

Escuela Politécnica superior
Alcalá de henares

Practica de POO Mayo 2014

**Grado En Ingeniería De Computadores
Curso 2013-2014**

Barquin Ayuso, Pedro

Descripción:

El programa desarrollado para esta entrega, trata de simular el funcionamiento de un concesionario de vehículos de ocasión, la base del programa es la realización de ciertas tareas dependiendo del identificador usado al entrar a la aplicación, como son la búsqueda de vehículos para su posterior posible compra o en el caso de entrar como administrador del sistema, la opción de dar de alta un vehículo, darlo de baja, comprobar las ofertas o las ventas, también cuenta con una opción de registro para los nuevos usuarios.

Se busca que la interface sea lo mas sencilla e intuitiva posible para facilitar el acceso al usuario o administrador.

Manteniendo siempre toda la información guardada entre ejecución y ejecución del programa.

Y mostrando algunos avisos sobre errores que se cometen al rellenar o completar los campos a la hora de realizar alguna de las acciones disponibles del programa.

En todas las ventanas se muestra el anagrama de la empresa de vehículos de ocasión.



Objetivos:

Los objetivos de la entrega de la practica son el afianzar los conceptos teórico prácticos obtenidos durante el curso y así demostrar que se han comprendido tales conceptos mediante la entrega de esta practica.

Especificación de datos registrados:

Datos insertados con anterioridad en el ejercicio para su correcta comprobación a la hora de ejecutar el programa.

Usuarios	Nombre	DNI	Relación	UserName	PW
1	Pedro Barquin	03142206S	Ninguna	pedro@gmail.com	1234
2	Juan Alto	12341234D	Familiar	juan@gmail.com	4321
3	Pepe Perez	32345670Z	Empleado	pepe@gmail.com	1212
4	Ana Martin	12345678W	Empleado	ana@gmail.com	9870
5	Laura Tomillo	45362718L	Ninguna	laura@gmail.com	5674
6	Pedro Barquin	03142200D	Familiar	pedro@hotmail.com	1234

Motocicleta:

Matricula	Marca	Modelo	Potencia	Color	Año fab	Kilómetros	Precio	tipo moto	cilindrada
1234ADC	honda	cb	56	verde	1989	111900	2500	Sport	600
4321EDA	honda	cbr	69	negra	2011	2900	6000	Custom	1000
8765DFC	suzuki	samurai	35	roja	2000	4500	3200	Trail	450

Turismo:

Matricula	Marca	Modelo	Potencia	Color	Año fab	Kilómetros	Precio	Combustible	Cambio	Nº plazas	Nº puertas
1234HDC	seat	ibiza	110	verde	2005	120000	5600	Diesel	Manual	5	3
2345FRZ	seat	ibiza	90	negro	2010	15000	9000	Diesel	Automatico	5	5
1256DFR	seat	toledo	140	blanco	2012	5000	11000	gasolina	Manual	4	3
2390KLJ	opel	vectra	220	negro	2006	125074	7500	gasolina	automatico	5	5

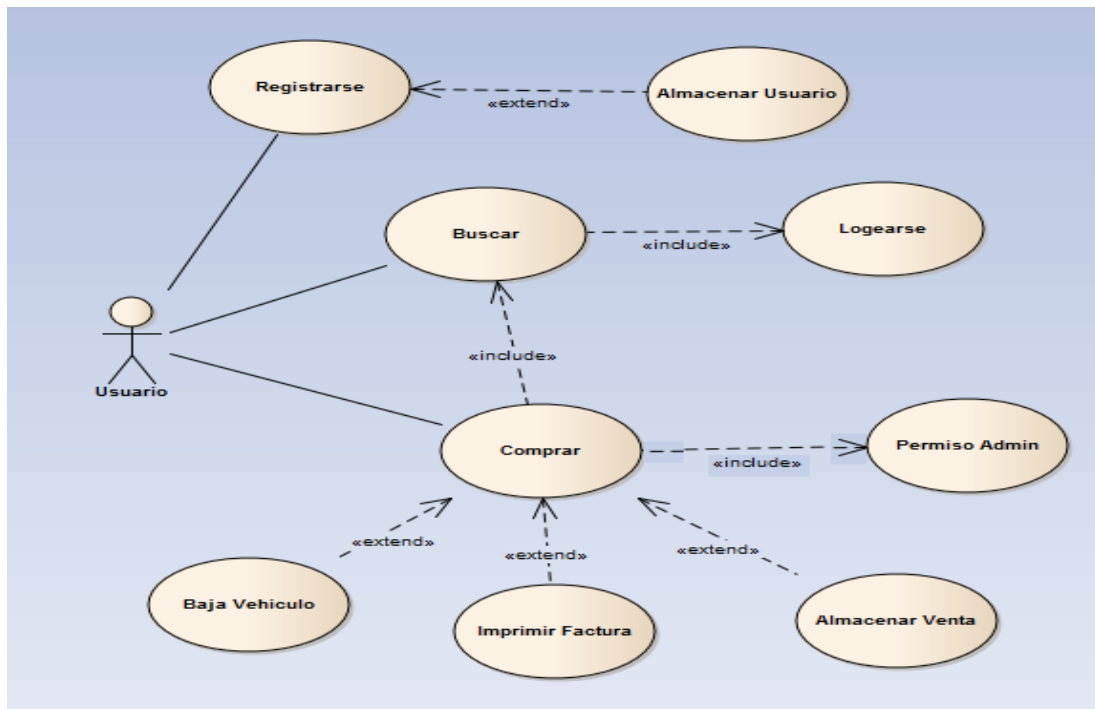
Todoterreno:

Matricula	Marca	Modelo	Potencia	Color	Año fab	Kilómetros	Precio	Combustible	Cambio	Nº plazas	Tracción
7812RTY	nissan	terrano	120	verde	2010	55000	9300	gasolina	automatico	7	4x4
8989PPD	lada	niva	90	blanco	1890	12000	1500	gasolina	Manual	5	4x4
3453FFF	nissan	terrano	150	negro	2000	19000	5000	gasolina	Manual	5	4x2

Especificación de requisitos:

Casos de uso de Usuario:

- En este diagrama se especifican los casos de uso de los cuales dispone un usuario normal con respecto al programa. Dichos casos de uso se comentaran uno por uno a continuación.



1. Registrarse:

En este caso de uso el usuario insertara los datos en los campos de la ventana registrarse y el programa obtendrá tratara y verificara esos datos antes de almacenarlos en el lugar que le corresponde.

