



Instituto Politécnico Nacional
Escuela Superior de Cómputo



Programación Orientada a Objetos

Reporte de Práctica #7 JDBC

Profesor: Roberto Tecla Parra

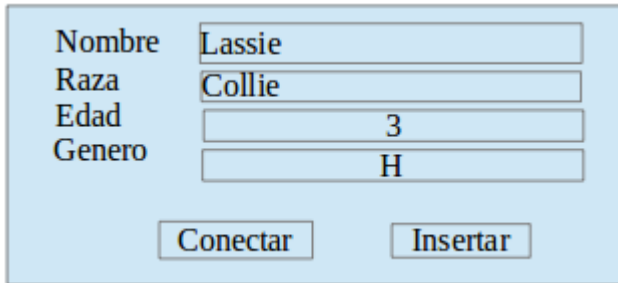
Alumno: Calva Hernández José Manuel

Grupo: 2CM3

Insertar Perros en una base de datos

Hacer un programa que utilice la API JDBC y que tenga una GUI que permita insertar "perros" en una base de datos que tiene una tabla Perro con el siguiente esquema Perro(nombre, raza, edad, genero). La GUI consistirá en 4 etiquetas, 4 campos de entrada y 2 botones.

Un botón para conectar a la base de datos y otro para insertar los datos de un perro.



Objetivos

- Entender la funcionalidad de JavaSQL
- Entender cada uno de los pasos del protocolo para comunicación entre la base de datos y el código en Java
- Entender el manejo de credenciales de acceso a la base de datos
- Hacer uso del manejo de errores mediante el catch y ejecutar todo lo que se requiera hacer en el try, funcionando como código "en un mundo ideal".

Desarrollo

Se importan las librerías necesarias para el desarrollo de la práctica, entre ellas la referente a sql.

```
import java.sql.*;
import java.util.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.applet.*;
```

Se crea una clase llamada ResultApp que nos servirá como Interfaz gráfica para el usuario, y que éste pueda interactuar con la base de datos.

```
class ResultApp extends JPanel implements ActionListener{

    static Connection conex;
    JButton bConnect, bInsert, bShow, bClose;
    JLabel lName, lRace, lAge, lGender;
    JTextField tName, tRace, tAge, tGender;
    JPanel p1,p2;
    boolean stateConnection = false;
}
```

Se crea a continuación el constructor para nuestra clase ResultApp que se encargará de modelar la interfaz gráfica, además de añadir la función de escucha a los botones para su posterior manejo.

```
public ResultApp(){
    bConnect = new JButton("Connect");
    bInsert = new JButton("Insert");
    bShow = new JButton("Show");
    bClose = new JButton("Close");
    lName = new JLabel("Name: ");
    tName = new JTextField(20);
    lRace = new JLabel("Race: ");
    tRace = new JTextField(20);
    lAge = new JLabel("Age: ");
    tAge = new JTextField(2);
    lGender = new JLabel("Gender: ");
    tGender = new JTextField(1);
    p1 = new JPanel();
    p2 = new JPanel();
    p1.setLayout(new GridLayout(3,3));
    p2.setLayout(new BorderLayout());
    add(lName);
    add(lRace);
    add(lAge);
    add(lGender);
    add(tName);
    add(tRace);
    add(tAge);
    add(tGender);
    p2.add(bConnect, BorderLayout.NORTH);
    p2.add(bInsert, BorderLayout.WEST);
    p2.add(bShow, BorderLayout.EAST);
    p2.add(bClose, BorderLayout.SOUTH);
    add(p2);
}
```

Se inicializa el main mandando llamar a la propia clase con parámetros de construcción a modo de JPanel.

```
public static void main(String[] args) {
    JFrame f = new JFrame("Dog Registration");
    f.add("Center", new ResultApp());
    f.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    f.setSize(250, 250); f.setVisible(true);
}
```

Implementamos el método actionPerformed que nos permitirá escuchar a los botones que sean seleccionados por el usuario, así, nuestros botones tendrán diferentes funciones.

El primero de ellos, bConnect, se encargará de establecer la conexión con la base de datos por medio de una clase encontrada en la librería de sql, a la cual le vamos a dar la url de nuestra base de datos, así como el nombre, password y

contraseña. Ésta intentará conectarse por medio de un bloque try, y en su defecto nos arrojará un error de conexión con la misma.

