

## CÓDIGO: Mostrar Datos

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

public class MostrarDatos {
    public static void mostrarDatos(DefaultTableModel model) {
        String sql = "SELECT fruta, precio FROM frutas WHERE precio IS NOT NULL";
        try (Connection conn = ConexionBD.getConnection());
            PreparedStatement pstmt = conn.prepareStatement(sql);
            ResultSet rs = pstmt.executeQuery() {
                model.setRowCount(0); // Limpiar la tabla antes de agregar nuevos datos
                while (rs.next()) {
                    Object[] rowData = {
                        rs.getString("fruta"),
                        rs.getDouble("precio")
                    };
                    model.addRow(rowData);
                }
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(null, "Error al recuperar datos de la base de
                datos: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}
```

## CÓDIGO: Calcular la Relación de Frutas

```
private void calcularRelacion() {  
    // Obtener el precio del plátano del JTextField  
    double precioPlatano = Double.parseDouble(jTextField1.getText());  
  
    // Variables para la conexión a la base de datos  
    String url = "jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";  
    String user = "kevin_123";  
    String password = "8baO6G0RmY0HjY2PqSrScxaR3p4aI3We";  
  
    // Consulta SQL para obtener los precios de las otras frutas  
    String sql = "SELECT fruta, precio FROM frutas WHERE fruta <> 'Plátano' AND precio  
    >= ? ";  
  
    try (Connection conn = DriverManager.getConnection(url, user, password);  
        PreparedStatement pstmt = conn.prepareStatement(sql)) {  
  
        // Establecer el parámetro de precio del plátano en la consulta SQL  
        pstmt.setDouble(1, precioPlatano);  
  
        try (ResultSet rs = pstmt.executeQuery()) {  
            // Limpiar la tabla antes de agregar nuevos datos  
            DefaultTableModel model = (DefaultTableModel) jTable2.getModel();  
            model.setRowCount(0);  
  
            // Iterar sobre los resultados y calcular la relación con el precio del plátano  
            while (rs.next()) {  
                String fruta = rs.getString("fruta");  
                double precioFruta = rs.getDouble("precio");  
                double relacion = precioFruta / precioPlatano;
```

```

        int relacion_entero = Math.round((float) relacion);

        model.addRow(new Object[]{fruta, "vale " + relacion_entero + " veces más"});
    }
}

} catch (SQLException e) {
    JOptionPane.showMessageDialog(this, "Error al recuperar datos de la base de datos: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
}
}

```

## CÓDIGO: Cargar Datos de la tabla escoger la fruta para las interacciones

```

private void cargarFrutasEnComboBox() {
    try {
        Connection con = DriverManager.getConnection("jdbc:postgresql://oregon-
postgres.render.com:5432/sisinfo2db", "kevin_123",
"8baO6G0RmY0HjY2PqSrScxaR3p4aI3We");

        String query = "SELECT fruta FROM frutas where 1=1";
        PreparedStatement pst = con.prepareStatement(query);
        ResultSet rs = pst.executeQuery();

        while (rs.next()) {
            String nombreFruta = rs.getString("fruta");
            // Imprime el nombre de la fruta
            jComboBox1.addItem(nombreFruta);
        }
        con.close();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

```

## CÓDIGO: Actualizar para ordenar los elementos.

```
import javax.swing.table.DefaultTableModel;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class Actualizar {
    public static void actualizarTabla(DefaultTableModel model) {
        model.setRowCount(0); // Limpiar la tabla

        try {
            Connection con = ConexionBD.getConnection();
            String query = "SELECT fruta, precio FROM frutas ORDER BY precio";
            PreparedStatement pst = con.prepareStatement(query);
            ResultSet rs = pst.executeQuery();
            while (rs.next()) {
                Object[] row = {
                    rs.getString("fruta"),
                    rs.getDouble("precio")
                };
                model.addRow(row);
            }
            ConexionBD.close(con, pst, rs);
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

## CÓDIGO: Eliminar los datos

```
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
public class EliminarDatos {
    public static void eliminarDatos(DefaultTableModel model, int selectedRow) {
        if (selectedRow != -1) {
            try (Connection con = ConexionBD.getConnection()) {
                String query = "DELETE FROM frutas WHERE fruta = ? AND precio = ?";
                PreparedStatement pst = con.prepareStatement(query);
                pst.setString(1, (String) model.getValueAt(selectedRow, 0)); // Suponiendo que
la primera columna es el nombre de la fruta
                pst.setDouble(2, (Double) model.getValueAt(selectedRow, 1)); // Suponiendo
que la segunda columna es el precio de la fruta
                pst.executeUpdate();
                model.removeRow(selectedRow);
                JOptionPane.showMessageDialog(null, "Fruta eliminada correctamente.");
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(null, "Error al eliminar la fruta: " +
ex.getMessage());
            }
        } else {
            JOptionPane.showMessageDialog(null, "Por favor, selecciona una fila para
eliminar.");
        }
    }
}
```

