


|  |                               |                               |
|--|-------------------------------|-------------------------------|
|    | <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 |                               |                               |


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

---

Universidad Politécnica Salesiana

## Vicerrectorado Docente

|                            |  |
|----------------------------|--|
| Código del Formato:        | GUIA-PRL-001   |
| Versión:                   | VF1.0  |
| Elaborado por:             | Directores de Área del Conocimiento<br>Integrantes Consejo Académico |
| Fecha de elaboración:      | 2016/04/01   |
| Revisado por:              | Consejo Académico  |
| Fecha de revisión:         | 2016/04/06   |
| Aprobado por:              | Lauro Fernando Pesántez Avilés<br>Vicerrector Docente                |
| Fecha de aprobación:       | 2016/14/06   |
| Nivel de confidencialidad: | Interno  |

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

## Descripción General

### Propósito


El propósito del presente documento es definir un estándar para elaborar documentación de guías de práctica de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana, con la finalidad de lograr una homogenización en la presentación de la información por parte del personal académico y técnico docente.


### Alcance


El presente estándar será aplicado a toda la documentación referente a informes de prácticas de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana.

### Formatos

- Formato de Guía de Práctica de Laboratorio / Talleres / Centros de Simulación – para Docentes
- Formato de Informe de Práctica de Laboratorio / Talleres / Centros de Simulación – para Estudiantes

|  |                               |                               |
|--|-------------------------------|-------------------------------|
|   | <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 GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b> |  |
| <b>CARRERA:</b> COMPUTACIÓN   |   | <b>ASIGNATURA:</b> Programación Aplicada   |  |
| <b>NRO. PRÁCTICA:</b>   | 1 | <b>TÍTULO PRÁCTICA:</b> Base de Datos  |  |
| <b>OBJETIVO:</b><br>Identificar las sentencias SQL<br>Diseñar e Implementar codigos DDL, DML<br>Entender cada una de las características del uso de una Base de Datos |   |  |  |
| <b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):  |   | 1. Revisar los conceptos fundamentales de SQL  |  |
|   |   | 2. Establecer como implementar DDL y DML   |  |
|   |   | 3. Implementar y diseñar nuevas estructuras de base de datos y datos.                                |  |
|   |   | 4. Realizar el informe respectivo según los datos solicitados.                                       |  |
| <b>ACTIVIDADES POR DESARROLLAR</b><br>(Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)                                    |   |  |  |
| 1. Revisar la teoría y conceptos de Base de Datos   |   |  |  |
| 3. Probar y modificar datos utilizando el IDE.  |   |  |  |

|  |                               |                               |
|--|-------------------------------|-------------------------------|
|   | <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 |                               |                               |

## Práctica en clase

Se quiere diseñar una base de datos relacional para almacenar información sobre los asuntos que lleva un gabinete de abogados. Cada asunto tiene un número de expediente que lo identifica, y corresponde a un solo cliente. Del asunto se debe almacenar el período (fecha de inicio y fecha de archivo o finalización), su estado (en trámite, archivado, etc.), así como los datos personales del cliente al que pertenece (Cédula, nombre, dirección, etc.). Algunos asuntos son llevados por uno o varios procuradores, de los que nos interesa también los datos personales.

**4. Fecha de Entrega:** 18 Enero del 2021 23:55

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

Realizar procesos de Bases de Datos  
Entender las aplicaciones de codificación de las nuevas características de base de datos

### CONCLUSIONES:


Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java y SQL.


