DOCUMENTO DE DISEÑO PROYECTO 3 – ENTREGA ÚNICA

Martín Andrés Del Gordo Caballero - 202024239 Juan Martin Vásquez - 202113314

Daniel Piñeros – 202013147

**Instrucciones de uso para cada usuario:**

* Usuario Cliente:

El usuario Cliente empieza por ejecutar el main de clase VClienteAPP, para luego ingresar su respectivo usuario y contraseña. Este también tiene la opción de crear una cuenta nueva. Luego, se le despliega un menú en el que puede ver la disponibilidad de una sede o crear una nueva reserva. Para ver la disponibilidad de una sede, basta con seleccionar la sede que quiere consultar para luego dar click en el botón MOSTRAR DIPO. Para crear una reserva, el cliente debe especificar la categoría que desea reservar, la sede de reclamo, la sede de devolución, la fecha de reclamo (en formato yyyy-MM-dd), fecha de devolución (en formato yyyy-MM-dd), la hora a la que va a retornar el vehículo (en formato hh:mm:ss), los seguros que desea tener y el login de los conductores adicionales.

* Usuario Empleado:

El usuario Empleado empieza por ejecutar el main de la clase VLogIn, y luego ingresa su respectivo usuario y contraseña. Luego, puede escoger alguna de las siguientes opciones: entregarle el carro a un Cliente, recibir un carro de un Cliente (devolverlo) o cambiar la disponibilidad de un carro. Para entregar y devolver un carro, solo se debe seleccionar el ID de este. Para cambiar la disponibilidad, primero se debe indicar la fecha (en formato yyyy-MM-dd) a partir la cual el vehículo no va a estar disponible, además del número de días en los que no lo estará y el ID de este.

* Usuario AdminGeneral:

El usuario AdminGeneral empieza por ejecutar el main de la clase VLogIn, y luego ingresa su respectivo usuario y contraseña. Luego, puede escoger alguna de las siguientes opciones: eliminar un carro, crear un nuevo carro, eliminar un seguro, crear un nuevo seguro, editar un seguro existente, cambiar el horario de una sede, cambiar la temporada y general un archivo Log. Para eliminar un vehículo, solo se debe seleccionar su ID, mientras que para crear uno se debe especificar: su placa, modelo, marca, color, precio, seleccionar su tipo de transmisión, categoría y sede en la que se va a encontrar. Para eliminar un seguro basta con seleccionar uno de los ya existentes y dar click en el botón “Eliminar Seguro”, mientras que para crear uno se debe digitar su nombre y su precio. Para editar un seguro solo se debe seleccionar uno de los ya existentes y digitar el nuevo precio que se le quiera asignar. Para cambiar el horario de una sede, primero se debe seleccionar la sede que se desea modificar, luego escribir “0” si se le quiere cambiar su horario de apertura o “0” para cambiar su horario de sede y, por último, la nueva hora (en formato hh:mm:ss). Para cambiar la temporada solo se debe seleccionar True para temporada alta y False para baja. Finalmente, para generar un archivo Log se debe seleccionar el ID del vehículo para el cual se quiere realizar.

* Usuario AdminLocal:

El usuario AdminLocal empieza por ejecutar el main de la clase VLogIn, y luego ingresa su respectivo usuario y contraseña. Luego, puede seleccionar crear un nuevo Cliente o crear un nuevo Empleado. Para lo primero, debe digitar: su nombre, su apellido, fecha de nacimiento (en formato yyyy-MM-dd), nacionalidad, login, contraseña, numero de tarjeta de crédito, cvv, ID de su licencia de conducción, país de origen, fecha de vencimiento de su licencia, foto de la persona (URL) y teléfono. Para crear un Empleado, se debe señalar su nombre, apellido, fecha de nacimiento (en formato yyyy-MM-dd), nacionalidad, usuario y contraseña.

