PECL3

Grupo de laboratorio: Martes 12:00-14:00

Profesor: José A. Gutiérrez de Mesa

Alumnos: Javier Martín Gómez 47231977M

Christian Ayala Urbanos 51196082Z

**ANÁLISIS DE LAS FORMAS NORMALES**

**RELACIONES Y DEPENDENCIAS FUNCIONALES**

**MUSICO** (DNI (PK), CP, TLF, NOMBRE, APELLIDOS, CIUDAD, PROVINCIA, INSTRUMENTO,

COD\_GRUPO, CALLE, N.º PISO)

**DNI** 🡪Nombre, Apellidos, Ciudad, Provincia, TLF, Instrumento, CP, Código Grupo, Calle, N.º, Piso

**GRUPO** (GENERO, COD\_GRUPO (PK))

**Código Grupo** 🡪 Género

**DISCO** (TITULO, GENERO, N.º REFERENCIA (PK), FECHA, FORMATO, TAMAÑO, CODIGO\_GRUPO, TIPO)

**Número de referencia** 🡪 Titulo, Género, Fecha, Formato, Tamaño, Tipo, Código Grupo

**CANCION** (NOMBRE (PK), DURACION, FECHA GRABACION (PK), DNI\_MUSICO, NUMERO\_DISCO)

**Nombre, Fecha Grabación** 🡪 Nombre, Duración, DNI\_Musico, Numero\_Disco

**USUARIO** (NOMBRE, APELLIDOS, DNI (PK), EMAIL, PUNTOS)

**DNI** 🡪Nombre, Apellidos, e-mail, Puntos

**ENTRADA** (PRECIO, CODIGO (PK), CODIGO\_CONCIERTO, DNI\_USUARIO)

**Código** 🡪Precio, Código\_Concierto, DNI\_Usuario

**CONCIERTO** (FECHA, PAIS, RECINTO, CIUDAD, CODIGO (PK))

**Código** 🡪Recinto, Ciudad, País, Fecha

**OPINION** (OPINION\_CONCIERTO, OPINION\_DISCO, N.º OPINION (PK), NUMERO\_DISCO

CODIGO\_CONCIERTO, DNI\_USUARIO)

**N.º Opinión** 🡪 Opinion\_Concierto, Opinión\_Disco, Número\_Disco, DNI\_Usuario, Código\_Concierto

**DIRECCION** (CALLE (PK), NUMERO (PK), PISO (PK))

**MUCHOS\_CONCIERTO\_TIENE\_MUCHOS\_GRUPOS** (CODIGO\_CONCIERTO (PK), CODIGO\_GRUPO(PK))

**MUCHOS\_USUARIOS\_TIENE\_MUCHOS\_DISCO** (DNI\_USUARIO (PK), NUMERO\_DISCO (PK))

**DISPARADORES**

**Trigger DNI**

Este trigger consiste en si el DNI introducido es válido, es decir, si la letra es la adecuada con los 8 números.

CREATE OR REPLACE FUNCTION funcion\_valida\_dni()

RETURNS "trigger" AS

$validardni$

DECLARE

letrasValidas CHAR(23) := 'TRWAGMYFPDXBNJZSQVHLCKE';

letraCorrecta CHAR;

letraLeida CHAR := substring(NEW."DNI",9);

numero INTEGER :=substring(NEW."DNI" from 1 for 8)::integer;

resto integer;

BEGIN

resto = numero % 23;

letraCorrecta = substring(letrasValidas, resto+1, 1);

IF (letraCorrecta = letraLeida) THEN

NEW."DNI":= (numero::text||letraLeida);

ELSE

RAISE EXCEPTION 'El DNI introducido es incorrecto';

END IF;

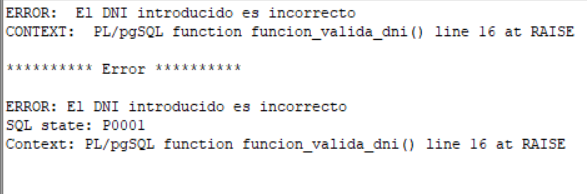
RETURN NEW;

END;

$validardni$

LANGUAGE 'plpgsql' VOLATILE;

ALTER FUNCTION funcion\_valida\_dni() OWNER TO postgres;



**ESTA ES UNA POSIBLE SALIDA SI INTRODUCIMOS UN DNI INCORRECTO**

**Trigger puntos**

La función de este Trigger es sumar 10 puntos a los Usuarios que han introducido una opinión.

create or replace function insertar\_puntos() returns trigger as $actualizar$

declare begin

update "Usuario" set "Puntos"="Puntos"+10 where "DNI"=NEW."DNI\_Usuario";

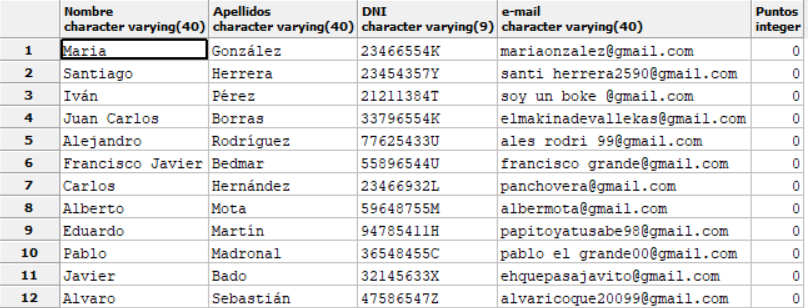
return null;

end;

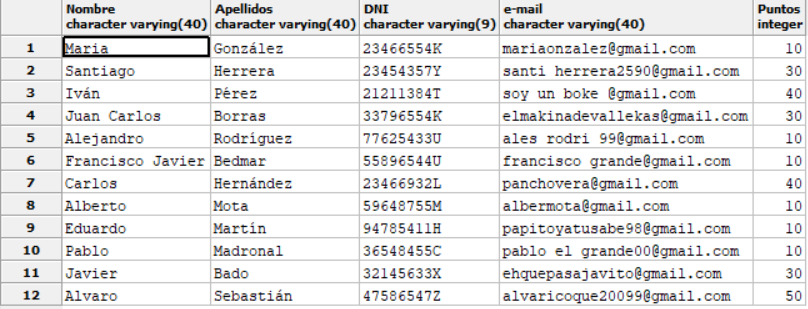
$actualizar$ language plpgsql;

create trigger inserta\_punto after insert on "Opinion"

for each row execute procedure insertar\_puntos();



**ESTOS SON LOS USUARIOS ANTES DE REALIZAR LAS OPINIONES**

****

**ESTOS SON LOS USUARIOS DESPUÉS DE REALIZAR LAS OPINIONES (SE LE SUMAN 10 PUNTOS POR CADA OPINIÓN)**

**Trigger precios correctos**

Este Trigger indica una excepción si los precios son negativos

CREATE OR REPLACE FUNCTION precio\_incorrecto()

RETURNS "trigger" AS

$precio\_incorrecto$

begin

IF (NEW."Precio"<cast(0 as money)) THEN

RAISE EXCEPTION 'El precio no puede ser negativo';

END IF;

RETURN NEW;

END;

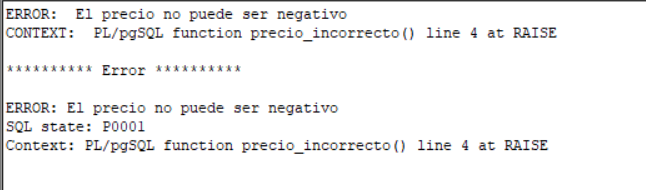
$precio\_incorrecto$

LANGUAGE 'plpgsql' VOLATILE;

ALTER FUNCTION precio\_incorrecto() OWNER TO postgres;

create trigger precio\_incorrecto after insert on "Entrada"

for each row execute procedure precio\_incorrecto()