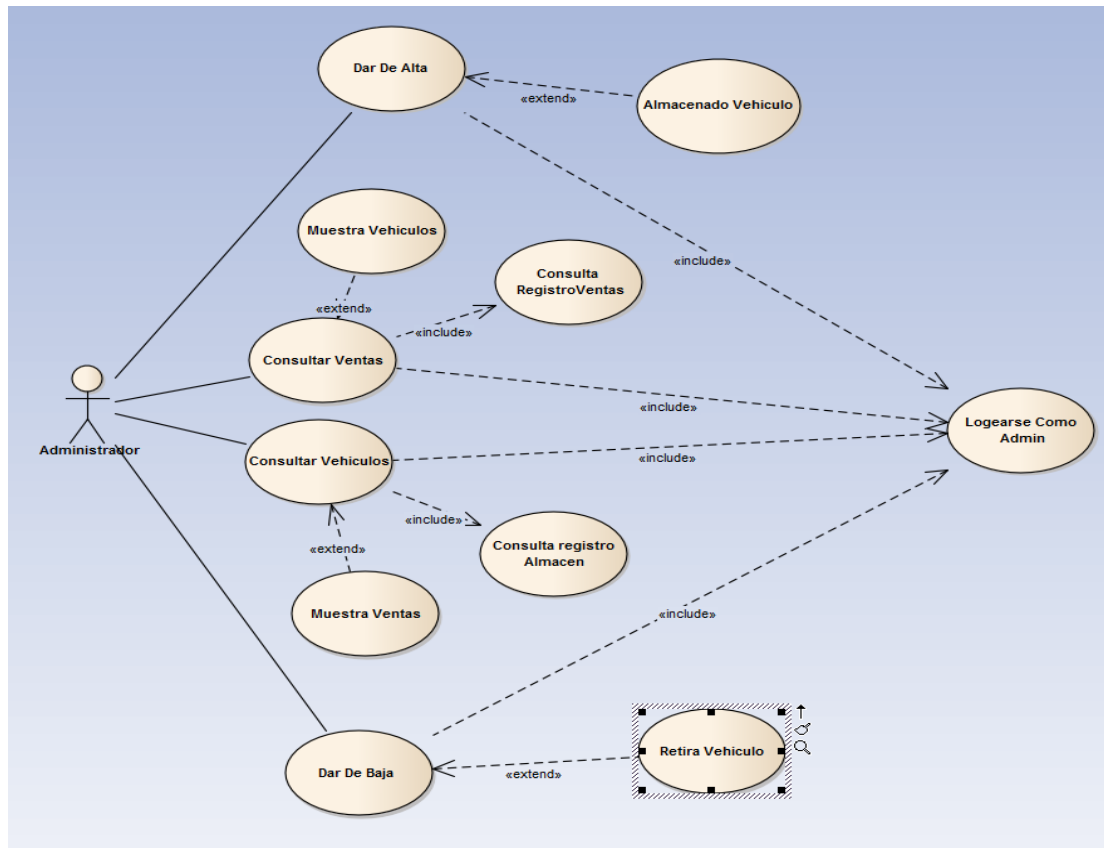
2. Buscar:

En este caso el usuario primero tendrá que logearse y una dentro se rellenaran los campos para poder realizar la búsqueda de acuerdo a los parámetros introducidos, esta parte esta relacionada con la compra.

3. Comprar:

En este caso el usuario a tenido que completar antes todo lo descrito en buscar y una vez que se ha cumplido este paso ha de verificara los permisos del administrador del sistema, en caso de tener dichos permisos entonces podre comprar lo cual desencadena los eventos de almacenamiento de la venta, impresión de la factura y retirada del vehículo del almacén.

- En este diagrama se especifican los casos de uso de los cuales dispone un usuario administrador al acceder con el username “admin” y la clave “1111” respecto al programa. Dichos casos de uso se comentaran uno por uno.



1. Dar de Alta:

En este caso el administrador se registrara con sus credenciales insertara los datos en los campos de la ventana dar de alta y el programa obtendrá tratara y verificara esos datos antes de almacenarlos en el lugar que le corresponde en este caso el almacén de vehículos.

2. Consultar ventas:

En este caso el administrador se registrara con sus credenciales de acceso, una vez realizado este paso se consultara la base de datos de las ventas para obtener el conjunto, cuando se tengan las ventas se mostraran en su plantilla correspondiente.

3. Consultar vehículos:

En este caso el administrador se registrara con sus credenciales de acceso, una vez realizado este paso se consultara la base de datos de los vehículos, se obtendrán los mismos para poder mostrarlos en la plantilla correspondiente.

4. Dar de Baja:

En este caso el administrador se registrara con sus credenciales, una vez se a cumplido este apartado el programa obtendrá la información de la baja en cuestión y procederá a eliminar el vehículo del almacén de vehículos.

Análisis de la aplicación:

La aplicación cuenta con una sección gráfica o swing la cual es por donde se moverá el usuario, en este apartado del programa sera el encargado de recoger toda la información que el usuario ira insertando en los cuadros indicados para este uso.

La información que es recogida en estos campos sera enviada a las clases del núcleo del programa mediante los tratamientos o asignaciones que se realizaran en la propia ventana dejando así ya limpia la información que sera enviada para su almacenamiento o para la creación de objetos que posteriormente serán utilizados en otras ventanas por el usuario o el administrador.

Dichas clases se agrupan en dos clases las cuales son citadas en el siguiente párrafo.

- Clases Interface

En estas se encuentran las ventanas y dentro de esta los campos de texto donde se insertaran los datos y los botones que son los encargados de operar con esos datos para luego mandarlos a los otros tipos de clase o obtener de alguna clase información y tratarla mostrando así dicha información en la ventana.