## CÓDIGO: Modificar

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

public class Modificar {

    public static void modificarFruta(DefaultTableModel model, int filaSeleccionada, String
nuevoNombreFruta, double nuevoPrecioFruta) {

        String nombreFrutaSeleccionada = (String) model.getValueAt(filaSeleccionada, 0);

        // Verificar si el nombre de la fruta seleccionada es igual al nuevo nombre de la fruta
        if (!nombreFrutaSeleccionada.equals(nuevoNombreFruta)) {

            try {

                Connection con = ConexionBD.getConnection();

                // Consulta para verificar si el nuevo nombre de fruta ya existe
                String queryVerificar = "SELECT COUNT(*) FROM frutas WHERE fruta = ?";
                PreparedStatement pstVerificar = con.prepareStatement(queryVerificar);
                pstVerificar.setString(1, nuevoNombreFruta);
                ResultSet rs = pstVerificar.executeQuery();

                // Verificar si hay algún resultado en la consulta
                rs.next();
                int count = rs.getInt(1);
                if (count > 0) {

                    JOptionPane.showMessageDialog(null, "El nombre de la fruta ya existe en la
base de datos.");
```

```

        return;
    }
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, "Error al verificar la fruta en la base de
datos: " + ex.getMessage());
    return;
}
}

// Si el nombre de la fruta seleccionada es igual al nuevo nombre de la fruta
// o si el nuevo nombre de la fruta no existe en la base de datos, procedemos con la
modificación
try {
    Connection con = ConexionBD.getConnection();

    // Consulta para actualizar la fruta
    String queryUpdate = "UPDATE frutas SET fruta = ?, precio = ? WHERE fruta = ?";
    PreparedStatement pst = con.prepareStatement(queryUpdate);
    pst.setString(1, nuevoNombreFruta);
    pst.setDouble(2, nuevoPrecioFruta);
    pst.setString(3, nombreFrutaSeleccionada); // Utilizamos el nombre de la fruta
seleccionada como criterio de búsqueda
    pst.executeUpdate();

    // Actualizar la tabla mostrando la fruta modificada
    model.setValueAt(nuevoNombreFruta, filaSeleccionada, 0); // Suponiendo que el
nombre de la fruta está en la primera columna
    model.setValueAt(nuevoPrecioFruta, filaSeleccionada, 1); // Suponiendo que el
precio de la fruta está en la segunda columna

    JOptionPane.showMessageDialog(null, "Fruta modificada correctamente.");
} catch (SQLException ex) {

```

```

        JOptionPane.showMessageDialog(null, "Error al modificar la fruta: " +
ex.getMessage());
    }
}
}

```

## CÓDIGO: Para los bonotes de NetBeans

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    MostrarDatos.mostrarDatos((DefaultTableModel) jTable1.getModel());
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    calcularRelacion();
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    int selectedRow = jTable1.getSelectedRow();
    EliminarDatos.eliminarDatos((DefaultTableModel) jTable1.getModel(), selectedRow);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    String nombreFruta = jTextField2.getText();
    double precioFruta = Double.parseDouble(jTextField3.getText());

    Agregar.agregarFruta((DefaultTableModel) jTable1.getModel(), nombreFruta,
precioFruta);
}

```



```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    int filaSeleccionada = jTable1.getSelectedRow();

    if (filaSeleccionada == -1) {
        JOptionPane.showMessageDialog(null, "Por favor, selecciona una fila para
        modificar.");
        return;
    }

    String nuevoNombreFruta = jTextField2.getText();
    double nuevoPrecioFruta;

    try {
        nuevoPrecioFruta = Double.parseDouble(jTextField3.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(null, "Por favor, ingresa un valor numérico válido
        para el precio.");
        return;
    }

    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    Modificar.modificarFruta(model, filaSeleccionada, nuevoNombreFruta,
    nuevoPrecioFruta);
}

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
    cargarFrutasEnComboBox();
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    Actualizar.actualizarTabla((DefaultTableModel) jTable1.getModel());
}

```

```
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    guardarCambios();
}
```

