DESAROLLO

Ejemplo de flujo:

Registrar una huella: Se guarda en la base de datos.

Verificar acceso: Si la cédula es par y está registrada, se concede el acceso y se almacena en la pila.

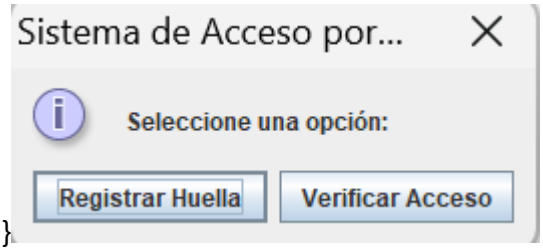
Mostrar ingresos: Lista los nombres de los estudiantes que ingresaron exitosamente.

Este código simula correctamente el comportamiento requerido y almacena los resultados

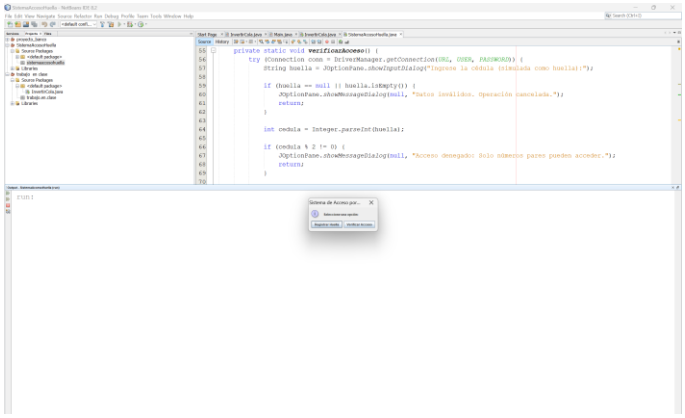
```
import java.sql.*;
import java.util.Scanner;

public class SistemaAccesoHuella {
    private static final String URL = "jdbc:mysql://localhost:3306/universidad";
    private static final String USER = "root";
    private static final String PASSWORD = "password";

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("1. Registrar Huella");
        System.out.println("2. Verificar Acceso");
        System.out.print("Seleccione una opción: ");
        int opcion = scanner.nextInt();

        switch (opcion) {
            case 1:
                registrarHuella();
                break;
            case 2:
```

```
            verificarAcceso();
            break;
        default:
            System.out.println("Opción inválida.");
    }
}

private static void registrarHuella() {
    try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el ID del usuario: ");
        int idUsuario = scanner.nextInt();
        System.out.print("Ingrese la huella (simulada como cadena): ");
        String huella = scanner.next();

        String query = "INSERT INTO huellas (id_usuario, huella) VALUES (?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, idUsuario);
            stmt.setString(2, huella);
            stmt.executeUpdate();
            System.out.println("Huella registrada exitosamente.");
        }
    } catch (SQLException e) {
        System.out.println("Error al registrar la huella: " + e.getMessage());
    }
}

private static void verificarAcceso() {
```

```
    try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese la huella (simulada como cadena): ");
        String huella = scanner.next();

        String query = "SELECT u.nombre, r.rol " +
                       "FROM usuarios u " +
                       "JOIN huellas h ON u.id = h.id_usuario " +
                       "JOIN roles r ON u.id_rol = r.id " +
                       "WHERE h.huella = ?";

        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setString(1, huella);
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                String nombre = rs.getString("nombre");
                String rol = rs.getString("rol");
                System.out.println("Acceso concedido a: " + nombre + " (" + rol + ")");
            } else {
                System.out.println("Huella no reconocida.");
            }
        }
    } catch (SQLException e) {
        System.out.println("Error al verificar el acceso: " + e.getMessage());
    }
}
```



}



En otra fase encontramos más en el desarrollo y encontramos errores y solucionamos haciéndonos a entender este proceso en el cual hay que hacer un registro unitario.

```
import java.sql.*;

import javax.swing.*;

import java.util.Stack;


public class SistemaAccesoHuella {

    private static final String URL = "jdbc:mysql://localhost:3306/universidad";

    private static final String USER = "root";

    private static final String PASSWORD = "password";


    // Pila para almacenar a los estudiantes que ingresaron exitosamente

    private static Stack<String> pilaEstudiantes = new Stack<>();


    public static void main(String[] args) {
```

