



UM GUIA DE INFORMAÇÕES RÁPIDAS

PROJETO COM INTERFACE GRÁFICA (GUI)

Atividade 03 - Construção de uma tela de
login

Profº Claudio Benossi
Gabriel Henrique Santos Amorim

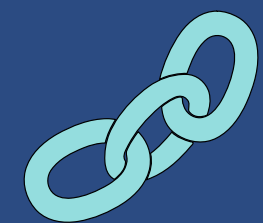


1a

**Download de
alguns recursos**

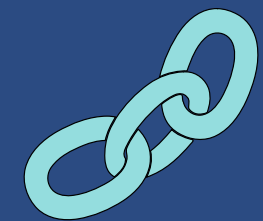
Acesse estes links:

1 - Driver MySQL para Java



<https://dev.mysql.com/downloads/connector/j/>

2 - Icones que vamos usar



https://github.com/amorim66/Login_Java

1 - Driver MySQL para Java

Selecione a opção
de Plataforma
Independente



MySQL Community Downloads

Connector/J

General Availability (GA) Releases

Archives



Connector/J 8.0.29

Select Operating System:

Select Operating System...

Select Operating System...

Microsoft Windows

Ubuntu Linux

Debian Linux

SUSE Linux Enterprise Server

Red Hat Enterprise Linux / Oracle Linux

Fedora

Platform Independent

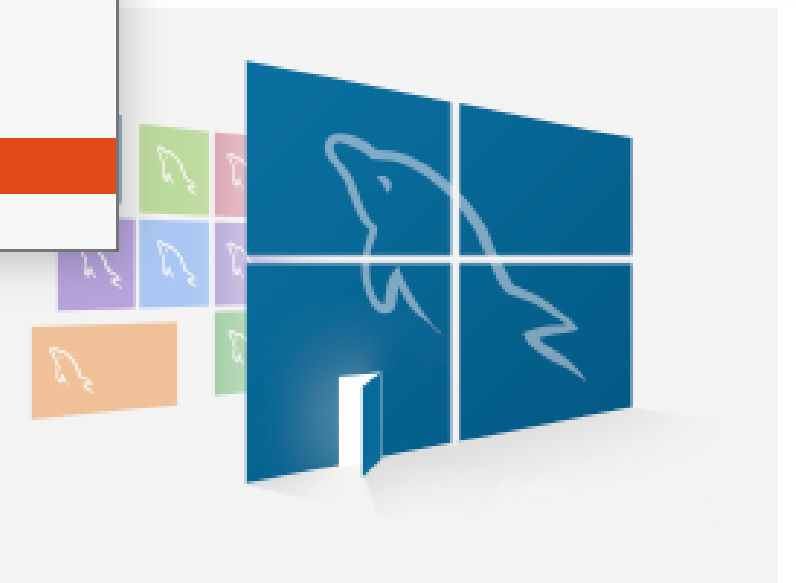
Source Code

All MySQL Products. For All Windows Platforms.
In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

Go to Download Page >



1 - Driver MySQL para Java

Selecione a
segunda
opção .zip



MySQL Community Downloads

Connector/J

General Availability (GA) Releases

Archives



Connector/J 8.0.29

Select Operating System:

Platform Independent

Platform Independent (Architecture Independent),
Compressed TAR Archive

(mysql-connector-java-8.0.29.tar.gz)

8.0.29

4.1M

Download

MD5: 25fea8c2e6b6ec630db7438bf6d29 | Signature

Platform Independent (Architecture Independent),
ZIP Archive

(mysql-connector-java-8.0.29.zip)

8.0.29

4.9M

Download

MD5: d1a49c0395d3b7d30322b766d4e63e51 | Signature



We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

2 - Icones que vamos usar

The screenshot shows the GitHub interface for the repository 'amorim66 / Login_Java'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted in green. A dropdown menu is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI, and a 'Download ZIP' button which is highlighted with a red box. A large blue arrow points to the 'Download ZIP' button.