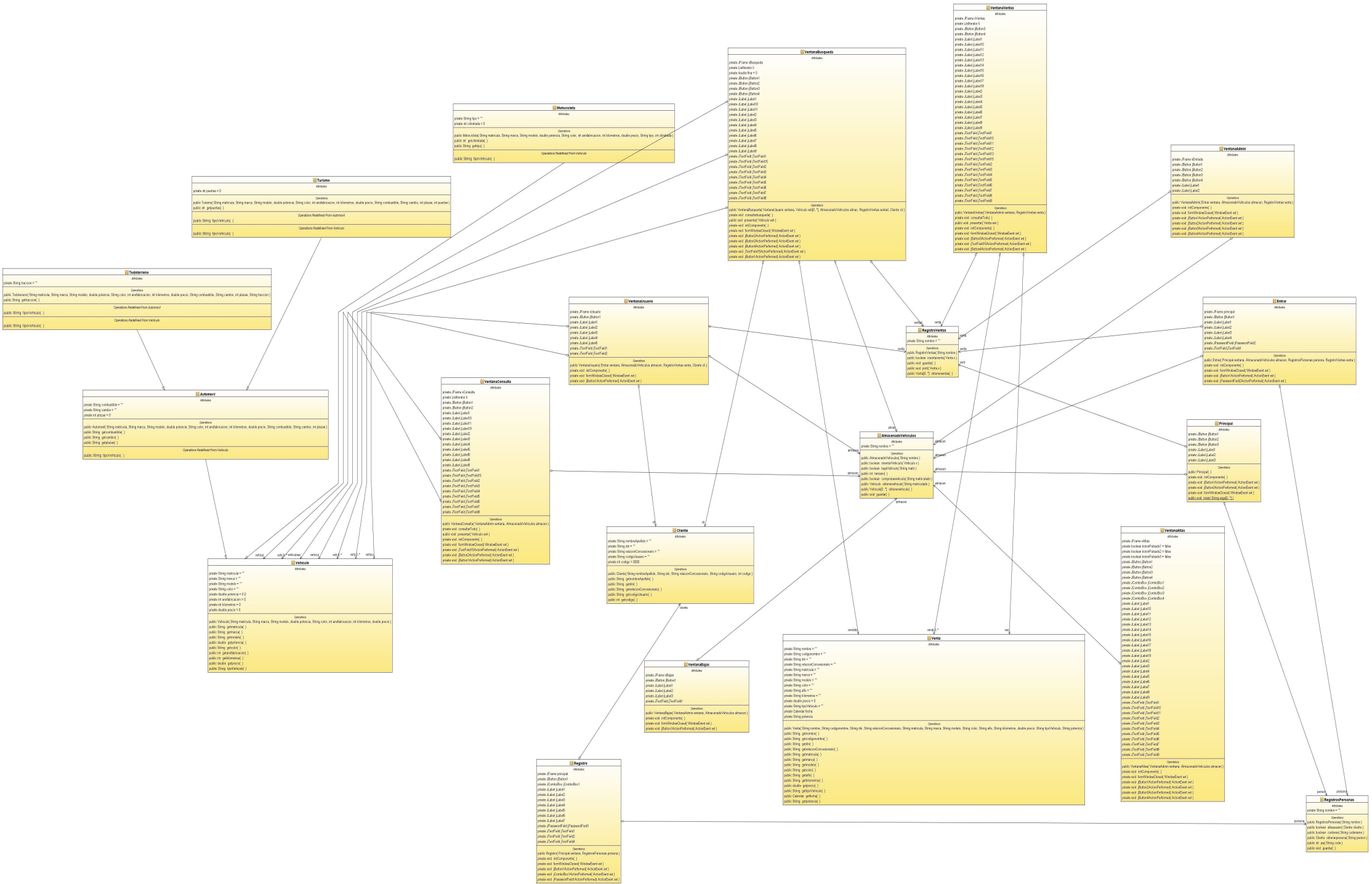
Estas clases son: Principal, Entrar, Registro, VentanaAdmin, VentanaAltas, VentanaBajas, VentanaVentas, VentanaConsultas, VentanaUsuario y VentanaBusqueda.

- Clases Normales

En estas clases se encuentran las clases básicas de java las cuales recibirán la información desde las clases ventanas y con ella crearan buscaran u operaran según el tipo de llamada que se realice.

Estas clases son: Vehículo, Automóvil, Turismo, Motocicleta; TodoTerreno, Cliente, Venta, AlmacenadoVehiculos, RegistroVentas, RegistroPersonas.

Se añade diagrama de clases del programa en cuestión donde se ven el tipo de relación que hay entre las clases con sus métodos y los atributos



Diseño e implementación:

Principal:

Núcleo del programa donde se instancian todos los almacenes o registros y el cual es la base de la interface gráfica del programa, de la cual se irán creando los tipos de ventanas del programa.

AlmacenadoVehiculos:

En esta clase es donde se realiza el almacenado de los vehículos que han sido dados de alta y la retirada de los que son dados de baja todo esto por el administrador del sistema, la clase cuenta con un método para serializar y almacenar la información y otro para al empezar traer esa información de vuelta.

Atributos:

Lugar donde se almacenan los vehículos registrados dentro de un hashmap con la key matricula y almacena objetos vehículos

Constructor:

En el es donde se crea el almacén que sera usado por el resto de clases, al crearlo al inicio en la clase principal, se carga la información almacenada en anteriores sesiones, para ello obtiene el archivo definido y lo trata obteniendo así un objetoinputstream el cual se ha de pasar a hashmap y entonces ya tendremos la información, después ha de cerrarse el flujo.

Guardar:

Este método almacena la información el hashmap al igual que el cargar obtiene el archivo especificado y lo trata creando un objetostream que es en el cual escribiremos la información del hashmap a guardar, una vez completado se cierra el flujo.

InsertarVehiculo:

Con este método recibimos un objeto vehículo, primero se comprueba que la matricula del vehículo en cuestión no esta ya introducida de ser así no se hace nada y se retorna un false, si por el contrario no esta se añade el vehículo con su key y el objeto al hashmap se llama a guardar y se retorna un true.

BajaVehiculo:

Al igual que en el alta en este caso se recibe la matricula y se comprueba que esta en el hashmap de no ser así no se hace nada y se retorna un false, si por el contrario esta se retira del hashmap con su key que es la matricula se llama a guardar y se retorna un true.

Tamaño:

Retorna un valor el cual es el tamaño actual del hashmap.

ComprobarVehiculo:

Recibiendo una matricula comprueba si esta en el hashmap de ser así retorna un true si no un false.

ObtenerVehiculo;

Este método vuelca el hashmap completo en un arraylist de vehículos y lo envía.

obtenerVehiculo(String matricula):

Este método recibe una matricula en concreto y con ella extrae el vehículo del hashmap para devolverlo como dato.

RegistroVentas:

En la clase registroventas se almacenan las ventas que se han realizado durante la ejecución del programa en un arraylist para ello usamos se almacena el objeto venta, cuenta con los métodos cargar y guardar al igual que almacenadoVehiculos.

