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



PRÁCTICA DE LABORATORIO

CARRERA: COMPUTACION

ASIGNATURA: Programación aplicada.

NRO. PRÁCTICA:

TÍTULO PRÁCTICA: Bases de datos

OBJETIVO ALCANZADO:

- Afianzar conocimientos sobre bases de datos

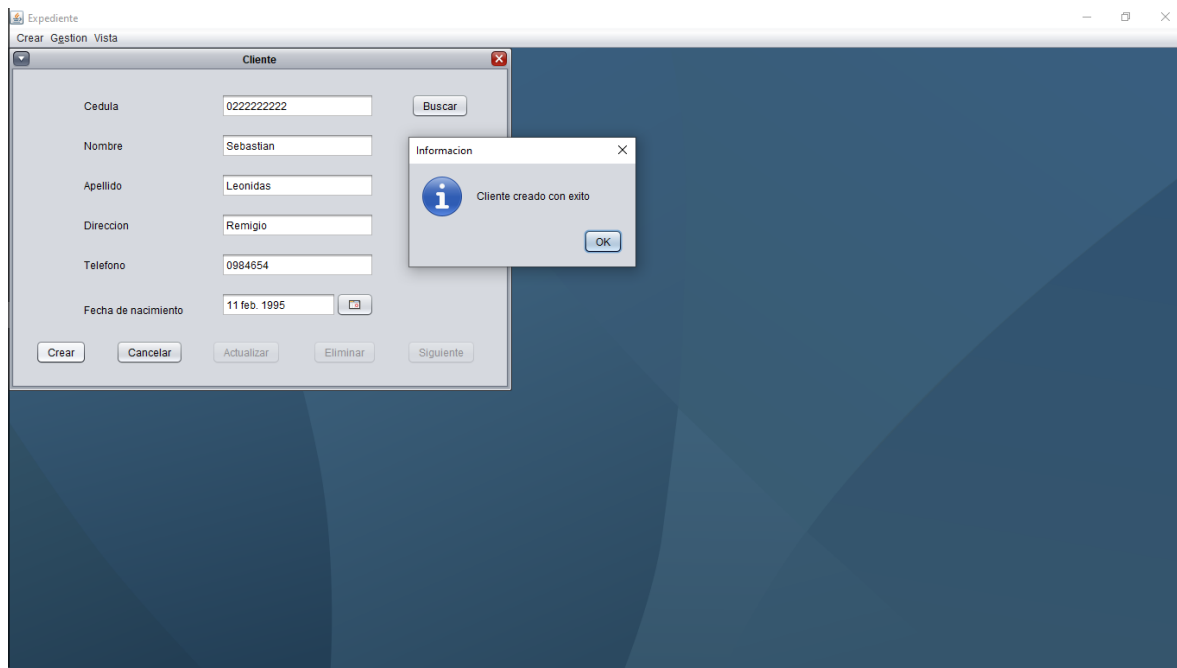
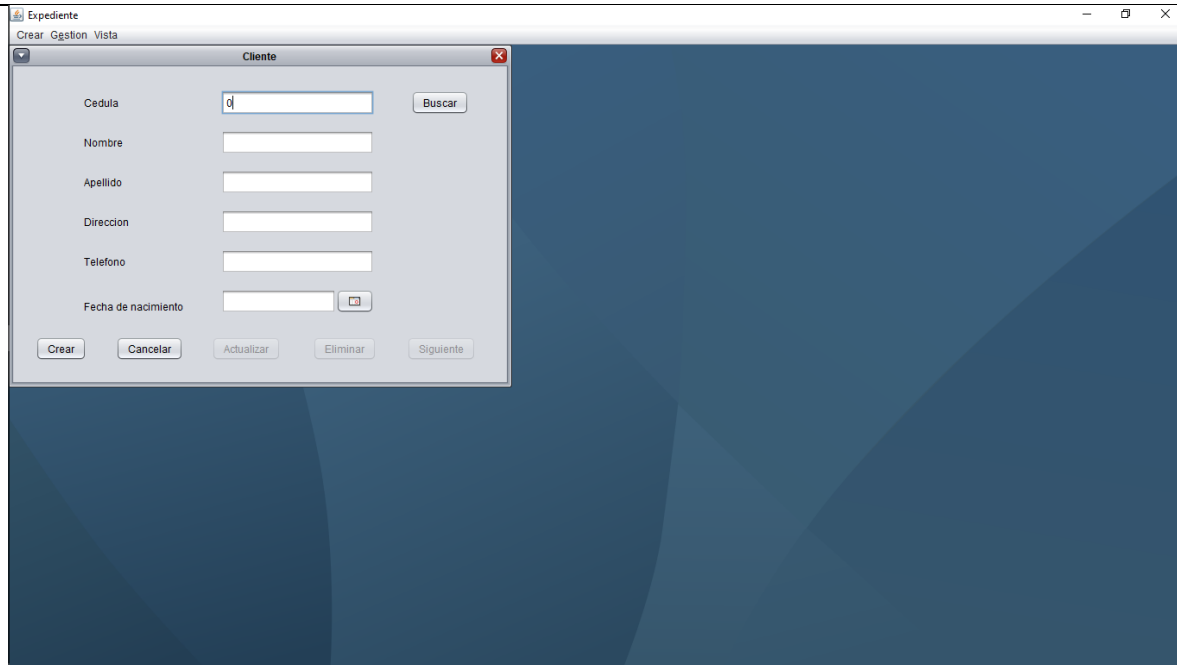
ACTIVIDADES DESARROLLADAS

1. Simulación del programa


Cuando iniciamos el programa esta es la ventana principal:

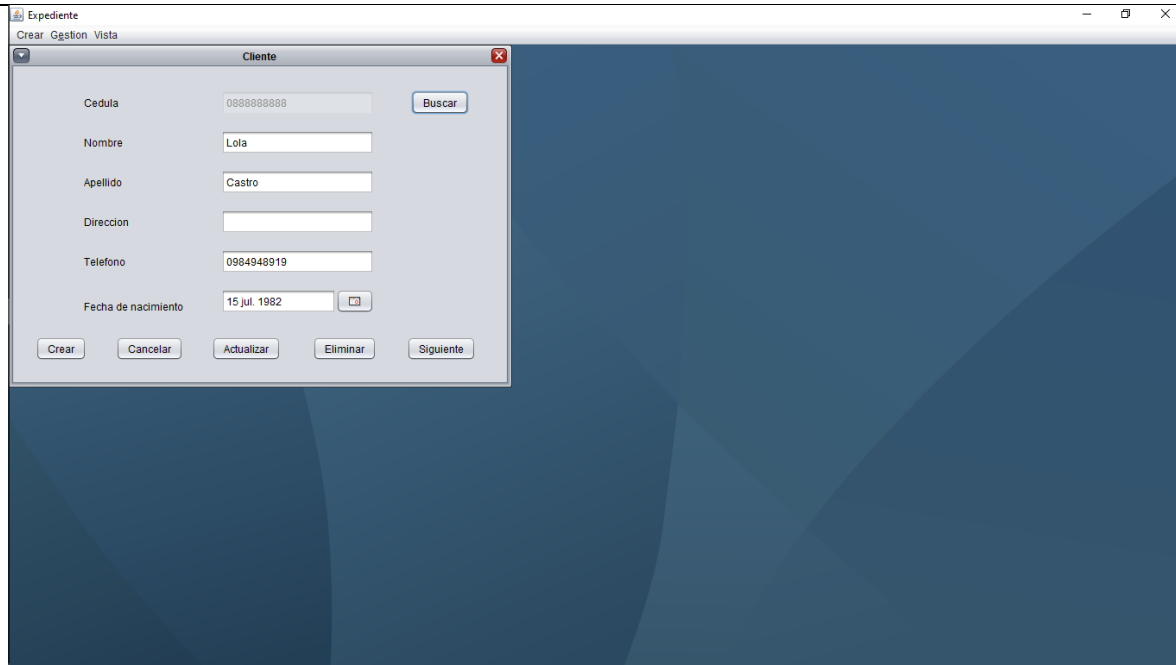


Para crear un cliente:



Ahora para actualizar un cliente se lo busca por número de cédula:

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



Expediente
Crear Gestion Vista

Cliente

Cedula: 000000000

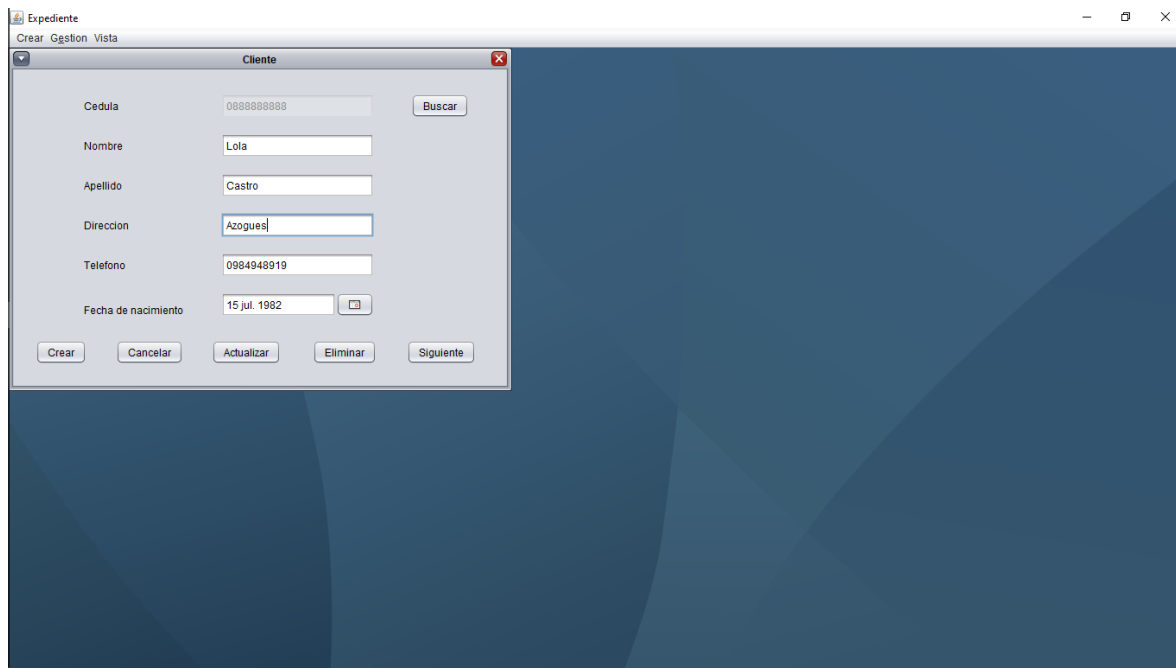
Nombre: Lola

Apellido: Castro

Direccion:

Telefono: 0984948919

Fecha de nacimiento: 15 jul. 1982



Expediente
Crear Gestion Vista

Cliente

Cedula: 000000000

Nombre: Lola

Apellido: Castro


Direccion: Azogues

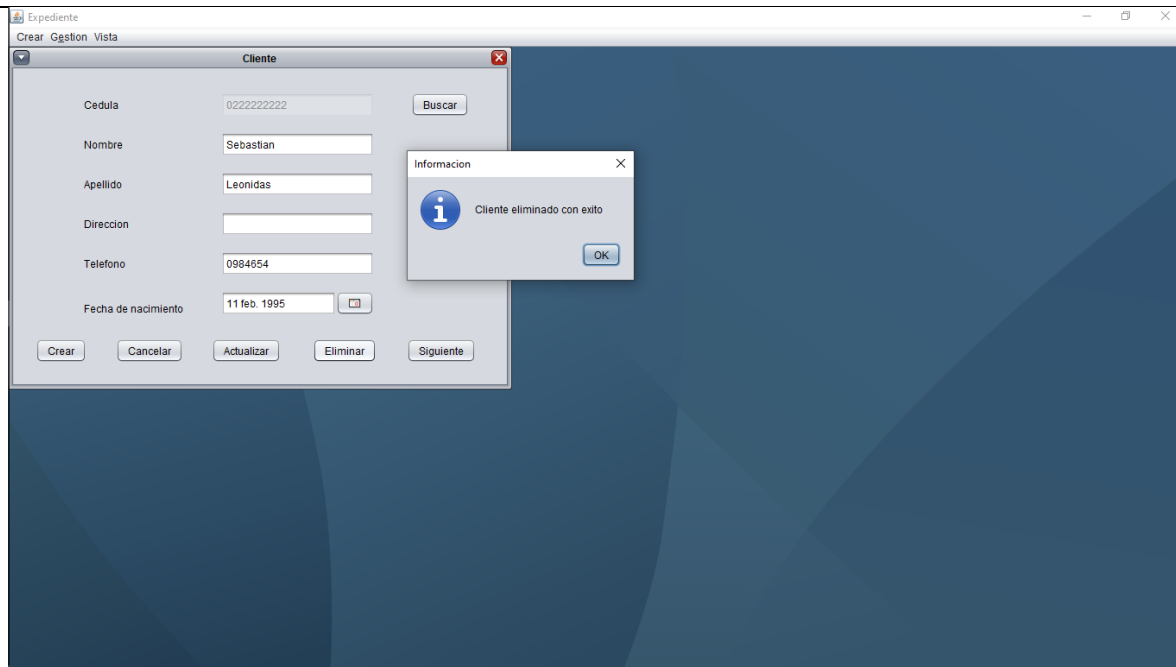
Telefono: 0984948919

Fecha de nacimiento: 15 jul. 1982

Solamente se aplasta el botón actualizar y el cliente ha sido actualizado.

