

1. Introducción

El presente documento detalla el diseño orientado a objetos de una aplicación de gestión para una empresa de alquiler de vehículos, considerando los cambios realizados en su operación. La aplicación aborda las responsabilidades de distintos usuarios, como administradores, empleados y clientes, proporcionando interfaces específicas para cada tipo de actor, diagramas que estructuran el funcionamiento de la aplicación, y su prevención de posibles errores. Adicionalmente, se integran elementos como la generación de facturas en formato PDF, la gestión de características de vehículos y la implementación de pagos con tarjeta de crédito a través de diversas pasarelas.

2. Elementos, aspirantes y estereotipos

A continuación, se presenta la estructura de clases, con una asignación de responsabilidades para la creación del diseño, esperando que la aplicación de software pueda abordarlas. En consideración al caso de uso, las funciones del programa se distribuyen entre los usuarios (administrador, cliente, empleado), ya que cada uno gestiona una parte específica del sistema. Esto facilita la identificación de los elementos candidatos que definen el control del programa.

En primer lugar, el componente inicial, dentro de la función estructural, se refiere a la administración de puntos de venta y usuarios de la aplicación. Según el modelo de dominio, es evidente que el primer componente es el administrador general, quien maneja los puntos de venta y desempeña un papel especial como el usuario de mayor rango con poder para manipular la estructura operativa de la empresa. En consecuencia, se ajusta al estereotipo de structurer, siendo el controlador de POS.

De manera similar, un segundo componente, encargado de tareas administrativas relacionadas con la aplicación, es el administrador del punto de venta. Este gestiona a los usuarios (clientes, empleados) que operan la aplicación para ofrecer el servicio de alquiler de vehículos. Puede ser etiquetado con el estereotipo de controller, ya que puede crear, eliminar y modificar la base de datos de estos usuarios. Además, certifica facturas a través de su firma digital y gestión del estado y disponibilidad de vehículos en una sede en especial. Regresando a los dos controladores, es esencial destacar que ambos operan sobre bases de datos acumuladas por el software, permitiendo la creación, modificación e incluso eliminación de datos. Por lo tanto, se les asigna el estereotipo de information holders para POS (catalogo vehículos, facturación, etc..) y usuarios.

En relación con los usuarios que utilizan los servicios de la aplicación, tanto clientes como empleados heredan atributos y métodos, pero también tienen atributos específicos según su estatus. En cuanto a la operación de la aplicación por parte de los clientes existe una división de funciones con respecto a sus métodos ya que su parte de la aplicación es independiente de lo demás. Con la opción de crear un nuevo usuario, tienen la capacidad de registrarse con toda la información solicitada, así como inscribir tarjeta de crédito y licencia de conducir; o si ya cuentan con una cuenta de usuario, proceden a la opción realizar una reserva, cancelarla, o editar cualquier información de la cuenta. Cabe mencionar que dentro del proceso de inscripción de tarjeta de crédito se deberá validar y bloquear para que la reserva sea exitosa.

Por otro lado, los empleados son responsables de verificar las reservas del usuario (verificar disponibilidad) y gestionar el catálogo de vehículos, con la capacidad de crear, modificar y eliminar elementos del catálogo, así como definir el estado del vehículo (disponibilidad). Esto también crea un information holder de automóviles, constituyendo la base de datos del catálogo de vehículos. Y además se encarga de realizar el pago con los diferentes métodos, además de realizar la facturación PDF que también se almacene como un information holder.

Es por esta razón que la interfaz de la aplicación varía según el tipo de actor, ya que la funcionalidad de la aplicación implica la interacción de todos los actores entre sí para llevar a cabo la parte operacional de la empresa. Esto comienza con una vista que interactúa con el actor, definiendo sus funciones y accesos según su designación (cliente, empleado, administrador...). En este contexto, se hace evidente y necesario el estereotipo de Interfacer, ya que transforma información y solicitudes entre diversas partes de un sistema.