**Login y contraseña para cada tipo de usuario:**

* Usuario Cliente:
  + Login: z.zidane; contraseña: Zidane
* Usuario AdminLocal:
  + Login: c.valderrama; contraseña: Valderrama
* Usuario AdminGeneral:
  + Login: l.yamal; contraseña: Yamal
* Usuario Empleado:
  + Login: l.messi; constraseña: Messi

**Clases:**

Aplicación: La clase aplicación es la clase central del sistema, esta clase funciona como herramienta central sobre la que se va a construir el resto del programa. Sus atributos son: una lista con los vehículos alquilados, inventario de vehículos no alquilados, una lista con las sedes. Esta clase también es la encargada de realizar la carga de datos a través de sus métodos.

Vehículo: Esta clase funciona principalmente como la estructura base de las características que deben tener todos los vehículos de la empresa. Sus atributos son: el id del vehículo, su placa, marca, modelo, color, tipo de transmisión, precio, la categoría a la que pertenece, la sede en la que se encuentra, el cliente que tiene el vehículo y su disponibilidad. Para la disponibilidad, es una ArrayList en donde cada posición representa un dia y cada una pude tomar el valor de 1 si el vehículo está disponible o 0 de lo contrario. Sus métodos constan de un constructor, así como de getters y setters para sus distintos atributos.

Reserva: Esta clase contiene la información relevante a la que se debe tener acceso sobre una reserva, Sus atributos son: el Cliente que hizo la reserva, el MedioPago con el que la realizará, el monto a pagar, la sede de devolución del alquiler, la fecha para la cual se hizo la reserva, la hora a la que se va a regresar el carro, el carro que se le asignó a la reserva (toma el valor de Null hasta que un empleado asigne un vehículo, una lista de los conductores adicionales que pueden recoger el carro, una lista con los seguros de la reserva y la fecha de creación de esta. Sus métodos constan de un contructor y de getters y setter para sus atributos. La clase reserva también puede revisar la disponibilidad de una categoría y retornar un vehículo dado la sede de devolución de este, así como calcular el costo total que tendrá el realizar una reserva.

Sede: Esta clase guarda la información de cada sede de la empresa. Sus atributos son: el nombre de la sede, la hora de apertura, hora de cierre, una lista de los empleados que trabajan allí, el empleado local encargado de ella y una lista con los autos que tiene la sede. Sus métodos incluyen un constructor, así como getters y setters para los atributos de la clase.

Persona: Esta clase representa a los diferentes usuarios que puede tener la aplicación. Sus atributos son: su nombre, apellido, fecha de nacimiento, nacionalidad y su login a la aplicación con su respectiva contraseña. Sus métodos constan de un constructo, setters para todos sus atributos y getters para atributos cuya información es necesaria acceder como su nombre, apellido, login y password.

AdminGeneral: Esta clase hereda de la clase persona y representa al usuario de administrador general de la aplicación. Esta clase no tiene atributos pues toda su información necesaria es la que hereda de Persona. Puede realizar los requerimientos de agregar y/o eliminar los vehículos de la empresa.

AdminLocal: Esta clase hereda de la clase persona y representa al usuario de administrador local de una sede. Sus atributos son: un booleano que indica si el empleado sigue activo con la empresa y la sede en la que trabaja. Sus funcionalidades son las de crear un nuevo empleado y cambiar la hora de apertura/cierre de una sede.

Empleado: Esta clase hereda de la clase persona y representa al usuario empleado de la empresa. Sus atributos son: un booleano que indica si el empleado sigue activo con la empresa, la sede en la que trabaja y una lista con los carros entregados por ese empleado. Los empleados pueden asignarle un vehículo a una reserva, asignarle a una reserva un vehículo y retornar un vehículo que había sido alquilado.

Cliente: Esta clase hereda de la clase persona y representa al usuario cliente de la aplicación. Sus atributos son: una lista con sus medios de pago, su licencia de conducción, una foto del cliente en formato URL y su teléfono. Los clientes tienen la capacidad de realizar reservas. Los conductores adicionales entran también dentro de esta clase (se toma la lista de métodos de pago como una lista vacía).

Licencia: Esta clase contiene la información de una licencia de conducción. Sus atributos son: el id de la licencia, el país del que procede, la fecha de vencimiento y una foto en formato URL. Sus métodos son un contructor así como sus respectivos getters y setters de los atributos.

