SISTEMA “COMO & COMO”

Entregable No. Tres

Elaborado por:

Ana Karen Hdez. Murrieta, Jesús Alberto Rodríguez Hernández, Alberto Sirio Torres Cruz, Isay Armando González Cruz.

Proyecto de Base de Datos Distribuidos.

Código relevante de la aplicación

Esta parte de código está hecha en una clase java dónde se hace la conexión a Oracle y se hacen las consultas para cada tabla de nuestra base de datos.

```
/**
 *
 * @author Karen, Isay, Sirio, Jesús.
 */
import java.sql.*;
public class ComoComo {
    static Connection con=null;
    static Statement sta=null;
    static ResultSet re=null;
    public static Connection conexion(Connection con) throws SQLException{
        try{
            Class.forName("oracle.jdbc.OracleDriver");
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE",
"sysdba", "123");


        }
        catch (ClassNotFoundException e){
            System.out.println("No se pudo realizar la conexion");
        }
        return con;
    }
public static Statement State (Statement sta) throws SQLException{
    con=conexion(con);
    sta=con.createStatement();
    return sta;
}
//hacaer un resultset para cada tabla
```

```
public static ResultSet Resu (ResultSet re) throws SQLException{
    //consulta para centro
    sta=State(sta);
    re=sta.executeQuery("select * from Centro");
    return re;
}
public static ResultSet ResuClie (ResultSet re) throws SQLException{
    //consulta para cliente
    sta = State(sta);
    re = sta.executeQuery("select * from Cliente");
    return re;
}
public static ResultSet ResuEmple (ResultSet re) throws SQLException{
    //consulta para empleado
    sta = State(sta);
    re = sta.executeQuery("select * from Empleado");
    return re;
}
public static ResultSet ResuFact (ResultSet re) throws SQLException{
    //consulta para factura
    sta = State(sta);
    re = sta.executeQuery("select * from Factura");
    return re;
}
public static ResultSet ResuPedi (ResultSet re) throws SQLException{
    //consulta para pedido
    sta = State(sta);
    re = sta.executeQuery("select * from Pedido");
    return re;
}
```

```
public static ResultSet ResuPlat (ResultSet re) throws SQLException{
    //consulta para plato
    sta = State(sta);
    re = sta.executeQuery("select * from Plato");
    return re;
}
public static ResultSet Result(ResultSet re) throws SQLException{
    //Consulta para Población
    sta=State(sta);
    //Comando de consulta tipo Query + comando en oracle para mostrar
    re=sta.executeQuery("select * from Poblacion");
    return re;
}
}
```

Nuestra aplicación tiene una ventana o "JFrame" principal en donde mandamos a llamar a otras ventanas.

El código que hicimos para mandar a llamar cada ventana quedó así:

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

```
        Centro cntro = new Centro();
        cntro.setVisible(true);
        cntro.pack();
    }
private void ClienteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Cliente clinte = new Cliente();
        clinte.setVisible(true);
        clinte.pack();
    }
private void EmpleadoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Empleado mpleado = new Empleado();
        mpleado.setVisible(true);
        mpleado.pack();
    }
private void FacturaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Factura fctura = new Factura();
        fctura.setVisible(true);
        fctura.pack();
    }
private void PedidoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Pedido pdido = new Pedido();
        pdido.setVisible(true);
        pdido.pack();
    }
private void PlatoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
        Plato plto = new Plato();
        plto.setVisible(true);
        plto.pack();
    }
private void PoblacionActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Poblacion pblacion = new Poblacion();
        pblacion.setVisible(true);
        pblacion.pack();
    }
```

