

Nombre: Richard Padilla

1. **Crear la ventana de login.**
2. **Conexión a la base de datos.**
3. **Mostrar datos y realizar inserciones.**
4. **Registrar usuarios con validación de edad.**

Login:

- El botón de **login** verifica las credenciales de usuario con la base de datos. Si son correctas, muestra un mensaje de éxito y cambia a la vista de datos.

Registro de usuario:

- Si el usuario tiene una edad válida (mayor de 18 y hasta 40), se registra en la base de datos en la tabla UsuariosNatacion.

Mostrar datos:

- Al presionar **Cargar Datos**, se ejecuta la consulta para cargar los usuarios registrados en la tabla UsuariosNatacion, y los muestra en el JTextArea.

Para mejorar las indicaciones de la tarea y que se comprendan mejor, podemos estructurarlas de forma clara y detallada, con un enfoque más descriptivo para cada paso en el contexto de la interfaz gráfica de Java (Swing) y las interacciones con la base de datos. Aquí tienes una versión mejorada:

Pasos para Implementar la Interfaz Gráfica de Login, Registro y Gestión de Datos en Java con Base de Datos

1. Crear la ventana de login

En este primer paso, debes diseñar una interfaz gráfica con **Swing** que incluya los siguientes elementos esenciales:

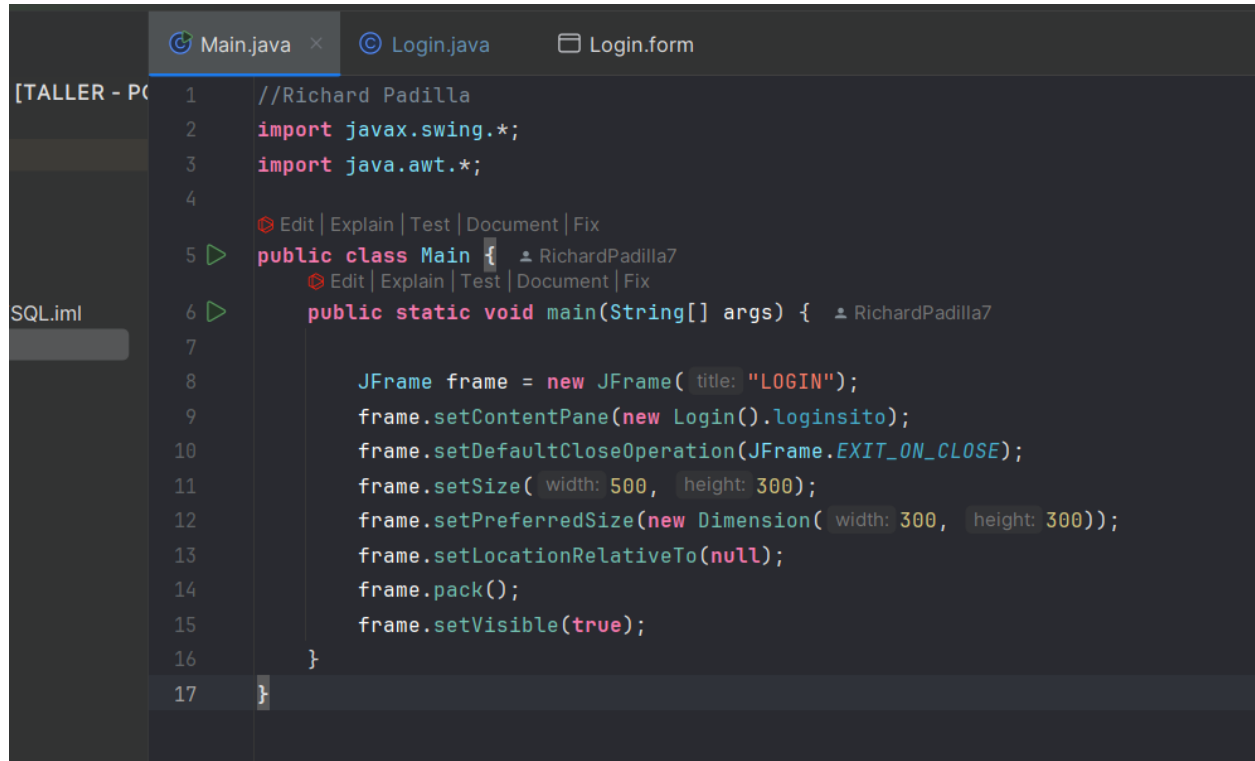
- **Campos de texto** para ingresar el **nombre de usuario** y la **contraseña**.
- **Botón de "Login"** para que el usuario intente ingresar al sistema.

