

David Caro - 202222073
Alejandro Abril - 202224328
Gabriel Martinez - 202214559

Documento de Diseño
Proyecto 3 DPOO

Descomposición del problema.

Sistema completo: Se debe construir un sistema para controlar una parte de las operaciones de una empresa que se encarga de rentar automóviles.

Primera descomposición:

Inventario	Alquiler	Sedes	Interfaz	Funcionalidades
Debe tener un sistema de inventario, de los carros disponibles en cada una de las sedes.	Debe tener un sistema de alquiler que gestione el proceso desde que se reserva un vehículo hasta que es regresado, incluyendo el cálculo de tarifas, la gestión de clientes, conductores y pagos.	Manejará la información de las sedes incluyendo la información de sus empleados, y el control de los mantenimientos de los vehículos.	Se debe hacer una interfaz gráfica agradable para el usuario, en busca de que no se interactúe con el usuario mediante la consola	La información debe guardarse en archivos, los cuales deben ser persistentes, dentro de una carpeta, la cual solo puede editar la aplicación. La persistencia no necesariamente debe hacerse en un archivo, puede haber más de un archivo. La carpeta donde se guardan los archivos no debe ser la misma que la del código fuente. Todos los usuarios deben tener un login y un password y usaran la misma app dependiendo del tipo de

				usuario. Debe ser capaz de generar un archivo de log con el historial de un vehículo.
--	--	--	--	--

Segunda descomposición:

ALQUILER		
Controla las reservas y alquileres de vehículos, los únicos que pueden reservar o alquilar vehículos, son clientes registrados.		
Cliente: <ol style="list-style-type: none"> 1. Nombres 2. Datos de contacto 3. Fecha de Nacimiento 4. Imagen doc de identidad 	Tarjeta credito: <ol style="list-style-type: none"> 1. Número 2. Fecha Vencimiento 3. CCV 4. Banco 	Alquileres: <ol style="list-style-type: none"> 1. Categoría 2. Tarifa diaria 3. Sede entrega 4. Sede devolución 5. Seguros 6. Conductor adicional 7. Cobro final
Licencia: <ol style="list-style-type: none"> 1. Número 2. País exp 3. Imagen doc 4. Fecha Venc 	Reserva: <ol style="list-style-type: none"> 1. Tipo de carro 2. Fecha y hora retorno 3. Cobro inicial 	Reserva Especial: <ol style="list-style-type: none"> 1. Sede donde se encuentra el carro

SEDES		
Esta parte es la que maneja las sedes y los empleados		
Sedes <ol style="list-style-type: none"> 1. Nombre 2. Ubicación 	Administrador general: <ol style="list-style-type: none"> 1. Administrar sede 2. Agregar sede 	Empleado: <ol style="list-style-type: none"> 1. Actualizar estado carro

3. Horarios de atención		2. Mantenimiento 3. Fecha disponibilidad
<p style="text-align: center;">Administrador local:</p> 1. Sede 2. Nuevo empleado 3. Nuevo usuario 4. Cambiar empleado		

INVENTARIO	
Maneja el inventario de los vehículos que son propiedad de la empresa.	
Vehículo: 1. Placa 2. Marca 3. Modelo 4. Color 5. Tipo de transmisión	Categorías: 1. Pequeños 2. SUV 3. Vans 4. Lujo 5. Eléctricos

FUNCIONALIDADES		
Archivo sedes: 1. Datos sede (Nombre, Ubicación, Horario atención) 2. Administrador general (Nombre)	Archivo Empleado: 1. Datos empleados (Nombre, sede) 2. Datos administración local (Nombre, sede)	Archivo pagos: 1. Duración alquiler. 2. Seguros 3. Sede entrega 4. Sede devolución 5. Conductor adicional 6. Cobro final
Archivo inventario: 1. Vehículo 2. Categoría 3. Donde se encuentra	Archivo log: 1. Vehículo 2. Fecha compra vehiculo 3. Alquileres (Fecha Inicio, Fecha fin) 4. Mantenimientos (Fecha inicio, Fecha fin)	Login/Password: 1. Data base de users y password. 2. Registrarse (usuario, empleado, administrador) 3. Ingresar. 4. Direccionar a la interfaz para usuario, empleado o administrar.

