



Colegio Mexiquense Universitario

Ingeniería en Sistemas Computacionales

Manual Técnico: Car Agency

Docente: Vianey Soltero Pérez

Alumnos: Sergio Rangel Vargas

Axel Isaac Cejudo Román

4to Semestre

Contenido

Objetivos	4
Pseudocódigo	5
Diagrama de flujo	7
Instalación de NetBeans IDE 8.2	8
Requisitos de software	8
Opciones de descarga del instalador	8
Personalización de la instalación	8
Inicio de la descarga	9
Instalación del software	9
Microsoft Windows, Solaris OS y Linux	9
Mac OS X	10
Paquete independiente de la plataforma	11
Instalación de AppServ	12
Hosting	13
Registro, Creación y Subida de la base de datos Mysql	14
Conexión y Uso de la base de datos	21
Código fuente	22
ConnectionSQL	22
FILES	24
MethodsSQL	26
Mylcon	50
UsersEntity	51
Main	53
Login	53
Menu	60
ModifyUser	66
Register	94
Inventory	108
MenuAdmin	115
PanelAdd	115
PanelDelete	130

PanelEdit	144
PanelSearch	158
Sales	163
Users.....	170
MenuUser.....	180
Purchase	190
Quotation	221
Vehicle.....	240

Objetivos

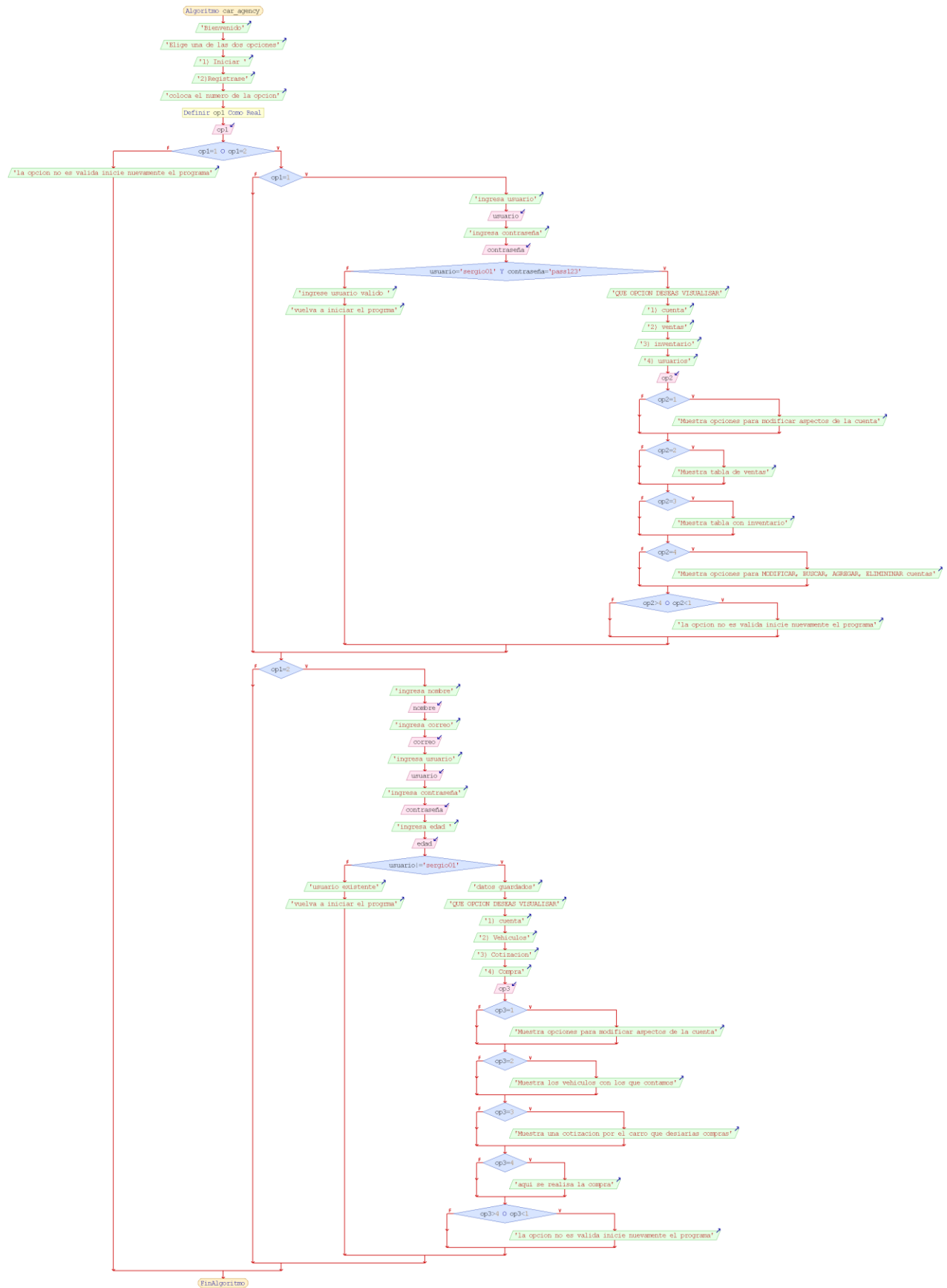
- ✓ Realizar una conexión a una base de datos en NetBeans
- ✓ Aplicar conocimientos aprendidos durante el semestre
- ✓ Usar medios de investigación para facilitar la construcción del programa ya sea libros, audiolibros, navegadores web.

Pseudocódigo

1. Algoritmo car_agency
2. Escribir "Bienvenido";
3. Escribir "Elige una de las dos opciones";
4. Escribir "1) Iniciar ";
5. Escribir "2) Registrase";
6. Escribir "coloca el numero de la opción";
7. Definir op1 como numero;
8. Leer op1;
9. Si op1=1 || op1=2 Entonces
10. Si op1=1 Entonces
11. Escribir "ingresa usuario";
12. Leer usuario
13. Escribir "ingresa contraseña";
14. Leer contraseña
15. Si usuario="sergio01" && contraseña="pass123" Entonces
16. Escribir "QUE OPCION DESEAS VISUALISAR";
17. Escribir "1) cuenta";
18. Escribir "2) ventas";
19. Escribir "3) inventario";
20. Escribir "4) usuarios";
21. Leer op2
22. Si op2=1 Entonces
23. Escribir "Muestra opciones para modificar aspectos de la cuenta";
24. Fin Si
25. Si op2=2 Entonces
26. Escribir "Muestra tabla de ventas";
27. Fin Si
28. Si op2=3 Entonces
29. Escribir "Muestra tabla con inventario";
30. Fin Si
31. Si op2=4 Entonces
32. Escribir "Muestra opciones para MODIFICAR, BUSCAR, AGREGAR, ELIMINAR cuentas";
33. Fin Si
34. Si op2>4 || op2<1 Entonces
35. Escribir "la opción no es válida inicie nuevamente el programa";
36. Fin Si
37. SiNo
38. Escribir "ingrese usuario valido ";
39. Escribir "vuelva a iniciar el programa";
40. Fin Si
41. Fin Si

42. Si op1=2 Entonces
43. Escribir "ingresa nombre";
44. Leer nombre;
45. Escribir "ingresa correo";
46. Leer correo;
47. Escribir "ingresa usuario";
48. Leer usuario;
49. Escribir "ingresa contraseña";
50. Leer contraseña;
51. Escribir "ingresa edad ";
52. Leer edad;
53. Si usuario!="sergio01" Entonces
54. Escribir "datos guardados";
55. Escribir "QUE OPCION DESEAS VISUALISAR";
56. Escribir "1) cuenta";
57. Escribir "2) Vehículos";
58. Escribir "3) Cotización";
59. Escribir "4) Compra";
60. Leer op3;
61. Si op3=1 Entonces
62. Escribir "Muestra opciones para modificar aspectos de la cuenta";
63. Fin Si
64. Si op3=2 Entonces
65. Escribir "Muestra los vehículos con los que contamos";
66. Fin Si
67. Si op3=3 Entonces
68. Escribir "Muestra una cotización por el carro que desearías comprar";
69. Fin Si
70. Si op3=4 Entonces
71. Escribir "aquí se realiza la compra";
72. Fin Si
73. Si op3>4 || op3<1 Entonces
74. Escribir "la opción no es válida inicie nuevamente el programa";
75. Fin Si
76. SiNo
77. Escribir "usuario existente";
78. Escribir "vuelva a iniciar el programa";
79. Fin Si
80. Fin Si
81. SiNo
82. Escribir "la opción no es válida inicie nuevamente el programa";
83. Fin Si
84. FinAlgoritmo

Diagrama de flujo



Instalación de NetBeans IDE 8.2

Requisitos de software

Antes de instalar el IDE, debe tener instalado en el sistema la actualización 19 de Java SE Development Kit (JDK) (versión 1.5.0_19) o la actualización 14 de JDK 6 o posterior. Si no tiene una instalación de JDK, no puede continuar con la instalación. Puede descargar la versión más reciente de JDK en <http://java.sun.com/javase/downloads>.

Para las opciones de descarga de PHP, Ruby y C/C++ puede instalar JRE en lugar de JDK. Sin embargo, si piensa usar cualquiera de las funciones de Java, necesitará JDK.

Opciones de descarga del instalador

En la página de descarga de NetBeans IDE puede obtener uno de los diversos instaladores, cada uno de los cuales contiene el IDE básico y herramientas adicionales.

Los siguientes instaladores están disponibles:

Java. Proporciona herramientas para desarrollar aplicaciones de Java SE, Java EE y Java ME. Esta opción de descarga también incluye el software GlassFish v3 y Apache Tomcat 6.0.20.

JavaFX. Proporciona compatibilidad para el desarrollo de aplicaciones multiplataforma de JavaFX.

Java SE. Proporciona todas las características estándar para el desarrollo de Java SE. También se incluye la compatibilidad con el desarrollo de RCP de la plataforma NetBeans.

Ruby. Proporciona herramientas para el desarrollo de Ruby y es compatible con Rails y JRuby. También contiene GlassFish v3.

PHP. Proporciona herramientas para el desarrollo de PHP.

C/C++. Es compatible con el desarrollo en los lenguajes C, C++, Qt, Fortran y Assembly.

All (Todo). Es una opción de descarga completa que contiene todos los tiempos de ejecución y los paquetes disponibles para NetBeans IDE.

Personalización de la instalación

Si realiza la descarga mediante las opciones Java o "All" (Todo), puede seleccionar qué herramientas y tiempos de ejecución que desea instalar. Haga clic en Personalizar en la página de bienvenida del instalador y seleccione las características que desee usar. Para obtener más información, consulte Instalación del software.

Si desea agregar funciones a la configuración del NetBeans IDE, utilice NetBeans Plugin Manager. Por ejemplo, suponga que descarga e instala el paquete de Ruby. A continuación, decide que desea probar las funciones de C/C++. Para ello, hay que acceder a NetBeans Plugin Manager desde el IDE (seleccione Herramientas > Plugins) y agregar el paquete C/C++ a la instalación existente.

Hay varias notas importantes sobre la instalación y la configuración de algunas funciones de NetBeans IDE:

Apache Tomcat 6.0.20 se incluye en las opciones de descarga "Java" y "All" (Todo), pero no se instala de forma predeterminada con ninguna de ellas. Para instalar Apache Tomcat a partir de estas descargas, inicie el instalador y seleccione Apache Tomcat 6.0.20 en el cuadro de diálogo Personalizar la instalación.

Para el desarrollo de C/C++, aplicable a los tipos de descarga "C/C++" y "All" (Todo), deberá instalar además compiladores y herramientas. Para obtener más información, consulte Configuración de NetBeans IDE para C/C++/Fortran.

En Mac OS X, puede personalizar la instalación de NetBeans IDE en las opciones de descarga de Java, Ruby, y "All" (Todo). Esta posibilidad no está disponible en las descargas JavaSE, PHP y C/C++.

También puede descargar un archivo comprimido independiente de la plataforma y ejecutar el archivo ejecutable de NetBeans IDE. Tenga en cuenta que los tiempos de ejecución, como los servidores de aplicaciones de GlassFish y Apache Tomcat, sólo están disponibles con instaladores para plataformas específicas.

Inicio de la descarga

Vaya a la página de descargas de NetBeans IDE.

En la parte superior derecha de la página, seleccione el idioma y la plataforma en la lista desplegable. También puede optar por descargar y utilizar el archivo zip independiente de la plataforma.

Haga clic en el botón Download (Descargar) de la opción de descarga que desee instalar.

Guarde el archivo del instalador en el sistema.

Instalación del software

Siga las instrucciones que aparecen en esta sección para instalar el IDE en el sistema. Estas instrucciones de instalación se aplican a todas las plataformas compatibles. Para obtener la lista de las plataformas compatibles, consulte las notas de la versión.

Microsoft Windows, Solaris OS y Linux

Para instalar el software:

Una vez finalizada la descarga, ejecute el instalador.

El archivo del instalador de Windows tiene la extensión .exe. Haga doble clic en él para ejecutarlo.

El archivo del instalador de las plataformas Solaris y Linux tiene la extensión .sh. En estas plataformas debe convertir los archivos del instalador en ejecutables mediante el siguiente comando: `chmod +x <nombre-archivo-instalador>`

Para seleccionar las herramientas y tiempos de ejecución que se van a instalar, siga estos pasos en la página de bienvenida del asistente de instalación:

Haga clic en Personalizar.

En el cuadro de diálogo Personalizar la instalación, realice las selecciones.

Haga clic en Aceptar.

En la página de bienvenida del asistente para la instalación, haga clic en Siguiente.

En la página del contrato de licencia, revise el contrato, marque la casilla de aceptación y haga clic en Siguiente.

En la página de instalación de NetBeans IDE, siga estos pasos:

Acepte el directorio de instalación predeterminado de NetBeans IDE o especifique otro directorio. Nota: el directorio de instalación debe estar vacío y el perfil de usuario que utilice para ejecutar el instalador debe disponer de permisos de lectura/escritura en dicho directorio.

Acepte la instalación de JDK predeterminada para usarlo con NetBeans IDE o seleccione una instalación diferente en la lista desplegable. Si el asistente de instalación no encuentra una instalación de JDK compatible para usar con NetBeans IDE, no se instalará en la ubicación predeterminada. En este caso, especifique la ruta de un JDK instalado o cancele la instalación actual, instale la versión de JDK requerida y reinicie esta instalación.

Haga clic en Siguiente.

Si se abre la página de instalación de GlassFish v3, acepte el directorio de instalación predeterminado o especifique otra ubicación para la instalación.

Si va a instalar Apache Tomcat: en la página de instalación, acepte el directorio de instalación predeterminado o especifique otra ubicación para la instalación.

En la página de resumen, compruebe que la lista de componentes que se van a instalar es correcta y que dispone de espacio suficiente en el sistema para la instalación.

Haga clic en Instalar para comenzar la instalación.

Nota: si surgen problemas para instalar el software correctamente, consulte Resolución de problemas para obtener descripciones y sugerencias para solucionar problemas no resueltos que pueden afectar al proceso de instalación.

Mac OS X

Para instalar el software:

Una vez finalizada la descarga, ejecute el instalador. La extensión del archivo del instalador es .dmg.

En los paneles que se abren, haga clic en el icono del paquete. La extensión del paquete es .mpkg. El asistente de instalación comienza.

En la página de bienvenida del asistente para la instalación, haga clic en Continuar.

Revise el contrato de licencia y haga clic en Continuar. Haga clic en Aceptar en la ventana emergente para aceptar la licencia.

En la página de selección de destino, elija la unidad y haga clic en Continuar.

Si ha descargado los paquetes de "All" (Todo), Java o Ruby, puede personalizar la instalación. En el último panel del asistente de instalación, pulse el botón Personalizar de la parte inferior izquierda del panel. Aparece el árbol de productos.

Seleccione los productos que desea instalar.

Especifique el nombre y la contraseña del administrador del sistema y haga clic en Aceptar para comenzar la instalación.

Paquete independiente de la plataforma

Para ejecutar el software:

Cuando finalice la descarga del archivo ZIP independiente de la plataforma, extraígallo a cualquier carpeta del sistema.

Ejecute el archivo ejecutable ubicado en el directorio netbeans/bin.

Acepte el contrato de licencia. NetBeans IDE se inicia.

Nota: tenga en cuenta que los tiempos de ejecución, como el servidor de aplicaciones de GlassFish y Apache Tomcat, sólo están disponibles con instaladores para plataformas específicas.

Nota: si no puede encontrar la instalación de JDK compatible, quizá deba agregar la ruta del directorio de instalación de JDK de forma manual:

Abra el archivo netbeans.conf que se encuentra en netbeans/etc en un editor de texto.

Introduzca la ubicación de la instalación de JDK compatible como valor de la opción netbeans_jdkhome.

Guarde el archivo netbeans.conf y ejecute el archivo ejecutable de netbeans/bin

Instalación de AppServ

AppServ es un software que nos permite instalar sobre Windows Apache, PHP, MySQL y phpMyAdmin (interfaz gráfica para administrar MySQL) de forma conjunta. Es una aplicación muy útil para empezar a familiarizarnos con Gestores de Contenidos, aprender a configurarlos e instalarlos, ya que nos permite realizar pruebas sobre nuestro pc y una vez que dominemos el tema podemos decidirnos a comprar un dominio y un alojamiento.

Para instalar AppServ os dejamos un paso a paso que os hará la tarea más facil:

- En primer lugar descargaremos el paquete de SourceForge, contando con dos versiones AppServ 2.5.5 (apache 2.0.55, MySQL 5.0.15 y PHP 5) o AppServ 2.4.5 (apache 1.3.34, MySQL 5.0.16 y PHP 4).
- Una vez descargado el ejecutable procedemos a ejecutar la instalación, recibiendo un mensaje de bienvenida y hacemos Click en Next
- **Directorio de Instalación:** en esta pantalla nos solicita el directorio en el que queremos instalar la aplicación, por defecto nos marca "c:/appserv", lo cambiamos si queremos y pulsamos Next.
- **Tipo de Instalación:** Elegimos Typical.
- **Server Information:** La casilla Server Name la dejamos como viene por defecto con "localhost" y en la casilla Administrator E-mail Address o lo dejamos como viene por defecto y luego lo cambiamos, o le introducimos una cuenta de correo que sera la del administrador. El campo "HTTP Port" lo dejamos como viene por defecto (80), salvo que queramos que el servidor atienda las peticiones en otro puerto. Pulsamos en Next
- **MySql Information:** Este es uno de los apartados más importantes ya que empezamos a instalar MySQL. En la casilla Username introduciremos el nombre del administrador de la base de datos y en la casilla Password le introducimos la contraseña, aunque si queremos hacerlo más fácil ponemos como usuario: root y el apartado contraseña lo dejamos vacío ya que posteriormente podemos cambiarlo con phpMyAdmin. El apartado Charset lo dejamos tal como viene por defecto. Pulsamos en Next
- **Progreso de la instalación:** Empieza realmente la instalación mostrándonos una barra de progreso hasta que aparece una pantalla que nos avisa de que ha finalizado la misma, debiendo dejar marcadas las opciones que vienen por defecto. Finalmente pulsamos en Close.
- **Comprobación de la instalación:** Si hemos hecho bien los deberes al poner en nuestro navegador: <http://localhost> nos debe aparecer la pantalla siguiente:

Si esta pantalla no aparece supone que bien Apache o MySQL no están funcionando bien, y debemos volver a instalarlo

Link de descarga: <https://www.appserv.org/en/download/>

Hosting

Es un servicio en línea que te permite publicar un sitio o aplicación web en Internet. Cuando te registras en un servicio de hosting, básicamente alquilas un espacio en un servidor donde puedes almacenar todos los archivos y datos necesarios para que tu sitio web funcione correctamente.



En este caso se utilizó el hosting para alojar la base de datos del programa “car agency” para poder acceder desde cualquier computadora mediante el programa creado. Se utilizó el servicio de HostBuddy cuyos precios son razonables a lo que ofrece sin embargo la base de datos está alojada por medio de un paquete de prueba de 60 días que cuenta con los requerimientos necesarios para el proyecto.

Free Hosting package

1 x Hosting plan "FREE PLAN"

Forced advertising	None
Storage capacity:	1000MB
Monthly transfer:	10GB/Month
# of Sites hosted:	1

ASP.Net 4.8, ASP.Net 2.0-4.7.2

PHP 5.2.X - 7.X

AJAX 3.5/1.0

Silverlight

MS Access 2003, 2007

Dedicated web application pool

1 x MS SQL2017/2016/2014 /2012/2008

MSSQL database size:	1000MB
----------------------	---------------

Backup/Restore/Attach Feature

1 x MySQL 5.x

MySQL database size:	1000MB
----------------------	---------------

Backup/Restore/Repair Feature

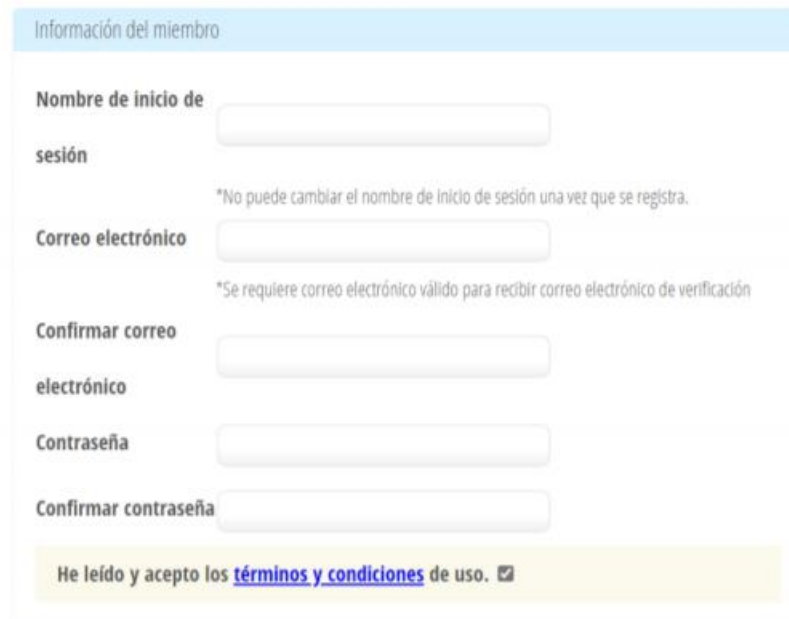
Las características de este servicio gratuito de alojamiento son:

- Sin publicidad
- Almacenamiento 1000MB MySQL
- MSSQL
- Transferencia 10GB/mes
- Copia de seguridad de BD
- Restauración de BD
- Reparación de BD
- ASP.Net 4.8, ASP.Net 2.0-4.7.2
- PHP 5.2.X - 7.X
- AJAX 3.5 / 1.0
- MS Access 2003, 2007

Registro, Creación y Subida de la base de datos Mysql

PASO 1 » Lo primero es entrar al link para registrarse y contratar el servicio gratuito de HostBuddy. Link: https://www.hostbuddy.com/secured_signup?plantype=&affiliate=

PASO 2 » Nos registraremos con nuestros datos en el formulario lo demás lo dejamos como están y procedemos a darle al botón “únete” si todo está correcto nos enviara a la siguiente ventana.



Formulario de registro de HostBuddy. El formulario tiene un título "Información del miembro" en un recuadro azul. Contiene los siguientes campos:

- Nombre de inicio de sesión:** Un campo de texto con un icono de ojo para alternar visibilidad. Debajo hay una nota: "*No puede cambiar el nombre de inicio de sesión una vez que se registra."
- Correo electrónico:** Un campo de texto. Debajo hay una nota: "*Se requiere correo electrónico válido para recibir correo electrónico de verificación"
- Confirmar correo electrónico:** Un campo de texto.
- Contraseña:** Un campo de texto.
- Confirmar contraseña:** Un campo de texto.

En la parte inferior, hay una barra amarilla que dice: "He leído y acepto los [términos y condiciones](#) de uso." con una casilla de verificación marcada.

PASO 3 » Ya registrado se tendrá que revisar el correo para verificar la cuenta



HostBuddy
fast & reliable webhosting

Servicio al cliente 24x7 **1-877-936-2284**

Casa Alojamiento ASP.NET gratis Planes de alojamiento \$12/Dominio Afiliados

¡Inscripción exitosa a la cuenta de hosting gratuita!

 **Verificación por correo electrónico pendiente**

Se envió una verificación por correo electrónico a su buzón de correo. Compruebe su correo electrónico y haga clic en el enlace de verificación en el correo electrónico. Si no recibe el correo electrónico en 5 a 10 minutos, póngase en contacto con nuestro soporte a través de nuestro servicio de asistencia o chat en vivo.
(Por favor, asegúrese de comprobar su carpeta SPAM en caso de que se marcó erróneamente como Spam.)

SMS Verificación (opcional)

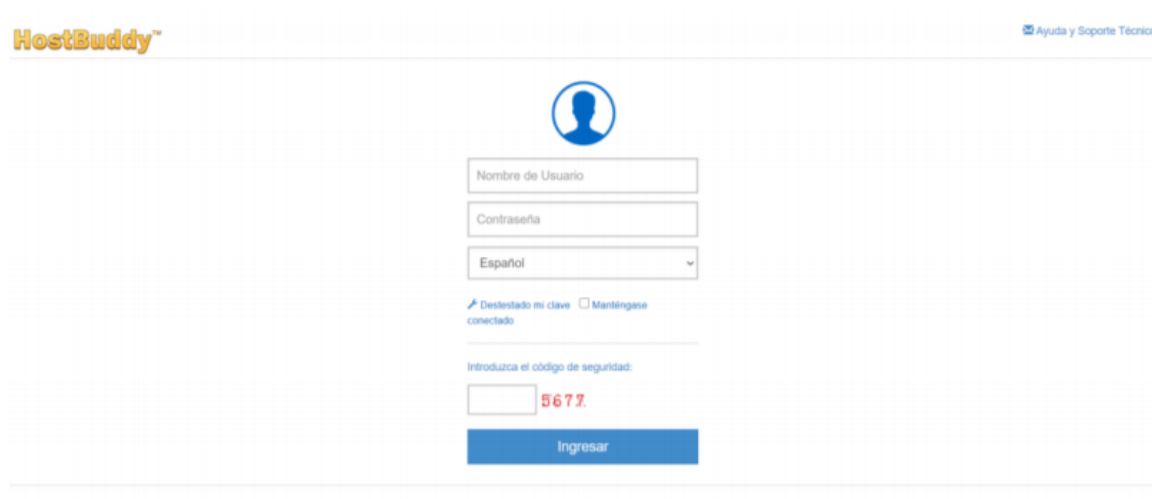
También puede verificar su acout utilizando la verificación de SMS. Para verificar a través de SMS, inicie sesión en su cuenta ahora y verá la opción de verificar su cuenta a través de un mensaje SMS.

¿Cómo activar el plan de alojamiento gratuito?

Una vez que su cuenta esté verificada, inicie sesión en su cuenta y vaya a "Cuentas de alojamiento" >> "Plan de alojamiento gratuito".

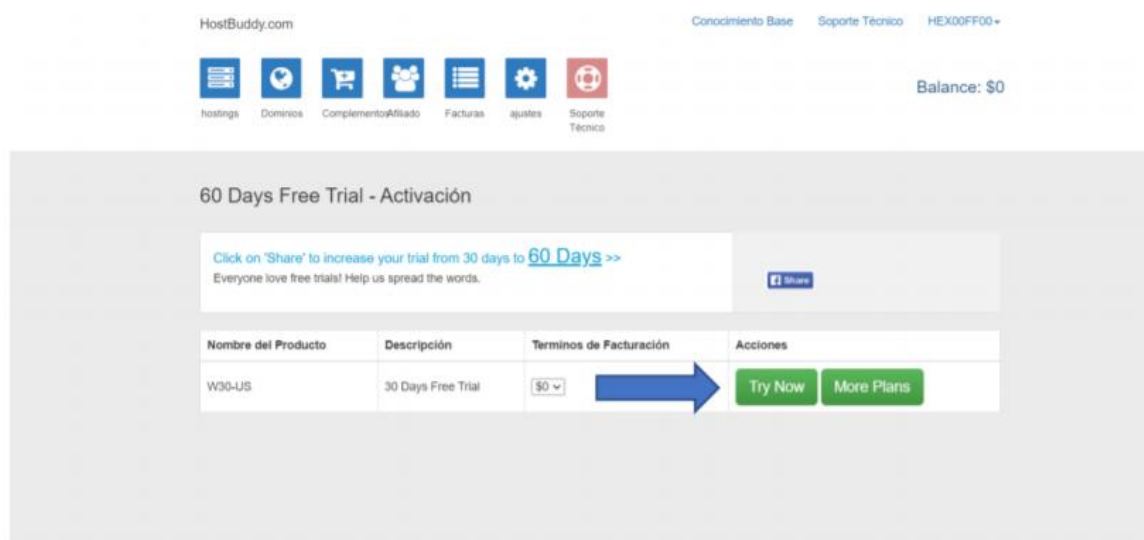


Pase 4 » Luego de verificar nuestra cuenta nos enviara a la ventana de login donde ingresaremos el usuario y contraseña con el que nos registramos.



The image shows the HostBuddy login page. At the top left is the HostBuddy logo, and at the top right is a link for 'Ayuda y Soporte Técnico'. The main content area features a user icon placeholder, followed by input fields for 'Nombre de Usuario' and 'Contraseña'. Below these is a language dropdown menu set to 'Español'. There are two checkboxes: 'Desactivado mi clave' (checked) and 'Manténgase conectado'. A security code field shows '5677' with the prompt 'Introduzca el código de seguridad:'. At the bottom is a blue 'Ingresar' button.

PASO 5 » Al ingresar tendremos esta ventana que nos dice que para tener los 60 días de prueba gratis tenemos que compartir la página por Facebook, en caso de no querer hacerlo simplemente le damos al botón verde que dice “try now” para comenzar el servicio.



The image shows the HostBuddy dashboard after login. The top navigation bar includes 'HostBuddy.com', 'Conocimiento Base', 'Soporte Técnico', and a user profile 'HEX00FF00+'. Below this is a row of icons for 'hostings', 'Dominios', 'Complementos/Afiliado', 'Facturas', 'ajustes', and 'Soporte Técnico'. The main content area is titled '60 Days Free Trial - Activación'. It contains a message: 'Click on "Share" to increase your trial from 30 days to 60 Days >>'. Below this is a table with trial plans.

Nombre del Producto	Descripción	Terminos de Facturación	Acciones
W30-US	30 Days Free Trial	\$0 v	Try Now More Plans

PASO 6 » Ingresaremos un nombre de dominio para empezar a usar el servidor y marcamos la casilla para aceptar los términos y condiciones los cuales menciona que no usaremos el servicio con la intención de hacer phishing luego le daremos al botón “next”.

☺ Instalación New Hosting Account

¿Qué es el ID de la Cuenta Hosting?

Normalmente, usted ingresa a su 'Panel de Control' / 'Centro de Cuenta' con su ID de Cuenta ('hex00f00'). Sin embargo, cada cuenta hosting viene con un ID de Cuenta Hosting. Usted puede dar el ID de la Cuenta Hosting de más abajo a los miembros que puedan SOLO acceder a su panel de control del hosting, no a su Centro de Cuenta.







Puede además crear más de una ID de Cuenta Hosting si es necesitada.

☺ Por favor Ingrese New Hosting Account Información

PHP 5.6.x
-->


Hosting ID de la cuenta:	hex00f00-001 (Usted puede crear más adelante Login ID adicional para acceder a esta cuenta.)
	<input checked="" type="checkbox"/> Usar mi contraseña actual 'Cuenta de usuario ID'.
Primer portal carpeta raíz Nombre:	site1 (Usted puede cambiar esto a cualquier nombre que desee)
	<input type="checkbox"/> I understand that All Facebook Bot and Phishing sites will be deleted without any notice.
	Next

PASO 7 » Para finalizar el registro solo tenemos que darle al botón “enviar” y esperar entre 1 a 3 minutos a que termine el registro para poder utilizar los servicios.

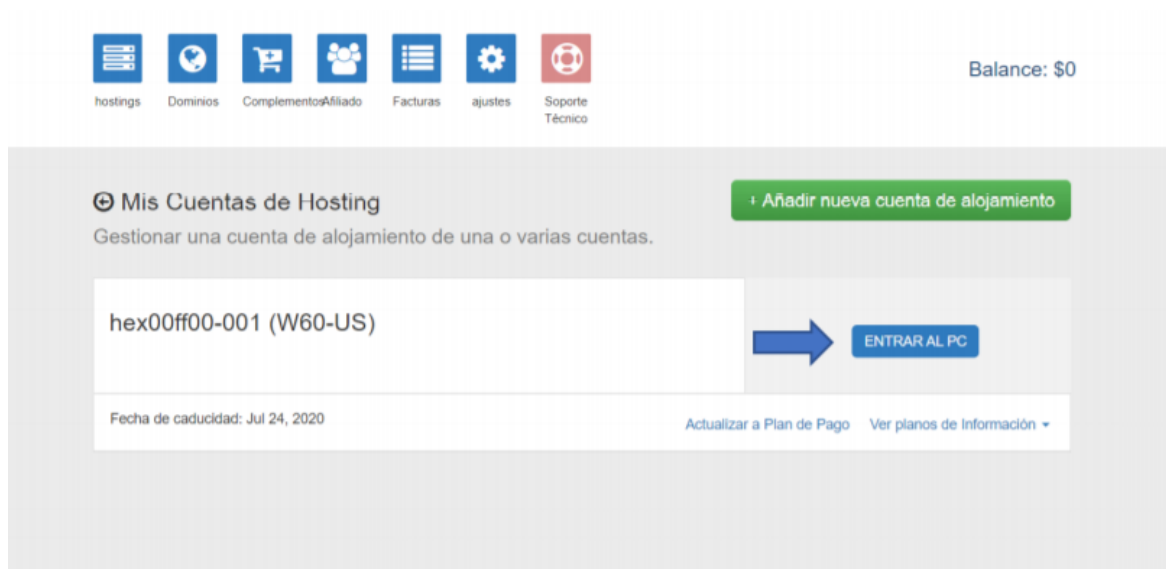


Balance: \$0

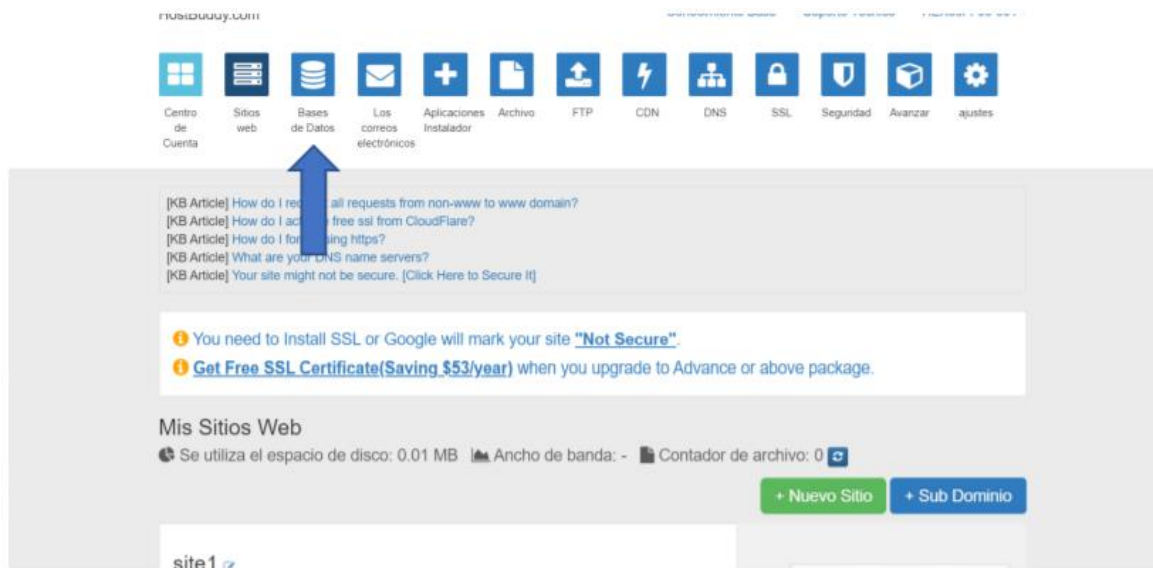
☺ Confirmar New Hosting Account Información

Hosting ID de la cuenta:	hex00f00-001
Primer portal carpeta raíz Nombre:	site1
Hosting Nombre del plan:	W60-US
	Por favor registre la siguiente FTP information:
Dirección FTP :	FTP:SITE4NOW.NET
Nombre de Usuario FTP :	hex00f00-001
Contraseña FTP :	(la misma que la contraseña del panel de control)
Directorio FTP	\
	 Enviar

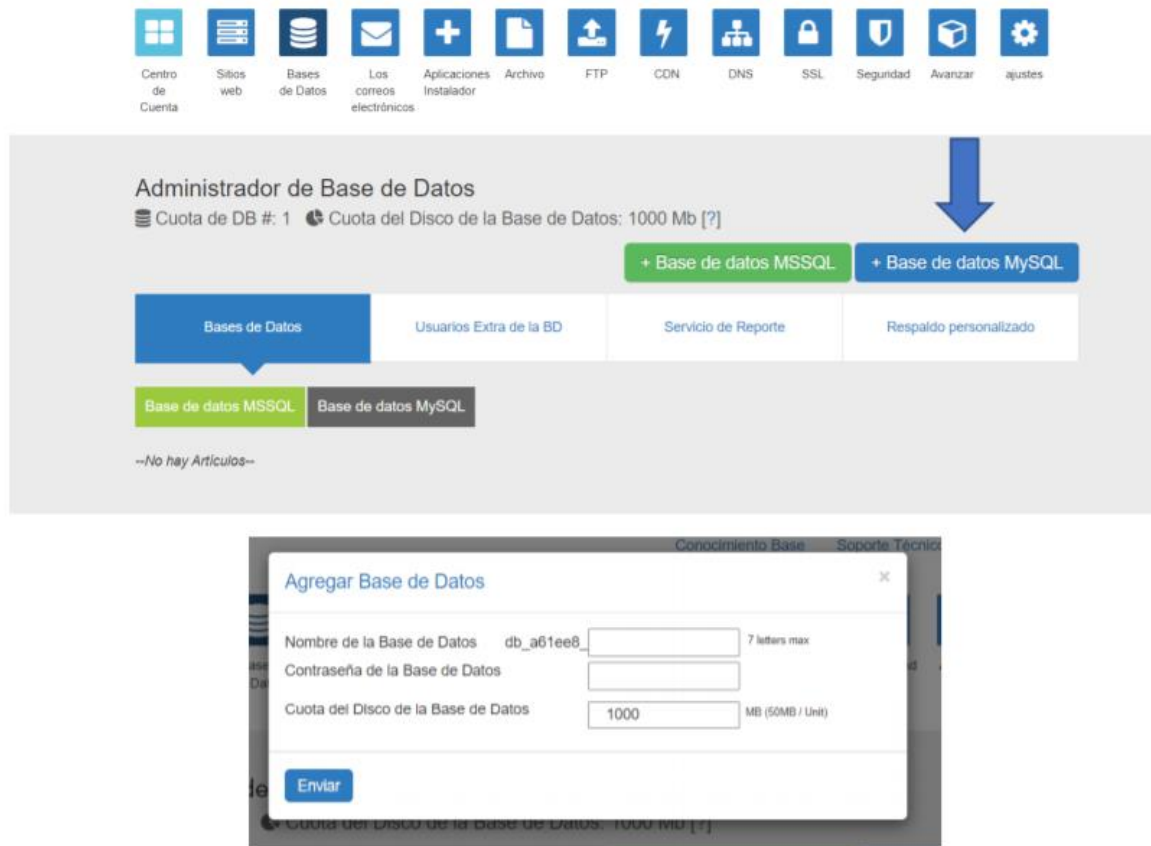
PASO 8 » Una vez terminado el registro podremos empezar dándole al botón azul que dice “ENTRAR AL PC” esto nos enviara al menú central del servidor.



PASO 9 » Como solo utilizaremos el servicio de alojamiento para bases de datos ignoraremos todas las demás opciones e iremos directo al botón que dice “Bases de Datos”.



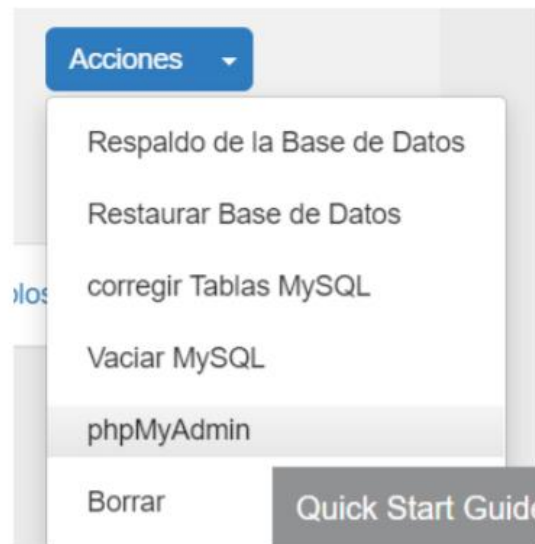
PASO 10 » Seleccionar la opción de “+ Base de datos MySQL” se nos desplegara un recuadro donde se ingresara el nombre de la base de datos, la contraseña y el tamaño para la misma.



PASO 11 » Se creara la base y para acceder le damos al botón “Acciones” para desplegar las opciones disponibles para trabajar la base de datos, además de mostrarnos los datos como el: nombre de la base, host, usuario, estos datos se utilizan en la conexión con java.



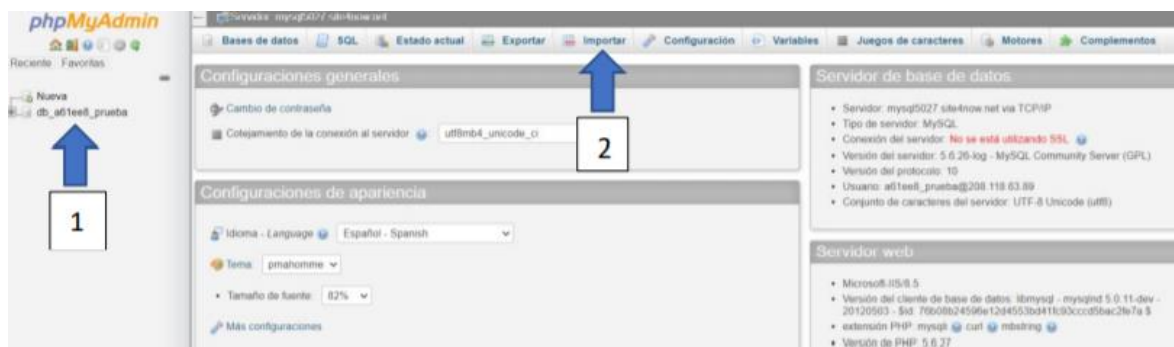
PASO 12 » Para acceder a la base de datos seleccionamos la opción de "phpMyAdmin" para después subir el archivo .sql de nuestra base de datos.



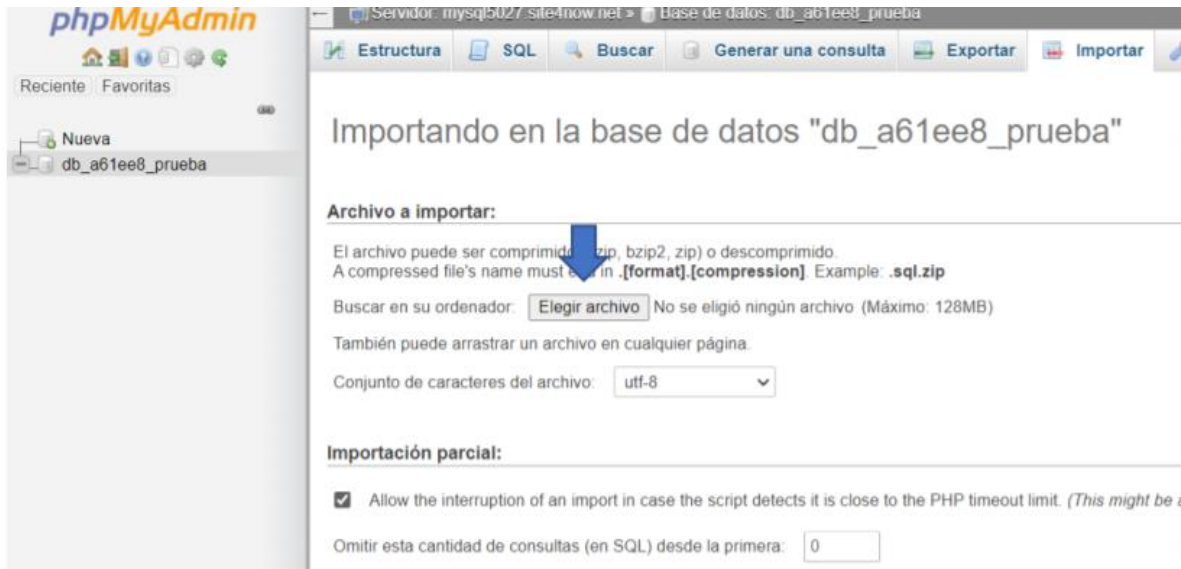
PASO 13 » Para ingresar nos pedirá el nombre del servidor, usuario y contraseña, estos datos nos lo dan anteriormente en el panel de menú de las bases de datos PASO 11.



PASO 14 » Una vez adentro de "phpMyAdmin" le daremos clic a la base creada y luego a la opción "Importar" para subir la base de datos.



PASO 15 » Aquí le daremos al botón que dice “Elegir archivo” y abrirá otra ventada donde se tendrá que buscar al archivo .sql de nuestra base de datos para poder subirla al servidor hosting.



PASO 16 » Una vez seleccionado el archivo daremos clic al botón de “Continuar” en la parte inferior y si todo sale correctamente se verán mensajes en color verde además de que aparecerá.

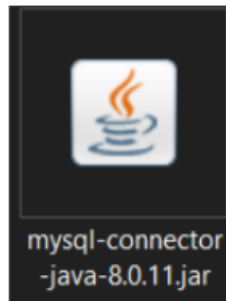


Una vez realizado todos los paso ya tendremos nuestro registro del servicio de hosting gratuito en HostBuddy junto a la base de datos creada en el con todos los datos subido de nuestra base de datos creada en el localhost pero ahora en un servidor.

Conexión y Uso de la base de datos

Una conexión mediante un servidor hosting no es diferente a la conexión realizada mediante un servidor local solo se necesita contar los siguientes requisitos:

- El controlador .jar encargado de hacer la conexión y el que nos permite implementar el código dentro de nuestro compilador de código ya no viene por defecto en él.



- El código para nuestro compilador de código que son las instrucciones que tiene que realizar nuestro programa para conectarse a la base de datos y así poder interactuar con ella mediante nuestro programa.

```
8 public class ConnectionSQL {
9
10     Connection con;
11
12     public ConnectionSQL(){
13         con = null;
14         try {
15             Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
16             String url = "jdbc:mysql://mysql5027.site4now.net/db_a6112a_caragyn?useTimezone=true&serverTimezone=UTC";
17             String user = "a6112a_caragyn";
18             String password = "pass1234";
19             con = DriverManager.getConnection(url, user, password);
20             if(con != null){
21                 System.out.println("Conexion Establecida..");
22             }
23         } catch (ClassNotFoundException e) {
24             JOptionPane.showMessageDialog(
25                 null, "EL CONTROLADOR DE MYSQL NO SE ENCUENTRA INSTALADO.",
26                 "ERROR DE CONTROLADOR", JOptionPane.ERROR_MESSAGE);
27         } catch (SQLException e) {
28             JOptionPane.showMessageDialog(
29                 null, "NO SE PUDO ESTABLECER LA CONEXIÓN CON LA BASE DE DATOS",
30                 "ERROR EN LA CONEXIÓN", JOptionPane.ERROR_MESSAGE);
31             System.out.println(e);
32         } catch (IllegalAccessException | InstantiationException e) {
33             JOptionPane.showMessageDialog(
34                 null, ""+e.getMessage(),
35                 "ERROR!!!", JOptionPane.ERROR_MESSAGE);
36             System.exit(0);
37         }
38     }
39
40     public java.sql.Connection getConnection(){
41         return con;
42     }
43
44 }
```

Para la conexión se declara el controlador y después solo colocamos los datos que nos pide que son: nombre del servidor, usuario y contraseña de la base de datos, todos estos- datos ya nos la dan el hosting.

```
try {
    Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
    String url = "jdbc:mysql:// nombre del servidor host / nombre de la base de datos ?use"
    String user = " nombre del usuario de la base de datos ";
    String password = " contraseña de la base de datos ";
    con = DriverManager.getConnection(url, user, password);
    if(con != null){
        System.out.println("Conexion Establecida..");
    }
}
```

db_a61ee8_prueba

MYSQL: mysql5027.site4now.net

Servidor Host

Nombre de Usuario: a61ee8_prueba

Nombre del Usuario

Contraseña: ***** 

Contraseña de la base de datos

Código fuente

ConnectionSQL

```
1. package code;
2.
3. import java.sql.Connection;
4. import java.sql.DriverManager;
5. import java.sql.SQLException;
6. import javax.swing.JOptionPane;
7.
8. public class ConnectionSQL {
9.
10.     Connection con;
11.
12.     public ConnectionSQL(){
13.         con = null;
14.         try {
15.
16.             Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
17.             // HOSTING
18.             String url =
19.                 "jdbc:mysql://mysql5027.site4now.net/db_a6112a_caragyn?useTimezone=
20.                 true&serverTimezone=UTC";//obtenemos el url
21.             String user = "a6112a_caragyn";//obtenemos el usuario
22.             de la base a la que se va a conectar
```

```
19.         String password = "pass1234";//obtenemos la contraseña
    de la base a la que se va a conectar
20.
21.         con = DriverManager.getConnection(url, user,
    password);
22.         if(con != null){//indicamos que si todo el
    procedimiento es correcto nos imprima el siguiente mensaje
23.             System.out.println("Conexion Establecida..");
24.         }
25.
26.     } catch (ClassNotFoundException e) { //indicamos que si
    ocurre un error nos muestre el tipo de error
27.         JOptionPane.showMessageDialog(
28.             null, "EL CONTROLADOR DE MYSQL NO SE ENCUENTRA
    INSTALADO.",
29.             "ERROR DE CONTROLADOR",
    JOptionPane.ERROR_MESSAGE);
30.     } catch (SQLException e) {
31.         JOptionPane.showMessageDialog(
32.             null, "NO SE PUDO ESTABLECER LA CONEXIÓN CON
    LA BASE DE DATOS",
33.             "ERROR EN LA CONEXIÓN",
    JOptionPane.ERROR_MESSAGE);
34.         System.out.println(e);
35.     } catch (IllegalAccessException | InstantiationException
    e) {
36.         JOptionPane.showMessageDialog(
37.             null, ""+e.getMessage(),
38.             "ERROR!!!", JOptionPane.ERROR_MESSAGE);
39.         System.exit(0);
40.     }
41. }
42.
43.     public java.sql.Connection getConection(){//creamos un método
    para ver si la conexión sigue
44.         return con;
45.     }
46.
47.     public void setDisconnect(){//creamos un método para ver si la
    conexión sigue
48.         con = null;
49.         if(con == null){
50.             System.err.print("x");
51.         }
52.     }
53.
54.
55.
56. }
57. }
```

FILES

```
1.  * To change this license header, choose License Headers in Project
    Properties.
2.  * To change this template file, choose Tools | Templates
3.  * and open the template in the editor.
4.  */
5.  package code;
6.
7.  import com.itextpdf.text.BaseColor;
8.  import com.itextpdf.text.Document;
9.  import com.itextpdf.text.DocumentException;
10. import com.itextpdf.text.FontFactory;
11. import com.itextpdf.text.Paragraph;
12. import com.itextpdf.text.pdf.PdfPTable;
13. import com.itextpdf.text.pdf.PdfWriter;
14. import java.awt.Font;
15. import java.io.FileOutputStream;
16. import java.io.IOException;
17. import java.text.DateFormat;
18. import java.text.SimpleDateFormat;
19. import java.util.Date;
20. import java.util.Random;
21. import view.Login;
22. import view.user.Purchase;
23.
24. /**
25.  *
26.  * @author Restf
27.  */
28. public class Files {
29.     public void pdfCreateFile(String url, String price, String
        extra, String total, int value){
30.         String txtCompany =
        Purchase.selectCompany.getSelectedItemAt().toString();
31.         String companyPDF = txtCompany.substring(2);
32.         String txtState =
        Purchase.selectState.getSelectedItemAt().toString();
33.         String statePDF = txtState.substring(2);
34.         String txtModel =
        Purchase.selectModel.getSelectedItemAt().toString();
35.         String modelPDF = txtModel.substring(2);
36.         String txtSerial =
        Purchase.selectSerial.getSelectedItemAt().toString();
37.         String serialPDF = txtSerial.substring(2);
38.         String txtColors =
        Purchase.selectColors.getSelectedItemAt().toString();
39.         String colorsPDF = txtColors.substring(2);
40.         String txtColorsNew =
        Purchase.selectColorsNew.getSelectedItemAt().toString();
41.         String colorNewPDF = txtColorsNew.substring(2);
42.         String txtTypePayment =
        Purchase.selectTypePayment.getSelectedItemAt().toString();
43.         String typePDF = txtTypePayment.substring(2);
```



```

44.
45.         Date date = new Date();
46.         DateFormat format = new SimpleDateFormat("dd/MM/yyyy");
47.         PdfPTable table = new PdfPTable(2);
48.         float[] mediaCeldas ={3.30f,3.50f};
49.         PdfPTable tablePDF = new PdfPTable(1);
50.         float[] mediaCeldass ={3.30f};
51.
52.         try {
53.             tablePDF.setWidths(mediaCeldass);
54.         } catch (DocumentException ex) {
55.             System.out.println(ex);
56.         }
57.         tablePDF.addCell(new Paragraph("
DESCRIPCION      ",
FontFactory.getFont("Arial",20,Font.BOLD,BaseColor.BLUE)));
58.         try {
59.             table.setWidths(mediaCeldas);
60.         } catch (DocumentException ex) {
61.             System.out.println(ex);
62.         }
63.         table.addCell(new
Paragraph("COMPañIA"+ "\n\nESTADO"+ "\n\nMODELO"+ "\n\nNUMERO
SERIAL"+ "\n\nCOLOR", FontFactory.getFont("Arial",20)));
64.         table.addCell(new
Paragraph(companyPDF+ "\n\n"+statePDF+ "\n\n"+modelPDF+ "\n\n"+serialP
DF+ "\n\n"+colorsPDF, FontFactory.getFont("Arial",20)));
65.
66.         Random random = new Random();
67.         int f = random.nextInt(10000);
68.         int c = random.nextInt(1000);
69.         try{
70.             FileOutputStream archivo = new
FileOutputStream(url+".pdf");
71.             Document doc= new Document();
72.             PdfWriter.getInstance(doc, archivo);
73.             doc.open();
74.             doc.add(new Paragraph("CAR AGENCY
FACTURA",
FontFactory.getFont("Arial",25,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
75.             doc.add(new Paragraph("Un auto nuevo es un sueño nuevo
Fecha: "+format.format(date),
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
76.             doc.add(new Paragraph("
N° de Factura: "+f,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
77.             doc.add(new Paragraph("
N° de Cliente: "+c,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
78.             doc.add(new Paragraph("FACTURA A:
"+Login.userEntity.getName(),
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));

```

```

79.         doc.add(new Paragraph("Blvrd Toluca Metepec 300, Zona
Sin Asignación de Nombre de Col 1, 52140 Metepec, Méx.",
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
80.         doc.add(new Paragraph("7291307338",
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
81.         doc.add(new Paragraph("\n\n"));
82.         doc.add(tablePDF);
83.         doc.add(table);
84.         doc.add(new Paragraph("\n\n
TIPO DE PAGO: "+typePDF,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
85.         doc.add(new Paragraph("
PRECIO: $" + price,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
86.         doc.add(new Paragraph("
ADICIONES: $" + extra,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
87.         doc.add(new Paragraph("
TOTAL: $" + total,
FontFactory.getFont("Arial",12,Font.ROMAN_BASELINE,BaseColor.BLACK)
));
88.         doc.close();
89.     } catch (Exception e) {
90.         System.out.println("Error: " + e);
91.     }
92.
93.     ProcessBuilder p= new ProcessBuilder();
94.     p.command("cmd.exe", "/c", url + ".pdf");
95.     try {
96.         p.start();
97.     } catch (IOException ex) {
98.         System.out.println(ex);
99.     }
100.
101. }
102.
103.
104. }

```

MethodsSQL

```

1.  /*
2.  * To change this license header, choose License Headers in Project
Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package code;
7.
8.  import com.itextpdf.text.BaseColor;

```

```

9. import com.itextpdf.text.Document;
10. import com.itextpdf.text.DocumentException;
11. import com.itextpdf.text.Element;
12. import com.itextpdf.text.FontFactory;
13. import com.itextpdf.text.PageSize;
14. import com.itextpdf.text.Paragraph;
15. import com.itextpdf.text.pdf.PdfPTable;
16. import com.itextpdf.text.pdf.PdfWriter;
17. import java.awt.Desktop;
18. import java.awt.Font;
19. import java.awt.HeadlessException;
20. import java.io.File;
21. import java.io.FileNotFoundException;
22. import java.io.FileOutputStream;
23. import java.io.IOException;
24. import java.io.OutputStream;
25. import java.sql.Connection;
26. import java.sql.PreparedStatement;
27. import java.sql.ResultSet;
28. import java.sql.SQLException;
29. import java.sql.Statement;
30. import javax.swing.DefaultComboBoxModel;
31. import javax.swing.JOptionPane;
32. import javax.swing.table.DefaultTableModel;
33.
34. /**
35.  *
36.  * @author Restf
37.  */
38. public class MethodsSQL {
39.
40.     private static Connection conn;
41.     private static ConnectionSQL con;
42.     private static PreparedStatement ps;
43.     private static ResultSet rs;
44.     MyIcon icon = new MyIcon();
45.
46.     public Connection setConnection(){
47.         conn = main.Main.conn.getConnection();
48.         return conn;
49.     }
50.
51.     //-----
    -----
52.     public UsersEntity Login(String user, String pass){
53.         UsersEntity ev = new UsersEntity();
54.         conn = setConnection();
55.         String sql = "SELECT * FROM users WHERE user=? and
pass=?";
56.         try {
57.             ps = conn.prepareStatement(sql);
58.             ps.setString(1, user);
59.             ps.setString(2, pass);
60.             rs = ps.executeQuery();
61.             while (rs.next()) {
62.                 ev.setState (rs.getInt(2));

