

# Diseño y Programación Orientada a Objetos

## Proyecto 3

Arévalo M. Anderson<sup>1</sup>, Ricaurte P. Natalia<sup>2</sup>, Alfonso Andrés F.<sup>3</sup>

Departamento de Ingeniería de Sistemas y Computación, Facultad de Ingeniería

Universidad de los Andes,

Bogotá, Colombia

### 1. Contexto problema

En el sistema de alquiler de vehículos, se busca desarrollar un sistema enfocado en el alquiler de vehículos con 3 módulos: inventario por sede, gestión del alquiler vehicular desde que se renta hasta su entrega y el manejo de información de la compañía. Los usuarios del sistema corresponden a clientes, empleados y administrador. Es de esta forma que, para satisfacer la interacción de la interfaz se tendrá en cuenta las funcionalidades nivel alto del sistema de alquiler de vehículos.

Esto es esencial con el fin de tener un sistema que gestione efectivamente el inventario de vehículos y las distintas interacciones con los usuarios, cabe resaltar que las distintas relaciones que más se resaltan son usuario con interfaz, sin embargo, hay que tener en cuenta que los usuarios se encuentran en dos categorías, cliente y empleados (empleado o administrador).

Esta relación se establece con el fin de poder reservar, alquilar, gestionar datos administrativos y otras actividades que se realizan con la utilización del sistema. Para entender de manera más dinámica y global el contexto del problema se puede observar la ilustración 1.

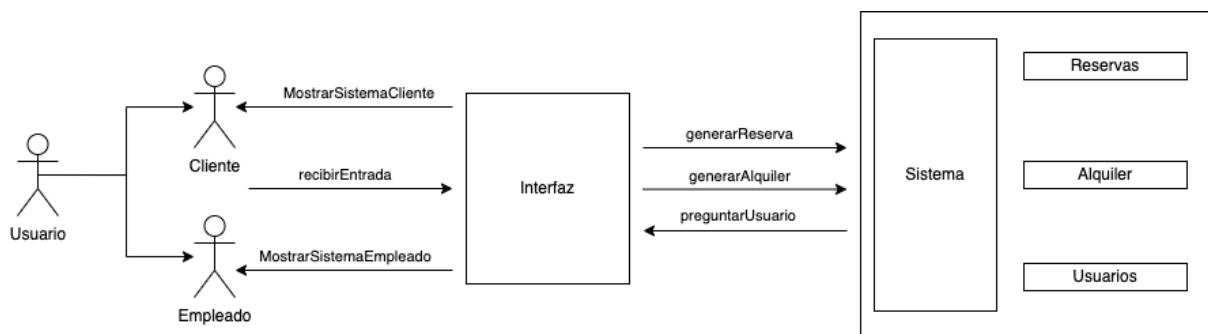


Ilustración 1. Digrama contexto

### 2. Nivel 1

#### 2.1. Componentes candidatos y estereotipos

Al tener claridad sobre el contexto del problema y como el sistema funciona, se pudieron evidenciar 4 componentes candidatos con sus estereotipos correspondientes, tal y como se evidencia en la ilustración 2.

- Se evidenció la función de un componente que se encarga de gestionar el manejo principal de sistema el cual es fundamental para la aplicación del contexto del problema, y la interacción con los demás componentes siendo este el ciclo de vida del sistema. Este componente, se encuentra relacionado con el estereotipo “Controlador del Sistema”.
- Por otro lado, se pudo observar un componente encargado de gestionar el proceso de reservas de vehículo el cual se encargue de realizar el proceso de registro e información

pertinente relacionada a esta. Este componente, está relacionado con el estereotipo “Controlador de reservas” e *Information holder*.

- Asimismo, se encontró otro componente que debe estar enfocado en gestionar el estado en el que se encuentran los vehículos, identificando si este está disponible, en qué lugar de sede se encuentra, su estado de limpieza y otras características que lo conforman. Este componente, se encuentra relacionado con el estereotipo “Controlador Estado Vehículo”. Sin embargo, como este es el que provee la información del estado, se adopta también como estereotipo *Information holder*.
- Finalmente, se puede evidenciar la necesidad de un componente controlador de gestionar la entrada y salida de cada vehículo que es reservado teniendo en cuenta las sedes de cada uno de estos. Esto teniendo en cuenta, que contiene el proceso de alquiler donde se registran las compras de seguros, se establecen los precios finales y se tiene en cuenta la información del actor que reservo el vehículo para el proceso que lo compone. Teniendo esto claro, se puede evidenciar que el estereotipo también se denomina como *Information holder*. Siendo este un “Controlador Entrega vehículo”

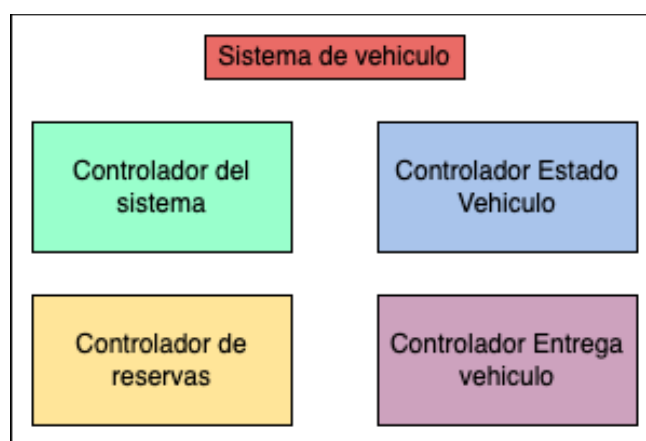


Ilustración 2. Componentes y estereotipos

## 2.2. Responsabilidades

Teniendo en cuenta, los componentes candidatos se pueden identificar ciertas responsabilidades que estos tendrán a cargo. Tal y como se evidencia en la tabla 1.