Instrucciones:

- Utiliza un JTextField para el **nombre de usuario** y un JPasswordField para la **contraseña**.
- Crea un JButton que se utilizará para verificar las credenciales del usuario.

Comportamiento esperado:

- Cuando el usuario haga clic en el botón de **Login**, se debe verificar si las credenciales coinciden con los datos almacenados en la base de datos. Si la validación es exitosa, el sistema cambiará a una nueva vista donde se gestionarán los datos.



```
1 //Richard Padilla
2 import javax.swing.*;
3 import java.awt.*;
4
5 public class Main {
6     public static void main(String[] args) {
7
8         JFrame frame = new JFrame( title: "LOGIN");
9         frame.setContentPane(new Login().loginsito);
10        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        frame.setSize( width: 500, height: 300);
12        frame.setPreferredSize(new Dimension( width: 300, height: 300));
13        frame.setLocationRelativeTo(null);
14        frame.pack();
15        frame.setVisible(true);
16    }
17 }
```

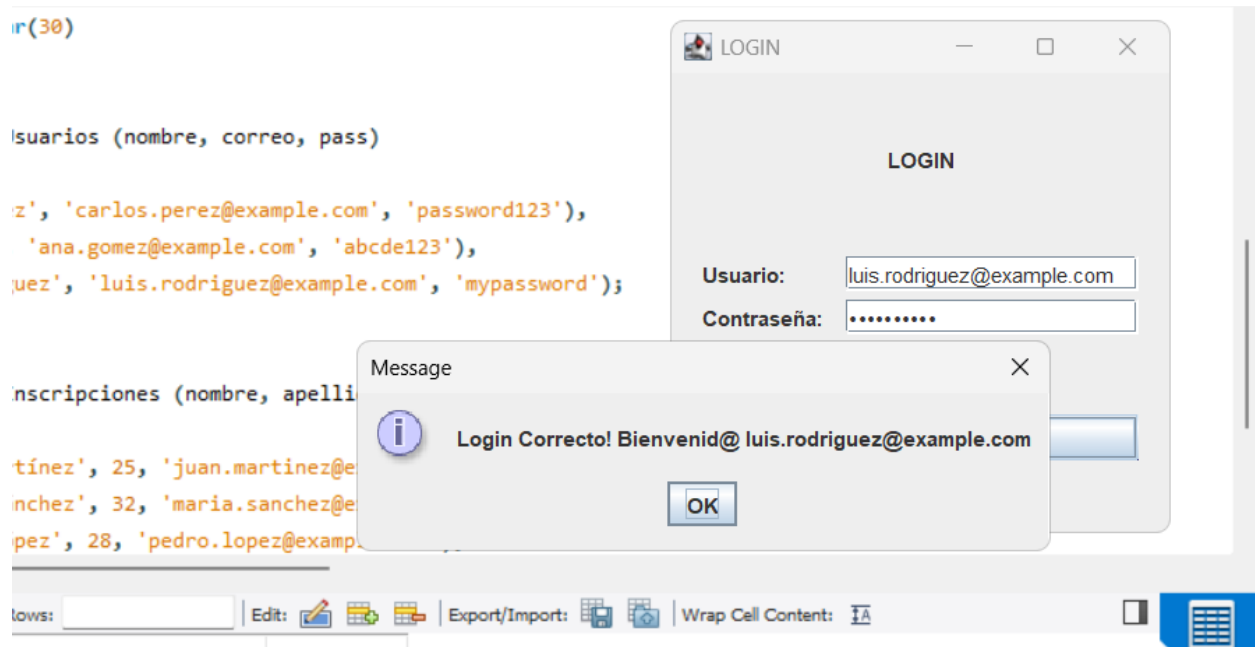
```
10 public class Login { 4 usages 1 RichardPadilla7 *
11 public JButton iniciarSesionButton; 3 usages
12
13
14 public JPanel loginsito; 5 usages
15 public JLabel titulo; 2 usages
16
17
18 public Login() { 2 usages 1 RichardPadilla7 *
19     iniciarSesionButton.addActionListener(new ActionListener() { 1 RichardPadilla7 *
20         @Override 1 RichardPadilla7 *
21         public void actionPerformed(ActionEvent e) {
22             String url = "jdbc:mysql://localhost:3306/Concursos";
23             String user = "root";
24             String password = "123456";
25
26             try (Connection conexionMySQL = DriverManager.getConnection(url, user, password)) {
27
28                 String query = "SELECT * FROM Usuarios WHERE correo = ? AND pass = ?";
29                 PreparedStatement statement = conexionMySQL.prepareStatement(query);
30
31                 statement.setString(1, textField1.getText());
32                 statement.setString(2, passwordField1.getText());
33
34                 ResultSet rs = statement.executeQuery();
35                 if (rs.next()) {
36
37                     JOptionPane.showMessageDialog(parentComponent: null, message: "Login Correcto! Bienvenid@ " + textField1.getText());
38
39                     JFrame frame = new JFrame(title: "LOGIN");
40
41                     frame.setContentPane(new Login().loginsito);
42                     frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

ER - PC Component Tree

Form (Login)

- loginsito : JPanel
 - titulo : JLabel
 - "Usuario:" : JLabel
 - "Contraseña:" : JLabel
 - textField1 : JTextField
 - passwordField1 : JPasswordField
 - Vertical Spacer
 - Vertical Spacer
 - iniciarSesionButton : JButton
 - Vertical Spacer

Property	Value
bind to class	Login



2. Conexión a la base de datos

El siguiente paso es establecer una **conexión con la base de datos** que almacena los usuarios y sus credenciales. Esta conexión se realiza utilizando el **JDBC**.

Instrucciones:

- Usa `DriverManager.getConnection()` para conectarte a la base de datos.
- Configura la URL de la base de datos, el nombre de usuario y la contraseña para acceder a la base de datos MySQL.

Comportamiento esperado:

- Si la conexión es exitosa, podrás realizar consultas para verificar las credenciales del usuario y también insertar o mostrar datos en la base de datos.