**ESTE ES EL RESULTADO AL INTRODUCIR UN PRECIO NEGATIVO**

**CONEXIÓN A BASE DE DATOS (JAVA)**

**CLASE CONEXIÓN**

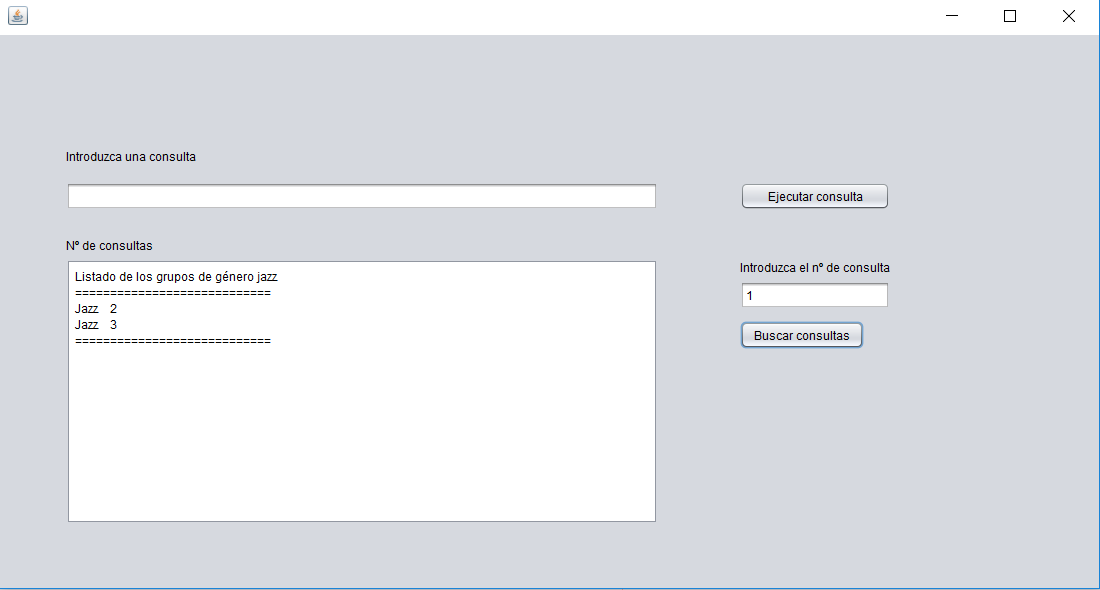
Para conectar la base de datos desde Java, lo primero que tenemos que hacer es instalarnos el JDBC Driver. Después, tras crear el proyecto en NetBeans, introducimos en el apartado bibliotecas, el archivo JDBC.

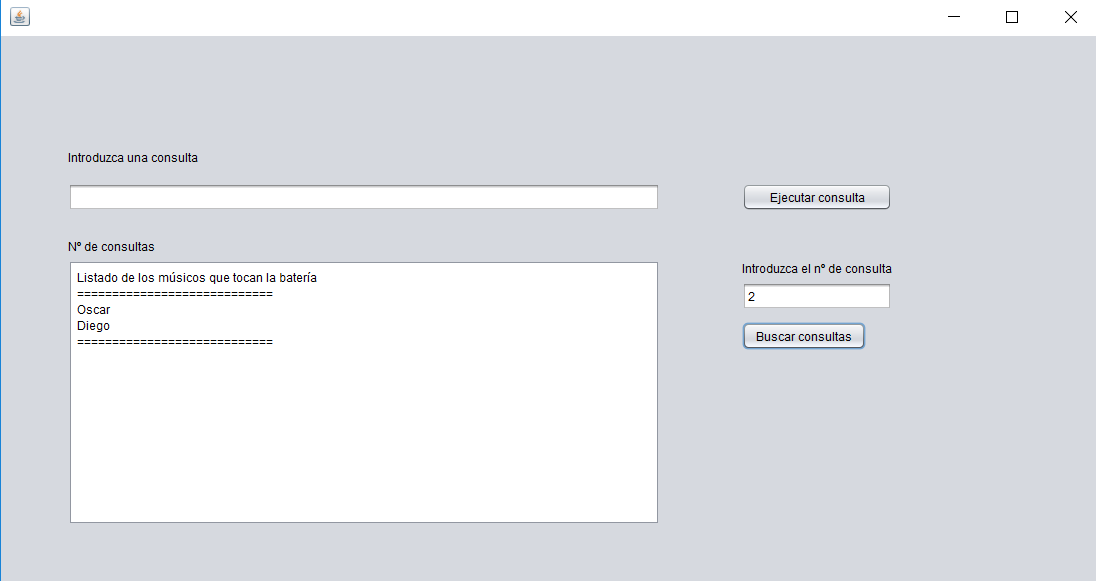
Tras realizar los imports necesarios, creamos unas variables para almacenar la URL del servidor, el usuario y la contraseña.

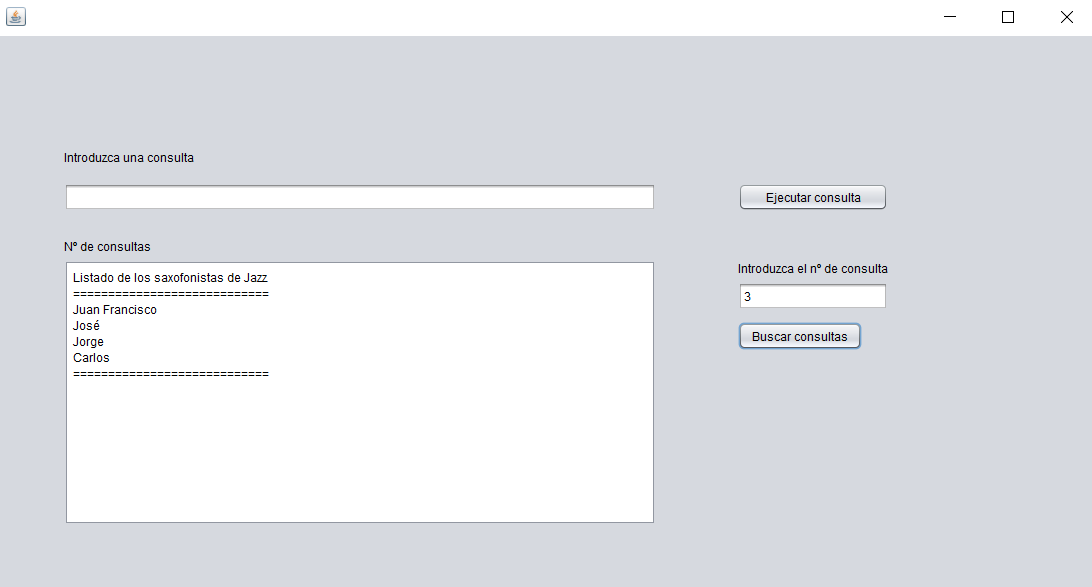
Posteriormente, creamos un método connect() para abrir la conexión y otro método disconect() para cerrarla.

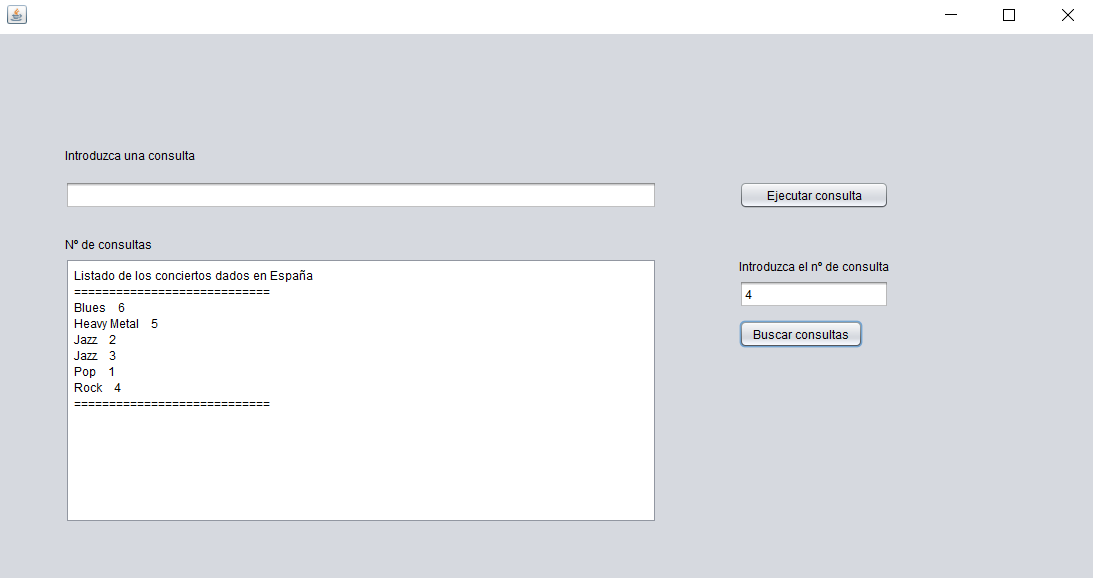
Con estos métodos, nos conectaremos a la base de datos, crearemos nuestros métodos para ejecutar nuestras consultas de la PECL2 y las ejecutaremos desde nuestro proyecto NetBeans.