```

```

63.         ev.setName    (rs.getString(3));
64.         ev.setEmail   (rs.getString(4));
65.         ev.setRfc      (rs.getString(5));
66.         ev.setUser     (rs.getString(6));
67.         ev.setPass     (rs.getString(7));
68.         ev.setAge      (rs.getString(8));
69.     }
70.     } catch (SQLException e) {
71.     }
72.     return ev;
73. }
74.
75. public UsersEntity RFC(String rfc){
76.     UsersEntity ev = new UsersEntity();
77.     conn = setConnection();
78.     String sql = "SELECT * FROM users WHERE rfc = ?";
79.     try {
80.         ps = conn.prepareStatement(sql);
81.         ps.setString(1, rfc);
82.         rs = ps.executeQuery();
83.         while (rs.next()) {
84.             ev.setId      (rs.getInt(1));
85.             ev.setState   (rs.getInt(2));
86.             ev.setName    (rs.getString(3));
87.             ev.setEmail   (rs.getString(4));
88.             ev.setRfc      (rs.getString(5));
89.             ev.setUser     (rs.getString(6));
90.             ev.setPass     (rs.getString(7));
91.             ev.setAge      (rs.getString(8));
92.         }
93.     } catch (SQLException e) {
94.     }
95.     return ev;
96. }
97. //-----
-----
98.     public boolean Register(int id, int state, String name, String
email, String rfc, String user, String pass, String age){
99.         boolean okey = false;
100.        try {
101.            conn = setConnection();
102.            String SQL = "INSERT INTO users (id, state, name,
email, rfc, user, pass, age) VALUES (?,?,?,?,?,?,?,?)";
103.            ps = conn.prepareStatement(SQL);
104.            ps.setInt(1, id);
105.            ps.setInt(2, state);
106.            ps.setString(3, name);
107.            ps.setString(4, email);
108.            ps.setString(5, rfc);
109.            ps.setString(6, user);
110.            ps.setString(7, pass);
111.            ps.setString(8, age);
112.            int i = ps.executeUpdate();
113.            if(i>0){
114.                okey = true;
115.                MyIcon icon = new MyIcon();

```

```

116.         JOptionPane.showMessageDialog(
117.             null, "Datos Guardados.",
118.             "", JOptionPane.DEFAULT_OPTION, icon);
119.     } else {
120.         okay = false;
121.         JOptionPane.showMessageDialog(
122.             null, "Error al guardar los datos.",
123.             "", JOptionPane.ERROR_MESSAGE);
124.     }
125. } catch (HeadlessException | SQLException e) {
126.     System.out.println("Error: "+e);
127. }
128. return okay;
129. }
130.
131. public void Purchase(int id, int folio, String name, String
    model, int serial_number, int price_car, String additional, int
    price_additional, int price_total, int type_payment){
132.     try {
133.         conn = setConnection();
134.         String SQL = "INSERT INTO sales (id, folio, name,
            model, serial_number, price_car, additional, price_additional,
            price_total, type_payment) VALUES (?,?,?,?,?,?,?,?,?,?)";
135.         ps = conn.prepareStatement(SQL);
136.         ps.setInt    (1, id);
137.         ps.setInt    (2, folio);
138.         ps.setString (3, name);
139.         ps.setString (4, model);
140.         ps.setInt    (5, serial_number);
141.         ps.setInt    (6, price_car);
142.         ps.setString (7, additional);
143.         ps.setInt    (8, price_additional);
144.         ps.setInt    (9, price_total);
145.         ps.setInt    (10, type_payment);
146.         int i = ps.executeUpdate();
147.         if(i>0){
148.             MyIcon icon = new MyIcon();
149.             JOptionPane.showMessageDialog(
150.                 null, "Compra Completada.",
151.                 "", JOptionPane.DEFAULT_OPTION, icon);
152.         } else {
153.             JOptionPane.showMessageDialog(
154.                 null, "Error al completar la compra.",
155.                 "", JOptionPane.ERROR_MESSAGE);
156.         }
157.     } catch (HeadlessException | SQLException e) {
158.         System.out.println("Error: "+e);
159.     }
160. }
161. //-----
    -----
162. public DefaultTableModel loadTableUser(){
163.     DefaultTableModel model;
164.     String com[] = {"ID", "ESTADO", "NOMBRE", "CORREO", "RFC",
        "USUARIO", "CONTRASEÑA", "EDAD" };
165.     String row[] = new String[8];

```

```

166.         model = new DefaultTableModel(null, com);
167.         try {
168.             conn = setConnection();
169.             String SQL = "SELECT * FROM users;";
170.             ps = conn.prepareStatement(SQL);
171.             rs = ps.executeQuery();
172.             while (rs.next()) {
173.                 row[0] = rs.getString(1);
174.                 String state = "";
175.                 switch(rs.getInt(2)){
176.                     case 0:
177.                         state = "USUARIO";
178.                         break;
179.                     case 1:
180.                         state = "ADMINISTRADOR";
181.                         break;
182.                 }
183.                 row[1] = state;
184.                 row[2] = rs.getString(3);
185.                 row[3] = rs.getString(4);
186.                 row[4] = rs.getString(5);
187.                 row[5] = rs.getString(6);
188.                 row[6] = rs.getString(7);
189.                 row[7] = rs.getString(8);
190.                 model.addRow(row);
191.             }
192.         } catch (Exception e) {
193.         }
194.         return model;
195.     }
196.
197.     public DefaultTableModel loadTableUserFilter(String rfc){
198.         DefaultTableModel model;
199.         String com[] = {"ID", "ESTADO", "NOMBRE", "CORREO", "RFC",
200.             "USUARIO", "CONTRASEÑA", "EDAD" };
201.         String row[] = new String[8];
202.         model = new DefaultTableModel(null, com);
203.         try {
204.             conn = setConnection();
205.             String SQL = "SELECT * FROM users WHERE rfc LIKE " +
206.                 "'" + rfc + "%" + "'";
207.             ps = conn.prepareStatement(SQL);
208.             rs = ps.executeQuery();
209.             while (rs.next()) {
210.                 row[0] = rs.getString(1);
211.                 String state = "";
212.                 switch(rs.getInt(2)){
213.                     case 0:
214.                         state = "USUARIO";
215.                         break;
216.                     case 1:
217.                         state = "ADMINISTRADOR";
218.                         break;
219.                 }
220.                 row[1] = state;
221.                 row[2] = rs.getString(3);
222.                 row[3] = rs.getString(4);

```

```

221.         row[4] = rs.getString(5);
222.         row[5] = rs.getString(6);
223.         row[6] = rs.getString(7);
224.         row[7] = rs.getString(8);
225.         model.addRow(row);
226.     }
227. } catch (Exception e) {
228. }
229. return model;
230. }
231.
232. public DefaultTableModel loadTableInventory(){
233.     DefaultTableModel model;
234.     String com[] = {"ID", "COMPAÑIA", "MODELO", "N° SERIE",
"COLOR", "ESTADO", "PRECIO", "PAIS", "HP", "N° REGISTRO",
"KILOMETRAJE" };
235.     String row[] = new String[11];
236.     model = new DefaultTableModel(null, com);
237.     try {
238.         conn = setConnection();
239.         String SQL = "SELECT * FROM car;";
240.         ps = conn.prepareStatement(SQL);
241.         rs = ps.executeQuery();
242.         while (rs.next()) {
243.             row[0] = rs.getString(1);
244.             row[1] = rs.getString(2);
245.             row[2] = rs.getString(3);
246.             row[3] = rs.getString(4);
247.             String color = "";
248.             switch(rs.getInt(5)){
249.                 case 1:
250.                     color = "1 BLANCO";
251.                     break;
252.                 case 2:
253.                     color = "2 ROJO";
254.                     break;
255.                 case 3:
256.                     color = "3 NARANJA";
257.                     break;
258.                 case 4:
259.                     color = "4 AMARILLO";
260.                     break;
261.                 case 5:
262.                     color = "5 VERDE";
263.                     break;
264.                 case 6:
265.                     color = "6 AZUL";
266.                     break;
267.                 case 7:
268.                     color = "7 GRIS";
269.                     break;
270.             }
271.             String state = "";
272.             switch(rs.getInt(6)){
273.                 case 0:
274.                     state = "0 NUEVO";
275.                     break;

```

```

276.             case 1:
277.                 state = "1 SEMINUEVO";
278.                 break;
279.             case 2:
280.                 state = "2 USADO";
281.                 break;
282.             }
283.             row[4] = color;
284.             row[5] = state;
285.             row[6] = rs.getString(7);
286.             row[7] = rs.getString(8);
287.             row[8] = rs.getString(9);
288.             row[9] = rs.getString(10);
289.             row[10] = rs.getString(11);
290.             model.addRow(row);
291.         }
292.     } catch (Exception e) {
293.     }
294.     return model;
295. }
296.
297. public DefaultTableModel loadTableInventoryFilter(int select,
String value){
298.     String txtSql = null;
299.     switch(select){
300.         case 1:
301.             txtSql = "SELECT * FROM car WHERE id LIKE " + "'"
+ value + "%" + "'";
302.             break;
303.         case 2:
304.             txtSql = "SELECT * FROM car WHERE company LIKE " +
"'" + value + "%" + "'";
305.             break;
306.         case 3:
307.             txtSql = "SELECT * FROM car WHERE model LIKE " +
"'" + value + "%" + "'";
308.             break;
309.         case 4:
310.             txtSql = "SELECT * FROM car WHERE serial_number
LIKE " + "'" + value + "%" + "'";
311.             break;
312.         case 5:
313.             txtSql = "SELECT * FROM car WHERE color LIKE " +
"'" + value + "%" + "'";
314.             break;
315.         case 6:
316.             txtSql = "SELECT * FROM car WHERE state LIKE " +
"'" + value + "%" + "'";
317.             break;
318.         case 7:
319.             txtSql = "SELECT * FROM car WHERE price LIKE " +
"'" + value + "%" + "'";
320.             break;
321.         case 8:
322.             txtSql = "SELECT * FROM car WHERE country LIKE " +
"'" + value + "%" + "'";
323.             break;

```



```

324.             case 9:
325.                 txtSql = "SELECT * FROM car WHERE horsepower LIKE
" + "'" + value + "%" + "'";
326.                 break;
327.             case 10:
328.                 txtSql = "SELECT * FROM car WHERE car_registration
LIKE " + "'" + value + "%" + "'";
329.                 break;
330.             case 11:
331.                 txtSql = "SELECT * FROM car WHERE mileage LIKE " +
"'" + value + "%" + "'";
332.                 break;
333.         }
334.         DefaultTableModel model;
335.         String com[] = {"ID", "COMPAÑIA", "MODELO", "N° SERIE",
"COLOR", "ESTADO", "PRECIO", "PAIS", "HP", "N° REGISTRO",
"KILOMETRAJE" };
336.         String row[] = new String[11];
337.         model = new DefaultTableModel(null, com);
338.         try {
339.             conn = setConnection();
340.             String SQL = txtSql;
341.             ps = conn.prepareStatement(SQL);
342.             rs = ps.executeQuery();
343.             while (rs.next()) {
344.                 row[0] = rs.getString(1);
345.                 row[1] = rs.getString(2);
346.                 row[2] = rs.getString(3);
347.                 row[3] = rs.getString(4);
348.                 String color = "";
349.                 switch(rs.getInt(5)){
350.                     case 1:
351.                         color = "1 BLANCO";
352.                         break;
353.                     case 2:
354.                         color = "2 ROJO";
355.                         break;
356.                     case 3:
357.                         color = "3 NARANJA";
358.                         break;
359.                     case 4:
360.                         color = "4 AMARILLO";
361.                         break;
362.                     case 5:
363.                         color = "5 VERDE";
364.                         break;
365.                     case 6:
366.                         color = "6 AZUL";
367.                         break;
368.                     case 7:
369.                         color = "7 GRIS";
370.                         break;
371.                 }
372.                 String state = "";
373.                 switch(rs.getInt(6)){
374.                     case 0:
375.                         state = "0 NUEVO";

```

```

376.             break;
377.         case 1:
378.             state = "1 SEMINUEVO";
379.             break;
380.         case 2:
381.             state = "2 USADO";
382.             break;
383.     }
384.     row[4] = color;
385.     row[5] = state;
386.     row[6] = rs.getString(7);
387.     row[7] = rs.getString(8);
388.     row[8] = rs.getString(9);
389.     row[9] = rs.getString(10);
390.     row[10] = rs.getString(11);
391.     model.addRow(row);
392. }
393. } catch (Exception e) {
394. }
395. return model;
396. }
397.
398. public DefaultTableModel loadTableSales(){
399.     DefaultTableModel model;
400.     String com[] = {"ID", "FOLIO", "NOMBRE", "MODELO", "N°
SERIE", "PRECIO VEHICULO", "ADICIONALES", "PRECIO ADICIONAL",
"PRECIO TOTAL", "TIPO DE PAGO"};
401.     String row[] = new String[10];
402.     model = new DefaultTableModel(null, com);
403.     try {
404.         conn = setConnection();
405.         String SQL = "SELECT * FROM sales;";
406.         ps = conn.prepareStatement(SQL);
407.         rs = ps.executeQuery();
408.         while (rs.next()) {
409.             row[0] = rs.getString(1);
410.             row[1] = rs.getString(2);
411.             row[2] = rs.getString(3);
412.             row[3] = rs.getString(4);
413.             row[4] = rs.getString(5);
414.             row[5] = rs.getString(6);
415.             if(rs.getString(7) == null){
416.                 row[6] = "-----";
417.                 row[7] = "-----"; } else {
418.                 row[6] = rs.getString(7);
419.                 row[7] = rs.getString(8); }
420.             row[8] = rs.getString(9);
421.             if(rs.getInt(10) == 1){ row[9] = "TARJETA
CREDITO"; }
422.             if(rs.getInt(10) == 2){ row[9] = "PAYPAL"; }
423.             model.addRow(row);
424.         }
425.     } catch (Exception e) {
426.     }
427.     return model;
428. }
429.

```

```

430.     public DefaultTableModel loadTableSalesFilter(int select,
String value){
431.         String txtSql = null;
432.         switch(select){
433.             case 1:
434.                 txtSql = "SELECT * FROM sales WHERE id LIKE " +
"" + value + "%" + "'";
435.                 break;
436.             case 2:
437.                 txtSql = "SELECT * FROM sales WHERE folio LIKE " +
"" + value + "%" + "'";
438.                 break;
439.             case 3:
440.                 txtSql = "SELECT * FROM sales WHERE name LIKE " +
"" + value + "%" + "'";
441.                 break;
442.             case 4:
443.                 txtSql = "SELECT * FROM sales WHERE model LIKE " +
"" + value + "%" + "'";
444.                 break;
445.             case 5:
446.                 txtSql = "SELECT * FROM sales WHERE serial_number
LIKE " + "" + value + "%" + "'";
447.                 break;
448.             case 6:
449.                 txtSql = "SELECT * FROM sales WHERE price_car LIKE
" + "" + value + "%" + "'";
450.                 break;
451.             case 7:
452.                 txtSql = "SELECT * FROM sales WHERE additional
LIKE " + "" + value + "%" + "'";
453.                 break;
454.             case 8:
455.                 txtSql = "SELECT * FROM sales WHERE
price_additional LIKE " + "" + value + "%" + "'";
456.                 break;
457.             case 9:
458.                 txtSql = "SELECT * FROM sales WHERE price_total
LIKE " + "" + value + "%" + "'";
459.                 break;
460.             case 10:
461.                 txtSql = "SELECT * FROM sales WHERE type_payment
LIKE " + "" + value + "%" + "'";
462.                 break;
463.             }
464.             DefaultTableModel model;
465.             String com[] = {"ID", "FOLIO", "NOMBRE", "MODELO", "N°
SERIE", "PRECIO VEHICULO", "ADICIONALES", "PRECIO ADICIONAL",
"PRECIO TOTAL", "TIPO DE PAGO"};
466.             String row[] = new String[10];
467.             model = new DefaultTableModel(null, com);
468.             try {
469.                 conn = setConnection();
470.                 String SQL = txtSql;
471.                 ps = conn.prepareStatement(SQL);
472.                 rs = ps.executeQuery();
473.                 while (rs.next()) {

```

```

474.         row[0] = rs.getString(1);
475.         row[1] = rs.getString(2);
476.         row[2] = rs.getString(3);
477.         row[3] = rs.getString(4);
478.         row[4] = rs.getString(5);
479.         row[5] = rs.getString(6);
480.         if(rs.getString(7) == null){
481.             row[6] = "-----";
482.             row[7] = "-----"; } else {
483.             row[6] = rs.getString(7);
484.             row[7] = rs.getString(8); }
485.         row[8] = rs.getString(9);
486.         if(rs.getInt(10) == 1){ row[9] = "TARJETA
CREDITO"; }
487.         if(rs.getInt(10) == 2){ row[9] = "PAYPAL"; }
488.         model.addRow(row);
489.     }
490. } catch (Exception e) {
491. }
492. return model;
493. }
494.
495. //-----
-----

496. public boolean ValidateUser(String user){
497.     boolean okey = false;
498.     try {
499.         conn = setConnection();
500.         String SQL = "SELECT user FROM users WHERE user = ?";
501.         ps = conn.prepareStatement(SQL);
502.         ps.setString(1, user);
503.         rs = ps.executeQuery();
504.         if(rs.next()){ okey = true; }
505.     } catch (SQLException e) {
506.         System.out.println("Error: "+e);
507.     }
508.     return okey;
509. }

510.
511. public boolean ValidateRFC(String rfc){
512.     boolean okey = false;
513.     try {
514.         conn = setConnection();
515.         String SQL = "SELECT rfc FROM users WHERE rfc = ?";
516.         ps = conn.prepareStatement(SQL);
517.         ps.setString(1, rfc);
518.         rs = ps.executeQuery();
519.         if(rs.next()){ okey = true; }
520.     } catch (SQLException e) {
521.         System.out.println("Error: "+e);
522.     }
523.     return okey;
524. }

525.
526. public boolean ValidateRFCicon(String rfc){
527.     boolean okey = false;

```

```

528.         try {
529.             conn = setConnection();
530.             String SQL = "SELECT * FROM users WHERE rfc LIKE " +
                "'" + rfc + "%" + "'";
531.             ps = conn.prepareStatement(SQL);
532.             rs = ps.executeQuery();
533.             if(rs.next()){ okey = true; }
534.         } catch (SQLException e) {
535.             System.out.println("Error: "+e);
536.         }
537.         return okey;
538.     }
539.
540.     public DefaultComboBoxModel loadModels(int selectCompany, int
        selectState){
541.         DefaultComboBoxModel model = new DefaultComboBoxModel();
542.         model.addElement("Seleccionar");
543.         try {
544.             conn = setConnection();
545.             String SQL = "SELECT model FROM car WHERE company = ?
                and state = ?";
546.             ps = conn.prepareStatement(SQL);
547.             String company = null;
548.             switch(selectCompany){
549.                 case 1:
550.                     company = "NISSAN";
551.                     break;
552.                 case 2:
553.                     company = "AUDI";
554.                     break;
555.                 case 3:
556.                     company = "VOLKSWAGEN";
557.                     break;
558.             }
559.             ps.setString(1, company);
560.             ps.setInt(2, selectState);
561.             rs = ps.executeQuery();
562.             while (rs.next()) {
563.                 model.addElement("• "+rs.getString(1));
564.             }
565.         } catch (Exception e) {
566.         }
567.         return model;
568.     }
569.
570.     public DefaultComboBoxModel loadSerial(String selectModel){
571.         DefaultComboBoxModel model = new DefaultComboBoxModel();
572.         model.addElement("Seleccionar");
573.         try {
574.             conn = setConnection();
575.             String SQL = "SELECT serial_number FROM car WHERE
                model = ?";
576.             ps = conn.prepareStatement(SQL);
577.             ps.setString(1, selectModel);
578.             rs = ps.executeQuery();
579.             while (rs.next()) {
580.                 model.addElement("• "+rs.getString(1));

```

```

581.         }
582.     } catch (Exception e) {
583.     }
584.     return model;
585. }
586.
587.     public DefaultComboBoxModel loadColorsQuotation(int
selectSerial){
588.         DefaultComboBoxModel model = new DefaultComboBoxModel();
589.         model.addElement("Seleccionar");
590.         try {
591.             conn = setConnection();
592.             String SQL = "SELECT color FROM car WHERE
serial_number = ?";
593.             ps = conn.prepareStatement(SQL);
594.             ps.setInt(1, selectSerial);
595.             rs = ps.executeQuery();
596.             while (rs.next()) {
597.                 String color = null;
598.                 switch(rs.getInt(1)){
599.                     case 1:
600.                         color = "BLANCO";
601.                         break;
602.                     case 2:
603.                         color = "ROJO";
604.                         break;
605.                     case 3:
606.                         color = "NARANJA";
607.                         break;
608.                     case 4:
609.                         color = "AMARILLO";
610.                         break;
611.                     case 5:
612.                         color = "VERDE";
613.                         break;
614.                     case 6:
615.                         color = "AZUL";
616.                         break;
617.                     case 7:
618.                         color = "GRIS";
619.                         break;
620.                 }
621.                 model.addElement("• "+color);
622.             }
623.             model.addElement("• Otro: ");
624.         } catch (Exception e) {
625.         }
626.         return model;
627.     }
628.
629.     public DefaultComboBoxModel loadColorsVehicle(String
selectModel){
630.         DefaultComboBoxModel model = new DefaultComboBoxModel();
631.         model.addElement("Seleccionar");
632.         try {
633.             conn = setConnection();
634.             String SQL = "SELECT color FROM car WHERE model = ?";

```

```

635.         ps = conn.prepareStatement(SQL);
636.         ps.setString(1, selectModel);
637.         rs = ps.executeQuery();
638.         while (rs.next()) {
639.             String color = null;
640.             switch(rs.getInt(1)){
641.                 case 1:
642.                     color = "BLANCO";
643.                     break;
644.                 case 2:
645.                     color = "ROJO";
646.                     break;
647.                 case 3:
648.                     color = "NARANJA";
649.                     break;
650.                 case 4:
651.                     color = "AMARILLO";
652.                     break;
653.                 case 5:
654.                     color = "VERDE";
655.                     break;
656.                 case 6:
657.                     color = "AZUL";
658.                     break;
659.                 case 7:
660.                     color = "GRIS";
661.                     break;
662.             }
663.             model.addElement("• "+color);
664.         }
665.     } catch (Exception e) {
666.     }
667.     return model;
668. }
669.
670. public int loadPriceCar(int serial_number){
671.     int txt = 0;
672.     try {
673.         conn = setConnection();
674.         String SQL = "SELECT price FROM car WHERE
serial_number = ?";
675.         ps = conn.prepareStatement(SQL);
676.         ps.setInt(1, serial_number);
677.         rs = ps.executeQuery();
678.         if(rs.next()){
679.             txt = rs.getInt(1);
680.         }
681.     } catch (SQLException e) {
682.     }
683.     return txt;
684. }
685. //-----
-----
686. public void DeleteUserAdmin(String rfc){
687.     try {
688.         conn = setConnection();

```

```

689.         String SQL = "DELETE FROM users WHERE rfc = ? ";
690.         ps = conn.prepareStatement(SQL);
691.         ps.setString(1, rfc);
692.         int i = ps.executeUpdate();
693.         if (i > 0) {
694.             JOptionPane.showMessageDialog(
695.                 null, "Usuario eliminado correctamente.",
696.                 "", JOptionPane.DEFAULT_OPTION, icon);
697.         } else {
698.             JOptionPane.showMessageDialog(
699.                 null, "Error al eliminar el usuario.",
700.                 "", JOptionPane.ERROR_MESSAGE);
701.         }
702.     } catch (HeadlessException | SQLException e) {
703.         System.out.println("Error: "+e);
704.     }
705. }
706.
707. public void DeleteCarPurchase(int serial_number){
708.     try {
709.         conn = setConnection();
710.         String SQL = "DELETE FROM car WHERE serial_number = ?
711. ";
712.         ps = conn.prepareStatement(SQL);
713.         ps.setInt(1, serial_number);
714.         int i = ps.executeUpdate();
715.     } catch (HeadlessException | SQLException e) {
716.         System.out.println("Error: "+e);
717.     }
718. //-----
719. -----
720.     public void ModifyUserData(int id, String name, String email,
721. String rfc, String user, String pass, String age){
722.         try {
723.             conn = setConnection();
724.             String SQL = "UPDATE users SET name = ?, email = ?,
725. rfc = ?, user = ?, pass = ?, age = ? WHERE id = ?";
726.             ps = conn.prepareStatement(SQL);
727.             ps.setString(1, name);
728.             ps.setString(2, email);
729.             ps.setString(3, rfc);
730.             ps.setString(4, user);
731.             ps.setString(5, pass);
732.             ps.setString(6, age);
733.             ps.setInt(7, id);
734.             int i = ps.executeUpdate();
735.             if(i>0){
736.                 JOptionPane.showMessageDialog(
737.                     null, "Datos Guardados.",
738.                     "", JOptionPane.DEFAULT_OPTION, icon);
739.             } else {
740.                 JOptionPane.showMessageDialog(
741.                     null, "Error al guardar los datos.",
742.                     "", JOptionPane.ERROR_MESSAGE);
743.             }
744.         }

```



```

741.         } catch (HeadlessException | SQLException e) {
742.             System.out.println("Error: "+e);
743.         }
744.     }
745.
746.     public void ModifyUserDataAdmin(String rfc,int state, String
name, String email, String rfcNew, String user, String pass, String
age){
747.         try {
748.             conn = setConnection();
749.             String SQL = "UPDATE users SET state = ?, name = ?,
email = ?, rfc = ?, user = ?, pass = ?, age = ? WHERE rfc = ?";
750.             ps = conn.prepareStatement(SQL);
751.             ps.setInt(1, state);
752.             ps.setString(2, name);
753.             ps.setString(3, email);
754.             ps.setString(4, rfcNew);
755.             ps.setString(5, user);
756.             ps.setString(6, pass);
757.             ps.setString(7, age);
758.             ps.setString(8, rfc);
759.             int i = ps.executeUpdate();
760.             if(i>0){
761.                 JOptionPane.showMessageDialog(
762.                     null, "Datos Guardados.",
763.                     "", JOptionPane.DEFAULT_OPTION, icon);
764.             } else {
765.                 JOptionPane.showMessageDialog(
766.                     null, "Error al guardar los datos.",
767.                     "", JOptionPane.ERROR_MESSAGE);
768.             }
769.         } catch (HeadlessException | SQLException e) {
770.             System.out.println("Error: "+e);
771.         }
772.     }
773.
774.     public void ModifyUserDataAdminFilter(int select, String
value){
775.         String txtSql = null;
776.         switch(select){
777.             case 1:
778.                 txtSql = "UPDATE users SET state = ? WHERE rfc =
?";
779.                 break;
780.             case 2:
781.                 txtSql = "UPDATE users SET name = ? WHERE rfc =
?";
782.                 break;
783.             case 3:
784.                 txtSql = "UPDATE users SET email = ? WHERE rfc =
?";
785.                 break;
786.             case 4:
787.                 txtSql = "UPDATE users SET rfc = ? WHERE id = ?";
788.                 break;
789.             case 5:

```

```

790.             txtSql = "UPDATE users SET user = ? WHERE id =
?";
791.             break;
792.         case 6:
793.             txtSql = "UPDATE users SET pass = ? WHERE id =
?";
794.             break;
795.         case 7:
796.             txtSql = "UPDATE users SET age = ? WHERE id = ?";
797.             break;
798.     }
799.     try {
800.         conn = setConnection();
801.         String SQL = txtSql;
802.         ps = conn.prepareStatement(SQL);
803.         ps.setString(1, value);
804.         int i = ps.executeUpdate();
805.         if(i>0){
806.             JOptionPane.showMessageDialog(
807.                 null, "Datos Guardados.",
808.                 "", JOptionPane.DEFAULT_OPTION, icon);
809.         } else {
810.             JOptionPane.showMessageDialog(
811.                 null, "Datos Guardados.",
812.                 "", JOptionPane.ERROR_MESSAGE);
813.         }
814.     } catch (HeadlessException | SQLException e) {
815.         System.out.println("Error: "+e);
816.     }
817. }
818.
819. //-----
-----
820. //sergio
821.
822. public void pdfCreateTableSales(String url, int select, String
value) {
823.     String txtSql, filter;
824.     switch(select){
825.         case 1:
826.             txtSql = "SELECT * FROM sales WHERE id LIKE " +
            "'" + value + "%" + "'";
827.             filter = "ID";
828.             break;
829.         case 2:
830.             txtSql = "SELECT * FROM sales WHERE folio LIKE " +
            "'" + value + "%" + "'";
831.             filter = "FOLIO";
832.             break;
833.         case 3:
834.             txtSql = "SELECT * FROM sales WHERE name LIKE " +
            "'" + value + "%" + "'";
835.             filter = "NOMBRE";
836.             break;
837.         case 4:

```

```

838.         txtSql = "SELECT * FROM sales WHERE model LIKE " +
            "'" + value + "%" + "'";
839.         filter = "MODELO";
840.         break;
841.         case 5:
842.             txtSql = "SELECT * FROM sales WHERE serial_number
LIKE " + "'" + value + "%" + "'";
843.             filter = "N° SERIE";
844.             break;
845.         case 6:
846.             txtSql = "SELECT * FROM sales WHERE price_car LIKE
" + "'" + value + "%" + "'";
847.             filter = "PRECIO";
848.             break;
849.         case 7:
850.             txtSql = "SELECT * FROM sales WHERE additional
LIKE " + "'" + value + "%" + "'";
851.             filter = "EXTRAS";
852.             break;
853.         case 8:
854.             txtSql = "SELECT * FROM sales WHERE
price_additional LIKE " + "'" + value + "%" + "'";
855.             filter = "PRECIO EXTRA";
856.             break;
857.         case 9:
858.             txtSql = "SELECT * FROM sales WHERE price_total
LIKE " + "'" + value + "%" + "'";
859.             filter = "PRECIO TOTAL";
860.             break;
861.         case 10:
862.             txtSql = "SELECT * FROM sales WHERE type_payment
LIKE " + "'" + value + "%" + "'";
863.             filter = "SIN FILTRO";
864.             break;
865.         default:
866.             txtSql = "SELECT * FROM sales ";
867.             filter = "";
868.             break;
869.     }
870.
871.     try{
872.         conn = setConnection();
873.         Statement ps = conn.createStatement();
874.         ResultSet rs = ps.executeQuery(txtSql);
875.
876.         OutputStream file = null;
877.         try {
878.             file = new FileOutputStream(new
File(url+".pdf"));
879.         } catch (FileNotFoundException ex) {
880.             System.out.println(ex);
881.         }
882.
883.         Document document = new Document();
884.         PdfWriter.getInstance(document, file);
885.
886.         document.open();

```

```

887.         PdfPTable table = new PdfPTable(9);
888.         Paragraph p = new Paragraph("LISTA DE VENTAS DE
PRODUCTOS (" + filter + ") \n\n",
FontFactory.getFont("Arial", 16, Font.ITALIC, BaseColor.BLUE));
889.
890.         p.setAlignment(Element.ALIGN_CENTER);
891.         document.add(p);
892.         document.add(new Paragraph(""));
893.
894.         float[] size
={3.30f, 3.50f, 3.50f, 3.70f, 3.70f, 3.50f, 3.70f, 3.70f, 3.50f};
895.
896.         table.setWidths(size);
897.         table.addCell(new Paragraph("ID",
FontFactory.getFont("Arial", 12)));
898.         table.addCell(new Paragraph("FOLIO",
FontFactory.getFont("Arial", 12)));
899.         table.addCell(new Paragraph("MODELO",
FontFactory.getFont("Arial", 12)));
900.         table.addCell(new Paragraph("N° SERIAL",
FontFactory.getFont("Arial", 12)));
901.         table.addCell(new Paragraph("PRECIO",
FontFactory.getFont("Arial", 12)));
902.         table.addCell(new Paragraph("EXTRAS",
FontFactory.getFont("Arial", 12)));
903.         table.addCell(new Paragraph("PRECIO EXTRA",
FontFactory.getFont("Arial", 12)));
904.         table.addCell(new Paragraph("TOTAL",
FontFactory.getFont("Arial", 12)));
905.         table.addCell(new Paragraph("TIPO DE PAGO",
FontFactory.getFont("Arial", 12)));
906.
907.         while (rs.next()) {
908.             table.addCell(new Paragraph(rs.getString("id"),
FontFactory.getFont("Arial", 10)));
909.             table.addCell(new Paragraph(rs.getString("folio"),
FontFactory.getFont("Arial", 10)));
910.             table.addCell(new Paragraph(rs.getString("model"),
FontFactory.getFont("Arial", 10)));
911.             table.addCell(new
Paragraph(rs.getString("serial_number"),
FontFactory.getFont("Arial", 10)));
912.             table.addCell(new
Paragraph(rs.getString("price_car"),
FontFactory.getFont("Arial", 10)));
913.             table.addCell(new
Paragraph(rs.getString("additional"),
FontFactory.getFont("Arial", 10)));
914.             table.addCell(new
Paragraph(rs.getString("price_additional"),
FontFactory.getFont("Arial", 10)));
915.             table.addCell(new
Paragraph(rs.getString("price_total"),
FontFactory.getFont("Arial", 10)));
916.             table.addCell(new
Paragraph(rs.getString("type_payment"),
FontFactory.getFont("Arial", 10)));

```

```

917.         }
918.
919.         document.add(table);
920.         document.close();
921.         file.close();
922.
923.     } catch (SQLException e) {
924.         System.out.println(e);
925.     } catch (DocumentException ex) {
926.         System.out.println(ex);
927.     } catch (IOException ex) {
928.         System.out.println(ex);
929.     }
930.
931.     try{
932.         File file = new File(url+".pdf");
933.         Desktop.getDesktop().open(file);
934.     } catch (IOException e) {
935.         System.out.println(e);
936.     }
937.
938. }
939.
940.
941.     public void pdfCreateTableInventory(String url, int select,
String value) {
942.         String txtSql, filter;
943.         switch(select){
944.             case 1:
945.                 txtSql = "SELECT * FROM car WHERE id LIKE " + "'"
+ value + "%" + "'";
946.                 filter = "ID";
947.                 break;
948.             case 2:
949.                 txtSql = "SELECT * FROM car WHERE company LIKE " +
"'" + value + "%" + "'";
950.                 filter = "COMPAÑIA";
951.                 break;
952.             case 3:
953.                 txtSql = "SELECT * FROM car WHERE model LIKE " +
"'" + value + "%" + "'";
954.                 filter = "MODELO";
955.                 break;
956.             case 4:
957.                 txtSql = "SELECT * FROM car WHERE serial_number
LIKE " + "'" + value + "%" + "'";
958.                 filter = "N° SERIE";
959.                 break;
960.             case 5:
961.                 txtSql = "SELECT * FROM car WHERE color LIKE " +
"'" + value + "%" + "'";
962.                 filter = "COLOR";
963.                 break;
964.             case 6:
965.                 txtSql = "SELECT * FROM car WHERE state LIKE " +
"'" + value + "%" + "'";
966.                 filter = "ESTADO";

```

```

967.             break;
968.         case 7:
969.             txtSql = "SELECT * FROM car WHERE price LIKE " +
970.                 "'" + value + "%" + "'";
971.             filter = "PRECIO";
972.             break;
973.         case 8:
974.             txtSql = "SELECT * FROM car WHERE country LIKE " +
975.                 "'" + value + "%" + "'";
976.             filter = "PAIS";
977.             break;
978.         case 9:
979.             txtSql = "SELECT * FROM car WHERE horsepower LIKE
980.                 " + "'" + value + "%" + "'";
981.             filter = "HP";
982.             break;
983.         case 10:
984.             txtSql = "SELECT * FROM car WHERE car_registration
985.                 LIKE " + "'" + value + "%" + "'";
986.             filter = "PLACAS";
987.             break;
988.         case 11:
989.             txtSql = "SELECT * FROM car WHERE mileage LIKE " +
990.                 "'" + value + "%" + "'";
991.             filter = "KILOMETRAJE";
992.             break;
993.         default:
994.             txtSql = "SELECT * FROM car ";
995.             filter = "SIN FILTRO";
996.             break;
997.     }
998.     try{
999.         conn = setConnection();
1000.        Statement ps = conn.createStatement();
1001.        ResultSet rs = ps.executeQuery(txtSql);
1002.
1003.        OutputStream file = null;
1004.        try {
1005.            file = new FileOutputStream(new
1006.                File(url+".pdf"));
1007.        } catch (FileNotFoundException ex) {
1008.            System.out.println(ex);
1009.        }
1010.
1011.        Document document = new
1012.            Document(PageSize.A4.rotate());
1013.        PdfWriter.getInstance(document, file);
1014.
1015.        document.open();
1016.        PdfPTable table = new PdfPTable(11);
1017.        Paragraph p = new Paragraph("LISTA DE
1018.            INVENTARIO DE VEHICULOS (" + filter + ")\n\n",
1019.                FontFactory.getFont("Arial", 16, Font.ITALIC, BaseColor.BLUE));
1020.
1021.        p.setAlignment(Element.ALIGN_CENTER);
1022.        document.add(p);
1023.
1024.    }

```

```

1015.                document.add(new Paragraph(""));
1016.
1017.                float[] size
1018.                = {3.30f, 3.50f, 3.50f, 3.70f, 3.70f, 3.50f, 3.70f, 3.70f, 3.50f, 3.70f, 3.50f
1019.                };
1020.                table.setWidths(size);
1021.                table.addCell(new Paragraph("ID",
1022.                FontFactory.getFont("Arial", 12)));
1023.                table.addCell(new Paragraph("COMPANIA",
1024.                FontFactory.getFont("Arial", 12)));
1025.                table.addCell(new Paragraph("MODELO",
1026.                FontFactory.getFont("Arial", 12)));
1027.                table.addCell(new Paragraph("N° SERIE",
1028.                FontFactory.getFont("Arial", 12)));
1029.                table.addCell(new Paragraph("COLOR",
1030.                FontFactory.getFont("Arial", 12)));
1031.                table.addCell(new Paragraph("ESTADO",
1032.                FontFactory.getFont("Arial", 12)));
1033.                table.addCell(new Paragraph("PRECIO",
1034.                FontFactory.getFont("Arial", 12)));
1035.                table.addCell(new Paragraph("PAIS",
1036.                FontFactory.getFont("Arial", 12)));
1037.                table.addCell(new Paragraph("HP",
1038.                FontFactory.getFont("Arial", 12)));
1039.                table.addCell(new Paragraph("PLACAS",
1040.                FontFactory.getFont("Arial", 12)));
1041.                table.addCell(new Paragraph("KILOMETRAJE",
1042.                FontFactory.getFont("Arial", 12)));
1043.
1044.                while (rs.next()) {
1045.                    table.addCell(new
1046.                    Paragraph(rs.getString("id"), FontFactory.getFont("Arial", 10)));
1047.                    table.addCell(new
1048.                    Paragraph(rs.getString("company"),
1049.                    FontFactory.getFont("Arial", 10)));
1050.                    table.addCell(new
1051.                    Paragraph(rs.getString("model"), FontFactory.getFont("Arial", 10)));
1052.                    table.addCell(new
1053.                    Paragraph(rs.getString("serial_number"),
1054.                    FontFactory.getFont("Arial", 10)));
1055.                    String color = "";
1056.                    switch (rs.getInt(5)) {
1057.                        case 1:
1058.                            color = "BLANCO";
1059.                            break;
1060.                        case 2:
1061.                            color = "ROJO";
1062.                            break;
1063.                        case 3:
1064.                            color = "NARANJA";
1065.                            break;
1066.                        case 4:
1067.                            color = "AMARILLO";
1068.                            break;
1069.                        case 5:
1070.                            color = "VERDE";
1071.                            break;

```

```

1053.                case 6:
1054.                    color = "AZUL";
1055.                    break;
1056.                case 7:
1057.                    color = "GRIS";
1058.                    break;
1059.            }
1060.            String state = "";
1061.            switch(rs.getInt(6)){
1062.                case 0:
1063.                    state = "NUEVO";
1064.                    break;
1065.                case 1:
1066.                    state = "SEMINUEVO";
1067.                    break;
1068.                case 2:
1069.                    state = "USADO";
1070.                    break;
1071.            }
1072.            table.addCell(new Paragraph(color,
1073.                FontFactory.getFont("Arial",10)));
1074.            table.addCell(new Paragraph(state,
1075.                FontFactory.getFont("Arial",10)));
1076.            table.addCell(new
1077.                Paragraph(rs.getString("price"), FontFactory.getFont("Arial",10)));
1078.            table.addCell(new
1079.                Paragraph(rs.getString("country"),
1080.                FontFactory.getFont("Arial",10)));
1081.            table.addCell(new
1082.                Paragraph(rs.getString("horsepower"),
1083.                FontFactory.getFont("Arial",10)));
1084.            table.addCell(new
1085.                Paragraph(rs.getString("car_registration"),
1086.                FontFactory.getFont("Arial",10)));
1087.            table.addCell(new
1088.                Paragraph(rs.getString("mileage"),
1089.                FontFactory.getFont("Arial",10)));
1090.        }
1091.
1092.        document.add(table);
1093.        document.close();
1094.        file.close();
1095.    } catch (SQLException e){
1096.        System.out.println(e);
1097.    } catch (DocumentException ex) {
1098.        System.out.println(ex);
1099.    } catch (IOException ex) {
1100.        System.out.println(ex);
1101.    }
1102.
1103.    try{
1104.        File file = new File(url+".pdf");
1105.        Desktop.getDesktop().open(file);
1106.    } catch (IOException e){
1107.        System.out.println(e);
1108.    }
1109.
1110.    }

```



```

1099.
1100.         public void pdfCreateTableUsers(String url) {
1101.             try{
1102.                 conn = setConnection();
1103.                 Statement ps = conn.createStatement();
1104.                 ResultSet rs = ps.executeQuery("SELECT * FROM
users;");
1105.
1106.                 OutputStream file = null;
1107.                 try {
1108.                     file = new FileOutputStream(new
File(url+".pdf"));
1109.                 } catch (FileNotFoundException ex) {
1110.                     System.out.println(ex);
1111.                 }
1112.
1113.                 Document document = new Document();
1114.                 PdfWriter.getInstance(document, file);
1115.
1116.                 document.open();
1117.                 PdfPTable tabla = new PdfPTable(8);
1118.                 Paragraph p = new Paragraph("LISTA DE USUARIOS
\n\n", FontFactory.getFont("Arial",16,Font.ITALIC,BaseColor.BLUE));
1119.
1120.                 p.setAlignment(Element.ALIGN_CENTER);
1121.                 document.add(p);
1122.
1123.                 document.add(new Paragraph(""));
1124.                 float[] size ={2.00f, 4.90f, 3.50f, 3.70f,
3.70f, 3.50f, 3.70f, 3.90f};
1125.
1126.                 tabla.setWidths(size);
1127.                 tabla.addCell(new Paragraph("id",
FontFactory.getFont("Arial",12)));
1128.                 tabla.addCell(new Paragraph("state",
FontFactory.getFont("Arial",12)));
1129.                 tabla.addCell(new Paragraph("name",
FontFactory.getFont("Arial",12)));
1130.                 tabla.addCell(new Paragraph("email",
FontFactory.getFont("Arial",12)));
1131.                 tabla.addCell(new Paragraph("rfc",
FontFactory.getFont("Arial",12)));
1132.                 tabla.addCell(new Paragraph("user",
FontFactory.getFont("Arial",12)));
1133.                 tabla.addCell(new Paragraph("pass",
FontFactory.getFont("Arial",12)));
1134.                 tabla.addCell(new Paragraph("age",
FontFactory.getFont("Arial",12)));
1135.
1136.                 while (rs.next()){
1137.                     tabla.addCell(new
Paragraph(rs.getString(1), FontFactory.getFont("Arial",10)));
1138.                     String state = "";
1139.                     switch(rs.getInt(2)){
1140.                         case 0:
1141.                             state = "usuario";
1142.                             break;

```

```

1143.                case 1:
1144.                    state = "administrador";
1145.                    break;
1146.                }
1147.                tabla.addCell(new Paragraph(state,
1148.                    FontFactory.getFont("Arial",10)));
1149.                tabla.addCell(new
1150.                    Paragraph(rs.getString("name"), FontFactory.getFont("Arial",10)));
1151.                tabla.addCell(new
1152.                    Paragraph(rs.getString("email"), FontFactory.getFont("Arial",10)));
1153.                tabla.addCell(new
1154.                    Paragraph(rs.getString("rfc"), FontFactory.getFont("Arial",10)));
1155.                tabla.addCell(new
1156.                    Paragraph(rs.getString("user"), FontFactory.getFont("Arial",10)));
1157.                tabla.addCell(new
1158.                    Paragraph(rs.getString("pass"), FontFactory.getFont("Arial",10)));
1159.                tabla.addCell(new
1160.                    Paragraph(rs.getString("age"), FontFactory.getFont("Arial",10)));
1161.            }
1162.            document.add(tabla);
1163.            document.close();
1164.            file.close();
1165.        } catch (SQLException e){
1166.            System.out.println(e);
1167.        } catch (DocumentException ex) {
1168.            System.out.println(ex);
1169.        } catch (IOException ex) {
1170.            System.out.println(ex);
1171.        }
1172.        try{
1173.            File file = new File(url+".pdf");
1174.            Desktop.getDesktop().open(file);
1175.        } catch (IOException e){
1176.            System.out.println(e);
1177.        }
1178.    }

```

Mylcon

```

1.  package code;
2.
3.  import java.awt.Component;
4.  import java.awt.Graphics;
5.  import java.awt.Image;
6.  import javax.swing.Icon;
7.  import javax.swing.ImageIcon;

```

```

8.
9. public class MyIcon implements Icon{
10.
11.     @Override
12.     public void paintIcon(Component c, Graphics g, int x, int y) {
13.         Image image = new
ImageIcon(getClass().getResource("/resource/ready-
on.png")).getImage();
14.         g.drawImage(image, x, y, c);
15.     }
16.
17.     @Override
18.     public int getIconWidth() {
19.         return 50;
20.     }
21.
22.     @Override
23.     public int getIconHeight() {
24.         return 50;
25.     }
26.
27. }

```

UsersEntity

```

1. package code;
2.
3. public class UsersEntity {
4.     private int id;
5.     private int state;
6.     private String name;
7.     private String email;
8.     private String rfc;
9.     private String user;
10.    private String pass;
11.    private String age;
12.
13.    public UsersEntity(){
14.
15.    }
16.
17.    public UsersEntity(int id, int state, String name, String
email, String rfc, String user, String pass){
18.        this.id = id;
19.        this.state = state;
20.        this.name = name;
21.        this.email = email;
22.        this.rfc = rfc;
23.        this.user = user;
24.        this.pass = pass;
25.    }
26.
27.    public int getId() {
28.        return id;
29.    }

```

```
30.
31.     public void setId(int id) {
32.         this.id = id;
33.     }
34.
35.     public int getState() {
36.         return state;
37.     }
38.
39.     public void setState(int state) {
40.         this.state = state;
41.     }
42.
43.     public String getName() {
44.         return name;
45.     }
46.
47.     public void setName(String name) {
48.         this.name = name;
49.     }
50.
51.     public String getEmail() {
52.         return email;
53.     }
54.
55.     public void setEmail(String email) {
56.         this.email = email;
57.     }
58.
59.     public String getRfc() {
60.         return rfc;
61.     }
62.
63.     public void setRfc(String rfc) {
64.         this.rfc = rfc;
65.     }
66.
67.     public String getUser() {
68.         return user;
69.     }
70.
71.     public void setUser(String user) {
72.         this.user = user;
73.     }
74.
75.     public String getPass() {
76.         return pass;
77.     }
78.
79.     public void setPass(String pass) {
80.         this.pass = pass;
81.     }
82.
83.     public String getAge() {
84.         return age;
85.     }
86.
```

```
87.     public void setAge(String age) {
88.         this.age = age;
89.     }
90.
91.
92.
93.
94.
95. }
```

Main

```
1. package main;
2.
3. import code.ConnectionSQL;
4. import java.sql.Connection;
5. import view.Menu;
6.
7. public class Main {
8.
9.     public static ConnectionSQL conn;
10.    public static Connection reg;
11.
12.
13.    public static void main(String[] args) {
14.        Menu.main(args);
15.        Main main = new Main();
16.    }
17.
18.    public Main(){
19.        conn = new ConnectionSQL();
20.        reg = conn.getConnection();
21.    }
22.
23. }
```

Login

```
1. package view;
2.
3. import code.MethodsSQL;
4. import code.UsersEntity;
5. import java.awt.Color;
6. import javax.swing.JOptionPane;
7. import view.admin.MenuAdmin;
8. import view.user.MenuUser;
9.
10. public class Login extends javax.swing.JFrame {
11.
12.     MethodsSQL methodsSql = new MethodsSQL();
13.     public static UsersEntity userEntity = new UsersEntity();
14.
15.     public Login() {
16.         initComponents();
```

```

17.         setTitle("Car Agency");
18.         setLocationRelativeTo(null);
19.     }
20.
21.
22.
23.     /**
24.      * This method is called from within the constructor to
        initialize the form.
25.      * WARNING: Do NOT modify this code. The content of this
        method is always
26.      * regenerated by the Form Editor.
27.      */
28.     @SuppressWarnings("unchecked")
29.     // <editor-fold defaultstate="collapsed" desc="Generated
        Code">
30.     private void initComponents() {
31.
32.         a1 = new javax.swing.JPanel();
33.         txtUser = new javax.swing.JTextField();
34.         txtPass = new javax.swing.JPasswordField();
35.         btnLogin = new javax.swing.JButton();
36.         jLabel1 = new javax.swing.JLabel();
37.         jLabel2 = new javax.swing.JLabel();
38.         jLabel3 = new javax.swing.JLabel();
39.         jLabel4 = new javax.swing.JLabel();
40.
41.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
            ;
42.         setResizable(false);
43.
44.         a1.setBackground(new java.awt.Color(128, 153, 155));
45.         a1.setPreferredSize(new java.awt.Dimension(400, 600));
46.
47.         txtUser.setBackground(new java.awt.Color(255, 255, 255));
48.         txtUser.setFont(new java.awt.Font("Dialog", 0, 18)); //
        NOI18N
49.         txtUser.setForeground(new java.awt.Color(0, 0, 0));
50.         txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
51.             public void keyReleased(java.awt.event.KeyEvent evt) {
52.                 txtUserKeyReleased(evt);
53.             }
54.             public void keyTyped(java.awt.event.KeyEvent evt) {
55.                 txtUserKeyTyped(evt);
56.             }
57.         });
58.
59.         txtPass.setBackground(new java.awt.Color(255, 255, 255));
60.         txtPass.setFont(new java.awt.Font("Dialog", 0, 18)); //
        NOI18N
61.         txtPass.setForeground(new java.awt.Color(0, 0, 0));
62.         txtPass.setToolTipText("");
63.         txtPass.addKeyListener(new java.awt.event.KeyAdapter() {
64.             public void keyReleased(java.awt.event.KeyEvent evt) {
65.                 txtPassKeyReleased(evt);
66.             }

```

```

67.         public void keyTyped(java.awt.event.KeyEvent evt) {
68.             txtPassKeyTyped(evt);
69.         }
70.     });
71.
72.     btnLogin.setBackground(new java.awt.Color(255, 255, 255));
73.     btnLogin.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
74.     btnLogin.setText("Conectar");
75.     btnLogin.setEnabled(false);
76.     btnLogin.setPreferredSize(new java.awt.Dimension(150,
60));
77.     btnLogin.addMouseListener(new
java.awt.event.MouseAdapter() {
78.         public void mouseClicked(java.awt.event.MouseEvent
evt) {
79.             btnLoginMouseClicked(evt);
80.         }
81.     });
82.     btnLogin.addActionListener(new
java.awt.event.ActionListener() {
83.         public void actionPerformed(java.awt.event.ActionEvent
evt) {
84.             btnLoginActionPerformed(evt);
85.         }
86.     });
87.
88.     jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/login.png")
)); // NOI18N
89.
90.     jLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
91.     jLabel2.setForeground(new java.awt.Color(0, 0, 0));
92.     jLabel2.setText("Contraseña");
93.
94.     jLabel3.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
95.     jLabel3.setForeground(new java.awt.Color(0, 0, 0));
96.     jLabel3.setText("Usuario");
97.
98.     jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
); // NOI18N
99.     jLabel4.addMouseListener(new java.awt.event.MouseAdapter()
{
100.         public void mouseClicked(java.awt.event.MouseEvent
evt) {
101.             jLabel4MouseClicked(evt);
102.         }
103.     });
104.
105.     javax.swing.GroupLayout alLayout = new
javax.swing.GroupLayout(al);
106.     al.setLayout(alLayout);
107.     alLayout.setHorizontalGroup(

```

```

108.    alLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
109.        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
110.            alLayout.createSequentialGroup())
111.        .addGroup(alLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
112.            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
113.                alLayout.createSequentialGroup())
114.                .addGap(72, 72, 72)
115.            .addGroup(alLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
116.                .addComponent(txtPass)
117.                .addComponent(txtUser)
118.                .addComponent(jLabel3)
119.                .addComponent(jLabel2)))
120.            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
121.                alLayout.createSequentialGroup())
122.                .addComponent(jLabel4)
123.            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
124.                .addComponent(jLabel1)
125.                .addGap(0, 0, Short.MAX_VALUE)))
126.            .addGap(66, 66, 66))
127.        .addGroup(alLayout.createSequentialGroup()
128.            .addGap(124, 124, 124)
129.            .addComponent(btnLogin,
130.                javax.swing.GroupLayout.PREFERRED_SIZE, 150,
131.                javax.swing.GroupLayout.PREFERRED_SIZE)
132.            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
133.                Short.MAX_VALUE))
134.        );
135.        alLayout.setVerticalGroup(
136.            alLayout.createSequentialGroup()
137.                .addGroup(alLayout.createSequentialGroup()
138.                    .addGroup(alLayout.createSequentialGroup()
139.                        .addComponent(jLabel4,
140.                            javax.swing.GroupLayout.PREFERRED_SIZE, 66,
141.                            javax.swing.GroupLayout.PREFERRED_SIZE)
142.                        .addGroup(alLayout.createSequentialGroup()
143.                            .addGap(15, 15, 15)
144.                            .addComponent(jLabel1)))
145.                    .addGap(18, 18, 18)
146.                    .addComponent(jLabel3)
147.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

140.                .addComponent(txtUser,
    javax.swing.GroupLayout.PREFERRED_SIZE, 48,
    javax.swing.GroupLayout.PREFERRED_SIZE)
141.                .addGap(18, 18, 18)
142.                .addComponent(jLabel2)
143.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
144.                .addComponent(txtPass,
    javax.swing.GroupLayout.PREFERRED_SIZE, 45,
    javax.swing.GroupLayout.PREFERRED_SIZE)
145.                .addGap(18, 18, 18)
146.                .addComponent(btnLogin,
    javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
147.                .addContainerGap(62, Short.MAX_VALUE))
148.            );
149.
150.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
151.            getContentPane().setLayout(layout);
152.            layout.setHorizontalGroup(
153.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
154.                    .addComponent(a1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
155.            );
156.            layout.setVerticalGroup(
157.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
158.                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    layout.createSequentialGroup())
159.                    .addGap(0, 0, Short.MAX_VALUE)
160.                    .addComponent(a1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
161.            );
162.
163.            pack();
164.        } // </editor-fold>
165.
166.        private void btnLoginMouseClicked(java.awt.event.MouseEvent
    evt) {
167.
168.        }
169.
170.        private void jLabel4MouseClicked(java.awt.event.MouseEvent
    evt) {
171.            this.setVisible(false);
172.            Menu menu = new Menu();
173.            menu.setVisible(true);
174.            dispose();

```

```

175.     }
176.
177.     private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
178.         int c = evt.getKeyChar();
179.         if((c<48 || c>57) && (c<64 || c>90) && (c<97 || c>122))
            evt.consume();
180.         if(txtUser.getText().length()>=12) evt.consume();
181.         txtUser.setForeground(Color.BLACK);
182.     }
183.
184.     private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
185.         if(txtPass.getText().length()>=12) evt.consume();
186.         txtPass.setForeground(Color.BLACK);
187.     }
188.
189.     private void
        btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
190.         txtValidate();
191.     }
192.
193.     private void txtUserKeyReleased(java.awt.event.KeyEvent evt) {
194.         btnValidate();
195.     }
196.
197.     private void txtPassKeyReleased(java.awt.event.KeyEvent evt) {
198.         btnValidate();
199.     }
200.
201.     public void txtValidate(){
202.         String user = txtUser.getText().toLowerCase();
203.         String pass = txtPass.getText().toLowerCase();
204.         if(methodsSql.ValidateUser(user)){
205.             userEntity = methodsSql.Login(user, pass);
206.             if(userEntity.getPass() == null){
207.                 txtPass.setForeground(Color.RED);
208.                 JOptionPane.showMessageDialog(this, "CONTRASEÑA
INCORRECTA", "",
209.                                     JOptionPane.WARNING_MESSAGE);
210.                 txtPass.requestFocus();
211.             } else {
212.
213.                 if(userEntity.getState() == 1){
214.                     this.setVisible(false);
215.                     MenuAdmin menu = new MenuAdmin();
216.                     menu.setVisible(true);
217.                     dispose();
218.                 }
219.                 if(userEntity.getState() == 0){
220.                     this.setVisible(false);
221.                     MenuUser menu = new MenuUser();
222.                     menu.setVisible(true);
223.                     dispose();
224.                 }
225.             }
226.         } else {
227.             txtUser.setForeground(Color.RED);
228.             txtPass.setForeground(Color.RED);

```

```

229.             JOptionPane.showMessageDialog(
230.                 this, "USUARIO INCORRECTO O NO REGISTRADO",
231.                 "",
232.                 JOptionPane.WARNING_MESSAGE);
233.             txtUser.requestFocus();
234.         }
235.     }
236.     private void btnValidate(){
237.         if(txtUser.getText().isEmpty() ||
238.            txtPass.getText().isEmpty()){
239.             btnLogin.setEnabled(false);
240.         } else {
241.             btnLogin.setEnabled(true);
242.         }
243.     }
244.     /**
245.      * @param args the command line arguments
246.      */
247.     public static void main(String args[]) {
248.         /* Set the Nimbus look and feel */
249.         //

```

```

270.         //</editor-fold>
271.
272.         /* Create and display the form */
273.         java.awt.EventQueue.invokeLater(new Runnable() {
274.             public void run() {
275.                 new Login().setVisible(true);
276.
277.             }
278.
279.         });
280.     }
281.
282.     // Variables declaration - do not modify
283.     private javax.swing.JPanel al;
284.     private javax.swing.JButton btnLogin;
285.     private javax.swing.JLabel jLabel1;
286.     private javax.swing.JLabel jLabel2;
287.     private javax.swing.JLabel jLabel3;
288.     private javax.swing.JLabel jLabel4;
289.     private javax.swing.JPasswordField txtPass;
290.     private javax.swing.JTextField txtUser;
291.     // End of variables declaration
292. }