Search or jump to...

Pull requests Issues Marketplace Explore

amorim66 / Login_Java Public

Pin Unwatch

Code Issues Pull requests Actions Projects Wiki Security Insights

main 1 branch 0 tags

Go to file Add file Code

amorim66 icones banco de dados

README.md	first commit
dberror.png	icones banco de dados
dbok.png	icones banco de dados

Help people interested in this repository understand your project by adding a README file.

Clone

HTTPS SSH GitHub CLI

https://github.com/amorim66/Login_Java

Use Git or checkout with SVN using the web URL.

Download ZIP

ATENÇÃO!

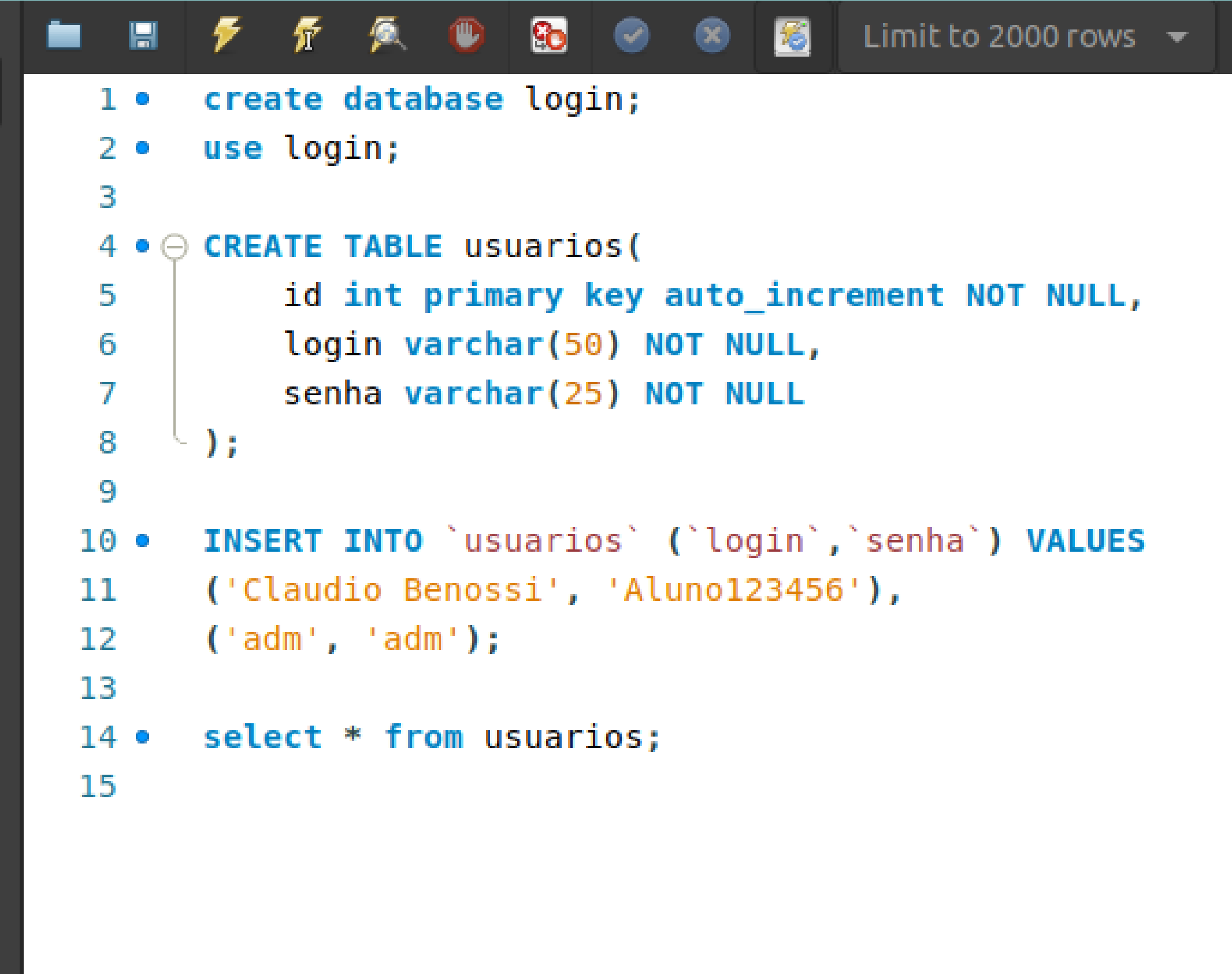
Salve os arquivos baixados na pasta que desejar, mas ambos precisam ser extraídos. Vamos acessá-los posteriormente.