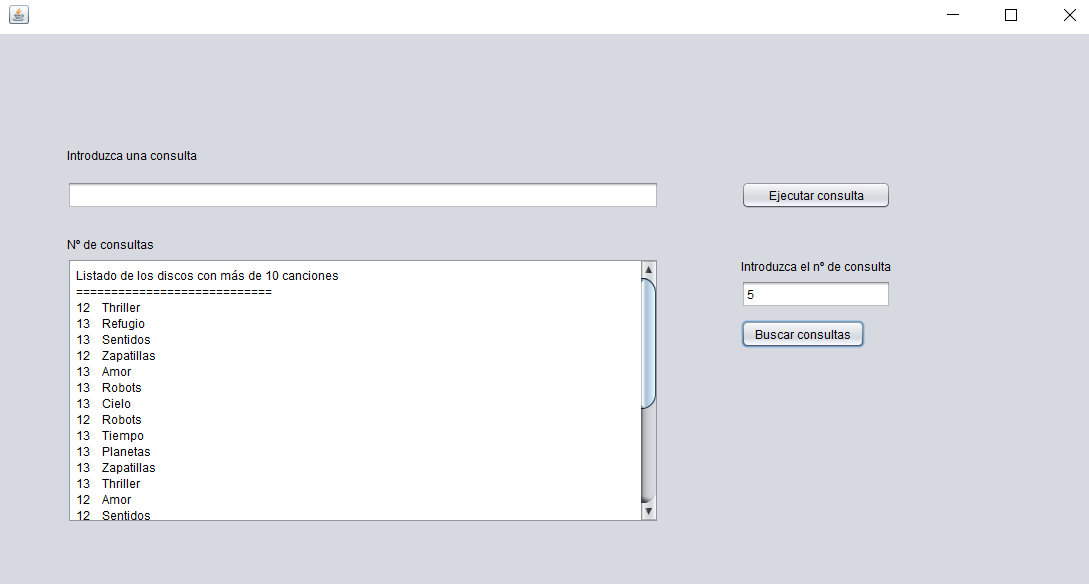
RESULTADO CONSULTAS

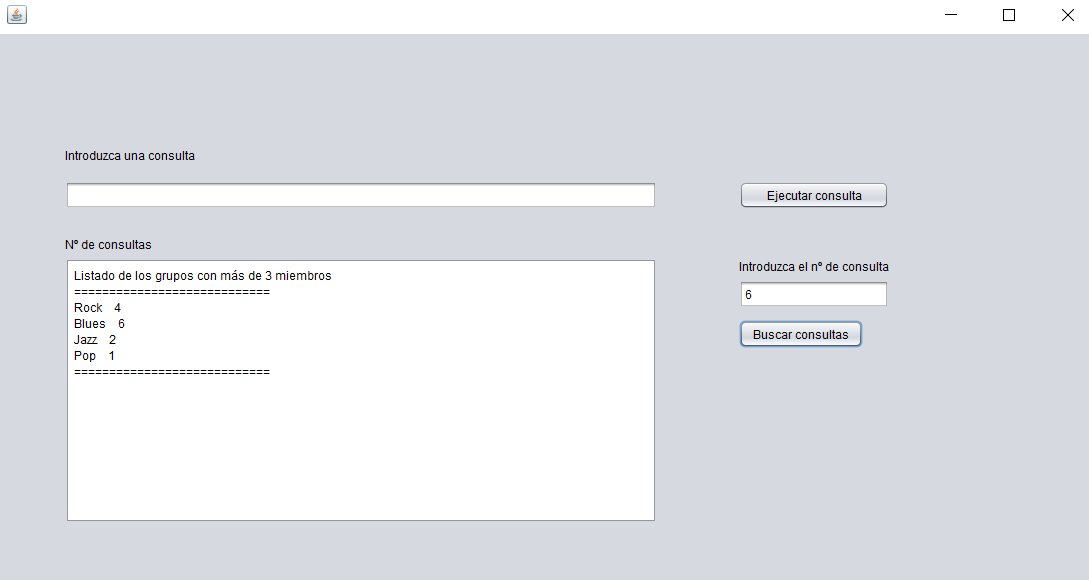




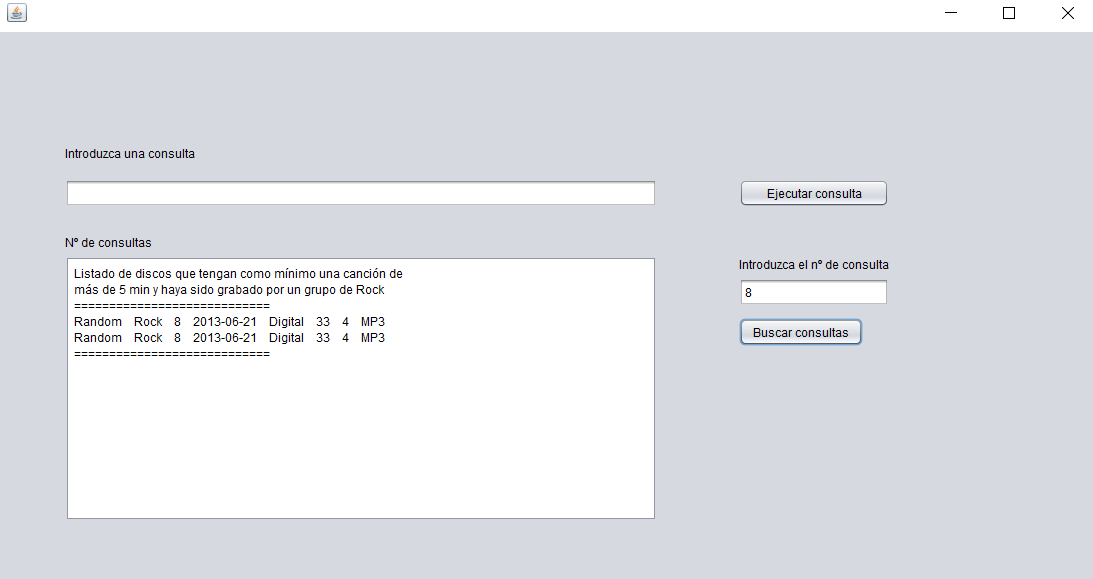


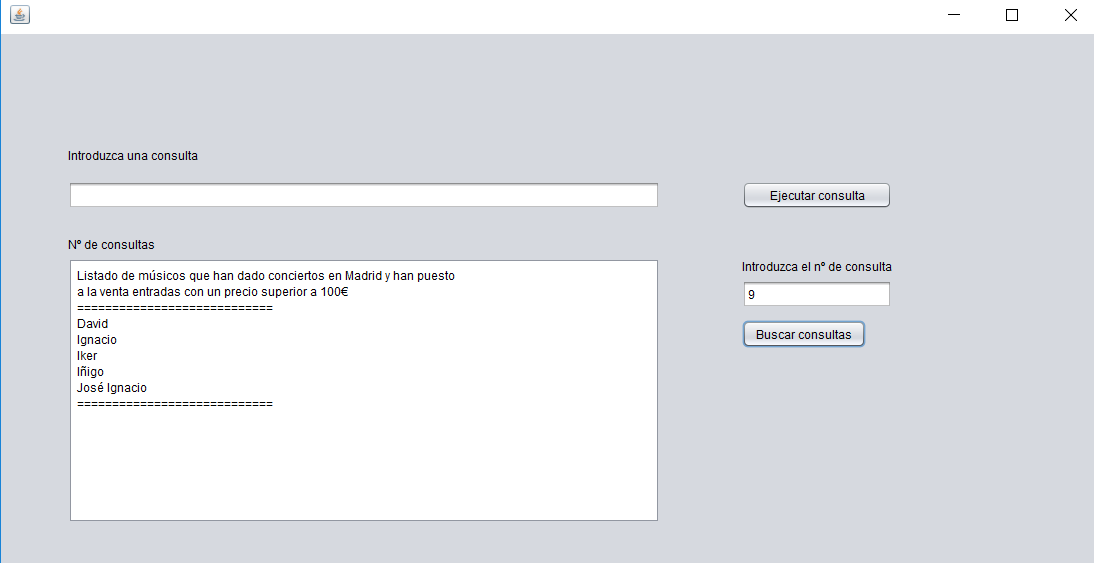


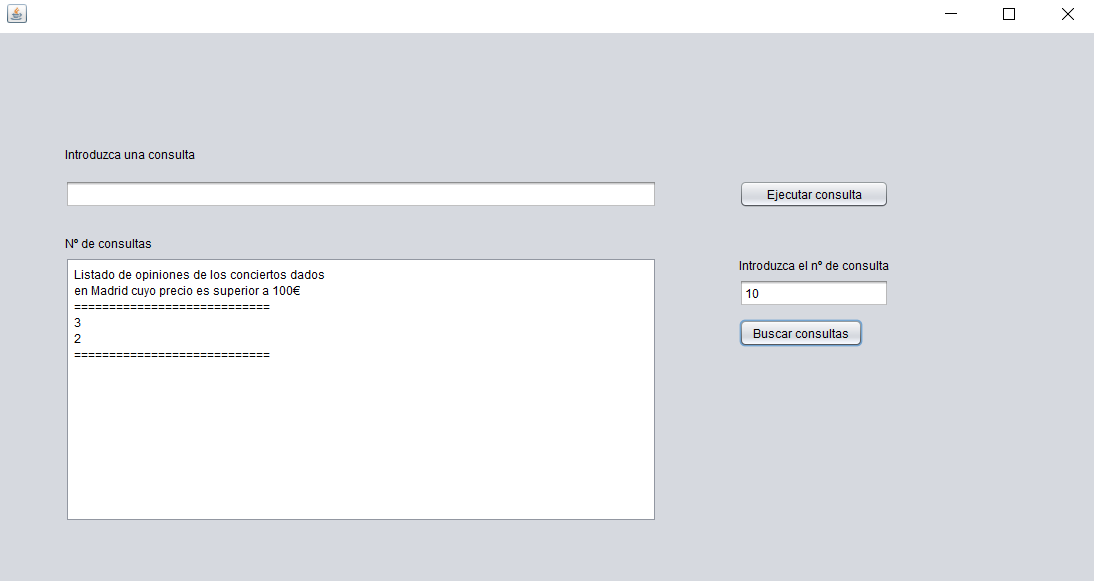


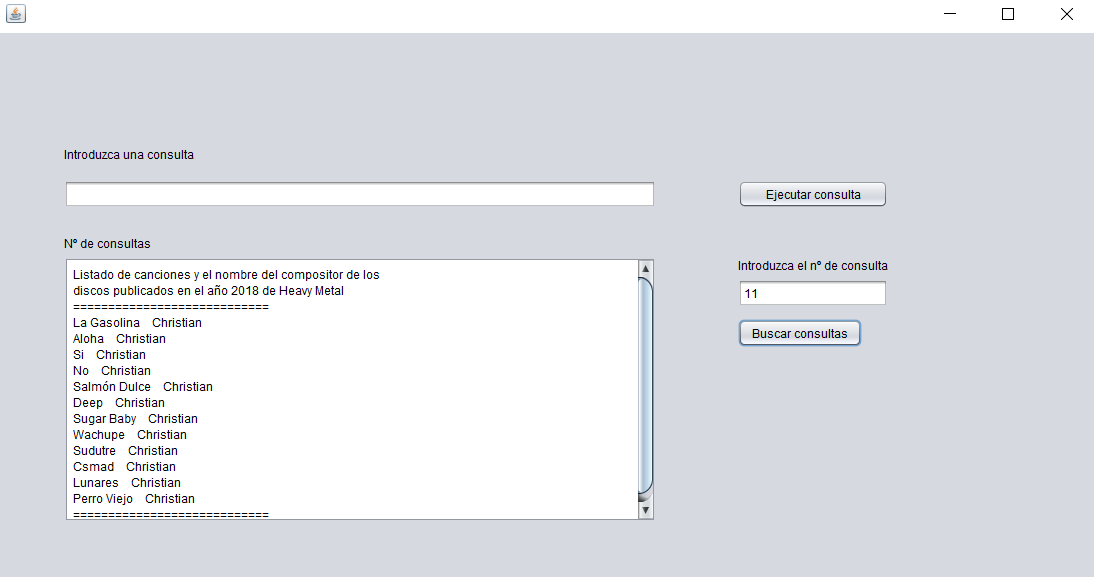


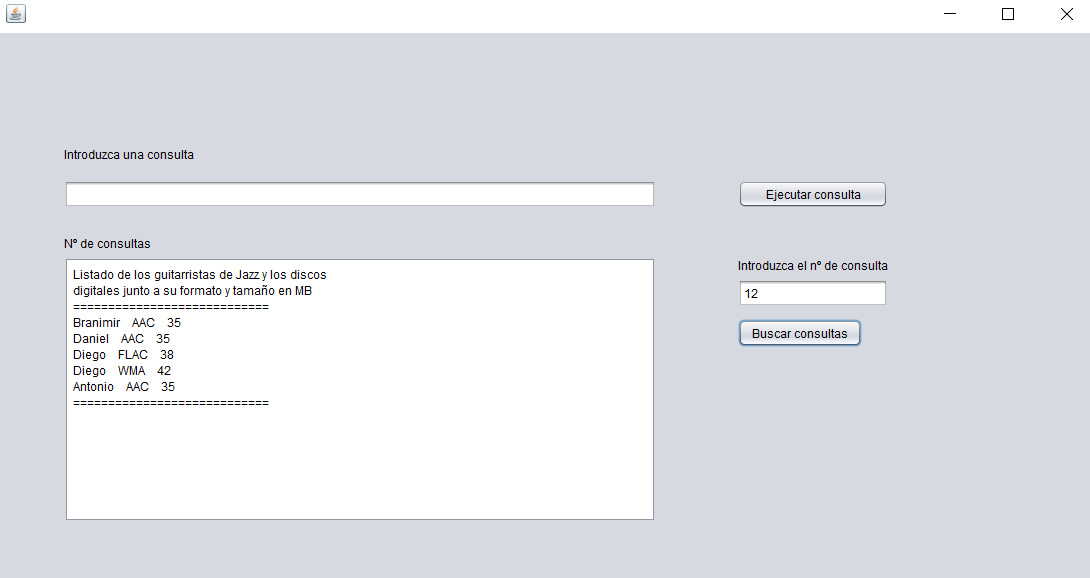


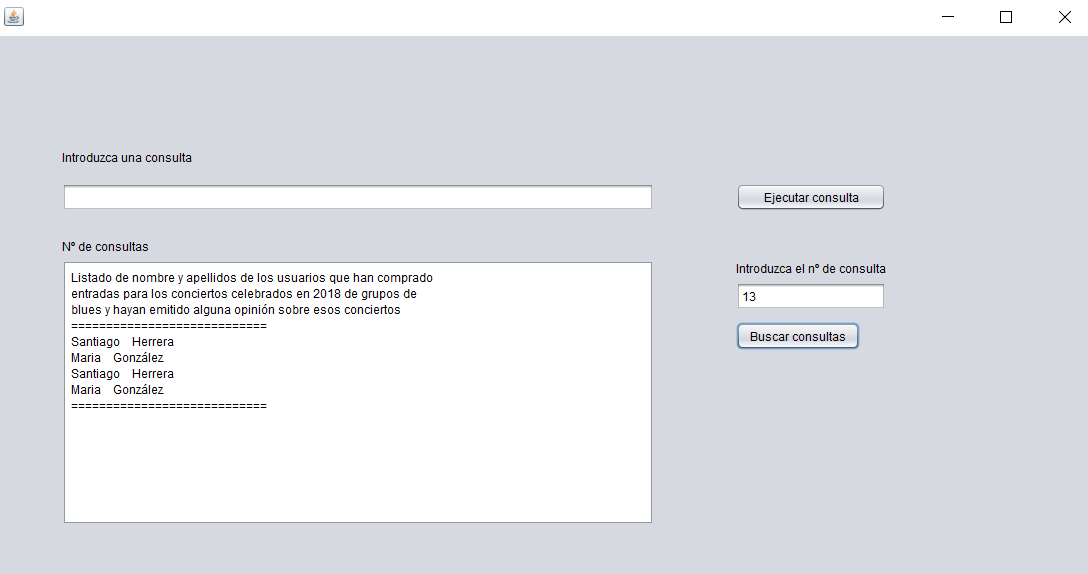


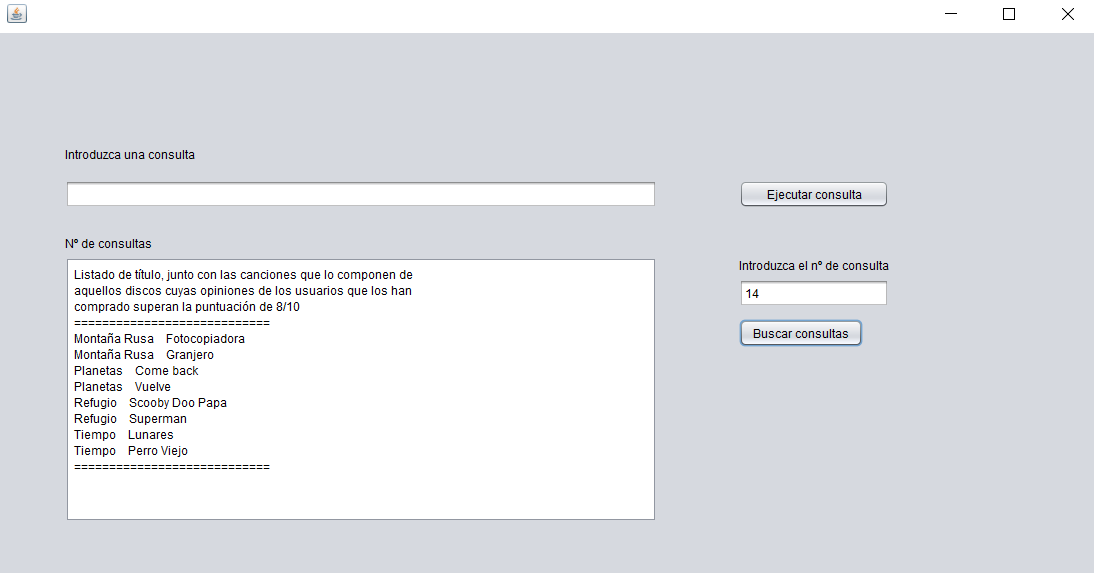


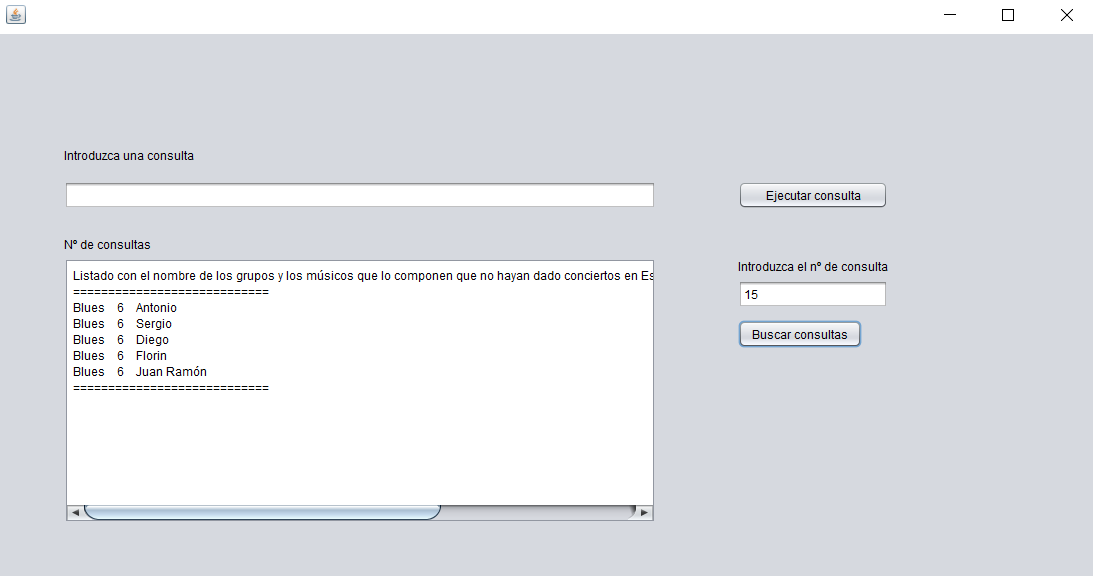














**CÓDIGO FUENTE CLASE CONEXIÓN**

package conexion;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.ResultSetMetaData;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.JTextArea;

public class Conexion {

private final String url = "jdbc:postgresql://127.0.0.1:5432/pl2";

private final String user;

private final String password;

private final Logger logger = Logger.getLogger(Conexion.class.getName());

private Connection conn = null;

private boolean b = false;

public Connection connect() {

try {

Class.forName("org.postgresql.Driver").newInstance();

conn = DriverManager.getConnection(url, user, password);

System.out.println("Conectado!!");

this.b = true;

} catch (SQLException e) {

System.out.println(e.getMessage());

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException ex) {

System.out.println("Driver no detectado");

}

return conn;

}

public Conexion(String user, String password) {

this.user = user;

this.password = password;

}

public boolean getB() {

return b;

}

public void disconnect() {

try {

conn.close();

System.out.println("Desconectado!!");

} catch (SQLException ex) {

