LENGUAJE DE PROGRAMACION II

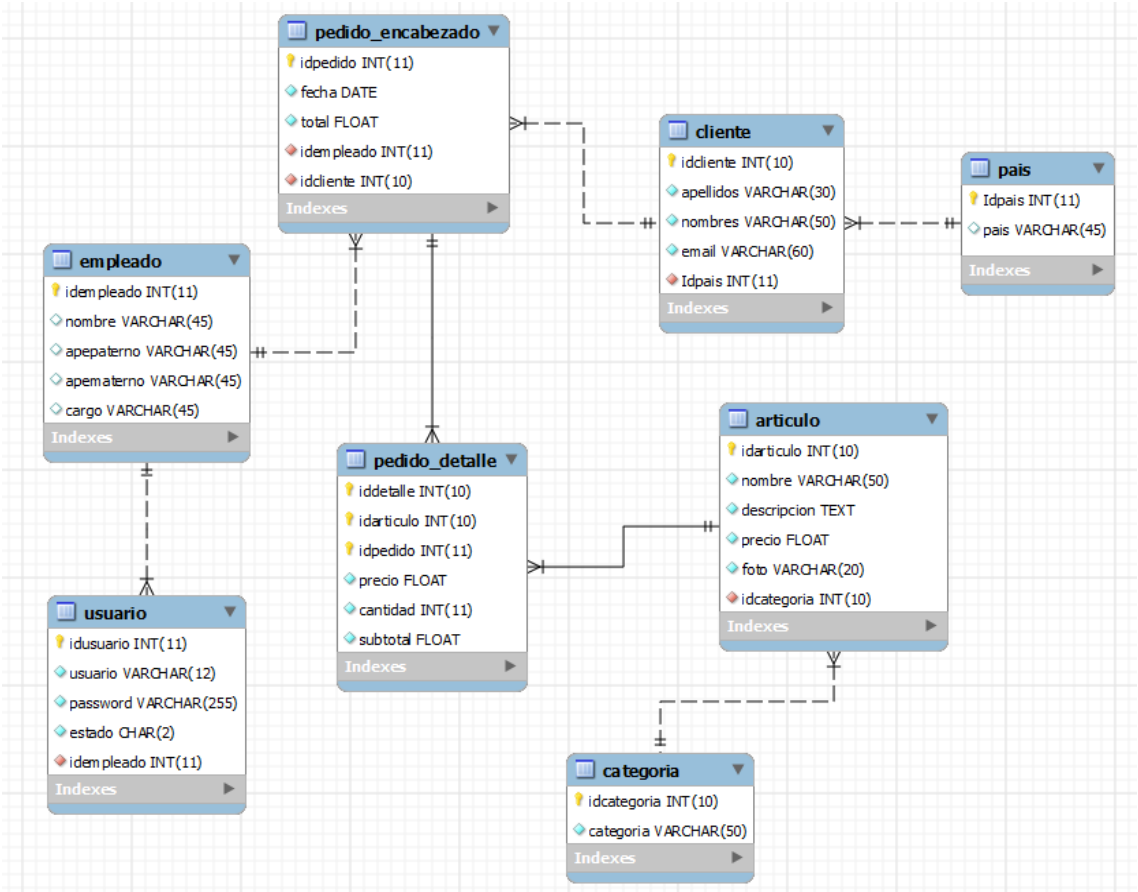
Docente: Ing. Díaz Leyva Teodoro

Tema: **Consultas a Base de Datos**

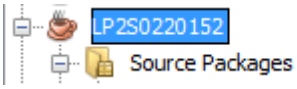
Objetivo

Crear una aplicación para realizar consultas a varias tablas de la base de datos tienda.

