

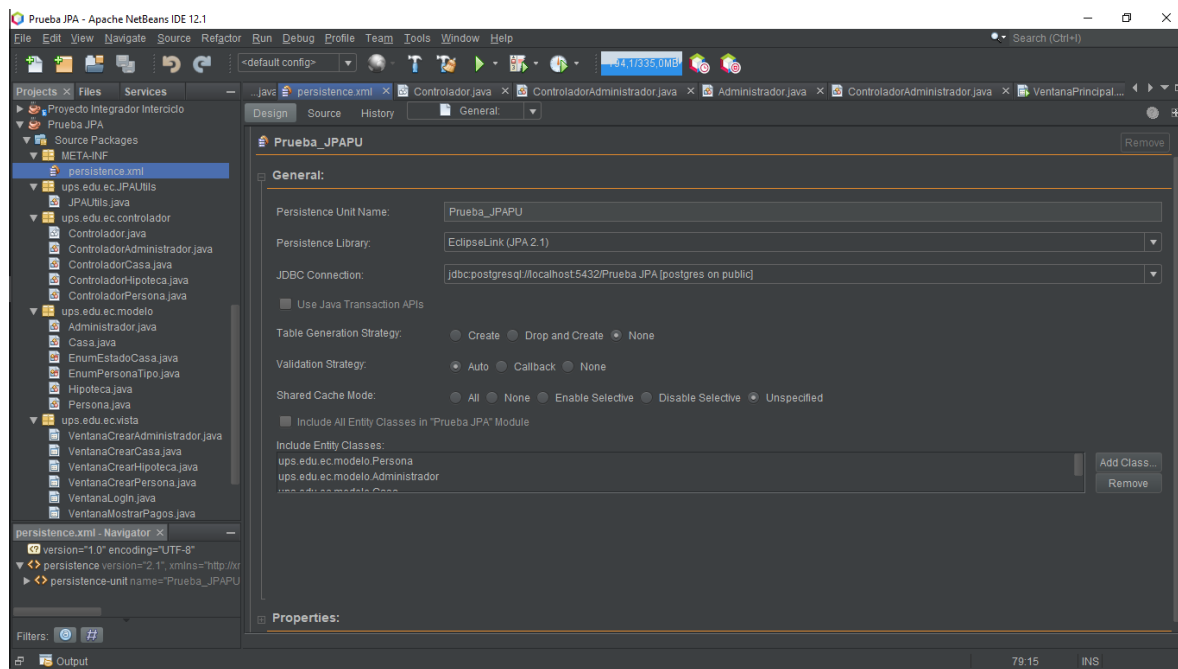
	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>PRÁCTICA DE LABORATORIO</b>	
<b>CARRERA:</b> COMPUTACION		<b>ASIGNATURA:</b> PROGRAMACION APLICADA	
<b>NRO. PRÁCTICA:</b>	3	<b>TÍTULO PRÁCTICA:</b> BASE DE DATOS DE JAVA	
<b>OBJETIVO ALCANZADO:</b> <ul style="list-style-type: none"> <li>Consolidar los conocimientos adquiridos en clase sobre JPA.</li> </ul>			
<b>ACTIVIDADES DESARROLLADAS</b>			
<b>1. ENUNCIADO.</b> <p>Realizar un sistema implementando todos los conceptos vistos en clases para gestionar la hipoteca de las casas con las siguientes características:</p> <ul style="list-style-type: none"> <li>Las personas compran casas y se convierten en propietarios.</li> <li>Para pagarlas es habitual que el propietario formalice un préstamo hipotecario con una entidad bancaria.</li> <li>El banco toma la casa en forma de aval en caso de impago de las mensualidades.</li> <li>En el caso de que el capital fiado supera el valor de tasación de la casa y el sueldo del propietario no es suficiente, el banco suele exigir la presencia de un avalista (garante).</li> <li>Para formalizar la hipoteca se necesitan los datos personales del propietario, además de su cédula, dirección de la casa, su dirección, nombres, apellidos y fecha de nacimiento y del garante de ser necesario.</li> <li>El capital de la hipoteca se ajusta teniendo en cuenta el valor de tasación de la casa y los datos de dirección.</li> <li>Toda hipoteca se formaliza detallando el capital, el interés (8,99 - 16,99%) y la duración (fecha de inicio y fecha de fin).</li> <li>A partir de estos datos se calcula el importe de cada mensualidad para el total del tiempo que pide el préstamo.</li> <li>No es necesario guardar los datos del banco pero si un sistema de autenticación.</li> <li>Generar los datos con el sistema de amortización Alemán [1].</li> </ul> <p><b>Ejemplo</b> simulador de crédito para guía: <a href="https://www.pichincha.com/portal/simuladores/simulador-de-creditos">https://www.pichincha.com/portal/simuladores/simulador-de-creditos</a></p> <p>Se calificará de la siguiente forma:</p> <ul style="list-style-type: none"> <li>JPA: 40%</li> <li>Excepciones: 10%</li> <li>MVC: 20%</li> <li>Diagrama de clases: 10%</li> <li>Usabilidad - Vista: 15%</li> <li>Programación genérica, Java 8, reflexión: 15%</li> </ul> <p>Bibliografía</p>			

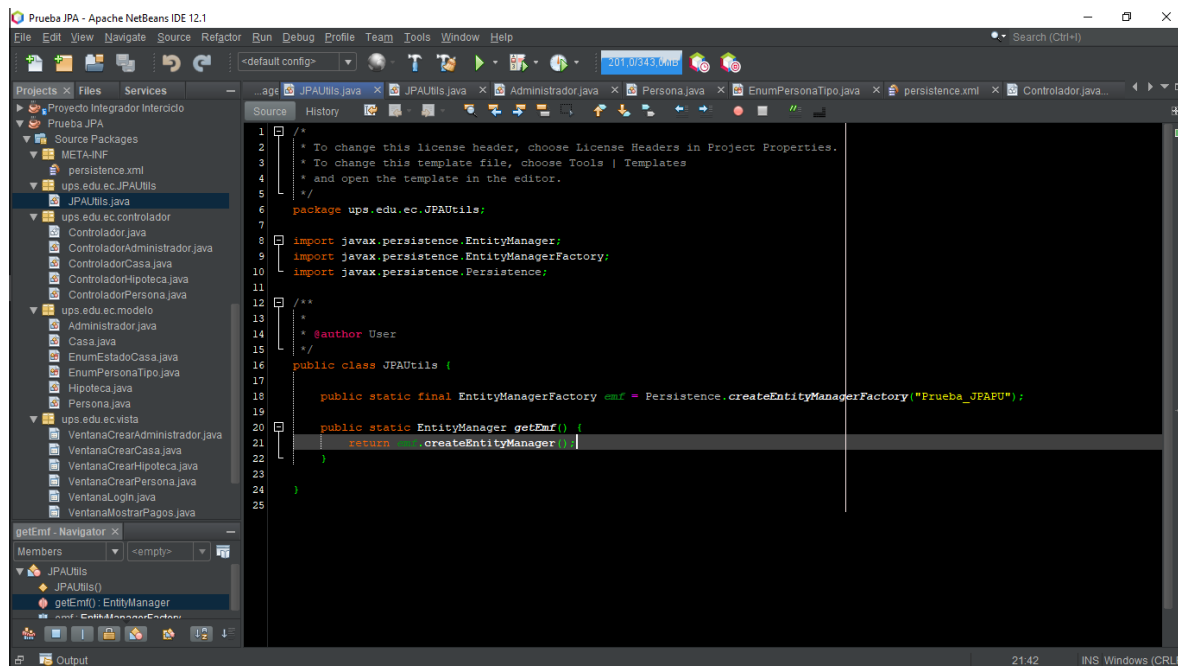
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


## 2. JPA

A continuación, se presentará el JPA de esta prueba:

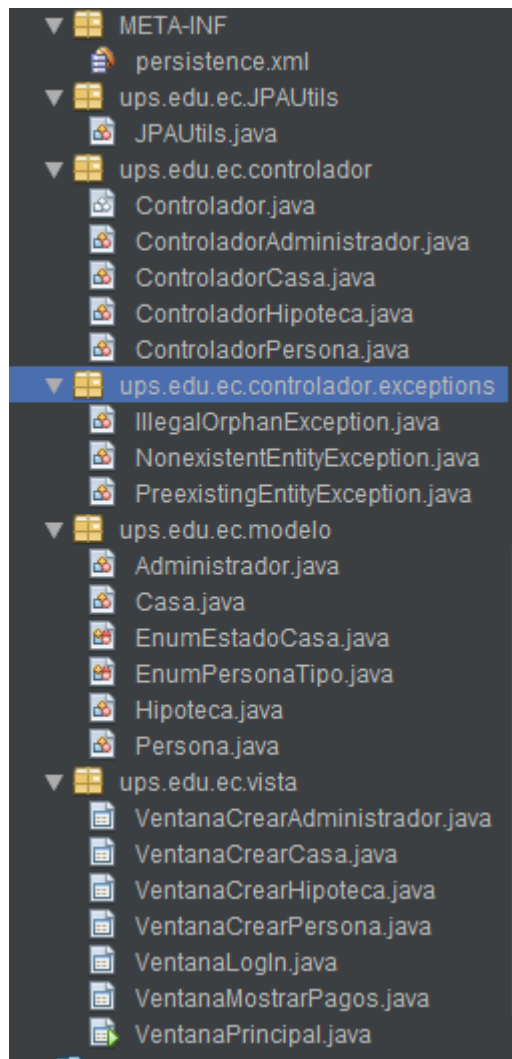


Ahora la clase JPA Utils:



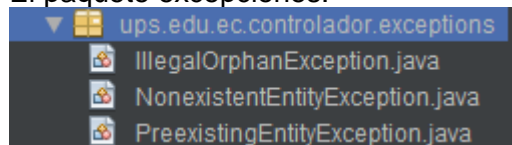
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Por último, los paquetes y clases creados para esta prueba:



- Excepciones

El paquete excepciones:




**IllegalOrphanException:**

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

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

```
public class IllegalOrphanException extends Exception {
```

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

```

private List<String> messages;
public IllegalOrphanException(List<String> messages) {
    super((messages != null && messages.size() > 0 ? messages.get(0) : null));
    if (messages == null) {
        this.messages = new ArrayList<String>();
    }
    else {
        this.messages = messages;
    }
}
public List<String> getMessages() {
    return messages;
}
}

```

#### **NonExistenceEntityException:**

```

package ups.edu.ec.controlador.exceptions;

public class NonexistentEntityException extends Exception {
    public NonexistentEntityException(String message, Throwable cause) {
        super(message, cause);
    }
    public NonexistentEntityException(String message) {
        super(message);
    }
}

```


#### **PreExistingEntityException:**

```

package ups.edu.ec.controlador.exceptions;

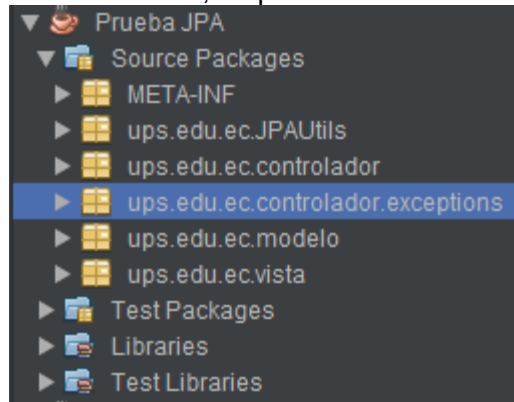
public class PreexistingEntityException extends Exception {
    public PreexistingEntityException(String message, Throwable cause) {
        super(message, cause);
    }
    public PreexistingEntityException(String message) {
        super(message);
    }
}

```

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

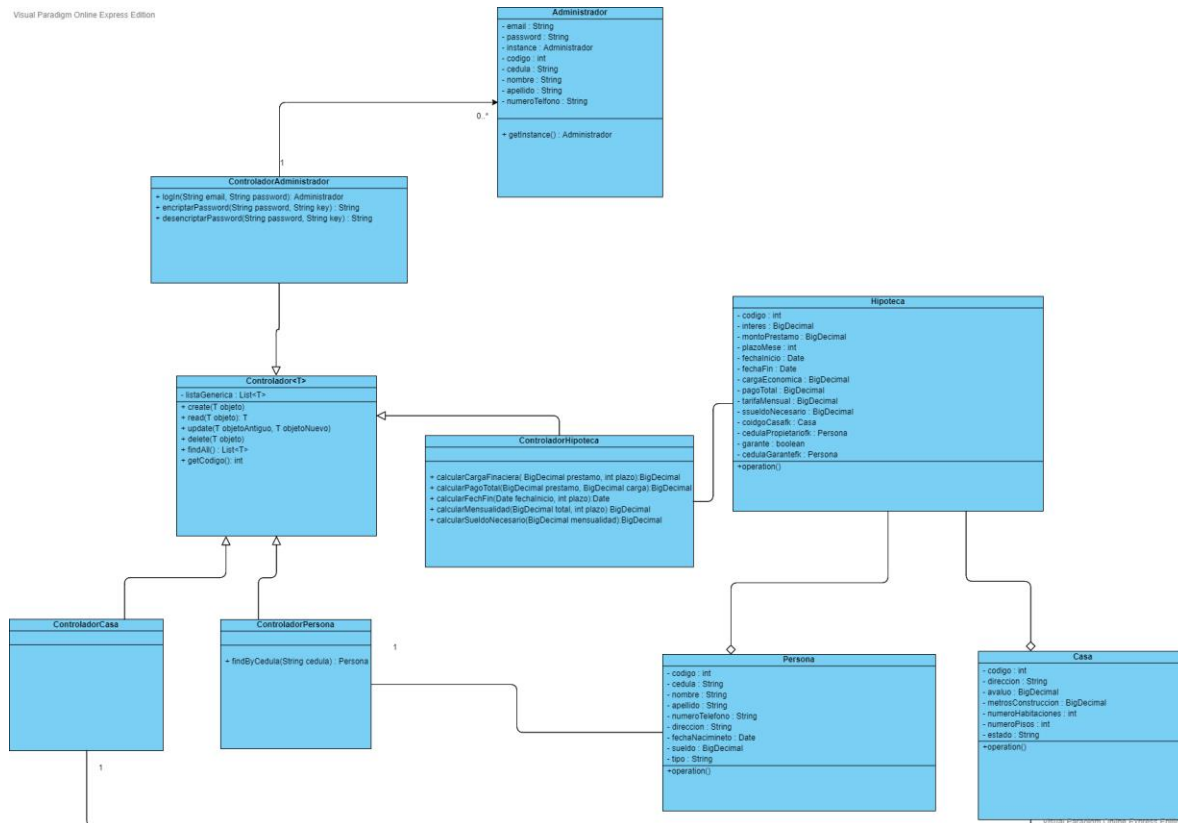
- MVC

A continuación, se presentará el MVC del proyecto:




Como podemos observar, tenemos el MVC, y como utilizamos el JPA en esta prueba, fue necesario añadir el paquete JPAUtils.

- Diagrama de clases

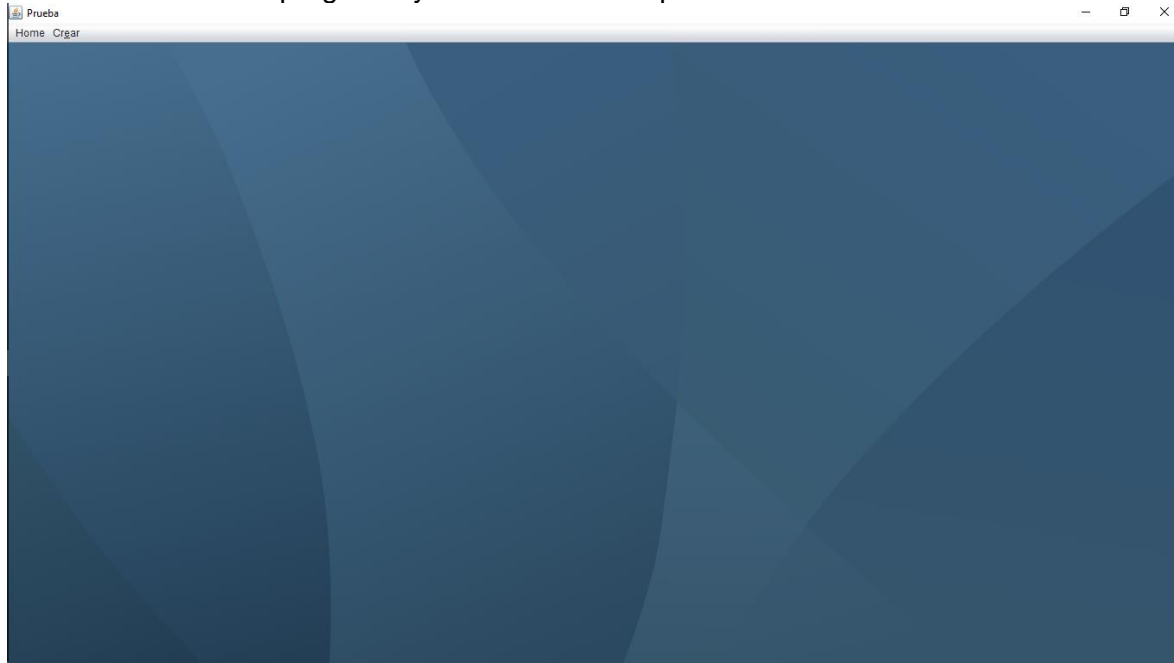


Para una mejor apreciación de la vista, se subirá la imagen al repositorio de GitHub.

