



**UNIVERSIDAD CATÓLICA LOS ÁNGELES
CHIMBOTE**

**FACULTAD D EINGENIERIA
ESCUELA PROFESIONAL DE INGENIERIA DE
SISTEMAS**

CURSO: INGENIERIA DE SOFTWARE II

DOCENTE: GOMEZ HURTADO HEBER

ALUMNO: ACERO ACOSTA RONALD

CICLO: VI

CHIBMOTE – 2017

ACTIVIDAD N° 5

Luego de haber socializado y mejorados las GUI proceda a implementarlos. El código fuente que genere deberá alojarlo en la plataforma GITHUB.

GUI_PEDIDO – CODIGO FUENTE NETBEANS

Clase Conexión:

```
package Restaurant;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.Statement;

public class Conexion {
    static Connection conexion = null;
    public static Connection getConexion(){
String url =
"jdbc:sqlserver://localhost\\MSSQLSERVER:1433;databaseName=Restaurante";
        //conexion del driver
        try{
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

        }catch (ClassNotFoundException e){
            JOptionPane.showMessageDialog(null, "No se pudo establecer la Conexion"
+e.getMessage(),
            "Error de Conexion",JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

    }
    //conexion del usuario y contraseña
    try{
        conexion = DriverManager.getConnection(url,"ronald","123456");
        System.out.print("Conexion Establecida");
    }catch (SQLException e){
        System.out.print("Conexion Erronea");
        JOptionPane.showMessageDialog(null, "Error: " +e.getMessage(),"Error de
Conexion",
        JOptionPane.ERROR_MESSAGE);
    }
    return conexion;
}

//metodo de consulta directamente a la base de datos
public static ResultSet Consulta(String consulta){
    Connection con = getConexion();
    Statement declara;
    try{
        declara=con.createStatement();
        ResultSet respuesta = declara.executeQuery(consulta);
        return respuesta;

    }catch (SQLException e){
        JOptionPane.showMessageDialog(null, "Error" +e.getMessage(),
            "Error de Conexion",JOptionPane.ERROR_MESSAGE);
    }
    return null;

}

}

```

Clase Procedimientos:

```
package Restaurant;

import java.sql.CallableStatement;
import java.sql.SQLException;

public class Procedimientos {

    public static void GuardarPedidos(String a, String b, String c) throws SQLException{

        CallableStatement entrada=Conexion.getConnection().prepareCall("{ call GuardarPedido(?,?,?) }");

        entrada.setString(1, a);
        entrada.setString(2, b);
        entrada.setString(3, c);
        entrada.execute();

    }

}
```

JFrame Form – frmPedidos:

```
package Restaurant;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;
```

```

public class frmPedidos extends javax.swing.JFrame {

    static ResultSet rs;

    PreparedStatement ps;

    public frmPedidos() {
        initComponents();
        CargarPedidos();
    }

    public void CargarPedidos() {
        DefaultTableModel modelo = (DefaultTableModel) jTablePed.getModel();
        modelo.setRowCount(0);

        rs = Conexion.Consulta("select * from pedidos ORDER BY Numero_Mesa");
        try {
            while (rs.next()) {
                Vector v = new Vector();

                v.add(rs.getInt(1));
                v.add(rs.getString(2));
                v.add(rs.getInt(3));

                modelo.addRow(v);

                jTablePed.setModel(modelo);
            }
        } catch (SQLException e) {
        }
    }

    @SuppressWarnings("unchecked")

    private void btnAceptarActionPerformed(java.awt.event.ActionEvent evt) {

        DefaultTableModel modelo = (DefaultTableModel) jTablePed.getModel();

        Object[] fila = new Object[3];
    }

```

```

fila[0] = txtnm.getText();
fila[1] = txtcon.getText();
fila[2] = txtcod.getText();
modelo.addRow(fila);
jTablePed.setModel(modelo);

if (txtnm.getText().isEmpty() || txtcon.getText().isEmpty() || txtcod.getText().isEmpty()) {
    JOptionPane.showMessageDialog(this, "Ingrese los Datos");
    txtnm.setText("");
    txtcon.setText("");
    txtcod.setText("");
    txtnm.requestFocus();

} else {
    try {
        Procedimientos.GuardarPedidos(txtnm.getText(), txtcon.getText(), txtcod.getText());
        txtnm.setText("");
        txtcon.setText("");
        txtcod.setText("");
        txtnm.requestFocus();
        JOptionPane.showMessageDialog(this, "Pedido Guardado Correctamente");
    } catch (SQLException e) {

    }

}

}

private void btnFinActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */


    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new frmPedidos().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btnAceptar;
private javax.swing.JButton btnFin;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTablePed;
private javax.swing.JTextField txtcod;
private javax.swing.JTextField txtcon;
private javax.swing.JTextField txtnm;

// End of variables declaration
}

```

FORMULARIO

 — □ ×

PEDIDOS

Numero de Mesa :

Consumicion :

Codigo de Consumicion :

PEDIDOS

Numero de Mesa	Consumicion	Codigo de Consumicion