```

Menu

```

1.  /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package view;
7.
8.  /**
9.   *
10.   * @author serk9
11.   */
12.  public class Menu extends javax.swing.JFrame {
13.
14.      public Menu() {
15.          initComponents();
16.          setTitle("Car Agency");
17.          setLocationRelativeTo(null);
18.      }
19.
20.      /**
21.       * This method is called from within the constructor to
  initialize the form.
22.       * WARNING: Do NOT modify this code. The content of this
  method is always
23.       * regenerated by the Form Editor.
24.       */
25.      @SuppressWarnings("unchecked")

```

```

26.      // <editor-fold defaultstate="collapsed" desc="Generated
      Code">
27.      private void initComponents() {
28.
29.          jButton4 = new javax.swing.JButton();
30.          jButton3 = new javax.swing.JButton();
31.          jPanel1 = new javax.swing.JPanel();
32.          jLabel1 = new javax.swing.JLabel();
33.          btnStart = new javax.swing.JPanel();
34.          l2 = new javax.swing.JLabel();
35.          btnRegister = new javax.swing.JPanel();
36.          l3 = new javax.swing.JLabel();
37.
38.          jButton4.setText("registrarse");
39.
40.          jButton3.setText("iniciar ");
41.
42.          setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
          ;
43.          setFocusCycleRoot(false);
44.          setResizable(false);
45.
46.          jPanel1.setBackground(new java.awt.Color(128, 153, 155));
47.          jPanel1.setForeground(new java.awt.Color(0, 51, 204));
48.          jPanel1.setPreferredSize(new java.awt.Dimension(400,
          600));
49.
50.          jLabel1.setBackground(new java.awt.Color(0, 0, 0));
51.          jLabel1.setFont(new java.awt.Font("Times New Roman", 0,
          48)); // NOI18N
52.          jLabel1.setForeground(new java.awt.Color(255, 255, 255));
53.
54.          jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
55.          jLabel1.setText("Bienvenido ");
56.          jLabel1.addMouseListener(new java.awt.event.MouseAdapter()
          {
57.              public void mousePressed(java.awt.event.MouseEvent
          evt) {
58.                  jLabel1MousePressed(evt);
59.              }
60.          });
61.          btnStart.setBackground(new java.awt.Color(128, 153, 155));
62.
63.          btnStart.setBorder(javax.swing.BorderFactory.createBevelBorder(java
          x.swing.border.BevelBorder.RAISED));
64.          btnStart.setForeground(new java.awt.Color(128, 153, 155));
65.
66.          l2.setBackground(new java.awt.Color(0, 0, 0));
67.          l2.setFont(new java.awt.Font("Times New Roman", 0, 48));
68.          // NOI18N
69.          l2.setForeground(new java.awt.Color(255, 255, 255));
70.
71.          l2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

```

```

69.         l2.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/enter.png")
    )); // NOI18N
70.         l2.setText("  Iniciar");
71.         l2.setToolTipText("");
72.         l2.addMouseListener(new java.awt.event.MouseAdapter() {
73.             public void mousePressed(java.awt.event.MouseEvent
    evt) {
74.                 l2MousePressed(evt);
75.             }
76.         });
77.
78.         javax.swing.GroupLayout btnStartLayout = new
    javax.swing.GroupLayout(btnStart);
79.         btnStart.setLayout(btnStartLayout);
80.         btnStartLayout.setHorizontalGroup(
81.             btnStartLayout.createParallelGroup(javax.swing.GroupLayout.Alignmen
    t.LEADING)
82.                 .addComponent(l2,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
83.             );
84.         btnStartLayout.setVerticalGroup(
85.             btnStartLayout.createParallelGroup(javax.swing.GroupLayout.Alignmen
    t.LEADING)
86.                 .addComponent(l2,
    javax.swing.GroupLayout.Alignment.TRAILING,
    javax.swing.GroupLayout.DEFAULT_SIZE, 78, Short.MAX_VALUE)
87.             );
88.
89.         btnRegister.setBackground(new java.awt.Color(128, 153,
    155));
90.
91.         btnRegister.setBorder(javax.swing.BorderFactory.createBevelBorder(j
    avax.swing.border.BevelBorder.RAISED));
92.         l3.setBackground(new java.awt.Color(0, 0, 0));
93.         l3.setFont(new java.awt.Font("Times New Roman", 0, 48));
    // NOI18N
94.         l3.setForeground(new java.awt.Color(255, 255, 255));
95.
96.         l3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
97.         l3.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/register.pn
    g"))); // NOI18N
98.         l3.setText("  Registrarse");
99.         l3.addMouseListener(new java.awt.event.MouseAdapter() {
100.             public void mousePressed(java.awt.event.MouseEvent
    evt) {
101.                 l3MousePressed(evt);
102.             }
103.         });
104.         javax.swing.GroupLayout btnRegisterLayout = new
    javax.swing.GroupLayout(btnRegister);

```

```

105.         btnRegister.setLayout(btnRegisterLayout);
106.         btnRegisterLayout.setHorizontalGroup(
107.             btnRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Align
                ment.LEADING)
108.                 .addGroup(btnRegisterLayout.createSequentialGroup())
109.                     .addComponent(13,
110.                         javax.swing.GroupLayout.PREFERRED_SIZE, 328,
111.                         javax.swing.GroupLayout.PREFERRED_SIZE)
112.                             .addGap(0, 22, Short.MAX_VALUE))
113.             );
114.         btnRegisterLayout.setVerticalGroup(
115.             btnRegisterLayout.createParallelGroup(javax.swing.GroupLayout.Align
                ment.LEADING)
116.                 .addComponent(13,
117.                     javax.swing.GroupLayout.DEFAULT_SIZE, 86, Short.MAX_VALUE)
118.                     );
119.         javax.swing.GroupLayout jPanel1Layout = new
120.             javax.swing.GroupLayout(jPanel1);
121.         jPanel1.setLayout(jPanel1Layout);
122.         jPanel1Layout.setHorizontalGroup(
123.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                .LEADING)
124.                 .addGroup(jPanel1Layout.createSequentialGroup()
125.                     .addContainerGap()
126.                     .addComponent(jLabel1,
127.                         javax.swing.GroupLayout.DEFAULT_SIZE,
128.                         javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
129.                     .addGroup(jPanel1Layout.createSequentialGroup()
130.                         .addGap(26, 26, 26)
131.                         .addGroup(jPanel1Layout.createSequentialGroup()
132.                             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                                .LEADING, false)
133.                                 .addComponent(btnRegister,
134.                                     javax.swing.GroupLayout.DEFAULT_SIZE,
135.                                     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
136.                                 .addComponent(btnStart,
137.                                     javax.swing.GroupLayout.DEFAULT_SIZE,
138.                                     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
139.                             .addGap(0, 0, Short.MAX_VALUE))
140.                         );
141.             jPanel1Layout.setVerticalGroup(
142.                 jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                    .LEADING)
143.                     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
144.                         jPanel1Layout.createSequentialGroup()
145.                             .addGap(58, 58, 58)
146.                             .addComponent(jLabel1)
147.                             .addGap(85, 85, 85)
148.                             .addComponent(btnStart,
149.                                 javax.swing.GroupLayout.PREFERRED_SIZE,
150.                                 javax.swing.GroupLayout.DEFAULT_SIZE,
151.                                 javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

138.             .addGap(54, 54, 54)
139.             .addComponent(btnRegister,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
140.
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE))
141.         );
142.
143.         javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
144.         getContentPane().setLayout(layout);
145.         layout.setHorizontalGroup(
146.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
147.
    .addComponent(jPanel1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
148.        );
149.        layout.setVerticalGroup(
150.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
151.
    .addComponent(jPanel1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
152.        );
153.
154.        pack();
155.    } // </editor-fold>
156.
157.
158.    private void l3MousePressed(java.awt.event.MouseEvent evt) {
159.        this.setVisible(false);
160.        Register reg = new Register();
161.        reg.setVisible(true);
162.        dispose();
163.    }
164.
165.    private void jLabel1MousePressed(java.awt.event.MouseEvent
    evt) {
166.
167.    }
168.
169.    private void l2MousePressed(java.awt.event.MouseEvent evt) {
170.        this.setVisible(false);
171.        Login login = new Login();
172.        login.setVisible(true);
173.        dispose();
174.
175.    }
176.
177.    /**
178.     * @param args the command line arguments
179.     */

```



```

180.     public static void main(String args[]) {
181.         /* Set the Nimbus look and feel */
182.         //editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">
183.         /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.
184.         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
185.         */
186.         try {
187.             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
188.                 if ("Nimbus".equals(info.getName())) {
189.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
190.                     break;
191.                 }
192.             }
193.         } catch (ClassNotFoundException ex) {
194.             java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
195.         } catch (InstantiationException ex) {
196.             java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
197.         } catch (IllegalAccessException ex) {
198.             java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
199.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
200.             java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
201.         }
202.         //editor-fold>
203.         //editor-fold>
204.         //editor-fold>
205.         //editor-fold>
206.
207.         /* Create and display the form */
208.         java.awt.EventQueue.invokeLater(new Runnable() {
209.             public void run() {
210.                 new Menu().setVisible(true);
211.             }
212.         });
213.     }
214.
215.     // Variables declaration - do not modify
216.     private javax.swing.JPanel btnRegister;
217.     private javax.swing.JPanel btnStart;
218.     private javax.swing.JButton jButton3;
219.     private javax.swing.JButton jButton4;
220.     private javax.swing.JLabel jLabel1;
221.     private javax.swing.JPanel jPanel1;
222.     private javax.swing.JLabel l2;

```

```
223.     private javax.swing.JLabel l3;
224.     // End of variables declaration
225. }
```

ModifyUser

```
1.  /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package view;
7.
8.  import code.MethodsSQL;
9.  import java.awt.Color;
10. import java.text.DateFormat;
11. import java.text.SimpleDateFormat;
12. import java.util.regex.Matcher;
13. import java.util.regex.Pattern;
14. import javax.swing.ImageIcon;
15. import view.admin.MenuAdmin;
16. import view.user.MenuUser;
17.
18.
19. /**
20.  *
21.  * @author Restf
22.  */
23. public class ModifyUser extends javax.swing.JFrame {
24.
25.     MethodsSQL methodsSql = new MethodsSQL();
26.     ImageIcon green = new
27. ImageIcon(getClass().getResource("/resource/Check-True.png"));
28.     ImageIcon red = new
29. ImageIcon(getClass().getResource("/resource/Check-False.png"));
30.     private boolean check1, check2, check3, check4, check5;
31.
32.     public ModifyUser() {
33.         initComponents();
34.         setLocationRelativeTo(null);
35.         String STATE = null;
36.         if(Login.userEntity.getState()==1){
37.             STATE = "Administrador";
38.             txtStateNew.setText(STATE);
39.         } else {
40.             STATE = "Usuario";
41.             txtStateNew.setText(STATE);
42.         }
43.         setTitle("Car Agency - Editar datos de "+STATE+" - [
44. "+Login.userEntity.getName()+" ]");
45.         txtId.setText(Integer.toString(Login.userEntity.getId()));
46.         txtState.setText(STATE);
47.         txtName.setText(Login.userEntity.getName());
48.         txtEmail.setText(Login.userEntity.getEmail());
49.     }
50. }
```

```

46.         txtRFC.setText(Login.userEntity.getRfc());
47.         txtUser.setText(Login.userEntity.getUser());
48.         txtPass.setText(Login.userEntity.getPass());
49.         txtAge.setText(Login.userEntity.getAge());
50.     }
51.
52.     /**
53.      * This method is called from within the constructor to
        initialize the form.
54.      * WARNING: Do NOT modify this code. The content of this
        method is always
55.      * regenerated by the Form Editor.
56.      */
57.     @SuppressWarnings("unchecked")
58.     // <editor-fold defaultstate="collapsed" desc="Generated
        Code">
59.     private void initComponents() {
60.
61.         jPanel1 = new javax.swing.JPanel();
62.         jLabel4 = new javax.swing.JLabel();
63.         jLabel11 = new javax.swing.JLabel();
64.         jPanel2 = new javax.swing.JPanel();
65.         jLabel2 = new javax.swing.JLabel();
66.         txtId = new javax.swing.JTextField();
67.         txtState = new javax.swing.JTextField();
68.         jLabel3 = new javax.swing.JLabel();
69.         jLabel5 = new javax.swing.JLabel();
70.         txtName = new javax.swing.JTextField();
71.         txtEmail = new javax.swing.JTextField();
72.         jLabel6 = new javax.swing.JLabel();
73.         jLabel7 = new javax.swing.JLabel();
74.         txtUser = new javax.swing.JTextField();
75.         txtPass = new javax.swing.JTextField();
76.         jLabel8 = new javax.swing.JLabel();
77.         jLabel9 = new javax.swing.JLabel();
78.         txtAge = new javax.swing.JTextField();
79.         txtRFC = new javax.swing.JTextField();
80.         jLabel10 = new javax.swing.JLabel();
81.         jPanel3 = new javax.swing.JPanel();
82.         jLabel12 = new javax.swing.JLabel();
83.         txtIdNew = new javax.swing.JTextField();
84.         txtStateNew = new javax.swing.JTextField();
85.         jLabel13 = new javax.swing.JLabel();
86.         jLabel14 = new javax.swing.JLabel();
87.         txtNameNew = new javax.swing.JTextField();
88.         txtEmailNew = new javax.swing.JTextField();
89.         jLabel15 = new javax.swing.JLabel();
90.         jLabel16 = new javax.swing.JLabel();
91.         txtUserNew = new javax.swing.JTextField();
92.         txtPassNew = new javax.swing.JTextField();
93.         jLabel17 = new javax.swing.JLabel();
94.         txtRFCNew = new javax.swing.JTextField();
95.         jLabel19 = new javax.swing.JLabel();
96.         txtAgeNew = new com.toedter.calendar.JDateChooser();
97.         jLabel18 = new javax.swing.JLabel();
98.         btnSave = new javax.swing.JButton();
99.         jPanel5 = new javax.swing.JPanel();

```

```

100.         UserLabel = new javax.swing.JLabel();
101.         jSeparator1 = new javax.swing.JSeparator();
102.         iconUserExist = new javax.swing.JLabel();
103.         iconUserCharacter = new javax.swing.JLabel();
104.         UserLabel1 = new javax.swing.JLabel();
105.         jSeparator2 = new javax.swing.JSeparator();
106.         iconRFCEXist = new javax.swing.JLabel();
107.         iconRFCChracter = new javax.swing.JLabel();
108.         iconEmailValidate = new javax.swing.JLabel();
109.         jSeparator3 = new javax.swing.JSeparator();
110.         UserLabel2 = new javax.swing.JLabel();
111.
112.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
113.         ;
114.         setResizable(false);
115.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
116.         jPanel1.setPreferredSize(new java.awt.Dimension(1280,
117.         720));
118.         jPanel1.setLayout(null);
119.         jLabel4.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
120.         jLabel4.setForeground(new java.awt.Color(0, 0, 0));
121.         jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
); // NOI18N
122.         jLabel4.setText("VOLVER");
123.         jLabel4.addMouseListener(new java.awt.event.MouseAdapter()
124.         {
125.             public void mouseClicked(java.awt.event.MouseEvent
evt) {
126.                 jLabel4MouseClicked(evt);
127.             }
128.         });
129.         jPanel1.add(jLabel4);
130.         jLabel4.setBounds(6, 6, 174, 66);
131.         jLabel1.setFont(new java.awt.Font("Dialog", 1, 36)); //
NOI18N
132.         jLabel1.setForeground(new java.awt.Color(255, 255, 255));
133.         jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
edit-on.png"))); // NOI18N
134.         jLabel1.setText("EDITAR DATOS DE USUARIO");
135.         jLabel1.setToolTipText("");
136.         jPanel1.add(jLabel1);
137.         jLabel1.setBounds(299, 14, 641, 134);
138.
139.         jPanel2.setBackground(new java.awt.Color(128, 153, 155));
140.         jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "DATOS ACTUALES",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new

```

```

        java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
255))); // NOI18N
141.
142.        jLabel2.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
143.        jLabel2.setForeground(new java.awt.Color(255, 255, 255));
144.        jLabel2.setText("ID");
145.
146.        txtId.setBackground(new java.awt.Color(255, 255, 255));
147.        txtId.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
148.        txtId.setForeground(new java.awt.Color(255, 255, 255));
149.        txtId.setEnabled(false);
150.        txtId.setPreferredSize(new java.awt.Dimension(220, 40));
151.        txtId.addActionListener(new
        java.awt.event.ActionListener() {
152.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
153.                txtIdActionPerformed(evt);
154.            }
155.        });
156.        txtId.addKeyListener(new java.awt.event.KeyAdapter() {
157.            public void keyTyped(java.awt.event.KeyEvent evt) {
158.                txtIdKeyTyped(evt);
159.            }
160.        });
161.
162.        txtState.setBackground(new java.awt.Color(255, 255, 255));
163.        txtState.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
164.        txtState.setForeground(new java.awt.Color(51, 51, 51));
165.        txtState.setEnabled(false);
166.        txtState.setPreferredSize(new java.awt.Dimension(220,
        40));
167.        txtState.addActionListener(new
        java.awt.event.ActionListener() {
168.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
169.                txtStateActionPerformed(evt);
170.            }
171.        });
172.        txtState.addKeyListener(new java.awt.event.KeyAdapter() {
173.            public void keyTyped(java.awt.event.KeyEvent evt) {
174.                txtStateKeyTyped(evt);
175.            }
176.        });
177.
178.        jLabel3.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
179.        jLabel3.setForeground(new java.awt.Color(255, 255, 255));
180.        jLabel3.setText("ESTADO");
181.
182.        jLabel5.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
183.        jLabel5.setForeground(new java.awt.Color(255, 255, 255));
184.        jLabel5.setText("NOMBRE");
185.

```

```

186.         txtName.setBackground(new java.awt.Color(255, 255, 255));
187.         txtName.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
188.         txtName.setForeground(new java.awt.Color(51, 51, 51));
189.         txtName.setEnabled(false);
190.         txtName.setPreferredSize(new java.awt.Dimension(220, 40));
191.         txtName.addActionListener(new
    java.awt.event.ActionListener() {
192.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
193.                 txtNameActionPerformed(evt);
194.             }
195.         });
196.         txtName.addKeyListener(new java.awt.event.KeyAdapter() {
197.             public void keyTyped(java.awt.event.KeyEvent evt) {
198.                 txtNameKeyTyped(evt);
199.             }
200.         });
201.
202.         txtEmail.setBackground(new java.awt.Color(255, 255, 255));
203.         txtEmail.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
204.         txtEmail.setForeground(new java.awt.Color(51, 51, 51));
205.         txtEmail.setEnabled(false);
206.         txtEmail.setPreferredSize(new java.awt.Dimension(220,
    40));
207.         txtEmail.addActionListener(new
    java.awt.event.ActionListener() {
208.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
209.                 txtEmailActionPerformed(evt);
210.             }
211.         });
212.         txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
213.             public void keyTyped(java.awt.event.KeyEvent evt) {
214.                 txtEmailKeyTyped(evt);
215.             }
216.         });
217.
218.         jLabel6.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
219.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
220.         jLabel6.setText("CORREO");
221.
222.         jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
223.         jLabel7.setForeground(new java.awt.Color(255, 255, 255));
224.         jLabel7.setText("USUARIO");
225.
226.         txtUser.setBackground(new java.awt.Color(255, 255, 255));
227.         txtUser.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
228.         txtUser.setForeground(new java.awt.Color(51, 51, 51));
229.         txtUser.setEnabled(false);
230.         txtUser.setPreferredSize(new java.awt.Dimension(220, 40));
231.         txtUser.addActionListener(new
    java.awt.event.ActionListener() {

```

```

232.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
233.             txtUserActionPerformed(evt);
234.         }
235.     });
236.     txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
237.         public void keyTyped(java.awt.event.KeyEvent evt) {
238.             txtUserKeyTyped(evt);
239.         }
240.     });
241.
242.     txtPass.setBackground(new java.awt.Color(255, 255, 255));
243.     txtPass.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
244.     txtPass.setForeground(new java.awt.Color(51, 51, 51));
245.     txtPass.setEnabled(false);
246.     txtPass.setPreferredSize(new java.awt.Dimension(220, 40));
247.     txtPass.addActionListener(new
    java.awt.event.ActionListener() {
248.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
249.             txtPassActionPerformed(evt);
250.         }
251.     });
252.     txtPass.addKeyListener(new java.awt.event.KeyAdapter() {
253.         public void keyTyped(java.awt.event.KeyEvent evt) {
254.             txtPassKeyTyped(evt);
255.         }
256.     });
257.
258.     jLabel8.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
259.     jLabel8.setForeground(new java.awt.Color(255, 255, 255));
260.     jLabel8.setText("CONTRASEÑA");
261.
262.     jLabel9.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
263.     jLabel9.setForeground(new java.awt.Color(255, 255, 255));
264.     jLabel9.setText("EDAD");
265.
266.     txtAge.setBackground(new java.awt.Color(255, 255, 255));
267.     txtAge.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
268.     txtAge.setForeground(new java.awt.Color(51, 51, 51));
269.     txtAge.setEnabled(false);
270.     txtAge.setPreferredSize(new java.awt.Dimension(220, 40));
271.     txtAge.addActionListener(new
    java.awt.event.ActionListener() {
272.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
273.             txtAgeActionPerformed(evt);
274.         }
275.     });
276.     txtAge.addKeyListener(new java.awt.event.KeyAdapter() {
277.         public void keyTyped(java.awt.event.KeyEvent evt) {
278.             txtAgeKeyTyped(evt);
279.         }

```

```

280.         });
281.
282.         txtRFC.setBackground(new java.awt.Color(255, 255, 255));
283.         txtRFC.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
284.         txtRFC.setForeground(new java.awt.Color(51, 51, 51));
285.         txtRFC.setEnabled(false);
286.         txtRFC.setPreferredSize(new java.awt.Dimension(220, 40));
287.         txtRFC.addActionListener(new
java.awt.event.ActionListener() {
288.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
289.                 txtRFCActionPerformed(evt);
290.             }
291.         });
292.         txtRFC.addKeyListener(new java.awt.event.KeyAdapter() {
293.             public void keyTyped(java.awt.event.KeyEvent evt) {
294.                 txtRFCKeyTyped(evt);
295.             }
296.         });
297.
298.         jLabel10.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
299.         jLabel10.setForeground(new java.awt.Color(255, 255, 255));
300.         jLabel10.setText("RFC");
301.
302.         javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
303.         jPanel2.setLayout(jPanel2Layout);
304.         jPanel2Layout.setHorizontalGroup(
305.             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
306.                 .addGroup(jPanel2Layout.createSequentialGroup()
307.                     .addGap(19, 19, 19)
308.                     .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING, false)
309.                         .addGroup(jPanel2Layout.createSequentialGroup()
310.                             .addComponent(jLabel10)
311.                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
312.                             .addComponent(txtRFC,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
313.                         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
314.                             .addComponent(jLabel17)
315.                             .addGap(70, 70, 70)
316.                             .addComponent(txtUser,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```



```
317.        .addGroup(jPanel2Layout.createSequentialGroup())
318.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
319.                .addComponent(jLabel8)
320.                .addComponent(jLabel9))
321.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
322.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
323.                .addComponent(txtAge,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
324.                .addComponent(txtPass,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
325.        .addGroup(jPanel2Layout.createSequentialGroup())
326.                .addComponent(jLabel6)
327.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
328.                .addComponent(txtEmail,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
329.        .addGroup(jPanel2Layout.createSequentialGroup())
330.                .addComponent(jLabel5)
331.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
332.                .addComponent(txtName,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
333.        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanel2Layout.createSequentialGroup())
334.                .addComponent(jLabel2)
335.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
336.                .addComponent(txtId,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
337.        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanel2Layout.createSequentialGroup())
338.                .addComponent(jLabel3)
339.                .addGap(80, 80, 80)
```

```
340.                .addComponent(txtState,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
341.                .addContainerGap(16, Short.MAX_VALUE))
342.            );
343.            jPanel2Layout.setVerticalGroup(
344.                jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
345.                .addGroup(jPanel2Layout.createSequentialGroup())
346.                .addGap(14, 14, 14)
347.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
348.                .addComponent(txtId,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE)
349.                .addComponent(jLabel2))
350.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
351.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
352.                .addComponent(jLabel3)
353.                .addComponent(txtState,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
354.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
355.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
356.                .addComponent(jLabel5)
357.                .addComponent(txtName,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
358.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
359.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
360.                .addComponent(jLabel6)
361.                .addComponent(txtEmail,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
362.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
363.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
364.                .addComponent(jLabel10)
```

```

365.                .addComponent(txtRFC,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
366.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
367.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
368.                .addComponent(jLabel7)
369.                .addComponent(txtUser,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
370.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
371.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
372.                .addComponent(jLabel8)
373.                .addComponent(txtPass,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
374.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
375.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
376.                .addComponent(jLabel9)
377.                .addComponent(txtAge,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
378.        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE))
379.    );
380.
381.    jPanel1.add(jPanel2);
382.    jPanel2.setBounds(338, 175, 444, 413);
383.
384.    jPanel3.setBackground(new java.awt.Color(128, 153, 155));
385.    jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null
    , "DATOS NUEVIOS",
    javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
    javax.swing.border.TitledBorder.DEFAULT_POSITION, new
    java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
    255))); // NOI18N
386.
387.    jLabel12.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
388.    jLabel12.setForeground(new java.awt.Color(255, 255, 255));
389.    jLabel12.setText("ID");
390.
391.    txtIdNew.setBackground(new java.awt.Color(255, 255, 255));

```

```

392.         txtIdNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
393.         txtIdNew.setForeground(new java.awt.Color(51, 51, 51));
394.         txtIdNew.setEnabled(false);
395.         txtIdNew.setPreferredSize(new java.awt.Dimension(220,
    40));
396.         txtIdNew.addActionListener(new
    java.awt.event.ActionListener() {
397.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
398.                 txtIdNewActionPerformed(evt);
399.             }
400.         });
401.         txtIdNew.addKeyListener(new java.awt.event.KeyAdapter() {
402.             public void keyTyped(java.awt.event.KeyEvent evt) {
403.                 txtIdNewKeyTyped(evt);
404.             }
405.         });
406.
407.         txtStateNew.setBackground(new java.awt.Color(255, 255,
    255));
408.         txtStateNew.setFont(new java.awt.Font("Dialog", 0, 18));
    // NOI18N
409.         txtStateNew.setForeground(new java.awt.Color(51, 51, 51));
410.         txtStateNew.setEnabled(false);
411.         txtStateNew.setPreferredSize(new java.awt.Dimension(220,
    40));
412.         txtStateNew.addActionListener(new
    java.awt.event.ActionListener() {
413.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
414.                 txtStateNewActionPerformed(evt);
415.             }
416.         });
417.         txtStateNew.addKeyListener(new java.awt.event.KeyAdapter()
    {
418.             public void keyTyped(java.awt.event.KeyEvent evt) {
419.                 txtStateNewKeyTyped(evt);
420.             }
421.         });
422.
423.         jLabel13.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
424.         jLabel13.setForeground(new java.awt.Color(255, 255, 255));
425.         jLabel13.setText("ESTADO");
426.
427.         jLabel14.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
428.         jLabel14.setForeground(new java.awt.Color(255, 255, 255));
429.         jLabel14.setText("NOMBRE");
430.
431.         txtNameNew.setBackground(new java.awt.Color(255, 255,
    255));
432.         txtNameNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
433.         txtNameNew.setForeground(new java.awt.Color(102, 102,
    102));

```

```

434.         txtNameNew.setPreferredSize(new java.awt.Dimension(220,
435.         40));
436.         txtNameNew.addActionListener(new
437.         java.awt.event.ActionListener() {
438.             public void actionPerformed(java.awt.event.ActionEvent
439.             evt) {
440.                 txtNameNewActionPerformed(evt);
441.             }
442.         });
443.         txtNameNew.addKeyListener(new java.awt.event.KeyAdapter()
444.         {
445.             public void keyReleased(java.awt.event.KeyEvent evt) {
446.                 txtNameNewKeyReleased(evt);
447.             }
448.             public void keyTyped(java.awt.event.KeyEvent evt) {
449.                 txtNameNewKeyTyped(evt);
450.             }
451.         });
452.         txtEmailNew.setBackground(new java.awt.Color(255, 255,
453.         255));
454.         txtEmailNew.setFont(new java.awt.Font("Dialog", 0, 18));
455.         // NOI18N
456.         txtEmailNew.setForeground(new java.awt.Color(102, 102,
457.         102));
458.         txtEmailNew.setPreferredSize(new java.awt.Dimension(220,
459.         40));
460.         txtEmailNew.addActionListener(new
461.         java.awt.event.ActionListener() {
462.             public void actionPerformed(java.awt.event.ActionEvent
463.             evt) {
464.                 txtEmailNewActionPerformed(evt);
465.             }
466.         });
467.         txtEmailNew.addKeyListener(new java.awt.event.KeyAdapter()
468.         {
469.             public void keyReleased(java.awt.event.KeyEvent evt) {
470.                 txtEmailNewKeyReleased(evt);
471.             }
472.             public void keyTyped(java.awt.event.KeyEvent evt) {
473.                 txtEmailNewKeyTyped(evt);
474.             }
475.         });
476.         jLabel15.setFont(new java.awt.Font("Dialog", 1, 24)); //
477.         NOI18N
478.         jLabel15.setForeground(new java.awt.Color(255, 255, 255));
479.         jLabel15.setText("CORREO");
480.         jLabel16.setFont(new java.awt.Font("Dialog", 1, 24)); //
481.         NOI18N
482.         jLabel16.setForeground(new java.awt.Color(255, 255, 255));
483.         jLabel16.setText("USUARIO");
484.         txtUserNew.setBackground(new java.awt.Color(255, 255,
485.         255));

```

```

476.         txtUserNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
477.         txtUserNew.setForeground(new java.awt.Color(102, 102,
102));
478.         txtUserNew.setPreferredSize(new java.awt.Dimension(220,
40));
479.         txtUserNew.addActionListener(new
java.awt.event.ActionListener() {
480.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
481.                 txtUserNewActionPerformed(evt);
482.             }
483.         });
484.         txtUserNew.addKeyListener(new java.awt.event.KeyAdapter()
{
485.             public void keyReleased(java.awt.event.KeyEvent evt) {
486.                 txtUserNewKeyReleased(evt);
487.             }
488.             public void keyTyped(java.awt.event.KeyEvent evt) {
489.                 txtUserNewKeyTyped(evt);
490.             }
491.         });
492.
493.         txtPassNew.setBackground(new java.awt.Color(255, 255,
255));
494.         txtPassNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
495.         txtPassNew.setForeground(new java.awt.Color(102, 102,
102));
496.         txtPassNew.setPreferredSize(new java.awt.Dimension(220,
40));
497.         txtPassNew.addActionListener(new
java.awt.event.ActionListener() {
498.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
499.                 txtPassNewActionPerformed(evt);
500.             }
501.         });
502.         txtPassNew.addKeyListener(new java.awt.event.KeyAdapter()
{
503.             public void keyReleased(java.awt.event.KeyEvent evt) {
504.                 txtPassNewKeyReleased(evt);
505.             }
506.             public void keyTyped(java.awt.event.KeyEvent evt) {
507.                 txtPassNewKeyTyped(evt);
508.             }
509.         });
510.
511.         jLabel17.setFont(new java.awt.Font("Dialog", 1, 24)); //
    NOI18N
512.         jLabel17.setForeground(new java.awt.Color(255, 255, 255));
513.         jLabel17.setText("CONTRASEÑA");
514.
515.         txtRFCNew.setBackground(new java.awt.Color(255, 255,
255));
516.         txtRFCNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N

```

```

517.         txtRFCNew.setForeground(new java.awt.Color(102, 102,
102));
518.         txtRFCNew.setPreferredSize(new java.awt.Dimension(220,
40));
519.         txtRFCNew.addActionListener(new
java.awt.event.ActionListener() {
520.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
521.                 txtRFCNewActionPerformed(evt);
522.             }
523.         });
524.         txtRFCNew.addKeyListener(new java.awt.event.KeyAdapter() {
525.             public void keyReleased(java.awt.event.KeyEvent evt) {
526.                 txtRFCNewKeyReleased(evt);
527.             }
528.             public void keyTyped(java.awt.event.KeyEvent evt) {
529.                 txtRFCNewKeyTyped(evt);
530.             }
531.         });
532.
533.         jLabel19.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
534.         jLabel19.setForeground(new java.awt.Color(255, 255, 255));
535.         jLabel19.setText("RFC");
536.
537.         txtAgeNew.setBackground(new java.awt.Color(255, 255,
255));
538.         txtAgeNew.setToolTipText("");
539.         txtAgeNew.setDateFormatString("yyyy-mm-dd");
540.         txtAgeNew.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
541.         txtAgeNew.setPreferredSize(new java.awt.Dimension(220,
40));
542.         txtAgeNew.addKeyListener(new java.awt.event.KeyAdapter() {
543.             public void keyReleased(java.awt.event.KeyEvent evt) {
544.                 txtAgeNewKeyReleased(evt);
545.             }
546.             public void keyTyped(java.awt.event.KeyEvent evt) {
547.                 txtAgeNewKeyTyped(evt);
548.             }
549.         });
550.
551.         jLabel18.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
552.         jLabel18.setForeground(new java.awt.Color(255, 255, 255));
553.         jLabel18.setText("EDAD");
554.
555.         javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
556.         jPanel3.setLayout(jPanel3Layout);
557.         jPanel3Layout.setHorizontalGroup(
558.             jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
559.                 .addGroup(jPanel3Layout.createSequentialGroup()
560.                     .addGroup(jPanel3Layout.createParallelGroup(

```

```

561.        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
562.        .addGroup(jPanel3Layout.createSequentialGroup()
563.                .addComponent(jLabel18)
564.                .addGap(113, 113, 113)
565.                .addComponent(txtAgeNew,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
566.        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.TRAILING, false)
567.        .addGroup(jPanel3Layout.createSequentialGroup()
568.                .addComponent(jLabel19)
569.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
570.                .addComponent(txtRFCNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
571.        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
572.        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        jPanel3Layout.createSequentialGroup()
573.                .addComponent(jLabel16)
574.                .addGap(70, 70, 70)
575.                .addComponent(txtUserNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
576.        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.TRAILING, false)
577.        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
        jPanel3Layout.createSequentialGroup()
578.                .addComponent(jLabel15)
579.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
580.                .addComponent(txtEmailNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
581.        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
        jPanel3Layout.createSequentialGroup()
582.                .addComponent(jLabel14)
583.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```



```

584.         .addComponent(txtNameNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
585.
        .addGroup(jPanel3Layout.createSequentialGroup())
586.         .addComponent(jLabel12)
587.
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
588.         .addComponent(txtIdNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
589.
        .addGroup(jPanel3Layout.createSequentialGroup())
590.         .addComponent(jLabel13)
591.         .addGap(80, 80, 80)
592.         .addComponent(txtStateNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
593.
        .addGroup(jPanel3Layout.createSequentialGroup())
594.         .addComponent(jLabel17)
595.
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
        ED)
596.         .addComponent(txtPassNew,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))))))
597.         .addContainerGap(17, Short.MAX_VALUE))
598.     );
599.     jPanel3Layout.setVerticalGroup(
600.     jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
601.         .addGroup(jPanel3Layout.createSequentialGroup())
602.         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)
603.         .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.BASELINE)
604.             .addComponent(txtIdNew,
        javax.swing.GroupLayout.PREFERRED_SIZE, 40,
        javax.swing.GroupLayout.PREFERRED_SIZE)
605.             .addComponent(jLabel12))
606.         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        )
607.         .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.BASELINE)
608.             .addComponent(jLabel13)

```

```
609.                .addComponent(txtStateNew,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
610.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
611.    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
612.        .addComponent(jLabel14)
613.        .addComponent(txtNameNew,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
614.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
615.    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
616.        .addComponent(jLabel15)
617.        .addComponent(txtEmailNew,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
618.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
619.    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
620.        .addComponent(jLabel19)
621.        .addComponent(txtRFCNew,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
622.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
623.    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
624.        .addComponent(jLabel18)
625.        .addComponent(txtAgeNew,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
626.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
627.    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
628.        .addComponent(jLabel16)
629.        .addComponent(txtUserNew,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
630.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
```

```

631.        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.BASELINE)
632.                .addComponent(jLabel17)
633.                .addComponent(txtPassNew,
        javax.swing.GroupLayout.PREFERRED_SIZE, 40,
        javax.swing.GroupLayout.PREFERRED_SIZE))
634.                .addGap(61, 61, 61))
635.        );
636.
637.        jPanel1.add(jPanel3);
638.        jPanel3.setBounds(788, 175, 448, 413);
639.
640.        btnSave.setBackground(new java.awt.Color(255, 255, 255));
641.        btnSave.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
642.        btnSave.setText("ACTUALIZAR");
643.        btnSave.setEnabled(false);
644.        btnSave.addActionListener(new
        java.awt.event.ActionListener() {
645.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
646.                btnSaveActionPerformed(evt);
647.            }
648.        });
649.        jPanel1.add(btnSave);
650.        btnSave.setBounds(660, 610, 245, 68);
651.
652.        jPanel5.setBackground(new java.awt.Color(128, 153, 155));
653.        jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(""))
        ;
654.
655.        UserLabel.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
656.        UserLabel.setForeground(new java.awt.Color(255, 255,
        255));
657.        UserLabel.setText("USUARIO");
658.        UserLabel.setFocusable(false);
659.
660.        iconUserExist.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
661.        iconUserExist.setForeground(new java.awt.Color(255, 255,
        255));
662.        iconUserExist.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/Check-
        False.png"))); // NOI18N
663.        iconUserExist.setText(" » Usuario no existente");
664.        iconUserExist.setFocusable(false);
665.
666.        iconUserCharacter.setFont(new java.awt.Font("Dialog", 1,
        18)); // NOI18N
667.        iconUserCharacter.setForeground(new java.awt.Color(255,
        255, 255));
668.        iconUserCharacter.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/Check-
        False.png"))); // NOI18N

```

```

669.         iconUserCharacter.setText(" » Minimo 8 caracteres");
670.         iconUserCharacter.setFocusable(false);
671.
672.         UserLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
673.         UserLabel1.setForeground(new java.awt.Color(255, 255,
255));
674.         UserLabel1.setText("RFC");
675.         UserLabel1.setFocusable(false);
676.
677.         iconRFCExist.setFont(new java.awt.Font("Dialog", 1, 18));
// NOI18N
678.         iconRFCExist.setForeground(new java.awt.Color(255, 255,
255));
679.         iconRFCExist.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/Check-
False.png"))); // NOI18N
680.         iconRFCExist.setText(" » RFC no registrado");
681.         iconRFCExist.setFocusable(false);
682.
683.         iconRFCChracter.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
684.         iconRFCChracter.setForeground(new java.awt.Color(255, 255,
255));
685.         iconRFCChracter.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/Check-
False.png"))); // NOI18N
686.         iconRFCChracter.setText(" » Minimo 12 caracteres");
687.         iconRFCChracter.setFocusable(false);
688.
689.         iconEmailValidate.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
690.         iconEmailValidate.setForeground(new java.awt.Color(255,
255, 255));
691.         iconEmailValidate.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/Check-
False.png"))); // NOI18N
692.         iconEmailValidate.setText(" » Correo no valido");
693.         iconEmailValidate.setFocusable(false);
694.
695.         UserLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
696.         UserLabel2.setForeground(new java.awt.Color(255, 255,
255));
697.         UserLabel2.setText("CORREO");
698.         UserLabel2.setFocusable(false);
699.
700.         javax.swing.GroupLayout jPanel5Layout = new
javax.swing.GroupLayout(jPanel5);
701.         jPanel5.setLayout(jPanel5Layout);
702.         jPanel5Layout.setHorizontalGroup(
703.             jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
704.                 .addGroup(jPanel5Layout.createSequentialGroup()
705.                     .addGroup(jPanel5Layout.createParallelGroup(javax.swing.

```

```

706.    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
707.        .addComponent(UserLabel,
    javax.swing.GroupLayout.PREFERRED_SIZE, 90,
    javax.swing.GroupLayout.PREFERRED_SIZE)
708.        .addComponent(iconUserExist,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
709.        .addComponent(iconUserCharacter,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
710.        .addComponent(jSeparator1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
    javax.swing.GroupLayout.PREFERRED_SIZE)
711.        .addComponent(UserLabel1)))
712.    .addGroup(jPanel5Layout.createSequentialGroup())
713.    .addContainerGap()
714.    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
715.        .addComponent(iconRFCEXist,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
716.        .addComponent(iconRFCChracter,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
717.        .addComponent(jSeparator2,
    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
718.    .addGroup(jPanel5Layout.createSequentialGroup())
719.    .addContainerGap()
720.    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
721.    .addGroup(jPanel5Layout.createSequentialGroup())
722.        .addGap(9, 9, 9)
723.        .addComponent(UserLabel2))
724.        .addComponent(iconEmailValidate,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
725.        .addComponent(jSeparator3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
726.    );
727.    jPanel5Layout.setVerticalGroup(
728.    jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
729.    .addGroup(jPanel5Layout.createSequentialGroup())
730.        .addContainerGap()
731.        .addComponent(UserLabel)
732.        .addGap(2, 2, 2)
733.    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)

```

```

734.        .addGroup(jPanel5Layout.createSequentialGroup())
735.                .addGap(10, 10, 10)
736.                .addComponent(iconUserExist))
737.                .addComponent(jSeparator1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
738.                .addGap(2, 2, 2)
739.                .addComponent(iconUserCharacter)
740.                .addGap(18, 18, 18)
741.                .addComponent(UserLabel1)
742.                .addGap(2, 2, 2)
743.
    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
744.
    .addGroup(jPanel5Layout.createSequentialGroup())
745.                .addGap(10, 10, 10)
746.                .addComponent(iconRFCEExist))
747.                .addComponent(jSeparator2,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
748.                .addGap(2, 2, 2)
749.                .addComponent(iconRFCCChracter)
750.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
751.                .addComponent(UserLabel2)
752.                .addGap(2, 2, 2)
753.
    .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
754.
    .addGroup(jPanel5Layout.createSequentialGroup())
755.                .addGap(10, 10, 10)
756.                .addComponent(iconEmailValidate))
757.                .addComponent(jSeparator3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
758.                .addContainerGap(20, Short.MAX_VALUE))
759.        );
760.
    jPanel1.add(jPanel5);
762.    jPanel5.setBounds(25, 249, 307, 330);
763.
    javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
765.    getContentPane().setLayout(layout);
766.    layout.setHorizontalGroup(
767.
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
768.
        .addComponent(jPanel1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
769.    );

```

```
770.         layout.setVerticalGroup(
771.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
772.         .addComponent(jPanell1,
773.         javax.swing.GroupLayout.PREFERRED_SIZE,
774.         javax.swing.GroupLayout.DEFAULT_SIZE,
775.         javax.swing.GroupLayout.PREFERRED_SIZE)
776.         );
777.         pack();
778.     }// </editor-fold>
779.     private void jLabel4MouseClicked(java.awt.event.MouseEvent
780.     evt) {
781.         this.setVisible(false);
782.         if (Login.userEntity.getState() == 1) {
783.             MenuAdmin menu = new MenuAdmin();
784.             menu.setVisible(true); }
785.         if (Login.userEntity.getState() == 0) {
786.             MenuUser menu = new MenuUser();
787.             menu.setVisible(true); }
788.         dispose();
789.     }
790.     private void txtIdActionPerformed(java.awt.event.ActionEvent
791.     evt) {
792.         // TODO add your handling code here:
793.     }
794.     private void txtIdKeyTyped(java.awt.event.KeyEvent evt) {
795.     }
796.     private void
797.     txtStateActionPerformed(java.awt.event.ActionEvent evt) {
798.         // TODO add your handling code here:
799.     }
800.     private void txtStateKeyTyped(java.awt.event.KeyEvent evt) {
801.         // TODO add your handling code here:
802.     }
803.     private void txtNameActionPerformed(java.awt.event.ActionEvent
804.     evt) {
805.         // TODO add your handling code here:
806.     }
807.     private void txtNameKeyTyped(java.awt.event.KeyEvent evt) {
808.         // TODO add your handling code here:
809.     }
810.     private void
811.     txtEmailActionPerformed(java.awt.event.ActionEvent evt) {
812.         // TODO add your handling code here:
813.     }
814.     private void
815.     txtPasswordActionPerformed(java.awt.event.ActionEvent evt) {
816.         // TODO add your handling code here:
```

```
817.     private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
818.         // TODO add your handling code here:
819.     }
820.
821.     private void txtUserActionPerformed(java.awt.event.ActionEvent
822.     evt) {
823.         // TODO add your handling code here:
824.     }
825.     private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
826.         // TODO add your handling code here:
827.     }
828.
829.     private void txtPassActionPerformed(java.awt.event.ActionEvent
830.     evt) {
831.         // TODO add your handling code here:
832.     }
833.     private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
834.         // TODO add your handling code here:
835.     }
836.
837.     private void txtAgeActionPerformed(java.awt.event.ActionEvent
838.     evt) {
839.         // TODO add your handling code here:
840.     }
841.     private void txtAgeKeyTyped(java.awt.event.KeyEvent evt) {
842.         // TODO add your handling code here:
843.     }
844.
845.     private void
846.     txtIdNewActionPerformed(java.awt.event.ActionEvent evt) {
847.         // TODO add your handling code here:
848.     }
849.     private void txtIdNewKeyTyped(java.awt.event.KeyEvent evt) {
850.         // TODO add your handling code here:
851.     }
852.
853.     private void
854.     txtStateNewActionPerformed(java.awt.event.ActionEvent evt) {
855.         // TODO add your handling code here:
856.     }
857.     private void txtStateNewKeyTyped(java.awt.event.KeyEvent evt)
858.     {
859.         // TODO add your handling code here:
860.     }
861.     private void
862.     txtNameNewActionPerformed(java.awt.event.ActionEvent evt) {
863.         // TODO add your handling code here:
864.     }
865.     private void txtNameNewKeyTyped(java.awt.event.KeyEvent evt) {
866.         if(txtNameNew.getText().length()>=25) evt.consume();
```



```
867.     }
868.
869.     private void
txtEmailNewActionPerformed(java.awt.event.ActionEvent evt) {
870.         // TODO add your handling code here:
871.     }
872.
873.     private void txtEmailNewKeyTyped(java.awt.event.KeyEvent evt)
{
874.         if(txtEmailNew.getText().length()>=25) evt.consume();
875.     }
876.
877.     private void
txtUserNewActionPerformed(java.awt.event.ActionEvent evt) {
878.         // TODO add your handling code here:
879.     }
880.
881.     private void txtUserNewKeyTyped(java.awt.event.KeyEvent evt) {
882.         if(txtUserNew.getText().length()>=12) evt.consume();
883.     }
884.
885.     private void
txtPassNewActionPerformed(java.awt.event.ActionEvent evt) {
886.         // TODO add your handling code here:
887.     }
888.
889.     private void txtPassNewKeyTyped(java.awt.event.KeyEvent evt) {
890.         if(txtPassNew.getText().length()>=12) evt.consume();
891.     }
892.
893.     private void txtRFCActionPerformed(java.awt.event.ActionEvent
evt) {
894.         // TODO add your handling code here:
895.     }
896.
897.     private void txtRFCKeyTyped(java.awt.event.KeyEvent evt) {
898.         // TODO add your handling code here:
899.     }
900.
901.     private void
txtRFCNewActionPerformed(java.awt.event.ActionEvent evt) {
902.         // TODO add your handling code here:
903.     }
904.
905.     private void txtRFCNewKeyTyped(java.awt.event.KeyEvent evt) {
906.         if(txtRFCNew.getText().length()>=13) evt.consume();
907.     }
908.
909.     private void txtAgeNewKeyTyped(java.awt.event.KeyEvent evt) {
910.         // TODO add your handling code here:
911.     }
912.
913.     private void btnSaveActionPerformed(java.awt.event.ActionEvent
evt) {
914.         validateData();
915.         clear();
916.     }
```

```

917.
918.     private void txtRFCNewKeyReleased(java.awt.event.KeyEvent evt)
919.     {
920.         validateRFC();
921.         validateButton();
922.         Mayusculas(txtRFCNew);
923.         iconRFCExist.setForeground(Color.WHITE);
924.         iconRFCChracter.setForeground(Color.WHITE);
925.     }
926.     private void txtUserNewKeyReleased(java.awt.event.KeyEvent
927.     evt) {
928.         validateUser();
929.         validateButton();
930.         iconUserExist.setForeground(Color.WHITE);
931.         iconUserCharacter.setForeground(Color.WHITE);
932.     }
933.     private void txtNameNewKeyReleased(java.awt.event.KeyEvent
934.     evt) {
935.         validateButton();
936.     }
937.     private void txtEmailNewKeyReleased(java.awt.event.KeyEvent
938.     evt) {
939.         check5 = isEmail(txtEmailNew.getText());
940.         if(check5){
941.             iconEmailValidate.setIcon(green);
942.             iconEmailValidate.setText("» Correo Valido");
943.         } else { iconEmailValidate.setIcon(red); }
944.         validateButton();
945.     }
946.     private void txtPassNewKeyReleased(java.awt.event.KeyEvent
947.     evt) {
948.         validateButton();
949.     }
950.     private void txtAgeNewKeyReleased(java.awt.event.KeyEvent evt)
951.     {
952.         validateButton();
953.     }
954.     public void validateData(){
955.         String name = txtNameNew.getText();
956.         String email = txtEmailNew.getText();
957.         String rfc = txtRFCNew.getText().toUpperCase();
958.         String user = txtUserNew.getText().toLowerCase();
959.         String pass = txtPassNew.getText().toLowerCase();
960.         if(check1 == true && check2 == true && check3 == true &&
961.         check4 == true){
962.             int id = (int) (Math.random()*1500+1000);
963.             DateFormat formatter = new SimpleDateFormat("yyyy-MM-
964.             dd");
965.             String age = formatter.format(txtAgeNew.getDate());
966.             methodsSql.ModifyUserData(Login.userEntity.getId(),
967.             name, email, rfc, user, pass, age);

```

```

965.         Login.userEntity = methodsSql.Login(user, pass);
966.         this.setVisible(false);
967.         MenuUser menu = new MenuUser();
968.         menu.setVisible(true);
969.         dispose();
970.     }
971. }
972.
973. public void validateUser(){
974.     if(methodsSql.ValidateUser(txtUserNew.getText())){
975.         iconUserExist.setText(" » Usuario ya existente");
976.         iconUserExist.setIcon(red);
977.         check1 = false;
978.     } else {
979.         iconUserExist.setText(" » Usuario no existente");
980.         iconUserExist.setIcon(green);
981.         check1 = true;
982.     }
983.     if(txtUserNew.getText().length() <= 8){
984.         iconUserCharacter.setText(" » Minimo 8 caracteres");
985.         iconUserCharacter.setIcon(red);
986.         check2 = false;
987.     } else {
988.         iconUserCharacter.setIcon(green);
989.         check2 = true;
990.     }
991.     if(txtUser.getText().equals("")){
992.         iconUserExist.setIcon(red);
993.         iconUserCharacter.setIcon(red);
994.     }
995. }
996.
997. public void validateRFC(){
998.     if(methodsSql.ValidateRFC(txtRFCNew.getText())){
999.         iconRFCExist.setText(" » RFC ya registrado");
1000.         iconRFCExist.setIcon(red);
1001.         check3 = false;
1002.     } else {
1003.         iconRFCExist.setText(" » RFC no registrado");
1004.         iconRFCExist.setIcon(green);
1005.         check3 = true;
1006.     }
1007.     if(txtRFCNew.getText().length() < 12){
1008.         iconRFCChracter.setText(" » Minimo 12
1009. caracteres");
1009.         iconRFCChracter.setIcon(red);
1010.         check4 = false;
1011.     } else {
1012.         iconRFCChracter.setIcon(green);
1013.         check4 = true;
1014.     }
1015.     if(txtRFC.getText().equals("")){
1016.         iconRFCExist.setIcon(red);
1017.         iconRFCChracter.setIcon(red);
1018.     }
1019. }
1020.

```

```

1021.         public void validateButton(){
1022.             if(txtNameNew.getText().equals("") ||
               txtEmailNew.getText().equals("") ||
1023.                 txtRFCNew.getText().equals("") ||
               txtUserNew.getText().equals("") ||
1024.                 txtPassNew.getText().equals("") ||
               txtAgeNew.getDate() == null ||
1025.                 check1 == false || check2 == false ||
1026.                 check3 == false || check4 == false || check5 ==
               false){
1027.                 btnSave.setEnabled(false);
1028.             } else {
1029.                 btnSave.setEnabled(true);
1030.             }
1031.         }
1032.
1033.         public void Mayusculas(javax.swing.JTextField
               jTextfieldS){
1034.             String cadena =
               (txtRFCNew.getText()).toUpperCase();
1035.             txtRFCNew.setText(cadena);
1036.         }
1037.
1038.         public boolean isEmail(String correo) {
1039.             Pattern pattern = Pattern.compile("^([0-9a-zA-
               Z]([_.w]*[0-9a-zA-Z])*@[0-9a-zA-Z]([-w]*[0-9a-zA-Z]
               .)+([a-zA-
               Z]{2,9}.)+[a-zA-Z]{2,3})$");
1040.             Matcher mather = pattern.matcher(correo);
1041.             if(mather.find()) {
1042.                 return true;
1043.             }else{
1044.                 return false;
1045.             }
1046.         }
1047.
1048.         public void clear(){
1049.             txtIdNew.setText(null);
1050.             txtStateNew.setText(null);
1051.             txtNameNew.setText(null);
1052.             txtEmailNew.setText(null);
1053.             txtRFCNew.setText(null);
1054.             txtUserNew.setText(null);
1055.             txtPassNew.setText(null);
1056.             iconUserExist.setIcon(red);
1057.             iconUserCharacter.setIcon(red);
1058.             iconRFCEXist.setIcon(red);
1059.             iconRFCChracter.setIcon(red);
1060.         }
1061.
1062.         /**
1063.          * @param args the command line arguments
1064.          */
1065.         public static void main(String args[]) {
1066.             /* Set the Nimbus look and feel */
1067.             //<editor-fold defaultstate="collapsed" desc="
               Look and feel setting code (optional) ">

```

```

1068.          /* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.
1069.          * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
1070.          */
1071.          try {
1072.              for (javax.swing.UIManager.LookAndFeelInfo
info : javax.swing.UIManager.getInstalledLookAndFeels()) {
1073.                  if ("Nimbus".equals(info.getName())) {
1074.                      javax.swing.UIManager.setLookAndFeel(info.getClassName());
1075.                      break;
1076.                  }
1077.              }
1078.              } catch (ClassNotFoundException ex) {
1079.                  java.util.logging.Logger.getLogger(ModifyUser.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
1080.              } catch (InstantiationException ex) {
1081.                  java.util.logging.Logger.getLogger(ModifyUser.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
1082.              } catch (IllegalAccessException ex) {
1083.                  java.util.logging.Logger.getLogger(ModifyUser.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
1084.              } catch
(javax.swing.UnsupportedLookAndFeelException ex) {
1085.                  java.util.logging.Logger.getLogger(ModifyUser.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
1086.              }
1087.          } //</editor-fold>
1088.
1089.          /* Create and display the form */
1090.          java.awt.EventQueue.invokeLater(new Runnable() {
1091.              public void run() {
1092.                  new ModifyUser().setVisible(true);
1093.              }
1094.          });
1095.      }
1096.
1097.      // Variables declaration - do not modify
1098.      private javax.swing.JLabel UserLabel;
1099.      private javax.swing.JLabel UserLabel1;
1100.      private javax.swing.JLabel UserLabel2;
1101.      private javax.swing.JButton btnSave;
1102.      private javax.swing.JLabel iconEmailValidate;
1103.      private javax.swing.JLabel iconRFCCChracter;
1104.      private javax.swing.JLabel iconRFCEExist;
1105.      private javax.swing.JLabel iconUserCharacter;
1106.      private javax.swing.JLabel iconUserExist;
1107.      private javax.swing.JLabel jLabel1;
1108.      private javax.swing.JLabel jLabel10;
1109.      private javax.swing.JLabel jLabel12;
1110.      private javax.swing.JLabel jLabel13;

```

```

1111.         private javax.swing.JLabel jLabel14;
1112.         private javax.swing.JLabel jLabel15;
1113.         private javax.swing.JLabel jLabel16;
1114.         private javax.swing.JLabel jLabel17;
1115.         private javax.swing.JLabel jLabel18;
1116.         private javax.swing.JLabel jLabel19;
1117.         private javax.swing.JLabel jLabel2;
1118.         private javax.swing.JLabel jLabel3;
1119.         private javax.swing.JLabel jLabel4;
1120.         private javax.swing.JLabel jLabel5;
1121.         private javax.swing.JLabel jLabel6;
1122.         private javax.swing.JLabel jLabel7;
1123.         private javax.swing.JLabel jLabel8;
1124.         private javax.swing.JLabel jLabel9;
1125.         private javax.swing.JPanel jPanel1;
1126.         private javax.swing.JPanel jPanel2;
1127.         private javax.swing.JPanel jPanel3;
1128.         private javax.swing.JPanel jPanel5;
1129.         private javax.swing.JSeparator jSeparator1;
1130.         private javax.swing.JSeparator jSeparator2;
1131.         private javax.swing.JSeparator jSeparator3;
1132.         private javax.swing.JTextField txtAge;
1133.         private com.toedter.calendar.JDateChooser txtAgeNew;
1134.         private javax.swing.JTextField txtEmail;
1135.         private javax.swing.JTextField txtEmailNew;
1136.         private javax.swing.JTextField txtId;
1137.         private javax.swing.JTextField txtIdNew;
1138.         private javax.swing.JTextField txtName;
1139.         private javax.swing.JTextField txtNameNew;
1140.         private javax.swing.JTextField txtPass;
1141.         private javax.swing.JTextField txtPassNew;
1142.         private javax.swing.JTextField txtRFC;
1143.         private javax.swing.JTextField txtRFCNew;
1144.         private javax.swing.JTextField txtState;
1145.         private javax.swing.JTextField txtStateNew;
1146.         private javax.swing.JTextField txtUser;
1147.         private javax.swing.JTextField txtUserNew;
1148.         // End of variables declaration
1149.     }

```

Register

```

1. package view;
2.
3. import code.MethodsSQL;
4. import java.awt.Color;
5. import java.text.DateFormat;
6. import java.text.SimpleDateFormat;
7. import java.util.regex.Matcher;
8. import java.util.regex.Pattern;
9. import javax.swing.ImageIcon;
10. import javax.swing.JOptionPane;
11. import static view.Login.userEntity;
12. import view.user.MenuUser;
13.