INTERFAZ		
InterfazRenta: <ol style="list-style-type: none"> 1. Renta Vehiculos 2. Paneles 3. Login 4. Cargar Temporada / cargar datos 	Panel login <ol style="list-style-type: none"> 1. Usuario TextField 2. Contraseña TextField 3. Login JButton 4. Registro JButton 	Panel empleado <ol style="list-style-type: none"> 1. Mostrar sedes JButton 2. Devolver carro JButton 3. Salir JButton
Panel Mostrar sedes <ol style="list-style-type: none"> 1. Mostrar info JTextArea 	Panel admin <ol style="list-style-type: none"> 1. Manejar seguros JButton 2. Elegir temporada JButton 3. Agregar carros JButton 4. Agregar sedes JButton 5. Eliminar seguros JButton 	Panel Eliminar Seguros <ol style="list-style-type: none"> 1. Si JButton 2. No JButton
Panel nueva sede <ol style="list-style-type: none"> 1. Nombre JTextField 2. Ubicación JTextField 3. Horario atención JTextField 4. Guardar JButton 	Panel temporada <ol style="list-style-type: none"> 1. temporada JTextField 2. Guardar JButton 	Panel manejar seguros <ol style="list-style-type: none"> 1. Nombre seguro JTextField 2. Precio seguro JTextField 3. Guardar JButton
Panel carro de nuevo <ol style="list-style-type: none"> 1. Placa JTextField 2. Marca JTextField 3. Modelo JTextField 4. Color JTextField 5. Transmision JTextField 6. Capacidad JTextField 7. Sede JTextField 8. Guardar JButton 	Panel Cliente <ol style="list-style-type: none"> 1. Renta Vehiculos 	Panel Alquiler <ol style="list-style-type: none"> 1. Nombre JTextField 2. Telefono JTextField 3. Fecha de nacimiento JTextField 4. Nacionalidad JTextField 5. Doc de identidad JTextField 6. Paisexp JTextField 7. licencia JTextField 8. Fecha de vencimiento licencia JTextField 9. CVV JTextField 10. Banco JTextField 11. Tipo Carro JTextField 12. Sede Entrega JTextField 13. Sede Devolucion JTextField 14. Dias alquiler JTextField 15. Conductor adicional 16. Seguros

Panel Elegir Seguros 1. Seguros JCheckBox	Panel conductor adicional 1. Nombre JTextField 2. Telefono JTextField 3. Fecha de nacimiento JTextField 4. Nacionalidad JTextField 5. Doc de identidad TextField 6. Paisexp JTextField 7. licencia JTextField 8. Fecha de vencimiento licencia JTextField	
---	--	--

Toma de decisiones:

Después de haber hecho la descomposición del problema se explicará la toma de decisiones propuestas en los diagramas.

Se crearon 5 carpetas principales, se realizó una clase vehículo que contiene los atributos de un vehículo que puede ser alquilado, dicha clase se relaciona con la clase categoría que contiene atributos relacionados con las tarifas y los tipos de carros, de esta manera podemos manejar diferentes tarifas por categoría y recargos por conductores adicionales, tenemos una clase Alquiler que tiene los atributos y métodos necesarios para que un cliente pueda realizar un alquiler, y una clase Reserva que hereda a la clase alquiler, por ende hereda sus atributos y métodos, esto lo hicimos debido a que el cliente siempre tendrá la opción de reservar o alquilar, y esencialmente es el mismo proceso salvo algunas excepciones que se definen con los atributos y métodos propios de la clase reserva, de igual forma hicimos otra clase llamada reserva especial que hereda a la clase reserva, por ende, se heredan los atributos y métodos de reserva sumados a sus atributos propios. Ahora para realizar una reserva o un alquiler se necesitan clientes registrados, por ende, tenemos una clase cliente con todos los atributos que debe tener un cliente, esa clase se relaciona con la clase reserva para especificar que pueden haber de 0 a n clientes haciendo una reserva, por otro lado, están las clases datosLicencia y datosTarjetaCredito que se relacionan con la clase clientes, puede haber 1 licencia y 1 tarjeta de crédito por cada cliente y sus atributos referencian la respectiva información de las entidades que se relaciona con los clientes y con los conductores adicionales. Las clases reserva y alquiler se relacionan con la clase categoría para así poder tener la disponibilidad de los tipos de carros.

