**Entregable Documento de diseño Proyecto 3 DPOO**

**Diseño Empresa Alquiler Carros**

**Estudiante:** Camilo Daza **Código:** 201416461

**Estudiante:** Mateo Cante **Código:** 202223518

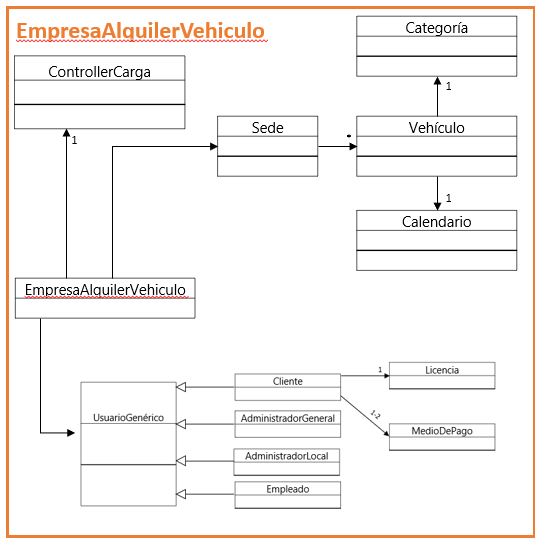
**Estudiante:** Juan Nicolás Suarez **Código:** 202222678

**Entregable 3, diseño actualizado + Interfaz**

En la sección de Anexos se puede encontrar el primer y segundo acercamiento a este documento de diseño, en dónde se establecieron las reglas y distintas iteraciones para la generación de la aplicación basada en consola y en un segundo momento basada en interfaz, pero sin los cambios solicitados en la tercera entrega. Para esta segunda versión del documento, se hicieron algunos ajustes en la lógica implementada para lograr los requerimientos adicionales solicitados. En esta tercera iteración los cambios agregados fueron bastante menores a iteraciones anteriores, los nuevos requerimientos fueron agregados de manera más sencilla y las clases nuevas no fueron tantas.

1. **Diseño Actualizado**
2. **Componentes y estereotipos finales**

* **Estereotipos de la lógica**

****

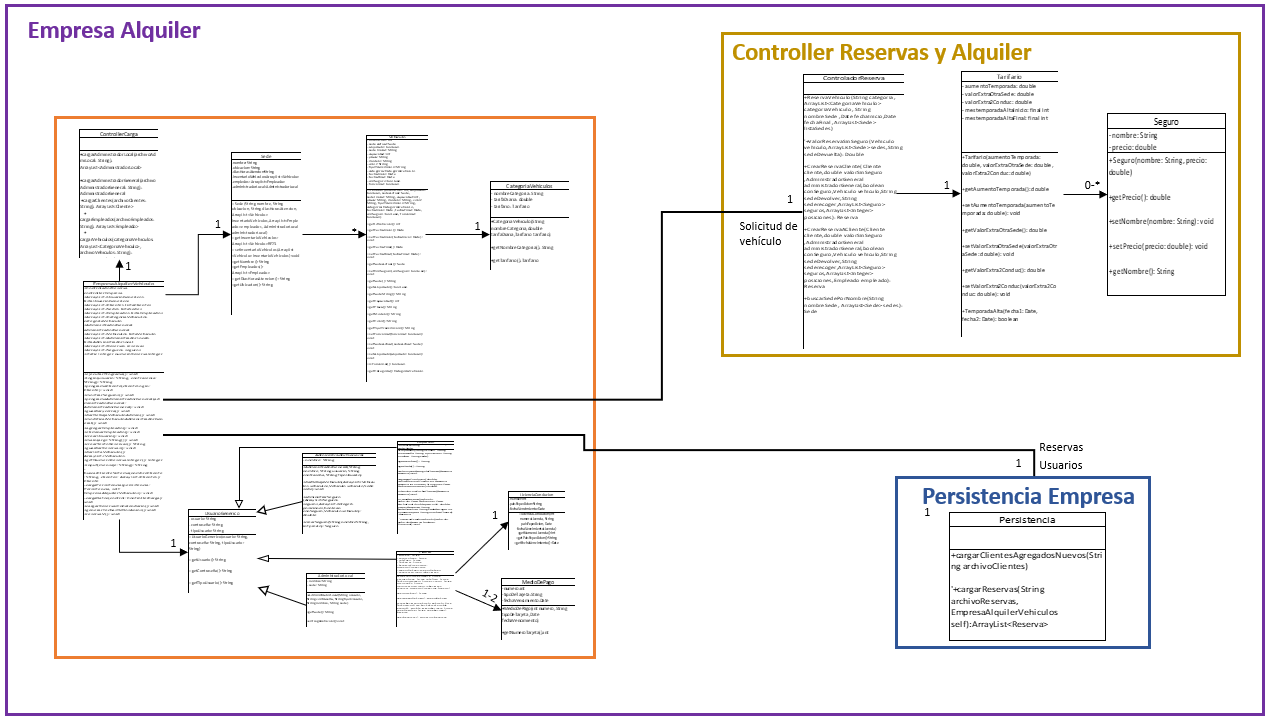
Para controllerReservas, alquiler y Persistencia se mantuvo la misma estructura a alto nivel.

**A diagram of a computer

Description automatically generated**

**A close-up of a document

Description automatically generated**

****

En las anteriores imágenes se documentan los estereotipos finales de la lógica alto nivel, cómo se puede observar, estos varían mínimamente con relación a los estereotipos definidos en una primera fase del proyecto. Lo anterior, dado que esta estructura estuvo muy bien definida, los cambios se presentaron principalmente en los cambios realizados al medio de pago, requerimiento nuevo que se solicitó para esta tercera entrega del proyecto. El resto de los estereotipos fueron correctamente definidos, lo que hizo que no fueran necesarios cambios muy grandes en la lógica.

