



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>PRÁCTICA DE LABORATORIO</b>	
<b>CARRERA:</b> COMPUTACION		<b>ASIGNATURA:</b> PROGRAMACION APLICADA	
<b>NRO. PRÁCTICA:</b>	1.1	<b>TÍTULO PRÁCTICA:</b> TÍTULO PROYECTO: Proyecto Integrador Inter ciclo. Desarrollo e implementación de un sistema de gestión de datos del parqueadero de la empresa EMOV-EC.	
<b>OBJETIVO ALCANZADO:</b> Reforzar los conocimientos adquiridos en clase sobre la programación aplicada (POO, Interfaz gráfica, etc) en un contexto real.			
<b>ACTIVIDADES DESARROLLADAS</b>			
<ul style="list-style-type: none"> <li>Investigue, diseñe y desarrolle e implemente un sistema informático que permita gestionar los espacios de parqueo de la empresa EMOV-EC del cancho del parque de la Madre que tiene una capacidad de más de 50 estacionamientos.</li> </ul> <p>Link GitHub:  <a href="https://github.com/adolfojara10/Practicas-de-programacion-aplicada">https://github.com/adolfojara10/Practicas-de-programacion-aplicada</a> </p> <p>Link página web:  <a href="https://sites.google.com/view/proyecto-parqueadero-jara/inicio">https://sites.google.com/view/proyecto-parqueadero-jara/inicio</a> </p> <p>Librerías extras utilizadas:</p>  <p>La primera librería nos sirve para poder seleccionar una fecha, y las demás son para encriptar las contraseñas de los administradores.</p> <p>Para la solución de este problema se planteó la siguiente resolución:          Primero, se generó el diagrama UML. Éste será subido al repositorio de github para una mejor comprensión de este, ya que la imagen es muy grande para un documento de Word. En este diagrama se planteó las siguientes clases del paquete modelo:</p> <ul style="list-style-type: none"> <li>Administrador: éste es el encargado de poder gestionar la aplicación del parqueadero. Existen 2 tipos: Completo y Simple. El completo puede crear administradores simples, pero lo simples no.</li> </ul>			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Esto se los diferencié con un Enum Class. Además, esta clase hereda los atributos de la clase padre Persona.

- Cliente: está sirve para crear un cliente que entra al parqueadero. También para la factura, contrato.
- Contrato: está sirve para cuando un cliente quiere ya tener reservado por un tiempo definido un parqueadero. Aquí se guardan los datos de un cliente, un sitio de parqueadero, fecha de inicio y de fin, el subtotal, total, si se aplica descuento o no, y si paga por adelantado el contrato. Este contrato guarda una variable de tipo Enum Class para saber si se cobra por día, semana o mes.
- Factura: esta clase nos sirve para emitir facturas para clientes que solamente entran al parqueadero ese día y salen el mismo día. Esto nos sirve para tener por separado a clientes por contrato y a clientes ocasionales. Al igual que en el contrato, esta cuenta con un cliente, un sitio, la fecha de entrada y salida, el tiempo que estuvo parqueado, el subtotal, el total, si se aplica descuento o no.
- Persona: esta clase fue creada para heredar tanto a la clase cliente y administrador. Esto nos sirve para reutilizar código.
- Sitio: esta clase representa como es un espacio en un parqueadero, esta cuenta con código, un Enum Class que nos sirve para cambiarle de estados al sitio entre ocupado, desocupado y contratado, y, por último, un precio que se cobra cada intervalo de 10 minutos.

Para el paquete controlador se generaron las siguientes clases:

- Controlador: esta clase es abstracta. Hereda a los demás controladores del paquete. Aquí se generan los métodos CRUD y el get de la lista en la que se guardan los objetos.
- ControladorAdministrador: esta hereda del Controlador, y además tiene 5 métodos propios: iniciarSesion() el cual nos sirve para leer los datos que alguien intenta iniciar sesión en la aplicación y devolver el usuario encontrado; encriptarPassword() nos sirve para encriptar la contraseña de un usuario y que así no sea visible; y por último desencriptarPassword() nos sirve para desencriptar una contraseña de algún usuario, generarCodigo() para generar el ID del nuevo administrador, y validarEmail() que nos sirve para comprobar una dirección de correo electrónico sea la correcta. Nos sirve para manipular los objetos de tipo Administrador.
- ControladorCliente: esta clase contiene el método generarCodigo() para cargar el código del cliente. Nos sirve para manipular los objetos de tipo Cliente.
- ControladorContrato: esta clase tiene 3 métodos propios: obtenerSubtotal() que nos sirve para calcular el subtotal de un contrato, obtenerTotal() que nos sirve para calcular el total de un contrato, generarCodigo() para cargar el código del contrato. Nos sirve para manipular los objetos de tipo Contrato.
- ControladorFactura: esta clase cuenta con 4 métodos propios: calcularSubtotal() que nos sirve para calcular el subtotal de una factura; calcularTiempo() que nos sirve para calcular el tiempo que una persona usó un sitio del parqueadero; y por último calcularTotal() que nos sirve para calcular el valor total a pagar, generarCodigo() para cargar el código del contrato. Nos sirve para manipular los objetos de tipo Factura.
- ControladorSitio: esta clase cuenta con 5 métodos: cargarSitios() que nos sirve para crear los sitios solamente al inicio del programa; encontrarSitioDesocupado() que nos sirve para encontrar un sitio desocupado y asignarle a un cliente; cambiarAEstadoOcupado() que nos sirve para cambiar el Enum Class de un sitio del parqueadero a ocupado; cambiarAEstadoDesocupado() que nos sirve para cambiar el Enum Class de un sitio del parqueadero a desocupado; cambiarAEstadoContratado() que nos sirve para cambiar el Enum Class de un sitio del parqueadero a contratado. Nos sirve para manipular los objetos de tipo Sitio.
- ControladorVehículo: cuenta con el método generarCodigo() para cargar el código del contrato pero que nos sirve para manejar a los vehículos.

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Luego continuamos con el paquete vista, en este se crearon las siguientes ventanas:

- VentanaPrincipal: es la ventana con la que se inicializa el proyecto y que guarda a las demás ventanas.
- VentanaContrato: es la ventana encargada de mostrar los elementos de un contrato y además de mostrar los registros de todos los contratos.
- VentanaCrearCliente: es la ventana encargada de los clientes, se puede realizar todos los métodos crud sobre él.
- VentanaCrearUsuario: sirve para manipular a los administradores de tipo Enum Class simple.
- VentanaEntradaTicket: sirve para emitir un ticket cuando un cliente ingresa al parqueadero.
- VentanaGestionFacturas: en esta ventana solo sirve para poder observar todas las facturas emitidas.
- VentanaLogIn: sirve para que un administrador pueda iniciar sesión.
- VentanaSalidaTicket: sirve para emitir una factura cuando un cliente está saliendo del parqueadero.
- VentanaVehiculo: sirve para crear un vehículo y todo el crud sobre él.
- VentanaVisualizarFactura: sirve para visualizar los datos de la factura en un Jlist. esta factura también tiene la funcionalidad de guardar una factura para que posteriormente sea impresa.

- **Diagramas de Clases.**



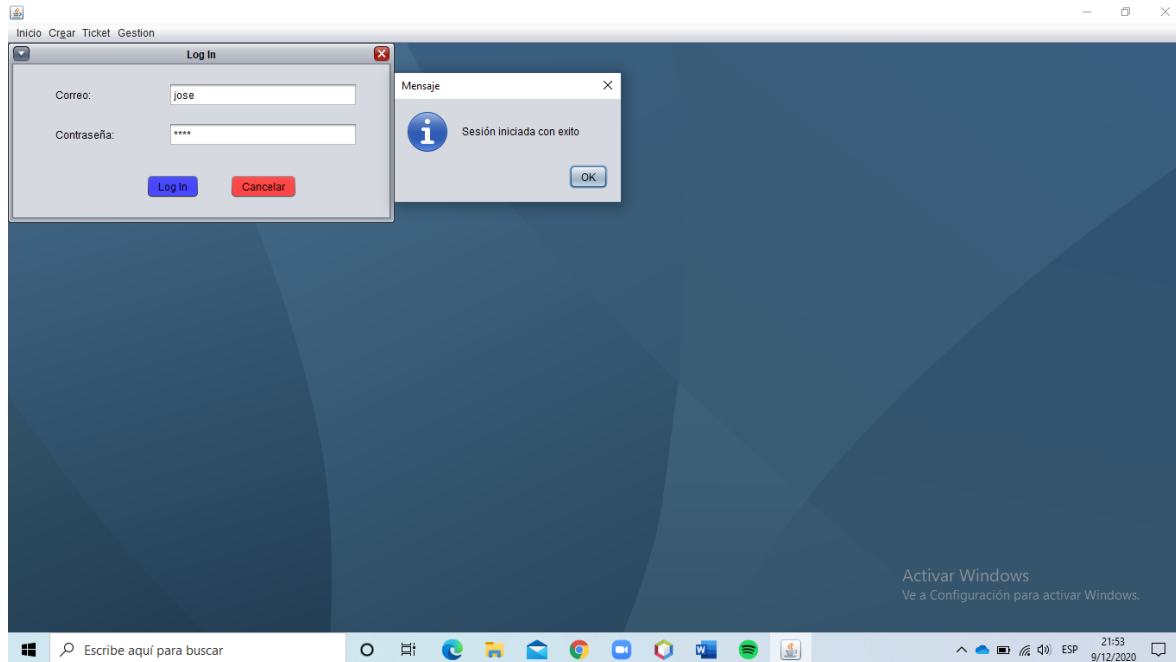
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

## Presentación del sistema:

La ventana principal:

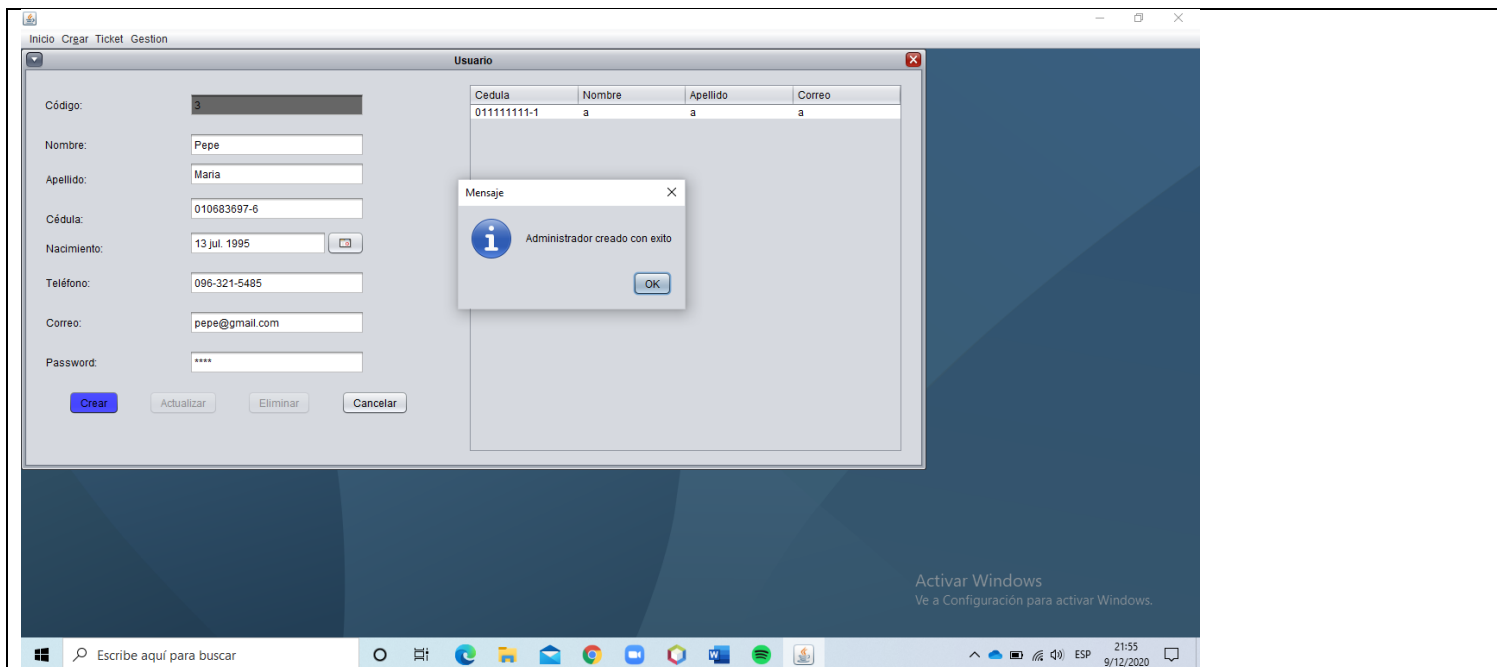


Para iniciar sesión:

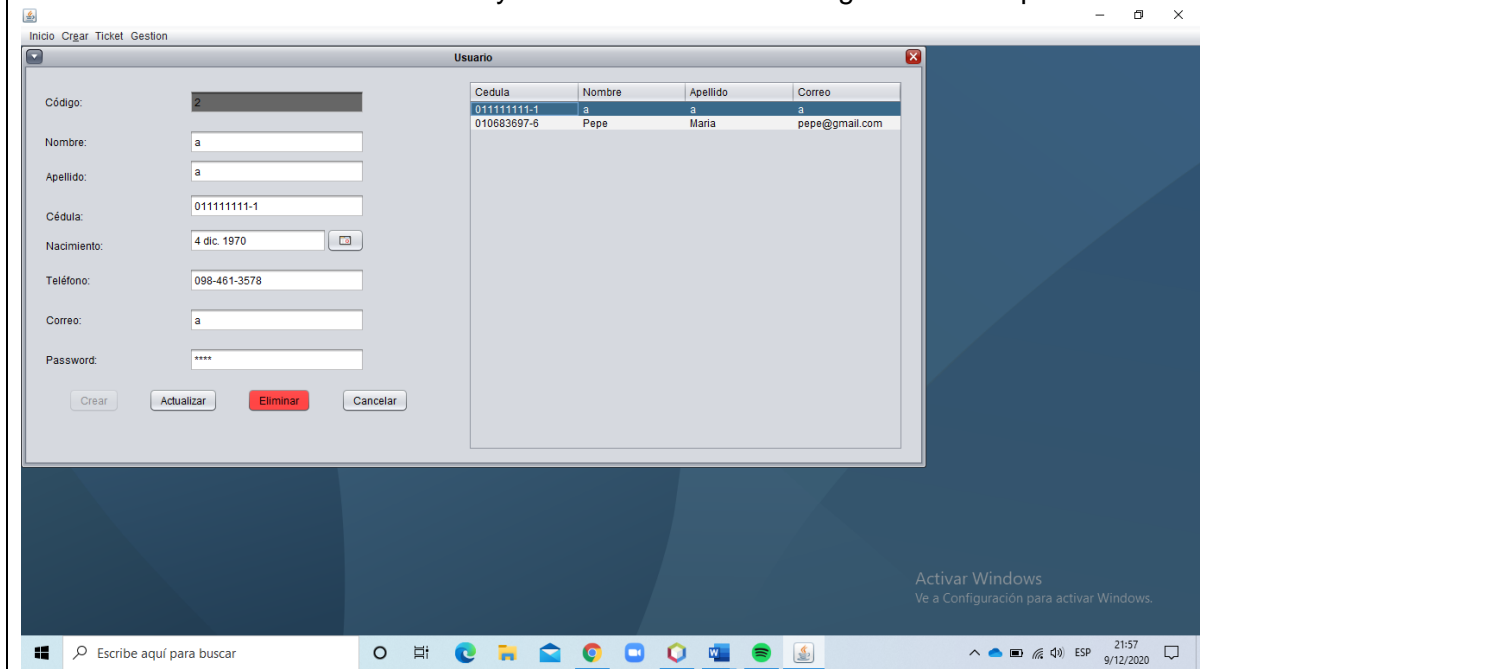


Se inició sesión con el administrador principal, éste es el único que puede crear a los demás administradores, pero estos no pueden ingresar a dicha ventana. A continuación, se mostrará la ventana para crear administradores:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Una vez creado un administrador, procederemos a editar el otro administrador que está en la tabla, para eso damos click en el administrador y automáticamente nos cagan los datos para editarlos:



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Inicio Cargar Ticket Gestion

Usuario

Código: 2

Nombre: Sebastian

Apellido: Lopez

Cédula: 011111111-1

Nacimiento: 4 dic. 1970

Teléfono: 095-148-5365

Correo: sebastian@gmail.com

Password: \*\*\*\*

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
011111111-1	a	a	a
010683697-6	Pepe	Maria	pepe@gmail.com

Mensaje

Administrador actualizado con éxito

OK

Activar Windows  
Ve a Configuración para activar Windows.


Escribe aquí para buscar

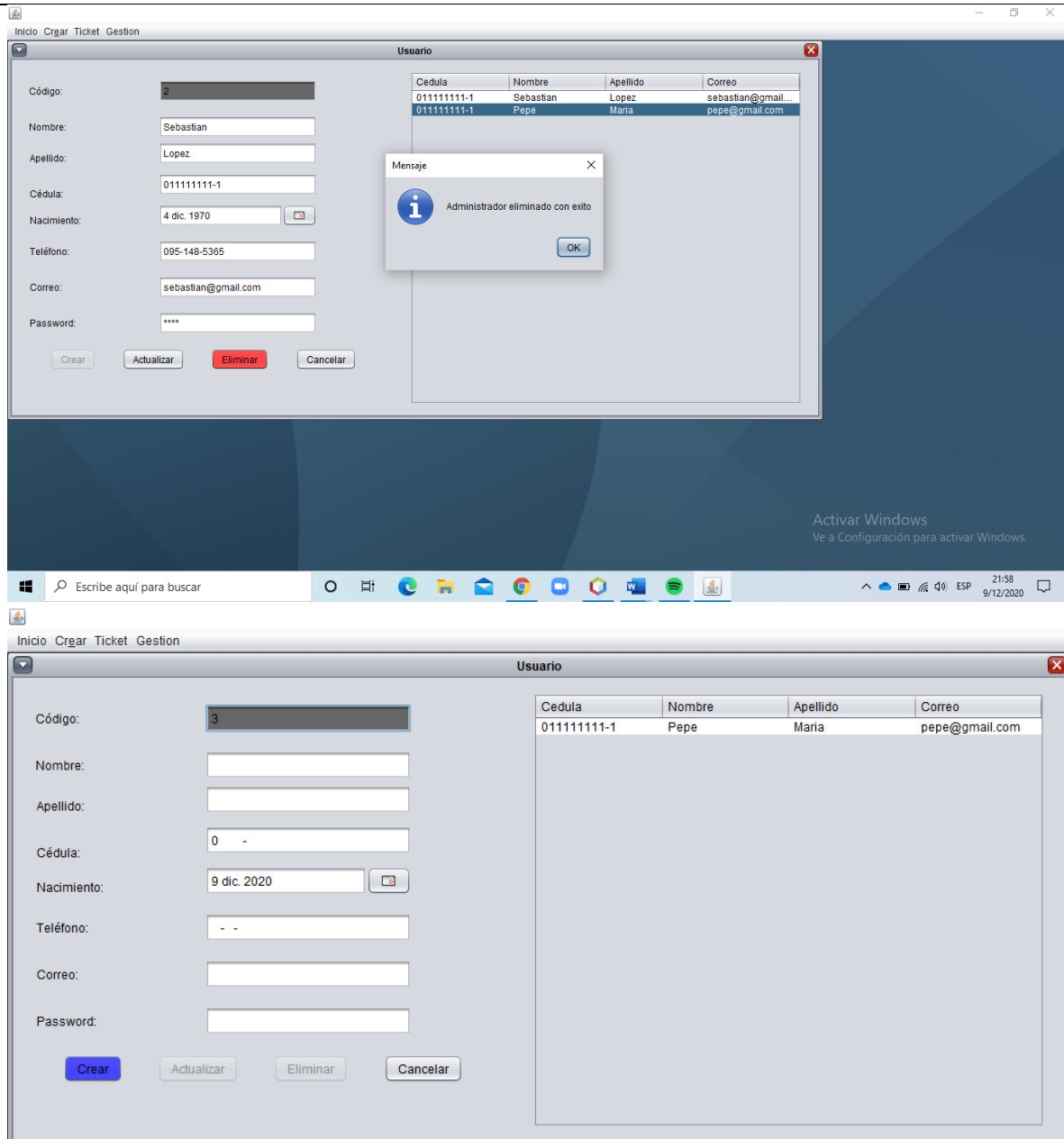
21:57  
9/12/2020

Usuario

Cedula	Nombre	Apellido	Correo
011111111-1	Sebastian	Lopez	sebastian@gmail....
011111111-1	Pepe	Maria	pepe@gmail.com

Ahora eliminaremos al segundo administrador, para esto se da click en el administrador en la tabla y presionamos el botón eliminar:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Inicio Crgar Ticket Gestion

Usuario

Código: 2

Nombre: Sebastian

Apellido: Lopez

Cédula: 011111111-1

Nacimiento: 4 dic. 1970

Teléfono: 095-148-5365

Correo: sebastian@gmail.com

Password: \*\*\*\*

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
011111111-1	Sebastian	Lopez	sebastian@gmail.com
011111111-1	Pepe	Maria	pepe@gmail.com

Mensaje

Administrador eliminado con éxito

OK

Activar Windows  
Ve a Configuración para activar Windows.

21:58  
9/12/2020

Inicio Crgar Ticket Gestion

Usuario

Código: 3

Nombre:

Apellido:

Cédula: 0 -

Nacimiento: 9 dic. 2020

Teléfono: - -

Correo:

Password:

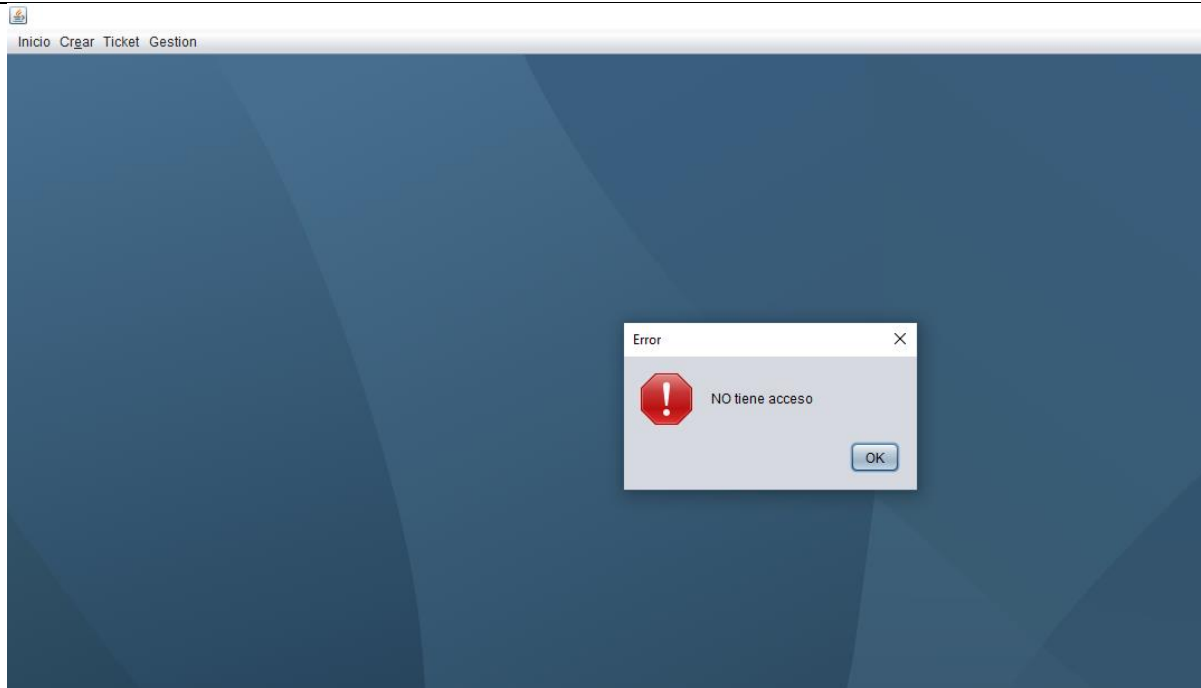
Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
011111111-1	Pepe	Maria	pepe@gmail.com

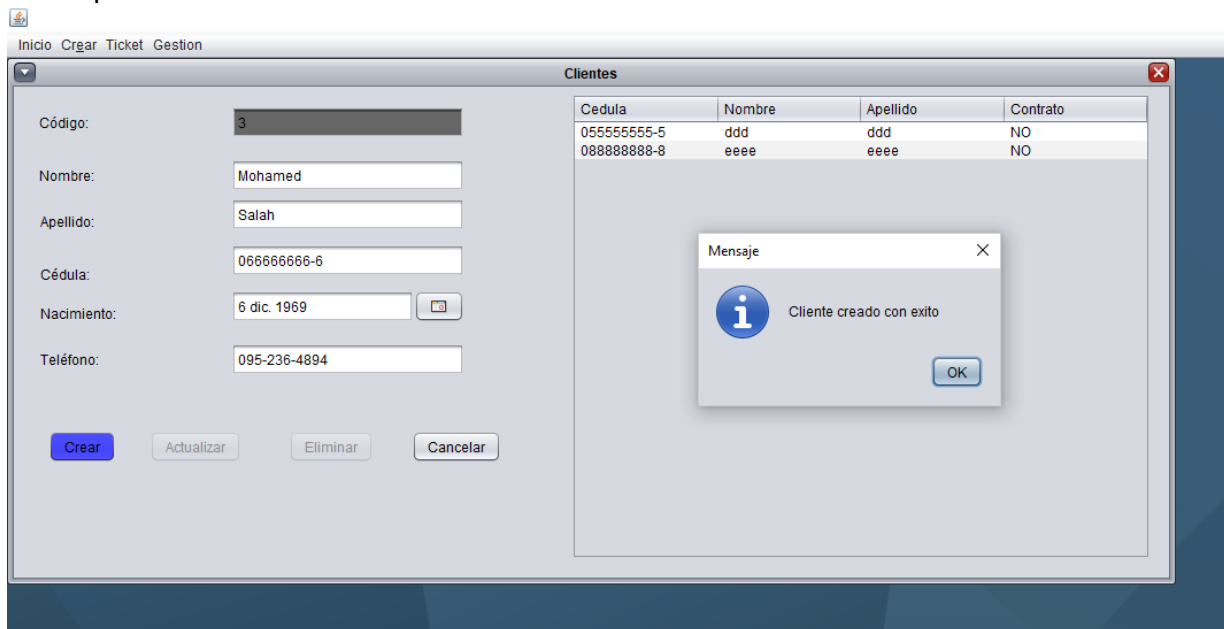
Ahora procedemos a iniciar sesión con un administrador simple e intentamos abrir la misma ventana y nos aparecerá este mensaje:



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Ahora procedemos a crear un Cliente:



Ahora para editar un cliente, se siguen los mismos que con la ventana del administrador:

Inicio Crgar Ticket Gestion

**Cientes**

Código: 2

Nombre: Esteban

Apellido: Rooney

Cédula: 088888888-8

Nacimiento: 5 dic. 1987

Teléfono: 088-949-8494

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Contrato
055555555-5	ddd	ddd	NO
088888888-8	eeee	eeee	NO
066666666-6	Mohamed	Salah	NO

Mensaje

Cliente actualizado con éxito

OK

Inicio Crgar Ticket Gestion

**Cientes**

Código: 4

Nombre:

Apellido:

Cédula: 0 -

Nacimiento: 9 dic. 2020

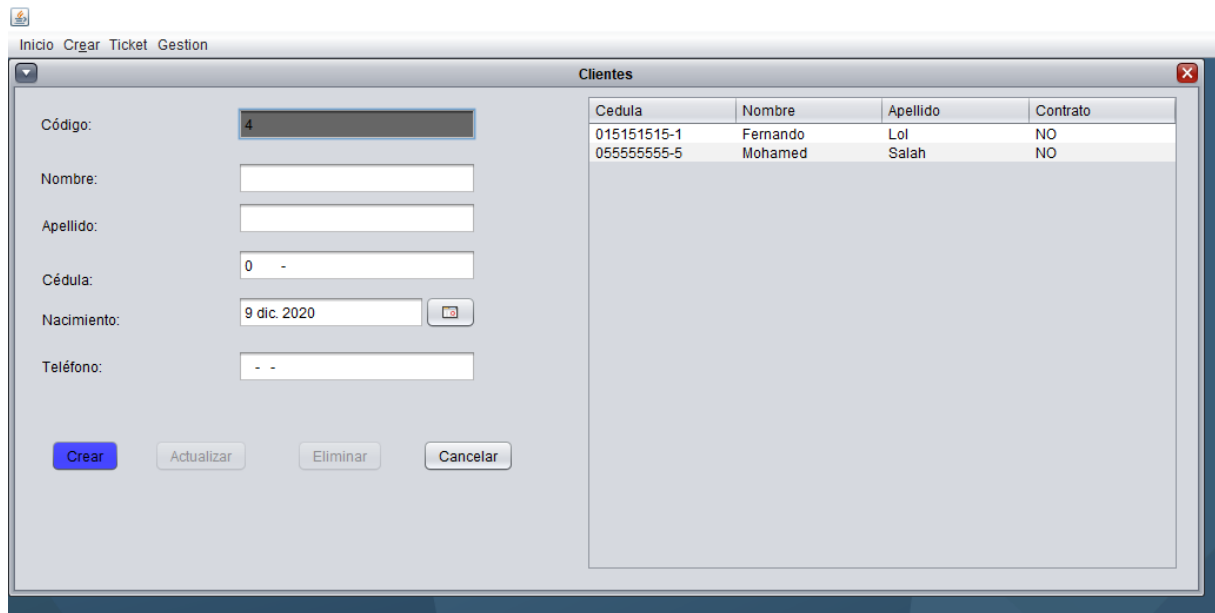
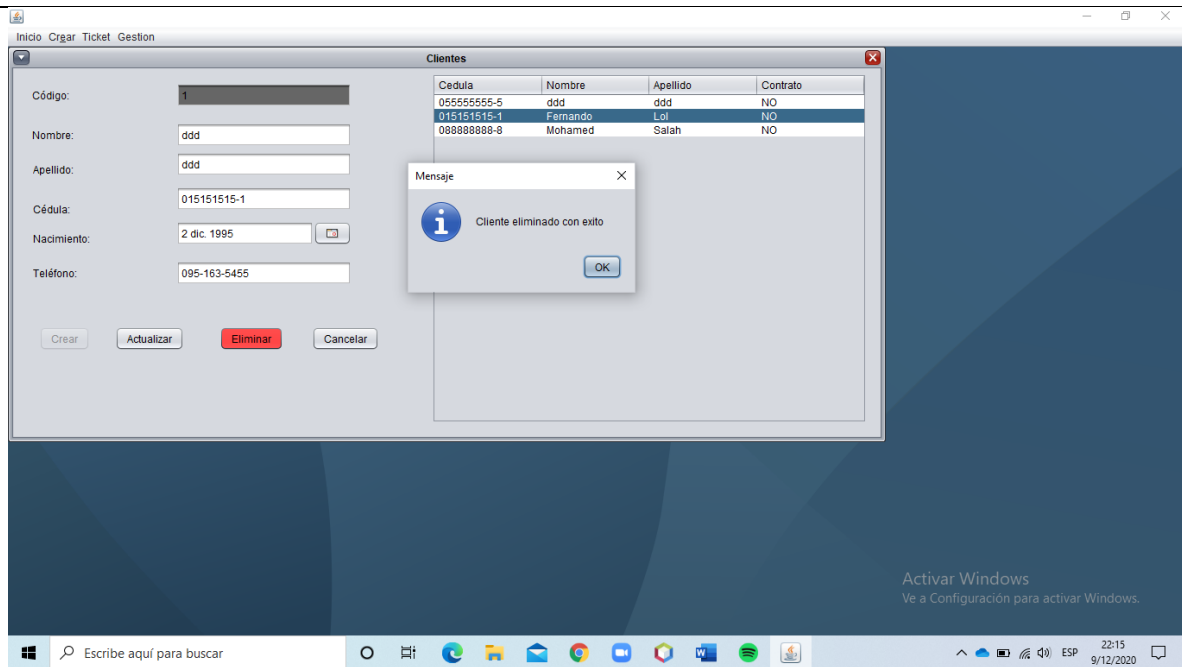
Teléfono: - -

Crear Actualizar Eliminar Cancelar

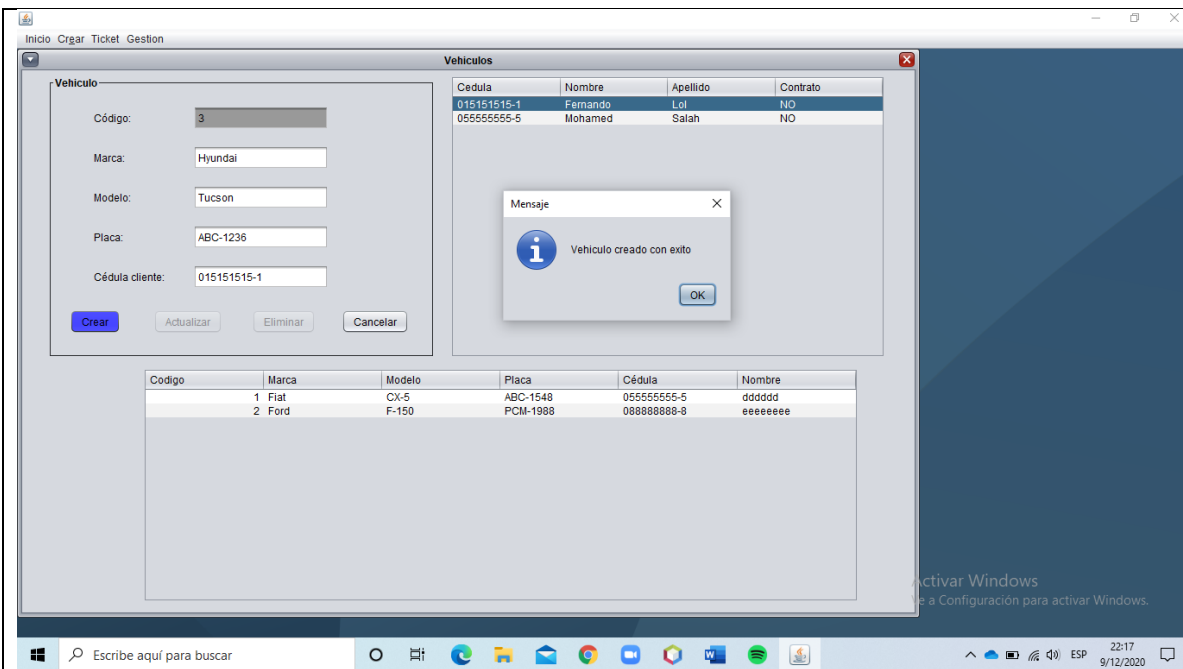
Cedula	Nombre	Apellido	Contrato
055555555-5	ddd	ddd	NO
088888888-8	Esteban	Rooney	NO
088888888-8	Mohamed	Salah	NO

Ahora eliminaremos un cliente, y se siguen los mismos pasos que con el administrador:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



A continuación, se creará un vehículo, para esto ponemos las características del vehículo y seleccionamos un cliente de la tabla de clientes:

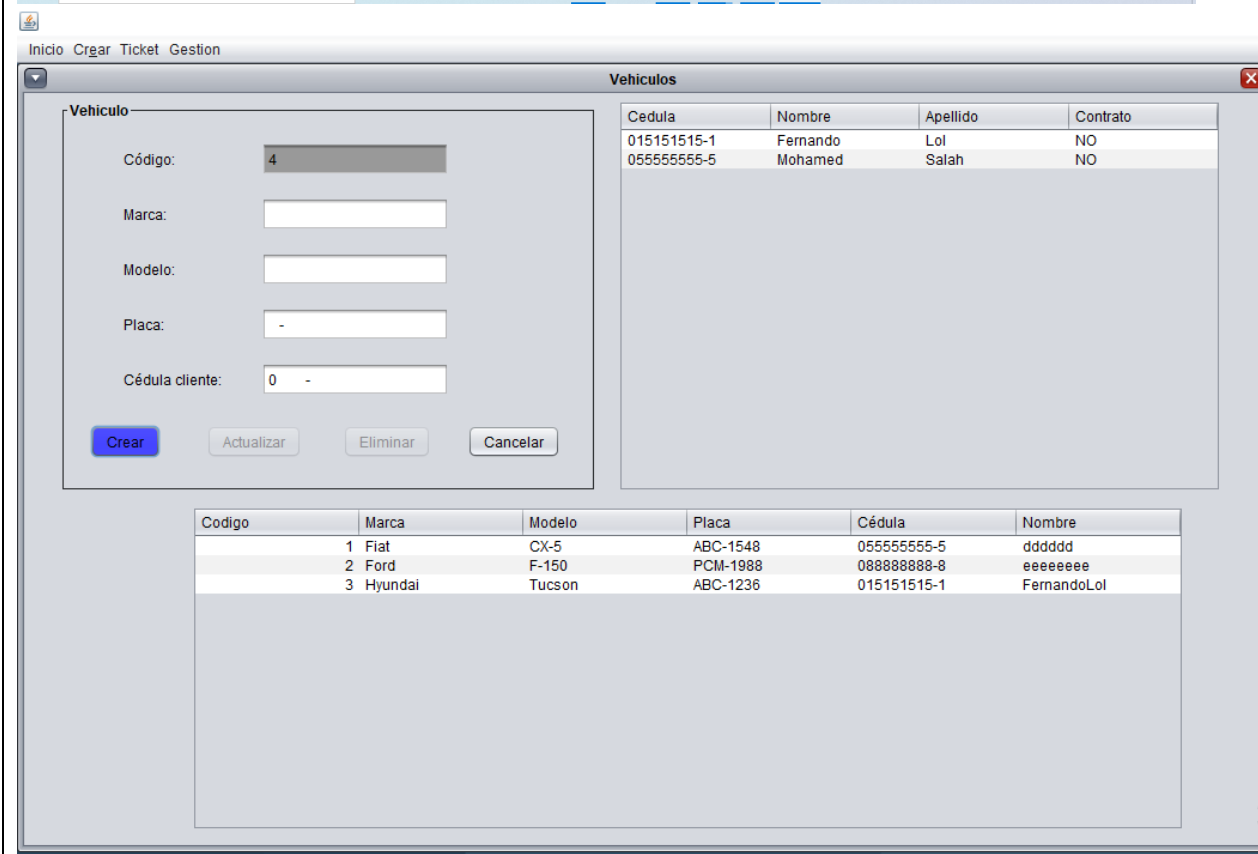


The screenshot shows the 'Vehiculos' application window. On the left, the 'Vehiculo' form has the following values: Código: 3, Marca: Hyundai, Modelo: Tucson, Placa: ABC-1236, and Cédula cliente: 01515151-1. The 'Crear' button is highlighted. In the center, a 'Mensaje' dialog box displays an information icon and the text 'Vehiculo creado con exito' with an 'OK' button. On the right, a table lists existing vehicles:

Cedula	Nombre	Apellido	Contrato
01515151-1	Fernando	Lol	NO
05555555-5	Mohamed	Salah	NO

Below the table, a summary table is visible:

Codigo	Marca	Modelo	Placa	Cédula	Nombre
1	Fiat	CX-5	ABC-1548	05555555-5	dddddd
2	Ford	F-150	PCM-1988	08888888-8	eeeeeeee



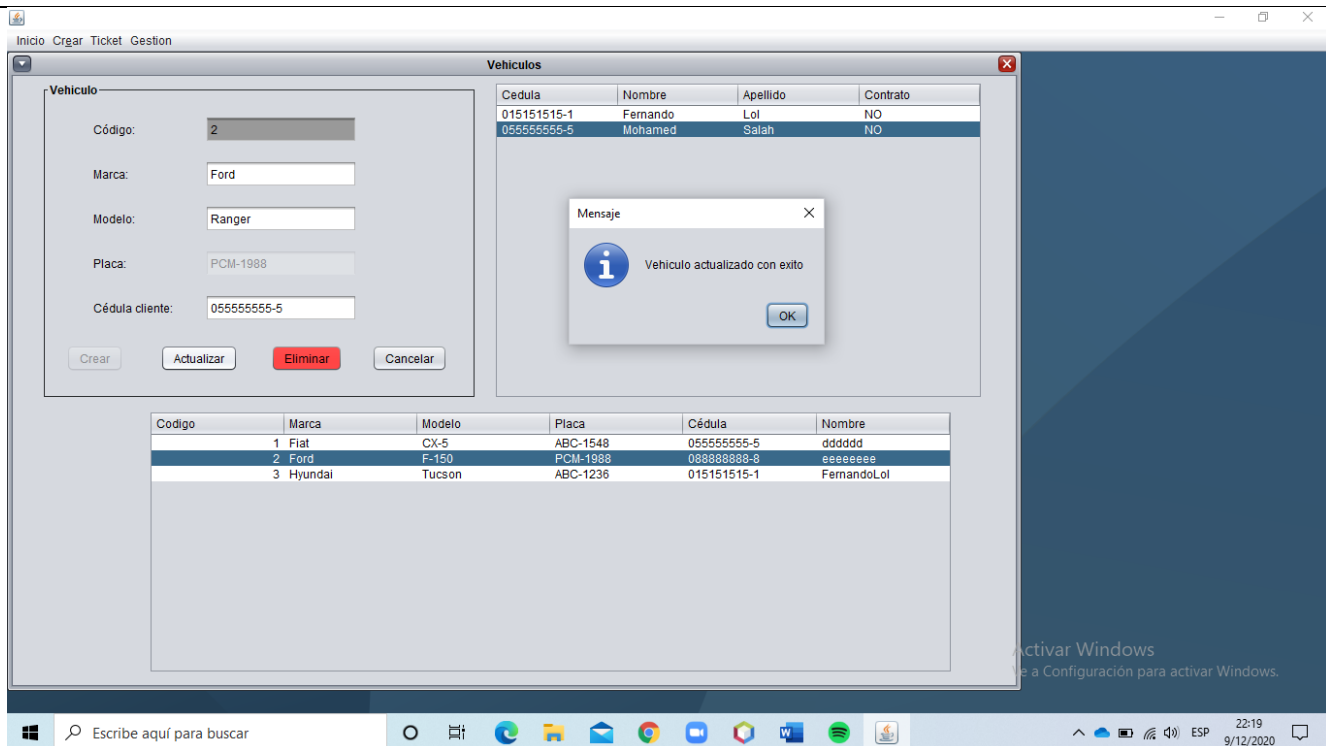
The screenshot shows the 'Vehiculos' application window after an update. The 'Vehiculo' form on the left now has: Código: 4, Marca: (empty), Modelo: (empty), Placa: -, and Cédula cliente: 0 -. The 'Actualizar' button is highlighted. The table on the right now includes the new vehicle:

Cedula	Nombre	Apellido	Contrato
01515151-1	Fernando	Lol	NO
05555555-5	Mohamed	Salah	NO

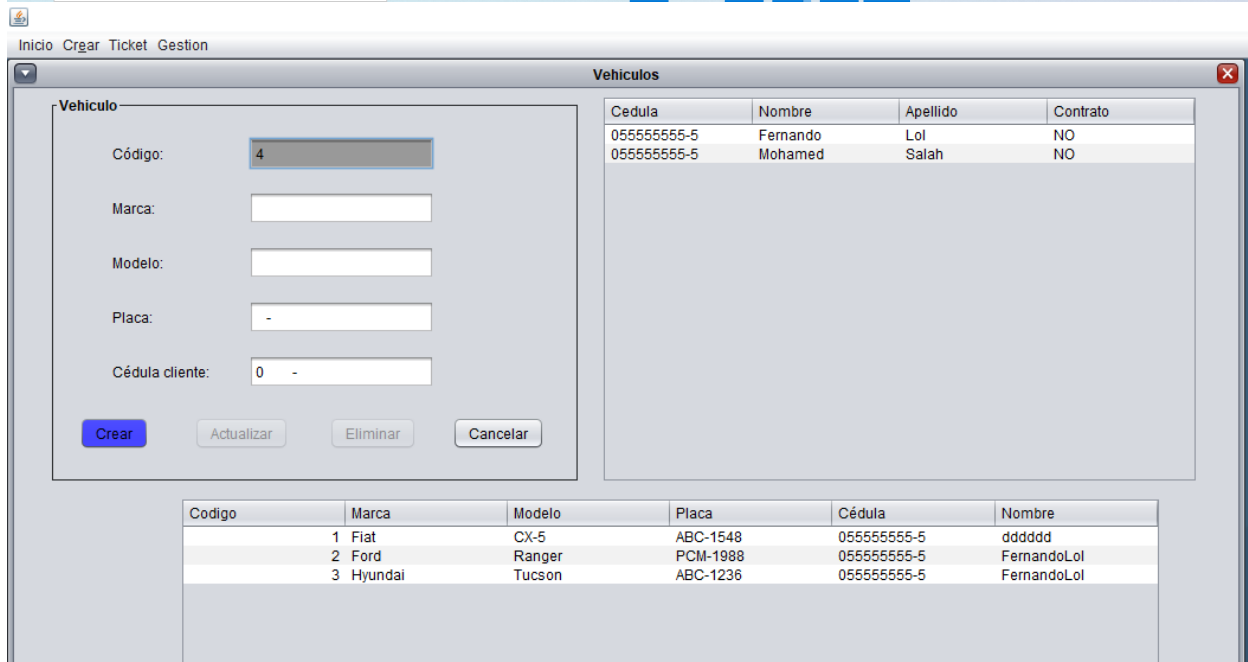
The summary table at the bottom is also updated:

Codigo	Marca	Modelo	Placa	Cédula	Nombre
1	Fiat	CX-5	ABC-1548	05555555-5	dddddd
2	Ford	F-150	PCM-1988	08888888-8	eeeeeeee
3	Hyundai	Tucson	ABC-1236	01515151-1	FernandoLol

Ahora actualizaremos un vehículo para esto hacemos los mismos que con el cliente:



The screenshot shows the 'Vehiculos' application window. On the left, there is a form for adding or updating a vehicle. The 'Código' field contains the value '2'. The 'Marca' field contains 'Ford', 'Modelo' contains 'Ranger', 'Placa' contains 'PCM-1988', and 'Cédula cliente' contains '05555555-5'. Below the form are buttons for 'Crear', 'Actualizar', 'Eliminar', and 'Cancelar'. On the right, there is a table with columns 'Cedula', 'Nombre', 'Apellido', and 'Contrato'. It contains two rows: '015151515-1' with 'Fernando' and 'Lol', and '055555555-5' with 'Mohamed' and 'Salah'. A modal dialog box titled 'Mensaje' is displayed in the center, showing an information icon and the text 'Vehiculo actualizado con exito', with an 'OK' button. At the bottom of the window, there is another table with columns 'Codigo', 'Marca', 'Modelo', 'Placa', 'Cédula', and 'Nombre', containing three rows of vehicle data.



The screenshot shows the 'Vehiculos' application window. The 'Código' field now contains the value '4'. The 'Marca' field is empty, 'Modelo' is empty, 'Placa' contains a hyphen '-', and 'Cédula cliente' contains '0 -'. The 'Crear' button is highlighted in blue. The table on the right now contains three rows: '055555555-5' with 'Fernando' and 'Lol', '055555555-5' with 'Mohamed' and 'Salah', and '055555555-5' with 'Fernando' and 'Lol'. The bottom table remains the same as in the previous screenshot.

Por último, eliminaremos un vehículo siguiendo los mismos pasos:

Inicio Crgar Ticket Gestion

**Vehiculo**

Código:

Marca:

Modelo:


Placa:

Cédula cliente:

**Vehiculos**

Cedula	Nombre	Apellido	Contrato
05555555-5	Fernando	Lol	NO
05555555-5	Mohamed	Salah	NO

Mensaje

 Vehiculo eliminado con exito

Codigo	Marca	Modelo	Placa	Cédula	Nombre
1	Fiat	CX-5	ABC-1548	05555555-5	ddddd
2	Ford	Ranger	PCM-1988	05555555-5	FernandoLol
3	Hyundai	Tucson	ABC-1236	05555555-5	FernandoLol

Activar Windows  
Ve a Configuración para activar Windows.