3. Participantes: Responsabilidades y Colaboraciones

3.1 Administrador:

3.1.1 Responsabilidades:

- Registrar la compra de nuevos vehículos.
- Asignar ubicaciones a los vehículos en sedes específicas.
- Generar facturas en formato PDF al momento de la entrega y pago de un vehículo.
- Gestionar las características de los vehículos, incluyendo el tipo y el porcentaje adicional de la prima del seguro.
- Firmar la factura como imagen.
- Mantener la configuración de las pasarelas de pago disponibles.
- Cambiar la implementación de las pasarelas según sea necesario.

3.1.2 Colaboraciones:

- Colabora con la interfaz de usuario para la administración de vehículos y generación de facturas.
- Colabora con las pasarelas de pago para la configuración y cambio de implementación.

3.2 Cliente:

3.2.1 Responsabilidades:

- Utilizar la aplicación para clientes para registrar un nuevo usuario.
- Consultar la disponibilidad de vehículos para una sede en fechas específicas.
- Realizar reservas de vehículos con descuento si se paga a través de la aplicación de clientes.

3.2.2 Colaboraciones:

- Colabora con la interfaz de usuario para realizar acciones específicas destinadas a clientes.
- Colabora con las pasarelas de pago al realizar pagos simulados.

3.3 Empleado:

3.3.1 Responsabilidades:

- Manejar la entrega y devolución de vehículos.
- Realizar el bloqueo del cupo en la tarjeta de crédito al momento de la entrega del vehículo.
- Utilizar la aplicación de empleados para realizar cobros y bloquear cupos con pasarelas de pago.

3.3.2 Colaboraciones:

- Colabora con la interfaz de usuario para llevar a cabo acciones relacionadas con la entrega y devolución de vehículos.
- Colabora con las pasarelas de pago para realizar cobros y bloquear cupos.

4. Colaboraciones Estructurales

4.1 Interfaz de Usuario

4.1.1 Responsabilidades:

- Proporcionar interfaces visuales diferenciadas para administradores, clientes y empleados.
- Facilitar la interacción del usuario con la aplicación.

4.1.2 Colaboraciones:

- Colabora con el administrador para la gestión de vehículos y generación de facturas.
- Colabora con el cliente para realizar acciones específicas destinadas a clientes.
- Colabora con el empleado para acciones relacionadas con la entrega y devolución de vehículos.

4.2 Pasarelas de Pago

4.2.1 Responsabilidades:

- Proveer servicios para recibir información de tarjetas de crédito y detalles de pagos.
- Retornar resultados indicando el éxito o los problemas en el proceso de pago.

4.2.2 Colaboraciones:

- Colabora con el administrador para la configuración y cambio de implementación.
- Colabora con el empleado al realizar cobros y bloquear cupos.
- Colabora con el cliente al realizar pagos simulados.

4.3 Aplicación para Clientes

4.3.1 Responsabilidades:

- Proporcionar una interfaz gráfica para que los clientes realicen acciones específicas.
- Realizar operaciones de registro de usuario, consulta de disponibilidad y reservas con descuento.

4.3.2 Colaboraciones:

- Colabora con el cliente para acciones destinadas a clientes.
- Colabora con la interfaz de usuario para la presentación visual de opciones.