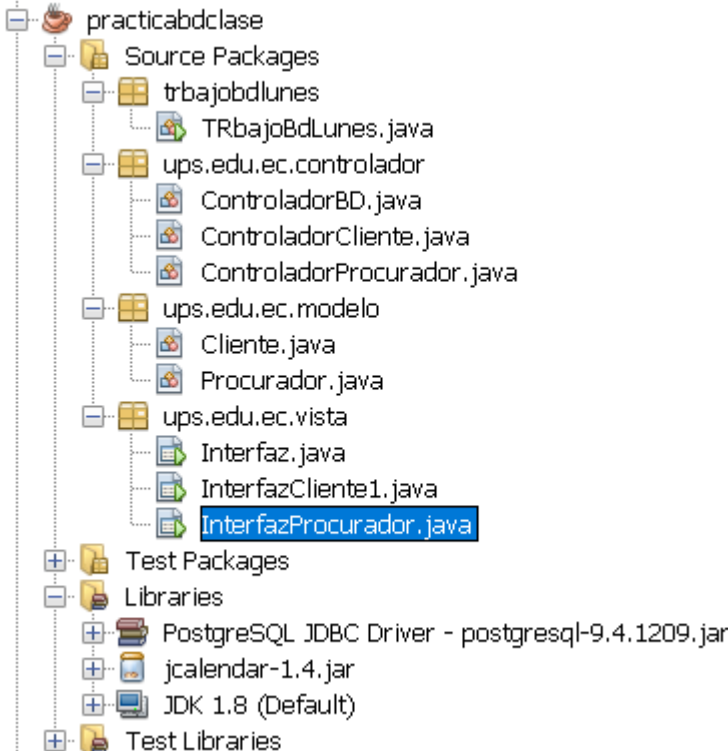
### RECOMENDACIONES:

Realizar el trabajo dentro del tiempo establecido.

**Docente / Técnico Docente:** \_\_\_\_\_

**Firma:** \_\_\_\_\_

|   |                        |                        |
|---|------------------------|------------------------|
|  | VICERRECTORADO DOCENTE | Código: GUIA-PRL-001   |
|   | CONSEJO ACADÉMICO      | Aprobación: 2016/04/06 |
| Formato: 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>   |  |
| <b>NRO. PRÁCTICA:</b>  |  | <b>TÍTULO PRÁCTICA:</b>  |  |
| <b>OBJETIVO ALCANZADO:</b>   |  |  |  |
| <b>ACTIVIDADES DESARROLLADAS</b>   |  |  |  |
|  |  |  |  |

```
InterfazCliente1.java x ControladorBD.java x ControladorCliente.java x SQL 1 [jdbc:postgresql://localho... x InterfazProcurador.java x Cliente.java x
Source History
17  * @author japch
18  */
19  public class ControladorBD {
20
21      public static Connection conectar(){
22          Connection con= null;
23          try{
24              Class.forName("org.postgresql.Driver");
25              con = DriverManager.getConnection("jdbc:postgresql://localhost:5432/bdpractclase","postgres","abc123");
26          }catch( Exception e){
27              System.err.println("Ocurrio un error "+e.getMessage());
28          }
29          System.out.println("la conexcion se dio con exito");
30          return con;
31      }
32
33      public static void desconectar(Connection con){
34          try {
35              con.close();
36          } catch (Exception e) {
37              e.printStackTrace();
38          }
39      }
40
41      public static ResultSet getTabla(String Consulta){
42
43          Connection cn = conectar();
44          Statement st;
45          ResultSet datos= null;
46
47          try {
48              st=cn.createStatement();
49              datos=st.executeQuery(Consulta);
50          } catch (Exception e) {
51
52          }
53          return datos;
54      }
55  }
```

2.

```
public class ControladorCliente {  
    public boolean insertar(Cliente cliente){  
        Connection con = ControladorBD.conectar();  
  
        try{  
            String sql = "INSERT INTO CLIENTE1(codigo, cedula, nombre, apellido, "  
                + "direccion, numero, fecha_nac) VALUES (?, ?, ?, ?, ?, ?, ?)";  
            PreparedStatement ps= con.prepareStatement(sql);  
            ps.setInt(1, cliente.getId());  
            ps.setString(2, cliente.getCedula());  
            ps.setString(3, cliente.getNombre());  
            ps.setString(4, cliente.getApellido());  
            ps.setString(5, cliente.getDireccion());  
            ps.setString(6, cliente.getNumero());  
            ps.setString(7, cliente.getFecchan());  
            ps.executeUpdate();  
  
        }catch(Exception e){  
            e.printStackTrace();  
            return false;  
        } finally{  
            ControladorBD.desconectar(con);  
        }  
  
        return true;  
    }  
}
```

