UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE INGENIERIA INDUSTRIAL

INGENIERIA EN SISTEMAS DE INFORMACION

MATERIA:

PROGRAMACION

ESTUDIANTE:

ROSERO LOPEZ MARYLIN NICOLLE

TEMA

TAREA INDIVIDUAL

CODIGO

```java
package InterfazGrafica;

import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableCellRenderer;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;
import java.util.Vector;

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

    public InterfazGrafica() {
        // 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 = "password";
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("Conexión establecida con
PostgreSQL.");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

```java
    private void createGUI() {
        frame = new JFrame("Tabla de Drivers por Año de Carrera");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(950, 450);

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

        // Tabla para mostrar los datos de corredores y carreras
        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().add(comboBox, BorderLayout.NORTH);
        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) {
```

```java
                    e.printStackTrace();
        }
    }

    private void updateTable() {
        try {
            String selectedYear = (String)
comboBox.getSelectedItem();
            if (selectedYear != null) {
                // Consulta para obtener los datos necesarios
                String query = "SELECT d.forename || ' ' || d.surname
AS driver_name, " +
                    "(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 wins, " +
                    "(SELECT SUM(points) 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 total_points, " +
                    "ds.position AS rank " +
                    "FROM drivers d " +
                    "JOIN driver_standings ds ON d.driver_id =
ds.driver_id " +
                    "JOIN races r ON ds.race_id = r.race_id " +
                    "WHERE r.year = ? " +
                    "ORDER BY ds.position ASC";

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

                // Obtener columnas
                Vector<String> columnNames = new Vector<>();
                columnNames.add("Nombre del Corredor");
                columnNames.add("Victorias");
                columnNames.add("Puntos Totales");
                columnNames.add("Posición");

                // Obtener filas sin repetir corredores
```

```java
                    Vector<Vector<Object>> data = new Vector<>();
                    Vector<String> processedDrivers = new Vector<>(); //
Para controlar duplicados
                while (rs.next()) {
                    String driverName = rs.getString("driver_name");
                    if (!processedDrivers.contains(driverName)) {
                        Vector<Object> row = new Vector<>();
                        row.add(driverName);
                        row.add(rs.getInt("wins"));
                        row.add(rs.getInt("total_points"));
                        row.add(rs.getInt("rank"));
                        data.add(row);
                        processedDrivers.add(driverName);
                    }
                }

                // Actualizar modelo de la tabla
                tableModel.setDataVector(data, columnNames);

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

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

    public static void main(String[] args) {
        SwingUtilities.invokeLater(InterfazGrafica::new);
    }
}
```
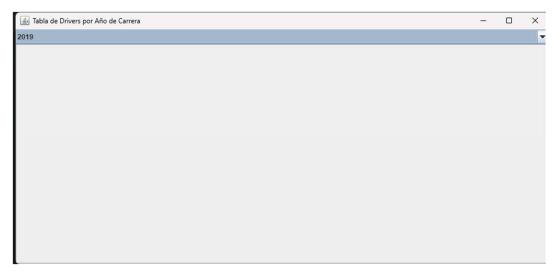
UNIVERSIDAD DE GUAYAQUIL

LA MÁS
GRANDE,
IMPORTANTE
Y GLORIOSA
del Ecuador

Tabla de Drivers por Año de Carrera — 2019

(Tabla vacía)

Tabla de Drivers por Año de Carrera — 2015

Nombre del Corredor	Victorias	Puntos Totales	Posición
Lewis Hamilton	19	3908	1
Nico Rosberg	0	3231	2
Sebastian Vettel	0	2917	2
Kimi Räikkönen	0	1540	4
Valtteri Bottas	0	1422	4
Felipe Massa	0	1374	4
Felipe Nasr	0	341	5
Daniel Ricciardo	0	924	6
Daniil Kvyat	0	816	7
Nico Hülkenberg	0	464	7
Marcus Ericsson	0	128	8
Romain Grosjean	0	515	8
Carlos Sainz	0	198	9
Sergio Pérez	0	546	9
Max Verstappen	0	415	10
Jenson Button	0	118	11
Pastor Maldonado	0	222	13
Fernando Alonso	0	111	15
Roberto Merhi	0	0	17
Kevin Magnussen	0	0	17
Will Stevens	0	0	19
Alexander Rossi	0	0	20

Tabla de Drivers por Año de Carrera — 2012

Lista desplegable:
2019
2018
2017
2016
2015
2014
2013
2012

Nombre del Corredor	Victorias	Puntos Totales	Posición
Kamui Kobayashi	0	630	6
Romain Grosjean	0	1181	7
Felipe Massa	0	803	7
Bruno Senna	0	398	8
Pastor Maldonado	0	530	9
Daniel Ricciardo	0	91	9
Paul di Resta	0	569	10
Michael Schumacher	0	493	10
Nico Hülkenberg	0	484	11
Jean-Éric Vergne	0	136	11
Timo Glock	0	0	14
Charles Pic	0	0	15
Heikki Kovalainen	0	0	18
Vitaly Petrov	0	0	19
Jérôme d'Ambrosio	0	0	21
Narain Karthikeyan	0	0	22
Pedro de la Rosa	0	0	23