

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

		<b>FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES</b>	
<b>CARRERA:</b>		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>		<b>TÍTULO PRÁCTICA:</b> Trabajo_en_Clases	
<b>OBJETIVO ALCANZADO:</b>			
<b>ACTIVIDADES DESARROLLADAS</b>			
<p><b>1. Se pidió desarrollar una aplicación que utilice lenguaje de programación JDBC y que este conectado a la base de datos PostgreSQL.</b>  Se desarrollo los siguientes paquetes para la realización de esta práctica</p> <ul style="list-style-type: none"> <li>• Ec.edu.ups.modelo</li> <li>• Ec.edu.ups.controlador</li> <li>• Ec.edu.ups.vista</li> </ul>			
<p><b>2. Ec.edu.ups.modelo</b>  <b>En este paquete se crearon todos los objetos que se utilizaran para simular un programa de consultas entre clientes y abogados.</b></p> <ul style="list-style-type: none"> <li>• Asunto</li> <li>• Cliente</li> <li>• Procurador</li> </ul>			
<p><b>3. Asunto.java</b>  Esta clase se utiliza para distinguir el tipo de caso que se encontrara en la base de datos. Esta clase tendrá las cedulas de ambos los procuradores y de los clientes.  Atributos:</p> <ul style="list-style-type: none"> <li>• private int numeroCaso;</li> <li>• private Date FechaDelInicio;</li> <li>• private Date fechaFinal;</li> <li>• private String estado;</li> <li>• private String cedulaCliente;</li> <li>• private String cedulaProcurador;</li> </ul> <p>De la misma manera se han creado dos constructores, el primero constructor se ha creado sin ningún parámetro. El segundo constructor tiene de parámetro los atributos de esta clase y de la misma manera inicializa todos los atributos. Por ultimo se ha creado los métodos getter y setters para cada atributo.</p>			
<p><b>4. Cliente.java</b>  Esta clase se utiliza para guardar los datos de un cliente dentro de la base de datos.</p>			

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

Atributos:

- private String cedula;
- private String nombre;
- private String apellido;
- private Date fechaDeNacimiento;
- private String direccion;
- private String telefono;

De la misma manera se han creado dos constructores, el primero constructor se ha creado sin ningún parámetro. El segundo constructor tiene de parámetro los atributos de esta clase y de la misma manera inicializa todos los atributos. Por ultimo se ha creado los métodos getter y setters para cada atributo.

## 5. Procurador.java

Esta clase se ha creado con la intención de guardar los datos del procurador dentro de la base de datos.

Atributos:

- private String cedula;
- private String nombre;
- private String apellido;
- private Date fechaDeNacimiento;
- private String direccion;
- private String telefono;

De la misma manera se han creado dos constructores, el primero constructor se ha creado sin ningún parámetro. El segundo constructor tiene de parámetro los atributos de esta clase y de la misma manera inicializa todos los atributos. Por ultimo se ha creado los métodos getter y setters para cada atributo.

## 6. Ec.edu.ups.controlador

Este paquete contiene todas las clases que interactuaran con la interfaz del usuario y a la misma vez las consultas con la base de datos postgresQL

Clases:

- controladorAsunto
- ControladorBD
- ControladorPersona
- ControladorProcurador
- ControladorRegex

## 7. ControladorAsunto.java


Esta clase se utiliza para crear y actualizar casos de un abogado y también para poder listar todos los asuntos que pueda tener un abogado.

Atributos:

- private Asunto asunto;

se ha creado un solo controlador que no recibe ningún objeto dentro de los parametros.

Métodos:

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

- **createAsunto:** Este método recibe en su parámetro todos los atributos de la clase Asunto. Este método se utiliza para crear un asunto dentro de la base de datos y se utiliza lenguaje de programación JDBC para la conexión a la base de datos
- **updateAsunto:** Este método recibe en su parámetro todos los atributos de la clase Asunto. Este método se utiliza para actualizar un asunto dentro de la base de datos y se utiliza lenguaje de programación JDBC para la conexión a la base de datos
- **findAll:** Este método se utiliza para devolver un listado de asuntos que se encuentre dentro de la base de datos.

#### 8. **ControladorBD**

Esta clase se utiliza para realizar la conexión a la base de datos y la desconexión.

Métodos:

- **Conexion:** Este método realiza la conexión a la base de datos postgresSQL.
- **Disconnect:** Este método realiza la desconexión a la base de datos postgresSQL

#### 9. **ControladorPersona**

Este controlador se utiliza en su mayoría para realizar las consultas de un cliente dentro del sistema.

Atributos:

- `private Cliente cliente;`

Esta clase solamente tiene creado un controlador sin ningún atributo en su parámetro.