3.

```
public class ControladorProcurador {  
    public boolean insertar(Procurador procurador){  
        Connection con = ControladorBD.conectar();  
  
        try{  
            String sql = "INSERT INTO PROCURADOR(codigo, cedula, nombre, apellido, "  
                +"direccion, numero,salario, fecha_nac) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";  
            PreparedStatement ps= con.prepareStatement(sql);  
            ps.setInt(1, procurador.getId());  
            ps.setString(2, procurador.getCedula());  
            ps.setString(3, procurador.getNombre());  
            ps.setString(4, procurador.getApellido());  
            ps.setString(5, procurador.getDireccion());  
            ps.setString(6, procurador.getNumero());  
            ps.setDouble(7, procurador.getSueldo());  
            ps.setString(8, procurador.getFecchan());  
            ps.executeUpdate();  
  
        }catch(Exception e){  
            e.printStackTrace();  
            return false;  
        } finally{  
            ControladorBD.desconectar(con);  
        }  
  
        return true;  
    }  
}
```



```

public InterfazCliente1() {
    initComponents();
}

public void ingresarDatos() {
    String id= txtid.getText();
    String cedula = txtcedula.getText();
    String nombre = txtnombre.getText();
    String apellido= txtapellido.getText();
    String direccion = txtdireccion.getText();
    String numero = txtnumero.getText();
    String fechan = df.format(datafechan.getDate());
    System.out.println("ID:"+id);
    Cliente cliente = new Cliente(Integer.parseInt(id), cedula, nombre, apellido, direccion, numero, fechan);
    cp.insertar(cliente);
    System.out.println("Lista "+cliente);
}

private void mostrar() {
    ResultSet rs = ControladorBD.getTabla("select codigo, cedula, nombre, apellido, "
        + "direccion, numero, fecha_nac from cliente1");
    modelo.setColumnIdentifiers(new Object[] {"codigo", "cedula", "nombre", "apellido",
        "direccion", "numero", "fecha_nac" });

    try {
        while(rs.next()) {
            modelo.addRow(new Object[] {rs.getString("codigo"), rs.getString("cedula"),
            rs.getString("nombre"), rs.getString("apellido"), rs.getString("direccion"),
            rs.getString("numero"), rs.getString("fecha_nac")});
        }
        jTableCliente.setModel(modelo);
    } catch (Exception e) {
        System.err.println(e);
    }
}
}

4.
/**
 *
 */
public InterfazProcurador() {
    initComponents();
}

public void ingresarDatos() {
    String id= txtid.getText();
    String cedula = txtcedula.getText();
    String nombre = txtnombre.getText();
    String apellido= txtapellido.getText();
    String direccion = txtdireccion.getText();
    String numero = txtnumero.getText();
    String sueldo = txtsueldo.getText();
    String fechan = df.format(datafechan.getDate());
    System.out.println("ID:"+id);

    Procurador procurador = new Procurador(Integer.parseInt(id), cedula, nombre, apellido, direccion, numero, Double.parseDouble(sueldo));
    cp.insertar(procurador);
    // System.out.println("Lista "+cliente);
}

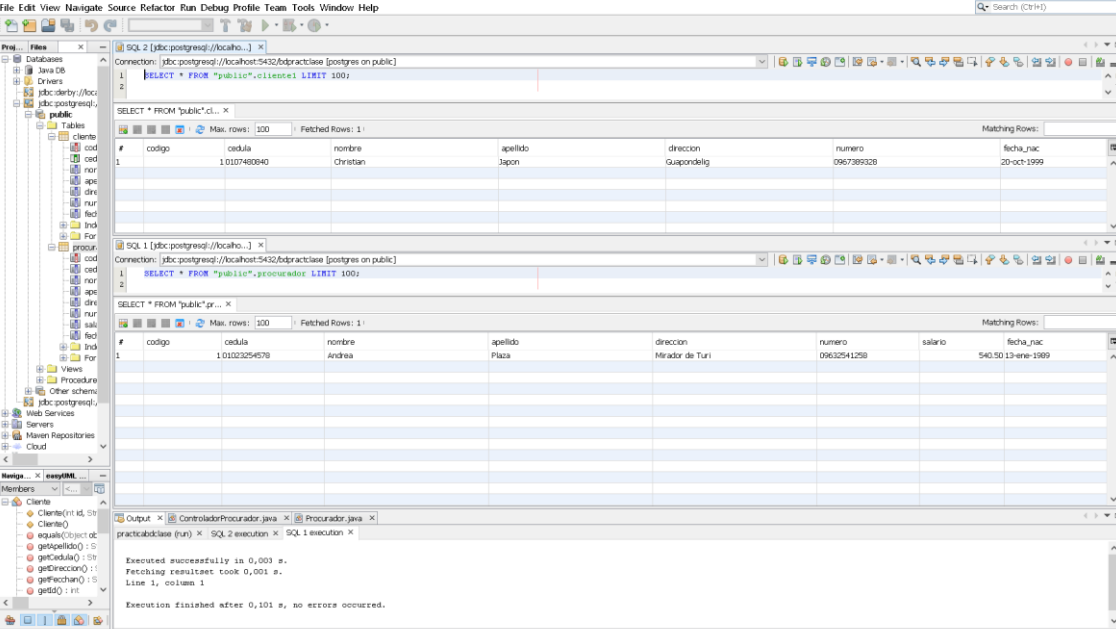
private void mostrar() {
    ResultSet rs = ControladorBD.getTabla("select codigo, cedula, nombre, apellido, "
        + "direccion, numero, salario, fecha_nac from procurador");
    modelo.setColumnIdentifiers(new Object[] {"codigo", "cedula", "nombre", "apellido",
        "direccion", "numero", "salario", "fecha_nac" });

    try {
        while(rs.next()) {
            modelo.addRow(new Object[] {rs.getString("codigo"), rs.getString("cedula"),
            rs.getString("nombre"), rs.getString("apellido"), rs.getString("direccion"),
            rs.getString("numero"), rs.getDouble("salario"), rs.getString("fecha_nac")});
        }
        jTableCliente.setModel(modelo);
    } catch (Exception e) {
        System.err.println(e);
    }
}
}

/**