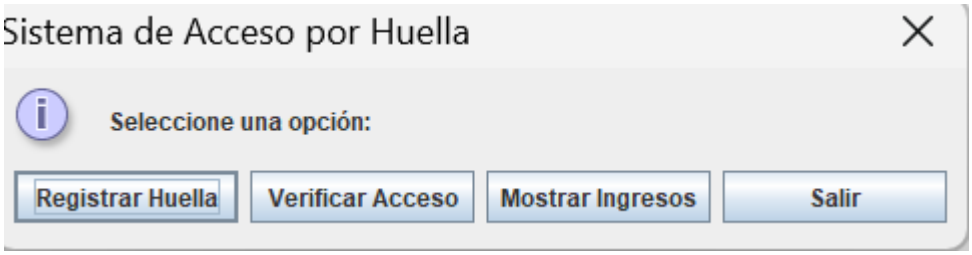
```
        while (true) {
            String[] opciones = {"Registrar Huella", "Verificar Acceso", "Mostrar Ingresos",
"Salir"};
            int opcion = JOptionPane.showOptionDialog(
                    null,
                    "Seleccione una opción:",
                    "Sistema de Acceso por Huella",
                    JOptionPane.DEFAULT_OPTION,
                    JOptionPane.INFORMATION_MESSAGE,
                    null,
                    opciones,
                    opciones[0]
            );

            switch (opcion) {
                case 0:
                    registrarHuella();
                    break;
                case 1:
                    verificarAcceso();
                    break;
                case 2:
                    mostrarIngresos();
                    break;
                case 3:
                    JOptionPane.showMessageDialog(null, "Saliendo del sistema. ¡Hasta luego!");
                    System.exit(0);
                default:
```

```
                JOptionPane.showMessageDialog(null, "Opción no válida.");
            }
        }
    }

    private static void registrarHuella() {
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String idUsuario = JOptionPane.showInputDialog("Ingrese el ID del usuario:");
            String huella = JOptionPane.showInputDialog("Ingrese la cédula (simulada como huella):");

            if (idUsuario == null || huella == null || idUsuario.isEmpty() || huella.isEmpty()) {
                JOptionPane.showMessageDialog(null, "Datos inválidos. Operación cancelada.");
                return;
            }

            String query = "INSERT INTO huellas (id_usuario, huella) VALUES (?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(query)) {
                stmt.setInt(1, Integer.parseInt(idUsuario));
                stmt.setString(2, huella);
                stmt.executeUpdate();
                JOptionPane.showMessageDialog(null, "Huella registrada exitosamente.");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error al registrar la huella: " + e.getMessage());
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Error: ID o huella inválida.");
```

```
    }
  }

  private static void verificarAcceso() {
    try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
      String huella = JOptionPane.showInputDialog("Ingrese la cédula (simulada como huella):");

      if (huella == null || huella.isEmpty()) {
        JOptionPane.showMessageDialog(null, "Datos inválidos. Operación cancelada.");
        return;
      }

      int cedula = Integer.parseInt(huella);

      if (cedula % 2 != 0) {
        JOptionPane.showMessageDialog(null, "Acceso denegado: Solo números pares pueden acceder.");
        return;
      }

      String query = "SELECT u.nombre, r.rol " +
                     "FROM usuarios u " +
                     "JOIN huellas h ON u.id = h.id_usuario " +
                     "JOIN roles r ON u.id_rol = r.id " +
                     "WHERE h.huella = ?";

      try (PreparedStatement stmt = conn.prepareStatement(query)) {
```

```
            stmt.setString(1, huella);
            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {
                String nombre = rs.getString("nombre");
                String rol = rs.getString("rol");
                JOptionPane.showMessageDialog(null, "Acceso concedido a: " + nombre + " (" +
rol + ")");
                // Agregar a la pila de ingresos exitosos
                pilaEstudiantes.push(nombre + " (" + rol + ")");
            } else {
                JOptionPane.showMessageDialog(null, "Huella no reconocida.");
            }
          }
      } catch (SQLException e) {
          JOptionPane.showMessageDialog(null, "Error al verificar el acceso: " +
e.getMessage());
      } catch (NumberFormatException e) {
          JOptionPane.showMessageDialog(null, "Error: Cédula inválida.");
      }
  }

  private static void mostrarIngresos() {
      if (pilaEstudiantes.isEmpty()) {
          JOptionPane.showMessageDialog(null, "No hay ingresos registrados.");
      } else {
          StringBuilder listaIngresos = new StringBuilder("Estudiantes que ingresaron
exitosamente:\n");
          for (String estudiante : pilaEstudiantes) {
```

```
            listaIngresos.append("- ").append(estudiante).append("\n");
        }
        JOptionPane.showMessageDialog(null, listaIngresos.toString());
    }
  }
}
```



Con respecto ala base de datos

```
-- Crear la base de datos
CREATE DATABASE universidad;

-- Usar la base de datos
USE universidad;

-- Crear la tabla de roles
CREATE TABLE roles (
    id INT AUTO_INCREMENT PRIMARY KEY,
    rol VARCHAR(50) NOT NULL
);

-- Crear la tabla de usuarios
CREATE TABLE usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
```

```
    id_rol INT,
    FOREIGN KEY (id_rol) REFERENCES roles(id)
);

-- Crear la tabla de huellas
CREATE TABLE huellas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    id_usuario INT,
    huella VARCHAR(255) UNIQUE NOT NULL,
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id)
);
```

Con respecto a datos provenientes a las tablas que se anexaron

```
-- Insertar roles
INSERT INTO roles (rol) VALUES
('Estudiante'),
('Docente'),
('Administrativo');

-- Insertar usuarios
INSERT INTO usuarios (nombre, id_rol) VALUES
('Juan Pérez', 1),
('María Gómez', 2),
('Carlos Sánchez', 1),
('Luisa Torres', 3);

-- Insertar huellas (simuladas como números de cédulas)
INSERT INTO huellas (id_usuario, huella) VALUES
```

(1, '10203040'),

(2, '11223344'),

(3, '12345678'),

(4, '87654321');