Se tiene la clase sede, decidimos abstraer de la realidad el nombre para poder distinguir las sedes, la ubicación debido a que se necesita saber en qué lugar se encuentran los vehículos, los horarios de atención son necesarios debido para los clientes saber a qué horas pueden recoger o dejar un automóvil en la sede. Cada sede tiene un administrador local que se encarga de administrar la sede, agregar empleados y expulsarlos, estos

tambien tienen la opcion de crear un nuevo cliente, Por otro lado tenemos a los empleados, los cuales hacen el manejo de todos los carros, estos pueden sacar los carros a mantenimiento y recibirlos, por lo que es necesario tener métodos que logren esa función, entre esos estan sacarDelInventario, actualizarEstado. Cada empleado debe poder ser reconocido por lo que tiene un atributo para esta función, el cual es IdEmpleado, que logra reconocer a los empleados por su número de identificación, el empleado al manipular los vehiculos tiene un vinculo directo con la clase vehiculo, por ultimo se tiene un administrador general, el cual se encarga de agregar nuevas sedes o gestionarlasy dependiendo si se quiere cambiar ciertos aspectos de la sede, dentro de la empresa solo existe un administrador general por lo que las cardinalidades que este tiene con otras clases sólo puede ser 1.

En última instancia, abordamos la clase denominada "Interfaz" en nuestro sistema. Su propósito inherente radica en la creación de una interfaz visual estéticamente agradable y que facilite la utilización de la aplicación. Se inicia con un panel general, conocido como el Panel de Interfaz de Renta, que, a su vez, da inicio al Panel Login. Dependiendo de las credenciales del usuario, se activan tres paneles distintos.

En primer lugar, discutamos el Panel de Empleado. En este espacio, los usuarios tienen la capacidad de realizar la devolución de vehículos, posteriormente se inicializa un panel adicional que permite la introducción de datos relevantes necesarios para completar la devolución del automóvil. Este mismo panel despliega información sobre las sedes de alquiler de vehículos.

El segundo panel principal, generado a partir del inicio de sesión, es el Panel de Administrador. Aquí se pueden agregar nuevos automóviles, desencadenando la aparición de un panel adicional que solicita la información pertinente para la incorporación del vehículo. Además, se incluye la opción de cambiar la temporada, dando lugar a otro panel que permite la modificación correspondiente. En relación con los seguros, se despliegan dos paneles, según la acción deseada: "Eliminar Seguros", para la eliminación de pólizas, y "Agregar Seguros", para su incorporación. Asimismo, se presenta un panel para la inclusión de una nueva sede, solicitando los datos necesarios para tal fin.

