Deber 2 : ERNESTO ROJAS

```
import javax.swing.*;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;
import java.util.Vector;

public class Main {
    private Connection conn;
    private JFrame frame;
    private JComboBox<String> comboBox;
    private JTable table;
    private DefaultTableModel tableModel;
    private JProgressBar progressBar;

    public Main() {
        // Establecer conexión a la base de datos PostgreSQL
        connectDB();

        // Crear la interfaz gráfica
        createGUI();
    }

    private void connectDB() {
        try {
            String url = "jdbc:postgresql://localhost:5432/formula1";
            String user = "postgres";
            String password = "1234";
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("Conexión establecida con PostgreSQL.");
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

private void createGUI() {
    frame = new JFrame("Tabla de Conductores y Constructores por Año de Carrera");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(1000, 600);

    // Combo box para seleccionar el año de carrera
    comboBox = new JComboBox<>();
    populateComboBox();
    comboBox.addActionListener(e -> {
        // Cuando se seleccione un año, actualizar la tabla de conductores y constructores
        updateTableInBackground();
    });

    // Barra de progreso para mostrar mientras se carga la tabla
    progressBar = new JProgressBar();
    progressBar.setStringPainted(true);

    // Tabla para mostrar los datos de conductores y constructores
    tableModel = new DefaultTableModel();
    table = new JTable(tableModel);
    JScrollPane scrollPane = new JScrollPane(table);

    // Centrar el contenido de las celdas
    DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment(JLabel.CENTER);
    table.setDefaultRenderer(Object.class, centerRenderer);
```

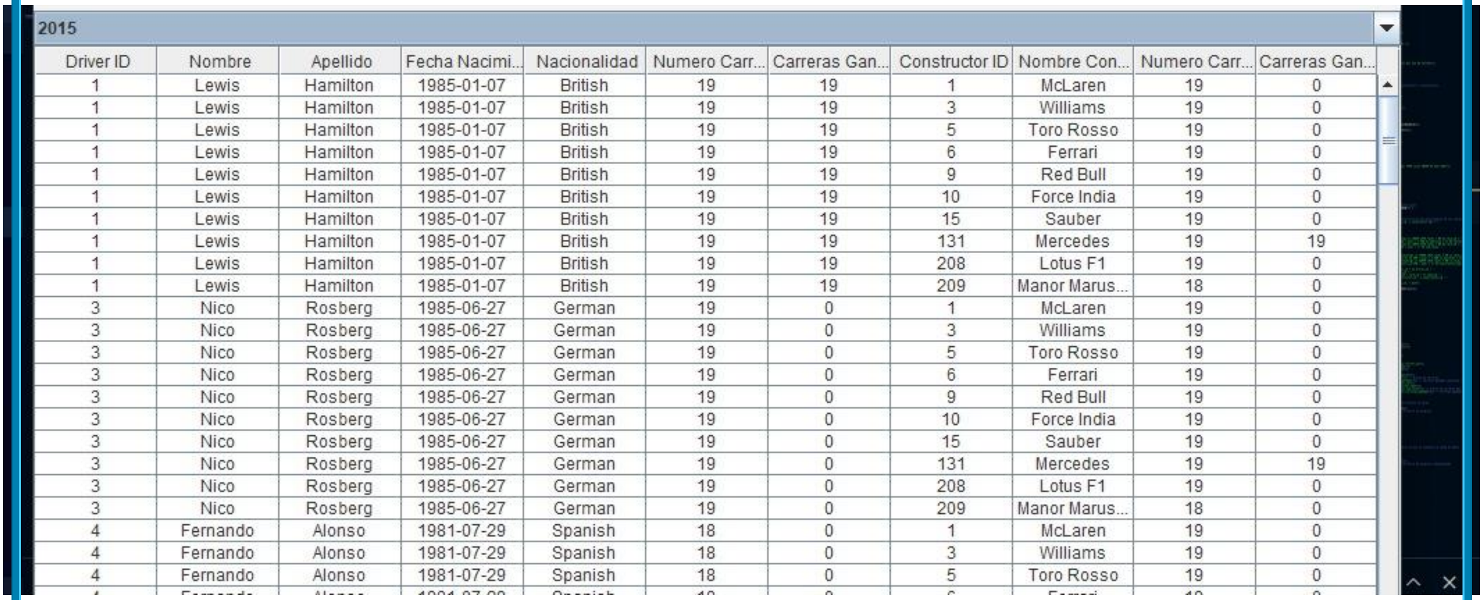
```
        frame.getContentPane().setLayout(new BorderLayout());
        frame.getContentPane().add(comboBox, BorderLayout.NORTH);
        frame.getContentPane().add(progressBar, BorderLayout.SOUTH);
        frame.getContentPane().add(scrollPane, BorderLayout.CENTER);

        frame.setVisible(true);
    }

    private void populateComboBox() {
        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT DISTINCT year FROM races ORDER BY year DESC");
            while (rs.next()) {
                comboBox.addItem(rs.getString("year"));
            }
            rs.close();
            stmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void updateTableInBackground() {
        String selectedYear = (String) comboBox.getSelectedItem();
        if (selectedYear != null) {
            // Crear un SwingWorker para ejecutar la consulta en segundo plano
            SwingWorker<Void, Void> worker = new SwingWorker<Void, Void>() {
                @Override
                protected Void doInBackground() throws Exception {
```

```
        try {
            // Consulta para obtener los conductores y constructores que participaron en las carreras del año seleccionado
            String query = "SELECT DISTINCT ON (d.driver_id, c.constructor_id) " +
                    "d.driver_id, " +
                    "d.forename, " +
                    "d.surname, " +
                    "d.dob, " +