logger.log(Level.SEVERE, "No se ha podido cerrar la conexión", ex);

} catch (Exception ex) {

logger.log(Level.WARNING, "Excepción capturada", ex);

}

}

public void getConsulta1(JTextArea panel) {

try {

String a = "";

String query = "select \* from \"Grupo\" where \"Género\"='Jazz';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los grupos de género jazz\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() +" "+ rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta2(JTextArea panel) {

try {

String a = "";

String query = "select \"Nombre\" from \"Músico\" where \"Instrumento\"='Batería'";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los músicos que tocan la batería\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta3(JTextArea panel) {

try {

String a = "";

String query = "select \"Nombre\" from \"Músico\" inner join \"Grupo\" on \n"

+ "\"Grupo\".\"Código Grupo\"=\"Músico\".\"Código Grupo\_Grupo\" where \"Grupo\".\"Género\"='Jazz' and \"Músico\".\"Instrumento\"='Saxofón';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los saxofonistas de Jazz\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta4(JTextArea panel) {

try {

String a = "";

String query = "select distinct \"Grupo\".\* from \"Grupo\" inner join \"muchos\_Concierto\_tiene\_muchos\_Grupo\" on \"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\"\n"

+ "inner join \"Concierto\" on \"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código\_Concierto\"=\"Concierto\".\"Código\" where \"Concierto\".\"País\"='España';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los conciertos dados en España\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + (rs.getString(1).trim() +" "+ rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta5(JTextArea panel) {

try {

String a = "";

String query = "select distinct count(\"Número de referencia\_Disco\"),\"Disco\".\"Título\" from \"Disco\", \"Canción\" group by \"Número de referencia\_Disco\",\"Disco\".\"Título\" having count(\"Número de referencia\_Disco\")>10 ;";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los discos con más de 10 canciones\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+ rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta6(JTextArea panel) {

try {

String a = "";

String query = "select \"Grupo\".\* from \"Grupo\" inner join \"Músico\" on \"Músico\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\" group by \"Código Grupo\" having count(\"Código Grupo\")>3;";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los grupos con más de 3 miembros\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+ rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta7(JTextArea panel) {

try {

String a = "";

String query = "select \"Opinion Disco\" from \"Opinion\" where \"DNI\_Usuario\" in"

+ "(select \"DNI\_Usuario\" from \"Opinion\" group by \"DNI\_Usuario\" having count(\"DNI\_Usuario\")>3);";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de todas las opiniones de los discos emitidas por usuarios que hayan comprado al menos tres discos \n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta8(JTextArea panel) {

try {

String a = "";