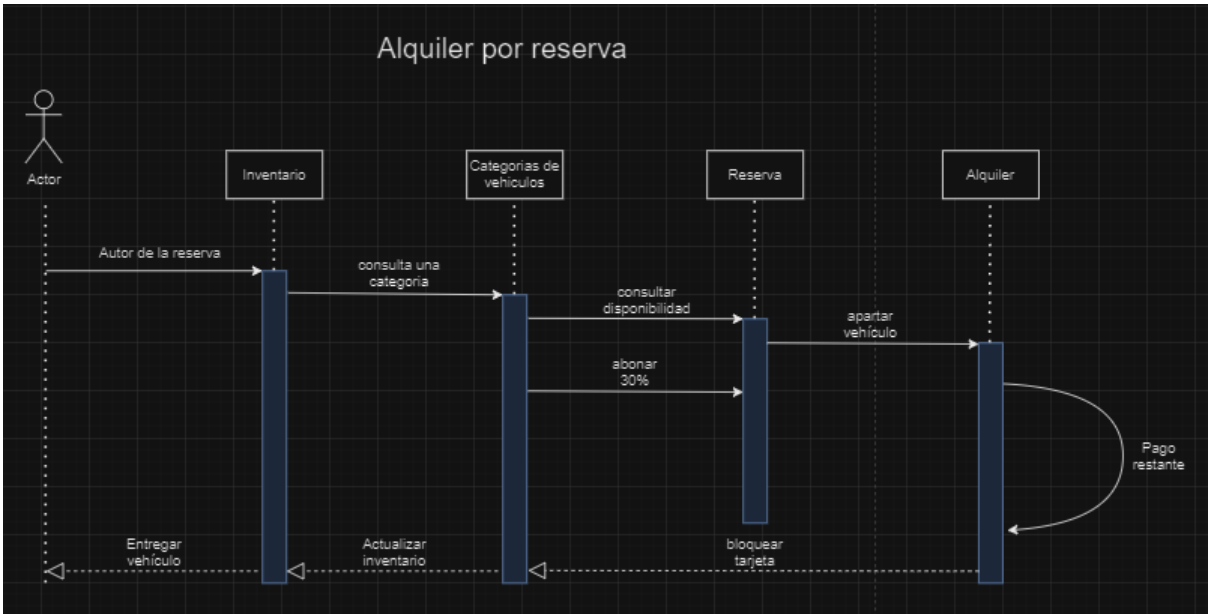
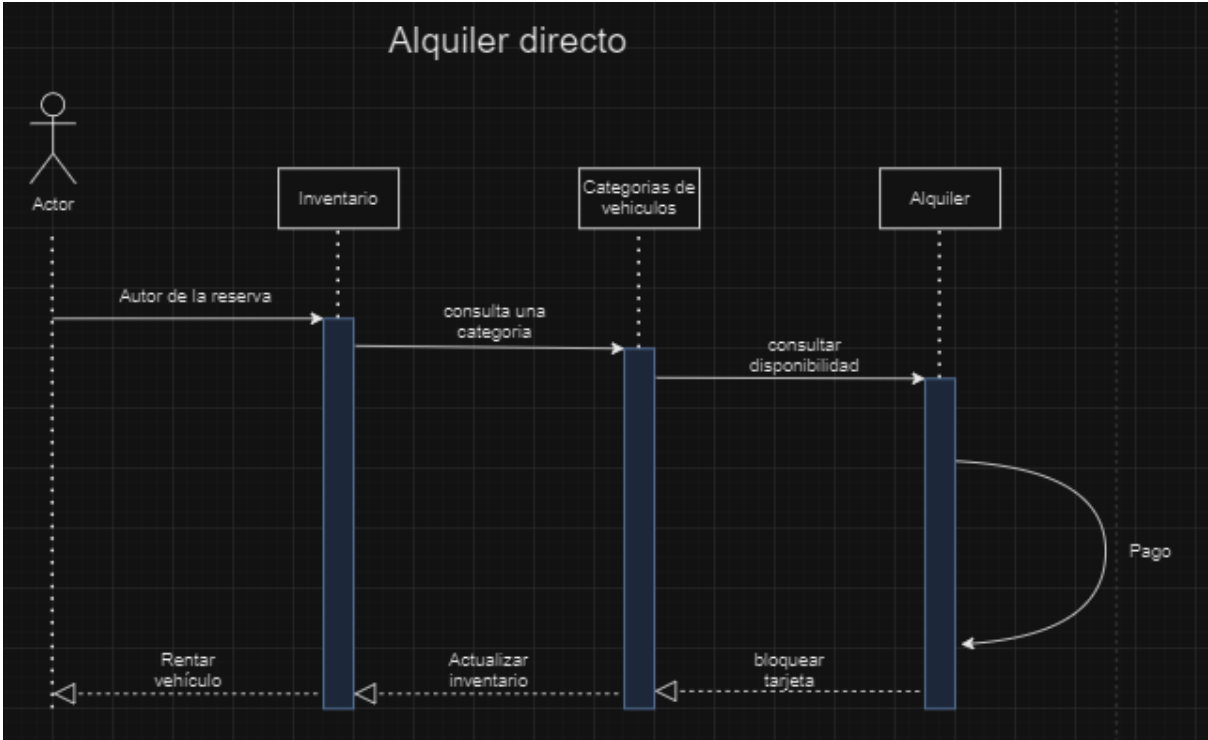
Finalmente, el Panel de Inicio de Sesión inicializa el Panel de Clientes. Dentro de este último, los usuarios tienen la capacidad de realizar reservas o alquileres, activando el Panel de Alquiler. Aquí, se solicitan al cliente los datos esenciales para llevar a cabo el alquiler de un vehículo. Una vez realizada la reserva o alquiler, se inicia un panel que presenta las opciones de seguros disponibles, brindando la posibilidad de seleccionar aquellos necesarios. Finalmente, se muestra el precio al completar la introducción de datos.