Atributos:

Lugar donde se almacenan las ventas registradas dentro de un arraylist que almacena objetos venta.

Constructor:

En el es donde se crea el registro que sera usado por el resto de clases, al crearlo al inicio en la clase principal, se carga la información almacenada en anteriores sesiones, para ello obtiene el archivo definido y lo trata obteniendo así un objetoinputstream el cual se ha de pasar a arraylist y entonces ya tendremos la información, después ha de cerrarse el flujo.

InsertarVenta:

Se recibe un objeto del tipo venta se almacena la venta en el arralist con el objeto venta después se llama a guardar se llama a imprimir pasando la venta y se retorna true.

Guardar:

Este método registra la información el arraylist al igual que el cargar obtiene el archivo especificado y lo trata creando un objetostream que es en el cual escribiremos la información del arraylit a guardar, una vez completado se cierra el flujo.

Print:

En este método se recibe un objeto venta, con ese objeto creamos un archivo de salida con nombre la matricula+nombreapellido.txt en el cual escribiremos toda la información de la venta, al terminar cerramos el flujo.

ObtenerVehiculo:

Este método da el arraylist de venta y lo envía.

RegistroPersonas:

En la clase registropersonas se almacenan los clientes que se han registrado durante la ejecución del programa en un hashmap para ello usamos la key username del cliente y se almacena el objeto cliente, cuenta con los métodos cargar y guardar al igual que almacenadoVehiculos.

Atributos:

Lugar donde se almacenan los clientes registrados dentro de un hashmap con la key username y almacena objetos cliente.

Constructor:

En el es donde se crea el registro que sera usado por el resto de clases, al

crearlo al inicio en la clase principal, se carga la información almacenada en anteriores sesiones, para ello obtiene el archivo definido y lo trata obteniendo así un objetoinputstream el cual se ha de pasar a hashmap y entonces ya tendremos la información cargada, después ha de cerrarse el flujo.

Guardar:

Este método registra la información el hashmap al igual que el cargar obtiene el archivo especificado y lo trata creando un objetostream que es en el cual escribiremos la información del hashmap a guardar, una vez completado se cierra el flujo.

Contiene:

Recibiendo un username comprueba si esta en el hashmap de ser así retorna un true si no un false.

PW:

Este método recibe un username con el obtiene el cliente en concreto sacándolo del hahmap, con el cliente ya en la mano obtenemos su código secreto el cual enviamos como retorno.

ObtenerPersona:

Con este método recibimos el username y con el extraemos el cliente del hashmap ese cliente es retornado como dato.

AltaUsuario:

Se recibe un objeto del tipo cliente con el cual comprobamos que no se encuentra el username de ese objeto en el hashmap, de ser así se almacena el cliente en el hashmap con el username como key y el objeto cliente después se llama a guardar se llama y se retorna true, si por el contrario el username ya esta no hace nada y se retorna un valor false.

Cliente:

La clase cliente donde se crean los objetos de este tipo los cuales son los usuarios de nuestro programa, esta es la encargada de crear dichos objetos clientes con sus atributos, es serializable ya que los objetos que se almacenan son de este tipo .

Atributos:

La clase cliente cuenta con un atributo que es el nombre y el apellido del cliente el cual es de tipo string, el dni del cliente el cual también es un string, la relación con el concesionario que al igual que el resto es un string, el usernarme o código de usuario el cual es string y por ultimo el código o pw que es de tipo int.

Constructor:

Aquí se referencia todos los campos que hemos recibido al solicitar la creación del objeto de tipo cliente, asignándolo a su lugar.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Venta:

En esta clase se crean los objetos del tipo venta, el objeto venta junta toda la información de los objetos cliente y vehículo involucrados en lo que es la venta, es serializable ya que los objetos que se almacenan son de este tipo .

Atributos:

Esta clase cuenta con una amplia lista de atributos ya que junta la información de cliente y vehículo.

Los atributos del tipo String son: nombre, username o codigonombre, dni, relación concesionario, matricula, marca, modelo, color, año, kilómetros, tipovehiculo, potencia. Algunos de estos no están en su forma original ya que es mejor tenerlos en formato string ya que solo sirven para referencias o mostrarlos lo cual es mas cómodo.

Double: El precio final del vehículo no el de venta ya que según el tipo de relación con el concesionario se hace un descuento o no.

Calendar: Es la fecha de creación del objeto y por lo tanto de la venta.

Constructor:

Aquí se referencia todos los datos recibidos al solicitar la construcción del objeto con sus respectivos, y por ultimo se asigna a la fecha la del momento de creación del objeto, la cual sera la fecha de la venta.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Vehículo:

Clase padre de motocicleta y automóvil la cual cuenta con un método abstract y es serializable ya que los objetos que se almacenan son de este tipo.

Atributos:

Esta clase cuenta con los atributos que necesitan todos sus hijos y que por consiguiente comparten entre ellos, los hay de varios tipos los cuales son.

String: Matricula, marca, modelo y color.

Int: Anodefabricacion y kilómetros.

Double: Potencia y precio.

Constructor:

Aquí se referencia todos los datos recibidos al solicitar la construcción del objeto con sus respectivos.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Método Abstract:

Solo se declara ya que serán los hijos los que lo realicen.

Motocicleta:

En esta clase se crean los objetos del tipo motocicleta los cuales son hijos de la clase vehículo, esta clase cuenta con la información de la padre mas información suya extra.

Atributos:

Como atributos tiene el tipo de motocicleta el cual es un string y la cilindrada que es un entero.

Constructor:

Aquí se crea el objeto del tipo motocicleta con la información que recibe la cual es la que tiene el padre y se asigna añadiendo la palabra super y los atributos, y luego se referencia los atributos propios de la motocicleta.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Método Abstract:

Es el método que viene del padre el cual retorna un string con toda la información especial de la motocicleta.

Automóvil:

Clase hija de vehículo abstract ya que contiene un método de ese tipo y clase padre de turismo y todoterreno, recoge la información conjunta de las dos clases hijas.

Atributos:

Cuenta con los atributos de tipo string que son el tipo de cambio y el tipo de combustible, y el numero e plazas que es de tipo int.

Constructor:

Aquí se crea el objeto del tipo automóvil con la información que recibe la cual es la que tiene el padre y se asigna añadiendo la palabra super y los atributos, y luego se referencia los atributos propios de la automóvil.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Método Abstract:

Solo se declara ya que serán los hijos los que lo realicen.