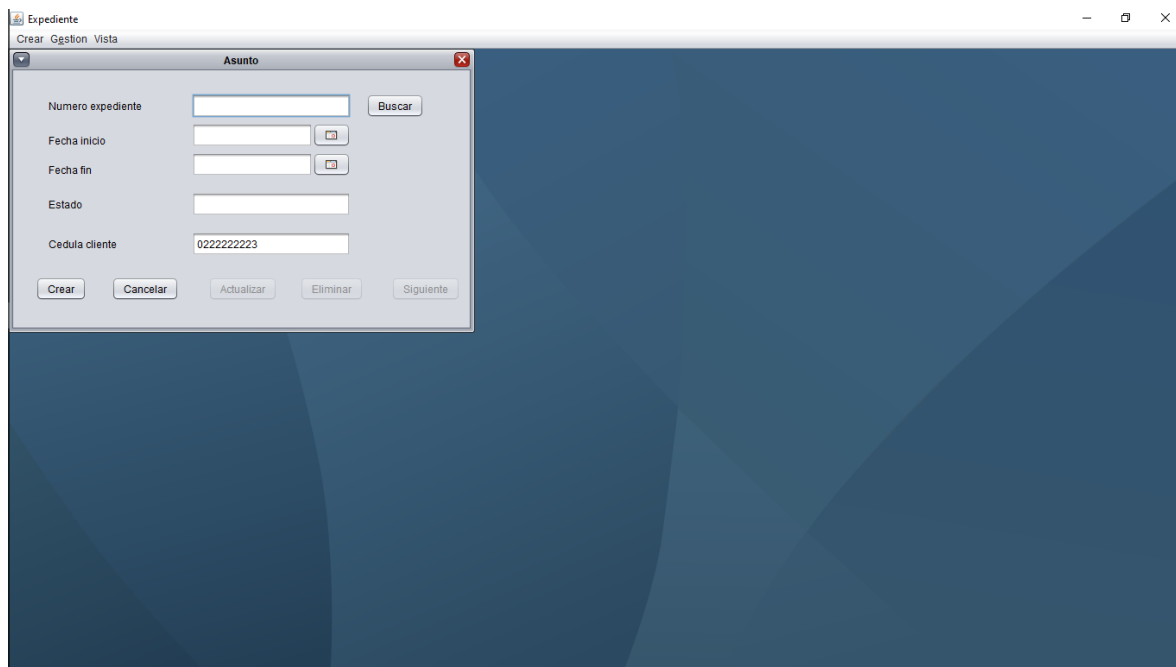
Para eliminar:

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




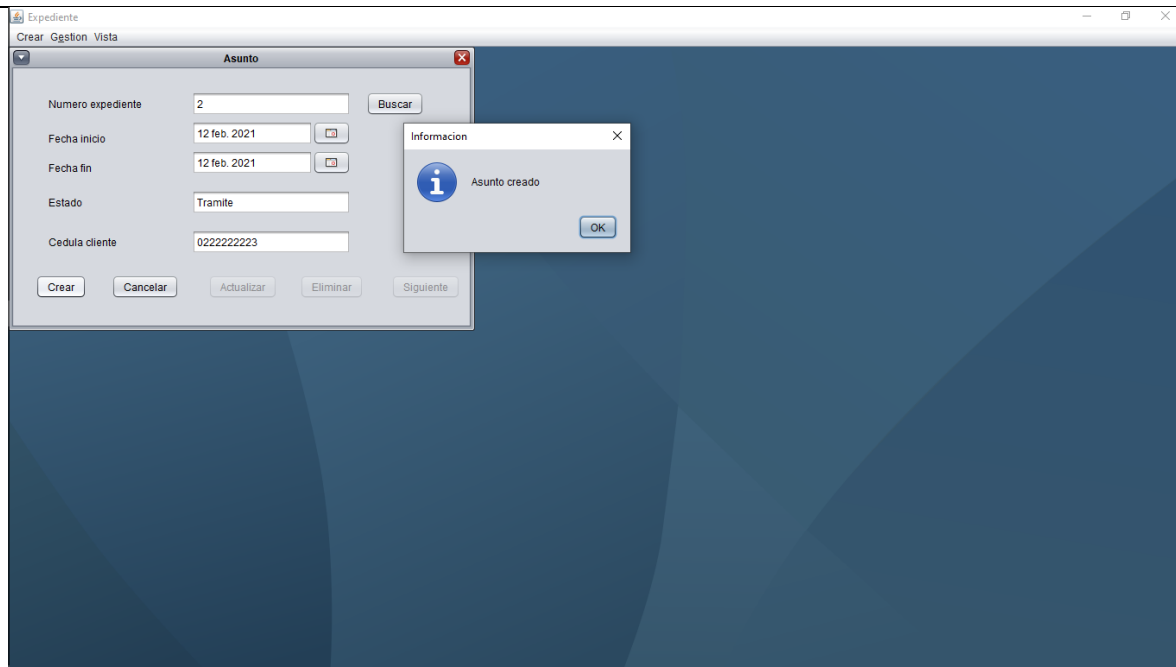
The screenshot shows the 'Expediente' application window with a menu bar (Crear, Gestion, Vista) and a toolbar. The 'Cliente' form is open, displaying fields for Cedula (022222222), Nombre (Sebastian), Apellido (Leonidas), Direccion, Telefono (0984654), and Fecha de nacimiento (11 feb. 1995). A 'Buscar' button is next to the Cedula field. Below the fields are buttons for 'Crear', 'Cancelar', 'Actualizar', 'Eliminar', and 'Siguiente'. An 'Información' dialog box is overlaid on the form, displaying an information icon and the message 'Cliente eliminado con éxito' with an 'OK' button.

Para crear el asunto:

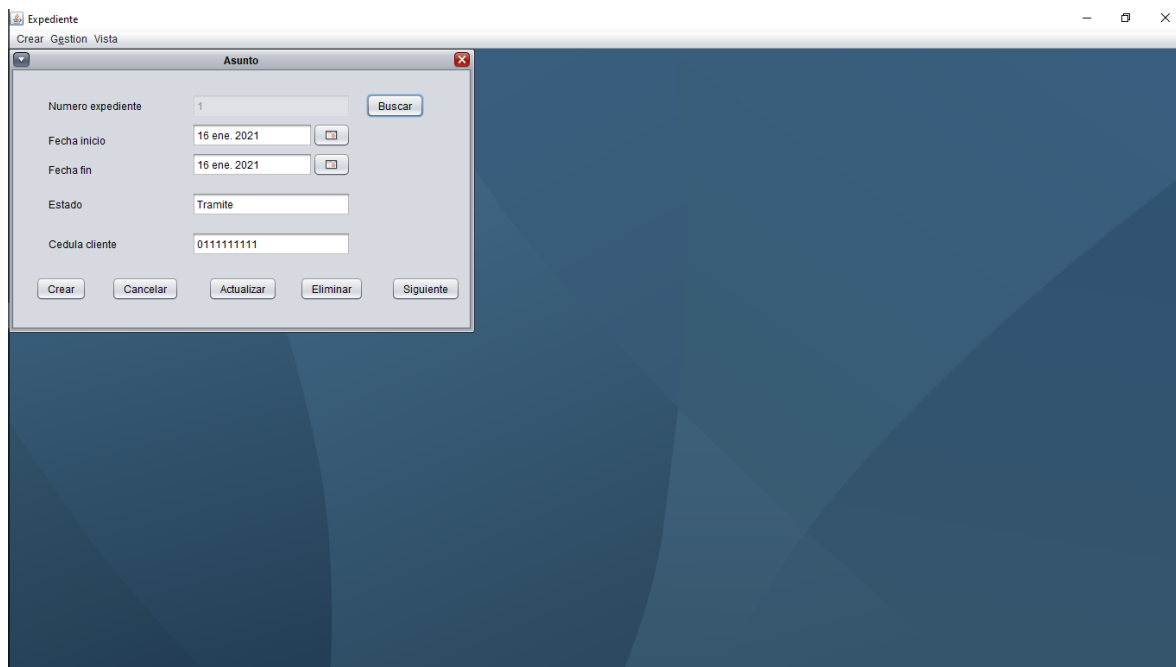



The screenshot shows the 'Expediente' application window with the 'Asunto' form open. The form contains fields for Numero expediente, Fecha inicio, Fecha fin, Estado, and Cedula cliente (022222223). A 'Buscar' button is next to the Numero expediente field. Below the fields are buttons for 'Crear', 'Cancelar', 'Actualizar', 'Eliminar', and 'Siguiente'.

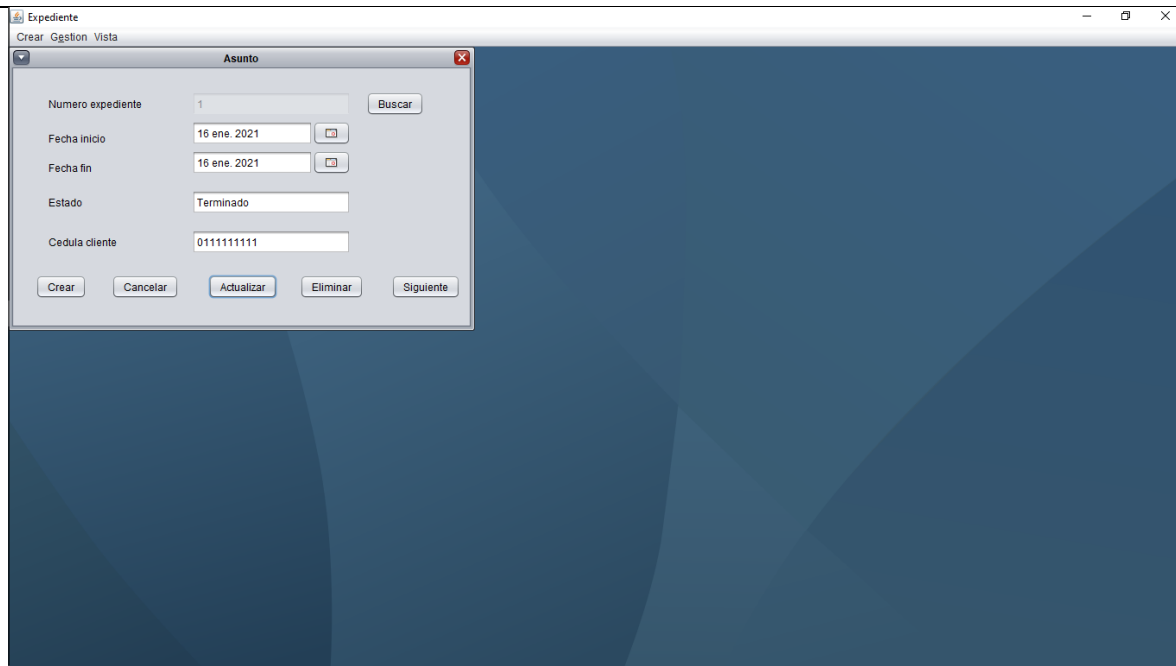
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



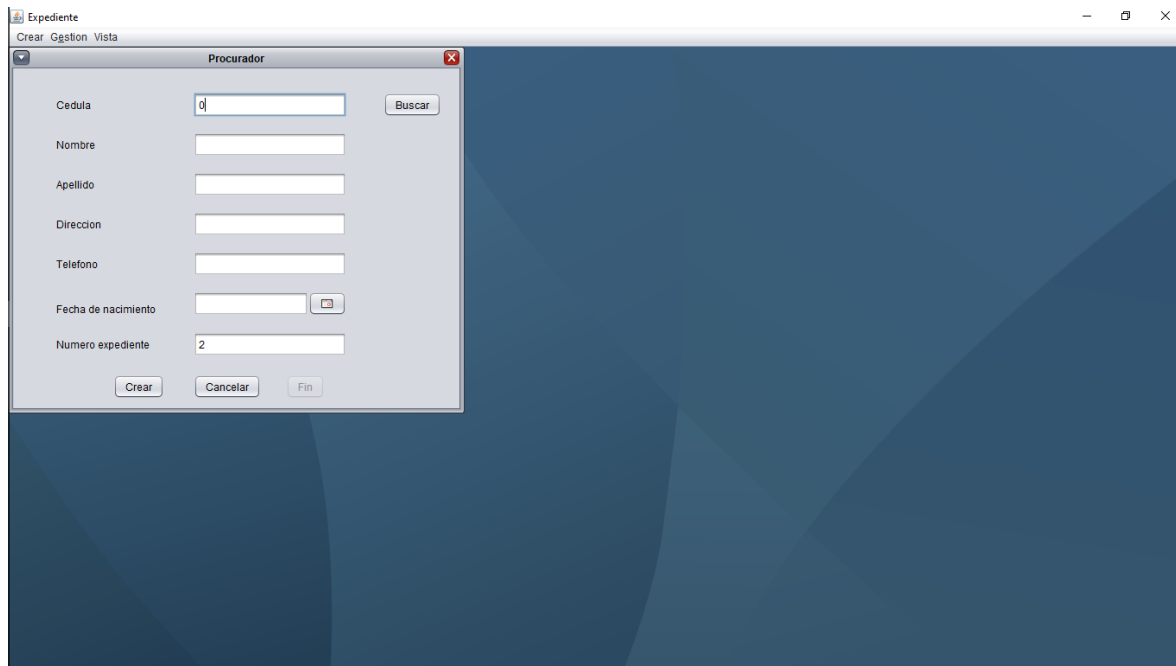
Ahora para editar:




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

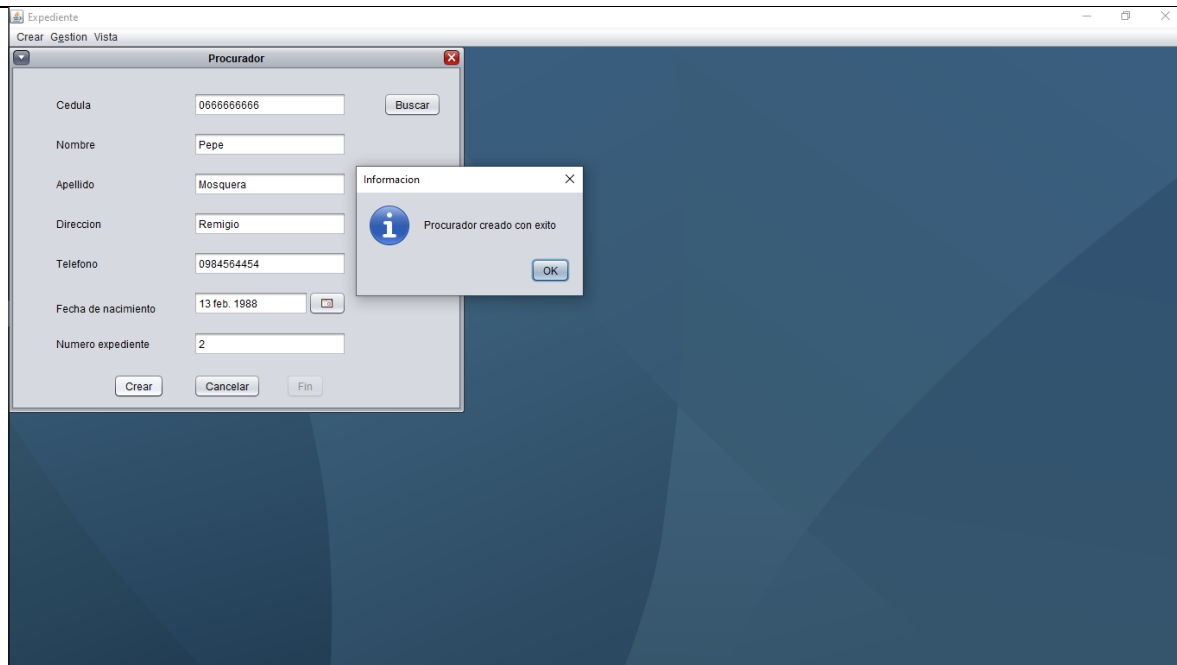


Ahora aplastamos siguiente para asignar un procurador al expediente:



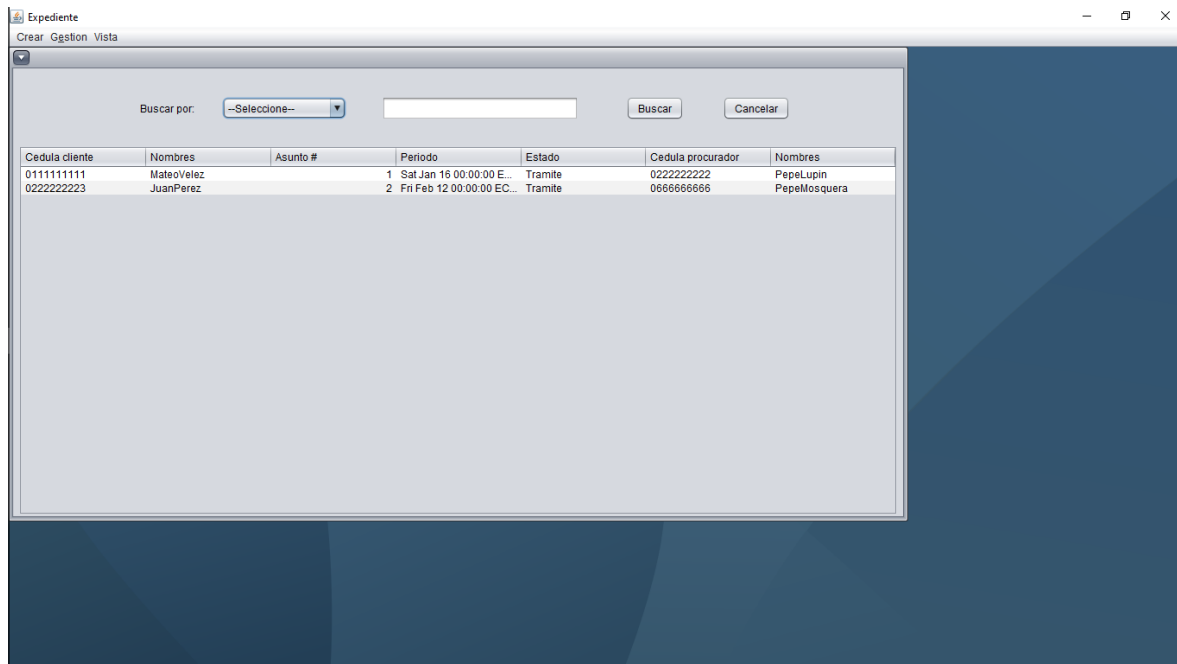
Aquí se lo puedo crear uno nuevo o buscar uno ya creado anteriormente:

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




En este caso se creó uno nuevo.

En la siguiente ventana podremos observar los expedientes creados hasta el momento:



Ahora procederemos a buscar por cedula del cliente:

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

Expediente
Crear Ggston Vista

Buscar por: Cedula cliente 022222223

Cedula cliente	Nombres	Asunto #	Periodo	Estado	Cedula procurador	Nombres
022222223	JuanPerez		2	Fri Feb 12 00 00:00 EC...	066666666	PepelMosquera

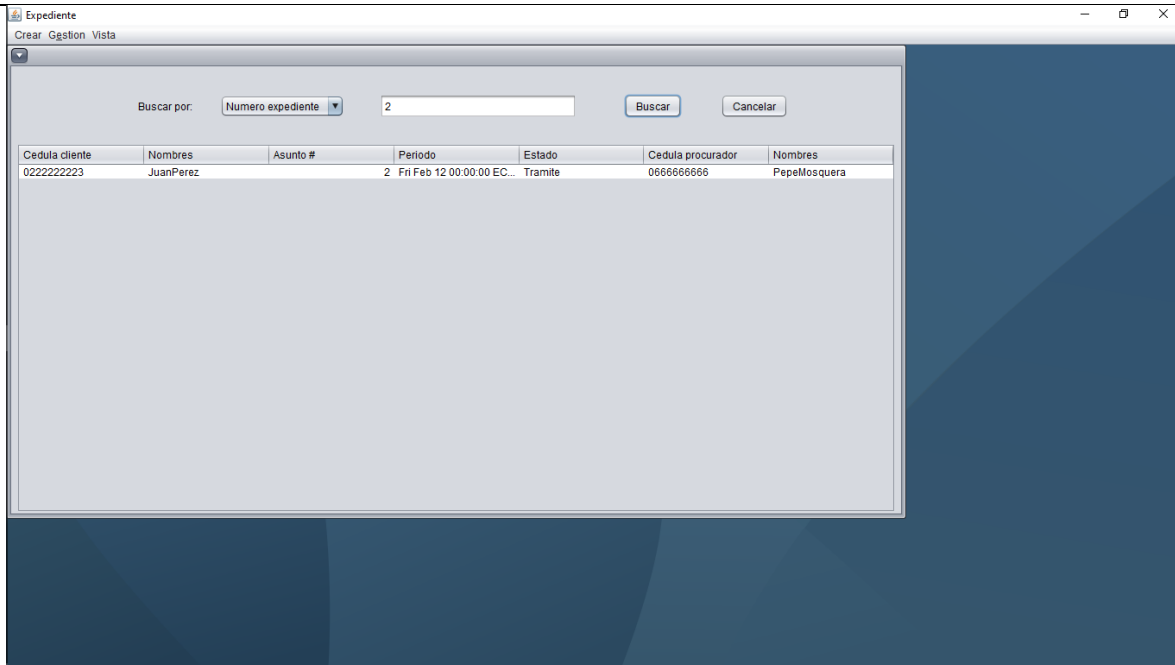
Ahora buscaremos por cedula procurador:

Expediente
Crear Ggston Vista

Buscar por: Cedula procurador 022222222

Cedula cliente	Nombres	Asunto #	Periodo	Estado	Cedula procurador	Nombres
011111111	MateoVelez		1	Sat Jan 16 00:00:00 E...	022222222	PepelLupin

Y por último, buscaremos por número expediente:



2. Código del proyecto.

Paquete modelo:

Clase cliente:

/*

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

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

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

```
import java.io.Serializable;
```


```
import java.util.Collection;
```

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

```
import javax.persistence.Basic;
```

```
import javax.persistence.CascadeType;
```

```
import javax.persistence.Column;
```

	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.Entity;

import javax.persistence.Id;

import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToOne;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;


/**
 *
 * @author User
 */
@Entity
@NamedQueries({
    @NamedQuery(name = "Cliente.findAll", query = "SELECT c FROM Cliente c"),
    @NamedQuery(name = "Cliente.findByCedula", query = "SELECT c FROM Cliente c WHERE c.cedula = :cedula"),
    @NamedQuery(name = "Cliente.findByNombre", query = "SELECT c FROM Cliente c WHERE c.nombre = :nombre"),
    @NamedQuery(name = "Cliente.findByApellido", query = "SELECT c FROM Cliente c WHERE c.apellido = :apellido"),
    @NamedQuery(name = "Cliente.findByDireccion", query = "SELECT c FROM Cliente c WHERE c.direccion = :direccion"),
    @NamedQuery(name = "Cliente.findByTelefono", query = "SELECT c FROM Cliente c WHERE c.telefono = :telefono"),
    @NamedQuery(name = "Cliente.findByFechaNacimiento", query = "SELECT c FROM Cliente c WHERE c.fechaNacimiento = :fechaNacimiento"))
})
public class Cliente implements Serializable {

    private static final long serialVersionUID = 1L;

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

    @Basic(optional = false)

```

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

```

private String nombre;
@Basic(optional = false)
private String apellido;
private String direccion;
@Basic(optional = false)
private String telefono;
@Basic(optional = false)
@Column(name = "fecha_nacimiento")
@Temporal(TemporalType.DATE)
private Date fechaNacimiento;
@OneToMany(cascade = CascadeType.ALL, mappedBy = "cedulaClienteFk")
private Collection<Asunto> asuntoCollection;


public Cliente() {
}

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

public Cliente(String cedula, String nombre, String apellido, String telefono, Date fechaNacimiento) {
    this.cedula = cedula;
    this.nombre = nombre;
    this.apellido = apellido;
    this.telefono = telefono;
    this.fechaNacimiento = fechaNacimiento;
}

public String getCedula() {

```

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

```

return cedula;
}

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 getDireccion() {
    return direccion;
}

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

```

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

```

}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public Date getFechaNacimiento() {
    return fechaNacimiento;
}


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

public Collection<Asunto> getAsuntoCollection() {
    return asuntoCollection;
}

public void setAsuntoCollection(Collection<Asunto> asuntoCollection) {
    this.asuntoCollection = asuntoCollection;
}

@Override
public int hashCode() {
    int hash = 0;

```

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

```
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 Cliente)) {
```

```
        return false;
```

```
    }
```

```
    Cliente other = (Cliente) 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 " cedula=" + cedula + " nombre: " + nombre + apellido;
```


```
}
```

```
}
```

Clase Asunto:

```
/*
```

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

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

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

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

```
import java.io.Serializable;
```

```
import java.util.Collection;
```

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

```
import javax.persistence.Basic;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.ManyToOne;
```

```
import javax.persistence.NamedQueries;
```

```
import javax.persistence.NamedQuery;
```

```
import javax.persistence.OneToMany;
```

```
import javax.persistence.Temporal;
```

```
import javax.persistence.TemporalType;
```

```
/**
```

```
*
```

```
* @author User
```


```
*/
```

```
@Entity
```

```
@NamedQueries({
```

```
    @NamedQuery(name = "Asunto.findAll", query = "SELECT a FROM Asunto a"),
```

```
    @NamedQuery(name = "Asunto.findByNumeroExpediente", query = "SELECT a FROM Asunto a WHERE  
a.numeroExpediente = :numeroExpediente"),
```

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

```

@NamedQuery(name = "Asunto.findByFechaInicio", query = "SELECT a FROM Asunto a WHERE a.fechaInicio = :fechaInicio"),

@NamedQuery(name = "Asunto.findByFechaFin", query = "SELECT a FROM Asunto a WHERE a.fechaFin = :fechaFin"),

@NamedQuery(name = "Asunto.findByEstado", query = "SELECT a FROM Asunto a WHERE a.estado = :estado"))

public class Asunto implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id

    @Basic(optional = false)

    @Column(name = "numero_expediente")

    private Integer numeroExpediente;

    @Basic(optional = false)

    @Column(name = "fecha_inicio")