5. Estilos de Control y Patrón de Diseño

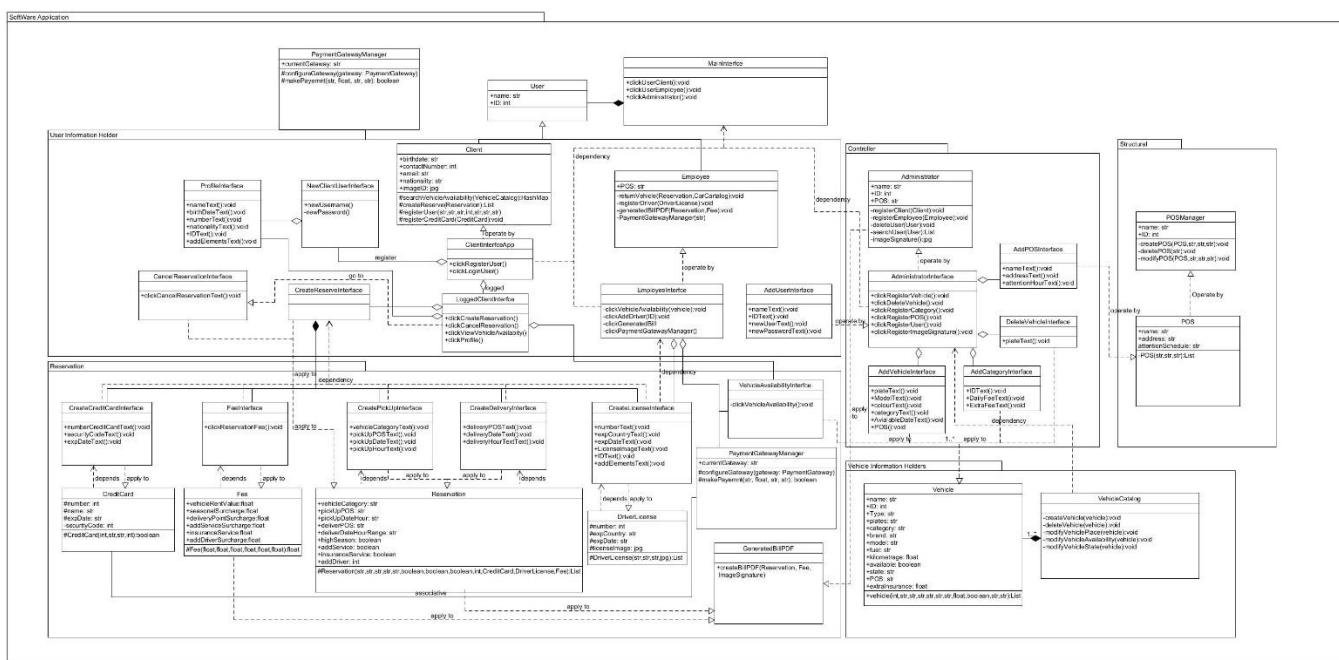
5.1 La aplicación sigue un estilo de control basado en el modelo MVC (Modelo-Vista-Controlador) para asegurar una estructura modular y una separación clara de las responsabilidades. Este enfoque facilita la mantenibilidad, escalabilidad y extensibilidad del sistema. Puede visualizarse la vista como las interfaces que se le muestran al usuario con las cuales interactúa, el controlador sería la funcionalidad de los botones, cuadros de texto etc... donde se toma toda la información, y el modelo son las funciones de almacenamiento de datos y validación para la operación de estos.

5.2 Patrón de Diseño Observador

Este patrón permite que la Vista observe los cambios en el Modelo. Cuando el Modelo se actualiza, notifica a todas las Vistas registradas, asegurando que la interfaz de usuario se actualice correctamente. Por esta razón se selecciona para mantener una sincronización efectiva entre el Modelo y la Vista en la aplicación. Este patrón establece una relación uno a muchos entre objetos, permitiendo que múltiples observadores (las Vistas) sean notificados automáticamente cuando el estado del objeto observado (el Modelo) cambia. En el contexto de la aplicación, la Vista puede observar cambios en el Modelo y actualizar dinámicamente la interfaz de usuario sin depender de acoples directos.

6. Diagrama de Clases: Métodos y Atributos del Diseño

Recopilando la estructura planteada anteriormente, sus conceptos y el funcionamiento de la aplicación se presenta el diagrama de clases de la aplicación, que funcionara como base para la realización de la implementación.