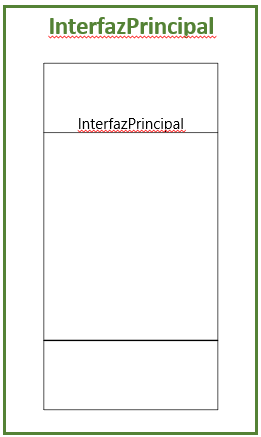
De igual forma, aunque en la segunda entrega se realizaron una serie de cambios importantes en cuanto a la interfaz y la eliminación de múltiples métodos de consola. Esta vez no fueron necesarios muchos cambios adicionales. De igual manera, la correcta implementación de la interfaz

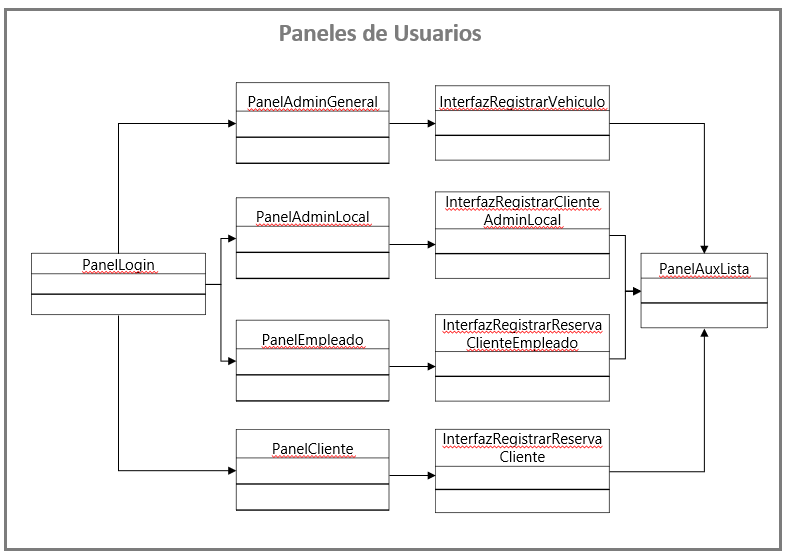
Aun así, para esta oportunidad se realizaron algunos cambios dentro de los métodos clave que se conectaban a consola. Los cambios que se listan a continuación, enfocados principalmente en la creación de un código más compacto y que pasa de recibir inputs de usuario por consola a opciones de interfaz:

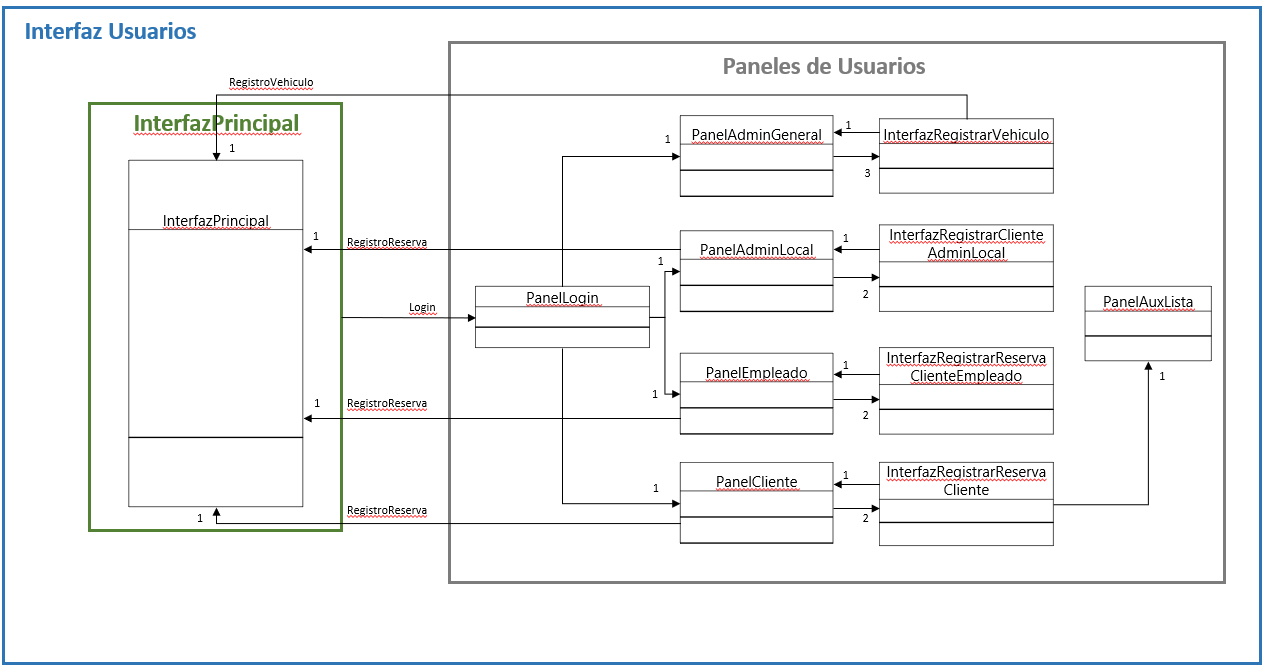
**Cambios:**

* Se ajustó método cargarReservas() para solucionar error en el cargue en la entrega anterior.
* Se ajustaron los métodos programaEmpleado(), dado que ya no se necesita solicitar inputs por consola a clientes
* **Estereotipos de la interfaz**