    @Temporal(TemporalType.DATE)

    private Date fechaInicio;

    @Basic(optional = false)

    @Column(name = "fecha_fin")

    @Temporal(TemporalType.DATE)

    private Date fechaFin;

    @Basic(optional = false)

    private String estado;

    @JoinColumn(name = "cedula_cliente_fk", referencedColumnName = "cedula")

    @ManyToOne(optional = false)

    private Cliente cedulaClienteFk;


    @OneToMany(mappedBy = "numeroAsuntoFk")

    private Collection<Procurador> procuradorCollection;

    public Asunto() {

    }

```


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

```

public Asunto(Integer numeroExpediente) {
    this.numeroExpediente = numeroExpediente;
}

public Asunto(Integer numeroExpediente, Date fechalnicio, Date fechaFin, String estado) {
    this.numeroExpediente = numeroExpediente;
    this.fechalnicio = fechalnicio;
    this.fechaFin = fechaFin;
    this.estado = estado;
}

public Integer getNumeroExpediente() {
    return numeroExpediente;
}


public void setNumeroExpediente(Integer numeroExpediente) {
    this.numeroExpediente = numeroExpediente;
}

public Date getFechalnicio() {
    return fechalnicio;
}

public void setFechalnicio(Date fechalnicio) {
    this.fechalnicio = fechalnicio;
}

public Date getFechaFin() {

```

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

```

return fechaFin;
}

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

public String getEstado() {
    return estado;
}

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


public Cliente getCedulaClienteFk() {
    return cedulaClienteFk;
}

public void setCedulaClienteFk(Cliente cedulaClienteFk) {
    this.cedulaClienteFk = cedulaClienteFk;
}

public Collection<Procurador> getProcuradorCollection() {
    return procuradorCollection;
}

public void setProcuradorCollection(Collection<Procurador> procuradorCollection) {
    this.procuradorCollection = procuradorCollection;
}

```

	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 += (numeroExpediente != null ? numeroExpediente.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 Asunto)) {
        return false;
    }
    Asunto other = (Asunto) object;
    if ((this.numeroExpediente == null && other.numeroExpediente != null) || (this.numeroExpediente != null &&
!this.numeroExpediente.equals(other.numeroExpediente))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return " numeroExpediente=" + numeroExpediente + " cliente: " + cedulaClienteFk;
}
}

```

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

Clase Procurador:

```


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

package ups.edu.ec.modelo;

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

@NamedQueries({

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

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

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

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

    @NamedQuery(name = "Procurador.findByTelefono", query = "SELECT p FROM Procurador p WHERE p.telefono = :telefono"),

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

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

    @NamedQuery(name = "Procurador.findById", query = "SELECT p FROM Procurador p WHERE p.id = :id")))

public class Procurador implements Serializable {

    private static final long serialVersionUID = 1L;

    @Basic(optional = false)

    private String nombre;

    @Basic(optional = false)

    private String apellido;

    private String direccion;

    @Basic(optional = false)

    private String telefono;


    @Basic(optional = false)

    @Column(name = "fecha_nacimiento")

    @Temporal(TemporalType.DATE)

    private Date fechaNacimiento;

    private String cedula;
```

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

```

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Basic(optional = false)
private Integer id;

@JoinColumn(name = "numero_asunto_fk", referencedColumnName = "numero_expediente")
@ManyToOne
private Asunto numeroAsuntoFk;

public Procurador() {
}


public Procurador(Integer id) {
    this.id = id;
}

public Procurador(Integer id, String nombre, String apellido, String telefono, Date fechaNacimiento) {
    this.id = id;
    this.nombre = nombre;
    this.apellido = apellido;
    this.telefono = telefono;
    this.fechaNacimiento = fechaNacimiento;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    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		

```

}

public String getApellido() {
    return apellido;
}

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

public String getDireccion() {
    return direccion;
}


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

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public Date getFechaNacimiento() {
    return fechaNacimiento;
}

```

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

```

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

public String getCedula() {
    return cedula;
}

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


public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public Asunto getNumeroAsuntoFk() {
    return numeroAsuntoFk;
}

public void setNumeroAsuntoFk(Asunto numeroAsuntoFk) {
    this.numeroAsuntoFk = numeroAsuntoFk;
}

```


	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 += (id != null ? id.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 Procurador)) {
        return false;
    }
    Procurador other = (Procurador) object;
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
        return false;
    }
    return true;
}
```


@Override

```
public String toString() {
    return "Procurador id=" + id + " nombre: " + nombre + " " + apellido + " Asunto: " + numeroAsuntoFk;
}

}
```

Paquete Controlador:

ControladorAsunto:

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

```

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

```


```

package ups.edu.ec.controlador;

import java.io.Serializable;
import javax.persistence.Query;
import javax.persistence.EntityNotFoundException;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;
import ups.edu.ec.modelo.Cliente;
import ups.edu.ec.modelo.Procurador;
import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import ups.edu.ec.controlador.exceptions.NonexistentEntityException;
import ups.edu.ec.controlador.exceptions.PreexistingEntityException;
import ups.edu.ec.modelo.Asunto;

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

```

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

```

public AsuntoJpaController(EntityManagerFactory emf) {
    this.emf = emf;
}


public AsuntoJpaController() {
    this.emf = Persistence.createEntityManagerFactory("Practica_en_clasePU");
}

private EntityManagerFactory emf = null;

public EntityManager getEntityManager() {
    return emf.createEntityManager();
}

public void create(Asunto asunto) throws PreexistingEntityException, Exception {
    if (asunto.getProcuradorCollection() == null) {
        asunto.setProcuradorCollection(new ArrayList<Procurador>());
    }
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Cliente cedulaClienteFk = asunto.getCedulaClienteFk();
        if (cedulaClienteFk != null) {
            cedulaClienteFk = em.getReference(cedulaClienteFk.getClass(), cedulaClienteFk.getCedula());
            asunto.setCedulaClienteFk(cedulaClienteFk);
        }
        Collection<Procurador> attachedProcuradorCollection = new ArrayList<Procurador>();


```

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

```

for (Procurador procuradorCollectionProcuradorToAttach : asunto.getProcuradorCollection()) {
    procuradorCollectionProcuradorToAttach = em.getReference(procuradorCollectionProcuradorToAttach.getClass(),
procuradorCollectionProcuradorToAttach.getCedula());
    attachedProcuradorCollection.add(procuradorCollectionProcuradorToAttach);
}
asunto.setProcuradorCollection(attachedProcuradorCollection);
em.persist(asunto);
if (cedulaClienteFk != null) {
    cedulaClienteFk.getAsuntoCollection().add(asunto);
    cedulaClienteFk = em.merge(cedulaClienteFk);
}
for (Procurador procuradorCollectionProcurador : asunto.getProcuradorCollection()) {
    Asunto oldNumeroAsuntoFkOfProcuradorCollectionProcurador =
procuradorCollectionProcurador.getNumeroAsuntoFk();
    procuradorCollectionProcurador.setNumeroAsuntoFk(asunto);
    procuradorCollectionProcurador = em.merge(procuradorCollectionProcurador);
    if (oldNumeroAsuntoFkOfProcuradorCollectionProcurador != null) {
        oldNumeroAsuntoFkOfProcuradorCollectionProcurador.getProcuradorCollection().
remove(procuradorCollectionProcurador);
        oldNumeroAsuntoFkOfProcuradorCollectionProcurador =
em.merge(oldNumeroAsuntoFkOfProcuradorCollectionProcurador);
    }
}
em.getTransaction().commit();
} catch (Exception ex) {
    if (findAsunto(asunto.getNumeroExpediente()) != null) {
        throw new PreexistingEntityException("Asunto " + asunto + " already exists.", ex);
    }
    throw ex;
} finally {

```


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

```

if (em != null) {
    em.close();
}
}
}

public void edit(Asunto asunto) throws NonexistentEntityException, Exception {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Asunto persistentAsunto = em.find(Asunto.class, asunto.getNumeroExpediente());
        Cliente cedulaClienteFkOld = persistentAsunto.getCedulaClienteFk();
        Cliente cedulaClienteFkNew = asunto.getCedulaClienteFk();
        Collection<Procurador> procuradorCollectionOld = persistentAsunto.getProcuradorCollection();
        Collection<Procurador> procuradorCollectionNew = asunto.getProcuradorCollection();
        if (cedulaClienteFkNew != null) {
            cedulaClienteFkNew = em.getReference(cedulaClienteFkNew.getClass(), cedulaClienteFkNew.getCedula());
            asunto.setCedulaClienteFk(cedulaClienteFkNew);
        }
        Collection<Procurador> attachedProcuradorCollectionNew = new ArrayList<Procurador>();
        for (Procurador procuradorCollectionNewProcuradorToAttach : procuradorCollectionNew) {
            procuradorCollectionNewProcuradorToAttach =
em.getReference(procuradorCollectionNewProcuradorToAttach.getClass(),
procuradorCollectionNewProcuradorToAttach.getId());
            attachedProcuradorCollectionNew.add(procuradorCollectionNewProcuradorToAttach);
        }
        procuradorCollectionNew = attachedProcuradorCollectionNew;
        asunto.setProcuradorCollection(procuradorCollectionNew);
        asunto = em.merge(asunto);
    } catch (Exception ex) {
        throw new NonexistentEntityException("The asunto with id " + asunto.getId() + " does not exist.", ex);
    }
}

```

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

```

if (cedulaClienteFkOld != null && !cedulaClienteFkOld.equals(cedulaClienteFkNew)) {
    cedulaClienteFkOld.getAsuntoCollection().remove(asunto);
    cedulaClienteFkOld = em.merge(cedulaClienteFkOld);
}

if (cedulaClienteFkNew != null && !cedulaClienteFkNew.equals(cedulaClienteFkOld)) {
    cedulaClienteFkNew.getAsuntoCollection().add(asunto);
    cedulaClienteFkNew = em.merge(cedulaClienteFkNew);
}

for (Procurador procuradorCollectionOldProcurador : procuradorCollectionOld) {
    if (!procuradorCollectionNew.contains(procuradorCollectionOldProcurador)) {
        procuradorCollectionOldProcurador.setNumeroAsuntoFk(null);
        procuradorCollectionOldProcurador = em.merge(procuradorCollectionOldProcurador);
    }
}

for (Procurador procuradorCollectionNewProcurador : procuradorCollectionNew) {
    if (!procuradorCollectionOld.contains(procuradorCollectionNewProcurador)) {
        Asunto oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador =
procuradorCollectionNewProcurador.getNumeroAsuntoFk();

        procuradorCollectionNewProcurador.setNumeroAsuntoFk(asunto);
        procuradorCollectionNewProcurador = em.merge(procuradorCollectionNewProcurador);


        if (oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador != null &&
!oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador.equals(asunto)) {
            oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador.getProcuradorCollection().
remove(procuradorCollectionNewProcurador);

            oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador =
em.merge(oldNumeroAsuntoFkOfProcuradorCollectionNewProcurador);
        }
    }
}

em.getTransaction().commit();

```

```
} catch (Exception ex) {  
    String msg = ex.getLocalizedMessage();  
    if (msg == null || msg.length() == 0) {  
        Integer id = asunto.getNumeroExpediente();  
        if (findAsunto(id) == null) {  
            throw new NonexistentEntityException("The asunto with id " + id + " no longer exists.");  
        }  
    }  
    throw ex;  
}  
} finally {  
    if (em != null) {  
        em.close();  
    }  
}  
}  
  
public void destroy(Integer id) throws NonexistentEntityException {  
    EntityManager em = null;  
    try {  
        em = getEntityManager();  
        em.getTransaction().begin();  
        Asunto asunto;  
        try {  
            asunto = em.getReference(Asunto.class, id);  
            asunto.getNumeroExpediente();  
        } catch (EntityNotFoundException enfe) {  
            throw new NonexistentEntityException("The asunto with id " + id + " no longer exists.", enfe);  
        }  
        Cliente cedulaClienteFk = asunto.getCedulaClienteFk();
```

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

```

if (cedulaClienteFk != null) {
    cedulaClienteFk.getAsuntoCollection().remove(asunto);
    cedulaClienteFk = em.merge(cedulaClienteFk);
}

Collection<Procurador> procuradorCollection = asunto.getProcuradorCollection();
for (Procurador procuradorCollectionProcurador : procuradorCollection) {
    procuradorCollectionProcurador.setNumeroAsuntoFk(null);
    procuradorCollectionProcurador = em.merge(procuradorCollectionProcurador);
}


em.remove(asunto);
em.getTransaction().commit();
} finally {
    if (em != null) {
        em.close();
    }
}
}

public List<Asunto> findAsuntoEntities() {
    return findAsuntoEntities(true, -1, -1);
}

public List<Asunto> findAsuntoEntities(int maxResults, int firstResult) {
    return findAsuntoEntities(false, maxResults, firstResult);
}

private List<Asunto> findAsuntoEntities(boolean all, int maxResults, int firstResult) {
    EntityManager em = getEntityManager();
    try {

```


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


```

CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
cq.select(cq.from(Asunto.class));
Query q = em.createQuery(cq);
if (!all) {
    q.setMaxResults(maxResults);
    q.setFirstResult(firstResult);
}
return q.getResultList();
} finally {
    em.close();
}
}

public Asunto findAsunto(Integer id) {
    EntityManager em = getEntityManager();
    try {
        return em.find(Asunto.class, id);
    } finally {
        em.close();
    }
}

public int getAsuntoCount() {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
        Root<Asunto> rt = cq.from(Asunto.class);
        cq.select(em.getCriteriaBuilder().count(rt));
        Query q = em.createQuery(cq);

