DATOS GENERALES

Programa: Ingeniería de Sistemas

Periodo académico: 2022-2

Semestre en curso: 4

Grupo de trabajo: 4

Fecha de entrega del informe: 09/11/2022

Nombre del proyecto:

**SISTEMA DE ALQUILER DE AUTOS PARA RENTACAR INTERNACIONAL**

Estudiantes:

Michael Fernney Pérez Ochoa, código:1005235157

Miguel Angel Becerra Solano, código: 1005371895

Jheison Alexander De la rosa Pineda, código: 1005290319

INTRODUCCIÓN

A lo largo de este proyecto se buscará construir un sistema de software para la empresa Rentacar internacional, que le permita realizar sus funciones, gestionando la información relacionada con el alquiler, compra, venta los vehículos, para esto, se emplearán tres áreas de conocimiento, donde cada una ayudara con una parte específica del proyecto. El análisis y la planificación se llevarán a cabo con la metodología que nos plantea la ingeniería de software. El diseño y la implementación estarán bajo la pauta del área de programación. La recolección, almacenamiento y gestión de los datos estarán bajo la estructura de bases de datos.

Todo esto para que al final del proyecto se refleje todo lo aprendido en las diferentes áreas para que el programa a realizar cumpla con las expectativas del cliente, que es que se gestionen las actividades de la empresa y que dicho programa inserte, modifique y borre datos que sean de la conveniencia de los usuarios que la manipulen.

GLOSARIO

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

**Ingeniería de software:** La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos.

**Base de datos:** Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.

**Programar:** Establecer o planificar el programa de una serie de actividades.

**Gestor de base de datos:** Es un conjunto de programas que permite administrar y manipular la información de la base de datos.

**Árbol de refinamiento funcional (ARF):** Es el modelo que descompone el sistema en grupos funcionales, sub-grupos funcionales y funcionalidades que tiene el sistema.

**Léxico orientado al modelado (LOM):** Es el modelo que descompone en los posibles objetos, acciones, estados y atributos en un sistema.

**Dominio (BD):** Un dominio describe un conjunto de posibles valores para cierto atributo.

**Clase:** Una clase es una plantilla para el objetivo de la creación de objetos de datos según un modelo predefinido.

**Entidades:** Son los objetos de un sistema a los que se les quiere almacenar información.

**Atributos:** Son los campos con los que se llena de información cada entidad.

**Diagrama de transición de estados (DTE):** Muestran las transiciones de los estados que pueden tener los procesos dentro de su ciclo de vida dentro de un sistema.

**Diagrama de clases:** Es un diagrama que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones y las relaciones entre los objetos.

CARACTERÍSTICAS DE LAS NECESIDADES DEL PROYECTO

La empresa Rentacar Internacional tiene en sus manos una gran labor de gestión de información, para poder prestar un buen servicio a sus clientes y no tener inconvenientes, ellos deben saber exactamente qué automóviles de su flota están disponibles para el alquiler, y cuáles no lo están, los automóviles pueden no estar disponibles por varias razones, pueden estar, en mantenimiento, reparación, vendidos, alquilados o por último fuera de cobertura del seguro.

Rentacar Internacional debe tener claras las tarifas de cada uno de los automóviles según su grupo y categoría, así como también, de su seguro y extras, esto para poder informar al cliente de lo que debe pagar a la hora de llegar a un acuerdo, teniendo en cuenta que a lo largo de la duración del contrato el cliente puede solicitar extras por períodos determinados.

También deben llevar la cuenta de lo que cada uno de los clientes debe pagar al finalizar el contrato para poder hacer el respectivo cobro, todo esto debe quedar registrado, desde los datos del cliente al que se le realizó el alquiler si fue directo, o los datos de la empresa intermediaria, así como todos los datos del automóvil y su alquiler.

En cuestión de sus automóviles, la empresa encargada debe tener todos los datos actualizados, esto para poder habilitarlos o deshabilitarlos en cualquier momento, además de presentar el vehículo a los posibles compradores. Llevar esta cantidad de información de manera humana puede llevar a muchos errores que la empresa no puede permitir si quiere brindar el mejor servicio posible para seguir creciendo, por esto, es necesario automatizar gran parte de este sistema, facilitando la gestión empresarial, minimizando el personal requerido, aumentando la seguridad de los procesos, asi como tambien la velocidad de estos mismos. Los beneficios a largo plazo serán mucho mayores que la inversión realizada. Todo lo expuesto anteriormente nos permite concluir que el proyecto es factible.

OBJETIVOS

**Objetivo general**

Construir un sistema de software para el fácil manejo de los procesos de la empresa Rentacar internacional.

**Objetivos Específicos**

1. Analizar los requerimientos de la empresa Rentacar internacional, para la comprensión del problema.
2. Planificar el desarrollo del sistema para que este sea ordenado y eficaz.
3. Diseñar las soluciones para los requerimientos establecidos.
4. Implementar el software para el avance del desarrollo del sistema.
5. Recolectar la información necesaria para el funcionamiento del sistema.
6. Probar los datos recolectados en el sistema para la verificación del correcto funcionamiento del sistema.

JUSTIFICACIÓN

La razón por la que se realiza este proyecto es para abastecer los problemas planteados por la empresa Rentacar, entre ellos el que se pueda consultar, modificar y almacenar información que se ingrese a la empresa Rentacar, esto nos lleva a una investigación para saber cómo funciona la empresa y qué necesidades tiene, con el fin de conseguir por medio de esta investigación crear las clases, entidades y los atributos que nos sirvan para la creación de los diferentes modelos y la base de datos, a parte de esto, se debe realizar un programa en el que primero los clientes y administrativos se puedan registrar y acceder al sistema Rentacar, en el que podrán acceder al catálogo de autos, seguros, compañía de seguros y contrato, dentro de este podrán alquilar autos y realizar otras funciones implementadas en el sistema como pueden ser consultar y almacenar los datos que se ingresen por un usuario administrativo.

Además se debe hacer investigación con el fin de tener conocimiento de que tiene un sistema para el alquiler de autos, tener una base y poder realizar un sistema propio.

Todo esto le ayuda a los desarrolladores a unificar los diferentes desarrollos y competencias que se aprenden en las diferentes áreas las cuales son programación, base de datos e ingeniería de software.

MARCO DE REFERENCIA

**Marco Teórico**

Para la realización de este proyecto las materias que debemos abordar son base de datos, programación e ingeniería de software de estas se debe tener una base de los siguientes temas: en la base de datos lo que se va a utilizar es, primero que todo la creación del modelo entidad-relación el cual nos permitirá organizar la información obtenida para crear entidades y atributos que utilizaremos en el desarrollo de nuestra base de datos, luego se creará el modelo relacional el cual nos ayudará más adelante y nos permitirá hacer la realización del diccionario de datos el cual servirá para saber cuales son los tipos de datos que se deben ingresar a la base de datos para cada uno de los atributos y tener una idea de su dominio y tipo de llave si se requiere, se procedera a crear las tablas en la base de datos y a buscar y ordenar informacion para abasteserla, con esto ya tendríamos la base para la creacion de esta además de esto debemos manipular la información suministrada a la base de datos por medio de la interfaz de tablas o directamente en oracle, con la ayuda de lo visto en programación se logrará el desarrollo de una aplicación de escritorio en el que se creara un login por medio del cual el usuario se registrara y ingresara a una interfaz con las funcionalidades que se nos asignaron, dentro de esta podra borrar, crear y modificar datos todo esto se realizara en el lenguaje c# y por último en ingeniería de software se añadirá el léxico orientado a objetos, árbol de refinamiento funcional, casos de usos, diagramas de secuencia y diagramas de clases, incluyendo lo visto en ingeniería de software se realizará la creación de todos los modelos necesarios ya creados y faltantes para el desarrollo del programa.

**Marco tecnológico**

las tecnologías que se van a utilizar para la realización del proyecto son las que se utilizaran para la creación de lo propuesto, para la creación de la base de datos se va a emplear oracle o vertabelo y para la aplicación de escritorio se utilizará c#. Además de esto para la presentación de los modelos entidad-relación , relacional y modelos de ingeniería de software se utilizará la plataforma Miro.

**Marco normativo (Legal o jurídico)**

Según la universidad externado de colombia EMI (11 de septiembre 2016).Departamento de Propiedad Intelectual.<https://propintel.uexternado.edu.co> la normativa colombiana que regula la protección del software en Colombia es la ley 23 de 1982, el decreto 1360 de 1989 y la ley 44 de 1993 (las decisiones 351 y 486, si bien tienen aplicación preferente en Colombia, son normativas subregionales, a las cuales se refiere el siguiente numeral). El decreto 1360 y la ley 44, son normas que se encargan de regular aspectos operativos y en los cuales se menciona al software como una obra independiente de las obras literarias y que se protege mediante el Derecho de Autor, sin perjuicio de su posible protección mediante otras ramas del derecho, por lo cual no se configura una antinomia entre estas normas con la normativa que regule de manera especial al software, por la que se propende.

A lo largo de este proyecto se tendrán en cuenta las leyes anteriormente mencionadas, para poder entregar un software legal, transparente y seguro, con el que la empresa Rentacar Internacional pueda continuar sus labores sin tener inconvenientes.

Para que el proyecto tenga un desarrollo sostenible, tendremos en cuenta los planes de desarrollo sostenibles que existen tanto a nivel internacional como nacional, departamental y municipal.

Tanto en el plan de desarrollo internacional de las naciones unidas como en el plan nacional de desarrollo sostenible está el objetivo número nueve, el cual es construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación. En el plan de desarrollo departamental publicado el 20/03/2016 encontramos el apartado 2.3.3.3 de la pagina 101 el Tema de Desarrollo Ciencia, Tecnología e Innovación, por parte del municipio está el plan de desarrollo municipal (Bucaramanga) 2020-2023 el cual en su apartado 2.3.1.2 pagina 121 nos habla de Emprendimiento, Innovación y Desarrollo Empresarial. Estos serán los lineamientos por los cuales se guiará el proyecto.

APROXIMACIÓN METODOLÓGICA DEL PROYECTO

La decisión de la metodología de investigación que se usará en el proyecto, es una de las partes más importantes de todas, esta definirá el rumbo que tomará toda la investigación, siendo la estrategia que se seguirá para el desarrollo de esta misma. Teniendo en cuenta que para este proyecto estamos analizando un problema de la empresa Rentacar que necesita solución, esto encaja con la definición de una investigación-acción la cual según la universidad de Colima UCOL (S.F).Investigación-acción. <https://recursos.ucol.mx> Este tipo de metodología se ocupa del estudio de una problemática social específica que requiere solución y que afecta a un determinado grupo de personas, sea una comunidad, asociación, escuela o empresa.

Es apropiada para aquellos que realizan investigaciones en pequeña escala, preferentemente en las áreas de educación, salud y asistencia social e incluso en administración. Constituye un método idóneo para emprender cambios en las organizaciones (Blaxter, Hughes y Tight, 2000), por lo que es usada por aquellos investigadores que han identificado un problema en su centro de trabajo y desean estudiarlo para contribuir a la mejora (Bell, 2005).

Por otro lado, sería una metodología de investigación-acción de tipo práctica, existen cuatro pasos a seguir según esta metodología:

* Detección y diagnóstico del problema de investigación.
* Elaboración del plan para solucionar el problema o introducir el cambio.
* Implementación del plan y evaluación de resultados.
* Realimentación, la cual conduce a un nuevo diagnóstico y a una nueva espiral de reflexión y acción.

Estos serán los pasos que se seguirán a lo largo de toda la investigación, se pueden llevar de la mano con los pasos de ingeniería, para lograr un proceso completo, eficaz y funcional, con el que se pueda lograr la mejor solución.

CRONOGRAMA

29/08/2022 - 04/09/2022 Detección y diagnóstico del problema de investigación.

05/09/2022 - 11/09/2022 Inicio de desarrollo del proyecto (Introducción, justificación, objetivos, características de las necesidades del proyecto, marco de referencia, aproximación metodológica del proyecto, cronograma, presupuesto y comienzo del desarrollo del proyecto).

12/09/2022 - 14/09/2022 Finalización del desarrollo del proyecto y primera entrega.

15/09/2022 - 18/09/2022 Contextualización de todo lo que tiene la nueva entrega

19/09/2022 - 25/09/2022 Árbol de refinamiento, Montaje y pruebas de requisitos de la Base de Datos.

26/09/2022 - 02/10/2022 Comienzo de modelamiento de software (modelo avanzado de clases, casos de uso, diagrama de secuencia)

20/10/2022 - 09/10/2022 Seguir avanzando con los modelos y diagramas tratados en la semana anterior.

10/10/2022 - 16/10/2022 Finalización de los modelos y diagramas de la semana anterior y comienzos de diseño de software y cliente servidor.

17/10/2022 - 23/10/2022 Finalización del diseño de software y cliente servidor además del prototipo de interfaz gráfica, al final de la semana segunda entrega del proyecto.

24/10/2022 - 30/10/2022 Comenzar el modelamiento detallado del sistema.

31/10/2022 - 06/11/2022 Finalización del modelamiento detallado del sistema y comienzo de la implementación de la aplicación

07/10/2022 - 13/11/2022 Modelamiento de la base de datos e implementación de la aplicación.

14/11/2022 - 20/11/2022 montaje y prueba de requisito de la base de datos e implementación de la aplicación.

21/11/2022 - 27/11/2022 Detalles finales y preparación de la entrega final.

PRESUPUESTO

Para este proyecto de dispone de tres investigadores y desarrolladores de software, que cuentan con tres meses para la total finalización del proyecto, estos trabajarán con software gratuito lo que permitirá bajar los costos de la operación, teniendo en cuenta el tiempo requerido no sólo para desarrollar el software si no para hacer toda la investigación, modelos y diseños requeridos, la mano de obra tendrá un costo de 7.000.000 por trabajador, dando un total de 21.000.00 de pesos, esta será la inversión total que tendrá que hacer la empresa Rentacar Internacional.

DESARROLLO DEL PROYECTO

Para comenzar a construir el software, se debe entender el problema que se tiene que solucionar, una de las formas de analizar más a fondo la situación es el Léxico orientado al modelado (LOM), este es un procedimiento en el que se califican a los sustantivos como objetos, atributos o sujetos, los verbos como acciones, además de los adjetivos y acciones como estados. Así también se puede extraer los posibles nombres con los que se pueda tratar una misma cosa y la relación que tenga con otros elementos del LOM.

A continuación se encuentran las cinco clasificaciones divididas en tablas.

| **objetos** | | |
| --- | --- | --- |
| **Nombre** | **Alias** | **Relación** |
| Automovil | Automovil/Coche |  |
| Alquiler | Contrato de coche |  |
| Mantenimiento |  |  |
| Reparacion |  |  |
| Alquiler de extras |  |  |
| Registro de venta |  |  |
| Taller |  |  |
| Cliente Directo |  |  |
| Cliente Agencia | Intermediaria |  |
| Tarifa Extra |  |  |
| Tatifa alquiler |  |  |
| Empleado |  |  |
| Venta de Vehiculo |  |  |
| Tarifa Seguro |  |  |
| Contrato |  |  |

Los elementos de la tabla objetos se convertirán en objetos del sistema, los cuales tendrán sus propios atributos.

| **Acciones** | | |
| --- | --- | --- |
| **Nombre** | **Alias** | **Relación** |
| alquilar |  | vehiculo |
| Compra |  | vehiculo |
| Venta |  | vehiculo |
| alquilar |  | extras |
| Reparar |  | vehiculo |
| Preparar |  | vehiculo |

Los elementos de la tabla acciones se convertirán en funciones que el programa debe permitir realizar, estos estarán relacionados a los objetos del sistema.

| **Estados** | | |
| --- | --- | --- |
| **Nombre** | **Alias** | **Relación** |
| Alquilado |  | vehiculo |
| A preparar |  | vehiculo |
| Listo |  | vehiculo |
| En actividad especial |  | vehiculo |
| Mantenimiento |  | vehiculo |
| Reparación |  | vehiculo |
| Vendido |  | vehiculo |
| Kilometrage excedido |  | vehiculo |
| Fuera de cobertura del seguro |  | vehiculo |

Los elementos de la tabla estados se convertirán en los estados de los objetos a los cuales están relacionados.

| **Sujetos** | | |
| --- | --- | --- |
| **Nombre** | **Alias** | **Relacion** |
| Empleado común |  | Usuarios |
| Empleado con mayores privilegios |  | Usuarios |
| Cliente Directo |  | Contrato |
| Agencia o intermediaria |  | Contrato |

Los elementos de la tabla sujetos son los usuarios que usarán el software.

| **Atributos** | | |
| --- | --- | --- |
| **Nombre** | **Alias** | **Relación** |
| Código |  | Vehiculo/Mantenimiento/Reparacion/Registro de venta/Cliente directo/Cliente agencia/Tarifa extra |
| Matricula |  | Vehiculo/Venta de vehiculo/Registro de venta/Contrato |
| Marca |  | Vehiculo |
| Clase |  | Vehiculo/Contrato |
| Grupo |  | Vehiculo/Tarifa alquiler/Tarifa seguro/Contrato |
| Fecha de compra |  | Vehiculo |
| Kilometraje |  | Vehiculo |
| Fecha |  | Mantenimiento/Reparacion/Contrato |
| Codigo vehículo |  | Mantenimiento /Reparacion |
| Fecha de entrega |  | Mantenimiento /Reparacion/Registro de venta/Contrato |
| Enviado por |  | Mantenimiento Reparacion |
| Descripción |  | Mantenimiento/Reparacion/Tarifa extra |
| Número |  | Taller/Empleado |
| Nombre |  | Taller/Cliente directo/Cliente agencia/Tarifa extra/Empleado |
| Dirección |  | Taller/Cliente directo/Cliente agencia |
| Teléfono |  | Taller/Cliente directo |
| Precio |  | Reparacion/ Tarifa extra |
| Precio de venta |  | Registro de venta |
| Fecha de operación |  | Registro de venta |
| Realizado por |  | Registro de venta |
| Observaciones |  | Registro de venta /Empleado |
| Trato |  | Cliente Directo |
| Población |  | Cliente Directo/Cliente agencia |
| DNI |  | Cliente Directo |
| Carnet de conducir |  | Cliente Directo |
| Tarjeta de crédito No. |  | Cliente Directo |
| Tipo de tarjeta |  | Cliente Directo |
| Fax |  | Cliente Directo |
| NIF |  | Cliente agencia |
| Persona de contacto |  | Cliente agencia |
| Comisión |  | Cliente agencia |
| Hora |  | Contrato |
| Nº de contrato |  | Contrato |
| Sucursal |  | Contrato |
| Modelo |  | Contrato |
| Kms |  | Contrato |
| Cliente |  | Contrato |
| Lugar de entrega |  | Contrato |
| Descuento |  | Contrato |
| Subtotal alquiler |  | Contrato |
| Cantidad de días de renta del vehículo |  | Contrato |
| Valor por dia del vehículo |  | Contrato |
| Cantidad de días de extras |  | Contrato |
| valor por dia de extras |  | Contrato |
| Subtotal complementos |  | Contrato |
| IVA |  | Contrato |
| Combustible |  | Contrato |
| Total importe |  | Contrato |
| Importe fianza |  | Contrato |
| Total importe |  | Contrato |
| Fecha de ingreso |  | Empleado |
| Total contrato |  | Contrato |

Los elementos de la tabla atributos serán los atributos de los objetos a los cuales están relacionados.

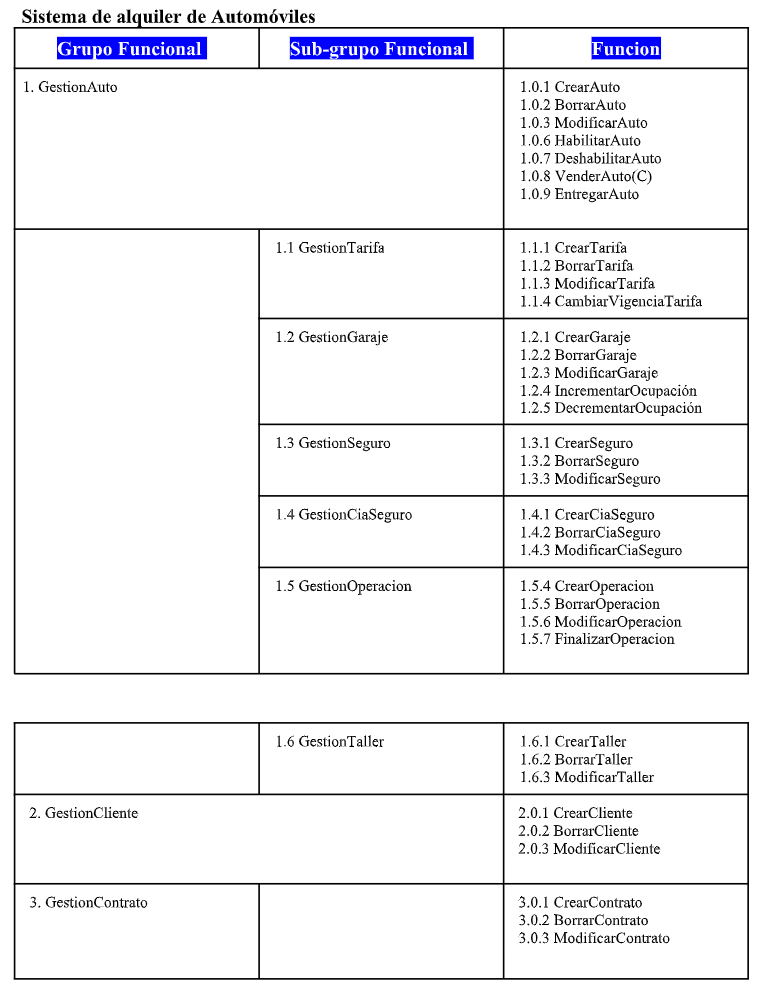
Para continuar con la construcción del software se debe hacer una descomposición jerárquica de las funciones del sistema, para dejar claro cada una de las funciones con las que contará y facilitará la implementación de los siguientes pasos, para esto se utilizara el Árbol de refinamiento funcional (ARF) el cual es un método para definir y categorizar las funciones del sistema, en el cual por medio de unos objetivos y objetos se descomponen sucesivamente en funciones, subfunciones hasta que no se pueda separar los elementos en más funciones.