```

```

14. public class Register extends javax.swing.JFrame {
15.
16.     MethodsSQL methodsSql = new MethodsSQL();
17.     ImageIcon green = new
ImageIcon(getClass().getResource("/resource/Check-True.png"));
18.     ImageIcon red = new
ImageIcon(getClass().getResource("/resource/Check-False.png"));
19.     private boolean check1, check2, check3, check4, check5;
20.
21.     public Register() {
22.         initComponents();
23.         setTitle("Car Agency");
24.         setLocationRelativeTo(null);
25.     }
26.
27.
28.     /**
29.      * This method is called from within the constructor to
initialize the form.
30.      * WARNING: Do NOT modify this code. The content of this
method is always
31.      * regenerated by the Form Editor.
32.      */
33.     @SuppressWarnings("unchecked")
34.     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
35.     private void initComponents() {
36.
37.         jPanel1 = new javax.swing.JPanel();
38.         jLabel7 = new javax.swing.JLabel();
39.         jPanel2 = new javax.swing.JPanel();
40.         jLabel11 = new javax.swing.JLabel();
41.         txtName = new javax.swing.JTextField();
42.         txtEmail = new javax.swing.JTextField();
43.         jLabel12 = new javax.swing.JLabel();
44.         jLabel13 = new javax.swing.JLabel();
45.         txtUser = new javax.swing.JTextField();
46.         txtPass = new javax.swing.JTextField();
47.         jLabel15 = new javax.swing.JLabel();
48.         jLabel14 = new javax.swing.JLabel();
49.         txtAge = new com.toedter.calendar.JDateChooser();
50.         btnSave = new javax.swing.JButton();
51.         jLabel18 = new javax.swing.JLabel();
52.         txtRFC = new javax.swing.JTextField();
53.         jLabel16 = new javax.swing.JLabel();
54.         jPanel3 = new javax.swing.JPanel();
55.         UserLabel = new javax.swing.JLabel();
56.         jSeparator1 = new javax.swing.JSeparator();
57.         iconUserExist = new javax.swing.JLabel();
58.         iconUserCharacter = new javax.swing.JLabel();
59.         UserLabel1 = new javax.swing.JLabel();
60.         jSeparator2 = new javax.swing.JSeparator();
61.         iconRFCEXist = new javax.swing.JLabel();
62.         iconRFCCharacter = new javax.swing.JLabel();
63.         UserLabel2 = new javax.swing.JLabel();
64.         jSeparator3 = new javax.swing.JSeparator();
65.         iconEmailValidate = new javax.swing.JLabel();

```

```

66.
67.     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
    ;
68.         setResizable(false);
69.
70.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
71.         jPanel1.setPreferredSize(new java.awt.Dimension(850,
640));
72.         jPanel1.setLayout(null);
73.
74.         jLabel7.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
    ); // NOI18N
75.         jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
    {
76.             public void mouseClicked(java.awt.event.MouseEvent
    evt) {
77.                 jLabel7MouseClicked(evt);
78.             }
79.         });
80.         jPanel1.add(jLabel7);
81.         jLabel7.setBounds(0, 0, 72, 66);
82.
83.         jPanel2.setBackground(new java.awt.Color(128, 153, 155));
84.
    jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(""))
    ;
85.         jPanel2.setLayout(null);
86.
87.         jLabel11.setFont(new java.awt.Font("Times New Roman", 0,
    20)); // NOI18N
88.         jLabel11.setForeground(new java.awt.Color(255, 255, 255));
89.         jLabel11.setText(">> NOMBRE");
90.         jPanel2.add(jLabel11);
91.         jLabel11.setBounds(15, 143, 120, 40);
92.
93.         txtName.setBackground(new java.awt.Color(255, 255, 255));
94.         txtName.setFont(new java.awt.Font("Dialog", 0, 18)); //
    NOI18N
95.         txtName.setForeground(new java.awt.Color(0, 0, 0));
96.         txtName.setPreferredSize(new java.awt.Dimension(220, 40));
97.         txtName.addActionListener(new
    java.awt.event.ActionListener() {
98.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
99.                 txtNameActionPerformed(evt);
100.            }
101.        });
102.         txtName.addKeyListener(new java.awt.event.KeyAdapter() {
103.             public void keyReleased(java.awt.event.KeyEvent evt) {
104.                 txtNameKeyReleased(evt);
105.            }
106.             public void keyTyped(java.awt.event.KeyEvent evt) {
107.                 txtNameKeyTyped(evt);
108.            }
109.        });

```



```

110.         jPanel2.add(txtName);
111.         txtName.setBounds(150, 140, 220, 40);
112.
113.         txtEmail.setBackground(new java.awt.Color(255, 255, 255));
114.         txtEmail.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
115.         txtEmail.setForeground(new java.awt.Color(0, 0, 0));
116.         txtEmail.setPreferredSize(new java.awt.Dimension(220,
40));
117.         txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
118.             public void keyReleased(java.awt.event.KeyEvent evt) {
119.                 txtEmailKeyReleased(evt);
120.             }
121.             public void keyTyped(java.awt.event.KeyEvent evt) {
122.                 txtEmailKeyTyped(evt);
123.             }
124.         });
125.         jPanel2.add(txtEmail);
126.         txtEmail.setBounds(150, 200, 220, 40);
127.
128.         jLabel2.setFont(new java.awt.Font("Times New Roman", 0,
20)); // NOI18N
129.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
130.         jLabel2.setText(">> CORREO");
131.         jPanel2.add(jLabel2);
132.         jLabel2.setBounds(15, 203, 120, 40);
133.
134.         jLabel3.setFont(new java.awt.Font("Times New Roman", 0,
20)); // NOI18N
135.         jLabel3.setForeground(new java.awt.Color(255, 255, 255));
136.         jLabel3.setText(">> USUARIO");
137.         jLabel3.setToolTipText("");
138.         jPanel2.add(jLabel3);
139.         jLabel3.setBounds(10, 380, 130, 40);
140.
141.         txtUser.setBackground(new java.awt.Color(255, 255, 255));
142.         txtUser.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
143.         txtUser.setForeground(new java.awt.Color(0, 0, 0));
144.         txtUser.setPreferredSize(new java.awt.Dimension(220, 40));
145.         txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
146.             public void keyReleased(java.awt.event.KeyEvent evt) {
147.                 txtUserKeyReleased(evt);
148.             }
149.             public void keyTyped(java.awt.event.KeyEvent evt) {
150.                 txtUserKeyTyped(evt);
151.             }
152.         });
153.         jPanel2.add(txtUser);
154.         txtUser.setBounds(150, 380, 220, 40);
155.
156.         txtPass.setBackground(new java.awt.Color(255, 255, 255));
157.         txtPass.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
158.         txtPass.setForeground(new java.awt.Color(0, 0, 0));
159.         txtPass.setPreferredSize(new java.awt.Dimension(220, 40));
160.         txtPass.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

161.         public void keyReleased(java.awt.event.KeyEvent evt) {
162.             txtPassKeyReleased(evt);
163.         }
164.         public void keyTyped(java.awt.event.KeyEvent evt) {
165.             txtPassKeyTyped(evt);
166.         }
167.     });
168.     jPanel2.add(txtPass);
169.     txtPass.setBounds(150, 440, 220, 40);
170.
171.     jLabel5.setFont(new java.awt.Font("Times New Roman", 0,
172.         18)); // NOI18N
173.     jLabel5.setForeground(new java.awt.Color(255, 255, 255));
174.     jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
175.     jLabel5.setText(">> CONTRASEÑA");
176.     jPanel2.add(jLabel5);
177.     jLabel5.setBounds(0, 440, 140, 40);
178.
179.     jLabel4.setFont(new java.awt.Font("Times New Roman", 0,
180.         24)); // NOI18N
181.     jLabel4.setForeground(new java.awt.Color(255, 255, 255));
182.     jLabel4.setText(">> Edad");
183.     jPanel2.add(jLabel4);
184.     jLabel4.setBounds(10, 320, 80, 40);
185.
186.     txtAge.setBackground(new java.awt.Color(255, 255, 255));
187.     txtAge.setForeground(new java.awt.Color(0, 0, 0));
188.     txtAge.setToolTipText("");
189.     txtAge.setDateFormatString("yyyy-mm-dd");
190.     txtAge.setFont(new java.awt.Font("Dialog", 0, 18)); //
191.     NOI18N
192.     txtAge.setPreferredSize(new java.awt.Dimension(220, 40));
193.     txtAge.addKeyListener(new java.awt.event.KeyAdapter() {
194.         public void keyReleased(java.awt.event.KeyEvent evt) {
195.             txtAgeKeyReleased(evt);
196.         }
197.         public void keyTyped(java.awt.event.KeyEvent evt) {
198.             txtAgeKeyTyped(evt);
199.         }
200.     });
201.     jPanel2.add(txtAge);
202.     txtAge.setBounds(150, 320, 220, 40);
203.
204.     btnSave.setBackground(new java.awt.Color(255, 255, 255));
205.     btnSave.setFont(new java.awt.Font("Dialog", 1, 18)); //
206.     NOI18N
207.     btnSave.setText("Siguiente");
208.     btnSave.setEnabled(false);
209.     btnSave.setPreferredSize(new java.awt.Dimension(150, 60));
210.     btnSave.addMouseListener(new java.awt.event.MouseAdapter()
211.     {
212.         public void mouseClicked(java.awt.event.MouseEvent
213.         evt) {
214.             btnSaveMouseClicked(evt);
215.         }
216.     });

```

```

211.         btnSave.addActionListener(new
        java.awt.event.ActionListener() {
212.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
213.                 btnSaveActionPerformed(evt);
214.             }
215.         });
216.         jPanel2.add(btnSave);
217.         btnSave.setBounds(110, 510, 150, 60);
218.
219.         jLabel8.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/register-
        menu.png"))); // NOI18N
220.         jPanel2.add(jLabel8);
221.         jLabel8.setBounds(120, 7, 128, 128);
222.
223.         txtRFC.setBackground(new java.awt.Color(255, 255, 255));
224.         txtRFC.setFont(new java.awt.Font("Dialog", 0, 18)); //
        NOI18N
225.         txtRFC.setForeground(new java.awt.Color(0, 0, 0));
226.         txtRFC.setPreferredSize(new java.awt.Dimension(220, 40));
227.         txtRFC.addKeyListener(new java.awt.event.KeyAdapter() {
228.             public void keyReleased(java.awt.event.KeyEvent evt) {
229.                 txtRFCKeyReleased(evt);
230.             }
231.             public void keyTyped(java.awt.event.KeyEvent evt) {
232.                 txtRFCKeyTyped(evt);
233.             }
234.         });
235.         jPanel2.add(txtRFC);
236.         txtRFC.setBounds(150, 260, 220, 40);
237.
238.         jLabel6.setFont(new java.awt.Font("Times New Roman", 0,
        24)); // NOI18N
239.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
240.         jLabel6.setText(">> RFC");
241.         jPanel2.add(jLabel6);
242.         jLabel6.setBounds(10, 260, 130, 40);
243.
244.         jPanel1.add(jPanel2);
245.         jPanel2.setBounds(415, 14, 400, 590);
246.
247.         jPanel3.setBackground(new java.awt.Color(128, 153, 155));
248.         jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(""))
        ;
249.
250.         UserLabel.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
251.         UserLabel.setForeground(new java.awt.Color(255, 255,
        255));
252.         UserLabel.setText("USUARIO");
253.         UserLabel.setFocusable(false);
254.
255.         iconUserExist.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N

```

```
256.         iconUserExist.setForeground(new java.awt.Color(255, 255,
257.         iconUserExist.setIcon(new
258.         javax.swing.ImageIcon(getClass().getResource("/resource/Check-
259.         False.png")))); // NOI18N
260.         iconUserExist.setText(" » Usuario no existente");
261.         iconUserExist.setFocusable(false);
262.         iconUserCharacter.setFont(new java.awt.Font("Dialog", 1,
263.         18)); // NOI18N
264.         iconUserCharacter.setForeground(new java.awt.Color(255,
265.         255, 255));
266.         iconUserCharacter.setIcon(new
267.         javax.swing.ImageIcon(getClass().getResource("/resource/Check-
268.         False.png")))); // NOI18N
269.         iconUserCharacter.setText(" » Minimo 8 caracteres");
270.         iconUserCharacter.setFocusable(false);
271.         UserLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
272.         NOI18N
273.         UserLabel1.setForeground(new java.awt.Color(255, 255,
274.         255));
275.         UserLabel1.setText("RFC");
276.         UserLabel1.setFocusable(false);
277.         iconRFCEXist.setFont(new java.awt.Font("Dialog", 1, 18));
278.         // NOI18N
279.         iconRFCEXist.setForeground(new java.awt.Color(255, 255,
280.         255));
281.         iconRFCEXist.setIcon(new
282.         javax.swing.ImageIcon(getClass().getResource("/resource/Check-
283.         False.png")))); // NOI18N
284.         iconRFCEXist.setText(" » RFC no registrado");
285.         iconRFCEXist.setFocusable(false);
286.         iconRFCCCharacter.setFont(new java.awt.Font("Dialog", 1,
287.         18)); // NOI18N
288.         iconRFCCCharacter.setForeground(new java.awt.Color(255,
289.         255, 255));
290.         iconRFCCCharacter.setIcon(new
291.         javax.swing.ImageIcon(getClass().getResource("/resource/Check-
292.         False.png")))); // NOI18N
293.         iconRFCCCharacter.setText(" » Minimo 12 caracteres");
294.         iconRFCCCharacter.setFocusable(false);
295.         UserLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
296.         NOI18N
297.         UserLabel2.setForeground(new java.awt.Color(255, 255,
298.         255));
299.         UserLabel2.setText("CORREO");
300.         UserLabel2.setFocusable(false);
301.         iconEmailValidate.setFont(new java.awt.Font("Dialog", 1,
302.         18)); // NOI18N
303.         iconEmailValidate.setForeground(new java.awt.Color(255,
304.         255, 255));
```

```

291.         iconEmailValidate.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/resource/Check-
                False.png"))); // NOI18N
292.         iconEmailValidate.setText(" » Correo no valido");
293.         iconEmailValidate.setFocusable(false);
294.
295.         javax.swing.GroupLayout jPanel3Layout = new
            javax.swing.GroupLayout(jPanel3);
296.         jPanel3.setLayout(jPanel3Layout);
297.         jPanel3Layout.setHorizontalGroup(
298.             jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                .LEADING)
299.                 .addGroup(jPanel3Layout.createSequentialGroup()
300.                     .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
                        .Alignment.LEADING)
301.                         .addGroup(jPanel3Layout.createSequentialGroup()
302.                             .addGap(15, 15, 15)
303.                             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
                                .Alignment.LEADING)
304.                                 .addComponent(UserLabel,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 90,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
305.                                 .addComponent(iconUserExist,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
306.                                 .addComponent(iconUserCharacter,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
307.                                 .addComponent(jSeparator1,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
308.                                 .addComponent(UserLabel1)))
309.                         .addGroup(jPanel3Layout.createSequentialGroup()
310.                             .addGap(15, 15, 15)
311.                             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
                                .Alignment.LEADING)
312.                                 .addComponent(iconRFCEXist,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
313.                                 .addComponent(iconRFCCharacter,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)
314.                                 .addComponent(jSeparator2,
                                    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
                                    javax.swing.GroupLayout.PREFERRED_SIZE)))
315.                         .addGroup(jPanel3Layout.createSequentialGroup()
316.                             .addGap(15, 15, 15)
317.                             .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
                                .Alignment.LEADING)

```

```

318.        .addGroup(jPanel3Layout.createSequentialGroup())
319.                .addGap(9, 9, 9)
320.                .addComponent(UserLabel2))
321.                .addComponent(iconEmailValidate,
    javax.swing.GroupLayout.PREFERRED_SIZE, 290,
    javax.swing.GroupLayout.PREFERRED_SIZE)
322.                .addComponent(jSeparator3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
323.                .addContainerGap(13, Short.MAX_VALUE))
324.        );
325.        jPanel3Layout.setVerticalGroup(
326.        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
327.                .addGroup(jPanel3Layout.createSequentialGroup())
328.                .addContainerGap()
329.                .addComponent(UserLabel)
330.                .addGap(2, 2, 2)
331.                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
332.                .addGroup(jPanel3Layout.createSequentialGroup())
333.                .addGap(10, 10, 10)
334.                .addComponent(iconUserExist))
335.                .addComponent(jSeparator1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
336.                .addGap(2, 2, 2)
337.                .addComponent(iconUserCharacter)
338.                .addGap(18, 18, 18)
339.                .addComponent(UserLabel1)
340.                .addGap(2, 2, 2)
341.                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
342.                .addGroup(jPanel3Layout.createSequentialGroup())
343.                .addGap(10, 10, 10)
344.                .addComponent(iconRFCExist))
345.                .addComponent(jSeparator2,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
346.                .addGap(2, 2, 2)
347.                .addComponent(iconRFCCharacter)
348.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
349.                .addComponent(UserLabel2)
350.                .addGap(2, 2, 2)
351.                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
352.                .addGroup(jPanel3Layout.createSequentialGroup())

```

```

353.                .addGap(10, 10, 10)
354.                .addComponent(iconEmailValidate))
355.                .addComponent(jSeparator3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 20,
    javax.swing.GroupLayout.PREFERRED_SIZE))
356.                .addContainerGap(20, Short.MAX_VALUE))
357.            );
358.
359.            jPanell1.add(jPanel3);
360.            jPanel3.setBounds(40, 140, 320, 330);
361.
362.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
363.            getContentPane().setLayout(layout);
364.            layout.setHorizontalGroup(
365.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
366.                .addComponent(jPanell1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
367.            );
368.            layout.setVerticalGroup(
369.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
370.                .addComponent(jPanell1,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
371.            );
372.
373.            pack();
374.        } // </editor-fold>
375.
376.        private void btnSaveMouseClicked(java.awt.event.MouseEvent
    evt) {
377.            // TODO add your handling code here:
378.        }
379.
380.        private void txtNameActionPerformed(java.awt.event.ActionEvent
    evt) {
381.            // TODO add your handling code here:
382.        }
383.
384.        private void jLabel7MouseClicked(java.awt.event.MouseEvent
    evt) {
385.            this.setVisible(false);
386.            Menu menu = new Menu();
387.            menu.setVisible(true);
388.            dispose();
389.        }
390.
391.        private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
392.            int rfc = evt.getKeyChar();
393.            if((rfc<65 || rfc>90) && (rfc<97 || rfc>122) && (rfc<48 ||
    rfc>57)){
394.                evt.consume(); }

```

```

395.         if(txtUser.getText().length()>=12) evt.consume();
396.     }
397.
398.     private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
399.         if(txtPass.getText().length()>=12) evt.consume();
400.     }
401.
402.     private void txtNameKeyTyped(java.awt.event.KeyEvent evt) {
403.         int name = evt.getKeyChar();
404.         if((name<65 || name>90) && (name<97 || name>122) && (name
            != 32)){
405.             evt.consume(); }
406.         if(txtName.getText().length()>=25) evt.consume();
407.     }
408.
409.     private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
410.         int email = evt.getKeyChar();
411.         if((email<48 || email>57) && (email<64 || email>90) &&
412.             (email<97 || email>122) && (email!=45) && (email!=46)
            &&
413.             (email!=95)){ evt.consume(); }
414.         if(txtEmail.getText().length()>=30) evt.consume();
415.     }
416.
417.     private void btnSaveActionPerformed(java.awt.event.ActionEvent
        evt) {
418.         validateData();
419.     }
420.
421.     private void txtAgeKeyTyped(java.awt.event.KeyEvent evt) {
422.         // TODO add your handling code here:
423.     }
424.
425.     private void txtUserKeyReleased(java.awt.event.KeyEvent evt) {
426.         validateUser();
427.         validateButton();
428.         iconUserExist.setForeground(Color.WHITE);
429.         iconUserCharacter.setForeground(Color.WHITE);
430.     }
431.
432.     private void txtRFCKeyTyped(java.awt.event.KeyEvent evt) {
433.         int rfc = evt.getKeyChar();
434.         if((rfc<65 || rfc>90) && (rfc<97 || rfc>122) && (rfc<48 ||
            rfc>57)){
435.             evt.consume(); }
436.         if(txtRFC.getText().length()>=13) evt.consume();
437.     }
438.
439.     private void txtRFCKeyReleased(java.awt.event.KeyEvent evt) {
440.         validateRFC();
441.         validateButton();
442.         Mayusculas(txtRFC);
443.         iconRFCExist.setForeground(Color.WHITE);
444.         iconRFCCharacter.setForeground(Color.WHITE);
445.     }
446.
447.     private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {

```



```

448.         Mayuscula(txtName);
449.         validateButton();
450.     }
451.
452.     private void txtEmailKeyReleased(java.awt.event.KeyEvent evt)
453.     {
454.         check5 = isEmail(txtEmail.getText());
455.         if(check5){
456.             iconEmailValidate.setIcon(green);
457.             iconEmailValidate.setText("» Correo Valido");
458.         } else { iconEmailValidate.setIcon(red); }
459.         validateButton();
460.     }
461.     private void txtPassKeyReleased(java.awt.event.KeyEvent evt) {
462.         validateButton();
463.     }
464.
465.     private void txtAgeKeyReleased(java.awt.event.KeyEvent evt) {
466.         validateButton();
467.     }
468.
469.     public void validateData(){
470.         String name = txtName.getText();
471.         String email = txtEmail.getText();
472.         String rfc = txtRFC.getText().toUpperCase();
473.         String user = txtUser.getText().toLowerCase();
474.         String pass = txtPass.getText().toLowerCase();
475.         if(check1 == true && check2 == true && check3 == true
476.            && check4 == true && check5){
477.             int id = (int) (Math.random()*1500+1000);
478.             DateFormat formatter = new SimpleDateFormat("yyyy-
MM-dd");
479.             String age = formatter.format(txtAge.getDate());
480.             boolean okey = methodsSql.Register(id, 0, name,
email, rfc, user, pass, age);
481.             if(okey){
482.                 userEntity = methodsSql.Login(user, pass);
483.                 this.setVisible(false);
484.                 MenuUser menu = new MenuUser();
485.                 menu.setVisible(true);
486.                 dispose();
487.             } else {
488.                 JOptionPane.showMessageDialog(
489.                     null, "Error al guardar los datos.",
490.                     "", JOptionPane.ERROR_MESSAGE);
491.                 System.exit(0);
492.             }
493.         }
494.     }
495.     public void validateUser(){
496.         if(methodsSql.ValidateUser(txtUser.getText())){
497.             iconUserExist.setText(" » Usuario ya existente");
498.             iconUserExist.setIcon(red);
499.             check1 = false;
500.         } else {

```

```

501.         iconUserExist.setText(" » Usuario no existente");
502.         iconUserExist.setIcon(green);
503.         check1 = true;
504.     }
505.     if(txtUser.getText().length() <= 8){
506.         iconUserCharacter.setText(" » Minimo 8 caracteres");
507.         iconUserCharacter.setIcon(red);
508.         check2 = false;
509.     } else {
510.         iconUserCharacter.setIcon(green);
511.         check2 = true;
512.     }
513.     if(txtUser.getText().equals("")){
514.         iconUserExist.setIcon(red);
515.         iconUserCharacter.setIcon(red);
516.     }
517. }
518.
519. public void validateRFC(){
520.     if(methodsSql.ValidateRFC(txtRFC.getText())){
521.         iconRFCExist.setText(" » RFC ya registrado");
522.         iconRFCExist.setIcon(red);
523.         check3 = false;
524.     } else {
525.         iconRFCExist.setText(" » RFC no registrado");
526.         iconRFCExist.setIcon(green);
527.         check3 = true;
528.     }
529.     if(txtRFC.getText().length() < 12){
530.         iconRFCCCharacter.setText(" » Minimo 12 caracteres");
531.         iconRFCCCharacter.setIcon(red);
532.         check4 = false;
533.     } else {
534.         iconRFCCCharacter.setIcon(green);
535.         check4 = true;
536.     }
537.     if(txtRFC.getText().equals("")){
538.         iconRFCExist.setIcon(red);
539.         iconRFCCCharacter.setIcon(red);
540.     }
541. }
542.
543. public void validateButton(){
544.     if(txtName.getText().equals("") ||
545.        txtEmail.getText().equals("") ||
546.        txtRFC.getText().equals("") || txtUser.getText()
547.        .equals("") ||
548.        txtPass.getText().equals("") || txtAge.getDate() ==
549.        null ||
550.        check1 == false || check2 == false ||
551.        check3 == false || check4 == false || check5 == false){
552.         btnSave.setEnabled(false);
553.     } else {
554.         btnSave.setEnabled(true);
555.     }
556. }

```

```

555.     public void Mayusculas(javax.swing.JTextField jTextFieldS){
556.         String cadena = (txtRFC.getText()).toUpperCase();
557.         txtRFC.setText(cadena);
558.     }
559.
560.     public void Mayuscula(javax.swing.JTextField jTextFieldS){
561.         String cadena = (txtName.getText()).toUpperCase();
562.         txtName.setText(cadena);
563.     }
564.
565.     public boolean isEmail(String correo) {
566.         Pattern pattern = Pattern.compile("^([0-9a-zA-Z]([_\\.w]*[0-9a-zA-Z])*)@([0-9a-zA-Z]([-w]*[0-9a-zA-Z])\\.)+([a-zA-Z]{2,9})\\.([a-zA-Z]{2,3})$");
567.         Matcher mather = pattern.matcher(correo);
568.         if (mather.find()) {
569.             return true;
570.         }else{
571.             return false;
572.         }
573.     }
574.
575.     /**
576.      * @param args the command line arguments
577.      */
578.     public static void main(String args[]) {
579.         /* Set the Nimbus look and feel */
580.         //editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">
581.         /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.
582.          * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
583.          */
584.         try {
585.             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
586.                 if ("Nimbus".equals(info.getName())) {
587.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
588.                     break;
589.                 }
590.             }
591.         } catch (ClassNotFoundException ex) {
592.             java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
593.         } catch (InstantiationException ex) {
594.             java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
595.         } catch (IllegalAccessException ex) {
596.             java.util.logging.Logger.getLogger(Register.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
597.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

598.     java.util.logging.Logger.getLogger(Register.class.getName()).log(ja
va.util.logging.Level.SEVERE, null, ex);
599.     }
600.     //</editor-fold>
601.     //</editor-fold>
602.
603.     /* Create and display the form */
604.     java.awt.EventQueue.invokeLater(new Runnable() {
605.         public void run() {
606.             new Register().setVisible(true);
607.         }
608.     });
609. }
610.
611. // Variables declaration - do not modify
612. private javax.swing.JLabel UserLabel;
613. private javax.swing.JLabel UserLabel1;
614. private javax.swing.JLabel UserLabel2;
615. private javax.swing.JButton btnSave;
616. private javax.swing.JLabel iconEmailValidate;
617. private javax.swing.JLabel iconRFCCharacter;
618. private javax.swing.JLabel iconRFCExist;
619. private javax.swing.JLabel iconUserCharacter;
620. private javax.swing.JLabel iconUserExist;
621. private javax.swing.JLabel jLabel1;
622. private javax.swing.JLabel jLabel2;
623. private javax.swing.JLabel jLabel3;
624. private javax.swing.JLabel jLabel4;
625. private javax.swing.JLabel jLabel5;
626. private javax.swing.JLabel jLabel6;
627. private javax.swing.JLabel jLabel7;
628. private javax.swing.JLabel jLabel8;
629. private javax.swing.JPanel jPanel1;
630. private javax.swing.JPanel jPanel2;
631. private javax.swing.JPanel jPanel3;
632. private javax.swing.JSeparator jSeparator1;
633. private javax.swing.JSeparator jSeparator2;
634. private javax.swing.JSeparator jSeparator3;
635. private com.toedter.calendar.JDateChooser txtAge;
636. private javax.swing.JTextField txtEmail;
637. private javax.swing.JTextField txtName;
638. private javax.swing.JTextField txtPass;
639. private javax.swing.JTextField txtRFC;
640. private javax.swing.JTextField txtUser;
641. // End of variables declaration
642. }

```

Inventory

1. /*
2. * To change this license header, choose License Headers in Project Properties.
3. * To change this template file, choose Tools | Templates
4. * and open the template in the editor.

```

5.  */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.io.File;
10. import javax.swing.ImageIcon;
11. import javax.swing.JFileChooser;
12. import view.Login;
13.
14.
15. /**
16.  *
17.  * @author Restf
18.  */
19. public class Inventory extends javax.swing.JFrame {
20.
21.     MethodsSQL methodsSql = new MethodsSQL();
22.     ImageIcon green = new
ImageIcon(getClass().getResource("/resource/Check-True.png"));
23.     ImageIcon red = new
ImageIcon(getClass().getResource("/resource/Check-False.png"));
24.     public Inventory() {
25.         initComponents();
26.         setTitle("Car Agency - Inventario: [
"+Login.userEntity.getName()+" ]");
27.         setLocationRelativeTo(null);
28.         TableData.setModel(methodsSql.loadTableInventory());
29.     }
30.
31.     /**
32.      * This method is called from within the constructor to
initialize the form.
33.      * WARNING: Do NOT modify this code. The content of this
method is always
34.      * regenerated by the Form Editor.
35.      */
36.     @SuppressWarnings("unchecked")
37.     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
38.     private void initComponents() {
39.
40.         jPanel1 = new javax.swing.JPanel();
41.         jLabel7 = new javax.swing.JLabel();
42.         jScrollPane1 = new javax.swing.JScrollPane();
43.         TableData = new javax.swing.JTable();
44.         btnSelectData = new javax.swing.JComboBox<>();
45.         txtDataFilter = new javax.swing.JTextField();
46.         jLabel11 = new javax.swing.JLabel();
47.         jButton1 = new javax.swing.JButton();
48.
49.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
;
50.         setResizable(false);
51.
52.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
53.         jPanel1.setMinimumSize(new java.awt.Dimension(1280, 720));

```

```

54.
55.         jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
56.         jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
); // NOI18N
57.         jLabel7.setText("VOLVER");
58.         jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
{
59.             public void mouseClicked(java.awt.event.MouseEvent
evt) {
60.                 jLabel7MouseClicked(evt);
61.             }
62.         });
63.
64.         TableData.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
65.         TableData.setModel(new
javax.swing.table.DefaultTableModel(
66.             new Object [][] {
67.                 {null, "", null, null, null, null, null, null,
null, null, null},
68.                 {null, null, null, null, null, null, null, null,
null, null, null},
69.                 {null, null, null, null, null, null, null, null,
null, null, null},
70.                 {null, null, null, null, null, null, null, null,
null, null, null}
71.             },
72.             new String [] {
73.                 "ID", "COMPAÑIA", "MODELO", "N° SERIE", "COLOR",
"ESTADO", "PRECIO", "PAIS", "HP", "N° REGISTRO", "KILOMETRAJE"
74.             }
75.         ) {
76.             boolean[] canEdit = new boolean [] {
77.                 false, false, false, false, false, false, false,
false, false, false, false
78.             };
79.
80.             public boolean isCellEditable(int rowIndex, int
columnIndex) {
81.                 return canEdit [columnIndex];
82.             }
83.         });
84.         jScrollPane1.setViewportView(TableData);
85.
86.         btnSelectData.setFont(new java.awt.Font("Dialog", 1, 16));
// NOI18N
87.         btnSelectData.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "SELECCIONAR", ".•
Identificador", ".• Compañia", ".• IModelo", ".• N° Serie", ".• Color",
".• Estado", ".• Precio", ".• Pais", ".• Horsepower", ".• N° Registro",
".• Kilometraje" }));
88.         btnSelectData.setPreferredSize(new java.awt.Dimension(220,
40));
89.         btnSelectData.addActionListener(new
java.awt.event.ActionListener() {

```

```

90.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
91.             btnSelectDataActionPerformed(evt);
92.         }
93.     });
94.
95.     txtDataFilter.setFont(new java.awt.Font("Dialog", 0, 16));
    // NOI18N
96.     txtDataFilter.setEnabled(false);
97.     txtDataFilter.setPreferredSize(new java.awt.Dimension(220,
    40));
98.     txtDataFilter.addKeyListener(new
    java.awt.event.KeyAdapter() {
99.         public void keyReleased(java.awt.event.KeyEvent evt) {
100.             txtDataFilterKeyReleased(evt);
101.         }
102.         public void keyTyped(java.awt.event.KeyEvent evt) {
103.             txtDataFilterKeyTyped(evt);
104.         }
105.     });
106.
107.     jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
108.     jLabel1.setForeground(new java.awt.Color(255, 255, 255));
109.     jLabel1.setText("> Selecciona el dato a buscar:");
110.
111.     jButton1.setText("Generar PDF");
112.     jButton1.addActionListener(new
    java.awt.event.ActionListener() {
113.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
114.             jButton1ActionPerformed(evt);
115.         }
116.     });
117.
118.     javax.swing.GroupLayout jPanel1Layout = new
    javax.swing.GroupLayout(jPanel1);
119.     jPanel1.setLayout(jPanel1Layout);
120.     jPanel1Layout.setHorizontalGroup(
121.         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
122.             .addGroup(jPanel1Layout.createSequentialGroup()
123.                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing
    .GroupLayout
    .Alignment.LEADING)
124.                     .addGroup(jPanel1Layout.createSequentialGroup()
125.                         .addContainerGap()
126.                         .addComponent(jLabel7)
127.                         .addGap(37, 37, 37)
128.                         .addComponent(jLabel1)
129.                         .addPreferredGap(javax.swing.LayoutStyle
    .ComponentPlacement.UNRELAT
    ED)
130.                         .addComponent(btnSelectData,
    javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
131.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
            ED)
132.        .addComponent(txtDataFilter,
            javax.swing.GroupLayout.PREFERRED_SIZE, 328,
            javax.swing.GroupLayout.PREFERRED_SIZE)
133.        .addGap(31, 31, 31)
134.        .addComponent(jButton1))
135.
        .addGroup(jPanell1Layout.createSequentialGroup())
136.        .addGap(39, 39, 39)
137.        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 1200,
            javax.swing.GroupLayout.PREFERRED_SIZE)))
138.        .addContainerGap(41, Short.MAX_VALUE))
139.    );
140.    jPanell1Layout.setVerticalGroup(
141.        jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
            .LEADING)
142.        .addGroup(jPanell1Layout.createSequentialGroup())
143.        .addContainerGap())
144.
        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
            .Alignment.BASELINE)
145.        .addComponent(jLabel7,
            javax.swing.GroupLayout.PREFERRED_SIZE, 66,
            javax.swing.GroupLayout.PREFERRED_SIZE)
146.        .addComponent(jLabel1)
147.        .addComponent(btnSelectData,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
148.        .addComponent(txtDataFilter,
            javax.swing.GroupLayout.PREFERRED_SIZE, 41,
            javax.swing.GroupLayout.PREFERRED_SIZE)
149.        .addComponent(jButton1))
150.        .addGap(18, 18, 18)
151.        .addComponent(jScrollPane1,
            javax.swing.GroupLayout.PREFERRED_SIZE, 611,
            javax.swing.GroupLayout.PREFERRED_SIZE)
152.        .addContainerGap(19, Short.MAX_VALUE))
153.    );
154.
155.    javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
156.    getContentPane().setLayout(layout);
157.    layout.setHorizontalGroup(
158.        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
159.        .addGroup(layout.createSequentialGroup()
160.        .addComponent(jPanell1,
            javax.swing.GroupLayout.PREFERRED_SIZE,

```



```

        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
161.            .addGap(0, 0, Short.MAX_VALUE))
162.        );
163.        layout.setVerticalGroup(
164.            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
165.                .addGroup(layout.createSequentialGroup()
166.                    .addComponent(jPanell,
167.                        javax.swing.GroupLayout.PREFERRED_SIZE,
168.                        javax.swing.GroupLayout.DEFAULT_SIZE,
169.                        javax.swing.GroupLayout.PREFERRED_SIZE)
170.                    .addGap(0, 0, Short.MAX_VALUE))
171.                );
172.        pack();
173.    } // </editor-fold>
174.
175.    private void jLabel7MouseClicked(java.awt.event.MouseEvent
176.        evt) {
177.        this.setVisible(false);
178.        MenuAdmin menu = new MenuAdmin();
179.        menu.setVisible(true);
180.        dispose();
181.    }
182.
183.    private void
184.        btnSelectDataActionPerformed(java.awt.event.ActionEvent evt) {
185.        if (btnSelectData.getSelectedIndex() > 0) {
186.            txtDataFilter.setEnabled(true);
187.        } else {
188.            txtDataFilter.setEnabled(false);
189.        }
190.        clear();
191.    }
192.
193.    private void txtDataFilterKeyReleased(java.awt.event.KeyEvent
194.        evt) {
195.        MayusculasRFC(txtDataFilter);
196.    }
197.
198.    TableData.setModel(methodsSql.loadTableInventoryFilter(btnSelectDat
199.        a.getSelectedIndex(), txtDataFilter.getText()));
200.    }
201.
202.    private void txtDataFilterKeyTyped(java.awt.event.KeyEvent
203.        evt) {
204.        // TODO add your handling code here:
205.    }
206.
207.    private void
208.        jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
209.        String url = null;
210.        JFileChooser dgl= new JFileChooser();
211.        int o=dgl.showSaveDialog(this);
212.        if(o== JFileChooser.APPROVE_OPTION){
213.            File f= dgl.getSelectedFile();

```

```

204.         url =f.toString();
205.     }
206.     methodsSql.pdfCreateTableInventory(url,
    btnSelectData.getSelectedIndex(), txtDataFilter.getText());
207. }
208.
209.     public void MayusculasRFC(javax.swing.JTextField jTextFieldS){
210.         String cadena = (txtDataFilter.getText()).toUpperCase();
211.         txtDataFilter.setText(cadena);
212.     }
213.
214.     public void clear(){
215.         txtDataFilter.setText(null);
216.         TableData.setModel(methodsSql.loadTableInventory());
217.     }
218.
219.     /**
220.      * @param args the command line arguments
221.      */
222.     public static void main(String args[]) {
223.         /* Set the Nimbus look and feel */
224.         //

```

```

246.         /* Create and display the form */
247.         java.awt.EventQueue.invokeLater(new Runnable() {
248.             public void run() {
249.                 new Inventory().setVisible(true);
250.             }
251.         });
252.     }
253.
254.     // Variables declaration - do not modify
255.     private javax.swing.JTable TableData;
256.     private javax.swing.JComboBox<String> btnSelectData;
257.     private javax.swing.JButton jButton1;
258.     private javax.swing.JLabel jLabel1;
259.     private javax.swing.JLabel jLabel7;
260.     private javax.swing.JPanel jPanel1;
261.     private javax.swing.JScrollPane jScrollPane1;
262.     private javax.swing.JTextField txtDataFilter;
263.     // End of variables declaration
264. }

```

MenuAdmin

PanelAdd

```

1.  /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.awt.Color;
10. import java.text.DateFormat;
11. import java.text.SimpleDateFormat;
12. import java.util.regex.Matcher;
13. import java.util.regex.Pattern;
14. import javax.swing.ImageIcon;
15.
16. /**
17.  *
18.  * @author Restf
19.  */
20. public class PanelAdd extends javax.swing.JPanel {
21.
22.     MethodsSQL methodsSql = new MethodsSQL();
23.     ImageIcon green = new
  ImageIcon(getClass().getResource("/resource/Check-True.png"));
24.     ImageIcon red = new
  ImageIcon(getClass().getResource("/resource/Check-False.png"));
25.     private boolean check1, check2, check3, check4, check5;
26.
27.     public PanelAdd() {
28.         initComponents();
29.         btnSave.setEnabled(false);

```

```

30.     }
31.
32.     /**
33.      * This method is called from within the constructor to
        initialize the form.
34.      * WARNING: Do NOT modify this code. The content of this
        method is always
35.      * regenerated by the Form Editor.
36.      */
37.     @SuppressWarnings("unchecked")
38.     // <editor-fold defaultstate="collapsed" desc="Generated
        Code">
39.     private void initComponents() {
40.
41.         jPanel1 = new javax.swing.JPanel();
42.         jPanel2 = new javax.swing.JPanel();
43.         jLabel1 = new javax.swing.JLabel();
44.         txtEmail = new javax.swing.JTextField();
45.         jLabel2 = new javax.swing.JLabel();
46.         jLabel3 = new javax.swing.JLabel();
47.         txtUser = new javax.swing.JTextField();
48.         txtPass = new javax.swing.JTextField();
49.         jLabel5 = new javax.swing.JLabel();
50.         txtAge = new com.toedter.calendar.JDateChooser();
51.         jLabel4 = new javax.swing.JLabel();
52.         jLabel6 = new javax.swing.JLabel();
53.         txtState = new javax.swing.JComboBox<>();
54.         txtName = new javax.swing.JTextField();
55.         txtRFC = new javax.swing.JTextField();
56.         jLabel8 = new javax.swing.JLabel();
57.         jPanel3 = new javax.swing.JPanel();
58.         btnRemove = new javax.swing.JButton();
59.         btnSave = new javax.swing.JButton();
60.         iconUserExist = new javax.swing.JLabel();
61.         iconUserCharacter = new javax.swing.JLabel();
62.         UserLabel = new javax.swing.JLabel();
63.         jSeparator2 = new javax.swing.JSeparator();
64.         iconRFCEXist = new javax.swing.JLabel();
65.         iconRFCCharacter = new javax.swing.JLabel();
66.         jSeparator3 = new javax.swing.JSeparator();
67.         iconEmailValidate = new javax.swing.JLabel();
68.         jLabel7 = new javax.swing.JLabel();
69.
70.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
71.         jPanel1.setPreferredSize(new java.awt.Dimension(982,
        641));
72.         jPanel1.setLayout(null);
73.
74.         jPanel2.setBackground(new java.awt.Color(128, 153, 155));
75.
        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null
        , "INSERTAR DATOS DE USUARIO",
        javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
        javax.swing.border.TitledBorder.DEFAULT_POSITION, new
        java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
        255))); // NOI18N
76.

```

```

77.         jLabel1.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
78.         jLabel1.setForeground(new java.awt.Color(255, 255, 255));
79.         jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
80.         jLabel1.setText(">> Nombre");
81.
82.         txtEmail.setBackground(new java.awt.Color(255, 255, 255));
83.         txtEmail.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
84.         txtEmail.setForeground(new java.awt.Color(0, 0, 0));
85.         txtEmail.setPreferredSize(new java.awt.Dimension(220,
40));
86.         txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
87.             public void keyReleased(java.awt.event.KeyEvent evt) {
88.                 txtEmailKeyReleased(evt);
89.             }
90.             public void keyTyped(java.awt.event.KeyEvent evt) {
91.                 txtEmailKeyTyped(evt);
92.             }
93.         });
94.
95.         jLabel2.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
96.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
97.         jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
98.         jLabel2.setText(">> Correo");
99.
100.        jLabel3.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
101.        jLabel3.setForeground(new java.awt.Color(255, 255, 255));
102.        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
103.        jLabel3.setText(">> Usuario");
104.        jLabel3.setToolTipText("");
105.
106.        txtUser.setBackground(new java.awt.Color(255, 255, 255));
107.        txtUser.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
108.        txtUser.setForeground(new java.awt.Color(0, 0, 0));
109.        txtUser.setPreferredSize(new java.awt.Dimension(220, 40));
110.        txtUser.addActionListener(new
java.awt.event.ActionListener() {
111.            public void actionPerformed(java.awt.event.ActionEvent
evt) {
112.                txtUserActionPerformed(evt);
113.            }
114.        });
115.        txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
116.            public void keyReleased(java.awt.event.KeyEvent evt) {
117.                txtUserKeyReleased(evt);
118.            }
119.            public void keyTyped(java.awt.event.KeyEvent evt) {
120.                txtUserKeyTyped(evt);
121.            }
122.        });

```

```

123.
124.         txtPass.setBackground(new java.awt.Color(255, 255, 255));
125.         txtPass.setFont(new java.awt.Font("Dialog", 0, 16)); //
    NOI18N
126.         txtPass.setForeground(new java.awt.Color(0, 0, 0));
127.         txtPass.setPreferredSize(new java.awt.Dimension(220, 40));
128.         txtPass.addKeyListener(new java.awt.event.KeyAdapter() {
129.             public void keyReleased(java.awt.event.KeyEvent evt) {
130.                 txtPassKeyReleased(evt);
131.             }
132.             public void keyTyped(java.awt.event.KeyEvent evt) {
133.                 txtPassKeyTyped(evt);
134.             }
135.         });
136.
137.         jLabel5.setFont(new java.awt.Font("Times New Roman", 0,
    24)); // NOI18N
138.         jLabel5.setForeground(new java.awt.Color(255, 255, 255));
139.         jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
140.         jLabel5.setText(">> Contraseña");
141.
142.         txtAge.setBackground(new java.awt.Color(255, 255, 255));
143.         txtAge.setForeground(new java.awt.Color(0, 0, 0));
144.         txtAge.setToolTipText("");
145.         txtAge.setDateFormatString("yyyy-mm-dd");
146.         txtAge.setFont(new java.awt.Font("Dialog", 0, 16)); //
    NOI18N
147.         txtAge.setPreferredSize(new java.awt.Dimension(220, 40));
148.         txtAge.addKeyListener(new java.awt.event.KeyAdapter() {
149.             public void keyReleased(java.awt.event.KeyEvent evt) {
150.                 txtAgeKeyReleased(evt);
151.             }
152.             public void keyTyped(java.awt.event.KeyEvent evt) {
153.                 txtAgeKeyTyped(evt);
154.             }
155.         });
156.
157.         jLabel4.setFont(new java.awt.Font("Times New Roman", 0,
    24)); // NOI18N
158.         jLabel4.setForeground(new java.awt.Color(255, 255, 255));
159.         jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
160.         jLabel4.setText(">> Edad");
161.
162.         jLabel6.setFont(new java.awt.Font("Times New Roman", 0,
    24)); // NOI18N
163.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
164.         jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
165.         jLabel6.setText(">> Estado");
166.
167.         txtState.setBackground(new java.awt.Color(255, 255, 255));
168.         txtState.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
169.         txtState.setForeground(new java.awt.Color(0, 0, 0));
170.         txtState.setMaximumRowCount(2);

```

```

171.         txtState.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Usuario",
"Administrador" }));
172.
173.         txtName.setBackground(new java.awt.Color(255, 255, 255));
174.         txtName.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
175.         txtName.setForeground(new java.awt.Color(0, 0, 0));
176.         txtName.setPreferredSize(new java.awt.Dimension(220, 40));
177.         txtName.addKeyListener(new java.awt.event.KeyAdapter() {
178.             public void keyReleased(java.awt.event.KeyEvent evt) {
179.                 txtNameKeyReleased(evt);
180.             }
181.             public void keyTyped(java.awt.event.KeyEvent evt) {
182.                 txtNameKeyTyped(evt);
183.             }
184.         });
185.
186.         txtRFC.setBackground(new java.awt.Color(255, 255, 255));
187.         txtRFC.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
188.         txtRFC.setForeground(new java.awt.Color(0, 0, 0));
189.         txtRFC.setPreferredSize(new java.awt.Dimension(220, 40));
190.         txtRFC.addKeyListener(new java.awt.event.KeyAdapter() {
191.             public void keyReleased(java.awt.event.KeyEvent evt) {
192.                 txtRFCKeyReleased(evt);
193.             }
194.             public void keyTyped(java.awt.event.KeyEvent evt) {
195.                 txtRFCKeyTyped(evt);
196.             }
197.         });
198.
199.         jLabel8.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
200.         jLabel8.setForeground(new java.awt.Color(255, 255, 255));
201.
202.         jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
203.         jLabel8.setText(">> RFC");
204.         javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
205.         jPanel2.setLayout(jPanel2Layout);
206.         jPanel2Layout.setHorizontalGroup(
207.             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
208.                 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
209.                     .addGap(20, 20, 20)
210.                     .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.TRAILING)
211.                         .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup()

```

```

212.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
213.                .addComponent(jLabel2,
        javax.swing.GroupLayout.PREFERRED_SIZE, 95,
        javax.swing.GroupLayout.PREFERRED_SIZE)
214.                .addComponent(jLabel1,
        javax.swing.GroupLayout.PREFERRED_SIZE, 110,
        javax.swing.GroupLayout.PREFERRED_SIZE)
215.                .addComponent(jLabel6,
        javax.swing.GroupLayout.PREFERRED_SIZE, 110,
        javax.swing.GroupLayout.PREFERRED_SIZE))
216.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
217.        .addGroup(jPanel2Layout.createSequentialGroup())
218.                .addGap(46, 46, 46)
219.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
220.                .addComponent(txtState, 0,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
221.                .addComponent(txtName,
        javax.swing.GroupLayout.Alignment.TRAILING,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
222.        .addGroup(jPanel2Layout.createSequentialGroup())
223.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
224.                .addComponent(txtEmail,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
225.        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
        jPanel2Layout.createSequentialGroup())
226.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.TRAILING, false)
227.        .addGroup(jPanel2Layout.createSequentialGroup())
228.                .addComponent(jLabel3)
229.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
230.                .addComponent(txtUser,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE))
231.        .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
        jPanel2Layout.createSequentialGroup())
232.                .addComponent(jLabel4)
233.                .addGap(88, 88, 88)

```



```

234.                .addComponent(txtAge,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
235.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
    jPanel2Layout.createSequentialGroup())
236.                .addComponent(jLabel5,
    javax.swing.GroupLayout.PREFERRED_SIZE, 138,
    javax.swing.GroupLayout.PREFERRED_SIZE)
237.                .addGap(18, 18, 18)
238.                .addComponent(txtPass,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
239.                .addGap(0, 0, Short.MAX_VALUE))
240.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
    jPanel2Layout.createSequentialGroup())
241.                .addComponent(jLabel8,
    javax.swing.GroupLayout.PREFERRED_SIZE, 95,
    javax.swing.GroupLayout.PREFERRED_SIZE)
242.                .addGap(61, 61, 61)
243.                .addComponent(txtRFC,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
244.                .addGap(46, 46, 46))
245.        );
246.        jPanel2Layout.setVerticalGroup(
247.        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
248.                .addGroup(jPanel2Layout.createSequentialGroup())
249.                .addGap(35, 35, 35)
250.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
251.                .addComponent(jLabel6,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE)
252.                .addComponent(txtState,
    javax.swing.GroupLayout.PREFERRED_SIZE, 36,
    javax.swing.GroupLayout.PREFERRED_SIZE))
253.                .addGap(23, 23, 23)
254.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
255.                .addComponent(jLabel1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE)
256.                .addComponent(txtName,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
257.                .addGap(20, 20, 20)
258.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.TRAILING)

```

```

259.                .addComponent(jLabel2,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE)
260.                .addComponent(txtEmail,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
261.                .addGap(18, 18, 18)
262.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
263.                .addComponent(txtRFC,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
264.                .addComponent(jLabel8,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
265.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
266.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
267.                .addComponent(txtAge,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
268.                .addComponent(jLabel4,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
269.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
270.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
271.                .addComponent(txtUser,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
272.                .addComponent(jLabel3,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
273.                .addGap(20, 20, 20)
274.
    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
275.                .addComponent(txtPass,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
276.                .addComponent(jLabel5,
    javax.swing.GroupLayout.PREFERRED_SIZE, 41,
    javax.swing.GroupLayout.PREFERRED_SIZE))
277.                .addGap(72, 72, 72))
278.    );

```

```

279.
280.         jPanel1.add(jPanel2);
281.         jPanel2.setBounds(40, 82, 452, 500);
282.
283.         jPanel3.setBackground(new java.awt.Color(128, 153, 155));
284.
        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null
        , "OPCIONES",
        javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
        javax.swing.border.TitledBorder.DEFAULT_POSITION, new
        java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
        255))); // NOI18N
285.         jPanel3.setLayout(null);
286.
287.         btnRemove.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
288.         btnRemove.setForeground(new java.awt.Color(255, 255,
        255));
289.         btnRemove.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/remove-
        128px.png"))); // NOI18N
290.         btnRemove.setText("LIMPIAR DATOS");
291.         btnRemove.setContentAreaFilled(false);
292.
        btnRemove.setHorizontalTextPosition(javax.swing.SwingConstants.CENT
        ER);
293.
        btnRemove.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM
        );
294.         btnRemove.addActionListener(new
        java.awt.event.ActionListener() {
295.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
296.                 btnRemoveActionPerformed(evt);
297.             }
298.         });
299.         jPanel3.add(btnRemove);
300.         btnRemove.setBounds(217, 301, 185, 172);
301.
302.         btnSave.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
303.         btnSave.setForeground(new java.awt.Color(255, 255, 255));
304.         btnSave.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/ready-
        128px.png"))); // NOI18N
305.         btnSave.setText("GUARDAR DATOS");
306.         btnSave.setContentAreaFilled(false);
307.
        btnSave.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER
        );
308.
        btnSave.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
309.         btnSave.addActionListener(new
        java.awt.event.ActionListener() {
310.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
311.                 btnSaveActionPerformed(evt);

```

```

312.         }
313.     });
314.     jPanel3.add(btnSave);
315.     btnSave.setBounds(21, 301, 190, 172);
316.
317.     iconUserExist.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
318.     iconUserExist.setForeground(new java.awt.Color(255, 255,
    255));
319.     iconUserExist.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
    False.png"))); // NOI18N
320.     iconUserExist.setText(" » Usuario no existente");
321.     iconUserExist.setFocusable(false);
322.     jPanel3.add(iconUserExist);
323.     iconUserExist.setBounds(60, 80, 290, 32);
324.
325.     iconUserCharacter.setFont(new java.awt.Font("Dialog", 1,
    18)); // NOI18N
326.     iconUserCharacter.setForeground(new java.awt.Color(255,
    255, 255));
327.     iconUserCharacter.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
    False.png"))); // NOI18N
328.     iconUserCharacter.setText(" » Minimo 8 caracteres");
329.     iconUserCharacter.setFocusable(false);
330.     jPanel3.add(iconUserCharacter);
331.     iconUserCharacter.setBounds(60, 120, 290, 32);
332.
333.     UserLabel.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
334.     UserLabel.setForeground(new java.awt.Color(255, 255,
    255));
335.     UserLabel.setText("USUARIO");
336.     UserLabel.setFocusable(false);
337.     jPanel3.add(UserLabel);
338.     UserLabel.setBounds(60, 40, 90, 24);
339.     jPanel3.add(jSeparator2);
340.     jSeparator2.setBounds(60, 190, 0, 2);
341.
342.     iconRFCExist.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
343.     iconRFCExist.setForeground(new java.awt.Color(255, 255,
    255));
344.     iconRFCExist.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
    False.png"))); // NOI18N
345.     iconRFCExist.setText(" » RFC no registrado");
346.     iconRFCExist.setFocusable(false);
347.     jPanel3.add(iconRFCExist);
348.     iconRFCExist.setBounds(60, 160, 214, 32);
349.
350.     iconRFCCharacter.setFont(new java.awt.Font("Dialog", 1,
    18)); // NOI18N
351.     iconRFCCharacter.setForeground(new java.awt.Color(255,
    255, 255));

```

```

352.         iconRFCCharacter.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/resource/Check-
                False.png"))); // NOI18N
353.         iconRFCCharacter.setText(" » Minimo 12 caracteres");
354.         iconRFCCharacter.setFocusable(false);
355.         jPanel3.add(iconRFCCharacter);
356.         iconRFCCharacter.setBounds(60, 200, 242, 32);
357.         jPanel3.add(jSeparator3);
358.         jSeparator3.setBounds(60, 70, 272, 20);
359.
360.         iconEmailValidate.setFont(new java.awt.Font("Dialog", 1,
            18)); // NOI18N
361.         iconEmailValidate.setForeground(new java.awt.Color(255,
            255, 255));
362.         iconEmailValidate.setIcon(new
            javax.swing.ImageIcon(getClass().getResource("/resource/Check-
                False.png"))); // NOI18N
363.         iconEmailValidate.setText(" » Correo no valido");
364.         iconEmailValidate.setFocusable(false);
365.         jPanel3.add(iconEmailValidate);
366.         iconEmailValidate.setBounds(60, 240, 201, 32);
367.
368.         jPanel1.add(jPanel3);
369.         jPanel3.setBounds(503, 77, 430, 500);
370.
371.         jLabel7.setFont(new java.awt.Font("Dialog", 1, 36)); //
            NOI18N
372.         jLabel7.setForeground(new java.awt.Color(255, 255, 255));
373.         jLabel7.setText("INGRESAR DE USUARIOS");
374.         jPanel1.add(jLabel7);
375.         jLabel7.setBounds(265, 0, 464, 58);
376.
377.         javax.swing.GroupLayout layout = new
            javax.swing.GroupLayout(this);
378.         this.setLayout(layout);
379.         layout.setHorizontalGroup(
380.             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
381.                 .addComponent(jPanel1,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 996, Short.MAX_VALUE)
382.                 );
383.         layout.setVerticalGroup(
384.             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
385.                 .addComponent(jPanel1,
                    javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
386.                 );
387.     } // </editor-fold>
388.
389.     private void btnSaveActionPerformed(java.awt.event.ActionEvent
        evt) {
390.         validateData();
391.     }
392.