```

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

```

        return ((Long) q.getSingleResult()).intValue();
    } finally {
        em.close();
    }
}
}
}
}

```


ControladorCliente:

```

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

import java.io.Serializable;
import javax.persistence.Query;
import javax.persistence.EntityNotFoundException;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;
import ups.edu.ec.modelo.Asunto;
import java.util.ArrayList;
import java.util.Collection;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

```

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

```

import ups.edu.ec.controlador.exceptions.IllegalOrphanException;

import ups.edu.ec.controlador.exceptions.NonexistentEntityException;

import ups.edu.ec.controlador.exceptions.PreexistingEntityException;

import ups.edu.ec.modelo.Cliente;

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

    public ClienteJpaController(EntityManagerFactory emf) {
        this.emf = emf;
    }


    public ClienteJpaController() {
        this.emf = Persistence.createEntityManagerFactory("Practica_en_clasePU");
    }

    private EntityManagerFactory emf = null;

    public EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    public void create(Cliente cliente) throws PreexistingEntityException, Exception {
        if (cliente.getAsuntoCollection() == null) {
            cliente.setAsuntoCollection(new ArrayList<Asunto>());
        }
    }

```

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

```

EntityManager em = null;

try {
    em = getEntityManager();
    em.getTransaction().begin();

    Collection<Asunto> attachedAsuntoCollection = new ArrayList<Asunto>();

    for (Asunto asuntoCollectionAsuntoToAttach : cliente.getAsuntoCollection()) {
        asuntoCollectionAsuntoToAttach = em.getReference(asuntoCollectionAsuntoToAttach.getClass(),
asuntoCollectionAsuntoToAttach.getNumeroExpediente());

        attachedAsuntoCollection.add(asuntoCollectionAsuntoToAttach);
    }


    cliente.setAsuntoCollection(attachedAsuntoCollection);
    em.persist(cliente);

    for (Asunto asuntoCollectionAsunto : cliente.getAsuntoCollection()) {
        Cliente oldCedulaClienteFkOfAsuntoCollectionAsunto = asuntoCollectionAsunto.getCedulaClienteFk();
        asuntoCollectionAsunto.setCedulaClienteFk(cliente);
        asuntoCollectionAsunto = em.merge(asuntoCollectionAsunto);
        if (oldCedulaClienteFkOfAsuntoCollectionAsunto != null) {
            oldCedulaClienteFkOfAsuntoCollectionAsunto.getAsuntoCollection().remove(asuntoCollectionAsunto);
            oldCedulaClienteFkOfAsuntoCollectionAsunto = em.merge(oldCedulaClienteFkOfAsuntoCollectionAsunto);
        }
    }

    em.getTransaction().commit();
} catch (Exception ex) {
    if (findCliente(cliente.getCedula()) != null) {
        throw new PreexistingEntityException("Cliente " + cliente + " already exists.", ex);
    }

    throw ex;
} finally {
    if (em != null) {

```

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

```

em.close();


}

}

}

public void edit(Cliente cliente) throws IllegalOrphanException, NonexistentEntityException, Exception {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Cliente persistentCliente = em.find(Cliente.class, cliente.getCedula());
        Collection<Asunto> asuntoCollectionOld = persistentCliente.getAsuntoCollection();
        Collection<Asunto> asuntoCollectionNew = cliente.getAsuntoCollection();
        List<String> illegalOrphanMessages = null;
        for (Asunto asuntoCollectionOldAsunto : asuntoCollectionOld) {
            if (!asuntoCollectionNew.contains(asuntoCollectionOldAsunto)) {
                if (illegalOrphanMessages == null) {
                    illegalOrphanMessages = new ArrayList<String>();
                }
                illegalOrphanMessages.add("You must retain Asunto " + asuntoCollectionOldAsunto + " since its cedulaClienteFk field is not nullable.");
            }
        }
        if (illegalOrphanMessages != null) {
            throw new IllegalOrphanException(illegalOrphanMessages);
        }
        Collection<Asunto> attachedAsuntoCollectionNew = new ArrayList<Asunto>();
        for (Asunto asuntoCollectionNewAsuntoToAttach : asuntoCollectionNew) {
            asuntoCollectionNewAsuntoToAttach = em.getReference(asuntoCollectionNewAsuntoToAttach.getClass(), asuntoCollectionNewAsuntoToAttach.getNumeroExpediente());

```


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

```

        attachedAsuntoCollectionNew.add(asuntoCollectionNewAsuntoToAttach);
    }

    asuntoCollectionNew = attachedAsuntoCollectionNew;
    cliente.setAsuntoCollection(asuntoCollectionNew);
    cliente = em.merge(cliente);
    for (Asunto asuntoCollectionNewAsunto : asuntoCollectionNew) {
        if (!asuntoCollectionOld.contains(asuntoCollectionNewAsunto)) {
            Cliente oldCedulaClienteFkOfAsuntoCollectionNewAsunto = asuntoCollectionNewAsunto.getCedulaClienteFk();
            asuntoCollectionNewAsunto.setCedulaClienteFk(cliente);
            asuntoCollectionNewAsunto = em.merge(asuntoCollectionNewAsunto);
            if (oldCedulaClienteFkOfAsuntoCollectionNewAsunto != null &&
!oldCedulaClienteFkOfAsuntoCollectionNewAsunto.equals(cliente)) {
                oldCedulaClienteFkOfAsuntoCollectionNewAsunto.getAsuntoCollection().remove(asuntoCollectionNewAsunto);
                oldCedulaClienteFkOfAsuntoCollectionNewAsunto =
em.merge(oldCedulaClienteFkOfAsuntoCollectionNewAsunto);
            }
        }
    }
    em.getTransaction().commit();
} catch (Exception ex) {
    String msg = ex.getLocalizedMessage();
    if (msg == null || msg.length() == 0) {
        String id = cliente.getCedula();
        if (findCliente(id) == null) {
            throw new NonexistentEntityException("The cliente with id " + id + " no longer exists.");
        }
    }
    throw ex;
} finally {
    if (em != null) {

```

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

```

em.close();


}

}

}

public void destroy(String id) throws IllegalOrphanException, NonexistentEntityException {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Cliente cliente;
        try {
            cliente = em.getReference(Cliente.class, id);
            cliente.getCedula();
        } catch (EntityNotFoundException enfe) {
            throw new NonexistentEntityException("The cliente with id " + id + " no longer exists.", enfe);
        }
        List<String> illegalOrphanMessages = null;
        Collection<Asunto> asuntoCollectionOrphanCheck = cliente.getAsuntoCollection();
        for (Asunto asuntoCollectionOrphanCheckAsunto : asuntoCollectionOrphanCheck) {
            if (illegalOrphanMessages == null) {
                illegalOrphanMessages = new ArrayList<String>();
            }
            illegalOrphanMessages.add("This Cliente (" + cliente + ") cannot be destroyed since the Asunto " +
asuntoCollectionOrphanCheckAsunto + " in its asuntoCollection field has a non-nullable cedulaClienteFk field.");
        }
        if (illegalOrphanMessages != null) {
            throw new IllegalOrphanException(illegalOrphanMessages);
        }
    }
}

```

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

```

em.remove(cliente);

em.getTransaction().commit();

} finally {
    if (em != null) {
        em.close();
    }
}

}


}

public List<Cliente> findClienteEntities() {
    return findClienteEntities(true, -1, -1);
}

public List<Cliente> findClienteEntities(int maxResults, int firstResult) {
    return findClienteEntities(false, maxResults, firstResult);
}

private List<Cliente> findClienteEntities(boolean all, int maxResults, int firstResult) {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
        cq.select(cq.from(Cliente.class));
        Query q = em.createQuery(cq);
        if (!all) {
            q.setMaxResults(maxResults);
            q.setFirstResult(firstResult);
        }
        return q.getResultList();
    } finally {

```


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

```


        em.close();
    }
}

public Cliente findCliente(String id) {
    EntityManager em = getEntityManager();
    try {
        return em.find(Cliente.class, id);
    } finally {
        em.close();
    }
}

public int getClienteCount() {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
        Root<Cliente> rt = cq.from(Cliente.class);
        cq.select(em.getCriteriaBuilder().count(rt));
        Query q = em.createQuery(cq);
        return ((Long) q.getSingleResult()).intValue();
    } finally {
        em.close();
    }
}
}

```

ControladorProcurador:

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

```

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

```

```

*/
package ups.edu.ec.controlador;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Query;
import javax.persistence.EntityNotFoundException;
import javax.persistence.Persistence;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;
import ups.edu.ec.controlador.exceptions.NonexistentEntityException;
import ups.edu.ec.modelo.Asunto;
import ups.edu.ec.modelo.Procurador;


```

```

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

    public ProcuradorJpaController(EntityManagerFactory emf) {

```

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

```

this.emf = emf;
}


public ProcuradorJpaController() {
    this.emf = Persistence.createEntityManagerFactory("Practica_en_clasePU");
}

private EntityManagerFactory emf = null;

public EntityManager getEntityManager() {
    return emf.createEntityManager();
}

public void create(Procurador procurador) {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Asunto numeroAsuntoFk = procurador.getNumeroAsuntoFk();
        if (numeroAsuntoFk != null) {
            numeroAsuntoFk = em.getReference(numeroAsuntoFk.getClass(), numeroAsuntoFk.getNumeroExpediente());
            procurador.setNumeroAsuntoFk(numeroAsuntoFk);
        }
        em.persist(procurador);
        if (numeroAsuntoFk != null) {
            numeroAsuntoFk.getProcuradorCollection().add(procurador);
            numeroAsuntoFk = em.merge(numeroAsuntoFk);
        }
        em.getTransaction().commit();
    }
}

```

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

```

    } finally {
        if (em != null) {
            em.close();
        }
    }
}


public void edit(Procurador procurador) throws NonexistentEntityException, Exception {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();

        Procurador persistentProcurador = em.find(Procurador.class, procurador.getId());
        Asunto numeroAsuntoFkOld = persistentProcurador.getNumeroAsuntoFk();
        Asunto numeroAsuntoFkNew = procurador.getNumeroAsuntoFk();
        if (numeroAsuntoFkNew != null) {
            numeroAsuntoFkNew = em.getReference(numeroAsuntoFkNew.getClass(),
numeroAsuntoFkNew.getNumeroExpediente());
            procurador.setNumeroAsuntoFk(numeroAsuntoFkNew);
        }
        procurador = em.merge(procurador);
        if (numeroAsuntoFkOld != null && !numeroAsuntoFkOld.equals(numeroAsuntoFkNew)) {
            numeroAsuntoFkOld.getProcuradorCollection().remove(procurador);
            numeroAsuntoFkOld = em.merge(numeroAsuntoFkOld);
        }
        if (numeroAsuntoFkNew != null && !numeroAsuntoFkNew.equals(numeroAsuntoFkOld)) {
            numeroAsuntoFkNew.getProcuradorCollection().add(procurador);
            numeroAsuntoFkNew = em.merge(numeroAsuntoFkNew);
        }
    }
}

```

```
em.getTransaction().commit();
} catch (Exception ex) {
    String msg = ex.getMessage();
    if (msg == null || msg.length() == 0) {
        Integer id = procurador.getId();
        if (findProcurador(id) == null) {
            throw new NonexistentEntityException("The procurador with id " + id + " no longer exists.");
        }
    }
    throw ex;
} finally {
    if (em != null) {
        em.close();
    }
}
}

public void destroy(Integer id) throws NonexistentEntityException {
    EntityManager em = null;
    try {
        em = getEntityManager();
        em.getTransaction().begin();
        Procurador procurador;
        try {
            procurador = em.getReference(Procurador.class, id);
            procurador.getId();
        } catch (EntityNotFoundException enfe) {
            throw new NonexistentEntityException("The procurador with id " + id + " no longer exists.", enfe);
        }
    }
```

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

```


Asunto numeroAsuntoFk = procurador.getNumeroAsuntoFk();
if (numeroAsuntoFk != null) {
    numeroAsuntoFk.getProcuradorCollection().remove(procurador);
    numeroAsuntoFk = em.merge(numeroAsuntoFk);
}
em.remove(procurador);
em.getTransaction().commit();
} finally {
    if (em != null) {
        em.close();
    }
}
}

public List<Procurador> findProcuradorEntities() {
    return findProcuradorEntities(true, -1, -1);
}

public List<Procurador> findProcuradorEntities(int maxResults, int firstResult) {
    return findProcuradorEntities(false, maxResults, firstResult);
}

private List<Procurador> findProcuradorEntities(boolean all, int maxResults, int firstResult) {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
        cq.select(cq.from(Procurador.class));
        Query q = em.createQuery(cq);
        if (!all) {

```

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


```

        q.setMaxResults(maxResults);
        q.setFirstResult(firstResult);
    }
    return q.getResultList();
} finally {
    em.close();
}
}

public Procurador findProcurador(Integer id) {
    EntityManager em = getEntityManager();
    try {
        return em.find(Procurador.class, id);
    } finally {
        em.close();
    }
}

public int getProcuradorCount() {
    EntityManager em = getEntityManager();
    try {
        CriteriaQuery cq = em.getCriteriaBuilder().createQuery();
        Root<Procurador> rt = cq.from(Procurador.class);
        cq.select(em.getCriteriaBuilder().count(rt));
        Query q = em.createQuery(cq);
        return ((Long) q.getSingleResult()).intValue();
    } finally {
        em.close();
    }
}