Escribe aquí para buscar

22:20  
9/12/2020

Inicio Crgar Ticket Gestion

**Vehiculo**

Código:

Marca:

Modelo:

Placa:

Cédula cliente:

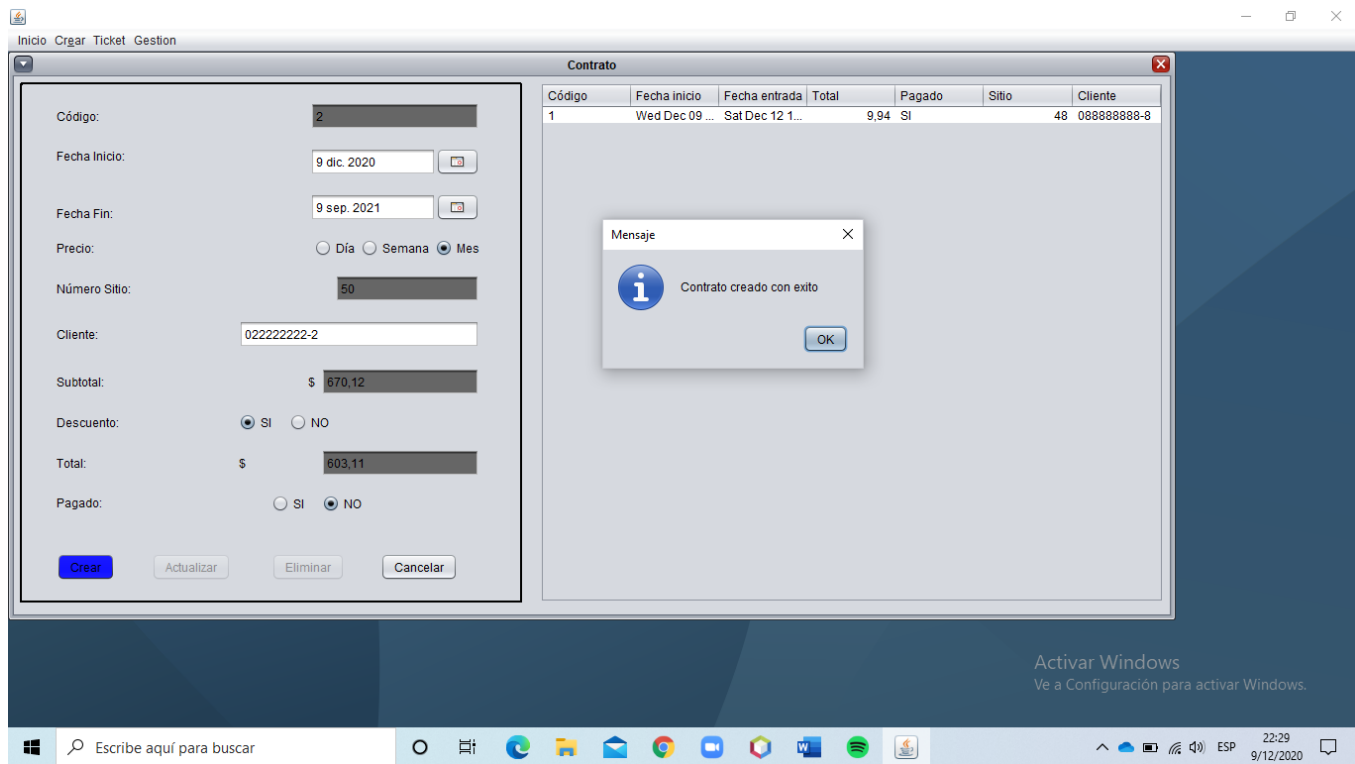
**Vehiculos**

Cedula	Nombre	Apellido	Contrato
05555555-5	Fernando	Lol	NO
05555555-5	Mohamed	Salah	NO

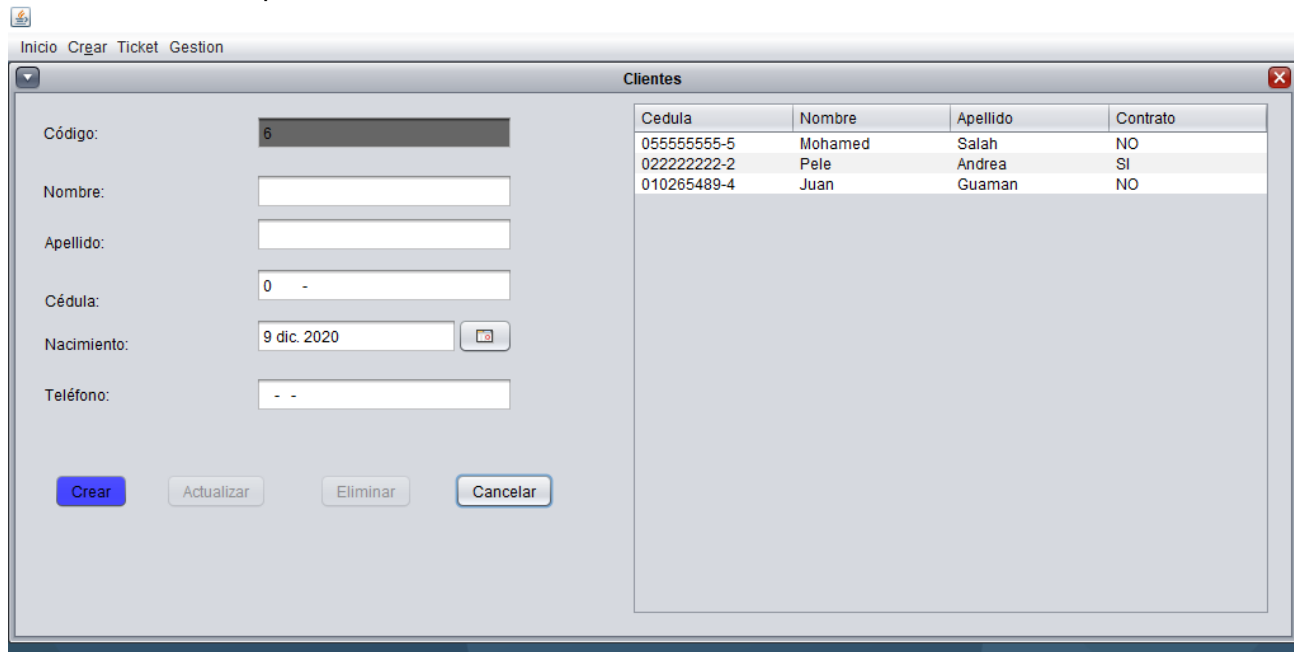
Codigo	Marca	Modelo	Placa	Cédula	Nombre
2	Ford	Ranger	ABC-1548	05555555-5	FernandoLol
3	Hyundai	Tucson	ABC-1236	05555555-5	FernandoLol

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Ahora crearemos un contrato:



Para crear un contrato necesitamos llenar los siguientes datos como se muestra en la imagen. Siempre se da un descuento de 10% si escogen la opción descuento. El tipo de precio dependerá de la duración del contrato. La cedula del cliente se llena y automáticamente se cambiará el estado del cliente ya que tendrá un contrato, para esto lo mostraremos en la ventana del cliente:

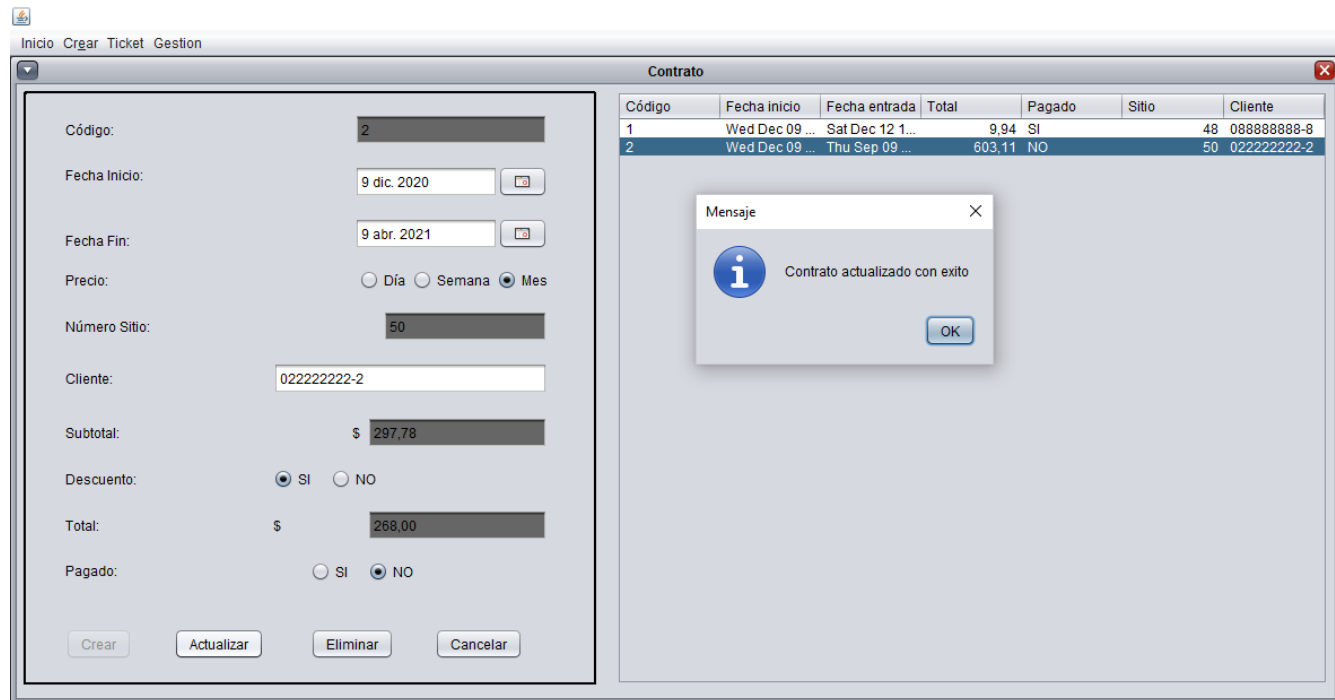


Cedula	Nombre	Apellido	Contrato
055555555-5	Mohamed	Salah	NO
022222222-2	Pele	Andrea	SI
010265489-4	Juan	Guaman	NO

Como podemos observar cambio su estado.

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Ahora actualizaremos un contrato, se siguen los mismos que con el administrador y el cliente y el vehículo:



The screenshot shows the 'Contrato' form with the following data:

- Código: 2
- Fecha Inicio: 9 dic. 2020
- Fecha Fin: 9 abr. 2021
- Precio: ☐ Día ☐ Semana ☒ Mes
- Número Sitio: 50
- Cliente: 02222222-2
- Subtotal: \$ 297,78
- Descuento: ☒ SI ☐ NO
- Total: \$ 268,00
- Pagado: ☐ SI ☒ NO

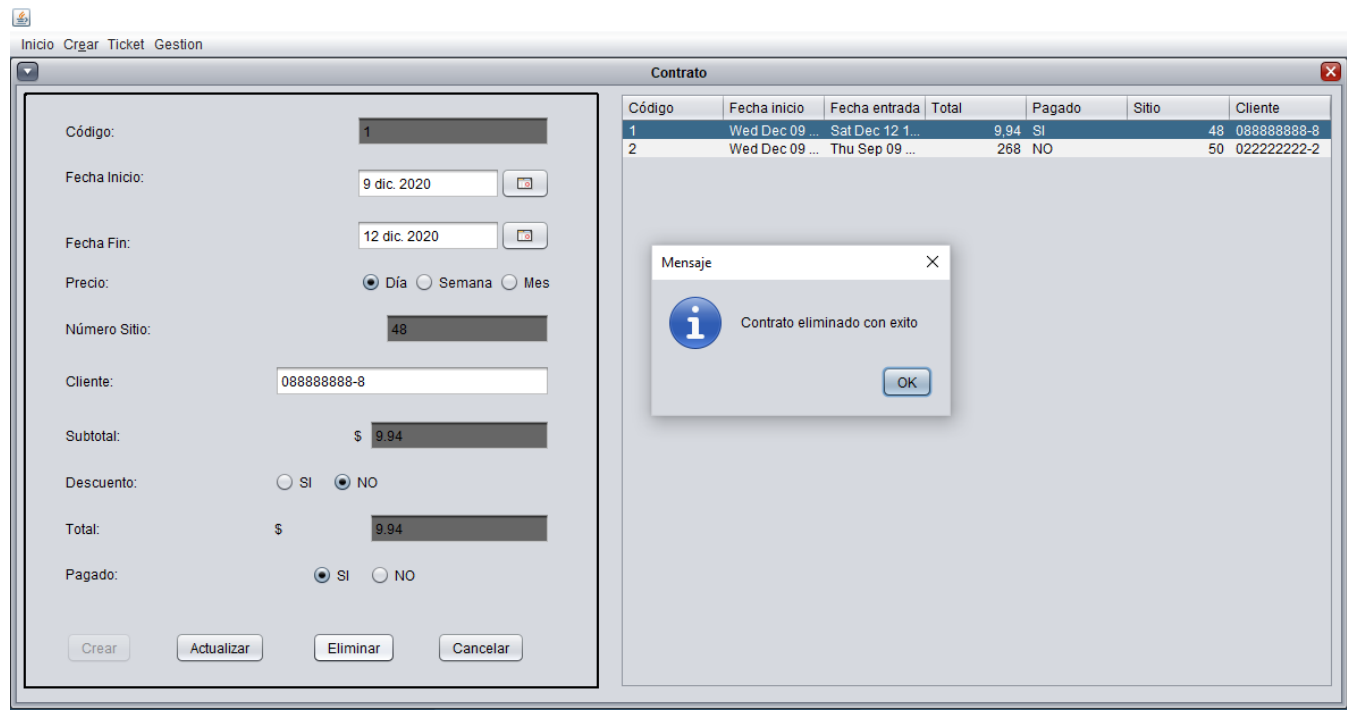
Buttons: Crear, Actualizar, Eliminar, Cancelar.

Table 'Contrato' (visible in the background):

Código	Fecha inicio	Fecha entrada	Total	Pagado	Sitio	Cliente
1	Wed Dec 09 ...	Sat Dec 12 1...	9,94	SI	48	088888888-8
2	Wed Dec 09 ...	Thu Sep 09 ...	603,11	NO	50	022222222-2

Message box: Mensaje. Contrato actualizado con éxito. OK

Ahora eliminamos un contrato:



The screenshot shows the 'Contrato' form with the following data:

- Código: 1
- Fecha Inicio: 9 dic. 2020
- Fecha Fin: 12 dic. 2020
- Precio: ☒ Día ☐ Semana ☐ Mes
- Número Sitio: 48
- Cliente: 088888888-8
- Subtotal: \$ 9,94
- Descuento: ☐ SI ☒ NO
- Total: \$ 9,94
- Pagado: ☒ SI ☐ NO

Buttons: Crear, Actualizar, Eliminar, Cancelar.

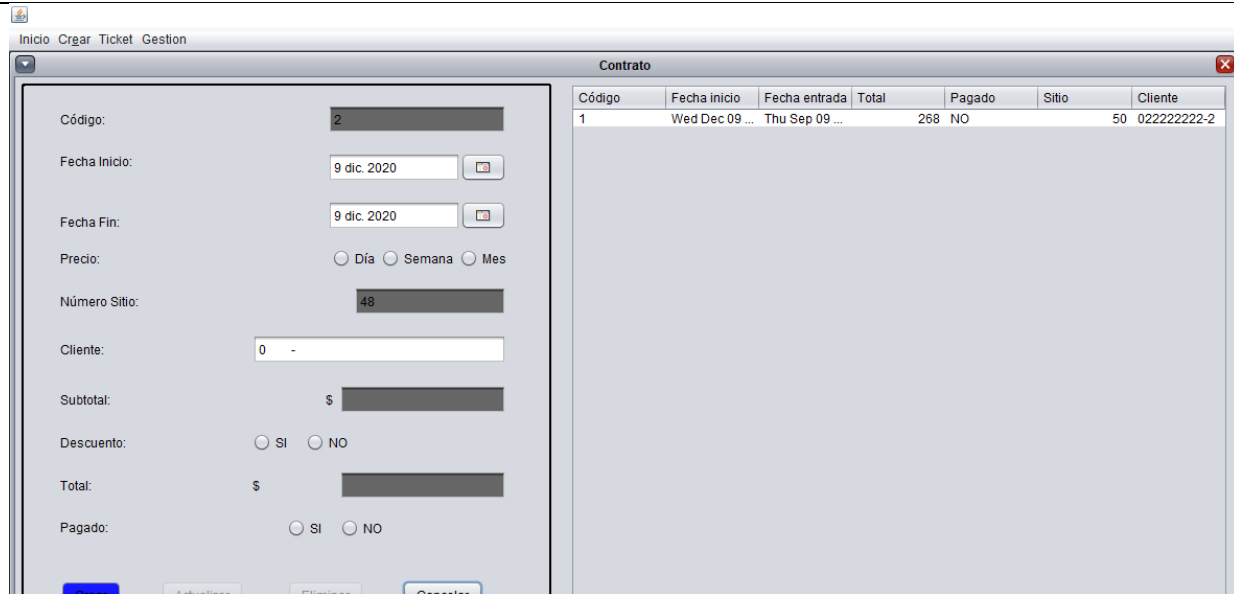
Table 'Contrato' (visible in the background):

Código	Fecha inicio	Fecha entrada	Total	Pagado	Sitio	Cliente
1	Wed Dec 09 ...	Sat Dec 12 1...	9,94	SI	48	088888888-8
2	Wed Dec 09 ...	Thu Sep 09 ...	268	NO	50	022222222-2

Message box: Mensaje. Contrato eliminado con éxito. OK

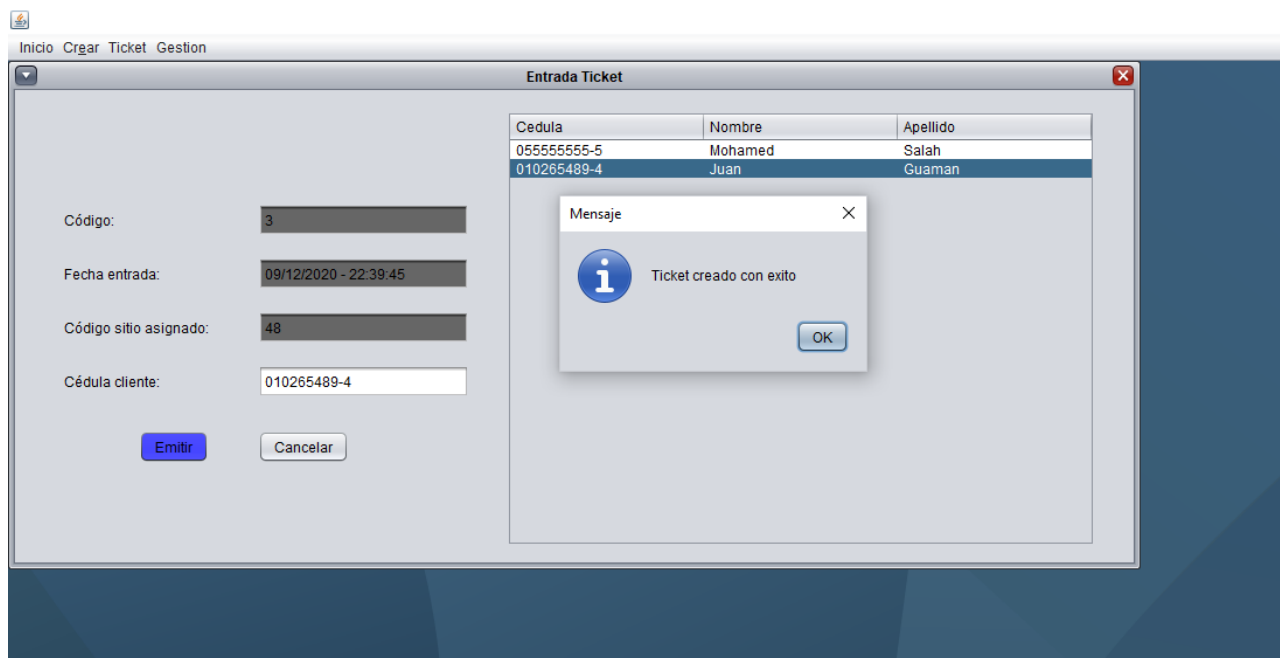


	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	<b>CONSEJO ACADÉMICO</b>	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Código	Fecha inicio	Fecha entrada	Total	Pagado	Sitio	Cliente
1	Wed Dec 09 ...	Thu Sep 09 ...	268	NO	50	02222222-2

Ahora pasaremos con la emisión de un ticket de entrada. Para esto solamente es necesario llenar la cedula del cliente:

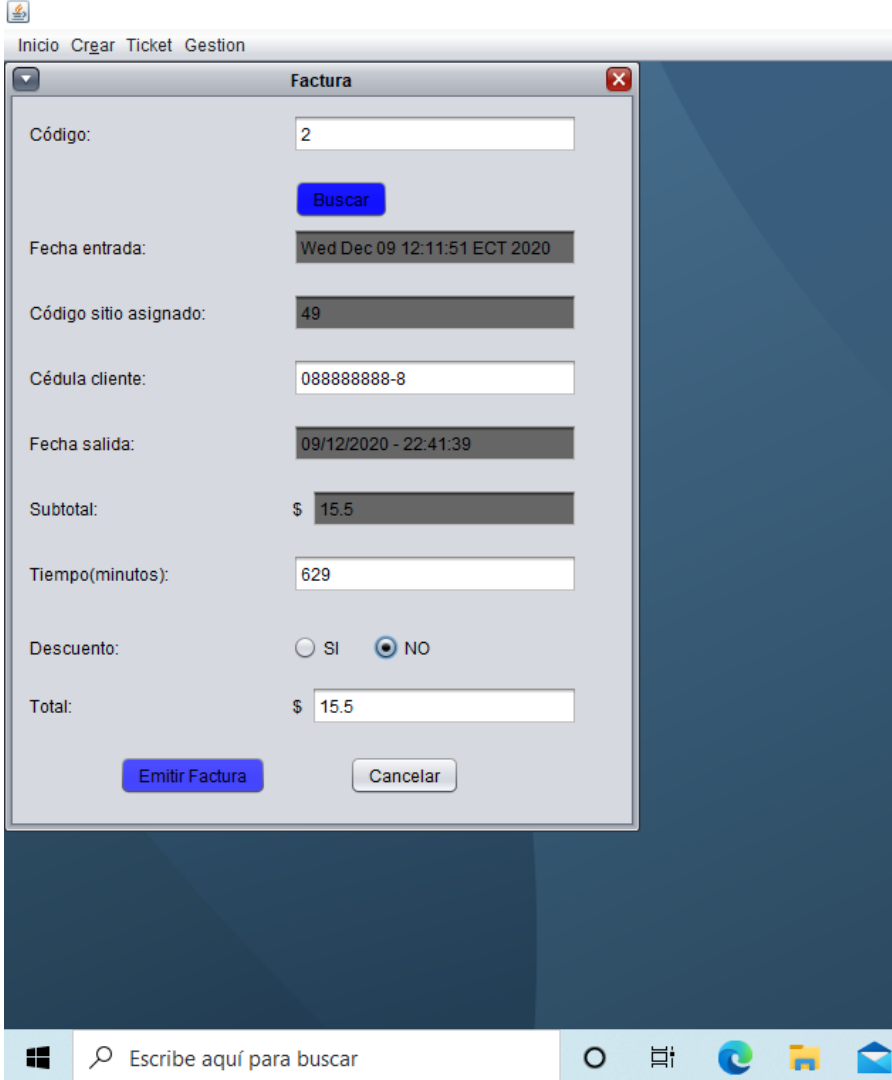


Cedula	Nombre	Apellido
05555555-5	Mohamed	Salah
010265489-4	Juan	Guaman

Cabe recalcar que los clientes que tienen contrato no aparecerán en la tabla de clientes de esta ventana.

Ahora emitiremos un ticket de salida:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Para emitir la factura solamente es necesario escribir el código del ticket emitido de entrada y buscarlo. Luego seleccionaremos si podemos dar o no descuento.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Inicio Crgar Ticket Gestion

**Factura**

Código:

Fecha entrada:

Código sitio asignado:

Cédula cliente:

Fecha salida:


Subtotal:

Tiempo(minutos):

Descuento: ☐ SI ☒ NO

Total: \$

**Mensaje**

 Factura emitida con éxito

Windows taskbar: Escribe aquí para buscar

Ahora mostraremos la ventana donde podemos observar todas las facturas emitidas:

Inicio Crgar Ticket Gestion

**Facturas**

Código:

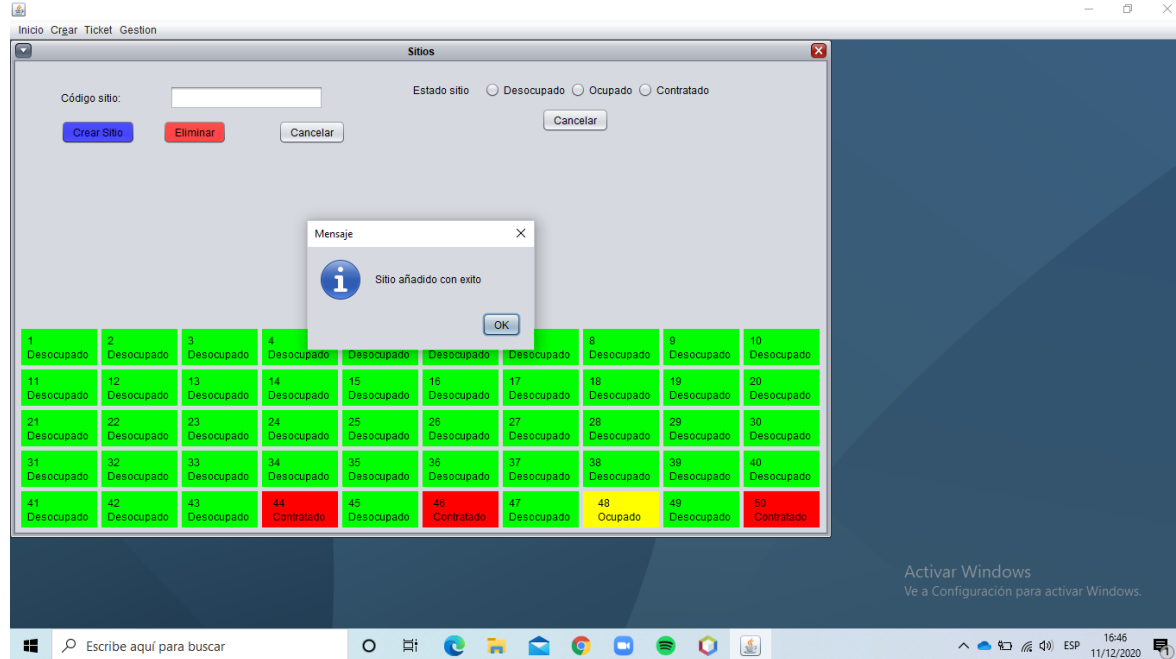
Código	Entrada	Salida	Tiempo	Subtotal	Total	Cliente	Sitio
1	Wed Dec 09 12:11:47 ...	Wed Dec 09 12:27:13 ...	15	0,25	0,25	088888888-8 ddd	50
2	Wed Dec 09 12:11:51 ...	Wed Dec 09 22:43:13 ...	629	15,5	15,5	088888888-8 ddd	49

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Por último, se presentará la interfaz gráfica de los parqueaderos:



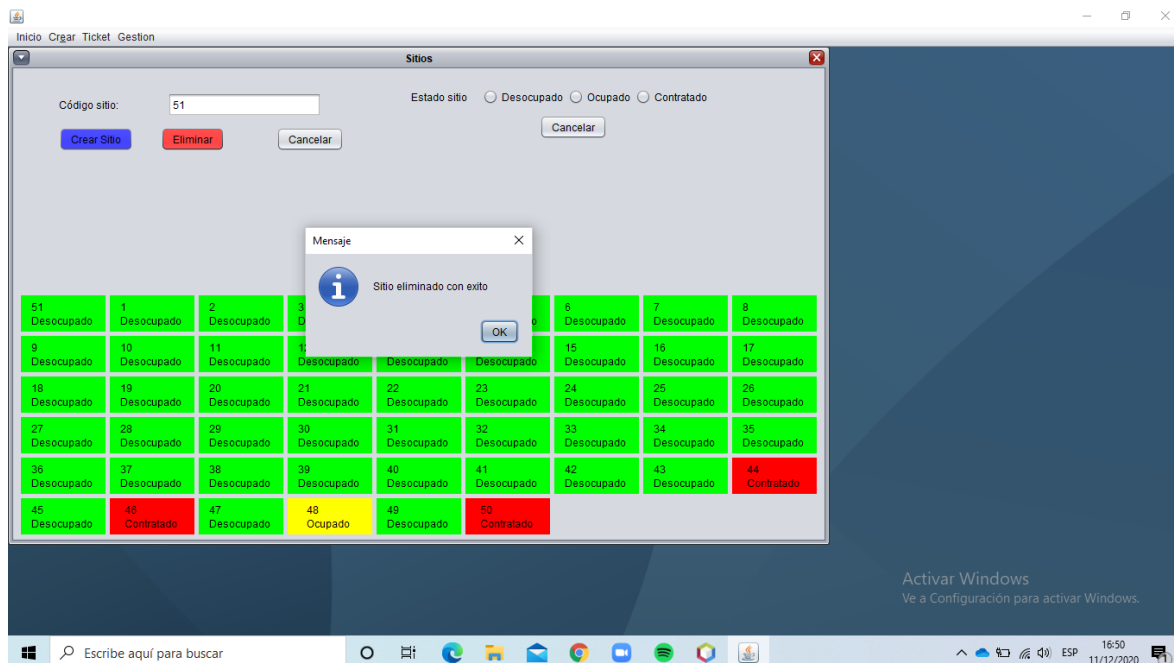
Ahora crearemos un nuevo parqueadero aplastando el botón de Crear Sitio:



	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	<b>CONSEJO ACADÉMICO</b>	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



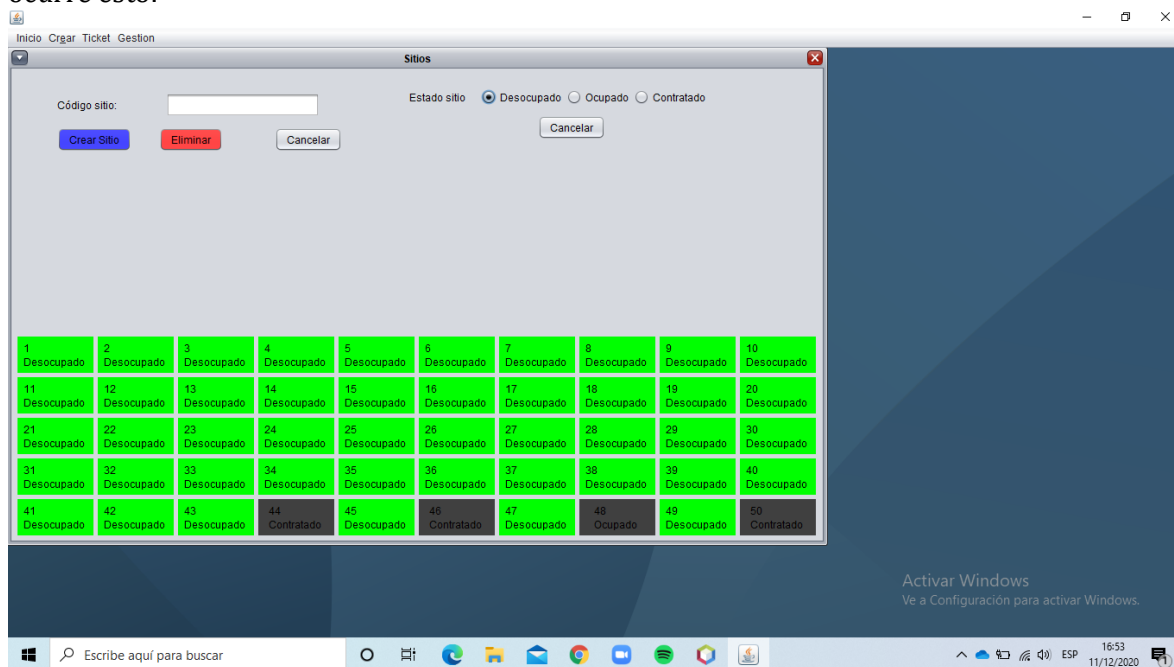
Ahora eliminaremos el mismo parqueadero que fue creado, para esto escribimos el número del sitio y aplastamos el botón eliminar:



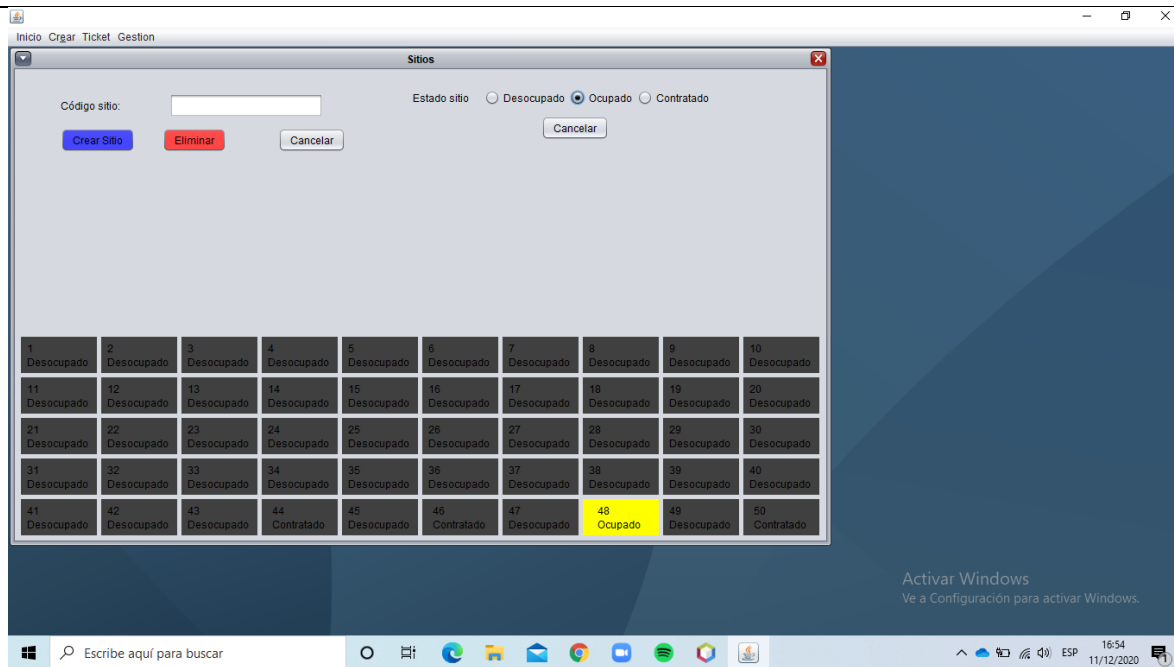
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



Tenemos también opciones de filtrado, vamos a demostrar el uso de las 3: primero daremos click en desocupado y ocurre esto:



Como podemos ver se pintan de gris oscuro los sitios que no cumplen con el filtro. Ahora damos click en Ocupado:



Inicio Crgar Ticket Gestion

Sitios

Código sitio:

Estado sitio: ☐ Desocupado ☒ Ocupado ☐ Contratado

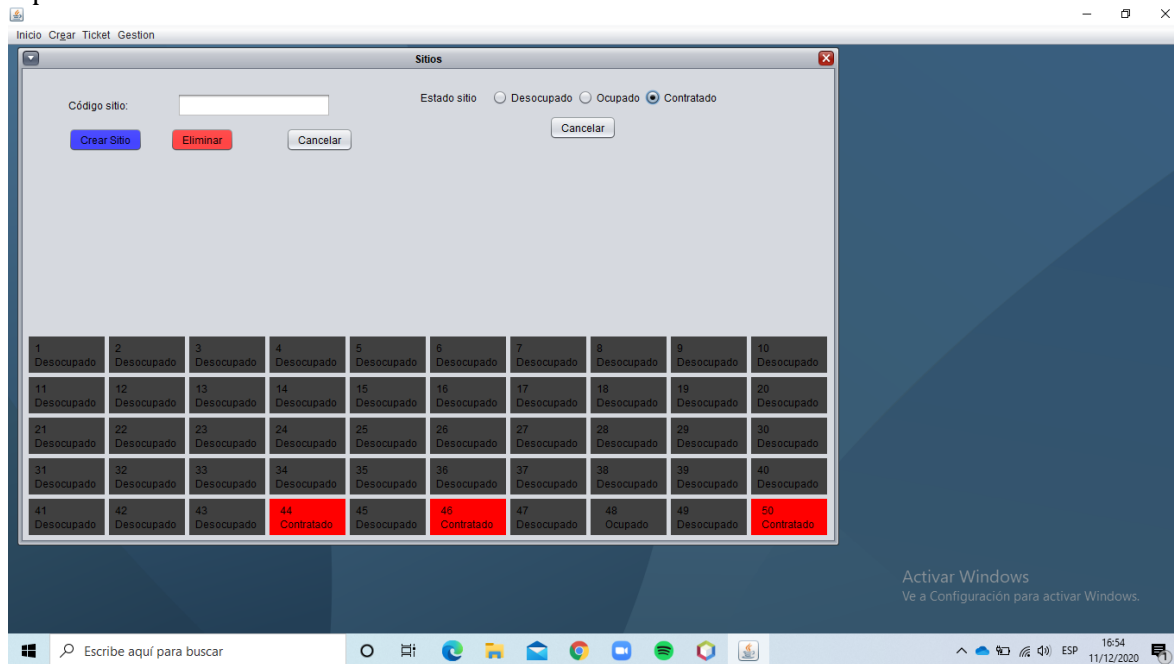
1 Desocupado	2 Desocupado	3 Desocupado	4 Desocupado	5 Desocupado	6 Desocupado	7 Desocupado	8 Desocupado	9 Desocupado	10 Desocupado
11 Desocupado	12 Desocupado	13 Desocupado	14 Desocupado	15 Desocupado	16 Desocupado	17 Desocupado	18 Desocupado	19 Desocupado	20 Desocupado
21 Desocupado	22 Desocupado	23 Desocupado	24 Desocupado	25 Desocupado	26 Desocupado	27 Desocupado	28 Desocupado	29 Desocupado	30 Desocupado
31 Desocupado	32 Desocupado	33 Desocupado	34 Desocupado	35 Desocupado	36 Desocupado	37 Desocupado	38 Desocupado	39 Desocupado	40 Desocupado
41 Desocupado	42 Desocupado	43 Desocupado	44 Contratado	45 Desocupado	46 Contratado	47 Desocupado	48 Ocupado	49 Desocupado	50 Contratado

Activar Windows  
Ve a Configuración para activar Windows.

Escribe aquí para buscar

16:54 11/12/2020

Y por último damos click en Contratado:



Inicio Crgar Ticket Gestion

Sitios

Código sitio:

Estado sitio: ☐ Desocupado ☐ Ocupado ☒ Contratado

1 Desocupado	2 Desocupado	3 Desocupado	4 Desocupado	5 Desocupado	6 Desocupado	7 Desocupado	8 Desocupado	9 Desocupado	10 Desocupado
11 Desocupado	12 Desocupado	13 Desocupado	14 Desocupado	15 Desocupado	16 Desocupado	17 Desocupado	18 Desocupado	19 Desocupado	20 Desocupado
21 Desocupado	22 Desocupado	23 Desocupado	24 Desocupado	25 Desocupado	26 Desocupado	27 Desocupado	28 Desocupado	29 Desocupado	30 Desocupado
31 Desocupado	32 Desocupado	33 Desocupado	34 Desocupado	35 Desocupado	36 Desocupado	37 Desocupado	38 Desocupado	39 Desocupado	40 Desocupado
41 Desocupado	42 Desocupado	43 Desocupado	44 Contratado	45 Desocupado	46 Contratado	47 Desocupado	48 Ocupado	49 Desocupado	50 Contratado

Activar Windows  
Ve a Configuración para activar Windows.

Escribe aquí para buscar

16:54 11/12/2020

## 1. Tutorial técnico del uso (Manual técnico):

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

### 3.- Pseudocódigo del proyecto:

paquete modelo:

- Administrador:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;
```

```
import java.io.Serializable;
import java.lang.reflect.Constructor;
import java.util.Calendar;
```

```
/**
 *
 * @author User
 */
```

```
public class Administrador extends Persona implements Serializable {
```

```
    private String correo;
    private String password;
    private EnumTipoAdministrador enumTipoAdministrador;
    public static Administrador instance;
```

```
    private Administrador() {

    }
}
```

```
    private Administrador(String correo, String password, int codigo, String codigoldentificacion,
        String nombre, String apellido, Calendar fechaNacimiento, String numeroTelefono,
        EnumTipoAdministrador enumTipoAdministrador) {
        super(codigo, codigoldentificacion, nombre, apellido, fechaNacimiento, numeroTelefono);
        this.correo = correo;
        this.password = password;
        this.enumTipoAdministrador = enumTipoAdministrador;
    }
}
```

```
public Administrador getInstance() {
    if (Administrador.instance == null) {
        Constructor<Administrador> constructor;
        try {
```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    constructor = Administrador.class.getDeclaredConstructor();
    constructor.setAccessible(true);
    Administrador admin = constructor.newInstance();

    Administrador.instance = admin;

    return Administrador.instance;
} catch (Exception e) {
    e.printStackTrace();
}

return Administrador.instance;
}

return Administrador.instance;
}

public String getCorreo() {
    return correo;
}

public void setCorreo(String correo) {
    this.correo = correo;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public EnumTipoAdministrador getEnumTipoAdministrador() {
    return enumTipoAdministrador;
}

public void setEnumTipoAdministrador(EnumTipoAdministrador enumTipoAdministrador) {
    this.enumTipoAdministrador = enumTipoAdministrador;
}

@Override
public String toString() {
    return super.toString() + "Administrador: " + "correo=" + correo + ", password="

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

+ password + ", enumTipoAdministrador=" + enumTipoAdministrador;
    }
}

    • Cliente:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

/**
 *
 * @author User
 */
public class Cliente extends Persona implements Serializable {

    private EnumClienteContrato enumClienteContrato;

    public Cliente() {

    }

    public Cliente(int codigo, String codigoidentificacion, String nombre, String apellido,
        Calendar fechaNacimiento, String numeroTelefono, EnumClienteContrato enumClienteContrato)
    {
        super(codigo, codigoidentificacion, nombre, apellido, fechaNacimiento, numeroTelefono);

        this.enumClienteContrato = enumClienteContrato;
    }

    public EnumClienteContrato getEnumClienteContrato() {
        return enumClienteContrato;
    }

    public void setEnumClienteContrato(EnumClienteContrato enumClienteContrato) {
        this.enumClienteContrato = enumClienteContrato;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

@Override
public String toString() {
    return "Cliente{" + "enumClienteContrato=" + enumClienteContrato + '}';
}

}

    • Contrato:
    /*
    * To change this license header, choose License Headers in Project Properties.
    * To change this template file, choose Tools | Templates
    * and open the template in the editor.
    */
package ups.edu.ec.modelo;

import java.io.Serializable;
import java.util.Calendar;
import java.util.Date;

/**
 *
 * @author User
 */
public class Contrato implements Serializable {

    private int codigo;
    private Cliente cliente;
    private Sitio sitio;
    private Date fechaInicio;
    private Date fechaFin;
    private EnumTipoPrecioContrato enumTipoPrecioContrato;
    private boolean pago;
    private double subtotal;
    private boolean descuento;
    private double total;

    public Contrato(int codigo, Cliente cliente, Sitio sitio, Date fechaInicio,
        Date fechaFin, EnumTipoPrecioContrato enumTipoPrecioContrato, boolean pago, double
    subtotal,
        boolean descuento, double total) {
        this.codigo = codigo;
        this.cliente = cliente;
        this.sitio = sitio;
        this.fechaInicio = fechaInicio;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

this.fechaFin = fechaFin;
this.enumTipoPrecioContrato = enumTipoPrecioContrato;
this.pago = pago;
this.subtotal = subtotal;
this.descuento = descuento;
this.total = total;
}

public Contrato(int codigo, Cliente cliente, Sitio sitio, Date fechaInicio,
    Date fechaFin, EnumTipoPrecioContrato enumTipoPrecioContrato) {
    this.codigo = codigo;
    this.cliente = cliente;
    this.sitio = sitio;
    this.fechaInicio = fechaInicio;
    this.fechaFin = fechaFin;
    this.enumTipoPrecioContrato = enumTipoPrecioContrato;
}

public Contrato() {
}

public EnumTipoPrecioContrato getEnumTipoPrecioContrato() {
    return enumTipoPrecioContrato;
}

public void setEnumTipoPrecioContrato(EnumTipoPrecioContrato enumTipoPrecioContrato) {
    this.enumTipoPrecioContrato = enumTipoPrecioContrato;
}

public boolean isDescuento() {
    return descuento;
}

public void setDescuento(boolean descuento) {
    this.descuento = descuento;
}

public double getTotal() {
    return total;
}

public void setTotal(double total) {
    this.total = total;
}

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public int getCodigo() {
    return codigo;
}

public void setCodigo(int codigo) {
    this.codigo = codigo;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Sitio getSitio() {
    return sitio;
}

public void setSitio(Sitio sitio) {
    this.sitio = sitio;
}

public Date getFechaInicio() {
    return fechaInicio;
}

public void setFechaInicio(Date fechaInicio) {
    this.fechaInicio = fechaInicio;
}

public Date getFechaFin() {
    return fechaFin;
}

public void setFechaFin(Date fechaFin) {
    this.fechaFin = fechaFin;
}

public boolean isPago() {
    return pago;
}

public void setPago(boolean pago) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

this.pago = pago;
}

public double getSubtotal() {
    return subtotal;
}

public void setSubtotal(double subtotal) {
    this.subtotal = subtotal;
}

@Override
public int hashCode() {
    int hash = 7;
    hash = 19 * hash + this.codigo;
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Contrato other = (Contrato) obj;
    if (this.codigo != other.codigo) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Contrato:" + "codigo=" + codigo + ", cliente=" + cliente + ", sitio="
        + sitio + ", fechaInicio=" + fechaInicio + ", fechaFin=" + fechaFin
        + ", pago=" + pago + ", subtotal=" + subtotal;
}
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Factura:

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;

import java.io.Serializable;
import java.util.Date;

/**
 *
 * @author User
 */
public class Factura implements Serializable {

    private int codigo;
    private Cliente cliente;
    private Date fechaEntrada;
    private Date fechaSalida;
    private int tiempo;
    private boolean descuento;
    private double subtotal;
    private double total;
    private Sitio sitio;

    public Factura(int codigo, Cliente cliente, Date fechaEntrada, Date fechaSalida,
        int tiempo, boolean descuento, double total, Sitio sitio) {
        this.codigo = codigo;
        this.cliente = cliente;
        this.fechaEntrada = fechaEntrada;
        this.fechaSalida = fechaSalida;
        this.tiempo = tiempo;
        this.descuento = descuento;
        this.total = total;
        this.sitio = sitio;
    }

    public Factura(int codigo, Cliente cliente, Date fechaEntrada, Sitio sitio) {
        this.codigo = codigo;
        this.cliente = cliente;
        this.fechaEntrada = fechaEntrada;
        this.sitio = sitio;
        this.subtotal = 0;
    }

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public Factura() {
}

public double getSubtotal() {
    return subtotal;
}

public void setSubtotal(double subtotal) {
    this.subtotal = subtotal;
}

public int getCodigo() {
    return codigo;
}

public void setCodigo(int codigo) {
    this.codigo = codigo;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

public Date getFechaEntrada() {
    return fechaEntrada;
}

public void setFechaEntrada(Date fechaEntrada) {
    this.fechaEntrada = fechaEntrada;
}

public Date getFechaSalida() {
    return fechaSalida;
}

public void setFechaSalida(Date fechaSalida) {
    this.fechaSalida = fechaSalida;
}

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public int getTiempo() {
    return tiempo;
}

public void setTiempo(int tiempo) {
    this.tiempo = tiempo;
}

public boolean isDescuento() {
    return descuento;
}

public void setDescuento(boolean descuento) {
    this.descuento = descuento;
}

public double getTotal() {
    return total;
}

public void setTotal(double total) {
    this.total = total;
}

public Sitio getSitio() {
    return sitio;
}

public void setSitio(Sitio sitio) {
    this.sitio = sitio;
}

@Override
public int hashCode() {
    int hash = 7;
    hash = 97 * hash + this.codigo;
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {

```

```
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Factura other = (Factura) obj;
    if (this.codigo != other.codigo) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Factura: " + "codigo=" + codigo + ", cliente=" + cliente + ", fechaEntrada="
        + fechaEntrada + ", fechaSalida=" + fechaSalida + ", tiempo=" + tiempo
        + ", descuento=" + descuento + ", total=" + total + ", sitio:" + sitio;
}
}
```

- Persona:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;

import java.io.Serializable;
import java.util.Calendar;
import java.util.Objects;

/**
 *
 * @author User
 */
public class Persona implements Serializable {

    private int codigo;
    private String cedula;
    private String nombre;
    private String apellido;
    private Calendar fechaNacimiento;
    private String numeroTelefono;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public Persona() {

}

public Persona(int codigo, String cedula, String nombre, String apellido,
    Calendar fechaNacimiento, String numeroTelefono) {
    this.codigo = codigo;
    this.cedula = cedula;
    this.nombre = nombre;
    this.apellido = apellido;
    this.fechaNacimiento = fechaNacimiento;
    this.numeroTelefono = numeroTelefono;
}

public Persona(String cedula) {
    this.cedula = cedula;
}

public int getCodigo() {
    return codigo;
}

public void setCodigo(int codigo) {
    this.codigo = codigo;
}

public String getCedula() {
    return cedula;
}

public void setCedula(String codigoidentificacion) {
    this.cedula = codigoidentificacion;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getApellido() {
    return apellido;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public Calendar getFechaNacimiento() {
    return fechaNacimiento;
}

public void setFechaNacimiento(Calendar fechaNacimiento) {
    this.fechaNacimiento = fechaNacimiento;
}

public String getNumeroTelefono() {
    return numeroTelefono;
}

public void setNumeroTelefono(String numeroTelefono) {
    this.numeroTelefono = numeroTelefono;
}

@Override
public int hashCode() {
    int hash = 5;
    hash = 67 * hash + Objects.hashCode(this.cedula);
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Persona other = (Persona) obj;
    if (!Objects.equals(this.cedula, other.cedula)) {
        return false;
    }
    return true;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

@Override
public String toString() {
    return "Persona{" + "codigo=" + codigo + ", codigoidentificacion=" + cedula
        + ", nombre=" + nombre + ", apellido=" + apellido + ", fechaNacimiento="
        + fechaNacimiento + ", numeroTelefono=" + numeroTelefono + "}";
}

}

    • Sitio:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;

import java.io.Serializable;

/**
 *
 * @author User
 */
public class Sitio implements Serializable {

    private int codigo;
    private EnumEstadoSitio enumEstadoSitio;
    private double precioIntervalo;

    public Sitio(int codigo, EnumEstadoSitio enumEstadoSitio) {
        this.codigo = codigo;
        this.enumEstadoSitio = enumEstadoSitio;
        this.precioIntervalo = 0.25;
    }

    public Sitio() {
    }

    public int getCodigo() {
        return codigo;
    }

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void setCodigo(int codigo) {
    this.codigo = codigo;
}

public EnumEstadoSitio getEnumEstadoSitio() {
    return enumEstadoSitio;
}

public void setEnumEstadoSitio(EnumEstadoSitio enumEstadoSitio) {
    this.enumEstadoSitio = enumEstadoSitio;
}

public double getPrecioHora() {
    return precioIntervalo;
}

public void setPrecioHora(double precioHora) {
    this.precioIntervalo = precioHora;
}

@Override
public int hashCode() {
    int hash = 7;
    hash = 97 * hash + this.codigo;
    return hash;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Sitio other = (Sitio) obj;
    if (this.codigo != other.codigo) {
        return false;
    }
    return true;
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

@Override
public String toString() {
    return "Sitio:" + "codigo=" + codigo + ", estado=" + enumEstadoSitio + ", precioHora="
        + precioIntervalo;
}
}

    • Vehículo:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;

import java.io.Serializable;
import java.util.Objects;

/**
 *
 * @author User
 */
public class Vehiculo implements Serializable {

    private int codigo;
    private String placa;
    private String marca;
    private String modelo;
    private Cliente cliente;

    public Vehiculo() {
    }

    public Vehiculo(int codigo, String placa, String marca, String modelo, Cliente cliente) {
        this.codigo = codigo;
        this.placa = placa;
        this.marca = marca;
        this.modelo = modelo;
        this.cliente = cliente;
    }

    public int getCodigo() {
        return codigo;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void setCodigo(int codigo) {
    this.codigo = codigo;
}

public String getPlaca() {
    return placa;
}

public void setPlaca(String placa) {
    this.placa = placa;
}

public String getMarca() {
    return marca;
}

public void setMarca(String marca) {
    this.marca = marca;
}

public String getModelo() {
    return modelo;
}

public void setModelo(String modelo) {
    this.modelo = modelo;
}

public Cliente getCliente() {
    return cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

@Override
public int hashCode() {
    int hash = 5;
    hash = 79 * hash + Objects.hashCode(this.placa);
    return hash;
}

@Override

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (getClass() != obj.getClass()) {
        return false;
    }
    final Vehiculo other = (Vehiculo) obj;
    if (!Objects.equals(this.placa, other.placa)) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Auto: " + "codigo=" + codigo + ", placa=" + placa + ", marca=" + marca
        + ", modelo=" + modelo + ", cliente: " + cliente;
}
}

```

EnumClienteContrato:

```

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

```

```
package ups.edu.ec.modelo;
```

```

/**
 *
 * @author User
 */

```

```

public enum EnumClienteContrato {
    SI,
    NO
}

```

EnumEstadoSitio:

```

/*
 * To change this license header, choose License Headers in Project Properties.