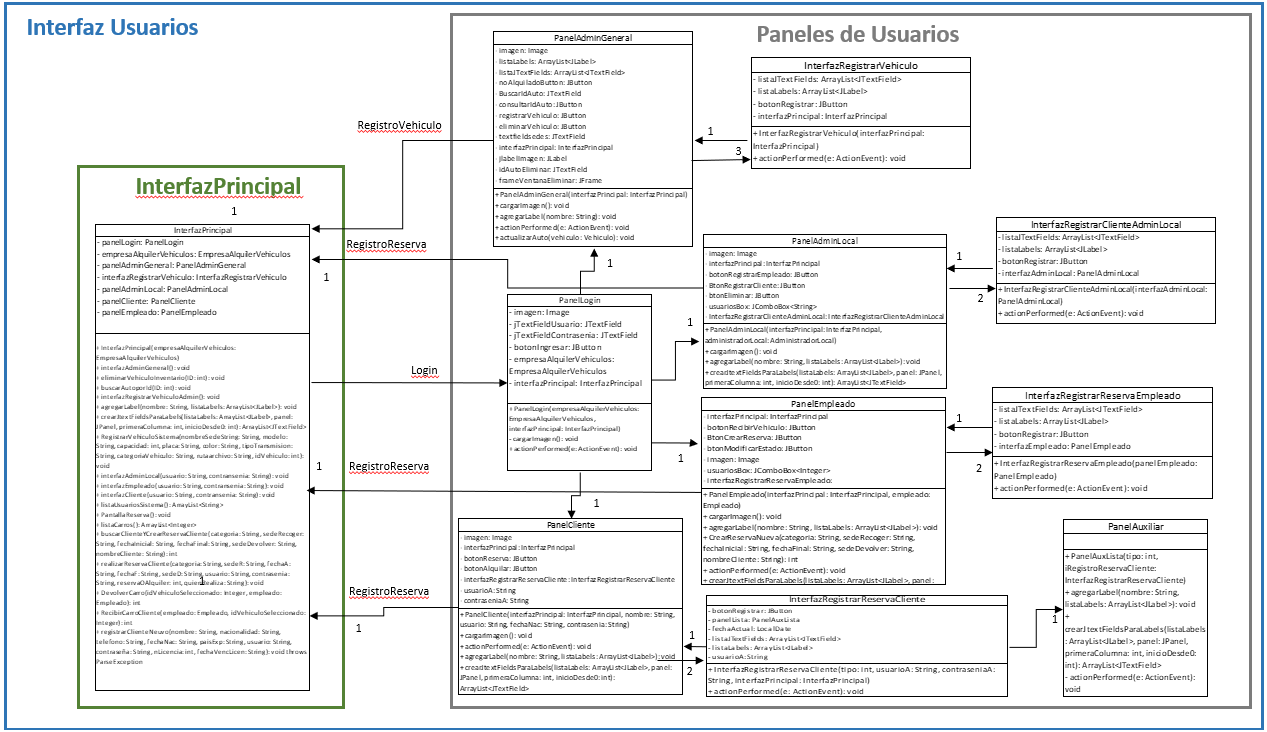
Ahora, para desarrollar la interfaz también se generaron dos estereotipos que permitieron un correcto desarrollo de nuestro aplicativo. Estos estereotipos se presentan a continuación:







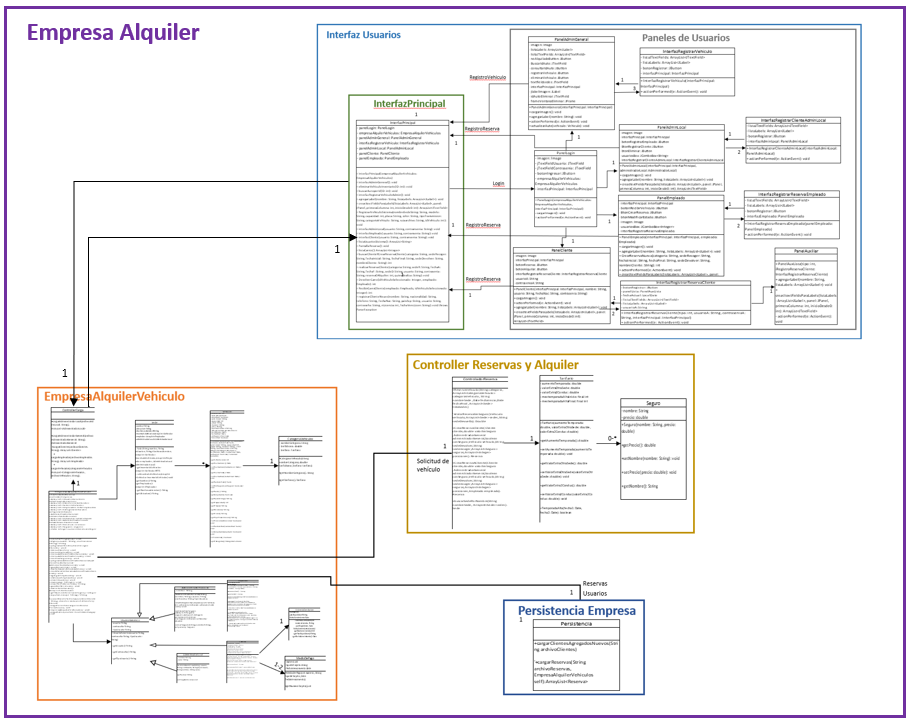
Con los métodos definidos, el diagrama anterior de diseño completo, solo para la parte de interfaz, se definió de la siguiente manera:

****

Para mejor visualización, revisar el powerpoint “Diagrama de Clases.pptx”