```

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

```

}

public Procurador findByCedula(String cedula) {
    var procuradores = this.findProcuradorEntities();

    for (Procurador p : procuradores) {
        if (p.getCedula().equals(cedula)) {
            return p;
        }
    }

    return null;
}

public List<Procurador> findCedulaCliente(String cedula) {


    var listaEncuentro = new ArrayList<Procurador>();
    var procuradores = this.findProcuradorEntities();

    procuradores.stream().filter(p ->
(p.getNumeroAsuntoFk().getCedulaClienteFk().getCedula().equals(cedula))).forEachOrdered(p -> {
        listaEncuentro.add(p);
    });

    return listaEncuentro;
}

public List<Procurador> findNumeroAsunto(int id) {

```


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

```

var listaEncuentro = new ArrayList<Procurador>();
var procuradores = this.findProcuradorEntities();

procuradores.stream().filter(p -> (p.getNumeroAsuntoFk().getNumeroExpediente() == id)).forEachOrdered(p -> {
    listaEncuentro.add(p);
});

return listaEncuentro;
}

```

```

public List<Procurador> findListaProcuradorCedula(String cedula) {

var listaEncuentro = new ArrayList<Procurador>();
var procuradores = this.findProcuradorEntities();

procuradores.stream().filter(p -> (p.getCedula().equals(cedula))).forEachOrdered(p -> {
    listaEncuentro.add(p);
});

return listaEncuentro;
}
}

```


Paquete Vista:

VentanaPrincipal:

/*

* 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.vista;

import javax.persistence.EntityManagerFactory;

import ups.edu.ec.controlador.AsuntoJpaController;

import ups.edu.ec.controlador.ClienteJpaController;

import ups.edu.ec.controlador.ProcuradorJpaController;

import ups.edu.ec.modelo.Procurador;

/**

*

* @author User

*/

public class VentanaPrincipal extends javax.swing.JFrame {

private AsuntoJpaController controladorAsunto;

private ClienteJpaController controladorCliente;

private ProcuradorJpaController controladorProcurador;

private VentanaAsunto ventanaAsunto;

private VentanaCliente ventanaCliente;


private VentanaProcurador ventanaProcurador;

private VentanaVista ventanaVista;

private EntityManagerFactory emf;

/**

* Creates new form VentanaPrincipal

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

*/

```

public VentanaPrincipal() {
    initComponents();

    controladorAsunto = new AsuntoJpaController();
    controladorCliente = new ClienteJpaController();
    controladorProcurador = new ProcuradorJpaController();

    ventanaAsunto = new VentanaAsunto(controladorAsunto, controladorCliente, controladorProcurador);
    ventanaCliente = new VentanaCliente(controladorCliente, controladorProcurador);
    ventanaProcurador = new VentanaProcurador(controladorProcurador, controladorAsunto);
    ventanaVista = new VentanaVista(controladorProcurador);


    ventanaCliente.setVentanaAsunto(ventanaAsunto);
    ventanaAsunto.setVentanaProcurador(ventanaProcurador);

    desktopPane.add(ventanaCliente);
    desktopPane.add(ventanaAsunto);
    desktopPane.add(ventanaProcurador);
    desktopPane.add(ventanaVista);

}

public void cerrarVentanas() {
    ventanaAsunto.hide();
    ventanaCliente.hide();
    ventanaProcurador.hide();
    ventanaVista.hide();
}

```

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

```

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


    desktopPane = new javax.swing.JDesktopPane();
    menuBar = new javax.swing.JMenuBar();
    fileMenu = new javax.swing.JMenu();
    asuntoMenuItem = new javax.swing.JMenuItem();
    clienteMenuItem = new javax.swing.JMenuItem();
    procuradorMenuItem = new javax.swing.JMenuItem();
    exitMenuItem = new javax.swing.JMenuItem();
    editMenu = new javax.swing.JMenu();
    gestionProcuradorMenuItem = new javax.swing.JMenuItem();
    helpMenu = new javax.swing.JMenu();
    vistaExpedietnesMenuItem = new javax.swing.JMenuItem();

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

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

    asuntoMenuItem.setMnemonic('o');

```

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

```

asuntoMenuItem.setText("Asunto");

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

fileMenu.add(asuntoMenuItem);

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


fileMenu.add(clienteMenuItem);

procuradorMenuItem.setMnemonic('a');
procuradorMenuItem.setText("Procurador");
procuradorMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        procuradorMenuItemActionPerformed(evt);
    }
});

fileMenu.add(procuradorMenuItem);

exitMenuItem.setMnemonic('x');
exitMenuItem.setText("Exit");
exitMenuItem.addActionListener(new java.awt.event.ActionListener() {

```

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

```

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

menuBar.add(fileMenu);

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


gestionProcuradorMenuItem.setMnemonic('p');
gestionProcuradorMenuItem.setText("Procurador");
editMenu.add(gestionProcuradorMenuItem);

menuBar.add(editMenu);

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

vistaExpedietnesMenuItem.setMnemonic('c');
vistaExpedietnesMenuItem.setText("Expedientes");
vistaExpedietnesMenuItem.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        vistaExpedietnesMenuItemActionPerformed(evt);
    }
});
helpMenu.add(vistaExpedietnesMenuItem);

```

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

```

menuBar.add(helpMenu);

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, 737, Short.MAX_VALUE)
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(desktopPane, javax.swing.GroupLayout.DEFAULT_SIZE, 452, Short.MAX_VALUE)
);

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


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

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

    cerrarVentanas();
    ventanaAsunto.setVisible(true);

}

```

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

```

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

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


    cerrarVentanas();
    ventanaProcurador.setVisible(true);
}

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

    cerrarVentanas();
    ventanaVista.setVisible(true);
}

/**
 * @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.

```


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

* For details see <http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html>


```

*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {
    java.util.logging.Logger.getLogger(VentanaPrincipal.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

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

// 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 asuntoMenuItem;
private javax.swing.JMenuItem clienteMenuItem;
private javax.swing.JDesktopPane desktopPane;
private javax.swing.JMenu editMenu;
private javax.swing.JMenuItem exitMenuItem;
private javax.swing.JMenu fileMenu;
private javax.swing.JMenuItem gestionProcuradorMenuItem;
private javax.swing.JMenu helpMenu;
private javax.swing.JMenuBar menuBar;
private javax.swing.JMenuItem procuradorMenuItem;
private javax.swing.JMenuItem vistaExpedietnesMenuItem;
// End of variables declaration

}

```

VentanaAsunto:


```

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

package ups.edu.ec.vista;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;

```

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

```

import javax.swing.JOptionPane;

import ups.edu.ec.controlador.AsuntoJpaController;
import ups.edu.ec.controlador.ClienteJpaController;
import ups.edu.ec.controlador.ProcuradorJpaController;
import ups.edu.ec.controlador.exceptions.NonexistentEntityException;
import ups.edu.ec.modelo.Asunto;
import ups.edu.ec.modelo.Cliente;
import ups.edu.ec.modelo.Procurador;


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

    private Cliente cliente;
    private AsuntoJpaController controladorAsunto;
    private ClienteJpaController controladorCliente;
    private ProcuradorJpaController controladorProcurador;

    private Asunto asunto;
    private VentanaProcurador ventanaProcurador;

    /**
     * Creates new form VentanaAsunto
     */
    public VentanaAsunto(AsuntoJpaController controladorAsunto, ClienteJpaController controladorCliente,
        ProcuradorJpaController controladorProcurador) {
        initComponents();
    }

```

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

```

this.controladorAsunto = controladorAsunto;

this.controladorCliente = controladorCliente;

this.controladorProcurador = controladorProcurador;
}

public Cliente getCliente() {
    return cliente;
}

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


public VentanaProcurador getVentanaProcurador() {
    return ventanaProcurador;
}

public void setVentanaProcurador(VentanaProcurador ventanaProcurador) {
    this.ventanaProcurador = ventanaProcurador;
}

public void limpiar() {

    txtNumeroExpediente.setText("");
    txtEstado.setText("");
    txtCedula.setText("");
    dtChooserFechaInicio.setDate(new Date());

```

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

```

btnActualizar.setEnabled(false);

btnEliminar.setEnabled(false);

btnSiguiente.setEnabled(false);


txtNumeroExpediente.setEnabled(true);

}

/**
 * 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() {

    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();
    txtNumeroExpediente = new javax.swing.JTextField();
    dtChooserFechalInicio = new com.toedter.calendar.JDateChooser();
    dtChooserFechaFin = new com.toedter.calendar.JDateChooser();
    txtEstado = new javax.swing.JTextField();
    txtCedula = new javax.swing.JFormattedTextField();
    btnCrear = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    btnSiguiente = 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		

```

btnBuscar = new javax.swing.JButton();

btnActualizar = new javax.swing.JButton();

btnEliminar = new javax.swing.JButton();


setClosable(true);

setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);

setTitle("Asunto");

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("Numero expediente");

jLabel2.setText("Fecha inicio");

```

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

```

jLabel3.setText("Fecha fin");

jLabel4.setText("Estado");

jLabel5.setText("Cedula cliente");


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

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

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

btnSiguiente.setText("Siguiente");

```

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

```

btnSiguiente.setEnabled(false);


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

btnBuscar.setText("Buscar");
btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(evt);
    }
});

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

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


```


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

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(41, 41, 41)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addComponent(jLabel2)
                        .addComponent(jLabel3)
                        .addComponent(jLabel4)
                        .addComponent(jLabel5))
                    .addGap(58, 58, 58)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                        .addComponent(dtChooserFechaInicio, javax.swing.GroupLayout.DEFAULT_SIZE, 184, Short.MAX_VALUE)
                        .addComponent(dtChooserFechaFin, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(txtNumeroExpediente)
                        .addComponent(txtEstado)
                        .addComponent(txtCedula))
                    .addGap(18, 18, 18)
                    .addComponent(btnBuscar))
                .addGroup(layout.createSequentialGroup()
                    .addGap(26, 26, 26)
                    .addComponent(btnCrear)
                    .addGap(29, 29, 29)
            )
        )
    )

```

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

```

.addComponent(btnCancelar)

.addGap(34, 34, 34)

.addComponent(btnActualizar)

.addGap(26, 26, 26)

.addComponent(btnEliminar)

.addGap(32, 32, 32)

.addComponent(btnSiguiente)))

.addContainerGap(11, Short.MAX_VALUE))

);

layout.setVerticalGroup(

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

        .addGroup(layout.createSequentialGroup())

            .addGap(27, 27, 27)

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

                .addGroup(layout.createSequentialGroup())

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

                        .addGroup(layout.createSequentialGroup())

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

                                .addComponent(jLabel1)

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

                                .addComponent(btnBuscar))

                            .addGap(18, 18, 18)

                            .addComponent(jLabel2))


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

                    .addGap(18, 18, 18)

                    .addComponent(jLabel3))

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

```

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

```

.addGap(18, 18, 18)

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

    .addComponent(jLabel4)

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

.addGap(18, 18, 18)

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

    .addComponent(jLabel5)

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

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 30, Short.MAX_VALUE)

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

    .addComponent(btnCrear)

    .addComponent(btnCancelar)

    .addComponent(btnActualizar)

    .addComponent(btnEliminar)

    .addComponent(btnSiguiente))

.addGap(31, 31, 31))

);

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

```

```

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

    // TODO add your handling code here:


    int numeroExpediente = Integer.valueOf(txtNumeroExpediente.getText());

    Date fechaInicio = dtChooserFechaInicio.getDate();

    Date fechaFin = dtChooserFechaFin.getDate();

    String estado = txtEstado.getText();

```

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

```
String cedulaCliente = txtCedula.getText();


if (numeroExpediente == 0 || fechalnicio.getTime() == 0 || estado.isBlank() || cedulaCliente.isBlank()) {
    JOptionPane.showMessageDialog(this, "Llene todos los campos", "Error", JOptionPane.ERROR_MESSAGE);
} else {

    cliente = controladorCliente.findCliente(cedulaCliente);
    if (cliente != null) {
        asunto = new Asunto(numeroExpediente, fechalnicio, fechaFin, estado);
        asunto.setCedulaClienteFk(cliente);

        try {
            controladorAsunto.create(asunto);
            /*
            int opcion = JOptionPane.showConfirmDialog(this, "Asunto creado con exito! ¿Desea escoger un Procurador",
                "Información", JOptionPane.INFORMATION_MESSAGE);*/

            JOptionPane.showMessageDialog(this, "Asunto creado", "Informacion", JOptionPane.INFORMATION_MESSAGE);
            btnSiguiente.setEnabled(true);

        } catch (Exception ex) {
            Logger.getLogger(VentanaAsunto.class.getName()).log(Level.SEVERE, null, ex);
            JOptionPane.showMessageDialog(this, "El asunto ya existe", "Error", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Cliente no encontrado", "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 formInternalFrameActivated(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:

    if (cliente != null) {
        txtCedula.setText(cliente.getCedula());
    }
}


private void formInternalFrameOpened(javax.swing.event.InternalFrameEvent evt) {
    // TODO add your handling code here:
    if (cliente != null) {
        txtCedula.setText(cliente.getCedula());
    }
}

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

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

    ventanaProcurador.setAsunto(asunto);