MedioPago: Esta clase contiene la información de un método de pago. Sus atributos son: el número de la tarjeta, un booleano que representa el estado de la tarjeta y el cvv de la tarjeta. Sus métodos son un contructor así como los setters de cada uno de sus atributos y un getter para el atrubuto estadoTarjeta(pues es el único que es necesario acceder.

Seguro: Esta clase contiene la información de un seguro que se le ofrece al cliente. Sus atributos son: el nombre del seguro y el costo extra que este genera. Sus métodos son un contructor así como sus respectivos getters y setters de los atributos.

**Clases interfaz gráfica:**

Las clases que constituyen la interfaz gráfica de la aplicación se encuentran en el paquete GUI y son:

* PIqDisp: Esta clase contiene el JPanel que despliega y colorea una matriz usada dentro de la clase VconsultarDispo.
* VAdminGeneral: Esta clase contiene el JPanel que muestra todas las funcionalidades que puede realizar un AdminGeneral, representadas a través de botones.
* VAdminLoc: : Esta clase contiene el JPanel que muestra todas las funcionalidades que puede realizar un AdminLocal, representadas a través de botones.
* VCambiarDisponibilidad: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de Empleado de cambiar la disponibilidad de un vehículo.
* VCambiarHorario: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminLocal de cambiar el horario de una sede.
* VCambiarTemporada: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de cambiar la temporada (alta o baja).
* VCliente: Esta clase contiene el JPanel que muestra todas las funcionalidades que puede realizar un Cliente, representadas a través de botones.
* VconsultarDispo: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de Cliente de consultar la disponibilidad de una sede.
* VCrearLog: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de crear un archivo Log con el historial de un vehículo.
* VcrearReserva: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de Cliente de crear una nueva reserva.
* VDevolverCarro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de Empleado de regresar un vehículo a una sede.
* VEditarSeguro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de editar el precio de un seguro ya existente.
* VEliminarSeguro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de eliminar un seguro existente.
* VEliminarVehiculo: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de eliminar un vehículo.
* VEmpleado: Esta clase contiene el JPanel que muestra la todas las funcionalidades que puede realizar un Empelado, representadas a través de botones.
* VEntregarCarro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de Empleado de entregar un vehículo a un Cliente.
* VLogIn: Esta clase contiene la interfaz en la que un Empleado, AdminLocal o AdminGeneral pueden ingresar su usuario y contraseña.
* VNuevoCarro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de crear un nuevo vehículo.
* VNuevoCliente: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminLocal de crear un nuevo usuario de tipo Cliente.
* VNuevoEmpelado: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminLocal de crear un nuevo usuario de tipo Empleado.
* VNuevoSeguro: Esta clase contiene el JPanel que muestra la interfaz para la funcionalidad de AdminGeneral de crear un nuevo seguro.

En el siguiente diagrama se observa de mejor manera la relación entre las diferentes clases de la interfaz gráfica:

Diagrama

Descripción generada automáticamente

**Clases adicionales (proyecto 3):**

* FacturaPDF: Como su nombre lo indica, esta clase se encarga de tomar una reserva (de la clase Reserva) y genera un PDF con el recibo de pago que debe realizar el Cliente. Es importante notar que este recibo se le entrega al cliente al momento que este recibe el carro o, en otras palabras, cuando el administrador local de la sede le hace entrega del vehículo.
* VClienteAPP: Esta clase contiene el main que debe ser ejecutado por un Cliente para acceder a la aplicación, haciendo uso de su usuario y de su contraseña.