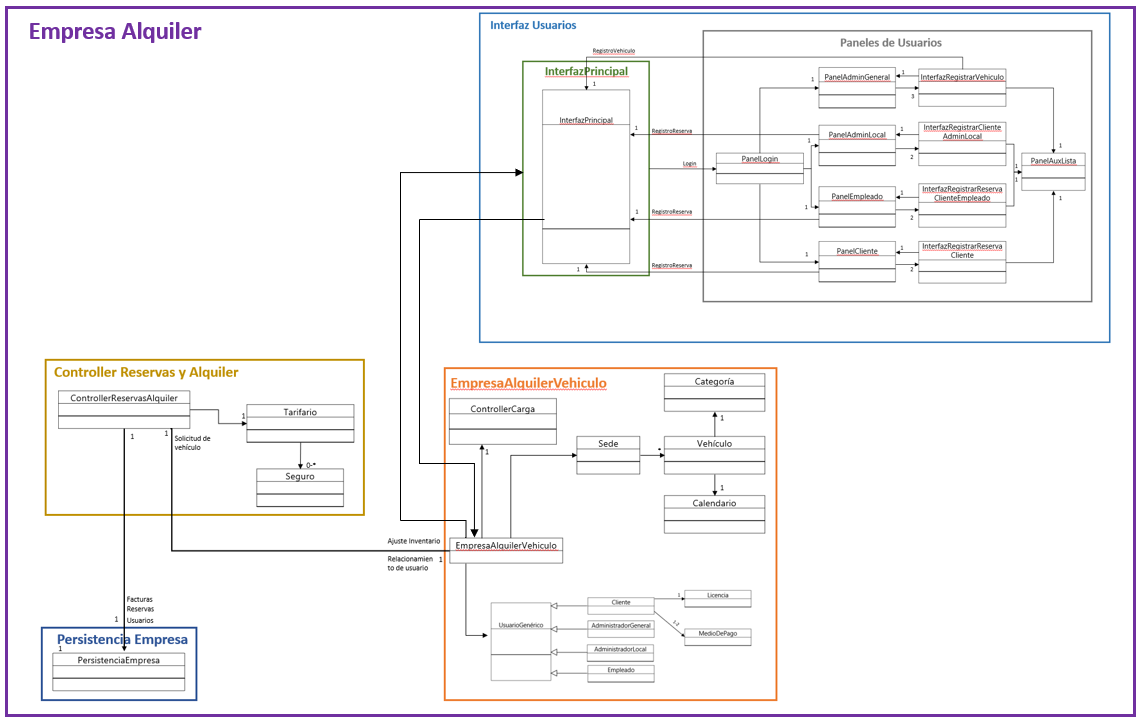
**2. Diagramas de clase**

Teniendo esto en cuenta los estereotipos, las responsabilidades y colaboraciones, a continuación, se presenta un diagrama de clases final, detallado con las relaciones, atributos y métodos.

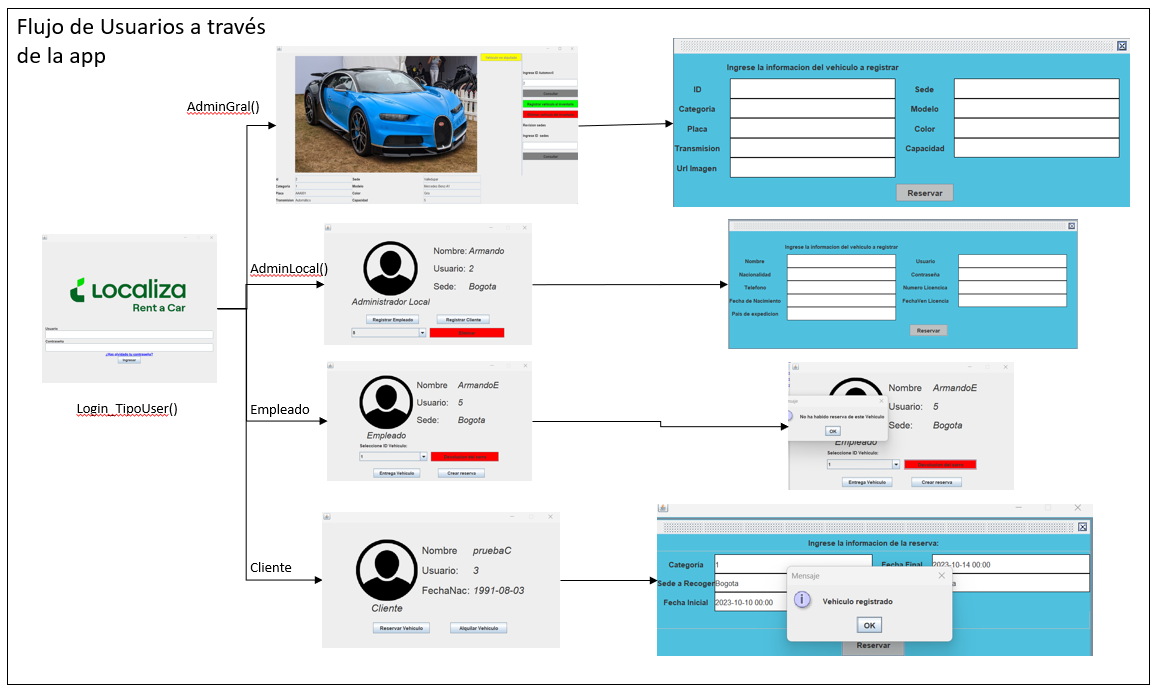
****

Para mejor visualización, revisar Excel adjunto “Diagrama de Clases.xlsx” o el powerpoint “Diagrama de Clases.pptx”

Para mejor entendimiento, también se presenta el diagrama **a alto nivel** para entendimiento de las relaciones entre clases, incluyendo tanto la lógica como la interfaz



1. **Diagrama de Flujo de Usuarios y acciones en la aplicación**

****

**A black car with blue lights

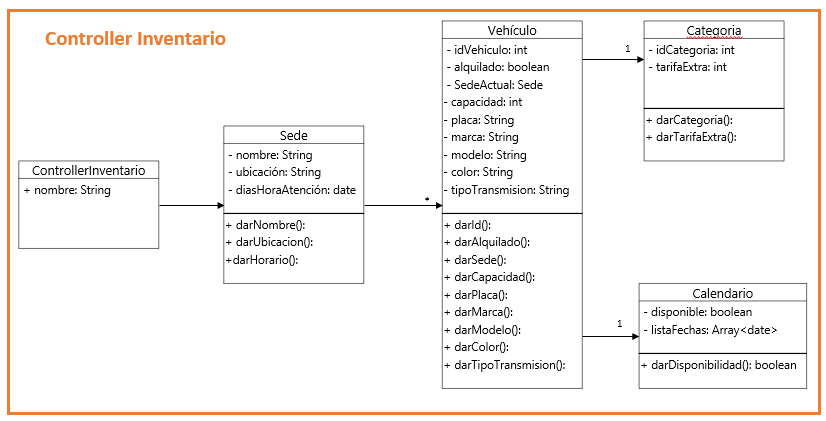
Description automatically generated**

**Anexos**

**Iteración Número 1**

**1. Componentes candidatos y estereotipos**

1. Al reconocer la necesidad de llevar un inventario de los vehículos, el estado de estos y las características, se hace evidente que se requiere un componente específicamente encargue de tal responsabilidad dada su magnitud. Dicho componente, haciendo referencia al estereotipo que lo justifica, se denomina “Controlador Inventario Vehículos”.



1. A razón de que se debe controlar las reservas y el alquiler de los automóviles, con todo lo que esto implica, es necesario integrar una instancia que se encargue del control de estas necesidades. Para esto, se asignará un componente con el estereotipo de controlador asociado a las reservas, llamado “Controlador Reservas y Alquiler”

A screenshot of a computer

Description automatically generated

1. Respecto al manejo, actualización y registro de usuarios, se requiere otro componente al que le son asignadas estas responsabilidades. La “plataforma de usuario” es quien provee la información del estado actual, adopta el estereotipo de information Holder.