Esquema de la base de datos tienda:



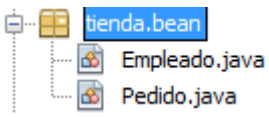
1. Crear proyecto:



2. En el proyecto, en el paquete “tienda.conexion” crear la clase Conexión :

```
package tienda.conexion;
import java.sql.Connection;
import java.sql.DriverManager;

public class Conexion {
 private   static final String url = "jdbc:mysql://localhost/tienda";
private   static final String usuario = "root";
private   static final String Password = "123";

    public static Connection abrir() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection cn = (Connection) DriverManager.getConnection(url, usuario, Password);
            return cn;
        } catch (Exception ex) {
            return null;
        }
    }
}
```

3. En paquete “tienda.bean”, crear la clase Empleado y Pedido
   Sus datos de estas clases representan a los campos de las respectivas tablas de la base de datos tienda.



- Clase Empleado:

```
package tienda.bean;
public class Empleado {
  private  int idempleado;
  private String nombre;
  private String apepaterno;
  private String apematerno;
```

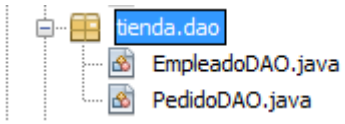
```
    private String cargo;
     public int getIdempleado() {
        return idempleado;
     }
     public void setIdempleado(int idempleado) {
        this.idempleado = idempleado;
     }
     public String getNombre() {
        return nombre;
     }
     public void setNombre(String nombre) {
        this.nombre = nombre;
     }
     public String getApepaterno() {
        return apepaterno;
     }
     public void setApepaterno(String apepaterno) {
        this.apepaterno = apepaterno;
     }
     public String getApematerno() {
        return apematerno;
     }
     public void setApematerno(String apematerno) {
        this.apematerno = apematerno;
     }
     public String getCargo() {
        return cargo;
     }

     public void setCargo(String cargo) {
        this.cargo = cargo;
     }
  }
```

- Pedido:

```
package tienda.bean;
import java.sql.Date;
public class Pedido {
    private int idpedido;
    private Date fecha;
    private float total;
    private int idempleado;
    private int idcliente;
    public int getIdpedido() {
        return idpedido;
    }
    public void setIdpedido(int idpedido) {
        this.idpedido = idpedido;
    }
    public Date getFecha() {
        return fecha;
    }
    public void setFecha(Date fecha) {
        this.fecha = fecha;
    }
    public float getTotal() {
        return total;
    }
    public void setTotal(float total) {
        this.total = total;
    }
    public int getIdempleado() {
        return idempleado;
    }
    public void setIdempleado(int idempleado) {
        this.idempleado = idempleado;
    }
    public int getIdcliente() {
        return idcliente;
    }
    public void setIdcliente(int idcliente) {
        this.idcliente = idcliente;
    }
}
```

4. En el paquete “tienda.dao”, crear la clase EmpleadoDAO y PedidoDAO

Estas clases tendrán los métodos para realizar operaciones sobre la base de datos tienda.



