Universidad de los Andes
Ingeniería de Sistemas y de Computación
Diseño y Programación O.O.
Proyecto 1 | Entrega 2

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

Diseño del Proyecto

1. Componentes candidatos y estereotipos

A continuación, se presenta el diagrama de diseño de clases, para la realización de este se elaboró una división de las tareas que se espera que la aplicación de Software resuelva. Teniendo en cuenta el caso de usuario, las funciones del programa son divididos en sus usuarios (administrador, cliente, empleado) ya que cada uno opera cierta parte del sistema. De está manera se puede hacer una inferencia de los componentes candidatos que forjan el control del programa.

En primera instancia, el primer componente como parte de la función estructural es la administración de los puntos de venta y de los usuarios de la aplicación, de acuerdo con el modelo de dominio es evidente que el primer componente es el administrador general, quien se ocupa de gestionar los puntos de venta, teniendo un rol especial ya que es el usuario con mayor rango y poder para manipular el funcionamiento estructural de la empresa, por ello cumple con el estereotipo de *controlador de POS*.

Del mismo modo, un segundo componente que realiza labores de administración con respecto a la aplicación es el administrador del punto de venta, ya que este se encarga de gestionar a los usuarios (clientes, empleados) que operan la aplicación para realizar el servicio de alquiler de vehículo. A este se le puede asignar el estereotipo de *controlador de usuarios* al poder crear, eliminar y modificar la base de datos de estos.

Retomando a los dos controladores es pertinente mencionar que cada uno opera bajo una base de datos que el software acumula y da acceso de crear, modificar e incluso eliminar, por tanto, corresponden al estereotipo de *information holders* para POS y usuarios. Por otra parte, continuando con los usuarios que se ocupan de operar los servicios de la aplicación, es de notar que los clientes y empleados heredando los atributos y métodos, también tiene atributos propios los cuales van de acuerdo con su estatus. Con respecto a los métodos los clientes tienen la completa gestión de realizar las reservas donde se les solicita información extra, además de inscripción de objetos como una tarjeta de crédito y licencia de conducir (teniendo atributos y métodos propios), y además la aplicación crea una tarifa el cual debe de pagar el usuario cuando realice el alquiler. Del otro lado, los empleados tienen la labor de comprobar las reservas del usuario (verificar disponibilidad) y realizar la gestión del catálogo de vehículos, poder crear, modificar y eliminar del catálogo; así como definir el estado del auto (disponibilidad). Esto también crea un *information holder* de automóviles, y es la base de datos de lo que se compone el catalogo de vehículos.

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

2. Responsabilidades

Teniendo en cuenta lo anterior, considerando los componentes candidatos, se contemplaron algunas de las responsabilidades de las cuales se ocuparán. La siguiente tabla presenta en detalle cada una de las responsabilidades y el componente asociado que debe asumirlas.

#	Responsabilidades	Componente
1	Gestionar los puntos de venta (crear, modificar, eliminar)	Administrador General
2	Realizar creación y eliminación de vehiculos para añadir al catalogo	
3	Administra las operaciones que relacionen a dos puntos de venta entre si	
4	Registrar y gestionar la información de los empleados de la sede.	Administrador POS
5	Crear y gestionar cuentas de usuario para los empleados de la sede.	
6	Registrarse en el sistema proporcionando información personal, datos de licencia de conducción y detalles de pago.	Usuario Cliente
7	Realizar reservas de vehículos especificando categoría, sede de recogida, fecha y hora.	
8	Pagar el 30% del valor proyectado de alquiler al hacer una reserva.	
9	Realizar el pago completo del alquiler cuando se recoge el vehículo.	
10	Elegir seguros adicionales, si se ofrecen, al formalizar el alquiler.	
11	Proporcionar información de licencia de conducir y detalles de otros conductores, si es necesario.	
12	Registrar la entrega y devolución de vehículos, verificando su estado.	Usuario Empleado
13	Asistir a los clientes en la recogida y devolución de vehículos.	
14	Gestionar el bloqueo y desbloqueo de la tarjeta de crédito del cliente en el momento de la recogida y devolución del vehículo.	
15	Registrar y reportar la necesidad de mantenimiento de vehículos.	
16	Gestionar el catálogo de vehículos (modificando su información)	

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

3. Colaboraciones

Dentro de las responsabilidades establecidas, también se ha tenido en cuenta la necesidad de cooperación entre los diversos elementos para cumplirlas. Esto se describe a continuación.