# Requerimientos funcionales (Método historias de usuario):

**Inventario de vehículos**

* **Historia de usuario:** Como administrador, se quiere poder registrar un nuevo vehículo.
  + Descripción: El administrador debe poder proporcionar la información básica del vehículo, incluyendo la placa, la marca, el modelo, el color, el tipo de transmisión y la categoría.
  + Criterios de aceptación: El vehículo debe ser agregado al inventario de la empresa.
* **Historia de usuario:** Como administrador, se quiere poder actualizar la información de un vehículo existente.
  + Descripción: El administrador debe poder actualizar la información básica del vehículo, incluyendo la placa, la marca, el modelo, el color, el tipo de transmisión y la categoría.
  + Criterios de aceptación: La información del vehículo debe ser actualizada correctamente.
* **Historia de usuario:** Como administrador, se quiere poder dar de baja un vehículo.
  + Descripción: El administrador debe poder marcar un vehículo como dado de baja.
  + Criterios de aceptación: El vehículo debe ser eliminado del inventario de la empresa.
* **Historia de usuario:** Como cliente, se quiere poder ver la disponibilidad de los vehículos en tiempo real.
  + Descripción: El cliente debe poder ver una lista de los vehículos disponibles, indicando la categoría, la sede de ubicación y la fecha de disponibilidad.
  + Criterios de aceptación: El cliente debe poder ver la disponibilidad de los vehículos de forma actualizada.

# Reservas y alquiler

* **Historia de usuario:** Como cliente, se quiere poder reservar un vehículo.
  + Descripción: El cliente debe poder seleccionar el tipo de vehículo, la sede de recogida, la fecha y hora de recogida, la sede de devolución y la fecha y hora de devolución.
  + Criterios de aceptación: las reservas se deben crear y relacionar con el perfil que se tenga el cliente en el sistema teniendo presente la disponibilidad de vehículos en la sede fecha hora y categoría del vehículo solicitadas.
* **Historia de usuario:** Como cliente, se quiere poder alquilar un vehículo con ayuda de un empleado.
  + Descripción: para poder realizar una reserva el cliente primero de facilitar varios datos al empleado con el fin de primero crear su propio perfil dentro del sistema, luego de dicho proceso se realizará la reserva teniendo en cuenta la disponibilidad de vehículos y sus categorías, las fechas de disponibilidad de los

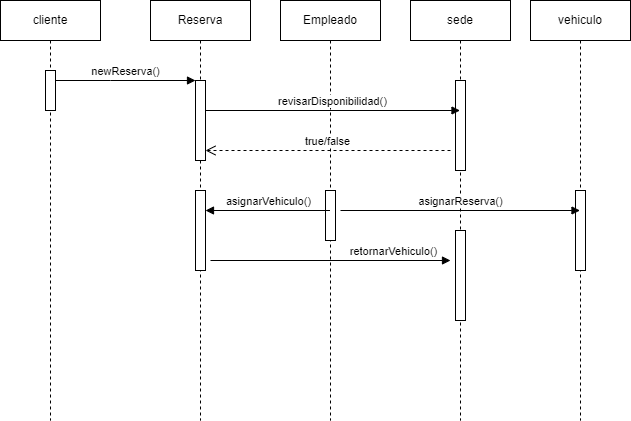
mismos y por parte del empleado saber si es o no temporada alta en dicho momento. Al ya tener la reserva hoy proyecta por parte del cliente se debe cancelar un 30% del valor total para poder formalizar la reserva tanto para el cliente como la para la compañía.

* + Criterios de aceptación: El vehículo debe ser asignado al cliente y el cliente luego del pago correspondiente al 30% del valor total de la reserva confirma su reserva con los parámetros acordados.
* **Historia de usuario:** Como empleado, se quiere poder gestionar las reservas y los alquileres.
  + Descripción: El empleado debe poder ver una lista de las reservas y los alquileres, actualizar el estado de las reservas y los alquileres, y cancelar las reservas y los alquileres.
  + Criterios de aceptación: Las reservas y los alquileres deben ser gestionados correctamente.