A diagram of a computer

Description automatically generated with medium confidence

1. Puesto que el programa debe mantener el estado actual de la empresa de alquiler y debe permitir cargar el estado inicial del mismo, se debe adicionar un componente que se ocupe de esta labor. El componente encargado que guarde y recupere la información del estado actual adopta el estereotipo de information Holder, con nombre “Persistencia Empresa”.

A screenshot of a computer

Description automatically generated

**2. Responsabilidades**



**3. Colaboraciones**

Ahora bien, entre las responsabilidades definidas se ha considerado también la colaboración entre los distintos componentes a fin de satisfacerlas. Lo anterior se caracteriza a continuación:

* **Reservar un automóvil:** Implica que el controlador de reservas y alquiler al recibir la instrucción por parte de la interfaz:

1. Le indica a la plataforma usuarios el usuario asociado a la reserva para que los relacione
2. Le pide a plataforma usuarios, datos adicionales del usuario para asignar a la reserva
3. Le pregunta al inventario de vehículos si hay vehículo de la categoría escogida en las fechas escogidas y en la sede escogida
4. Si la respuesta es positiva sobre la disponibilidad del vehículo, le indica al inventario de vehículos la sede dónde se devolverá el vehículo.
5. Si la respuesta es positiva sobre la disponibilidad del vehículo, le indica al inventario de vehículos la fecha en la que empieza la reserva y también la fecha en la que se devolverá el vehículo
6. Le indica a persistencia empresa que guarde el nuevo estado del vehículo reservado y la información del cliente asociado.

* **Retirar un automóvil en sede:** Implica que el controlador de reservas y alquiler al recibir la instrucción por parte de la interfaz (Usuario en sede):

1. Le dice a plataforma usuario que registra a nuevo usuario si aún no está registrado en plataforma para alquilar/retirar vehículo
2. Le pregunta al inventario de vehículos si hay vehículo de la categoría escogida en las fechas escogidas y en la sede escogida o de una categoría mejor
3. Si la respuesta es positiva sobre la disponibilidad del vehículo, le confirma a inventario de vehículos el retiro del vehículo de la sede
4. Le indica a persistencia empresa que guarde el nuevo estado del vehículo retirado y la información del cliente asociado.

* **Gestión de reserva:** Implica que el controlador de reservas y alquiler al recibir la instrucción por parte de la interfaz:

1. Le pregunta al inventario de vehículos si hay disponibilidad con los cambios hechos por el usuario.
2. Si la respuesta es positiva sobre la disponibilidad, le indica al inventario de vehículos los datos de la reserva para que ajuste su inventario y disponibilidad en estas fechas
3. Le indica a persistencia empresa que actualice y guarde el nuevo estado del vehículo reservado.

* **Cobro de tarifa:** Implica que el controlador de reservas y alquiler al recibir la instrucción por parte de la interfaz:

1. Le pregunte a plataforma usuario los datos de facturación del cliente
2. Le indica a persistencia empresa que guarde la factura generada y que la asocie al cliente en cuestión

El resultado final de la iteración Número 1 sería del tipo:

A diagram of a machine

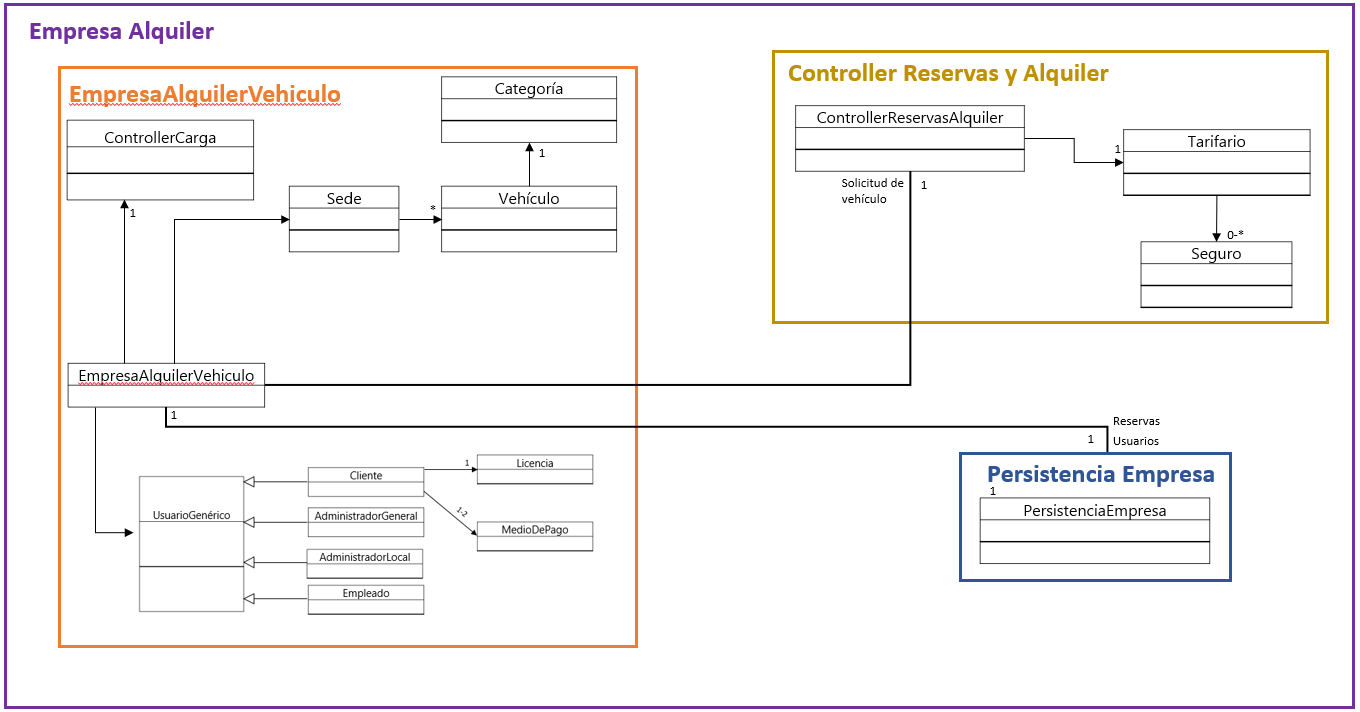
Description automatically generated with medium confidence

**Iteración Número 2**

Dado que Plataforma Usuario es solo una manera de interactuar con reserva y de conocer acerca de los clientes, un mismo controller puede dirigir tanto inventario como plataforma usuario, esta información se agrupa en una sola clase y esto facilitará el manejo de la información de cada una, y el hecho de poder guardar listas de información en conjunto.