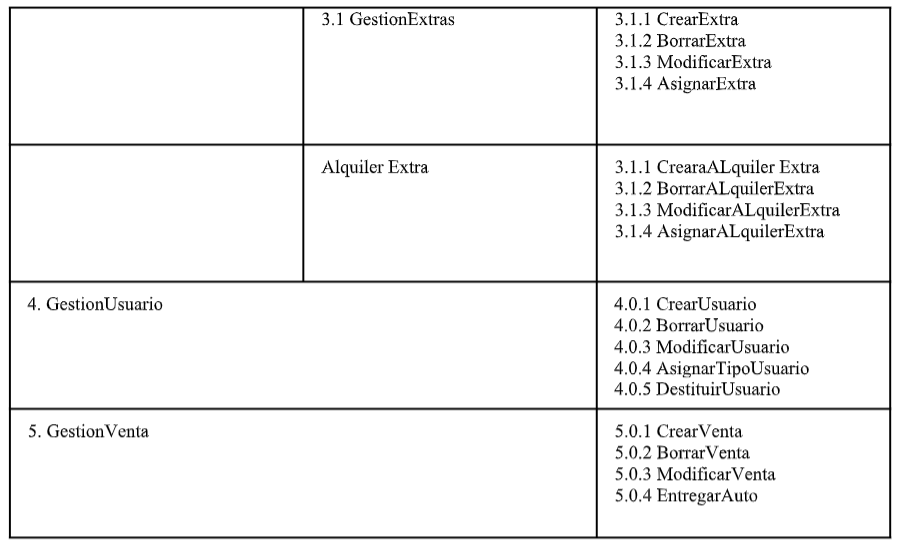
En nuestro caso se creará el árbol de refinamiento funcional (ARF) de las funcionalidades del sistema Rentacar con base a el estudio realizado y el léxico orientado al modelado (LOM), estas funcionalidades serán: grupos funcionales, sub grupos funcionales o funciones.

En el sistemas Rentacar en grupos funcionales se puede encontrar la gestión de auto, gestion cliente, gestión de contrato, gestión de usuario, gestión de venta y gestión de compra.

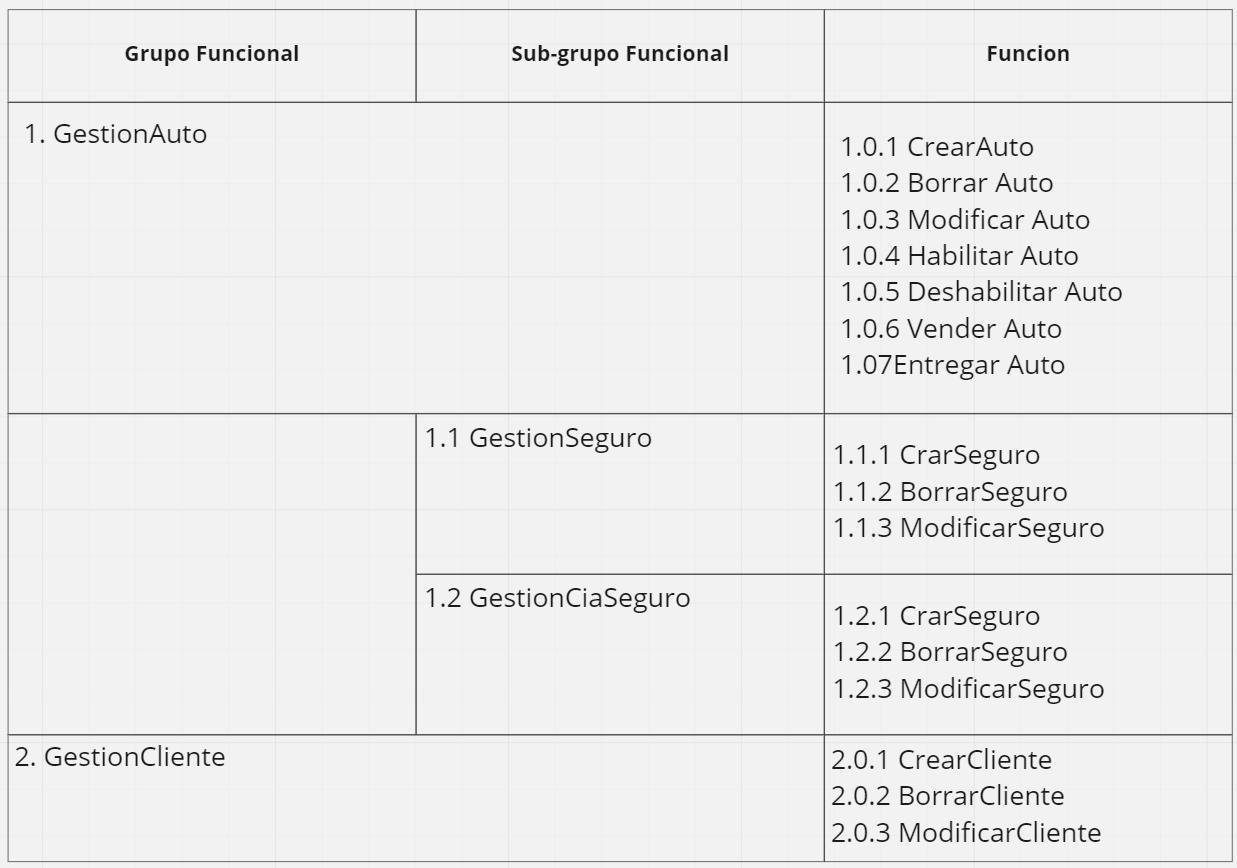
Cada uno de estos grupos tiene dentro de ellos subgrupos funcionales, en el caso de gestión de auto cuenta con los subgrupos funcionales gestión de tarifa, garaje, seguro, operación, taller y extras, por otra parte la gestión de contrato el cual tiene el subgrupo funcional gestión de contrato extra y alquiler extras, cada uno de los grupos y subgrupos cuenta con funciones, en el sistema Rentacar estas funciones serán el último nivel de nuestra descomposición jerárquica de las funcionalidades.

Ahora se mostrará el árbol de refinamiento funcional basado en toda la información, análisis y LOM del sistema Rentacar realizado anteriormente, este será el ARF general del sistema.



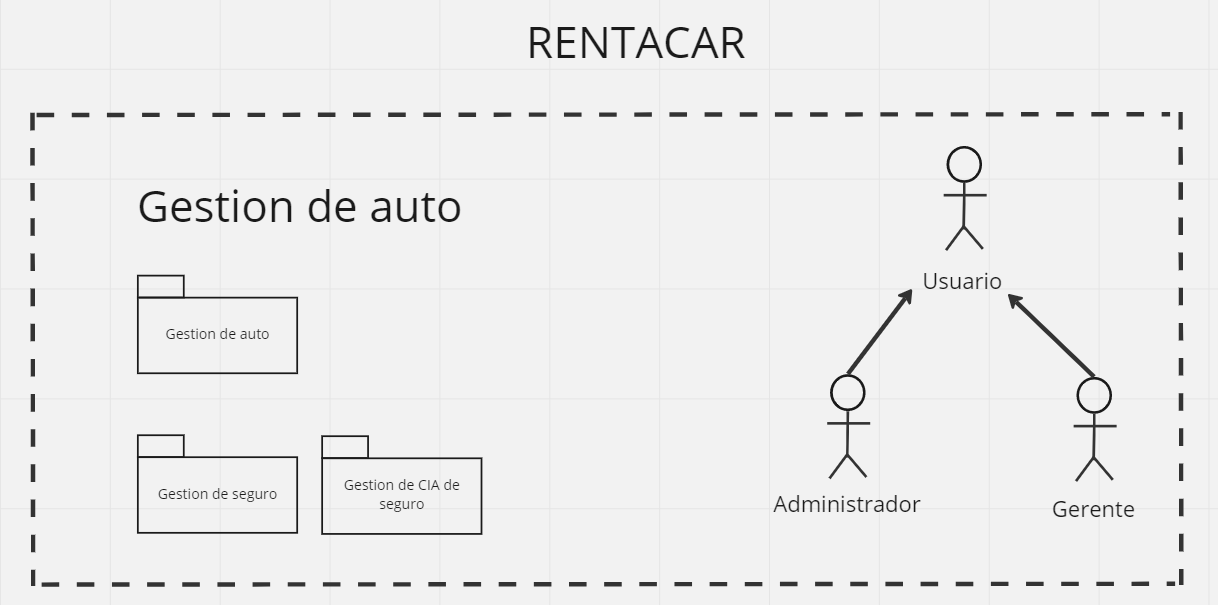


En la realización de este proyecto se vio pertinente separa las funciones del sistema y dividirla en varios grupos de trabajo para una mejor optimización del tiempo y facilidad del trabajo, éste grupo de trabajo contará con los grupos funcionales de Gestión de auto con los subgrupos funcionales de gestión de seguro y gestión de compañía de seguro cada uno con sus funciones y aparte también se contará con el grupo funcional gestión de cliente con sus funciones, a continuación se enseñara el ARF actualizado con las funcionalidades asignadas a nuestro grupo de trabajo.



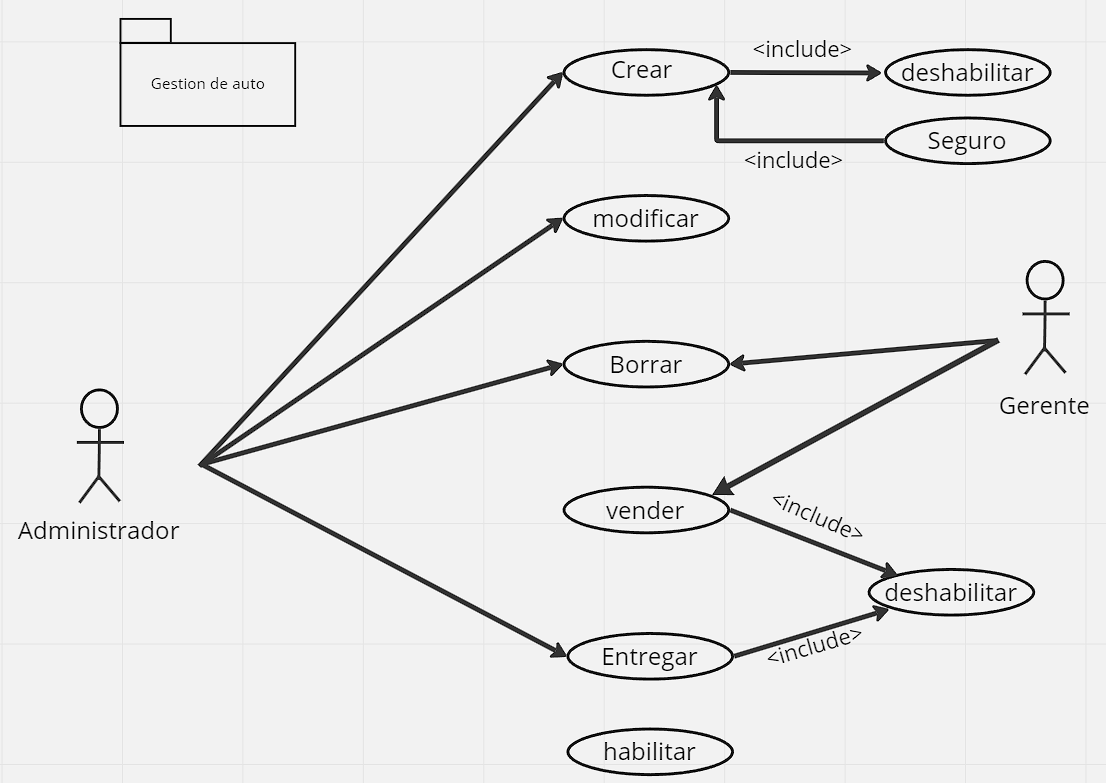
En lo que se presentará a continuación se encuentran los diagramas de casos de uso, de secuencia y de clases. Todos estos basados en el ARF de nuestro grupo de trabajo. los cuales nos ayudarán a controlar el programa para nuestro sistema que es la gestión de auto, gestión de seguro, gestión de cia seguro y gestión de cliente.

Primero se mostrarán los diagramas de casos de usos de los temas a trabajar por nuestro grupo, el diagrama de caso de uso nos servirá para representar nuestro sistema como un conjunto de interacciones que se desarrollarán entre casos de usos y sus actores.



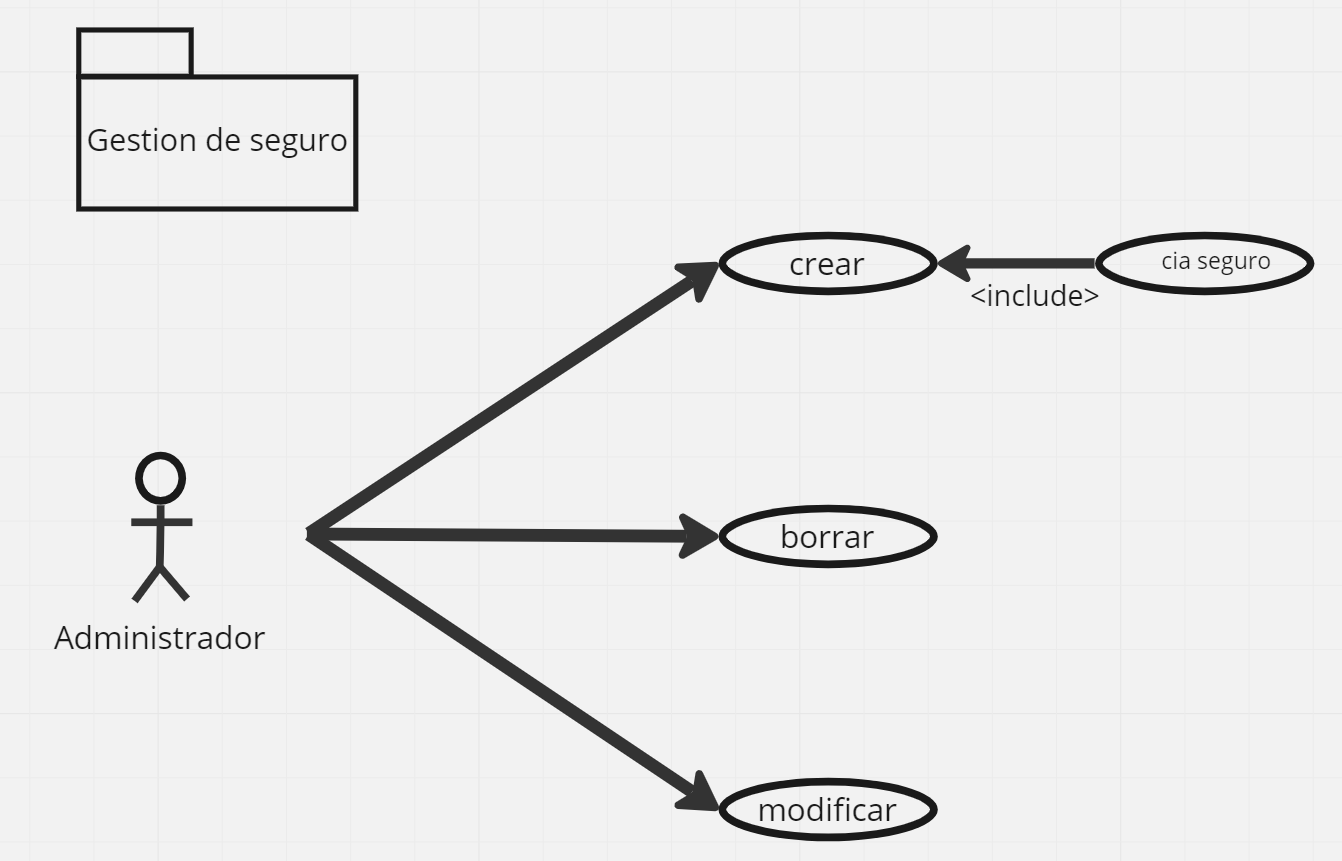
En esta imagen vemos nuestro sistema Rentacar el cual cuenta con nuestro grupo funcional gestión de auto y sus subgrupos funcionales Gestión de seguro y gestión de Cia seguro, cada uno de estos con sus funciones, observamos que nuestros actores del sistema son el administrador y el gerente, ahora se procederá a realizar el diagrama de caso de usos y secuencia para estos grupos denuesto sistema Rentacar.

**Gestion de auto**



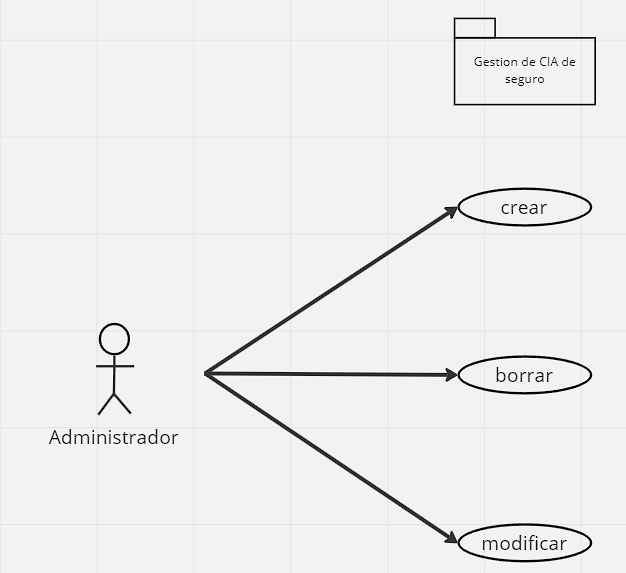
Lo que se hizo en este diagrama de casos de uso fue básicamente conectar lo que pueden hacer el gerente o el administrador dentro del programa, en este caso el administrador podrá crear, modificar, borrar y entregar el auto y por otro lado el gerente puede borrar y vender los autos, al momento de crear el auto se deshabilita para conseguirle el seguro obligatorio y al venderse o entregarse éste se también deshabilitará.

**Gestion seguro**



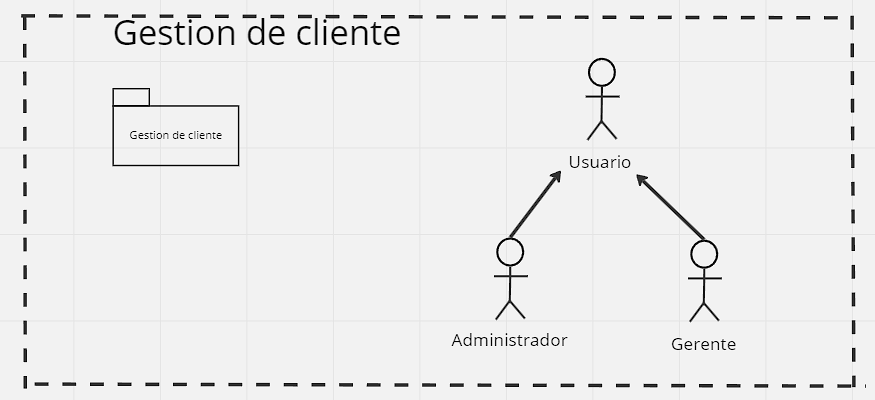
En este caso de uso, lo que hace el usuario que en este caso es el administrador es crear, borrar y modificar los seguros para los automóviles.

**Gestión de cia de seguro**

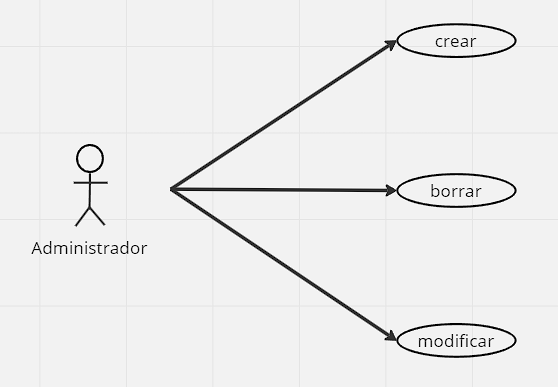


En este caso de uso, lo que hace el usuario que en este caso es el administrador es crear, borrar y modificar las cias de seguros para los seguros.

**Gestión de cliente**



En esta imagen encontramos nuestro grupo funcional gestión de cliente el cual hace parte de nuestro sistema Rentacar.

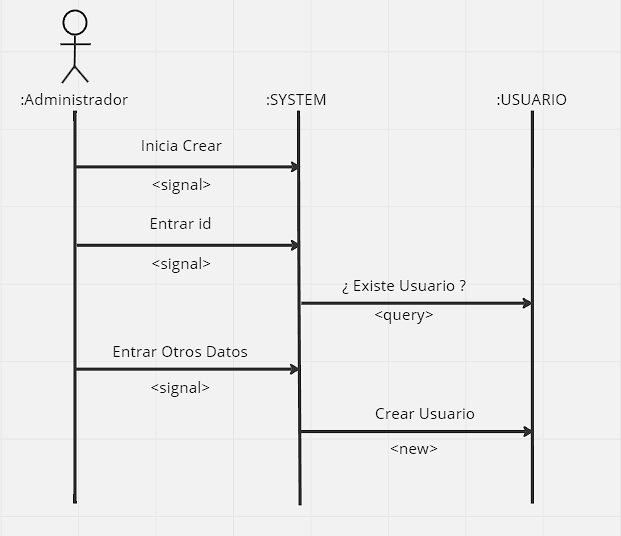


Este caso de uso se creó para la gestión de los clientes, en este caso el administrador tiene permitido crear, borrar y modificar clientes para el sistema.

**DIAGRAMA DE SECUENCIA**

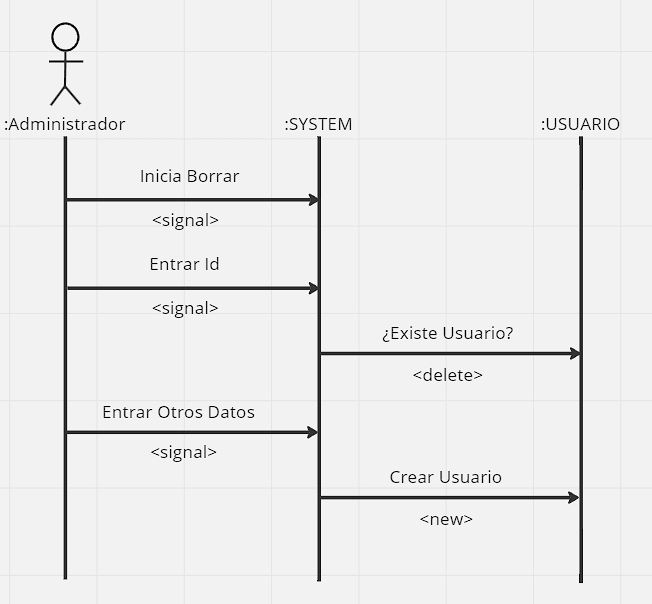
En este diagrama lo que se hace es hacer un paso a paso para representar el comportamiento de nuestro sistema, nos muestra cómo interactúan los participantes y los objetos del sistema y el orden en que se producen.

**CREAR USUARIO**

****

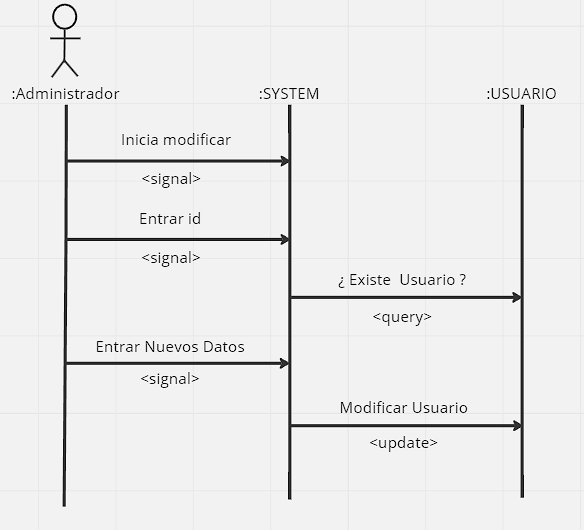
El diagrama de secuencia lo que hace es el proceso paso a paso para hacer alguna acción, en este caso para crear el usuario.

