**Multi302 Mayo 29 Base de Datos III CRUD**

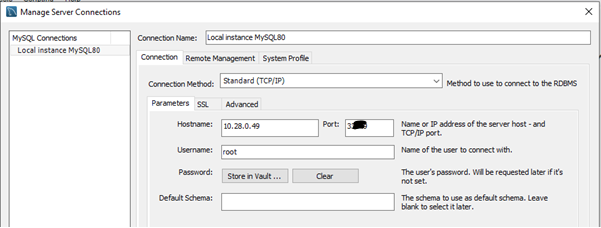
**Temas Tratados**

1. CRUD en proyVehiculos

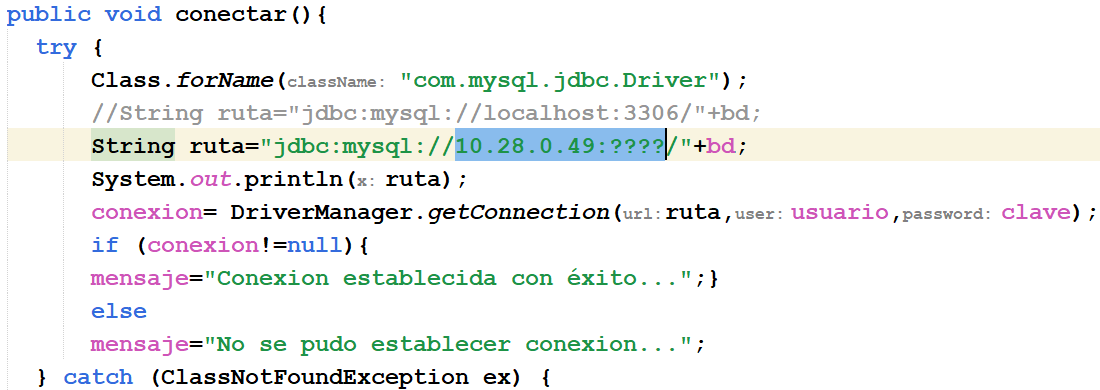
**Desarrollo**

1. **CRUD en proyVehiculos**

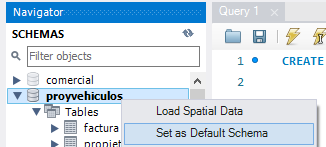
Verificar datos de conexión, servidor y puerto para configurar en clase **ConexionBD**



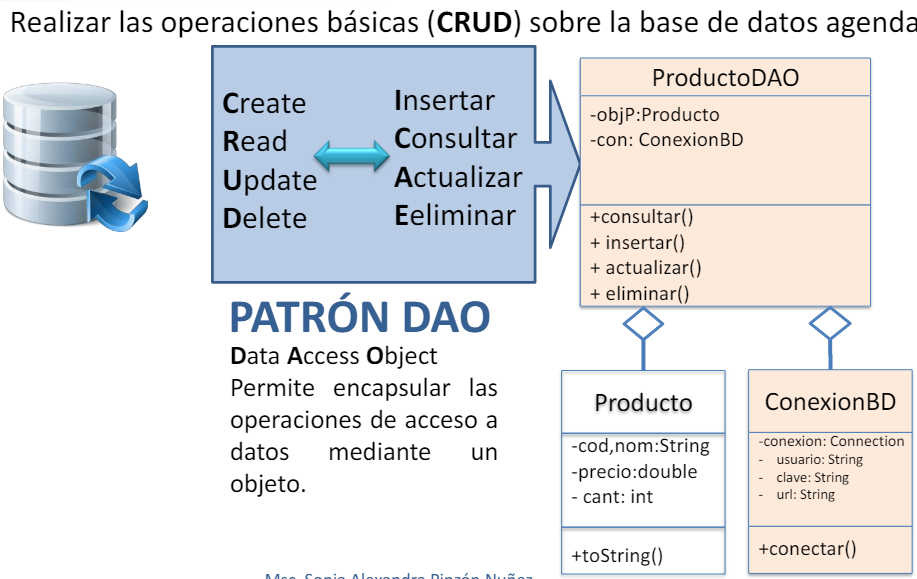
En la clase ConexionBD se debe actualizar cada dato



Establecer esquema BD por defecto



* **Realizar CRUD aplicando patrón DAO**

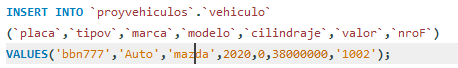


En clase Factura DAO configurar métodos Insertar Actualizar y Eliminar

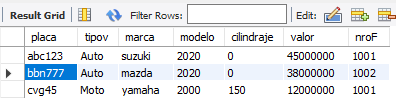
**Configurar comando insertar**

En Java se pueden usar dos tipos se estamentos para realizar la inserción de registros, el primero consiste en configurar la cadena del comando incluyendo los valores que se desean insertar. Para ello inicialmente se identificará el comando se usa en MySQL workbench.

Se debe identificar los datos y nombre de la tabla para hacer el comando de inserción



Verificar con select



Así se puede identificar en el siguiente código para concatenar los datos considerando el tipo de datos de cada campo de la tabla.

Importante:

Si el tipo de dato es VARCHAR, se debe enviar el dato dentro de comillas simples, tal como se observa en el método para insertar los vehículos desde el arreglo de la Factura.