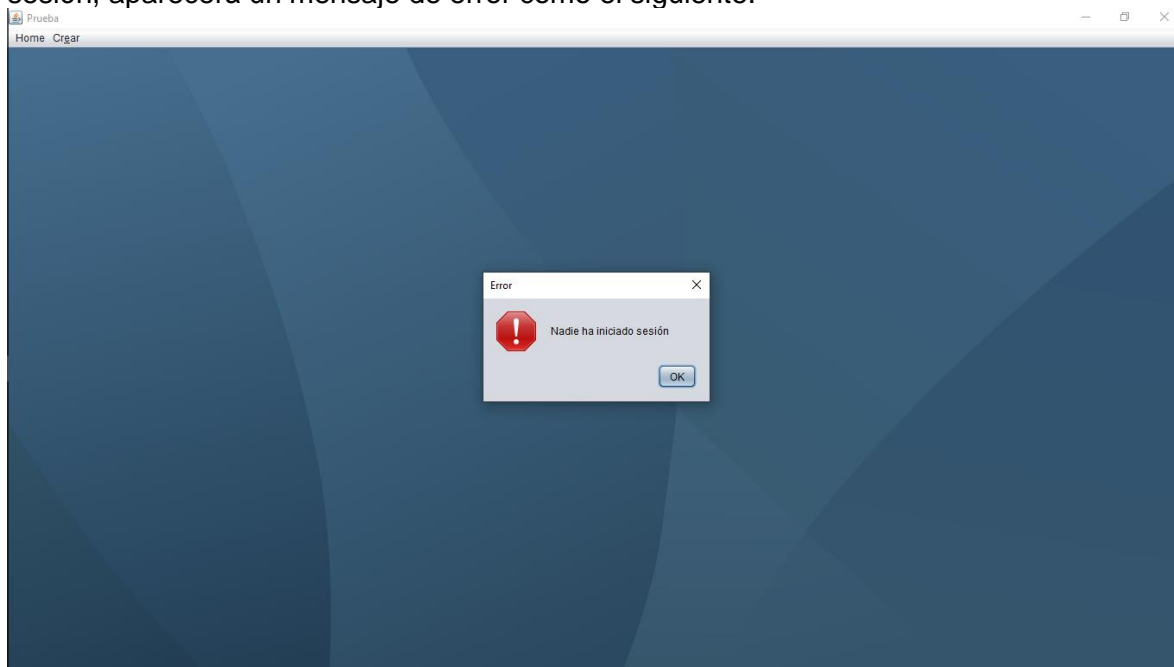
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


- Usabilidad – Vista

En este apartado vamos a explicar cómo funciona el programa.  
Primero iniciamos el programa y esta es la vista que obtenemos:

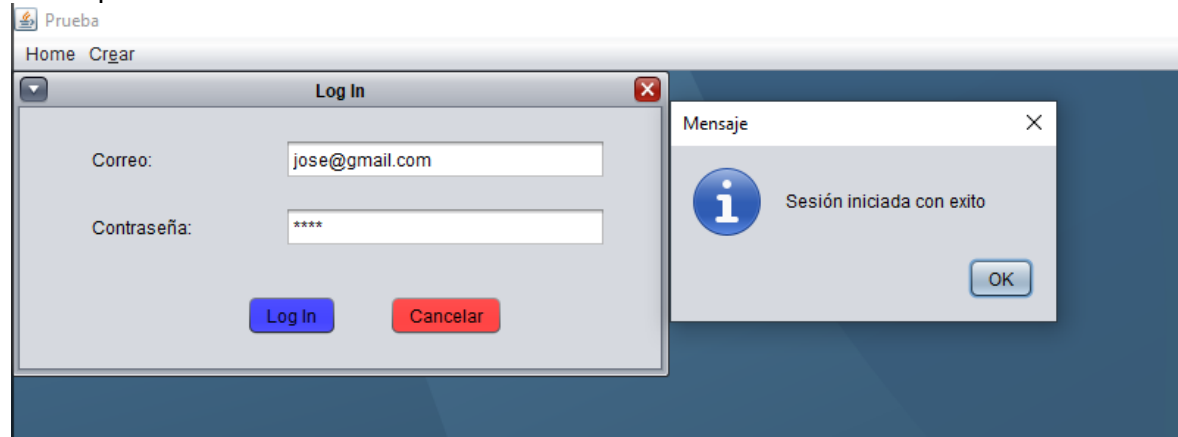


Ahora para iniciar sesión se requiere ser un administrador. Cabe recalcar que esta es la única ventana en la que podemos acceder ya que si intentamos abrir las otras y no hay un administrador iniciado sesión, aparecerá un mensaje de error como el siguiente:

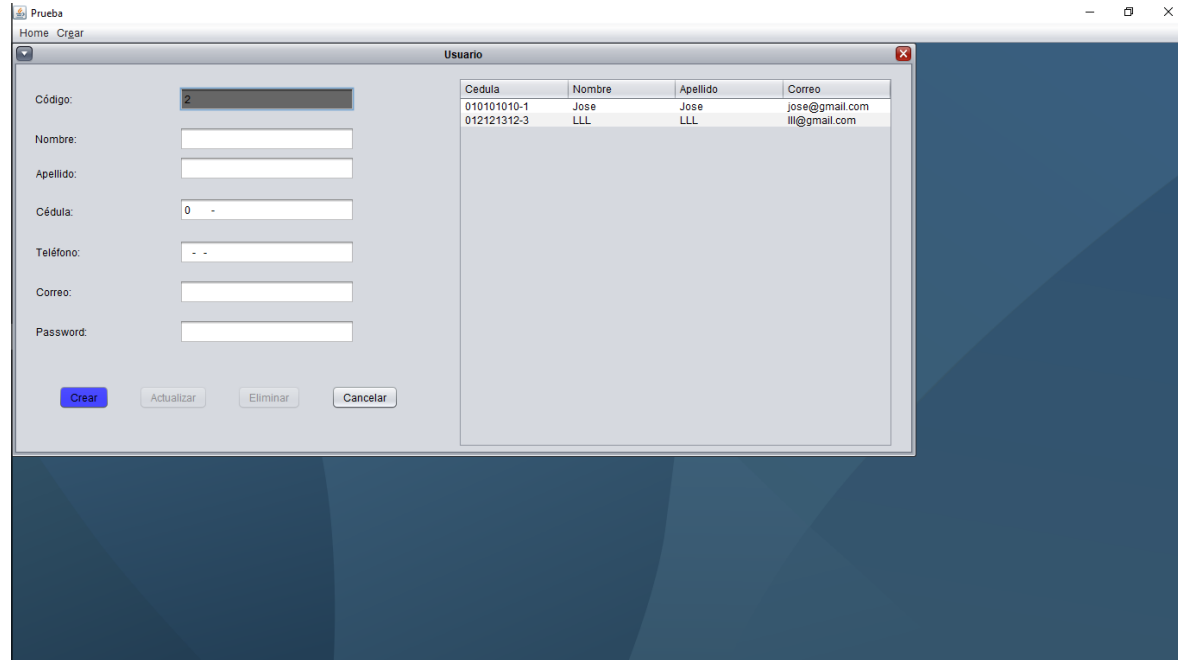


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Ahora procedemos a iniciar sesión:



La ventana del administrador:



Ahora vamos a crear un administrador:

Prueba  
Home Crgar

**Usuario**

Código: 2

Nombre: Josefina

Apellido: Carrasco

Cédula: 010101010-4

Teléfono: 098-756-5646

Correo: jf@gmail.com

Password: \*\*\*\*

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
012121312-3	LLL	LLL	lll@gmail.com

Mensaje

Administrador creado con éxito

OK

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
012121312-3	LLL	LLL	lll@gmail.com
010101010-4	Josefina	Carrasco	jf@gmail.com



Ahora procedemos a editar un administrador:

Prueba

Home Crear

Usuario

Código: 1

Nombre: Pepecho

Apellido: Malo

Cédula: 012121312-3

Teléfono: 094-897-8789

Correo: pepe@gmail.com

Password: \*\*\*\*

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
012121312-3	LLL	LLL	lll@gmail.com
010101010-4	Josefina	Carrasco	jf@gmail.com

Mensaje

Administrador actualizado con éxito

OK

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
010101010-4	Josefina	Carrasco	jf@gmail.com
012121312-3	Pepecho	Malo	pepe@gmail.com

Ahora eliminaremos a un administrador:

Prueba

Home Crear

Usuario

Código: 1

Nombre: Pepecho

Apellido: Malo

Cédula: 012121312-3

Teléfono: 094-897-8789

Correo: pepe@gmail.com

Password: \*\*\*\*

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
010101010-4	Josefina	Carrasco	jf@gmail.com
012121312-3	Pepecho	Malo	pepe@gmail.com


Mensaje

Administrador eliminado con éxito

OK

Crear Actualizar Eliminar Cancelar

Cedula	Nombre	Apellido	Correo
010101010-1	Jose	Jose	jose@gmail.com
010101010-4	Josefina	Carrasco	jf@gmail.com

	<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 pasaremos a la ventana para crear personas:

Prueba  
Home Crear

Código: 3

Nombre:

Apellido:

Cédula: 0 -

Nacimiento: 30 ene. 2021

Teléfono: - -

Sueldo:

Dirección:

Tipo: --Seleccione--

**Crear** Actualizar Eliminar Cancelar

**Cientes**

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-3	PPP	PPP	2600.23	Garante
022222222-1	Ete	Sech	1200.56	Garante

Crear una persona:

Prueba  
Home Crear

Código: 3

Nombre: Lola

Apellido: Gonzalez

Cédula: 022222222-4

Nacimiento: 30 ene. 1991

Teléfono: 098-452-1647

Sueldo: 2500.31

Dirección: Av. 12 de Abril


Tipo: Propietario

**Crear** Actualizar Eliminar Cancelar


**Cientes**

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-3	PPP	PPP	2600.23	Garante
022222222-1	Ete	Sech	1200.56	Garante

**Mensaje**

 Cliente creado con éxito

**OK**

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

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-3	PPP	PPP	2600.23	Garante
022222222-1	Ete	Sech	1200.56	Garante
022222222-4	Lola	Gonzalez	2500.31	Propietario

Ahora procedemos a editar a una persona:

Prueba

Home Crear

Código: 2

Nombre: Piero

Apellido: Peña

Cédula: 022222222-3

Nacimiento: 29 ene. 1956

Teléfono: 069-849-8746

Sueldo: 1662.13

Dirección: Estadio


Tipo: Garante

Crear Actualizar Eliminar Cancelar


Cientes

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-3	PPP	PPP	2600.23	Garante
022222222-1	Ete	Sech	1200.56	Garante
022222222-4	Lola	Gonzalez	2500.31	Propietario

Mensaje

 Cliente actualizado con éxito

OK

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

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-1	Ete	Sech	1200.56	Garante
022222222-4	Lola	Gonzalez	2500.31	Propietario
022222222-3	Piero	Peña	1662.13	Garante

Ahora eliminaremos una persona:

Prueba

Home Crear

Código: 4

Nombre: aaaa

Apellido: aaa

Cédula: 016487987-9

Nacimiento: 30 ene. 2021

Teléfono: 068-498-7984

Sueldo: 1553.0

Dirección: Remigio

Tipo: Garante

Crear Actualizar Eliminar Cancelar


**Cientes**

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-1	Ete	Sech	1200.56	Garante
022222222-4	Lola	Gonzalez	2500.31	Propietario
022222222-3	Piero	Peña	1662.13	Garante
016487987-9	aaaa	aaa	1553.0	Garante

Mensaje

Cliente eliminado con éxito

OK

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

Cedula	Nombre	Apellido	Sueldo	Tipo
022222222-2	Lorena	Flores	4500.56	Propietario
022222222-1	Ete	Sech	1200.56	Garante
022222222-4	Lola	Gonzalez	2500.31	Propietario
022222222-3	Piero	Peña	1662.13	Garante

Ahora procedemos con la ventana de la casa:

Prueba  
Home Cargar

Código:

Dirección:

Avaluo:

M² de construcción:

Número habitaciones:

Número pisos:

Estado:

Código	Dirección	Avaluo	Habitaciones	m² construcc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA

## Ahora creamos una casa:

Prueba  
Home Cargar

Código:

Dirección:

Avaluo:

M² de construcción:


Número habitaciones:

Número pisos:

Estado:

Código	Direccion	Avaluo	Habitaciones	m² construc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA

Mensaje

 Casa creada con éxito

Código	Direccion	Avaluo	Habitaciones	m² construc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA
3	Primero de ...	126.000	4	225,31	Enventa

## Ahora editamos una casa:

Prueba

Home Cargar

Código:

Dirección:

Avaluo:

M² de construcción:


Número habitaciones:

Número pisos:

Estado:

Código	Dirección	Avaluo	Habitaciones	m² construcc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA
3	Primero de ...	126.000	4	225,31	Enventa
4	San Joaquin	189.654	3	200	Enventa
5	Vista linda	236.000	5	451,32	Enventa

Mensaje

 Casa actualizada con éxito

Código	Dirección	Avaluo	Habitaciones	m² construc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA
4	San Joaquin	189.654	3	200	Enventa
5	Vista linda	236.000	5	451,32	Enventa
3	Primero de ...	156.000	5	225,31	Enventa



## Ahora eliminamos una casa:

Prueba  
Home Cargar

Código: 5

Dirección: Vista linda

Avaluo: 236000.0

M² de construcción: 451.32

Número habitaciones: 5

Número pisos: 3


Estado: Enventa

Crear Actualizar Eliminar Cancelar

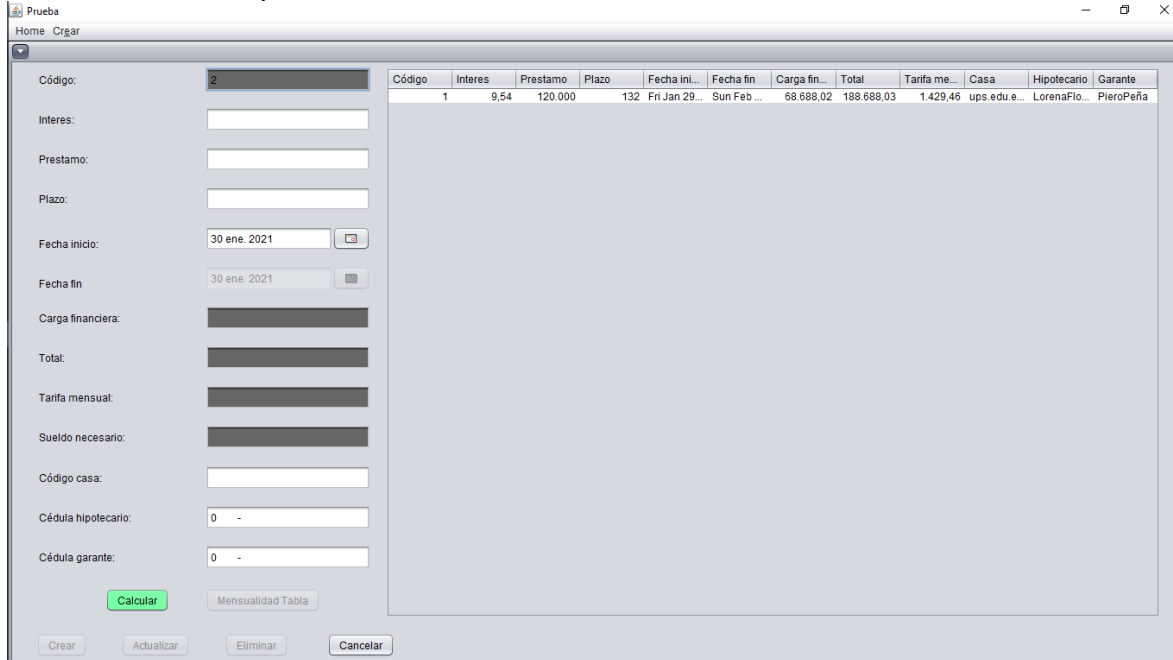
Código	Dirección	Avaluo	Habitaciones	m² construcc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA
4	San Joaquin	189.654	3	200	Enventa
5	Vista linda	236.000	5	451,32	Enventa
3	Primero de ...	156.000	5	225,31	Enventa