Todoterreno:

En esta clase se crean los objetos del tipo todoterreno los cuales son hijos de la clase automóvil la cual es hija de vehículo, esta clase cuenta con la información de la padre mas información suya extra.

Atributos:

Como atributos tiene el tipo de tracción el cual es un string.

Constructor:

Aquí se crea el objeto del tipo todoterreno con la información que recibe la cual es la que tiene el padre y se asigna añadiendo la palabra super y los atributos, y luego se referencia los atributos propios del todoterreno.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Método Abstract:

Es el método que viene del padre el cual retorna un string con toda la información especial de la todoterreno y la información del automóvil.

Turismo:

En esta clase se crean los objetos del tipo turismo los cuales son hijos de la clase automóvil la cual es hija de vehículo, esta clase cuenta con la información de la padre mas información suya extra.

Atributos:

Como atributos tiene el numero de puertas el cual es un int.

Constructor:

Aquí se crea el objeto del tipo turismo con la información que recibe la cual es la que tiene el padre y se asignara añadiendo la palabra super y los atributos, y luego se referencia los atributos propios del turismo.

Métodos Get*:

Esta clase cuenta con los métodos get* los cuales retorna el valor de aquello que se esta pidiendo en el mismo formato del cual es el original.

Método Abstract:

Es el método que viene del padre el cual retorna un string con toda la información especial de la turismo y la información del automóvil.

Ventanas

Entrar:

Esta ventana viene desde la ventana principal, contiene las importaciones de los paquetes de las clases que referencia, su cometido es desviar el flujo en un sentido para el usuario registrado o para el administrador.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos registroPersonas y registroVentas ya que estos objetos son necesarios para operar con ellos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra.

Botón Entrar:

Este botón en primer lugar comprueba que no hay campos vacíos si es así muestra un mensaje, si todo esta de acuerdo verifica si los datos son de administrador si es correcto lanza la ventana de administrador, si por el contrario no coinciden intenta con los datos almacenados en el registroPersonas si están lanza la ventana de usuario si no muestra un mensaje de que no hay usuarios con esos datos.

Registro:

Esta ventana viene desde la ventana principal, contiene las importaciones de los paquetes de las clases que referencia, su cometido registrar a los usuarios en el registroPersonas.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo registroPersonas y Cliente ya que estos objetos son necesarios para operar con ellos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra.

Botón registrar:

Con este botón lo primero se verifica que los campos estén completos, si es así se comprueba que la contraseña tiene na mas y na menos que 4 de longitud y si pasa este requisito se crea un objeto del tipo cliente con esa información y se da de alta en el registro recibiendo la respuesta si es true es que el objeto se a registrado en caso contrario es que el objeto ya estaba en el registro.

VentanaAdmin:

Esta ventana viene desde la ventana entrar, contiene las importaciones de los paquetes de las clases que referencia, su cometido es desviar el flujo en un sentido o otro dependiendo del botón pulsado.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos registroPersonas y registroVentas ya que estos objetos son necesarios para operar con ellos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra.

Botones:

En función del botón pulsado se desvía la ejecución a una ventana u otra en la cual se pasaran como datos cierta información que se sera necesaria.

VentanaVentas y VentanaConsulta:

Estas ventanas tienen un mecanismo similar por eso se explican juntas, viene desde la ventana admin, contiene las importaciones de los paquetes de las clases que referencia, su es mostrar la información almacenada en los registros.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos registroPersonas y registroVentas ya que estos objetos son necesarios para operar con ellos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible

la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra y por ultimo se llama al método consultartodo.

Consultartodo:

Este método lo que hace es importa un arraylist con la lista completa a tratar después de eso crea un listIterator que con el cual avanzaremos en la muestra de la información, primero verifica que el arraylists no esta vacío si es así enseña un mensaje y deshabilita los botones, si por el contrario tiene información obtiene el objeto del arraylist y ve si no es null (significa que es el ultimo) si es no nulo llama a mostrar parando el objeto.

Mostrar:

Se encarga de mostrar en los campos de la ventana toda la información del objeto recibido.

Siguiente:

Este botón obtiene el siguiente objeto al igual que en el método consultarTodo y lo muestra.

Anterior:

Este botón obtiene el anterior objeto al igual que en el método consultarTodo y lo muestra.

VentanaBajas:

Esta ventana viene desde la ventana admin, contiene las importaciones de los paquetes de las clases que referencia, su cometido dar de baja un vehículo.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos ya que es el objeto necesarios para operar con la información.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra.

Baja:

Al pulsar el botón baja captura el compra de texto que es la matricula siempre y cuando no este vacío, con esa información va al almacén a dar de baja el vehículo con lo que recibe un respuesta true si se a dado de baja y otra false si no esta dicho vehículo y por lo tanto no se puede dar de baja.

VentanaAltas:

Esta ventana viene desde la ventana admin, contiene las importaciones de los paquetes de las clases que referencia, su cometido dar de alta un vehículo.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos ya que es el objeto necesarios para operar con la información y 3 boolean los cuales sirven para saber de que tipo de vehículo hablamos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la

imagen y se muestra.

Botones Motocicleta Turismo Todoterreno:

Estos botones ponen a true el boolean de su tipo y habilitan o deshabilitan los campos que son o no son necesarios para el.

Alta:

Primero comprueba que ha sido pulsado uno de los botones y se va a la zona de dicho boolean pulsado, si es así comprueba que no este ningún campo sin rellenar, si se pasa este punto se crea el objeto en cuestión con la información de los campos de texto de la ventana y se registra en el almacén recibiendo una respuesta de la acción de alta, si el valor devuelto es true entonces se a registrado y se limpian los campos y se muestra un mensaje si no se enseña un mensaje de error diciendo que ya esta el vehículo.

VentanaUsuario

Esta ventana viene de insertar los datos de un usuario registrado en la ventana entrar, su función es la de buscar vehículos en concreto.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos y registroVentas ya que es el objeto necesarios para operar con la información, ademas cuenta con un ArrayList de vehículos un objeto cliente y ventas.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra.

Botón:

Primero comprueba que no hay ningún campo sin rellenar si es así obtiene la información de los campos y obtiene el ArrayList con los vehículos almacenados entonces uno a uno va comprobando si cumplen las condiciones de ser así los almacena en un ArrayList aux el cual sera el que envía a la ventana para mostrar los que coinciden con la búsqueda aparte de pasar el cliente que busca también.

VentanaBusqueda:

Al igual que la ventana consulta o ventas muestra la información de los vehículos por pantalla.