El segundo nos permitirá insertar los valores que haya colocado el usuario en los campos de texto, nuevamente se realiza en un bloque try porque puede generar excepciones.

El tercer botón nos mostrará los datos existentes en el momento en nuestra base de datos. Y para terminar el último botón cerrará la conexión con la base y nos cerrará el programa.

```
public void actionPerformed(ActionEvent e){
    JButton selected = (JButton)e.getSource();
    if (selected == bConnect && !stateConnection) {
        String url = "jdbc:mysql://localhost:3306/", dbName = "p7";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conex=DriverManager.getConnection(url+dbName, "root", " ");
            System.out.println("Connection established.");
            stateConnection = true;
        }
        catch (Exception ex){ System.out.println(ex); System.exit(0); }
    }
    else if (selected != bConnect && !stateConnection) {
        System.out.println("Connection not established yet.");
    }
    else if (selected == bInsert) {
        String sName = tName.getText().toUpperCase();
        String sRace = tRace.getText().toUpperCase();
        int nAge = Integer.parseInt(tAge.getText());
        char cGender = tGender.getText().toUpperCase().charAt(0);
        String sqlInsert="INSERT INTO Dog VALUES('"+sName+"','"+sRace+"','"+nAge+"','"+cGender+"')";
        try{
            int i;
            Statement statement=conex.createStatement();
            i=statement.executeUpdate(sqlInsert);
            System.out.println("QUERY OK, "+i+" row affected.");
        } catch (Exception ex){ System.out.println(ex); System.exit(0); }
    }
    else if(selected == bShow){
        String sqlConsult="SELECT * FROM Dog";
        try{
            Statement statement=conex.createStatement();
            ResultSet result=statement.executeQuery(sqlConsult);
            displayResults(result);
        } catch (Exception ex){ System.out.println(ex); System.exit(0); }
    }
    else if (selected == bClose) {
        try{
            conex.close();
            System.out.println("Connection closed.");
            System.exit(0);
        } catch (Exception ex){ System.out.println(ex); System.exit(0); }
    }
}
```

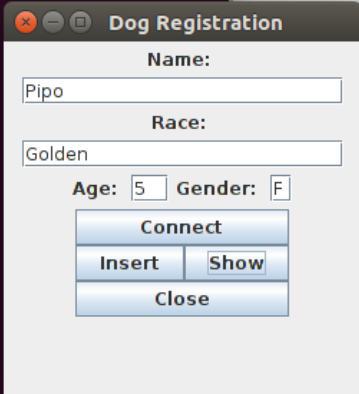
Para ejecutar el código seguiremos el siguiente comando:

```
akotadi@Laptop: ~/Documents/POO/Práctica 7
akotadi@Laptop:~/Documents/POO/Práctica 7$ javac ResultAppv2.java
akotadi@Laptop:~/Documents/POO/Práctica 7$ java -cp ./mysqlcon.jar ResultApp
Connection established.
```

NAME	RACE	AGE	GENDER
BLACK	LABRADOR	6	F
COFFEE	MEXA	7	M
DOGGY	PITBULL	10	M
HUGGY	HUSKY	20	M
JUSTIN	PODDLE	13	M
TOMMY	CHIHUAHUA	9	F

QUERY OK, 1 row affected.

NAME	RACE	AGE	GENDER
BLACK	LABRADOR	6	F
COFFEE	MEXA	7	M
DOGGY	PITBULL	10	M
HUGGY	HUSKY	20	M
JUSTIN	PODDLE	13	M
PIPO	GOLDEN	5	F
TOMMY	CHIHUAHUA	9	F



The image shows a 'Dog Registration' window with the following fields and buttons:

- Name:
- Race:
- Age: Gender:
- Buttons: Connect, Insert, Show, Close

Conclusiones

Las Bases de Datos nos permiten guardar información de forma que no se vea afectada por el programa principal una vez se inserten en la tabla (a menos que incluyamos en el mismo una opción para modificarlos o eliminarlos).

Como se sabe, aprender a usar esos API nos puede brindar muchas opciones para futuros proyectos que nos dejen a lo largo de nuestra vida académica u profesional por lo que explorar este tema a mayor profundidad será la prioridad.