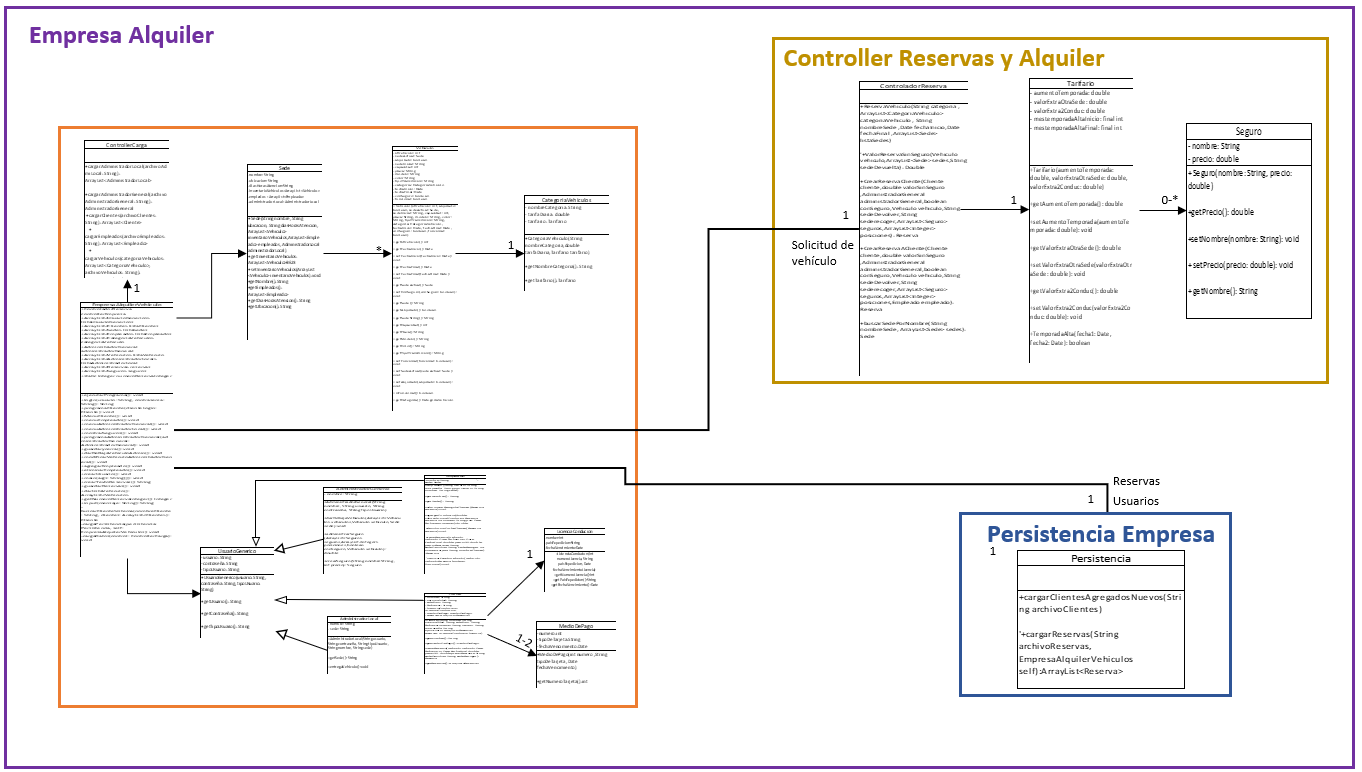
De esta manera, esta clase incluirá la información de los usuarios y del inventario en conjunto, esto facilita también la recopilación de la información para generar la persistencia, por lo que la persistencia se llevará a cabo desde esta clase también.