```

```
393.     private void
      btnRemoveActionPerformed(java.awt.event.ActionEvent evt) {
394.         clear();
395.     }
396.
397.     private void txtRFCKeyTyped(java.awt.event.KeyEvent evt) {
398.         if(txtRFC.getText().length()>=13) evt.consume();
399.     }
400.
401.     private void txtRFCKeyReleased(java.awt.event.KeyEvent evt) {
402.         validateRFC();
403.         validateButton();
404.         Mayusculas(txtRFC);
405.         iconRFCExist.setForeground(Color.WHITE);
406.         iconRFCCharacter.setForeground(Color.WHITE);
407.     }
408.
409.     private void txtNameKeyTyped(java.awt.event.KeyEvent evt) {
410.         int name = evt.getKeyChar();
411.         if((name<65 || name>90) && (name<97 || name>122) && (name
            != 32)){
412.             evt.consume(); }
413.         if(txtName.getText().length()>=25) evt.consume();
414.     }
415.
416.     private void txtNameKeyReleased(java.awt.event.KeyEvent evt) {
417.         validateButton();
418.     }
419.
420.     private void txtAgeKeyTyped(java.awt.event.KeyEvent evt) {
421.         txtAge.setBackground(Color.WHITE);
422.     }
423.
424.     private void txtAgeKeyReleased(java.awt.event.KeyEvent evt) {
425.         validateButton();
426.     }
427.
428.     private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
429.         if(txtPass.getText().length()>=12) evt.consume();
430.     }
431.
432.     private void txtPassKeyReleased(java.awt.event.KeyEvent evt) {
433.         validateButton();
434.     }
435.
436.     private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
437.         if(txtUser.getText().length()>=12) evt.consume();
438.     }
439.
440.     private void txtUserKeyReleased(java.awt.event.KeyEvent evt) {
441.         validateUser();
442.         validateButton();
443.         iconUserExist.setForeground(Color.WHITE);
444.         iconUserCharacter.setForeground(Color.WHITE);
445.     }
446.
```

```

447.     private void txtUserActionPerformed(java.awt.event.ActionEvent
    evt) {
448.         // TODO add your handling code here:
449.     }
450.
451.     private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
452.         int c = evt.getKeyChar();
453.         if((c<48 || c>57) && (c<64 || c>90) &&
454.             (c<97 || c>122) && (c!=45) && (c!=46) &&
455.             (c!=95)){ evt.consume(); }
456.         int characterLimit=25;
457.         if(txtEmail.getText().length()>=characterLimit){
458.             evt.consume();
459.         }
460.     }
461.
462.     private void txtEmailKeyReleased(java.awt.event.KeyEvent evt)
    {
463.         check5 = isEmail(txtEmail.getText());
464.         if(check5){
465.             iconEmailValidate.setIcon(green);
466.             iconEmailValidate.setText(">> Correo Valido");
467.         } else { iconEmailValidate.setIcon(red); }
468.         validateButton();
469.     }
470.
471.     public void clear(){
472.         txtName.setText(null);
473.         txtEmail.setText(null);
474.         txtRFC.setText(null);
475.         txtUser.setText(null);
476.         txtPass.setText(null);
477.         txtAge.setDate(null);
478.         iconUserExist.setForeground(Color.WHITE);
479.         iconUserCharacter.setForeground(Color.WHITE);
480.         iconUserExist.setIcon(red);
481.         iconUserCharacter.setIcon(red);
482.         iconRFCExist.setIcon(red);
483.         iconRFCCharacter.setIcon(red);
484.         iconUserExist.setText(" » Usuario no existente");
485.         iconRFCExist.setText(" » RFC no registrado");
486.         iconEmailValidate.setIcon(red);
487.         iconEmailValidate.setText(" » Correo no valido");
488.         btnSave.setEnabled(false);
489.     }
490.
491.     public void validateData(){
492.         int state = txtState.getSelectedIndex();
493.         String name = txtName.getText();
494.         String email = txtEmail.getText();
495.         String rfc = txtRFC.getText().toUpperCase();
496.         String user = txtUser.getText().toLowerCase();
497.         String pass = txtPass.getText().toLowerCase();
498.         if(check1 == true && check2 == true && check3 == true &&
    check4 == true && check5 == true){
499.             int id = (int) (Math.random()*1500+1000);

```

```
500.         DateFormat formatter = new SimpleDateFormat("yyyy-MM-
dd");
501.         String age = formatter.format(txtAge.getDate());
502.         methodsSql.Register(id, state, name, email, rfc, user,
pass, age);
503.         clear();
504.     }
505. }
506.
507. public void validateUser(){
508.     if(methodsSql.ValidateUser(txtUser.getText())){
509.         iconUserExist.setText(" » Usuario ya existente");
510.         iconUserExist.setIcon(red);
511.         check1 = false;
512.     } else {
513.         iconUserExist.setText(" » Usuario no existente");
514.         iconUserExist.setIcon(green);
515.         check1 = true;
516.     }
517.     if(txtUser.getText().length() <= 8){
518.         iconUserCharacter.setText(" » Minimo 8 caracteres");
519.         iconUserCharacter.setIcon(red);
520.         check2 = false;
521.     } else {
522.         check2 = true;
523.         iconUserCharacter.setIcon(green);
524.     }
525.     if(txtUser.getText().equals("")){
526.         iconUserExist.setIcon(red);
527.         iconUserCharacter.setIcon(red);
528.     }
529. }
530.
531. public void validateRFC(){
532.     if(methodsSql.ValidateRFC(txtRFC.getText())){
533.         iconRFCExist.setText(" » RFC ya registrado");
534.         iconRFCExist.setIcon(red);
535.         check3 = false;
536.     } else {
537.         iconRFCExist.setText(" » RFC no registrado");
538.         iconRFCExist.setIcon(green);
539.         check3 = true;
540.     }
541.     if(txtRFC.getText().length() < 12){
542.         iconRFCCCharacter.setText(" » Minimo 12 caracteres");
543.         iconRFCCCharacter.setIcon(red);
544.         check4 = false;
545.     } else {
546.         iconRFCCCharacter.setIcon(green);
547.         check4 = true;
548.     }
549.     if(txtRFC.getText().equals("")){
550.         iconRFCExist.setIcon(red);
551.         iconRFCCCharacter.setIcon(red);
552.     }
553. }
554.
```



```

555.
556.     public void validateButton(){
557.         if(txtName.getText().equals("") ||
558.            txtEmail.getText().equals("") ||
559.            txtRFC.getText().equals("") || txtUser.getText()
560.            .equals("") ||
561.            txtPass.getText().equals("") || txtAge.getDate() ==
562.            null ||
563.            check1 == false || check2 == false ||
564.            check3 == false || check4 == false || check5 == false){
565.                btnSave.setEnabled(false);
566.            } else {
567.                btnSave.setEnabled(true);
568.            }
569.        }
570.
571.     public void Mayusculas(javax.swing.JTextField jTextFieldS){
572.         String cadena = (txtRFC.getText()).toUpperCase();
573.         txtRFC.setText(cadena);
574.     }
575.
576.     public boolean isEmail(String correo) {
577.         Pattern pattern = Pattern.compile("^([0-9a-zA-Z]([_\\.w]*[0-
578.         9a-zA-Z])*@[0-9a-zA-Z]([-w]*[0-9a-zA-Z].)+([a-zA-Z]{2,9}.)+[a-zA-
579.         Z]{2,3})$");
580.         Matcher mather = pattern.matcher(correo);
581.         if (mather.find()) {
582.             return true;
583.         } else {
584.             return false;
585.         }
586.     }
587.
588.     // Variables declaration - do not modify
589.     private javax.swing.JLabel UserLabel;
590.     private javax.swing.JButton btnRemove;
591.     private javax.swing.JButton btnSave;
592.     private javax.swing.JLabel iconEmailValidate;
593.     private javax.swing.JLabel iconRFCCharacter;
594.     private javax.swing.JLabel iconRFCExist;
595.     private javax.swing.JLabel iconUserCharacter;
596.     private javax.swing.JLabel iconUserExist;
597.     private javax.swing.JLabel jLabel1;
598.     private javax.swing.JLabel jLabel2;
599.     private javax.swing.JLabel jLabel3;
600.     private javax.swing.JLabel jLabel4;
601.     private javax.swing.JLabel jLabel5;
602.     private javax.swing.JLabel jLabel6;
603.     private javax.swing.JLabel jLabel7;
604.     private javax.swing.JLabel jLabel8;
605.     private javax.swing.JPanel jPanel1;
606.     private javax.swing.JPanel jPanel2;
607.     private javax.swing.JPanel jPanel3;
608.     private javax.swing.JSeparator jSeparator2;
609.     private javax.swing.JSeparator jSeparator3;
610.     private com.toedter.calendar.JDateChooser txtAge;
611.     private javax.swing.JTextField txtEmail;

```

```

607.     private javax.swing.JTextField txtName;
608.     private javax.swing.JTextField txtPass;
609.     private javax.swing.JTextField txtRFC;
610.     private javax.swing.JComboBox<String> txtState;
611.     private javax.swing.JTextField txtUser;
612.     // End of variables declaration
613. }

```

PanelDelete

```

1.  /*
2.   * To change this license header, choose License Headers in Project
   Properties.
3.   * To change this template file, choose Tools | Templates
4.   * and open the template in the editor.
5.   */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.awt.Color;
10. import java.awt.Component;
11. import javax.swing.ImageIcon;
12. import javax.swing.JOptionPane;
13. import static view.Login.userEntity;
14.
15. /**
16.  *
17.  * @author Restf
18.  */
19. public class PanelDelete extends javax.swing.JPanel {
20.
21.     MethodsSQL methodsSql = new MethodsSQL();
22.     ImageIcon green = new
   ImageIcon(getClass().getResource("/resource/Check-True.png"));
23.     ImageIcon red = new
   ImageIcon(getClass().getResource("/resource/Check-False.png"));
24.     private boolean check1, check2;
25.
26.     public PanelDelete() {
27.         initComponents();
28.         lock();
29.     }
30.
31.
32.     @SuppressWarnings("unchecked")
33.     // <editor-fold defaultstate="collapsed" desc="Generated
   Code">
34.     private void initComponents() {
35.
36.         PanelDelete = new javax.swing.JPanel();
37.         PanelInfo = new javax.swing.JPanel();
38.         jLabel1 = new javax.swing.JLabel();
39.         txtEmail = new javax.swing.JTextField();
40.         jLabel2 = new javax.swing.JLabel();
41.         jLabel3 = new javax.swing.JLabel();

```

```

42.         txtUser = new javax.swing.JTextField();
43.         txtPass = new javax.swing.JTextField();
44.         jLabel5 = new javax.swing.JLabel();
45.         jLabel4 = new javax.swing.JLabel();
46.         jLabel6 = new javax.swing.JLabel();
47.         txtName = new javax.swing.JTextField();
48.         txtAge = new javax.swing.JTextField();
49.         txtState = new javax.swing.JTextField();
50.         txtRFC = new javax.swing.JTextField();
51.         jLabel7 = new javax.swing.JLabel();
52.         PanelButtons = new javax.swing.JPanel();
53.         btnDelete = new javax.swing.JButton();
54.         btnClear = new javax.swing.JButton();
55.         iconRFCCharacter = new javax.swing.JLabel();
56.         iconRFCEXist = new javax.swing.JLabel();
57.         jSeparator1 = new javax.swing.JSeparator();
58.         UserLabel1 = new javax.swing.JLabel();
59.         jLabel8 = new javax.swing.JLabel();
60.         jLabel9 = new javax.swing.JLabel();
61.         txtSearchUsers = new javax.swing.JTextField();
62.
63.         PanelDelete.setBackground(new java.awt.Color(128, 153,
155));
64.         PanelDelete.setPreferredSize(new java.awt.Dimension(982,
641));
65.
66.         PanelInfo.setBackground(new java.awt.Color(128, 153,
155));
67.         PanelInfo.setBorder(javax.swing.BorderFactory.createTitledBorder(nu
ll, "DATOS DE USUARIO",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
255))); // NOI18N
68.
69.         jLabel11.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
70.         jLabel11.setForeground(new java.awt.Color(255, 255, 255));
71.         jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
72.         jLabel11.setText(">> Nombre");
73.
74.         txtEmail.setEditable(false);
75.         txtEmail.setBackground(new java.awt.Color(255, 255, 255));
76.         txtEmail.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
77.         txtEmail.setForeground(new java.awt.Color(0, 0, 0));
78.         txtEmail.setPreferredSize(new java.awt.Dimension(220,
40));
79.         txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
80.             public void keyTyped(java.awt.event.KeyEvent evt) {
81.                 txtEmailKeyTyped(evt);
82.             }
83.         });
84.

```

```

85.         jLabel2.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
86.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
87.         jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
88.         jLabel2.setText(">> Corrreo");
89.
90.         jLabel3.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
91.         jLabel3.setForeground(new java.awt.Color(255, 255, 255));
92.         jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
93.         jLabel3.setText(">> Usuario");
94.         jLabel3.setToolTipText("");
95.
96.         txtUser.setEditable(false);
97.         txtUser.setBackground(new java.awt.Color(255, 255, 255));
98.         txtUser.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
99.         txtUser.setForeground(new java.awt.Color(0, 0, 0));
100.        txtUser.setPreferredSize(new java.awt.Dimension(220, 40));
101.        txtUser.addActionListener(new
java.awt.event.ActionListener() {
102.            public void actionPerformed(java.awt.event.ActionEvent
evt) {
103.                txtUserActionPerformed(evt);
104.            }
105.        });
106.        txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
107.            public void keyTyped(java.awt.event.KeyEvent evt) {
108.                txtUserKeyTyped(evt);
109.            }
110.        });
111.
112.        txtPass.setEditable(false);
113.        txtPass.setBackground(new java.awt.Color(255, 255, 255));
114.        txtPass.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
115.        txtPass.setForeground(new java.awt.Color(0, 0, 0));
116.        txtPass.setPreferredSize(new java.awt.Dimension(220, 40));
117.        txtPass.addKeyListener(new java.awt.event.KeyAdapter() {
118.            public void keyTyped(java.awt.event.KeyEvent evt) {
119.                txtPassKeyTyped(evt);
120.            }
121.        });
122.
123.        jLabel5.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
124.        jLabel5.setForeground(new java.awt.Color(255, 255, 255));
125.        jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
126.        jLabel5.setText(">> Contraseña");
127.
128.        jLabel4.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
129.        jLabel4.setForeground(new java.awt.Color(255, 255, 255));

```

```

130.        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
131.        jLabel4.setText(">> Edad");
132.
133.        jLabel6.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
134.        jLabel6.setForeground(new java.awt.Color(255, 255, 255));
135.
136.        jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
137.        jLabel6.setText(">> Estado");
138.
139.        txtName.setEditable(false);
140.        txtName.setBackground(new java.awt.Color(255, 255, 255));
141.        txtName.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
142.        txtName.setForeground(new java.awt.Color(0, 0, 0));
143.        txtName.setPreferredSize(new java.awt.Dimension(220, 40));
144.        txtName.addKeyListener(new java.awt.event.KeyAdapter() {
145.            public void keyTyped(java.awt.event.KeyEvent evt) {
146.                txtNameKeyTyped(evt);
147.            }
148.        });
149.
150.        txtAge.setEditable(false);
151.        txtAge.setBackground(new java.awt.Color(255, 255, 255));
152.        txtAge.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
153.        txtAge.setForeground(new java.awt.Color(0, 0, 0));
154.
155.        txtState.setEditable(false);
156.        txtState.setBackground(new java.awt.Color(255, 255, 255));
157.        txtState.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
158.        txtState.setForeground(new java.awt.Color(0, 0, 0));
159.
160.        txtRFC.setEditable(false);
161.        txtRFC.setBackground(new java.awt.Color(255, 255, 255));
162.        txtRFC.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
163.        txtRFC.setForeground(new java.awt.Color(0, 0, 0));
164.
165.        jLabel7.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
166.        jLabel7.setForeground(new java.awt.Color(255, 255, 255));
167.
168.        jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
169.        jLabel7.setText(">> RFC");
170.
171.        javax.swing.GroupLayout PanelInfoLayout = new
javax.swing.GroupLayout(PanelInfo);
172.        PanelInfo.setLayout(PanelInfoLayout);
173.        PanelInfoLayout.setHorizontalGroup(
174.            PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
175.                .addGroup(PanelInfoLayout.createSequentialGroup()
176.                    .addGroup(PanelInfoLayout.createParallelGroup()
177.                        .addGap(35, 35, 35)

```

```

175.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
176.        .addGroup (PanelInfoLayout.createSequentialGroup ()
177.            .addComponent (jLabel7,
178.                javax.swing.GroupLayout.PREFERRED_SIZE, 138,
179.                javax.swing.GroupLayout.PREFERRED_SIZE)
180.            .addGap (18, 18, 18)
181.            .addComponent (txtRFC))
182.        .addGroup (PanelInfoLayout.createSequentialGroup ()
183.            .addComponent (jLabel5,
184.                javax.swing.GroupLayout.PREFERRED_SIZE, 138,
185.                javax.swing.GroupLayout.PREFERRED_SIZE)
186.            .addGap (18, 18, 18)
187.            .addComponent (txtPass,
188.                javax.swing.GroupLayout.DEFAULT_SIZE,
189.                javax.swing.GroupLayout.DEFAULT_SIZE,
190.                javax.swing.GroupLayout.PREFERRED_SIZE)
191.            .addGap (0, 0, Short.MAX_VALUE))
192.        .addGroup (PanelInfoLayout.createSequentialGroup ()
193.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
194.            .addComponent (jLabel4)
195.            .addComponent (jLabel3)
196.            .addComponent (jLabel2,
197.                javax.swing.GroupLayout.PREFERRED_SIZE, 95,
198.                javax.swing.GroupLayout.PREFERRED_SIZE)
199.            .addComponent (jLabel1,
200.                javax.swing.GroupLayout.PREFERRED_SIZE, 110,
201.                javax.swing.GroupLayout.PREFERRED_SIZE)
202.            .addComponent (jLabel6,
203.                javax.swing.GroupLayout.PREFERRED_SIZE, 110,
204.                javax.swing.GroupLayout.PREFERRED_SIZE))
205.            .addGap (46, 46, 46)
206.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
207.            .addComponent (txtName,
208.                javax.swing.GroupLayout.Alignment.TRAILING,
209.                javax.swing.GroupLayout.DEFAULT_SIZE,
210.                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
211.            .addComponent (txtAge)
212.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING, false)
213.            .addComponent (txtEmail,
214.                javax.swing.GroupLayout.DEFAULT_SIZE,
215.                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
216.            .addComponent (txtUser,
217.                javax.swing.GroupLayout.PREFERRED_SIZE,
218.                javax.swing.GroupLayout.DEFAULT_SIZE,
219.                javax.swing.GroupLayout.PREFERRED_SIZE))
220.            .addComponent (txtState)))

```

```
200.                .addGap(31, 31, 31))
201.            );
202.            PanelInfoLayout.setVerticalGroup(
203.                PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
                nt.LEADING)
204.                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                PanelInfoLayout.createSequentialGroup())
205.                .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayo
                ut.Alignment.TRAILING)
206.                    .addComponent(txtRFC,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
207.                    .addComponent(jLabel7,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
208.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
                , 15, Short.MAX_VALUE)
209.                .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayo
                ut.Alignment.BASELINE)
210.                    .addComponent(txtState,
                javax.swing.GroupLayout.PREFERRED_SIZE, 40,
                javax.swing.GroupLayout.PREFERRED_SIZE)
211.                    .addComponent(jLabel6,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
212.                .addGap(17, 17, 17)
213.                .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayo
                ut.Alignment.BASELINE)
214.                    .addComponent(txtName,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
215.                    .addComponent(jLabel1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
216.                .addGap(18, 18, 18)
217.                .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayo
                ut.Alignment.BASELINE)
218.                    .addComponent(txtEmail,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
219.                    .addComponent(jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
220.                .addGap(25, 25, 25)
221.                .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayo
                ut.Alignment.BASELINE)
222.                    .addComponent(txtUser,
                javax.swing.GroupLayout.PREFERRED_SIZE,
```

```

        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
223.        .addComponent(jLabel3,
        javax.swing.GroupLayout.PREFERRED_SIZE, 33,
        javax.swing.GroupLayout.PREFERRED_SIZE))
224.        .addGap(26, 26, 26)
225.        .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
226.        .addComponent(txtPass,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
227.        .addComponent(jLabel5,
        javax.swing.GroupLayout.PREFERRED_SIZE, 41,
        javax.swing.GroupLayout.PREFERRED_SIZE))
228.        .addGap(25, 25, 25)
229.        .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
230.        .addComponent(jLabel4,
        javax.swing.GroupLayout.PREFERRED_SIZE, 33,
        javax.swing.GroupLayout.PREFERRED_SIZE)
231.        .addComponent(txtAge,
        javax.swing.GroupLayout.PREFERRED_SIZE, 41,
        javax.swing.GroupLayout.PREFERRED_SIZE))
232.        .addContainerGap())
233.    );
234.
235.    PanelButtons.setBackground(new java.awt.Color(128, 153,
155));
236.    PanelButtons.setBorder(javax.swing.BorderFactory.createTitledBorder(
null, "OPCIONES",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
255))); // NOI18N
237.
238.    btnDelete.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
239.    btnDelete.setForeground(new java.awt.Color(255, 255,
255));
240.    btnDelete.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/delete-
128px.png"))); // NOI18N
241.    btnDelete.setText("ELIMINAR USUARIO");
242.    btnDelete.setContentAreaFilled(false);
243.    btnDelete.setEnabled(false);
244.
    btnDelete.setHorizontalTextPosition(javax.swing.SwingConstants.CENT
ER);
245.    btnDelete.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM
);
246.    btnDelete.addActionListener(new
java.awt.event.ActionListener() {

```



```

247.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
248.             btnDeleteActionPerformed(evt);
249.         }
250.     });
251.
252.     btnClear.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
253.     btnClear.setForeground(new java.awt.Color(255, 255, 255));
254.     btnClear.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/remove-
    128px.png"))); // NOI18N
255.     btnClear.setText("LIMPIAR DATOS");
256.     btnClear.setContentAreaFilled(false);
257.     btnClear.setHorizontalTextPosition(javax.swing.SwingConstants.CENTE
    R);
258.     btnClear.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM)
    ;
259.     btnClear.addActionListener(new
    java.awt.event.ActionListener() {
260.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
261.             btnClearActionPerformed(evt);
262.         }
263.     });
264.
265.     iconRFCCharacter.setFont(new java.awt.Font("Dialog", 1,
    18)); // NOI18N
266.     iconRFCCharacter.setForeground(new java.awt.Color(255,
    255, 255));
267.     iconRFCCharacter.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
    False.png"))); // NOI18N
268.     iconRFCCharacter.setText(" » Minimo 12 caracteres");
269.     iconRFCCharacter.setFocusable(false);
270.
271.     iconRFCExist.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
272.     iconRFCExist.setForeground(new java.awt.Color(255, 255,
    255));
273.     iconRFCExist.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
    False.png"))); // NOI18N
274.     iconRFCExist.setText(" » RFC no encontrado");
275.     iconRFCExist.setFocusable(false);
276.
277.     UserLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
278.     UserLabel1.setForeground(new java.awt.Color(255, 255,
    255));
279.     UserLabel1.setText("RFC");
280.     UserLabel1.setFocusable(false);
281.
282.     javax.swing.GroupLayout PanelButtonsLayout = new
    javax.swing.GroupLayout(PanelButtons);

```

```

283.         PanelButtons.setLayout (PanelButtonsLayout);
284.         PanelButtonsLayout.setHorizontalGroup (
285.             PanelButtonsLayout.createParallelGroup (javax.swing.GroupLayout.Align
                ment.LEADING)
286.                 .addGroup (PanelButtonsLayout.createSequentialGroup ())
287.                 .addGroup (PanelButtonsLayout.createParallelGroup (javax.swing.GroupL
                    ayout.Alignment.LEADING)
288.                     .addGroup (PanelButtonsLayout.createSequentialGroup ())
289.                         .addGap (12, 12, 12)
290.                         .addComponent (btnDelete)
291.                     .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
                        )
292.                     .addComponent (btnClear,
                javax.swing.GroupLayout.PREFERRED_SIZE, 205,
                javax.swing.GroupLayout.PREFERRED_SIZE))
293.                 .addGroup (PanelButtonsLayout.createSequentialGroup ())
294.                     .addGap (71, 71, 71)
295.                 .addGroup (PanelButtonsLayout.createParallelGroup (javax.swing.GroupL
                    ayout.Alignment.LEADING)
296.                     .addComponent (UserLabel1)
297.                     .addComponent (jSeparator1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 272,
                javax.swing.GroupLayout.PREFERRED_SIZE)
298.                     .addComponent (iconRFCEXist)
299.                     .addComponent (iconRFCCharacter)))
300.                 .addContainerGap (12, Short.MAX_VALUE))
301.         );
302.         PanelButtonsLayout.setVerticalGroup (
303.             PanelButtonsLayout.createParallelGroup (javax.swing.GroupLayout.Align
                ment.LEADING)
304.                 .addGroup (javax.swing.GroupLayout.Alignment.TRAILING,
                PanelButtonsLayout.createSequentialGroup ())
305.                 .addGap (23, 23, 23)
306.                 .addComponent (UserLabel1)
307.                 .addGap (6, 6, 6)
308.                 .addGroup (PanelButtonsLayout.createParallelGroup (javax.swing.GroupL
                    ayout.Alignment.LEADING)
309.                     .addComponent (jSeparator1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 20,
                javax.swing.GroupLayout.PREFERRED_SIZE)
310.                     .addGroup (PanelButtonsLayout.createSequentialGroup ())
311.                         .addGap (10, 10, 10)
312.                         .addComponent (iconRFCEXist)))
313.                 .addGap (8, 8, 8)
314.                 .addComponent (iconRFCCharacter)
315.                 .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
                    , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

316.        .addGroup (PanelButtonsLayout.createParallelGroup (javax.swing.GroupL
          ayout.Alignment.TRAILING)
317.                .addComponent (btnDelete)
318.                .addComponent (btnClear))
319.                .addGap (15, 15, 15))
320.        );
321.
322.        jLabel8.setFont (new java.awt.Font ("Dialog", 1, 36)); //
      NOI18N
323.        jLabel8.setForeground (new java.awt.Color (255, 255, 255));
324.        jLabel8.setText ("ELIMINACION DE USUARIOS");
325.
326.        jLabel9.setFont (new java.awt.Font ("Times New Roman", 0,
          24)); // NOI18N
327.        jLabel9.setForeground (new java.awt.Color (255, 255, 255));
328.
329.        jLabel9.setHorizontalAlignment (javax.swing.SwingConstants.LEFT);
330.        jLabel9.setText ("» Ingrese RFC de usuario para eliminar
          datos:");
331.
332.        txtSearchUsers.setBackground (new java.awt.Color (255, 255,
          255));
333.        txtSearchUsers.setFont (new java.awt.Font ("Dialog", 0,
          16)); // NOI18N
334.        txtSearchUsers.setForeground (new java.awt.Color (0, 0, 0));
335.        txtSearchUsers.addKeyListener (new
          java.awt.event.KeyAdapter () {
336.            public void keyReleased (java.awt.event.KeyEvent evt) {
337.                txtSearchUsersKeyReleased (evt);
338.            }
339.            public void keyTyped (java.awt.event.KeyEvent evt) {
340.                txtSearchUsersKeyTyped (evt);
341.            }
342.        });
343.
344.        javax.swing.GroupLayout PanelDeleteLayout = new
          javax.swing.GroupLayout (PanelDelete);
345.        PanelDelete.setLayout (PanelDeleteLayout);
346.        PanelDeleteLayout.setHorizontalGroup (
          PanelDeleteLayout.createParallelGroup (javax.swing.GroupLayout.Align
            ment.LEADING)
347.            .addGroup (javax.swing.GroupLayout.Alignment.TRAILING,
          PanelDeleteLayout.createSequentialGroup ()
348.                .addContainerGap (31, Short.MAX_VALUE)
349.
350.                .addGroup (PanelDeleteLayout.createParallelGroup (javax.swing.GroupLa
          yout.Alignment.LEADING)
351.                    .addGroup (javax.swing.GroupLayout.Alignment.TRAILING,
          PanelDeleteLayout.createSequentialGroup ()
352.                        .addGap (0, 0, Short.MAX_VALUE)
353.                        .addComponent (jLabel8)
354.                        .addGap (240, 240, 240))

```

```

354.        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        PanelDeleteLayout.createSequentialGroup())
355.            .addComponent(PanelInfo,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
356.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
        ED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
357.            .addComponent(PanelButtons,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
358.            .addGap(51, 51, 51))
359.        .addGroup(PanelDeleteLayout.createSequentialGroup())
360.            .addGap(71, 71, 71)
361.            .addComponent(jLabel9)
362.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
        ED)
363.            .addComponent(txtSearchUsers,
        javax.swing.GroupLayout.PREFERRED_SIZE, 291,
        javax.swing.GroupLayout.PREFERRED_SIZE)
364.        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE)))
365.    );
366.    PanelDeleteLayout.setVerticalGroup(
367.        PanelDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Align
        ment.LEADING)
368.            .addGroup(PanelDeleteLayout.createSequentialGroup())
369.                .addComponent(jLabel8,
        javax.swing.GroupLayout.PREFERRED_SIZE, 59,
        javax.swing.GroupLayout.PREFERRED_SIZE)
370.                .addGap(19, 19, 19)
371.            .addGroup(PanelDeleteLayout.createParallelGroup(javax.swing.GroupLa
        yout.Alignment.BASELINE)
372.                .addComponent(jLabel9,
        javax.swing.GroupLayout.PREFERRED_SIZE, 33,
        javax.swing.GroupLayout.PREFERRED_SIZE)
373.                .addComponent(txtSearchUsers,
        javax.swing.GroupLayout.PREFERRED_SIZE, 40,
        javax.swing.GroupLayout.PREFERRED_SIZE))
374.                .addGap(33, 33, 33)
375.            .addGroup(PanelDeleteLayout.createParallelGroup(javax.swing.GroupLa
        yout.Alignment.LEADING, false)
376.                .addComponent(PanelInfo,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
377.                .addComponent(PanelButtons,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

```

```

378.         .addContainerGap(44, Short.MAX_VALUE))
379.     );
380.
381.     javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(this);
382.     this.setLayout(layout);
383.     layout.setHorizontalGroup(
384.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
385.             .addComponent(PanelDelete,
    javax.swing.GroupLayout.DEFAULT_SIZE, 996, Short.MAX_VALUE)
386.             );
387.     layout.setVerticalGroup(
388.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
389.             .addComponent(PanelDelete,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
390.             );
391.     } // </editor-fold>
392.
393.     private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
394.
395.     }
396.
397.     private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
398.
399.     }
400.
401.     private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
402.
403.     }
404.
405.     private void txtUserActionPerformed(java.awt.event.ActionEvent
    evt) {
406.         // TODO add your handling code here:
407.     }
408.
409.     private void txtNameKeyTyped(java.awt.event.KeyEvent evt) {
410.         // TODO add your handling code here:
411.     }
412.
413.     private void txtSearchUsersKeyReleased(java.awt.event.KeyEvent
    evt) {
414.         validateRFC();
415.         iconRFCExist.setForeground(Color.WHITE);
416.         iconRFCCharacter.setForeground(Color.WHITE);
417.         Mayusculas(txtSearchUsers);
418.         if(check1 == true && check2 == true){
419.             userEntity = methodsSql.RFC(txtSearchUsers.getText());
420.             unlock();
421.             showData();
422.         }
423.     }
424.

```

```
425.     private void txtSearchUsersKeyTyped(java.awt.event.KeyEvent
    evt) {
426.         int characterLimit=13;
427.         if(txtSearchUsers.getText().length()>=characterLimit){
428.             evt.consume();
429.         }
430.     }
431.
432.     private void
    btnClearActionPerformed(java.awt.event.ActionEvent evt) {
433.         clear();
434.     }
435.
436.     private void
    btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
437.         DeleteUserValidate();
438.     }
439.
440.
441.     public void lock(){
442.         for(Component a: PanelInfo.getComponents()){
443.             a.setEnabled(false);
444.         }
445.
446.         btnDelete.setEnabled(false);
447.     }
448.
449.
450.     public void unlock(){
451.         for(Component a: PanelInfo.getComponents()){
452.             a.setEnabled(true);
453.         }
454.
455.         for(Component b: PanelButtons.getComponents()){
456.             b.setEnabled(true);
457.         }
458.     }
459.
460.     public void clear(){
461.         txtSearchUsers.setText(null);
462.         txtRFC.setText(null);
463.         txtState.setText(null);
464.         txtName.setText(null);
465.         txtEmail.setText(null);
466.         txtUser.setText(null);
467.         txtPass.setText(null);
468.         txtAge.setText(null);
469.         iconRFCExist.setIcon(red);
470.         iconRFCCharacter.setIcon(red);
471.         iconRFCExist.setText(" » RFC no encontrado");
472.         lock();
473.     }
474.
475.     public void Mayusculas(javax.swing.JTextField jTextFieldS){
476.         String cadena = (txtSearchUsers.getText()).toUpperCase();
477.         txtSearchUsers.setText(cadena);
478.     }
```

```

479.
480.     public void showData(){
481.         txtRFC.setText(userEntity.getRfc());
482.         String state= "";
483.         if (userEntity.getState() == 1) {
484.             state = "Administrador";
485.         } else {
486.             state = "Usuario";
487.         }
488.         txtState.setText(state);
489.         txtName.setText(userEntity.getName());
490.         txtEmail.setText(userEntity.getEmail());
491.         txtUser.setText(userEntity.getUser());
492.         txtPass.setText(userEntity.getPass());
493.         txtAge.setText(userEntity.getAge());
494.     }
495.
496.     public void validateRFC(){
497.         if(methodsSql.ValidateRFC(txtSearchUsers.getText())){
498.             iconRFCExist.setText(" » RFC encontrado");
499.             iconRFCExist.setIcon(green);
500.             check1 = true;
501.         } else {
502.             iconRFCExist.setText(" » RFC no encontrado");
503.             iconRFCExist.setIcon(red);
504.             check1 = false;
505.             lock();
506.         }
507.
508.         if(txtSearchUsers.getText().length() < 12){
509.             iconRFCCharacter.setText(" » Minimo 12 caracteres");
510.             iconRFCCharacter.setIcon(red);
511.             check2 = false;
512.             lock();
513.         } else {
514.             iconRFCCharacter.setIcon(green);
515.             check2 = true;
516.         }
517.     }
518.
519.     public void DeleteUserValidate(){
520.         int i = JOptionPane.showConfirmDialog(null, "¿REALMENTE
DESEAS ELIMINAR EL USUARIO?", "¡ALERTA!",
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
521.         if(i == 0){
522.             methodsSql.DeleteUserAdmin(txtSearchUsers.getText());
523.             clear();
524.         }
525.     }
526.
527.
528.
529.     // Variables declaration - do not modify
530.     private javax.swing.JPanel PanelButtons;
531.     private javax.swing.JPanel PanelDelete;
532.     private javax.swing.JPanel PanelInfo;
533.     private javax.swing.JLabel UserLabel1;

```

```

534.     private javax.swing.JButton btnClear;
535.     private javax.swing.JButton btnDelete;
536.     private javax.swing.JLabel iconRFCCharacter;
537.     private javax.swing.JLabel iconRFCExist;
538.     private javax.swing.JLabel jLabel1;
539.     private javax.swing.JLabel jLabel2;
540.     private javax.swing.JLabel jLabel3;
541.     private javax.swing.JLabel jLabel4;
542.     private javax.swing.JLabel jLabel5;
543.     private javax.swing.JLabel jLabel6;
544.     private javax.swing.JLabel jLabel7;
545.     private javax.swing.JLabel jLabel8;
546.     private javax.swing.JLabel jLabel9;
547.     private javax.swing.JSeparator jSeparator1;
548.     private javax.swing.JTextField txtAge;
549.     private javax.swing.JTextField txtEmail;
550.     private javax.swing.JTextField txtName;
551.     private javax.swing.JTextField txtPass;
552.     private javax.swing.JTextField txtRFC;
553.     private javax.swing.JTextField txtSearchUsers;
554.     private javax.swing.JTextField txtState;
555.     private javax.swing.JTextField txtUser;
556.     // End of variables declaration
557. }

```

PanelEdit

```

1.  /*
2.   * To change this license header, choose License Headers in Project
    Properties.
3.   * To change this template file, choose Tools | Templates
4.   * and open the template in the editor.
5.   */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.awt.Color;
10. import java.awt.Component;
11. import javax.swing.ImageIcon;
12. import javax.swing.JOptionPane;
13. import static view.Login.userEntity;
14.
15. /**
16.  *
17.  * @author Restf
18.  */
19. public class PanelDelete extends javax.swing.JPanel {
20.
21.     MethodsSQL methodsSql = new MethodsSQL();
22.     ImageIcon green = new
    ImageIcon(getClass().getResource("/resource/Check-True.png"));
23.     ImageIcon red = new
    ImageIcon(getClass().getResource("/resource/Check-False.png"));
24.     private boolean check1, check2;
25.

```



```

26.     public PanelDelete() {
27.         initComponents();
28.         lock();
29.     }
30.
31.
32.     @SuppressWarnings("unchecked")
33.     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
34.     private void initComponents() {
35.
36.         PanelDelete = new javax.swing.JPanel();
37.         PanelInfo = new javax.swing.JPanel();
38.         jLabel1 = new javax.swing.JLabel();
39.         txtEmail = new javax.swing.JTextField();
40.         jLabel2 = new javax.swing.JLabel();
41.         jLabel3 = new javax.swing.JLabel();
42.         txtUser = new javax.swing.JTextField();
43.         txtPass = new javax.swing.JTextField();
44.         jLabel5 = new javax.swing.JLabel();
45.         jLabel4 = new javax.swing.JLabel();
46.         jLabel6 = new javax.swing.JLabel();
47.         txtName = new javax.swing.JTextField();
48.         txtAge = new javax.swing.JTextField();
49.         txtState = new javax.swing.JTextField();
50.         txtRFC = new javax.swing.JTextField();
51.         jLabel7 = new javax.swing.JLabel();
52.         PanelButtons = new javax.swing.JPanel();
53.         btnDelete = new javax.swing.JButton();
54.         btnClear = new javax.swing.JButton();
55.         iconRFCCharacter = new javax.swing.JLabel();
56.         iconRFCExist = new javax.swing.JLabel();
57.         jSeparator1 = new javax.swing.JSeparator();
58.         UserLabel1 = new javax.swing.JLabel();
59.         jLabel8 = new javax.swing.JLabel();
60.         jLabel9 = new javax.swing.JLabel();
61.         txtSearchUsers = new javax.swing.JTextField();
62.
63.         PanelDelete.setBackground(new java.awt.Color(128, 153,
155));
64.         PanelDelete.setPreferredSize(new java.awt.Dimension(982,
641));
65.
66.         PanelInfo.setBackground(new java.awt.Color(128, 153,
155));
67.
68.         PanelInfo.setBorder(javax.swing.BorderFactory.createTitledBorder(nu
ll, "DATOS DE USUARIO",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
255))); // NOI18N
69.
69.         jLabel1.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
70.         jLabel1.setForeground(new java.awt.Color(255, 255, 255));

```

```

71.     jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
72.         jLabel1.setText("> Nombre");
73.
74.         txtEmail.setEditable(false);
75.         txtEmail.setBackground(new java.awt.Color(255, 255, 255));
76.         txtEmail.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
77.         txtEmail.setForeground(new java.awt.Color(0, 0, 0));
78.         txtEmail.setPreferredSize(new java.awt.Dimension(220,
40));
79.         txtEmail.addKeyListener(new java.awt.event.KeyAdapter() {
80.             public void keyTyped(java.awt.event.KeyEvent evt) {
81.                 txtEmailKeyTyped(evt);
82.             }
83.         });
84.
85.         jLabel2.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
86.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
87.
88.         jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
89.         jLabel2.setText("> Correo");
90.
91.         jLabel3.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
92.         jLabel3.setForeground(new java.awt.Color(255, 255, 255));
93.         jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
94.         jLabel3.setText("> Usuario");
95.         jLabel3.setToolTipText("");
96.
97.         txtUser.setEditable(false);
98.         txtUser.setBackground(new java.awt.Color(255, 255, 255));
99.         txtUser.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
100.         txtUser.setForeground(new java.awt.Color(0, 0, 0));
101.         txtUser.setPreferredSize(new java.awt.Dimension(220, 40));
102.         txtUser.addActionListener(new
java.awt.event.ActionListener() {
103.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
104.                 txtUserActionPerformed(evt);
105.             }
106.         });
107.         txtUser.addKeyListener(new java.awt.event.KeyAdapter() {
108.             public void keyTyped(java.awt.event.KeyEvent evt) {
109.                 txtUserKeyTyped(evt);
110.             }
111.         });
112.
113.         txtPass.setEditable(false);
114.         txtPass.setBackground(new java.awt.Color(255, 255, 255));
115.         txtPass.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
116.         txtPass.setForeground(new java.awt.Color(0, 0, 0));
117.         txtPass.setPreferredSize(new java.awt.Dimension(220, 40));

```

```

117.         txtPass.addKeyListener(new java.awt.event.KeyAdapter() {
118.             public void keyTyped(java.awt.event.KeyEvent evt) {
119.                 txtPassKeyTyped(evt);
120.             }
121.         });
122.
123.         jLabel5.setFont(new java.awt.Font("Times New Roman", 0,
124.             24)); // NOI18N
125.         jLabel5.setForeground(new java.awt.Color(255, 255, 255));
126.         jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
127.         jLabel5.setText(">> Contraseña");
128.         jLabel4.setFont(new java.awt.Font("Times New Roman", 0,
129.             24)); // NOI18N
130.         jLabel4.setForeground(new java.awt.Color(255, 255, 255));
131.         jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
132.         jLabel4.setText(">> Edad");
133.         jLabel6.setFont(new java.awt.Font("Times New Roman", 0,
134.             24)); // NOI18N
135.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
136.         jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
137.         jLabel6.setText(">> Estado");
138.         txtName.setEditable(false);
139.         txtName.setBackground(new java.awt.Color(255, 255, 255));
140.         txtName.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
141.         txtName.setForeground(new java.awt.Color(0, 0, 0));
142.         txtName.setPreferredSize(new java.awt.Dimension(220, 40));
143.         txtName.addKeyListener(new java.awt.event.KeyAdapter() {
144.             public void keyTyped(java.awt.event.KeyEvent evt) {
145.                 txtNameKeyTyped(evt);
146.             }
147.         });
148.
149.         txtAge.setEditable(false);
150.         txtAge.setBackground(new java.awt.Color(255, 255, 255));
151.         txtAge.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
152.         txtAge.setForeground(new java.awt.Color(0, 0, 0));
153.
154.         txtState.setEditable(false);
155.         txtState.setBackground(new java.awt.Color(255, 255, 255));
156.         txtState.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
157.         txtState.setForeground(new java.awt.Color(0, 0, 0));
158.
159.         txtRFC.setEditable(false);
160.         txtRFC.setBackground(new java.awt.Color(255, 255, 255));
161.         txtRFC.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
162.         txtRFC.setForeground(new java.awt.Color(0, 0, 0));
163.

```

```

164.         jLabel7.setFont(new java.awt.Font("Times New Roman", 0,
24)); // NOI18N
165.         jLabel7.setForeground(new java.awt.Color(255, 255, 255));
166.         jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
167.         jLabel7.setText("» RFC");
168.
169.         javax.swing.GroupLayout PanelInfoLayout = new
javax.swing.GroupLayout(PanelInfo);
170.         PanelInfo.setLayout(PanelInfoLayout);
171.         PanelInfoLayout.setHorizontalGroup(
172.
PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
173.             .addGroup(PanelInfoLayout.createSequentialGroup()
174.                 .addGap(35, 35, 35)
175.                 .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
176.                     .addGroup(PanelInfoLayout.createSequentialGroup()
177.                         .addComponent(jLabel7,
javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)
178.                         .addGap(18, 18, 18)
179.                         .addComponent(txtRFC))
180.                     .addGroup(PanelInfoLayout.createSequentialGroup()
181.                         .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE)
182.                         .addGap(18, 18, 18)
183.                         .addComponent(txtPass,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
184.                         .addGap(0, 0, Short.MAX_VALUE))
185.                     .addGroup(PanelInfoLayout.createSequentialGroup()
186.                         .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
187.                             .addComponent(jLabel4)
188.                             .addComponent(jLabel3)
189.                             .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 95,
javax.swing.GroupLayout.PREFERRED_SIZE)
190.                             .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE)
191.                             .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE, 110,
javax.swing.GroupLayout.PREFERRED_SIZE))
192.                         .addGap(46, 46, 46)
193.                         .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

```

194.                .addComponent(txtName,
    javax.swing.GroupLayout.Alignment.TRAILING,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
195.                .addComponent(txtAge)
196.
    .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.
    Alignment.LEADING, false)
197.                .addComponent(txtEmail,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
198.                .addComponent(txtUser,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
199.                .addComponent(txtState)))
200.            .addGap(31, 31, 31))
201.        );
202.        PanelInfoLayout.setVerticalGroup(
203.
    PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout.Alignme
    nt.LEADING)
204.            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    PanelInfoLayout.createSequentialGroup())
205.
    .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout
    Alignment.TRAILING)
206.            .addComponent(txtRFC,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE)
207.            .addComponent(jLabel7,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
208.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    , 15, Short.MAX_VALUE)
209.
    .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout
    Alignment.BASELINE)
210.            .addComponent(txtState,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE)
211.            .addComponent(jLabel6,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
212.            .addGap(17, 17, 17)
213.
    .addGroup(PanelInfoLayout.createParallelGroup(javax.swing.GroupLayout
    Alignment.BASELINE)
214.            .addComponent(txtName,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
215.            .addComponent(jLabel11,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE))
216.            .addGap(18, 18, 18)

```

```

217.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
218.                .addComponent (txtEmail,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
219.                .addComponent (jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
220.        .addGap (25, 25, 25)
221.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
222.                .addComponent (txtUser,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
223.                .addComponent (jLabel3,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE))
224.        .addGap (26, 26, 26)
225.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
226.                .addComponent (txtPass,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
227.                .addComponent (jLabel5,
                javax.swing.GroupLayout.PREFERRED_SIZE, 41,
                javax.swing.GroupLayout.PREFERRED_SIZE))
228.        .addGap (25, 25, 25)
229.        .addGroup (PanelInfoLayout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
230.                .addComponent (jLabel4,
                javax.swing.GroupLayout.PREFERRED_SIZE, 33,
                javax.swing.GroupLayout.PREFERRED_SIZE)
231.                .addComponent (txtAge,
                javax.swing.GroupLayout.PREFERRED_SIZE, 41,
                javax.swing.GroupLayout.PREFERRED_SIZE))
232.        .addContainerGap ()
233.    );
234.
235.    PanelButtons.setBackground (new java.awt.Color (128, 153,
236.        155));
237.
238.    PanelButtons.setBorder (javax.swing.BorderFactory.createTitledBorder
239.        (null, "OPCIONES",
240.        javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
241.        javax.swing.border.TitledBorder.DEFAULT_POSITION, new
242.        java.awt.Font ("Dialog", 1, 18), new java.awt.Color (255, 255,
243.        255))); // NOI18N
244.
245.    btnDelete.setFont (new java.awt.Font ("Dialog", 1, 18)); //
246.    NOI18N

```

```

239.         btnDelete.setForeground(new java.awt.Color(255, 255,
255));
240.         btnDelete.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/delete-
128px.png"))); // NOI18N
241.         btnDelete.setText("ELIMINAR USUARIO");
242.         btnDelete.setContentAreaFilled(false);
243.         btnDelete.setEnabled(false);
244.
    btnDelete.setHorizontalTextPosition(javax.swing.SwingConstants.CENT
ER);
245.
    btnDelete.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM
);
246.         btnDelete.addActionListener(new
java.awt.event.ActionListener() {
247.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
248.                 btnDeleteActionPerformed(evt);
249.             }
250.         });
251.
252.         btnClear.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
253.         btnClear.setForeground(new java.awt.Color(255, 255, 255));
254.         btnClear.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/remove-
128px.png"))); // NOI18N
255.         btnClear.setText("LIMPIAR DATOS");
256.         btnClear.setContentAreaFilled(false);
257.
    btnClear.setHorizontalTextPosition(javax.swing.SwingConstants.CENTE
R);
258.
    btnClear.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM)
;
259.         btnClear.addActionListener(new
java.awt.event.ActionListener() {
260.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
261.                 btnClearActionPerformed(evt);
262.             }
263.         });
264.
265.         iconRFCCharacter.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
266.         iconRFCCharacter.setForeground(new java.awt.Color(255,
255, 255));
267.         iconRFCCharacter.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/Check-
False.png"))); // NOI18N
268.         iconRFCCharacter.setText(" » Minimo 12 caracteres");
269.         iconRFCCharacter.setFocusable(false);
270.
271.         iconRFCExist.setFont(new java.awt.Font("Dialog", 1, 18));
// NOI18N

```

```

272.         iconRFCExist.setForeground(new java.awt.Color(255, 255,
255));
273.         iconRFCExist.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/Check-
False.png"))); // NOI18N
274.         iconRFCExist.setText(" » RFC no encontrado");
275.         iconRFCExist.setFocusable(false);
276.
277.         UserLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
278.         UserLabel1.setForeground(new java.awt.Color(255, 255,
255));
279.         UserLabel1.setText("RFC");
280.         UserLabel1.setFocusable(false);
281.
282.         javax.swing.GroupLayout PanelButtonsLayout = new
    javax.swing.GroupLayout(PanelButtons);
283.         PanelButtons.setLayout(PanelButtonsLayout);
284.         PanelButtonsLayout.setHorizontalGroup(
285.             PanelButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
286.                 .addGroup(PanelButtonsLayout.createSequentialGroup()
287.                     .addGroup(PanelButtonsLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
288.                         .addGroup(PanelButtonsLayout.createSequentialGroup()
289.                             .addGap(12, 12, 12)
290.                             .addComponent(btnDelete)
291.                             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacemen
t.RELATED)
292.                             .addComponent(btnClear,
    javax.swing.GroupLayout.PREFERRED_SIZE, 205,
    javax.swing.GroupLayout.PREFERRED_SIZE))
293.                         .addGroup(PanelButtonsLayout.createSequentialGroup()
294.                             .addGap(71, 71, 71)
295.                             .addGroup(PanelButtonsLayout.createParallelGroup(javax.swing.GroupL
ayout.Alignment.LEADING)
296.                                 .addComponent(UserLabel1)
297.                                 .addComponent(jSeparator1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 272,
    javax.swing.GroupLayout.PREFERRED_SIZE)
298.                                 .addComponent(iconRFCExist)
299.                                 .addComponent(iconRFCCharacter)))
300.                             .addContainerGap(12, Short.MAX_VALUE))
301.                     );
302.         PanelButtonsLayout.setVerticalGroup(
303.             PanelButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
304.                 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    PanelButtonsLayout.createSequentialGroup()
305.                     .addGap(23, 23, 23)

```



```

306.                .addComponent(UserLabel1)
307.                .addGap(6, 6, 6)
308.
309.                .addGroup(PanelButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
310.                .addComponent(jSeparator1,
311.                javax.swing.GroupLayout.PREFERRED_SIZE, 20,
312.                javax.swing.GroupLayout.PREFERRED_SIZE)
313.                .addGroup(PanelButtonsLayout.createSequentialGroup()
314.                .addGap(10, 10, 10)
315.                .addComponent(iconRFCEXist)))
316.                .addGap(8, 8, 8)
317.                .addComponent(iconRFCCharacter)
318.
319.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
320.                , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
321.                .addGroup(PanelButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
322.                .addComponent(btnDelete)
323.                .addComponent(btnClear))
324.                .addGap(15, 15, 15))
325.        );
326.
327.        jLabel8.setFont(new java.awt.Font("Dialog", 1, 36)); //
NOI18N
328.        jLabel8.setForeground(new java.awt.Color(255, 255, 255));
329.        jLabel8.setText("ELIMINACION DE USUARIOS");
330.
331.        jLabel9.setFont(new java.awt.Font("Times New Roman", 0,
332.        24)); // NOI18N
333.        jLabel9.setForeground(new java.awt.Color(255, 255, 255));
334.
335.        jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
336.        jLabel9.setText(">> Ingrese RFC de usuario para eliminar
337.        datos:");
338.
339.        txtSearchUsers.setBackground(new java.awt.Color(255, 255,
340.        255));
341.        txtSearchUsers.setFont(new java.awt.Font("Dialog", 0,
342.        16)); // NOI18N
343.        txtSearchUsers.setForeground(new java.awt.Color(0, 0, 0));
344.        txtSearchUsers.addKeyListener(new
345.        java.awt.event.KeyAdapter() {
346.            public void keyReleased(java.awt.event.KeyEvent evt) {
347.                txtSearchUsersKeyReleased(evt);
348.            }
349.            public void keyTyped(java.awt.event.KeyEvent evt) {
350.                txtSearchUsersKeyTyped(evt);
351.            }
352.        });
353.
354.        javax.swing.GroupLayout PanelDeleteLayout = new
355.        javax.swing.GroupLayout(PanelDelete);
356.        PanelDelete.setLayout(PanelDeleteLayout);
357.        PanelDeleteLayout.setHorizontalGroup(

```

```

346.     PanelDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
347.         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
PanelDeleteLayout.createSequentialGroup())
348.         .addContainerGap(31, Short.MAX_VALUE)
349.     .addGroup(PanelDeleteLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.LEADING)
350.     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
PanelDeleteLayout.createSequentialGroup())
351.         .addGap(0, 0, Short.MAX_VALUE)
352.         .addComponent(jLabel8)
353.         .addGap(240, 240, 240))
354.     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
PanelDeleteLayout.createSequentialGroup())
355.         .addComponent(PanelInfo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
356.     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
357.         .addComponent(PanelButtons,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
358.         .addGap(51, 51, 51))
359.     .addGroup(PanelDeleteLayout.createSequentialGroup())
360.         .addGap(71, 71, 71)
361.         .addComponent(jLabel9)
362.     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
363.         .addComponent(txtSearchUsers,
javax.swing.GroupLayout.PREFERRED_SIZE, 291,
javax.swing.GroupLayout.PREFERRED_SIZE)
364.     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
365.     );
366.     PanelDeleteLayout.setVerticalGroup(
367.     PanelDeleteLayout.createParallelGroup(javax.swing.GroupLayout.Align
ment.LEADING)
368.     .addGroup(PanelDeleteLayout.createSequentialGroup())
369.         .addComponent(jLabel8,
javax.swing.GroupLayout.PREFERRED_SIZE, 59,
javax.swing.GroupLayout.PREFERRED_SIZE)
370.         .addGap(19, 19, 19)
371.     .addGroup(PanelDeleteLayout.createParallelGroup(javax.swing.GroupLa
yout.Alignment.BASELINE)

```

```

372.                .addComponent(jLabel9,
    javax.swing.GroupLayout.PREFERRED_SIZE, 33,
    javax.swing.GroupLayout.PREFERRED_SIZE)
373.                .addComponent(txtSearchUsers,
    javax.swing.GroupLayout.PREFERRED_SIZE, 40,
    javax.swing.GroupLayout.PREFERRED_SIZE))
374.                .addGap(33, 33, 33)
375.
    .addGroup(PanelDeleteLayout.createParallelGroup(javax.swing.GroupLa
    yout.Alignment.LEADING, false)
376.                .addComponent(PanelInfo,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
377.                .addComponent(PanelButtons,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
378.                .addContainerGap(44, Short.MAX_VALUE))
379.            );
380.
381.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(this);
382.            this.setLayout(layout);
383.            layout.setHorizontalGroup(
384.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
385.                .addComponent(PanelDelete,
    javax.swing.GroupLayout.DEFAULT_SIZE, 996, Short.MAX_VALUE)
386.            );
387.            layout.setVerticalGroup(
388.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
389.                .addComponent(PanelDelete,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
390.            );
391.        } // </editor-fold>
392.
393.        private void txtEmailKeyTyped(java.awt.event.KeyEvent evt) {
394.
395.        }
396.
397.        private void txtUserKeyTyped(java.awt.event.KeyEvent evt) {
398.
399.        }
400.
401.        private void txtPassKeyTyped(java.awt.event.KeyEvent evt) {
402.
403.        }
404.
405.        private void txtUserActionPerformed(java.awt.event.ActionEvent
    evt) {
406.            // TODO add your handling code here:
407.        }
408.
409.        private void txtNameKeyTyped(java.awt.event.KeyEvent evt) {

```

```
410.         // TODO add your handling code here:
411.     }
412.
413.     private void txtSearchUsersKeyReleased(java.awt.event.KeyEvent
    evt) {
414.         validateRFC();
415.         iconRFCExist.setForeground(Color.WHITE);
416.         iconRFCCharacter.setForeground(Color.WHITE);
417.         Mayusculas(txtSearchUsers);
418.         if(check1 == true && check2 == true){
419.             userEntity = methodsSql.RFC(txtSearchUsers.getText());
420.             unlock();
421.             showData();
422.         }
423.     }
424.
425.     private void txtSearchUsersKeyTyped(java.awt.event.KeyEvent
    evt) {
426.         int characterLimit=13;
427.         if(txtSearchUsers.getText().length()>=characterLimit){
428.             evt.consume();
429.         }
430.     }
431.
432.     private void
    btnClearActionPerformed(java.awt.event.ActionEvent evt) {
433.         clear();
434.     }
435.
436.     private void
    btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
437.         DeleteUserValidate();
438.     }
439.
440.
441.     public void lock(){
442.         for(Component a: PanelInfo.getComponents()){
443.             a.setEnabled(false);
444.         }
445.
446.         btnDelete.setEnabled(false);
447.     }
448.
449.
450.     public void unlock(){
451.         for(Component a: PanelInfo.getComponents()){
452.             a.setEnabled(true);
453.         }
454.
455.         for(Component b: PanelButtons.getComponents()){
456.             b.setEnabled(true);
457.         }
458.     }
459.
460.     public void clear(){
461.         txtSearchUsers.setText(null);
462.         txtRFC.setText(null);
```

```
463.         txtState.setText(null);
464.         txtName.setText(null);
465.         txtEmail.setText(null);
466.         txtUser.setText(null);
467.         txtPass.setText(null);
468.         txtAge.setText(null);
469.         iconRFCExist.setIcon(red);
470.         iconRFCCharacter.setIcon(red);
471.         iconRFCExist.setText(" » RFC no encontrado");
472.         lock();
473.     }
474.
475.     public void Mayusculas(javax.swing.JTextField jTextFieldS){
476.         String cadena = (txtSearchUsers.getText()).toUpperCase();
477.         txtSearchUsers.setText(cadena);
478.     }
479.
480.     public void showData(){
481.         txtRFC.setText(userEntity.getRfc());
482.         String state= "";
483.         if (userEntity.getState() == 1) {
484.             state = "Administrador";
485.         } else {
486.             state = "Usuario";
487.         }
488.         txtState.setText(state);
489.         txtName.setText(userEntity.getName());
490.         txtEmail.setText(userEntity.getEmail());
491.         txtUser.setText(userEntity.getUser());
492.         txtPass.setText(userEntity.getPass());
493.         txtAge.setText(userEntity.getAge());
494.     }
495.
496.     public void validateRFC(){
497.         if(methodsSql.ValidateRFC(txtSearchUsers.getText())){
498.             iconRFCExist.setText(" » RFC encontrado");
499.             iconRFCExist.setIcon(green);
500.             check1 = true;
501.         } else {
502.             iconRFCExist.setText(" » RFC no encontrado");
503.             iconRFCExist.setIcon(red);
504.             check1 = false;
505.             lock();
506.         }
507.
508.         if(txtSearchUsers.getText().length() < 12){
509.             iconRFCCharacter.setText(" » Minimo 12 caracteres");
510.             iconRFCCharacter.setIcon(red);
511.             check2 = false;
512.             lock();
513.         } else {
514.             iconRFCCharacter.setIcon(green);
515.             check2 = true;
516.         }
517.     }
518.
519.     public void DeleteUserValidate(){
```