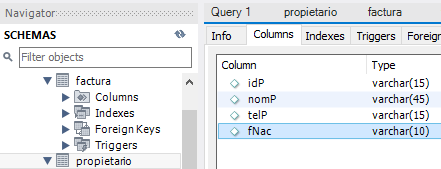
Este método de unir los valores con el comando insert, aunque es complicado realiza la acción tal como lo requiere MYSQL.

En el segundo caso, se crea un PrepareStatemt con el comando insert a partir de un string que incluye los parámetros mediante el símbolo ?, de esta forma mediante los métodos set, se incorporan los valores dentro de la cadena que ejecutará la consulta,

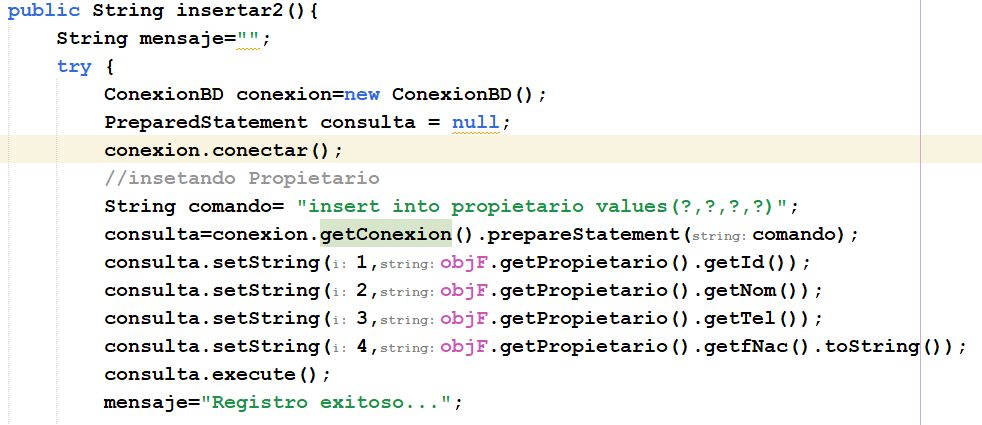


* **Proceso para para insertar propietario**

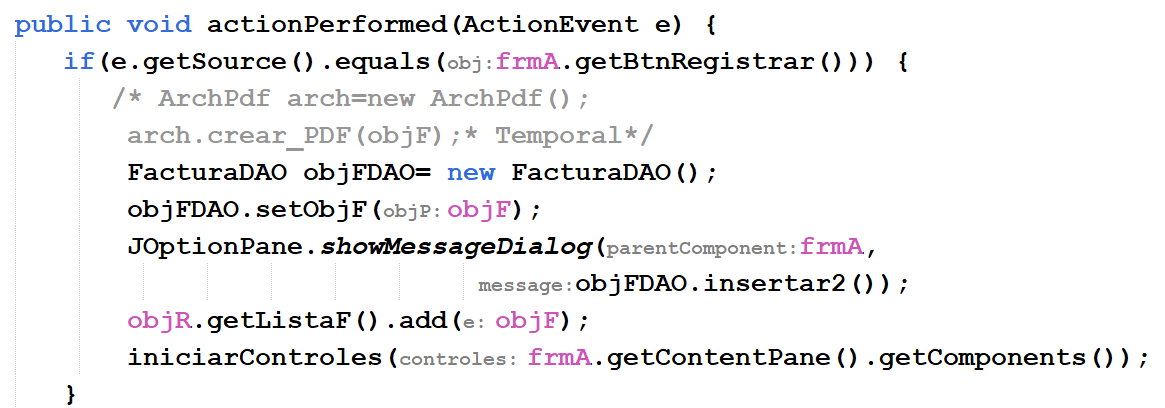
verificar nombre y tipos de datos de la tabla



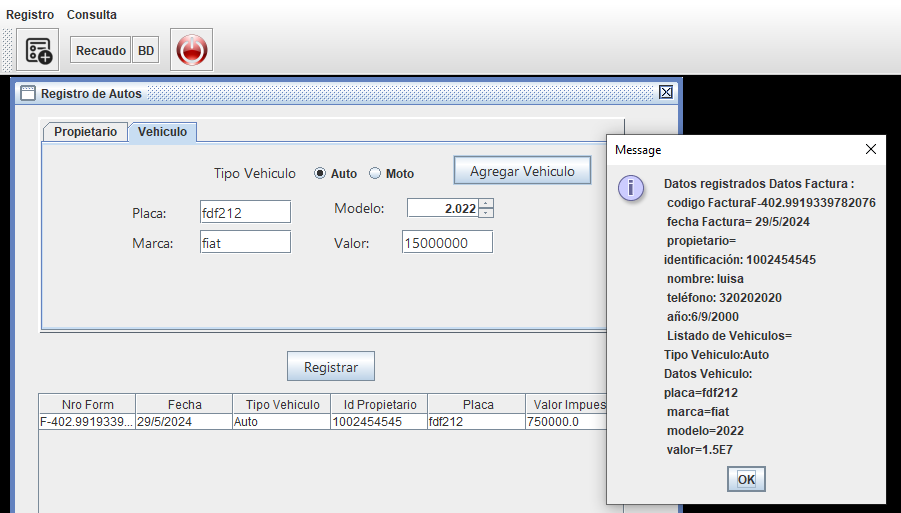
En clase **FacturaDAO** configurar el método insertar2

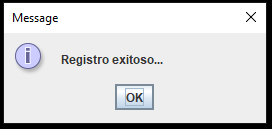


En **ControladorJIFRM** del formulario **jIfrmVehiculo** agregar el código de inserción en el botón registrar