Atributos:

Tiene objetos del tipo JFrame para las ventanas y del tipo almacenadoVehiculos y registroVentas ya que estos objetos son necesarios para operar con ellos, también cuenta con un objeto venta y cliente y un ArrayList de vehículos.

Constructor:

Aquí se referencia los datos recibidos con sus respectivos y se hace invisible la ventana de la que venimos haciendo visible esta, también se obtiene la imagen y se muestra y por ultimo se llama al método consultartodo.

Consultartodo:

Este método lo que hace es importa ArrayList el cual viene dado desde la

ventana de usuario con la lista completa de los vehículos que cumplían las condiciones, después de eso crea un listIterator que con el cual avanzaremos en la muestra de la información, primero verifica que el arraylist no esta vacío si es así enseña un mensaje y deshabilita los botones, si por el contrario tiene información obtiene el objeto del arraylist y ve si no es null (significa que es el ultimo) si es no nulo llama a mostrar parando el objeto.

Mostrar:

Se encarga de mostrar en los campos de la ventana toda la información del objeto recibido.

Siguiente:

Este botón obtiene el siguiente objeto al igual que en el método consultarTodo y lo muestra.

Anterior:

Este botón obtiene el anterior objeto al igual que en el método consultarTodo y lo muestra.

Comprar:

Una vez en el vehículo que queremos comprar pulsamos el botón y aparecerá un mensaje pidiendo datos de administrador si se insertan estos datos se continua si no se muestra un mensaje diciendo que no tenemos permiso, si por el contrario tenemos permiso, primero se calcula el precio en función del tipo de relación del cliente una vez echo esto se crea el objeto venta con toda la información necesario ya sea obtenida de la ventana o por otros métodos, completado este paso se inserta la venta en el registroventa para que quede almacenada y para terminar se retira el vehículo en cuestión del almacenVehiculos, se muestra un mensaje de venta realizada y se vuelve a la ventana de usuario.

Volver:

Cierra la ventana y vuelve a la ventana de usuario.

Manual de usuario:

En este apartado se va a detallar el uso del programa contando todas las opciones de las cuales dispone y de como se ha de usar, además de la explicación de algunos conceptos y de los posibles mensajes emergentes que tiene el programa.

Nota: Si no hay botón de volver solo hay que cerrar la ventana para volver a donde estábamos anteriormente.

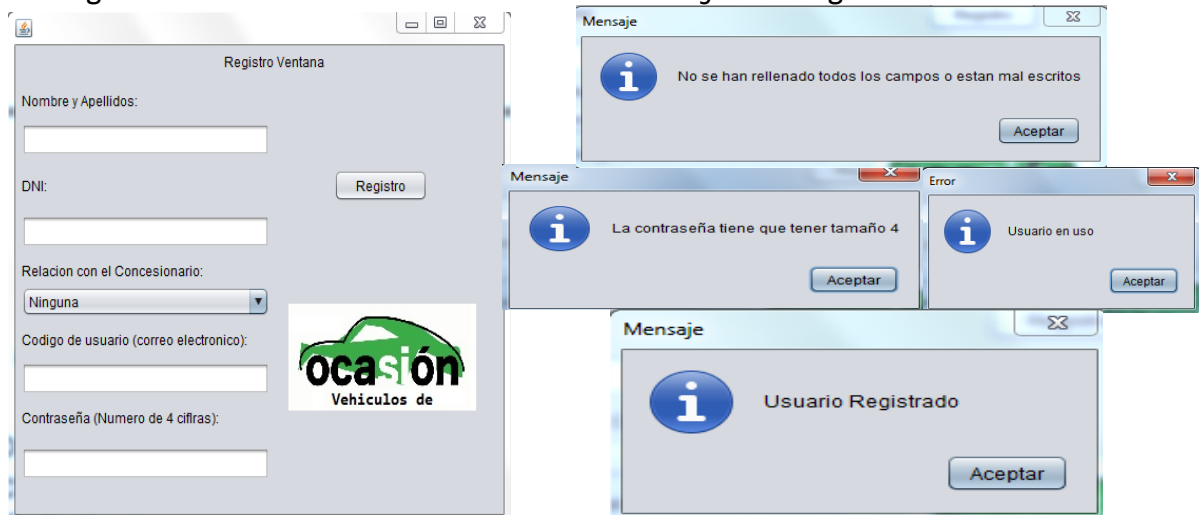
- **Principal:**

Nada más arrancar el programa nos encontramos con dos opciones posibles a elegir que son entrar para empezar con el programa o la opción de registrarnos en caso de no contar con una cuenta para el programa la cual es necesaria.



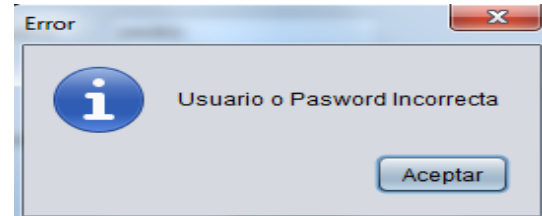
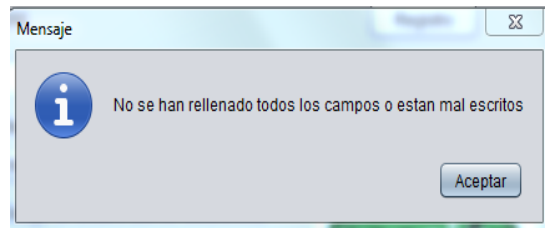
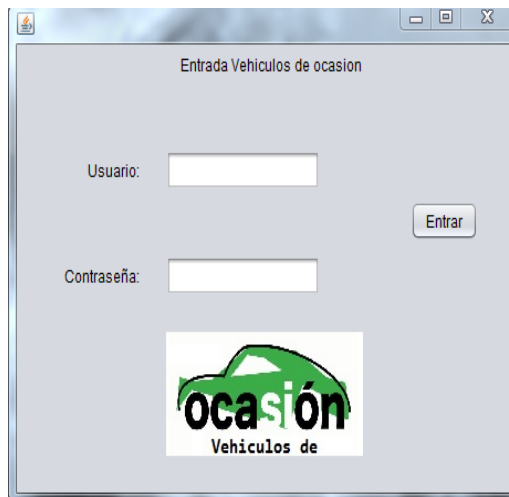
- **Registrarse:**

En esta sección se nos pedirán nuestros datos para la creación de un usuario para el programa, los campos necesarios son el nombre y apellido, el DNI completo "01231450S", el tipo de relación que tenemos con el concesionario pudiendo elegir entre varias opciones, el correo electrónico el cual será el username para el programa y su contraseña la cual tiene que tener 4 caracteres, en caso de que falte algún campo por rellenar, que el username este en uso o que la contraseña no tenga 4 dígitos saltará un mensaje de aviso, si se cumplen todos los requisitos se registrará el usuario mostrando un mensaje de registrado.