```
PC 1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.sql.Connection;
6  import java.sql.DriverManager;
7  import java.sql.PreparedStatement;
8  import java.sql.ResultSet;
9
10  public class Login { 4 usages  RichardPadilla7 *
11      public JTextField textField1; 4 usages
12      public JPasswordField passwordField1; 3 usages
13      public JButton iniciarSesionButton; 3 usages
14      public JPanel loginsito; 5 usages
15      public JLabel titulo; 2 usages
16
17  public Login() { 2 usages  RichardPadilla7 *
18      iniciarSesionButton.addActionListener(new ActionListener() {  RichardPadilla7 *
19          @Override  RichardPadilla7 *
20          public void actionPerformed(ActionEvent e) {
21              String url = "jdbc:mysql://localhost:3306/Concursos";
22              String user = "root";
23              String password = "123456";
24
25              try (Connection conexionMySQL = DriverManager.getConnection(url, user, password)) {
26
27                  String query = "SELECT * FROM Usuarios WHERE correo = ? AND pass = ?";
28                  PreparedStatement statement = conexionMySQL.prepareStatement(query);
29
30                  statement.setString(1, textField1.getText());
```

```
iniciarSesionButton.addActionListener(new ActionListener() {  RichardPadilla7 *
    public void actionPerformed(ActionEvent e) {
        String url = "jdbc:mysql://localhost:3306/Concursos";
        String user = "root";
        String password = "123456";

        try (Connection conexionMySQL = DriverManager.getConnection(url, user, password)) {

            String query = "SELECT * FROM Usuarios WHERE correo = ? AND pass = ?";
            PreparedStatement statement = conexionMySQL.prepareStatement(query);

            statement.setString(1, textField1.getText());
            statement.setString(2, passwordField1.getText());