No	Responsabilidad	Componente
1	Gestionar la verificación de usuarios	Controlador Sistema
2	Generar números de reserva y alquiler	
3	Crear nuevo usuario	
4	Acceso de información a vehículos y sedes	
5	Registro de reserva	Controlador Reservas
6	Información de la reserva	
7	Estimación de precio de la reserva	
8	Actualización estado de los vehículos	Controlador Estado Vehículo
9	Notificar y confirmar el cambio del estado	
10	Solicitar entrega de vehículo	
11	Consultar reserva	
12	Consultar disponibilidad del vehículo	

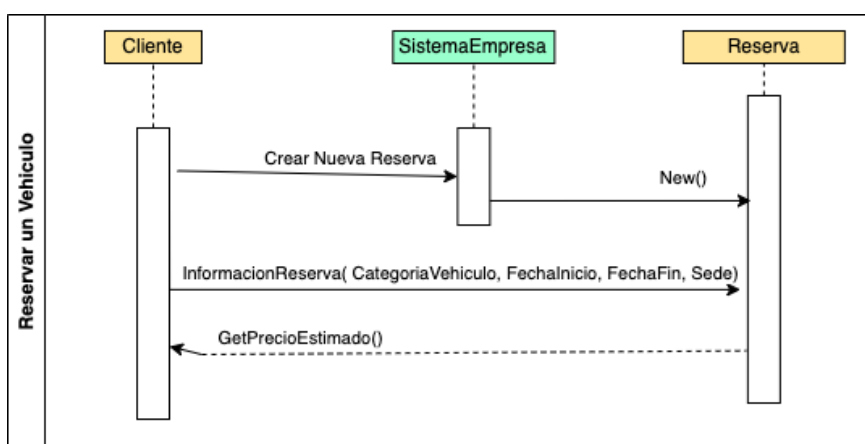
13	Consultar información seguros	Controlador Entrega Vehículo
14	Registrar seguros y conductores adicionales	
15	Estipular precio final de alquiler	
16	Generar factura de cobro	

*Tabla 1. Responsabilidades.*

## 2.3. Colaboraciones

### 2.3.1. Colaboraciones al realizar reserva:

El controlador de reservas y controlador del sistema son los encargados de esta ejecución. En esta se puede analizar como la interacción esta dada por el cliente que es el actor encargado de las reservas, dado que este crea la solicitud, este genera una información pertinente a esta reserva y a partir de este el sistema atrae la solicitud, realiza el procesamiento respectivo, enviando al cliente la confirmación y estimación de precio respectivo.



*Ilustración 3. Colaboraciones al realizar una reserva*

### 2.3.2. Colaboración entrega de vehículo:

En esta colaboración se puede observar distintas relaciones en las cuales están el controlador del sistema, controlador entrega de vehículo y controlador de reserva. En esta se puede evidenciar:

1. Inicia el cliente con interacción con el empleado solicitando entrega vehículo (encargado de realizar proceso de entrega)
2. Información de la reserva si posee.
3. El empleado realiza el procesamiento de la solicitud con preferencia de cliente.
4. Ofrece distintos seguros.
5. Verifica disponibilidad vehículo con el sistema.
6. El sistema notifica la disponibilidad.
7. Empleado registra conductores y seguros.
8. Sistema genera el precio final.
9. Empleado realiza cobro y entregar vehículo.