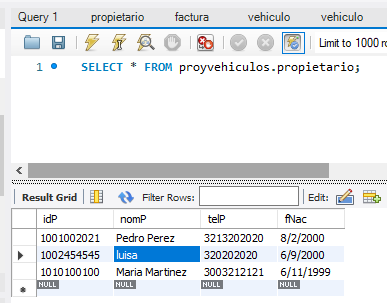


Ejecutar el registro de datos



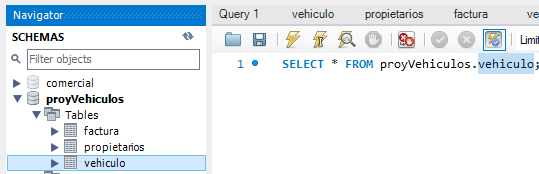


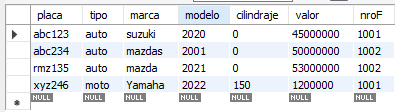
Verificar la base de datos



* **Procedimiento para Actualizar registros:**

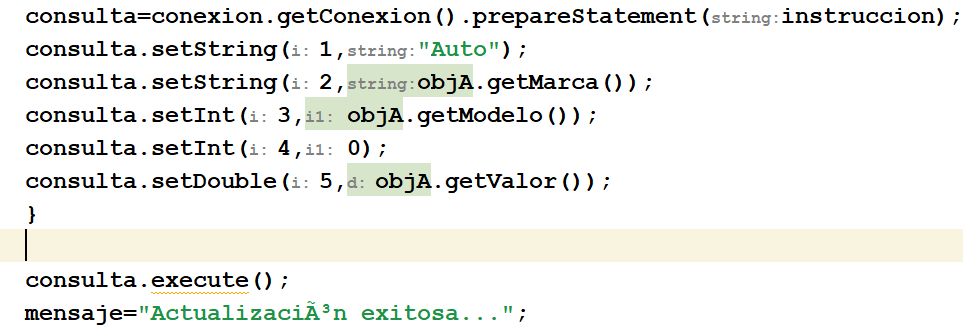
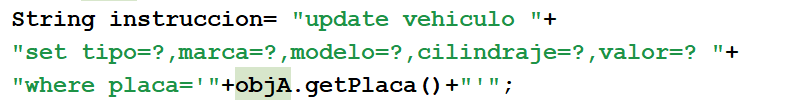
Para actualizar se verifica nombre y campos de tabla





Insertando imagen...

El comando de actualización no permitirá modificar los campos de placa y nroF, dado que son llaves primaria y foránea

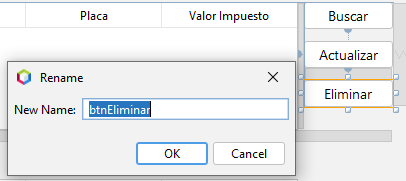
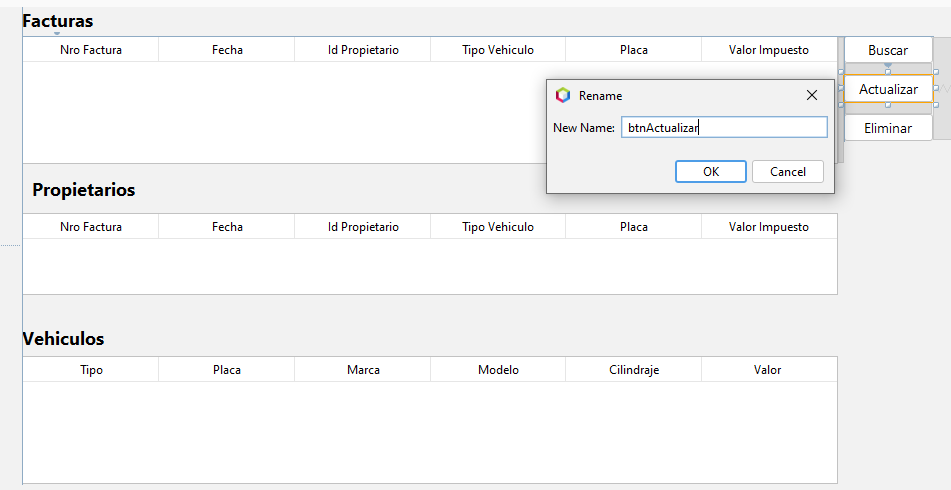


**En la clase FacturaDAO,** se implementa el método actualizar que recibirá un objeto de tipo Object, el cual corresponderá a los tipos de objeto que se están trabajando en la BD (Auto, Moto, Persona o Factura), con los cuales se configurará la instrucción **update** que será ejecutada, finalmente el método retornará un mensaje indicando si la actualización ha sido exitosa o se ha generado error.