```

520.         int i = JOptionPane.showConfirmDialog(null, "¿REALMENTE
DESEAS ELIMINAR EL USUARIO?", "¡ALERTA!",
JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE);
521.         if(i == 0){
522.             methodsSql.DeleteUserAdmin(txtSearchUsers.getText());
523.             clear();
524.         }
525.     }
526.
527.
528.
529.     // Variables declaration - do not modify
530.     private javax.swing.JPanel PanelButtons;
531.     private javax.swing.JPanel PanelDelete;
532.     private javax.swing.JPanel PanelInfo;
533.     private javax.swing.JLabel UserLabel1;
534.     private javax.swing.JButton btnClear;
535.     private javax.swing.JButton btnDelete;
536.     private javax.swing.JLabel iconRFCCharacter;
537.     private javax.swing.JLabel iconRFCExist;
538.     private javax.swing.JLabel jLabel1;
539.     private javax.swing.JLabel jLabel2;
540.     private javax.swing.JLabel jLabel3;
541.     private javax.swing.JLabel jLabel4;
542.     private javax.swing.JLabel jLabel5;
543.     private javax.swing.JLabel jLabel6;
544.     private javax.swing.JLabel jLabel7;
545.     private javax.swing.JLabel jLabel8;
546.     private javax.swing.JLabel jLabel9;
547.     private javax.swing.JSeparator jSeparator1;
548.     private javax.swing.JTextField txtAge;
549.     private javax.swing.JTextField txtEmail;
550.     private javax.swing.JTextField txtName;
551.     private javax.swing.JTextField txtPass;
552.     private javax.swing.JTextField txtRFC;
553.     private javax.swing.JTextField txtSearchUsers;
554.     private javax.swing.JTextField txtState;
555.     private javax.swing.JTextField txtUser;
556.     // End of variables declaration
557. }

```

PanelSearch

```

1.  /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.io.File;
10. import javax.swing.ImageIcon;
11. import javax.swing.JFileChooser;

```

```

12.
13.  /**
14.   *
15.   * @author Restf
16.   */
17.  public class PanelSearch extends javax.swing.JPanel {
18.
19.      MethodsSQL methodsSql = new MethodsSQL();
20.      ImageIcon green = new
ImageIcon(getClass().getResource("/resource/Check-True.png"));
21.      ImageIcon red = new
ImageIcon(getClass().getResource("/resource/Check-False.png"));
22.      private boolean check1, check2;
23.
24.      public PanelSearch() {
25.          initComponents();
26.          TableData.setModel(methodsSql.loadTableUser());
27.
28.      }
29.
30.
31.      @SuppressWarnings("unchecked")
32.      // <editor-fold defaultstate="collapsed" desc="Generated
Code">
33.      private void initComponents() {
34.
35.          jPanel1 = new javax.swing.JPanel();
36.          jPanel2 = new javax.swing.JPanel();
37.          jScrollPane1 = new javax.swing.JScrollPane();
38.          TableData = new javax.swing.JTable();
39.          txtSearchUsers = new javax.swing.JTextField();
40.          RFCExistsSearch = new javax.swing.JLabel();
41.          jButton2 = new javax.swing.JButton();
42.          jLabel8 = new javax.swing.JLabel();
43.
44.          jPanel1.setBackground(new java.awt.Color(128, 153, 155));
45.          jPanel1.setPreferredSize(new java.awt.Dimension(982,
641));
46.          jPanel1.setLayout(null);
47.
48.          jPanel2.setBackground(new java.awt.Color(128, 153, 155));
49.
jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "» Ingrese RFC de usuario para actualizar datos",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Dialog", 1, 18), new java.awt.Color(255, 255,
255))); // NOI18N
50.
51.          TableData.setFont(new java.awt.Font("Dialog", 0, 16)); //
NOI18N
52.          TableData.setModel(new
javax.swing.table.DefaultTableModel(
53.              new Object [][] {
54.                  {null, null, null, null, null, null, null, null},
55.                  {null, null, null, null, null, null, null, null},
56.                  {null, null, null, null, null, null, null, null},

```

```

57.         {null, null, null, null, null, null, null, null}
58.     },
59.     new String [] {
60.         "ID", "ESTADO", "NOMBRE", "CORREO", "RFC",
        "USUARIO", "CONTRASEÑA", "NACIMIENTO"
61.     }
62. ) {
63.     boolean[] canEdit = new boolean [] {
64.         false, false, false, false, false, false, false, false,
        false
65.     };
66.
67.     public boolean isCellEditable(int rowIndex, int
        columnIndex) {
68.         return canEdit [columnIndex];
69.     }
70. };
71. jScrollPane1.setViewportViewView(TableData);
72.
73. txtSearchUsers.setBackground(new java.awt.Color(255, 255,
        255));
74. txtSearchUsers.setFont(new java.awt.Font("Dialog", 0,
        16)); // NOI18N
75. txtSearchUsers.setForeground(new java.awt.Color(0, 0, 0));
76. txtSearchUsers.setPreferredSize(new
        java.awt.Dimension(220, 40));
77. txtSearchUsers.addKeyListener(new
        java.awt.event.KeyAdapter() {
78.     public void keyReleased(java.awt.event.KeyEvent evt) {
79.         txtSearchUsersKeyReleased(evt);
80.     }
81.     public void keyTyped(java.awt.event.KeyEvent evt) {
82.         txtSearchUsersKeyTyped(evt);
83.     }
84. });
85.
86. RFCExistsSearch.setFont(new java.awt.Font("Dialog", 1,
        18)); // NOI18N
87. RFCExistsSearch.setForeground(new java.awt.Color(255, 255,
        255));
88. RFCExistsSearch.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/Check-
        False.png"))); // NOI18N
89. RFCExistsSearch.setText(" » RFC no encontrado");
90. RFCExistsSearch.setFocusable(false);
91.
92. jButton2.setText("Generar PDF");
93. jButton2.addActionListener(new
        java.awt.event.ActionListener() {
94.     public void actionPerformed(java.awt.event.ActionEvent
        evt) {
95.         jButton2ActionPerformed(evt);
96.     }
97. });
98.
99. javax.swing.GroupLayout jPanel2Layout = new
        javax.swing.GroupLayout(jPanel2);

```



```

100.         jPanel2.setLayout (jPanel2Layout);
101.         jPanel2Layout.setHorizontalGroup(
102.             jPanel2Layout.createParallelGroup (javax.swing.GroupLayout.Alignment
103.                 .LEADING)
104.                 .addGroup (jPanel2Layout.createSequentialGroup ())
105.                 .addContainerGap ()
106.                 .addGroup (jPanel2Layout.createParallelGroup (javax.swing.GroupLayout
107.                     .Alignment.LEADING)
108.                     .addGroup (jPanel2Layout.createSequentialGroup ())
109.                     .addComponent (jScrollPane1,
110.                         javax.swing.GroupLayout.DEFAULT_SIZE, 933, Short.MAX_VALUE)
111.                     .addContainerGap ())
112.                     .addGroup (jPanel2Layout.createSequentialGroup ())
113.                     .addComponent (txtSearchUsers,
114.                         javax.swing.GroupLayout.PREFERRED_SIZE,
115.                         javax.swing.GroupLayout.DEFAULT_SIZE,
116.                         javax.swing.GroupLayout.PREFERRED_SIZE)
117.                     .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
118.                         ED)
119.                     .addComponent (RFCEExistsSearch)
120.                     .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
121.                         , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
122.                     .addComponent (jButton2)
123.                     .addGap (27, 27, 27)))
124.             );
125.         jPanel2Layout.setVerticalGroup(
126.             jPanel2Layout.createParallelGroup (javax.swing.GroupLayout.Alignment
127.                 .LEADING)
128.                 .addGroup (javax.swing.GroupLayout.Alignment.TRAILING,
129.                     jPanel2Layout.createSequentialGroup ())
130.                 .addContainerGap ()
131.                 .addGroup (jPanel2Layout.createParallelGroup (javax.swing.GroupLayout
132.                     .Alignment.BASELINE)
133.                     .addComponent (txtSearchUsers,
134.                         javax.swing.GroupLayout.PREFERRED_SIZE,
135.                         javax.swing.GroupLayout.DEFAULT_SIZE,
136.                         javax.swing.GroupLayout.PREFERRED_SIZE)
137.                     .addComponent (RFCEExistsSearch)
138.                     .addComponent (jButton2)
139.                     .addPreferredGap (javax.swing.LayoutStyle.ComponentPlacement.RELATED
140.                         )
141.                     .addComponent (jScrollPane1,
142.                         javax.swing.GroupLayout.DEFAULT_SIZE, 491, Short.MAX_VALUE)
143.                     .addContainerGap ())
144.             );
145.         jPanel1.add (jPanel2);
146.         jPanel2.setBounds (15, 50, 955, 580);

```

```

132.
133.         jLabel8.setFont(new java.awt.Font("Dialog", 1, 36)); //
NOI18N
134.         jLabel8.setForeground(new java.awt.Color(255, 255, 255));
135.         jLabel8.setText("CONSULTA DE USUARIOS");
136.         jPanel1.add(jLabel8);
137.         jLabel8.setBounds(251, 0, 475, 59);
138.
139.         javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(this);
140.         this.setLayout(layout);
141.         layout.setHorizontalGroup(
142.             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
143.                 .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE, 996, Short.MAX_VALUE)
144.             );
145.         layout.setVerticalGroup(
146.             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
147.                 .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE, 645, Short.MAX_VALUE)
148.             );
149.         } // </editor-fold>
150.
151.         private void txtSearchUsersKeyReleased(java.awt.event.KeyEvent
evt) {
152.             validateRFCsearch();
153.             MayusculasRFC(txtSearchUsers);
154.             TableData.setModel(methodsSql.loadTableUserFilter(txtSearchUsers.ge
tText()));
155.             if(txtSearchUsers.getText().equals("")){
156.                 clear();
157.             }
158.         }
159.
160.         private void txtSearchUsersKeyTyped(java.awt.event.KeyEvent
evt) {
161.             int characterLimit=13;
162.             if(txtSearchUsers.getText().length()>=characterLimit){
163.                 evt.consume();
164.             }
165.         }
166.
167.         private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
168.             String url = null;
169.             JFileChooser dgl= new JFileChooser();
170.             int o=dgl.showSaveDialog(this);
171.             if(o== JFileChooser.APPROVE_OPTION){
172.                 File f= dgl.getSelectedFile();
173.                 url =f.toString();
174.             }
175.             methodsSql.pdfCreateTableUsers(url);

```

```

176.     }
177.
178.     public void MayusculasRFC(javax.swing.JTextField jTextFieldS){
179.         String cadena = (txtSearchUsers.getText()).toUpperCase();
180.         txtSearchUsers.setText(cadena);
181.     }
182.
183.     public void validateRFCsearch(){
184.         if(methodsSql.ValidateRFCicon(txtSearchUsers.getText())){
185.             RFCExistsSearch.setText(" » RFC encontrado");
186.             RFCExistsSearch.setIcon(green);
187.         } else {
188.             RFCExistsSearch.setText(" » RFC no encontrado");
189.             RFCExistsSearch.setIcon(red);
190.         }
191.     }
192.
193.     public void clear(){
194.         RFCExistsSearch.setIcon(red);
195.         RFCExistsSearch.setText(" » RFC no encontrado");
196.         txtSearchUsers.setText(null);
197.         TableData.setModel(methodsSql.loadTableUser());
198.     }
199.
200.     // Variables declaration - do not modify
201.     private javax.swing.JLabel RFCExistsSearch;
202.     private javax.swing.JTable TableData;
203.     private javax.swing.JButton jButton2;
204.     private javax.swing.JLabel jLabel8;
205.     private javax.swing.JPanel jPanel1;
206.     private javax.swing.JPanel jPanel2;
207.     private javax.swing.JScrollPane jScrollPane1;
208.     private javax.swing.JTextField txtSearchUsers;
209.     // End of variables declaration
210. }

```

Sales

```

1.  /*
2.   * To change this license header, choose License Headers in Project
   * Properties.
3.   * To change this template file, choose Tools | Templates
4.   * and open the template in the editor.
5.   */
6.  package view.admin;
7.
8.  import code.MethodsSQL;
9.  import java.io.File;
10. import javax.swing.JFileChooser;
11. import view.Login;
12.
13. /**
14.  *
15.  * @author Restf
16.  */

```

```

17. public class Sales extends javax.swing.JFrame {
18.
19.     MethodsSQL methodsSql = new MethodsSQL();
20.
21.     public Sales() {
22.         initComponents();
23.         setTitle("Car Agency - Ventas: [
"+Login.userEntity.getName()+" ]");
24.         setLocationRelativeTo(null);
25.         TableData.setModel(methodsSql.loadTableSales());
26.     }
27.
28.     /**
29.      * This method is called from within the constructor to
        initialize the form.
30.      * WARNING: Do NOT modify this code. The content of this
        method is always
31.      * regenerated by the Form Editor.
32.      */
33.     @SuppressWarnings("unchecked")
34.     // <editor-fold defaultstate="collapsed" desc="Generated
        Code">
35.     private void initComponents() {
36.
37.         jPanel1 = new javax.swing.JPanel();
38.         jLabel7 = new javax.swing.JLabel();
39.         jScrollPane1 = new javax.swing.JScrollPane();
40.         TableData = new javax.swing.JTable();
41.         txtDataFilter = new javax.swing.JTextField();
42.         jLabel11 = new javax.swing.JLabel();
43.         btnSelectData = new javax.swing.JComboBox<>();
44.         jButton2 = new javax.swing.JButton();
45.
46.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        ;
47.         setResizable(false);
48.
49.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
50.         jPanel1.setMinimumSize(new java.awt.Dimension(1280, 720));
51.
52.         jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
        NOI18N
53.         jLabel7.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
        ); // NOI18N
54.         jLabel7.setText("VOLVER");
55.         jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
        {
56.             public void mouseClicked(java.awt.event.MouseEvent
        evt) {
57.                 jLabel7MouseClicked(evt);
58.             }
59.         });
60.
61.         TableData.setFont(new java.awt.Font("Dialog", 0, 16)); //
        NOI18N

```

```

62.         TableData.setModel(new
javax.swing.table.DefaultTableModel(
63.             new Object [][] {
64.                 {null, "", null, null, null, null, null, null,
null, null},
65.                 {null, null, null, null, null, null, null, null,
null, null},
66.                 {null, null, null, null, null, null, null, null,
null, null},
67.                 {null, null, null, null, null, null, null, null,
null, null}
68.             },
69.             new String [] {
70.                 "ID", "FOLIO", "NOMBRE", "MODELO", "N° SERIE",
"PRECIO VEHICULO", "ADICIONALES", "PRECIO ADICIONAL", "PRECIO
TOTAL", "TIPO DE PAGO"
71.             }
72.         ) {
73.             boolean[] canEdit = new boolean [] {
74.                 false, false, false, false, false, false, false, false,
false, false, false
75.             };
76.
77.             public boolean isCellEditable(int rowIndex, int
columnIndex) {
78.                 return canEdit [columnIndex];
79.             }
80.         });
81.         jScrollPane1.setViewportViewView(TableData);
82.
83.         txtDataFilter.setFont(new java.awt.Font("Dialog", 0, 16));
// NOI18N
84.         txtDataFilter.setEnabled(false);
85.         txtDataFilter.setPreferredSize(new java.awt.Dimension(220,
40));
86.         txtDataFilter.addKeyListener(new
java.awt.event.KeyAdapter() {
87.             public void keyReleased(java.awt.event.KeyEvent evt) {
88.                 txtDataFilterKeyReleased(evt);
89.             }
90.             public void keyTyped(java.awt.event.KeyEvent evt) {
91.                 txtDataFilterKeyTyped(evt);
92.             }
93.         });
94.
95.         jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
96.         jLabel1.setForeground(new java.awt.Color(255, 255, 255));
97.         jLabel1.setText(" » Selecciona el dato a buscar:");
98.
99.         btnSelectData.setFont(new java.awt.Font("Dialog", 1, 16));
// NOI18N
100.        btnSelectData.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "SELECCIONAR", "•
Identificador", "• Folio", "• Nombre", "• IModelo", "• N° Serie",
"• Precio Vehiculo", "• Adicionales", "• Precio Adicional", "•
Precio Total", "• Tipo de Pago" }));

```

```

101.         btnSelectData.setPreferredSize(new java.awt.Dimension(220,
40));
102.         btnSelectData.addActionListener(new
java.awt.event.ActionListener() {
103.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
104.                 btnSelectDataActionPerformed(evt);
105.             }
106.         });
107.
108.         jButton2.setText("Generar PDF");
109.         jButton2.addActionListener(new
java.awt.event.ActionListener() {
110.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
111.                 jButton2ActionPerformed(evt);
112.             }
113.         });
114.
115.         javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
116.         jPanel1.setLayout(jPanel1Layout);
117.         jPanel1Layout.setHorizontalGroup(
118.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
119.                 .addGroup(jPanel1Layout.createSequentialGroup()
120.                     .addContainerGap()
121.                     .addComponent(jLabel7)
122.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
896, Short.MAX_VALUE)
123.                     .addComponent(jButton2)
124.                     .addGap(103, 103, 103)
125.                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
126.                         .addGroup(jPanel1Layout.createSequentialGroup()
127.                             .addGap(40, 40, 40)
128.                             .addGroup(jPanel1Layout.createParallelGroup(javax.swing
.GroupLayout.Alignment.LEADING)
129.                                 .addGroup(jPanel1Layout.createSequentialGroup()
130.                                     .addGap(178, 178, 178)
131.                                     .addComponent(jLabel11)
132.                                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement
UNRELATED)
133.                                     .addComponent(btnSelectData,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
134.                                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement
UNRELATED)

```

```

135.                .addComponent(txtDataFilter,
    javax.swing.GroupLayout.PREFERRED_SIZE, 328,
    javax.swing.GroupLayout.PREFERRED_SIZE))
136.                .addComponent(jScrollPane1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 1200,
    javax.swing.GroupLayout.PREFERRED_SIZE))
137.                .addContainerGap(40, Short.MAX_VALUE)))
138.            );
139.            jPanell1Layout.setVerticalGroup(
140.                jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
141.                .addGroup(jPanell1Layout.createSequentialGroup())
142.                .addContainerGap())
143.                .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
144.                .addComponent(jLabel7,
    javax.swing.GroupLayout.PREFERRED_SIZE, 66,
    javax.swing.GroupLayout.PREFERRED_SIZE)
145.                .addComponent(jButton2))
146.                .addContainerGap(648, Short.MAX_VALUE))
147.                .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
148.                .addGroup(jPanell1Layout.createSequentialGroup())
149.                .addGap(25, 25, 25)
150.                .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
151.                .addComponent(jLabel1)
152.                .addComponent(btnSelectData,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
153.                .addComponent(txtDataFilter,
    javax.swing.GroupLayout.PREFERRED_SIZE, 41,
    javax.swing.GroupLayout.PREFERRED_SIZE))
154.                .addGap(18, 18, 18)
155.                .addComponent(jScrollPane1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 611,
    javax.swing.GroupLayout.PREFERRED_SIZE)
156.                .addContainerGap(25, Short.MAX_VALUE)))
157.            );
158.
159.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
160.            getContentPane().setLayout(layout);
161.            layout.setHorizontalGroup(
162.                layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
163.                .addComponent(jPanell1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
164.            );

```

```

165.         layout.setVerticalGroup(
166.             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
167.                 .addComponent(jPanell1,
168.                     javax.swing.GroupLayout.PREFERRED_SIZE,
169.                     javax.swing.GroupLayout.DEFAULT_SIZE,
170.                     javax.swing.GroupLayout.PREFERRED_SIZE)
171.             );
172.         pack();
173.     } // </editor-fold>
174.
175.     private void jLabel7MouseClicked(java.awt.event.MouseEvent
176.         evt) {
177.         this.setVisible(false);
178.         MenuAdmin menu = new MenuAdmin();
179.         menu.setVisible(true);
180.         dispose();
181.     }
182.
183.     private void txtDataFilterKeyReleased(java.awt.event.KeyEvent
184.         evt) {
185.         MayusculasRFC(txtDataFilter);
186.
187.         TableData.setModel(methodsSql.loadTableSalesFilter(btnSelectData.ge
188.             tSelectedIndex(), txtDataFilter.getText()));
189.     }
190.
191.     private void txtDataFilterKeyTyped(java.awt.event.KeyEvent
192.         evt) {
193.         // TODO add your handling code here:
194.     }
195.
196.     private void
197.         btnSelectDataActionPerformed(java.awt.event.ActionEvent evt) {
198.         if (btnSelectData.getSelectedIndex() > 0) {
199.             txtDataFilter.setEnabled(true);
200.         } else {
201.             txtDataFilter.setEnabled(false);
202.         }
203.         clear();
204.     }
205.
206.     private void
207.         jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
208.         String url = null;
209.         JFileChooser dgl= new JFileChooser();
210.         int o=dgl.showSaveDialog(this);
211.         if(o== JFileChooser.APPROVE_OPTION){
212.             File f= dgl.getSelectedFile();
213.             url =f.toString();
214.         }
215.         methodsSql.pdfCreateTableSales(url,
216.             btnSelectData.getSelectedIndex(), txtDataFilter.getText());
217.     }
218.

```



```

209.     public void MayusculasRFC(javax.swing.JTextField jTextFieldS){
210.         String cadena = (txtDataFilter.getText()).toUpperCase();
211.         txtDataFilter.setText(cadena);
212.     }
213.
214.     public void clear(){
215.         txtDataFilter.setText(null);
216.         TableData.setModel(methodsSql.loadTableSales());
217.     }
218.
219.     /**
220.      * @param args the command line arguments
221.      */
222.     public static void main(String args[]) {
223.         /* Set the Nimbus look and feel */
224.         //<editor-fold defaultstate="collapsed" desc=" Look and
        feel setting code (optional) ">
225.         /* If Nimbus (introduced in Java SE 6) is not available,
        stay with the default look and feel.
226.          * For details see
        http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf
        .html
227.          */
228.         try {
229.             for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
230.                 if ("Nimbus".equals(info.getName())) {
231.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
232.                     break;
233.                 }
234.             }
235.         } catch (ClassNotFoundException ex) {
236.             java.util.logging.Logger.getLogger(Sales.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
237.         } catch (InstantiationException ex) {
238.             java.util.logging.Logger.getLogger(Sales.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
239.         } catch (IllegalAccessException ex) {
240.             java.util.logging.Logger.getLogger(Sales.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
241.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
242.             java.util.logging.Logger.getLogger(Sales.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
243.         }
244.         //</editor-fold>
245.
246.         /* Create and display the form */
247.         java.awt.EventQueue.invokeLater(new Runnable() {
248.             public void run() {
249.                 new Sales().setVisible(true);
250.             }
251.         });

```

```

252.     }
253.
254.     // Variables declaration - do not modify
255.     private javax.swing.JTable TableData;
256.     private javax.swing.JComboBox<String> btnSelectData;
257.     private javax.swing.JButton jButton2;
258.     private javax.swing.JLabel jLabel1;
259.     private javax.swing.JLabel jLabel7;
260.     private javax.swing.JPanel jPanel1;
261.     private javax.swing.JScrollPane jScrollPane1;
262.     private javax.swing.JTextField txtDataFilter;
263.     // End of variables declaration
264. }

```

Users

```

1. package view.admin;
2.
3. /*
4.  * To change this license header, choose License Headers in Project
  Properties.
5.  * To change this template file, choose Tools | Templates
6.  * and open the template in the editor.
7.  */
8.
9.
10. import java.awt.CardLayout;
11. import javax.swing.SwingUtilities;
12. import view.Login;
13.
14. /**
15.  *
16.  * @author Restf
17.  */
18. public class Users extends javax.swing.JFrame {
19.
20.     PanelSearch searchPanel = new PanelSearch();
21.     PanelEdit editPanel = new PanelEdit();
22.     PanelAdd addPanel = new PanelAdd();
23.     PanelDelete deletePanel = new PanelDelete();
24.
25.     CardLayout view;
26.
27.
28.     public Users() {
29.         initComponents();
30.         setTitle("Car Agency - Lista de Usuarios - [
31. "+Login.userEntity.getName()+" ]");
32.         setLocationRelativeTo(null);
33.         view = (CardLayout) ContainerPanel.getLayout();
34.     }
35.
36.     /**

```

```

37.      * This method is called from within the constructor to
      initialize the form.
38.      * WARNING: Do NOT modify this code. The content of this
      method is always
39.      * regenerated by the Form Editor.
40.      */
41.      @SuppressWarnings("unchecked")
42.      // <editor-fold defaultstate="collapsed" desc="Generated
      Code">
43.      private void initComponents() {
44.
45.          jPanel1 = new javax.swing.JPanel();
46.          btnBack = new javax.swing.JLabel();
47.          MenuButton = new javax.swing.JPanel();
48.          jPanel2 = new javax.swing.JPanel();
49.          btnSearch = new javax.swing.JToggleButton();
50.          jPanel6 = new javax.swing.JPanel();
51.          btnEdit = new javax.swing.JToggleButton();
52.          jPanel7 = new javax.swing.JPanel();
53.          btnAdd = new javax.swing.JToggleButton();
54.          jPanel3 = new javax.swing.JPanel();
55.          btnDelete = new javax.swing.JToggleButton();
56.          ContainerPanel = new javax.swing.JPanel();
57.
58.          setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
59.          ;
60.          setMinimumSize(new java.awt.Dimension(1280, 720));
61.          setResizable(false);
62.
63.          jPanel1.setBackground(new java.awt.Color(128, 153, 155));
64.          jPanel1.setName(""); // NOI18N
65.          jPanel1.setPreferredSize(new java.awt.Dimension(1280,
66.          820));
67.
68.          btnBack.setFont(new java.awt.Font("Dialog", 1, 24)); //
        NOI18N
69.          btnBack.setForeground(new java.awt.Color(0, 0, 0));
70.          btnBack.setIcon(new
71.          javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
72.          ); // NOI18N
73.          btnBack.setText("VOLVER");
74.          btnBack.addMouseListener(new java.awt.event.MouseAdapter()
75.          {
76.              public void mouseClicked(java.awt.event.MouseEvent
77.              evt) {
78.                  btnBackMouseClicked(evt);
79.              }
80.          });
81.
82.          MenuButton.setBackground(new java.awt.Color(128, 153,
83.          155));
84.
85.          jPanel2.setBackground(new java.awt.Color(128, 153, 155));
86.
87.          jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
88.          .swing.border.BevelBorder.RAISED));

```

```

80.
81.         btnSearch.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
82.         btnSearch.setForeground(new java.awt.Color(255, 255,
    255));
83.         btnSearch.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
    SubMenuUser-search-off.png"))); // NOI18N
84.         btnSearch.setText("BUSCAR");
85.         btnSearch.setContentAreaFilled(false);
86.
    btnSearch.setHorizontalAlignment(javax.swing.SwingConstants.LEADING
    );
87.         btnSearch.setRolloverIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
    SubMenuUser-search-on.png"))); // NOI18N
88.         btnSearch.addActionListener(new
    java.awt.event.ActionListener() {
89.             public void actionPerformed(java.awt.event.ActionEvent
    evt) {
90.                 btnSearchActionPerformed(evt);
91.             }
92.         });
93.
94.         javax.swing.GroupLayout jPanel2Layout = new
    javax.swing.GroupLayout(jPanel2);
95.         jPanel2.setLayout(jPanel2Layout);
96.         jPanel2Layout.setHorizontalGroup(
97.             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
98.                 .addGroup(jPanel2Layout.createSequentialGroup()
99.                     .addComponent(btnSearch,
    javax.swing.GroupLayout.PREFERRED_SIZE, 221,
    javax.swing.GroupLayout.PREFERRED_SIZE)
100.                    .addGroup(0, 0, Short.MAX_VALUE))
101.                 );
102.         jPanel2Layout.setVerticalGroup(
103.             jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
104.                 .addComponent(btnSearch,
    javax.swing.GroupLayout.DEFAULT_SIZE, 101, Short.MAX_VALUE)
105.                 );
106.
107.         jPanel6.setBackground(new java.awt.Color(128, 153, 155));
108.
    jPanel6.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
    .swing.border.BevelBorder.RAISED));
109.
110.         btnEdit.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
111.         btnEdit.setForeground(new java.awt.Color(255, 255, 255));
112.         btnEdit.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
    SubMenuUser-edit-off.png"))); // NOI18N
113.         btnEdit.setText("MODIFICAR");

```

```

114.         btnEdit.setContentAreaFilled(false);
115.
116.         btnEdit.setHorizontalAlignment(javax.swing.SwingConstants.LEADING);
117.         btnEdit.setRolloverIcon(new
118.             javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
119.                 SubMenuUser-edit-on.png"))); // NOI18N
120.         btnEdit.addActionListener(new
121.             java.awt.event.ActionListener() {
122.                 public void actionPerformed(java.awt.event.ActionEvent
123.                     evt) {
124.                     btnEditActionPerformed(evt);
125.                 }
126.             });
127.
128.         javax.swing.GroupLayout jPanel6Layout = new
129.             javax.swing.GroupLayout(jPanel6);
130.         jPanel6.setLayout(jPanel6Layout);
131.         jPanel6Layout.setHorizontalGroup(
132.             jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
133.                 .LEADING)
134.                 .addGroup(jPanel6Layout.createSequentialGroup()
135.                     .addComponent(btnEdit,
136.                         javax.swing.GroupLayout.PREFERRED_SIZE, 220,
137.                         javax.swing.GroupLayout.PREFERRED_SIZE)
138.                     .addGap(0, 0, Short.MAX_VALUE))
139.             );
140.         jPanel6Layout.setVerticalGroup(
141.             jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
142.                 .LEADING)
143.                 .addGroup(jPanel6Layout.createSequentialGroup()
144.                     .addComponent(btnEdit,
145.                         javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
146.                     );
147.
148.         jPanel7.setBackground(new java.awt.Color(128, 153, 155));
149.
150.         jPanel7.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
151.             .swing.border.BevelBorder.RAISED));
152.
153.         btnAdd.setFont(new java.awt.Font("Dialog", 1, 18)); //
154.             NOI18N
155.         btnAdd.setForeground(new java.awt.Color(255, 255, 255));
156.         btnAdd.setIcon(new
157.             javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
158.                 SubMenuUser-add-off.png"))); // NOI18N
159.         btnAdd.setText("AGREGAR");
160.         btnAdd.setContentAreaFilled(false);
161.
162.         btnAdd.setHorizontalAlignment(javax.swing.SwingConstants.LEADING);
163.         btnAdd.setRolloverIcon(new
164.             javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
165.                 SubMenuUser-add-on.png"))); // NOI18N
166.         btnAdd.addActionListener(new
167.             java.awt.event.ActionListener() {
168.                 public void actionPerformed(java.awt.event.ActionEvent
169.                     evt) {

```

```

148.         btnAddActionPerformed(evt);
149.     }
150. });
151.
152.     javax.swing.GroupLayout jPanel7Layout = new
    javax.swing.GroupLayout(jPanel7);
153.     jPanel7.setLayout(jPanel7Layout);
154.     jPanel7Layout.setHorizontalGroup(
155.         jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
156.             .addComponent(btnAdd,
    javax.swing.GroupLayout.DEFAULT_SIZE, 218, Short.MAX_VALUE)
157.         );
158.     jPanel7Layout.setVerticalGroup(
159.         jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
160.             .addComponent(btnAdd,
    javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
161.         );
162.
163.     jPanel3.setBackground(new java.awt.Color(128, 153, 155));
164.
165.     jPanel3.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
    .swing.border.BevelBorder.RAISED));
166.
167.     btnDelete.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
168.     btnDelete.setForeground(new java.awt.Color(255, 255,
    255));
169.     btnDelete.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
    SubMenuUser-delete-off.png"))); // NOI18N
170.     btnDelete.setText("ELIMINAR");
171.     btnDelete.setContentAreaFilled(false);
172.
173.     btnDelete.setHorizontalAlignment(javax.swing.SwingConstants.LEADING
    );
174.
175.     btnDelete.setRolloverIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
    SubMenuUser-delete-on.png"))); // NOI18N
176.
177.     btnDelete.addActionListener(new
    java.awt.event.ActionListener() {
178.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
179.             btnDeleteActionPerformed(evt);
180.         }
181.     });
182.
183.     javax.swing.GroupLayout jPanel3Layout = new
    javax.swing.GroupLayout(jPanel3);
184.     jPanel3.setLayout(jPanel3Layout);
185.     jPanel3Layout.setHorizontalGroup(
186.         jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)

```

```
183.         .addComponent(btnDelete,
184.             javax.swing.GroupLayout.DEFAULT_SIZE, 218, Short.MAX_VALUE)
185.         );
186.         jPanel3Layout.setVerticalGroup(
187.             jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
188.                 .LEADING)
189.             .addComponent(btnDelete,
190.                 javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
191.             );
192.         javax.swing.GroupLayout MenuButtonLayout = new
193.             javax.swing.GroupLayout(MenuButton);
194.         MenuButton.setLayout(MenuButtonLayout);
195.         MenuButtonLayout.setHorizontalGroup(
196.             MenuButtonLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
197.                 ent.LEADING)
198.                 .addGroup(MenuButtonLayout.createSequentialGroup())
199.                 .addContainerGap()
200.                 .addGroup(MenuButtonLayout.createParallelGroup(javax.swing.GroupLay
201.                     out.Alignment.LEADING)
202.                     .addComponent(jPanel2,
203.                         javax.swing.GroupLayout.PREFERRED_SIZE,
204.                         javax.swing.GroupLayout.DEFAULT_SIZE,
205.                         javax.swing.GroupLayout.PREFERRED_SIZE)
206.                     .addComponent(jPanel6,
207.                         javax.swing.GroupLayout.PREFERRED_SIZE,
208.                         javax.swing.GroupLayout.DEFAULT_SIZE,
209.                         javax.swing.GroupLayout.PREFERRED_SIZE)
210.                     .addComponent(jPanel7,
211.                         javax.swing.GroupLayout.PREFERRED_SIZE,
212.                         javax.swing.GroupLayout.DEFAULT_SIZE,
213.                         javax.swing.GroupLayout.PREFERRED_SIZE))
214.                 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
215.                     Short.MAX_VALUE))
216.                 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
217.                     MenuButtonLayout.createSequentialGroup())
218.                 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
219.                     Short.MAX_VALUE)
220.             );
221.         MenuButtonLayout.setVerticalGroup(
222.             MenuButtonLayout.createParallelGroup(javax.swing.GroupLayout.Alignm
223.                 ent.LEADING)
224.                 .addGroup(MenuButtonLayout.createSequentialGroup())
225.                 .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
226.                     Short.MAX_VALUE)
```

```

210.                .addComponent(jPanel2,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
211.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
212.                .addComponent(jPanel6,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
213.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
214.                .addComponent(jPanel7,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
215.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
    ED)
216.                .addComponent(jPanel3,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
217.                .addContainerGap()
218.            );
219.
220.            ContainerPanel.setBackground(new java.awt.Color(128, 153,
    155));
221.            ContainerPanel.setPreferredSize(new
    java.awt.Dimension(982, 641));
222.            ContainerPanel.setLayout(new java.awt.CardLayout());
223.
224.            javax.swing.GroupLayout jPanel1Layout = new
    javax.swing.GroupLayout(jPanel1);
225.            jPanel1.setLayout(jPanel1Layout);
226.            jPanel1Layout.setHorizontalGroup(
227.                jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
228.                .addGroup(jPanel1Layout.createSequentialGroup()
229.                    .addContainerGap()
230.                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
231.                        .addComponent(btnBack,
    javax.swing.GroupLayout.PREFERRED_SIZE, 180,
    javax.swing.GroupLayout.PREFERRED_SIZE)
232.                        .addComponent(MenuButton,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
233.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )

```



```

234.                .addComponent(ContainerPanel,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
235.                .addContainerGap(49, Short.MAX_VALUE))
236.            );
237.            jPanell1Layout.setVerticalGroup(
238.
    jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
239.                .addGroup(jPanell1Layout.createSequentialGroup())
240.                .addContainerGap()
241.                .addComponent(btnBack,
    javax.swing.GroupLayout.PREFERRED_SIZE, 60,
    javax.swing.GroupLayout.PREFERRED_SIZE)
242.                .addGap(59, 59, 59)
243.                .addComponent(MenuButton,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
244.                .addContainerGap(95, Short.MAX_VALUE))
245.                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanell1Layout.createSequentialGroup())
246.
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE)
247.                .addComponent(ContainerPanel,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
248.                .addGap(22, 22, 22))
249.            );
250.
251.            javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());
252.            getContentPane().setLayout(layout);
253.            layout.setHorizontalGroup(
254.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
255.                .addComponent(jPanell1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
256.            );
257.            layout.setVerticalGroup(
258.
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
259.                .addComponent(jPanell1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 685,
    javax.swing.GroupLayout.PREFERRED_SIZE)
260.            );
261.
262.            pack();
263.        } // </editor-fold>
264.

```

```

265.     private void btnBackMouseClicked(java.awt.event.MouseEvent
    evt) {
266.         this.setVisible(false);
267.         MenuAdmin menu = new MenuAdmin();
268.         menu.setVisible(true);
269.         dispose();
270.     }
271.
272.     private void
    btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
273.         if(btnSearch.isSelected()){
274.             ContainerPanel.add(searchPanel, "search");
275.             view.show(ContainerPanel, "search");
276.             SwingUtilities.updateComponentTreeUI(this);
277.             this.repaint();
278.             btnAdd.setEnabled(false);
279.             btnDelete.setEnabled(false);
280.             btnEdit.setEnabled(false);
281.             searchPanel.clear();
282.         } else {
283.             ContainerPanel.removeAll();
284.             SwingUtilities.updateComponentTreeUI(this);
285.             btnAdd.setEnabled(true);
286.             btnDelete.setEnabled(true);
287.             btnEdit.setEnabled(true);
288.         }
289.     }
290.
291.     private void btnEditActionPerformed(java.awt.event.ActionEvent
    evt) {
292.         if(btnEdit.isSelected()){
293.             ContainerPanel.add(editPanel, "edit");
294.             view.show(ContainerPanel, "edit");
295.             SwingUtilities.updateComponentTreeUI(this);
296.             this.repaint();
297.             btnSearch.setEnabled(false);
298.             btnAdd.setEnabled(false);
299.             btnDelete.setEnabled(false);
300.             editPanel.clear();
301.         } else {
302.             ContainerPanel.removeAll();
303.             SwingUtilities.updateComponentTreeUI(this);
304.             btnSearch.setEnabled(true);
305.             btnAdd.setEnabled(true);
306.             btnDelete.setEnabled(true);
307.         }
308.     }
309.
310.     private void btnAddActionPerformed(java.awt.event.ActionEvent
    evt) {
311.         if(btnAdd.isSelected()){
312.             ContainerPanel.add(addPanel, "add");
313.             view.show(ContainerPanel, "add");
314.             SwingUtilities.updateComponentTreeUI(this);
315.             this.repaint();
316.             btnSearch.setEnabled(false);
317.             btnDelete.setEnabled(false);

```

```

318.         btnEdit.setEnabled(false);
319.         addPanel.clear();
320.     } else {
321.         ContainerPanel.removeAll();
322.         SwingUtilities.updateComponentTreeUI(this);
323.         btnSearch.setEnabled(true);
324.         btnDelete.setEnabled(true);
325.         btnEdit.setEnabled(true);
326.     }
327. }
328.
329. private void
    btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
330.     if(btnDelete.isSelected()){
331.         ContainerPanel.add(deletePanel, "delete");
332.         view.show(ContainerPanel, "delete");
333.         SwingUtilities.updateComponentTreeUI(this);
334.         this.repaint();
335.         btnSearch.setEnabled(false);
336.         btnAdd.setEnabled(false);
337.         btnEdit.setEnabled(false);
338.         deletePanel.clear();
339.     } else {
340.         ContainerPanel.removeAll();
341.         SwingUtilities.updateComponentTreeUI(this);
342.         btnSearch.setEnabled(true);
343.         btnAdd.setEnabled(true);
344.         btnEdit.setEnabled(true);
345.     }
346. }
347.
348.
349. /**
350.  * @param args the command line arguments
351.  */
352. public static void main(String args[]) {
353.     /* Set the Nimbus look and feel */
354.     //<editor-fold defaultstate="collapsed" desc=" Look and
    feel setting code (optional) ">
355.     /* If Nimbus (introduced in Java SE 6) is not available,
    stay with the default look and feel.
356.     * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
357.     */
358.     try {
359.         for (javax.swing.UIManager.LookAndFeelInfo info :
    javax.swing.UIManager.getInstalledLookAndFeels()) {
360.             if ("Nimbus".equals(info.getName())) {
361.                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
362.                 break;
363.             }
364.         }
365.     } catch (ClassNotFoundException ex) {

```

```

366.     java.util.logging.Logger.getLogger(Users.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
367.     } catch (InstantiationException ex) {
368.     java.util.logging.Logger.getLogger(Users.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
369.     } catch (IllegalAccessException ex) {
370.     java.util.logging.Logger.getLogger(Users.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
371.     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
372.     java.util.logging.Logger.getLogger(Users.class.getName()).log(java.
        util.logging.Level.SEVERE, null, ex);
373.     }
374.     //</editor-fold>
375.     //</editor-fold>
376.     //</editor-fold>
377.     //</editor-fold>
378.
379.     /* Create and display the form */
380.     java.awt.EventQueue.invokeLater(new Runnable() {
381.         public void run() {
382.             new Users().setVisible(true);
383.         }
384.     });
385. }
386.
387. // Variables declaration - do not modify
388. private javax.swing.JPanel ContainerPanel;
389. private javax.swing.JPanel MenuButton;
390. private javax.swing.JToggleButton btnAdd;
391. private javax.swing.JLabel btnBack;
392. private javax.swing.JToggleButton btnDelete;
393. private javax.swing.JToggleButton btnEdit;
394. private javax.swing.JToggleButton btnSearch;
395. private javax.swing.JPanel jPanel1;
396. private javax.swing.JPanel jPanel2;
397. private javax.swing.JPanel jPanel3;
398. private javax.swing.JPanel jPanel6;
399. private javax.swing.JPanel jPanel7;
400. // End of variables declaration
401. }

```

MenuUser

```

1. /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6. package view.user;
7.

```

```

8. import view.Login;
9. import view.Menu;
10. import view.ModifyUser;
11.
12. /**
13.  *
14.  * @author serk9
15.  */
16. public class MenuUser extends javax.swing.JFrame {
17.     static Boolean ReloadData;
18.     /**
19.      * Creates new form prin
20.      */
21.     public MenuUser() {
22.         initComponents();
23.         setTitle("Car Agency - Menu de Usuario - [
24. "+Login.userEntity.getName()+" ]");
25.         setLocationRelativeTo(null);
26.     }
27.
28.     /**
29.      * This method is called from within the constructor to
30.      initialize the form.
31.      * WARNING: Do NOT modify this code. The content of this
32.      method is always
33.      * regenerated by the Form Editor.
34.      */
35.     @SuppressWarnings("unchecked")
36.     // <editor-fold defaultstate="collapsed" desc="Generated
37.     Code">
38.     private void initComponents() {
39.
40.         jPanel2 = new javax.swing.JPanel();
41.         leave = new javax.swing.JLabel();
42.         jLabel7 = new javax.swing.JLabel();
43.         jLabel9 = new javax.swing.JLabel();
44.         jLabel10 = new javax.swing.JLabel();
45.         jLabel11 = new javax.swing.JLabel();
46.         jPanel1 = new javax.swing.JPanel();
47.         jButton1 = new javax.swing.JButton();
48.         jPanel3 = new javax.swing.JPanel();
49.         btnVehicle = new javax.swing.JButton();
50.         jPanel7 = new javax.swing.JPanel();
51.         btnQuotation = new javax.swing.JButton();
52.         jPanel6 = new javax.swing.JPanel();
53.         btnPurchase = new javax.swing.JButton();
54.
55.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
56.         ;
57.         setResizable(false);
58.
59.         jPanel2.setBackground(new java.awt.Color(128, 153, 155));
60.         jPanel2.setPreferredSize(new java.awt.Dimension(1280,
61. 720));

```

```

58.         leave.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
59.         leave.setForeground(new java.awt.Color(0, 0, 0));
60.         leave.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
        ); // NOI18N
61.         leave.setText("CERRAR SESION");
62.         leave.addMouseListener(new java.awt.event.MouseAdapter() {
63.             public void mouseClicked(java.awt.event.MouseEvent
        evt) {
64.                 leaveMouseClicked(evt);
65.             }
66.         });
67.
68.         jLabel7.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/user-
        128px.png"))); // NOI18N
69.
70.         jLabel9.setFont(new java.awt.Font("Dialog", 1, 36)); //
NOI18N
71.         jLabel9.setForeground(new java.awt.Color(255, 255, 255));
72.         jLabel9.setText("BIENVENIDO");
73.
74.         jLabel10.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
75.         jLabel10.setForeground(new java.awt.Color(255, 255, 255));
76.         jLabel10.setText(">> Nombre: "+Login.userEntity.getName());
77.
78.         jLabel11.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
79.         jLabel11.setForeground(new java.awt.Color(255, 255, 255));
80.         jLabel11.setText(">> Usuario:
        "+Login.userEntity.getUser());
81.
82.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
83.
        jPanel1.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
        .swing.border.BevelBorder.RAISED));
84.
85.         jButton1.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
86.         jButton1.setForeground(new java.awt.Color(255, 255, 255));
87.         jButton1.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
        edit-off.png"))); // NOI18N
88.         jButton1.setText("CUENTA");
89.         jButton1.setContentAreaFilled(false);
90.
        jButton1.setHorizontalAlignment(javax.swing.SwingConstants.LEADING)
        ;
91.         jButton1.setPreferredSize(new java.awt.Dimension(236,
        144));
92.         jButton1.setRolloverIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/MenuAdmin-
        edit-on.png"))); // NOI18N
93.         jButton1.addActionListener(new
        java.awt.event.ActionListener() {

```

```

94.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
95.             jButton1ActionPerformed(evt);
96.         }
97.     });
98.
99.     javax.swing.GroupLayout jPanel1Layout = new
    javax.swing.GroupLayout(jPanel1);
100.     jPanel1.setLayout(jPanel1Layout);
101.     jPanel1Layout.setHorizontalGroup(
102.         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
103.             .addComponent(jButton1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 293, Short.MAX_VALUE)
104.         );
105.     jPanel1Layout.setVerticalGroup(
106.         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
107.             .addComponent(jButton1,
    javax.swing.GroupLayout.DEFAULT_SIZE, 152, Short.MAX_VALUE)
108.         );
109.
110.     jPanel3.setBackground(new java.awt.Color(128, 153, 155));
111.
112.     jPanel3.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
    .swing.border.BevelBorder.RAISED));
113.     btnVehicle.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
114.     btnVehicle.setForeground(new java.awt.Color(255, 255,
    255));
115.     btnVehicle.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
    vehicle-off.png"))); // NOI18N
116.     btnVehicle.setText(" VEHICULOS");
117.     btnVehicle.setContentAreaFilled(false);
118.
119.     btnVehicle.setHorizontalAlignment(javax.swing.SwingConstants.LEADIN
    G);
120.     btnVehicle.setRolloverIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
    vehicle-on.png"))); // NOI18N
121.     btnVehicle.addActionListener(new
    java.awt.event.ActionListener() {
122.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
123.             btnVehicleActionPerformed(evt);
124.         }
125.     });
126.     javax.swing.GroupLayout jPanel3Layout = new
    javax.swing.GroupLayout(jPanel3);
127.     jPanel3.setLayout(jPanel3Layout);
128.     jPanel3Layout.setHorizontalGroup(

```

```

129.        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
130.            .addComponent(btnVehicle,
        javax.swing.GroupLayout.DEFAULT_SIZE, 295, Short.MAX_VALUE)
131.        );
132.        jPanel3Layout.setVerticalGroup(
133.
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
134.            .addComponent(btnVehicle,
        javax.swing.GroupLayout.DEFAULT_SIZE, 152, Short.MAX_VALUE)
135.        );
136.
137.        jPanel7.setBackground(new java.awt.Color(128, 153, 155));
138.
        jPanel7.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
        .swing.border.BevelBorder.RAISED));
139.
140.        btnQuotation.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
141.        btnQuotation.setForeground(new java.awt.Color(255, 255,
        255));
142.        btnQuotation.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
        price-off.png"))); // NOI18N
143.        btnQuotation.setText(" COTIZACION");
144.        btnQuotation.setContentAreaFilled(false);
145.
        btnQuotation.setHorizontalAlignment(javax.swing.SwingConstants.LEAD
        ING);
146.        btnQuotation.setRolloverIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
        price-on.png"))); // NOI18N
147.        btnQuotation.addActionListener(new
        java.awt.event.ActionListener() {
148.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
149.                btnQuotationActionPerformed(evt);
150.            }
151.        });
152.
153.        javax.swing.GroupLayout jPanel7Layout = new
        javax.swing.GroupLayout(jPanel7);
154.        jPanel7.setLayout(jPanel7Layout);
155.        jPanel7Layout.setHorizontalGroup(
156.
        jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
157.            .addComponent(btnQuotation,
        javax.swing.GroupLayout.DEFAULT_SIZE, 297, Short.MAX_VALUE)
158.        );
159.        jPanel7Layout.setVerticalGroup(
160.
        jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)

```



```

161.         .addComponent(btnQuotation,
    javax.swing.GroupLayout.DEFAULT_SIZE, 152, Short.MAX_VALUE)
162.     );
163.
164.     jPanel6.setBackground(new java.awt.Color(128, 153, 155));
165.
    jPanel6.setBorder(javax.swing.BorderFactory.createBevelBorder(javax
    .swing.border.BevelBorder.RAISED));
166.
167.     btnPurchase.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
168.     btnPurchase.setForeground(new java.awt.Color(255, 255,
    255));
169.     btnPurchase.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
    sell-off.png"))); // NOI18N
170.     btnPurchase.setText("COMPRA");
171.     btnPurchase.setContentAreaFilled(false);
172.
    btnPurchase.setHorizontalAlignment(javax.swing.SwingConstants.LEADI
    NG);
173.     btnPurchase.setRolloverIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
    sell-on.png"))); // NOI18N
174.     btnPurchase.addActionListener(new
    java.awt.event.ActionListener() {
175.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
176.             btnPurchaseActionPerformed(evt);
177.         }
178.     });
179.
180.     javax.swing.GroupLayout jPanel6Layout = new
    javax.swing.GroupLayout(jPanel6);
181.     jPanel6.setLayout(jPanel6Layout);
182.     jPanel6Layout.setHorizontalGroup(
183.         jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
184.         .addComponent(btnPurchase,
    javax.swing.GroupLayout.DEFAULT_SIZE, 297, Short.MAX_VALUE)
185.     );
186.     jPanel6Layout.setVerticalGroup(
187.         jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)
188.         .addComponent(btnPurchase,
    javax.swing.GroupLayout.DEFAULT_SIZE, 152, Short.MAX_VALUE)
189.     );
190.
191.     javax.swing.GroupLayout jPanel2Layout = new
    javax.swing.GroupLayout(jPanel2);
192.     jPanel2.setLayout(jPanel2Layout);
193.     jPanel2Layout.setHorizontalGroup(
194.         jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
    .LEADING)

```

```

195.         .addGroup(jPanel2Layout.createSequentialGroup())
196.         .addGap(24, 24, 24)
197.
198.         .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
199.             .Alignment.LEADING)
200.         .addGroup(jPanel2Layout.createSequentialGroup())
201.             .addComponent(jPanel6,
202.                 javax.swing.GroupLayout.PREFERRED_SIZE,
203.                 javax.swing.GroupLayout.DEFAULT_SIZE,
204.                 javax.swing.GroupLayout.PREFERRED_SIZE)
205.             .addGap(0, 0, Short.MAX_VALUE))
206.         .addGroup(jPanel2Layout.createSequentialGroup())
207.         .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
208.             .Alignment.TRAILING)
209.             .addComponent(jPanel1,
210.                 javax.swing.GroupLayout.PREFERRED_SIZE,
211.                 javax.swing.GroupLayout.DEFAULT_SIZE,
212.                 javax.swing.GroupLayout.PREFERRED_SIZE)
213.             .addComponent(jPanel3,
214.                 javax.swing.GroupLayout.PREFERRED_SIZE,
215.                 javax.swing.GroupLayout.DEFAULT_SIZE,
216.                 javax.swing.GroupLayout.PREFERRED_SIZE))
217.         .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
218.             .Alignment.LEADING)
219.         .addGroup(jPanel2Layout.createSequentialGroup())
220.             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
221.                 , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
222.             .addComponent(leave)
223.             .addContainerGap())
224.         .addGroup(jPanel2Layout.createSequentialGroup())
225.             .addGap(245, 245, 245)
226.             .addComponent(jLabel7)
227.             .addContainerGap(584,
228.                 Short.MAX_VALUE))))
229.         .addGroup(jPanel2Layout.createSequentialGroup())
230.             .addComponent(jPanel7,
231.                 javax.swing.GroupLayout.PREFERRED_SIZE,
232.                 javax.swing.GroupLayout.DEFAULT_SIZE,
233.                 javax.swing.GroupLayout.PREFERRED_SIZE)
234.             .addContainerGap()))
235.         .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
236.             .Alignment.LEADING)
237.         .addGroup(jPanel2Layout.createSequentialGroup())
238.             .addGap(529, 529, 529)
239.         .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
240.             .Alignment.LEADING)
241.             .addComponent(jLabel9)

```

```

222.        .addGroup(jPanel2Layout.createSequentialGroup())
223.                .addGap(6, 6, 6)
224.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
225.                .Alignment.LEADING)
226.                .addComponent(jLabel11)
227.                .addComponent(jLabel10)))
228.        .addContainerGap(529, Short.MAX_VALUE))
229.        );
230.        jPanel2Layout.setVerticalGroup(
231.        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
232.                .LEADING)
233.                .addGroup(jPanel2Layout.createSequentialGroup())
234.                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
235.                .Alignment.LEADING)
236.                .addGroup(jPanel2Layout.createSequentialGroup())
237.                .addGap(20, 20, 20)
238.                .addComponent(jPanel1,
239.                javax.swing.GroupLayout.PREFERRED_SIZE,
240.                javax.swing.GroupLayout.DEFAULT_SIZE,
241.                javax.swing.GroupLayout.PREFERRED_SIZE)
242.                .addGap(18, 18, 18)
243.                .addComponent(jPanel3,
244.                javax.swing.GroupLayout.PREFERRED_SIZE,
245.                javax.swing.GroupLayout.DEFAULT_SIZE,
246.                javax.swing.GroupLayout.PREFERRED_SIZE)
247.                .addComponent(leave,
248.                javax.swing.GroupLayout.PREFERRED_SIZE, 66,
249.                javax.swing.GroupLayout.PREFERRED_SIZE)
250.                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
251.                jPanel2Layout.createSequentialGroup())
252.                .addContainerGap()
253.                .addComponent(jLabel7)))
254.                .addGap(22, 22, 22)
255.                .addComponent(jPanel7,
256.                javax.swing.GroupLayout.PREFERRED_SIZE,
257.                javax.swing.GroupLayout.DEFAULT_SIZE,
258.                javax.swing.GroupLayout.PREFERRED_SIZE)
259.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
260.                ED)
261.                .addComponent(jPanel6,
262.                javax.swing.GroupLayout.PREFERRED_SIZE,
263.                javax.swing.GroupLayout.DEFAULT_SIZE,
264.                javax.swing.GroupLayout.PREFERRED_SIZE)
265.                .addContainerGap(24, Short.MAX_VALUE))
266.        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
267.                .Alignment.LEADING)
268.                .addGroup(jPanel2Layout.createSequentialGroup())
269.                .addGap(365, 365, 365)
270.                .addComponent(jLabel9)

```

```

251.         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
252.                 .addComponent(jLabel10)
253.         .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
254.                 .addComponent(jLabel11)
255.                 .addContainerGap(226, Short.MAX_VALUE)))
256.         );
257.
258.         javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
259.         getContentPane().setLayout(layout);
260.         layout.setHorizontalGroup(
261.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
262.                 .addGroup(layout.createSequentialGroup()
263.                 .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
264.                 .addGap(0, 0, Short.MAX_VALUE))
265.                 );
266.         layout.setVerticalGroup(
267.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
268.                 .addGroup(layout.createSequentialGroup()
269.                 .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
270.                 .addGap(0, 0, Short.MAX_VALUE))
271.                 );
272.
273.         pack();
274.     } // </editor-fold>
275.
276.     private void leaveMouseClicked(java.awt.event.MouseEvent evt)
    {
277.         this.setVisible(false);
278.         Menu menu = new Menu();
279.         menu.setVisible(true);
280.         dispose();
281.     }
282.
283.     private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
284.         this.setVisible(false);
285.         ModifyUser modify = new ModifyUser();
286.         modify.setVisible(true);
287.         dispose();
288.     }
289.

```

```

290.     private void
      btnVehicleActionPerformed(java.awt.event.ActionEvent evt) {
291.         this.setVisible(false);
292.         Vehicle vehicle = new Vehicle();
293.         vehicle.setVisible(true);
294.         dispose();
295.     }
296.
297.     private void
      btnQuotationActionPerformed(java.awt.event.ActionEvent evt) {
298.         this.setVisible(false);
299.         Quotation quotation = new Quotation();
300.         quotation.setVisible(true);
301.         dispose();
302.     }
303.
304.     private void
      btnPurchaseActionPerformed(java.awt.event.ActionEvent evt) {
305.         MenuUser.ReloadData = false;
306.         this.setVisible(false);
307.         Purchase purchase = new Purchase();
308.         purchase.setVisible(true);
309.         dispose();
310.     }
311.
312.     /**
313.      * @param args the command line arguments
314.      */
315.     public static void main(String args[]) {
316.         /* Set the Nimbus look and feel */
317.         //<editor-fold defaultstate="collapsed" desc=" Look and
          feel setting code (optional) ">
318.         /* If Nimbus (introduced in Java SE 6) is not available,
          stay with the default look and feel.
319.          * For details see
          http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
320.          */
321.         try {
322.             for (javax.swing.UIManager.LookAndFeelInfo info :
          javax.swing.UIManager.getInstalledLookAndFeels()) {
323.                 if ("Nimbus".equals(info.getName())) {
324.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
325.                     break;
326.                 }
327.             }
328.         } catch (ClassNotFoundException ex) {
329.             java.util.logging.Logger.getLogger(MenuUser.class.getName()).log(ja
          va.util.logging.Level.SEVERE, null, ex);
330.         } catch (InstantiationException ex) {
331.             java.util.logging.Logger.getLogger(MenuUser.class.getName()).log(ja
          va.util.logging.Level.SEVERE, null, ex);
332.         } catch (IllegalAccessException ex) {

```