- Clase EmpleadoDAO

```
package tienda.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import tienda.bean.Empleado;
import tienda.conexion.Conexion;
public class EmpleadoDAO {
  //-------buscar empleado por nombre y retornar en un arreglo--------
  public static ArrayList<Empleado> buscarEmpleadoPorNombre(String nom) {

    //lista es un objeto de tipo ArrayList, sus elementos son objetos tipo Empleado.
     ArrayList<Empleado> lista = new ArrayList<Empleado>();

     //sentecia sql para consultar un empleado por su nombre
     String sql = "select* from Empleado where nombre like ?";

    //abrir la conexion
     Connection cn = Conexion.abrir();
     //variable tipo Empleado
     Empleado emp = null;

     try {
        //presparedStatement para ejecutar sentencia sql
        PreparedStatement ps = cn.prepareStatement(sql);

      //asignar valor al parametro ?
        ps.setString(1, nom + '%');
        //ejecutar sentencia sql  a traves de ps y asignar resultados en resultset
        ResultSet rs = ps.executeQuery();
```

```
        //leer resultset
          while (rs.next()) {
             //objeto tipo Empleado para encapsular los datos  que se lee del resultset
             emp = new Empleado();
             //encapsular datos en el objeto emp
             emp.setIdempleado(rs.getInt(1));
             emp.setNombre(rs.getString(2));
             emp.setApematerno(rs.getString(3));
             emp.setApematerno(rs.getString(4));
             emp.setCargo(rs.getString(5));
             //agrear objeto emppleado al arrelo
             lista.add(emp);
          }
          //cerrar objetos
          rs.close();
          ps.close();
          cn.close();
       } catch (SQLException ex) {
          return null;
       }
       return lista;
    }
    //-------buscar empleado por código y retornar en un objeto empleado-----
    public static Empleado buscarEmpleadoPorCodigo(int  cod) {

       //sentencia sql para consultar un empleado por su codigo
       String sql = "select* from Empleado where idempleado=?";

      //abrir la conexion
       Connection cn = Conexion.abrir();
       //variable tipo Empleado
       Empleado emp = null;

       try {
          //presparedStatement para ejecutar sentencia sql
          PreparedStatement ps = cn.prepareStatement(sql);
          //asignar valor al parametro ?
          ps.setInt(1, cod);
          //ejecutar sentencia sql  a traves de ps y asignar resultados en resultset
          ResultSet rs = ps.executeQuery();
```

```
        //leer resultset
          while (rs.next()) {
             //objeto tipo Empleado para encapsular los datos  que se lee del resultset
             emp = new Empleado();
             //encapsular datos en el objeto emp
             emp.setIdempleado(rs.getInt(1));
             emp.setNombre(rs.getString(2));
             emp.setApepaterno(rs.getString(3));
             emp.setApematerno(rs.getString(4));
             emp.setCargo(rs.getString(5));
                      }

      //cerrar objetos
          rs.close();
          ps.close();
          cn.close();
       } catch (SQLException ex) {
          return null;
       }
       return emp;
    }

    //-------lista de empelados----------------------------------
    public static ArrayList<Empleado> listar() {
       //lista es un objeto de tipo ArrayList, sus elementos son objetos tipo Empleado.
       ArrayList<Empleado> lista = new ArrayList<Empleado>();

     //sentecia sql para consultar un empleado por su nombre
       String sql = "select* from Empleado";
       //abrir la conexion

       Connection cn = Conexion.abrir();
       //variable tipo Empleado
       Empleado emp = null;

       try {
          //presparedStatement para ejecutar sentencia sql
          PreparedStatement ps = cn.prepareStatement(sql);

        //ejecutar sentencia sql  a traves de ps y asignar resultados en resultset
          ResultSet rs = ps.executeQuery();
```

```
        //leer resultset
            while (rs.next()) {
                //objeto tipo Empleado para encapsular los datos  que se lee del resultset
                emp = new Empleado();
                //encapsular datos en el objeto emp
                emp.setIdempleado(rs.getInt(1));
                emp.setNombre(rs.getString(2));
                emp.setApematerno(rs.getString(3));
                emp.setApematerno(rs.getString(4));
                emp.setCargo(rs.getString(5));
                //agrear objeto emppleado al arrelo
                lista.add(emp);
            }

            //cerrar objetos
            rs.close();
            ps.close();
            cn.close();
        } catch (SQLException ex) {
            return null;
        }
        return lista;
    }
}
```