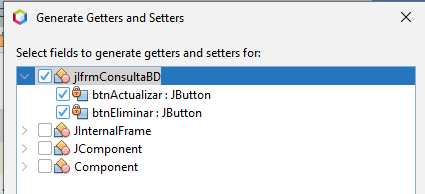


**En el formulario JIFRmConsultaBD**

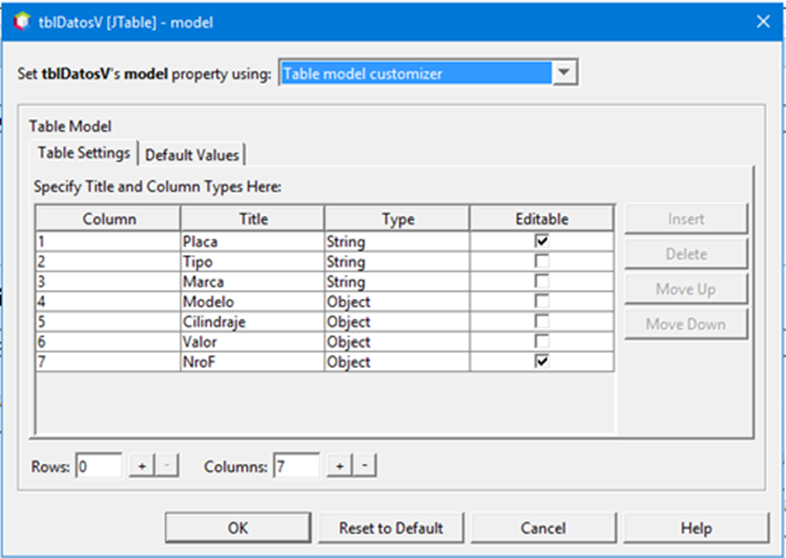
En formulario ConsutaBD agregar los botones Actualizar y Eliminar.



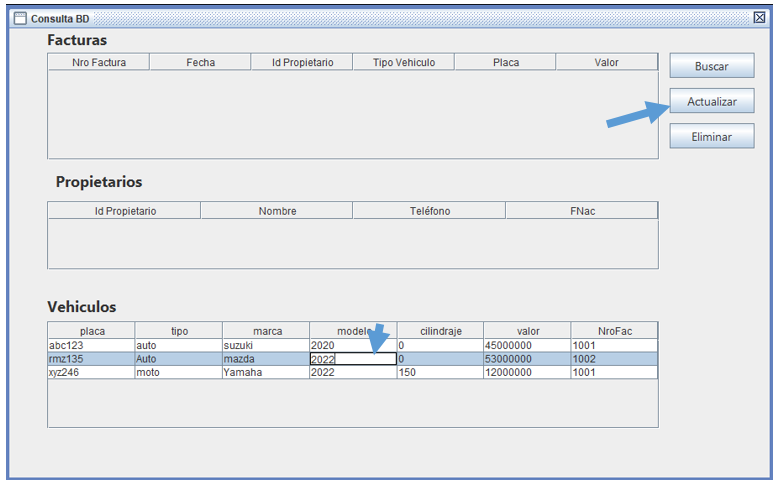
Generar Getter and setters



Este procedimiento se realizará a partir de la tabla que el usuario seleccione, por lo tanto, deberá permitirse la actualización de los campos de las tablas (excepto llave principal y foránea), verifique el modelo de cada tabla



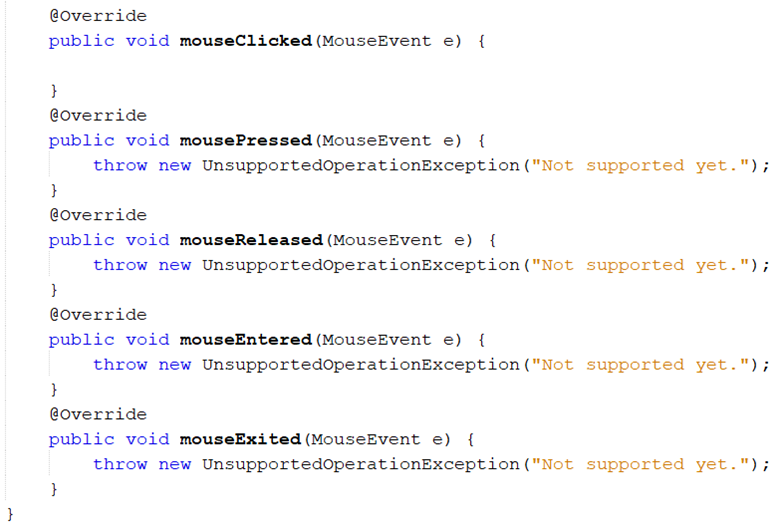
El proceso que deberá realizar el usuario es hacer doble clic para modificar el dato y luego clic en el botón Actualizar



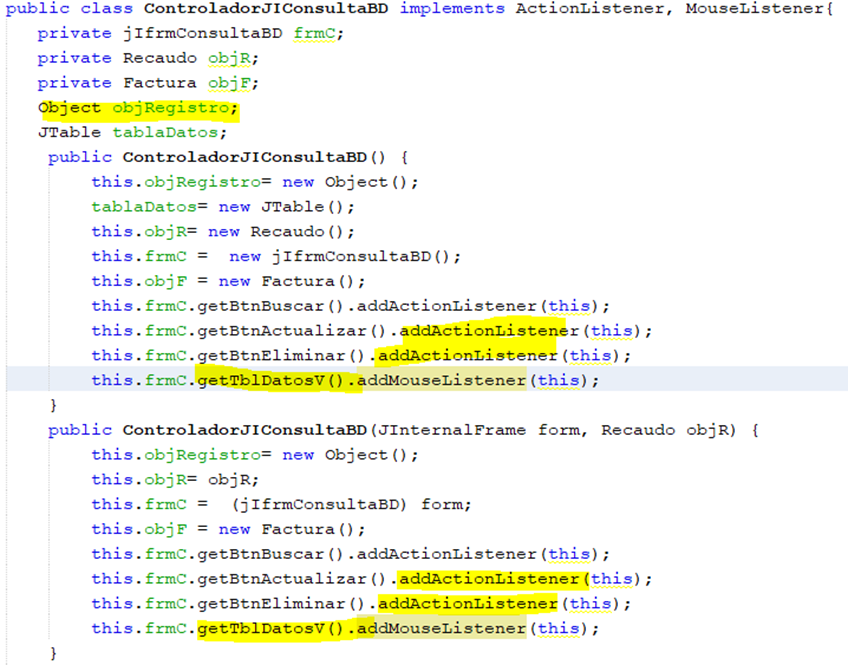
Para lo anterior, se agregará la interfaz **mouseListener** al controlador **ControladorJIConsultaBD**, la cual incorpora los métodos mouseClicked(), mousePressed(), mouseReleased(), mouseEntered() y mouseExited().