String query = "select \"Disco\".\* from \"Disco\" inner join \"Canción\" on \"Canción\".\"Número de referencia\_Disco\"=\"Disco\".\"Número de referencia\" inner join \"Grupo\" on \"Grupo\".\"Código Grupo\"=\"Disco\".\"Código Grupo\_Grupo\" \n"

+ "where \"Duración\">='05:00:00' and \"Grupo\".\"Género\"='Rock';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de discos que tengan como mínimo una canción de \nmás de 5 min y haya sido grabado por un grupo de Rock\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+rs.getString(2).trim() +" "+ rs.getString(3).trim() +" "+ rs.getString(4).trim() +" "+ rs.getString(5).trim() +

" "+rs.getString(6).trim() +" "+ rs.getString(7).trim() + " "+rs.getString(8).trim() +" "+ " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta9(JTextArea panel) {

try {

String a = "";

String query = "select distinct \"Nombre\" from \"Músico\" inner join \"Grupo\" on \"Grupo\".\"Código Grupo\"=\"Músico\".\"Código Grupo\_Grupo\" \n"

+ "inner join \"muchos\_Concierto\_tiene\_muchos\_Grupo\" on \"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\"\n"

+ "inner join \"Concierto\" on \"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código\_Concierto\"=\"Concierto\".\"Código\" \n"

+ "inner join \"Entrada\" on \"Entrada\".\"Código\_Concierto\"=\"Concierto\".\"Código\" \n"

+ "where \"Concierto\".\"Ciudad\"='Madrid' and \"Entrada\".\"Precio\">'100';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de músicos que han dado conciertos en Madrid y han puesto \na la venta entradas con un precio superior a 100€\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta10(JTextArea panel) {

try {

String a = "";

String query = "select \"Opinion concierto\" from \"Opinion\" inner join \"Concierto\" on \"Opinion\".\"Código\_Concierto\"=\"Concierto\".\"Código\" \n"

+ "inner join \"Entrada\" on \"Entrada\".\"Código\_Concierto\"=\"Concierto\".\"Código\" \n"

+ "where \"Concierto\".\"Ciudad\"='Madrid' and \"Entrada\".\"Precio\">'100';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de opiniones de los conciertos dados \nen Madrid cuyo precio es superior a 100€\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta11(JTextArea panel) {///////////////////

try {

String a = "";

String query = "select \"Canción\".\"Nombre\",\"Músico\".\"Nombre\" from \"Canción\" inner join \"Músico\" on \"Músico\".\"DNI\"=\"Canción\".\"DNI\_Músico\" \n"

+ "inner join \"Disco\" on \"Disco\".\"Número de referencia\"=\"Canción\".\"Número de referencia\_Disco\" where \"Disco\".\"Género\"='Heavy Metal' and (\"Disco\".\"Fecha\"<'31/12/2018 23:59:59' \n"

+ "or \"Disco\".\"Fecha\">'1/1/2018 00:00:00');";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de canciones y el nombre del compositor de los \ndiscos publicados en el año 2018 de Heavy Metal \n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta12(JTextArea panel) {/////////////

try {

String a = "";

String query = "select \"Nombre\",\"Disco\".\"Tipo\",\"Disco\".\"Tamaño (MB)\" from \"Músico\" inner join \"Grupo\" on \"Grupo\".\"Código Grupo\"=\"Músico\".\"Código Grupo\_Grupo\" inner join \n"

+ "\"Disco\" on \"Disco\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\"\n"

+ " where \"Músico\".\"Instrumento\"='Guitarra' and \"Grupo\".\"Género\"='Jazz' and \"Disco\".\"Formato\"='Digital';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de los guitarristas de Jazz y los discos \ndigitales junto a su formato y tamaño en MB\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+rs.getString(2).trim() + " "+rs.getString(3).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta13(JTextArea panel) {

try {

String a = "";

String query = "select \"Nombre\", \"Apellidos\" from \"Usuario\" inner join \"Entrada\" on \"Usuario\".\"DNI\"=\"Entrada\".\"DNI\_Usuario\" \n"

+ "inner join \"Concierto\" on \"Concierto\".\"Código\"=\"Entrada\".\"Código\_Concierto\" inner join \"muchos\_Concierto\_tiene\_muchos\_Grupo\" on \n"

+ "\"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código Grupo\_Grupo\"=\"Concierto\".\"Código\"\n"

+ "inner join \"Grupo\" on \"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código\_Concierto\"=\"Concierto\".\"Código\" inner join \"Opinion\" on\n"

+ "\"Opinion\".\"Código\_Concierto\"=\"Concierto\".\"Código\" where \"Grupo\".\"Género\"='Blues' and\n"

+ "\"Concierto\".\"Fecha\"<'31/12/2018 23:59:59' and \"Concierto\".\"Fecha\">'1/1/2018 00:00:00';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de nombre y apellidos de los usuarios que han comprado \nentradas para los conciertos celebrados en 2018 de grupos de \nblues y hayan emitido alguna opinión sobre esos conciertos\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() +" "+ rs.getString(2).trim()+ " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta14(JTextArea panel) {

try {

String a = "";

String query = "select \"Disco\".\"Título\", \"Canción\".\"Nombre\" as Disco from \"Disco\" inner join \"Canción\" on \"Disco\".\"Número de referencia\"=\"Canción\".\"Número de referencia\_Disco\"\n"

+ "where \"Disco\".\"Título\" in (select \"Título\" from \"Disco\" inner join \"Opinion\" on \"Opinion\".\"Número de referencia\_Disco\"=\"Disco\".\"Número de referencia\" where \"Opinion Disco\">8);";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado de título, junto con las canciones que lo componen de \naquellos discos cuyas opiniones de los usuarios que los han \ncomprado superan la puntuación de 8/10\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() + " "+rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta15(JTextArea panel) {

try {

String a = "";

String query = "select \"Grupo\".\*,\"Músico\".\"Nombre\" from \"Músico\" inner join \"Grupo\" on \"Grupo\".\"Código Grupo\"=\"Músico\".\"Código Grupo\_Grupo\"\n"

+ "inner join \"Disco\" on \"Grupo\".\"Código Grupo\"=\"Disco\".\"Código Grupo\_Grupo\" inner join \"muchos\_Concierto\_tiene\_muchos\_Grupo\" on \n"

+ "\"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\" inner join \"Concierto\" on \n"

+ "\"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código\_Concierto\"=\"Concierto\".\"Código\" where not(\"País\"='España')\n"

+ "and \"Disco\".\"Número de referencia\" in (select \"Número de referencia\_Disco\" from \"Canción\" group by \"Número de referencia\_Disco\" \n"

+ "having count(\"Número de referencia\_Disco\")>10);";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado con el nombre de los grupos y los músicos que lo componen que no hayan dado conciertos en España y hayan grabado discos con más de 10 canciones \n");

a = a + ("============================\n");

while (rs.next()) {

a = a + ( rs.getString(1).trim() +" "+ rs.getString(2).trim() + " "+rs.getString(3).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void getConsulta16(JTextArea panel) {///////////////

try {

String a = "";

String query = "select distinct \"Canción\".\"Nombre\",\"Músico\".\"Nombre\" from \"Canción\" inner join \"Músico\" on \"Músico\".\"DNI\"=\"Canción\".\"DNI\_Músico\" \n"

+ "inner join \"Grupo\" on \"Grupo\".\"Código Grupo\"=\"Músico\".\"Código Grupo\_Grupo\" inner join \"muchos\_Concierto\_tiene\_muchos\_Grupo\" on \n"

+ "\"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código Grupo\_Grupo\"=\"Grupo\".\"Código Grupo\" inner join \"Concierto\" on\n"

+ "\"Concierto\".\"Código\"=\"muchos\_Concierto\_tiene\_muchos\_Grupo\".\"Código\_Concierto\" where \"Concierto\".\"Fecha\"<'1/1/2017 00:00:00' or \"Concierto\".\"Fecha\">'31/12/2018 23:59:59';";

//String query ="select \* from \"Grupo\" where \"Género\"='Jazz';";

Statement stmnt = conn.createStatement();

ResultSet rs = stmnt.executeQuery(query);

a = a + ("Listado con los nombres de las canciones y los nombres de músicos que las han \ncompuesto grabadas por grupos que no han dado ningún concierto en el año 2017\n");

a = a + ("============================\n");

while (rs.next()) {

a = a + (rs.getString(1).trim() + " "+rs.getString(2).trim() + " \n");

