

CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES

CARRERA: ASIGNATURA: Programación Aplicada

NRO. PRÁCTICA: TÍTULO PRÁCTICA: Trabajo en Clases

OBJETIVO ALCANZADO:

ACTIVIDADES DESARROLLADAS

 Se pidió desarrollar una aplicación que utilice lenguaje de programación JDBC y que este conectado a la base de datos PostgreSQL.

Se desarrollo los siguientes paquetes para la realización de esta práctica

- Ec.edu.ups.modelo
- Ec.edu.ups.controlador
- Ec.edu.ups.vista

2. Ec.edu.ups.modelo

En este paquete se crearon todos los objetos que se utilizaran para simular un programa de consultas entre clientes y abogados.

- Asunto
- Cliente
- Procurador

3. Asunto.java

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:

- private int numeroCaso;
- private Date FechaDelnicio;
- private Date fechaFinal;
- private String estado;
- private String cedulaCliente;
- private String cedulaProcurador;

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.

4. Cliente.java

Esta clase se utiliza para guardar los datos de un cliente dentro de la base de datos.



CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: 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 postgeSQL

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:



CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

- createAsunto: Este método recibe en su parámetro todos loa 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 loa 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:

- Conection: Este método realiza la conexión a la base de datos postgresSQL.
- Disconect: Este método realiza la desconexión a la base de datos postgreSQL

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.



CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: 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 boleano True, caso contrario devolver un boleano 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.

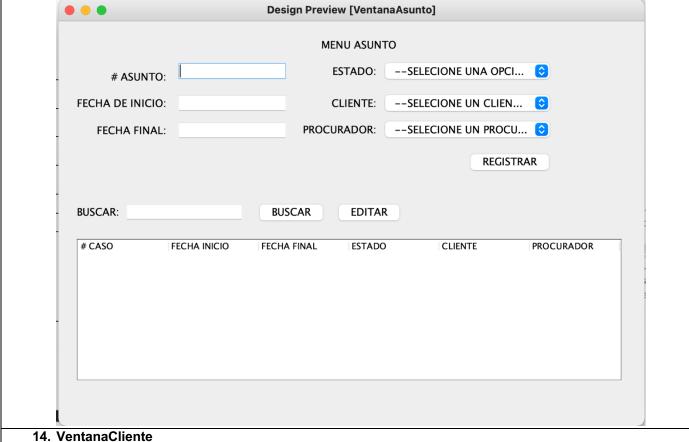


CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



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



CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

	Design Preview [VentanaCliente]							
	MEN	U USUARIO						
CEDULA:		DIRECCION:						
NOMBRE:		TELEFONO:						
APELLIDO:	FECHA D	DE NACIMIENTO:						
			REGISTRAR					
BUSCAR:	BUSCAR							
CEDULA	NOMBRE APELLIDO	DIRECCION	TELEFONO					
entanaPrincipal sta ventana será la qu	ue contiene todas las interf	aces dentro de ella.						