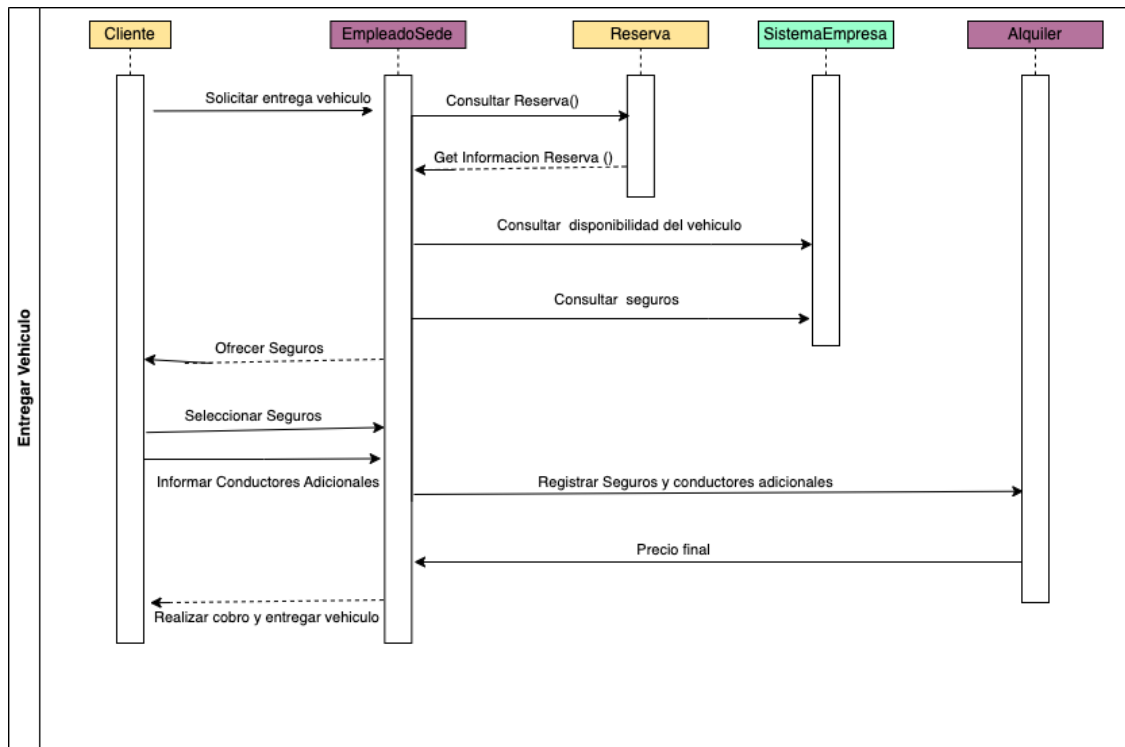


Ilustración 4. Colaboraciones al realizar una entrega de vehículo

Para continuar con este proceso, se encuentra el controlador de entrega de vehículo y controlador de sistema con el fin de realizar la devolución del vehículo por parte del cliente:

1. Entrega el vehículo en la sede
2. Empleado verifica estado del vehículo y el alquiler correspondiente.

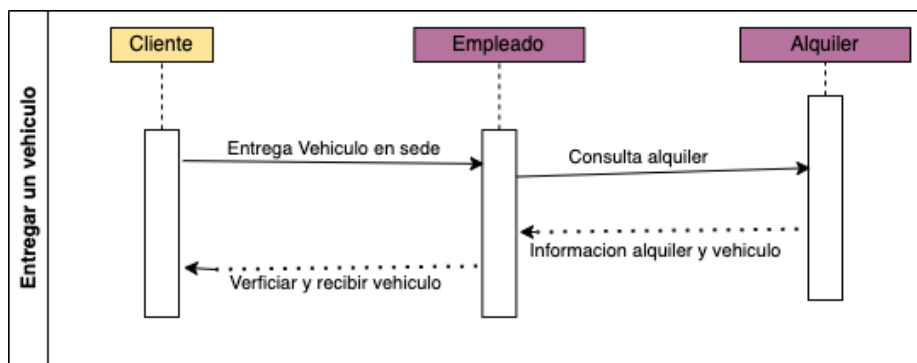


Ilustración 5. Colaboraciones al realizar entrega de un vehículo

### 2.3.3. Actualizar estado vehículo:

Para este escenario la colaboración del controlador estado vehículo, controlador empresa y controlador entrega vehículo son las que hacen parte del funcionamiento de esta:

1. Empleado realiza la actualización estado del vehículo en el sistema.
2. El sistema modifica la disponibilidad en la caracterización del vehículo
3. Confirma el cambio de estado

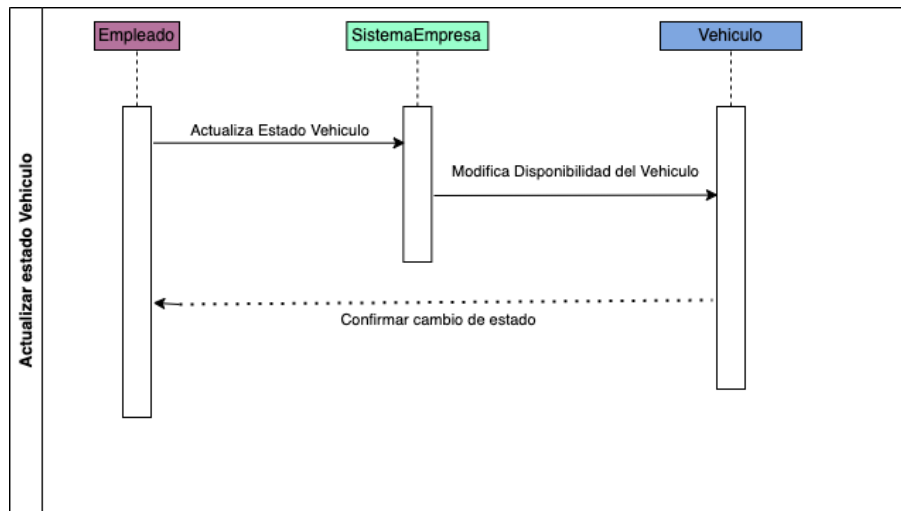


Ilustración 6. Colaboraciones al actualizar estado

Teniendo en cuenta estas distintas colaboraciones se puede observar que las relaciones entre los 4 componentes son:

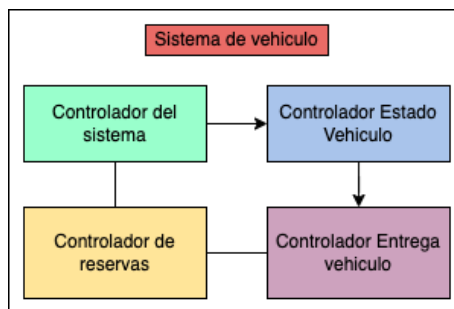


Ilustración 7. relaciones

### 3. Nivel 2

Siguiendo con el proceso de diseño por niveles, se procede a descomponer cada uno de los componentes: Controlador del Sistema, Controlador Reservas, Controlador Estado Vehículo, Controlador Entrega Vehículo.

#### 3.1. Controlador del Sistema

Es necesario recordar las responsabilidades asociadas al componente:

- Gestionar la verificación de usuarios
- Generar números de reserva y alquiler
- Crear nuevo usuario
- Acceso de información a vehículos y sedes

Teniendo en cuenta que a partir de estas responsabilidades se definen los aspectos claves e iniciales del funcionamiento del sistema en general, en el sentido en el que precisan la información de los diferentes procesos a realizar, estas deben estar organizadas, además de muy relacionadas, en una clase. En este orden de ideas, este punto debe permitir la recepción y posterior guardado de información para poder iniciar los respectivos procesos de la empresa. (Ver Ilustración 8).