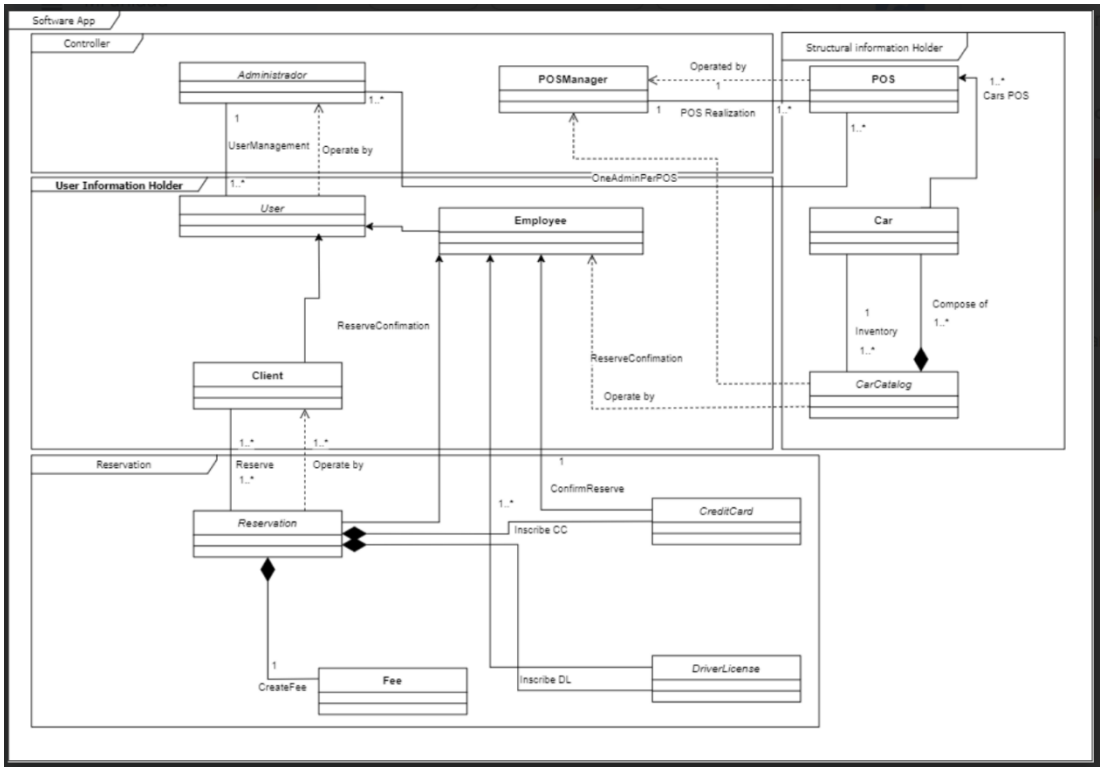
#	Descripción	Colaboración
1	El administrador general colabora con los administradores de las sedes locales para asignar vehículos a ubicaciones específicas y asegurarse de que haya un inventario adecuado en cada sede.	Administrador General con Administrador del Punto de Venta (Sede Local)
2	Los administradores de las sedes locales colaboran con los empleados en la gestión diaria de las operaciones de alquiler de vehículos. Además de supervisar su actividad de usuario.	Administrador del Punto de Venta con Empleados de la Sede Local
3	Los clientes interactúan con los empleados de las sedes locales al recoger y devolver vehículos.	Cliente con Empleados de la Sede Local
4	Los empleados ayudan a los clientes en el proceso de alquiler, proporcionando el vehículo reservado y asegurándose de que el vehículo se encuentre en buen estado.	
5	El administrador del punto de venta se encarga de supervisar la actividad del cliente a partir de su usuario	Cliente con Administrador del Punto de Venta

Universidad de los Andes
Ingeniería de Sistemas y de Computación
Diseño y Programación O.O.
Proyecto 1 | Entrega 2

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

4. Diagrama de Clases de Alto Nivel

A continuación, se presenta el diagrama de clases de alto nivel. El cual es una representación gráfica de las clases obtenidas en el primer punto, junto con las responsabilidades y colaboraciones que las relacionan de forma que la aplicación cumpla con los requerimientos y restricciones planteados en el análisis.