Métodos:

- **createCliente:** Este método recibe todos los atributos de un cliente dentro de su parámetro. Este método cumple con la tarea de crear el objeto dentro de la base de datos y almacenarla.
- **readCliente:** Este método recibe dentro de su parámetro un objeto de tipo String. Este método realiza una consulta a la base de datos y devuelve un cliente que tenga la misma cedula. En caso de no encontrar ningún cliente con la cedula devolverá un null.
- **updateCliente:** Este método recibe todos los atributos de un cliente dentro de su parámetro. Este método cumple con la tarea de actualizar un objeto dentro de la base de datos y almacenarla.
- **deleteCliente:** Este método recibe dentro de su parámetro un objeto de tipo String. Este método realiza la tarea de eliminar un objeto dentro de la base de datos.
- **findAll:** Este método solamente realiza la tarea de devolver un listado de clientes que se encuentre dentro de la base de datos.

#### 10. **controladorProcurador**

Este controlador se utiliza en su mayoría para realizar las consultas de un procurador dentro del sistema.


Atributos:

- `private Procurador procurador;`

Esta clase solamente tiene creado un controlador sin ningún atributo en su parámetro.

Métodos:

- **createProcurador:** Este método recibe todos los atributos de un procurador dentro de su parámetro. Este método cumple con la tarea de crear el objeto dentro de la base de datos y almacenarla.
- **readProcurador:** Este método recibe dentro de su parámetro un objeto de tipo String. Este método realiza una consulta a la base de datos y devuelve un procurador que tenga la misma cedula. En caso de no encontrar ningún procurador con la cedula devolverá un null.

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

- **updateProcurador:** Este método recibe todos los atributos de un procurador dentro de su parámetro. Este método cumple con la tarea de actualizar un objeto dentro de la base de datos y almacenarla.
- **deleteProcurador:** Este método recibe dentro de su parámetro un objeto de tipo String. Este método realiza la tarea de eliminar un objeto dentro de la base de datos.
- **findAll:** Este método solamente realiza la tarea de devolver un listado de procuradores que se encuentre dentro de la base de datos.

#### 11. **controladorRegex.java**

Este controlador solamente se ha creado para poder realizar validaciones dentro de la interfaz.

Atributos:

- `private Pattern patron;`
- `private Matcher corpus;`

metodos:

- **ingreseRegex:** Este metodo recibe un objeto de tipo String en su parametro. Este metodo realiza la tarea de crear un patron con el String que recibio en su parametro.
- **validar:** Este metodo recibe un objeto de tipo String en su parametro. Este metodo realiza la tarea de validar si el String cumple con las condiciones del patron. Si es que cumple con las condiciones devolvera un booleano True, caso contrario devolver un booleano False.

#### 12. **Ec.edu.ups.vista**

Este paquete contiene todas las interfaces en donde se interactuará con el usuario.

Clases:


- `ventanaAsunto`
- `VentanaCliente`
- `VentanaPrincipal`
- `VentanaProcurador`


#### 13. **VentanaAsunto**


Esta ventana permite que el usuario pueda registrar un asunto dentro de la base de datos. deberá llenar los datos y seleccionar la cedula del cliente y del procurador para guardar los datos dentro de la base de datos.

Design Preview [VentanaAsunto]

MENU ASUNTO

# ASUNTO:  ESTADO: --SELECCIONE UNA OPCIO... 

FECHA DE INICIO:  CLIENTE: --SELECCIONE UN CLIEN... 

FECHA FINAL:  PROCURADOR: --SELECCIONE UN PROCU... 

REGISTRAR

BUSCAR:  BUSCAR EDITAR

# CASO	FECHA INICIO	FECHA FINAL	ESTADO	CLIENTE	PROCURADOR
--------	--------------	-------------	--------	---------	------------

#### 14. VentanaCliente

Esta interfaz permite que el usuario pueda registrarse como cliente dentro de la base de datos.

Design Preview [VentanaCliente]

MENU USUARIO

CEDULA:  DIRECCION:

NOMBRE:  TELEFONO:


APELLIDO:  FECHA DE NACIMIENTO:

BUSCAR:

CEDULA	NOMBRE	APELLIDO	DIRECCION	TELEFONO
--------	--------	----------	-----------	----------

### 15. VentanaPrincipal

Esta ventana será la que contiene todas las interfaces dentro de ella.

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

MENU

☐ REGISTRAR C. shortcut  
☐ REGISTRAR P. shortcut  
☐ REGISTRAR A. shortcut  
☐ Exit shortcut

**16. VentanaProcurador**  
 Esta ventana permite que un usuario pueda registrarse como procurador dentro de la base de datos.

Design Preview [VentanaProcurador]

MENU PROCURADOR

CEDULA:  DIRECCION:

NOMBRE:  TELEFONO:

APELLIDO:  FECHA DE NACIMIENTO:

BUSCAR:


CEDULA	NOMBRE	APELLIDO	DIRECCION	TELEFONO
--------	--------	----------	-----------	----------

17.

RESULTADO(S) OBTENIDO(S):





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

**Firma de estudiante:** \_\_\_\_\_