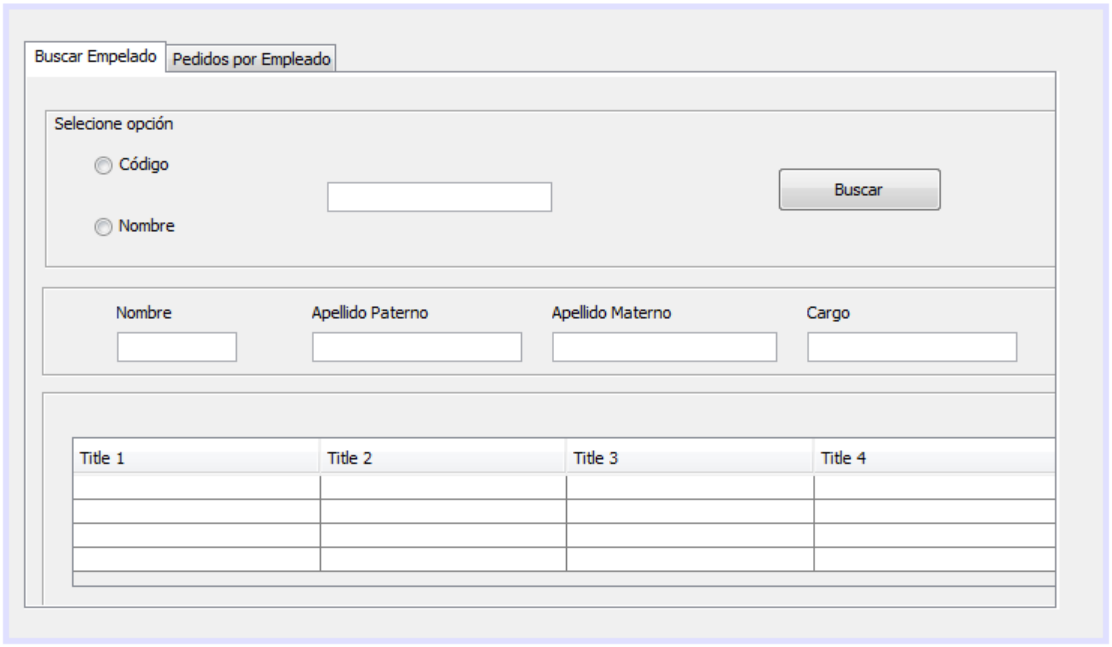
- Clase PedidoDAO

```
package tienda.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import tienda.bean.Pedido;
import tienda.conexion.Conexion;

public class PedidoDAO {
  //------buscar empleado por nombre y retornar en un arreglo
    public static ArrayList<Pedido> pedidosPorEmpleado(String nom) {

     //lista es un objeto de tipo ArrayList, sus elementos son objetos tipo Pedido.
       ArrayList<Pedido> lista = new ArrayList<Pedido>();
```