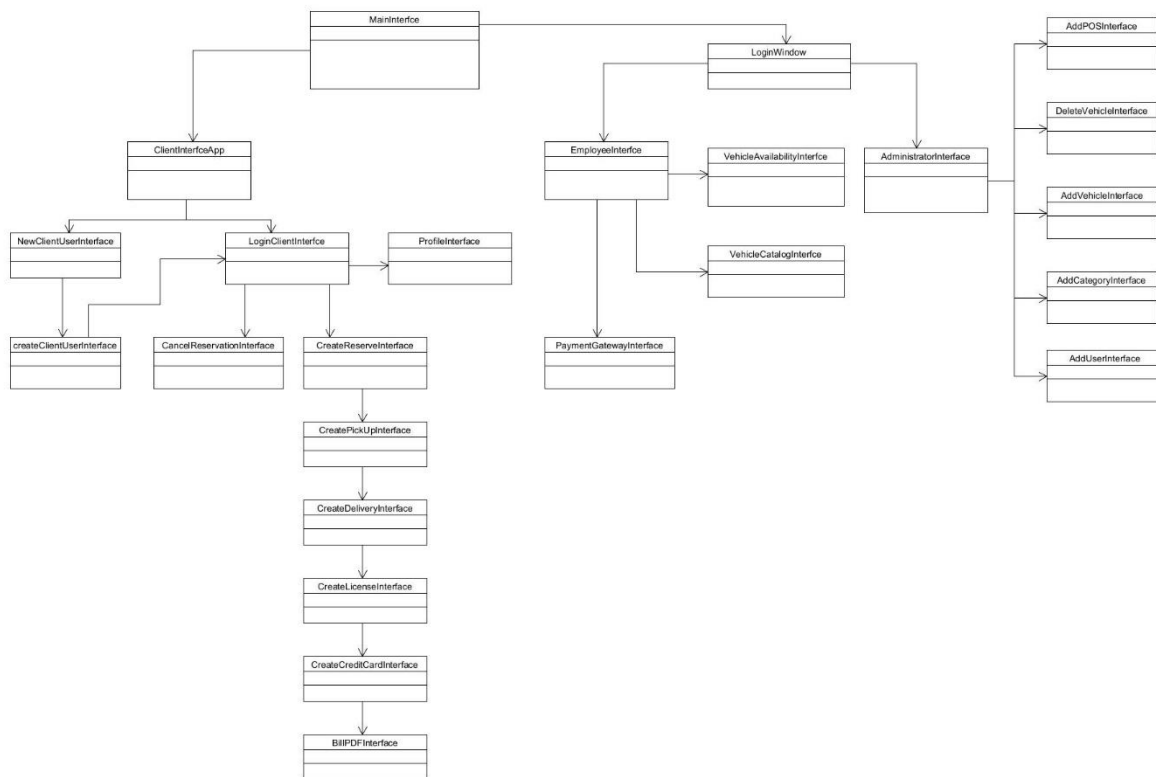
Entre sus clases más importantes se encuentran tales como:

- **Reservation:** Esta clase representa una reservación realizada por un cliente. Tiene atributos como, delivery address, delivery date, delivery hour, payment gateway, clase del vehículo, etc. También tiene métodos como creating a reservation, modifying a reservation, canceling a reservation.
- **Client:** Esta clase representa a un cliente. Tiene atributos como name, ID, address, etc. También tiene métodos como registering a client, crear perfil, realizar la reserva, inscribir tarjeta de crédito, inscribir licencia de conducción.
- **Employee:** Esta clase representa a un empleado. Tiene atributos como name, ID, address, etc. Y sus métodos son todos asociados a las funciones del empleado como realizar el pago, modificar el estado del vehículo etc.....
- **Vehicle:** Esta clase representa a un vehículo. Tiene atributos como type, license plate, model, color, category, brand, fuel type, kilometerage, availability, etc. También tiene métodos como registering a vehicle, modifying a vehicle, deleting a vehicle, etc. Y cabe mencionar que los atributos van siendo funcionales dependiendo del tipo de este.
- **PaymentGateway:** Esta clase representa a un sistema de pago. Tiene atributos como gateway name, configuration details, etc. También tiene métodos como creating a payment gateway, que funciona como las pasarelas de pago.
- **VehicleCatalog:** Esta clase representa a un catálogo de vehículos. Tiene métodos como creating a vehicle, deleting a vehicle, modifying a vehicle, etc.
- **POSManager:** Esta clase representa a un administrador de puntos de venta (POS). Tiene métodos como registering a POS, modifying a POS, deleting a POS, etc.
- **User:** Esta clase representa a un usuario en el sistema. Tiene atributos como name, ID, POS, role, etc. Que van dependiendo del tipo usuario, del mismo modo sus métodos.
- **GenerateBillPDF:** Esta clase genera la factura del alquiler del vehículo, la cual será por medio de un PDF y su información contiene la imagen de la firma del administrador, la factura y la reserva respectivamente.