A continuación pondremos el código de "agregar" o "insertar" datos de una sola ventana, puesto que las demás ventanas tienen lo mismo, sólo cambian los campos. Iniciamos la conexión con la base de datos mandando a llamar a nuestra clase principal, ya que ahí tenemos la conexión, luego hacemos la sentencia en Oracle "insert into… values…", es importante que al guardar los datos de nuestra ventana se guarden en el mismo orden que tenemos en nuestra base, ya que de otra manera, se insertarán los datos en campos que no son correctos.

```
private void GuardarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFrame marco = new JFrame("DialogDemo");
        marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        marco.pack();
        try{
            con = ComoComo.conexion(con);
            String insertar = "insert into Cliente values(?,?,?,?,?,?,?,?)";
            PreparedStatement estado = con.prepareStatement(insertar);
            estado.setString(1, CodCliente.getText());
            estado.setString(2, NomCliente.getText());
            estado.setString(3, APClie.getText());
            estado.setString(4, AMClie.getText());
            estado.setInt(5, Integer.parseInt(TelClie.getText()));
```

```
            estado.setString(6, CalleClie.getText());
            estado.setString(7, ColoniaClie.getText());
            estado.setInt(8, Integer.parseInt(NumCasaClie.getText()));
            estado.execute();
            estado.close();
            JOptionPane.showMessageDialog(marco, "Registro insertado");
            con.close();
        }catch (Exception e){
            System.out.println("No se pudo guardar el registro");
        }
    }
```

En esta parte de código eliminaremos de Población los datos, al igual que en insertar, sólo pondremos el código de una ventana, ya que se utiliza prácticamente el mismo código para todas las demás. Para hacer la eliminación de datos utilizaremos la sentencia "delete" y utilizaremos sólo la llave primaria de nuestra tabla en la base y la compararemos con nuestro Text Field, si coincide entonces se borrará de la tabla.

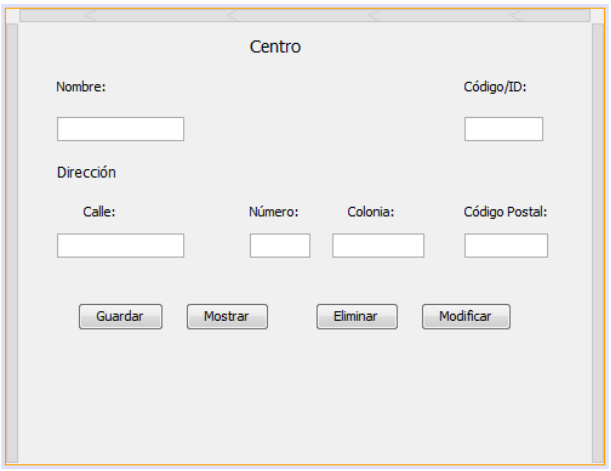
```
private void EliminarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFrame marco = new JFrame("DialogDemo");
        marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        marco.pack();
        try{
            con=ComoComo.conexion(con);
            String eliminar = ("delete from Cliente where codigo = '" + CodCliente.getText() +
"'");
            PreparedStatement estado = con.prepareStatement (eliminar);
            estado.execute();
            estado.close();
            con.close();
            JOptionPane.showMessageDialog(marco, "Registro Eliminado");
        }catch (Exception e){
```

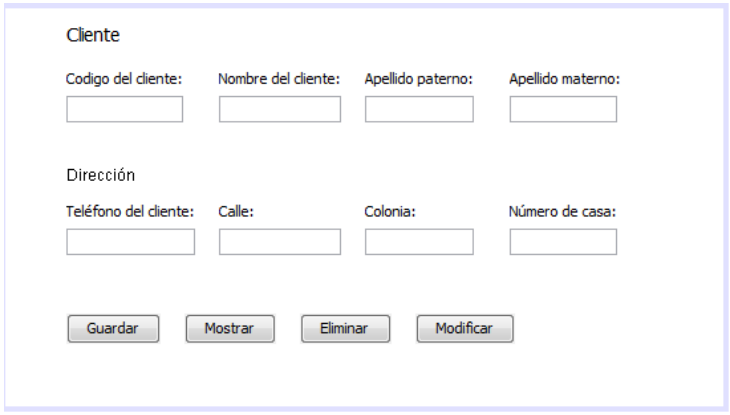
```
            System.out.println("No se pudo eliminar el registro");
        }
    }
```