**BORRAR USUARIO**

****

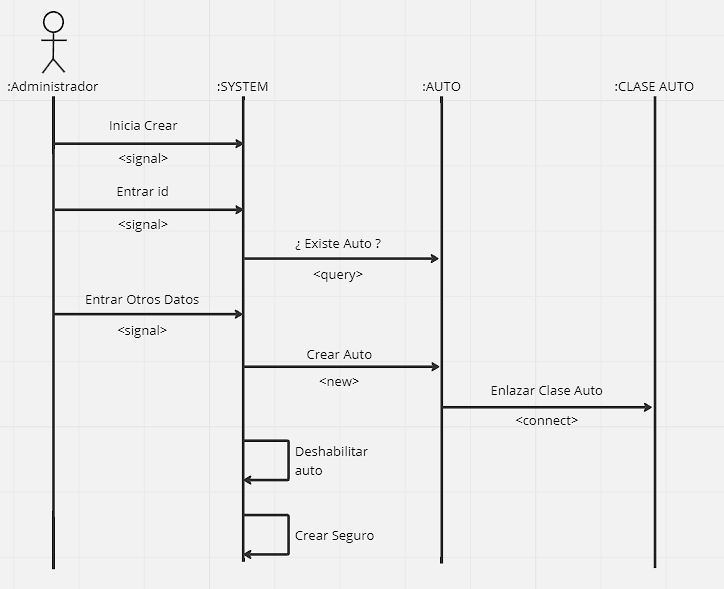
Este diagrama de secuencia lo que hace es el paso a paso para borrar un usuario.

**MODIFICAR USUARIO**

****

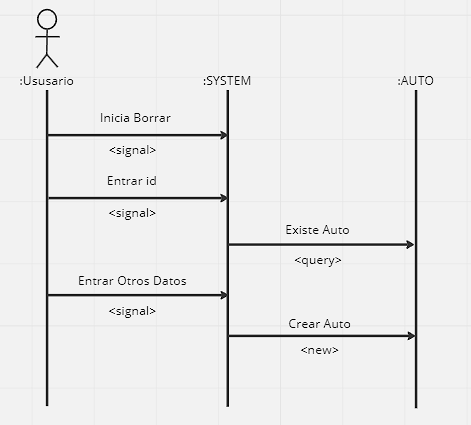
Este diagrama de secuencia lo que hace es el paso a paso para modificar los datos de un usuario creado.

**CREAR AUTOMÓVIL**

****

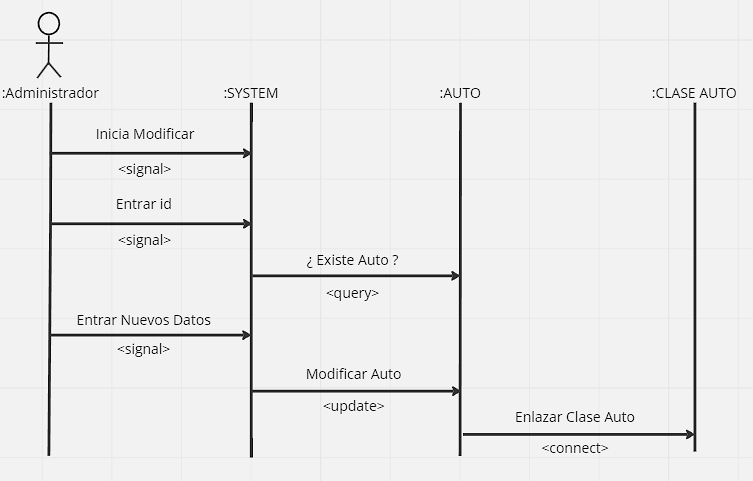
Este diagrama de secuencia lo que hace es el paso a paso para crear un auto.

**BORRAR AUTOMÓVIL**

****

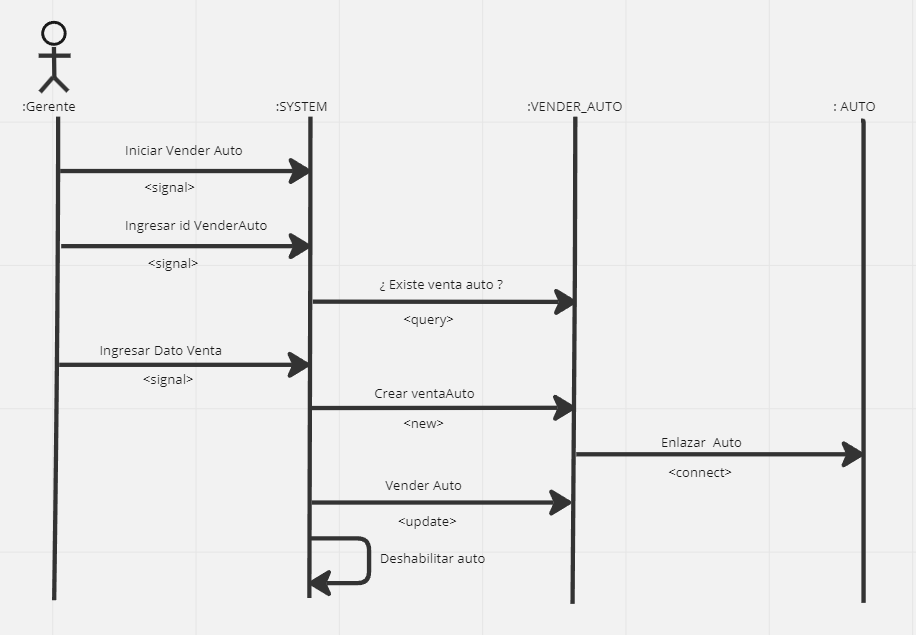
Este diagrama de secuencia lo que hace es el paso a paso para borrar un auto.

**MODIFICAR AUTOMÓVIL**

****

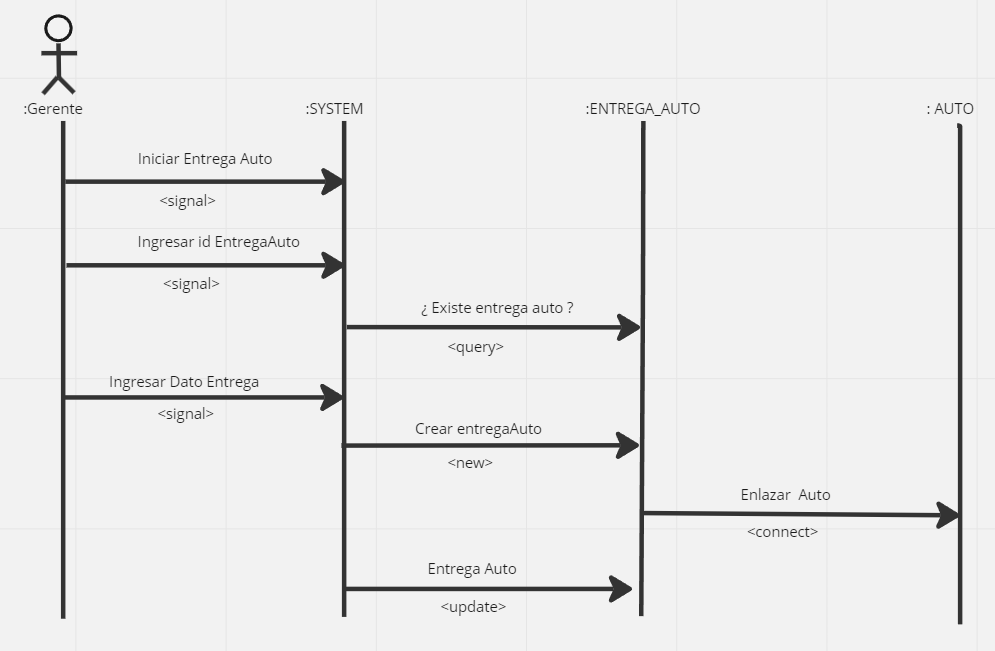
Este diagrama de secuencia lo que hace es el paso a paso para modificar los datos de un auto creado.

**VENDER AUTOMÓVIL**

****

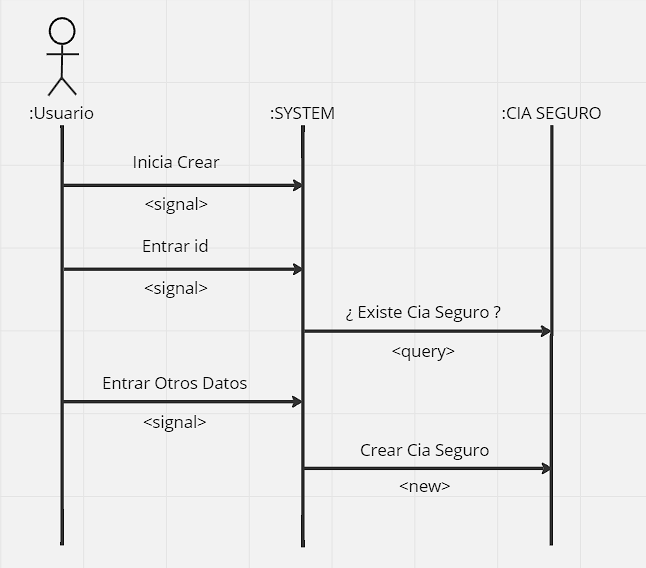
Este diagrama de secuencia lo que hace es el paso a paso para la hora de vender el auto el programa lo sepa y lo deshabilite.

**ENTREGAR AUTOMÓVIL**

****

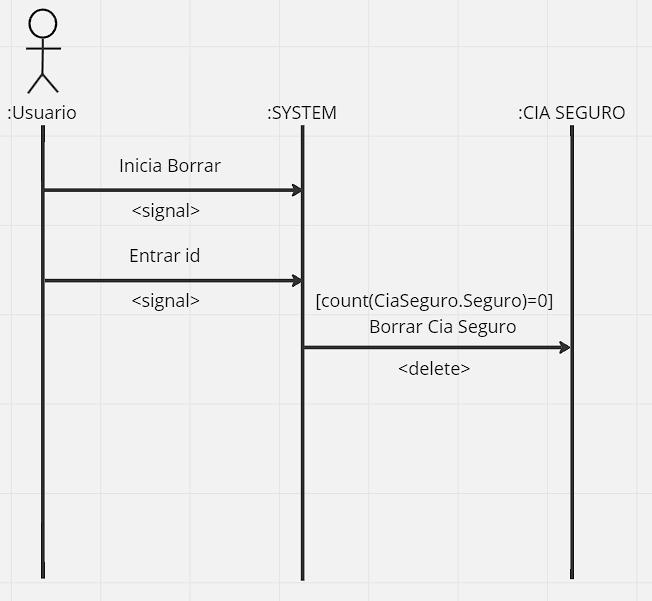
Este diagrama de secuencia lo que hace es el paso a paso para la entrega del auto

**CREAR CIA SEGURO**

****

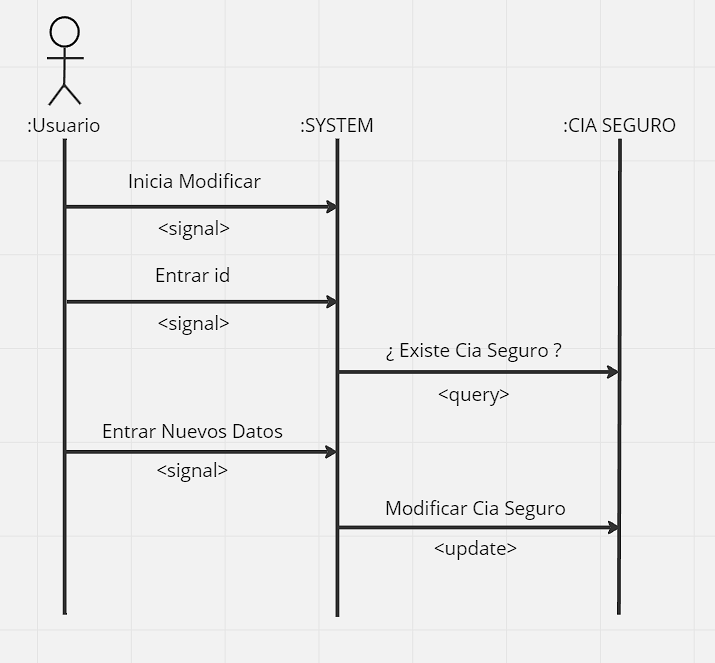
Este diagrama de secuencia lo que hace es el paso a paso para crear una cia de seguro.

**BORRAR CIA SEGURO**

****

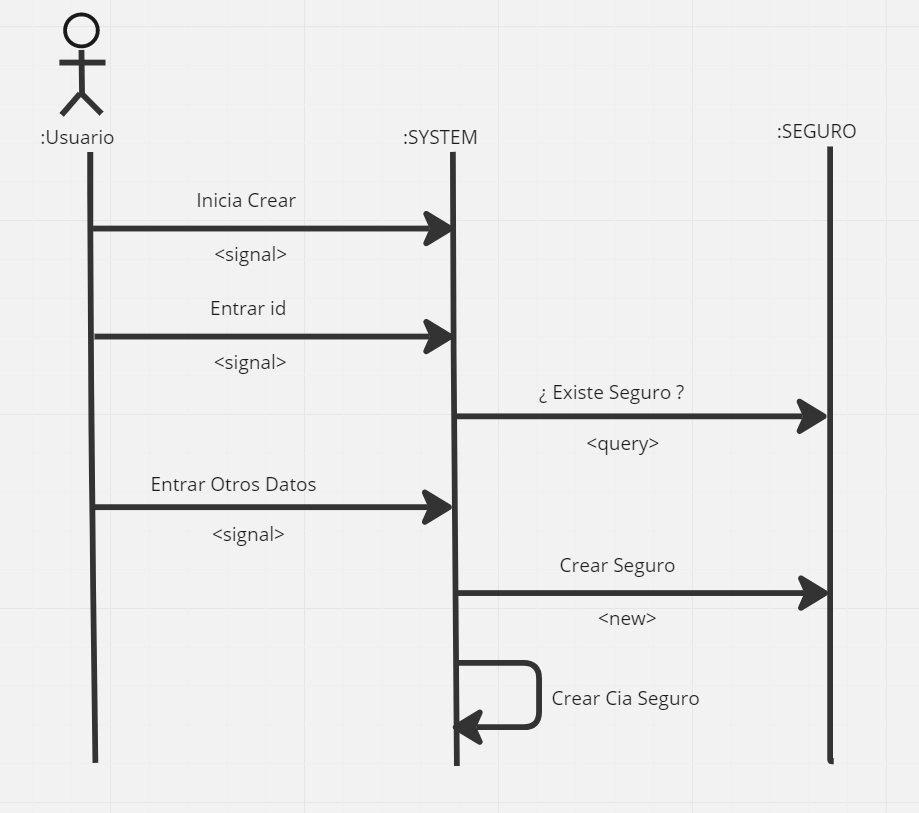
Este diagrama de secuencia lo que hace es el paso a paso para borrar una cia de seguro.

**MODIFICAR CIA SEGURO**

****

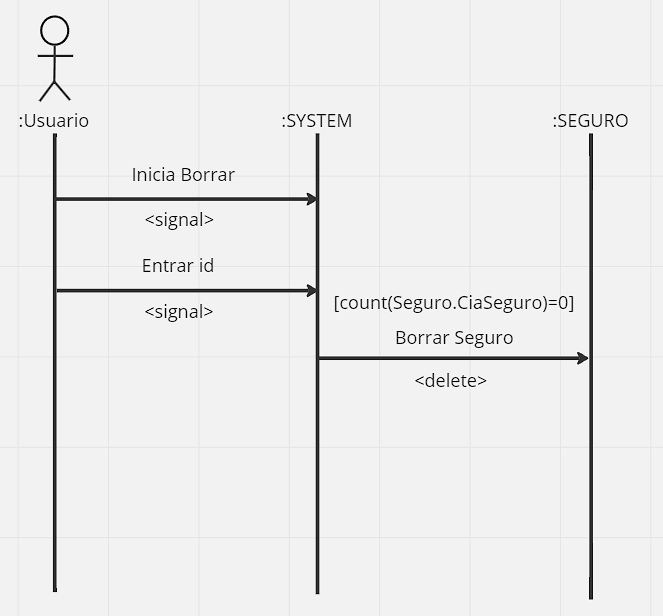
Este diagrama de secuencia lo que hace es el paso a paso para modificar los datos de una cia de seguro creada.

**CREAR SEGURO**

****

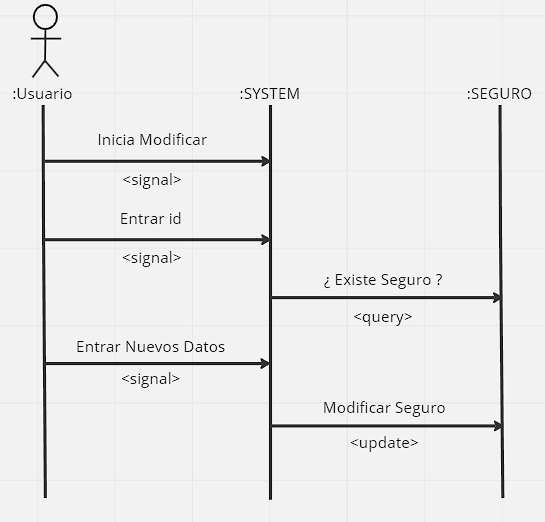
Este diagrama de secuencia lo que hace es el paso a paso para crear un seguro.

**BORRAR SEGURO**

****

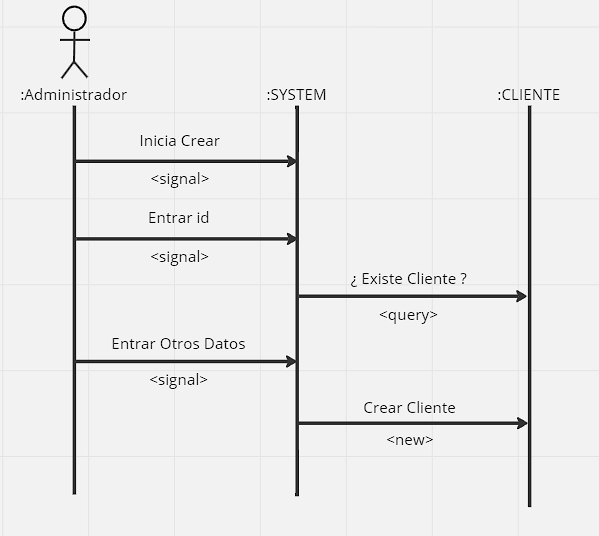
Este diagrama de secuencia lo que hace es el paso a paso para borrar un seguro.

**MODIFICAR SEGURO**

****

Este diagrama de secuencia lo que hace es el paso a paso para modificar los datos de un seguro creado.

**CREAR CLIENTE**

****

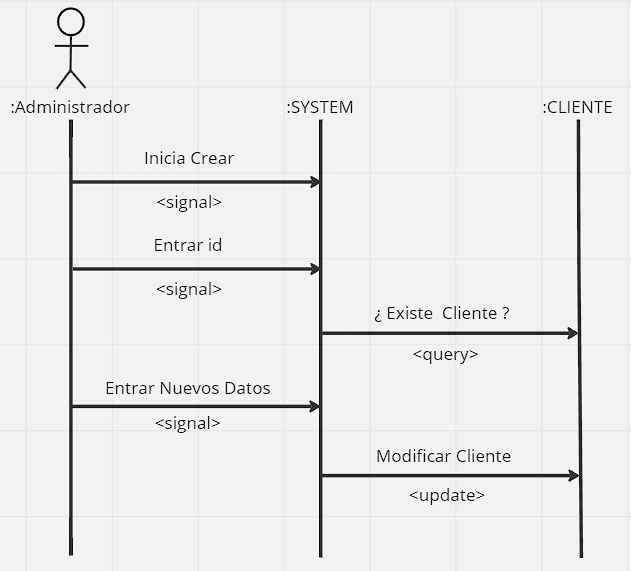
Este diagrama de secuencia lo que hace es el paso a paso para crear un cliente.

**BORRAR CLIENTE**

****

Este diagrama de secuencia lo que hace es el paso a paso para borrar un cliente.

**MODIFICAR CLIENTE**

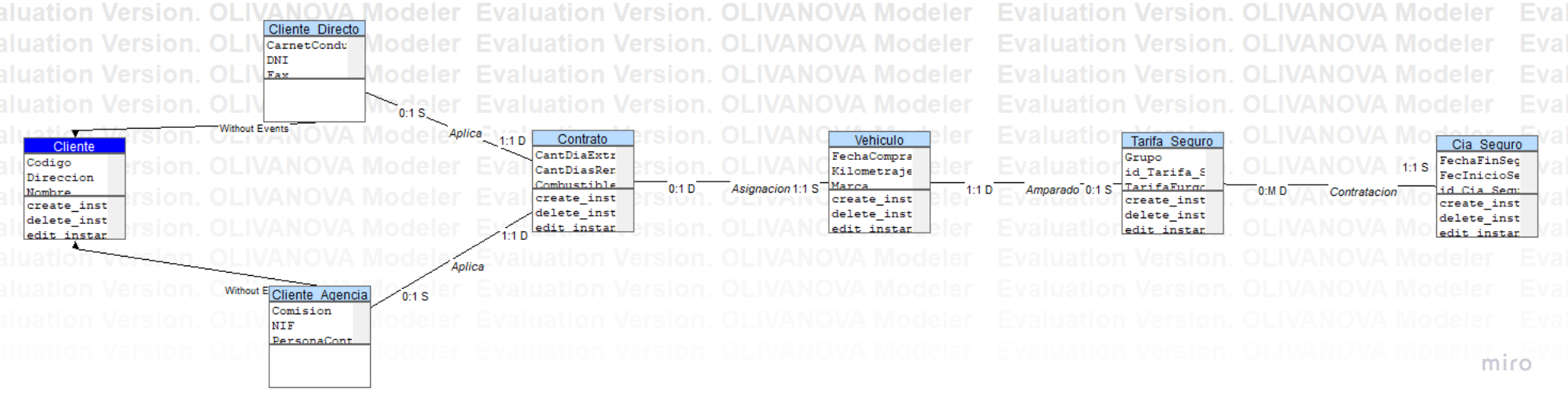
****

Este diagrama de secuencia lo que hace es el paso a paso para modificar los datos de un cliente creado.

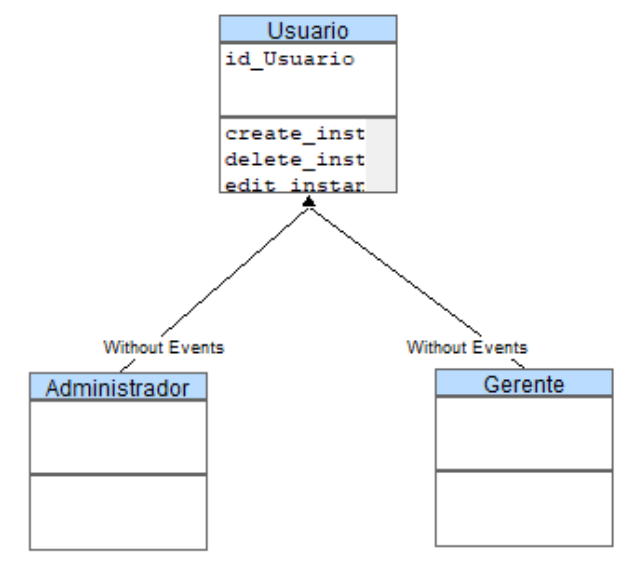
**DIAGRAMA DE CLASES**

Un diagrama de clases es una estructura estática que se usa para mostrar los tipos de relaciones entre los objetos que se están programando. También es una buena manera de mostrar la estructura de clases en un sistema al modelar las clases, atributos, operaciones y relaciones entre objetos.

