

TRAINING 10 – Listar hamburguesas desde la base de datos PostgreSQL

Objetivo

Aprender a conectarse a una base de datos PostgreSQL alojada en AWS usando JDBC.
Crear una consulta SQL que devuelva hamburguesas almacenadas en la base de datos y devolverlas en formato JSON al cliente.

Conocimientos previos recomendados

- Conexión a base de datos con JDBC
 - SQL básico (SELECT)
 - Transformar un `ResultSet` en una lista
 - Cómo devolver JSON desde un servlet
-

Herramientas necesarias

- IntelliJ IDEA
 - PostgreSQL en AWS (ya configurado)
 - Driver JDBC PostgreSQL (`org.postgresql.Driver`)
 - Clase `MotorSQL` ya incluida (ver más abajo)
-

Paso 1: Usar la clase MotorSQL

Copia esta clase en el mismo paquete que `Controller.java` (`com.alumnos.j2ee`)

// --- INICIO DE CÓDIGO ---

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class MotorSQL {

    private Connection conn;

    private static final String URL =
"jdbc:postgresql://netflix-01.cupetnmlumhn.us-east-1.rds.amazonaws.com/postgres";
    private static final String USER = "postgres";
    private static final String PASSWORD = "LUCASLUCAS";

    public void conectar() {
        try {
            Class.forName("org.postgresql.Driver");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }

        Properties properties = new Properties();
        properties.setProperty("user", USER);
        properties.setProperty("password", PASSWORD);
        properties.setProperty("ssl", "false");

        try {
            conn = DriverManager.getConnection(URL, properties);
            if (conn != null) {
                System.out.println("Connected to the database!");
            } else {
                System.out.println("Error al conectar");
            }
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public Connection getConnection() {
        return conn;
    }
}
```

— --- FIN DE CÓDIGO ---

Paso 2: Crear tabla de hamburguesas en PostgreSQL

Lanza este SQL en pgAdmin o cualquier cliente conectado a tu base de datos:

```
CREATE TABLE hamburguesas (  
    id SERIAL PRIMARY KEY,  
    nombre VARCHAR(100),  
    precio DECIMAL  
);  
  
INSERT INTO hamburguesas (nombre, precio) VALUES  
('Cheeseburger', 7.5),  
('Veggie Burger', 6.8),  
('Doble BBQ', 8.9);
```

Paso 3: Crear **Hamburguesa.java** como entidad

```
public class Hamburguesa {  
    private String nombre;  
    private double precio;  
  
    public Hamburguesa(String nombre, double precio) {  
        this.nombre = nombre;  
        this.precio = precio;  
    }  
  
    public String getNombre() { return nombre; }  
    public double getPrecio() { return precio; }  
}
```

Paso 4: Crear **HamburguesaDAO** con SELECT a PostgreSQL

```
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;

public class HamburguesaDAO {

    MotorSQL motor = new MotorSQL();

    public ArrayList<Hamburguesa> getHamburguesas() {
        ArrayList<Hamburguesa> lista = new ArrayList<>();
        try {
            motor.conectar();
            Statement st = motor.getConnection().createStatement();
            ResultSet rs = st.executeQuery("SELECT nombre, precio FROM
hamburguesas");

            while (rs.next()) {
                String nombre = rs.getString("nombre");
                double precio = rs.getDouble("precio");
                lista.add(new Hamburguesa(nombre, precio));
            }
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return lista;
    }
}
```

Paso 5: Crear **HamburguesaService** para devolver JSON

```
package com.alumnos.j2ee;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```

import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

public class HamburguesaService {

    HamburguesaDAO dao = new HamburguesaDAO();

    public void listar(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        response.setContentType("application/json");
        PrintWriter escritor = response.getWriter();

        ArrayList<Hamburguesa> hamburguesas = dao.getHamburguesas();

        String json = "[";
        for (int i = 0; i < hamburguesas.size(); i++) {
            Hamburguesa h = hamburguesas.get(i);
            json += "{\"nombre\":\"" + h.getNombre() + "\",\"precio\":\"" + h.getPrecio() + "\"}";
            if (i < hamburguesas.size() - 1) json += ",";
        }
        json += "]";
        escritor.println(json);
    }
}

```

Paso 6: Añadir enrutamiento a **Controller.java**

Dentro del método `doPost()` o `doGet()`:

```

if (objeto.equalsIgnoreCase("HAMBURGUESA")) {
    HamburguesaService service = new HamburguesaService();
    if (operacion.equalsIgnoreCase("LIST")) {
        service.listar(req, resp);
    }
}

```



Prueba desde Thunder Client

Método: GET

URL:

<http://localhost:8080/TestControllerPostgre/Controller?ACTION=HAMBURGUESA.LIST>

Respuesta esperada:

```
[  
  {"nombre":"Cheeseburger","precio":7.5},  
  {"nombre":"Veggie Burger","precio":6.8},  
  {"nombre":"Doble BBQ","precio":8.9}  
]
```