2a

**Desenvolvendo
script SQL**

a) Abra o
MySQL e
Digite o
script ao
lado.

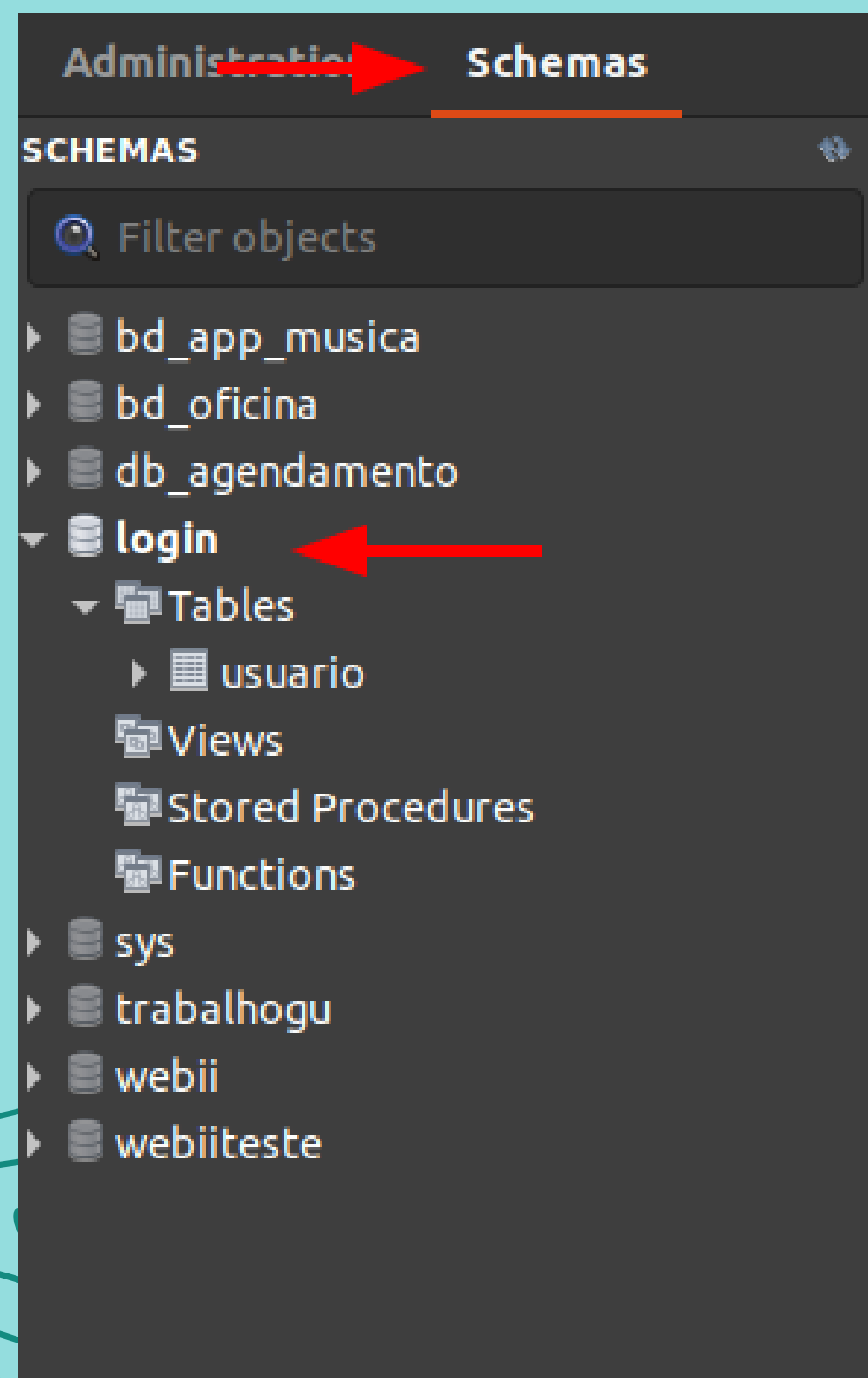


```
1 • create database login;
2 • use login;
3
4 • CREATE TABLE usuarios(
5     id int primary key auto_increment NOT NULL,
6     login varchar(50) NOT NULL,
7     senha varchar(25) NOT NULL
8 );
9
10 • INSERT INTO `usuarios` (`login`,`senha`) VALUES
11     ('Claudio Benossi', 'Aluno123456'),
12     ('adm', 'adm');
13
14 • select * from usuarios;
15
```