En la siguiente imagen se procederá a mostrar el diagrama de clases hecho con la herramienta Olivanova de las funcionalidades asignadas a nuestro grupo de trabajo.

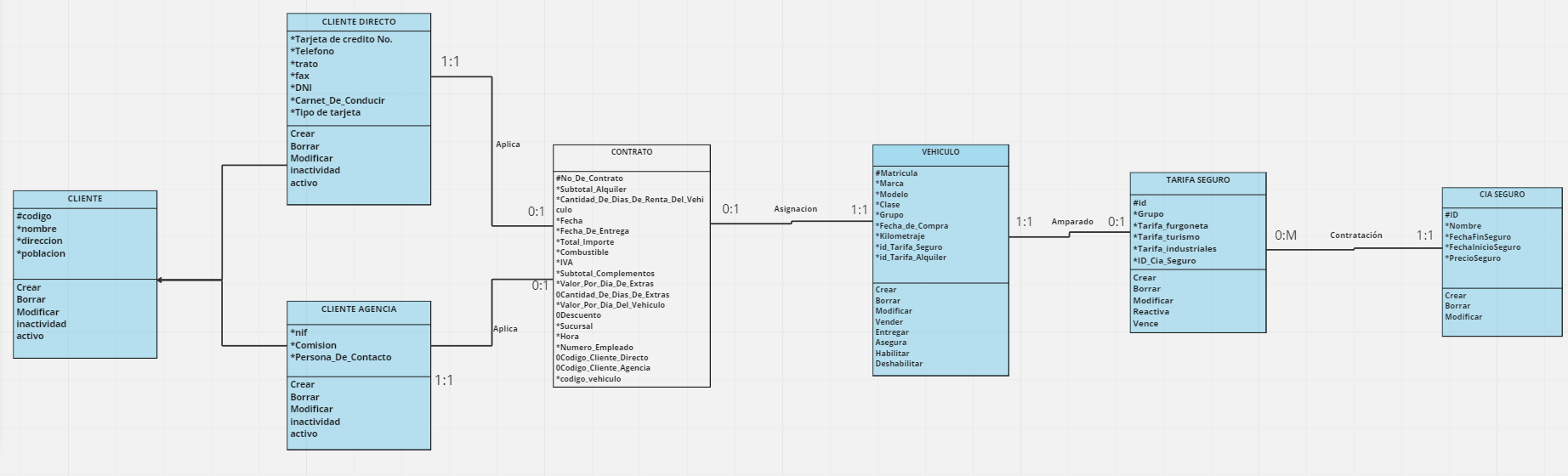


Aca podemos observar nuestro diagrama de clase con las funcionalidades asignadas a nuestro grupo, las cuales son cliente, vehiculo, tarifa seguro y cia seguro, primero tenemos nuestra funcionalidad cliente la cual en realidad cuenta con dos clases las cuales son cliente directo y cliente agencia, debido a que ambas clases hacen referencia a cliente y tienen atributos compartidos se realizó una herencia para estas dos clases la cual se llamará “Cliente” y esta tendrá dentro de ella los atributos compartidos los cuales son código, dirección y nombre de los clientes. Después los atributos propios de cada cliente se pondrán dentro de ellos en el caso de cliente directo tendrá los atributos propios tarjeta de crédito, teléfono, trato, fax, DNI, carnet de conducir y tipo de tarjeta mientras que cliente agencia tendrá de atributos propios nif, comisión y persona de contacto. La clase contrato no hace parte de las funcionalidades asignadas a nuestro grupo pero se decidió agregar para poder hacer las relaciones de las demás clases correctamente, en este caso observamos que para aplicar un contrato primero debemos tener creadas nuestras clases cliente directo y agencia ya que son las clases estáticas mientras contrato es la dinámica, la cardinalidad sería un contrato puede tener minimo cero y máximo un clientes directos o cliente agencia y los clientes directos y agencia pueden tener mínimo uno y máximo un contrato, después observamos que la clase contrato se relaciona a la clase vehículo en el que para asignar un contrato primero debe existir el vehículo por esta razón el vehículo es la clase estática y contrato es la dinámica, la cardinalidad sería tal que un contrato necesita mínimo uno y máximo un vehículo y vehículo necesita mínimo cero y máximo un contrato ya que estamos basándonos en ese momento. A continuación encontramos la clase tarifa seguro, sabemos que el vehículo está amparado por un seguro pero para esto primero debe existir el seguro debido a esto tarifa seguro es la clase estática y vehiculo es la clase dinámica, la cardinalidad es un vehículo está amparado mínimo por cero y máximo por un seguro y seguro tiene mínimo uno y máximo un vehículo. También encontramos la clase Cia seguro la cual sabemos que un seguro contrata una compañía de seguros, en este caso la compañía de seguros es necesario que exista primero para poder brindar un seguro por ello la clase Cia seguro es nuestra clase estática y tarifa seguro es nuestra clase dinámica, la cardinalidad de estas sería de tal forma que un seguro necesita mínimo una y máximo una compañía de seguros y la compañía de seguros puede tener minimo cero y maximo muchos seguros.

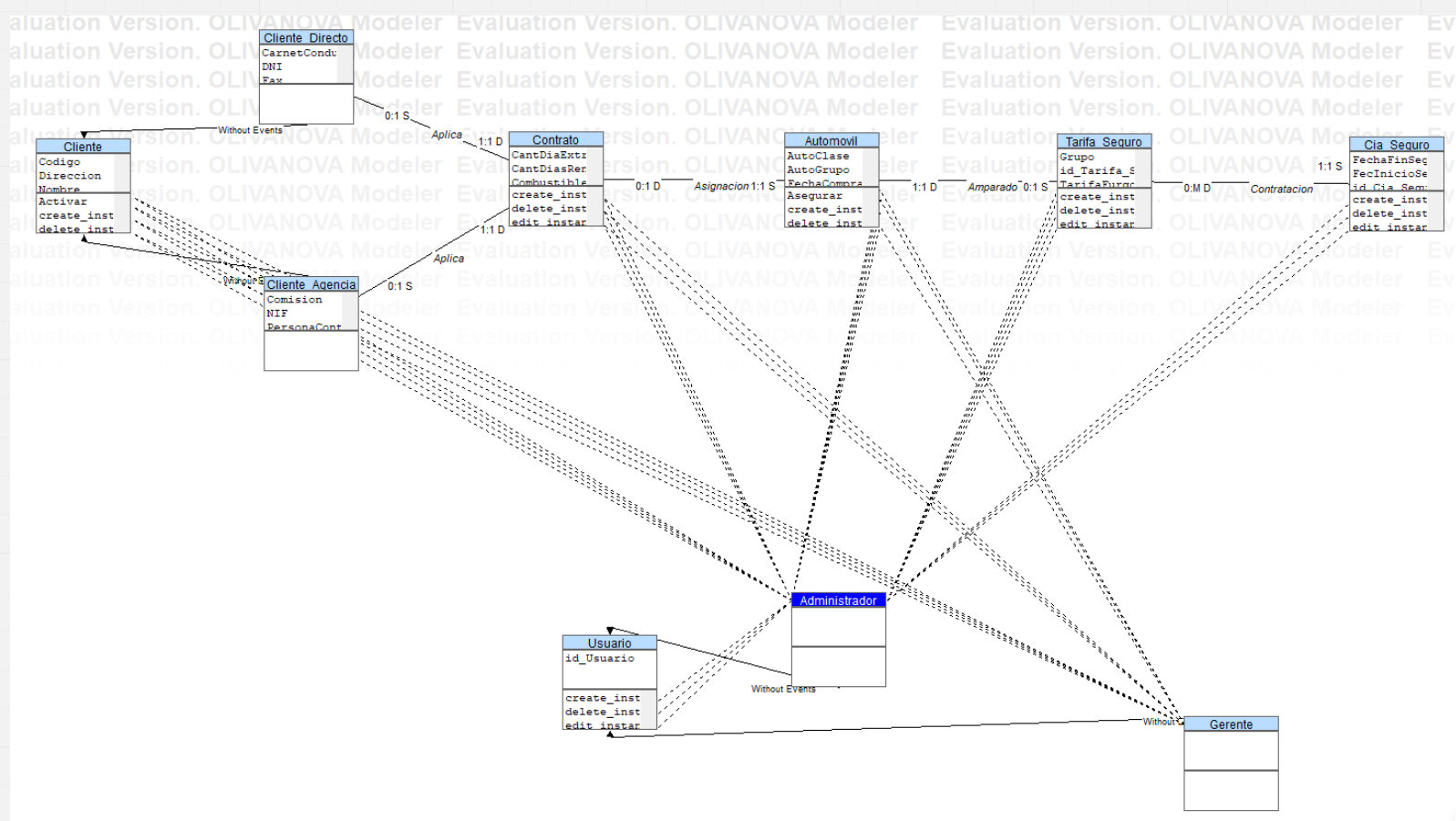


Por último en nuestro diagrama de clases tenemos a los usuarios, los cuales son administrador y gerente, debido a que estas clases tienen los mismos atributos y hacen las dos referencia a los trabajadores decidimos crear una herencia de estas dos clases llamada “Usuario” la cual tiene su id, por otro lado estas clases no van directamente relacionadas con las anteriores vistas.

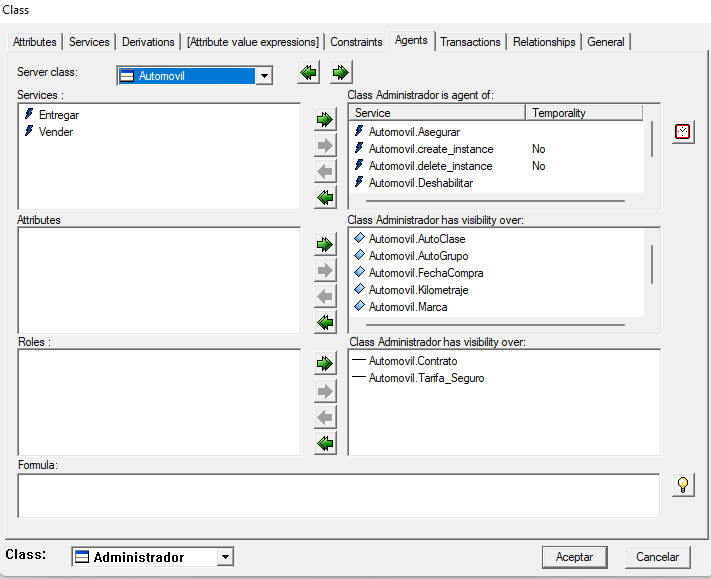
En esta imagen podemos observar un poco mejor todos los atributos que se encuentran dentro de nuestras clases.



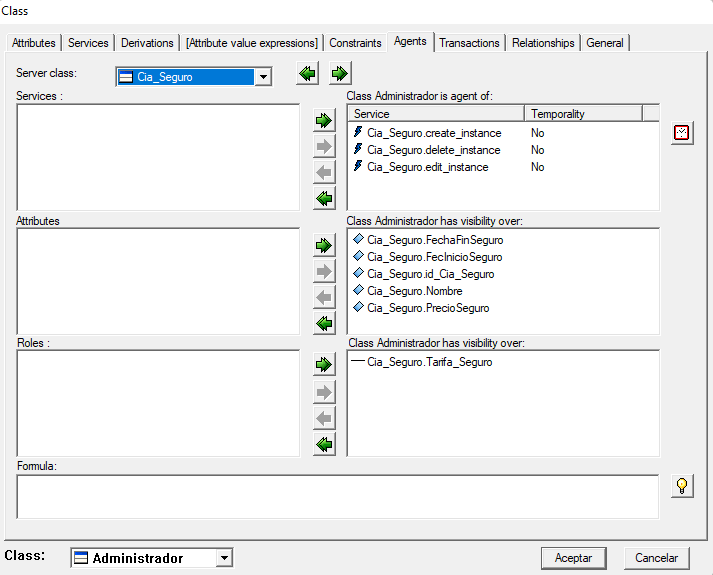
**Diagrama de clases con privilegios de los usuarios**



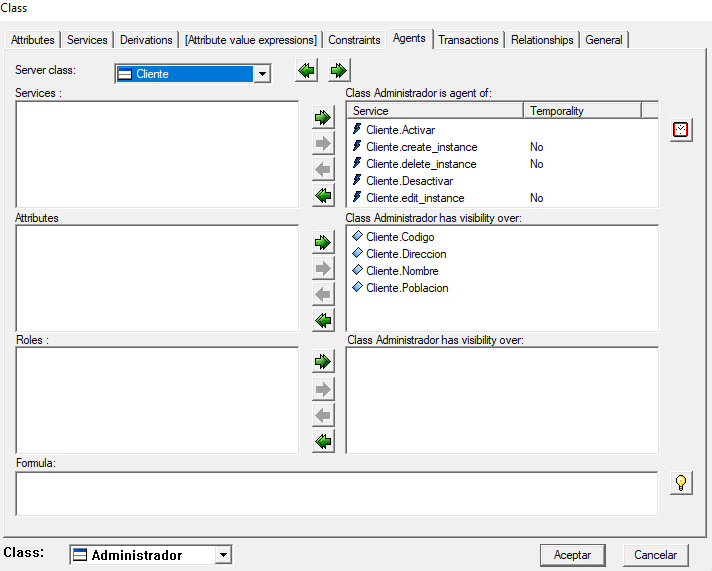
En este diagrama de clases podemos ver los privilegios asignados a los usuarios en el caso del gerente tiene privilegios sobre las clases automovil, contrato y cliente y en el caso de los administradores tienen privilegios sobre las clases Cia seguro, tarifa seguro, automovil, contrato, cliente y usuarios.



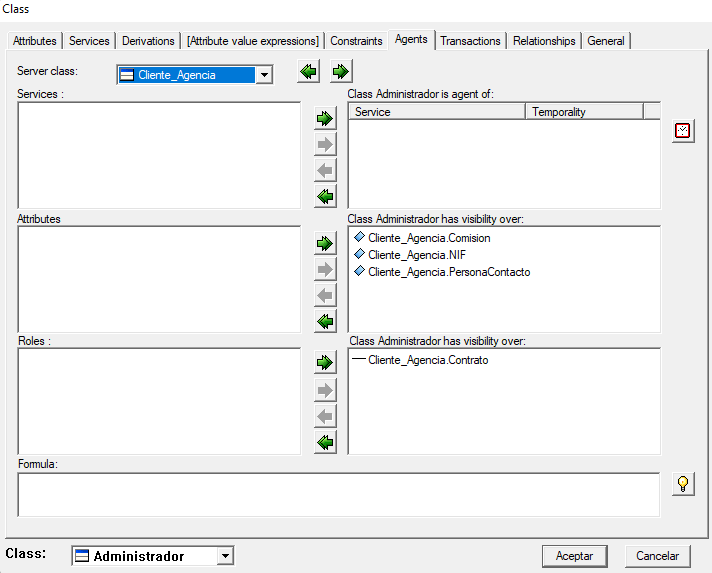
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado al automóvil.



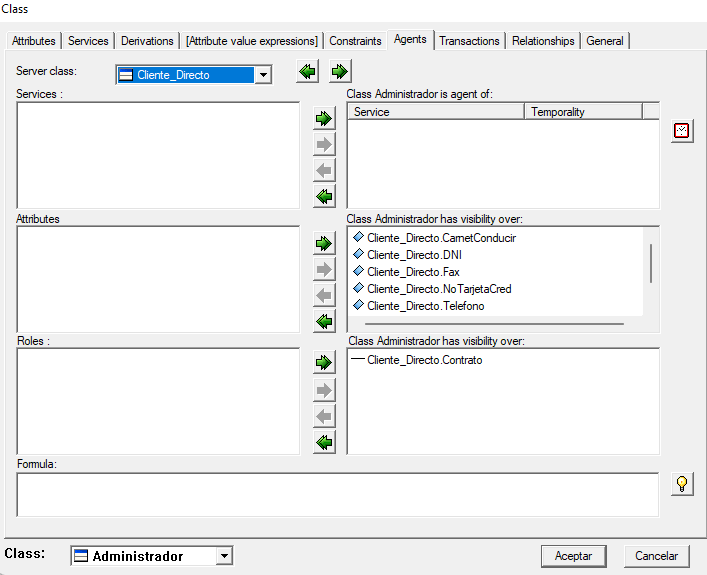
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado a la compañía de seguro.



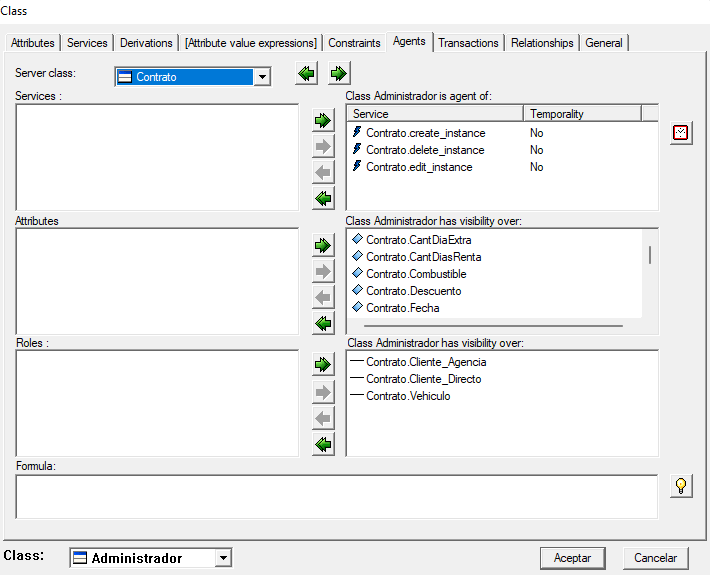
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado a los clientes.



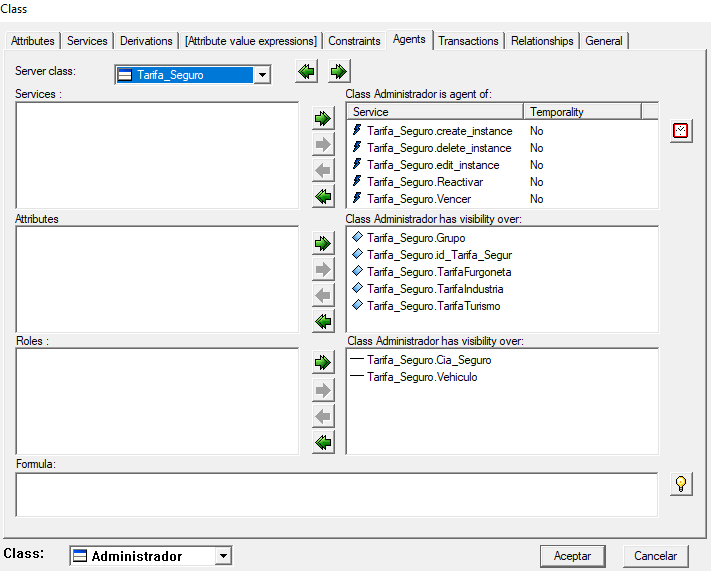
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado al cliente agencia.



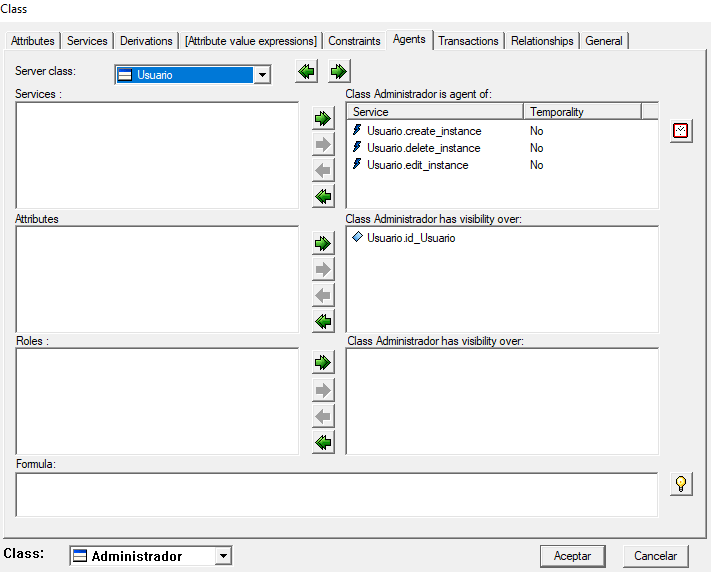
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado al cliente directo.



En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado al contrato.

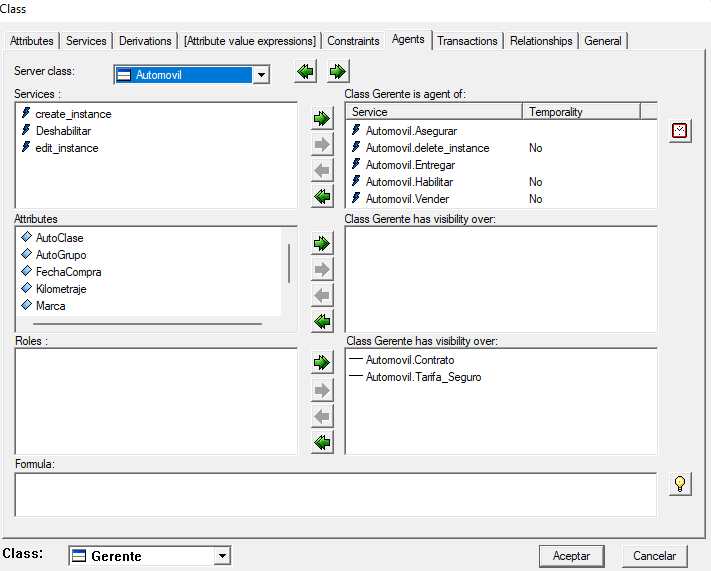


En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado a la tarifa del seguro.

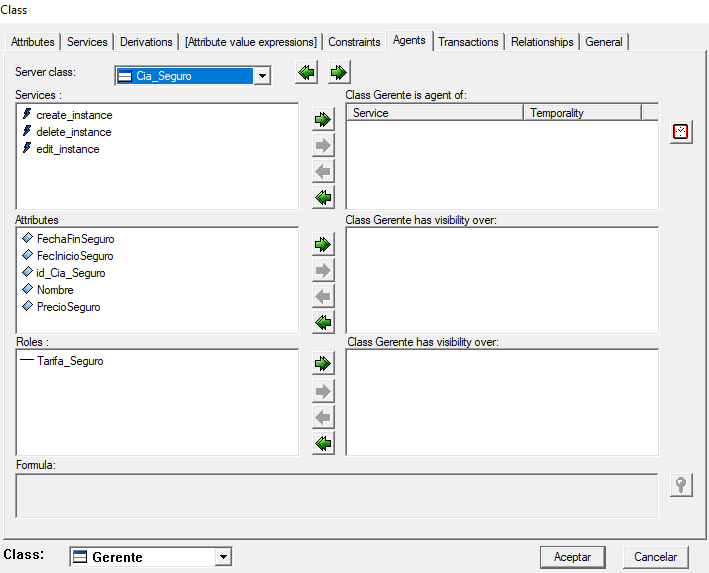


En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado a los usuarios.