Se sugiere **eliminar** los throw new, para evitar que se generen mensajes de Excepción al realizar las acciones con el mouse.

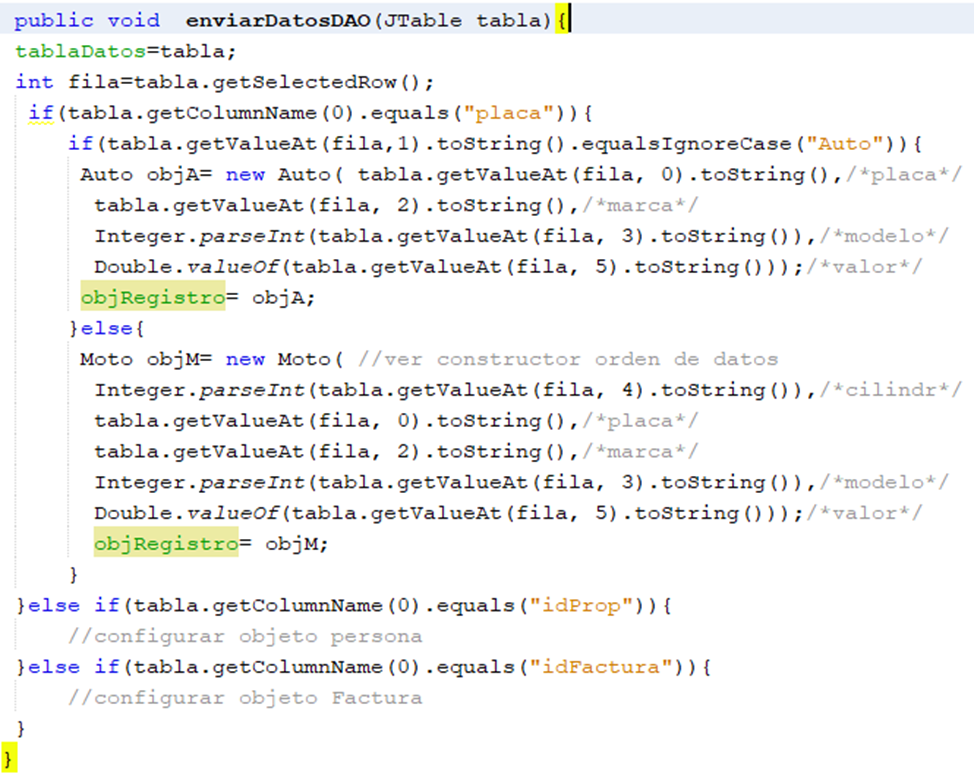




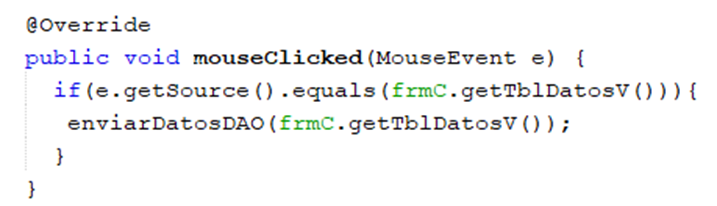
Es necesario agregar los **mouseListener** a las tablas del **ControladorJIConsultaBD**, igual que los **actionListener**() de los Botones **actualizar** y **eliminar** en los dos constructores. Adicionalmente se agregará el objeto de tipo **Object** llamado objRegistro para configurar cada tipo de objeto que se enviará a los métodos actualizar o eliminar de FacturaDAO.



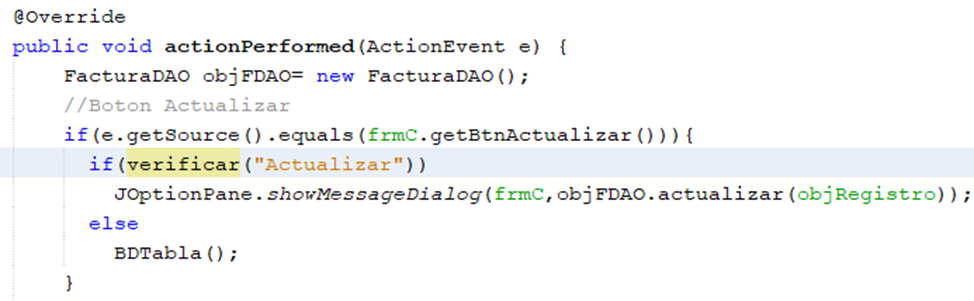
Se implementa el método **enviarDatosDAO**() dentro de **ControladorJIConsultaBD**, que permite generar cada objeto Auto, Moto, Persona o Factura, a partir de los datos que contenga la tabla que reciba el método y almacenarlo dentro del objeto objRegistro, de esta manera se enviarán los datos a la clase FacturaDAO para que se ejecute el comando correspondiente.