Mensaje  
Casa eliminada con éxito  
OK

Código	Direccion	Avaluo	Habitaciones	m^2 construc...	Estado
1	Remigio	226.500,35	5	321,2	VENDIDA
4	San Joaquin	189.654	3	200	Enventa
3	Primero de ...	156.000	5	225,31	Enventa

	<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 la ventana hipoteca:



Prueba

Home Crgar

Código: 2

Interes:

Prestamo:

Plazo:

Fecha inicio: 30 ene. 2021

Fecha fin: 30 ene. 2021

Carga financiera:

Total:

Tarifa mensual:

Sueldo necesario:

Código casa:

Cédula hipotecario: 0 -

Cédula garante: 0 -

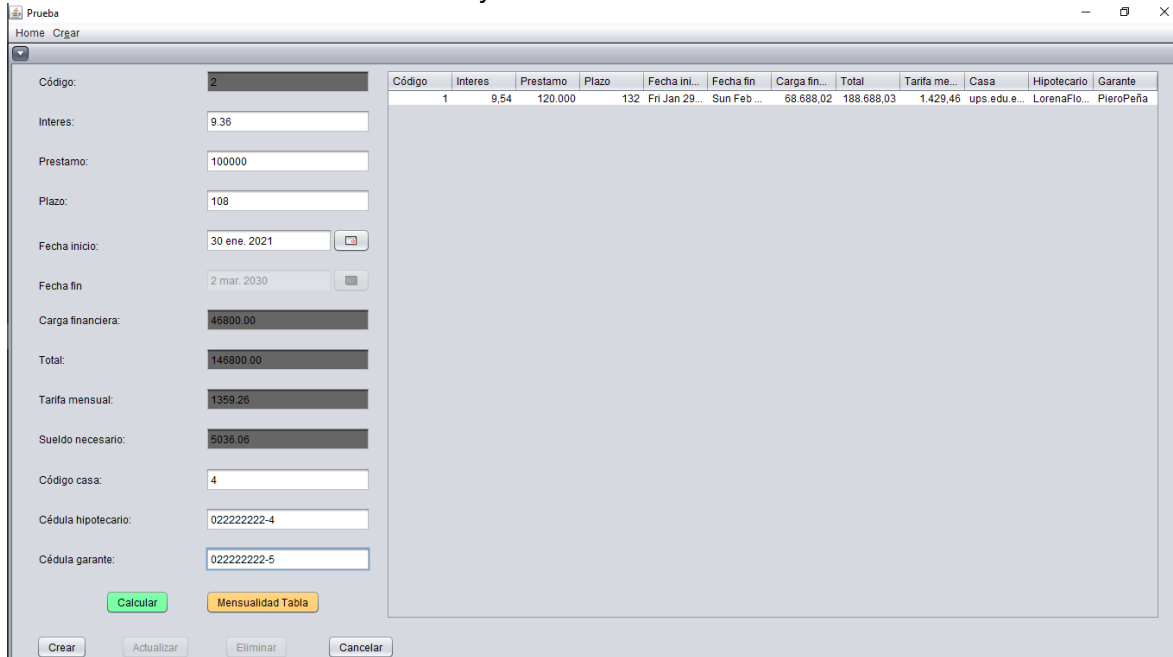
Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9,54	120.000	132	Fri Jan 29...	Sun Feb ...	68.688,02	188.688,03	1.429,46	ups.edu.e...	LorenaFlo...	PieroPeña

Crear una hipoteca:

Primero calculamos todos los datos y llenamos los datos:



Prueba

Home Crgar

Código: 2

Interes: 9.36

Prestamo: 100000

Plazo: 108

Fecha inicio: 30 ene. 2021

Fecha fin: 2 mar. 2030

Carga financiera: 46800.00

Total: 146800.00

Tarifa mensual: 1359.26

Sueldo necesario: 5036.06

Código casa: 4

Cédula hipotecario: 02222222-4


Cédula garante: 02222222-5

Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9,54	120.000	132	Fri Jan 29...	Sun Feb ...	68.688,02	188.688,03	1.429,46	ups.edu.e...	LorenaFlo...	PieroPeña

Ahora aplastamos el botón mensualidad tabla para ver los pagos:

	<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		

Prueba

Home Crgar

Fecha	Mensualidad	Valor restante
30/01/3921	0	146900.0
02/03/3921	1359.78	145440.22
30/03/3921	1359.78	144080.44
30/04/3921	1359.78	142720.66
30/05/3921	1359.78	141360.88
30/06/3921	1359.78	140001.10
30/07/3921	1359.78	138641.32
30/08/3921	1359.78	137281.54
30/09/3921	1359.78	135921.76
30/10/3921	1359.78	134561.98
30/11/3921	1359.78	133202.20
30/12/3921	1359.78	131842.42
30/01/3922	1359.78	130482.65
02/03/3922	1359.78	129122.87
30/03/3922	1359.78	127763.09
30/04/3922	1359.78	126403.31
30/05/3922	1359.78	125043.53
30/06/3922	1359.78	123683.75
30/07/3922	1359.78	122323.97
30/08/3922	1359.78	120964.19
30/09/3922	1359.78	119604.41
30/10/3922	1359.78	118244.63
30/11/3922	1359.78	116884.85
30/12/3922	1359.78	115525.07
30/01/3923	1359.78	114165.30
02/03/3923	1359.78	112805.52
30/03/3923	1359.78	111445.74
30/04/3923	1359.78	110085.96

Fecha ini...	Fecha fin...	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
2	Fri Jan 29...	Sun Feb ...	68 688.02	188 688.03	1 429.46	ups.edu...	LorenaFlo... PieroPeña

Cédula hipotecario: 022222222-4

Cédula garante: 022222222-5

Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

La tabla cuenta con los 108 meses de plazo que se tiene que pagar. Ahora aplastamos el botón crear:

Prueba

Home Crgar

Código: 2

Interes: 9.36

Prestamo: 100000

Plazo: 108

Fecha inicio: 30 ene. 2021

Fecha fin: 2 mar. 2030

Carga financiera: 46800.00

Total: 146900.00

Tarifa mensual: 1359.26

Sueldo necesario: 5036.06

Código casa: 4

Cédula hipotecario: 022222222-4

Cédula garante: 022222222-5

Calcular Mensualidad Tabla


Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin...	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9.54	120 000	132	Fri Jan 29...	Sun Feb ...	68 688.02	188 688.03	1 429.46	ups.edu...	LorenaFlo...	PieroPeña

Mensaje

Hipoteca creada 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		

Prueba

Home Crgar

Código: 4

Interes:

Prestamo:

Plazo:

Fecha inicio: 30 ene. 2021

Fecha fin: 30 ene. 2021

Carga financiera:

Total:

Tarifa mensual:

Sueldo necesario:

Código casa:

Cédula hipotecario: 0 -

Cédula garante: 0 -

Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9.54	120.000	132	Fri Jan 29...	Sun Feb...	68.688.02	188.688.03	1.429.46	ups.edu.e...	LorenaFlo...	PieroPeña
3	9.36	100.000	108	Sat Jan 3...	Sat Mar 0...	46.800	146.800	1.359.26	ups.edu.e...	LolaGonz...	IrinaTam...

Como podemos observar se tuvo que agregar un garante ya que el hipotecario no tenía el sueldo necesario para poder acceder al préstamo hipotecario.

Ahora editaremos una hipoteca:

Prueba

Home Crgar

Código: 3

Interes: 9.64

Prestamo: 120000

Plazo: 120

Fecha inicio: 30 ene. 2021

Fecha fin: 2 mar. 2031

Carga financiera: 63624.00

Total: 183624.00

Tarifa mensual: 1530.20

Sueldo necesario: 5869.40

Código casa: 4

Cédula hipotecario: 02222222-4

Cédula garante: 02222222-5

Calcular Mensualidad Tabla


Crear Actualizar Eliminar Cancelar

Mensaje

Hipoteca actualizada con éxito

OK

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9.54	120.000	132	Fri Jan 29...	Sun Feb...	68.688.02	188.688.03	1.429.46	ups.edu.e...	LorenaFlo...	PieroPeña
3	9.36	100.000	108	Sat Jan 3...	Sat Mar 0...	46.800	146.800	1.359.26	ups.edu.e...	LolaGonz...	IrinaTam...

	<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		

Prueba  
Home Crgar

Código: 4

Interes:

Prestamo:

Plazo:

Fecha inicio: 30 ene. 2021

Fecha fin: 30 ene. 2021

Carga financiera:

Total:

Tarifa mensual:

Sueldo necesario:

Código casa:

Cédula hipotecario: 0 -

Cédula garante: 0 -

Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin...	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9.54	120.000	132	Fri Jan 29...	Sun Feb...	68.688.02	188.688.03	1.429.46	ups.edu.e...	LorenaFlo...	PieroPeña
3	9.64	120.000	120	Sat Jan 3...	Sun Mar 0...	63.624	183.624	1.530.2	ups.edu.e...	LolaGonz...	IrinaTam...

Ahora eliminamos una hipoteca:

Prueba  
Home Crgar

Código: 1

Interes: 9.54

Prestamo: 120000.00

Plazo: 132

Fecha inicio: 29 ene. 2021

Fecha fin: 29 feb. 2032

Carga financiera: 68688.02

Total: 188688.03

Tarifa mensual: 1429.46

Sueldo necesario: 5295.15

Código casa: 1

Cédula hipotecario: 02222222-2


Cédula garante: 02222222-3

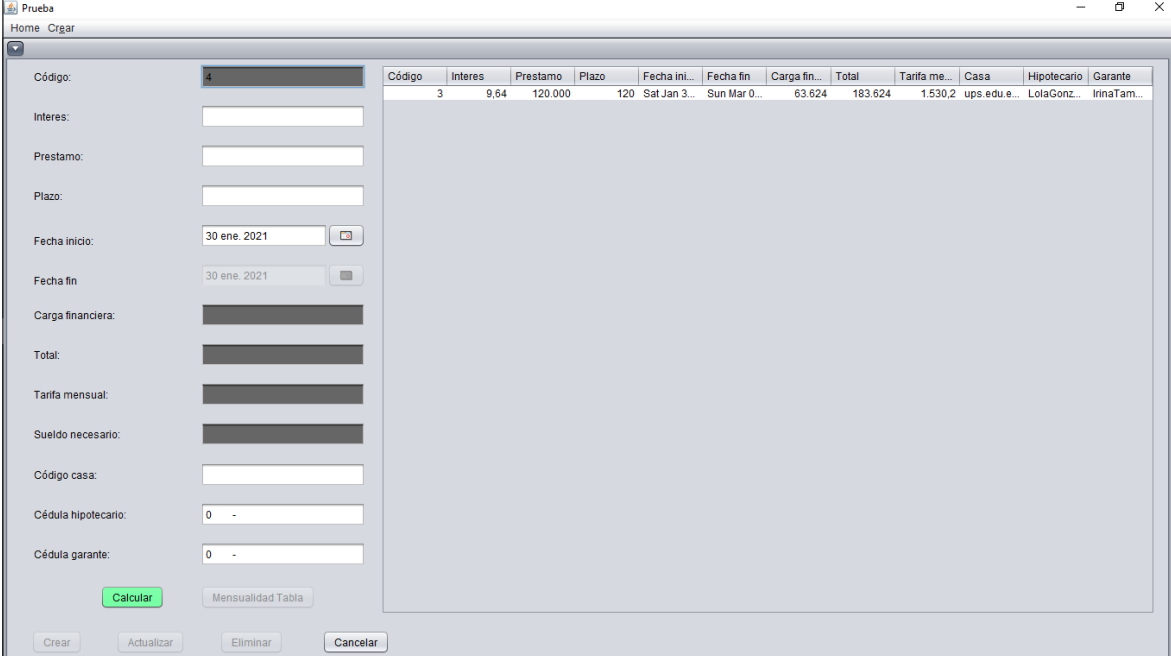
Calcular Mensualidad Tabla

Crear Actualizar Eliminar Cancelar

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin...	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
1	9.54	120.000	132	Fri Jan 29...	Sun Feb...	68.688.02	188.688.03	1.429.46	ups.edu.e...	LorenaFlo...	PieroPeña
3	9.64	120.000	120	Sat Jan 3...	Sun Mar 0...	63.624	183.624	1.530.2	ups.edu.e...	LolaGonz...	IrinaTam...

Mensaje  
i Hipoteca 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		



The application window 'Prueba' has a menu bar with 'Home' and 'Cargar'. The form on the left includes fields for:

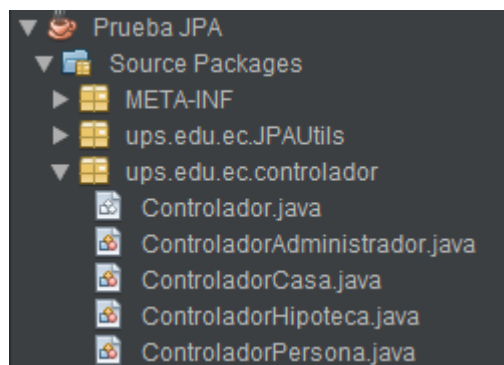
- Código: 4
- Interes: (empty)
- Prestamo: (empty)
- Plazo: (empty)
- Fecha inicio: 30 ene. 2021
- Fecha fin: 30 ene. 2021
- Carga financiera: (empty)
- Total: (empty)
- Tarifa mensual: (empty)
- Sueldo necesario: (empty)
- Código casa: (empty)
- Cédula hipotecario: 0 -
- Cédula garante: 0 -

Buttons at the bottom: Calcular (green), Mensualidad Tabla, Crear, Actualizar, Eliminar, Cancelar.

Código	Interes	Prestamo	Plazo	Fecha ini...	Fecha fin	Carga fin...	Total	Tarifa me...	Casa	Hipotecario	Garante
3	9,64	120.000	120	Sat Jan 3...	Sun Mar 0...	63.624	183.624	1.530,2	ups.edu...	LolaGonz...	IrinaTam...


- Programación genérica, Java 8, reflexión

Programación genérica:



Como podemos observar el controlador es abstracto y se extiende a las otras clases hijas. A continuación, el código de la clase:

```
/*
 * 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.lang.reflect.ParameterizedType;
import java.util.List;
import javax.persistence.EntityManager;
import ups.edu.ec.JPAUtils.JPAUtils;

/**
 *
 * @author User
 * @param <E>
 */
public abstract class Controlador<E> {


    private EntityManager em;
    private Class<E> clase;

    public Controlador() {
        this.em = (EntityManager) JPAUtils.getEmf();

        java.lang.reflect.Type t = getClass().getGenericSuperclass();
        ParameterizedType pt = (ParameterizedType) t;

        clase = (Class) pt.getActualTypeArguments()[0];
    }

    public EntityManager getEm() {
```

	<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		

```

return em;
}

public void setEm(EntityManager em) {
    this.em = em;
}

public boolean create(E objeto) {

    try {

        em.getTransaction().begin();
        em.persist(objeto);
        em.getTransaction().commit();
        return true;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }


}

public E read(int id) {
    return em.find(clase, id);
}

public boolean update(E objeto) {

```



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

```

em.getTransaction().begin();
em.merge(objeto);
em.getTransaction().commit();
return true;

}

public boolean delete(E objeto) {
    em.getTransaction().begin();
    if (!em.contains(objeto)) {
        objeto = em.merge(objeto);
    }

    em.remove(objeto);
    em.getTransaction().commit();
    return true;
}


/*
public List<E> findAll() {
    List<E> lista = new ArrayList<>();
    lista = em.createQuery("Select t from " + clase.getSimpleName() + " t").getResultList();

    return lista;

}*/

public abstract List<E> findAll();

```

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

```
public abstract int getCodigo();

}
```

### Reflexión:

Esto se utilizó para poder crear administradores ya que se utilizó el patrón de diseño singleton:

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

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

### Java 8:

Funciones lambda:

```
@Override
public int getCodigo() {
    var lista = findAll();

    Collections.sort(lista, (Administrador f1, Administrador f2) -> f1.getCodigo() - (f2.getCodigo()));

    return lista.get(lista.size() - 1).getCodigo();
}
```

```
public Administrador iniciarSesion(String correo, String password, String keyUnlock) {
    var listaAdmins = (List<Administrador>) findAll();
    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;
    }
}
}
```

Encriptar y desencriptar contraseña:

```
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");

        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 descriptarPassword(String key, String passwordEncriptado) {

    String desincriptacion = "";

    try {

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

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

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

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

        return desincriptacion;

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

    return desincriptacion;
}
```

### 3. Código de la prueba:

**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;


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

```

import java.io.Serializable;
import java.lang.reflect.Constructor;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;

/**
 *
 * @author User
 */
@Entity
@Table(name = "administrador")
@NamedQueries({
    @NamedQuery(name = "Administrador.findAll", query = "SELECT a FROM Administrador a"),
    @NamedQuery(name = "Administrador.findByCodigo", query = "SELECT a FROM Administrador a WHERE a.codigo = :codigo"),
    @NamedQuery(name = "Administrador.findByCedula", query = "SELECT a FROM Administrador a WHERE a.cedula = :cedula"),
    @NamedQuery(name = "Administrador.findByNombre", query = "SELECT a FROM Administrador a WHERE a.nombre = :nombre"),
    @NamedQuery(name = "Administrador.findByApellido", query = "SELECT a FROM Administrador a WHERE a.apellido = :apellido"),
    @NamedQuery(name = "Administrador.findByNumeroTelefono", query = "SELECT a FROM Administrador a WHERE a.numeroTelefono = :numeroTelefono"),