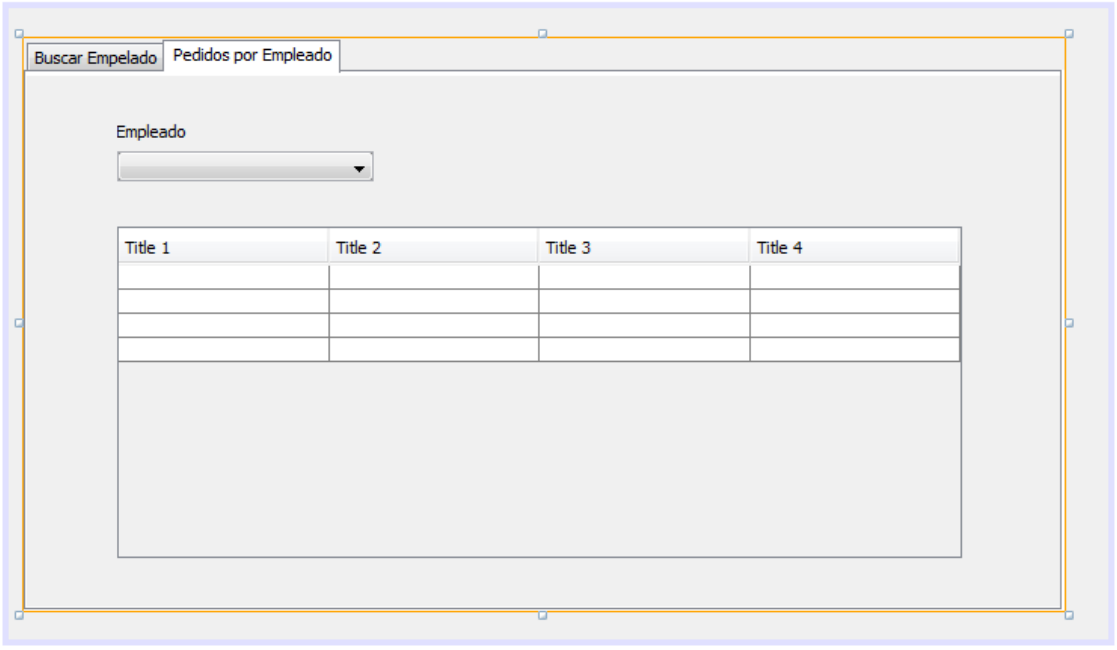
```
//sentecia sql para consultar pedidos por empleado
String sql = "SELECT
pedido_encabezado.idpedido,pedido_encabezado.fecha,pedido_encabezado.total,pedido_encabeza
do.idempleado,pedido_encabezado.idempleado" +
" FROM empleado, pedido_encabezado where
empleado.idempleado=pedido_encabezado.idempleado and empleado.nombre=?";

    //abrir la conexion
      Connection cn = Conexion.abrir();
      //variable tipo Pedido
      Pedido ped= null;
      try {
         //presparedStatement para ejecutar sentencia sql
         PreparedStatement ps = cn.prepareStatement(sql);

         //asignar valor al parametro ?
         ps.setString(1, nom);
         //ejecutar sentencia sql  a traves de ps y asignar resultados en resultset
         ResultSet rs = ps.executeQuery();
         //leer resultset
         while (rs.next()) {
            //objeto tipo Empleado para encapsular los datos  que se lee del resultset
            ped = new Pedido();
            //encapsular datos en el objeto emp
            ped.setIdpedido(rs.getInt(1));
            ped.setFecha(rs.getDate(2));
            ped.setTotal(rs.getFloat(3));
            ped.setIdempleado(rs.getInt(4));
            ped.setIdcliente(rs.getInt(5));
            //agrear objeto pedido al arrelo
            lista.add(ped);
         }
         //cerrar objetos
         rs.close();
         ps.close();
         cn.close();
      } catch (SQLException ex) {
         return null;
      }
      return lista;
   }
}
```

5. En el paquete" tienda. presentación", crear la interfaz grafica:

En la ficha Buscar Empleado:



En la ficha Pedidos por Empleado:



- CODIFICACION EN EL "frm_empleados":