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

\* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.

```
*/
package ups.edu.ec.modelo;
```

```
/**
 *
 * @author User
 */
public enum EnumEstadoSitio {
    OCUPADO,
    DESOCUPADO,
    CONTRATADO,
}
```

EnumTipoAdministrador:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;
```

```
/**
 *
 * @author User
 */
public enum EnumTipoAdministrador {
    COMPLETO,
    SIMPLE
}
```

EnumTipoPrecioContrato:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.modelo;
```

```
/**
 *
 * @author User
 */
public enum EnumTipoPrecioContrato {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

DIA,  
SEMANA,  
MES

}

Para el paquete controlador se generaron las siguientes clases:

- Controlador:

\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

```
package ups.edu.ec.controlador;
```

```
import java.io.FileInputStream;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.io.ObjectInputStream;
```

```
import java.io.ObjectOutputStream;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public abstract class Controlador<T> {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private List<T> listaGenerica;

private String ruta;

public Controlador(String ruta) {
    this.ruta = ruta;
    listaGenerica = new ArrayList<>();
    this.cargarDatos();
}

public void cargarDatos() {
    try {
        FileInputStream file = new FileInputStream(ruta);
        ObjectInputStream objeto = new ObjectInputStream(file);

        listaGenerica = (List<T>) objeto.readObject();

    } catch (IOException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Controlador.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public boolean guardarDatos(List<T> listaGuardar) {

    try {
        FileOutputStream file = new FileOutputStream(ruta);
        ObjectOutputStream objetoEscritura = new ObjectOutputStream(file);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

objetoEscritura.writeObject(listaGuardar);

return true;
} catch (IOException e) {
    e.printStackTrace();
}

return false;
}


public boolean create(T objeto) {

    if (!listaGenerica.contains(objeto)) {
        listaGenerica.add(objeto);
        return guardarDatos(listaGenerica);
    } else {
        return false;
    }

}

public T read(T objeto) {
    if (listaGenerica.contains(objeto)) {
        return listaGenerica.stream().filter(ob -> ob.equals(objeto)).findFirst().get();
    } else {
        return null;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public boolean update(T objetoAntiguo, T objetoNuevo) {
    int index = listaGenerica.indexOf(objetoAntiguo);
    listaGenerica.set(index, objetoNuevo);
    return guardarDatos(listaGenerica);
}

public boolean delete(T objetoEliminar) {
    listaGenerica.remove(objetoEliminar);
    return guardarDatos(listaGenerica);
}

public List<T> getListaGenerica() {
    return listaGenerica;
}
}

    • ControladoAdministrador:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.controlador;

import java.security.MessageDigest;
import java.util.Arrays;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import java.util.List;
import java.util.NoSuchElementException;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;
import ups.edu.ec.modelo.Administrador;
import ups.edu.ec.modelo.Cliente;


/**
 *
 * @author User
 */
public class ControladorAdministrador extends Controlador {

    public ControladorAdministrador(String ruta) {
        super(ruta);
    }

    public int generarCodigo() {
        int codigo = 0;
        for (Administrador c : (List<Administrador>) getListaGenerica()) {
            codigo = c.getCodigo();
        }

        return codigo + 1;
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public Administrador iniciarSesion(String correo, String password, String keyUnlock) {
    var listaAdmins = (List<Administrador>) getListGenerica();
    Administrador admin = null;
    try {
        admin = listaAdmins.stream().filter(ad -> ad.getCorreo().equals(correo) && desencriptarPassword(keyUnlock,
ad.getPassword()).equals(password)).findFirst().get();
    } catch (NoSuchElementException e) {
        e.printStackTrace();
    }

    if (Administrador.instance == null) {
        Administrador.instance = (Administrador) admin;
        return Administrador.instance;
    } else {
        return null;
    }
}

public String encriptarPassword(String key, String password) {


    String encriptacion = "";

    try {
        MessageDigest msmd = MessageDigest.getInstance("MD5");
        byte[] llavePassword = msmd.digest(key.getBytes("utf-8"));
        byte[] bytesKey = Arrays.copyOf(llavePassword, 24);

        SecretKey secretKey = new SecretKeySpec(bytesKey, "DESede");

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

Cipher cifrado = Cipher.getInstance("DESede");
cifrado.init(Cipher.ENCRYPT_MODE, secretKey);

byte[] bytesPassword = password.getBytes("utf-8");
byte[] buffer = cifrado.doFinal(bytesPassword);

byte[] base64Bytes = Base64.encodeBase64(buffer);

encriptacion = new String(base64Bytes);

return encriptacion;

} catch (Exception e) {
    e.printStackTrace();
}

return encriptacion;
}


public String desencriptarPassword(String key, String passwordEncriptado) {

    String desincriptacion = "";

    try {

        byte[] mensaje = Base64.decodeBase64(passwordEncriptado.getBytes("utf-8"));
        MessageDigest msmd = MessageDigest.getInstance("MD5");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

byte[] llavePassword = msmd.digest(key.getBytes("utf-8"));
byte[] bytesKey = Arrays.copyOf(llavePassword, 24);

SecretKey secretKey = new SecretKeySpec(bytesKey, "DESede");

Cipher decifrado = Cipher.getInstance("DESede");
decifrado.init(Cipher.DECRYPT_MODE, secretKey);

byte[] cadenaDesencriptada = decifrado.doFinal(mensaje);

desincryption = new String(cadenaDesencriptada, "utf-8");

return desincryption;

} catch (Exception e) {
    e.printStackTrace();
}

return desincryption;
}
}

```

- ControladorCliente:

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.controlador;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
import java.util.List;
import ups.edu.ec.modelo.Cliente;

/**
 *
 * @author User
 */
public class ControladorCliente extends Controlador {

    public ControladorCliente(String ruta) {
        super(ruta);
    }

    public int generarCodigo() {
        int codigo = 0;
        for (Cliente c : (List<Cliente>) getListaGenerica()) {
            codigo = c.getCodigo();
        }

        return codigo + 1;
    }
}

• ControladorContrato:

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

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

\*/

```
package ups.edu.ec.controlador;
```

```
import java.util.Calendar;
```

```
import java.util.Date;
```

```
import java.util.Enumeration;
```

```
import java.util.List;
```

```
import ups.edu.ec.modelo.Cliente;
```

```
import ups.edu.ec.modelo.Contrato;
```

```
import ups.edu.ec.modelo.EnumTipoPrecioContrato;
```

```
import ups.edu.ec.modelo.Sitio;
```

/\*\*

\*

\* @author User

\*/

```
public class ControladorContrato extends Controlador {
```

```
    public ControladorContrato(String ruta) {
```

```
        super(ruta);
```

```
    }
```

```
    public int generarCodigo() {
```

```
        int codigo = 0;
```

```
        for (Contrato c : (List<Contrato>) getListaGenerica()) {
```

```
            codigo = c.getCodigo();
```

```
        }
```

```
        return codigo + 1;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```


}

public double obtenerSubtotal(Sitio sitio, Date fechaEntrada, Date fechaSalida, EnumTipoPrecioContrato tipoPrecio) {

    long tiempoE = 0;
    long tiempoS = 0;
    int tiempo = 0;

    switch (tipoPrecio) {
        case MES -> {
            tiempoE = (fechaEntrada.getYear() + 1900) * 525600;
            tiempoE += (fechaEntrada.getMonth() + 1) * 43804.8;
            tiempoE += (fechaEntrada.getDate() * 1440);
            tiempoS = (fechaSalida.getYear() + 1900) * 525600;
            tiempoS += (fechaSalida.getMonth() + 1) * 43804.8;
            tiempoS += (fechaSalida.getDate() * 1440);
            tiempo = Math.round(tiempoS - tiempoE);
            break;
        }
        case SEMANA -> {
            tiempoE = (fechaEntrada.getYear() + 1900) * 525600;
            tiempoE += (fechaEntrada.getMonth() + 1) * 43804.8;
            tiempoE += (fechaEntrada.getDate() * 1440);
            tiempoS = (fechaSalida.getYear() + 1900) * 525600;
            tiempoS += (fechaSalida.getMonth() + 1) * 43804.8;
            tiempoS += (fechaSalida.getDate() * 1440);
            tiempo = Math.round(tiempoS - tiempoE);
            break;
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
default -> {
    tiempoE += (fechaEntrada.getMonth() + 1) * 43804.8;
    tiempoE += (fechaEntrada.getDate() * 1440);
    tiempoS += (fechaSalida.getMonth() + 1) * 43804.8;
    tiempoS += (fechaSalida.getDate() * 1440);
    tiempo = Math.round(tiempoS - tiempoE);
    break;
}
}
```

```
System.out.println(tiempoS + " " + tiempoE);
```

```
double subtotal = tiempo * sitio.getPrecioHora();
return subtotal;
```

```
}
```

```
public double obtenerTotal(double subtotal) {
```

```
    return subtotal - (subtotal * 0.10);
```

```
}
```

```
}
```

- ControladorFactura:

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

\*/

```
package ups.edu.ec.controlador;
```

```
import java.text.SimpleDateFormat;
```

```
import java.time.Period;
```

```
import java.time.format.DateTimeFormatter;
```

```
import java.util.Date;
```

```
import java.util.List;
```

```
import java.util.Locale;
```

```
import ups.edu.ec.modelo.Cliente;
```

```
import ups.edu.ec.modelo.Factura;
```

/\*\*

\*

\* @author User

\*/

```
public class ControladorFactura extends Controlador {
```

```
    public ControladorFactura(String ruta) {
```

```
        super(ruta);
```

```
    }
```

```
    public int generarCodigo() {
```

```
        int codigo = 0;
```

```
        for (Factura c : (List<Factura>) getListaGenerica()) {
```

```
            codigo = c.getCodigo();
```

```
        }
```

```
        return codigo + 1;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public double calcularSubtotal(Date fechaEntrada, Date fechaSalida) {
    /*
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

    String fechaE = formatter.format(fechaEntrada);
    String fechaS = formatter.format(fechaSalida);

    DateTimeFormatter dtf = DateTimeFormat.forPattern("dd/MM/yyyy HH:mm:ss").withLocale(Locale.ENGLISH);


    DateTime dateTime1 = dtf.parseDateTime(fechaE);
    DateTime dateTime2 = dtf.parseDateTime(fechaS);
    Period period = new Period(dateTime1, dateTime2);

    PeriodFormatter formatter = new PeriodFormatterBuilder()
        .appendYears().appendSuffix(" years ")
        .appendMonths().appendSuffix(" months ")
        .appendWeeks().appendSuffix(" weeks ")
        .appendDays().appendSuffix(" days ")
        .appendHours().appendSuffix(" hours ")
        .appendMinutes().appendSuffix(" minutes ")
        .appendSeconds().appendSuffix(" seconds ")
        .printZeroNever()
        .toFormatter();

    String elapsed = formatter.print(period);
    System.out.println(elapsed);*/

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

/\*

```
long numeroMilisegundosEntrada = fechaEntrada.getTime();
```

```
long numeroMilisegundosSalida = fechaSalida.getTime();
```

```
int totalMilisegundos = Math.round(numeroMilisegundosSalida - numeroMilisegundosEntrada);
```

```
double subtotal = 0.25 * ((totalMilisegundos / 1000 * 60) / 10);*/
```

```
int hora = (fechaEntrada.getHours() * 3600);
```

```
int minutos = (fechaEntrada.getMinutes() * 60);
```

```
int segundos = (fechaEntrada.getSeconds());
```

```
int tiempo1 = hora + minutos + segundos;
```

```
int hora2 = (fechaSalida.getHours() * 3600);
```

```
int minutos2 = (fechaSalida.getMinutes() * 60);
```

```
int segundos2 = (fechaSalida.getSeconds());
```

```
int tiempo2 = hora2 + minutos2 + segundos2;
```

```
int tiempoTotal = Math.round(((tiempo2 - tiempo1) / 60));
```

```
double subtotal = 0.25 * (tiempoTotal / 10);
```

```
return subtotal;
```

```
}
```

```
public int calcularTiempo(Date fechaEntrada, Date fechaSalida) {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/*
long numeroMilisegundosEntrada = fechaEntrada.getTime();
long numeroMilisegundosSalida = fechaSalida.getTime();

int totalMilisegundos = Math.round(numeroMilisegundosSalida - numeroMilisegundosEntrada);

int tiempo = Math.round((totalMilisegundos / 1000 * 60));

return tiempo;*/

int hora = (fechaEntrada.getHours() * 3600);
int minutos = (fechaEntrada.getMinutes() * 60);
int segundos = (fechaEntrada.getSeconds());

int tiempo1 = hora + minutos + segundos;

int hora2 = (fechaSalida.getHours() * 3600);
int minutos2 = (fechaSalida.getMinutes() * 60);
int segundos2 = (fechaSalida.getSeconds());

int tiempo2 = hora2 + minutos2 + segundos2;

int tiempoTotal = Math.round(((tiempo2 - tiempo1) / 60));

return tiempoTotal;
}

public double calcularTotal(double subtotal, boolean descuento) {
    if (descuento) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

return subtotal - (subtotal * 0.10);
    } else {
        return subtotal;
    }
}
}
}

```

- ControladorSitio:

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.controlador;

import java.util.List;
import ups.edu.ec.modelo.EnumEstadoSitio;
import ups.edu.ec.modelo.Sitio;

/**
 *
 * @author User
 */
public class ControladorSitio extends Controlador {

    public ControladorSitio(String ruta) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

super(ruta);

//cargarSitios();

}

public void cargarSitios() {
    var listaSitios = (List<Sitio>) getListGenerica();
    int numeroSitios = 51;

    while (numeroSitios > 1) {
        numeroSitios--;
        Sitio sitio = new Sitio(numeroSitios, EnumEstadoSitio.DESOCUPADO);
        listaSitios.add(sitio);

    }
    guardarDatos(listaSitios);
}

public Sitio encontrarSitioDesocupado() {
    var listaSitios = (List<Sitio>) getListGenerica();
    return listaSitios.stream().filter(sit
sit.getEnumEstadoSitio().equals(EnumEstadoSitio.DESOCUPADO)).findFirst().get();
}

public void cambiarAEstadoOcupado(Sitio sitio) {
    sitio.setEnumEstadoSitio(EnumEstadoSitio.OCUPADO);
    this.update(sitio, sitio);

}

```

->

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void cambiarAEstadoContratado(Sitio sitio) {
    sitio.setEnumEstadoSitio(EnumEstadoSitio.CONTRATADO);
    this.update(sitio, sitio);

}

public void cambiarAEstadoDescupado(Sitio sitio) {
    sitio.setEnumEstadoSitio(EnumEstadoSitio.DESOCUPADO);
    this.update(sitio, sitio);
}
}

```

- ControladorVehiculo:

```

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

package ups.edu.ec.controlador;

import java.util.List;
import ups.edu.ec.modelo.Vehiculo;

/**
 *
 * @author User
 */
public class ControladorVehiculo extends Controlador {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public ControladorVehiculo(String ruta) {
    super(ruta);
}

public int generarCodigo() {
    int codigo = 0;
    for (Vehiculo c : (List<Vehiculo>) getListaGenerica()) {
        codigo = c.getCodigo();
    }

    return codigo + 1;
}
}

```

Luego continuamos con el paquete vista, en este se crearon las siguientes ventanas:

- VentanaPrincipal:

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.vista;

import java.lang.reflect.Constructor;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import javax.swing.JOptionPane;

import ups.edu.ec.controlador.ControladorAdministrador;
import ups.edu.ec.controlador.ControladorCliente;
import ups.edu.ec.controlador.ControladorContrato;
import ups.edu.ec.controlador.ControladorFactura;
import ups.edu.ec.controlador.ControladorSitio;
import ups.edu.ec.controlador.ControladorVehiculo;
import ups.edu.ec.modelo.Administrador;
import ups.edu.ec.modelo.EnumTipoAdministrador;

/**
 *
 * @author User
 */
public class VentanaPrincipal extends javax.swing.JFrame {

    private Administrador administrador;
    private boolean inicioSesion;
    private String keyUnlock;

    private ControladorAdministrador controladorAdministrador;
    private ControladorCliente controladorCliente;
    private ControladorVehiculo controladorVehiculo;
    private ControladorFactura controladorFactura;
    private ControladorSitio controladorSitio;
    private ControladorContrato controladorContrato;

    private VentanaLogIn ventanaLogIn;
    private VentanaCrearUsuario ventanaCrearUsuario;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private VentanaCrearCliente ventanaCrearCliente;

private VentanaVehiculo ventanaVehiculo;

private VentanaEntradaTicket ventanaEntradaTicket;

private VentanaSalidaTicket ventanaSalidaTicket;

private VentanaContrato ventanaContrato;

private VentanaGestionFacturas ventanaGestionFacturas;

private VentanaSitios ventanaSitios;

/**
 * Creates new form VentanaPrincipal
 */
public VentanaPrincipal() {
    initComponents();
    inicioSesion = false;
    /*
    try {
        Constructor<Administrador> constructor;
        constructor = Administrador.class.getDeclaredConstructor();
        constructor.setAccessible(true);
        administrador = constructor.newInstance();
        administrador.setNombre("Jose");
        administrador.setApellido("Jose");
        administrador.setCodigo(1);
        administrador.setCedula("0101010101");
        administrador.setCorreo("jose");

        administrador.setEnumTipoAdministrador(EnumTipoAdministrador.COMPLETO);

        Calendar fecha = new GregorianCalendar(1980, 10, 15);
        administrador.setFechaNacimiento(fecha);

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

administrador.setNumeroTelefono("0123456789");

} catch (Exception e) {
    e.printStackTrace();
}*/

keyUnlock = "programacion";

controladorAdministrador = new ControladorAdministrador("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\administradores.obj");

/*
String passwordEncriptado = controladorAdministrador.encriptarPassword(keyUnlock, "1234");
administrador.setPassword(passwordEncriptado);
controladorAdministrador.create(administrador);*/

controladorCliente = new ControladorCliente("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\clientes.obj");

controladorVehiculo = new ControladorVehiculo("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\vehiculos.obj");

controladorFactura = new ControladorFactura("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\facturas.obj");

controladorSitio = new ControladorSitio("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\sitios.obj");

controladorContrato = new ControladorContrato("C:\\Users\\User\\Desktop\\ProgramacionAplicada\\Proyecto Integrador Interciclo\\contratos.obj");

//System.out.println(controladorSitio.getListGenerica().size());

ventanaLogIn = new VentanaLogIn(controladorAdministrador, keyUnlock);

ventanaCrearUsuario = new VentanaCrearUsuario(controladorAdministrador, keyUnlock);

ventanaCrearCliente = new VentanaCrearCliente(controladorCliente);

ventanaVehiculo = new VentanaVehiculo(controladorCliente, controladorVehiculo);

ventanaEntradaTicket = new VentanaEntradaTicket(controladorSitio, controladorCliente, controladorFactura);

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

ventanaSalidaTicket = new VentanaSalidaTicket(controladorFactura, controladorSitio);

ventanaContrato = new VentanaContrato(controladorContrato, controladorSitio, controladorCliente);

ventanaGestionFacturas = new VentanaGestionFacturas(controladorFactura);

ventanaSitios = new VentanaSitios(controladorSitio);


desktopPane.add(ventanaLogIn);
desktopPane.add(ventanaCrearUsuario);
desktopPane.add(ventanaCrearCliente);
desktopPane.add(ventanaVehiculo);
desktopPane.add(ventanaEntradaTicket);
desktopPane.add(ventanaSalidaTicket);
desktopPane.add(ventanaContrato);
desktopPane.add(ventanaGestionFacturas);
desktopPane.add(ventanaSitios);


this.setExtendedState(VentanaPrincipal.MAXIMIZED_BOTH);
}

public void cerrarVentanas() {
    ventanaLogIn.hide();
    ventanaCrearUsuario.hide();
    ventanaCrearCliente.hide();
    ventanaVehiculo.hide();
    ventanaEntradaTicket.hide();
    ventanaSalidaTicket.hide();
    ventanaContrato.hide();
    ventanaGestionFacturas.hide();
    ventanaSitios.hide();
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    desktopPane = new javax.swing.JDesktopPane();
    menuBar = new javax.swing.JMenuBar();
    fileMenu = new javax.swing.JMenu();
    logInMenuItem = new javax.swing.JMenuItem();
    cerrarSesionMenuItem = new javax.swing.JMenuItem();
    exitMenuItem = new javax.swing.JMenuItem();
    editMenu = new javax.swing.JMenu();
    usuarioMenuItem = new javax.swing.JMenuItem();
    clienteMenuItem = new javax.swing.JMenuItem();
    vehiculoMenuItem = new javax.swing.JMenuItem();
    contratoMenuItem = new javax.swing.JMenuItem();
    helpMenu = new javax.swing.JMenu();
    entradaMenuItem = new javax.swing.JMenuItem();
    salidaMenuItem = new javax.swing.JMenuItem();
    jMenu1 = new javax.swing.JMenu();
    facturaMenuItem = new javax.swing.JMenuItem();
    sitiosMenuItem = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

fileMenu.setMnemonic('f');
fileMenu.setText("Inicio");

logInMenuItem.setMnemonic('o');
logInMenuItem.setText("Log In");
logInMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        logInMenuItemActionPerformed(evt);
    }
});
fileMenu.add(logInMenuItem);

cerrarSesionMenuItem.setMnemonic('s');
cerrarSesionMenuItem.setText("Cerrar Sesión");
cerrarSesionMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionMenuItemActionPerformed(evt);
    }
});
fileMenu.add(cerrarSesionMenuItem);

exitMenuItem.setMnemonic('x');
exitMenuItem.setText("Exit");
exitMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        exitMenuItemActionPerformed(evt);
    }
});

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

editMenu.setMnemonic('e');
editMenu.setText("Crear");

usuarioMenuItem.setMnemonic('t');
usuarioMenuItem.setText("Usuario");
usuarioMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        usuarioMenuItemActionPerformed(evt);
    }
});
editMenu.add(usuarioMenuItem);

clienteMenuItem.setMnemonic('y');
clienteMenuItem.setText("Cliente");
clienteMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        clienteMenuItemActionPerformed(evt);
    }
});
editMenu.add(clienteMenuItem);

vehiculoMenuItem.setText("Vehiculo");
vehiculoMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        vehiculoMenuItemActionPerformed(evt);
    }
});

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }
});
editMenu.add(vehiculoMenuItem);

contratoMenuItem.setMnemonic('p');
contratoMenuItem.setText("Contrato");
contratoMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        contratoMenuItemActionPerformed(evt);
    }
});
editMenu.add(contratoMenuItem);

menuBar.add(editMenu);

helpMenu.setMnemonic('h');
helpMenu.setText("Ticket");

entradaMenuItem.setMnemonic('c');
entradaMenuItem.setText("Emitir ticket entrada");
entradaMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        entradaMenuItemActionPerformed(evt);
    }
});
helpMenu.add(entradaMenuItem);

salidaMenuItem.setMnemonic('a');
salidaMenuItem.setText("Emitir ticket salida");

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

salidaMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        salidaMenuItemActionPerformed(evt);
    }
});
helpMenu.add(salidaMenuItem);

menuBar.add(helpMenu);

jMenu1.setText("Gestion");

facturaMenuItem.setText("Facturas");
facturaMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        facturaMenuItemActionPerformed(evt);
    }
});
jMenu1.add(facturaMenuItem);

sitiosMenuItem.setText("Sitios");
sitiosMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        sitiosMenuItemActionPerformed(evt);
    }
});
jMenu1.add(sitiosMenuItem);

menuBar.add(jMenu1);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
setJMenuBar(menuBar);
```

```
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 632, Short.MAX_VALUE)
```

```
);
```

```
layout.setVerticalGroup(
```

```
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 411, Short.MAX_VALUE)
```

```
);
```

```
pack();
```

```
// </editor-fold>
```

```
private void exitMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    System.exit(0);
```

```
}
```

```
private void loginMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    cerrarVentanas();
```