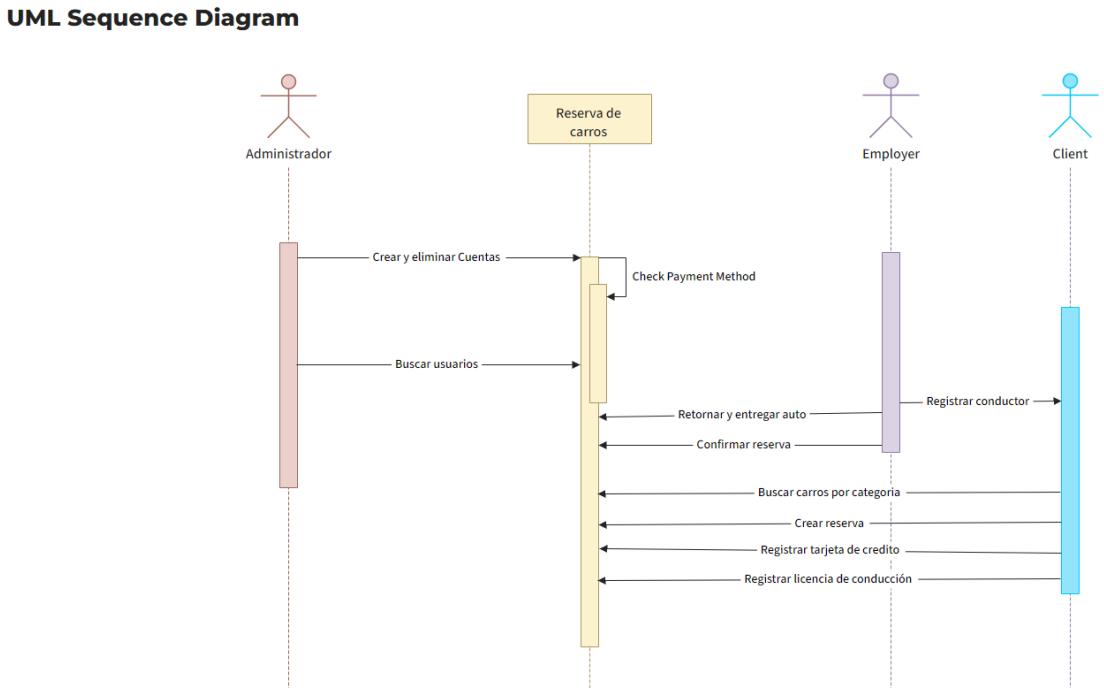


En este diagrama se pueden ver mucho más claras las relaciones entre las clases y la dependencia entre estas:

- Administrador: solo existe un administrador por POS (Points Of Sale) y de la creación de usuarios.
- Cliente: se encarga de las reservas.
- POS: este indica el punto de venta en que se encuentra el carro, a su vez indica la marca del mismo.
- Empleado: entrega y devolución de carros, registro de conductores adicionales, actualización del estado de los carros.
- Reservación: le indica al empleado que carro se busca y este puede indicar si se encuentra disponible, además se encuentra compuesta por la licencia del conductor, la tarjeta de crédito y el Fee que se toma como la tarifa que va adquiriendo la reserva con base a lo que diga el cliente.
- CarCatalog: el empleado puede ver el catálogo de carros así puede saber si lo que está pidiendo el cliente se encuentra disponible, está compuesta por características que definirán su interés.

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

5. Diagrama de Secuencias



Este diagrama de secuencia ilustra el proceso de reserva y alquiler de vehículos en el sistema de gestión de alquiler de vehículos de la empresa.