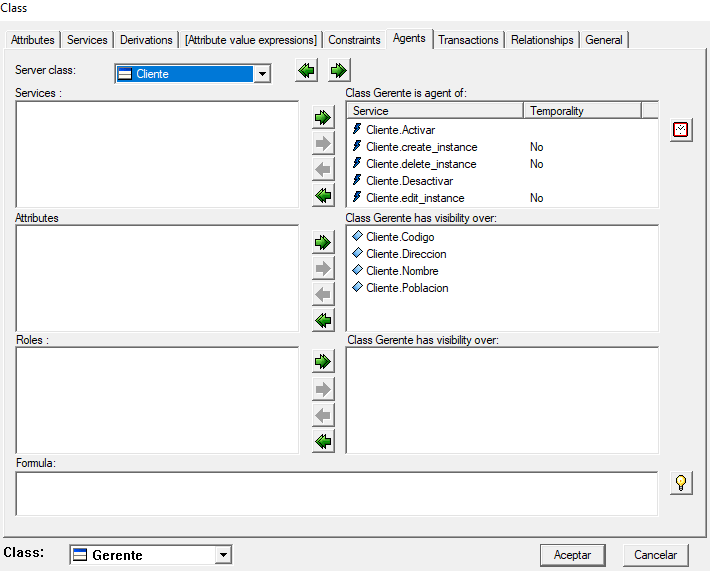
aaaaaaaa



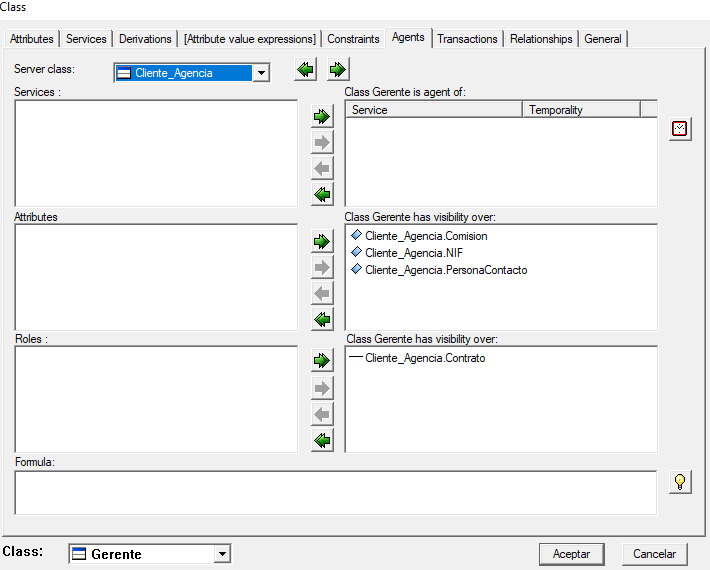
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado al automóvil.



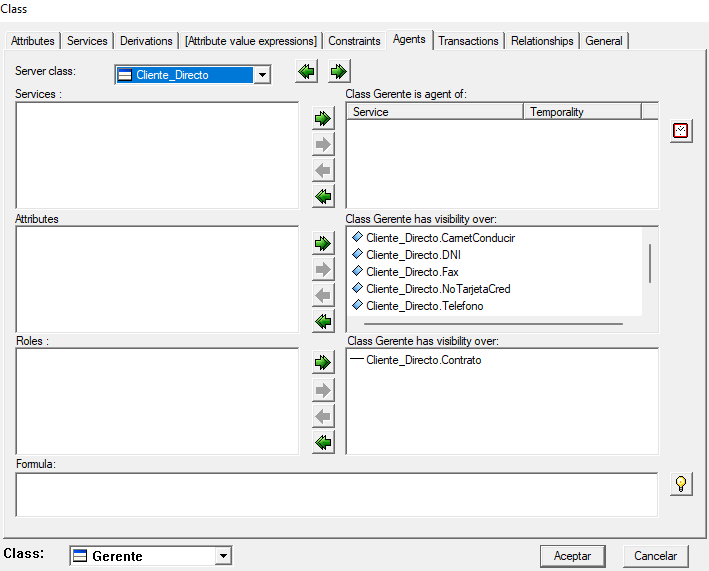
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado a la compañía de seguro.



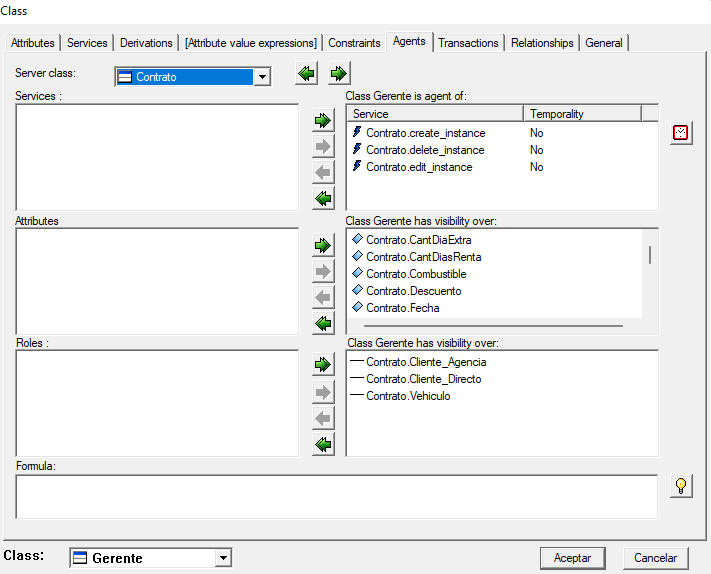
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado a los clientes.



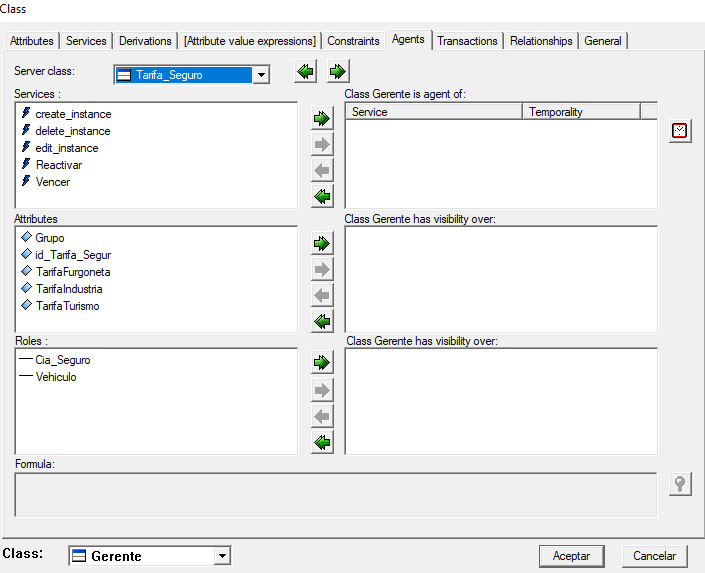
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado al cliente agencia.



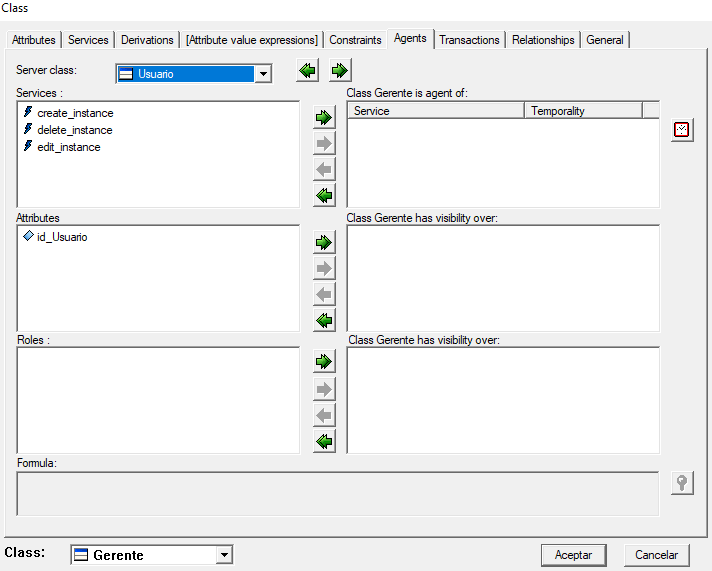
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado al cliente directo.



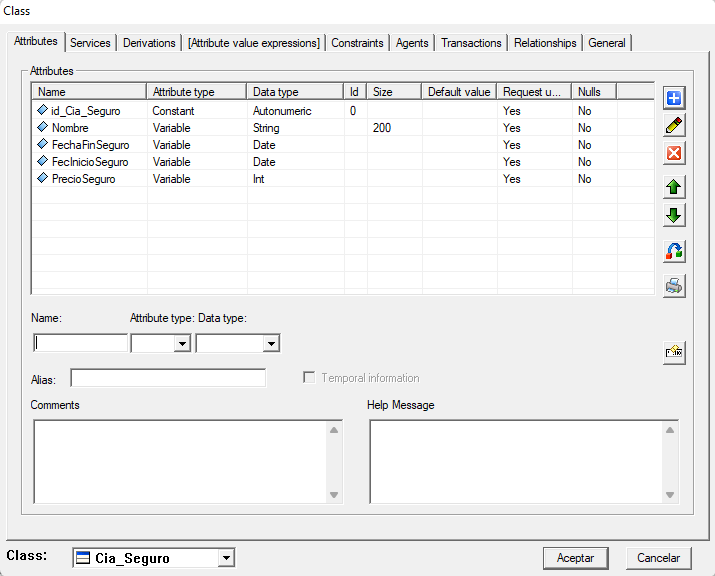
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado al contrato.



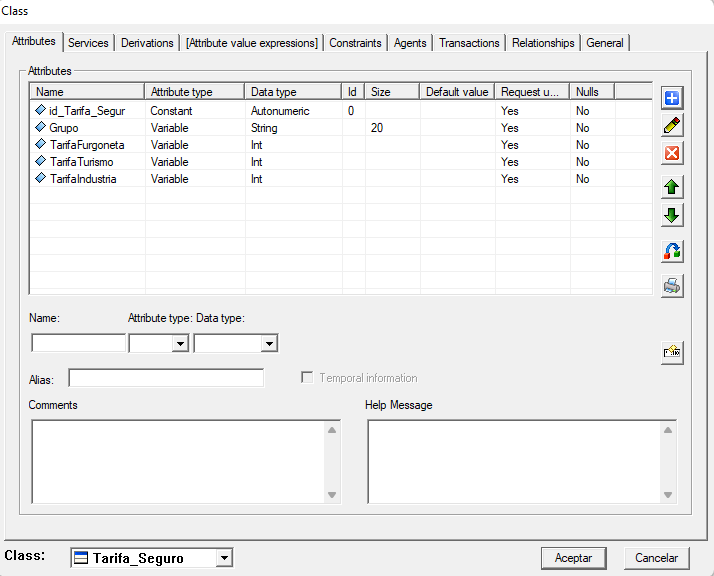
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el gerente frente a lo relacionado a la tarifa del seguro.



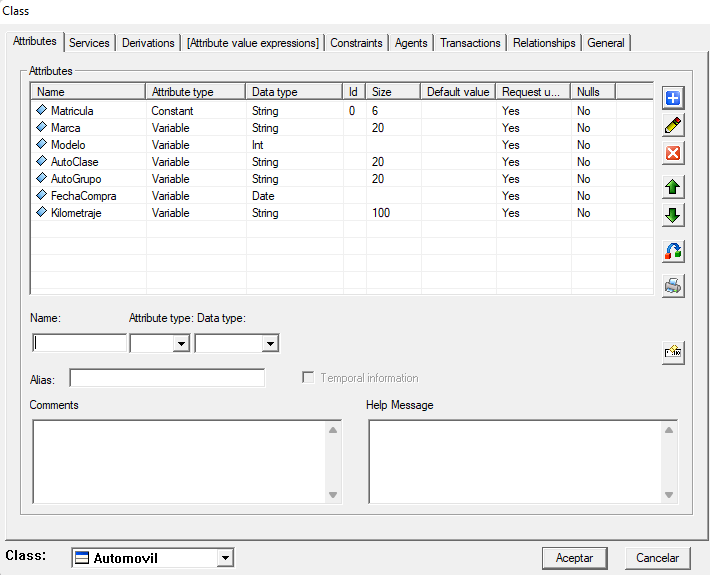
En esta ventana se puede ver que es lo que puede hacer y no puede hacer el administrador frente a lo relacionado a los usuarios.



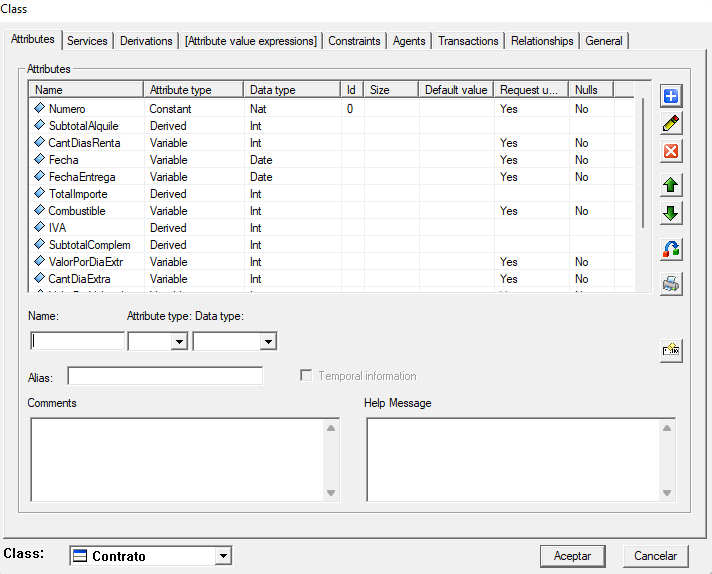
En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase cia\_seguro y podemos observar los atributos agregados junto a sus parámetros.



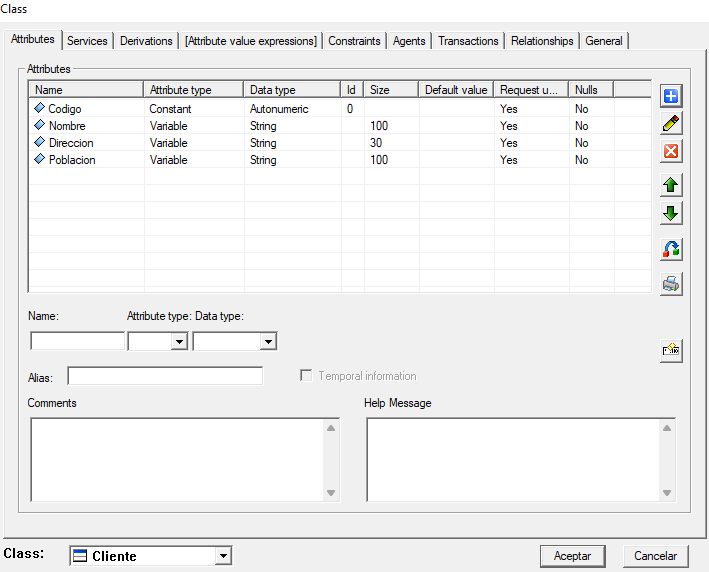
En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase tarifa\_seguro y podemos observar los atributos agregados junto a sus parámetros.



En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase automovil y podemos observar los atributos agregados junto a sus parámetros.

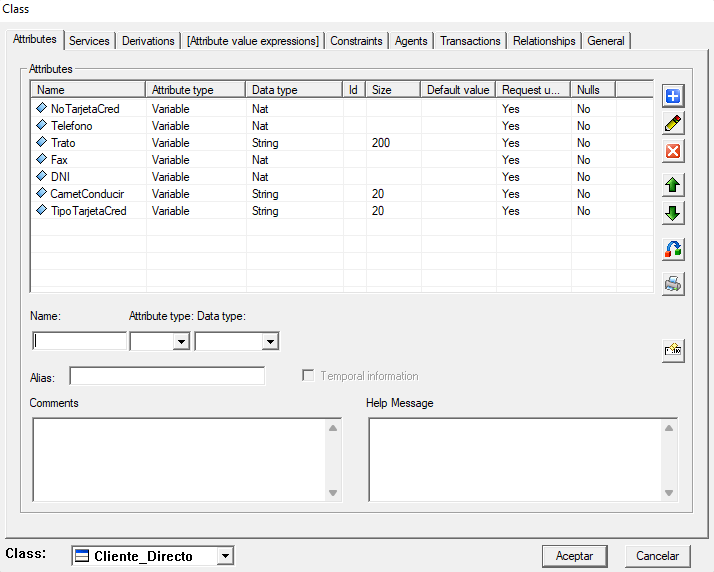
****

En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase contrato y podemos observar los atributos agregados junto a sus parámetros.

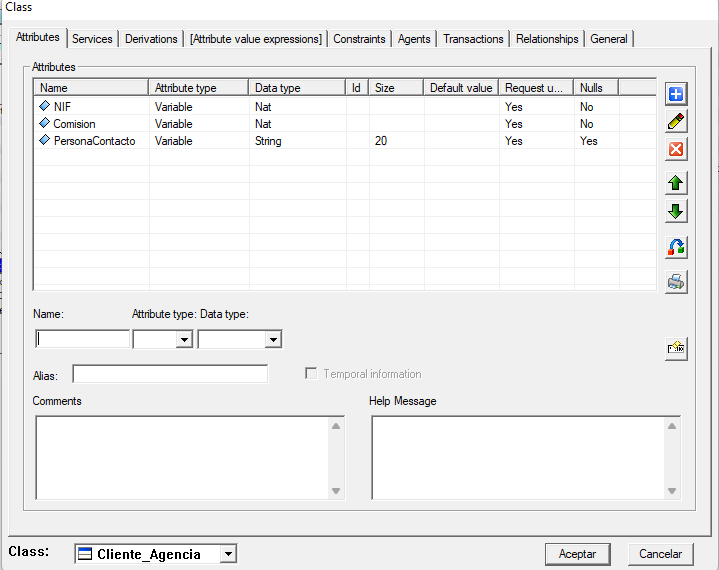
****

En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase cliente y podemos observar los atributos agregados junto a sus parámetros,

además ésta es una clase padre que tiene dos clases hijas que son el cliente directo y el cliente agencia.

****

En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase cliente\_directo y podemos observar los atributos agregados junto a sus parámetros.

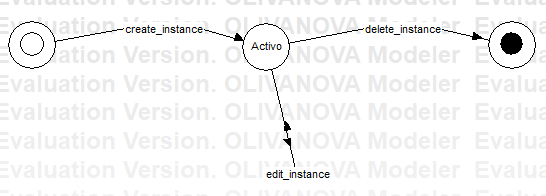
****

En esta imagen se puede ver una ventana que nos permite agregar los atributos a las clases en este caso tenemos la clase cliente\_agencia y podemos observar los atributos agregados junto a sus parámetros.

Diagrama de transición de estados (DTE)

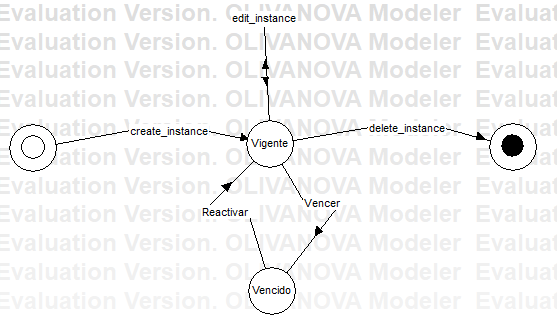
Un diagrama de transición de estados muestra el comportamiento dependiente del tiempo de un sistema de información. Representa los estados que puede tomar un componente o un sistema y muestra los eventos que implican el cambio de un estado a otro.

**Compañía de seguro**



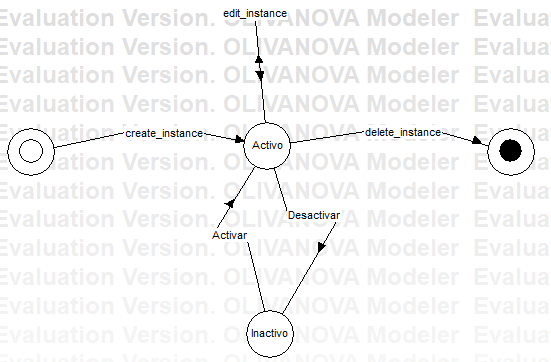
En este diagrama de transición de estados observamos que cia\_seguro a la hora de crearse pasa a estar en el estado activo después esta nos permite modificar o borrar y pasar al estado de muerta.

**Tarifa seguro**



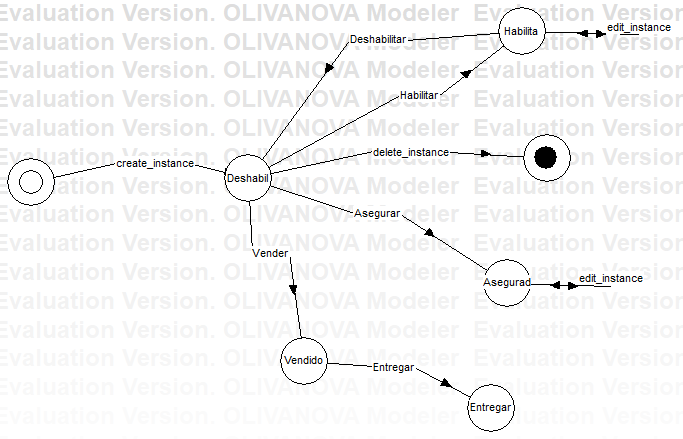
En este diagrama de transición de estados observamos que el seguro a la hora de crearse pasa a estar en el estado vigente y este puede quedar en estado vencido, mientras esté en vencido puede volverse a activar para que vuelva a quedar en vigente, mientras el seguro está en vigente se puede actualizar o eliminar.

**Cliente**



En este diagrama de transición de estados observamos que el cliente a la hora de crearse pasa a estar en el estado activo y este puede quedar en estado inactivo, mientras esté en estado inactivo puede volverse a activar para que vuelva a quedar en activo, mientras el cliente está en vigente se puede actualizar o eliminar.