```
    if (Administrador.instance == null) {
```

```
        ventanaLogin.setVisible(true);
```

```
        inicioSesion = true;
```

```
    } else {
```

```
        JOptionPane.showMessageDialog(this, "Ya existe un usuario que inicio sesión", "Error",
JOptionPane.ERROR_MESSAGE);
```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

}

private void cerrarSesionMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (inicioSesion) {
        Administrador.instance = null;
        cerrarVentanas();
        JOptionPane.showMessageDialog(this, "Sesión cerrada", "Mensaje", JOptionPane.INFORMATION_MESSAGE);
        inicioSesion = false;
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void usuarioMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        cerrarVentanas();
        if (Administrador.instance.getEnumTipoAdministrador() == EnumTipoAdministrador.COMPLETO) {

            ventanaCrearUsuario.setVisible(true);
        } else {
            JOptionPane.showMessageDialog(this, "NO tiene acceso", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    } else {
        cerrarVentanas();
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }

}

private void clienteMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaCrearCliente.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void vehiculoMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaVehiculo.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private void entradaMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaEntradaTicket.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void salidaMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaSalidaTicket.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void contratoMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaContrato.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

}

private void facturaMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaGestionFacturas.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void sitiosMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    if (Administrador.instance != null) {
        ventanaSitios.setVisible(true);
    } else {
        JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
 */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new VentanaPrincipal().setVisible(true);
    }
});
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
// Variables declaration - do not modify

private javax.swing.JMenuItem cerrarSesionMenuItem;

private javax.swing.JMenuItem clienteMenuItem;

private javax.swing.JMenuItem contratoMenuItem;

private javax.swing.JDesktopPane desktopPane;

private javax.swing.JMenu editMenu;

private javax.swing.JMenuItem entradaMenuItem;

private javax.swing.JMenuItem exitMenuItem;

private javax.swing.JMenuItem facturaMenuItem;

private javax.swing.JMenu fileMenu;

private javax.swing.JMenu helpMenu;

private javax.swing.JMenu jMenuItem1;

private javax.swing.JMenuItem loginMenuItem;

private javax.swing.JMenuBar menuBar;

private javax.swing.JMenuItem salidaMenuItem;

private javax.swing.JMenuItem sitiosMenuItem;

private javax.swing.JMenuItem usuarioMenuItem;

private javax.swing.JMenuItem vehiculoMenuItem;

// End of variables declaration

}

    • VentanaContrato:

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

package ups.edu.ec.vista;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import java.text.DecimalFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.TimeZone;
import javax.swing.ButtonModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorCliente;
import ups.edu.ec.controlador.ControladorContrato;
import ups.edu.ec.controlador.ControladorSitio;
import ups.edu.ec.modelo.Cliente;
import ups.edu.ec.modelo.Contrato;
import ups.edu.ec.modelo.EnumClienteContrato;
import ups.edu.ec.modelo.EnumTipoPrecioContrato;
import ups.edu.ec.modelo.Sitio;

/**
 *
 * @author User
 */
public class VentanaContrato extends javax.swing.JInternalFrame {

    private ControladorContrato controladorContrato;
    private ControladorSitio controladorSitio;
    private ControladorCliente controladorCliente;

    private Sitio sitio;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private Contrato contrato;

private Cliente cliente;


private static DecimalFormat decimalFormat;

/**
 * Creates new form VentanaContrato
 */
public VentanaContrato(ControladorContrato controladorContrato, ControladorSitio controladorSitio,
    ControladorCliente controladorCliente) {
    initComponents();

    this.controladorContrato = controladorContrato;
    this.controladorSitio = controladorSitio;
    this.controladorCliente = controladorCliente;

    contrato = new Contrato();
    cliente = new Cliente();

    //decimalFormat = new DecimalFormat("^(\d)+(\.)(\d){0,2}$");
    limpiar();
}

public void cargarCodigo() {

    txtCodigo.setText(String.valueOf(controladorContrato.generarCodigo()));
}

public void cargarSitio() {

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

sitio = controladorSitio.encontrarSitioDesocupado();
txtNumeroSitio.setText(String.valueOf(sitio.getCodigo()));
}

public void limpiar() {
    contrato = new Contrato();
    cliente = new Cliente();
    cargarCodigo();
    jDateChooserInicio.cleanup();
    jDateChooserFin.cleanup();
    btnGroupPrecio.clearSelection();
    cargarSitio();
    txtFormattedCedula.setValue("");
    txtFormattedCedula.setEditable(true);
    txtSubtotal.setText("");
    buttonGroupDescuento.clearSelection();
    txtTotal.setText("");
    buttonGroupPagado.clearSelection();
    Date date = java.util.Calendar.getInstance().getTime();
    jDateChooserInicio.setDate(date);
    jDateChooserFin.setDate(date);
    llenarTblContratos();
    tblContratos.clearSelection();

    btnCrear.setEnabled(true);
    btnActualizar.setEnabled(false);
    btnEliminar.setEnabled(false);
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void llenarTblContratos() {
    var contratos = (List<Contrato>) controladorContrato.getListaGenerica();

    DefaultTableModel modelo = (DefaultTableModel) tblContratos.getModel();
    modelo.setRowCount(0);

    for (Contrato con : contratos) {
        if (con.getFechaFin().before(new Date())) {
            controladorSitio.cambiarAEstadoDescupado(con.getSitio());
            con.setTotal(con.getTotal() + (con.getTotal() * 0.10));
            controladorContrato.update(con, con);
        }
        String pagado = "NO";
        if (con.isPago()) {
            pagado = "SI";
        }
        Object[] row = {con.getCodigo(), con.getFechaInicio().toString(), con.getFechaFin().toString(),
            con.getTotal(), pagado, con.getSitio().getCodigo(), con.getCliente().getCedula()};

        modelo.addRow(row);
    }

    tblContratos.setModel(modelo);
}

public void calcularSubtotal() {
    double subtotal = controladorContrato.obtenerSubtotal(sitio, jDateChooserInicio.getDate(),

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
jDateChooserFin.getDate(), contrato.getEnumTipoPrecioContrato());
```

```
String formateado = String.format("%.2f", subtotal);
```

```
//subtotal = decimalFormat.format(subtotal);
```

```
txtSubtotal.setText(formateado);
```

```
}
```

```
public double casteoString(String numero) {
```

```
numero = numero.replace(",", ".");
```

```
return Double.valueOf(numero);
```

```
}
```

```
/**
```

```
* This method is called from within the constructor to initialize the form.
```

```
* WARNING: Do NOT modify this code. The content of this method is always
```

```
* regenerated by the Form Editor.
```

```
*/
```

```
@SuppressWarnings("unchecked")
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">
```

```
private void initComponents() {
```

```
btnGroupPrecio = new javax.swing.ButtonGroup();
```

```
buttonGroupDescuento = new javax.swing.ButtonGroup();
```

```
buttonGroupPagado = new javax.swing.ButtonGroup();
```

```
jPanel1 = new javax.swing.JPanel();
```

```
jLabel1 = new javax.swing.JLabel();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
radioBtnDia = new javax.swing.JRadioButton();
radioBtnSemana = new javax.swing.JRadioButton();
radioBtnMes = new javax.swing.JRadioButton();
txtSubtotal = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
txtFormattedCedula = new javax.swing.JFormattedTextField();
txtCodigo = new javax.swing.JTextField();
jDateChooserFin = new com.toedter.calendar.JDateChooser();
jDateChooserInicio = new com.toedter.calendar.JDateChooser();
txtNumeroSitio = new javax.swing.JTextField();
radioBtnSi = new javax.swing.JRadioButton();
radioBtnNo = new javax.swing.JRadioButton();
jLabel10 = new javax.swing.JLabel();
txtTotal = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
radioBtnPagadoSi = new javax.swing.JRadioButton();
radioBtnPagadoNo = new javax.swing.JRadioButton();
btnCrear = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminar = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jScrollPane1 = new javax.swing.JScrollPane();
tblContratos = new javax.swing.JTable();

setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Contrato");
addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameActivated(evt);
    }
    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameClosed(evt);
    }
    public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameOpened(evt);
    }
});

jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0), 2, true));

jLabel1.setText("Código:");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jLabel2.setText("Fecha Inicio:");

jLabel3.setText("Fecha Fin:");

jLabel4.setText("Precio:");

btnGroupPrecio.add(radioBtnDia);
radioBtnDia.setText("Día");
radioBtnDia.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnDiaActionPerformed(evt);
    }
});

btnGroupPrecio.add(radioBtnSemana);
radioBtnSemana.setText("Semana");
radioBtnSemana.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnSemanaActionPerformed(evt);
    }
});

btnGroupPrecio.add(radioBtnMes);
radioBtnMes.setText("Mes");
radioBtnMes.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnMesActionPerformed(evt);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
});
```

```
txtSubtotal.setEditable(false);
```

```
txtSubtotal.setBackground(new java.awt.Color(102, 102, 102));
```

```
jLabel6.setText("Subtotal:");
```

```
jLabel5.setText("$");
```

```
jLabel7.setText("Descuento:");
```

```
jLabel8.setText("Número Sitio:");
```

```
jLabel9.setText("Cliente:");
```

```
try {
```

```
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("0#####-#"));
```

```
    } catch (java.text.ParseException ex) {
```

```
        ex.printStackTrace();
```

```
    }
```

```
txtCodigo.setEditable(false);
```

```
txtCodigo.setBackground(new java.awt.Color(102, 102, 102));
```

```
txtNumeroSitio.setEditable(false);
```

```
txtNumeroSitio.setBackground(new java.awt.Color(102, 102, 102));
```

```
buttonGroupDescuento.add(radioBtnSi);
```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

radioBtnSi.setText("SI");

radioBtnSi.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        radioBtnSiActionPerformed(evt);

    }

});

buttonGroupDescuento.add(radioBtnNo);

radioBtnNo.setText("NO");

radioBtnNo.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        radioBtnNoActionPerformed(evt);

    }

});

jLabel10.setText("Total:");

txtTotal.setEditable(false);

txtTotal.setBackground(new java.awt.Color(102, 102, 102));

jLabel11.setText("$");

jLabel12.setText("Pagado:");

buttonGroupPagado.add(radioBtnPagadoSi);

radioBtnPagadoSi.setText("SI");

radioBtnPagadoSi.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        radioBtnPagadoSiActionPerformed(evt);

    }

});

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }
});

buttonGroupPagado.add(radioBtnPagadoNo);
radioBtnPagadoNo.setText("NO");
radioBtnPagadoNo.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnPagadoNoActionPerformed(evt);
    }
});

btnCrear.setBackground(new java.awt.Color(0, 0, 255));
btnCrear.setText("Crear");
btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

btnActualizar.setText("Actualizar");
btnActualizar.setEnabled(false);
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

btnEliminar.setText("Eliminar");
btnEliminar.setEnabled(false);

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGap(136, 136, 136)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(txtFormattedCedula)

    .addGroup(jPanel1Layout.createSequentialGroup())

        .addComponent(radioBtnSi)

        .addGap(18, 18, 18)

        .addComponent(radioBtnNo)

        .addGap(33, 33, 33)))

.addGroup(jPanel1Layout.createSequentialGroup())

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel3)

        .addComponent(jLabel2))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(jDateChooserFin, javax.swing.GroupLayout.DEFAULT_SIZE, 171, Short.MAX_VALUE)

        .addComponent(jDateChooserInicio, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

.addGroup(jPanel1Layout.createSequentialGroup())

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jLabel4)

        .addComponent(jLabel8)

        .addComponent(jLabel7))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addComponent(radioBtnDia)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(radioBtnSemana)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(radioBtnMes))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(21, 21, 21)

.addComponent(txtNumeroSitio))))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(36, 36, 36)

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup())

.addComponent(jLabel10)

.addGap(154, 154, 154)

.addComponent(jLabel11,                                     javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGap(6, 6, 6)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(txtTotal, javax.swing.GroupLayout.DEFAULT_SIZE, 160, Short.MAX_VALUE)

.addComponent(txtSubtotal)))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(jLabel12)

.addGap(138, 138, 138))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())


.addComponent(btnCrear)

.addGap(37, 37, 37)

.addComponent(btnActualizar)

```

```
.addGap(41, 41, 41)))  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
.addGroup(jPanel1Layout.createSequentialGroup()  
.addComponent(radioBtnPagadoSi)  
.addGap(18, 18, 18)  
.addComponent(radioBtnPagadoNo))  
.addGroup(jPanel1Layout.createSequentialGroup()  
.addComponent(btnEliminar)  
.addGap(36, 36, 36)  
.addComponent(btnCancelar)))  
.addGap(22, 22, 22)))  
.addGap(41, 41, 41))  
);  
jPanel1Layout.setVerticalGroup(  
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
.addGroup(jPanel1Layout.createSequentialGroup()  
.addGap(18, 18, 18)  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
.addComponent(jLabel1)  
.addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))  
.addGap(18, 18, 18)  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
.addComponent(jLabel2)  
.addComponent(jDateChooserInicio, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))  
.addGap(18, 18, 18)  
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)  
.addComponent(jLabel3)
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addComponent(jDateChooserFin,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(radioBtnDia)

.addComponent(radioBtnSemana)

.addComponent(radioBtnMes))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(txtNumeroSitio,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(txtFormattedCedula,                            javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(20, 20, 20)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtSubtotal,                                    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6)

.addComponent(jLabel5))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(radioBtnSi)

.addComponent(radioBtnNo))

.addGap(18, 18, 18)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel10)
        .addComponent(txtTotal, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    .addComponent(jLabel11, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel12)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(radioBtnPagadoSi)
        .addComponent(radioBtnPagadoNo)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 41, Short.MAX_VALUE)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnCrear)
    .addComponent(btnActualizar)
    .addComponent(btnEliminar)
    .addComponent(btnCancelar))
.addGap(20, 20, 20)
);

tblContratos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Código", "Fecha inicio", "Fecha entrada", "Total", "Pagado", "Sitio", "Cliente"
    }
) {

```





	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(18, 18, 18)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 630, Short.MAX_VALUE)

        .addContainerGap())

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

                .addComponent(jPanel1,                                javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jScrollPane1))

            .addContainerGap(18, Short.MAX_VALUE))

);

pack();
} // </editor-fold>

private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    int codigo = Integer.valueOf(txtCodigo.getText());

    Date fechalnicio = jDateChooserInicio.getDate();

    Date fechaFin = jDateChooserFin.getDate();

    String cedula = (String) txtFormattedCedula.getValue();

    String subtotal = txtSubtotal.getText();

    String total = txtTotal.getText();

    if (cedula.isBlank()) {

```

 <b>UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR</b>	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {

    cliente.setCedula(cedula);

    cliente = (Cliente) controladorCliente.read(cliente);


    if (cliente != null) {
        contrato.setCodigo(codigo);
        contrato.setFechaInicio(fechaInicio);
        contrato.setFechaFin(fechaFin);
        contrato.setCliente(cliente);
        contrato.setSitio(sitio);

        contrato.setSubtotal(casteoString(subtotal));
        contrato.setTotal(casteoString(total));

        if (controladorContrato.create(contrato)) {
            JOptionPane.showMessageDialog(this, "Contrato creado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
            controladorSitio.cambiarAEstadoContratado(sitio);
            cliente.setEnumClienteContrato(EnumClienteContrato.SI);
            controladorCliente.update(cliente, cliente);
            limpiar();
        } else {
            JOptionPane.showMessageDialog(this, "Contrato ya existente", "Error", JOptionPane.ERROR_MESSAGE);
        }

    } else {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
JOptionPane.showMessageDialog(this, "Cliente no encontrado", "Error", JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
}
```

```
}
```

```
private void radioBtnDiaActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:
```

```
contrato.setEnumTipoPrecioContrato(EnumTipoPrecioContrato.DIA);
```

```
sitio.setPrecioHora(0.0023);
```

```
controladorSitio.update(sitio, sitio);
```

```
calcularSubtotal();
```

```
}
```

```
private void radioBtnSemanaActionPerformed(java.awt.event.ActionEvent evt) {
```

```
// TODO add your handling code here:
```

```
contrato.setEnumTipoPrecioContrato(EnumTipoPrecioContrato.SEMANA);
```

```
sitio.setPrecioHora(0.002);
```

```
controladorSitio.update(sitio, sitio);
```

```
calcularSubtotal();
```

```
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private void radioBtnMesActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    contrato.setEnumTipoPrecioContrato(EnumTipoPrecioContrato.MES);
    sitio.setPrecioHora(0.0017);
    controladorSitio.update(sitio, sitio);

    calcularSubtotal();
}

private void radioBtnSiActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    contrato.setDescuento(true);
    calcularSubtotal();

    double totalNumero = controladorContrato.obtenerTotal(casteoString(txtSubtotal.getText()));
    String formateado = String.format("%.2f", totalNumero);

    //subtotal = decimalFormat.format(subtotal);
    txtTotal.setText(formateado);
}

private void radioBtnNoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    contrato.setDescuento(false);

    calcularSubtotal();
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtTotal.setText(txtSubtotal.getText());
}

private void radioBtnPagadoSiActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    calcularSubtotal();
    contrato.setPago(true);
}

private void radioBtnPagadoNoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    calcularSubtotal();
    contrato.setPago(false);
}

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void tblContratosMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int fila = tblContratos.getSelectedRow();

    contrato = new Contrato();

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

contrato.setCodigo((int) (tblContratos.getValueAt(fila, 0)));

contrato = (Contrato) controladorContrato.read(contrato);

txtCodigo.setText(String.valueOf(contrato.getCodigo()));
jDateChooserInicio.setDate(contrato.getFechaInicio());
jDateChooserFin.setDate(contrato.getFechaFin());

switch (contrato.getEnumTipoPrecioContrato()) {
    case DIA -> {
        btnGroupPrecio.setSelected((ButtonModel) radioBtnDia.getModel(), true);
        break;
    }
    case MES -> {
        btnGroupPrecio.setSelected((ButtonModel) radioBtnMes.getModel(), true);
        break;
    }

    default -> {
        btnGroupPrecio.setSelected((ButtonModel) radioBtnSemana.getModel(), true);
        break;
    }
}

sitio = contrato.getSitio();
txtNumeroSitio.setText(String.valueOf(sitio.getCodigo()));
cliente = contrato.getCliente();
txtFormattedCedula.setValue(cliente.getCedula());

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtFormattedCedula.setEditable(false);

txtSubtotal.setText(String.valueOf(contrato.getSubtotal()));

if (contrato.isDescuento()) {
    buttonGroupDescuento.setSelected(radioBtnSi.getModel(), true);
} else {
    buttonGroupDescuento.setSelected(radioBtnNo.getModel(), true);
}

txtTotal.setText(String.valueOf(contrato.getTotal()));

if (contrato.isPago()) {
    buttonGroupPagado.setSelected(radioBtnPagadoSi.getModel(), true);
} else {
    buttonGroupPagado.setSelected(radioBtnPagadoNo.getModel(), true);
}

btnCrear.setEnabled(false);
btnActualizar.setEnabled(true);
btnEliminar.setEnabled(true);

}

private void btnActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    contrato.setSubtotal(casteoString(txtSubtotal.getText()));
    contrato.setTotal(casteoString(txtTotal.getText()));

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

if (controladorContrato.update(contrato, contrato)) {
    JOptionPane.showMessageDialog(this, "Contrato actualizado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    controladorSitio.cambiarAEstadoContratado(contrato.getSitio());
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}

}

private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    cliente = contrato.getCliente();

    if (controladorContrato.delete(contrato)) {
        JOptionPane.showMessageDialog(this, "Contrato eliminado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
        controladorSitio.cambiarAEstadoDescupado(contrato.getSitio());
        cliente.setEnumClienteContrato(EnumClienteContrato.NO);
        controladorCliente.update(cliente, cliente);
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
    }

}

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void formInternalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnEliminar;
private javax.swing.ButtonGroup btnGroupPrecio;
private javax.swing.ButtonGroup buttonGroupDescuento;
private javax.swing.ButtonGroup buttonGroupPagado;
private com.toedter.calendar.JDateChooser jDateChooserFin;
private com.toedter.calendar.JDateChooser jDateChooserInicio;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JRadioButton radioBtnDia;
private javax.swing.JRadioButton radioBtnMes;
private javax.swing.JRadioButton radioBtnNo;
private javax.swing.JRadioButton radioBtnPagadoNo;
private javax.swing.JRadioButton radioBtnPagadoSi;
private javax.swing.JRadioButton radioBtnSemana;
private javax.swing.JRadioButton radioBtnSi;
private javax.swing.JTable tblContratos;
private javax.swing.JTextField txtCodigo;
private javax.swing.JFormattedTextField txtFormattedCedula;
private javax.swing.JTextField txtNumeroSitio;
private javax.swing.JTextField txtSubtotal;
private javax.swing.JTextField txtTotal;

// End of variables declaration
}

```

- VentanaCrearCliente:

/\*

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package ups.edu.ec.vista;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import ups.edu.ec.controlador.ControladorCliente;

import ups.edu.ec.modelo.Administrador;

import ups.edu.ec.modelo.Cliente;

import ups.edu.ec.modelo.EnumClienteContrato;

import ups.edu.ec.modelo.EnumTipoAdministrador;

/\*\*

\*

\* @author User

\*/

public class VentanaCrearCliente extends javax.swing.JInternalFrame {

private ControladorCliente controladorCliente;

private Cliente cliente;

/\*\*

\* Creates new form VentanaCrearCliente

\*/

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public VentanaCrearCliente(ControladorCliente controladorCliente) {
    initComponents();

    this.controladorCliente = controladorCliente;

    cliente = new Cliente();
    limpiar();
}

public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorCliente.generarCodigo()));
}

public void limpiar() {
    txtNombre.setText("");
    txtApellido.setText("");
    txtFormattedCedula.setValue("");
    jDateChooser1.setDate(new Date());

    cargarCodigo();
    txtFormattedTelefono.setValue("");
    cargarBotones();
    tblClientes.clearSelection();
    txtFormattedCedula.setEditable(true);

    llenarTblClientes();
}

public void cargarBotones() {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

btnCrear.setEnabled(true);

btnActualizar.setEnabled(false);

btnEliminar.setEnabled(false);

btnCancelar.setEnabled(true);
}

public void cargarBotonesCambio() {
    btnCrear.setEnabled(false);
    btnActualizar.setEnabled(true);
    btnEliminar.setEnabled(true);
    btnCancelar.setEnabled(true);
}

public void llenarTblClientes() {
    DefaultTableModel modelo = (DefaultTableModel) tblClientes.getModel();
    modelo.setRowCount(0);

    for (Cliente admin : (List<Cliente>) controladorCliente.getListaGenerica()) {

        Object[] row = {admin.getCedula(), admin.getNombre(), admin.getApellido(),
            admin.getEnumClienteContrato().toString()};

        modelo.addRow(row);

    }

    tblClientes.setModel(modelo);
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jDateChooser1 = new com.toedter.calendar.JDateChooser();
    jLabel6 = new javax.swing.JLabel();
    txtFormattedTelefono = new javax.swing.JFormattedTextField();
    txtFormattedCedula = new javax.swing.JFormattedTextField();
    txtCodigo = new javax.swing.JTextField();
    txtNombre = new javax.swing.JTextField();
    txtApellido = new javax.swing.JTextField();
    btnCrear = new javax.swing.JButton();
    btnActualizar = new javax.swing.JButton();
    btnEliminar = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblClientes = new javax.swing.JTable();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

setClosable(true);

setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);

setTitle("Clientes");

addInternalFrameListener(new javax.swing.event.InternalFrameListener() {

    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {

        formInternalFrameActivated(evt);

    }

    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {

    }

    public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {

    }

    public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {

    }

    public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {

    }

    public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {

    }

    public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {

        formInternalFrameOpened(evt);

    }

});

jLabel1.setText("Nombre:");

jLabel2.setText("Apellido:");

jLabel3.setText("Cédula:");

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jLabel4.setText("Código:");

jLabel5.setText("Nacimiento:");

jLabel6.setText("Teléfono:");

try {
    txtFormattedTelefono.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("###-###-####"));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

try {
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("0#####-#"));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

txtCodigo.setEditable(false);
txtCodigo.setBackground(new java.awt.Color(102, 102, 102));

btnCrear.setBackground(new java.awt.Color(51, 51, 255));
btnCrear.setText("Crear");
btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

btnActualizar.setText("Actualizar");
btnActualizar.setEnabled(false);
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

btnEliminar.setBackground(new java.awt.Color(255, 51, 51));
btnEliminar.setText("Eliminar");
btnEliminar.setEnabled(false);
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarActionPerformed(evt);
    }
});

btnCancelar.setText("Cancelar");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(16, 16, 16)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel4)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(1, 1, 1)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jLabel3)

.addComponent(jLabel2)

.addComponent(jLabel6)))

.addComponent(btnCrear, javax.swing.GroupLayout.Alignment.TRAILING))

.addGap(100, 100, 100)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 203, Short.MAX_VALUE)

.addComponent(txtFormattedTelefono)

.addComponent(txtFormattedCedula)

.addComponent(txtCodigo)

.addComponent(txtNombre)

.addComponent(txtApellido)))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(112, 112, 112)

.addComponent(btnActualizar)

.addGap(41, 41, 41)

.addComponent(btnEliminar)

.addGap(33, 33, 33)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addComponent(btnCancelar)))

    .addContainerGap(55, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(11, 11, 11)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(txtCodigo,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(txtNombre,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addComponent(jLabel2))

.addComponent(txtApellido,                                javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,                                javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

.addComponent(jLabel3)

.addGap(18, 18, 18)

.addComponent(jLabel5))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup())

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addComponent(txtFormattedCedula,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addComponent(jDateChooser1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))

        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel6)

        .addComponent(txtFormattedTelefono,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(48, 48, 48)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(btnCrear)

        .addComponent(btnActualizar)

        .addComponent(btnEliminar)

        .addComponent(btnCancelar))

        .addContainerGap(90, Short.MAX_VALUE))


    );

tblClientes.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {

        "Cedula", "Nombre", "Apellido", "Contrato"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class
    };

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```


boolean[] canEdit = new boolean [] {
    false, false, false, false
};

public Class getColumnClass(int columnIndex) {
    return types [columnIndex];
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});
tblClientes.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblClientesMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblClientes);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        )
);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addGap(16, 16, 16))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(16, 16, 16))
    );

    pack();
} // </editor-fold>


private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int codigo = Integer.valueOf(txtCodigo.getText());
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    Calendar fechaNacimiento = jDateChooser1.getCalendar();

    String telefono = (String) txtFormattedTelefono.getValue();

    if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank() || fechaNacimiento == null
        || telefono.isBlank()) {

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {
    cliente = new Cliente();
    cliente.setCodigo(codigo);
    cliente.setNombre(nombre);
    cliente.setApellido(apellido);
    cliente.setCedula(cedula);
    cliente.setFechaNacimiento(fechaNacimiento);
    cliente.setNumeroTelefono(telefono);
    cliente.setEnumClienteContrato(EnumClienteContrato.NO);

    if (controladorCliente.create(cliente)) {
        JOptionPane.showMessageDialog(this, "Cliente creado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Cliente ya existente", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
}

private void btnActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int codigo = Integer.valueOf(txtCodigo.getText());
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    Calendar fechaNacimiento = jDateChooser1.getCalendar();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

String telefono = (String) txtFormattedTelefono.getValue();

if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank() || fechaNacimiento == null
    || telefono.isBlank()) {
    JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {

    cliente = new Cliente();
    cliente.setCodigo(codigo);
    cliente.setNombre(nombre);
    cliente.setApellido(apellido);
    cliente.setCedula(cedula);
    cliente.setFechaNacimiento(fechaNacimiento);
    cliente.setNumeroTelefono(telefono);
    cliente.setEnumClienteContrato(EnumClienteContrato.NO);


    if (controladorCliente.update(cliente, cliente)) {
        JOptionPane.showMessageDialog(this, "Cliente actualizado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
        limpiar();

    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
// TODO add your handling code here:

String cedula = (String) txtFormattedCedula.getValue();

cliente = new Cliente();
cliente.setCedula(cedula);

if (controladorCliente.delete(cliente)) {
    JOptionPane.showMessageDialog(this, "Cliente eliminado con éxito", "Mensaje",
    JOptionPane.INFORMATION_MESSAGE);
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}
}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void tblClientesMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int fila = tblClientes.getSelectedRow();

    String cedula = (String) tblClientes.getValueAt(fila, 0);
    cliente = new Cliente();
    cliente.setCedula(cedula);
    cliente = (Cliente) controladorCliente.read(cliente);
}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtCodigo.setText(String.valueOf(cliente.getCodigo()));
txtNombre.setText(cliente.getNombre());
txtApellido.setText(cliente.getApellido());
txtFormattedCedula.setValue(cliente.getCedula());
txtFormattedCedula.setEditable(false);
txtFormattedTelefono.setValue(cliente.getNumeroTelefono());
jDateChooser1.setCalendar(cliente.getFechaNacimiento());

cargarBotonesCambio();
}

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnEliminar;
private com.toedter.calendar.JDateChooser jDateChooser1;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblClientes;
private javax.swing.JTextField txtApellido;
private javax.swing.JTextField txtCodigo;
private javax.swing.JFormattedTextField txtFormattedCedula;
private javax.swing.JFormattedTextField txtFormattedTelefono;
private javax.swing.JTextField txtNombre;

// End of variables declaration
}


    • VentanaCrearUsuario:

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

package ups.edu.ec.vista;

import java.lang.reflect.Constructor;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import javax.swing.JOptionPane;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import javax.swing.table.DefaultTableModel;

import ups.edu.ec.controlador.ControladorAdministrador;

import ups.edu.ec.modelo.Administrador;

import ups.edu.ec.modelo.EnumTipoAdministrador;

/**
 *
 * @author User
 */
public class VentanaCrearUsuario extends javax.swing.JFrame {

    private Administrador administrador;

    private ControladorAdministrador controladorAdministrador;

    private String keyUnlock;

    /**
     * Creates new form VentanaCrearUsuario
     */
    public VentanaCrearUsuario(ControladorAdministrador controladorAdministrador, String keyUnlock) {
        initComponents();

        this.controladorAdministrador = controladorAdministrador;

        crearAdministrador();

        cargarCodigo();

        llenarTblAdministradores();

        this.keyUnlock = keyUnlock;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

public void crearAdministrador() {
    try {
        Constructor<Administrador> constructor = Administrador.class.getDeclaredConstructor();
        constructor.setAccessible(true);
        administrador = constructor.newInstance();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorAdministrador.generarCodigo()));
}

public void limpiar() {
    txtNombre.setText("");
    txtApellido.setText("");
    txtFormattedCedula.setValue("");
    jDateChooser1.setDate(new Date());
    txtPassword.setText("");
    txtCorreo.setText("");
    cargarCodigo();
    txtFormattedTelefono.setValue("");
    cargarBotones();
    tblUsuarios.clearSelection();
    txtFormattedCedula.setEditable(true);
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

llenarTblAdministradores();
}

public void cargarBotones() {
    btnCrear.setEnabled(true);
    btnActualizar.setEnabled(false);
    btnEliminar.setEnabled(false);
    btnCancelar.setEnabled(true);
}

public void cargarBotonesCambio() {
    btnCrear.setEnabled(false);
    btnActualizar.setEnabled(true);
    btnEliminar.setEnabled(true);
    btnCancelar.setEnabled(true);
}

public void llenarTblAdministradores() {
    DefaultTableModel modelo = (DefaultTableModel) tblUsuarios.getModel();
    modelo.setRowCount(0);

    for (Administrador admin : (List<Administrador>) controladorAdministrador.getListaGenerica()) {

        if (admin.getEnumTipoAdministrador() != EnumTipoAdministrador.COMPLETO) {
            Object[] row = {admin.getCedula(), admin.getNombre(), admin.getApellido(),
                admin.getCorreo()};

            modelo.addRow(row);
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

tblUsuarios.setModel(modelo);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    tblUsuarios = new javax.swing.JTable();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jDateChooser1 = new com.toedter.calendar.JDateChooser();
    jLabel6 = new javax.swing.JLabel();
    txtFormattedTelefono = new javax.swing.JFormattedTextField();
    txtFormattedCedula = new javax.swing.JFormattedTextField();
    txtCodigo = new javax.swing.JTextField();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