Y sus subdivisiones de controller, structural, e informationholders hacen referencia al estilo de control y sus estereotipos y participantes, mientras que reservation contiene todo el proceso principal de la aplicación que es realizar la reserva ya que opera bajo la interacción directa e indirecta de todos los subprocesos de la aplicación.

7. Diagrama de Interfaces de alto nivel

Continuando con el funcionamiento de la aplicación, se presenta un diagrama de clases de alto nivel de la operación de las interfaces.



Este diagrama muestra cómo se organizan y se comunican entre sí las diferentes clases y funcionalidades en la aplicación. Las líneas de asociación entre las clases representan las interacciones y las relaciones entre las clases.

8. Diseño de Posibles Errores

8.1 Error en la Validación de Tarjeta de Crédito:

- Si un cliente intenta registrar una tarjeta de crédito y la validación falla (por ejemplo, número incorrecto o tarjeta vencida), se debe generar un mensaje de error y bloquear la reserva.

8.1.1 Manejo:

- La aplicación mostrará un mensaje de error específico indicando el problema en la validación de la tarjeta de crédito. La reserva no se procesará hasta que se proporcione una tarjeta válida.

8.2 Error en la Reserva de Vehículo:

- Si un cliente intenta realizar una reserva y hay un problema, como falta de disponibilidad de vehículos en las fechas deseadas, se debe informar al cliente.

8.2.1 Manejo:

- La aplicación mostrará un mensaje de error indicando que no hay vehículos disponibles para las fechas seleccionadas. Se ofrecerá al cliente la opción de modificar las fechas o elegir otra categoría de vehículo.

8.3 Error en el Proceso de Pago:

- Si hay un problema durante el proceso de pago (por ejemplo, tarjeta rechazada o transacción fallida), se debe informar al empleado y al cliente.

8.3.1 Manejo:

- La aplicación mostrará un mensaje de error detallando el problema con el pago. Se ofrecerán opciones para corregir la información de pago o seleccionar otra forma de pago.

8.4 Error en la Generación de Facturas en PDF:

- Si hay un error al generar la factura en formato PDF al momento de la entrega y pago de un vehículo, se debe registrar el problema.

8.4.1 Manejo:

- La aplicación mostrará un mensaje de error y registrará la incidencia. Se notificará al administrador para abordar el problema de generación de facturas.

8.5 Error en la Configuración de Pasarelas de Pago:

- Si hay un problema al cambiar la configuración de las pasarelas de pago por parte del administrador, se deben mantener las configuraciones anteriores y notificar al administrador sobre el error.

8.5.1 Manejo:

- La aplicación mostrará un mensaje de error y revertirá la configuración a la versión anterior. Se notificará al administrador para abordar el problema de configuración.

8.6 Error en la Autenticación del Usuario:

- Si un usuario (administrador, cliente, empleado) intenta autenticarse con credenciales incorrectas, se debe informar sobre el error.

8.6.1 Manejo:

- La aplicación mostrará un mensaje de error indicando que las credenciales son incorrectas. Se proporcionarán opciones para restablecer la contraseña o volver a intentar la autenticación.

8.7 Error en la Creación de Reservas:

- Si hay un problema al crear una reserva, como la falta de información necesaria, se debe informar al cliente.

8.7.1 Manejo:

- La aplicación mostrará un mensaje de error indicando la información faltante o incorrecta. Se guiará al cliente para completar la reserva correctamente.

8.8 Error en el Bloqueo del Cupo en la Tarjeta de Crédito:

- Si hay un problema al bloquear el cupo en la tarjeta de crédito al momento de la entrega del vehículo, se debe informar al empleado y al cliente.

8.8.1 Manejo:

- La aplicación mostrará un mensaje de error detallando el problema con el bloqueo del cupo. Se ofrecerán opciones para corregir la información de pago o seleccionar otra forma de bloqueo.