```

333.     java.util.logging.Logger.getLogger(MenuUser.class.getName()).log(ja
        va.util.logging.Level.SEVERE, null, ex);
334.     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
335.     java.util.logging.Logger.getLogger(MenuUser.class.getName()).log(ja
        va.util.logging.Level.SEVERE, null, ex);
336.     }
337.     //</editor-fold>
338.     //</editor-fold>
339.
340.     /* Create and display the form */
341.     java.awt.EventQueue.invokeLater(new Runnable() {
342.         public void run() {
343.             new MenuUser().setVisible(true);
344.         }
345.     });
346. }
347.
348. // Variables declaration - do not modify
349. private javax.swing.JButton btnPurchase;
350. private javax.swing.JButton btnQuotation;
351. private javax.swing.JButton btnVehicle;
352. private javax.swing.JButton jButton1;
353. private javax.swing.JLabel jLabel10;
354. private javax.swing.JLabel jLabel11;
355. private javax.swing.JLabel jLabel7;
356. private javax.swing.JLabel jLabel9;
357. private javax.swing.JPanel jPanel1;
358. private javax.swing.JPanel jPanel2;
359. private javax.swing.JPanel jPanel3;
360. private javax.swing.JPanel jPanel6;
361. private javax.swing.JPanel jPanel7;
362. private javax.swing.JLabel leave;
363. // End of variables declaration
364. }

```

Purchase

```

1. /*
2.  * To change this license header, choose License Headers in Project
  Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6. package view.user;
7.
8. import code.Files;
9. import code.MethodsSQL;
10. import java.awt.Component;
11. import java.io.File;
12. import java.text.DecimalFormat;
13. import java.util.Properties;
14. import java.util.regex.Matcher;
15. import java.util.regex.Pattern;

```

```
16. import javax.activation.DataHandler;
17. import javax.activation.FileDataSource;
18. import javax.mail.BodyPart;
19. import javax.mail.Message;
20. import javax.mail.Session;
21. import javax.mail.Transport;
22. import javax.mail.internet.InternetAddress;
23. import javax.mail.internet.MimeBodyPart;
24. import javax.mail.internet.MimeMessage;
25. import javax.mail.internet.MimeMultipart;
26. import javax.swing.DefaultComboBoxModel;
27. import javax.swing.ImageIcon;
28. import javax.swing.JFileChooser;
29. import javax.swing.JOptionPane;
30. import view.Login;
31.
32. /**
33.  *
34.  * @author Restf
35.  */
36. public class Purchase extends javax.swing.JFrame {
37.
38.     MethodsSQL methodsSql = new MethodsSQL();
39.     Files file = new Files();
40.     Boolean checkCar1, checkCar2, checkCar3, checkCar4, checkPay1,
    checkPay2;
41.     int price;
42.     ImageIcon car = new
    ImageIcon(getClass().getResource("/resource/car.png"));
43.     ImageIcon nissanWhite = new
    ImageIcon(getClass().getResource("/resource/car-nissan-
    white.png"));
44.     ImageIcon nissanRed = new
    ImageIcon(getClass().getResource("/resource/car-nissan-red.png"));
45.     ImageIcon nissanOrange = new
    ImageIcon(getClass().getResource("/resource/car-nissan-
    orange.png"));
46.     ImageIcon nissanYellow = new
    ImageIcon(getClass().getResource("/resource/car-nissan-
    yellow.png"));
47.     ImageIcon nissanGreen = new
    ImageIcon(getClass().getResource("/resource/car-nissan-
    green.png"));
48.     ImageIcon nissanBlue = new
    ImageIcon(getClass().getResource("/resource/car-nissan-blue.png"));
49.     ImageIcon nissanGray = new
    ImageIcon(getClass().getResource("/resource/car-nissan-gray.png"));
50.
51.     ImageIcon audiWhite = new
    ImageIcon(getClass().getResource("/resource/car-audi-white.png"));
52.     ImageIcon audiRed = new
    ImageIcon(getClass().getResource("/resource/car-audi-red.png"));
53.     ImageIcon audiOrange = new
    ImageIcon(getClass().getResource("/resource/car-audi-orange.png"));
54.     ImageIcon audiYellow = new
    ImageIcon(getClass().getResource("/resource/car-audi-yellow.png"));
```

```

55.     ImageIcon audiGreen  = new
ImageIcon(getClass().getResource("/resource/car-audi-green.png"));
56.     ImageIcon audiBlue   = new
ImageIcon(getClass().getResource("/resource/car-audi-blue.png"));
57.     ImageIcon audiGray   = new
ImageIcon(getClass().getResource("/resource/car-audi-gray.png"));
58.
59.     ImageIcon volkswagenWhite = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
white.png"));
60.     ImageIcon volkswagenRed   = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
red.png"));
61.     ImageIcon volkswagenOrange = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
orange.png"));
62.     ImageIcon volkswagenYellow = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
yellow.png"));
63.     ImageIcon volkswagenGreen = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
green.png"));
64.     ImageIcon volkswagenBlue   = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
blue.png"));
65.     ImageIcon volkswagenGray   = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
gray.png"));
66.
67.     public Purchase() {
68.         initComponents();
69.         setTitle("Car Agency - Compra - [
"+Login.userEntity.getName()+" ]");
70.         setLocationRelativeTo(null);
71.         lockTypePaymentPanel();
72.         if(MenuUser.ReloadData){
73.             load();
74.         } else {
75.             selectColorsNew.setVisible(false);
76.         }
77.     }
78.
79.     /**
80.      * This method is called from within the constructor to
initialize the form.
81.      * WARNING: Do NOT modify this code. The content of this
method is always
82.      * regenerated by the Form Editor.
83.      */
84.     @SuppressWarnings("unchecked")
85.     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
86.     private void initComponents() {
87.
88.         jPanel1 = new javax.swing.JPanel();
89.         jLabel7 = new javax.swing.JLabel();
90.         selectCompany = new javax.swing.JComboBox<>();

```



```

91.         jLabel1 = new javax.swing.JLabel();
92.         jLabel2 = new javax.swing.JLabel();
93.         selectState = new javax.swing.JComboBox<>();
94.         selectModel = new javax.swing.JComboBox<>();
95.         jLabel3 = new javax.swing.JLabel();
96.         jLabel10 = new javax.swing.JLabel();
97.         selectSerial = new javax.swing.JComboBox<>();
98.         selectColors = new javax.swing.JComboBox<>();
99.         jLabel4 = new javax.swing.JLabel();
100.        selectColorsNew = new javax.swing.JComboBox<>();
101.        jLabel6 = new javax.swing.JLabel();
102.        txtPriceCar = new javax.swing.JLabel();
103.        txtExtra = new javax.swing.JLabel();
104.        jLabel8 = new javax.swing.JLabel();
105.        jLabel9 = new javax.swing.JLabel();
106.        txtPriceTotal = new javax.swing.JLabel();
107.        iconCar = new javax.swing.JLabel();
108.        jLabel5 = new javax.swing.JLabel();
109.        jLabel11 = new javax.swing.JLabel();
110.        selectTypePayment = new javax.swing.JComboBox<>();
111.        typePaymentPanel = new javax.swing.JPanel();
112.        txtCard = new javax.swing.JTextField();
113.        labelCar = new javax.swing.JLabel();
114.        txtCVV = new javax.swing.JTextField();
115.        labelCVV = new javax.swing.JLabel();
116.        labelMM = new javax.swing.JLabel();
117.        labelAA = new javax.swing.JLabel();
118.        txtMM = new javax.swing.JTextField();
119.        txtAA = new javax.swing.JTextField();
120.        txtPaypalEmail = new javax.swing.JTextField();
121.        labelPaypalEmail = new javax.swing.JLabel();
122.        labelPaypalPass = new javax.swing.JLabel();
123.        txtPaypalPass = new javax.swing.JPasswordField();
124.        btnPurchase = new javax.swing.JButton();
125.
126.        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        ;
127.        setResizable(false);
128.
129.        jPanel1.setBackground(new java.awt.Color(128, 153, 155));
130.        jPanel1.setForeground(new java.awt.Color(51, 51, 51));
131.        jPanel1.setAutoscrolls(true);
132.        jPanel1.setPreferredSize(new java.awt.Dimension(1280,
720));
133.        jPanel1.setLayout(null);
134.
135.        jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
136.        jLabel7.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
); // NOI18N
137.        jLabel7.setText("VOLVER");
138.        jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
{
139.            public void mouseClicked(java.awt.event.MouseEvent
evt) {

```

```

140.         jLabel7MouseClicked(evt);
141.     }
142. });
143. jPanel1.add(jLabel7);
144. jLabel7.setBounds(1086, 6, 174, 72);
145.
146. selectCompany.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
147. selectCompany.setMaximumRowCount(4);
148. selectCompany.setModel(new
    javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", "•
    NISSAN", "• AUDI", "• VOLKSWAGEN" }));
149. selectCompany.setPreferredSize(new java.awt.Dimension(220,
    40));
150. selectCompany.addActionListener(new
    java.awt.event.ActionListener() {
151.     public void actionPerformed(java.awt.event.ActionEvent
    evt) {
152.         selectCompanyActionPerformed(evt);
153.     }
154. });
155. jPanel1.add(selectCompany);
156. selectCompany.setBounds(207, 127, 320, 40);
157.
158. jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
159. jLabel1.setForeground(new java.awt.Color(255, 255, 255));
160. jLabel1.setText(">> COMPAÑIA");
161. jLabel1.setToolTipText("");
162. jPanel1.add(jLabel1);
163. jLabel1.setBounds(76, 135, 113, 24);
164.
165. jLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
    NOI18N
166. jLabel2.setForeground(new java.awt.Color(255, 255, 255));
167. jLabel2.setText(">> ESTADO");
168. jLabel2.setToolTipText("");
169. jPanel1.add(jLabel2);
170. jLabel2.setBounds(80, 183, 90, 24);
171.
172. selectState.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
173. selectState.setMaximumRowCount(4);
174. selectState.setModel(new
    javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", "•
    NUEVO", "• SEMINUEVO", "• USADO", " " }));
175. selectState.setEnabled(false);
176. selectState.setPreferredSize(new java.awt.Dimension(220,
    40));
177. selectState.addActionListener(new
    java.awt.event.ActionListener() {
178.     public void actionPerformed(java.awt.event.ActionEvent
    evt) {
179.         selectStateActionPerformed(evt);
180.     }
181. });
182. jPanel1.add(selectState);

```

```

183.         selectState.setBounds(207, 175, 320, 40);
184.
185.         selectModel.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
186.         selectModel.setMaximumRowCount(4);
187.         selectModel.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
        }));
188.         selectModel.setEnabled(false);
189.         selectModel.setPreferredSize(new java.awt.Dimension(220,
        40));
190.         selectModel.addActionListener(new
        java.awt.event.ActionListener() {
191.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
192.                 selectModelActionPerformed(evt);
193.             }
194.         });
195.         jPanel1.add(selectModel);
196.         selectModel.setBounds(207, 221, 320, 40);
197.
198.         jLabel3.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
199.         jLabel3.setForeground(new java.awt.Color(255, 255, 255));
200.         jLabel3.setText(">> MODELO");
201.         jLabel3.setToolTipText("");
202.         jPanel1.add(jLabel3);
203.         jLabel3.setBounds(76, 229, 94, 24);
204.
205.         jLabel10.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
206.         jLabel10.setForeground(new java.awt.Color(255, 255, 255));
207.         jLabel10.setText(">> N! SERIE");
208.         jLabel10.setToolTipText("");
209.         jPanel1.add(jLabel10);
210.         jLabel10.setBounds(78, 275, 92, 24);
211.
212.         selectSerial.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
213.         selectSerial.setMaximumRowCount(4);
214.         selectSerial.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
        }));
215.         selectSerial.setEnabled(false);
216.         selectSerial.setPreferredSize(new java.awt.Dimension(220,
        40));
217.         selectSerial.addActionListener(new
        java.awt.event.ActionListener() {
218.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
219.                 selectSerialActionPerformed(evt);
220.             }
221.         });
222.         jPanel1.add(selectSerial);
223.         selectSerial.setBounds(207, 267, 320, 40);
224.

```

```

225.         selectColors.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
226.         selectColors.setMaximumRowCount(4);
227.         selectColors.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
        }));
228.         selectColors.setEnabled(false);
229.         selectColors.setPreferredSize(new java.awt.Dimension(220,
        40));
230.         selectColors.addActionListener(new
        java.awt.event.ActionListener() {
231.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
232.                 selectColorsActionPerformed(evt);
233.             }
234.         });
235.         jPanel1.add(selectColors);
236.         selectColors.setBounds(207, 313, 320, 40);
237.
238.         jLabel4.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
239.         jLabel4.setForeground(new java.awt.Color(255, 255, 255));
240.         jLabel4.setText(">> COLORES");
241.         jLabel4.setToolTipText("");
242.         jPanel1.add(jLabel4);
243.         jLabel4.setBounds(78, 321, 104, 24);
244.
245.         selectColorsNew.setFont(new java.awt.Font("Dialog", 1,
        18)); // NOI18N
246.         selectColorsNew.setMaximumRowCount(4);
247.         selectColorsNew.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
        BLANCO", ". ROJO", ". NARANJA", ". AMARILLO", ". VERDE", ". AZUL",
        ". GRIS" }));
248.         selectColorsNew.setEnabled(false);
249.         selectColorsNew.setPreferredSize(new
        java.awt.Dimension(220, 40));
250.         selectColorsNew.addActionListener(new
        java.awt.event.ActionListener() {
251.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
252.                 selectColorsNewActionPerformed(evt);
253.             }
254.         });
255.         jPanel1.add(selectColorsNew);
256.         selectColorsNew.setBounds(207, 359, 320, 40);
257.
258.         jLabel6.setFont(new java.awt.Font("Dialog", 1, 24)); //
        NOI18N
259.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
260.         jLabel6.setText("PRECIO:          $");
261.         jPanel1.add(jLabel6);
262.         jLabel6.setBounds(716, 459, 167, 32);
263.
264.         txtPriceCar.setFont(new java.awt.Font("Dialog", 1, 24));
        // NOI18N

```

```

265.         txtPriceCar.setForeground(new java.awt.Color(255, 255,
255));
266.         txtPriceCar.setText("-----");
267.         jPanel1.add(txtPriceCar);
268.         txtPriceCar.setBounds(889, 459, 250, 32);
269.
270.         txtExtra.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
271.         txtExtra.setForeground(new java.awt.Color(255, 255, 255));
272.         txtExtra.setText("-----");
273.         jPanel1.add(txtExtra);
274.         txtExtra.setBounds(889, 499, 250, 32);
275.
276.         jLabel8.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
277.         jLabel8.setForeground(new java.awt.Color(255, 255, 255));
278.         jLabel8.setText("ADICIONES: $");
279.         jPanel1.add(jLabel8);
280.         jLabel8.setBounds(716, 497, 167, 32);
281.
282.         jLabel9.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
283.         jLabel9.setForeground(new java.awt.Color(255, 255, 255));
284.         jLabel9.setText("TOTAL: $");
285.         jPanel1.add(jLabel9);
286.         jLabel9.setBounds(719, 539, 167, 32);
287.
288.         txtPriceTotal.setFont(new java.awt.Font("Dialog", 1, 24));
// NOI18N
289.         txtPriceTotal.setForeground(new java.awt.Color(255, 255,
255));
290.         txtPriceTotal.setText("-----");
291.         jPanel1.add(txtPriceTotal);
292.         txtPriceTotal.setBounds(889, 539, 250, 32);
293.
294.         iconCar.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/car.png")))
; // NOI18N
295.         jPanel1.add(iconCar);
296.         iconCar.setBounds(596, 127, 575, 253);
297.
298.         jLabel5.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
299.         jLabel5.setForeground(new java.awt.Color(255, 255, 255));
300.         jLabel5.setText("► SELECCIONA LOS SIGUIENTES DATOS ◀");
301.         jLabel5.setToolTipText("");
302.         jPanel1.add(jLabel5);
303.         jLabel5.setBounds(108, 75, 391, 24);
304.
305.         jLabel11.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
306.         jLabel11.setForeground(new java.awt.Color(255, 255, 255));
307.         jLabel11.setText("» TIPO DE PAGO");
308.         jLabel11.setToolTipText("");
309.         jPanel1.add(jLabel11);
310.         jLabel11.setBounds(78, 428, 145, 24);
311.

```

```

312.         selectTypePayment.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
313.         selectTypePayment.setMaximumRowCount(4);
314.         selectTypePayment.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
TARJETA", ". PAYPAL" }));
315.         selectTypePayment.setEnabled(false);
316.         selectTypePayment.setPreferredSize(new
java.awt.Dimension(220, 40));
317.         selectTypePayment.addActionListener(new
java.awt.event.ActionListener() {
318.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
319.                 selectTypePaymentActionPerformed(evt);
320.             }
321.         });
322.         jPanel1.add(selectTypePayment);
323.         selectTypePayment.setBounds(235, 420, 284, 40);
324.
325.         typePaymentPanel.setBackground(new java.awt.Color(128,
153, 155));
326.         typePaymentPanel.setBorder(javax.swing.BorderFactory.createTitledBo
rder(""));
327.         typePaymentPanel.setEnabled(false);
328.
329.         txtCard.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
330.         txtCard.setHorizontalAlignment(javax.swing.JTextField.CENTER);
331.         txtCard.setPreferredSize(new java.awt.Dimension(73, 34));
332.         txtCard.addActionListener(new
java.awt.event.ActionListener() {
333.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
334.                 txtCardActionPerformed(evt);
335.             }
336.         });
337.         txtCard.addKeyListener(new java.awt.event.KeyAdapter() {
338.             public void keyReleased(java.awt.event.KeyEvent evt) {
339.                 txtCardKeyReleased(evt);
340.             }
341.             public void keyTyped(java.awt.event.KeyEvent evt) {
342.                 txtCardKeyTyped(evt);
343.             }
344.         });
345.
346.         labelCar.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
347.         labelCar.setForeground(new java.awt.Color(255, 255, 255));
348.         labelCar.setText(">> N° TARJETA");
349.         labelCar.setToolTipText("");
350.
351.         txtCVV.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
352.         txtCVV.setHorizontalAlignment(javax.swing.JTextField.CENTER);

```

```

353.         txtCVV.setPreferredSize(new java.awt.Dimension(73, 34));
354.         txtCVV.addKeyListener(new java.awt.event.KeyAdapter() {
355.             public void keyReleased(java.awt.event.KeyEvent evt) {
356.                 txtCVVKeyReleased(evt);
357.             }
358.             public void keyTyped(java.awt.event.KeyEvent evt) {
359.                 txtCVVKeyTyped(evt);
360.             }
361.         });
362.
363.         labelCVV.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
364.         labelCVV.setForeground(new java.awt.Color(255, 255, 255));
365.         labelCVV.setText(">> CVV");
366.         labelCVV.setToolTipText("");
367.
368.         labelMM.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
369.         labelMM.setForeground(new java.awt.Color(255, 255, 255));
370.         labelMM.setText(">> MES");
371.         labelMM.setToolTipText("");
372.
373.         labelAA.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
374.         labelAA.setForeground(new java.awt.Color(255, 255, 255));
375.         labelAA.setText(">> AÑO");
376.         labelAA.setToolTipText("");
377.
378.         txtMM.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
379.         txtMM.setHorizontalAlignment(javax.swing.JTextField.CENTER);
380.         txtMM.setPreferredSize(new java.awt.Dimension(73, 34));
381.         txtMM.addActionListener(new
java.awt.event.ActionListener() {
382.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
383.                 txtMMActionPerformed(evt);
384.             }
385.         });
386.         txtMM.addKeyListener(new java.awt.event.KeyAdapter() {
387.             public void keyReleased(java.awt.event.KeyEvent evt) {
388.                 txtMMKeyReleased(evt);
389.             }
390.             public void keyTyped(java.awt.event.KeyEvent evt) {
391.                 txtMMKeyTyped(evt);
392.             }
393.         });
394.
395.         txtAA.setFont(new java.awt.Font("Dialog", 0, 18)); //
NOI18N
396.         txtAA.setHorizontalAlignment(javax.swing.JTextField.CENTER);
397.         txtAA.setPreferredSize(new java.awt.Dimension(73, 34));
398.         txtAA.addActionListener(new
java.awt.event.ActionListener() {

```

```

399.         public void actionPerformed(java.awt.event.ActionEvent
    evt) {
400.             txtAAActionPerformed(evt);
401.         }
402.     });
403.     txtAA.addKeyListener(new java.awt.event.KeyAdapter() {
404.         public void keyReleased(java.awt.event.KeyEvent evt) {
405.             txtAAKeyReleased(evt);
406.         }
407.         public void keyTyped(java.awt.event.KeyEvent evt) {
408.             txtAAKeyTyped(evt);
409.         }
410.     });
411.
412.     txtPaypalEmail.setFont(new java.awt.Font("Dialog", 0,
18)); // NOI18N
413.     txtPaypalEmail.setHorizontalAlignment(javax.swing.JTextField.CENTER
);
414.     txtPaypalEmail.setPreferredSize(new java.awt.Dimension(73,
34));
415.     txtPaypalEmail.addKeyListener(new
java.awt.event.KeyAdapter() {
416.         public void keyReleased(java.awt.event.KeyEvent evt) {
417.             txtPaypalEmailKeyReleased(evt);
418.         }
419.         public void keyTyped(java.awt.event.KeyEvent evt) {
420.             txtPaypalEmailKeyTyped(evt);
421.         }
422.     });
423.
424.     labelPaypalEmail.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
425.     labelPaypalEmail.setForeground(new java.awt.Color(255,
255, 255));
426.     labelPaypalEmail.setText(">> CORREO");
427.     labelPaypalEmail.setToolTipText("");
428.
429.     labelPaypalPass.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
430.     labelPaypalPass.setForeground(new java.awt.Color(255, 255,
255));
431.     labelPaypalPass.setText(">> CONTRASEÑA");
432.     labelPaypalPass.setToolTipText("");
433.
434.     txtPaypalPass.setFont(new java.awt.Font("Dialog", 0, 18));
// NOI18N
435.     txtPaypalPass.setHorizontalAlignment(javax.swing.JTextField.CENTER
);
436.     txtPaypalPass.setPreferredSize(new java.awt.Dimension(73,
34));
437.     txtPaypalPass.addKeyListener(new
java.awt.event.KeyAdapter() {
438.         public void keyReleased(java.awt.event.KeyEvent evt) {
439.             txtPaypalPassKeyReleased(evt);
440.         }

```



```

441.         public void keyTyped(java.awt.event.KeyEvent evt) {
442.             txtPaypalPassKeyTyped(evt);
443.         }
444.     });
445.
446.     javax.swing.GroupLayout typePaymentPanelLayout = new
javax.swing.GroupLayout(typePaymentPanel);
447.     typePaymentPanel.setLayout(typePaymentPanelLayout);
448.     typePaymentPanelLayout.setHorizontalGroup(
449.         typePaymentPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
450.         .addGroup(typePaymentPanelLayout.createSequentialGroup()
451.             .addGroup(typePaymentPanelLayout.createParallelGroup(javax.s
wing.Gr
oupLayout.Alignment.TRAILING)
452.             .addGroup(typePaymentPanelLayout.createSequentialGroup()
453.                 .addGap(28, 28, 28)
454.                 .addGroup(typePaymentPanelLayout.createParallelGroup(j
avax.s
wing.Gr
oupLayout.Alignment.LEADING)
455.                 .addGroup(typePaymentPanelLayout.createSequentialGroup()
456.                     .addComponent(labelCVV)
457.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPl
acement.RELATED
458.                     .addComponent(txtCVV,
javax.swing.GroupLayout.PREFERRED_SIZE, 1, Short.MAX_VALUE))
459.                 .addComponent(labelCar,
javax.swing.GroupLayout.Alignment.TRAILING))
460.                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPl
acement.RELATED
461.                 .addGroup(typePaymentPanelLayout.createParallelGroup(j
avax.s
wing.Gr
oupLayout.Alignment.LEADING, false)
462.                 .addComponent(txtCard,
javax.swing.GroupLayout.PREFERRED_SIZE, 258,
javax.swing.GroupLayout.PREFERRED_SIZE)
463.                 .addGroup(typePaymentPanelLayout.createSequentialGroup()
464.                     .addComponent(labelMM)
465.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPl
acement.RELATED
, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
466.                     .addComponent(txtMM,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
467.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPl
acement.RELATED
468.                     .addComponent(labelAA)

```

```

469.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
470.        .addComponent(txtAA,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))))
471.        .addGroup(typePaymentPanelLayout.createSequentialGroup())
472.        .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.TRAILING)
473.        .addGroup(typePaymentPanelLayout.createSequentialGroup())
474.        .addGap(0, 0, Short.MAX_VALUE)
475.        .addComponent(labelPaypalPass))
476.        .addGroup(typePaymentPanelLayout.createSequentialGroup())
477.        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
478.        .addComponent(labelPaypalEmail)))
479.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
480.        .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.LEADING, false)
481.        .addComponent(txtPaypalEmail,
javax.swing.GroupLayout.DEFAULT_SIZE, 258, Short.MAX_VALUE)
482.        .addComponent(txtPaypalPass,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
483.        .addContainerGap(44, Short.MAX_VALUE))
484.    );
485.    typePaymentPanelLayout.setVerticalGroup(
486.    typePaymentPanelLayout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
487.    .addGroup(typePaymentPanelLayout.createSequentialGroup())
488.    .addGap(14, 14, 14)
489.    .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE)
490.    .addComponent(labelCar)
491.    .addComponent(txtCard,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
492.    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
493.    .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
oupLayout.Alignment.BASELINE)

```

```

494.                .addComponent(txtCVV,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
495.                .addComponent(labelCVV)
496.                .addComponent(txtMM,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
497.                .addComponent(txtAA,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
498.                .addComponent(labelMM)
499.                .addComponent(labelAA))
500.            .addGap(27, 27, 27)
501.
    .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
    oupLayout.Alignment.BASELINE)
502.            .addComponent(labelPaypalEmail)
503.            .addComponent(txtPaypalEmail,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
504.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
505.
    .addGroup(typePaymentPanelLayout.createParallelGroup(javax.swing.Gr
    oupLayout.Alignment.BASELINE)
506.            .addComponent(labelPaypalPass)
507.            .addComponent(txtPaypalPass,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE))
508.            .addContainerGap(31, Short.MAX_VALUE))
509.        );
510.
511.        jPanell1.add(typePaymentPanel);
512.        typePaymentPanel.setBounds(67, 472, 476, 228);
513.
514.        btnPurchase.setBackground(new java.awt.Color(255, 255,
    255));
515.        btnPurchase.setFont(new java.awt.Font("Dialog", 1, 18));
    // NOI18N
516.        btnPurchase.setIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/ready-
    off.png"))); // NOI18N
517.        btnPurchase.setText("COMPRAR");
518.        btnPurchase.setEnabled(false);
519.        btnPurchase.setRolloverIcon(new
    javax.swing.ImageIcon(getClass().getResource("/resource/ready-
    on.png"))); // NOI18N
520.        btnPurchase.addActionListener(new
    java.awt.event.ActionListener() {
521.            public void actionPerformed(java.awt.event.ActionEvent
    evt) {

```

```

522.         btnPurchaseActionPerformed(evt);
523.     }
524. });
525. jPanell1.add(btnPurchase);
526. btnPurchase.setBounds(700, 600, 340, 80);
527.
528. javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
529. getContentPane().setLayout(layout);
530. layout.setHorizontalGroup(
531.     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
532.         .addGroup(layout.createSequentialGroup()
533.             .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
534.             .addGap(0, 0, Short.MAX_VALUE))
535.         );
536. layout.setVerticalGroup(
537.     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
538.         .addGroup(layout.createSequentialGroup()
539.             .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
540.             .addGap(0, 0, Short.MAX_VALUE))
541.         );
542.
543. pack();
544. }// </editor-fold>
545.
546. private void jLabel7MouseClicked(java.awt.event.MouseEvent
evt) {
547.     MenuUser.ReloadData = false;
548.     this.setVisible(false);
549.     MenuUser menu = new MenuUser();
550.     menu.setVisible(true);
551.     dispose();
552. }
553.
554. private void
selectCompanyActionPerformed(java.awt.event.ActionEvent evt) {
555.     if(MenuUser.ReloadData){
556.         load();
557.     } else {
558.         if (selectCompany.getSelectedIndex() > 0) {
559.             selectState.setEnabled(true);
560.             selectState.setSelectedIndex(0);
561.             iconCar.setIcon(car);
562.             txtPriceCar.setText("-----");
563.             txtExtra.setText("-----");
564.             txtPriceTotal.setText("-----");
565.             switch(selectCompany.getSelectedIndex()){

```

```

566.             case 1:
567.                 iconCar.setIcon(nissanWhite);
568.                 break;
569.             case 2:
570.                 iconCar.setIcon(audiWhite);
571.                 break;
572.             case 3:
573.                 iconCar.setIcon(volkswagenWhite);
574.                 break;
575.         }
576.     } else {
577.         clear();
578.     }
579. }
580. }
581.
582.     private void
selectStateActionPerformed(java.awt.event.ActionEvent evt) {
583.         if(MenuUser.ReloadData) {
584.             load();
585.         } else {
586.             if (selectState.getSelectedIndex() > 0) {
587.                 selectModel.setEnabled(true);
588.                 selectSerial.setEnabled(false);
589.                 selectSerial.setSelectedIndex(0);
590.                 txtPriceCar.setText("-----");
591.                 txtExtra.setText("-----");
592.                 txtPriceTotal.setText("-----");
593.                 switch(selectCompany.getSelectedIndex()) {
594.                     case 1:
595.                         iconCar.setIcon(nissanWhite);
596.                         break;
597.                     case 2:
598.                         iconCar.setIcon(audiWhite);
599.                         break;
600.                     case 3:
601.                         iconCar.setIcon(volkswagenWhite);
602.                         break;
603.                 }
604.
selectModel.setModel(methodsSql.loadModels(selectCompany.getSelectedIndex(), selectState.getSelectedIndex()));
605.             } else {
606.                 iconCar.setIcon(car);
607.                 selectModel.setEnabled(false);
608.                 selectSerial.setEnabled(false);
609.                 selectColors.setEnabled(false);
610.                 selectModel.setSelectedIndex(0);
611.                 selectSerial.setSelectedIndex(0);
612.                 selectColors.setSelectedIndex(0);
613.             }
614.         }
615.
616.     private void
selectModelActionPerformed(java.awt.event.ActionEvent evt) {
617.         if(MenuUser.ReloadData) {
618.             load();

```

```

619.         } else {
620.             if (selectModel.getSelectedIndex() > 0) {
621.                 selectSerial.setEnabled(true);
622.                 String txtmodel =
selectModel.getSelectedItem().toString();
623.                 String model = txtmodel.substring(2);
624.                 selectSerial.setModel(methodsSql.loadSerial(model));
625.                 txtPriceCar.setText("-----");
626.                 txtExtra.setText("-----");
627.                 txtPriceTotal.setText("-----");
628.             } else {
629.                 selectSerial.setEnabled(false);
630.                 selectColors.setEnabled(false);
631.                 selectSerial.setSelectedIndex(0);
632.                 selectColors.setSelectedIndex(0);
633.             }}
634.     }
635.
636.     private void
selectSerialActionPerformed(java.awt.event.ActionEvent evt) {
637.         if(MenuUser.ReloadData){
638.             load();
639.         } else {
640.             if (selectSerial.getSelectedIndex() > 0) {
641.                 selectColors.setEnabled(true);
642.                 String txt =
selectSerial.getSelectedItem().toString();
643.                 String txtmodel = txt.substring(2);
644.                 int model = Integer.parseInt(txtmodel);
645.                 selectColors.setModel(methodsSql.loadColorsQuotation(model));
646.                 String s =
moneyFormat(methodsSql.loadPriceCar(model));
647.                 txtPriceCar.setText(s);
648.                 if(!selectColorsNew.isEnabled()){
649.                     txtPriceTotal.setText(s);
650.                 }
651.
652.             } else {
653.                 selectColors.setEnabled(false);
654.                 selectColors.setSelectedIndex(0);
655.             }
656.         }
657.     }
658.
659.     private void
selectColorsActionPerformed(java.awt.event.ActionEvent evt) {
660.         if(MenuUser.ReloadData){
661.             load();
662.         } else {
663.             String txtcompany =
selectCompany.getSelectedItem().toString();
664.             String company = txtcompany.substring(2);
665.             String txtcolor =
selectColors.getSelectedItem().toString();
666.             String color = txtcolor.substring(2);
667.             switch(company){

```

```
668.         case "NISSAN":
669.         switch(color){
670.             case "BLANCO":
671.                 iconCar.setIcon(nissanWhite);
672.                 break;
673.             case "ROJO":
674.                 iconCar.setIcon(nissanRed);
675.                 break;
676.             case "NARANJA":
677.                 iconCar.setIcon(nissanOrange);
678.                 break;
679.             case "AMARILLO":
680.                 iconCar.setIcon(nissanYellow);
681.                 break;
682.             case "VERDE":
683.                 iconCar.setIcon(nissanGreen);
684.                 break;
685.             case "AZUL":
686.                 iconCar.setIcon(nissanBlue);
687.                 break;
688.             case "GRIS":
689.                 iconCar.setIcon(nissanGray);
690.                 break;
691.         }
692.         break;
693.         case "AUDI":
694.         switch(color){
695.             case "BLANCO":
696.                 iconCar.setIcon(audiWhite);
697.                 break;
698.             case "ROJO":
699.                 iconCar.setIcon(audiRed);
700.                 break;
701.             case "NARANJA":
702.                 iconCar.setIcon(audiOrange);
703.                 break;
704.             case "AMARILLO":
705.                 iconCar.setIcon(audiYellow);
706.                 break;
707.             case "VERDE":
708.                 iconCar.setIcon(audiGreen);
709.                 break;
710.             case "AZUL":
711.                 iconCar.setIcon(audiBlue);
712.                 break;
713.             case "GRIS":
714.                 iconCar.setIcon(audiGray);
715.                 break;
716.         }
717.         break;
718.         case "VOLKSWAGEN":
719.         switch(color){
720.             case "BLANCO":
721.                 iconCar.setIcon(volkswagenWhite);
722.                 break;
723.             case "ROJO":
724.                 iconCar.setIcon(volkswagenRed);
```

```

725.             break;
726.             case "NARANJA":
727.                 iconCar.setIcon(volkswagenOrange);
728.             break;
729.             case "AMARILLO":
730.                 iconCar.setIcon(volkswagenYellow);
731.             break;
732.             case "VERDE":
733.                 iconCar.setIcon(volkswagenGreen);
734.             break;
735.             case "AZUL":
736.                 iconCar.setIcon(volkswagenBlue);
737.             break;
738.             case "GRIS":
739.                 iconCar.setIcon(volkswagenGray);
740.             break;
741.         }
742.         break;
743.     }
744.
745.     if(!selectColors.getSelectedItemAt().equals("Seleccionar") &&
        !selectColors.getSelectedItemAt().equals("• Otro:
        ")){
746.         selectTypePayment.setEnabled(true);
747.     } else {
748.         selectTypePayment.setSelectedIndex(0);
749.         selectTypePayment.setEnabled(false);
750.     }
751.     if(selectColors.getSelectedItemAt().equals("• Otro: ")){
752.         selectColorsNew.setVisible(true);
753.         selectColorsNew.setEnabled(true);
754.         selectColorsNew.setSelectedIndex(0);
755.         switch(selectCompany.getSelectedIndex()){
756.             case 1:
757.                 iconCar.setIcon(nissanWhite);
758.             break;
759.             case 2:
760.                 iconCar.setIcon(audiWhite);
761.             break;
762.             case 3:
763.                 iconCar.setIcon(volkswagenWhite);
764.             break;
765.         }
766.     } else {
767.         selectColorsNew.setSelectedIndex(0);
768.         selectColorsNew.setVisible(false);
769.     }
770.
771. }
772. }
773.
774. private void
    selectTypePaymentActionPerformed(java.awt.event.ActionEvent evt) {
775.     switch(selectTypePayment.getSelectedIndex()){
776.         case 1:
777.             txtCard.setEnabled(true);
778.             txtCVV.setEnabled(true);

```



```

779.         txtMM.setEnabled(true);
780.         txtAA.setEnabled(true);
781.         labelCar.setEnabled(true);
782.         labelCVV.setEnabled(true);
783.         labelMM.setEnabled(true);
784.         labelAA.setEnabled(true);
785.
786.         txtPaypalEmail.setEnabled(false);
787.         txtPaypalPass.setEnabled(false);
788.         labelPaypalEmail.setEnabled(false);
789.         labelPaypalPass.setEnabled(false);
790.         txtPaypalEmail.setText(null);
791.         txtPaypalPass.setText(null);
792.
793.         break;
794.     case 2:
795.         txtPaypalEmail.setEnabled(true);
796.         txtPaypalPass.setEnabled(true);
797.         labelPaypalEmail.setEnabled(true);
798.         labelPaypalPass.setEnabled(true);
799.
800.         txtCard.setEnabled(false);
801.         txtCVV.setEnabled(false);
802.         txtMM.setEnabled(false);
803.         txtAA.setEnabled(false);
804.         labelCar.setEnabled(false);
805.         labelCVV.setEnabled(false);
806.         labelMM.setEnabled(false);
807.         labelAA.setEnabled(false);
808.         txtCard.setText(null);
809.         txtCVV.setText(null);
810.         txtMM.setText(null);
811.         txtAA.setText(null);
812.         break;
813.     default:
814.         lockTypePaymentPanel();
815.         break;
816.     }
817.     btnPurchase.setEnabled(false);
818. }
819.
820. private void txtMMActionPerformed(java.awt.event.ActionEvent
    evt) {
821.     // TODO add your handling code here:
822. }
823.
824. private void txtAAAActionPerformed(java.awt.event.ActionEvent
    evt) {
825.     // TODO add your handling code here:
826. }
827.
828. private void
    btnPurchaseActionPerformed(java.awt.event.ActionEvent evt) {
829.     int i = JOptionPane.showConfirmDialog(
830.         null, "CONFIRME PORFAVOR SU COMPRA PARA PROCESAR
        EL PAGO!",

```

```

831.             ";ALERTA!", JOptionPane.YES_NO_OPTION,
JOptionPane.PLAIN_MESSAGE);
832.         if(i == 0){
833.             int id = Login.userEntity.getId();
834.             int folio = (int) (Math.random()*1500+1000);
835.             String name = Login.userEntity.getName();
836.             String model, additional = null;
837.             int serial_number;
838.             if(MenuUser.ReloadData){
839.                 model = selectModel.getSelectedItem().toString();
840.                 serial_number =
Integer.parseInt(selectSerial.getSelectedItem().toString());
841.             } else {
842.                 String txtmodel =
selectModel.getSelectedItem().toString();
843.                 String txttmodel = txtmodel.substring(2);
844.                 model = txttmodel;
845.                 String txtserial =
selectSerial.getSelectedItem().toString();
846.                 String txttserial = txtserial.substring(2);
847.                 serial_number = Integer.parseInt(txttserial);
848.             }
849.             int price_car =
methodsSql.loadPriceCar(serial_number);
850.             if(selectColorsNew.isEnabled()){
851.                 additional = "YES";
852.             } else {
853.                 additional = "NOT";
854.             }
855.             int price_additional = price;
856.             int price_total = price_car + price_additional;
857.             int type_payment =
selectTypePayment.getSelectedIndex();
858.             methodsSql.Purchase(id, folio, name, model,
serial_number, price_car, additional, price_additional,
price_total, type_payment);
859. //             methodsSql.DeleteCarPurchase(serial_number);
860.             validateRecive();
861.         }
862.     }
863.
864.     private void
selectColorsNewActionPerformed(java.awt.event.ActionEvent evt) {
865.         if(MenuUser.ReloadData){
866.             load();
867.         } else {
868.             if(selectColorsNew.isEnabled()){
869.                 String txtcompany =
selectCompany.getSelectedItem().toString();
870.                 String company = txtcompany.substring(2);
871.                 String txtcolor =
selectColorsNew.getSelectedItem().toString();
872.                 String color = txtcolor.substring(2);
873.                 switch(company){
874.                     case "NISSAN":
875.                         switch(color){
876.                             case "BLANCO":

```

```
877.         price = 15000;
878.         iconCar.setIcon(nissanWhite);
879.         break;
880.         case "ROJO":
881.         price = 10000;
882.         iconCar.setIcon(nissanRed);
883.         break;
884.         case "NARANJA":
885.         price = 5000;
886.         iconCar.setIcon(nissanOrange);
887.         break;
888.         case "AMARILLO":
889.         price = 8000;
890.         iconCar.setIcon(nissanYellow);
891.         break;
892.         case "VERDE":
893.         price = 9000;
894.         iconCar.setIcon(nissanGreen);
895.         break;
896.         case "AZUL":
897.         price = 12000;
898.         iconCar.setIcon(nissanBlue);
899.         break;
900.         case "GRIS":
901.         price = 13000;
902.         iconCar.setIcon(nissanGray);
903.         break;
904.     }
905.     break;
906.     case "AUDI":
907.     switch(color) {
908.         case "BLANCO":
909.         price = 15000;
910.         iconCar.setIcon(audiWhite);
911.         break;
912.         case "ROJO":
913.         price = 10000;
914.         iconCar.setIcon(audiRed);
915.         break;
916.         case "NARANJA":
917.         price = 5000;
918.         iconCar.setIcon(audiOrange);
919.         break;
920.         case "AMARILLO":
921.         price = 8000;
922.         iconCar.setIcon(audiYellow);
923.         break;
924.         case "VERDE":
925.         price = 9000;
926.         iconCar.setIcon(audiGreen);
927.         break;
928.         case "AZUL":
929.         price = 12000;
930.         iconCar.setIcon(audiBlue);
931.         break;
932.         case "GRIS":
933.         price = 13000;
```

```

934.         iconCar.setIcon(audiGray);
935.         break;
936.     }
937.     break;
938.     case "VOLKSWAGEN":
939.     switch(color){
940.         case "BLANCO":
941.             price = 15000;
942.             iconCar.setIcon(volkswagenWhite);
943.             break;
944.         case "ROJO":
945.             price = 10000;
946.             iconCar.setIcon(volkswagenRed);
947.             break;
948.         case "NARANJA":
949.             price = 5000;
950.             iconCar.setIcon(volkswagenOrange);
951.             break;
952.         case "AMARILLO":
953.             price = 8000;
954.             iconCar.setIcon(volkswagenYellow);
955.             break;
956.         case "VERDE":
957.             price = 9000;
958.             iconCar.setIcon(volkswagenGreen);
959.             break;
960.         case "AZUL":
961.             price = 12000;
962.             iconCar.setIcon(volkswagenBlue);
963.             break;
964.         case "GRIS":
965.             price = 13000;
966.             iconCar.setIcon(volkswagenGray);
967.             break;
968.     }
969.     break;
970.     }
971.     String s = moneyFormat(price);
972.     txtExtra.setText(s);
973.     String txt =
        selectSerial.getSelectedItemAt().toString();
974.     String txtmodel = txt.substring(2);
975.     int model = Integer.parseInt(txtmodel);
976.     int extra = methodsSql.loadPriceCar(model);
977.     int price_total = price + extra;
978.
979.     txtPriceTotal.setText(moneyFormat(price_total));
980.     if(!selectColorsNew.getSelectedItemAt().equals("Seleccionar")){
981.         selectTypePayment.setEnabled(true);
982.     } else {
983.         selectTypePayment.setEnabled(false);
984.         selectTypePayment.setSelectedIndex(0);
985.     }
986. }
987. }

```

```

988.
989.     private void txtCardKeyTyped(java.awt.event.KeyEvent evt) {
990.         char num = evt.getKeyChar();
991.         if(num<48 || num>57) evt.consume();
992.         if(txtCard.getText().length()>=16) evt.consume();
993.     }
994.
995.     private void txtCVVKeyTyped(java.awt.event.KeyEvent evt) {
996.         char num = evt.getKeyChar();
997.         if(num<48 || num>57) evt.consume();
998.         if(txtCVV.getText().length()>=3) evt.consume();
999.     }
1000.
1001.         private void txtMMKeyTyped(java.awt.event.KeyEvent
    evt) {
1002.             char num = evt.getKeyChar();
1003.             if(num<48 || num>57) evt.consume();
1004.             if(txtMM.getText().length()>=2) evt.consume();
1005.         }
1006.
1007.         private void txtAAKeyTyped(java.awt.event.KeyEvent
    evt) {
1008.             char num = evt.getKeyChar();
1009.             if(num<48 || num>57) evt.consume();
1010.             if(txtAA.getText().length()>=2) evt.consume();
1011.         }
1012.
1013.         private void
    txtPaypalPassKeyTyped(java.awt.event.KeyEvent evt) {
1014.             char c = evt.getKeyChar();
1015.             if((c<48 || c>57) && (c<65 || c>90) && (c<97 ||
    c>122)) {
1016.                 evt.consume(); }
1017.             if(txtPaypalPass.getText().length()>=30)
    evt.consume();
1018.         }
1019.
1020.         private void
    txtPaypalEmailKeyTyped(java.awt.event.KeyEvent evt) {
1021.             int c = evt.getKeyChar();
1022.             if((c<48 || c>57) && (c<64 || c>90) &&
    (c<97 || c>122) && (c!=45) && (c!=46) &&
    (c!=95)){ evt.consume(); }
1023.             if(txtPaypalEmail.getText().length()>=30)
    evt.consume();
1024.         }
1025.
1026.
1027.
1028.         private void
    txtCardKeyReleased(java.awt.event.KeyEvent evt) {
1029.             if(txtCard.getText().length() == 16){ checkCar1 =
    true;
1030.             } else { checkCar1 = false; }
1031.             btnValidateCard();
1032.         }
1033.
1034.         private void txtCVVKeyReleased(java.awt.event.KeyEvent
    evt) {

```

```

1035.             if(txtCVV.getText().length() == 3){ checkCar2 =
true;
1036.             } else { checkCar2 = false; }
1037.             btnValidateCard();
1038.         }
1039.
1040.         private void txtMMKeyReleased(java.awt.event.KeyEvent
evt) {
1041.             if(txtMM.getText().length() == 2){ checkCar3 =
true;
1042.             } else { checkCar3 = false; }
1043.             btnValidateCard();
1044.         }
1045.
1046.         private void txtAAKeyReleased(java.awt.event.KeyEvent
evt) {
1047.             if(txtAA.getText().length() == 2){ checkCar4 =
true;
1048.             } else { checkCar4 = false; }
1049.             btnValidateCard();
1050.         }
1051.
1052.         private void
txtPaypalEmailKeyReleased(java.awt.event.KeyEvent evt) {
1053.             checkPay1 = isEmail(txtPaypalEmail.getText());
1054.             btnValidatePaypal();
1055.         }
1056.
1057.         private void
txtPaypalPassKeyReleased(java.awt.event.KeyEvent evt) {
1058.             if(txtPaypalPass.getText().length() > 8){
checkPay2 = true;
1059.             } else { checkPay2 = false; }
1060.             btnValidatePaypal();
1061.         }
1062.
1063.         private void
txtCardActionPerformed(java.awt.event.ActionEvent evt) {
1064.             // TODO add your handling code here:
1065.         }
1066.
1067.         private String moneyFormat(int i){
1068.             DecimalFormat money = new
DecimalFormat("###,###,###");
1069.             String s = money.format(i);
1070.             return s;
1071.         }
1072.
1073.         private void clear(){
1074.             selectState.setEnabled(false);
1075.             selectModel.setEnabled(false);
1076.             selectColors.setEnabled(false);
1077.             selectColorsNew.setEnabled(false);
1078.             selectSerial.setEnabled(false);
1079.             selectTypePayment.setEnabled(false);
1080.             selectState.setSelectedIndex(0);
1081.             selectModel.setSelectedIndex(0);

```

```

1082.         selectSerial.setSelectedIndex(0);
1083.         selectColors.setSelectedIndex(0);
1084.         selectColorsNew.setSelectedIndex(0);
1085.         selectTypePayment.setSelectedIndex(0);
1086.         iconCar.setIcon(car);
1087.     }
1088.
1089.     private void load(){
1090.         selectCompany.setEnabled(true);
1091.         String txtCompany =
1092.             Quotation.selectCompany.getSelectedItem().toString();
1093.         String txttCompany = txtCompany.substring(2);
1094.         DefaultComboBoxModel company = new
1095.             DefaultComboBoxModel();
1096.         company.addElement(txttCompany);
1097.         selectCompany.setModel(company);
1098.
1099.         switch(selectCompany.getSelectedItem().toString()){
1100.             case "NISSAN":
1101.                 iconCar.setIcon(nissanWhite);
1102.                 break;
1103.             case "AUDI":
1104.                 iconCar.setIcon(audiWhite);
1105.                 break;
1106.             case "VOLKSWAGEN":
1107.                 iconCar.setIcon(volkswagenWhite);
1108.                 break;
1109.         }
1110.         selectState.setEnabled(true);
1111.         String txtState =
1112.             Quotation.selectState.getSelectedItem().toString();
1113.         String txttState = txtState.substring(2);
1114.         DefaultComboBoxModel state = new
1115.             DefaultComboBoxModel();
1116.         state.addElement(txttState);
1117.         selectState.setModel(state);
1118.
1119.         selectModel.setEnabled(true);
1120.         String txtModel =
1121.             Quotation.selectModel.getSelectedItem().toString();
1122.         String txttModel = txtModel.substring(2);
1123.         DefaultComboBoxModel model = new
1124.             DefaultComboBoxModel();
1125.         model.addElement(txttModel);
1126.         selectModel.setModel(model);
1127.
1128.         selectSerial.setEnabled(true);
1129.         String txtSerial =
1130.             Quotation.selectSerial.getSelectedItem().toString();
1131.         String txttSerial = txtSerial.substring(2);
1132.         DefaultComboBoxModel serial = new
1133.             DefaultComboBoxModel();
1134.         serial.addElement(txttSerial);
1135.         selectSerial.setModel(serial);
1136.
1137.         if(Quotation.selectColors.getSelectedItem().equals("• Otro: ")){

```

```

1129.             selectColors.setEnabled(true);
1130.             String txtColors =
1131.                 Quotation.selectColorsNew.getSelectedItem().toString();
1132.             String txtttColors = txtColors.substring(2);
1133.             DefaultComboBoxModel colors = new
1134.                 DefaultComboBoxModel();
1135.             colors.addElement(txtttColors);
1136.             selectColors.setModel(colors);
1137.         } else {
1138.             selectColors.setEnabled(true);
1139.             String txtColors =
1140.                 Quotation.selectColors.getSelectedItem().toString();
1141.             String txtttColors = txtColors.substring(2);
1142.             DefaultComboBoxModel colors = new
1143.                 DefaultComboBoxModel();
1144.             colors.addElement(txtttColors);
1145.             selectColors.setModel(colors);
1146.         }
1147.     }
1148.     switch(selectCompany.getSelectedItem().toString()){
1149.         case "NISSAN":
1150.             switch(selectColors.getSelectedItem().toString()){
1151.                 case "BLANCO":
1152.                     iconCar.setIcon(nissanWhite);
1153.                     break;
1154.                 case "ROJO":
1155.                     iconCar.setIcon(nissanRed);
1156.                     break;
1157.                 case "NARANJA":
1158.                     iconCar.setIcon(nissanOrange);
1159.                     break;
1160.                 case "AMARILLO":
1161.                     iconCar.setIcon(nissanYellow);
1162.                     break;
1163.                 case "VERDE":
1164.                     iconCar.setIcon(nissanGreen);
1165.                     break;
1166.                 case "AZUL":
1167.                     iconCar.setIcon(nissanBlue);
1168.                     break;
1169.                 case "GRIS":
1170.                     iconCar.setIcon(nissanGray);
1171.                     break;
1172.             }
1173.             break;
1174.         case "AUDI":
1175.             switch(selectColors.getSelectedItem().toString()){
1176.                 case "BLANCO":
1177.                     iconCar.setIcon(audiWhite);
1178.                     break;
1179.                 case "ROJO":
1180.                     iconCar.setIcon(audiRed);
1181.                     break;
1182.                 case "NARANJA":
1183.                     iconCar.setIcon(audiOrange);

```



```

1179.             break;
1180.             case "AMARILLO":
1181.                 iconCar.setIcon(audiYellow);
1182.             break;
1183.             case "VERDE":
1184.                 iconCar.setIcon(audiGreen);
1185.             break;
1186.             case "AZUL":
1187.                 iconCar.setIcon(audiBlue);
1188.             break;
1189.             case "GRIS":
1190.                 iconCar.setIcon(audiGray);
1191.             break;
1192.         }
1193.         break;
1194.         case "VOLKSWAGEN":
1195.             switch(selectColors.getSelectedItem().toString()){
1196.                 case "BLANCO":
1197.                     iconCar.setIcon(volkswagenWhite);
1198.                 break;
1199.                 case "ROJO":
1200.                     iconCar.setIcon(volkswagenRed);
1201.                 break;
1202.                 case "NARANJA":
1203.                     iconCar.setIcon(volkswagenOrange);
1204.                 break;
1205.                 case "AMARILLO":
1206.                     iconCar.setIcon(volkswagenYellow);
1207.                 break;
1208.                 case "VERDE":
1209.                     iconCar.setIcon(volkswagenGreen);
1210.                 break;
1211.                 case "AZUL":
1212.                     iconCar.setIcon(volkswagenBlue);
1213.                 break;
1214.                 case "GRIS":
1215.                     iconCar.setIcon(volkswagenGray);
1216.                 break;
1217.             }
1218.             break;
1219.         }
1220.         selectColorsNew.setVisible(false);
1221.         selectTypePayment.setEnabled(true);
1222.         txtPriceCar.setText(Quotation.txtPriceCar.getText());
1223.         txtExtra.setText(Quotation.txtExtra.getText());
1224.         txtPriceTotal.setText(Quotation.txtPriceTotal.getText());
1225.     }
1226.
1227.     public void lockTypePaymentPanel(){
1228.         for(Component a:
1229.             typePaymentPanel.getComponents()){
1230.             a.setEnabled(false);
1231.         }

```

```

1232.
1233.         public void btnValidateCard() {
1234.             if(txtCard.getText().equals("") ||
1235.                txtCVV.getText().equals("") ||
1236.                txtMM.getText().equals("") || txtAA.getText().
1237.                equals("")){
1238.                 btnPurchase.setEnabled(false);
1239.             } else {
1240.                 if(checkCar1 == true && checkCar2 == true &&
1241.                    checkCar3 == true && checkCar4 == true){
1242.                     btnPurchase.setEnabled(true);
1243.                 }
1244.             }
1245.         }
1246.
1247.         public void btnValidatePaypal() {
1248.             if(txtPaypalEmail.getText().equals("") ||
1249.                txtPaypalPass.getText().equals("")){
1250.                 btnPurchase.setEnabled(false);
1251.             } else {
1252.                 if(checkPay1 == true && checkPay2 == true){
1253.                     btnPurchase.setEnabled(true);
1254.                 }
1255.             }
1256.
1257.             public boolean isEmail(String correo) {
1258.                 Pattern pattern = Pattern.compile("^([0-9a-zA-
1259.                    Z]([_\\.w]*[0-9a-zA-Z])*)@([0-9a-zA-Z]([-w]*[0-9a-zA-Z].)+([a-zA-
1260.                    Z]{2,9}.)+[a-zA-Z]{2,3})$");
1261.                 Matcher mather = pattern.matcher(correo);
1262.                 if (mather.find()) {
1263.                     return true;
1264.                 }else{
1265.                     return false;
1266.                 }
1267.             }
1268.
1269.             public void validateRecive() {
1270.                 int i = JOptionPane.showConfirmDialog(
1271.                    null, "¿DESEAS GENERAR UNA FACTURA? \n\t
1272.                    SE CREARA UNA FACTURA PDF Y AL CORREO REGISTRADO",
1273.                    "", JOptionPane.YES_NO_OPTION,
1274.                    JOptionPane.DEFAULT_OPTION);
1275.                 if(i==0){
1276.                     String url = null;
1277.                     JFileChooser loadUrlFile1 = new
1278.                     JFileChooser();
1279.                     int o = loadUrlFile1.showSaveDialog(this);
1280.                     if(o== JFileChooser.APPROVE_OPTION){
1281.                         File fileUrl =
1282.                         loadUrlFile1.getSelectedFile();
1283.                         url = fileUrl.toString();
1284.                     }
1285.                     file.pdfCreateFile(url, txtPriceCar.getText(),
1286.                        txtExtra.getText(), txtPriceTotal.getText(), i);

```

```

1277.                pdfSendEmail(url,
Login.userEntity.getEmail());
1278.            }
1279.        }
1280.
1281.        public void pdfSendEmail(String url, String receiver){
1282.            try{
1283.                Properties props = new Properties();
1284.                props.put("mail.smtp.host", "smtp.gmail.com");
1285.                props.setProperty("mail.smtp.starttls.enable",
"true");
1286.                props.setProperty("mail.smtp.port", "587");
1287.                props.setProperty("mail.smtp.user",
"admicaragency01");
1288.                props.setProperty("mail.smtp.auth", "true");
1289.
1290.                Session session =
Session.getDefaultInstance(props, null);
1291.
1292.                BodyPart texto = new MimeBodyPart();
1293.                texto.setText("Gracias por comprar en Car
Agency :D");
1294.
1295.                BodyPart adjunto = new MimeBodyPart();
1296.                adjunto.setDataHandler(
new DataHandler(new
FileDataSource(url+".pdf")));
1298.                adjunto.setFileName("Factura.pdf");
1299.
1300.                MimeMultipart multiParte = new
MimeMultipart();
1301.                multiParte.addBodyPart(texto);
1302.                multiParte.addBodyPart(adjunto);
1303.
1304.                MimeMessage message = new
MimeMessage(session);
1305.                message.setFrom(new
InternetAddress("prueba@prueba.com"));
1306.                message.addRecipient(
Message.RecipientType.TO,
new InternetAddress(receiver));
1308.                message.setSubject("FACTURA - CAR AGENCY");
1309.                message.setContent(multiParte);
1310.
1311.
1312.                Transport t = session.getTransport("smtp");
1313.                t.connect("admicaragency01@gmail.com",
"contr@@10");
1314.                t.sendMessage(message,
message.getAllRecipients());
1315.                t.close();
1316.            } catch (Exception e) {
1317.                System.out.println(e);
1318.            }
1319.        }
1320.
1321.        /**
1322.         * @param args the command line arguments

```