```

txtNombre = new javax.swing.JTextField();
txtApellido = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
txtPassword = new javax.swing.JPasswordField();
txtCorreo = new javax.swing.JTextField();
btnCrear = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminar = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();

setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Usuario");
addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameActivated(evt);
    }
    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

formInternalFrameOpened(evt);

}

});

tblUsuarios.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {


    },
    new String [] {
        "Cedula", "Nombre", "Apellido", "Correo"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

tblUsuarios.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        tblUsuariosMouseClicked(evt);

    }

});

jScrollPane1.setViewportViewView(tblUsuarios);

jLabel1.setText("Nombre:");

jLabel2.setText("Apellido:");

jLabel3.setText("Cédula:");

jLabel4.setText("Código:");

jLabel5.setText("Nacimiento:");

jLabel6.setText("Teléfono:");

try {
    txtFormattedTelefono.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("###-###-####"));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

try {
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("0#####-#"));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtCodigo.setEditable(false);

txtCodigo.setBackground(new java.awt.Color(102, 102, 102));


jLabel7.setText("Correo:");



jLabel8.setText("Password:");


btnCrear.setBackground(new java.awt.Color(51, 51, 255));
btnCrear.setText("Crear");
btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

btnActualizar.setText("Actualizar");
btnActualizar.setEnabled(false);
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

btnEliminar.setBackground(new java.awt.Color(255, 51, 51));
btnEliminar.setText("Eliminar");
btnEliminar.setEnabled(false);
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        btnEliminarActionPerformed(evt);
    }
});

btnCancelar.setText("Cancelar");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel4)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(16, 16, 16)
                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel5)
                        .addComponent(jLabel3)
                        .addComponent(jLabel2)
                        .addComponent(jLabel6)
                    )
                )
            )
        )
);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addComponent(jLabel7)
        .addComponent(jLabel8))))
    .addGap(100, 100, 100)
    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 203, Short.MAX_VALUE)
        .addComponent(txtFormattedTelefono)
        .addComponent(txtFormattedCedula)
        .addComponent(txtCodigo)
        .addComponent(txtNombre)
        .addComponent(txtApellido)
        .addComponent(txtPassword)
        .addComponent(txtCorreo)))
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(43, 43, 43)
        .addComponent(btnCrear)
        .addGap(34, 34, 34)
        .addComponent(btnActualizar)
        .addGap(34, 34, 34)
        .addComponent(btnEliminar)
        .addGap(35, 35, 35)
        .addComponent(btnCancelar)))
    .addContainerGap(51, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(11, 11, 11)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)
    .addComponent(txtNombre, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addComponent(jLabel2))
.addComponent(txtApellido, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel3)
        .addGap(18, 18, 18)
        .addComponent(jLabel5))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addComponent(txtFormattedCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel6)
    .addComponent(txtFormattedTelefono, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel7)

    .addComponent(txtCorreo,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(jLabel8)

    .addComponent(txtPassword,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

    .addComponent(btnCrear)

    .addComponent(btnActualizar)

    .addComponent(btnEliminar)

    .addComponent(btnCancelar))


.addGap(44, 44, 44))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGap(18, 18, 18)
            .addComponent(jScrollPane1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

504,

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addGap(21, 21, 21))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(15, 15, 15)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jPanel1,                                javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jScrollPane1))
            .addContainerGap(33, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int codigo = Integer.valueOf(txtCodigo.getText());
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    Calendar fechaNacimiento = jDateChooser1.getCalendar();
    String email = txtCorreo.getText();
    String password = txtPassword.getText();
    String telefono = (String) txtFormattedTelefono.getValue();

    if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank() || fechaNacimiento == null

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

|| email.isBlank() || password.isBlank() || telefono.isBlank()) {
    JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {

    String passwordEncriptado = controladorAdministrador.encriptarPassword(keyUnlock, password);
    administrador.setPassword(passwordEncriptado);

    System.out.println(passwordEncriptado);
    administrador.setCodigo(codigo);
    administrador.setNombre(nombre);
    administrador.setApellido(apellido);
    administrador.setCedula(cedula);
    administrador.setFechaNacimiento(fechaNacimiento);
    administrador.setCorreo(email);

    administrador.setNumeroTelefono(telefono);
    administrador.setEnumTipoAdministrador(EnumTipoAdministrador.SIMPLE);

    if (controladorAdministrador.create(administrador)) {
        JOptionPane.showMessageDialog(this, "Administrador creado con exito", "Mensaje",
        JOptionPane.INFORMATION_MESSAGE);
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Administrador ya existente", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

```
private void tblUsuariosMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
```

```
int fila = tblUsuarios.getSelectedRow();
```

```
String cedula = (String) tblUsuarios.getValueAt(fila, 0);
```

```
administrador.setCedula(cedula);
```

```
administrador = (Administrador) controladorAdministrador.read(administrador);
```

```
String passwordDesencriptado = controladorAdministrador.desencriptarPassword(keyUnlock,
administrador.getPassword());
```

```
txtCodigo.setText(String.valueOf(administrador.getCodigo()));
```

```
txtNombre.setText(administrador.getNombre());
```

```
txtApellido.setText(administrador.getApellido());
```

```
txtFormattedCedula.setValue(administrador.getCedula());
```

```
txtFormattedCedula.setEditable(false);
```

```
txtCorreo.setText(administrador.getCorreo());
```

```
txtPassword.setText(passwordDesencriptado);
```

```
txtFormattedTelefono.setValue(administrador.getNumeroTelefono());
```

```
jDateChooser1.setCalendar(administrador.getFechaNacimiento());
```

```
cargarBotonesCambio();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}


private void btnActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int codigo = Integer.valueOf(txtCodigo.getText());
    String nombre = txtNombre.getText();
    String apellido = txtApellido.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    Calendar fechaNacimiento = jDateChooser1.getCalendar();
    String email = txtCorreo.getText();
    String password = txtPassword.getText();
    String telefono = (String) txtFormattedTelefono.getValue();

    if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank() || fechaNacimiento == null
        || email.isBlank() || password.isBlank() || telefono.isBlank()) {
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
    } else {

        String passwordEncriptado = controladorAdministrador.encriptarPassword(keyUnlock, password);
        administrador.setPassword(passwordEncriptado);

        administrador.setCodigo(codigo);
        administrador.setNombre(nombre);
        administrador.setApellido(apellido);
        administrador.setCedula(cedula);
        administrador.setFechaNacimiento(fechaNacimiento);
        administrador.setCorreo(email);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

administrador.setNumeroTelefono(telefono);

administrador.setEnumTipoAdministrador(EnumTipoAdministrador.SIMPLE);

if (controladorAdministrador.update(administrador, administrador)) {
    JOptionPane.showMessageDialog(this, "Administrador actualizado con éxito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    limpiar();

} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}
}

}

private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String cedula = (String) txtFormattedCedula.getValue();

    administrador.setCedula(cedula);

    if (controladorAdministrador.delete(administrador)) {
        JOptionPane.showMessageDialog(this, "Administrador eliminado con éxito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

```
private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

// Variables declaration - do not modify

```
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnEliminar;
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblUsuarios;
private javax.swing.JTextField txtApellido;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private javax.swing.JTextField txtCodigo;
private javax.swing.JTextField txtCorreo;
private javax.swing.JFormattedTextField txtFormattedCedula;
private javax.swing.JFormattedTextField txtFormattedTelefono;
private javax.swing.JTextField txtNombre;
private javax.swing.JPasswordField txtPassword;
// End of variables declaration
}

```

- VentanaEntradaTicket:

```

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

```

```


package ups.edu.ec.vista;

```

```

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorCliente;
import ups.edu.ec.controlador.ControladorFactura;
import ups.edu.ec.controlador.ControladorSitio;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import ups.edu.ec.modelo.Cliente;

import ups.edu.ec.modelo.EnumClienteContrato;

import ups.edu.ec.modelo.EnumEstadoSitio;

import ups.edu.ec.modelo.Factura;

import ups.edu.ec.modelo.Sitio;

/**
 *
 * @author User
 */
public class VentanaEntradaTicket extends javax.swing.JInternalFrame {

    private ControladorSitio controladorSitio;
    private ControladorCliente controladorCliente;
    private ControladorFactura controladorFactura;

    private Sitio sitio;
    private Cliente cliente;

    /**
     * Creates new form VentanaEntradaTicket
     */
    public VentanaEntradaTicket(ControladorSitio controladorSitio, ControladorCliente controladorCliente,
        ControladorFactura controladorFactura) {

        initComponents();

        this.controladorSitio = controladorSitio;
        this.controladorCliente = controladorCliente;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

this.controladorFactura = controladorFactura;

this.sitio = new Sitio();
this.cliente = new Cliente();
limpiar();

}

public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorFactura.generarCodigo()));
}

public void cargarFecha() {

    try {
        Date fecha = new Date(Calendar.getInstance().getTimeInMillis());
        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");

        String fechaTexto = formatter.format(fecha);

        txtFechaEntrada.setText(fechaTexto);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void cargarSitio() {

    sitio = controladorSitio.encontrarSitioDesocupado();

```



	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

if (sitio != null) {
    txtNumeroSitio.setText(String.valueOf(sitio.getCodigo()));
} else {
    JOptionPane.showMessageDialog(this, "Parqueadero lleno, no pueden ingresar más vehiculos",
        "Error", JOptionPane.ERROR_MESSAGE);
    this.hide();
}

}

public void limpiar() {
    cargarCodigo();
    cargarFecha();
    cargarSitio();
    txtFormattedCedula.setValue("");
    llenarTblClientes();
}

public void llenarTblClientes() {
    DefaultTableModel modelo = (DefaultTableModel) tblClientes.getModel();
    modelo.setRowCount(0);

    for (Cliente admin : (List<Cliente>) controladorCliente.getListaGenerica()) {

        if (admin.getEnumClienteContrato().equals(EnumClienteContrato.NO)) {
            Object[] row = {admin.getCedula(), admin.getNombre(), admin.getApellido()};

            modelo.addRow(row);
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }

}

tblClientes.setModel(modelo);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    txtCodigo = new javax.swing.JTextField();
    txtFechaEntrada = new javax.swing.JTextField();
    txtNumeroSitio = new javax.swing.JTextField();
    txtFormattedCedula = new javax.swing.JFormattedTextField();
    btnCrear = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
tblClientes = new javax.swing.JTable();

setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Entrada Ticket");
addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameActivated(evt);
    }
    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameClosed(evt);
    }
    public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
        formInternalFrameOpened(evt);
    }
});

jLabel1.setText("Código:");

jLabel2.setText("Fecha entrada:");
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jLabel3.setText("Código sitio asignado:");

jLabel4.setText("Cédula cliente:");

txtCodigo.setEditable(false);
txtCodigo.setBackground(new java.awt.Color(102, 102, 102));

txtFechaEntrada.setEditable(false);
txtFechaEntrada.setBackground(new java.awt.Color(102, 102, 102));

txtNumeroSitio.setEditable(false);
txtNumeroSitio.setBackground(new java.awt.Color(102, 102, 102));


try {
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("0#####-#")));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

btnCrear.setBackground(new java.awt.Color(51, 51, 255));
btnCrear.setText("Emitir");
btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

btnCancelar.setText("Cancelar");

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(txtFechaEntrada, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(txtNumeroSitio, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(txtFormattedCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(28, 28, 28)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(btnCrear)
                .addComponent(btnCancelar))
            .addContainerGap(25, Short.MAX_VALUE))
        );

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

tblClientes.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Cedula", "Nombre", "Apellido"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblClientes.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblClientesMouseClicked(evt);
    }
});
jScrollPane1.setViewportViewView(tblClientes);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(36, 36, 36)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 504,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(33, Short.MAX_VALUE))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(18, 18, 18)
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 372,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                    .addGap(91, 91, 91)
                    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(20, Short.MAX_VALUE))
        );

pack();

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

}// </editor-fold>

```
private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    int codigo = Integer.valueOf(txtCodigo.getText());
```

```
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");
```

```
    Date fechaEntrada = new Date();
```

```
    try {
```

```
        fechaEntrada = formatter.parse(txtFechaEntrada.getText());
```

```
    } catch (ParseException ex) {
```

```
        Logger.getLogger(VentanaEntradaTicket.class.getName()).log(Level.SEVERE, null, ex);
```

```
    }
```

```
    String cedula = (String) txtFormattedCedula.getValue();
```

```
    if (cedula.isBlank()) {
```

```
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
```

```
    } else {
```

```
        cliente.setCedula(cedula);
```

```
        cliente = (Cliente) controladorCliente.read(cliente);
```

```
        if (cliente != null) {
```

```
            Factura factura = new Factura(codigo, cliente, fechaEntrada, sitio);
```

```
            if (controladorFactura.create(factura)) {
```

```
                JOptionPane.showMessageDialog(this, "Ticket creado con éxito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

controladorSitio.cambiarAEstadoOcupado(sitio);

limpiar();

    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
    }
} else {
    JOptionPane.showMessageDialog(this, "Cliente no encontrado", "Error", JOptionPane.ERROR_MESSAGE);
}

}

}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void tblClientesMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int fila = tblClientes.getSelectedRow();

    String cedula = (String) tblClientes.getValueAt(fila, 0);
    cliente.setCedula(cedula);
    cliente = (Cliente) controladorCliente.read(cliente);
    txtFormattedCedula.setValue(cliente.getCedula());

}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

```
private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

```
private void formInternalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblClientes;
private javax.swing.JTextField txtCodigo;
private javax.swing.JTextField txtFechaEntrada;
```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private javax.swing.JFormattedTextField txtFormattedCedula;

private javax.swing.JTextField txtNumeroSitio;

// End of variables declaration
}

    • VentanaGestionFacturas:

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.vista;

import java.text.ParseException;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorFactura;
import ups.edu.ec.modelo.Factura;

/**
 *
 * @author User
 */
public class VentanaGestionFacturas extends javax.swing.JInternalFrame {

    private ControladorFactura controladorFactura;

    private Factura factura;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/**
 * Creates new form VentanaGestionFacturas
 */
public VentanaGestionFacturas(ControladorFactura controladorFactura) {
    initComponents();

    this.controladorFactura = controladorFactura;

    factura = new Factura();
    limpiar();
}

public void llenarTblFacturas() {

    DefaultTableModel modelo = (DefaultTableModel) tblFacturas.getModel();
    modelo.setRowCount(0);

    for (Factura fac : (List<Factura>) controladorFactura.getListaGenerica()) {

        if (fac.getFechaSalida() != null) {
            Object[] row = {fac.getCodigo(), fac.getFechaEntrada().toString(), fac.getFechaSalida().toString(),
                fac.getTiempo(), fac.getSubtotal(), fac.getTotal(),
                fac.getCliente().getCedula() + " " + fac.getCliente().getApellido(),
                fac.getSitio().getCodigo()};

            modelo.addRow(row);
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }

    tblFacturas.setModel(modelo);

}

public void llenarTblFacturaEncontrada(Factura factura) {
    DefaultTableModel modelo = (DefaultTableModel) tblFacturas.getModel();
    modelo.setRowCount(0);

    Object[] row = {factura.getCodigo(), factura.getFechaEntrada().toString(),
        factura.getFechaSalida().toString(), factura.getTiempo(), factura.getSubtotal(),
        factura.getTotal(), factura.getCliente().getCedula() + " " + factura.getCliente().getApellido(),
        factura.getSitio().getCodigo()};

    modelo.addRow(row);

    tblFacturas.setModel(modelo);
}

public void limpiar() {
    txtBuscar.setText("");
    llenarTblFacturas();
}

/**
 * This method is called from within the constructor to initialize the form.

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


```

* WARNING: Do NOT modify this code. The content of this method is always
* regenerated by the Form Editor.
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    tblFacturas = new javax.swing.JTable();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    txtBuscar = new javax.swing.JTextField();
    btnBuscar = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();

    setClosable(true);
    setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
    setTitle("Facturas");
    addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
        public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
        }
    }

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
}

public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    formInternalFrameOpened(evt);
}

});

tblFacturas.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Código", "Entrada", "Salida", "Tiempo", "Subtotal", "Total", "Cliente", "Sitio"
    }
) {
    Class[] types = new Class [] {
        java.lang.Integer.class,      java.lang.String.class,      java.lang.String.class,      java.lang.Integer.class,
        java.lang.Double.class, java.lang.Double.class, java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
}

```





	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

        .addGap(25, 25, 25)

        .addComponent(jLabel1)

        .addGap(75, 75, 75)

        .addComponent(txtBuscar,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))                                152,

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(92, 92, 92)

        .addComponent(btnBuscar)

        .addGap(41, 41, 41)

        .addComponent(btnCancelar)))

        .addContainerGap(31, Short.MAX_VALUE))

    );

    jPanel1Layout.setVerticalGroup(

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(20, 20, 20)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel1)

            .addComponent(txtBuscar,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))                                152,

        .addGap(18, 18, 18)

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(btnBuscar)

            .addComponent(btnCancelar))

        .addContainerGap(18, Short.MAX_VALUE))

    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(445, Short.MAX_VALUE)

.addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(384, 384, 384))

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jScrollPane1)

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 321, Short.MAX_VALUE)

.addContainerGap())

);

pack();
} // </editor-fold>


private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String codigoBusqueda = txtBuscar.getText();

if (codigoBusqueda.isBlank() || codigoBusqueda.contains("\\D")) {

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

OptionPane.showMessageDialog(this, "Llene el campo busqueda", "Error", JOptionPane.ERROR_MESSAGE);

} else {

    int cod = Integer.valueOf(codigoBusqueda);
    factura.setCodigo(cod);
    factura.setTiempo(0);
    factura = (Factura) controladorFactura.read(factura);
    if (factura != null) {
        llenarTblFacturaEncontrada(factura);
    } else {
        JOptionPane.showMessageDialog(this, "Factura no encontrada", "Error", JOptionPane.ERROR_MESSAGE);
        limpiar();
    }
}

}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
// Variables declaration - do not modify
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnCancelar;
private javax.swing.JLabel jLabel1;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable tblFacturas;
private javax.swing.JTextField txtBuscar;
// End of variables declaration
}

    • VentanaLogIn:
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package ups.edu.ec.vista;

import javax.swing.JOptionPane;
import ups.edu.ec.controlador.ControladorAdministrador;
import ups.edu.ec.modelo.Administrador;

/**
 *
 * @author User
 */
public class VentanaLogIn extends javax.swing.JInternalFrame {
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private ControladorAdministrador controladorAdministrador;

private String keyUnlock;

/**
 * Creates new form VentanaLogIn
 *
 * @param controladorAdministrador
 */
public VentanaLogIn(ControladorAdministrador controladorAdministrador, String keyUnlock) {
    initComponents();


    this.controladorAdministrador = controladorAdministrador;

    this.keyUnlock = keyUnlock;
}

public void limpiar() {
    txtCorreo.setText("");
    txtPassword.setText("");
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    txtCorreo = new javax.swing.JTextField();
    txtPassword = new javax.swing.JPasswordField();
    btnLogIn = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();

    setClosable(true);
    setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
    setTitle("Log In");
    addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
        public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
            formInternalFrameOpened(evt);
        }
    })
}
```





	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addGap(26, 26, 26)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel1)

    .addComponent(jLabel2))

.addGap(63, 63, 63)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

    .addComponent(txtCorreo)

    .addComponent(txtPassword, javax.swing.GroupLayout.DEFAULT_SIZE, 219, Short.MAX_VALUE)))

.addGroup(jPanel1Layout.createSequentialGroup())

    .addGap(131, 131, 131)

    .addComponent(btnLogIn)

    .addGap(35, 35, 35)

    .addComponent(btnCancelar)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

    .addGap(15, 15, 15)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

        .addComponent(jLabel1)

        .addComponent(txtCorreo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(18, 18, 18)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)


        .addComponent(jLabel2)

        .addComponent(txtPassword, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 32, Short.MAX_VALUE)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```


        .addComponent(btnLogIn)
        .addComponent(btnCancelar))
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(31, 31, 31))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(22, 22, 22))
    );

    pack();
} // </editor-fold>

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

}

private void btnLogInActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

    String correo = txtCorreo.getText();
    String password = txtPassword.getText();

    if (correo.isBlank() || password.isBlank()) {
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);

    } else {
        if (Administrador.instance == null) {
            var admin = controladorAdministrador.iniciarSesion(correo, password, keyUnlock);
            if (admin != null) {
                JOptionPane.showMessageDialog(this, "Sesión iniciada con éxito", "Mensaje",
                JOptionPane.INFORMATION_MESSAGE);
                limpiar();
                this.hide();
            } else {
                JOptionPane.showMessageDialog(this, "Usuario erroneo", "Error", JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(this, "Existe otro usuario conectado", "Error",
            JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}
```

// Variables declaration - do not modify

```
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnLogIn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField txtCorreo;
private javax.swing.JPasswordField txtPassword;
```