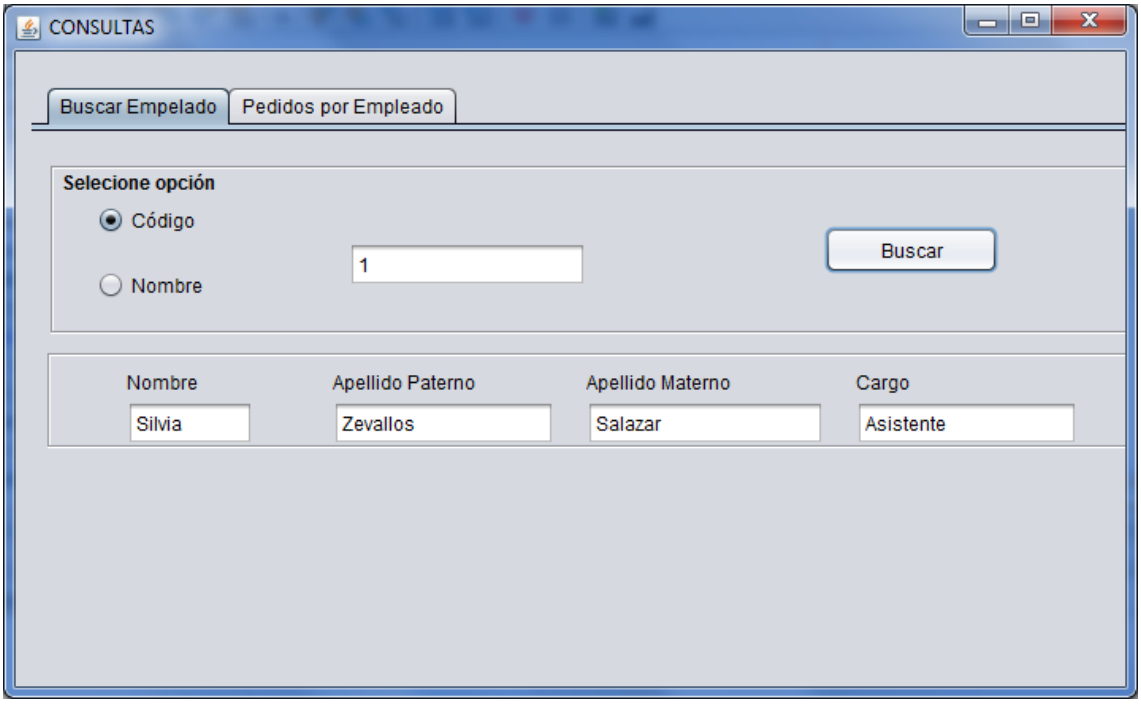
```
public class frm_empleado extends javax.swing.JFrame {
    //variable tipo modelo de tabla
    DefaultTableModel md1;
    DefaultTableModel md2;
    //columns del modelo de tabla
    String c1[] = {"CODIO", "NOMBRE", "APELLIDO PATERNO", "APELLIDO MATERNO", "CARGO"};
    String c2[] = {"ID", "FECHA", "TOTAL"};

//constructor
    public frm_empleado() {
        initComponents();
        setTitle("CONSULTAS");
//invocación de métodos
        ocultarPaneles();
        addEmpleado();
    }


    //-------------buscar empleado----------------
    public void buscarEmpleado() {

        String p = txtempleado.getText();

        if (rbCodigo.isSelected() == true) {
            if (p.length() > 0) {
                //obtenemos empleado por codigo
                Empleado emp = EmpleadoDAO.buscarEmpleadoPorCodigo(Integer.parseInt(p));
                //asignar datos de empleado a cuadros de text
                txtnombre.setText(emp.getNombre());
                txtpaterno.setText(emp.getApepaterno());
                txtmaterno.setText(emp.getApematerno());
                txtcargo.setText(emp.getCargo());
                //invocar a visibilidadPaneles
                visibilidadPaneles(true);
            } else {
                mensaje("Ingrese código  de empleado..");
                txtempleado.requestFocus();
            }
        } else if (rbNombre.isSelected() == true) {
            if (p.length() > 0) {
```

```
            md1 = new DefaultTableModel(null, c1);
            ArrayList<Empleado> lista = EmpleadoDAO.buscarEmpleadoPorNombre(p);
            for (Empleado x : lista) {

                md1.addRow(new Object[]{x.getIdempleado(), x.getNombre(), x.getApepaterno(),
x.getApematerno(), x.getCargo()});
                tablaEmpleados.setModel(md1);
            }
            visibilidadPaneles(false);
        } else {
            mensaje("Ingrese nombre  de empleado..");
            txtempleado.requestFocus();
        }
    } else {
        mensaje("Seleccione botón de busqueda...");

    }
}

    //------visibilidad de paneles----

        public void visibilidadPaneles(boolean v) {
            panelCodigo.setVisible(v);
            panelTabla.setVisible(!v);
        }

        public void ocultarPaneles() {
            panelCodigo.setVisible(false);
            panelTabla.setVisible(false);
        }
        //--------limpiar tabla--------

        public void limpiar() {
            while (tablaEmpleados.getRowCount() > 0) {
                ((DefaultTableModel) tablaEmpleados.getModel()).removeRow(0);
            }
        }

        //-----------------cuadro de dialogo mensaje----------
        public void mensaje(String m) {
            JOptionPane.showMessageDialog(this, m);
        }
```