**Automóvil**

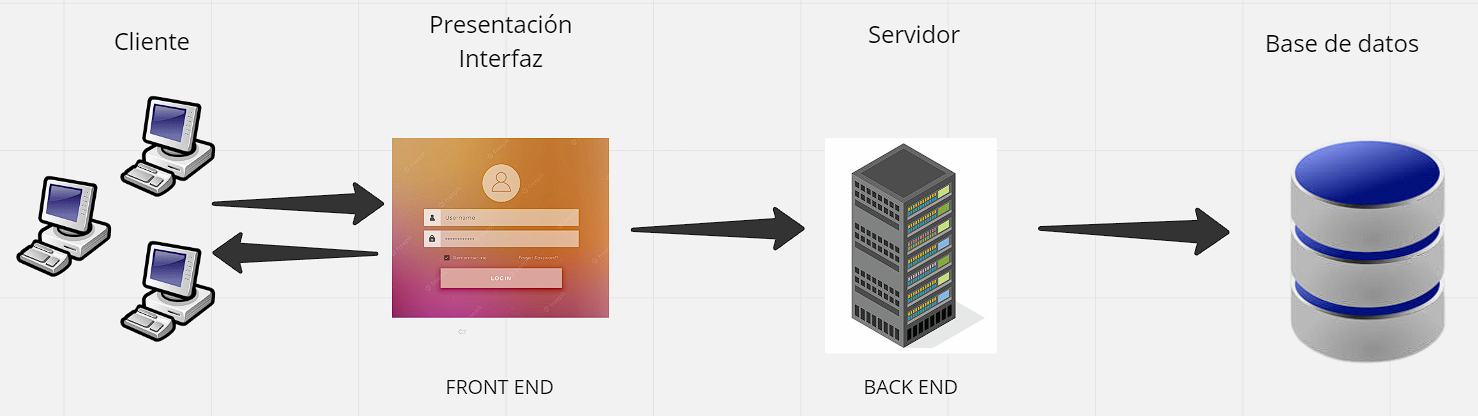


En este diagrama de transición de estados observamos que el auto al crearse queda en un estado deshabilitado, a la hora de que el auto se deshabilita puede pasar a un estado de habilitado y desde aquí se puede deshabilitar también, a su vez mientras el auto esté deshabilitado pasa a un estado de asegurado que es cuando se le asigna un seguro, a si mismo se puede pasar a un estado de vencido el cual se entrega, el auto si está deshabilitado se puede eliminar y por último el auto se puede modificar mientras esté asegurado y habilitado.

Sumado a lo anterior para cada una de las clases se le tienen que agregar servicios que son los que hacen que el diagrama de transición de estados tenga sentido las que están por defecto son la de crear, editar y eliminar, pero algunas de las clases tienen servicios adicionales como lo son: automovil, que tiene habilitar, deshabilitar, asegurar, vender y entregar, cliente que tiene activar y desactivar para saber si el cliente sea concurrente o no dentro del sistema y tarifa del seguro que tiene reactivar y vencer, ya que los seguros pueden pasar a los estados de vigente y vencido.

**Arquitectura cliente-servidor**

Además de haber hecho todo lo anterior se debe realizar la arquitectura cliente-servidor para tener una noción de lo que se realizará el programa, esta arquitectura lo que ayuda al proyecto es básicamente para saber qué es lo que hace el programa con los datos y los usuarios que ingresen a éste y interactúan por medio del front end (la interfaz), es decir, que los usuarios usen el programa y pidan información al back end (Servidor) sobre los datos que estén dentro de éste o tenga que consultarlos en la base de datos se les devuelva en forma de respuesta.



Por otro lado, el software a implementar deberá contar con una manera de almacenar los datos que serán administrados, para esto, se contará con una base de datos de tipo relacional, la cual debe almacenar los datos de los clientes, estos pueden ser de dos tipos el cliente directo y el cliente agencia, al cliente directo se le pedirán sus datos personales adicionando población, fax , carnet de conducción, población y tipo de tarjeta, por otro lado al cliente agencia se le pedirá el nombre, direccion, poblacion, nif, comisión y una persona de contacto.

Además, debe almacenar toda la información relacionada con los vehículos que hagan parte de la flota, registrando la marca, modelo, matrícula, clase, grupo, fecha de compra y kilometraje. Así mismo se deberán guardar todos los datos de las operaciones, almacenando datos como una descripción, fecha de envío, fecha de entrega, quien lo envió a mantenimiento y el taller al que fue enviado del cual se debe registrar nombre direccion y telefono.

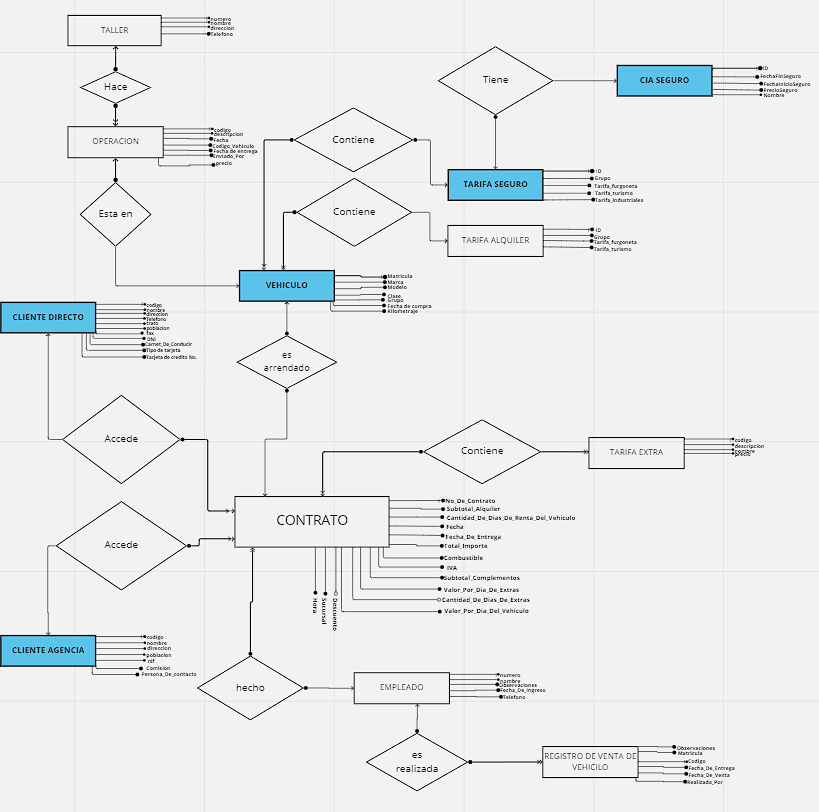
También, deberá tener registro del empleado y las ventas o envíos a reparación y mantenimiento que hagan, se registrará sobre estos su nombre, fecha de ingreso, teléfono y observaciones, si vende un vehículo se deberá registrar, matrícula. fecha de venta, fecha de entrega.

Sumado a esto, al contrato del vehículo se le deben adicionar una tarifa externa la cual será la tarifa extra, y al vehículo se le adicionará la tarifa de seguro, la compañía de seguros y tarifa de alquiler, la primera debe contener una descripción, el nombre y el precio que tenga, las otras deben tener el grupo del vehículo y la tarifa de furgoneta y turismo.

A la hora de firmar un contrato este debe tener los datos del cliente, los datos del empleado que lo realizó, los datos del vehículo arrendado, los datos de todas las tarifas, así como el subtotal del alquiler, cantidad de días que será alquilado el vehículo, fecha, fecha de entrega del vehículo, total del importe, combustible, IVA, subtotal complementos, valor total por dia de extras, valor total por dia del vehículo (incluyendo el seguro), descuento si aplica, sucursal donde se firmó y la hora.

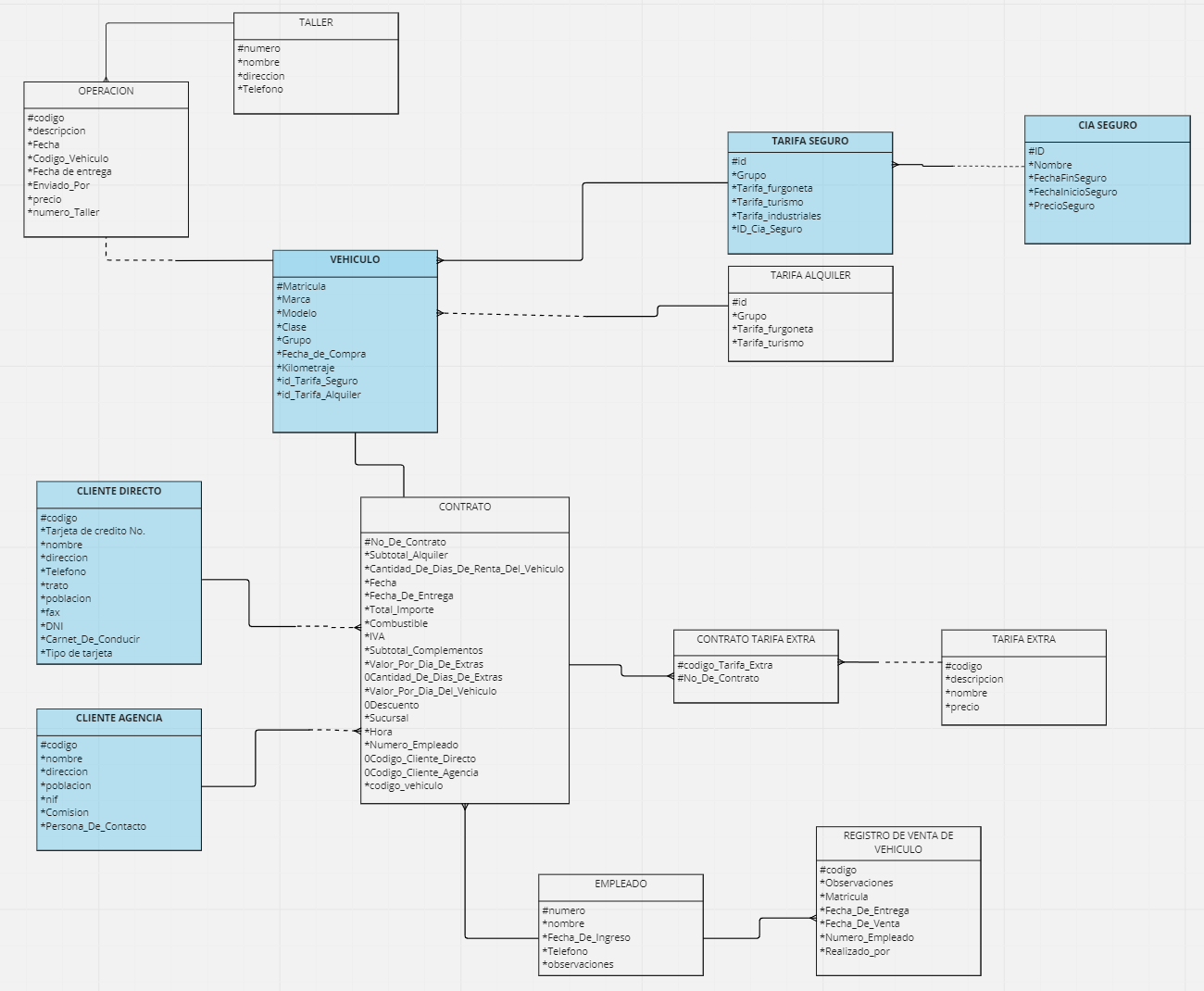
**MODELO ENTIDAD-RELACIÓN**

En esta imagen encontraremos el modelo entidad-relación propuesto para la realización del proyecto y en color azul las entidades que corresponden a nuestro grupo de trabajo



**MODELO RELACIONAL**

En la siguiente imagen encontraremos nuestro modelo relacional basado en el anterior modelo entidad-relación, se encontraran en color azul las entidades asignadas a nuestro grupo de trabajo.

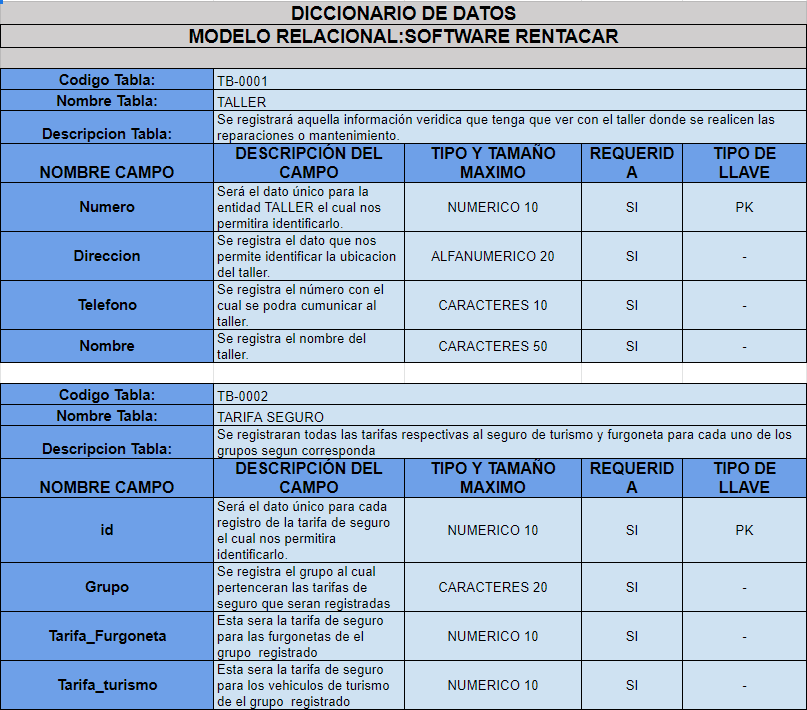


Vínculo para poder observar el modelo entidad-relación y el modelo relacional en Miro:

<https://miro.com/app/board/uXjVO8U5fxU=/?share_link_id=510265437257>

En las siguientes imágenes encontraremos el desarrollo del diccionario de datos basado en el modelo relacional el cual nos permitirá organizar la información para el desarrollo y creación de la base de datos.

<https://docs.google.com/spreadsheets/d/1KPGp9JxxbKSqnxCE4Wgl_bVCM-seguqj/edit?usp=sharing&ouid=118051998181042068789&rtpof=true&sd=true>

[](https://docs.google.com/spreadsheets/d/1KPGp9JxxbKSqnxCE4Wgl_bVCM-seguqj/edit?usp=sharing&ouid=118051998181042068789&rtpof=true&sd=true)

El diccionario de datos está en el enlace anclado a la imagen, ya que es muy extenso para añadirlo.

Luego de haber creado el diccionario de datos, lo que se procede a realizar es la creación de las tablas de la base de datos con respecto a las gestiones expuestas en el árbol de refinamiento funcional (ARF).

**Sentencias de creación de la base de datos**

1. **Creación de la tabla del cliente directo.**

CREATE TABLE cliente\_directo

(Codigo NUMBER(10) CONSTRAINT cld\_cod\_pk PRIMARY KEY,

Tarjeta\_de\_Credito NUMBER(20) CONSTRAINT cld\_tdc\_nn NOT NULL,

Nombre VARCHAR(20) CONSTRAINT cld\_nom\_nn NOT NULL,

Direccion VARCHAR(15) CONSTRAINT cld\_dir\_nn NOT NULL,

Telefono NUMBER(10) CONSTRAINT cld\_tel\_nn NOT NULL,

Trato VARCHAR(20) CONSTRAINT cld\_trt\_nn NOT NULL,

Poblacion VARCHAR(20) CONSTRAINT cld\_pob\_nn NOT NULL,

Fax VARCHAR(11) CONSTRAINT cld\_fax\_nn NOT NULL,

Dni NUMBER(12) CONSTRAINT cld\_dni\_nn NOT NULL,

Carnet\_De\_Conducir VARCHAR(15) CONSTRAINT cld\_cdc\_nn NOT NULL,

Tipo\_De\_Tarjeta VARCHAR(14) CONSTRAINT cld\_tdt\_nn NOT NULL);

1. **Creación de la tabla del cliente agencia.**

CREATE TABLE cliente\_agencia

(Codigo NUMBER(10) CONSTRAINT cla\_cod\_pk PRIMARY KEY,

Nombre VARCHAR(50) CONSTRAINT cla\_nom\_nn NOT NULL,

Direccion VARCHAR(15) CONSTRAINT cla\_dir\_nn NOT NULL,

Poblacion VARCHAR(20) CONSTRAINT cla\_pob\_nn NOT NULL,

Nif NUMBER(20) CONSTRAINT cla\_nif\_nn NOT NULL,

Comision NUMBER(15) CONSTRAINT cla\_com\_nn NOT NULL,

Persona\_Contacto VARCHAR(50) CONSTRAINT cla\_percon\_nn NOT NULL);

1. **Creación de la tabla de cia seguro.**

CREATE TABLE cia\_seguro

(Id NUMBER(10) CONSTRAINT cia\_id\_pk PRIMARY KEY,

Nombre VARCHAR(20) CONSTRAINT cia\_nom\_nn NOT NULL,

Fecha\_fin\_seguro DATE CONSTRAINT cia\_dir\_nn NOT NULL,

Fecha\_inicio\_seguro DATE CONSTRAINT cia\_pob\_nn NOT NULL,

Precio\_seguro NUMBER(10) CONSTRAINT cia\_nif\_nn NOT NULL);

1. **Creación de la tabla de tarifa seguro.**

CREATE TABLE tarifa\_seguro

(Id NUMBER(10) CONSTRAINT tfs\_id\_pk PRIMARY KEY,

Grupo VARCHAR(20) CONSTRAINT tfs\_grp\_nn NOT NULL,

Tarifa\_Furgoneta NUMBER(10) CONSTRAINT tfs\_trf\_nn NOT NULL,

Tarifa\_Turismo NUMBER(10) CONSTRAINT tfs\_tat\_nn NOT NULL,

Id\_Cia\_Seguro NUMBER(10) CONSTRAINT tfs\_ics\_fk REFERENCES cia\_seguro(Id));

1. **Creación de la tabla de vehículo.**

CREATE TABLE vehiculo

(Matricula VARCHAR(6) CONSTRAINT veh\_mat\_pk PRIMARY KEY,

Marca VARCHAR(20) CONSTRAINT veh\_mar\_nn NOT NULL,

Clase VARCHAR(10) CONSTRAINT veh\_clas\_nn NOT NULL,

Grupo VARCHAR(10) CONSTRAINT veh\_gru\_nn NOT NULL,

Fecha\_de\_compra DATE CONSTRAINT veh\_feccom\_nn NOT NULL,

Kilometraje NUMBER(15) CONSTRAINT veh\_kil\_nn NOT NULL,

Modelo VARCHAR(10) CONSTRAINT veh\_mod\_nn NOT NULL,

Id\_tarifa\_seguro NUMBER(10) CONSTRAINT veh\_idtfs\_nn NOT NULL CONSTRAINT veh\_idtfs\_fk REFERENCES tarifa\_seguro(Id));

1. **Creación de la tabla contrato.**

CREATE TABLE contrato

(No\_de\_Contrato NUMBER(10) CONSTRAINT con\_noc\_pk PRIMARY KEY,

Subtotal\_Alquiler NUMBER(15) CONSTRAINT con\_subal\_nn NOT NULL,

Cantidad\_De\_Renta\_Del\_Vehiculo NUMBER(10) CONSTRAINT con\_cantdia\_nn NOT NULL,

Fecha DATE CONSTRAINT con\_fec\_nn NOT NULL,

Fecha\_De\_Entrega DATE CONSTRAINT con\_fecent\_nn NOT NULL,

Total\_Importe NUMBER(15) CONSTRAINT con\_toim\_nn NOT NULL,

Combustible NUMBER(10) CONSTRAINT con\_com\_nn NOT NULL,

Iva NUMBER(10) CONSTRAINT con\_iva\_nn NOT NULL,

Subtotal\_Complementos VARCHAR(52) CONSTRAINT con\_scm\_nn NOT NULL,

Valor\_Por\_Dia\_De\_Extras NUMBER(10) CONSTRAINT con\_vdex\_nn NOT NULL,

Cantidad\_De\_Dias\_De\_Extras NUMBER(10),

Valor\_Por\_Dia\_Del\_Vehiculo NUMBER(10) CONSTRAINT con\_vdv\_nn NOT NULL,

Descuento NUMBER(10),

Sucursal VARCHAR(20) CONSTRAINT con\_suc\_nn NOT NULL,

Hora NUMBER(10) CONSTRAINT con\_hor\_nn NOT NULL,

Codigo\_cliente\_directo NUMBER(10) CONSTRAINT con\_codcld\_fk REFERENCES cliente\_directo(codigo),

Codigo\_cliente\_agencia NUMBER(10) CONSTRAINT con\_codcla\_fk REFERENCES cliente\_agencia(codigo),

Matricula\_vehiculo VARCHAR(6) CONSTRAINT con\_matveh\_nn NOT NULL CONSTRAINT con\_matveh\_fk REFERENCES vehiculo(Matricula));

**Tablas creadas**

A continuación se muestran las tablas creadas en el gestor de la base de datos Oracle.

Todas las tablas creadas.

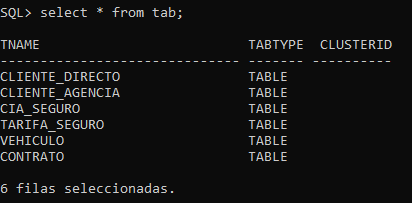


Tabla del cliente directo.

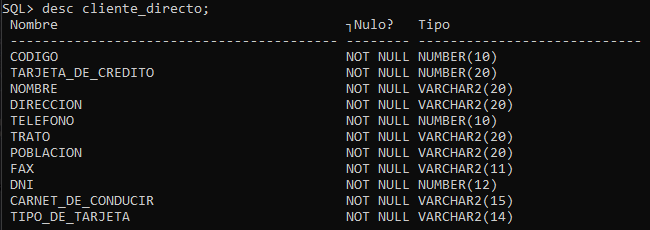


Tabla del cliente agencia.

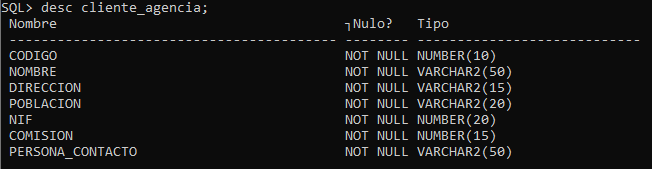


Tabla de Cia seguro.

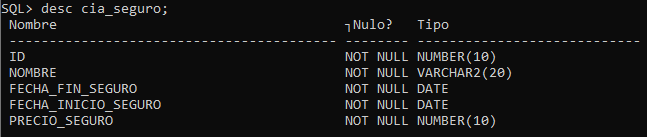


Tabla de tarifa seguro.

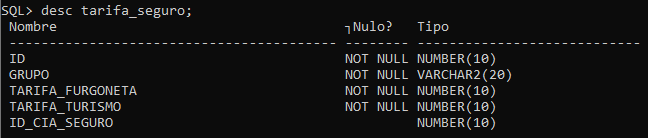


Tabla del vehículo.

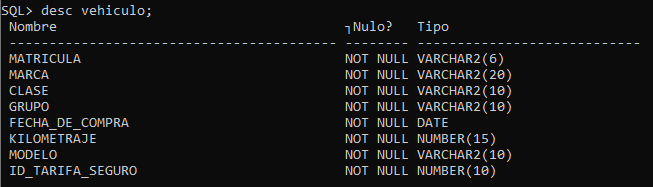


Tabla del contrato.