}

a = a + ("============================");

panel.setText(a);

} catch (SQLException ex) {

//panel.setText(ex.getMessage());

logger.log(Level.WARNING, "SQL Exception", ex);

}

}

public void consulta(String sql, JTextArea panel){

try{

Statement statement = conn.createStatement();

ResultSet rs = statement.executeQuery(sql);

String pantalla = "";

pantalla = pantalla + "Listado de la consulta SQL.\n";

pantalla = pantalla + "\*\*\*\*\*\*\*\*\*\*\*\n";

while(rs.next()){

for(int i=1; i<=rs.getMetaData().getColumnCount(); i++){

//Clausula if para que no se pongan por pantalla los espacios al final de fila

if(rs.getString(i)!=null){

if(i==rs.getMetaData().getColumnCount()){

pantalla = pantalla +" " + rs.getString(i)+ "\n";

}else{

pantalla = pantalla +" " + rs.getString(i);

}

}

}

}

panel.setText(pantalla);

}catch(SQLException e){

System.out.println(e.getMessage());

JOptionPane.showMessageDialog(null,e.getMessage());

}

}

}

**CÓDIGO FUENTE CLASE CONSULTAS**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package conexion;

import java.awt.Image;

import java.awt.image.ImageObserver;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Consultas extends javax.swing.JFrame {

Conexion c;

public Consultas(Conexion c){

this.c=c;

initComponents();

}

@Override

public int checkImage(Image image, ImageObserver io) {

return super.checkImage(image, io); //To change body of generated methods, choose Tools | Templates.

}

public Consultas() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

textConsulta = new javax.swing.JTextField();

buttonConsulta = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jLabel3 = new javax.swing.JLabel();

textNumeroConsultas = new javax.swing.JTextField();

buttonBuscar = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

textConsulta.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textConsultaActionPerformed(evt);

}

});

buttonConsulta.setText("Ejecutar consulta");

buttonConsulta.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

buttonConsultaActionPerformed(evt);

}

});

jLabel1.setText("Introduzca una consulta");

jLabel2.setText("Nº de consultas");

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

jLabel3.setText("Introduzca el nº de consulta");

textNumeroConsultas.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textNumeroConsultasActionPerformed(evt);

}

});

buttonBuscar.setText("Buscar consultas");

buttonBuscar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

buttonBuscarActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(67, 67, 67)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 592, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1)

.addComponent(textConsulta, javax.swing.GroupLayout.PREFERRED\_SIZE, 592, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 123, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(82, 82, 82)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(buttonConsulta, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(textNumeroConsultas)

.addComponent(buttonBuscar))

.addContainerGap(209, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(113, 113, 113)

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(buttonConsulta)

.addComponent(textConsulta, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(27, 27, 27)

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(textNumeroConsultas, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(buttonBuscar))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 265, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(64, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void textConsultaActionPerformed(java.awt.event.ActionEvent evt) {

}

private void buttonBuscarActionPerformed(java.awt.event.ActionEvent evt) {

c.connect();

if (textNumeroConsultas.getText().equals("1")) {

c.getConsulta1(jTextArea1);

}

if (textNumeroConsultas.getText().equals("2")) {

c.getConsulta2(jTextArea1);

}

if (textNumeroConsultas.getText().equals("3")) {

c.getConsulta3(jTextArea1);

}

if (textNumeroConsultas.getText().equals("4")) {

c.getConsulta4(jTextArea1);

}

if (textNumeroConsultas.getText().equals("5")) {

c.getConsulta5(jTextArea1);

}

if (textNumeroConsultas.getText().equals("6")) {

c.getConsulta6(jTextArea1);

}

if (textNumeroConsultas.getText().equals("7")) {

c.getConsulta7(jTextArea1);

}

if (textNumeroConsultas.getText().equals("8")) {

c.getConsulta8(jTextArea1);

}

if (textNumeroConsultas.getText().equals("9")) {

c.getConsulta9(jTextArea1);

}

if (textNumeroConsultas.getText().equals("10")) {

c.getConsulta10(jTextArea1);

}

if (textNumeroConsultas.getText().equals("11")) {

c.getConsulta11(jTextArea1);

}

if (textNumeroConsultas.getText().equals("12")) {

c.getConsulta12(jTextArea1);

}

if (textNumeroConsultas.getText().equals("13")) {

c.getConsulta13(jTextArea1);

}

if (textNumeroConsultas.getText().equals("14")) {

c.getConsulta14(jTextArea1);

}

if (textNumeroConsultas.getText().equals("15")) {

c.getConsulta15(jTextArea1);

}

if (textNumeroConsultas.getText().equals("16")) {

c.getConsulta16(jTextArea1);

}

}

private void buttonConsultaActionPerformed(java.awt.event.ActionEvent evt) {

c.connect();

c.consulta(textConsulta.getText(), jTextArea1);

}

private void textNumeroConsultasActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Consultas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Consultas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Consultas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Consultas.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

@Override

public void run() {

new Consultas().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton buttonBuscar;

private javax.swing.JButton buttonConsulta;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextField textConsulta;

private javax.swing.JTextField textNumeroConsultas;

// End of variables declaration

}

**CREACIÓN DE ROLES, GRUPOS Y USUARIOS**

**CÓDIGO FUENTE SQL**

Creamos 5 grupos de usuarios (Administrador, gestor, cliente, musico y mánager) y 5 tipos de usuarios (boss, gestor, mariaonzalez@gmail.com, JavierMartin, PacoElFlaco).

create role "ADMINISTRADOR" superuser inherit createrole noreplication;

create role "GESTOR" nosuperuser inherit createdb nocreaterole noreplication;

create role "CLIENTE" nosuperuser inherit createdb nocreaterole noreplication;

create role "MUSICO" nosuperuser inherit createdb nocreaterole noreplication;

create role "MANAGER" nosuperuser inherit createdb nocreaterole noreplication;

create user "boss" with password 'boss';

grant "ADMINISTRADOR" to "boss";

grant all on all tables in schema public to "boss";

create user "gestor" with password 'gestor';

grant "GESTOR" to "gestor";

grant select,insert,delete,update on all tables in schema public to "gestor";

create user "mariaonzalez@gmail.com" with password 'pepe99';

grant "CLIENTE" to "mariaonzalez@gmail.com";

grant INSERT("Opinion concierto","Opinion Disco"),update("Opinion concierto","Opinion Disco"),

select("Opinion concierto","Opinion Disco") on "Opinion" to "mariaonzalez@gmail.com";

grant select on "Disco" to "mariaonzalez@gmail.com";

grant select on "Entrada" to "mariaonzalez@gmail.com";

grant update("Nombre", "e-mail"), select on "Usuario" to "mariaonzalez@gmail.com";

create user "JavierMartin" with password 'javiermartin';

grant "MUSICO" TO "JavierMartin";

grant select on "Músico" to "JavierMartin";

create user "PacoElFlaco" with password 'pacoelflaco';

grant "MANAGER" to "PacoElFlaco";

grant select on "Concierto", "Músico", "Entrada", "Disco" to "PacoElFlaco";

grant insert on "Concierto" to "PacoElFlaco";

**CONEXIÓN CON PROGRAMAS EXTERNOS Y SEGURIDAD**

Para conectar a la base de datos a través del login, utilizamos un booleano privado que ponemos a true cuando llamamos al método connect(). Creamos también su método getter.