b) Para executar basta selecionar o script todo e clicar no primeiro raio.



```
1 • create database login;
2 • use login;
3
4 • CREATE TABLE usuarios(
5     id int primary key auto_increment NOT NULL,
6     login varchar(50) NOT NULL,
7     senha varchar(25) NOT NULL
8 );
9
10 • INSERT INTO `usuarios` (`login`,`senha`) VALUES
11     ('Claudio Benossi', 'Aluno123456'),
12     ('adm', 'adm');
13
14 • select * from usuarios;
15
```



OK! Se na barra lateral esquerda em "Schemas" já tiver o banco "Login" ativo, o script funcionou.

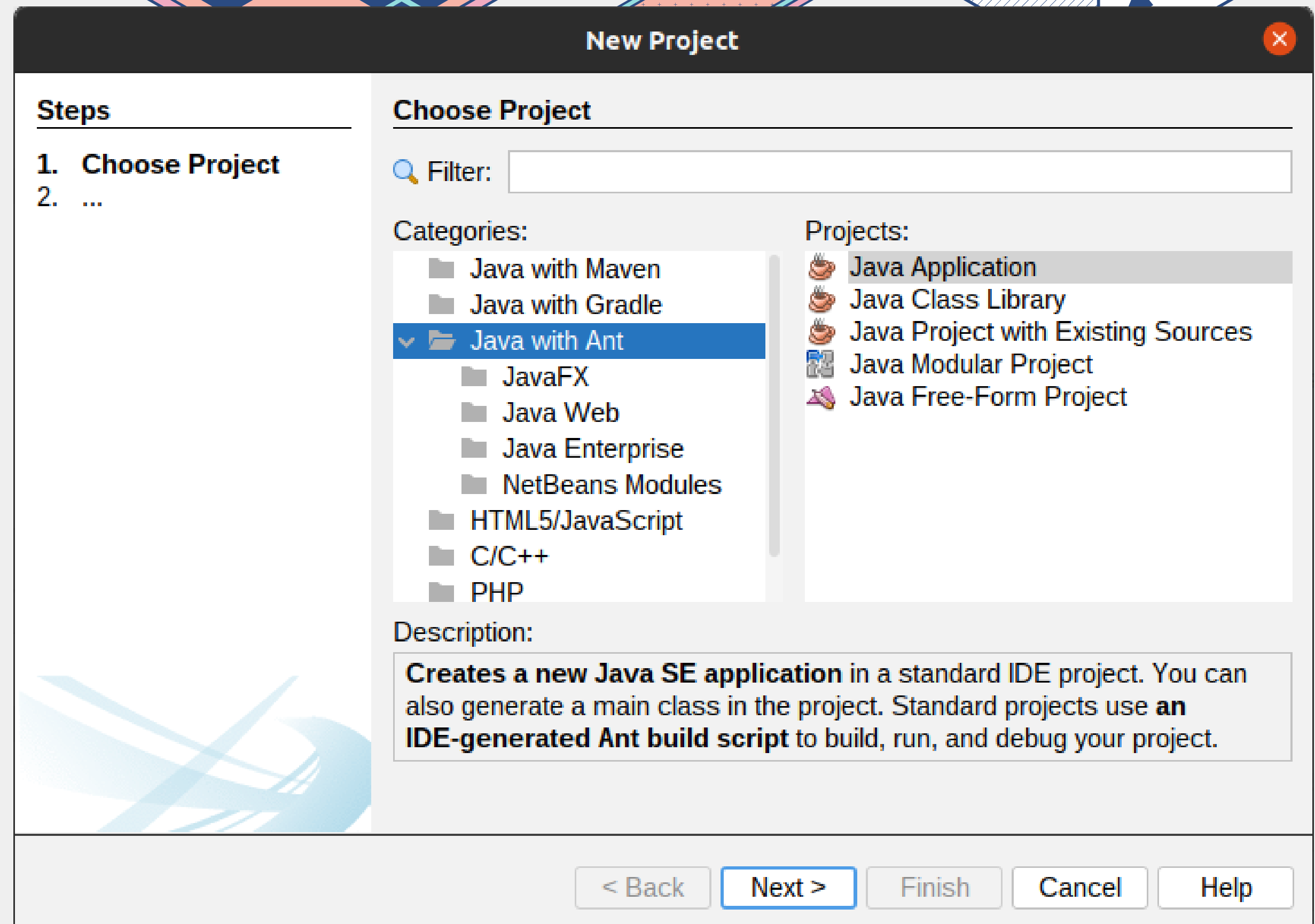
obs: é importante salvar o script caso queira criar a estrutura de novo.