##### 3.1.1. Componentes candidatos y estereotipos

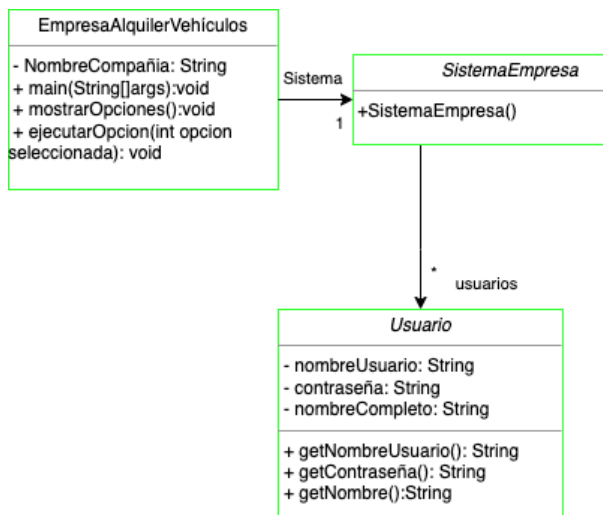


Ilustración 8. componentes sistema controlador

### 3.1.2. Responsabilidades y Colaboraciones

Este componente, cuenta con las colaboraciones de las clases EmpresaAlquiler, SistemaEmpresa, Usuario. En este caso EmpresaAlquilerVehiculo se encarga de la lógica del sistema de realizar los distintos registros de información igual que mostrar la información correspondiente mientras que el Sistema empresa, enseña como se vera y como se ejecutaran las distintas opciones teniendo en cuenta las filtraciones por usuario. En este caso hay que tener en cuenta que la clase Usuarios es la que permite el acceso a las distintas opciones de los roles, al igual que el registro de estos.

### 3.2. Controlador Reservas

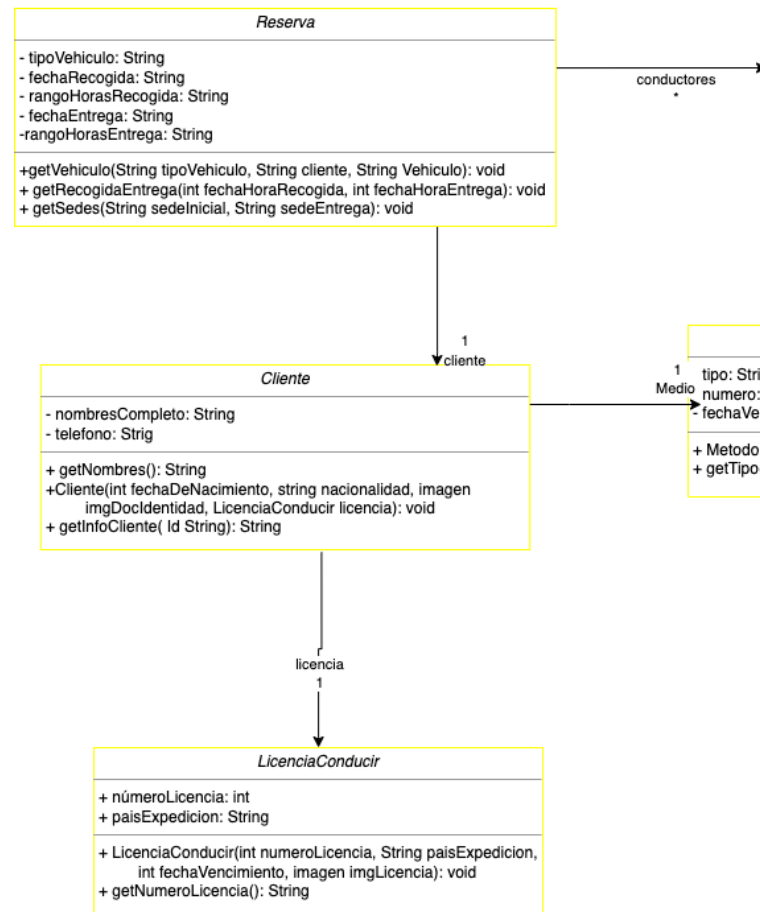
Nuevamente se considera apropiado reiterar las responsabilidades que el componente debe atender:

- Registro de reserva
- Información de la reserva
- Estimación de precio de la reserva

#### 3.2.1. Componentes candidatos y estereotipos

A partir de estas responsabilidades, que dentro del proceso de la empresa son los pasos iniciales para desarrollar las actividades del portafolio de esta, se deben

separar en diferentes componentes para manejar de forma adecuada la información suministrada por los usuarios en dos clases diferentes: Reservas, Cliente, ConductorAdicional, MedioPago, LicenciaConductor.



### 3.2.2. Responsabilidades

En este sentido, el controlador de las reservas será el encargado de registrar las reservas, estimar un precio de acuerdo con las características de esta y en caso dado de ser necesario brindar la información de sí. De igual forma, es necesaria la información de los clientes; las responsabilidades descritas a continuación se manejarán en las clases mencionadas anteriormente.

Inicialmente se tiene la información completa del cliente, destacando su nombre y teléfono de contacto, por lo cual ya se puede reconocer como *cliente* puntual de la empresa; esto facilita la flexibilidad a la hora de desarrollar diferentes procesos para

encontrar información de las reservas por cliente. En este paso se revisa también la licencia del conductor, la cual se encuentra en una clase exclusiva.