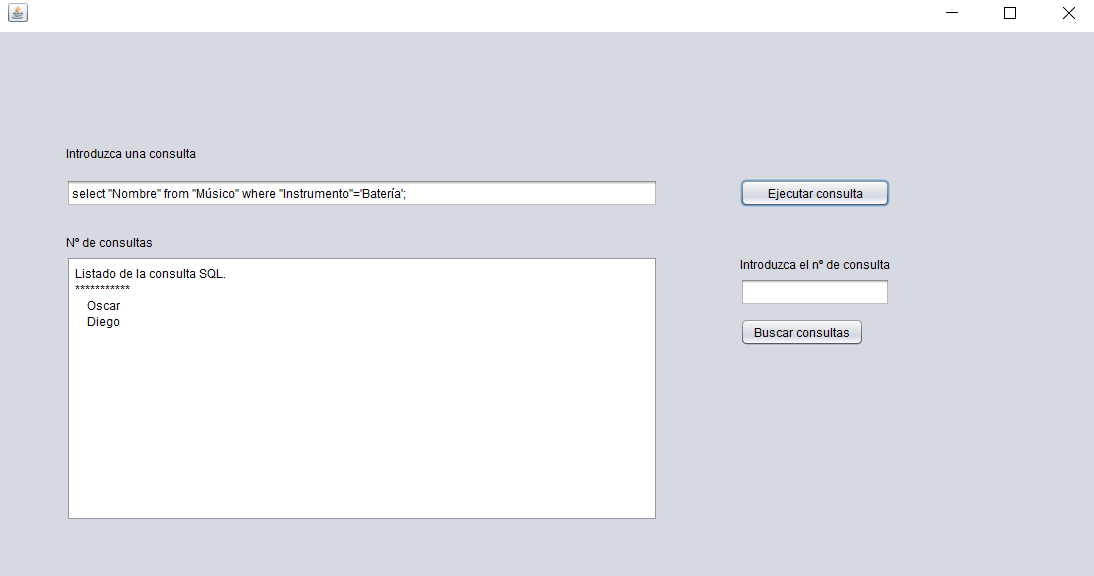
En la clase Login, hacemos un método para iniciar sesión, en el que pasamos por el constructor de la clase Conexión, el usuario y la contraseña introducida por pantalla. Llamamos al método del booleano y cuando sea true (usuario y contraseña correctos), aparece por pantalla la interfaz de la clase Consultas, donde se pueden realizar las consultas dependiendo de los permisos del usuario introducido.

**Ejemplo:**

Introducimos el usuario Boss (todos los permisos)



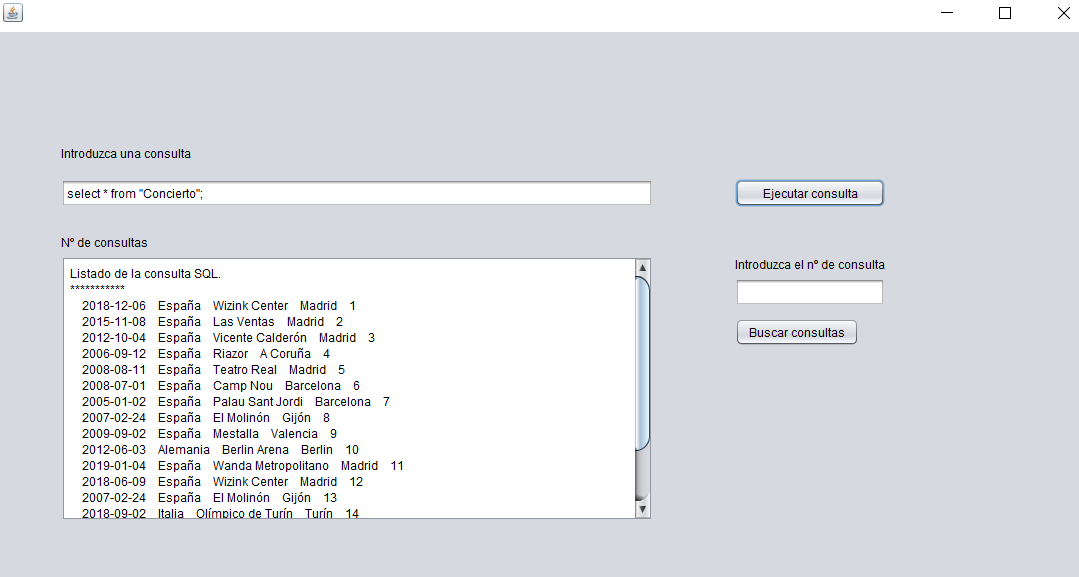
Ejecuta cualquier consulta sin problema



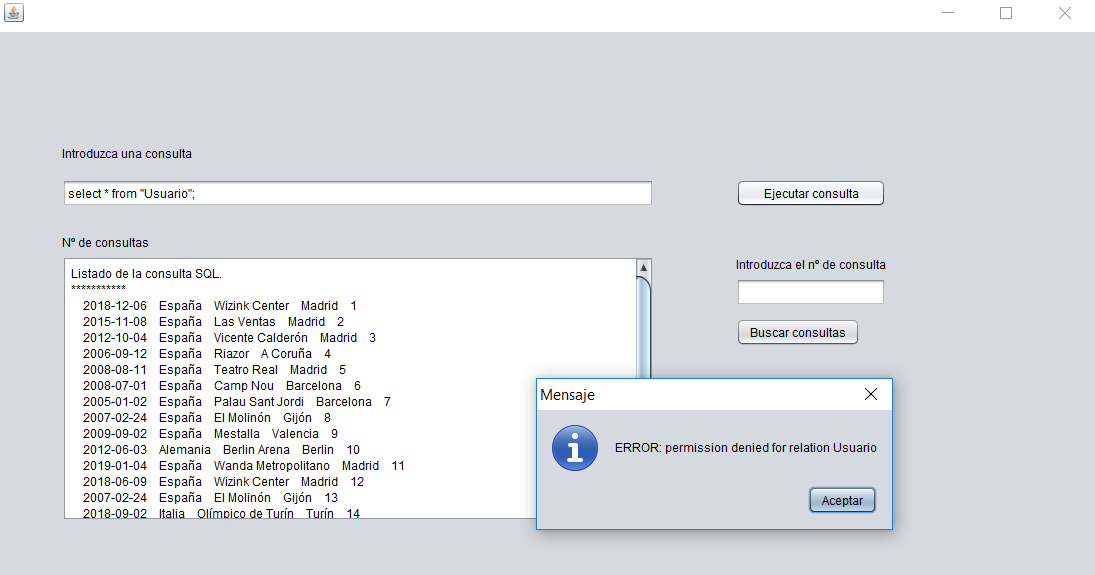
Introducimos el usuario PacoElFlaco, con los permisos select en "Concierto", "Músico", "Entrada", "Disco" e insert en "Concierto"



Hace select en Concierto sin ningún problema



Ahora cuando intentamos hacer select en Usuario (permiso no disponible) nos salta un Error que nos deniega el permiso.



**CÓDIGO FUENTE CLASE LOGIN**

package conexion;

import javax.swing.JOptionPane;

public class Login extends javax.swing.JFrame {

Conexion c;

public void iniciarSesion(String user, String password) {

c = new Conexion(user, password);

c.connect();

if (c.getB()) {

Consultas consulta = new Consultas(c);

consulta.setVisible(true);

this.setVisible(false);

} else {

JOptionPane.showMessageDialog(null, "Usuario y/o contraseña incorrectos");

textPassword.setText("");

textUsuario.setText("");

}

}

public Login() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

textUsuario = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

buttonIniciar = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

textPassword = new javax.swing.JPasswordField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setText("Usuario:");

textUsuario.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

textUsuarioActionPerformed(evt);

}

});

jLabel2.setText("Contraseña:");

buttonIniciar.setText("Iniciar");

buttonIniciar.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

buttonIniciarActionPerformed(evt);

}

});

jLabel3.setText("INICIO DE SESIÓN ");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(61, 61, 61)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 66, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(textUsuario, javax.swing.GroupLayout.DEFAULT\_SIZE, 188, Short.MAX\_VALUE)

.addComponent(textPassword)))

.addGroup(layout.createSequentialGroup()

.addGap(323, 323, 323)

.addComponent(jLabel3)))

.addContainerGap(322, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(buttonIniciar)

.addGap(334, 334, 334))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(21, 21, 21)

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 28, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(textUsuario, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(textPassword, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(43, 43, 43)

.addComponent(buttonIniciar)

.addGap(93, 93, 93))

);

pack();

}// </editor-fold>

private void textUsuarioActionPerformed(java.awt.event.ActionEvent evt) {

}

private void buttonIniciarActionPerformed(java.awt.event.ActionEvent evt) {

String usuario = textUsuario.getText();

String contraseña = textPassword.getText();

if (usuario.equals("") || contraseña.equals("")) {

JOptionPane.showMessageDialog(null, "Introduzca los campos usuario y contraseña");

}

else {

iniciarSesion(usuario, contraseña);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton buttonIniciar;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPasswordField textPassword;

private javax.swing.JTextField textUsuario;

// End of variables declaration

}