```

1323.         */
1324.         public static void main(String args[]) {
1325.             /* Set the Nimbus look and feel */
1326.             //<editor-fold defaultstate="collapsed" desc="
Look and feel setting code (optional) ">
1327.             /* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.
1328.             * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
1329.             */
1330.             try {
1331.                 for (javax.swing.UIManager.LookAndFeelInfo
info : javax.swing.UIManager.getInstalledLookAndFeels()) {
1332.                     if ("Nimbus".equals(info.getName())) {
1333.                         javax.swing.UIManager.setLookAndFeel(info.getClassName());
1334.                         break;
1335.                     }
1336.                 }
1337.             } catch (ClassNotFoundException ex) {
1338.                 java.util.logging.Logger.getLogger(Purchase.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
1339.             } catch (InstantiationException ex) {
1340.                 java.util.logging.Logger.getLogger(Purchase.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
1341.             } catch (IllegalAccessException ex) {
1342.                 java.util.logging.Logger.getLogger(Purchase.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
1343.             } catch (javax.swing.UnsupportedLookAndFeelException ex) {
1344.                 java.util.logging.Logger.getLogger(Purchase.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
1345.             }
1346.             //</editor-fold>
1347.             //</editor-fold>
1348.
1349.             /* Create and display the form */
1350.             java.awt.EventQueue.invokeLater(new Runnable() {
1351.                 public void run() {
1352.                     new Purchase().setVisible(true);
1353.                 }
1354.             });
1355.         }
1356.
1357.         // Variables declaration - do not modify
1358.         private javax.swing.JButton btnPurchase;
1359.         private javax.swing.JLabel iconCar;
1360.         private javax.swing.JLabel jLabel1;
1361.         private javax.swing.JLabel jLabel10;
1362.         private javax.swing.JLabel jLabel11;
1363.         private javax.swing.JLabel jLabel2;
1364.         private javax.swing.JLabel jLabel3;

```

```

1365.         private javax.swing.JLabel jLabel4;
1366.         private javax.swing.JLabel jLabel5;
1367.         private javax.swing.JLabel jLabel6;
1368.         private javax.swing.JLabel jLabel7;
1369.         private javax.swing.JLabel jLabel8;
1370.         private javax.swing.JLabel jLabel9;
1371.         private javax.swing.JPanel jPanel1;
1372.         private javax.swing.JLabel labelAA;
1373.         private javax.swing.JLabel labelCVV;
1374.         private javax.swing.JLabel labelCar;
1375.         private javax.swing.JLabel labelMM;
1376.         private javax.swing.JLabel labelPaypalEmail;
1377.         private javax.swing.JLabel labelPaypalPass;
1378.         public static javax.swing.JComboBox<String>
            selectColors;
1379.         public static javax.swing.JComboBox<String>
            selectColorsNew;
1380.         public static javax.swing.JComboBox<String>
            selectCompany;
1381.         public static javax.swing.JComboBox<String>
            selectModel;
1382.         public static javax.swing.JComboBox<String>
            selectSerial;
1383.         public static javax.swing.JComboBox<String>
            selectState;
1384.         public static javax.swing.JComboBox<String>
            selectTypePayment;
1385.         private javax.swing.JTextField txtAA;
1386.         private javax.swing.JTextField txtCVV;
1387.         private javax.swing.JTextField txtCard;
1388.         private javax.swing.JLabel txtExtra;
1389.         private javax.swing.JTextField txtMM;
1390.         private javax.swing.JTextField txtPaypalEmail;
1391.         private javax.swing.JPasswordField txtPaypalPass;
1392.         private javax.swing.JLabel txtPriceCar;
1393.         private javax.swing.JLabel txtPriceTotal;
1394.         private javax.swing.JPanel typePaymentPanel;
1395.         // End of variables declaration
1396.     }

```

Quotation

```

1.  /*
2.  * To change this license header, choose License Headers in Project
    Properties.
3.  * To change this template file, choose Tools | Templates
4.  * and open the template in the editor.
5.  */
6.  package view.user;
7.
8.  import code.MethodsSQL;
9.  import java.text.DecimalFormat;
10. import javax.swing.ImageIcon;
11. import javax.swing.Login;
12.

```

```
13.  /**
14.   *
15.   * @author Restf
16.   */
17.  public class Quotation extends javax.swing.JFrame {
18.
19.      MethodsSQL methodsSql = new MethodsSQL();
20.      ImageIcon car = new
21.      ImageIcon(getClass().getResource("/resource/car.png"));
22.      ImageIcon nissanWhite = new
23.      ImageIcon(getClass().getResource("/resource/car-nissan-
24.      white.png"));
25.      ImageIcon nissanRed = new
26.      ImageIcon(getClass().getResource("/resource/car-nissan-red.png"));
27.      ImageIcon nissanOrange = new
28.      ImageIcon(getClass().getResource("/resource/car-nissan-
29.      orange.png"));
30.      ImageIcon nissanYellow = new
31.      ImageIcon(getClass().getResource("/resource/car-nissan-
32.      yellow.png"));
33.      ImageIcon nissanGreen = new
34.      ImageIcon(getClass().getResource("/resource/car-nissan-
35.      green.png"));
36.      ImageIcon nissanBlue = new
37.      ImageIcon(getClass().getResource("/resource/car-nissan-blue.png"));
38.      ImageIcon nissanGray = new
39.      ImageIcon(getClass().getResource("/resource/car-nissan-gray.png"));
40.
41.      ImageIcon audiWhite = new
42.      ImageIcon(getClass().getResource("/resource/car-audi-white.png"));
43.      ImageIcon audiRed = new
44.      ImageIcon(getClass().getResource("/resource/car-audi-red.png"));
45.      ImageIcon audiOrange = new
46.      ImageIcon(getClass().getResource("/resource/car-audi-orage.png"));
47.      ImageIcon audiYellow = new
48.      ImageIcon(getClass().getResource("/resource/car-audi-yellow.png"));
49.      ImageIcon audiGreen = new
50.      ImageIcon(getClass().getResource("/resource/car-audi-green.png"));
51.      ImageIcon audiBlue = new
52.      ImageIcon(getClass().getResource("/resource/car-audi-blue.png"));
53.      ImageIcon audiGray = new
54.      ImageIcon(getClass().getResource("/resource/car-audi-gray.png"));
55.
56.      ImageIcon volkswagenWhite = new
57.      ImageIcon(getClass().getResource("/resource/car-volkswagen-
58.      white.png"));
59.      ImageIcon volkswagenRed = new
60.      ImageIcon(getClass().getResource("/resource/car-volkswagen-
61.      red.png"));
62.      ImageIcon volkswagenOrange = new
63.      ImageIcon(getClass().getResource("/resource/car-volkswagen-
64.      orange.png"));
65.      ImageIcon volkswagenYellow = new
66.      ImageIcon(getClass().getResource("/resource/car-volkswagen-
67.      yellow.png"));
```

```

41.         ImageIcon volkswagenGreen = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
green.png"));
42.         ImageIcon volkswagenBlue = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
blue.png"));
43.         ImageIcon volkswagenGray = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
gray.png"));
44.
45.     public Quotation() {
46.         initComponents();
47.         setTitle("Car Agency - Cotización - [
"+Login.userEntity.getName()+" ]");
48.         setLocationRelativeTo(null);
49.         selectColorsNew.setVisible(false);
50.         btnPurchase.setEnabled(false);
51.     }
52.
53.     /**
54.      * This method is called from within the constructor to
initialize the form.
55.      * WARNING: Do NOT modify this code. The content of this
method is always
56.      * regenerated by the Form Editor.
57.      */
58.     @SuppressWarnings("unchecked")
59.     // <editor-fold defaultstate="collapsed" desc="Generated
Code">
60.     private void initComponents() {
61.
62.         jPanel1 = new javax.swing.JPanel();
63.         jLabel7 = new javax.swing.JLabel();
64.         iconCar = new javax.swing.JLabel();
65.         jLabel5 = new javax.swing.JLabel();
66.         selectCompany = new javax.swing.JComboBox<>();
67.         jLabel11 = new javax.swing.JLabel();
68.         jLabel2 = new javax.swing.JLabel();
69.         selectState = new javax.swing.JComboBox<>();
70.         selectModel = new javax.swing.JComboBox<>();
71.         jLabel3 = new javax.swing.JLabel();
72.         jLabel4 = new javax.swing.JLabel();
73.         selectColors = new javax.swing.JComboBox<>();
74.         jLabel6 = new javax.swing.JLabel();
75.         jLabel8 = new javax.swing.JLabel();
76.         jLabel9 = new javax.swing.JLabel();
77.         txtPriceTotal = new javax.swing.JLabel();
78.         txtExtra = new javax.swing.JLabel();
79.         txtPriceCar = new javax.swing.JLabel();
80.         jLabel10 = new javax.swing.JLabel();
81.         selectSerial = new javax.swing.JComboBox<>();
82.         selectColorsNew = new javax.swing.JComboBox<>();
83.         btnPurchase = new javax.swing.JButton();
84.
85.         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
;

```

```

86.         setResizable(false);
87.
88.         jPanel1.setBackground(new java.awt.Color(128, 153, 155));
89.         jPanel1.setPreferredSize(new java.awt.Dimension(1280,
90.             720));
91.         jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
92.             NOI18N
93.         jLabel7.setForeground(new java.awt.Color(0, 0, 0));
94.         jLabel7.setIcon(new
95.             javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
96.             ); // NOI18N
97.         jLabel7.setText("VOLVER");
98.         jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
99.             {
100.                 public void mouseClicked(java.awt.event.MouseEvent
101.                     evt) {
102.                         jLabel7MouseClicked(evt);
103.                     }
104.             });
105.         iconCar.setIcon(new
106.             javax.swing.ImageIcon(getClass().getResource("/resource/car.png")))
107.             ; // NOI18N
108.         jLabel5.setFont(new java.awt.Font("Dialog", 1, 18)); //
109.             NOI18N
110.         jLabel5.setForeground(new java.awt.Color(255, 255, 255));
111.         jLabel5.setText("► SELECCIONA LOS SIGUIENTES DATOS ◀");
112.         jLabel5.setToolTipText("");
113.         selectCompany.setBackground(new java.awt.Color(255, 255,
114.             255));
115.         selectCompany.setFont(new java.awt.Font("Dialog", 1, 18));
116.             // NOI18N
117.         selectCompany.setForeground(new java.awt.Color(0, 0, 0));
118.         selectCompany.setMaximumRowCount(4);
119.         selectCompany.setModel(new
120.             javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
121.                 NISSAN", ". AUDI", ". VOLKSWAGEN" }));
122.         selectCompany.setPreferredSize(new java.awt.Dimension(220,
123.             40));
124.         selectCompany.addActionListener(new
125.             java.awt.event.ActionListener() {
126.                 public void actionPerformed(java.awt.event.ActionEvent
127.                     evt) {
128.                         selectCompanyActionPerformed(evt);
129.                     }
130.             });
131.         jLabel1.setFont(new java.awt.Font("Dialog", 1, 18)); //
132.             NOI18N
133.         jLabel1.setForeground(new java.awt.Color(255, 255, 255));
134.         jLabel1.setText("» COMPañIA");
135.         jLabel1.setToolTipText("");
136.

```



```

125.        jLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
126.        jLabel2.setForeground(new java.awt.Color(255, 255, 255));
127.        jLabel2.setText(">> ESTADO");
128.        jLabel2.setToolTipText("");
129.
130.        selectState.setBackground(new java.awt.Color(255, 255,
        255));
131.        selectState.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
132.        selectState.setForeground(new java.awt.Color(0, 0, 0));
133.        selectState.setMaximumRowCount(4);
134.        selectState.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".•
        NUEVO", ".• SEMINUEVO", ".• USADO", " " }));
135.        selectState.setEnabled(false);
136.        selectState.setPreferredSize(new java.awt.Dimension(220,
        40));
137.        selectState.addActionListener(new
        java.awt.event.ActionListener() {
138.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
139.                selectStateActionPerformed(evt);
140.            }
141.        });
142.
143.        selectModel.setBackground(new java.awt.Color(255, 255,
        255));
144.        selectModel.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
145.        selectModel.setForeground(new java.awt.Color(0, 0, 0));
146.        selectModel.setMaximumRowCount(4);
147.        selectModel.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
        }));
148.        selectModel.setEnabled(false);
149.        selectModel.setPreferredSize(new java.awt.Dimension(220,
        40));
150.        selectModel.addActionListener(new
        java.awt.event.ActionListener() {
151.            public void actionPerformed(java.awt.event.ActionEvent
        evt) {
152.                selectModelActionPerformed(evt);
153.            }
154.        });
155.
156.        jLabel3.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
157.        jLabel3.setForeground(new java.awt.Color(255, 255, 255));
158.        jLabel3.setText(">> MODELO");
159.        jLabel3.setToolTipText("");
160.
161.        jLabel4.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
162.        jLabel4.setForeground(new java.awt.Color(255, 255, 255));
163.        jLabel4.setText(">> COLORES");
164.        jLabel4.setToolTipText("");

```

```

165.
166.         selectColors.setBackground(new java.awt.Color(255, 255,
255));
167.         selectColors.setFont(new java.awt.Font("Dialog", 1, 18));
// NOI18N
168.         selectColors.setForeground(new java.awt.Color(0, 0, 0));
169.         selectColors.setMaximumRowCount(4);
170.         selectColors.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
}));
171.         selectColors.setEnabled(false);
172.         selectColors.setPreferredSize(new java.awt.Dimension(220,
40));
173.         selectColors.addActionListener(new
java.awt.event.ActionListener() {
174.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
175.                 selectColorsActionPerformed(evt);
176.             }
177.         });
178.
179.         jLabel6.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
180.         jLabel6.setForeground(new java.awt.Color(255, 255, 255));
181.         jLabel6.setText("PRECIO:      $");
182.
183.         jLabel8.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
184.         jLabel8.setForeground(new java.awt.Color(255, 255, 255));
185.         jLabel8.setText("ADICIONES:  $");
186.
187.         jLabel9.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
188.         jLabel9.setForeground(new java.awt.Color(255, 255, 255));
189.         jLabel9.setText("TOTAL:      $");
190.
191.         txtPriceTotal.setFont(new java.awt.Font("Dialog", 1, 24));
// NOI18N
192.         txtPriceTotal.setForeground(new java.awt.Color(255, 255,
255));
193.         txtPriceTotal.setText("-----");
194.
195.         txtExtra.setFont(new java.awt.Font("Dialog", 1, 24)); //
NOI18N
196.         txtExtra.setForeground(new java.awt.Color(255, 255, 255));
197.         txtExtra.setText("-----");
198.
199.         txtPriceCar.setFont(new java.awt.Font("Dialog", 1, 24));
// NOI18N
200.         txtPriceCar.setForeground(new java.awt.Color(255, 255,
255));
201.         txtPriceCar.setText("-----");
202.
203.         jLabel10.setFont(new java.awt.Font("Dialog", 1, 18)); //
NOI18N
204.         jLabel10.setForeground(new java.awt.Color(255, 255, 255));
205.         jLabel10.setText(">> N! SERIE");

```

```

206.         jLabel10.setToolTipText("");
207.
208.         selectSerial.setBackground(new java.awt.Color(255, 255,
255));
209.         selectSerial.setFont(new java.awt.Font("Dialog", 1, 18));
// NOI18N
210.         selectSerial.setForeground(new java.awt.Color(0, 0, 0));
211.         selectSerial.setMaximumRowCount(4);
212.         selectSerial.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
}));
213.         selectSerial.setEnabled(false);
214.         selectSerial.setPreferredSize(new java.awt.Dimension(220,
40));
215.         selectSerial.addActionListener(new
java.awt.event.ActionListener() {
216.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
217.                 selectSerialActionPerformed(evt);
218.             }
219.         });
220.
221.         selectColorsNew.setBackground(new java.awt.Color(255, 255,
255));
222.         selectColorsNew.setFont(new java.awt.Font("Dialog", 1,
18)); // NOI18N
223.         selectColorsNew.setForeground(new java.awt.Color(0, 0,
0));
224.         selectColorsNew.setMaximumRowCount(4);
225.         selectColorsNew.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
BLANCO", ". ROJO", ". NARANJA", ". AMARILLO", ". VERDE", ". AZUL",
". GRIS" }));
226.         selectColorsNew.setEnabled(false);
227.         selectColorsNew.setPreferredSize(new
java.awt.Dimension(220, 40));
228.         selectColorsNew.addActionListener(new
java.awt.event.ActionListener() {
229.             public void actionPerformed(java.awt.event.ActionEvent
evt) {
230.                 selectColorsNewActionPerformed(evt);
231.             }
232.         });
233.
234.         btnPurchase.setFont(new java.awt.Font("Dialog", 1, 18));
// NOI18N
235.         btnPurchase.setForeground(new java.awt.Color(255, 255,
255));
236.         btnPurchase.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
price-off.png"))); // NOI18N
237.         btnPurchase.setText("¿DESEAS COMPRAR LO BUSCADO?");
238.         btnPurchase.setContentAreaFilled(false);
239.         btnPurchase.setEnabled(false);
240.
        btnPurchase.setHorizontalAlignment(javax.swing.SwingConstants.LEADING);

```

```

241.         btnPurchase.setRolloverIcon(new
            javax.swing.ImageIcon(getClass().getResource("/resource/MenuUser-
                price-on.png"))); // NOI18N
242.         btnPurchase.addActionListener(new
            java.awt.event.ActionListener() {
243.             public void actionPerformed(java.awt.event.ActionEvent
                evt) {
244.                 btnPurchaseActionPerformed(evt);
245.             }
246.         });
247.
248.         javax.swing.GroupLayout jPanel1Layout = new
            javax.swing.GroupLayout(jPanel1);
249.         jPanel1.setLayout(jPanel1Layout);
250.         jPanel1Layout.setHorizontalGroup(
251.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                .LEADING)
252.                 .addGroup(jPanel1Layout.createSequentialGroup()
253.                     .addGap(1092, 1092, 1092)
254.                     .addComponent(jLabel7))
255.                 .addGroup(jPanel1Layout.createSequentialGroup()
256.                     .addGap(110, 110, 110)
257.                     .addComponent(jLabel5))
258.                 .addGroup(jPanel1Layout.createSequentialGroup()
259.                     .addGap(90, 90, 90)
260.                     .addComponent(btnPurchase,
                javax.swing.GroupLayout.PREFERRED_SIZE, 490,
                javax.swing.GroupLayout.PREFERRED_SIZE)
261.                     .addGap(140, 140, 140)
262.                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
263.                         .addComponent(jLabel8,
                javax.swing.GroupLayout.PREFERRED_SIZE, 167,
                javax.swing.GroupLayout.PREFERRED_SIZE)
264.                         .addComponent(jLabel9,
                javax.swing.GroupLayout.PREFERRED_SIZE, 167,
                javax.swing.GroupLayout.PREFERRED_SIZE))
265.                     .addGap(3, 3, 3)
266.                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
267.                         .addComponent(txtExtra,
                javax.swing.GroupLayout.PREFERRED_SIZE, 250,
                javax.swing.GroupLayout.PREFERRED_SIZE)
268.                         .addComponent(txtPriceTotal,
                javax.swing.GroupLayout.PREFERRED_SIZE, 250,
                javax.swing.GroupLayout.PREFERRED_SIZE))
269.                 .addGroup(jPanel1Layout.createSequentialGroup()
270.                     .addGap(78, 78, 78)
271.                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
272.                         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.TRAILING)

```

```

273.                .addComponent(jLabel2)
274.                .addComponent(jLabel3)
275.                .addComponent(jLabel10))
276.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
277.                .addComponent(jLabel1,
        javax.swing.GroupLayout.Alignment.TRAILING)
278.        .addGroup(jPanell1Layout.createSequentialGroup())
279.                .addGap(2, 2, 2)
280.                .addComponent(jLabel4)))
281.        .addGap(18, 18, 18)
282.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
283.        .addGroup(jPanell1Layout.createSequentialGroup())
284.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
285.                .addComponent(selectCompany,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE)
286.                .addComponent(selectState,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE)
287.                .addComponent(selectModel,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE)
288.                .addComponent(selectColors,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE)
289.                .addComponent(selectColorsNew,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE))
290.        .addGap(100, 100, 100)
291.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
        .Alignment.LEADING)
292.                .addComponent(iconCar)
293.        .addGroup(jPanell1Layout.createSequentialGroup())
294.                .addGap(90, 90, 90)
295.                .addComponent(jLabel6)
296.                .addGap(3, 3, 3)
297.                .addComponent(txtPriceCar,
        javax.swing.GroupLayout.PREFERRED_SIZE, 240,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
298.        .addComponent(selectSerial,
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,
        javax.swing.GroupLayout.PREFERRED_SIZE))
299.    );
300.    jPanell1Layout.setVerticalGroup(
301.    jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)

```

```

302.         .addGroup(jPanell1Layout.createSequentialGroup())
303.         .addGap(15, 15, 15)
304.         .addComponent(jLabel7,
    javax.swing.GroupLayout.PREFERRED_SIZE, 66,
    javax.swing.GroupLayout.PREFERRED_SIZE)
305.         .addGap(49, 49, 49)
306.         .addComponent(jLabel5)
307.         .addGap(26, 26, 26)
308.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
309.
    .addGroup(jPanell1Layout.createSequentialGroup())
310.        .addComponent(iconCar)
311.        .addGap(7, 7, 7)
312.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.LEADING)
313.        .addComponent(jLabel6)
314.        .addComponent(txtPriceCar))
315.
    .addGroup(jPanell1Layout.createSequentialGroup())
316.        .addGap(2, 2, 2)
317.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
318.        .addComponent(selectCompany,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
319.        .addComponent(jLabel1))
320.        .addGap(8, 8, 8)
321.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
322.        .addComponent(selectState,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
323.        .addComponent(jLabel2))
324.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
325.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)
326.        .addComponent(selectModel,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)
327.        .addComponent(jLabel3))
328.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
    )
329.
    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
    .Alignment.BASELINE)

```

```

330.                .addComponent(selectSerial,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
331.                .addComponent(jLabel10))
332.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
333.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
334.                .addComponent(selectColors,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
335.                .addComponent(jLabel14))
336.        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
337.        .addComponent(selectColorsNew,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
338.        .addGap(8, 8, 8)
339.        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
340.                .addGroup(jPanell1Layout.createSequentialGroup())
341.                .addGap(30, 30, 30)
342.                .addComponent(btnPurchase,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
343.        .addGroup(jPanell1Layout.createSequentialGroup())
344.                .addComponent(jLabel8)
345.                .addGap(8, 8, 8)
346.                .addComponent(jLabel9))
347.        .addGroup(jPanell1Layout.createSequentialGroup())
348.                .addComponent(txtExtra)
349.                .addGap(8, 8, 8)
350.                .addComponent(txtPriceTotal))))
351.    );
352.
353.    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
354.    getContentPane().setLayout(layout);
355.    layout.setHorizontalGroup(
356.        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
357.            .addGroup(layout.createSequentialGroup()
358.                .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

359.         .addGap(0, 0, Short.MAX_VALUE))
360.     );
361.     layout.setVerticalGroup(
362.         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
363.         .addGroup(layout.createSequentialGroup()
364.             .addComponent(jPanel1,
365.                 javax.swing.GroupLayout.PREFERRED_SIZE,
366.                 javax.swing.GroupLayout.DEFAULT_SIZE,
367.                 javax.swing.GroupLayout.PREFERRED_SIZE)
368.             .addGap(0, 0, Short.MAX_VALUE))
369.         );
370.     pack();
371. } // </editor-fold>
372.
373. private void jLabel7MouseClicked(java.awt.event.MouseEvent
374.     evt) {
375.     this.setVisible(false);
376.     MenuUser menu = new MenuUser();
377.     menu.setVisible(true);
378.     dispose();
379. }
380.
381. private void
382.     selectCompanyActionPerformed(java.awt.event.ActionEvent evt) {
383.         if (selectCompany.getSelectedIndex() > 0) {
384.             selectState.setEnabled(true);
385.             selectState.setSelectedIndex(0);
386.             iconCar.setIcon(car);
387.             txtPriceCar.setText("-----");
388.             txtExtra.setText("-----");
389.             txtPriceTotal.setText("-----");
390.             switch(selectCompany.getSelectedIndex()){
391.                 case 1:
392.                     iconCar.setIcon(nissanWhite);
393.                     break;
394.                 case 2:
395.                     iconCar.setIcon(audiWhite);
396.                     break;
397.                 case 3:
398.                     iconCar.setIcon(volkswagenWhite);
399.                     break;
400.             }
401.         } else {
402.             clear();
403.         }
404.     }
405.
406. private void
407.     selectStateActionPerformed(java.awt.event.ActionEvent evt) {
408.         if (selectState.getSelectedIndex() > 0) {
409.             selectModel.setEnabled(true);
410.             selectSerial.setEnabled(false);
411.             selectSerial.setSelectedIndex(0);
412.             txtPriceCar.setText("-----");

```



```

408.         txtExtra.setText("-----");
409.         txtPriceTotal.setText("-----");
410.         switch(selectCompany.getSelectedIndex()){
411.             case 1:
412.                 iconCar.setIcon(nissanWhite);
413.                 break;
414.             case 2:
415.                 iconCar.setIcon(audiWhite);
416.                 break;
417.             case 3:
418.                 iconCar.setIcon(volkswagenWhite);
419.                 break;
420.         }
421.
422.         selectModel.setModel(methodsSql.loadModels(selectCompany.getSelectedIndex(), selectState.getSelectedIndex()));
423.     } else {
424.         iconCar.setIcon(car);
425.         selectModel.setEnabled(false);
426.         selectSerial.setEnabled(false);
427.         selectColors.setEnabled(false);
428.         selectModel.setSelectedIndex(0);
429.         selectSerial.setSelectedIndex(0);
430.         selectColors.setSelectedIndex(0);
431.     }
432.
433.     private void
434.     selectModelActionPerformed(java.awt.event.ActionEvent evt) {
435.         if (selectModel.getSelectedIndex() > 0) {
436.             selectSerial.setEnabled(true);
437.             String txtmodel =
438.             selectModel.getSelectedItem().toString();
439.             String model = txtmodel.substring(2);
440.             selectSerial.setModel(methodsSql.loadSerial(model));
441.             txtPriceCar.setText("-----");
442.             txtExtra.setText("-----");
443.             txtPriceTotal.setText("-----");
444.         } else {
445.             selectSerial.setEnabled(false);
446.             selectColors.setEnabled(false);
447.             selectSerial.setSelectedIndex(0);
448.             selectColors.setSelectedIndex(0);
449.         }
450.
451.     private void
452.     selectColorsActionPerformed(java.awt.event.ActionEvent evt) {
453.         if(!selectColors.getSelectedItem().equals("Seleccionar")
454.         &&
455.         !selectColors.getSelectedItem().equals("• Otro: ")){
456.             String txtcompany =
457.             selectCompany.getSelectedItem().toString();
458.             String company = txtcompany.substring(2);
459.             String txtcolor =
460.             selectColors.getSelectedItem().toString();
461.             String color = txtcolor.substring(2);

```

```
457.         switch(company) {
458.             case "NISSAN":
459.                 switch(color) {
460.                     case "BLANCO":
461.                         iconCar.setIcon(nissanWhite);
462.                         break;
463.                     case "ROJO":
464.                         iconCar.setIcon(nissanRed);
465.                         break;
466.                     case "NARANJA":
467.                         iconCar.setIcon(nissanOrange);
468.                         break;
469.                     case "AMARILLO":
470.                         iconCar.setIcon(nissanYellow);
471.                         break;
472.                     case "VERDE":
473.                         iconCar.setIcon(nissanGreen);
474.                         break;
475.                     case "AZUL":
476.                         iconCar.setIcon(nissanBlue);
477.                         break;
478.                     case "GRIS":
479.                         iconCar.setIcon(nissanGray);
480.                         break;
481.                 }
482.             break;
483.             case "AUDI":
484.                 switch(color) {
485.                     case "BLANCO":
486.                         iconCar.setIcon(audiWhite);
487.                         break;
488.                     case "ROJO":
489.                         iconCar.setIcon(audiRed);
490.                         break;
491.                     case "NARANJA":
492.                         iconCar.setIcon(audiOrange);
493.                         break;
494.                     case "AMARILLO":
495.                         iconCar.setIcon(audiYellow);
496.                         break;
497.                     case "VERDE":
498.                         iconCar.setIcon(audiGreen);
499.                         break;
500.                     case "AZUL":
501.                         iconCar.setIcon(audiBlue);
502.                         break;
503.                     case "GRIS":
504.                         iconCar.setIcon(audiGray);
505.                         break;
506.                 }
507.             break;
508.             case "VOLKSWAGEN":
509.                 switch(color) {
510.                     case "BLANCO":
511.                         iconCar.setIcon(volkswagenWhite);
512.                         break;
513.                     case "ROJO":
```

```

514.         iconCar.setIcon(volkswagenRed);
515.         break;
516.         case "NARANJA":
517.             iconCar.setIcon(volkswagenOrange);
518.             break;
519.         case "AMARILLO":
520.             iconCar.setIcon(volkswagenYellow);
521.             break;
522.         case "VERDE":
523.             iconCar.setIcon(volkswagenGreen);
524.             break;
525.         case "AZUL":
526.             iconCar.setIcon(volkswagenBlue);
527.             break;
528.         case "GRIS":
529.             iconCar.setIcon(volkswagenGray);
530.             break;
531.     }
532.     break;
533. }
534.     btnPurchase.setEnabled(true);
535. } else if(selectColors.getSelectedItem().equals("• Otro:
    ")) {
536.     selectColorsNew.setVisible(true);
537.     selectColorsNew.setEnabled(true);
538.     selectColorsNew.setSelectedIndex(0);
539.     switch(selectCompany.getSelectedIndex()) {
540.         case 1:
541.             iconCar.setIcon(nissanWhite);
542.             break;
543.         case 2:
544.             iconCar.setIcon(audiWhite);
545.             break;
546.         case 3:
547.             iconCar.setIcon(volkswagenWhite);
548.             break;
549.     }
550.     } else {
551.         selectColorsNew.setSelectedIndex(0);
552.         selectColorsNew.setVisible(false);
553.     }
554.
555.
556.
557.
558.     }
559.
560.     private void
        selectSerialActionPerformed(java.awt.event.ActionEvent evt) {
561.         if (selectSerial.getSelectedIndex() > 0) {
562.             selectColors.setEnabled(true);
563.             String txt =
                selectSerial.getSelectedItem().toString();
564.             String txtmodel = txt.substring(2);
565.             int model = Integer.parseInt(txtmodel);
566.             selectColors.setModel(methodsSql.loadColorsQuotation(model));

```

```
567.         String s =
568.             moneyFormat (methodsSql.loadPriceCar (model));
569.             txtPriceCar.setText(s);
570.             if(!selectColorsNew.isEnabled()){
571.                 txtPriceTotal.setText(s);
572.             }
573.         } else {
574.             selectColors.setEnabled(false);
575.             selectColors.setSelectedIndex(0);
576.         }
577.     }
578. }
579.
580. private void
581.     selectColorsNewActionPerformed(java.awt.event.ActionEvent evt) {
582.         if(selectColorsNew.isEnabled()){
583.             String txtcompany =
584.                 selectCompany.getSelectedItem().toString();
585.             String company = txtcompany.substring(2);
586.             String txtcolor =
587.                 selectColorsNew.getSelectedItem().toString();
588.             String color = txtcolor.substring(2);
589.             int price = 0;
590.             switch(company){
591.                 case "NISSAN":
592.                     switch(color){
593.                         case "BLANCO":
594.                             price = 15000;
595.                             iconCar.setIcon(nissanWhite);
596.                             break;
597.                         case "ROJO":
598.                             price = 10000;
599.                             iconCar.setIcon(nissanRed);
600.                             break;
601.                         case "NARANJA":
602.                             price = 5000;
603.                             iconCar.setIcon(nissanOrange);
604.                             break;
605.                         case "AMARILLO":
606.                             price = 8000;
607.                             iconCar.setIcon(nissanYellow);
608.                             break;
609.                         case "VERDE":
610.                             price = 9000;
611.                             iconCar.setIcon(nissanGreen);
612.                             break;
613.                         case "AZUL":
614.                             price = 12000;
615.                             iconCar.setIcon(nissanBlue);
616.                             break;
617.                         case "GRIS":
618.                             price = 13000;
619.                             iconCar.setIcon(nissanGray);
620.                             break;
621.                     }
622.                 break;
623.             }
624.         }
625.     }
626. }
```

```
620.         case "AUDI":
621.         switch(color){
622.             case "BLANCO":
623.                 price = 15000;
624.                 iconCar.setIcon(audiWhite);
625.                 break;
626.             case "ROJO":
627.                 price = 10000;
628.                 iconCar.setIcon(audiRed);
629.                 break;
630.             case "NARANJA":
631.                 price = 5000;
632.                 iconCar.setIcon(audiOrange);
633.                 break;
634.             case "AMARILLO":
635.                 price = 8000;
636.                 iconCar.setIcon(audiYellow);
637.                 break;
638.             case "VERDE":
639.                 price = 9000;
640.                 iconCar.setIcon(audiGreen);
641.                 break;
642.             case "AZUL":
643.                 price = 12000;
644.                 iconCar.setIcon(audiBlue);
645.                 break;
646.             case "GRIS":
647.                 price = 13000;
648.                 iconCar.setIcon(audiGray);
649.                 break;
650.         }
651.         break;
652.         case "VOLKSWAGEN":
653.         switch(color){
654.             case "BLANCO":
655.                 price = 15000;
656.                 iconCar.setIcon(volkswagenWhite);
657.                 break;
658.             case "ROJO":
659.                 price = 10000;
660.                 iconCar.setIcon(volkswagenRed);
661.                 break;
662.             case "NARANJA":
663.                 price = 5000;
664.                 iconCar.setIcon(volkswagenOrange);
665.                 break;
666.             case "AMARILLO":
667.                 price = 8000;
668.                 iconCar.setIcon(volkswagenYellow);
669.                 break;
670.             case "VERDE":
671.                 price = 9000;
672.                 iconCar.setIcon(volkswagenGreen);
673.                 break;
674.             case "AZUL":
675.                 price = 12000;
676.                 iconCar.setIcon(volkswagenBlue);
```

```

677.             break;
678.             case "GRIS":
679.                 price = 13000;
680.                 iconCar.setIcon(volkswagenGray);
681.                 break;
682.             }
683.         break;
684.     }
685.
686.     String s = moneyFormat(price);
687.     txtExtra.setText(s);
688.     String txt =
        selectSerial.getSelectedItemAt().toString();
689.     String txtmodel = txt.substring(2);
690.     int model = Integer.parseInt(txtmodel);
691.     int extra = methodsSql.loadPriceCar(model);
692.     int price_total = price + extra;
693.     txtPriceTotal.setText(moneyFormat(price_total));
694.
        if(!selectColorsNew.getSelectedItemAt().equals("Seleccionar")){
695.            btnPurchase.setEnabled(true);
696.        } else {
697.            btnPurchase.setEnabled(false);
698.        }
699.    }
700.
701.    }
702.
703.    private void
        btnPurchaseActionPerformed(java.awt.event.ActionEvent evt) {
704.        MenuUser.ReloadData = true;
705.        this.setVisible(false);
706.        Purchase purchase = new Purchase();
707.        purchase.setVisible(true);
708.    }
709.
710.
711.    private String moneyFormat(int i){
712.        DecimalFormat money = new DecimalFormat("###,###,###");
713.        String s = money.format(i);
714.        return s;
715.    }
716.
717.    private void clear(){
718.        selectState.setEnabled(false);
719.        selectModel.setEnabled(false);
720.        selectColors.setEnabled(false);
721.        selectSerial.setEnabled(false);
722.        selectState.setSelectedIndex(0);
723.        selectModel.setSelectedIndex(0);
724.        selectSerial.setSelectedIndex(0);
725.        selectColors.setSelectedIndex(0);
726.        iconCar.setIcon(car);
727.    }
728.    /**
729.     * @param args the command line arguments
730.     */

```

```

731.     public static void main(String args[]) {
732.         /* Set the Nimbus look and feel */
733.         //editor-fold defaultstate="collapsed" desc=" Look and
feel setting code (optional) ">
734.         /* If Nimbus (introduced in Java SE 6) is not available,
stay with the default look and feel.
735.         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
736.         */
737.         try {
738.             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
739.                 if ("Nimbus".equals(info.getName())) {
740.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
741.                     break;
742.                 }
743.             }
744.         } catch (ClassNotFoundException ex) {
745.             java.util.logging.Logger.getLogger(Quotation.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
746.         } catch (InstantiationException ex) {
747.             java.util.logging.Logger.getLogger(Quotation.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
748.         } catch (IllegalAccessException ex) {
749.             java.util.logging.Logger.getLogger(Quotation.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
750.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
751.             java.util.logging.Logger.getLogger(Quotation.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
752.         }
753.         //editor-fold>
754.         //editor-fold>
755.
756.         /* Create and display the form */
757.         java.awt.EventQueue.invokeLater(new Runnable() {
758.             public void run() {
759.                 new Quotation().setVisible(true);
760.             }
761.         });
762.     }
763.
764.     // Variables declaration - do not modify
765.     private javax.swing.JButton btnPurchase;
766.     private javax.swing.JLabel iconCar;
767.     private javax.swing.JLabel jLabel1;
768.     private javax.swing.JLabel jLabel10;
769.     private javax.swing.JLabel jLabel2;
770.     private javax.swing.JLabel jLabel3;
771.     private javax.swing.JLabel jLabel4;
772.     private javax.swing.JLabel jLabel5;
773.     private javax.swing.JLabel jLabel6;

```

```

774.     private javax.swing.JLabel jLabel7;
775.     private javax.swing.JLabel jLabel8;
776.     private javax.swing.JLabel jLabel9;
777.     private javax.swing.JPanel jPanel1;
778.     public static javax.swing.JComboBox<String> selectColors;
779.     public static javax.swing.JComboBox<String> selectColorsNew;
780.     public static javax.swing.JComboBox<String> selectCompany;
781.     public static javax.swing.JComboBox<String> selectModel;
782.     public static javax.swing.JComboBox<String> selectSerial;
783.     public static javax.swing.JComboBox<String> selectState;
784.     public static javax.swing.JLabel txtExtra;
785.     public static javax.swing.JLabel txtPriceCar;
786.     public static javax.swing.JLabel txtPriceTotal;
787.     // End of variables declaration
788. }

```

Vehicle

```

1.  /*
2.   * To change this license header, choose License Headers in Project
  Properties.
3.   * To change this template file, choose Tools | Templates
4.   * and open the template in the editor.
5.   */
6.  package view.user;
7.
8.  import code.MethodsSQL;
9.  import javax.swing.ImageIcon;
10. import view.Login;
11.
12. /**
13.  *
14.  * @author Restf
15.  */
16. public class Vehicle extends javax.swing.JFrame {
17.
18.     MethodsSQL methodsSql = new MethodsSQL();
19.     ImageIcon car = new
  ImageIcon(getClass().getResource("/resource/car.png"));
20.     ImageIcon nissanWhite = new
  ImageIcon(getClass().getResource("/resource/car-nissan-
  white.png"));
21.     ImageIcon nissanRed = new
  ImageIcon(getClass().getResource("/resource/car-nissan-red.png"));
22.     ImageIcon nissanOrange = new
  ImageIcon(getClass().getResource("/resource/car-nissan-
  orange.png"));
23.     ImageIcon nissanYellow = new
  ImageIcon(getClass().getResource("/resource/car-nissan-
  yellow.png"));
24.     ImageIcon nissanGreen = new
  ImageIcon(getClass().getResource("/resource/car-nissan-
  green.png"));
25.     ImageIcon nissanBlue = new
  ImageIcon(getClass().getResource("/resource/car-nissan-blue.png"));

```



```

26.         ImageIcon nissanGray    = new
ImageIcon(getClass().getResource("/resource/car-nissan-gray.png"));
27.
28.         ImageIcon audiWhite    = new
ImageIcon(getClass().getResource("/resource/car-audi-white.png"));
29.         ImageIcon audiRed      = new
ImageIcon(getClass().getResource("/resource/car-audi-red.png"));
30.         ImageIcon audiOrange   = new
ImageIcon(getClass().getResource("/resource/car-audi-orage.png"));
31.         ImageIcon audiYellow   = new
ImageIcon(getClass().getResource("/resource/car-audi-yellow.png"));
32.         ImageIcon audiGreen    = new
ImageIcon(getClass().getResource("/resource/car-audi-green.png"));
33.         ImageIcon audiBlue     = new
ImageIcon(getClass().getResource("/resource/car-audi-blue.png"));
34.         ImageIcon audiGray     = new
ImageIcon(getClass().getResource("/resource/car-audi-gray.png"));
35.
36.         ImageIcon volkswagenWhite = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
white.png"));
37.         ImageIcon volkswagenRed   = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
red.png"));
38.         ImageIcon volkswagenOrange = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
orange.png"));
39.         ImageIcon volkswagenYellow = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
yellow.png"));
40.         ImageIcon volkswagenGreen = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
green.png"));
41.         ImageIcon volkswagenBlue  = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
blue.png"));
42.         ImageIcon volkswagenGray  = new
ImageIcon(getClass().getResource("/resource/car-volkswagen-
gray.png"));
43.
44.         public Vehicle() {
45.             initComponents();
46.             setTitle("Car Agency - Vehiculos - [
"+Login.userEntity.getName()+" ]");
47.             setLocationRelativeTo(null);
48.
49.         }
50.
51.         /**
52.          * This method is called from within the constructor to
initialize the form.
53.          * WARNING: Do NOT modify this code. The content of this
method is always
54.          * regenerated by the Form Editor.
55.          */
56.         @SuppressWarnings("unchecked")

```

```

57.      // <editor-fold defaultstate="collapsed" desc="Generated
      Code">
58.      private void initComponents() {
59.
60.          jPanel1 = new javax.swing.JPanel();
61.          jLabel7 = new javax.swing.JLabel();
62.          jLabel5 = new javax.swing.JLabel();
63.          jLabel11 = new javax.swing.JLabel();
64.          selectCompany = new javax.swing.JComboBox<>();
65.          jLabel2 = new javax.swing.JLabel();
66.          selectState = new javax.swing.JComboBox<>();
67.          jLabel3 = new javax.swing.JLabel();
68.          selectModel = new javax.swing.JComboBox<>();
69.          jLabel4 = new javax.swing.JLabel();
70.          selectColors = new javax.swing.JComboBox<>();
71.          iconCar = new javax.swing.JLabel();
72.
73.          setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
          ;
74.          setPreferredSize(new java.awt.Dimension(1280, 720));
75.          setResizable(false);
76.
77.          jPanel1.setBackground(new java.awt.Color(128, 153, 155));
78.          jPanel1.setPreferredSize(new java.awt.Dimension(1280,
79.          720));
80.          jLabel7.setFont(new java.awt.Font("Dialog", 1, 24)); //
      NOI18N
81.          jLabel7.setForeground(new java.awt.Color(0, 0, 0));
82.          jLabel7.setIcon(new
      javax.swing.ImageIcon(getClass().getResource("/resource/back.png"))
      ); // NOI18N
83.          jLabel7.setText("VOLVER");
84.          jLabel7.addMouseListener(new java.awt.event.MouseAdapter()
      {
85.              public void mouseClicked(java.awt.event.MouseEvent
      evt) {
86.                  jLabel7MouseClicked(evt);
87.              }
88.          });
89.
90.          jLabel5.setFont(new java.awt.Font("Dialog", 1, 18)); //
      NOI18N
91.          jLabel5.setForeground(new java.awt.Color(255, 255, 255));
92.          jLabel5.setText("► SELECCIONA LOS SIGUIENTES DATOS ◀");
93.          jLabel5.setToolTipText("");
94.
95.          jLabel11.setFont(new java.awt.Font("Dialog", 1, 18)); //
      NOI18N
96.          jLabel11.setForeground(new java.awt.Color(255, 255, 255));
97.          jLabel11.setText("» COMPAÑIA");
98.          jLabel11.setToolTipText("");
99.
100.         selectCompany.setBackground(new java.awt.Color(255, 255,
255));

```

```

101.         selectCompany.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
102.         selectCompany.setForeground(new java.awt.Color(0, 0, 0));
103.         selectCompany.setMaximumRowCount(4);
104.         selectCompany.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
        NISSAN", ". AUDI", ". VOLKSWAGEN" }));
105.         selectCompany.setPreferredSize(new java.awt.Dimension(220,
        40));
106.         selectCompany.addActionListener(new
        java.awt.event.ActionListener() {
107.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
108.                 selectCompanyActionPerformed(evt);
109.             }
110.         });
111.
112.         jLabel2.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
113.         jLabel2.setForeground(new java.awt.Color(255, 255, 255));
114.         jLabel2.setText(">> ESTADO");
115.         jLabel2.setToolTipText("");
116.
117.         selectState.setBackground(new java.awt.Color(255, 255,
        255));
118.         selectState.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
119.         selectState.setForeground(new java.awt.Color(0, 0, 0));
120.         selectState.setMaximumRowCount(4);
121.         selectState.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", ".
        NUEVO", ". SEMINUEVO", ". USADO", " " }));
122.         selectState.setEnabled(false);
123.         selectState.setPreferredSize(new java.awt.Dimension(220,
        40));
124.         selectState.addActionListener(new
        java.awt.event.ActionListener() {
125.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
126.                 selectStateActionPerformed(evt);
127.             }
128.         });
129.
130.         jLabel3.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
131.         jLabel3.setForeground(new java.awt.Color(255, 255, 255));
132.         jLabel3.setText(">> MODELO");
133.         jLabel3.setToolTipText("");
134.
135.         selectModel.setBackground(new java.awt.Color(255, 255,
        255));
136.         selectModel.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
137.         selectModel.setForeground(new java.awt.Color(0, 0, 0));
138.         selectModel.setMaximumRowCount(4);

```

```

139.         selectModel.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar"
        }));
140.         selectModel.setEnabled(false);
141.         selectModel.setPreferredSize(new java.awt.Dimension(220,
        40));
142.         selectModel.addActionListener(new
        java.awt.event.ActionListener() {
143.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
144.                 selectModelActionPerformed(evt);
145.             }
146.         });
147.
148.         jLabel4.setFont(new java.awt.Font("Dialog", 1, 18)); //
        NOI18N
149.         jLabel4.setForeground(new java.awt.Color(255, 255, 255));
150.         jLabel4.setText(">> COLORES");
151.         jLabel4.setToolTipText("");
152.
153.         selectColors.setBackground(new java.awt.Color(255, 255,
        255));
154.         selectColors.setFont(new java.awt.Font("Dialog", 1, 18));
        // NOI18N
155.         selectColors.setForeground(new java.awt.Color(0, 0, 0));
156.         selectColors.setMaximumRowCount(4);
157.         selectColors.setModel(new
        javax.swing.DefaultComboBoxModel<>(new String[] { "Seleccionar", "•
        BLANCO", "• ROJO", "• NARANJA", "• AMARILLO", "• VERDE", "• AZUL",
        "• GRIS" }));
158.         selectColors.setEnabled(false);
159.         selectColors.setPreferredSize(new java.awt.Dimension(220,
        40));
160.         selectColors.addActionListener(new
        java.awt.event.ActionListener() {
161.             public void actionPerformed(java.awt.event.ActionEvent
        evt) {
162.                 selectColorsActionPerformed(evt);
163.             }
164.         });
165.
166.         iconCar.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/resource/car.png")))
        ; // NOI18N
167.
168.         javax.swing.GroupLayout jPanel1Layout = new
        javax.swing.GroupLayout(jPanel1);
169.         jPanel1.setLayout(jPanel1Layout);
170.         jPanel1Layout.setHorizontalGroup(
171.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
        .LEADING)
172.                 .addGroup(jPanel1Layout.createSequentialGroup()
173.                     .addGap(1091, 1091, 1091)
174.                     .addComponent(jLabel7)

```

```
175.        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE))  
176.            .addGroup(jPanell1Layout.createSequentialGroup()  
177.  
        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout  
        .Alignment.LEADING)  
178.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
179.            .addGap(59, 59, 59)  
180.  
        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout  
        .Alignment.LEADING)  
181.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
182.            .addComponent(jLabel4)  
183.            .addGap(18, 18, 18)  
184.            .addComponent(selectColors,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,  
        javax.swing.GroupLayout.PREFERRED_SIZE))  
185.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
186.            .addGap(3, 3, 3)  
187.  
        .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout  
        .Alignment.LEADING)  
188.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
189.            .addComponent(jLabel1)  
190.            .addGap(6, 6, 6)  
191.  
        .addComponent(selectCompany,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,  
        javax.swing.GroupLayout.PREFERRED_SIZE))  
192.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
193.            .addComponent(jLabel2)  
194.            .addGap(29, 29, 29)  
195.            .addComponent(selectState,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,  
        javax.swing.GroupLayout.PREFERRED_SIZE))  
196.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
197.            .addComponent(jLabel3)  
198.            .addGap(25, 25, 25)  
199.            .addComponent(selectModel,  
        javax.swing.GroupLayout.PREFERRED_SIZE, 320,  
        javax.swing.GroupLayout.PREFERRED_SIZE))))))  
200.  
        .addGroup(jPanell1Layout.createSequentialGroup()  
201.            .addGap(85, 85, 85)  
202.            .addComponent(jLabel5)))  
203.  
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED  
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)  
204.            .addComponent(iconCar)  
205.            .addGap(110, 110, 110))
```

```

206.         );
207.         jPanell1Layout.setVerticalGroup(
208.             jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
                .LEADING)
209.                 .addGroup(jPanell1Layout.createSequentialGroup())
210.                     .addGap(6, 6, 6)
211.                     .addComponent(jLabel17,
                javax.swing.GroupLayout.PREFERRED_SIZE, 66,
                javax.swing.GroupLayout.PREFERRED_SIZE)
212.                 .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
213.                     .addGroup(jPanell1Layout.createSequentialGroup())
214.                         .addGap(151, 151, 151)
215.                         .addComponent(jLabel15)
216.                         .addGap(32, 32, 32)
217.                     .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
218.                         .addGroup(jPanell1Layout.createSequentialGroup())
219.                             .addGap(8, 8, 8)
220.                             .addComponent(jLabel11))
221.                             .addComponent(selectCompany,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE))
222.                             .addGap(6, 6, 6)
223.                         .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
224.                             .addGroup(jPanell1Layout.createSequentialGroup())
225.                                 .addGap(8, 8, 8)
226.                                 .addComponent(jLabel12))
227.                                 .addComponent(selectState,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE))
228.                                 .addGap(6, 6, 6)
229.                             .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)
230.                                 .addGroup(jPanell1Layout.createSequentialGroup())
231.                                     .addGap(8, 8, 8)
232.                                     .addComponent(jLabel13))
233.                                     .addComponent(selectModel,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE))
234.                                     .addGap(6, 6, 6)
235.                                 .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout
                .Alignment.LEADING)

```

```

236.        .addGroup(jPanell1Layout.createSequentialGroup())
237.                .addGap(8, 8, 8)
238.                .addComponent(jLabel4))
239.                .addComponent(selectColors,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)))
240.
241.        .addGroup(jPanell1Layout.createSequentialGroup())
242.                .addGap(173, 173, 173)
243.                .addComponent(iconCar)))
244.        );
245.
246.        javax.swing.GroupLayout layout = new
        javax.swing.GroupLayout(getContentPane());
247.        getContentPane().setLayout(layout);
248.        layout.setHorizontalGroup(
249.        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
250.                .addGroup(layout.createSequentialGroup()
251.                        .addComponent(jPanell1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
252.                        .addGap(0, 0, Short.MAX_VALUE))
253.                );
254.        layout.setVerticalGroup(
255.        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
256.                .addGroup(layout.createSequentialGroup()
257.                        .addComponent(jPanell1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
258.                        .addGap(0, 0, Short.MAX_VALUE))
259.                );
260.
261.        getAccessibleContext().setAccessibleDescription("");
262.
263.        pack();
264.    } // </editor-fold>
265.
266.    private void jLabel7MouseClicked(java.awt.event.MouseEvent
        evt) {
267.        this.setVisible(false);
268.        MenuUser menu = new MenuUser();
269.        menu.setVisible(true);
270.        dispose();
271.    }
272.
273.    private void
        selectCompanyActionPerformed(java.awt.event.ActionEvent evt) {
274.        if (selectCompany.getSelectedIndex() > 0) {

```

```

275.         selectState.setEnabled(true);
276.         selectState.setSelectedIndex(0);
277.         switch(selectCompany.getSelectedIndex()){
278.             case 1:
279.                 iconCar.setIcon(nissanWhite);
280.                 break;
281.             case 2:
282.                 iconCar.setIcon(audiWhite);
283.                 break;
284.             case 3:
285.                 iconCar.setIcon(volkswagenWhite);
286.                 break;
287.         }
288.     } else {
289.         clear();
290.     }
291. }
292.
293.     private void
        selectStateActionPerformed(java.awt.event.ActionEvent evt) {
294.         if (selectState.getSelectedIndex() > 0) {
295.             selectModel.setEnabled(true);
296.             selectColors.setEnabled(false);
297.             selectColors.setSelectedIndex(0);
298.             switch(selectCompany.getSelectedIndex()){
299.                 case 1:
300.                     iconCar.setIcon(nissanWhite);
301.                     break;
302.                 case 2:
303.                     iconCar.setIcon(audiWhite);
304.                     break;
305.                 case 3:
306.                     iconCar.setIcon(volkswagenWhite);
307.                     break;
308.             }
309.
            selectModel.setModel(methodsSql.loadModels(selectCompany.getSelectedIndex(), selectState.getSelectedIndex()));
310.        } else {
311.            selectModel.setEnabled(false);
312.            selectColors.setEnabled(false);
313.            selectModel.setSelectedIndex(0);
314.            selectColors.setSelectedIndex(0);
315.        }
316.    }
317.
318.     private void
        selectModelActionPerformed(java.awt.event.ActionEvent evt) {
319.         if (selectModel.getSelectedIndex() > 0) {
320.             selectColors.setEnabled(true);
321.             String txtmodel =
                selectModel.getSelectedItem().toString();
322.             String model = txtmodel.substring(2);
323.
            selectColors.setModel(methodsSql.loadColorsVehicle(model));
324.        } else {
325.            selectColors.setEnabled(false);

```



```

326.         selectColors.setSelectedIndex(0);
327.     }
328. }
329.
330.     private void
331.     selectColorsActionPerformed(java.awt.event.ActionEvent evt) {
332.         String txtcompany =
333.         selectCompany.getSelectedItemAt().toString();
334.         String company = txtcompany.substring(2);
335.         String txtcolor =
336.         selectColors.getSelectedItemAt().toString();
337.         String color = txtcolor.substring(2);
338.         switch(company) {
339.             case "NISSAN":
340.                 switch(color) {
341.                     case "BLANCO":
342.                         iconCar.setIcon(nissanWhite);
343.                         break;
344.                     case "ROJO":
345.                         iconCar.setIcon(nissanRed);
346.                         break;
347.                     case "NARANJA":
348.                         iconCar.setIcon(nissanOrange);
349.                         break;
350.                     case "AMARILLO":
351.                         iconCar.setIcon(nissanYellow);
352.                         break;
353.                     case "VERDE":
354.                         iconCar.setIcon(nissanGreen);
355.                         break;
356.                     case "AZUL":
357.                         iconCar.setIcon(nissanBlue);
358.                         break;
359.                     case "GRIS":
360.                         iconCar.setIcon(nissanGray);
361.                         break;
362.                 }
363.             break;
364.             case "AUDI":
365.                 switch(color) {
366.                     case "BLANCO":
367.                         iconCar.setIcon(audiWhite);
368.                         break;
369.                     case "ROJO":
370.                         iconCar.setIcon(audiRed);
371.                         break;
372.                     case "NARANJA":
373.                         iconCar.setIcon(audiOrange);
374.                         break;
375.                     case "AMARILLO":
376.                         iconCar.setIcon(audiYellow);
377.                         break;
378.                     case "VERDE":
379.                         iconCar.setIcon(audiGreen);
380.                         break;
381.                     case "AZUL":
382.                         iconCar.setIcon(audiBlue);
383.                         break;
384.                 }
385.             break;
386.         }
387.     }
388. }

```

```

380.             break;
381.             case "GRIS":
382.                 iconCar.setIcon(audiGray);
383.                 break;
384.         }
385.         break;
386.     case "VOLKSWAGEN":
387.         switch(color){
388.             case "BLANCO":
389.                 iconCar.setIcon(volkswagenWhite);
390.                 break;
391.             case "ROJO":
392.                 iconCar.setIcon(volkswagenRed);
393.                 break;
394.             case "NARANJA":
395.                 iconCar.setIcon(volkswagenOrange);
396.                 break;
397.             case "AMARILLO":
398.                 iconCar.setIcon(volkswagenYellow);
399.                 break;
400.             case "VERDE":
401.                 iconCar.setIcon(volkswagenGreen);
402.                 break;
403.             case "AZUL":
404.                 iconCar.setIcon(volkswagenBlue);
405.                 break;
406.             case "GRIS":
407.                 iconCar.setIcon(volkswagenGray);
408.                 break;
409.         }
410.         break;
411.     }
412. }
413.
414. private void clear(){
415.     selectState.setEnabled(false);
416.     selectModel.setEnabled(false);
417.     selectColors.setEnabled(false);
418.     selectState.setSelectedIndex(0);
419.     selectModel.setSelectedIndex(0);
420.     selectColors.setSelectedIndex(0);
421.     iconCar.setIcon(car);
422. }
423.
424. /**
425.  * @param args the command line arguments
426.  */
427. public static void main(String args[]) {
428.     /* Set the Nimbus look and feel */
429.     //http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
435.     */

```

```

433.         try {
434.             for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
435.                 if ("Nimbus".equals(info.getName())) {
436.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
437.                     break;
438.                 }
439.             }
440.         } catch (ClassNotFoundException ex) {
441.             java.util.logging.Logger.getLogger(Vehicle.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
442.         } catch (InstantiationException ex) {
443.             java.util.logging.Logger.getLogger(Vehicle.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
444.         } catch (IllegalAccessException ex) {
445.             java.util.logging.Logger.getLogger(Vehicle.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
446.         } catch (javax.swing.UnsupportedLookAndFeelException ex) {
447.             java.util.logging.Logger.getLogger(Vehicle.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
448.         }
449.         //</editor-fold>
450.         //</editor-fold>
451.
452.         /* Create and display the form */
453.         java.awt.EventQueue.invokeLater(new Runnable() {
454.             public void run() {
455.                 new Vehicle().setVisible(true);
456.             }
457.         });
458.     }
459.
460.     // Variables declaration - do not modify
461.     private javax.swing.JLabel iconCar;
462.     private javax.swing.JLabel jLabel1;
463.     private javax.swing.JLabel jLabel2;
464.     private javax.swing.JLabel jLabel3;
465.     private javax.swing.JLabel jLabel4;
466.     private javax.swing.JLabel jLabel5;
467.     private javax.swing.JLabel jLabel7;
468.     private javax.swing.JPanel jPanel1;
469.     private javax.swing.JComboBox<String> selectColors;
470.     private javax.swing.JComboBox<String> selectCompany;
471.     private javax.swing.JComboBox<String> selectModel;
472.     private javax.swing.JComboBox<String> selectState;
473.     // End of variables declaration
474. }

```