Participantes:

- **Cliente:** El usuario que desea reservar o alquilar un vehículo.
- **Empleado de la Sede:** El empleado de la sede de la empresa que atiende a los clientes y registra las reservas y alquileres.
- **Administrador:** El encargado de manejar toda la información de todos los usuarios y el manejo de ellos.
- **Sistema:** El sistema de gestión de alquiler de vehículos que facilita la interacción entre el cliente, el empleado de la sede y la base de datos.

El diagrama de secuencia describe el proceso de reserva y alquiler de vehículos en una empresa de gestión de alquiler de vehículos. Involucra a tres actores clave: el cliente, el empleado de la sede y el sistema. El proceso comienza con la autenticación del cliente, seguido de la selección de opciones en un menú intuitivo.

La elección de "Hacer Reserva" inicia un proceso de varios pasos, que incluye la elección de una categoría de vehículo, sede de recogida, fecha y hora de recogida, y opciones adicionales. Se confirma la reserva con un pago del 30% del valor proyectado a través de un sistema de pagos.

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

El empleado de la sede es notificado para preparar el vehículo, verificar su disponibilidad y entregarlo al cliente. La entrega implica la firma de documentos y la formalización del contrato de alquiler. La devolución del vehículo se registra en el sistema.

La facturación final se calcula considerando varios factores, y el cliente realiza el pago a través del sistema de pagos. El sistema actualiza la disponibilidad del vehículo en la base de datos para futuras reservas.

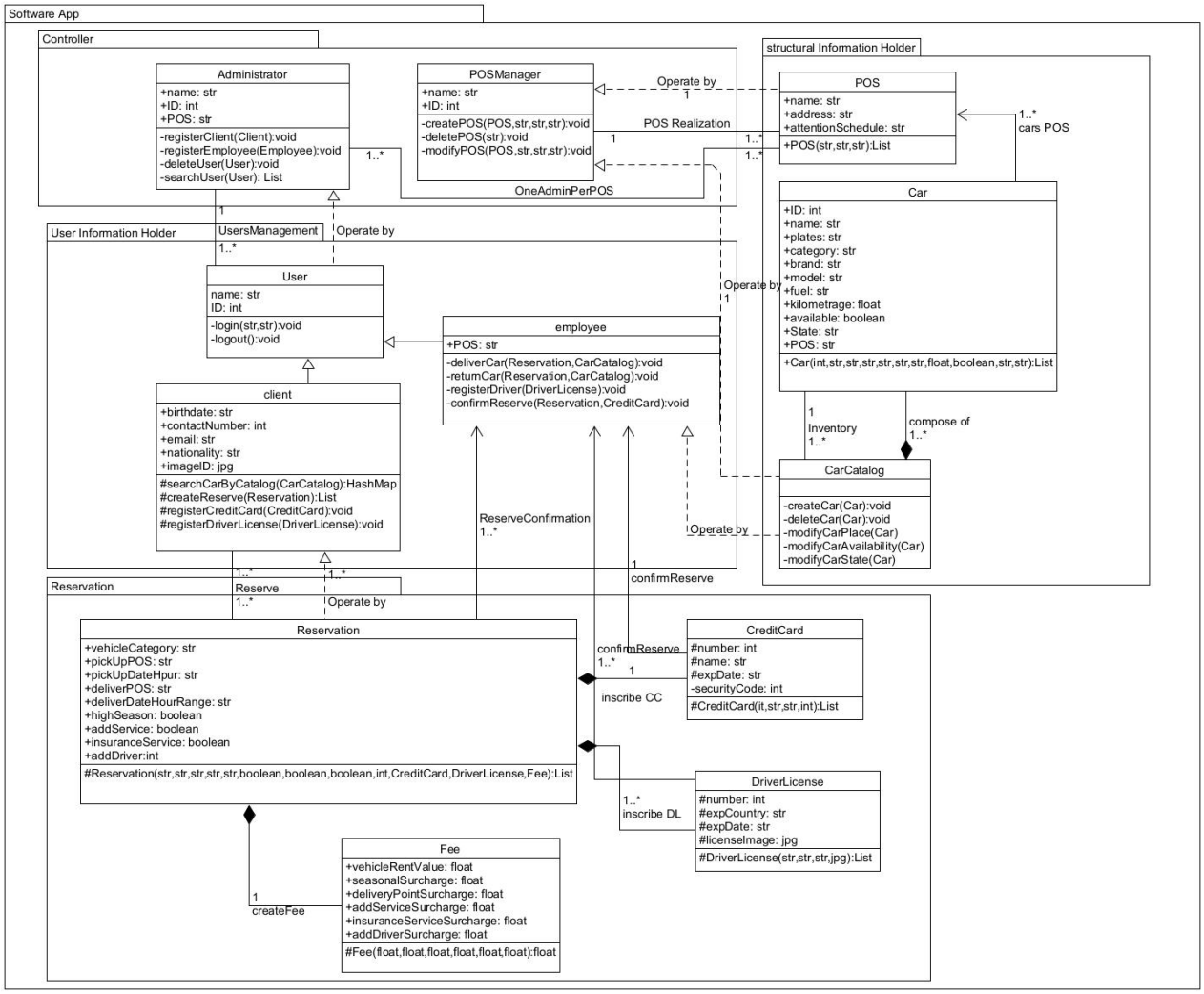
Este diagrama muestra cómo el cliente, el empleado de la sede y el sistema interactúan para garantizar una experiencia eficiente de alquiler de vehículos.