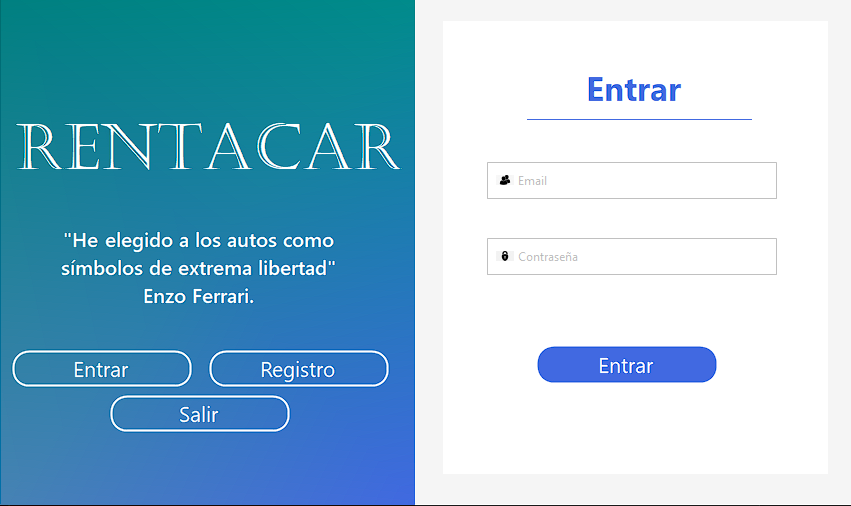
Luego de revisar que las tablas de la base de datos están creadas, se procede a realizar la aplicación de escritorio, la cual va a permitir crear los usuarios con su nombre y contraseña para que se pueda entrar y revisar lo que ofrece la aplicación, que en este caso sería crear, borrar y modificar los datos introducidos al apartado de seguros, clientes, compañía de seguros, contrato y cliente intermediario (cliente agencia), para el auto tenemos funciones adicionales que son habilitar, deshabilitar, vender y entregar. Además a cada uno de los apartados mencionados anteriormente se tiene para cada uno de estos la base de datos vinculada que hace referencia a los mismos.

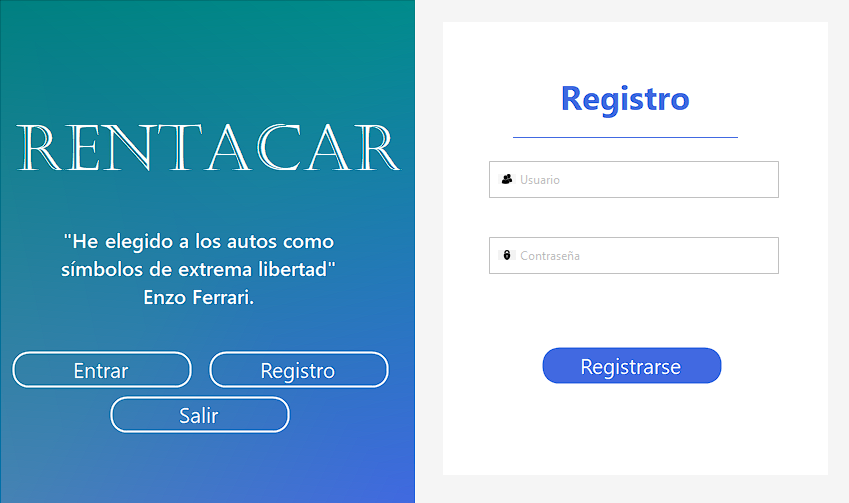
**SENTENCIAS**

En la base de datos tenemos cuatro clases de procedimientos, estos procedimientos son los que nos permiten hacer el CRUD desde el programa, así como también visualizar las vistas en el programa.

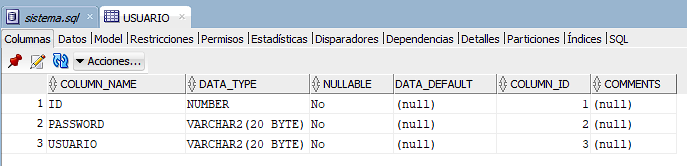
En el entregable se dejará el anexo SentenciasRentacar.txt en el que se podrán observar las sentencias utilizadas y el CRUD

En la primera imagen encontraremos la interfaz de Entrar que se despliega al seleccionar el botón entrar, en ella pondremos nuestro usuario y contraseña registrado para acceder al menú principal.

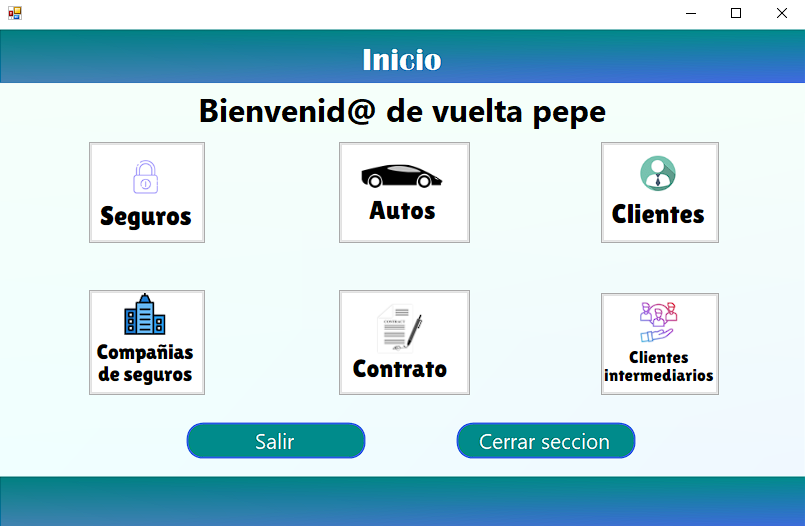


En esta imagen se mostrará la interfaz de registro de usuario, esta se desplegará al oprimir el botón registro y en ella se podrá ingresar el usuario y contraseña para registrarse.

Esta es la tabla donde se almacenarán los usuarios registrados en el sistema, estos serán los únicos en tener acceso.



Este es el menú principal del programa, este nos da la bienvenida con el nombre de usuario y nos permite ingresar a las diferentes gestiones del software.



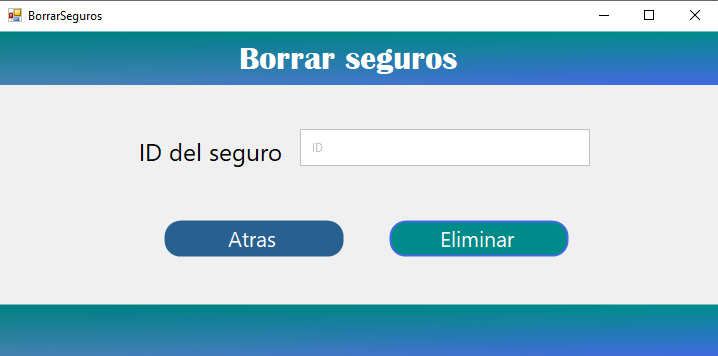
Al ingresar al menu de seguros podemos observar las cuatro funciones base



En el menú crear seguro encontramos los campos necesarios para ingresar los datos de un registro de seguro que queramos hacer



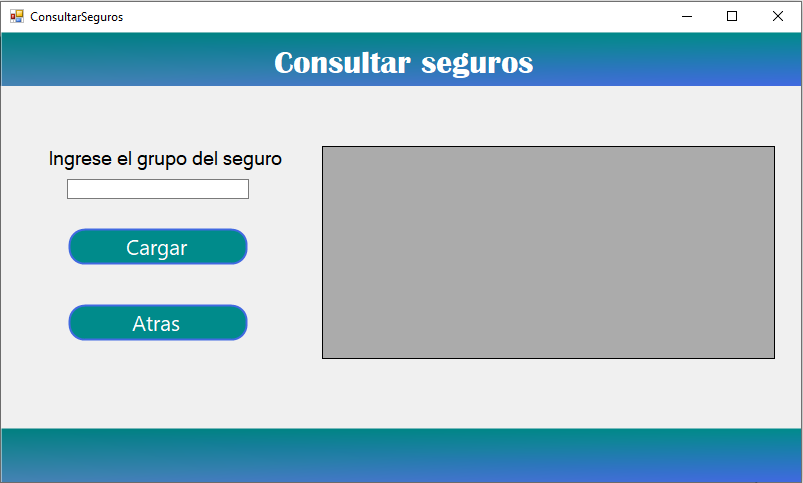
En el menú borrar seguros encontramos un campo que nos permite ingresar el id del seguro que deseamos borrar, al darle a eliminar se llamará un procedimiento que llevará a cabo el delete



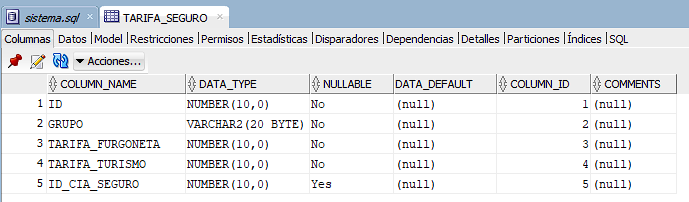
En el menú modificar seguros tenemos los campos necesarios para ingresar los datos con los que queremos que quede un registro que ya estaba creado



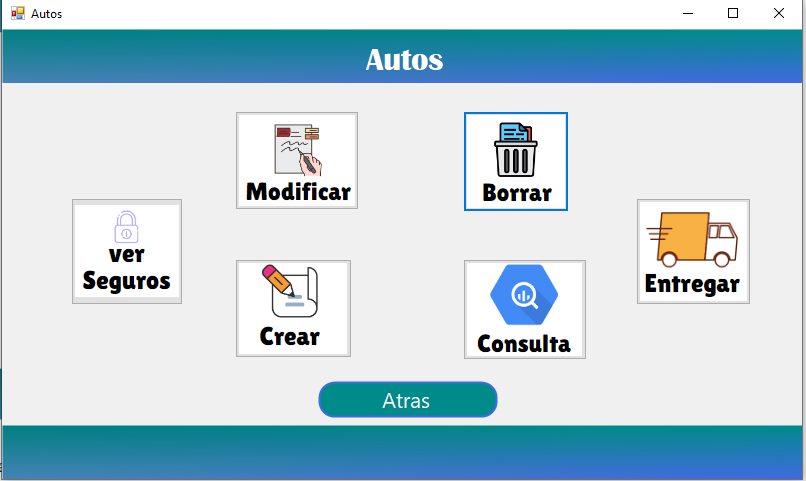
En el menú consultar seguro podemos hacer una búsqueda de los seguros mediante su grupo



Esta es la tabla seguro donde se almacenarán los registros de seguro



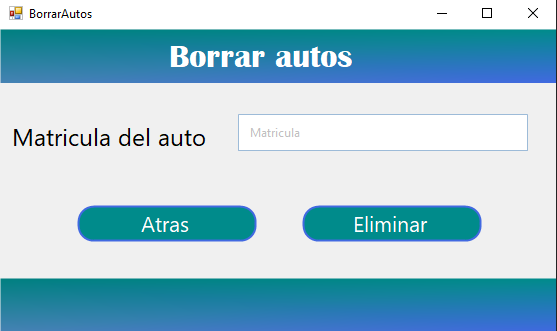
En el menú de autos tenemos algunas opciones extras ya que aquí podremos ver las vistas en los menús de entregas y ver seguros.



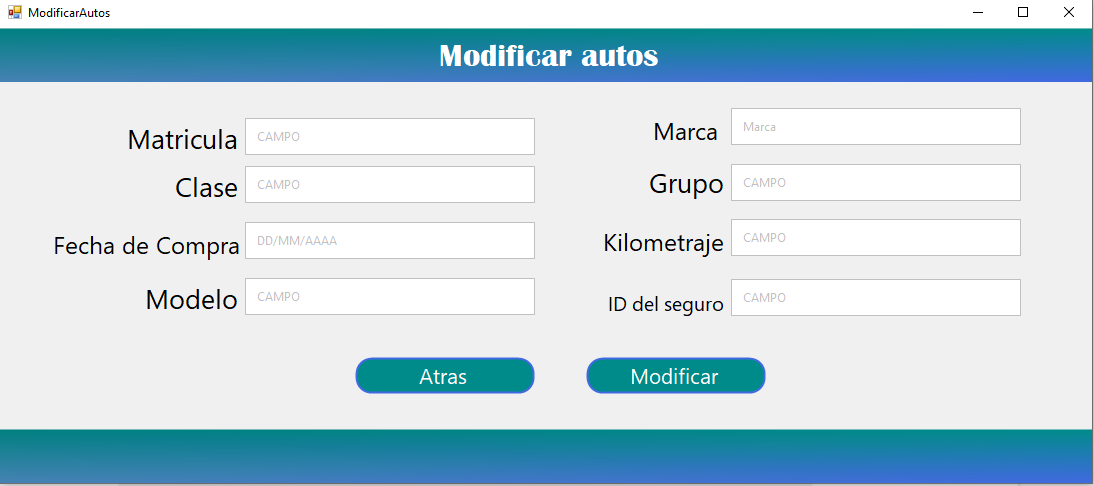
El menú de crear auto tiene todos los campos para ingresar los datos del auto que deseamos registrar



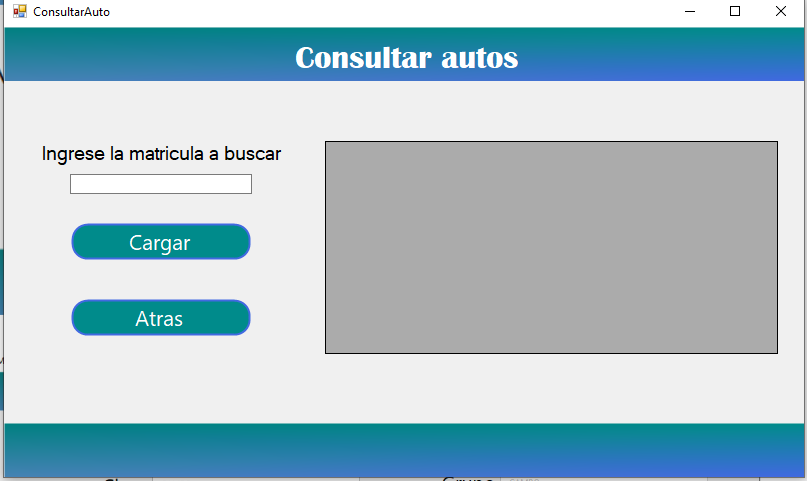
En el menú de borrar auto podemos ingresar la matrícula del auto que deseamos eliminar



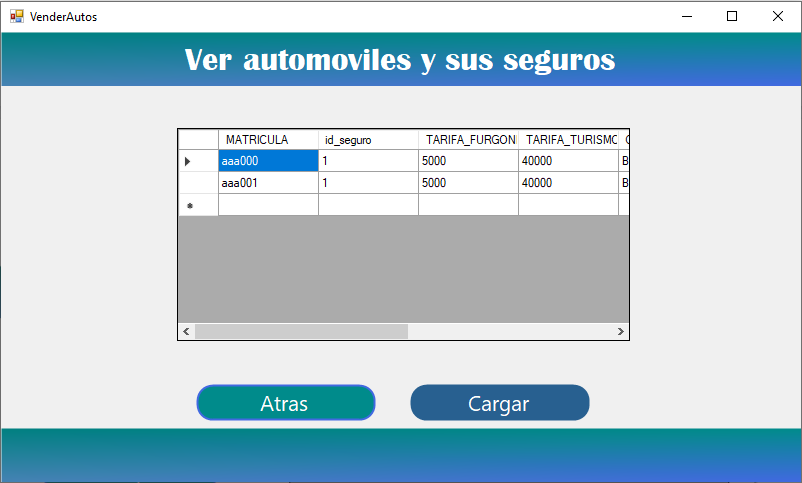
En el menú de modificar tenemos todos los campos necesarios para ingresar los datos que queremos que queden registrados en el auto



En el menú de consultar un auto podemos ingresar una matricula para ver si esta registrado



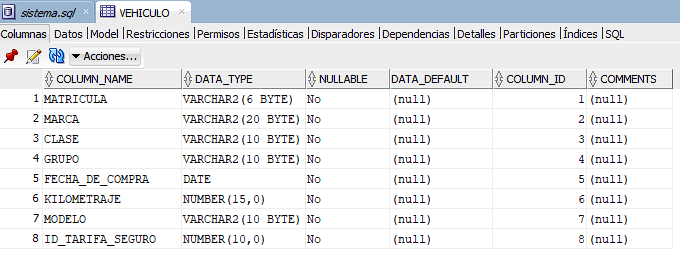
La vista de ver automóviles y seguro es para poder estar pendiente de los seguros y tarifas de los autos.



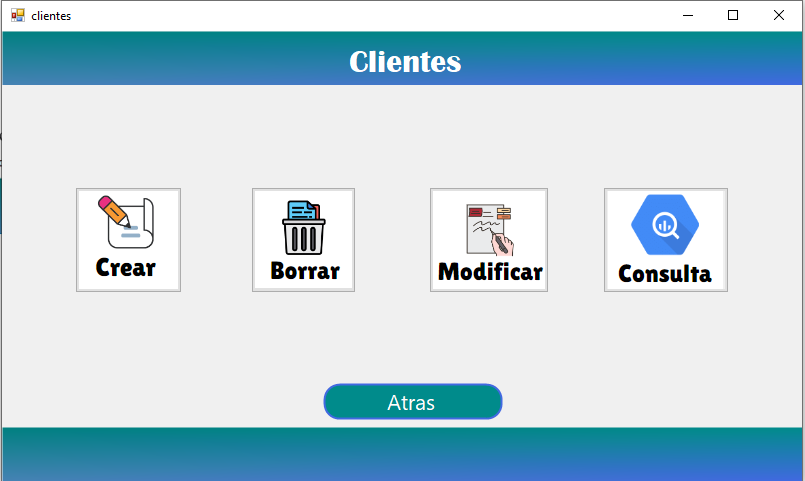
En la vista de entregar autos podemos ver la fecha que tienen de entrega los autos.



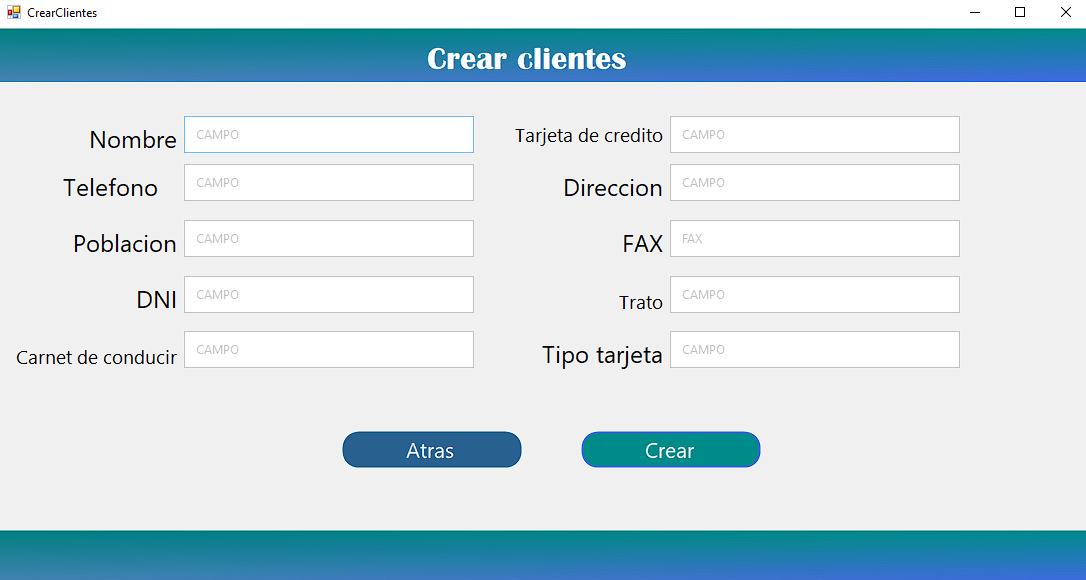
Esta es la tabla donde se almacenarán todos los registros de vehículo que se hagan en el programa.



En el menú cliente tenemos las cuatro funcionalidades básicas para poder interactuar con los datos de los clientes.



Este es el formulario para crear un cliente, están los botones para crear y de ir atrás.



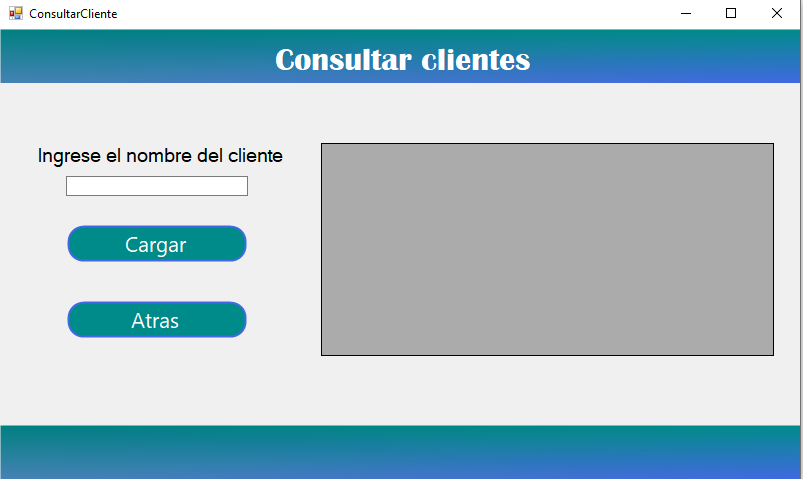
Este es el formulario para borrar un cliente registrado, tiene los botones de id y atrás, solo es necesario el código para realizar el borrado.



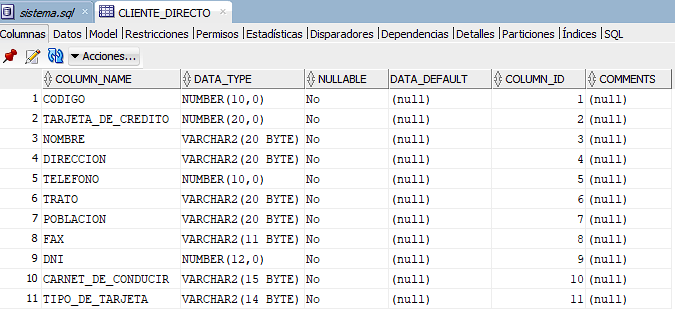
Este es el formulario que nos permite modificar un cliente registrado, tiene los botones de modificar y atrás.



En el menú consultar clientes podemos hacer una búsqueda de un registro mediante el nombre de un cliente.



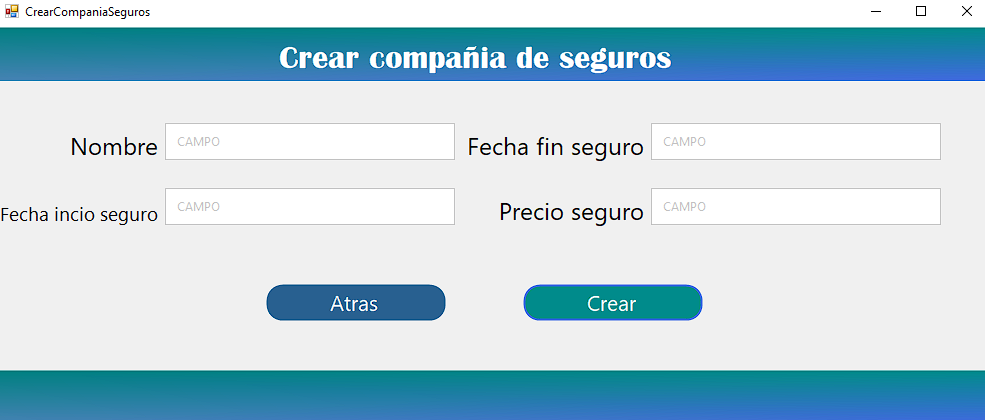
Esta es la tabla donde se almacenarán todos los registros de clientes (cliente directo) que se hagan en el programa.