COMMMIT REALIZADOS:

Commits on Apr 8, 2024		
Merge branch 'main' of <a href="https://github.com/SisInfo2/SprintZero">https://github.com/SisInfo2/SprintZero</a>	d84e805	
KevinOmega committed 1 minute ago		
commit Kevin C.	caef954	
KevinYrl committed 2 minutes ago		
agregando un nuevo Frame	d3926e7	
KevinOmega committed 2 minutes ago		
Merge branch 'main' of <a href="https://github.com/SisInfo2/SprintZero">https://github.com/SisInfo2/SprintZero</a>	dc644c2	
KevinYrl committed 3 minutes ago		
commit Kevin C.	229702d	
KevinYrl committed 6 minutes ago		
Initial commit	7e356e7	
KevinOmega committed 13 minutes ago		
conexion a la base de datos	6a3c716	
KevinYrl committed 1 hour ago		
Commits on Apr 6, 2024		
solucion conflicto	35e4aab	
KevinYrl committed 2 days ago		
avance del proyecto es un exito	245d650	
KevinYrl committed 2 days ago		
Update Test	371c4af	

<b>solucion conflicto</b>	35e4aab	<>
KevinYrI committed 2 days ago		
<b>avance del proyecto es un exito</b>	245d650	<>
KevinYrI committed 2 days ago		
<b>Update Test</b>	371c4af	<>
KevinOmega committed 2 days ago	Verified	
<b>Create Test</b>	c2c429f	<>
KevinOmega committed 2 days ago	Verified	
<b>inicializando el proyecto</b>	1f18f79	<>
KevinYrI committed 2 days ago		
<b>modificacion commit</b>	fb69241	<>
KevinYrI committed 2 days ago		
<b>commit 2</b>	28b6cef	<>
KevinYrI committed 2 days ago		
<b>Update Readme</b>	b2a2f00	<>
KevinOmega committed 2 days ago	Verified	
<b>Create NewFile</b>	5ac2786	<>
KevinOmega committed 2 days ago	Verified	
<b>Create Readme</b>	d8c6f8e	<>
KevinOmega committed 2 days ago	Verified	

## CONEXIÓN A LA BASE DE DATOS:

pgAdmin 4

File Object Tools Help

Object Explorer

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)
  - > public
    - > Aggregates
    - > Collations
    - > Domains
    - > FTS Configurations
    - > FTS Dictionaries
    - > FTS Parsers
    - > FTS Templates
    - > Foreign Tables
    - > Functions
    - > Materialized Views
    - > Operators
    - > Procedures
    - > Sequences
    - > Tables (1)
      - > frutas
        - > Columns (3)
          - fruta
          - precio
          - id\_fruta
        - > Constraints
        - > Indexes
        - > RLS Policies
        - > Rules
        - > Triggers

public.frutas/fruta... x public.frutas/frutas/postgres@PostgreSQL 16 x

public.frutas/frutas/postgres@PostgreSQL 16

Query Query History

```

1 SELECT * FROM public.frutas
2 ORDER BY id_fruta ASC

```

Scratch Pad x

Data Output Messages Notifications

	fruta character varying (100)	precio numeric (10,2)	id_fruta [PK] integer
1	platano	0.20	1
2	naranja	0.40	2
3	durazno	0.80	3
4	kiwi	1.30	5
5	papaya	3.00	6
6	granada	4.50	7
7	coco	7.00	8
8	chirimoya	10.00	9
9	sandia	25.00	10
10	nara	5.00	18

Total rows: 11 of 11 Query complete 00:00:00.082 Ln 1, Col 1

```

private static final String URL = "jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";
private static final String USER = "kevin_123";
private static final String PASSWORD = "8ba06G0RmY0HjY2PqSrScxaR3p4aI3We";

```

# ESTADO DEL GITHUB(MAIN) CON LOS COMMITS REALIZADOS

main

1 Branch

0 Tags

Go to file

t

Add file

> Code

KevinYrl

Merge branch 'main' of <https://github.com/SisInfo2/SprintZero>

4dd0e12 · now

20 Commits

<div>ConexionBD</div>	Commit Inicial KC	1 minute ago
<div>KevinFabianChavez</div>	Commit Inicial KC	1 minute ago
<div>KevinHuayllas</div>	cambiar el titulo al frame principal	now
<div>.gitignore</div>	agregando un nuevo Frame	13 minutes ago
<div>NewFile</div>	Create NewFile	2 days ago
<div>Readme</div>	modificacion commit	2 days ago

README