```

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

```

@NamedQuery(name = "Administrador.findByCorreo", query = "SELECT a FROM Administrador a WHERE
a.correo = :correo"),

@NamedQuery(name = "Administrador.findByPassword", query = "SELECT a FROM Administrador a WHERE
a.password = :password"))

public class Administrador implements Serializable {

    private static final long serialVersionUID = 1L;

    @Basic(optional = false)
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "codigo")
    private int codigo;

    @Id
    @Basic(optional = false)
    @Column(name = "cedula")
    private String cedula;

    @Column(name = "nombre")
    private String nombre;

    @Column(name = "apellido")
    private String apellido;

    @Column(name = "numero_telefono")
    private String numeroTelefono;


    @Column(name = "correo")
    private String correo;

    @Column(name = "password")
    private String password;

    public static Administrador instance;

    public Administrador() {
    }

```

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

```

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

public Administrador(String cedula, int codigo) {
    this.cedula = cedula;
    this.codigo = codigo;
}


public Administrador(String cedula, String nombre, String apellido, String numeroTelefono, String correo) {
    this.cedula = cedula;
    this.nombre = nombre;
    this.apellido = apellido;
    this.numeroTelefono = numeroTelefono;
    this.correo = correo;
}

public Administrador getInstance() {
    if (Administrador.instance == null) {
        Constructor<Administrador> constructor;
        try {
            constructor = Administrador.class.getDeclaredConstructor();
            constructor.setAccessible(true);
            Administrador admin = constructor.newInstance();

            Administrador.instance = admin;

            return Administrador.instance;
        }
    }
}

```

	<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		

```

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

    return Administrador.instance;
}

return Administrador.instance;
}

public int getCodigo() {
    return codigo;
}

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


public String getCedula() {
    return cedula;
}

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

public String getNombre() {
    return nombre;
}

```



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

```

}

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

public String getApellido() {
    return apellido;
}

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


public String getNumeroTelefono() {
    return numeroTelefono;
}

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

public String getCorreo() {
    return correo;
}

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

```

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

```


public String getPassword() {
    return password;
}

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

@Override
public int hashCode() {
    int hash = 0;
    hash += (cedula != null ? cedula.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Administrador)) {
        return false;
    }
    Administrador other = (Administrador) object;
    if ((this.cedula == null && other.cedula != null) || (this.cedula != null && !this.cedula.equals(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 "ups.edu.ec.modelo.Administrador[ cedula=" + cedula + " ]";
}


}
```

**Casa:**

```
/*
 * 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.math.BigDecimal;
import java.util.Collection;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Table;
```

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

```


/**
 *
 * @author User
 */
@Entity
@Table(name = "casa")
@NamedQueries({
    @NamedQuery(name = "Casa.findAll", query = "SELECT c FROM Casa c"),
    @NamedQuery(name = "Casa.findByCodigo", query = "SELECT c FROM Casa c WHERE c.codigo = :codigo"),
    @NamedQuery(name = "Casa.findByDireccion", query = "SELECT c FROM Casa c WHERE c.direccion = :direccion"),
    @NamedQuery(name = "Casa.findByAvaluo", query = "SELECT c FROM Casa c WHERE c.avaluo = :avaluo"),
    @NamedQuery(name = "Casa.findByMetrosConstruccion", query = "SELECT c FROM Casa c WHERE c.metrosConstruccion = :metrosConstruccion"),
    @NamedQuery(name = "Casa.findByNumeroHabitaciones", query = "SELECT c FROM Casa c WHERE c.numeroHabitaciones = :numeroHabitaciones"),
    @NamedQuery(name = "Casa.findByNumeroPisos", query = "SELECT c FROM Casa c WHERE c.numeroPisos = :numeroPisos"),
    @NamedQuery(name = "Casa.findByEstado", query = "SELECT c FROM Casa c WHERE c.estado = :estado"))})
public class Casa implements Serializable {

    @OneToMany(mappedBy = "codigoCasaFk")
    private Collection<Hipoteca> hipotecaCollection;

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "codigo")
    private Integer codigo;

```

	<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		

```

@Column(name = "direccion")
private String direccion;

// @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to
enforce field validation

@Column(name = "avaluo")
private BigDecimal avaluo;

@Column(name = "metros_construccion")
private BigDecimal metrosConstruccion;

@Column(name = "numero_habitaciones")
private Integer numeroHabitaciones;

@Column(name = "numero_pisos")
private Integer numeroPisos;


@Column(name = "estado")
private String estado;

public Casa() {
}

public Casa(String direccion, BigDecimal avaluo, BigDecimal metrosConstruccion, Integer numeroHabitaciones,
Integer numeroPisos, String estado) {
    this.direccion = direccion;
    this.avaluo = avaluo;
    this.metrosConstruccion = metrosConstruccion;
    this.numeroHabitaciones = numeroHabitaciones;
    this.numeroPisos = numeroPisos;
    this.estado = estado;
}

public Casa(Integer codigo) {
    this.codigo = codigo;
}

```

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

```

}

public Integer getCodigo() {
    return codigo;
}

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

public String getDireccion() {
    return direccion;
}


public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public BigDecimal getAvaluo() {
    return avaluo;
}

public void setAvaluo(BigDecimal avaluo) {
    this.avaluo = avaluo;
}

public BigDecimal getMetrosConstruccion() {
    return metrosConstruccion;
}

```

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

```

public void setMetrosConstruccion(BigDecimal metrosConstruccion) {
    this.metrosConstruccion = metrosConstruccion;
}

public Integer getNumeroHabitaciones() {
    return numeroHabitaciones;
}

public void setNumeroHabitaciones(Integer numeroHabitaciones) {
    this.numeroHabitaciones = numeroHabitaciones;
}


public Integer getNumeroPisos() {
    return numeroPisos;
}

public void setNumeroPisos(Integer numeroPisos) {
    this.numeroPisos = numeroPisos;
}

public String getEstado() {
    return estado;
}

public void setEstado(String estado) {
    this.estado = estado;
}

```

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

@Override

```
public int hashCode() {
    int hash = 0;
    hash += (codigo != null ? codigo.hashCode() : 0);
    return hash;
}
```

@Override


```
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Casa)) {
        return false;
    }
    Casa other = (Casa) object;
    if ((this.codigo == null && other.codigo != null) || (this.codigo != null && !this.codigo.equals(other.codigo))) {
        return false;
    }
    return true;
}
```

@Override

```
public String toString() {
    return "ups.edu.ec.modelo.Casa[ codigo=" + codigo + " ]";
}
```

```
public Collection<Hipoteca> getHipotecaCollection() {
    return hipotecaCollection;
}
```



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

```

public void setHipotecaCollection(Collection<Hipoteca> hipotecaCollection) {
    this.hipotecaCollection = hipotecaCollection;
}

}

```


### Hipoteca:

```

/*
 * 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.math.BigDecimal;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.Table;
import javax.persistence.Temporal;

```

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


```

import javax.persistence.TemporalType;

/**
 *
 * @author User
 */
@Entity
@Table(name = "hipoteca")
@NamedQueries({
    @NamedQuery(name = "Hipoteca.findAll", query = "SELECT h FROM Hipoteca h"),
    @NamedQuery(name = "Hipoteca.findByCodigo", query = "SELECT h FROM Hipoteca h WHERE h.codigo = :codigo"),
    @NamedQuery(name = "Hipoteca.findByFechaInicio", query = "SELECT h FROM Hipoteca h WHERE h.fechaInicio = :fechaInicio"),
    @NamedQuery(name = "Hipoteca.findByFechaFin", query = "SELECT h FROM Hipoteca h WHERE h.fechaFin = :fechaFin"),
    @NamedQuery(name = "Hipoteca.findByPlazoMeses", query = "SELECT h FROM Hipoteca h WHERE h.plazoMeses = :plazoMeses"),
    @NamedQuery(name = "Hipoteca.findByMontoPrestamo", query = "SELECT h FROM Hipoteca h WHERE h.montoPrestamo = :montoPrestamo"),
    @NamedQuery(name = "Hipoteca.findByInteres", query = "SELECT h FROM Hipoteca h WHERE h.interes = :interes"),
    @NamedQuery(name = "Hipoteca.findByCargaFinanciera", query = "SELECT h FROM Hipoteca h WHERE h.cargaFinanciera = :cargaFinanciera"),
    @NamedQuery(name = "Hipoteca.findByPagoTotal", query = "SELECT h FROM Hipoteca h WHERE h.pagoTotal = :pagoTotal"),
    @NamedQuery(name = "Hipoteca.findByTarifaMensual", query = "SELECT h FROM Hipoteca h WHERE h.tarifaMensual = :tarifaMensual"),
    @NamedQuery(name = "Hipoteca.findByGarante", query = "SELECT h FROM Hipoteca h WHERE h.garante = :garante")))
public class Hipoteca implements Serializable {

    @Column(name = "sueldo_necesario")

```

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

```

private BigDecimal sueldoNecesario;


private static final long serialVersionUID = 1L;

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Basic(optional = false)
@Column(name = "codigo")
private Integer codigo;
@Column(name = "fecha_inicio")
@Temporal(TemporalType.DATE)
private Date fechaInicio;
@Column(name = "fecha_fin")
@Temporal(TemporalType.DATE)
private Date fechaFin;
@Column(name = "plazo_meses")
private Integer plazoMeses;

// @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to
enforce field validation

@Column(name = "monto_prestamo")
private BigDecimal montoPrestamo;
@Column(name = "interes")
private BigDecimal interes;
@Column(name = "carga_financiera")
private BigDecimal cargaFinanciera;
@Column(name = "pago_total")
private BigDecimal pagoTotal;
@Column(name = "tarifa_mensual")
private BigDecimal tarifaMensual;
@Column(name = "garante")

```

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

```

private Boolean garante;

@JoinColumn(name = "codigo_casa_fk", referencedColumnName = "codigo")
@ManyToOne
private Casa codigoCasaFk;

@JoinColumn(name = "cedula_garante_fk", referencedColumnName = "cedula")
@ManyToOne
private Persona cedulaGaranteFk;


@JoinColumn(name = "cedula_propietario_fk", referencedColumnName = "cedula")
@ManyToOne
private Persona cedulaPropietarioFk;

public Hipoteca() {
}

public Hipoteca(Integer codigo) {
    this.codigo = codigo;
}

public Hipoteca(BigDecimal sueldoNecesario, Date fechalnicio, Date fechaFin, Integer plazoMeses, BigDecimal
montoPrestamo, BigDecimal interes, BigDecimal cargaFinanciera, BigDecimal pagoTotal, BigDecimal
tarifaMensual, Boolean garante, Casa codigoCasaFk, Persona cedulaGaranteFk, Persona cedulaPropietarioFk) {
    this.sueldoNecesario = sueldoNecesario;
    this.fechalnicio = fechalnicio;
    this.fechaFin = fechaFin;
    this.plazoMeses = plazoMeses;
    this.montoPrestamo = montoPrestamo;
    this.interes = interes;
    this.cargaFinanciera = cargaFinanciera;
    this.pagoTotal = pagoTotal;
    this.tarifaMensual = tarifaMensual;
}

```

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

```

this.garante = garante;

this.codigoCasaFk = codigoCasaFk;

this.cedulaGaranteFk = cedulaGaranteFk;

this.cedulaPropietarioFk = cedulaPropietarioFk;

}

public Integer getCodigo() {

    return codigo;

}

public void setCodigo(Integer codigo) {

    this.codigo = codigo;

}

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;

}

```

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

```

public Integer getPlazoMeses() {
    return plazoMeses;
}

public void setPlazoMeses(Integer plazoMeses) {
    this.plazoMeses = plazoMeses;
}

public BigDecimal getMontoPrestamo() {
    return montoPrestamo;
}


public void setMontoPrestamo(BigDecimal montoPrestamo) {
    this.montoPrestamo = montoPrestamo;
}

public BigDecimal getInteres() {
    return interes;
}

public void setInteres(BigDecimal interes) {
    this.interes = interes;
}

public BigDecimal getCargaFinanciera() {
    return cargaFinanciera;
}

```

	<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 setCargaFinanciera(BigDecimal cargaFinanciera) {
```

```
    this.cargaFinanciera = cargaFinanciera;
```

```
}
```

```
public BigDecimal getPagoTotal() {
```

```
    return pagoTotal;
```

```
}
```

```
public void setPagoTotal(BigDecimal pagoTotal) {
```

```
    this.pagoTotal = pagoTotal;
```

```
}
```

```
public BigDecimal getTarifaMensual() {
```

```
    return tarifaMensual;
```

```
}
```

```
public void setTarifaMensual(BigDecimal tarifaMensual) {
```

```
    this.tarifaMensual = tarifaMensual;
```

```
}
```

```
public Boolean getGarante() {
```

```
    return garante;
```


```
}
```

```
public void setGarante(Boolean garante) {
```

```
    this.garante = garante;
```

```
}
```

```
public Casa getCodigoCasaFk() {
```

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

```

return codigoCasaFk;
}

public void setCodigoCasaFk(Casa codigoCasaFk) {
    this.codigoCasaFk = codigoCasaFk;
}

public Persona getCedulaGaranteFk() {
    return cedulaGaranteFk;
}

public void setCedulaGaranteFk(Persona cedulaGaranteFk) {
    this.cedulaGaranteFk = cedulaGaranteFk;
}


public Persona getCedulaPropietarioFk() {
    return cedulaPropietarioFk;
}

public void setCedulaPropietarioFk(Persona cedulaPropietarioFk) {
    this.cedulaPropietarioFk = cedulaPropietarioFk;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (codigo != null ? codigo.hashCode() : 0);
    return hash;
}

```



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

@Override


```
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Hipoteca)) {
        return false;
    }
    Hipoteca other = (Hipoteca) object;
    if ((this.codigo == null && other.codigo != null) || (this.codigo != null && !this.codigo.equals(other.codigo))) {
        return false;
    }
    return true;
}
```

@Override

```
public String toString() {
    return "ups.edu.ec.modelo.Hipoteca[ codigo=" + codigo + " ]";
}

public BigDecimal getSueldoNecesario() {
    return sueldoNecesario;
}

public void setSueldoNecesario(BigDecimal sueldoNecesario) {
    this.sueldoNecesario = sueldoNecesario;
}
}
```

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

#### 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.math.BigDecimal;

import java.util.Collection;

import java.util.Date;

import javax.persistence.Basic;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.NamedQueries;

import javax.persistence.NamedQuery;

import javax.persistence.OneToOne;

import javax.persistence.Table;

import javax.persistence.Temporal;


import javax.persistence.TemporalType;

/\*\*

\*

\* @author User

\*/

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

@Entity

@Table(name = "persona")

@NamedQueries({

    @NamedQuery(name = "Persona.findAll", query = "SELECT p FROM Persona p"),

    @NamedQuery(name = "Persona.findByCodigo", query = "SELECT p FROM Persona p WHERE p.codigo = :codigo"),

    @NamedQuery(name = "Persona.findByCedula", query = "SELECT p FROM Persona p WHERE p.cedula = :cedula"),

    @NamedQuery(name = "Persona.findByNombre", query = "SELECT p FROM Persona p WHERE p.nombre = :nombre"),

    @NamedQuery(name = "Persona.findByApellido", query = "SELECT p FROM Persona p WHERE p.apellido = :apellido"),

    @NamedQuery(name = "Persona.findByNumeroTelefono", query = "SELECT p FROM Persona p WHERE p.numeroTelefono = :numeroTelefono"),

    @NamedQuery(name = "Persona.findByFechaNacimiento", query = "SELECT p FROM Persona p WHERE p.fechaNacimiento = :fechaNacimiento"),

    @NamedQuery(name = "Persona.findByDireccion", query = "SELECT p FROM Persona p WHERE p.direccion = :direccion"),

    @NamedQuery(name = "Persona.findBySueldo", query = "SELECT p FROM Persona p WHERE p.sueldo = :sueldo"),

    @NamedQuery(name = "Persona.findByTipo", query = "SELECT p FROM Persona p WHERE p.tipo = :tipo"))})

public class Persona implements Serializable {

    @OneToMany(mappedBy = "cedulaGaranteFk")

    private Collection<Hipoteca> hipotecaCollection;

    @OneToMany(mappedBy = "cedulaPropietarioFk")


    private Collection<Hipoteca> hipotecaCollection1;

    private static final long serialVersionUID = 1L;

    @Basic(optional = false)

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    @Column(name = "codigo")

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

```

private int codigo;

@Id

@Basic(optional = false)

@Column(name = "cedula")
private String cedula;