6. Diagrama de Clases: Atributos y Métodos

Finalmente, siguiendo las relaciones identificadas al definir las colaboraciones, responsabilidades y teniendo en cuenta la subdivisión en clases realizada por componente, se llega al diseño definitivo que servirá como base para la implementación de la solución, que se puede apreciar seguidamente:

Universidad de los Andes
Ingeniería de Sistemas y de Computación
Diseño y Programación O.O.
Proyecto 1 | Entrega 2

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González



Prosiguiendo con la comprensión del diseño definitivo, se procede a realizar una descripción de las clases con respecto a sus métodos y atributos.

Clase	Administrator
Atributos	name (str): El nombre del administrador.

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

	ID (int): Identificador único del administrador.
	POS (str): Posición u cargo del administrador en la empresa.
Métodos	registerClient(Client) (void): Este método permite al administrador registrar un nuevo cliente en el sistema. Recibe un objeto Client como parámetro y lo agrega a la base de datos de clientes.
	registerEmployee(Employee) (void): Permite al administrador registrar un nuevo empleado en el sistema. Recibe un objeto Employee como parámetro y lo agrega a la base de datos de empleados.
	deleteUser(User) (void): Permite al administrador dar de baja a un usuario (tanto cliente como empleado) del sistema. Recibe un objeto User como parámetro y lo elimina de la base de datos de usuarios.
	searchUser(User) (List): Este método busca usuarios en el sistema según un criterio específico, como nombre o ID, y devuelve una lista de usuarios que coinciden con el criterio especificado.

Clase	POSManage
Atributos	name (str): El nombre del administrador de punto de venta.
	ID (int): Identificador único del administrador de punto de venta.
Métodos	createPOS(POS, str, str, str) (void): Este método permite al administrador de punto de venta crear un nuevo punto de venta (POS) en el sistema. Toma como parámetros un objeto POS y tres strings para la ubicación, horario de apertura y horario de cierre del nuevo POS. Agrega esta información a la base de datos de puntos de venta.
	deletePOS(str) (void): Permite al administrador de punto de venta eliminar un punto de venta del sistema, identificado por su ubicación (string). Elimina la información correspondiente de la base de datos de puntos de venta.
	modifyPOS(POS, str, str, str) (void): Este método permite al administrador de punto de venta modificar la información de un punto de venta existente. Recibe un objeto POS para identificar el POS que se va a modificar y tres strings para actualizar la ubicación, horario de apertura y horario de cierre del POS en cuestión.

Clase	POS
Atributos	name (str): El nombre del punto de venta (POS).
	address (str): La dirección del punto de venta.
	attentionSchedule (str): El horario de atención del punto de venta.

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

Métodos	POS(str, str, str) (List): El método constructor recibe tres parámetros: name, address, y attentionSchedule. Crea una instancia de POS con los atributos proporcionados y devuelve una lista con estos atributos para representar un nuevo punto de venta.