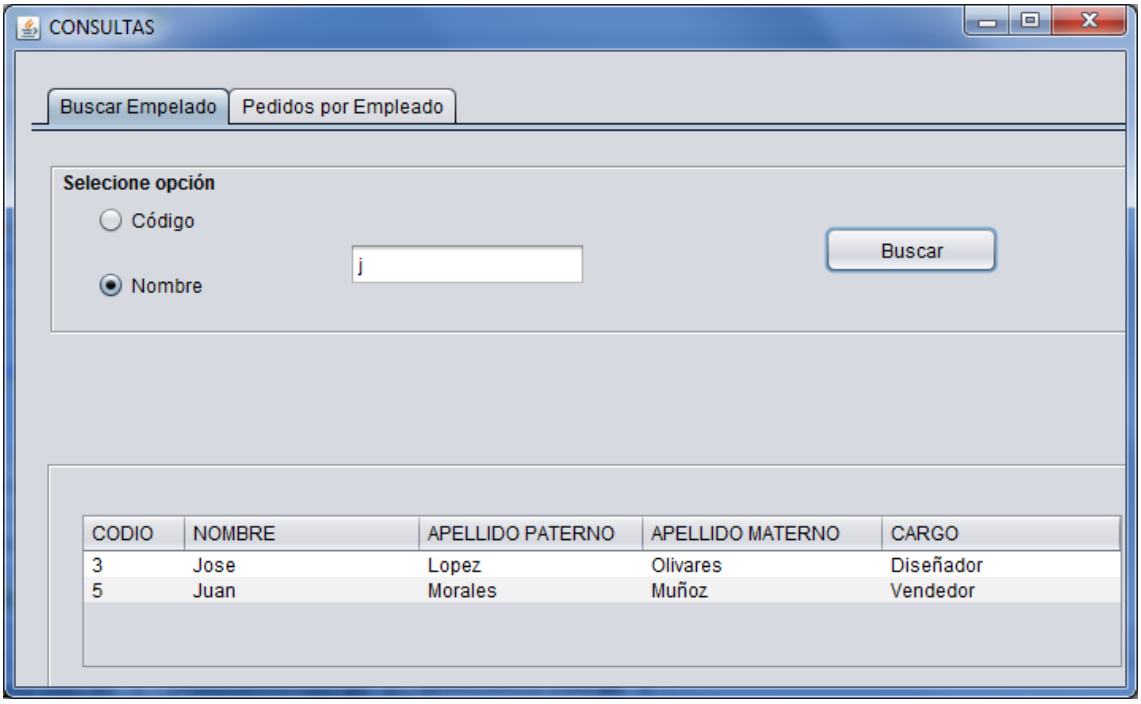
```
//------------------agregar empleados en combobox
    public void addEmpleado() {

        ArrayList<Empleado> lista = EmpleadoDAO.listar();
        for (Empleado x : lista) {
            cboEmpleado.addItem(x.getNombre());
        }
    }

//llamar  a los métodos en los eventos de los objetos
private void btnbuscarActionPerformed(java.awt.event.ActionEvent evt) {
        buscarEmpleado();
    }
//boton de opcion codigo:
    private void rbCodigoActionPerformed(java.awt.event.ActionEvent evt) {
        txtempleado.setText("");
        txtempleado.requestFocus();
        ocultarPaneles();
    }

//boton de opcion nombre:
    private void rbNombreActionPerformed(java.awt.event.ActionEvent evt) {
        txtempleado.setText("");
        txtempleado.requestFocus();
        ocultarPaneles();
        limpiar();
    }

//lista desplegable Empleado
    private void cboEmpleadoActionPerformed(java.awt.event.ActionEvent evt) {
        ArrayList<Pedido>lista =PedidoDAO.pedidosPorEmpleado((String)
cboEmpleado.getSelectedItem());
         md2 = new DefaultTableModel(null, c2);

        for (Pedido x : lista) {
            md2.addRow(new Object[]{x.getIdpedido(),x.getFecha(),x.getTotal()});
        }
         tablaPedidos.setModel(md2);
    }
```