```

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

```

ventanaProcurador.setVisible(true);

limpiar();

this.hide();

}

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

    // TODO add your handling code here:


    int numeroExpediente = Integer.valueOf(txtNumeroExpediente.getText());

    asunto = controladorAsunto.findAsunto(numeroExpediente);

    if (asunto != null) {
        txtEstado.setText(asunto.getEstado());
        dtChooserFechaInicio.setDate(asunto.getFechaInicio());
        try {
            dtChooserFechaFin.setDate(asunto.getFechaFin());
        } catch (Exception e) {
            e.printStackTrace();
        }
        txtCedula.setText(asunto.getCedulaClienteFk().getCedula());

        btnActualizar.setEnabled(true);
        btnEliminar.setEnabled(true);
        btnSiguiente.setEnabled(true);
        txtNumeroExpediente.setEnabled(false);
    } else {
        JOptionPane.showMessageDialog(this, "Asunto no encontrado", "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 numeroExpediente = Integer.valueOf(txtNumeroExpediente.getText());
```

```
    Date fechaInicio = dtChooserFechaInicio.getDate();
```

```
    Date fechaFin = dtChooserFechaFin.getDate();
```

```
    String estado = txtEstado.getText();
```

```
    String cedulaCliente = txtCedula.getText();
```

```
    if (numeroExpediente == 0 || fechaInicio.getTime() == 0 || estado.isBlank() || cedulaCliente.isBlank()) {
```

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

```
    } else {
```

```
        cliente = controladorCliente.findCliente(cedulaCliente);
```

```
        if (cliente != null) {
```

```
            asunto = new Asunto(numeroExpediente, fechaInicio, fechaFin, estado);
```


```
            asunto.setCedulaClienteFk(cliente);
```

```
            var listaProcuradoresAuxiliar = controladorProcurador.findNumeroAsunto(numeroExpediente);
```

```
            asunto.setProcuradorCollection(listaProcuradoresAuxiliar);
```

```
            try {
```

```
                controladorAsunto.edit(asunto);
```

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

```

/*
    int opcion = JOptionPane.showConfirmDialog(this, "Asunto creado con exito! ¿Desea escoger un Procurador",
        "Información", JOptionPane.INFORMATION_MESSAGE);*/

    JOptionPane.showMessageDialog(this, "Asunto actualizado", "Informacion",
JOptionPane.INFORMATION_MESSAGE);

    } catch (Exception ex) {
        Logger.getLogger(VentanaAsunto.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(this, "El asunto ya existe", "Error", JOptionPane.ERROR_MESSAGE);
    }
} else {
    JOptionPane.showMessageDialog(this, "Cliente no encontrado", "Error", JOptionPane.ERROR_MESSAGE);
}

}

}

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


    int numeroExpediente = Integer.valueOf(txtNumeroExpediente.getText());

    try {

        var as = controladorAsunto.findAsunto(numeroExpediente);

        var procus = controladorProcurador.findNumeroAsunto(numeroExpediente);

```


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

```
as.setProcuradorCollection(procus);
```

```
controladorAsunto.edit(as);
```

```
controladorAsunto.destroy(numeroExpediente);
```

```
JOptionPane.showMessageDialog(this, "Asunto eliminado", "Informacion", JOptionPane.INFORMATION_MESSAGE);
limpiar();
```

```
} catch (NonexistentEntityException ex) {
    Logger.getLogger(VentanaAsunto.class.getName()).log(Level.SEVERE, null, ex);
} catch (Exception ex) {
    Logger.getLogger(VentanaAsunto.class.getName()).log(Level.SEVERE, null, ex);
}
```

```
}
```

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

```
private javax.swing.JButton btnActualizar;
```

```
private javax.swing.JButton btnBuscar;
```

```
private javax.swing.JButton btnCancelar;
```

```
private javax.swing.JButton btnCrear;
```


```
private javax.swing.JButton btnEliminar;
```

```
private javax.swing.JButton btnSiguiente;
```

```
private com.toedter.calendar.JDateChooser dtChooserFechaFin;
```

```
private com.toedter.calendar.JDateChooser dtChooserFechaInicio;
```

```
private javax.swing.JLabel jLabel1;
```

	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 jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JFormattedTextField txtCedula;
private javax.swing.JTextField txtEstado;
private javax.swing.JTextField txtNumeroExpediente;

// End of variables declaration
}

```

VentanaCliente:


```

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

package ups.edu.ec.vista;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import ups.edu.ec.controlador.ClienteJpaController;
import ups.edu.ec.controlador.ProcuradorJpaController;
import ups.edu.ec.controlador.exceptions.IllegalOrphanException;
import ups.edu.ec.controlador.exceptions.NonexistentEntityException;

```

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

```

import ups.edu.ec.modelo.Asunto;

import ups.edu.ec.modelo.Cliente;

import ups.edu.ec.modelo.Procurador;

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

    private ClienteJpaController controladorCliente;
    private ProcuradorJpaController controladorProcurador;

    private VentanaAsunto ventanaAsunto;


    private Cliente cliente;

    /**
     * Creates new form Cliente
     */
    public VentanaCliente(ClienteJpaController controladorCliente, ProcuradorJpaController controladorProcurador) {
        initComponents();

        this.controladorProcurador = controladorProcurador;
        this.controladorCliente = controladorCliente;
    }

    public VentanaAsunto getVentanaAsunto() {

```

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

```

return ventanaAsunto;
}

public void setVentanaAsunto(VentanaAsunto ventanaAsunto) {
    this.ventanaAsunto = ventanaAsunto;
}


public void limpiar() {
    txtCedula.setText("");
    txtNombre.setText("");
    txtApellido.setText("");
    txtDireccion.setText("");
    txtTelefono.setText("");

    btnSiguiente.setEnabled(false);
    btnActualizar.setEnabled(false);
    txtCedula.setEnabled(true);
    btnEliminar.setEnabled(false);
}

/**
 * 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() {

```

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

```

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();
jLabel6 = new javax.swing.JLabel();
txtNombre = new javax.swing.JTextField();
txtApellido = new javax.swing.JTextField();
txtDireccion = new javax.swing.JTextField();
txtTelefono = new javax.swing.JTextField();
jDateChooser1 = new com.toedter.calendar.JDateChooser();
btnCrear = new javax.swing.JButton();
btnCancelar = new javax.swing.JButton();
txtCedula = new javax.swing.JFormattedTextField();
btnBuscar = new javax.swing.JButton();
btnSiguiente = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminar = new javax.swing.JButton();


setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Cliente");

jLabel1.setText("Cedula");

jLabel2.setText("Nombre");

jLabel3.setText("Apellido");

```

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

```
jLabel4.setText("Direccion");
```

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

```
jLabel6.setText("Fecha de nacimiento");
```

```
btnCrear.setText("Crear");
```

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

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

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


```
try {
```

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

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

```
btnBuscar.setText("Buscar");
```

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

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

```


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

btnSiguiente.setText("Siguiente");
btnSiguiente.setEnabled(false);
btnSiguiente.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSiguienteActionPerformed(evt);
    }
});

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

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


```

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

```

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(26, 26, 26)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addComponent(btnCrear)
            .addComponent(btnCancelar))
        .addGap(34, 34, 34)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jLabel4)
            .addComponent(jLabel5)
            .addComponent(jLabel6)))
    .addGroup(layout.createSequentialGroup()
        .addGap(44, 44, 44)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(txtNombre)
            .addComponent(txtApellido)
            .addComponent(txtDireccion)
            .addComponent(txtTelefono)
            .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
            .addComponent(txtCedula))
        .addGap(43, 43, 43)

```


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

```

.addComponent(btnBuscar)

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

.addGroup(layout.createSequentialGroup())

.addGap(33, 33, 33)

.addComponent(btnActualizar)

.addGap(37, 37, 37)

.addComponent(btnEliminar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 35, Short.MAX_VALUE)

.addComponent(btnSiguiente)

.addGap(36, 36, 36))))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

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

.addComponent(jLabel1)

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

.addComponent(btnBuscar))

.addGap(18, 18, 18)

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

.addComponent(jLabel2)

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


.addGap(18, 18, 18)

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

.addComponent(jLabel3)

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

```

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

```

.addGap(18, 18, 18)

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

    .addComponent(jLabel4)

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

.addGap(18, 18, 18)

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

    .addComponent(jLabel5)

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

.addGap(18, 18, 18)

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

    .addComponent(jLabel6)

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

.addGap(27, 27, 27)

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

    .addComponent(btnSiguiete)

    .addComponent(btnCancelar)

    .addComponent(btnCrear)

    .addComponent(btnActualizar)

    .addComponent(btnEliminar))

.addContainerGap(31, 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 cedula = txtCedula.getText();
String nombre = txtNombre.getText();
String apellido = txtApellido.getText();
String direccion = txtDireccion.getText();
String telefono = txtTelefono.getText();

Date fechaNacimiento = jDateChooser1.getDate();

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


    cliente = new Cliente(cedula, nombre, apellido, telefono, fechaNacimiento);

    try {
        controladorCliente.create(cliente);

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

        btnSiguiente.setEnabled(true);
    } catch (Exception ex) {
        Logger.getLogger(VentanaCliente.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(this, "El cliente ya existe", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

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

```
}
```

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

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

```
limpiar();
```

```
}
```

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

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

```
String cedula = txtCedula.getText();
```

```
cliente = controladorCliente.findCliente(cedula);
```

```
if (cliente != null) {
```

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

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

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

```
txtTelefono.setText(cliente.getTelefono());
```

```
jDateChooser1.setDate(cliente.getFechaNacimiento());
```

```
btnSiguiente.setEnabled(true);
```


```
btnActualizar.setEnabled(true);
```

```
txtCedula.setEnabled(false);
```

```
btnEliminar.setEnabled(true);
```

```
} else {
```

```
JOptionPane.showMessageDialog(this, "El cliente no existe", "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 btnSiguienteActionPerformed(java.awt.event.ActionEvent evt) {
```

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

```
    ventanaAsunto.setCliente(cliente);
```

```
    ventanaAsunto.setVisible(true);
```

```
    limpiar();
```

```
    this.hide();
```

```
}
```

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

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

```
    String cedula = txtCedula.getText();
```

```
    String nombre = txtNombre.getText();
```

```
    String apellido = txtApellido.getText();
```

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

```
    String telefono = txtTelefono.getText();
```


```
    Date fechaNacimiento = jDateChooser1.getDate();
```

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

```
        || telefono.isBlank()) {
```

```
        JOptionPane.showMessageDialog(this, "Llene todos los campos", "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		

```

cliente = new Cliente(cedula, nombre, apellido, telefono, fechaNacimiento);

try {

    var listaProcurador = controladorProcurador.findCedulaCliente(cedula);
    List<Asunto> asuntos = new ArrayList<>();

    listaProcurador.stream().map(p -> p.getNumeroAsuntoFk()).map(asunto -> {
        var listaProcusAsunto = controladorProcurador.findNumeroAsunto(asunto.getNumeroExpediente());
        asunto.setProcuradorCollection(listaProcusAsunto);
        return asunto;
    }).forEachOrdered(asunto -> {
        asuntos.add(asunto);
    });


    cliente.setAsuntoCollection(asuntos);

    controladorCliente.edit(cliente);

    JOptionPane.showMessageDialog(this, "Cliente actualizado con exito", "Informacion",
    JOptionPane.INFORMATION_MESSAGE);

    btnSiguiente.setEnabled(true);
} catch (Exception ex) {
    Logger.getLogger(VentanaCliente.class.getName()).log(Level.SEVERE, null, ex);
    JOptionPane.showMessageDialog(this, "El cliente ya existe", "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 btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    String cedula = txtCedula.getText();


    try {
        controladorCliente.destroy(cedula);

        JOptionPane.showMessageDialog(this, "Cliente eliminado con exito", "Informacion",
JOptionPane.INFORMATION_MESSAGE);

        limpiar();
    } catch (IllegalOrphanException | NonexistentEntityException ex) {
        Logger.getLogger(VentanaCliente.class.getName()).log(Level.SEVERE, null, ex);
    }
}

// Variables declaration - do not modify
private javax.swing.JButton btnActualizar;
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnEliminar;
private javax.swing.JButton btnSiguiente;

```

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

```

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.JTextField txtApellido;
private javax.swing.JFormattedTextField txtCedula;
private javax.swing.JTextField txtDireccion;
private javax.swing.JTextField txtNombre;
private javax.swing.JTextField txtTelefono;

// End of variables declaration
}

```

VentanaProcurador:


```

/*
 * 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.System.Logger;
import java.lang.System.Logger.Level;
import ups.edu.ec.modelo.Asunto;
import java.util.*;
import javax.swing.JOptionPane;

```


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

```

import ups.edu.ec.controlador.AsuntoJpaController;

import ups.edu.ec.controlador.ProcuradorJpaController;

import ups.edu.ec.modelo.Cliente;

import ups.edu.ec.modelo.Procurador;

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

    private Asunto asunto;

    private Procurador procurador;

    private ProcuradorJpaController controladorProcurador;

    private AsuntoJpaController controladorAsunto;


    /**
     * Creates new form VentanaProcurador
     */
    public VentanaProcurador(ProcuradorJpaController controladorProcurador, AsuntoJpaController controladorAsunto) {
        initComponents();

        this.controladorProcurador = controladorProcurador;

        this.controladorAsunto = controladorAsunto;
    }

    public Asunto getAsunto() {
        return asunto;
    }

```

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