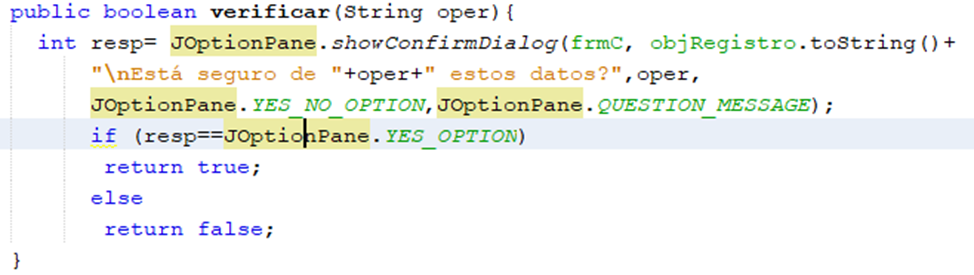
En el método **mouseClicked**() de **ControladorJIConsultaBD**, crear el condicional para verificar si se hizo clic en la tabla, en este caso la tabla de vehículos, la cual se enviará al método enviarDatosDAO().

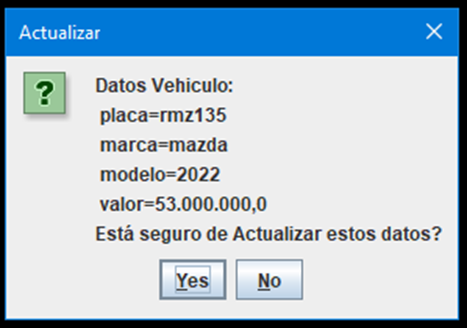


Finalmente, en el **actionPerformed**() de **ControladorJIConsultaBD**, se crea el objeto objFDAO de FacturaDAO y se agrega el condicional del botón actualizar, que deberá verificar si el usuario si desea realizar la actualización del registro para que se ejecute el método actualizar (), de lo contrario se restaurará la consulta mediante el método BDTabla().



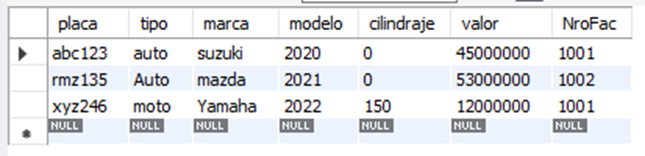
El método verificar genera una caja de dialogo que visualiza el contenido del objRegistro y la operación (**oper**) que se requiere ejecutar ("Actualizar " o "Eliminar") para que el usuario confirme la actualización de los datos.