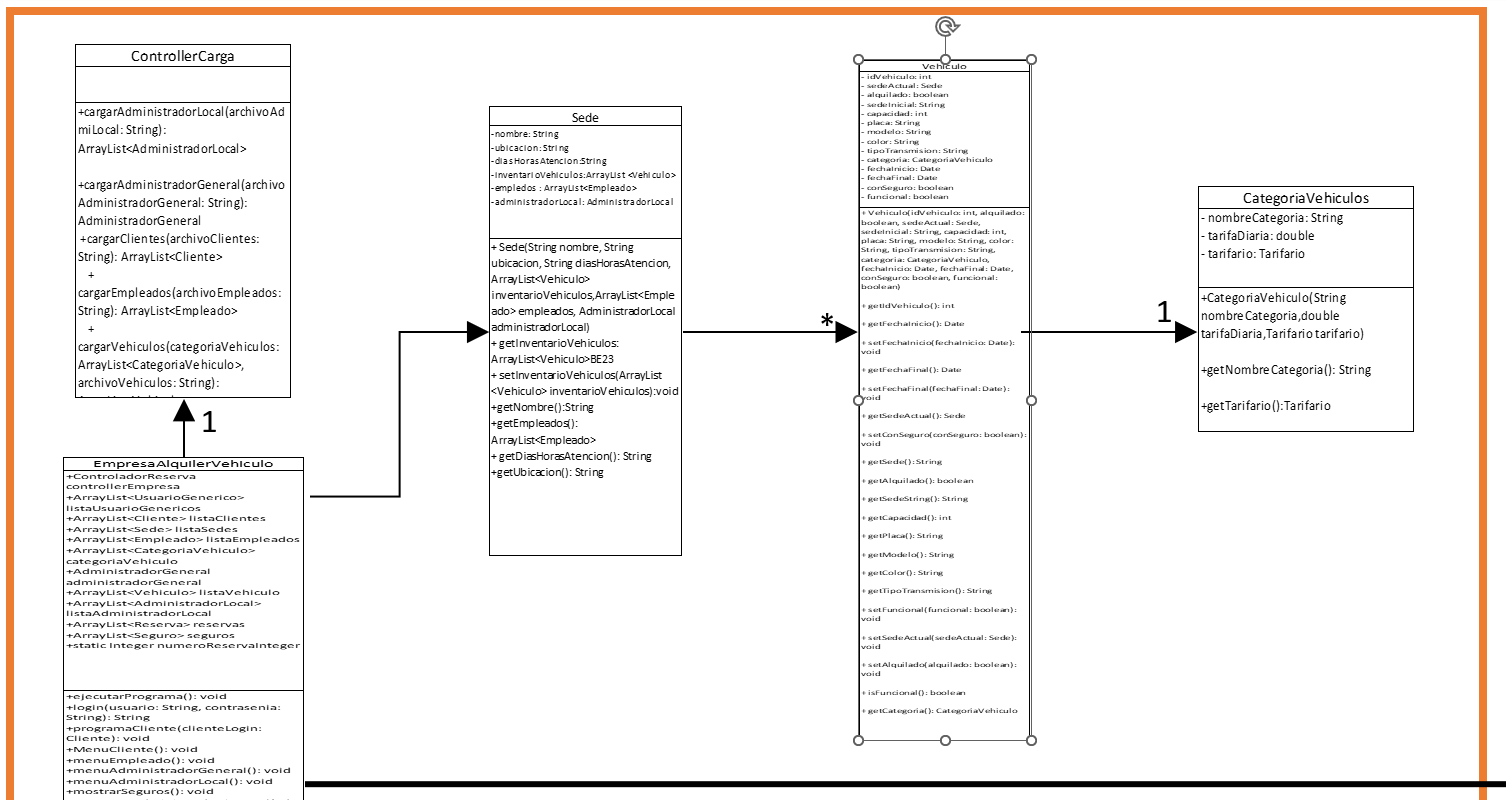
El resultado final de la iteración Número 2 sería del tipo:



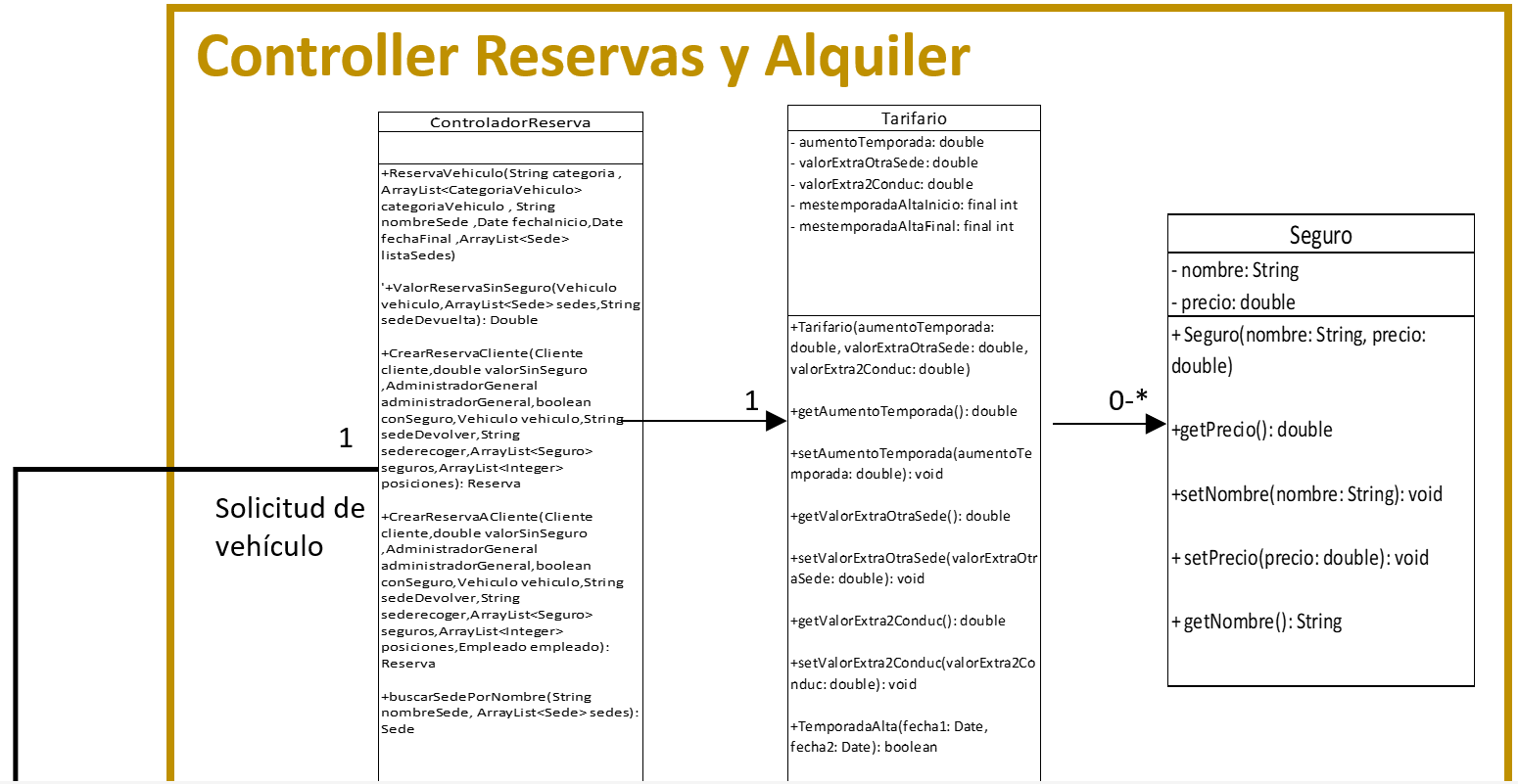
**4. Diagramas de clase**

Teniendo esto en cuenta los estereotipos, las responsabilidades y colaboraciones, a continuación, se presenta un diagrama de clases final, detallado con las relaciones, atributos y métodos.