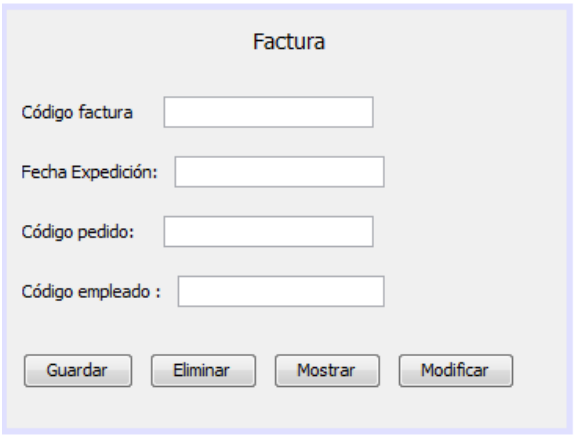
Ahora utilizaremos el código de modificar de la ventana Población, utilizaremos la sentencia "update" y nos aseguraremos de poner todos los campos que se van a modificar y lo validaremos con nuestra llave primaria.

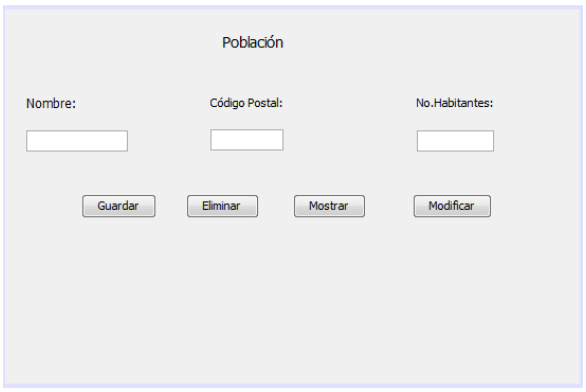
```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JFrame marco = new JFrame("DialogDemo");
        marco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        marco.pack();
        try{
            con=ComoComo.conexion(con);
            String modificar = ("update Poblacion set Nombre ='"+ Nombre.getText()+
"',nohabitantes='"+ Habitantes.getText()+"' where CP="+ CP.getText());
            PreparedStatement estado = con.prepareStatement(modificar);
            estado.execute();
            estado.close();
            con.close();
            JOptionPane.showMessageDialog(marco, "Registro Modificado");
        }
        catch (Exception e){
            System.out.println("No se pudo modificar el registro");
        }
    }
```

Ventanas de nuestro sistema “Como&Como”