//**********EJECUCION DEL PROYECTO**********

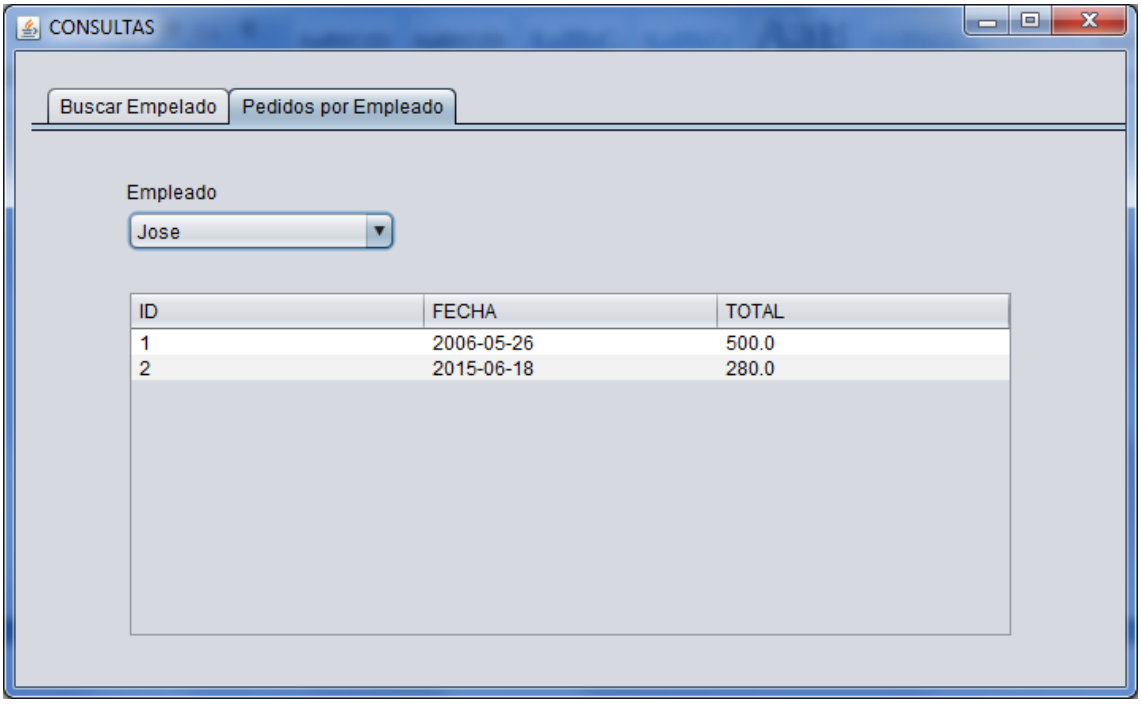
- Buscar empleado por código:



- Buscar empleado por Nombre:



Pedidos por empelado



Ejercicios propuestos:

1. Cree una consulta para listar los artículos por categoría, se debe ingresa como dato el nombre de la categoría.
2. Cree una consulta para listar los pedidos por empleado, se debe ingresa como dato el nombre del empleado para listar los pedidos atendidos por dicho empleado.
3. Cree una consulta para listar los pedidos por un intervalo de fechas, es decir se ingresa una fecha inicial y una fecha final, los pedidos a listar solamente estarán entre estas dos fechas.
4. Listar los artículos (campos a visualizar idarticulo, nombre, descripción y precio) que han sido pedidos por un determinado cliente, el nombre del cliente se ingresa por la interfaz grafica.
5. Listar articulo con el mayor precio
6. Buscar un articulo por su nombre
7. Listar el país, fecha de pedido, monto total de pedido y nombre del cliente.
8. Listar los artículos que no han sido pedidos
9. Listar los clientes de la base de datos que no han realizado ningún pedido
10. Listar idarticulo, nombre y precio de los artículos que han sido pedidos