@Column(name = "nombre")
private String nombre;

@Column(name = "apellido")
private String apellido;

@Column(name = "numero_telefono")
private String numeroTelefono;

@Column(name = "fecha_nacimiento")
@Temporal(TemporalType.DATE)
private Date fechaNacimiento;

@Column(name = "direccion")
private String direccion;

// @Max(value=?) @Min(value=?)//if you know range of your decimal fields consider using these annotations to
enforce field validation

@Column(name = "sueldo")
private BigDecimal sueldo;

@Column(name = "tipo")
private String tipo;


public Persona() {
}

public Persona(String cedula, String nombre, String apellido, String numeroTelefono, Date fechaNacimiento,
String direccion, BigDecimal sueldo, String tipo) {

    this.cedula = cedula;

    this.nombre = nombre;

```

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

```

this.apellido = apellido;

this.numeroTelefono = numeroTelefono;

this.fechaNacimiento = fechaNacimiento;

this.direccion = direccion;

this.sueldo = sueldo;

this.tipo = tipo;
}

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


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

public int getCodigo() {
    return codigo;
}

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

public String getCedula() {
    return cedula;
}

```

	<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 setCedula(String cedula) {
    this.cedula = cedula;
}

public String getNombre() {
    return nombre;
}

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

public String getApellido() {
    return apellido;
}


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

public String getNumeroTelefono() {
    return numeroTelefono;
}

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

public Date getFechaNacimiento() {

```

	<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		

```

return fechaNacimiento;
}

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

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}


public BigDecimal getSueldo() {
    return sueldo;
}

public void setSueldo(BigDecimal sueldo) {
    this.sueldo = sueldo;
}

public String getTipo() {
    return tipo;
}

public void setTipo(String tipo) {
    this.tipo = tipo;
}

```

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

```
}
```

```
@Override
```

```
public int hashCode() {
    int hash = 0;
    hash += (cedula != null ? cedula.hashCode() : 0);
    return hash;
}
```

```
@Override
```


```
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Persona)) {
        return false;
    }
    Persona other = (Persona) object;
    if ((this.cedula == null && other.cedula != null) || (this.cedula != null && !this.cedula.equals(other.cedula))) {
        return false;
    }
    return true;
}
```

```
@Override
```

```
public String toString() {
    return "ups.edu.ec.modelo.Persona[ cedula=" + cedula + " ]";
}
```

```
public Collection<Hipoteca> getHipotecaCollection() {
    return hipotecaCollection;
}
```



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

```

}

public void setHipotecaCollection(Collection<Hipoteca> hipotecaCollection) {
    this.hipotecaCollection = hipotecaCollection;
}

public Collection<Hipoteca> getHipotecaCollection1() {
    return hipotecaCollection1;
}

public void setHipotecaCollection1(Collection<Hipoteca> hipotecaCollection1) {
    this.hipotecaCollection1 = hipotecaCollection1;
}
}

```

#### **Paquete controlador:**


#### **ControladorAdministrador:**

```

/*
 * 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;
import java.util.Collections;

```

	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 java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.persistence.EntityManager;
import javax.persistence.Query;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;
import org.apache.commons.codec.binary.Base64;
import ups.edu.ec.modelo.Administrador;


/**
 *
 * @author User
 */
public class ControladorAdministrador extends Controlador<Administrador> {

    private Pattern patron;
    private Matcher corpus;

    public ControladorAdministrador() {
        patron = Pattern.compile("^(\\w){2,}(\\@)(\\w){2,}(\\.\\w{2,})+$");
    }

    public boolean validarEmail(String email) {

```

	<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		

```
corpus = patron.matcher(email);
```

```
return corpus.find();
```

```
}
```

```
public Administrador findByCedula(String cedula) {
```

```
    Query consulta = getEm().createNamedQuery("Administrador.findByCedula");
```

```
    consulta.setParameter("cedula", cedula);
```

```
    return (Administrador) consulta.getSingleResult();
```

```
}
```

```
public Administrador iniciarSesion(String correo, String password, String keyUnlock) {
```

```
    var listaAdmins = (List<Administrador>) findAll();
```

```
    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;
```

```
    }
```

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

```

}

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");

        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();
    }
}

```

	<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		

```
}
```

```
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");
```

```
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);
```

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

```
return desincriptacion;
```

```
} catch (Exception e) {
```

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

```
e.printStackTrace();
```

```
}
```

```
return desincryptacion;
```

```
}
```

```
@Override
```

```
public List<Administrador> findAll() {
```

```
    Query consulta = getEm().createNamedQuery("Administrador.findAll");
```

```
    //System.out.println("rrrrr" + consulta.getResultList());
```

```
    return consulta.getResultList();
```

```
}
```

```
@Override
```

```
public int getCodigo() {
```

```
    var lista = findAll();
```

```
    Collections.sort(lista, (Administrador f1, Administrador f2) -> f1.getCodigo() - (f2.getCodigo()));
```

```
    return lista.get(lista.size() - 1).getCodigo();
```

```
}
```


```
}
```

**ControladorPersona:**

```
/*
```

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

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

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

\* and open the template in the editor.

\*/

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

```
import java.util.Collections;
```

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

```
import javax.persistence.NoResultException;
```

```
import javax.persistence.Query;
```

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

/\*\*

\*

\* @author User

\*/

```
public class ControladorPersona extends Controlador<Persona> {
```

```
    public Persona findByCedula(String cedula) {
```

```
        try {
```

```
            Query consulta = getEm().createNamedQuery("Persona.findByCedula");
```

```
            consulta.setParameter("cedula", cedula);
```

```
            Persona p = (Persona) consulta.getSingleResult();
```

```
            return p;
```


```
        } catch (NoResultException e) {
```

```
            e.printStackTrace();
```

```
            return null;
```

```
        }
```

```
    }
```

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

@Override

```
public List<Persona> findAll() {
    Query consulta = getEm().createNamedQuery("Persona.findAll");
    return consulta.getResultList();
}
```

@Override

```
public int getCodigo() {
    var lista = findAll();

    Collections.sort(lista, (Persona p1, Persona p2) -> p1.getCodigo() - p2.getCodigo());


    if (!lista.isEmpty()) {
        return lista.get(lista.size() - 1).getCodigo();
    } else {
        return 0;
    }
}

}
```

**ControladorCasa:**

```
/*
 * 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.Collections;
import java.util.List;
import javax.persistence.Query;
import ups.edu.ec.modelo.Casa;

/**
 *
 * @author User
 */
public class ControladorCasa extends Controlador<Casa> {


    @Override
    public List<Casa> findAll() {
        Query consulta = getEm().createNamedQuery("Casa.findAll");
        return consulta.getResultList();
    }

    @Override
    public int getCodigo() {
        var lista = findAll();

        if (!lista.isEmpty()) {
            Collections.sort(lista, (Casa f1, Casa f2) -> f1.getCodigo().compareTo(f2.getCodigo()));

            return lista.get(lista.size() - 1).getCodigo();
        } else {
            return 0;
        }
    }
}

```

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

```
}
```

### ControladorHipoteca:

```
/*
```

```
* 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.math.BigDecimal;
```

```
import java.math.RoundingMode;
```

```
import java.util.Collections;
```

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

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

```
import javax.persistence.Query;
```

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

```
/**
```

```
*
```

```
* @author User
```


```
*/
```

```
public class ControladorHipoteca extends Controlador<Hipoteca> {
```

```
    public BigDecimal calcularCargaFinanciera(BigDecimal montoPrestamo, BigDecimal interes, int mesesPlazo) {
```

```
        BigDecimal totalCarga = new BigDecimal(0);
```

```
        double carga = 0;
```

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

```
double anios = (mesesPlazo / 12);
```

```
if (mesesPlazo % 12 != 0) {
```

```
    anios += 1;
```

```
}
```

```
BigDecimal aniosPago = new BigDecimal(anios);
```

```
BigDecimal aniosPagoRedondeado = aniosPago.setScale(0, RoundingMode.CEILING);
```

```
System.out.println(anios + " " + aniosPago + " " + aniosPagoRedondeado);
```

```
// double pagoSinIntereses = montoPrestamo.doubleValue() / aniosPagoRedondeado.doubleValue();
```

```
double pagoSinIntereses = (montoPrestamo.doubleValue() / mesesPlazo) * 12;
```

```
BigDecimal pago = new BigDecimal(pagoSinIntereses);
```

```
BigDecimal pagoRedondeado = pago.setScale(2, RoundingMode.CEILING);
```

```
double montoInicial = montoPrestamo.doubleValue();
```

```
BigDecimal montoRedondear = new BigDecimal(0);
```

```
for (int i = 1; i <= aniosPagoRedondeado.intValue(); i++) {
```

```
    double interesAnual = montoInicial * (interes.doubleValue() / 100);
```

```
    carga = totalCarga.doubleValue() + interesAnual;
```


```
    totalCarga = BigDecimal.valueOf(carga).setScale(2, RoundingMode.CEILING);
```

```
    montoInicial -= pagoRedondeado.doubleValue();
```

```
    montoRedondear = BigDecimal.valueOf(montoInicial).setScale(2, RoundingMode.CEILING);
```

```
    montoInicial = montoRedondear.doubleValue();
```

```
}
```

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

```

return totalCarga;

}

public BigDecimal calcularTotal(BigDecimal prestamo, BigDecimal carga) {

    return BigDecimal.valueOf(prestamo.doubleValue() + carga.doubleValue()).setScale(2,
RoundingMode.CEILING);
}

public BigDecimal calcularMensualidad(BigDecimal total, int plazo) {

    return BigDecimal.valueOf(total.doubleValue() / plazo).setScale(2, RoundingMode.CEILING);
}

public BigDecimal calcularSueldoNecesario(BigDecimal mensualidad, int plazo) {


    return BigDecimal.valueOf(mensualidad.doubleValue() * 3.705).setScale(2, RoundingMode.CEILING);
}

public Date calcularFechaFin(Date fechaInicio, int mesesPlazo) {
    int ano = (fechaInicio.getYear());
    int mes = fechaInicio.getMonth();

    int sumarAnios = mesesPlazo / 12;
    int sumarMes = mesesPlazo % 12;

    Date fechaFin = new Date((ano + sumarAnios), (mes + sumarMes + 1), fechaInicio.getDate());

```

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

```

return fechaFin;


}

@Override
public List<Hipoteca> findAll() {
    Query consulta = getEm().createNamedQuery("Hipoteca.findAll");
    return consulta.getResultList();
}

@Override
public int getCodigo() {
    var lista = findAll();

    if (!lista.isEmpty()) {
        Collections.sort(lista, (Hipoteca h1, Hipoteca h2) -> h1.getCodigo().compareTo(h2.getCodigo()));
        return lista.get(lista.size() - 1).getCodigo();
    } else {
        return 0;
    }
}
}

```

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

### Paquete vista:

### 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.math.BigDecimal;

import java.util.Date;

import javax.swing.JOptionPane;

import ups.edu.ec.controlador.ControladorAdministrador;

import ups.edu.ec.controlador.ControladorCasa;

import ups.edu.ec.controlador.ControladorHipoteca;

import ups.edu.ec.controlador.ControladorPersona;

import ups.edu.ec.modelo.Administrador;

/\*\*

\*

\* @author User


\*/

public class VentanaPrincipal extends javax.swing.JFrame {

private String keyUnlock;

private boolean inicioSesion;

private Administrador administrador;

	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 ControladorPersona controladorPersona;

private ControladorCasa controladorCasa;

private ControladorHipoteca controladorHipoteca;


private VentanaCrearAdministrador ventanaCrearAdministrador;

private VentanaLogIn ventanaLogIn;

private VentanaCrearPersona ventanaCrearPersona;

private VentanaCrearCasa ventanaCrearCasa;

private VentanaCrearHipoteca ventanaCrearHipoteca;

private VentanaMostrarPagos ventanaMostrarPagos;

```

```
/**
```

```
 * Creates new form VentanaPrincipal
```

```
*/
```

```
public VentanaPrincipal() {
```

```
    initComponents();
```

```
    keyUnlock = "programacion";
```

```
    inicioSesion = false;
```

```
/*
```

```
try {
```

```
    Constructor<Administrador> constructor;
```

```
    constructor = Administrador.class.getDeclaredConstructor();
```


```
    constructor.setAccessible(true);
```

```
    administrador = constructor.newInstance();
```

```
    administrador.setNombre("Jose");
```

```
    administrador.setApellido("Jose");
```

```
    administrador.setCedula("010101010-1");
```

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

```

administrador.setCorreo("jose@gmail.com");
administrador.setNumeroTelefono("012-345-6789");
} catch (Exception e) {
    e.printStackTrace();
}*/

controladorAdministrador = new ControladorAdministrador();
controladorPersona = new ControladorPersona();
controladorCasa = new ControladorCasa();
controladorHipoteca = new ControladorHipoteca();

/* System.out.println(controladorHipoteca.calcularCargaFinanciera(new BigDecimal(120000), new
BigDecimal(11.23), 16));

System.out.println(controladorHipoteca.calcularFechaFin(new Date(121, 0, 29), 16));*/
/*String passwordEncriptado = controladorAdministrador.encriptarPassword(keyUnlock, "1234");


administrador.setPassword(passwordEncriptado);
controladorAdministrador.create(administrador);*/

ventanaMostrarPagos = new VentanaMostrarPagos();
ventanaCrearAdministrador = new VentanaCrearAdministrador(controladorAdministrador, keyUnlock);
ventanaLogIn = new VentanaLogIn(controladorAdministrador, keyUnlock);
ventanaCrearPersona = new VentanaCrearPersona(controladorPersona);
ventanaCrearCasa = new VentanaCrearCasa(controladorCasa);
ventanaCrearHipoteca = new VentanaCrearHipoteca(controladorHipoteca, controladorPersona,
    controladorCasa, ventanaMostrarPagos);

desktopPane.add(ventanaLogIn);
desktopPane.add(ventanaCrearAdministrador);
desktopPane.add(ventanaCrearPersona);
desktopPane.add(ventanaCrearCasa);
desktopPane.add(ventanaCrearHipoteca);

```



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

```
desktopPane.add(ventanaMostrarPagos);
```

```
}
```

```
public void cerrarVentanas() {
```

```
    ventanaLogIn.hide();
```

```
    ventanaCrearAdministrador.hide();
```

```
    ventanaCrearPersona.hide();
```

```
    ventanaCrearCasa.hide();
```

```
    ventanaCrearHipoteca.hide();
```

```
}
```

```
/**
```

```
 * 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();
```


```
    openMenuIniciarSesion = new javax.swing.JMenuItem();
```

```
    cerrarSesionMenuItem = new javax.swing.JMenuItem();
```

```
    exitMenuItem = new javax.swing.JMenuItem();
```

```
    editMenu = new javax.swing.JMenu();
```

```
    adminMenuItem = new javax.swing.JMenuItem();
```

	<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		

```

personaMenuItem = new javax.swing.JMenuItem();
casaMenuItem = new javax.swing.JMenuItem();
hipotecaMenuItem = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Prueba");


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

openMenuIniciarSesion.setMnemonic('o');
openMenuIniciarSesion.setText("Iniciar Sesion");
openMenuIniciarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        openMenuIniciarSesionActionPerformed(evt);
    }
});
fileMenu.add(openMenuIniciarSesion);

cerrarSesionMenuItem.setText("Cerrar Sesion");
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");

```

	<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		

```

exitMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        exitMenuItemActionPerformed(evt);
    }
});
fileMenu.add(exitMenuItem);


menuBar.add(fileMenu);

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

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

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

```

	<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		

```

casaMenuItem.setText("Casa");
casaMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        casaMenuItemActionPerformed(evt);
    }
});
editMenu.add(casaMenuItem);


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

menuBar.add(editMenu);

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, 606, Short.MAX_VALUE)
);
layout.setVerticalGroup(

```

	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)
    .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 438, Short.MAX_VALUE)
);

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


private void exitMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

private void openMenuIniciarSesionActionPerformed(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);
    }
}

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

    cerrarVentanas();
    if (inicioSesion) {
        Administrador.instance = null;
    }
}

```

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

```

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 adminMenuItemActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

cerrarVentanas();

//ventanaCrearAdministrador.setVisible(true);

if (Administrador.instance != null) {

cerrarVentanas();

ventanaCrearAdministrador.setVisible(true);

} else {

cerrarVentanas();

JOptionPane.showMessageDialog(this, "Nadie ha iniciado sesión", "Error",
JOptionPane.ERROR_MESSAGE);


}

}

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

// TODO add your handling code here:

```

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

```

cerrarVentanas();

//ventanaCrearPersona.setVisible(true);

if (Administrador.instance != null) {
    cerrarVentanas();
    ventanaCrearPersona.setVisible(true);

} else {
    cerrarVentanas();

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

}


private void casaMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    //ventanaCrearCasa.setVisible(true);

    if (Administrador.instance != null) {
        cerrarVentanas();
        ventanaCrearCasa.setVisible(true);

    } else {
        cerrarVentanas();

        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 hipotecaMenuItemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cerrarVentanas();
    //ventanaCrearHipoteca.setVisible(true);

    if (Administrador.instance != null) {
        cerrarVentanas();
        ventanaCrearHipoteca.setVisible(true);


    } else {
        cerrarVentanas();
        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 */
    //<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 {

```



	<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		

```

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);
    }
});
}

// Variables declaration - do not modify

```

	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.JMenuItem adminMenuItem;
private javax.swing.JMenuItem casaMenuItem;
private javax.swing.JMenuItem cerrarSesionMenuItem;
private javax.swing.JDesktopPane desktopPane;
private javax.swing.JMenu editMenu;
private javax.swing.JMenuItem exitMenuItem;
private javax.swing.JMenu fileMenu;
private javax.swing.JMenuItem hipotecaMenuItem;
private javax.swing.JMenuBar menuBar;
private javax.swing.JMenuItem openMenuIniciarSesion;
private javax.swing.JMenuItem personaMenuItem;
// 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;

/**
 *
 */

```

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

```

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

    private ControladorAdministrador controladorAdministrador;

    private String keyUnlock;

    /**
     * Creates new form VentanaLogIn
     *
     * @param controladorAdministrador
     * @param keyUnlock
     */
    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.