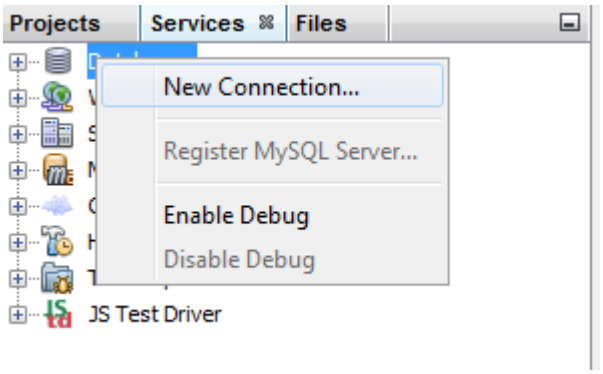




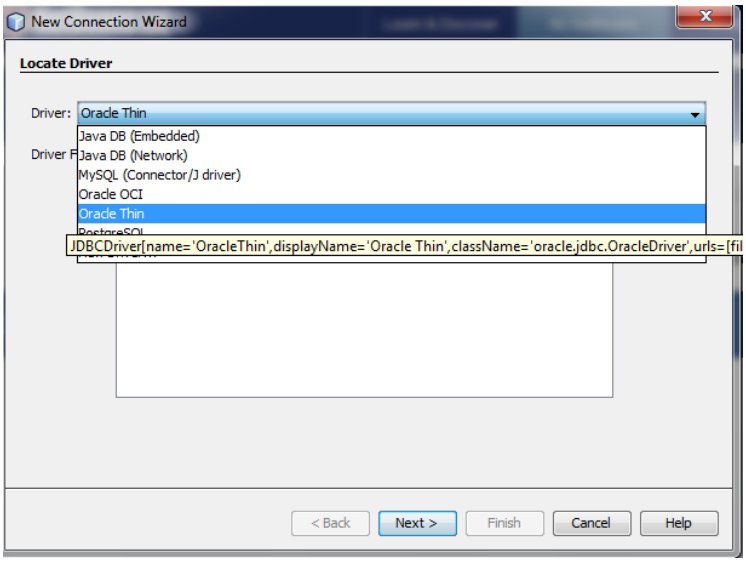


Conexión a la Base de Datos en Oracle

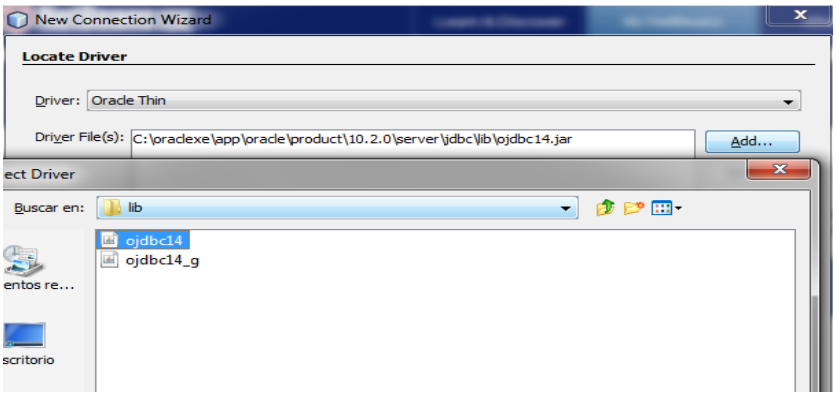
1. En servicios, hacemos una nueva conexión.



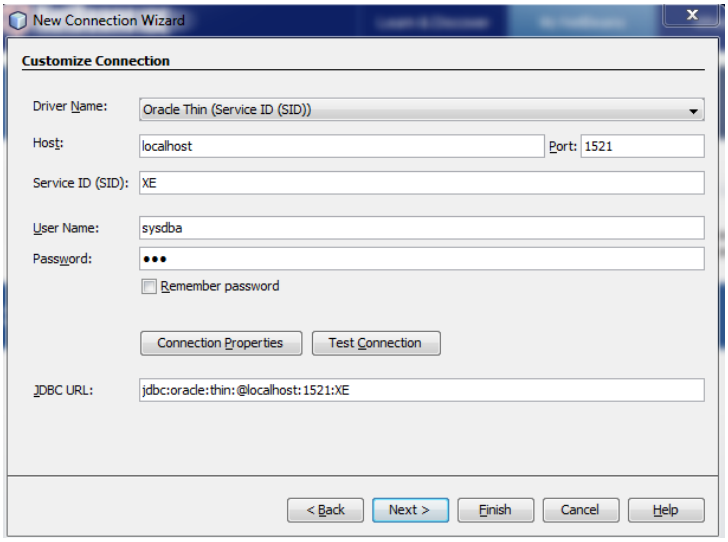
2. Seleccionamos Oracle Thin para poder poner nuestro “ojdbc”.



3. Seleccionamos “add” y buscamos en nuestra carpeta de Oracle donde tenemos guardado el “ojdbc”.



4. Iniciamos la base en Oracle y ponemos usuario y contraseña en Netbeans, probamos conexión.



5. Cambiamos el nombre y le damos terminar.