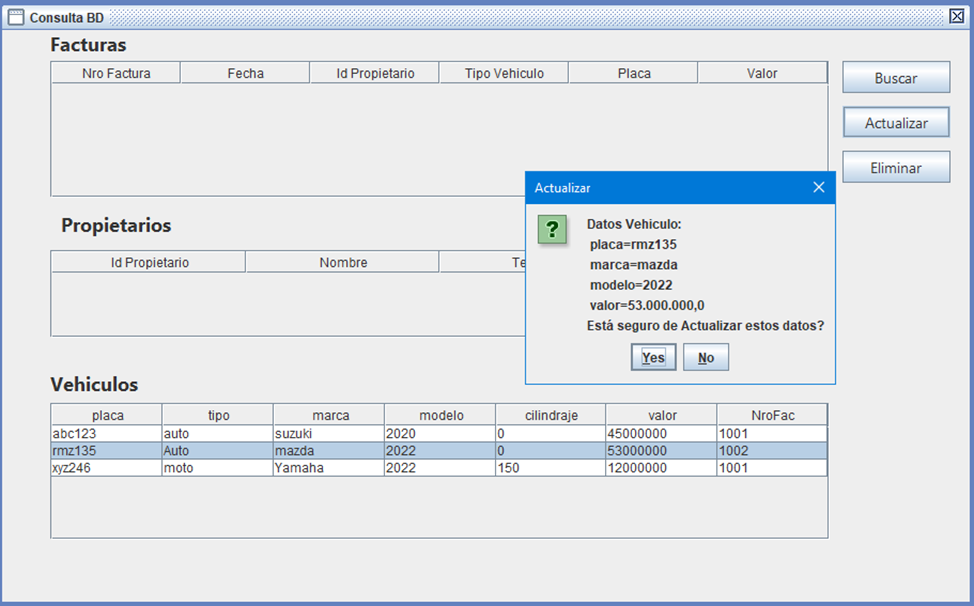


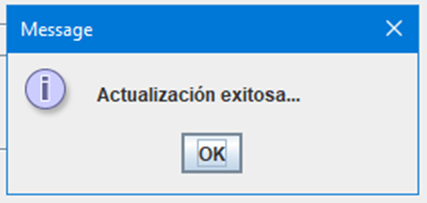
**Realizar ejecución**

Datos originales

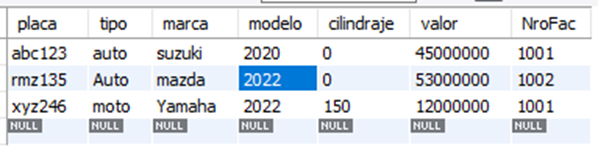


Ejecución modificando campo **modelo**





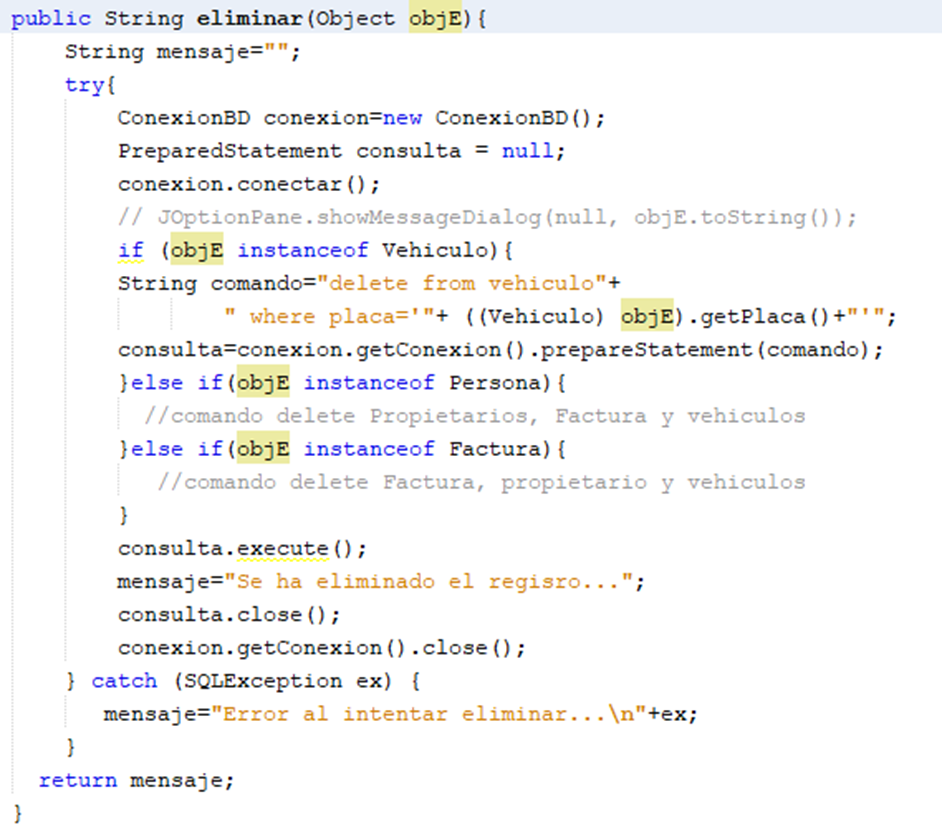
**Datos Finales**



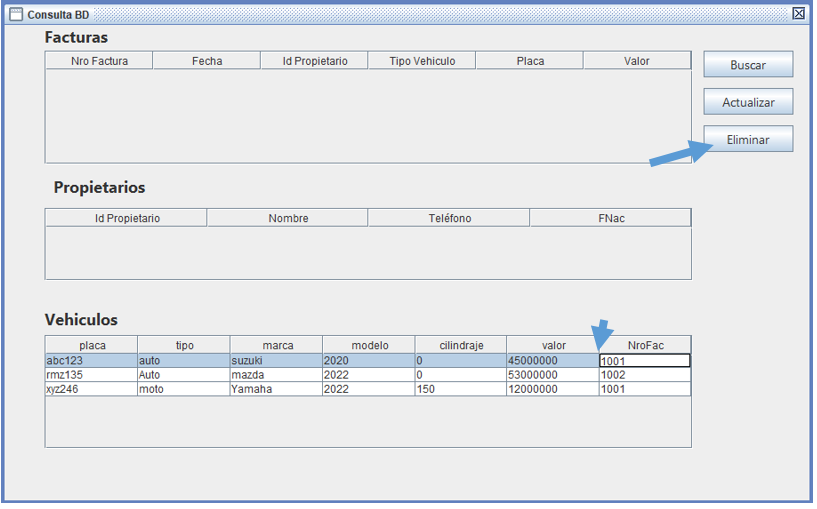
* **Procedimiento para Eliminar datos,**

**En la clase FacturaDAO,** se implementa el método eliminar que recibirá un objeto de tipo Object, de la misma manera que en el método actualizar, corresponderá a los tipos de objeto que se están trabajando en la BD (Auto, Moto, Persona o Factura), con los cuales se configurará la instrucción **delete** que será ejecutada, finalmente el método retornará un mensaje indicando si la eliminación ha sido exitosa o se ha generado error.

Es importante recordar que hay dependencias transitivas, es decir que si se desea eliminar una Factura o un Propietario se deberán eliminar todos los demás objetos asociados a estas entidades.



El proceso que debe realizar el usuario es hacer clic en el registro que desea eliminar dentro de la tabla y luego activar el botón Eliminar.

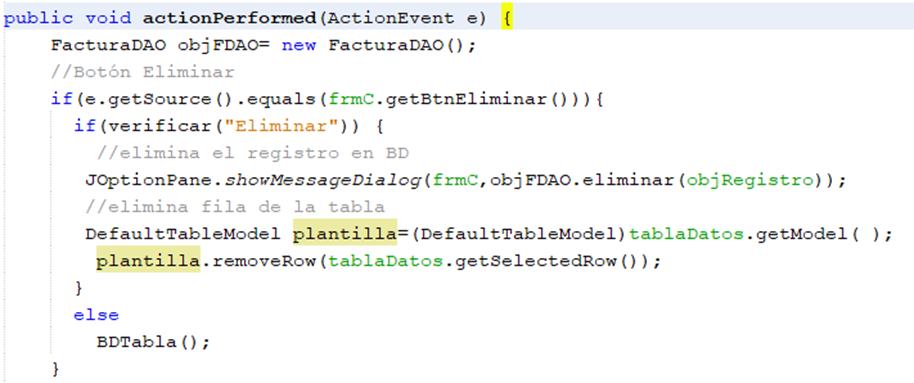


Este proceso también utilizará el método **enviarDatosDAO**() que se activa cada vez que se hace click en cualquiera de las tablas, mediante el método mouseClicked().