            ResultSet rs = statement.executeQuery();
            if (rs.next()) {

                JOptionPane.showMessageDialog(parentComponent: null, message: "Login Correcto! Bienvenid@ " + textField1.getText());

                JFrame frame = new JFrame(title: "LOGIN");

                frame.setContentPane(new Login().loginsito);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                frame.setSize(width: 500, height: 300);
                frame.setPreferredSize(new Dimension(width: 300, height: 300));
                frame.setLocationRelativeTo(null);
                frame.pack();
                frame.setVisible(true);
            }
        }
    }
}
```

```

10 public class Login { 4 usages RichardPadilla7 *
17 public Login() { 2 usages RichardPadilla7 *
18     iniciarSesionButton.addActionListener(new ActionListener() { RichardPadilla7 *
20         public void actionPerformed(ActionEvent e) {
36             JOptionPane.showMessageDialog( parentComponent: null, message: "Login Correcto! Bienvenid@ " + textField
37
38             JFrame frame = new JFrame( title: "LOGIN");
39
40             frame.setContentPane(new Login().loginsito);
41             frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
42             frame.setSize( width: 500, height: 300);
43             frame.setPreferredSize(new Dimension( width: 300, height: 300));
44             frame.setLocationRelativeTo(null);
45             frame.pack();
46             frame.setVisible(true);
47
48             // Cerrar la ventana de login actual
49             JFrame loginFrame = (JFrame) SwingUtilities.getWindowAncestor(loginsito);
50             loginFrame.dispose();
51
52         } else {
53             JOptionPane.showMessageDialog( parentComponent: null, message: "Usuario o contraseña incorrectos");
54         }
55     } catch (Exception ex) {
56         JOptionPane.showMessageDialog( parentComponent: null, message: "Error de conexión en la base de datos!");
57         ex.printStackTrace();
58     }
59 }
60 });
61 }
62 }

```

3. Mostrar datos y realizar inserciones