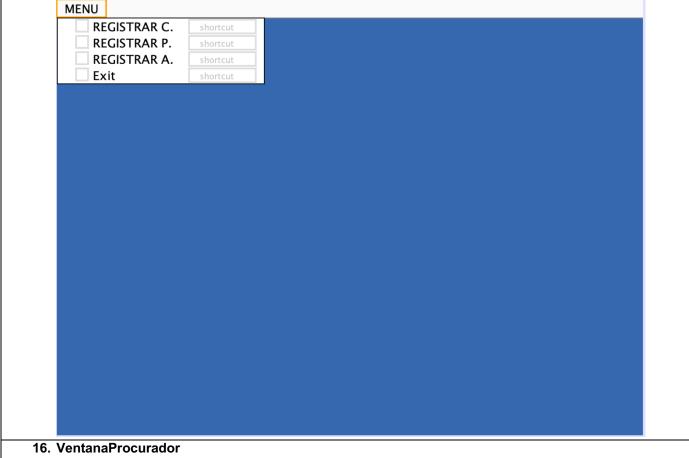


CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



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



Código: GUIA-PRL-001

Aprobación: 2016/04/06 CONSEJO ACADÉMICO

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

MENU PROCURADOR CEDULA: DIRECCION: NOMBRE: TELEFONO: APELLIDO: FECHA DE NACIMIENTO: REGISTRAR BUSCAR: CEDULA NOMBRE APELLIDO DIRECCION TELEFONO FADO(S) OBTENIDO(S):		Des	Design Preview [VentanaProcurador]							
NOMBRE: TELEFONO: APELLIDO: FECHA DE NACIMIENTO: REGISTRAR BUSCAR: CEDULA NOMBRE APELLIDO DIRECCION TELEFONO			MENU PROCURAI	OOR						
APELLIDO: FECHA DE NACIMIENTO: REGISTRAR BUSCAR: BUSCAR CEDULA NOMBRE APELLIDO DIRECCION TELEFONO	CEDULA:									
BUSCAR: BUSCAR CEDULA NOMBRE APELLIDO DIRECCION TELEFONO	NOMBRE:			TELEFONO:						
BUSCAR CEDULA NOMBRE APELLIDO DIRECCION TELEFONO	APELLIDO:		FECHA DE NA	ACIMIENTO:						
CEDULA NOMBRE APELLIDO DIRECCION TELEFONO					REGISTRAR					
	BUSCAR:		BUSCAR							
	CEDULA	NOMBRE	APELLIDO	DIRECCION	TELEFONO	7				
TADO(S) OBTENIDO(S):										
TADO(S) OBTENIDO(S):										
ΓADO(S) OBTENIDO(S):										
ΓADO(S) OBTENIDO(S):										
FADO(S) OBTENIDO(S):										
FADO(S) OBTENIDO(S):										
IADO(S) OBTENIDO(S):										
	TADO(S) OBTEN	NIDO(S):								
	TADO(S) OBTEN	NIDO(S):								
	TADO(S) OBTEN	NIDO(S):								

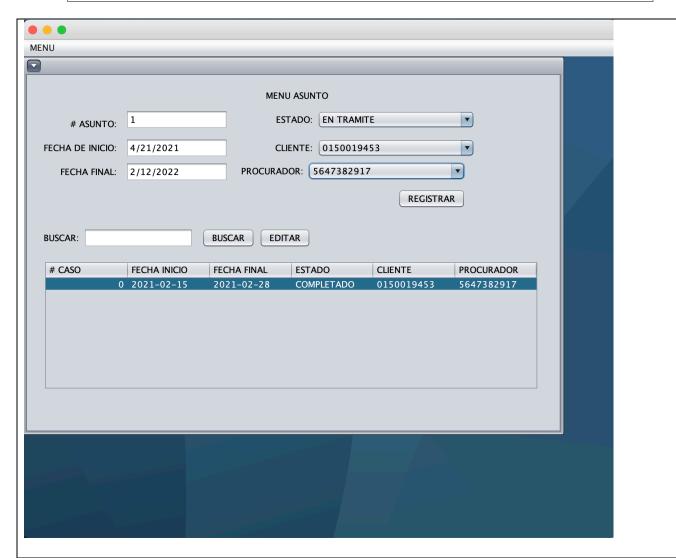


CONSEJO ACADÉMICO

Código: GUIA-PRL-001

PÉMICO Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación



CONCLUSIONES:

En conclusión, se debe tener un entendimiento de como programar utilizando el lenguaje de JDBC. Un punto que se debe tener en cuenta es que se debe siempre conectar la base de datos al rato de realizar una consulta, y por ultimo se debe desconectar la base de datos una vez realizado la consulta. Por ultimo se debe tener un entendimiento que pueden existir muchos lenguajes para diversas base de datos y se debe aprender a como utilizarlas dentro de java.

RECOMENDACIONES:

Nombre de estudiante: Denys Dutan



CONSEJO ACADÉMICO

Código: GUIA-PRL-001

Aprobación: 2016/04/06

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Firma de estudiante:	
Firma de estudiante:	