

MANUAL TÉCNICO

APP JAVA Y MYSQL

Pedro Samuel Hernández Pinzón

1990-22-3323

Programación II

Segundo Examen Parcial

Login

```
private void ingresarActionPerformed(java.awt.event.ActionEvent evt) {  
    String usu = usuario.getText();  
    String pwr = contrasena.getText();  
  
    if (usu.isEmpty() || pwr.isEmpty()) {  
        JOptionPane.showMessageDialog(null, "Algun campo esta vacio");  
    } else {  
        if (usu.equals("usuariol") && pwr.equals("12345")) {  
            JOptionPane.showMessageDialog(null, "Bienvenido");  
            Table t = new Table();  
            t.setVisible(true);  
            this.dispose();  
        } else {  
            JOptionPane.showMessageDialog(null, "Su usuario o contraseña son incorrectos");  
        }  
    }  
}  
  
private void salirActionPerformed(java.awt.event.ActionEvent evt) {  
    JOptionPane.showMessageDialog(null, "¡Adiós!");  
    System.exit(0);  
}
```

Ingresar

```
void Ingresar() {  
    String cod = cod_in.getText();  
    String nom = nom_in.getText();  
    String precio = precio_in.getText();  
    String can = can_in.getText();  
    String fec = fec_in.getText();  
  
    try{  
        if(cod.equals("") || nom.equals("") || precio.equals("") || can.equals("") || fec.equals(""))  
            JOptionPane.showMessageDialog(null, "Faltan ingresar datos");  
        limpiarTabla();  
    } else {  
        String sql = "insert into producto(codigoProducto, nombreProducto, precioUnitario, cantidad, fechaIngreso) values(?,?,?,?,?)";  
        conet=con1.getConnection();  
        st = conet.createStatement();  
        st.executeUpdate(sql);  
        JOptionPane.showMessageDialog(null, "Nuevo Producto Ingresado");  
        limpiarTabla();  
    }  
} catch (Exception e) {  
  
}  
}
```

Inventario

```
void consultar() {  
    String sql = "select*from producto";  
  
    try {  
        conet=con1.getConnection();  
        st = conet.createStatement();  
        rs = st.executeQuery(sql);  
        Object[] producto = new Object[5];  
        modelo = (DefaultTableModel) Tabla.getModel();  
        while (rs.next()){  
            producto [0] = rs.getString("codigoProducto");  
            producto [1] = rs.getString("nombreProducto");  
            producto [2] = rs.getFloat("precioUnitario");  
            producto [3] = rs.getInt("cantidadProducto");  
            producto [4] = rs.getDate("fechaIngreso");  
  
            modelo.addRow(producto);  
        }  
        Tabla.setModel(modelo);  
    }catch (Exception e) {  
  
    }  
}
```

Eliminar

```
void eliminar() {  
    int fila=Tabla.getSelectedRow();  
  
    try{  
        if (fila<0){  
            JOptionPane.showMessageDialog(null, "Cliente no seleccionado");  
            limpiarTabla();  
        }else{  
            String cod = cod_in.getText();  
            String sql = "delete from producto where codigoProducto="+cod;  
            conet=con1.getConnection();  
            st = conet.createStatement();  
            st.executeUpdate(sql);  
            JOptionPane.showMessageDialog(null, "Cliente eliminado");  
            limpiarTabla();  
        }  
    }catch (Exception e){  
  
    }  
}
```


Buscar

```
void buscar() {
String codigoONombre = jTextField1.getText();
String sql = "SELECT * FROM producto WHERE codigoProducto = ? OR nombreProducto = ?";

if (codigoONombre.isEmpty()) {
    limpiarTabla2();
    consultar();
} else {
try {
    conet = con1.getConnection();
    PreparedStatement statement = conet.prepareStatement(sql);
    statement.setString(1, codigoONombre);
    statement.setString(2, codigoONombre);
    ResultSet rs = statement.executeQuery();

    Object[] producto = new Object[5];
    modelo = (DefaultTableModel) Tabla.getModel();
    modelo.setRowCount(0); // Limpiar la tabla antes de agregar nuevos datos

    while (rs.next()) {
        producto[0] = rs.getString("codigoProducto");
        producto[1] = rs.getString("nombreProducto");
        producto[2] = rs.getFloat("precioUnitario");
        producto[3] = rs.getInt("cantidadProducto");
        producto[4] = rs.getDate("fechaIngreso");

        modelo.addRow(producto);
    }

    Tabla.setModel(modelo);
}
```

Modificar

```
void modificar() {
    String cod = cod_in.getText();
    String nom = nom_in.getText();
    String precio = precio_in.getText();
    String can = can_in.getText();
    String fec = fec_in.getText();

    try{
        if(cod.equals("") || nom.equals("") || precio.equals("") || can.equals("") || fec.equals(""))
            JOptionPane.showMessageDialog(null, "Faltan ingresar datos");
        limpiarTabla();
    } else {
        String sql = "Update producto set codigoProducto='"+cod+"', nombreProducto='"+nom+"',
        conet=con1.getConnection();
        st = conet.createStatement();
        st.executeUpdate(sql);
        JOptionPane.showMessageDialog(null, "Datos de cliente modificados");
        limpiarTabla();
    }
} catch (Exception e) {

}
}
```

Reportes

```
private void generarActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        Conexion con = new Conexion();  
        Connection conn = con.getConnection();  
  
        JasperReport reporte = null;  
        String path = "src\\Reportes\\reporte.jasper";  
        reporte = (JasperReport) JRLoader.loadObjectFromFile(path);  
  
        JasperPrint jprint = JasperFillManager.fillReport(reporte, null, conn);  
  
        JasperViewer view = new JasperViewer(jprint, false);  
  
        view.setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
  
        view.setVisible(true);  
    } catch (JRException ex) {  
        Logger.getLogger(Table.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```


Historial Compras

```
void consultar() {  
    String sql = "select*from historial_compras";  
  
    try {  
        conet=con1.getConnection();  
        st = conet.createStatement();  
        rs = st.executeQuery(sql);  
        Object[] historial_compras = new Object[5];  
        modelo = (DefaultTableModel) Tabla.getModel();  
        while (rs.next()){  
            historial_compras [0] = rs.getString("codigoproducto");  
            historial_compras [1] = rs.getString("nombreproducto");  
            historial_compras [2] = rs.getFloat("preciounitario");  
            historial_compras [3] = rs.getInt("cantidadproducto");  
            historial_compras [4] = rs.getDate("fechaingreso");  
            modelo.addRow(historial_compras);  
        }  
        Tabla.setModel(modelo);  
    }catch (Exception e) {  
    }  
}
```

App En Java y MySQL

Mi app consiste en una venta de electrodomésticos como estufas, refrigeradoras, batidoras, lavadoras, etc. Como expuse anteriormente la aplicación cuenta con diferentes opciones con las que podemos interactuar dentro de la app de java. Podemos realizar múltiples actividades, actividades muy básicas que son parte del CRUD y todas funcionan de manera exitosa y como se esperaba, tiene un diseño sencillo y es muy funcional y no presenta ningun error.

Nos vemos pronto
¡Gracias por la atención!