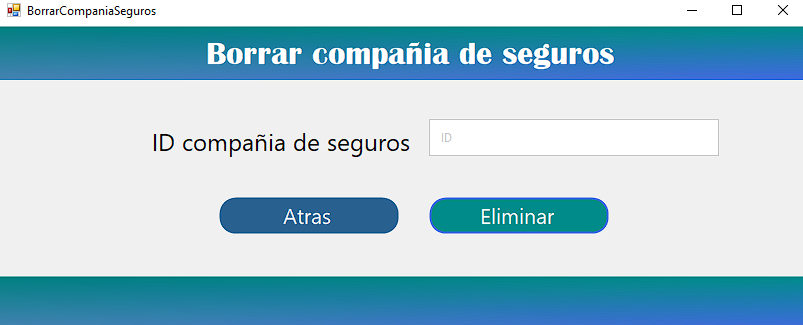
En el menú de compañía de seguros podemos encontrar las cuatro funcionalidades básicas, estas nos permiten gestionar los registros de compañía de seguros.



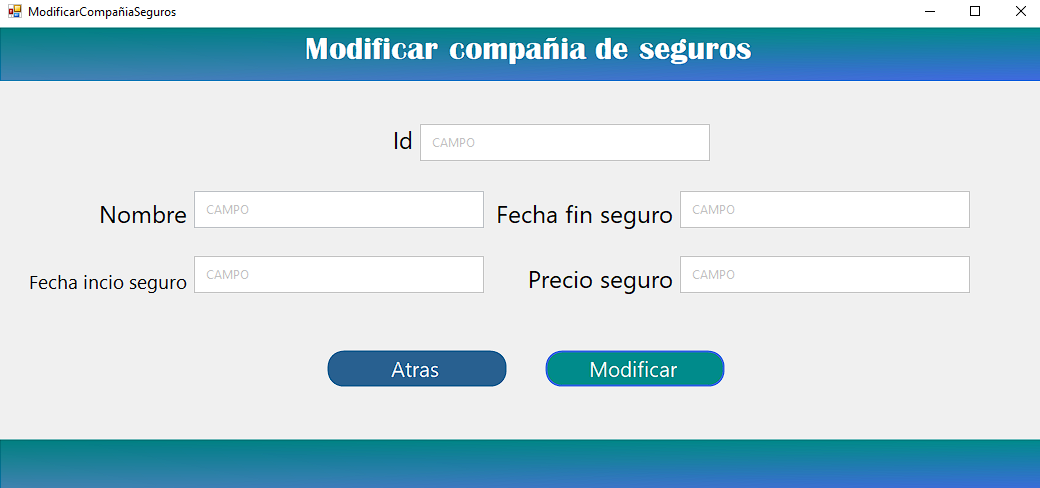
Este es el formulario para crear una compañía de seguros, están los botones para crear y de ir atrás.



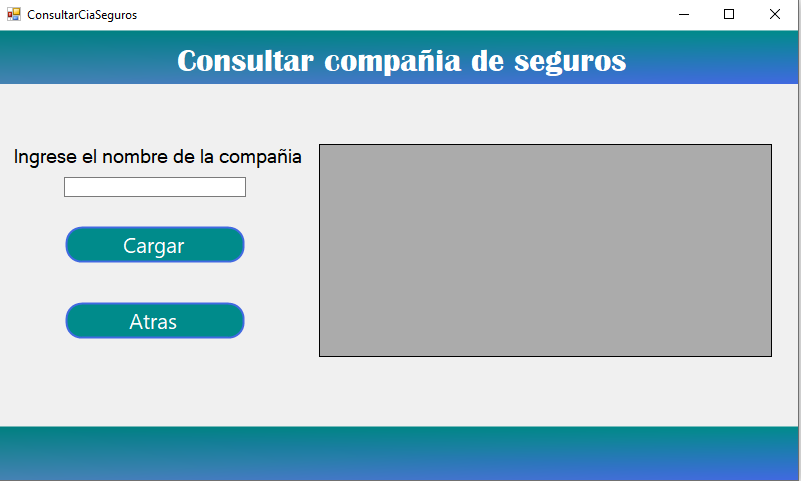
Este es el formulario para borrar una compañía de seguros registrada, tiene los botones de id y atrás, solo es necesario el código para realizar el borrado.



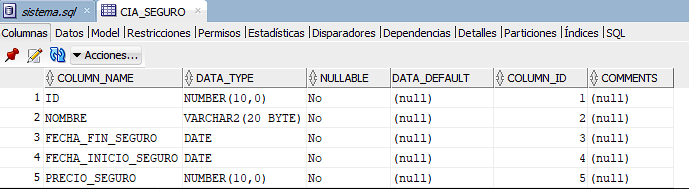
Este es el formulario que nos permite modificar una compañía de seguros registrada, tiene los botones de modificar y atrás.



En el menú de consultar compañía de seguros podemos buscar el registro de una compañía mediante su nombre



Esta es la tabla donde se almacenarán todos los registros de compañía de seguro(cia seguro) que se hagan en el programa.



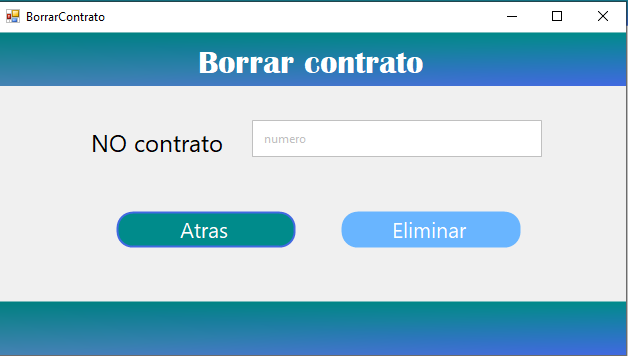
En el menú de contrato podemos encontrar las cuatro funcionalidades básicas, estas nos permitirán gestionar los registros de contrato



En el menú de crear contrato tenemos todos los campos necesarios para hacer el registro de un contrato, además nos permite dejar nulos los campos que pueden estar vacíos en la base de datos



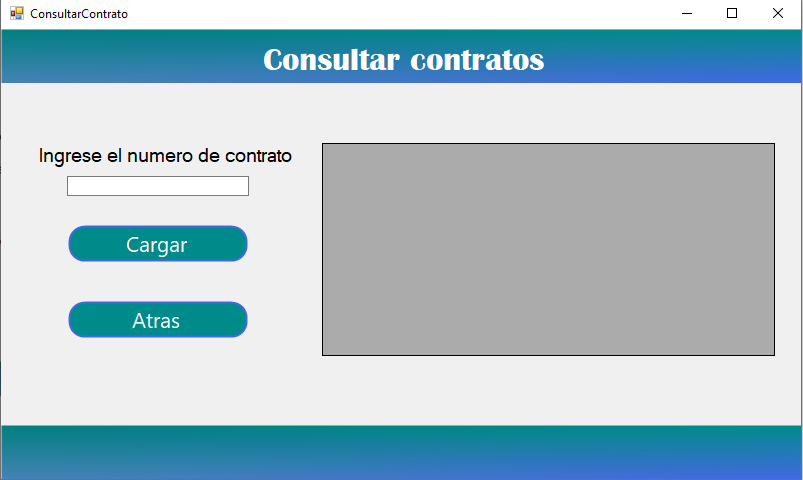
El menú de borrar contrato nos permite borrar un contrato ingresando su número de registro.



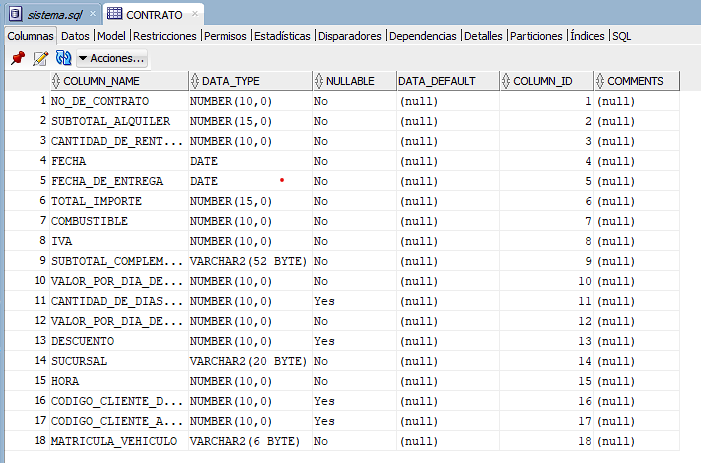
En el menú modificar contrato podremos encontrar los campos necesarios para poner los datos con los que queremos que quede un contrato ya registrado, muy útil para añadir extras y hacer cambios inesperados que el cliente podría solicitar



En el menú consultar contrato podemos buscar un registro de contrato mediante su numero.



Esta es la tabla donde se almacenarán todos los registros de contrato que se hagan en el programa.



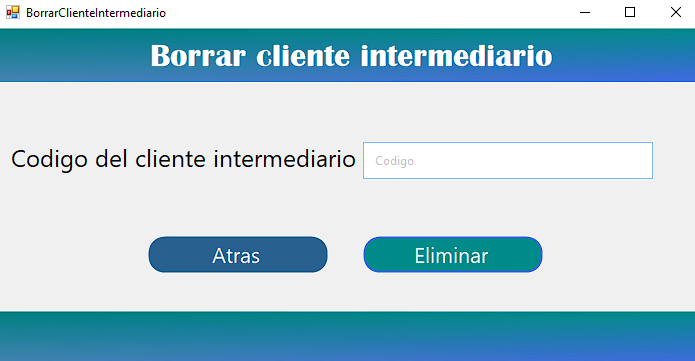
En el menú cliente intermediario podemos encontrar las cuatro funciones básicas, estas permitirán gestionar los registros que se tengan de clientes intermediarios.



Este es el formulario para crear un cliente intermediario (cliente agencia), están los botones para crear y de ir atrás.



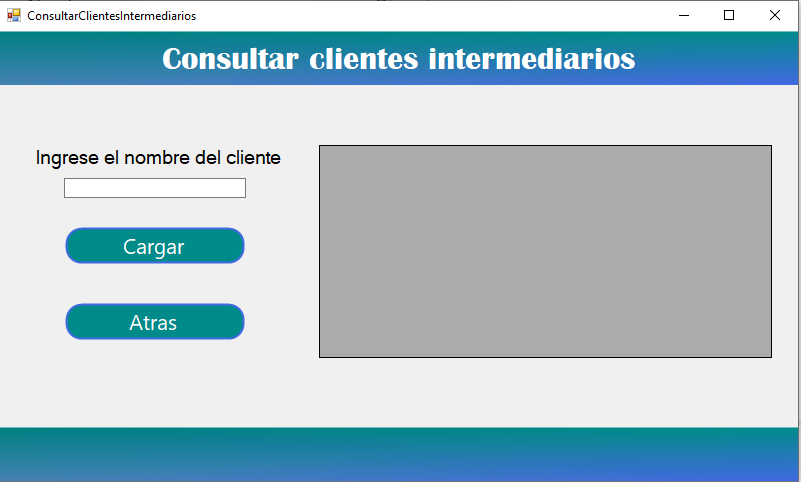
Este es el formulario para borrar un cliente intermediario registrado, tiene los botones de id y atrás, solo es necesario el código para realizar el borrado.



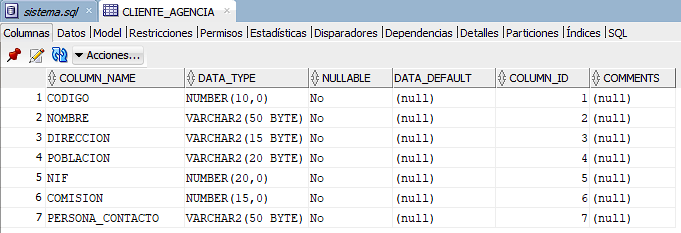
Este es el formulario que nos permite modificar un cliente intermediario registrado, tiene los botones de modificar y atrás.



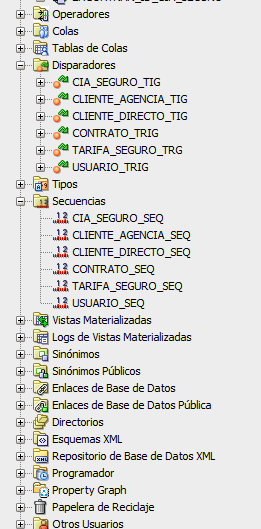
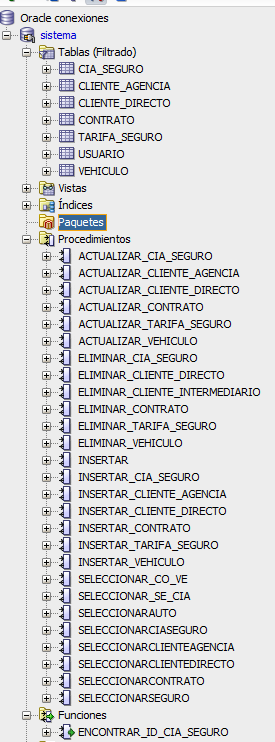
En el menú consultar clientes intermediarios nos permite buscar a un cliente int mediante su nombre para poder ver su registro,



Esta es la tabla donde se almacenarán todos los registros de cliente intermediaria(cliente agencia) que se hagan en el programa.



Contiene todo lo que hace referencia a la base de datos del programa.

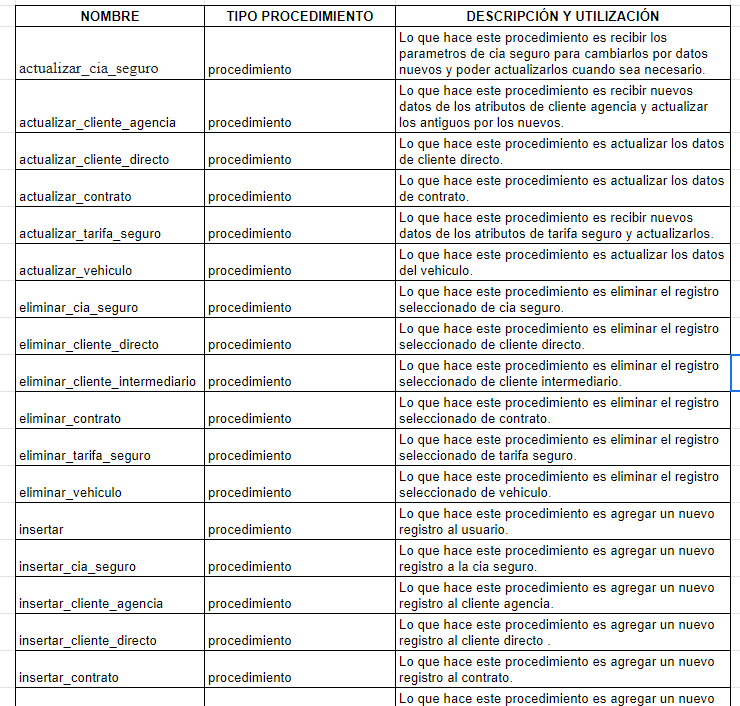


A continuación, estará el link del video donde se podrá ver toda la interfaz del login y menús, además de cómo podemos actualizar la base de datos desde el programa.

<https://drive.google.com/file/d/1-KhieNFzYER7S3IGIqJGbOGKGqC6p81H/view?usp=sharing>

**SENTENCIAS**

En la imagen está el enlace anclado para revisar las sentencias realizadas.

[](https://docs.google.com/spreadsheets/d/1elRVWDj1hXRGc8y0OSTGqsk2WCUOvfnOyd0jUrwHKlk/edit?usp=sharing)

**CONSULTAS**

**1. Incluir una nueva tarifa seguro al sistema( se da el nombre de la cia seguro)**

ENTRADA:

CREATE OR REPLACE FUNCTION f\_bus\_id\_ciaseg

(v\_nom\_cia\_seguro cia\_seguro.nombre%TYPE)

RETURN NUMBER

IS

v\_id\_ciaseg cia\_seguro.id%TYPE;

BEGIN

SELECT id INTO v\_id\_ciaseg

FROM cia\_seguro

WHERE UPPER(nombre) LIKE v\_nom\_cia\_seguro;

RETURN v\_id\_ciaseg;

END;

CREATE OR REPLACE PROCEDURE p\_incTarseg

(r\_tarifaSeguro tarifa\_seguro%ROWTYPE)

IS

BEGIN

INSERT INTO tarifa\_seguro

VALUES r\_tarifaSeguro;

END;

DECLARE

r\_tarifa\_Seguro tarifa\_seguro%ROWTYPE;

v\_nom\_cia\_seguro cia\_seguro.nombre%TYPE:=UPPER('&nombreCompañiaSeguro');

BEGIN

r\_tarifa\_Seguro.id:= &idTarifaseguro;

r\_tarifa\_Seguro.grupo:= '&grupoTarifaseguro';

r\_tarifa\_Seguro.Tarifa\_furgoneta:= &grupoTarifafurgoneta;

r\_tarifa\_Seguro.Tarifa\_turismo:= &grupoTarifaturismo;

r\_tarifa\_Seguro.Id\_cia\_seguro:= f\_bus\_id\_ciaseg(v\_nom\_cia\_seguro);

p\_incTarseg (r\_tarifa\_Seguro);

COMMIT;

END;

**2. Ver los seguros que se vencen este mes**

CREATE OR REPLACE FUNCTION f\_buscarseguros

RETURN varchar2

IS

CURSOR c\_seguro IS

SELECT cia.id,cia.nombre,cia.FECHA\_FIN\_SEGURO

FROM cia\_seguro cia

WHERE to\_char(cia.FECHA\_FIN\_SEGURO,'mm-yyyy') = to\_char(sysdate,'mm-yyyy') ;

mensaje VARCHAR2(5000) :='Los seguros que se vencen este mes son:'||

CHR(10);

BEGIN

FOR r\_cia\_seguro IN c\_seguro LOOP

mensaje:= mensaje||CHR(10)|| 'id: '||r\_cia\_seguro.id||' Nombre: '||r\_cia\_seguro.Nombre;

END LOOP;

RETURN MENSAJE;

END;

SET SERVEROUTPUT ON

BEGIN

DBMS\_OUTPUT.PUT\_LINE(f\_buscarseguros);

END;

DECLARE

v\_nom\_cia\_seguro cia\_seguro.nombre%TYPE:=UPPER('&nombreCompañiaSeguro');

**3. Mostrar los clientes agencias que tengan una comisión de un rango dado.**

CREATE OR REPLACE FUNCTION f\_bus\_clien\_com

(v\_comisionmin NUMBER,

v\_comisionmax NUMBER)

RETURN VARCHAR2

IS

CURSOR c\_clienteagencia IS

SELECT ca.\*

FROM cliente\_agencia ca

WHERE comision BETWEEN v\_comisionmin AND v\_comisionmax;

mensaje VARCHAR2(5000) :='Los Clientes agencia comision entre '||v\_comisionmin||' hasta '||v\_comisionmax|| ' son: '||CHR(10);

BEGIN

FOR r\_cliente\_agencia IN c\_clienteagencia LOOP

mensaje:=mensaje||CHR(10)||'\*\*\* Codigo: '||r\_cliente\_agencia.codigo||' Nombre: '||r\_cliente\_agencia.nombre||' Dirección: '||r\_cliente\_agencia.direccion||' Poblacion: '||r\_cliente\_agencia.poblacion||' Nif: '||r\_cliente\_agencia.nif||' comision: '||r\_cliente\_agencia.comision||' Persona contacto: '||r\_cliente\_agencia.persona\_contacto||CHR(10);

END LOOP;

RETURN mensaje;

END;

/

—----------------

SET SERVEROUTPUT ON

DECLARE

v\_comisionmin NUMBER:=&ComisionMinima;

v\_comisionmax NUMBER:=&ComisionMaxima;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(f\_bus\_clien\_com(v\_comisionmin,v\_comisionmax));

END;

/

CONCLUSIONES

Al finalizar el proyecto nos hemos dado cuenta que la producción de software es bastante importante, mucho más de lo que se piensa, ya que esto ayuda a mucha gente a facilitarles las cosas ya sea para manejar por ellos mismos los emprendimientos que realicen o por otro lado, ayudar a las empresas a controlar la gran cantidad de información que manejan y que tienen que monitorear casi constantemente.

A la hora de realizar este proyecto aprendimos que es importante primero realizar una planificación de todo lo que vamos a hacer antes de empezar a desarrollarlo, debido a que esto permitió que nuestro grupo de trabajo tuviera un mayor orden, supiera en todo momento que debíamos hacer ayudándonos a optimizar el tiempo de trabajo y organizándonos para en todo momento poder tener un avance continuo y eficiente.

Este proyecto nos enseñó a que no hay que quedarnos con el aprendizaje básico, si no que hay que ir más allá y tomar nuestro tiempo para aprender nuevas habilidades y competencias, además conocer diferentes herramientas que nos ayuden a crecer en conocimiento pues para completar este proyecto fue necesario aprender muchas cosas fuera de lo establecido en las clases.

BIBLIOGRAFÍA

enago academy(s.f)¿Cómo elegir la mejor metodología de investigación para su estudio?[https://www.enago.com/es/academy/choose-best-research-methodology/#:~:text=Las%20metodologías%20de%20investigación%20pueden,que%20interesan%20a%20la%20investigación](https://www.enago.com/es/academy/choose-best-research-methodology/#:~:text=Las%20metodolog%C3%ADas%20de%20investigaci%C3%B3n%20pueden,que%20interesan%20a%20la%20investigaci%C3%B3n).

Charles darwin university Australia (s.f) Choosing your research methodology and methods

<https://www.cdu.edu.au/research-and-innovation/current-students/useful-materials/choosing-your-research-methodology#which-cluster-of-quantitative-techniques-should-i-use->

[ttps://www.ejemplos.co/100-ejemplos-de-conectores/](https://www.ejemplos.co/100-ejemplos-de-conectores/) [h](https://www.ejemplos.co/100-ejemplos-de-conectores/)

Oxford Languages(s.f)<https://languages.oup.com/google-dictionary-es/>

universidad de colima(s.f)Investigación-acción

<https://recursos.ucol.mx/tesis/investigacion_accion.php>

universidad externado de colombia (11 de septiembre 2016)La protección del software desde la Propiedad Intelectual en Colombia: Conveniencia de la creación de una normativa especial que garantice los derechos de los desarrolladores

<https://propintel.uexternado.edu.co/la-proteccion-del-software-desde-la-propiedad-intelectual-en-colombia-conveniencia-de-la-creacion-de-una-normativa-especial-que-garantice-los-derechos-de-los-desarrolladores/#:~:text=Como%20fue%20presentado%20en%20el,las%20cuales%20se%20refiere%20el>

alcaldia de bucaramanga (12 de enero 2022) Bucaramanga una ciudad de oportunidades <https://www.bucaramanga.gov.co/wp-content/uploads/2022/02/Plan-de-Desarrollo-2020-2023_VF-1.pdf>