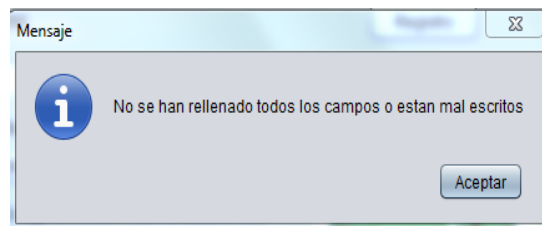
- **Entrar:**

Una vez pulsado entrar en la ventana principal, entramos a la ventana en la cual nos logeamos para entrar al grueso del programa, aquí nos encontramos con dos opciones, la primera que es logearnos con la clave del administrador (admin 1111) y acceder al apartado del administrador, el cual tiene unas opciones especiales o logearnos con un usuario que hayamos creado antes o que ya estuviera creado, para así acceder al apartado de usuario. También cuenta con un aviso en caso de no haber insertado ningún dato o en caso de no ser la información adecuada para acceder.



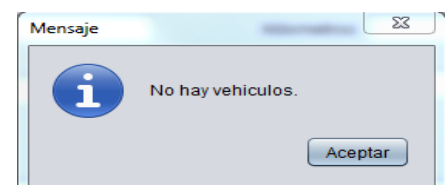
- **Entrada como Usuario:**

Si se han insertado en la ventana de entrada los datos adecuados, de un usuario en concreto, se entra a este apartado como el usuario logeado en cuestión, una vez en este apartado del programa tenemos la opción de buscar entre los vehículos que hay almacenado, para ello sera necesario insertar en los campos de búsqueda la información necesaria, en caso de faltar alguno de los campos por completar saldrá un mensaje de aviso, si todo esta correcto se pasara a la siguiente ventana en la cual se mostrara el resultado de la búsqueda en otra ventana.

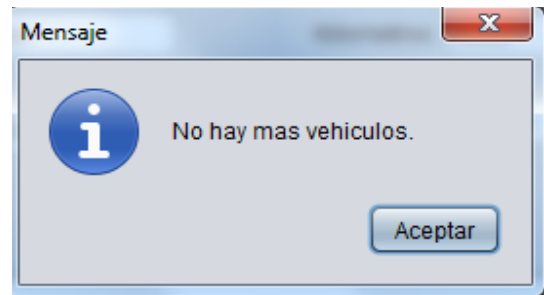


- **Ventana Búsqueda Usuario:**

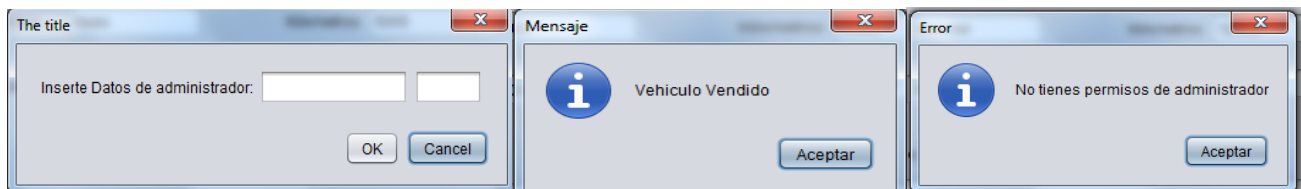
Esta ventana se abrirá de dos formas.
La primera en caso de no haber encontrado ningún vehículo con las características de la búsqueda, en este caso se mostrara un mensaje de que no hay vehículos disponible y los botones estarán deshabilitados salvo el de volver a la ventana anterior.



La segunda habiendo encontrado algunos vehículos con esas características, en este caso se habilitan todo los botones y se muestra la información referente al vehículo en cuestión en los campos de texto disponibles, en caso de haber mas de un vehículo podremos pasar entre todos los de la lista pulsando el botón de siguiente o anterior, cuando lleguemos al ultimo aparecerá un mensaje avisando de que no hay mas disponibles.



En cualquier momento podremos comprar un vehículo pulsando el botón de comprar al pulsarlo intentaremos comprar el vehículo mostrado en ese momento, para ello antes el administrador ha de insertar su contraseña en un mensaje emergente que aparecerá, en caso de que no se inserten los datos adecuados se mostrara un mensaje de error diciendo que no tenemos permisos de administrador, si por el contrario se introducen los datos correctos saldrá un mensaje avisando de la compra del vehículo lo que generara una factura y la amacena en un documento compuesto por la matricula y el nombre del usuario y retirara el vehículo del almacén de ocasión.



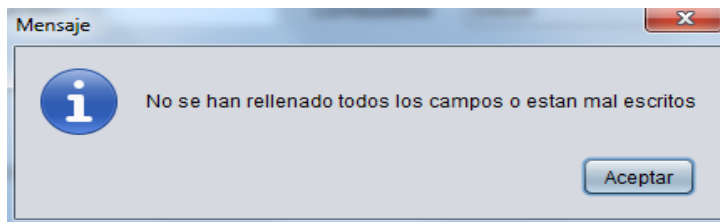
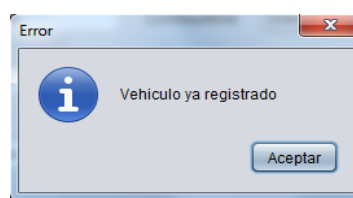
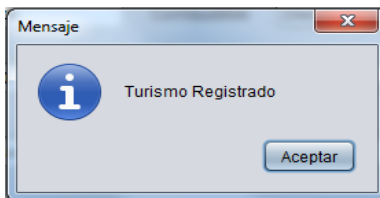
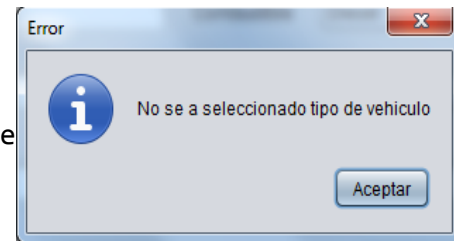
- Entrada como Administrador:**
 Si al insertar los datos en la ventana de entrar hemos puesto admin y 1111 entraremos al sistema como administradores por lo que podremos realizar ciertas acciones reservadas, dichas acciones son dar de alta un vehículo, dar de baja un vehículo, consultar las ofertas o consultar las ventas, para acceder a cada una de estas opciones solo es necesario pulsar el botón con su nombre.