// End of variables declaration

```
}
```

- VentanaSalidaTicket:

```
/*
```

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

```
*/
```

```
package ups.edu.ec.vista;
```


```
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.*;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

import javax.swing.JOptionPane;

import ups.edu.ec.controlador.ControladorFactura;
import ups.edu.ec.controlador.ControladorSitio;
import ups.edu.ec.modelo.Factura;

/**
 *
 * @author User
 */
public class VentanaSalidaTicket extends javax.swing.JInternalFrame {

    private Factura factura;

    private ControladorFactura controladorFactura;
    private ControladorSitio controladorSitio;

    /**
     * Creates new form VentanaSalidaTicket
     */
    public VentanaSalidaTicket(ControladorFactura controladorFactura, ControladorSitio controladorSitio) {
        initComponents();

        this.controladorFactura = controladorFactura;
        this.controladorSitio = controladorSitio;

        factura = new Factura();
    }

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void cargarFecha() {
    try {
        Date fecha = new Date(Calendar.getInstance().getTimeInMillis());
        SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");

        String fechaTexto = formatter.format(fecha);

        txtFechaSalida.setText(fechaTexto);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void limpiar() {
    txtCodigoBuscar.setText("");
    txtCodigoBuscar.setEditable(true);

    txtFechaEntrada.setText("");
    txtNumeroSitio.setText("");
    txtSubtotal.setText("");
    txtTotal.setText("");
    txtFormattedCedula.setValue("");
    btnGroupDescuento.clearSelection();
    txtTiempo.setText("");
    btnCrear.setEnabled(false);
}

public void cargarSubtotal() {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");
Date fechaSalida = null;
try {
    fechaSalida = formatter.parse(txtFechaSalida.getText());
} catch (ParseException ex) {
    Logger.getLogger(VentanaSalidaTicket.class.getName()).log(Level.SEVERE, null, ex);
}
double subtotal = controladorFactura.calcularSubtotal(factura.getFechaEntrada(), fechaSalida);
txtSubtotal.setText(String.valueOf(subtotal));
}

public void cargarTiempo() {
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");
    Date fechaSalida = null;
    try {
        fechaSalida = formatter.parse(txtFechaSalida.getText());
    } catch (ParseException ex) {
        Logger.getLogger(VentanaSalidaTicket.class.getName()).log(Level.SEVERE, null, ex);
    }
    txtTiempo.setText(String.valueOf(controladorFactura.calcularTiempo(factura.getFechaEntrada(), fechaSalida)));
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private void initComponents() {

    btnGroupDescuento = new javax.swing.ButtonGroup();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    txtCodigoBuscar = new javax.swing.JTextField();
    txtFechaEntrada = new javax.swing.JTextField();
    txtNumeroSitio = new javax.swing.JTextField();
    txtFormattedCedula = new javax.swing.JFormattedTextField();
    btnCrear = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    jLabel5 = new javax.swing.JLabel();
    txtFechaSalida = new javax.swing.JTextField();
    jLabel6 = new javax.swing.JLabel();
    txtSubtotal = new javax.swing.JTextField();
    btnBuscar = new javax.swing.JButton();
    radioBtnAplicar = new javax.swing.JRadioButton();
    radioBtnNoAplicar = new javax.swing.JRadioButton();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    txtTotal = new javax.swing.JTextField();
    jLabel7 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    txtTiempo = new javax.swing.JTextField();
}
```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

setClosable(true);

setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);

setTitle("Factura");


jLabel1.setText("Código:");


jLabel2.setText("Fecha entrada:");


jLabel3.setText("Código sitio asignado:");


jLabel4.setText("Cédula cliente:");


txtCodigoBuscar.setBackground(new java.awt.Color(255, 255, 255));


txtFechaEntrada.setEditable(false);
txtFechaEntrada.setBackground(new java.awt.Color(102, 102, 102));


txtNumeroSitio.setEditable(false);
txtNumeroSitio.setBackground(new java.awt.Color(102, 102, 102));


txtFormattedCedula.setEditable(false);
try {
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.MaskFormatter("0#####-#")));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}


btnCrear.setBackground(new java.awt.Color(51, 51, 255));

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

btnCrear.setText("Emitir Factura");

btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

btnCancelar.setText("Cancelar");

btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

jLabel5.setText("Fecha salida:");

txtFechaSalida.setEditable(false);
txtFechaSalida.setBackground(new java.awt.Color(102, 102, 102));

jLabel6.setText("Subtotal:");

txtSubtotal.setEditable(false);
txtSubtotal.setBackground(new java.awt.Color(102, 102, 102));

btnBuscar.setBackground(new java.awt.Color(0, 0, 255));
btnBuscar.setText("Buscar");
btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(evt);
    }
});

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    }
});

btnGroupDescuento.add(radioBtnAplicar);
radioBtnAplicar.setText("SI");
radioBtnAplicar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnAplicarActionPerformed(evt);
    }
});

btnGroupDescuento.add(radioBtnNoAplicar);
radioBtnNoAplicar.setText("NO");
radioBtnNoAplicar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        radioBtnNoAplicarActionPerformed(evt);
    }
});

jLabel8.setText("Descuento:");

jLabel9.setText("Total:");

txtTotal.setEditable(false);

jLabel7.setText("$");

jLabel10.setText("$");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jLabel11.setText("Tiempo(minutos):");

txtTiempo.setEditable(false);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addComponent(jLabel2)
                .addComponent(jLabel3)
                .addComponent(jLabel4)
                .addComponent(jLabel5)
                .addComponent(jLabel6)
                .addComponent(jLabel11)
                .addComponent(jLabel8)
                .addComponent(jLabel9))
            .addGap(61, 61, 61))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(btnCrear))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(txtCodigoBuscar))
    );

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addComponent(jLabel7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtTotal))

.addComponent(txtFormattedCedula)

.addComponent(txtFechaSalida)

.addComponent(txtFechaEntrada)

.addComponent(txtNumeroSitio)

.addGroup(jPanel1Layout.createSequentialGroup())

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(radioBtnAplicar)

.addGap(24, 24, 24)

.addComponent(radioBtnNoAplicar))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(40, 40, 40)

.addComponent(btnCancelar)))

.addGap(0, 83, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(txtSubtotal))

.addComponent(txtTiempo))

.addGap(22, 22, 22))

.addGroup(jPanel1Layout.createSequentialGroup())

.addGap(192, 192, 192)


.addComponent(btnBuscar)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

jPanel1Layout.setVerticalGroup(

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup()
.addContainerGap()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel1)
.addComponent(txtCodigoBuscar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addComponent(btnBuscar)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel2)
.addComponent(txtFechaEntrada, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel3)
.addComponent(txtNumeroSitio, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel4)
.addComponent(txtFormattedCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel5)
.addComponent(txtFechaSalida, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addComponent(jLabel6)

.addComponent(txtSubtotal,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel10))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11)

.addComponent(txtTiempo,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGap(29, 29, 29)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(radioBtnAplicar)

.addComponent(radioBtnNoAplicar)

.addComponent(jLabel8))

.addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(txtTotal,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel7))

.addGap(18, 21, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnCrear)

.addComponent(btnCancelar))

.addGap(20, 20, 20))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(16, Short.MAX_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

);

pack();
} // </editor-fold>

```

```

private void radioBtnNoAplicarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:

```

```

    factura.setDescuento(false);

```

```

    txtTotal.setText(txtSubtotal.getText());

```

```

}

```

```

private void radioBtnAplicarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    // TODO add your handling code here:

```

```

    double total = controladorFactura.calcularTotal(Double.valueOf(txtSubtotal.getText()), true);


```

```

    factura.setDescuento(true);

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtTotal.setText(String.valueOf(total));
}

private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int codigo = Integer.valueOf(txtCodigoBuscar.getText());
    factura.setCodigo(codigo);

    factura = (Factura) controladorFactura.read(factura);

    if (factura != null) {
        txtCodigoBuscar.setEditable(false);
        txtFechaEntrada.setText(factura.getFechaEntrada().toString());
        cargarFecha();
        txtFormattedCedula.setValue(factura.getCliente().getCedula());
        txtNumeroSitio.setText(String.valueOf(factura.getSitio().getCodigo()));
        btnCrear.setEnabled(true);
        cargarSubtotal();
        cargarTiempo();
    } else {
        JOptionPane.showMessageDialog(this, "Factura no encontrada", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy - HH:mm:ss");
    Date fechaSalida = new Date();
    try {
        fechaSalida = formatter.parse(txtFechaEntrada.getText());
    } catch (ParseException ex) {

    }

    factura.setFechaSalida(fechaSalida);
    factura.setSubtotal(Double.valueOf(txtSubtotal.getText()));
    factura.setTiempo(Integer.valueOf(txtTiempo.getText()));
    factura.setTotal(Double.valueOf(txtTotal.getText()));

    if (controladorFactura.update(factura, factura)) {
        JOptionPane.showMessageDialog(this, "Factura emitida con éxito", "Mensaje",
        JOptionPane.INFORMATION_MESSAGE);
        controladorSitio.cambiarAEstadoDescupado(factura.getSitio());
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
// Variables declaration - do not modify

private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.ButtonGroup btnGroupDescuento;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JRadioButton radioBtnAplicar;
private javax.swing.JRadioButton radioBtnNoAplicar;
private javax.swing.JTextField txtCodigoBuscar;
private javax.swing.JTextField txtFechaEntrada;
private javax.swing.JTextField txtFechaSalida;
private javax.swing.JFormattedTextField txtFormattedCedula;
private javax.swing.JTextField txtNumeroSitio;
private javax.swing.JTextField txtSubtotal;
private javax.swing.JTextField txtTiempo;
private javax.swing.JTextField txtTotal;

// End of variables declaration

}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- VentanaVehiculo:

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

```
*/
```

```
package ups.edu.ec.vista;
```

```
import java.util.Iterator;
```

```
import java.util.List;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.table.DefaultTableModel;
```

```
import ups.edu.ec.controlador.ControladorCliente;
```

```
import ups.edu.ec.controlador.ControladorVehiculo;
```

```
import ups.edu.ec.modelo.Cliente;
```

```
import ups.edu.ec.modelo.Vehiculo;
```

```
/**
```

```
 *
```

```
 * @author User
```

```
*/
```

```
public class VentanaVehiculo extends javax.swing.JInternalFrame {
```

```
    private ControladorCliente controladorCliente;
```

```
    private ControladorVehiculo controladorVehiculo;
```

```
    private Vehiculo vehiculo;
```

```
    private Cliente cliente;
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

/**
 * Creates new form VentanaVehiculo
 */
public VentanaVehiculo(ControladorCliente controladorCliente, ControladorVehiculo controladorVehiculo) {
    initComponents();

    this.controladorCliente = controladorCliente;
    this.controladorVehiculo = controladorVehiculo;

    vehiculo = new Vehiculo();
    cliente = new Cliente();
}

public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorVehiculo.generarCodigo()));
}

public void limpiar() {
    cargarCodigo();
    txtFormattedCedula.setValue("");
    txtFormattedPlaca.setValue("");
    txtFormattedPlaca.setEnabled(true);
    txtMarca.setText("");
    txtModelo.setText("");

    tblClientes.clearSelection();
    cargarBotones();
    llenarTblClientes();
}

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

llenarTblVehiculos();
}

public void cargarBotones() {
    btnCrear.setEnabled(true);
    btnActualizar.setEnabled(false);
    btnEliminar.setEnabled(false);
    btnCancelar.setEnabled(true);
}

public void cargarBotonesCambio() {
    btnCrear.setEnabled(false);
    btnActualizar.setEnabled(true);
    btnEliminar.setEnabled(true);
    btnCancelar.setEnabled(true);
}

public void llenarTblClientes() {
    DefaultTableModel modelo = (DefaultTableModel) tblClientes.getModel();
    modelo.setRowCount(0);

    for (Cliente admin : (List<Cliente>) controladorCliente.getListaGenerica()) {

        Object[] row = {admin.getCedula(), admin.getNombre(), admin.getApellido(),
            admin.getEnumClienteContrato().toString()};

        modelo.addRow(row);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

tblClientes.setModel(modelo);

}

public void llenarTblVehiculos() {
    DefaultTableModel modelo = (DefaultTableModel) tblVehiculos.getModel();
    modelo.setRowCount(0);

    for (var vehi : (List<Vehiculo>) controladorVehiculo.getListaGenerica()) {

        Object[] row = {vehi.getCodigo(), vehi.getMarca(), vehi.getModelo(),
            vehi.getPlaca(), vehi.getCliente().getCedula(),
            vehi.getCliente().getNombre().concat(vehi.getCliente().getApellido())};
        modelo.addRow(row);
    }

    tblVehiculos.setModel(modelo);

}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

jScrollPane1 = new javax.swing.JScrollPane();
tblClientes = new javax.swing.JTable();
jPanel1 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
txtCodigo = new javax.swing.JTextField();
txtMarca = new javax.swing.JTextField();
txtModelo = new javax.swing.JTextField();
txtFormattedPlaca = new javax.swing.JFormattedTextField();
btnCrear = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminar = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();
jLabel5 = new javax.swing.JLabel();
txtFormattedCedula = new javax.swing.JFormattedTextField();
jScrollPane2 = new javax.swing.JScrollPane();
tblVehiculos = new javax.swing.JTable();

setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Vehiculos");
addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
    public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    }
    public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
    }
}

```



	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
}

public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
}

public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
}

public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
}

public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    formInternalFrameOpened(evt);
}


});

tblClientes.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Cedula", "Nombre", "Apellido", "Contrato"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

return types [columnIndex];
}

public boolean isCellEditable(int rowIndex, int columnIndex) {
    return canEdit [columnIndex];
}
});

tblClientes.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblClientesMouseClicked(evt);
    }
});

jScrollPane1.setViewportView(tblClientes);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(41, 43, 45)), "Vehiculo", javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.TOP));

jLabel1.setText("Código:");

jLabel2.setText("Marca:");

jLabel3.setText("Modelo:");

jLabel4.setText("Placa:");

txtCodigo.setEditable(false);
txtCodigo.setBackground(new java.awt.Color(153, 153, 153));

try {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

txtFormattedPlaca.setFormatterFactory(new
javax.swing.text.MaskFormatter("UUU-####"));

} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

btnCrear.setBackground(new java.awt.Color(51, 51, 255));
btnCrear.setText("Crear");
btnCrear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCrearActionPerformed(evt);
    }
});

btnActualizar.setText("Actualizar");
btnActualizar.setEnabled(false);
btnActualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

btnEliminar.setBackground(new java.awt.Color(255, 51, 51));
btnEliminar.setText("Eliminar");
btnEliminar.setEnabled(false);
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnEliminarActionPerformed(evt);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

});

btnCancelar.setText("Cancelar");
btnCancelar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCancelarActionPerformed(evt);
    }
});

jLabel5.setText("Cédula cliente:");

try {
    txtFormattedCedula.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
    javax.swing.text.MaskFormatter("0#####-#")));
} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(20, 20, 20)
            .addComponent(btnCrear)
            .addGap(38, 38, 38)
            .addComponent(btnActualizar)
            .addGap(34, 34, 34)
            .addComponent(btnEliminar)
        )
    );


```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 30, Short.MAX_VALUE)
.addComponent(btnCancelar)
.addGap(25, 25, 25))
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(48, 48, 48)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel1)
    .addComponent(jLabel2)
    .addComponent(jLabel3)
    .addComponent(jLabel4)
    .addComponent(jLabel5))
.addGap(33, 33, 33)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(txtCodigo)
    .addComponent(txtMarca)
    .addComponent(txtModelo)
    .addComponent(txtFormattedPlaca)
    .addComponent(txtFormattedCedula, javax.swing.GroupLayout.DEFAULT_SIZE, 157, Short.MAX_VALUE))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(17, 17, 17)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel1)
    .addComponent(txtCodigo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


```

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(txtMarca, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(txtModelo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(txtFormattedPlaca, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(txtFormattedCedula, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(24, 24, 24)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(btnCrear)
    .addComponent(btnActualizar)
    .addComponent(btnEliminar)
    .addComponent(btnCancelar))
.addContainerGap(23, Short.MAX_VALUE))

);

tblVehiculos.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

    },
    new String [] {
        "Codigo", "Marca", "Modelo", "Placa", "Cédula", "Nombre"
    }
) {
    Class[] types = new Class [] {
        java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.String.class, java.lang.String.class,
        java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false
    };

    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
tblVehiculos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblVehiculosMouseClicked(evt);
    }
});
jScrollPane2.setViewportViewView(tblVehiculos);

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 504,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(34, 34, 34)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 829,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(65, 65, 65)))
        .addGap(18, 18, 18)
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 271,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
            )
        .addGap(18, 18, 18)
    );

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

);

pack();
} // </editor-fold>

private void tblClientesMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int fila = tblClientes.getSelectedRow();

    String cedula = (String) tblClientes.getValueAt(fila, 0);
    cliente.setCedula(cedula);
    cliente = (Cliente) controladorCliente.read(cliente);

    txtFormattedCedula.setValue(cedula);


}

private void btnCrearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    int codigo = Integer.valueOf(txtCodigo.getText());
    String marca = txtMarca.getText();
    String modelo = txtModelo.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    String placa = (String) txtFormattedPlaca.getValue();

    if (marca.isBlank() || modelo.isBlank() || cedula.isBlank() || placa == null) {
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

} else {

    cliente.setCedula(cedula);
    cliente = (Cliente) controladorCliente.read(cliente);
    if (cliente != null) {
        vehiculo = new Vehiculo(codigo, placa, marca, modelo, cliente);

        if (controladorVehiculo.create(vehiculo)) {
            JOptionPane.showMessageDialog(this, "Vehiculo creado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
            limpiar();

        } else {
            JOptionPane.showMessageDialog(this, "Vehiculo ya existente", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }


}

}

private void btnActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int codigo = Integer.valueOf(txtCodigo.getText());
    String marca = txtMarca.getText();
    String modelo = txtModelo.getText();
    String cedula = (String) txtFormattedCedula.getValue();
    String placa = (String) txtFormattedPlaca.getValue();

    if (marca.isBlank() || modelo.isBlank() || cedula.isBlank() || placa == null) {

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {

    cliente.setCedula(cedula);
    cliente = (Cliente) controladorCliente.read(cliente);
    if (cliente != null) {
        vehiculo = new Vehiculo(codigo, placa, marca, modelo, cliente);

        if (controladorVehiculo.update(vehiculo, vehiculo)) {
            JOptionPane.showMessageDialog(this, "Vehiculo actualizado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
            limpiar();

        } else {
            JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
        }
    }

}

}

private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String placa = (String) txtFormattedPlaca.getValue();

    vehiculo.setPlaca(placa);

    if (controladorVehiculo.delete(vehiculo)) {
        JOptionPane.showMessageDialog(this, "Vehiculo eliminado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}
}

private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    limpiar();
}

private void tblVehiculosMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

    int fila = tblVehiculos.getSelectedRow();

    txtCodigo.setText(String.valueOf(tblVehiculos.getValueAt(fila, 0)));
    txtMarca.setText((String) tblVehiculos.getValueAt(fila, 1));
    txtModelo.setText((String) tblVehiculos.getValueAt(fila, 2));
    txtFormattedPlaca.setValue(tblVehiculos.getValueAt(fila, 3));
    txtFormattedPlaca.setEnabled(false);
    txtFormattedCedula.setValue(tblVehiculos.getValueAt(fila, 4));

    cargarBotonesCambio();

}

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

// TODO add your handling code here:

limpiar();

}

// Variables declaration - do not modify

private javax.swing.JButton btnActualizar;

private javax.swing.JButton btnCancelar;

private javax.swing.JButton btnCrear;

private javax.swing.JButton btnEliminar;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JTable tblClientes;

private javax.swing.JTable tblVehiculos;

private javax.swing.JTextField txtCodigo;

private javax.swing.JFormattedTextField txtFormattedCedula;

private javax.swing.JFormattedTextField txtFormattedPlaca;


private javax.swing.JTextField txtMarca;

private javax.swing.JTextField txtModelo;

// End of variables declaration

}

- VentanaSitios:

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

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

```

```
package ups.edu.ec.vista;
```

```

import com.toedter.calendar.JMonthChooser;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.util.List;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.BevelBorder;
import javax.swing.border.Border;
import ups.edu.ec.controlador.ControladorSitio;
import ups.edu.ec.modelo.EnumEstadoSitio;
import ups.edu.ec.modelo.Sitio;

```

```

/**
 *
 * @author User
 */

```

```

public class VentanaSitios extends javax.swing.JInternalFrame {

    private ControladorSitio controladorSitio;

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```
private Sitio sitio;
```

```
/**
```

```
 * Creates new form VentanaSitios
```

```
 *
```

```
 * @param controladorSitio
```

```
 */
```

```
public VentanaSitios(ControladorSitio controladorSitio) {
```

```
    initComponents();
```

```
    this.controladorSitio = controladorSitio;
```

```
    sitio = new Sitio();
```

```
    anadirParqueaderos();
```

```
}
```

```
public void anadirParqueaderos() {
```

```
    var listaSitio = (List<Sitio>) controladorSitio.getListaGenerica();
```


```
    Border border = new BevelBorder(4, Color.BLACK, Color.GRAY);
```

```
    GridLayout layout = new GridLayout(5, 10, 10, 10);
```

```
    panelSitios.setLayout(layout);
```

```
    int codigoSitio = 49;
```

```
    panelSitios.removeAll();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

while (codigoSitio >= 0) {
    sitio = listaSitio.get(codigoSitio);
    JLabel labelSitio = new JLabel();

    JPanel panelito = new JPanel();
    panelito.setSize(5, 5);
    //panelito.setLayout(new BorderLayout());
    switch (sitio.getEnumEstadoSitio()) {
        case DESOCUPADO -> {
            panelito.setBackground(Color.GREEN);
            String texto = "<html><body> . <br> ? <br></body></html>";
            texto = texto.replace(".", String.valueOf(sitio.getCodigo()));
            texto = texto.replace("?", "Desocupado");
            labelSitio.setText(texto);
            break;
        }
        case OCUPADO -> {
            panelito.setBackground(Color.yellow);
            String texto = "<html><body> . <br> ? <br></body></html>";
            texto = texto.replace(".", String.valueOf(sitio.getCodigo()));
            texto = texto.replace("?", "Ocupado");
            labelSitio.setText(texto);
            break;
        }

        default -> {
            panelito.setBackground(Color.RED);
            String texto = "<html><body> . <br> ? <br></body></html>";
            texto = texto.replace(".", String.valueOf(sitio.getCodigo()));

```



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

texto = texto.replace("?", "Contratado");

labelSitio.setText(texto);

break;

}

}

panelito.add(labelSitio);

codigoSitio--;

panelSitios.add(panelito);

}

/*

for (int i = 0; i < layout.getRows(); i++) {
    for (int j = 0; j < layout.getColumns(); j++) {
        sitio = listaSitio.get(codigoSitio);
        JLabel labelSitio = new JLabel();

        JPanel panelito = new JPanel();
        panelito.setSize(5, 5);
        //panelito.setLayout(new BorderLayout());
        switch (sitio.getEnumEstadoSitio()) {
            case DESOCUPADO -> {
                panelito.setBackground(Color.GREEN);
                String texto = "<html><body> . <br> ? <br></body></html>";
                texto = texto.replace(".", String.valueOf(sitio.getCodigo()));
                texto = texto.replace("?", "Desocupado");
                labelSitio.setText(texto);
                break;
            }
            case OCUPADO -> {
                panelito.setBackground(Color.yellow);

```

```
String texto = "<html><body> . <br> ? <br></body></html>";  
texto = texto.replace(".", String.valueOf(sitio.getCodigo()));  
texto = texto.replace("?", "Ocupado");  
labelSitio.setText(texto);  
break;  
}
```

```
default -> {  
    panelito.setBackground(Color.RED);  
    String texto = "<html><body> . <br> ? <br></body></html>";  
    texto = texto.replace(".", String.valueOf(sitio.getCodigo()));  
    texto = texto.replace("?", "Contratado");  
    labelSitio.setText(texto);  
    break;  
}
```

```
}  
panelito.add(labelSitio);  
codigoSitio--;  
panelSitios.add(panelito);
```

```
*/
```

```
}
```

```
/**
```

- \* This method is called from within the constructor to initialize the form.
- \* WARNING: Do NOT modify this code. The content of this method is always
- \* regenerated by the Form Editor.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

    panelSitios = new javax.swing.JPanel();

    setClosable(true);
    setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
    setTitle("Sitios");
    addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
        public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
            formInternalFrameActivated(evt);
        }
        public void internalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
            formInternalFrameClosed(evt);
        }
        public void internalFrameClosing(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeactivated(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameDeiconified(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameIconified(javax.swing.event.InternalFrameEvent evt) {
        }
        public void internalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
            formInternalFrameOpened(evt);
        }
    });

```

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

javax.swing.GroupLayout panelSitiosLayout = new javax.swing.GroupLayout(panelSitios);
panelSitios.setLayout(panelSitiosLayout);
panelSitiosLayout.setHorizontalGroup(
    panelSitiosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 928, Short.MAX_VALUE)
);
panelSitiosLayout.setVerticalGroup(
    panelSitiosLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 516, Short.MAX_VALUE)
);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(panelSitios, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(panelSitios, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
);

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

pack();
} // </editor-fold>

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    anadirParqueaderos();
}

private void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    anadirParqueaderos();
}

private void formInternalFrameClosed(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    anadirParqueaderos();
}

// Variables declaration - do not modify
private javax.swing.JPanel panelSitios;
// End of variables declaration
}

```

N.

	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

#### RESULTADO(S) OBTENIDO(S):

- Lograr afianzar los conocimientos adquiridos sobre lo aprendido en el primer inter ciclo de la materia de Programación Aplicada.
- Aplicar buenas prácticas de programación.
- Aplicar patrones de diseño aprendidos en la materia de Programación Aplicada.
- Aplicar clases abstractas para así generar pseudocódigo para otras clases.
- Resolver problemas que se nos pueden presentar como programadores y encontrar su solución mediante la búsqueda en distintas fuentes.

#### CONCLUSIONES:

En conclusión, este trabajo ha sido de mucha ayuda para practicar todo lo aprendido a lo largo de este inter ciclo. Además, Esto nos ayuda como futuros programadores para poder crear aplicaciones en tiempos cortos de producción. Además, mejoramos nuestras técnicas de programación.

#### RECOMENDACIONES:

No existe ninguna recomendación ya que todo fue explicado en clase.

**Nombre de estudiante:** Adolfo Sebastián Jara Gavilanes.

**Firma de estudiante:** \_\_\_\_\_