****

 A diagram of text and information

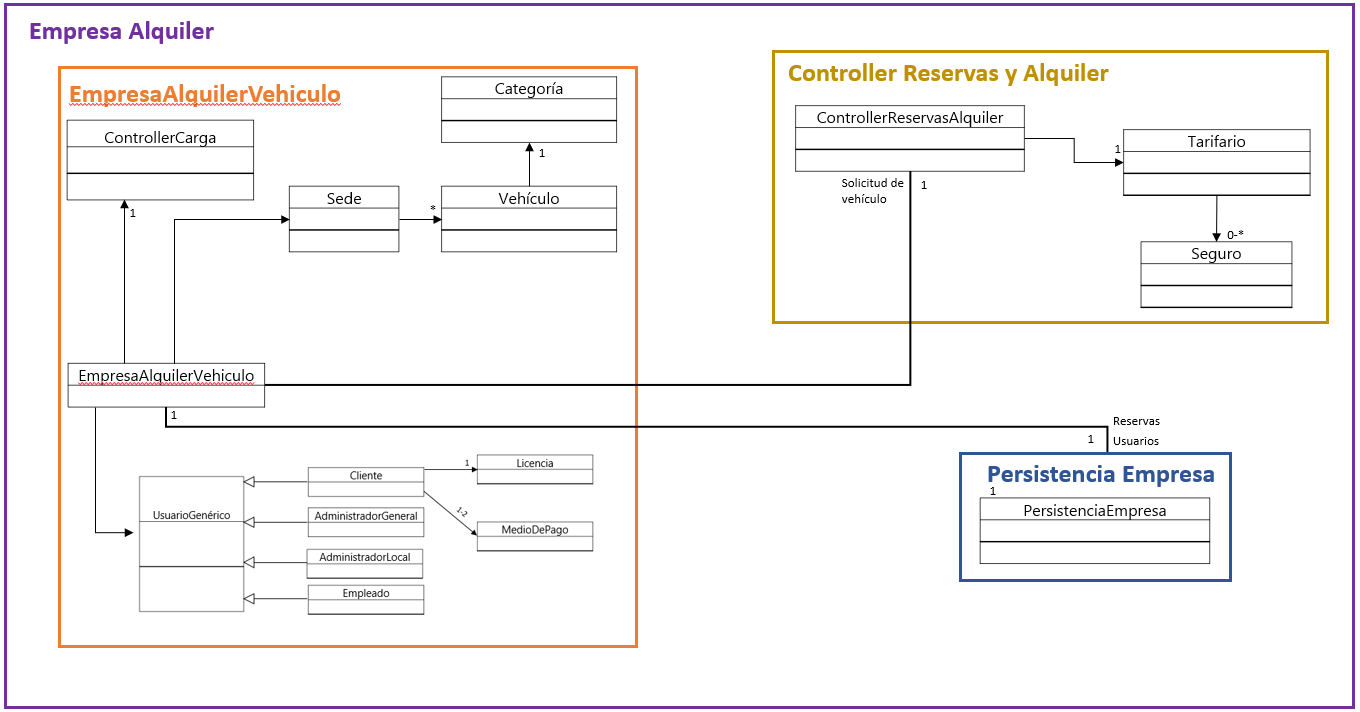
Description automatically generated with medium confidence

 A screenshot of a computer

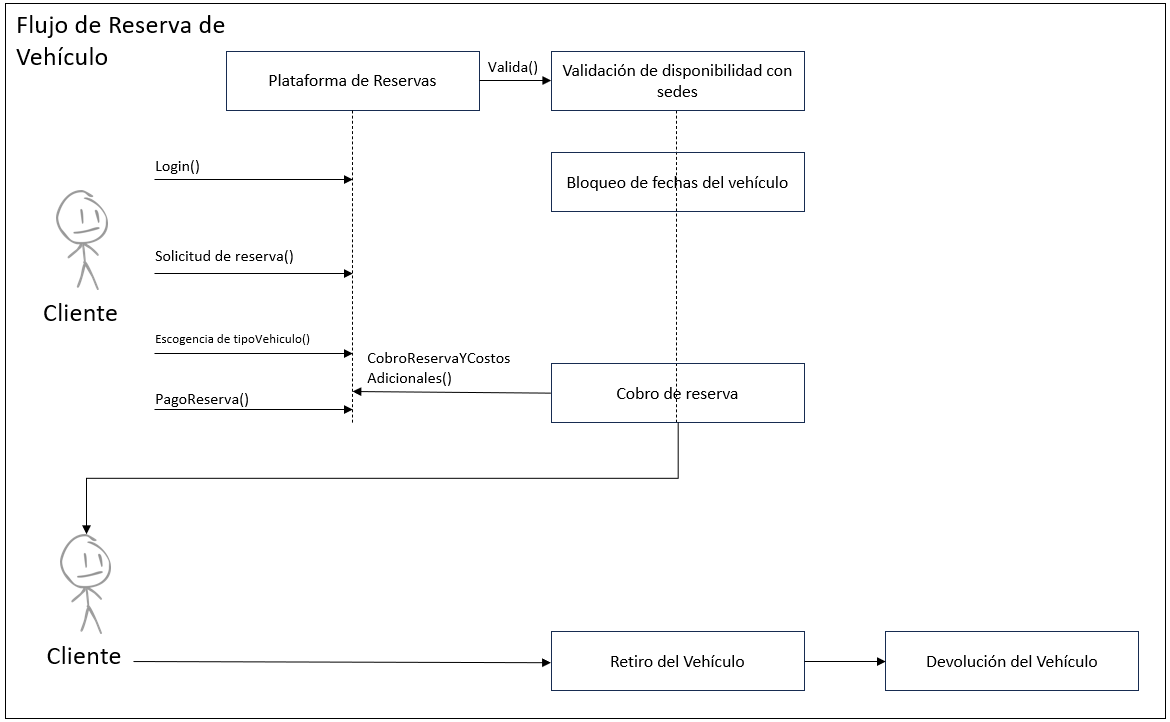
Description automatically generated

Para mejor visualización, revisar Excel adjunto “Diagrama de Clases.xlsx” o el powerpoint “Diagrama de Clases.pptx”

Para mejor entendimiento, también se presenta el diagrama **a alto nivel** para entendimiento de las relaciones entre clases



1. **Diagrama de Flujo de Reserva de Vehículo**

****