- **Alta vehículos:**

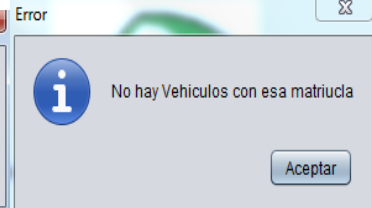
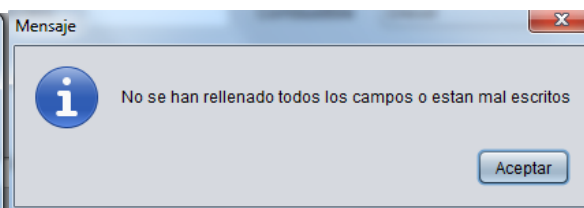
En la ventana de alta vehículos aparecen los 3 tipos de vehículos en la parte superior y hemos de seleccionar uno de ellos antes de proceder a insertar los datos, ya que en función del tipo de vehículo habrá campos que no estén disponibles y si no hemos pulsado ninguno y damos a registrar aparecerá un mensaje diciendo que no se a seleccionado ningún tipo de vehículo.

Una vez seleccionado el vehículo insertamos todos los datos necesarios para su registro ya que si no olvidamos de alguno de los campos aparecerá un mensaje avisando que faltan campos por rellenar como pasaba al registrar un usuario, si todos los campos están correctos y el vehículo no estaba registrado antes se mostrara en mensaje avisando de su registro cuando pulsemos el botón registrar, si por el contrario el vehículo ya estaba registrado saldrá un aviso diciendo que el vehículo ya esta registrado.



- **Baja vehículos:**

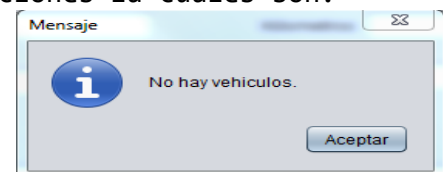
En este apartado podremos dar de baja un vehículo que este registrado en la base de datos del programa, para ello solo sera necesario insertar la matriculo del vehículo en cuestión si la matricula es la correcta y el vehículo esta saldrá un mensaje avisando se su eliminación, si por el contrario no se a rellenado el campo de texto o no esta dicha matricula saldrán mensaje avisando del error.



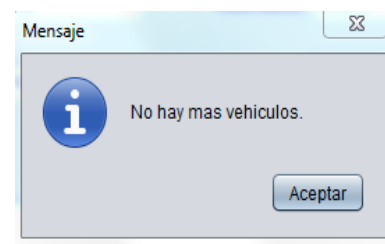
- **Consulta Vehículos:**

En esta ventana nada mas acceder a ella pulsando el botón desde la ventana de administrador se presentan dos opciones la cuales son.

Primera que no haya ningún vehículo almacenado por lo que la ventana aparecerá con algunos de los botones deshabilitados y mostrara un mensaje diciendo que no hay ningún vehículo.



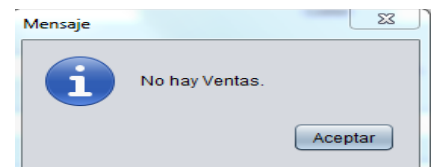
Segunda en el caso de que existan vehículos almacenados se mostraran todos lo que hay enseñando sus características en los campos que componen la ventana, con los botones siguiente y anterior podremos cambiar entre todos los vehículos disponible, en el caso de aparecer una venta avisando de que no hay mas vehículos habremos llegado al final de la lista.

 A form titled 'Consulta Vehículos' with navigation buttons 'Anterior' and 'Siguiente'. It contains input fields for: Matricula (1212ACB), Color (rosa), Marca (seat), Año fabricacion (2005), Modelo (ibiza), Kilometros (56000), Potencia (110.0), and Precio (8900.0). Below these is a text area for 'Vehiculo' containing 'turismo: Puertas:5 Combustible:Diesel Plazas:5 Cambio:Automatico'. At the bottom is the 'ocasión Vehículos de' logo.


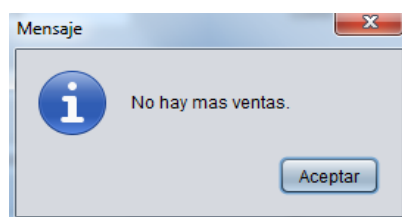
- **Consulta ventas:**

En esta parte del programa podremos ver todas las ventas almacenadas en el programa, al acceder a esta sección desde la ventana del administrado se nos presentan dos opciones.

Primera que no haya ningún venta almacenada por lo que la ventana aparecerá con algunos de los botones deshabilitados y mostrara un mensaje diciendo que no hay ninguna venta.



Segunda en el caso de que existan ventas almacenadas se mostraran todas las que hay enseñando sus características en los campos que componen la ventana, con los botones siguiente y anterior podremos cambiar entre todas las ventas disponible, en el caso de aparecer una venta avisando de que no hay mas vehículos habremos llegado al final de la lista.



 A form titled 'Historial Ventas' with navigation buttons 'Anterior' and 'Siguiente'. It contains sections for 'Datos Comprador' (Nombre: Pedro barquin, DNI: 3142206S, Usuario: pedro.barquin@gmail.com, Relacion: Empleado) and 'Datos Vehiculo' (Matricula: 1234, Color: azul, Marca: honda, Año fabricacion: 2005, Modelo: isuzu, Kilometros: 12050, Potencia: 45.0, Precio: 2625.0). Below these is a text area for 'Vehiculo' containing 'motocicleta: Cilindrada:1000 Tipo:Sport' and a 'Fecha' field with 'Sun May 25 01:05:21 CEST 2014'. At the bottom is the 'ocasión Vehículos de' logo.

Conclusiones:

La conclusión de este ejercicio/programa es cerciorarnos de que todos los contenidos aprendidos durante el curso han sido comprendidos y ser capaz de juntarlos para realizar una practica de cierta envergadura la cual cuenta con conversión entre tipos de datos, persistencia de información, creación de documentos de texto del ejercicio, etc.

Para ello se realiza la entrega de esta practica la cual consta de todos esos conceptos teórico prácticos.