Una vez conectado a la base de datos, debes ser capaz de mostrar los datos almacenados y permitir inserciones.

Mostrar datos:

- Crea un **JTextArea** o un **JTable** donde se mostrarán los datos de los usuarios registrados en la tabla UsuariosNatacion de la base de datos.
- Al presionar un botón de **Cargar Datos**, se ejecuta una consulta SELECT que extrae los datos de la base de datos y los presenta en la interfaz gráfica.

Realizar inserciones:

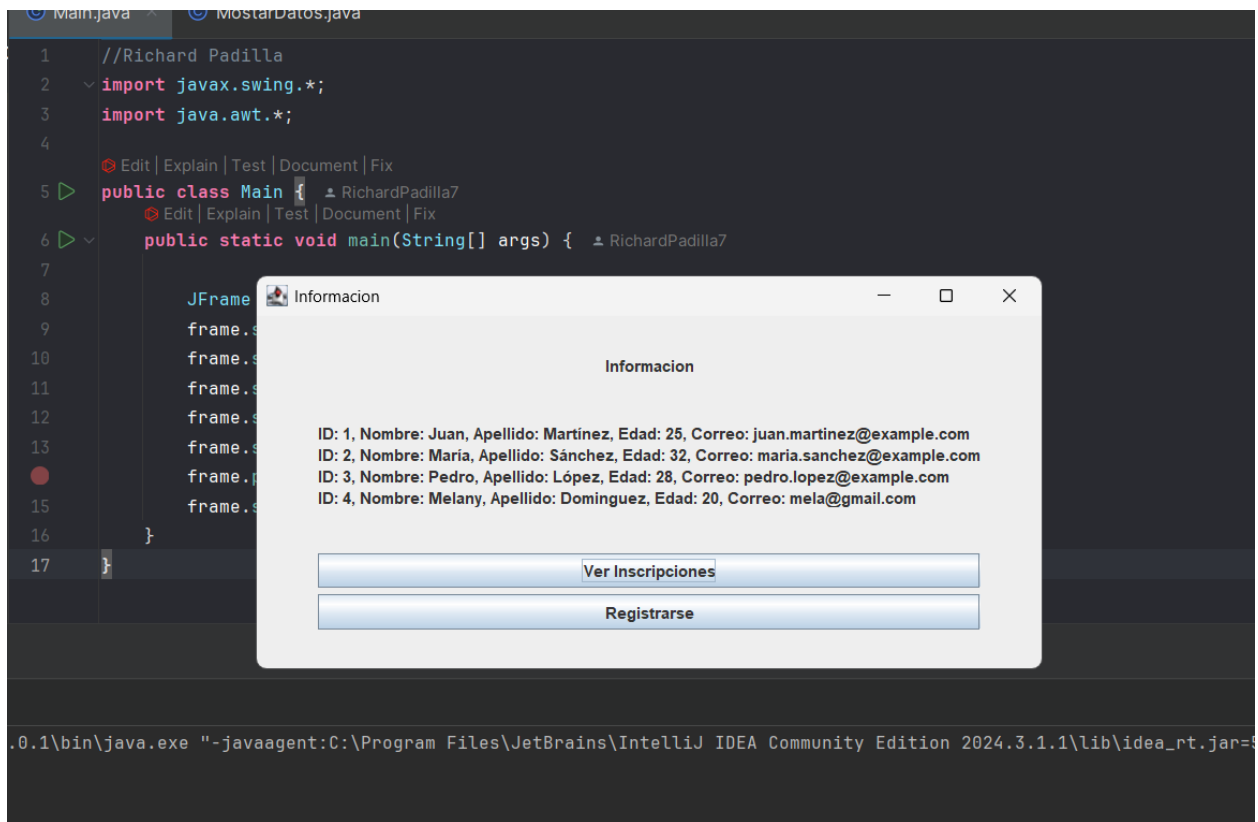
- Utiliza un formulario de registro con **campos de texto** para ingresar el **nombre** y la **edad** de los usuarios que quieren participar en natación.
- Al hacer clic en el botón **Registrar**, verifica si la edad del usuario es válida (mayor de 18 y hasta 40 años).

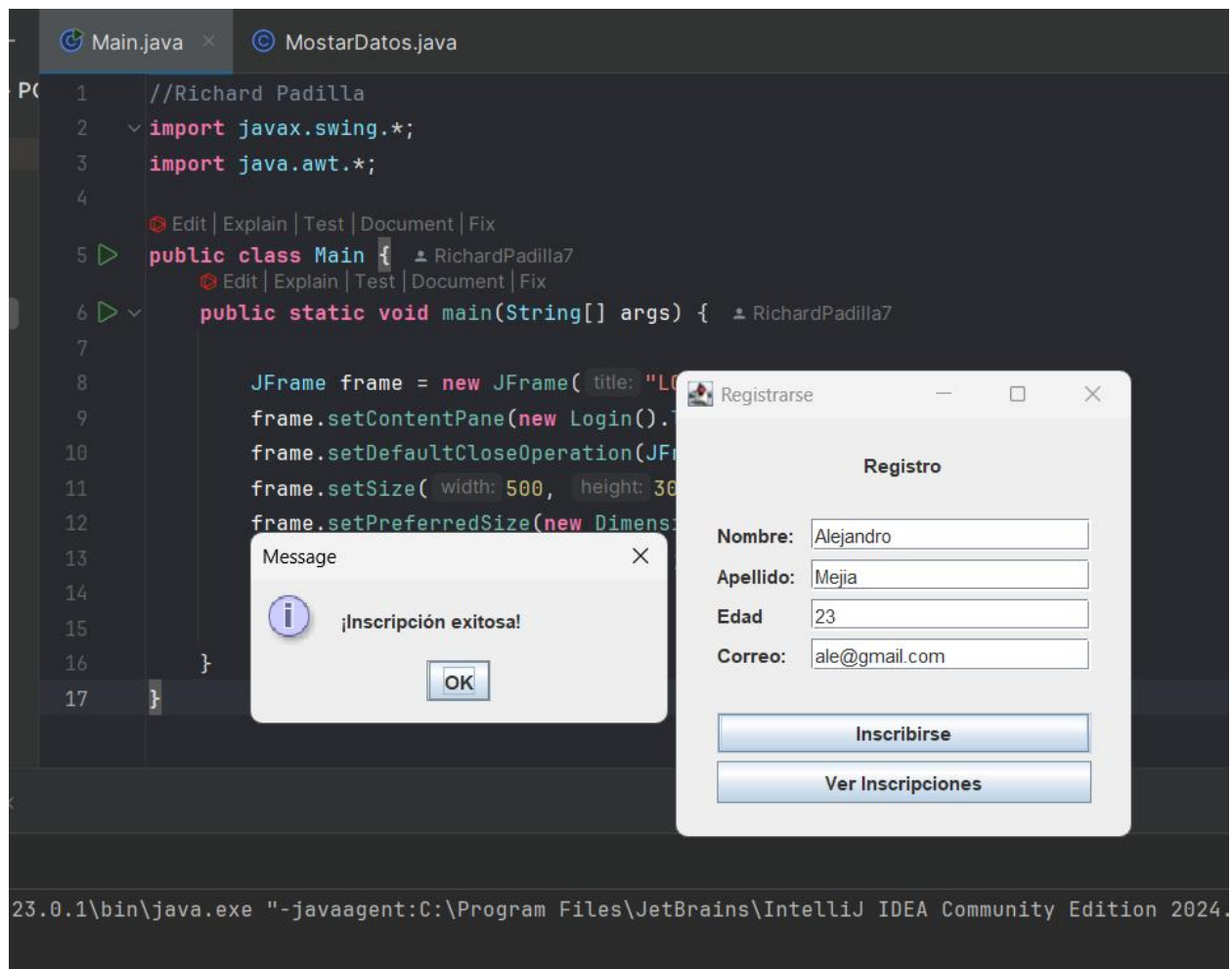
Comportamiento esperado:

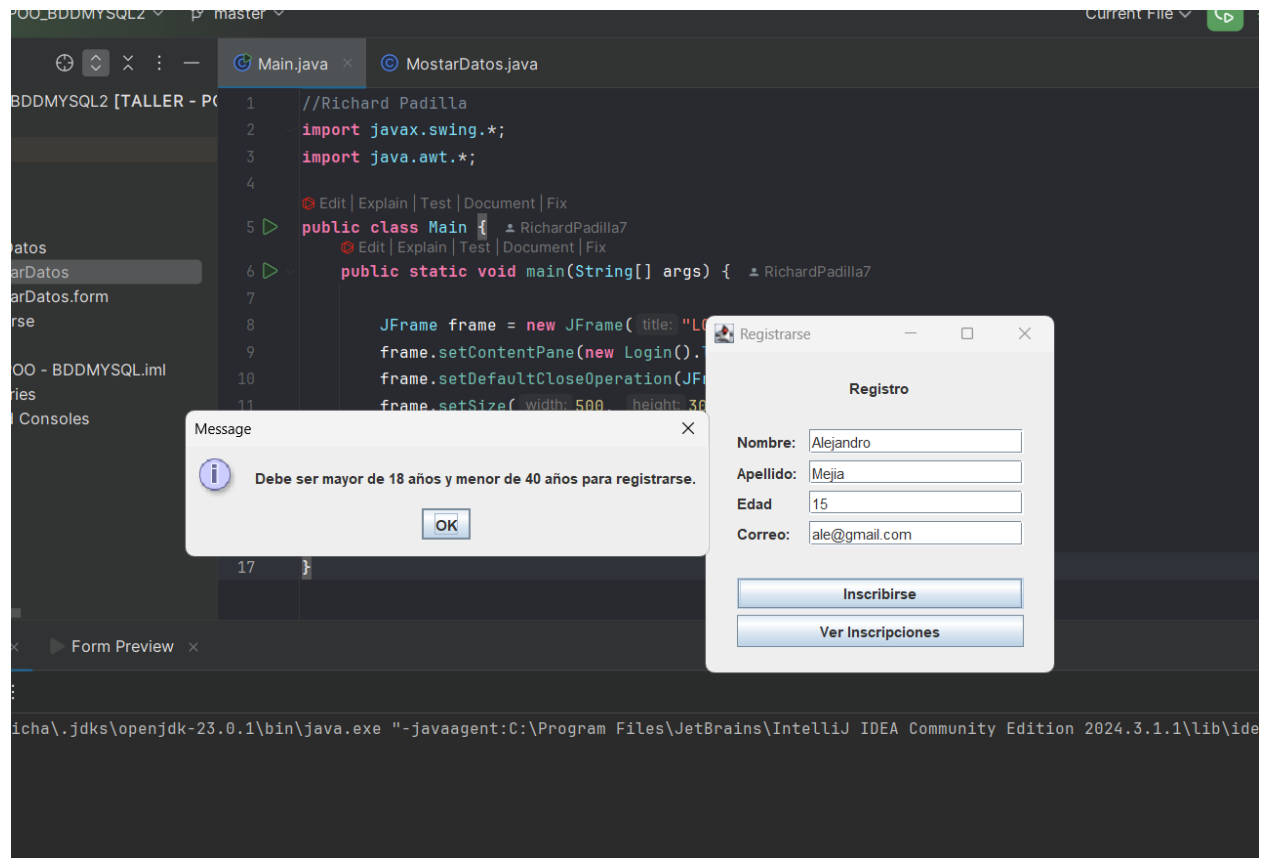
- Los datos extraídos de la base de datos se mostrarán en un área de texto o tabla.

- Los nuevos registros se insertarán en la base de datos cuando la edad del usuario esté dentro del rango permitido.