```

	<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		

\* 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();

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) {

}

	<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);
}

});

jLabel1.setText("Correo:");


jLabel2.setText("Contraseña:");

btnLogIn.setBackground(new java.awt.Color(51, 51, 255));
btnLogIn.setText("Log In");
btnLogIn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnLogInActionPerformed(evt);
    }
});

btnCancelar.setBackground(new java.awt.Color(255, 51, 51));
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);

```

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

```

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

    .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(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(31, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(btnLogIn, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(btnCancelar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(22, Short.MAX_VALUE))
);
```

	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:
    limpiar();
}

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);

```



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

```

    }

    } else {

        JOptionPane.showMessageDialog(this, "Existe otro usuario conectado", "Error",
JOptionPane.ERROR_MESSAGE);

    }

}

}

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


}

```

#### **VentanaCrearAdministrador:**

/\*

\* 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.vista;
```

```
import java.lang.reflect.Constructor;
```

```
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.ControladorAdministrador;
```

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

```
/**
```

```
*
```

```
* @author User
```

```
*/
```

```
public class VentanaCrearAdministrador extends javax.swing.JInternalFrame {
```

```
    private Administrador administrador;
```

```
    private ControladorAdministrador controladorAdministrador;
```


```
    private String keyUnlock;
```

```
/**
```

```
* Creates new form VentanaCrearUsuario
```

```
*/
```

```
public VentanaCrearAdministrador(ControladorAdministrador controladorAdministrador, String keyUnlock) {
```

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

```
initComponents();
```

```
this.controladorAdministrador = controladorAdministrador;
```

```
crearAdministrador();
```

```
cargarCodigo();
```

```
llenarTblAdministradores();
```

```
this.keyUnlock = keyUnlock;
```

```
}
```

```
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.getCodigo() + 1));
```


```
}
```

```
public void limpiar() {
```

```
    txtNombre.setText("");
```

```
    txtApellido.setText("");
```

```
    txtFormattedCedula.setValue("");
```

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

```

txtPassword.setText("");
txtCorreo.setText("");
cargarCodigo();
txtFormattedTelefono.setValue("");
cargarBotones();
tblUsuarios.clearSelection();
txtFormattedCedula.setEditable(true);

```

```

llenarTblAdministradores();
crearAdministrador();

```

```

}

```

```

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();

```

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

```

modelo.setRowCount(0);

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

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

    modelo.addRow(row);

}


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();

```


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

```

jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
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();
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) {

```

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

```

}

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);
}


});

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];
}

```

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

```

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) {
        tblUsuariosMouseClicked(evt);
    }
});
jScrollPane1.setViewportView(tblUsuarios);

jLabel1.setText("Nombre:");

jLabel2.setText("Apellido:");

jLabel3.setText("Cédula:");

jLabel4.setText("Código:");


jLabel6.setText("Teléfono:");

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

try {

```



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

```

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));


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);
    }
});

```

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

```

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)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addGroup(jPanel1Layout.createSequentialGroup()
                                .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addComponent(jLabel1)
                                    .addComponent(jLabel4)
                                )
                                .addGroup(jPanel1Layout.createSequentialGroup()
                                    .addGroup(jPanel1Layout.createSequentialGroup()
                                        .addGroup(jPanel1Layout.createSequentialGroup()
                                            .addGap(1, 1, 1)


```

	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)
    .addComponent(jLabel3)
    .addComponent(jLabel2)
    .addComponent(jLabel6)
    .addComponent(jLabel7)
    .addComponent(jLabel8))))
.addGap(107, 107, 107)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
    .addComponent(txtFormattedTelefono, javax.swing.GroupLayout.DEFAULT_SIZE, 203,
Short.MAX_VALUE)
    .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(


```

	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())
.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))
.addGap(20, 20, 20)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
.addComponent(jLabel3)
.addGap(25, 25, 25))
.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)
.addGap(21, 21, 21)))
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.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(txtFormattedTelefono, 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(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()
            .addContainerGap()
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .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		

```

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 504,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .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();

    String email = txtCorreo.getText();

    String password = txtPassword.getText();

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

```

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

    if (controladorAdministrador.validarEmail(email)) {
        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.setCorreo(email);

        administrador.setNumeroTelefono(telefono);

        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		

```

    } else {
        JOptionPane.showMessageDialog(this, "Email invalido", "Error", JOptionPane.ERROR_MESSAGE);
    }

}

}

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.findByCedula(cedula);

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

    txtCodigo.setText(String.valueOf(administrador.getCodigo()));
    txtNombre.setText(administrador.getNombre());
    txtApellido.setText(administrador.getApellido());

```



	<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		

```

txtFormattedCedula.setValue(administrador.getCedula());

txtFormattedCedula.setEditable(false);

txtCorreo.setText(administrador.getCorreo());

txtPassword.setText(passwordDesencriptado);

txtFormattedTelefono.setValue(administrador.getNumeroTelefono());

```

```

cargarBotonesCambio();

```

```

}

```

```

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();

```

```

    String email = txtCorreo.getText();

```

```

    String password = txtPassword.getText();

```

```

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

```

```

    if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank()

```

```

        || email.isBlank() || password.isBlank() || telefono.isBlank()) {

```

```

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

```

```


    } else {

```

```

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

```

	<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		

```

administrador.setPassword(passwordEncriptado);

//administrador.setCodigo(codigo);
administrador.setNombre(nombre);
administrador.setApellido(apellido);
administrador.setCedula(cedula);

administrador.setCorreo(email);

administrador.setNumeroTelefono(telefono);

if (controladorAdministrador.update(administrador)) {
    JOptionPane.showMessageDialog(this, "Administrador 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 cedula = (String) txtFormattedCedula.getValue();

    administrador.setCedula(cedula);

```

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

```


if (controladorAdministrador.delete(administrador)) {
    JOptionPane.showMessageDialog(this, "Administrador eliminado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}
}

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 javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;

```

	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 jLabel4;
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;
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
}

```


#### **VentanaCrearPersona:**

```

/*
 * 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.math.BigDecimal;
import java.util.Calendar;
import java.util.Date;
import java.util.List;

```

	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.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorPersona;
import ups.edu.ec.modelo.Persona;

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

    private ControladorPersona controladorPersona;


    private Persona persona;

    /**
     * Creates new form VentanaCrearCliente
     *
     * @param controladorPersona
     */
    public VentanaCrearPersona(ControladorPersona controladorPersona) {
        initComponents();

        this.controladorPersona = controladorPersona;

        persona = new Persona();
        limpiar();

```

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

```

}


public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorPersona.getCodigo() + 1));
}

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

    cargarCodigo();
    txtFormattedTelefono.setValue("");
    cargarBotones();
    tblClientes.clearSelection();
    txtFormattedCedula.setEditable(true);
    txtSueldo.setText("");
    txtDireccion.setText("");
    cbxTipo.setSelectedIndex(0);
    llenarTblClientes();
}

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

```

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

```

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 (Persona per : (List<Persona>) controladorPersona.findAll()) {

        Object[] row = {per.getCedula(), per.getNombre(), per.getApellido(),
            per.getSueldo(), per.getTipo()};

        modelo.addRow(row);


    }

    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.

```


	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() {

    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();
    jLabel7 = new javax.swing.JLabel();
    txtSueldo = new javax.swing.JTextField();
    jLabel8 = new javax.swing.JLabel();
    txtDireccion = new javax.swing.JTextField();
    jLabel9 = new javax.swing.JLabel();
    cbxTipo = new javax.swing.JComboBox<>();
    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("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:");

```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: 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);
```

```
    }
```

```
});
```

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

```
jLabel8.setText("Direccion:");
```

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

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

```
cbxTipo.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "--Seleccione--", "Propietario", "Garante" }));
```

```
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
```

```
jPanel1.setLayout(jPanel1Layout);
```

```
jPanel1Layout.setHorizontalGroup(
```

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

```
        .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(jLabel7)
```

```
            .addComponent(jLabel8)
```

```
            .addComponent(jLabel9)))
```

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

```
    .addGap(100, 100, 100)
```


	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)
        .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 203,
Short.MAX_VALUE)
        .addComponent(txtFormattedTelefono)
        .addComponent(txtFormattedCedula)
        .addComponent(txtCodigo)
        .addComponent(txtNombre)
        .addComponent(txtApellido)
        .addComponent(txtSueldo)
        .addComponent(txtDireccion)
        .addComponent(cbxTipo, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(112, 112, 112)
        .addComponent(btnActualizar)
        .addGap(41, 41, 41)
        .addComponent(btnEliminar)
        .addGap(33, 33, 33)
        .addComponent(btnCancelar)))
        .addContainerGap(55, Short.MAX_VALUE))
);

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

```

	<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		

```

        .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()

        .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(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(jLabel7)

        .addComponent(txtSueldo, 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(jLabel8)

            .addComponent(txtDireccion, 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(cbxTipo, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(37, 37, 37)

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

            .addComponent(btnCrear)

            .addComponent(btnActualizar)

            .addComponent(btnEliminar)

            .addComponent(btnCancelar))

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

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


    },
    new String [] {

        "Cedula", "Nombre", "Apellido", "Sueldo", "Tipo"
    }
));

```





	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.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

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

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 504,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .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();
    Date fechaNacimiento = jDateChooser1.getDate();
    BigDecimal sueldo = BigDecimal.valueOf(Double.valueOf(txtSueldo.getText()));

```

	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();

String direccion = txtDireccion.getText();

String tipo = (String) cbxTipo.getSelectedItem();

System.out.println(sueldo);

if (nombre.isBlank() || apellido.isBlank() || cedula.isBlank() || fechaNacimiento == null
    || telefono.isBlank() || sueldo == null || direccion == null || tipo.equals("--Seleccione--")) {

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

} else {

    persona = new Persona();

    //cliente.setCodigo(codigo);

    persona.setNombre(nombre);

    persona.setApellido(apellido);

    persona.setCedula(cedula);

    persona.setFechaNacimiento(fechaNacimiento);

    persona.setNumeroTelefono(telefono);

    persona.setSueldo(sueldo);

    persona.setDireccion(direccion);

    persona.setTipo(tipo);

    if (controladorPersona.create(persona)) {

        JOptionPane.showMessageDialog(this, "Cliente creado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

        limpiar();


    } else {

        JOptionPane.showMessageDialog(this, "Cliente 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 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();
    Date fechaNacimiento = jDateChooser1.getDate();
    BigDecimal sueldo = BigDecimal.valueOf(Double.valueOf(txtSueldo.getText()));
    String telefono = (String) txtFormattedTelefono.getValue();
    String tipo = (String) cbxTipo.getSelectedItem();
    String direccion = txtDireccion.getText();

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

        //cliente = controladorCliente.findCliente(cedula);
        //cliente.setCodigo(codigo);
        persona.setNombre(nombre);
        persona.setApellido(apellido);
        persona.setCedula(cedula);
        persona.setFechaNacimiento(fechaNacimiento);
        persona.setNumeroTelefono(telefono);
        persona.setSueldo(sueldo);
        persona.setDireccion(direccion);
    }
}

```

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

```

persona.setTipo(tipo);

/*cliente.setFacturaCollection(cliente.getFacturaCollection());
cliente.setVehiculoCollection(cliente.getVehiculoCollection());*/

if (controladorPersona.update(persona)) {
    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) {
    // TODO add your handling code here:
    String cedula = (String) txtFormattedCedula.getValue();

    persona = controladorPersona.findByCedula(cedula);

    if (controladorPersona.delete(persona)) {

        JOptionPane.showMessageDialog(this, "Cliente eliminado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
        limpiar();
    } else {
        JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_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		

```

}

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);
    persona = new Persona();

    persona = controladorPersona.findByCedula(cedula);

    txtCodigo.setText(String.valueOf(persona.getCodigo()));
    txtNombre.setText(persona.getNombre());
    txtApellido.setText(persona.getApellido());
    txtFormattedCedula.setValue(persona.getCedula());
    txtFormattedCedula.setEditable(false);
    txtFormattedTelefono.setValue(persona.getNumeroTelefono());
    jDateChooser1.setDate(persona.getFechaNacimiento());
    txtSueldo.setText(persona.getSueldo().toString());
    txtDireccion.setText(persona.getDireccion());
    cbxTipo.setSelectedItem(persona.getTipo());

```

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

```


    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 javax.swing.JComboBox<String> cbxTipo;
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;

```

	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 jLabel8;
private javax.swing.JLabel jLabel9;
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.JTextField txtDireccion;
private javax.swing.JFormattedTextField txtFormattedCedula;
private javax.swing.JFormattedTextField txtFormattedTelefono;
private javax.swing.JTextField txtNombre;
private javax.swing.JTextField txtSueldo;

// End of variables declaration
}

```

#### **VentanaCrearCasa:**


```

/*
 * 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.math.BigDecimal;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorCasa;
import ups.edu.ec.modelo.Casa;

```

	<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		

```

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

    private ControladorCasa controladorCasa;


    /**
     * Creates new form VentanaCrearCasa
     */
    public VentanaCrearCasa(ControladorCasa controladorCasa) {
        initComponents();
        this.controladorCasa = controladorCasa;
        limpiar();
    }

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

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

```



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

```

}

public void limpiar() {
    cargarCodigo();
    txtAvaluo.setText("");
    txtDireccion.setText("");
    txtMetrosConstruccion.setText("");
    txtNumeroHabitaciones.setText("");
    txtNumeroPisos.setText("");
    cbxEstado.setSelectedIndex(0);
    botonesInicio();
    llenarTabla();
    tblCasas.clearSelection();
}


public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorCasa.getCodigo() + 1));
}

public void llenarTabla() {
    var listacasas = controladorCasa.findAll();

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

    for (Casa casa : listacasas) {
        Object[] row = {casa.getCodigo(), casa.getDireccion(), casa.getAvaluo().doubleValue(),
            casa.getNumeroHabitaciones(), casa.getMetrosConstruccion().doubleValue(), casa.getEstado()
        }
    }
}

```

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

```

};

modelo.addRow(row);

}


tblCasas.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();
    txtCodigo = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    txtDireccion = new javax.swing.JTextField();
    txtAvaluo = new javax.swing.JTextField();
    txtMetrosConstruccion = new javax.swing.JTextField();
    txtNumeroHabitaciones = new javax.swing.JTextField();
    txtNumeroPisos = 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		


```

btnCrear = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminar = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
cbxEstado = new javax.swing.JComboBox<>();
jScrollPane1 = new javax.swing.JScrollPane();
tblCasas = new javax.swing.JTable();

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("Código:");

```

	<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		

```

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

jLabel2.setText("Dirección:");

jLabel3.setText("Avaluo:");

jLabel4.setText("M^2 de construcción:");


jLabel5.setText("Número habitaciones:");

jLabel6.setText("Número pisos:");

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.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnActualizarActionPerformed(evt);
    }
});

```

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

```
btnEliminar.setBackground(new java.awt.Color(255, 51, 51));
```

```
btnEliminar.setText("Eliminar");
```

```
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);
```

```
    }
```

```
});
```

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

```
cbxEstado.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "--Seleccione--", "Enventa",  
"Vendida", "Extraida" }));
```

```
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(41, 41, 41)
```

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

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

```
                    .addComponent(jLabel6)
```

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

```

.addGap(0, 0, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup())

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

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(jLabel1)

.addGap(85, 85, 85)

.addComponent(txtCodigo))

.addGroup(jPanel1Layout.createSequentialGroup())

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

.addComponent(jLabel2)

.addComponent(jLabel5)

.addComponent(jLabel4)

.addComponent(jLabel3)

.addComponent(jLabel7))

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

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

.addComponent(txtDireccion)

.addComponent(txtAvaluo)

.addComponent(txtMetrosConstruccion)

.addComponent(txtNumeroHabitaciones)

.addComponent(txtNumeroPisos)

.addComponent(cbxEstado, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))


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

.addGap(15, 15, 15)

.addComponent(btnCrear)

.addGap(37, 37, 37)