Ya que se pretende es borrar el registro de la BD, también deberá eliminarse la fila que el usuario ha seleccionado dentro de la tabla. Para ello es necesario almacenar dicha la tabla mediante el objeto tablaDatos, el cual debe definirse en el **ControladorJIConsutlaBD**.

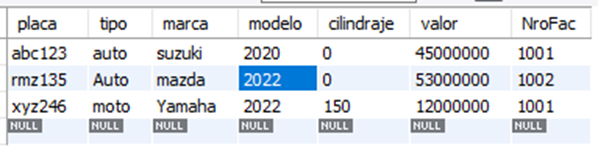


Posteriormente, en el **actionPerformed**() de **ControladorJIConsultaBD**, se agrega el condicional del botón eliminar, que al igual el método actualizar verifica si el usuario si desea eliminar el registro para enviar el objeto objRegistro y al mismo tiempo eliminando la fila de la tabla seleccionada por el usuario (mediante el uso de la clase **DefaultTableModel** y el método **removeRow**() de la tablaDatos), de lo contrario se restaurará la consulta mediante el método BDTabla().

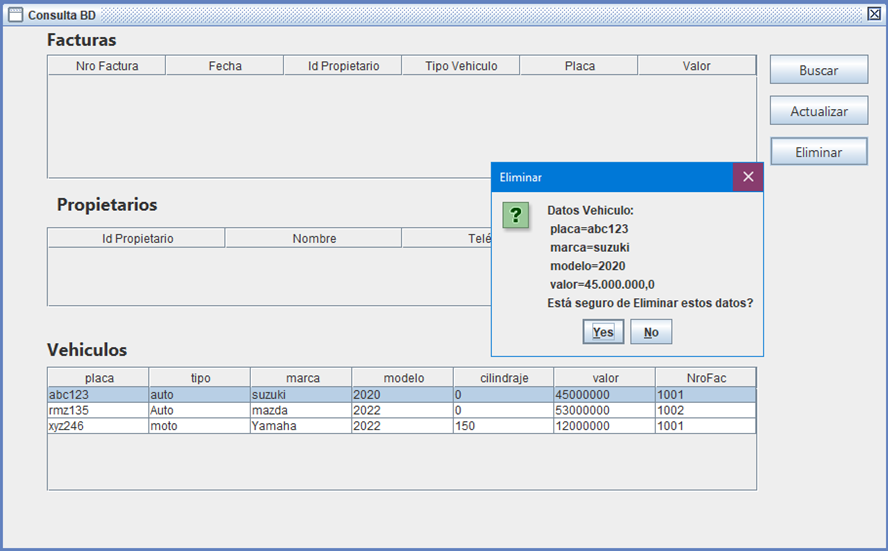


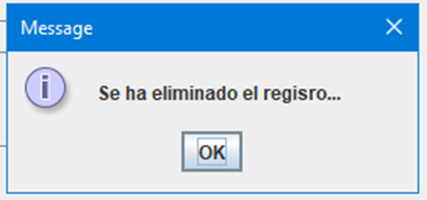
**Realizar la ejecución**

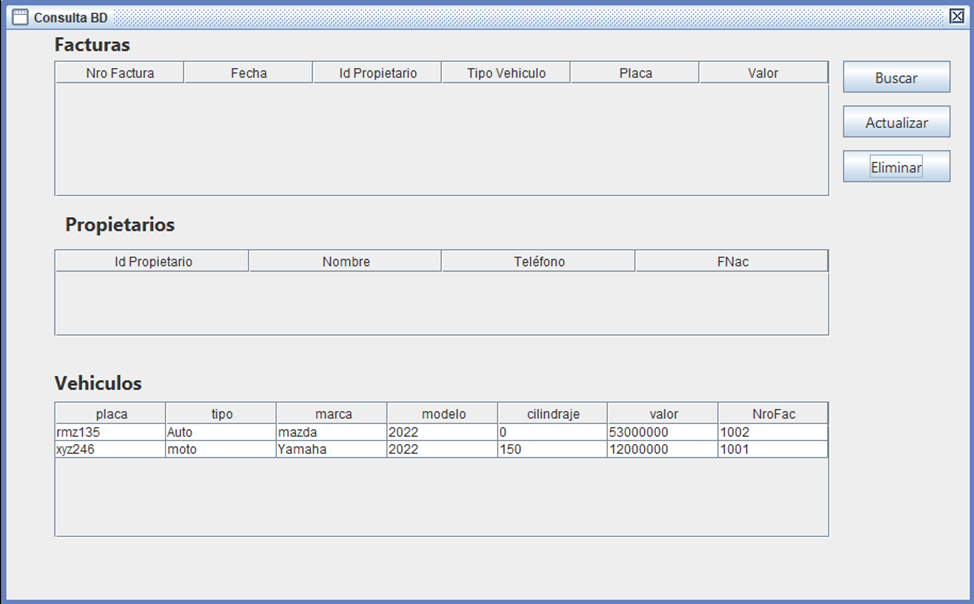
**Datos originales**



**Ejecución eliminando el primer registro**







Datos finales