Clase	User
Atributos	name (str): El nombre del usuario.
	ID (int): El identificador único del usuario.
Métodos	login(str, str) (void): Este método permite que el usuario inicie sesión en el sistema proporcionando un nombre de usuario (str) y una contraseña (str). El método verifica la identidad del usuario y establece su sesión en el sistema.
	logout() (void): Permite al usuario cerrar sesión en el sistema, lo que significa que el usuario finaliza su sesión actual.

Clase	Client
Atributos	birthdate (str): La fecha de nacimiento del cliente.
	contactNumber (int): El número de contacto del cliente.
	email (str): La dirección de correo electrónico del cliente.
	nationality (str): La nacionalidad del cliente.
	imageID (jpg): Una imagen que identifica al cliente.
Métodos	searchCarByCatalog(CarCatalog) (HashMap): Este método permite al cliente buscar automóviles disponibles en el catálogo de autos (objeto CarCatalog) y devuelve un mapa (HashMap) con la información de los autos encontrados.
	createReserve(Reservation) (List): Permite al cliente crear una reserva de un vehículo especificado por un objeto Reservation. Devuelve una lista que contiene información sobre la reserva realizada.
	registerCreditCard(CreditCard) (void): Este método permite al cliente registrar su tarjeta de crédito en el sistema para realizar pagos. Recibe un objeto CreditCard como parámetro y lo almacena en la base de datos de tarjetas de crédito.
	registerDriverLicense(DriverLicense) (void): Permite al cliente registrar su licencia de conducir en el sistema. Recibe un objeto DriverLicense como parámetro y lo almacena en la base de datos de licencias de conducir.

Universidad de los Andes
Ingeniería de Sistemas y de Computación
Diseño y Programación O.O.
Proyecto 1 | Entrega 2

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

Clase	Employee
Atributos	POS (str): El punto de venta (POS) al que está asignado el empleado.
Métodos	deliverCar(Reservation, CarCatalog) (void): Este método permite al empleado de la agencia entregar un automóvil al cliente como parte de una reserva. Recibe un objeto Reservation y el catálogo de automóviles CarCatalog. El empleado realiza la entrega del automóvil.
	returnCar(Reservation, CarCatalog) (void): Permite al empleado recibir la devolución de un automóvil por parte de un cliente. Recibe un objeto Reservation y el catálogo de automóviles CarCatalog. El empleado registra la devolución del automóvil en el sistema.
	registerDriver(DriverLicense) (void): Este método permite al empleado registrar la información de una licencia de conducir en el sistema. Recibe un objeto DriverLicense y lo almacena en la base de datos de licencias de conducir.
	confirmReserve(Reservation, CreditCard) (void): Permite al empleado confirmar una reserva realizada por un cliente. Recibe un objeto Reservation y una tarjeta de crédito CreditCard para realizar la confirmación y el cobro correspondiente.

Clase	Car
Atributos	ID (int): El identificador único del automóvil.
	name (str): El nombre o identificación del automóvil.
	plates (str): La placa del automóvil.
	category (str): La categoría del automóvil (por ejemplo, "pequeños", "SUV", "vans", "lujo", etc.).
	brand (str): La marca del automóvil.
	model (str): El modelo del automóvil.
	fuel (str): Tipo de combustible que utiliza el automóvil.
	kilometrage (float): El kilometraje del automóvil.
	available (boolean): Un indicador que señala si el automóvil está disponible para alquiler.
	State (str): El estado del automóvil (puede indicar si está en mantenimiento o limpieza, por ejemplo).
	POS (str): El punto de venta (POS) al que está asociado el automóvil.
Métodos	Car(int, str, str, str, str, str, str, float, boolean, str, str) (List): El método constructor recibe valores para todos los atributos de la clase y crea una instancia de Car con la información proporcionada. Devuelve una lista que contiene estos atributos para representar un nuevo automóvil.