```

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

```

        .addComponent(btnActualizar)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 50,
Short.MAX_VALUE)

        .addComponent(btnEliminar)

        .addGap(39, 39, 39)

        .addComponent(btnCancelar)))

        .addGap(47, 47, 47)))

);

jPanel1Layout.setVerticalGroup(

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

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(38, 38, 38)

.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(txtDireccion, 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(txtAvaluo, 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(txtMetrosConstruccion, 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(txtNumeroHabitaciones, 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(txtNumeroPisos, 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(jLabel7)

    .addComponent(cbxEstado, 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(btnActualizar)

    .addComponent(btnEliminar)

    .addComponent(btnCancelar))

.addContainerGap(35, Short.MAX_VALUE))

);

tblCasas.setModel(new javax.swing.table.DefaultTableModel(


    new Object [][] {

    },

    new String [] {

        "Código", "Direccion", "Avaluo", "Habitaciones", "m^2 construcción", "Estado"
```



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

```

    }
    ){
        Class[] types = new Class [] {
            java.lang.Integer.class, java.lang.String.class, java.lang.Double.class, java.lang.Integer.class,
            java.lang.Double.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];
        }
    });
    tblCasas.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            tblCasasMouseClicked(evt);
        }
    });
    jScrollPane1.setViewportView(tblCasas);

    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.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGap(18, 18, 18)

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 555,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addContainerGap(20, 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(14, 14, 14)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(layout.createSequentialGroup())

                .addGap(24, 24, 24)

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


        .addContainerGap(19, Short.MAX_VALUE))

);

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

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

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

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

```

String direccion = txtDireccion.getText();

BigDecimal avaluo = BigDecimal.valueOf(Double.valueOf(txtAvaluo.getText()));

BigDecimal metrosConstruccion = BigDecimal.valueOf(Double.valueOf(txtMetrosConstruccion.getText()));

int numeroHabitaciones = Integer.valueOf(txtNumeroHabitaciones.getText());

int numeroPisos = Integer.valueOf(txtNumeroPisos.getText());

String estado = (String) cbxEstado.getSelectedItem();

if (direccion.isBlank() || avaluo == null || metrosConstruccion == null || numeroHabitaciones == 0
    || numeroPisos == 0 || estado.equals("--Seleccione--")) {

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

} else {

    Casa casa = new Casa(direccion, avaluo, metrosConstruccion, numeroHabitaciones, numeroPisos,
estado);

    if (controladorCasa.create(casa)) {

        JOptionPane.showMessageDialog(this, "Casa creada con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

        limpiar();

    } else {

        JOptionPane.showMessageDialog(this, "Casa ya existente", "Error",
JOptionPane.ERROR_MESSAGE);

    }


}

}

private void tblCasasMouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:

```

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

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

```
int codigo = (int) (tblCasas.getValueAt(fila, 0));
```

```
Casa casa = controladorCasa.read(codigo);
```

```
txtCodigo.setText(codigo + "");
```

```
txtDireccion.setText(casa.getDireccion());
```

```
txtAvaluo.setText(casa.getAvaluo().toString());
```

```
txtMetrosConstruccion.setText(casa.getMetrosConstruccion().toString());
```

```
txtNumeroHabitaciones.setText(casa.getNumeroHabitaciones().toString());
```

```
txtNumeroPisos.setText(casa.getNumeroPisos().toString());
```

```
cbxEstado.setSelectedItem(casa.getEstado());
```

```
botonesActualizar();
```

```
tblCasas.clearSelection();
```

```
}
```

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

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

```
String direccion = txtDireccion.getText();
```


```
BigDecimal avaluo = BigDecimal.valueOf(Double.valueOf(txtAvaluo.getText()));
```

```
BigDecimal metrosConstruccion = BigDecimal.valueOf(Double.valueOf(txtMetrosConstruccion.getText()));
```

```
int numeroHabitaciones = Integer.valueOf(txtNumeroHabitaciones.getText());
```

```
int numeroPisos = Integer.valueOf(txtNumeroPisos.getText());
```

```
String estado = (String) cbxEstado.getSelectedItem();
```

	<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 (direccion.isBlank() || avaluo == null || metrosConstruccion == null || numeroHabitaciones == 0
    || numeroPisos == 0 || estado.equals("--Seleccione--")) {
    JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error",
JOptionPane.ERROR_MESSAGE);
} else {

    Casa casa = controladorCasa.read(Integer.valueOf(txtCodigo.getText()));

    casa.setDireccion(direccion);
    casa.setAvaluo(avaluo);
    casa.setMetrosConstruccion(metrosConstruccion);
    casa.setNumeroHabitaciones(numeroHabitaciones);
    casa.setNumeroPisos(numeroPisos);
    casa.setEstado(estado);


    if (controladorCasa.update(casa)) {
        JOptionPane.showMessageDialog(this, "Casa actualizada 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:

```

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

```
Casa casa = controladorCasa.read(Integer.valueOf(txtCodigo.getText()));
```

```
if (controladorCasa.delete(casa)) {
```

```
    JOptionPane.showMessageDialog(this, "Casa eliminada con exito", "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 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();
```

```
}
```

	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 btnActualizar;

private javax.swing.JButton btnCancelar;

private javax.swing.JButton btnCrear;

private javax.swing.JButton btnEliminar;

private javax.swing.JComboBox<String> cbxEstado;

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.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable tblCasas;

private javax.swing.JTextField txtAvaluo;

private javax.swing.JTextField txtCodigo;

private javax.swing.JTextField txtDireccion;


private javax.swing.JTextField txtMetrosConstruccion;

private javax.swing.JTextField txtNumeroHabitaciones;

private javax.swing.JTextField txtNumeroPisos;

// 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		

### VentanaCrearHipoteca:

```
/*
 * 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.math.BigDecimal;
import java.util.Date;
import javax.persistence.NoResultException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ControladorCasa;
import ups.edu.ec.controlador.ControladorHipoteca;
import ups.edu.ec.controlador.ControladorPersona;
import ups.edu.ec.modelo.Casa;
import ups.edu.ec.modelo.EnumEstadoCasa;
import ups.edu.ec.modelo.EnumPersonaTipo;
import ups.edu.ec.modelo.Hipoteca;
import ups.edu.ec.modelo.Persona;
```


```
/**
```

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

```
public class VentanaCrearHipoteca extends javax.swing.JInternalFrame {

    private ControladorHipoteca controladorHipoteca;
```



	<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 ControladorPersona controladorPersona;

private ControladorCasa controladorCasa;


private VentanaMostrarPagos ventanaMostrarPagos;


private boolean necesidadGarante;
private boolean botonesActualizar;



private Hipoteca hipotecaActualizar;


/**
 * Creates new form VentanaCrearHipoteca
 *
 * @param controladorHipoteca
 * @param controladorPersona
 * @param controladorCasa
 * @param ventanaMostrarPagos
 */
public VentanaCrearHipoteca(ControladorHipoteca controladorHipoteca, ControladorPersona
controladorPersona,
    ControladorCasa controladorCasa, VentanaMostrarPagos ventanaMostrarPagos) {
    initComponents();

    this.controladorHipoteca = controladorHipoteca;
    this.controladorPersona = controladorPersona;
    this.controladorCasa = controladorCasa;

    this.ventanaMostrarPagos = ventanaMostrarPagos;

```

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

necesidadGarante = false;

botonesActualizar = false;

limpiar();

}

public void bloquearTxts() {

txtCodigoCasa.setEnabled(false);

jDateFechaInicio.setEnabled(false);

}

public void botonesInicio() {

btnCalcular.setEnabled(true);

btnMensualidadTabla.setEnabled(false);

btnCrear.setEnabled(false);

btnActualizar.setEnabled(false);

btnEliminar.setEnabled(false);

btnCancelar.setEnabled(true);

}

public void botonesCrear() {


btnCalcular.setEnabled(true);

btnMensualidadTabla.setEnabled(true);

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		

```

btnCancelar.setEnabled(true);


}

public void botonesActualizar() {
    btnCalcular.setEnabled(true);
    btnMensualidadTabla.setEnabled(true);
    btnCrear.setEnabled(false);
    btnActualizar.setEnabled(true);
    btnEliminar.setEnabled(true);
    btnCancelar.setEnabled(true);
}

public void cargarCodigo() {
    txtCodigo.setText(String.valueOf(controladorHipoteca.getCodigo() + 1));
}

public void limpiar() {
    botonesInicio();
    cargarCodigo();
    txtCargaFinanciera.setText("");
    txtCedulaGarante.setText("");
    txtCedulaGarante.setEditable(false);
    txtCedulaHipotecario.setText("");
    txtCodigoCasa.setText("");
    txtInteres.setText("");
    txtPlazo.setText("");
    txtPrestamo.setText("");

```

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

```

txtTarifaMensual.setText("");
txtTotal.setText("");
jDateFechaInicio.setDate(new Date());
jDateFechaFin.setDate(new Date());
txtCodigoCasa.setEnabled(true);
jDateFechaInicio.setEnabled(true);
txtSueldoNecesario.setText("");

```

```

necesidadGarante = false;
botonesActualizar = false;
llenarTblHipotecas();

```

```

}

```

```

public void llenarTblHipotecas() {

```

```

    var listaHipotecas = controladorHipoteca.findAll();

```

```

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

```

```

    for (Hipoteca hipo : listaHipotecas) {

```

```

        String nombrePropietario =
        hipo.getCedulaPropietarioFk().getNombre().concat(hipo.getCedulaPropietarioFk().getApellido());

```

```

        String nombreGarante = "";

```

```

        if (hipo.getCedulaGaranteFk() != null) {

```

```


            nombreGarante =
            hipo.getCedulaGaranteFk().getNombre().concat(hipo.getCedulaGaranteFk().getApellido());

```

```

        } else {

```

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

```

nombreGarante = "Null";
}

Object[] row = {hipo.getCodigo(), hipo.getInteres(), hipo.getMontoPrestamo(),
    hipo.getPlazoMeses(), hipo.getFechaInicio().toString(), hipo.getFechaFin(),
    hipo.getCargaFinanciera(), hipo.getPagoTotal(), hipo.getTarifaMensual(),
    hipo.getCodigoCasaFk(), nombrePropietario, nombreGarante

};


modelo.addRow(row);
}

tblHipotecas.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();
    txtCodigo = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();


```

	<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		

```

txtInteres = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
txtPrestamo = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
txtPlazo = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
txtCargaFinanciera = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
txtTotal = new javax.swing.JTextField();
txtTarifaMensual = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jDateFechaInicio = new com.toedter.calendar.JDateChooser();
jDateFechaFin = new com.toedter.calendar.JDateChooser();
jLabel9 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
txtCodigoCasa = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
txtCedulaHipotecario = new javax.swing.JFormattedTextField();
jLabel12 = new javax.swing.JLabel();
txtCedulaGarante = new javax.swing.JFormattedTextField();
btnCalcular = new javax.swing.JButton();
jLabel13 = new javax.swing.JLabel();
txtSueldoNecesario = new javax.swing.JTextField();
btnMensualidadTabla = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
tblHipotecas = new javax.swing.JTable();
btnCrear = new javax.swing.JButton();
btnActualizar = 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		

```

btnEliminar = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();

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);
    }
});


jLabel1.setText("Código:");

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

jLabel2.setText("Interes:");

jLabel3.setText("Prestamo:");

```

	<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("Plazo:");

jLabel5.setText("Carga financiera:");

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

jLabel6.setText("Total:");

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

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

jLabel7.setText("Tarifa mensual:");

jLabel8.setText("Fecha inicio:");

jDateFechaFin.setEnabled(false);

jLabel9.setText("Fecha fin");


jLabel10.setText("Código casa:");

jLabel11.setText("Cédula hipotecario:");

try {

```



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

```

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

} catch (java.text.ParseException ex) {
    ex.printStackTrace();
}

jLabel12.setText("Cédula garante:");

txtCedulaGarante.setEditable(false);

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

btnCalcular.setBackground(new java.awt.Color(102, 255, 153));
btnCalcular.setText("Calcular");
btnCalcular.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCalcularActionPerformed(evt);
    }
});


jLabel13.setText("Sueldo necesario:");

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

btnMensualidadTabla.setBackground(new java.awt.Color(255, 204, 102));

```



	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())

.addGap(102, 102, 102)

.addComponent(btnCalcular)))

.addGap(42, 42, 42)

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

.addGroup(jPanel1Layout.createSequentialGroup())

.addComponent(btnMensualidadTabla)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup())

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

.addComponent(txtCedulaGarante, javax.swing.GroupLayout.DEFAULT_SIZE, 191,
Short.MAX_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup())

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

.addComponent(txtCodigo)

.addComponent(txtInteres)

.addComponent(txtPrestamo)

.addComponent(txtPlazo)

.addComponent(txtCargaFinanciera)

.addComponent(txtTotal)

.addComponent(txtTarifaMensual)

.addComponent(jDateFechaInicio, javax.swing.GroupLayout.DEFAULT_SIZE, 191,
Short.MAX_VALUE)

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


.addComponent(txtCodigoCasa)

.addComponent(txtCedulaHipotecario))

.addGap(0, 0, Short.MAX_VALUE))

.addComponent(txtSueldoNecesario))

```


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

```

        .addGap(12, 12, 12))))
    );
    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(txtInteres, 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(txtPrestamo, 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(txtPlazo, 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(jLabel8)
                .addComponent(jDateFechaInicio, 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(jDateFechaFin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel9))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(txtCargaFinanciera, 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(txtTotal, 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(txtTarifaMensual, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel7))
.addGap(18, 18, 18)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel13)
    .addComponent(txtSueldoNecesario, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addGap(19, 19, 19)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel10)
    .addComponent(txtCodigoCasa, 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)
```

	<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		

```

        .addComponent(jLabel11)

        .addComponent(txtCedulaHipotecario, 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(jLabel12)

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

        .addGap(22, 22, 22)

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

        .addComponent(btnCalcular)

        .addComponent(btnMensualidadTabla))


        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

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

    },
    new String [] {

        "Código", "Interes", "Prestamo", "Plazo", "Fecha inicio", "Fecha fin", "Carga financiera", "Total", "Tarifa
mensual", "Casa", "Hipotecario", "Garante"
    }
) {
    Class[] types = new Class [] {
        java.lang.Integer.class, java.lang.Double.class, java.lang.Double.class, java.lang.Double.class,
        java.lang.String.class, java.lang.String.class, java.lang.Double.class, java.lang.Double.class,
        java.lang.Double.class, java.lang.Integer.class, java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false, false, false, false, true, true
    };

```

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

```

};

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

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

});


tblHipotecas.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        tblHipotecasMouseClicked(evt);
    }
});

jScrollPane1.setViewportView(tblHipotecas);

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

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

```

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

```
});
```

```
btnEliminar.setText("Eliminar");
```

```
btnEliminar.addActionListener(new java.awt.event.ActionListener() {
```

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

```
        btnEliminarActionPerformed(evt);
```

```
    }
```

```
});
```

```
btnCancelar.setText("Cancelar");
```

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

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

```
layout.setHorizontalGroup(
```

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

```
        .addGroup(layout.createParallelGroup()
```

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

```
                .addGroup(layout.createParallelGroup()
```

```
                    .addContainerGap()
```

```
                    .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, 897,
```

```
                    javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
                .addGroup(layout.createParallelGroup()
```

```
                    .addGap(28, 28, 28)
```

```
                    .addComponent(btnCrear)
```

```
                    .addGap(39, 39, 39)
```

```
                    .addComponent(btnActualizar)
```

```
                    .addGap(40, 40, 40)
```



```
.addComponent(btnEliminar)

.addGap(45, 45, 45)

.addComponent(btnCancelar)))

.addContainerGap(15, Short.MAX_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

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

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

    .addGroup(layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(jScrollPane1)))

.addGap(18, 18, 18)

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

    .addComponent(btnCrear)

    .addComponent(btnActualizar)

    .addComponent(btnEliminar)

    .addComponent(btnCancelar))

.addGap(0, 20, Short.MAX_VALUE))


);

pack();

} // </editor-fold>

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {

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

    BigDecimal interes = BigDecimal.valueOf(Double.valueOf(txtInteres.getText()));
    BigDecimal prestamo = BigDecimal.valueOf(Double.valueOf(txtPrestamo.getText()));
    int plazo = Integer.valueOf(txtPlazo.getText());
    Date fechalnicio = jDateFechalnicio.getDate();

    if (interes == null || prestamo == null || plazo == 0 || fechalnicio == null) {
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error",
        JOptionPane.ERROR_MESSAGE);
    } else {
        BigDecimal carga = controladorHipoteca.calcularCargaFinaciera(prestamo, interes, plazo);
        txtCargaFinaciera.setText(carga.toString());


        Date fechaFin = controladorHipoteca.calcularFechaFin(fechalnicio, plazo);
        jDateFechaFin.setDate(fechaFin);

        BigDecimal total = controladorHipoteca.calcularTotal(prestamo, carga);
        txtTotal.setText(total.toString());

        BigDecimal mensualidad = controladorHipoteca.calcularMensualidad(total, plazo);
        txtTarifaMensual.setText(mensualidad.toString());

        BigDecimal sueldoNecesario = controladorHipoteca.calcularSueldoNecesario(mensualidad, plazo);
        txtSueldoNecesario.setText(sueldoNecesario.toString());
    }
}