```

5.



SQL 2 [jdbc:postgresql://localhost:5432/...]   
 Connection: jdbc:postgresql://localhost:5432/... [postgres on public]   
 1 SELECT \* FROM "public".cliente LIMIT 100;   
 2

SQL 1 [jdbc:postgresql://localhost:5432/...]   
 Connection: jdbc:postgresql://localhost:5432/... [postgres on public]   
 1 SELECT \* FROM "public".proceder LIMIT 100;   
 2

Output:   
 Executed successfully in 0,003 s.   
 Fetching resultset took 0,001 s.   
 Line 1, column 1   
 Execution finished after 0,101 s, no errors occurred.

6.

N.

CLIENTE

Actualizar Tabla

ID: 2

CEDULA: 01215457899°

NOMBRE: Andres

APELLIDO: Perez

DIRECCION: Totora

NUMERO: 0321456789


FECHA: 13-ene-2001

Añadir

Editar

Eliminar

| codigo | cedula       | nombre    | apellido | direccion   | numero     | fecha_nac   |
|--------|--------------|-----------|----------|-------------|------------|-------------|
| 1      | 0107480840   | Christian | Japon    | Guapondelig | 0967389328 | 20-oct-1999 |
| 2      | 01215457899° | Andres    | Perez    | Totora      | 0321456789 | 13-ene-2001 |

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

**RESULTADO(S) OBTENIDO(S):**

**CONCLUSIONES:**

**RECOMENDACIONES:**

**Nombre de estudiante:** \_\_\_\_\_

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