Posteriormente el registro de las reservas requiere que el cliente ya establecido defina características como la hora y sede de entrega y recogida de los vehículos. En este punto se tiene muy presente que tanto las fechas como los lugares tienen un valor fundamental dentro del valor final de la reserva, puesto que, por ejemplo, en cuanto a las fechas, se genera un valor adicional al excederse de la fecha límite acordada. De igual forma sucede al cambiar la sede de entrega del vehículo, puesto que estos cuentan con unas sedes ya asignadas previamente. Tener en cuenta esto deja al código un poco más complejo al definir estas características.

En cuanto a la estimación del precio, inicialmente se tienen en cuenta la categoría del vehículo, el tiempo de reserva y la cantidad de conductores que harán uso del automóvil, con ello se obtiene un valor base que se puede ver incrementado dados atributos que pueden cambiar (los mencionados anteriormente).

Finalmente, al tener el valor total del pago, se define por el cliente el método de pago, esperando que en su mayoría se realice por tarjeta de crédito o débito, y teniendo en cuenta factores característicos de estas al revisar el número y fecha de vencimiento de los diferentes métodos de pago posibles; en caso de no ser realizados de este modo, simplemente se ignoran categorizándolos como “No Aplica” o “0”.

### 3.2.3. Colaboraciones

A pesar de ser solo una clase, en este caso las reservas cuentan con la colaboración de las clases Vehículo y Seguro, que les brinda la información referente a estos aspectos para tener en cuenta.

## 3.3. Controlador Estado Vehículo

De igual forma, en este componente también recordamos las responsabilidades a desarrollar:

- Actualización estado de los vehículos
- Confirmar la fecha de disponibilidad del vehículo
- Notificar y confirmar el cambio del estado

### 3.3.1. Componentes candidatos y estereotipos

En este orden de ideas, el controlador de los vehículos se encuentra organizado en dos diferentes clases: Categoría, EstadoVehículo y Vehículo, puesto que en ellas se definen las características del automóvil a rentar.

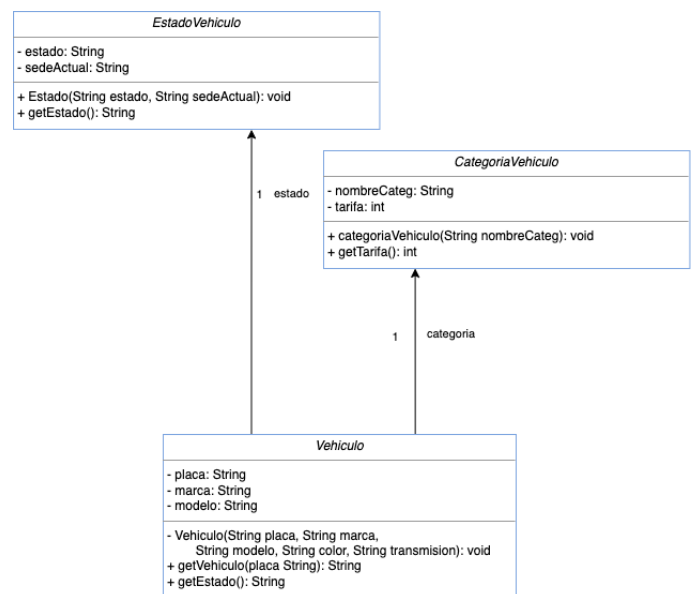


Ilustración 9. Componentes estado vehiculo

### 3.3.2. Responsabilidades

Así pues, teniendo en cuenta la relación directa entre clases, inicialmente se revisa Vehículo, puesto que predispone la información básica del automóvil, que a su vez funciona como referente en caso de ser necesario para obtener datos de este, tal como la placa, el modelo o la marca. En el desarrollo del proyecto se tiene muy presente la categoría del vehículo, pues esta define por una parte el precio final del alquiler debido a que cada categoría maneja una tarifa en su precio diferente, siendo mayor para las categorías más altas. Finalmente, se tiene en cuenta el Estado del Vehículo, en el sentido en que se revisa su disponibilidad cercana para su correspondiente alquiler. Al igual que la



CategoriaVehiculo quien define en que categoría y que tarifa correspondiente tiene cada vehículo.

### 3.3.3. Colaboraciones

El orden de las conexiones entre clases es muy importante al observar que permite detallar más fácil diferentes características que pueden ser utilizadas por otros controladores en otros momentos; sin embargo, su esencia se mantiene entre estos lazos al manejar entre todos la información del producto principal de la empresa.

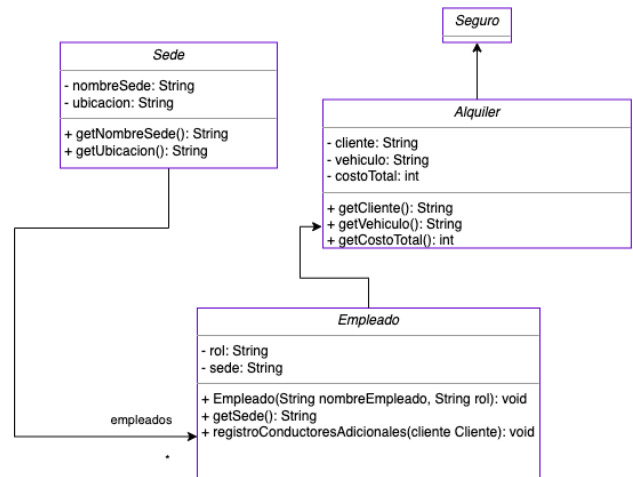


Ilustración 10. componentes controlador entrega vehiculo

### 3.4. Controlador Entrega Vehículo

Así mismo se deben reiterar las debidas responsabilidades de este componente:

- Solicitar entrega de vehículo
- Consultar reserva
- Consultar disponibilidad del vehículo
- Consultar información seguros
- Registrar seguros y conductores adicionales
- Estipular precio final de alquiler
- Generar factura de cobro

#### 3.4.1. Componentes candidatos y estereotipos

Así las cosas, este componente se encuentra organizado en 6 clases: Alquiler, Vehículo, Cliente, conductorAdicional, Sede, Seguro, que, si bien algunas ya se han utilizado en otros controladores, tienen información fundamental para llevar a cabo la entrega del vehículo.