```

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

```

var casa = controladorCasa.read(Integer.valueOf(txtCodigoCasa.getText()));

if (!txtCedulaHipotecario.getText().equals("0    - ")) {


    var persona = controladorPersona.findByCedula(txtCedulaHipotecario.getText());

    if (persona != null &&
persona.getTipo().equalsIgnoreCase(EnumPersonaTipo.PROPIETARIO.toString())) {
        if (sueldoNecesario.doubleValue() > persona.getSueldo().doubleValue() || total.doubleValue() >
casa.getAvaluo().doubleValue()) {
            txtCedulaGarante.setEditable(true);
            necesidadGarante = true;
        } else {
            txtCedulaGarante.setEditable(false);
            necesidadGarante = false;
        }
    }
}

}

if (botonesActualizar) {
    botonesActualizar();
} else {
    botonesCrear();
}
}
}

```

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


    ventanaMostrarPagos.setPlazos(Integer.valueOf(txtPlazo.getText()));
    ventanaMostrarPagos.setFechaInicio(jDateFechaInicio.getDate());
    ventanaMostrarPagos.setTotal(BigDecimal.valueOf(Double.valueOf(txtTotal.getText())));
    ventanaMostrarPagos.setMensualidad(BigDecimal.valueOf(Double.valueOf(txtTarifaMensual.getText())));
    ventanaMostrarPagos.setVisible(true);

}

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

    BigDecimal interes = BigDecimal.valueOf(Double.valueOf(txtInteres.getText()));
    if (interes == null || interes.doubleValue() < 8.99 || interes.doubleValue() > 16.99) {
        JOptionPane.showMessageDialog(this, "Error en los intereses", "Error",
JOptionPane.ERROR_MESSAGE);
    } else {
        BigDecimal prestamo = BigDecimal.valueOf(Double.valueOf(txtPrestamo.getText()));
        int plazo = Integer.valueOf(txtPlazo.getText());
        Date fechaInicio = jDateFechaInicio.getDate();
        Date fechaFin = jDateFechaFin.getDate();
        BigDecimal carga = BigDecimal.valueOf(Double.valueOf(txtCargaFinanciera.getText()));
        BigDecimal total = BigDecimal.valueOf(Double.valueOf(txtTotal.getText()));
        BigDecimal mensualidad = BigDecimal.valueOf(Double.valueOf(txtTarifaMensual.getText()));
        BigDecimal sueldoNecesario = BigDecimal.valueOf(Double.valueOf(txtSueldoNecesario.getText()));
        int codigoCasa = Integer.valueOf(txtCodigoCasa.getText());
        String cedulaPropietario = txtCedulaHipotecario.getText();
    }
}

```

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

```

Casa casa = controladorCasa.read(codigoCasa);

Persona persona = controladorPersona.findByCedula(cedulaPropietario);

if (casa == null || !(casa.getEstado().equalsIgnoreCase(EnumEstadoCasa.ENVENTA.toString()))
    || persona == null || persona.getTipo().equalsIgnoreCase(EnumPersonaTipo.GARANTE.toString())) {
    JOptionPane.showMessageDialog(this, "Error casa o hipotecario", "Error",
JOptionPane.ERROR_MESSAGE);

} else {
    if (necesidadGarante) {
        String cedulaGarante = txtCedulaGarante.getText();

        Persona p = controladorPersona.findByCedula(cedulaGarante);

        if (p == null || p.getTipo().equalsIgnoreCase(EnumPersonaTipo.PROPIETARIO.toString())) {
            JOptionPane.showMessageDialog(this, "Persona no existente o es propietario", "Error",
JOptionPane.ERROR_MESSAGE);
        } else {
            Hipoteca hipoteca = new Hipoteca(sueldoNecesario, fechaInicio, fechaFin, plazo,
                prestamo, interes, carga, total, mensualidad, necesidadGarante, casa, p, persona);


            if (controladorHipoteca.create(hipoteca)) {
                JOptionPane.showMessageDialog(this, "Hipoteca creada con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

                casa.setEstado(EnumEstadoCasa.VENDIDA.toString());

                controladorCasa.update(casa);

                limpiar();
            } else {
                JOptionPane.showMessageDialog(this, "Error", "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		

```

Hipoteca hipoteca = new Hipoteca(sueldoNecesario, fechaInicio, fechaFin, plazo,
    prestamo, interes, carga, total, mensualidad, necesidadGarante, casa, null, persona);

if (controladorHipoteca.create(hipoteca)) {
    JOptionPane.showMessageDialog(this, "Hipoteca creada con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    casa.setEstado(EnumEstadoCasa.VENDIDA.toString());
    controladorCasa.update(casa);
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Error", "Error", JOptionPane.ERROR_MESSAGE);
}
}
}


}

}

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

    BigDecimal interes = BigDecimal.valueOf(Double.valueOf(txtInteres.getText()));
    if (interes == null || interes.doubleValue() < 8.99 || interes.doubleValue() > 16.99) {
        JOptionPane.showMessageDialog(this, "Error en los intereses", "Error",
JOptionPane.ERROR_MESSAGE);
    } else {
        BigDecimal prestamo = BigDecimal.valueOf(Double.valueOf(txtPrestamo.getText()));
        int plazo = Integer.valueOf(txtPlazo.getText());
        Date fechaInicio = jDateFechaInicio.getDate();

```

	<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		

```

Date fechaFin = jDateFechaFin.getDate();

BigDecimal carga = BigDecimal.valueOf(Double.valueOf(txtCargaFinanciera.getText()));

BigDecimal total = BigDecimal.valueOf(Double.valueOf(txtTotal.getText()));

BigDecimal mensualidad = BigDecimal.valueOf(Double.valueOf(txtTarifaMensual.getText()));

BigDecimal sueldoNecesario = BigDecimal.valueOf(Double.valueOf(txtSueldoNecesario.getText()));

int codigoCasa = Integer.valueOf(txtCodigoCasa.getText());

String cedulaPropietario = txtCedulaHipotecario.getText();


var casa = hipotecaActualizar.getCodigoCasaFk();
var persona = hipotecaActualizar.getCedulaPropietarioFk();

if (necesidadGarante) {
    String cedulaGarante = txtCedulaGarante.getText();

    Persona p = controladorPersona.findByCedula(cedulaGarante);

    if (p == null || p.getTipo().equalsIgnoreCase(EnumPersonaTipo.PROPIETARIO.toString())) {
        JOptionPane.showMessageDialog(this, "Persona no existente o es propietario", "Error",
JOptionPane.ERROR_MESSAGE);
    } else {

        /*Hipoteca hipoteca = new Hipoteca(sueldoNecesario, fechaInicio, fechaFin, plazo,
            prestamo, interes, carga, total, mensualidad, necesidadGarante, casa, p, persona);*/

        hipotecaActualizar.setInteres(interес);

        hipotecaActualizar.setMontoPrestamo(prestamo);

        hipotecaActualizar.setPlazoMeses(plazo);

        hipotecaActualizar.setFechaInicio(fechaInicio);

        hipotecaActualizar.setFechaFin(fechaFin);


        hipotecaActualizar.setCargaFinanciera(carga);

        hipotecaActualizar.setPagoTotal(total);

        hipotecaActualizar.setTarifaMensual(mensualidad);

        hipotecaActualizar.setSueldoNecesario(sueldoNecesario);

```

	<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		

```

hipotecaActualizar.setCodigoCasaFk(casa);

hipotecaActualizar.setCedulaPropietarioFk(persona);

hipotecaActualizar.setGarante(necesidadGarante);

hipotecaActualizar.setCedulaGaranteFk(p);


if (controladorHipoteca.update(hipotecaActualizar)) {

    JOptionPane.showMessageDialog(this, "Hipoteca actualizada con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);

    casa.setEstado(EnumEstadoCasa.VENDIDA.toString());

    controladorCasa.update(casa);

    limpiar();

} else {

    JOptionPane.showMessageDialog(this, "Error", "Error", JOptionPane.ERROR_MESSAGE);

}

}

} else {

/*Hipoteca hipoteca = new Hipoteca(sueldoNecesario, fechaInicio, fechaFin, plazo,
prestamo, interes, carga, total, mensualidad, necesidadGarante, casa, null, persona);*/


hipotecaActualizar.setInteres(interres);

hipotecaActualizar.setMontoPrestamo(prestamo);

hipotecaActualizar.setPlazoMeses(plazo);

hipotecaActualizar.setFechaInicio(fechaInicio);

hipotecaActualizar.setFechaFin(fechaFin);

hipotecaActualizar.setCargaFinanciera(carga);

hipotecaActualizar.setPagoTotal(total);

hipotecaActualizar.setTarifaMensual(mensualidad);

hipotecaActualizar.setSueldoNecesario(sueldoNecesario);

hipotecaActualizar.setCodigoCasaFk(casa);

```



```
hipotecaActualizar.setCedulaPropietarioFk(persona);
hipotecaActualizar.setGarante(necesidadGarante);
hipotecaActualizar.setCedulaGaranteFk(null);

if (controladorHipoteca.update(hipotecaActualizar)) {
    JOptionPane.showMessageDialog(this, "Hipoteca actualizada con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    casa.setEstado(EnumEstadoCasa.VENDIDA.toString());
    controladorCasa.update(casa);
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Error", "Error", JOptionPane.ERROR_MESSAGE);
}
}

}

}


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

    int fila = tblHipotecas.getSelectedRow();

    int codigo = (int) (tblHipotecas.getValueAt(fila, 0));

    hipotecaActualizar = controladorHipoteca.read(codigo);

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

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

```

txtInteres.setText(hipotecaActualizar.getInteres().toString());
txtPrestamo.setText(hipotecaActualizar.getMontoPrestamo().toString());
txtPlazo.setText(hipotecaActualizar.getPlazoMeses().toString());
jDateFechaInicio.setDate(hipotecaActualizar.getFechaInicio());
jDateFechaFin.setDate(hipotecaActualizar.getFechaFin());
txtCargaFinanciera.setText(hipotecaActualizar.getCargaFinanciera().toString());
txtTotal.setText(hipotecaActualizar.getPagoTotal().toString());
txtTarifaMensual.setText(hipotecaActualizar.getTarifaMensual().toString());
txtSueldoNecesario.setText(hipotecaActualizar.getSueldoNecesario().toString());
txtCodigoCasa.setText(String.valueOf(hipotecaActualizar.getCodigoCasaFk().getCodigo()));
txtCedulaHipotecario.setText(hipotecaActualizar.getCedulaPropietarioFk().getCedula());

if (hipotecaActualizar.getGarante()) {
    txtCedulaGarante.setText(hipotecaActualizar.getCedulaGaranteFk().getCedula());
}


botonesActualizar();
botonesActualizar = true;
bloquearTxts();
tblHipotecas.clearSelection();

}

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

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

```

	<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		

```

hipotecaActualizar = controladorHipoteca.read(codigo);


var casa = hipotecaActualizar.getCodigoCasaFk();
casa.setEstado(EnumEstadoCasa.ENVENTA.toString());

if (controladorHipoteca.delete(hipotecaActualizar)) {
    JOptionPane.showMessageDialog(this, "Hipoteca eliminado con exito", "Mensaje",
JOptionPane.INFORMATION_MESSAGE);
    controladorCasa.update(casa);
    limpiar();
} else {
    JOptionPane.showMessageDialog(this, "Ha ocurrido un error", "Error", JOptionPane.ERROR_MESSAGE);
}

}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnCalcular;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnEliminar;
private javax.swing.JButton btnMensualidadTabla;
private com.toedter.calendar.JDateChooser jDateFechaFin;
private com.toedter.calendar.JDateChooser jDateFechaInicio;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;

```


	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 jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
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.JTable tblHipotecas;
private javax.swing.JTextField txtCargaFinanciera;
private javax.swing.JFormattedTextField txtCedulaGarante;
private javax.swing.JFormattedTextField txtCedulaHipotecario;
private javax.swing.JTextField txtCodigo;
private javax.swing.JTextField txtCodigoCasa;
private javax.swing.JTextField txtInteres;
private javax.swing.JTextField txtPlazo;
private javax.swing.JTextField txtPrestamo;
private javax.swing.JTextField txtSueldoNecesario;
private javax.swing.JTextField txtTarifaMensual;
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		

### VentanaMostrarPagos:

```
/*
 * 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.math.BigDecimal;
import java.math.RoundingMode;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.table.DefaultTableModel;
```


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

```
public class VentanaMostrarPagos extends javax.swing.JInternalFrame {
```

```
    private int plazos;
    private BigDecimal mensualidad;
    private BigDecimal total;
    private Date fechaInicio;
```

```
/**
 * Creates new form VentanaMostrarPagos
 */
```

```
public VentanaMostrarPagos() {
```

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

```

initComponents();
}

public int getPlazos() {
    return plazos;
}

public void setPlazos(int plazos) {
    this.plazos = plazos;
}

public BigDecimal getMensualidad() {
    return mensualidad;
}


public void setMensualidad(BigDecimal mensualidad) {
    this.mensualidad = mensualidad;
}

public Date getFechaInicio() {
    return fechaInicio;
}

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

public BigDecimal getTotal() {
    return total;
}

```

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

```

}

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

public void llenarTblPagos() {

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

    int mes = fechaInicio.getMonth();
    int anio = fechaInicio.getYear() + 1900;
    int dia = fechaInicio.getDate();
    boolean primerMes = false;
    BigDecimal sobrantePago = total;

    for (int i = 1; i <= (plazos + 1); i++) {
        if (mes > 11) {
            anio += 1;
            mes = 0;
        }

        if (!primerMes) {

            Date fecha = new Date(anio, mes, dia);
            SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
            String fechaTexto = formatter.format(fecha);

```

```
Object[] row = {fechaTexto, 0 + "", total.toString()
};

mes++;

primerMes = true;

modelo.addRow(row);

} else {

    sobrantePago = BigDecimal.valueOf(sobrantePago.doubleValue() -
mensualidad.doubleValue()).setScale(2, RoundingMode.CEILING);

    Date fecha = new Date(anio, mes, dia);

    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");

    String fechaTexto = formatter.format(fecha);

    if (i != (plazos + 1)) {

        Object[] row = {fechaTexto, mensualidad.toString(), sobrantePago.toString()

        };

        modelo.addRow(row);

    } else {

        mensualidad = BigDecimal.valueOf(mensualidad.doubleValue() + sobrantePago.doubleValue());

        sobrantePago = BigDecimal.valueOf(sobrantePago.doubleValue() -
sobrantePago.doubleValue()).setScale(2, RoundingMode.CEILING);

        Object[] row = {fechaTexto, mensualidad.toString(), sobrantePago.toString()


        };

        modelo.addRow(row);

    }

    mes++;
```



 <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		

```

    }

}

tblPagos.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();
    tblPagos = new javax.swing.JTable();

    setClosable(true);
    setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
    addInternalFrameListener(new javax.swing.event.InternalFrameListener() {
        public void internalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
            formInternalFrameActivated(evt);
        }
        public void internalFrameClosed(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		

```

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);
}
});

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

    },
    new String [] {
        "Fecha", "Mensualidad", "Valor restante"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false
    };

    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		

```

jScrollPane1.setViewportView(tblPagos);

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(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 676,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(23, true))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 477,
                true)
            .addContainerGap()
        )
);


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

private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {

    llenarTblPagos();

}

```

	<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 void formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
```

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

```
    llenarTblPagos();
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JScrollPane jScrollPane1;
```

```
private javax.swing.JTable tblPagos;
```

```
// End of variables declaration
```


```
}
```

#### RESULTADO(S) OBTENIDO(S):

- Afianzar los conocimientos sobre JPA.
- Completar una prueba en un periodo corto de tiempo con uso de bases de datos y JPA.
- Practicar todo lo que se ha aprendido a lo largo de esta unidad.

#### CONCLUSIONES:

En conclusión, con esta práctica he logrado mejorar en las técnicas implementación sobre Jpa. Cabe recalcar que este tema es muy importante para manejar bases de datos de una manera mucho más sencilla y rápida y no perder tanto tiempo escribiendo las sentencias SQL. Además, es un problema que ocurre en la vida real en las

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

entidades financieras, por lo que me ha ayudado para conocer más a fondo sobre como funcionan los prestamos hipotecarios.

#### RECOMENDACIONES

No existe ninguna recomendación.

**Nombre de estudiante:** Adolfo Sebastián Jara Gavilanes.

**Firma de estudiante:**