                    "d.nationality, " +
                    "(SELECT COUNT(*) FROM driver_standings ds INNER JOIN races r ON ds.race_id = r.race_id " +
                    "WHERE ds.driver_id = d.driver_id AND r.year = ? AND ds.position = 1) AS carreras_ganadas, " +
                    "(SELECT COUNT(*) FROM driver_standings ds INNER JOIN races r ON ds.race_id = r.race_id " +
                    "WHERE ds.driver_id = d.driver_id AND r.year = ?) AS num_races, " +
                    "c.constructor_id, " +
                    "c.name AS constructor_name, " +
                    "(SELECT COUNT(*) FROM constructor_standings cs INNER JOIN races r ON cs.race_id = r.race_id " +
                    "WHERE cs.constructor_id = c.constructor_id AND r.year = ? AND cs.position = 1) AS constructor_carreras_ganadas, " +
                    "(SELECT COUNT(*) FROM constructor_standings cs INNER JOIN races r ON cs.race_id = r.race_id " +
                    "WHERE cs.constructor_id = c.constructor_id AND r.year = ?) AS constructor_num_races " +
                    "FROM drivers d " +
                    "JOIN driver_standings ds ON d.driver_id = ds.driver_id " +
                    "JOIN races r ON ds.race_id = r.race_id " +
                    "LEFT JOIN constructor_standings cs ON r.race_id = cs.race_id " +
                    "LEFT JOIN constructors c ON cs.constructor_id = c.constructor_id " +
                    "WHERE r.year = ? " +
                    "ORDER BY d.driver_id, c.constructor_id, r.date";
```

```
PreparedStatement pstmt = conn.prepareStatement(query);
int year = Integer.parseInt(selectedYear);
pstmt.setInt(1, year);
pstmt.setInt(2, year);
pstmt.setInt(3, year);
pstmt.setInt(4, year);
pstmt.setInt(5, year);
ResultSet rs = pstmt.executeQuery();

// Obtener columnas
Vector<String> columnNames = new Vector<>();
columnNames.add("Driver ID");
columnNames.add("Nombre");
columnNames.add("Apellido");
columnNames.add("Fecha Nacimiento");
columnNames.add("Nacionalidad");
columnNames.add("Numero Carreras");
columnNames.add("Carreras Ganadas");
columnNames.add("Constructor ID");
columnNames.add("Nombre Constructor");
columnNames.add("Numero Carreras Constructor");
columnNames.add("Carreras Ganadas Constructor");

// Obtener filas
Vector<Vector<Object>> data = new Vector<>();
while (rs.next()) {
    Vector<Object> row = new Vector<>();
    row.add(rs.getInt("driver_id"));
    row.add(rs.getString("forename"));
    row.add(rs.getString("surname"));
    row.add(rs.getDate("dob"));
```

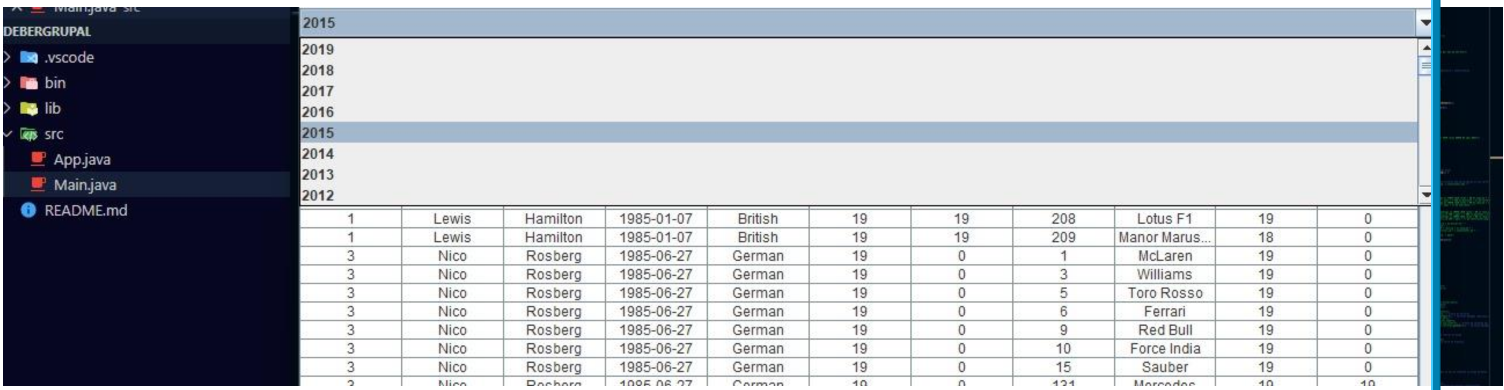
```
                            row.add(rs.getString("nationality"));
                            row.add(rs.getInt("num_races")); // Número de carreras
                            row.add(rs.getInt("carreras_ganadas")); // Carreras ganadas (position = 1)
                            row.add(rs.getInt("constructor_id"));
                            row.add(rs.getString("constructor_name"));
                            row.add(rs.getInt("constructor_num_races")); // Número de carreras del
constructor
                            row.add(rs.getInt("constructor_carreras_ganadas")); // Carreras ganadas del
constructor (position = 1)
                            data.add(row);
                        }

                        // Actualizar modelo de la tabla en el hilo de eventos de Swing