# Sedes

* **Historia de usuario:** Como administrador, se quiere poder registrar una nueva sede.
  + Descripción: El administrador debe poder proporcionar la información básica de la sede, incluyendo el nombre, la ubicación y los horarios de atención.
  + Criterios de aceptación: La sede debe ser agregada a la lista de sedes de la empresa.
* **Historia de usuario:** Como administrador, se quiere poder actualizar la información de una sede existente.
  + Descripción: El administrador debe poder actualizar la información básica de la sede, incluyendo el nombre, la ubicación y los horarios de atención.
  + Criterios de aceptación: La información de la sede debe ser actualizada correctamente.
* **Historia de usuario:** Como administrador, se quiere poder eliminar una sede.
  + Descripción: El administrador debe poder marcar una sede como dada de baja.
  + Criterios de aceptación: La sede debe ser eliminada de la lista de sedes de la empresa.
* **Historia de usuario:** Como administrador, se quiere poder registrar y gestionar la información de los empleados de las sedes.
  + Descripción: El administrador debe poder registrar la información básica de los empleados, incluyendo el nombre, el cargo, la información de contacto y la información de pago.
  + Criterios de aceptación: La información de los empleados debe estar actualizada y completa.
* **Historia de usuario:** Como empleado de la sede, se quiere poder actualizar el estado de los vehículos.
  + Descripción: El empleado debe poder indicar que un vehículo está disponible, en mantenimiento o en reparación.
  + Criterios de aceptación: El estado del vehículo debe ser actualizado correctamente.
* **Historia de usuario:** cómo empleado de la sede, hoy se quiere registrar una reserva en temporada alta de un vehículo con una categoría que no cuenta con disponibilidad en la sede en la que se solicita.
  + Descripción: el empleado debe indicar que es temporada alta para poder calcular los costos con exactitud, seguidamente ingresar en el sistema la categoría solicitada por el cliente y éste le indicará hoy que se le dará acceso a un vehículo de una categoría superior debido a la no disponibilidad del vehículo en la categoría inicialmente solicitada.
  + Criterios de aceptación: el sistema debe ser capaz de calcular con precisión el 30% correspondiente al total del valor a pagar por parte del cliente en la reserva teniendo presente el estado activo de temporada alta, adicionalmente como no hay disponibilidad de la categoría inicialmente solicitada por el cliente se debe instanciar a una categoría superior sin realizar ningún tipo de cobro adicional y relacionando un vehículo disponible en la sede solicitada para el cliente en las fechas que él ha solicitado de la reserva.

# Diagrama de Secuencia:



En este diagrama se pueden observar varios requerimientos funcionales. Por ejemplo, en el principio se puede observar la interacción que se asegura que siempre vaya a haber un vehículo disponible a la hora de hacer una reserva. Otro requerimiento que se expone es el empleado asignado un carro a una reserva, además se decidió que el empleado también asignara un cliente a un carro.

# En el siguiente párrafo se describirá una serie de restricciones y modificaciones hechas a la aplicación original:

La aclaración principal consiste en el horizonte de reservas en la aplicación creada. El horizonte consiste en un periodo válido de 30 días a partir de la fecha actual, 16-10-2023. Con el fin de simplificar las relaciones entre los datos de los archivos en el programa, las reservas se definen directamente a partir de la fecha de reserva para recogida y la fecha de reserva para devolución, así como cada una con sus debidas horas. Estos valores, en conjunto con la disponibilidad particular de cada vehículo y su respectiva categoría, permitirán emparejar cada una de las reservas con la disponibilidad de vehículos presente en cada sede.

Es importante aclarar que, en el proceso de reserva que realiza el empleado, es él quien indica si es o no temporada alta en este momento. En el programa, esto se interpreta como un booleano que determina si el precio será uno u otro con el fin de proporcionar el precio a cancelar por parte del cliente en el momento de la reserva. Debido a que el costo para cancelar es un valor calculado, se debe tener en cuenta el número de días de reserva, la categoría del vehículo y el valor que determina si es temporada alta o no. El conjunto de las variables anteriormente mencionadas se usa para indicar a qué corresponde el 30% del valor total que debe ser cancelado en el momento de la reserva por parte del cliente.

Se debe aclarar que el programa realizado cuenta con una restricción para el método de pago. Esta restricción consiste en que el método de pago debe ser por tarjeta y sólo puede

ser uno, debido a que este dato se relaciona con cada uno de los clientes, excepto si es un conductor adicional de un cliente por sistema.

El administrador general tendrá 3 capacidades básicas para poder gestionar seguros y horarios en sedes. Estas capacidades son la creación y eliminación de seguros para poder ofrecer a cada cliente. Por último, pero no menos importante, tendrá la capacidad de gestionar los horarios por cada una de las sedes de la compañía.