Clase	CarCatalog

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

Métodos	createCar(Car) (void): Este método permite agregar un automóvil (objeto Car) al catálogo de automóviles. El automóvil se registra en la lista de automóviles disponibles en el sistema.
	deleteCar(Car) (void): Permite eliminar un automóvil (objeto Car) del catálogo de automóviles. El automóvil se quita de la lista de automóviles disponibles.
	modifyCarPlace(Car) (void): Este método permite modificar la ubicación (POS) de un automóvil (objeto Car) en el catálogo. Esto puede ser útil si un automóvil se traslada a una ubicación diferente.
	modifyCarAvailability(Car) (void): Permite modificar la disponibilidad de un automóvil (objeto Car) en el catálogo. Esto se utiliza para indicar si un automóvil está disponible para alquiler o no.
	modifyCarState(Car) (void): Permite modificar el estado de un automóvil (objeto Car) en el catálogo. Por ejemplo, se puede cambiar el estado del automóvil a "mantenimiento" o "limpieza".

Clase	Reservation
Atributos	vehicleCategory (str): La categoría de vehículo reservado.
	pickUpPOS (str): El punto de venta (POS) donde se recogerá el vehículo.
	pickUpDateHour (str): La fecha y hora de recogida del vehículo.
	deliverPOS (str): El punto de venta (POS) donde se devolverá el vehículo.
	deliverDateHourRange (str): El rango de fechas y horas de entrega del vehículo.
	highSeason (boolean): Un indicador que señala si la reserva se realiza durante la temporada alta.
	addService (boolean): Un indicador que señala si se solicita un servicio adicional.
	insuranceService (boolean): Un indicador que señala si se solicita un servicio de seguro adicional.
	addDriver (int): La cantidad de conductores adicionales que se agregarán a la reserva.
Métodos	Reservation(str, str, str, str, str, boolean, boolean, boolean, int, CreditCard, DriverLicense, Fee) (List): El método constructor recibe valores para todos los atributos de la clase y crea una instancia de Reservation con la información proporcionada. Devuelve una lista que contiene estos atributos para representar una nueva reserva.

Clase	CreditCard

Andrés Juan Cárdenas Layton
Natalia Andrea Bohórquez Jiménez
Ángela Jiménez González

Atributos	number (int): El número de la tarjeta de crédito.
	name (str): El nombre del titular de la tarjeta de crédito.
	expDate (str): La fecha de vencimiento de la tarjeta de crédito.
	securityCode (int): El código de seguridad de la tarjeta de crédito.
Métodos	CreditCard(int, str, str, int) (List): El método constructor recibe valores para todos los atributos de la clase y crea una instancia de CreditCard con la información proporcionada. Devuelve una lista que contiene estos atributos para representar una nueva tarjeta de crédito.

Clase	DriverLicense
Atributos	number (int): El número de licencia de conducir.
	expCountry (str): El país de expedición de la licencia de conducir.
	expDate (str): La fecha de vencimiento de la licencia de conducir.
	licenseImage (jpg): Una imagen que representa la licencia de conducir.
Métodos	DriverLicense(str, str, str, jpg) (List): El método constructor recibe valores para todos los atributos de la clase y crea una instancia de DriverLicense con la información proporcionada. Devuelve una lista que contiene estos atributos para representar una nueva licencia de conducir.

Clase	Fee
Atributos	vehicleRentValue (float): El costo del alquiler del vehículo.
	seasonalSurcharge (float): El recargo por temporada.
	deliveryPointSurcharge (float): El recargo por entrega en un punto de venta diferente.
	addServiceSurcharge (float): El recargo por servicios adicionales.
	insuranceServiceSurcharge (float): El recargo por servicios de seguro adicional.
	addDriverSurcharge (float): El recargo por conductores adicionales.
Métodos	Fee(float, float, float, float, float, float) (float): Este método calcula el costo total de la reserva teniendo en cuenta los atributos de la clase, como el costo del alquiler, recargos por temporada, entrega en un punto de venta diferente, servicios adicionales, servicios de seguro adicional y recargos por conductores adicionales. Devuelve el costo total de la reserva como un valor float.

Para concluir con el diseño final de la aplicación, En conjunto, estas clases forman la estructura del sistema de alquiler de autos y permiten la gestión de clientes, empleados, vehículos, reservas y pagos, lo que facilita el funcionamiento eficiente de la agencia de alquiler de autos. Con esto ya está definido el programa por lo que finalmente se prosigue a desarrollarlo.