### 3.4.2. Responsabilidades

Se revisa en primera instancia la información encontrada dentro de la Reserva, la cual esencialmente refiere al Cliente, al conductorAdicional, en caso de la presencia de este, y del Vehículo gracias a los métodos implementados en ella. Aquí se podrá revisar la reserva realizada por el cliente y se tendrá en cuenta la cantidad de personas a la cuál estará disponible el vehículo, esto al tener en cuenta los nombres y apellidos sobre los cuales quedo realizada, y eventualmente al automóvil del cual harán uso.

Con esta información, que se conecta a su vez con el Controlador Estado Vehículo, se identifica la Sede en la cual se hará la respectiva entrega (esta sede fue elegida por el cliente) y se ofrece la ubicación. Cabe aclarar que tener la Sede en una sola clase permite facilidad dentro del código para manejar las ubicaciones de entrega y recogida de los vehículos.

Por otro lado, se puede revisar a la vez que la Reserva el Alquiler, pues son dos procesos diferentes pero conectados, en el sentido que definen la entrega del vehículo, pues de igual forma al final tienen en cuenta algunos factores distintos pero relacionados. Por ejemplo, acá se tienen en cuenta los seguros con los que contará el alquiler, los cuales de igual forma representan un factor importante para el cliente al recibir el vehículo (ser entregado



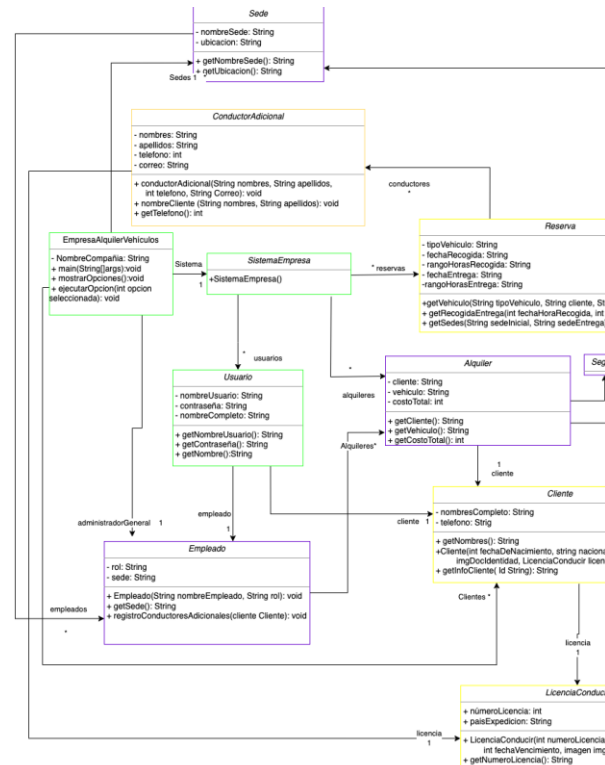
por la empresa). El manejo de esta información se lleva a cabo mediante un proceso manual realizado por los empleados de la empresa, que según su rol hacen diferentes tareas.

### 3.4.3. Colaboraciones

Si bien en una parte se apoya este componente en el controlador de Estado Vehículo, la mayor parte del proceso se realiza dentro de sí al tener relaciones bien establecidas y organizadas que manejan la información de forma fluida, beneficiando al código y su flexibilidad en caso de necesitar realizar algún cambio.

## 4. Diseño Final

En la ilustración



5. Ilustración 11. Diseño final nivel alto

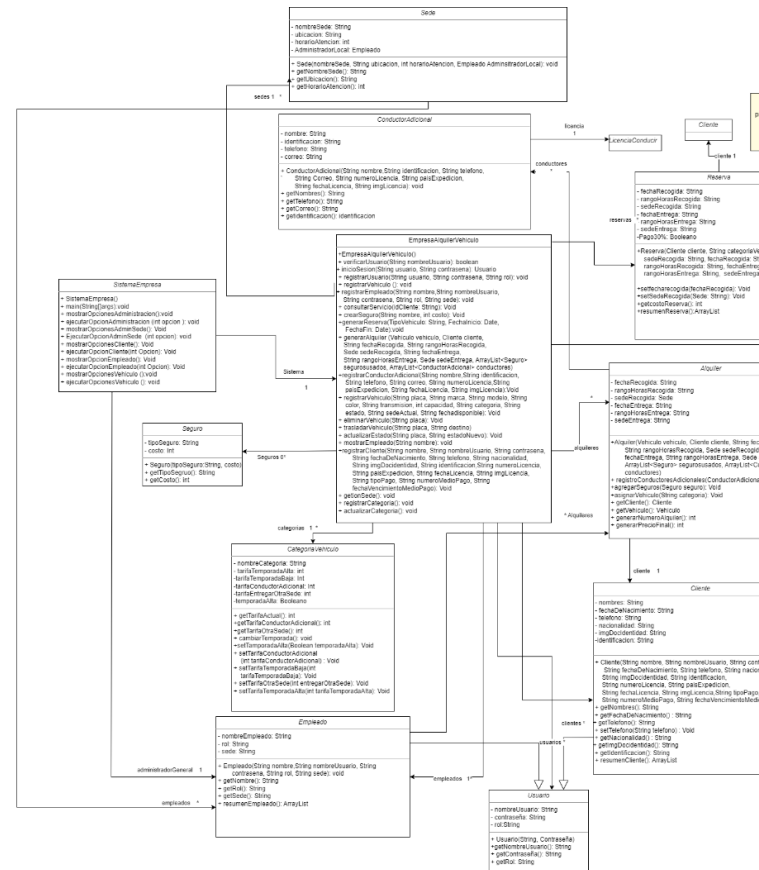


Ilustración 12. Diseño final nivel alto

## 6. MODIFICACIONES AL DISEÑO INICIAL

### 5.1 Emisión de facturas en pdf

El primer cambio que se realizó fue la emisión de facturas en PDF al momento de la entrega de un vehículo y del pago. Esto requirió:

- Crear una nueva clase "Facturar" que se encarga de generar el PDF con los datos de la factura.
- Utilizar la biblioteca iText para la generación del PDF. iText es una biblioteca estable, ampliamente utilizada y de código abierto para la generación de documentos PDF en Java.
- Permite crear fácilmente documentos PDF con formato desde cero. Tiene métodos para insertar texto, imágenes, tablas y otros elementos.

- Capturar la fecha/hora actual para incluirla en la factura. Para esto se utilizaron las clases `LocalDateTime` y `DateTimeFormatter` de Java.
- Leer la firma del administrador desde una imagen JPEG y agregarla al PDF. Se usó la clase `Image` de `iText`.

Manejar adecuadamente posibles excepciones tanto de la generación del PDF como de lectura/escritura de archivos. De esta manera, al integrar el código anterior con el proceso de pago y entrega de vehículos, se logra que se genere en PDF una factura estándar para cada transacción con los datos necesarios y la firma correspondiente. Esto da más formalidad y permite llevar un registro digital de las facturas. Lo anterior se implementa mediante interfaces que permite a los usuarios interactuar con el sistema de generar facturas PDF, la interfaz se muestra en la Fig. 13.

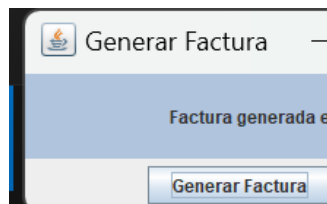


Ilustración 13. Interfaz generación PDFs

## 5.2 Características de los vehículos

Se agregan las características tanto a la clase `Vehículo` del modelo lógico como a las interfaces permitiendo a los usuarios incluir dichos parámetros en los nuevos registros de los vehículos. De igual forma el modelo se modifica para incluir las características de Tipo de Vehículo y el porcentaje de prima seguro. La Fig 14 muestra la modificación correspondiente en la interfaz para el ingreso de los parámetros dados.

Ilustración 14. Características adicionales a los vehículos

## 5.3 Pagos con Tarjeta de Crédito

El requerimiento de pagos con tarjeta de crédito se abordó definiendo una interfaz Pago que expone los métodos necesarios para procesar un pago y bloquear el cupo de la tarjeta. Esta interfaz es implementada por clases que simulan diferentes pasarelas de pago como PayPal, PayU o Sire. La lógica consiste en leer desde un archivo de configuración el nombre de las clases que implementan las pasarelas disponibles y cargarlas dinámicamente. Luego, en la interfaz gráfica se muestra un combo para que el usuario seleccione la pasarela deseada. Al momento de procesar el pago se captura la información de la tarjeta y del monto. Esta información se pasa a la pasarela seleccionada para intentar el pago y el bloqueo del cupo. Cada pasarela simulada registra las transacciones en un archivo de log con formato diferente (txt, json, etc). De esta manera se logra un acoplamiento flexible que permite agregar nuevas pasarelas con facilidad,

simplemente agregando la clase a el archivo de configuración. Esto cumple con el requerimiento de soportar múltiples opciones de pago de tarjeta de crédito de forma modular. La Fig. 15 demuestra las opciones para seleccionar los diferentes métodos de pago disponibles en la plataforma actualmente. Adicionalmente la fig. 16 demuestra el ingreso de los datos necesarios para realizar el pago correspondiente.

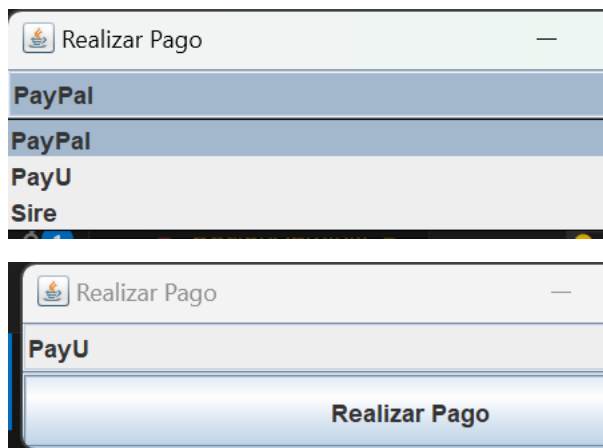


Ilustración 15. Selección de múltiples pasarelas de pago

Ilustración 16. Ingreso de los atributos necesarios para la pasarela de pagos

## 5.4 Aplicación para clientes

Para cumplir con el requerimiento de una aplicación independiente para clientes, se

desarrolló una interfaz gráfica compuesta por varias ventanas contenidas en el paquete "AplicacionClientes":

- La ventana principal "interfazUsuario" muestra las opciones iniciales de registro, consulta y reserva.
- La ventana "registroUsuario" captura los datos de nuevos clientes, almacenándolos en un archivo.
- La ventana "consultaDisponibilidad" permite consultar los vehículos disponibles dado una sede y rango de fechas. Muestra los resultados en la ventana "vehiculosDisponibles".
- La ventana "vehiculosDisponibles" contiene una lista de objetos "Vehiculo", encargados de modelar los datos básicos de cada vehículo disponible.
- La ventana "reservaVehiculo" genera la reserva de un vehículo en unas fechas, aplica el descuento del 10% y abre la ventana de pago.
- La ventana "pagoReserva" completa la transacción de pago por parte del cliente.