Decisiones de diseño proyecto 3:

Para el proyecto 3 se realizaron los siguientes cambios: Se importó y se hizo uso de la librería iText para poder crear y manejar documentos PDF dentro del proyecto, para ello se creo una carpeta lib dentro del proyecto para almacenar los archivos JAR necesarios para el funcionamiento de la librería, por tanto se agregaron 2 métodos en la clase Renta Vehículos los cuales se encargan de generar la factura de un alquiler o de una reserva, y la factura se crea en la carpeta data con el nombre de factura.pdf. Posteriormente se crearon 4 clases en

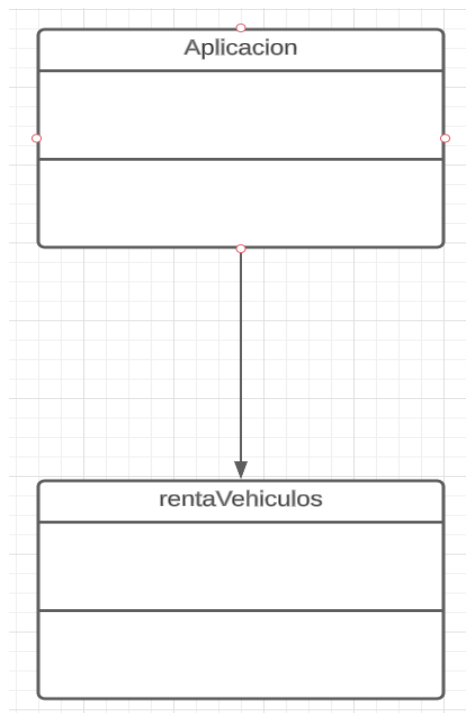
el paquete archivo: ArchivoPaypal, ArchivoPayU, ArchivoSire y ArchivoTarjeta, en donde se maneja todo lo relacionado con la Pasarela de Pagos, también, en el modelo se crearon 2 clases: EleccionPasarela y PasarelaPago, en donde se hizo uso de la librería fasterxml para poder guardar los datos de las transacciones en un archivo json, específicamente para el método de Sire, donde se almacena el precio, el número de la tarjeta, y el nombre del cliente para poder realizar el pago, para los demás métodos de pago, la información se almacena en la su respectivo archivo.txt en la carpeta data. En relación a la implementación, también se creó una clase PanelPago que fue agregada a interfazRenta, para que posterior a que el cliente reciba la información de su alquiler o de su reserva, salga un panel que le permita elegir el método de pago que desea. Por otro lado, se creó otro paquete programaCliente en donde se creó una clase AplicacionCliente, esta es una aplicación que está dedicada solamente al uso del cliente, por tanto, no tiene acceso a las funcionalidades del administrador ni del empleado, únicamente a las funcionalidades del cliente. Por último, se creó un nuevo paquete pruebas en donde se crearon 3 clases: AlquilerTest, ReservaTest y carga CargaArchivosTest, allí se realizan algunas pruebas unitarias y de integración con el framework JUnit que fue importado al proyecto, en CargaArchivosTest se realizan pruebas unitarias relacionadas a la carga de datos, por tanto se verifica que las sedes y los vehículos se creen y se carguen correctamente a partir de los datos en los archivos de prueba, en la clase AlquilerTest se prueban todos los métodos importantes de la clase Alquiler, es decir, todos los relacionados a el precio de un alquiler, decidimos hacer esto debido a que con probar estos métodos podemos hacer pruebas unitarias y de integración respecto a la creación y el funcionamiento de un alquiler, para asegurarnos de que todos los aspectos que hay que tener en cuenta a la hora de la creación de un alquiler se cumplan, al asegurar que el precio total de un alquiler da bien, podemos asumir que todos los aspectos están funcionando correctamente a la hora de su creación, por tanto, dado que la implementación de alquiler y reserva es muy similar no hace falta probar todo 2 veces, por eso en ReservaTest solo se prueba el metodo del cobrolnicial que se debe tener en cuenta a la hora de realizar una reserva.