```

}

public void setAsunto(Asunto asunto) {
    this.asunto = asunto;
}


public void limpiar() {

}

/**
 * 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() {

    jLabel5 = new javax.swing.JLabel();
    btnCrear = new javax.swing.JButton();
    jLabel6 = new javax.swing.JLabel();
    btnCancelar = new javax.swing.JButton();
    txtCedula = new javax.swing.JFormattedTextField();
    txtNombre = new javax.swing.JTextField();
    txtApellido = new javax.swing.JTextField();
    txtDireccion = new javax.swing.JTextField();
    txtTelefono = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();

```


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

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
btnBuscar = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jDateChooser1 = new com.toedter.calendar.JDateChooser();
btnFin = new javax.swing.JButton();
jLabel7 = new javax.swing.JLabel();
txtNumeroExpediente = new javax.swing.JTextField();

setClosable(true);
setDefaultCloseOperation(javax.swing.WindowConstants.HIDE_ON_CLOSE);
setTitle("Procurador");
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);
    }
}

```

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

```
});
```

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

```
btnCrear.setText("Crear");
```

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

```
jLabel6.setText("Fecha de nacimiento");
```

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


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

```
try {
```

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

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


```
jLabel1.setText("Cedula");
```


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

```

.addGroup(layout.createSequentialGroup())
    .addGap(50, 50, 50)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel1)
        .addComponent(jLabel2)
        .addComponent(jLabel3)
        .addComponent(jLabel4)
        .addComponent(jLabel5)
        .addComponent(jLabel6)
        .addComponent(jLabel7))
    .addGap(44, 44, 44))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
    .addContainerGap()
    .addComponent(btnCrear)
    .addGap(34, 34, 34)))
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(btnCancelar)
        .addGap(29, 29, 29)
        .addComponent(btnFin))
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(txtNumeroExpediente)
            .addComponent(txtNombre)
            .addComponent(txtApellido)
            .addComponent(txtDireccion)
            .addComponent(txtTelefono)
            .addComponent(jDateChooser1, javax.swing.GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
            .addComponent(txtCedula))

```

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

```

.addGap(43, 43, 43)

.addComponent(btnBuscar)))

.addContainerGap(21, Short.MAX_VALUE))

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addGap(28, 28, 28)

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

.addComponent(jLabel1)

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

.addComponent(btnBuscar))

.addGap(18, 18, 18)

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

.addComponent(jLabel2)

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

.addGap(18, 18, 18)

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

.addComponent(jLabel3)

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

.addGap(18, 18, 18)

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


.addComponent(jLabel4)

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

.addGap(18, 18, 18)

.addGroup(layout.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(jLabel5)

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

        .addGap(18, 18, 18)

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

            .addComponent(jLabel6)

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

        .addGap(19, 19, 19)

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

            .addComponent(jLabel7)

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

        .addGap(21, 21, 21)

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

            .addComponent(btnFin)

            .addComponent(btnCancelar)

            .addComponent(btnCrear))

        .addContainerGap(17, Short.MAX_VALUE))

    );

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

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


    // TODO add your handling code here:

    String cedula = txtCedula.getText();

    String nombre = txtNombre.getText();

    String apellido = txtApellido.getText();

```


	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();

String telefono = txtTelefono.getText();


Date fechaNacimiento = jDateChooser1.getDate();


int numeroExpediente = Integer.valueOf(txtNumeroExpediente.getText());


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

    asunto = controladorAsunto.findAsunto(numeroExpediente);


    if (asunto != null) {
        procurador = new Procurador(controladorProcurador.getProcuradorCount(), nombre, apellido, telefono,
fechaNacimiento);



        procurador.setCedula(cedula);
        procurador.setNumeroAsuntoFk(asunto);
        procurador.setDireccion(direccion);


        try {
            controladorProcurador.create(procurador);


            JOptionPane.showMessageDialog(this, "Procurador creado con exito", "Informacion",
JOptionPane.INFORMATION_MESSAGE);


            btnFin.setEnabled(true);

```

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

```

    } catch (Exception ex) {
        //Logger.getLogger(VentanaCliente.class.getName()).log(Level.SEVERE, null, ex);
        JOptionPane.showMessageDialog(this, "El procurador ya existe", "Error", JOptionPane.ERROR_MESSAGE);
    }
}

}

}

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

    limpiar();
}


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

    String cedula = txtCedula.getText();

    procurador = controladorProcurador.findByCedula(cedula);

    if (procurador != null) {
        txtNombre.setText(procurador.getNombre());
        txtApellido.setText(procurador.getApellido());
        txtDireccion.setText(procurador.getDireccion());
        txtTelefono.setText(procurador.getTelefono());
        jDateChooser1.setDate(procurador.getFechaNacimiento());
        btnFin.setEnabled(true);
    }
}

```

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

```

//btnActualizar.setEnabled(true);

txtCedula.setEnabled(false);

//btnEliminar.setEnabled(true);

} else {

    JOptionPane.showMessageDialog(this, "El cliente no existe", "Error", JOptionPane.ERROR_MESSAGE);

}

}

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

    // TODO add your handling code here:

    JOptionPane.showMessageDialog(this, "Asunto creado con exito", "Informacion",
JOptionPane.INFORMATION_MESSAGE);

    limpiar();

    this.hide();

}

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

    // TODO add your handling code here:

    if (asunto != null) {

        txtNumeroExpediente.setText(asunto.getNumeroExpediente() + "");

    }

}

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

    // TODO add your handling code here:


    if (asunto != null) {

        txtNumeroExpediente.setText(asunto.getNumeroExpediente() + "");

    }

}

```

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

```


}

}

// Variables declaration - do not modify
private javax.swing.JButton btnBuscar;
private javax.swing.JButton btnCancelar;
private javax.swing.JButton btnCrear;
private javax.swing.JButton btnFin;
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JTextField txtApellido;
private javax.swing.JFormattedTextField txtCedula;
private javax.swing.JTextField txtDireccion;
private javax.swing.JTextField txtNombre;
private javax.swing.JTextField txtNumeroExpediente;
private javax.swing.JTextField txtTelefono;
// End of variables declaration
}

```

VentanaVista:

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

```

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

```

```

package ups.edu.ec.vista;

import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import ups.edu.ec.controlador.ProcuradorJpaController;
import ups.edu.ec.modelo.Procurador;

```

```


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

    private ProcuradorJpaController controladorProcurador;

    /**
     * Creates new form VentanaVista
     */
    public VentanaVista(ProcuradorJpaController controladorProcurador) {
        initComponents();

        this.controladorProcurador = controladorProcurador;
    }

```

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

```

llenarTablaInicio();
}

public void llenarTablaInicio() {

    var procuradores = controladorProcurador.findProcuradorEntities();

    var modelo = (DefaultTableModel) tblInformacion.getModel();

    modelo.setRowCount(0);

    for (Procurador p : procuradores) {

        if (p.getNumeroAsuntoFk() != null) {

            Object[] row = {p.getNumeroAsuntoFk().getCedulaClienteFk().getCedula(),
p.getNumeroAsuntoFk().getCedulaClienteFk().getNombre().concat(p.getNumeroAsuntoFk().getCedulaClienteFk().getApellido()),
            p.getNumeroAsuntoFk().getNumeroExpediente(),
            p.getNumeroAsuntoFk().getFechaInicio().toString().concat(p.getNumeroAsuntoFk().getFechaFin().toString()),
            p.getNumeroAsuntoFk().getEstado(), p.getCedula(), p.getNombre().concat(p.getApellido())
            };

            modelo.addRow(row);


        }

    }

    tblInformacion.setModel(modelo);

}

```

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

```

public void llenarTabla(List<Procurador> procuradores) {

    var modelo = (DefaultTableModel) tblInformacion.getModel();

    modelo.setRowCount(0);

    for (Procurador p : procuradores) {
        Object[] row = {p.getNumeroAsuntoFk().getCedulaClienteFk().getCedula(),
p.getNumeroAsuntoFk().getCedulaClienteFk().getNombre().concat(p.getNumeroAsuntoFk().getCedulaClienteFk().getApellido()),
        p.getNumeroAsuntoFk().getNumeroExpediente(),
        p.getNumeroAsuntoFk().getFechaInicio().toString().concat(p.getNumeroAsuntoFk().getFechaFin().toString()),
        p.getNumeroAsuntoFk().getEstado(), p.getCedula(), p.getNombre().concat(p.getApellido())
        };

        modelo.addRow(row);
    }


    tblInformacion.setModel(modelo);

}

public void limpiar() {
    llenarTablaInicio();
    txtBusqueda.setText("");
    cbxOpcion.setSelectedIndex(0);

}

```

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


```

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

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    cbxOpcion = new javax.swing.JComboBox<>();
    txtBusqueda = new javax.swing.JTextField();
    btnBuscar = new javax.swing.JButton();
    btnCancelar = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    tblInformacion = new javax.swing.JTable();

    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) {

```


	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: 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("Buscar por:");


cbxOpcion.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "--Seleccione--", "Cedula cliente", "Numero
expediente", "Cedula procurador" }));

btnBuscar.setText("Buscar");
btnBuscar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnBuscarActionPerformed(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);


```

	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()
            .addGap(141, 141, 141)
            .addComponent(jLabel1)
            .addGap(32, 32, 32)
            .addComponent(cbxOpcion, javax.swing.GroupLayout.PREFERRED_SIZE, 144,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(41, 41, 41)
            .addComponent(txtBusqueda, javax.swing.GroupLayout.PREFERRED_SIZE, 228,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(55, 55, 55)
            .addComponent(btnBuscar)
            .addGap(45, 45, 45)
            .addComponent(btnCancelar)
            .addContainerGap(125, Short.MAX_VALUE))
        );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(26, 26, 26)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(cbxOpcion, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(txtBusqueda, javax.swing.GroupLayout.PREFERRED_SIZE,
                    javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(btnBuscar)
                .addComponent(btnCancelar))
            .addContainerGap(23, 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		

```

);


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

    },
    new String [] {
        "Cedula cliente", "Nombres", "Asunto #", "Periodo", "Estado", "Cedula procurador", "Nombres"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.String.class, java.lang.Integer.class, java.lang.String.class, java.lang.String.class,
        java.lang.String.class, java.lang.String.class
    };
    boolean[] canEdit = new boolean [] {
        false, false, false, false, false, false, false
    };
    public Class getColumnClass(int columnIndex) {
        return types [columnIndex];
    }
    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});

jScrollPane1.setViewportViewView(tblInformacion);

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

```

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

```

getContentPane().setLayout(layout);

layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 600, true)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, 300, true)
                    .addGap(10, 10, 10)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, 300, true)
                )
            )
        )
);

layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 400, true)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, 150, true)
                    .addGap(10, 10, 10)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, 150, true)
                )
            )
        )
);


pack();
}

// </editor-fold>

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

    String opcionCBX = (String) cbxOpcion.getSelectedItem();

```

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

```

if (opcionCBX.equals("--Seleccione--") || txtBusqueda.getText().isBlank()) {
    JOptionPane.showMessageDialog(this, "Seleccione parametro busqueda", "Error", JOptionPane.ERROR_MESSAGE);
} else if (opcionCBX.equals("Cedula cliente")) {

    String cedula = txtBusqueda.getText();

    var resultado = controladorProcurador.findCedulaCliente(cedula);

    if (resultado.isEmpty()) {
        JOptionPane.showMessageDialog(this, "No se han encontrado registros", "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        llenarTabla(resultado);
    }

} else if (opcionCBX.equals("Numero expediente")) {


    int idAsunto = Integer.valueOf(txtBusqueda.getText());

    var resultado = controladorProcurador.findNumeroAsunto(idAsunto);

    if (resultado.isEmpty()) {
        JOptionPane.showMessageDialog(this, "No se han encontrado registros", "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        llenarTabla(resultado);
    }

} else if (opcionCBX.equals("Cedula procurador")) {

```

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

```
String cedula = txtBusqueda.getText();
```

```
var resultado = controladorProcurador.findListaProcuradorCedula(cedula);
```

```
if (resultado.isEmpty()) {
```

```
    JOptionPane.showMessageDialog(this, "No se han encontrado registros", "Error", JOptionPane.ERROR_MESSAGE);
```

```
} else {
```

```
    llenarTabla(resultado);
```

```
}
```

```
}
```

```
}
```

```
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:
```

```
    llenarTablaInicio();
```

```
}
```

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

```
private javax.swing.JButton btnBuscar;
```

	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.JButton btnCancelar;

private javax.swing.JComboBox<String> cbxOpcion;

private javax.swing.JLabel jLabel1;

private javax.swing.JPanel jPanel1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable tblInformacion;

private javax.swing.JTextField txtBusqueda;

// End of variables declaration

}
```

RESULTADO(S) OBTENIDO(S):

- Emplear bases de datos a una práctica.
- Practicar comandos de bases de datos.
- Interactuar con bases de datos.

CONCLUSIONES:

En conclusión, esta practica ha sido de mucha ayuda para afianzar los conocimientos sobre bases de datos en un programa de java. Cabe recalcar que mi persona ya tenia conocimientos previos sobre JPA por lo que esta práctica utilicé dichos conocimientos.

RECOMENDACIONES:

No existe ninguna recomendación.

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