```
12 public class MostarDatos { 4 usages 1 RichardPadilla7
18 public MostarDatos() { 2 usages 1 RichardPadilla7
19     verInscripcionesButton.addActionListener(new ActionListener() { 1 RichardPadilla7
21         public void actionPerformed(ActionEvent e) {
25
26             try (Connection conexionMySQL = DriverManager.getConnection(url, user, password)) {
27                 String query = "SELECT * FROM Inscripciones";
28                 PreparedStatement statement = conexionMySQL.prepareStatement(query);
29                 ResultSet rs = statement.executeQuery();
30
31                 // Construir un String con los resultados
32                 StringBuilder resultados = new StringBuilder("<html>");
33                 while (rs.next()) {
34                     resultados.append("ID: ").append(rs.getString( columnLabel: "cod"))
35                     .append(", Nombre: ").append(rs.getString( columnLabel: "nombre"))
36                     .append(", Apellido: ").append(rs.getString( columnLabel: "apellido"))
37                     .append(", Edad: ").append(rs.getString( columnLabel: "edad"))
38                     .append(", Correo: ").append(rs.getString( columnLabel: "correo"))
39                     .append("<br>");
40                 }
41                 resultados.append("</html>");
42
43                 // Mostrar resultados en el JLabel
44                 VerDatos.setText(resultados.toString());
45
46             } catch (SQLException ex) {
47                 JOptionPane.showMessageDialog( parentComponent: null, message: "Error en la base de datos!");
48             }
49         }
50     }
51 }
```







4. Registrar usuarios con validación de edad

Este paso es fundamental para asegurar que solo los usuarios con una edad válida puedan ser registrados en el sistema.

Instrucciones:

- El formulario de registro debe contar con un **campo de texto** para el **nombre del usuario** y un **campo de texto** para la **edad**.
- Después de que el usuario ingrese los datos, al presionar el botón **Registrar**, se debe verificar que la edad esté en el rango permitido (18 - 40 años).

Comportamiento esperado:

- Si la edad ingresada es válida, el sistema registrará al usuario en la base de datos, en la tabla UsuariosNatacion.
- Si la edad no es válida, el sistema mostrará un mensaje de error indicando que la edad debe estar entre 18 y 40 años.

Resumen de Comportamiento General

- **Login:** Verifica las credenciales del usuario con la base de datos. Si la validación es exitosa, cambia la vista a la pantalla de gestión de datos.
- **Registro de usuario:** Solo los usuarios con edades entre 18 y 40 años pueden ser registrados en la base de datos. Se muestra un mensaje de confirmación si el registro es exitoso.

- **Mostrar datos:** Al presionar el botón de **Cargar Datos**, se consulta la base de datos y se muestran los usuarios registrados en el área de texto o en una tabla.