Todas estas ventanas modulares son mostradas desde una clase Main independiente al resto del sistema. De esta manera se cumple con el requisito de tener una aplicación reusable sólo para clientes, con su propia interfaz gráfica y lógica de negocio específica, funcionando sobre los mismos datos que la aplicación de empleados. Las Figuras 17, 18, 19 y 20 permiten visualizar el funcionamiento de la aplicación clientes y la incorporación de los requerimientos solicitados.

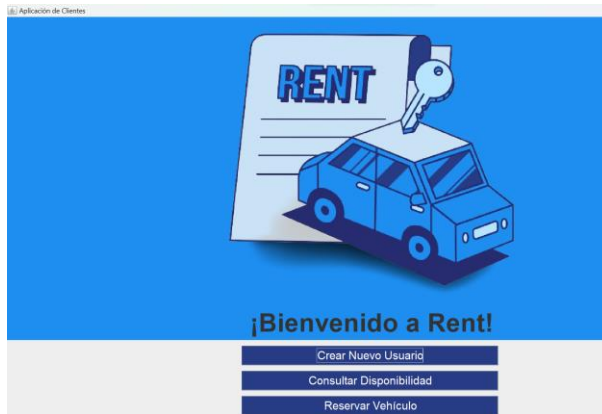


Ilustración 17. Ingreso de aplicación clientes

Ilustración 18. Registro de clientes en aplicación clientes

Ilustración 19. Consulta de reservas en aplicación clientes

Ilustración 20. Realizar reservas en aplicación cliente

Ilustración 21. Realizar pago de reservas en aplicación cliente

## 5.5 Pruebas automáticas

En este caso, para las pruebas automáticas, se completó la información que se guarda de forma automática en los documentos txt, verificando así que el proceso de guardado se llevara a cabo correctamente. A continuación, se mostrarán algunos ejemplos de estos procedimientos:

### 1. Creación de Seguro:

En la creación de seguro, para el menú inicial, se pide el tipo de seguro a implementar, con el cual nos guiaremos para definir un costo final a partir del propuesto inicialmente. En el ejemplo se puede observar un nombre y valor de seguro inventados para el ejemplo.

## 2. Agregar Vehículos

```

24 -----
25 null: ABC123
26 null: Audi
27 null: 2003
28 null: Gris
29 null: Manual
30 null: 5
31 null: Camioneta
32 null: Disponible
33 null: Norte
34 null: 11/12/2023
35 null: Camioneta 4x4
36 null: 10000
37 -----
38

```

Aquí se puede observar la creación del vehículo que se usará como ejemplo en este y los siguientes dos puntos, el cual es una Camioneta 4x4 de marca Audi, modelo 2003, con algunas características adicionales.

## 3. Trasladar Vehículos

En ese orden de ideas, el vehículo ejemplo se encontraba en la sede Norte, y será trasladado a la sede Norte.

## 4. Eliminar Vehículos

Finalmente, el vehículo por diferentes motivos será eliminado del sistema porque no será utilizado más por la empresa.

## 5. Información Administrador Local

Registrar Administrador Local

Nombre:

Pepe

Nombre de Usuario:

Pepe12

Contraseña:

abc123

Rol:

Administrador Sede Norte

Sede:

Sede Norte

Registrar

Regresar

25 null: Pepe

26 null: Pepe12

27 null: abc123

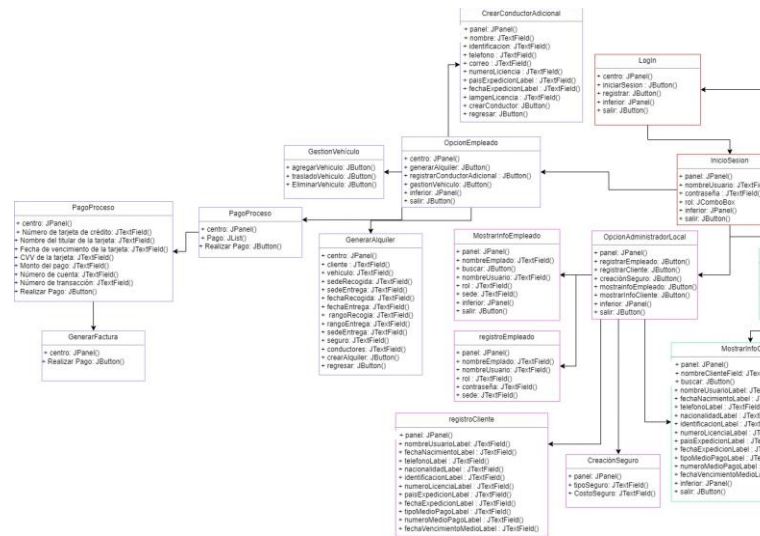
28 null: Administrador Sede Norte

29 null: Sede Norte

30 -----

31

e interacciones en el modelo principal se representan a continuación:



En el ejemplo actual se observa como se guardan los datos de un Administrador Local, en este caso del de la Sede Norte.

## 7. Diseño interfaz

Una vez aplicadas las modificaciones correspondientes el diseño de la interfaz realizado con las nuevas implementaciones