                        SwingUtilities.invokeLater(() -> {
                            tableModel.setDataVector(data, columnNames);
                            progressBar.setValue(100); // Completa la barra de progreso
                        });

                        rs.close();
                        pstmt.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                    return null;
                }

                @Override
                protected void done() {
                    // Aquí puedes realizar acciones adicionales después de que se completa la carga de
datos
                }
            };
```

```
      // Iniciar el SwingWorker y mostrar la barra de progreso
      progressBar.setValue(0); // Reiniciar la barra de progreso
      progressBar.setIndeterminate(true); // Mostrar una barra de progreso indeterminada
      worker.execute();
    }
  }

  public static void main(String[] args) {
    SwingUtilities.invokeLater(Main::new);
  }
```



Driver ID	Nombre	Apellido	Fecha Nacimi...	Nacionalidad	Numero Carr...	Carreras Gan...	Constructor ID	Nombre Con...	Numero Carr...	Carreras Gan...
1	Lewis	Hamilton	1985-01-07	British	19	19	1	McLaren	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	3	Williams	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	5	Toro Rosso	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	6	Ferrari	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	9	Red Bull	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	10	Force India	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	15	Sauber	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	131	Mercedes	19	19
1	Lewis	Hamilton	1985-01-07	British	19	19	208	Lotus F1	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	209	Manor Marus...	18	0
3	Nico	Rosberg	1985-06-27	German	19	0	1	McLaren	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	3	Williams	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	5	Toro Rosso	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	6	Ferrari	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	9	Red Bull	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	10	Force India	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	15	Sauber	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	131	Mercedes	19	19
3	Nico	Rosberg	1985-06-27	German	19	0	208	Lotus F1	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	209	Manor Marus...	18	0
4	Fernando	Alonso	1981-07-29	Spanish	18	0	1	McLaren	19	0
4	Fernando	Alonso	1981-07-29	Spanish	18	0	3	Williams	19	0
4	Fernando	Alonso	1981-07-29	Spanish	18	0	5	Toro Rosso	19	0
4	Fernando	Alonso	1981-07-29	Spanish	18	0	6	Ferrari	19	0

```
}
```



1	Lewis	Hamilton	1985-01-07	British	19	19	208	Lotus F1	19	0
1	Lewis	Hamilton	1985-01-07	British	19	19	209	Manor Marus...	18	0
3	Nico	Rosberg	1985-06-27	German	19	0	1	McLaren	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	3	Williams	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	5	Toro Rosso	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	6	Ferrari	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	9	Red Bull	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	10	Force India	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	15	Sauber	19	0
3	Nico	Rosberg	1985-06-27	German	19	0	131	Mercedes	19	19