Diagramas de secuencia:

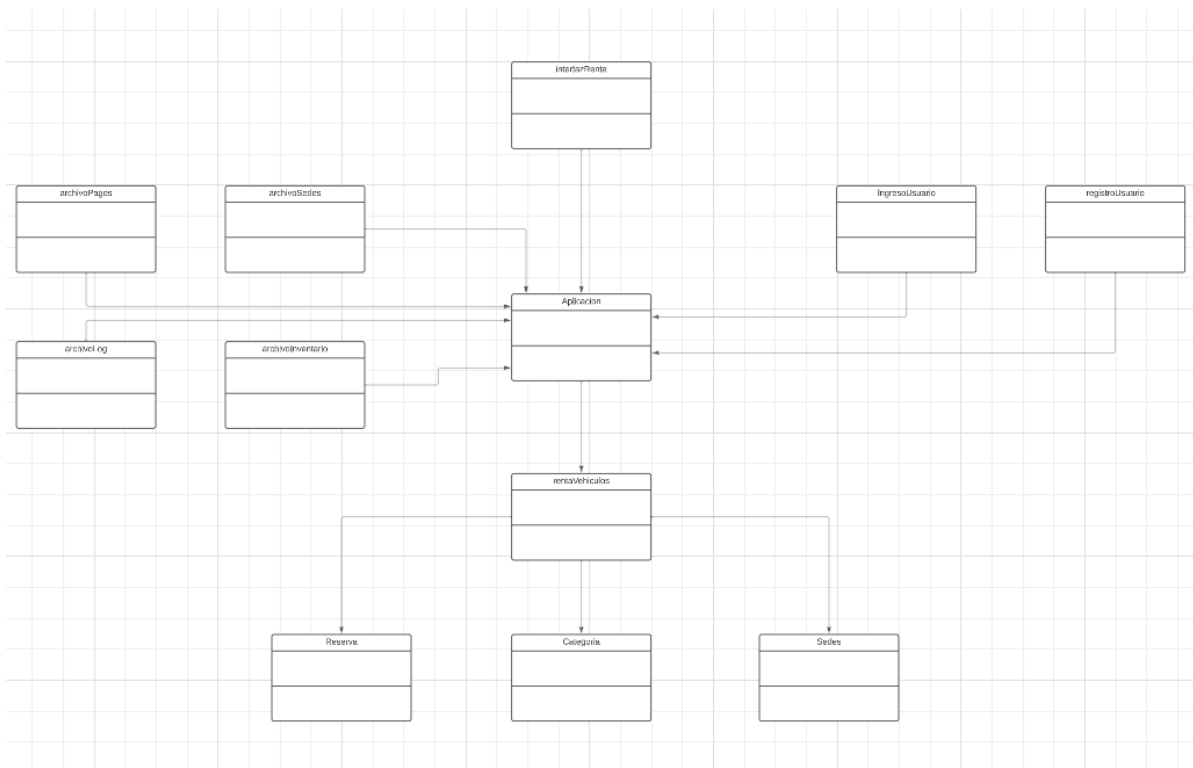


Descomposición:

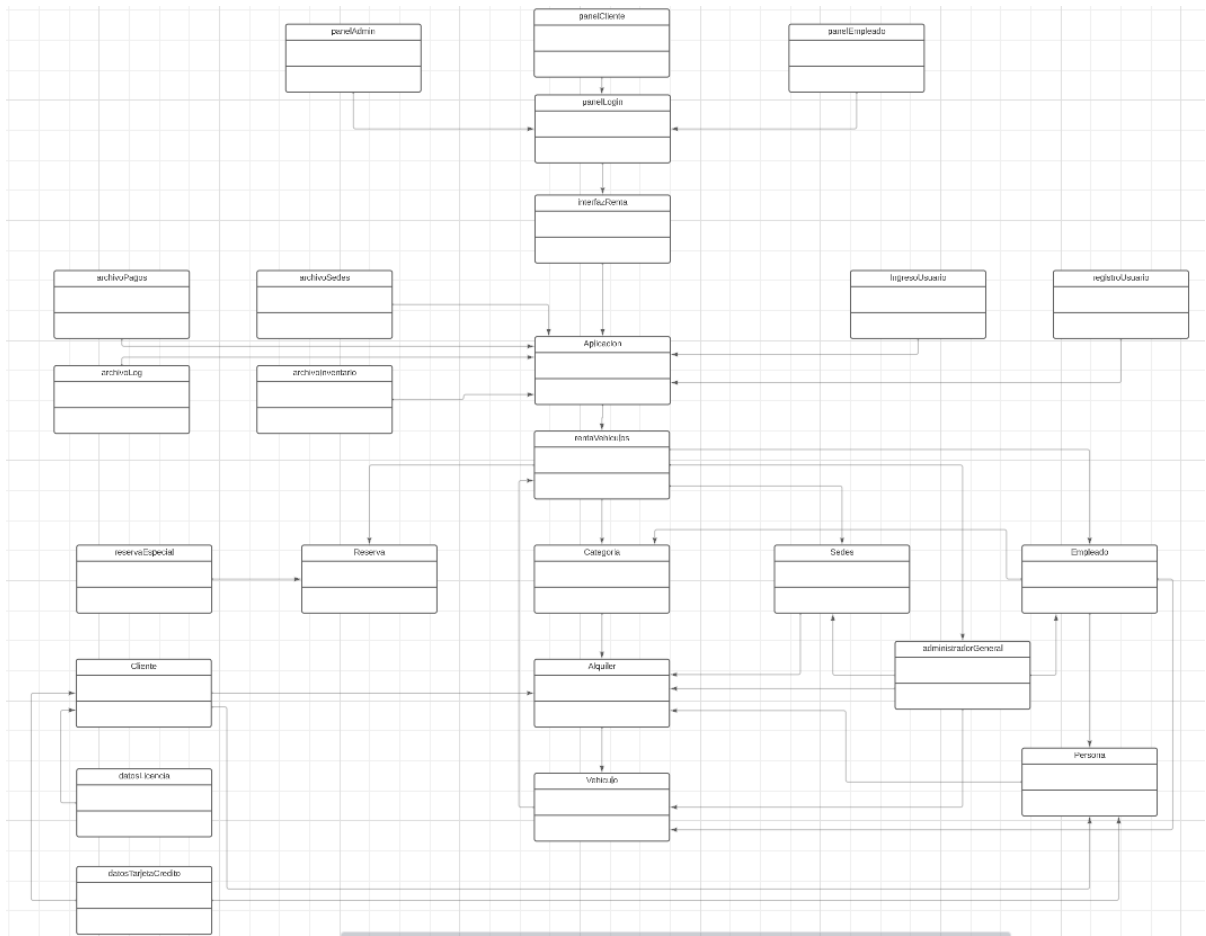
1.



2.



3.



4.