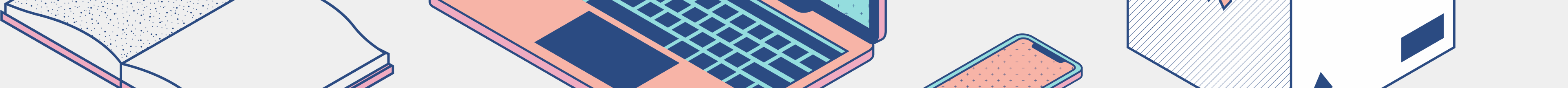


3ª

**Preparando
ambiente para
aplicação JAVA**

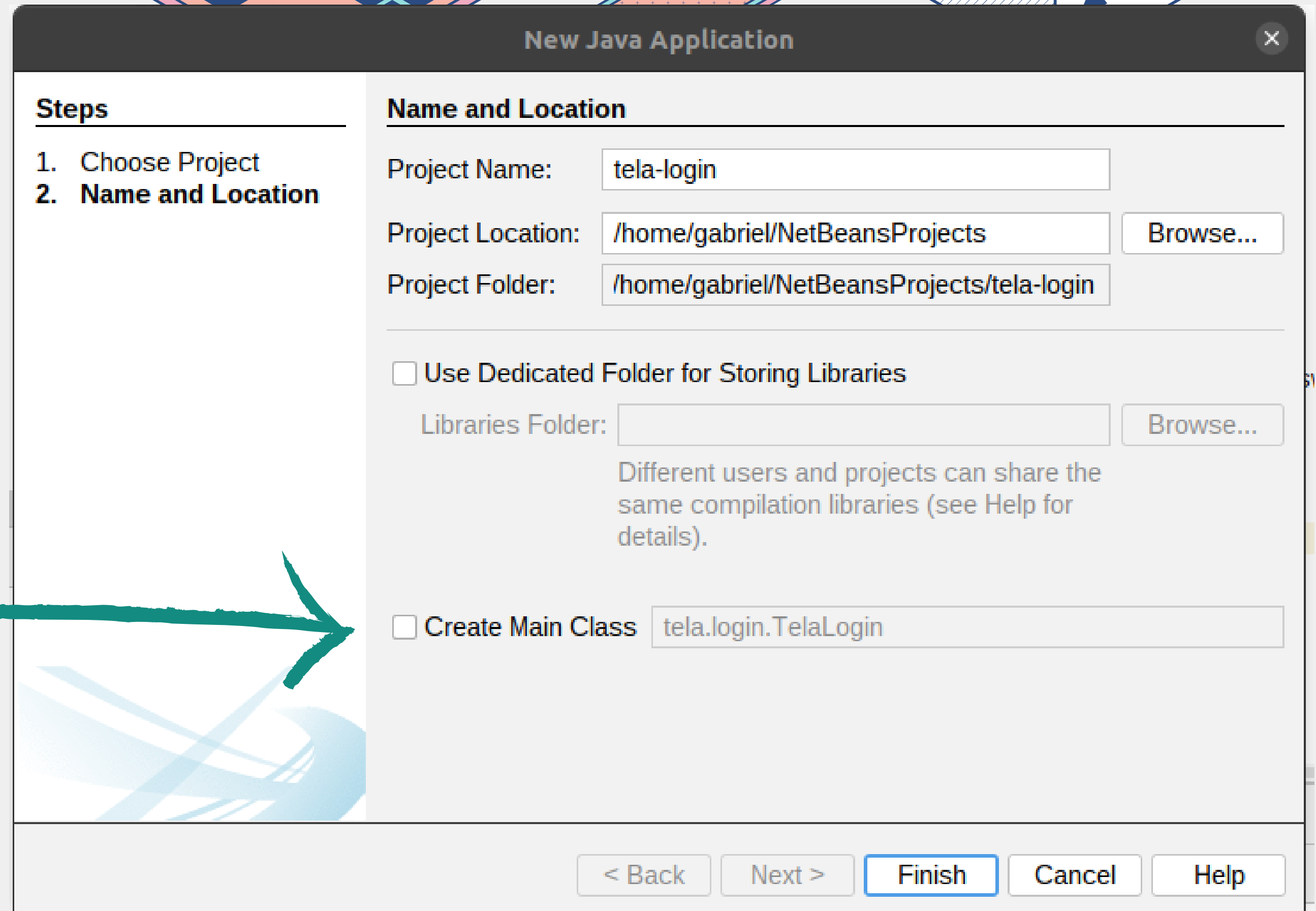

A) Escolha o tipo de projeto, no caso "Java Application"





B) Crie o novo projeto com o nome "tela-login"

Lembre-se de desmarcar a opção de criar classe principal



New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

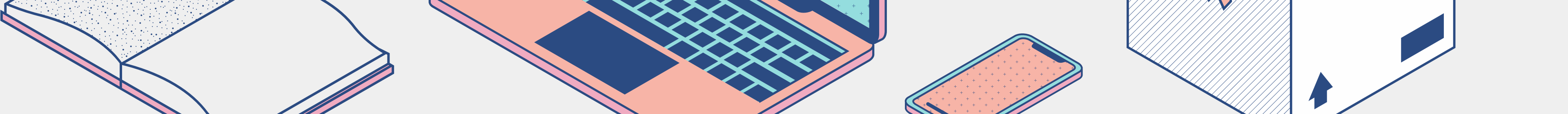
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☐ Create Main Class

< Back Next > **Finish** Cancel Help

D) Crie um pacote dentro do nosso projeto com o nome de "view".



New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name:

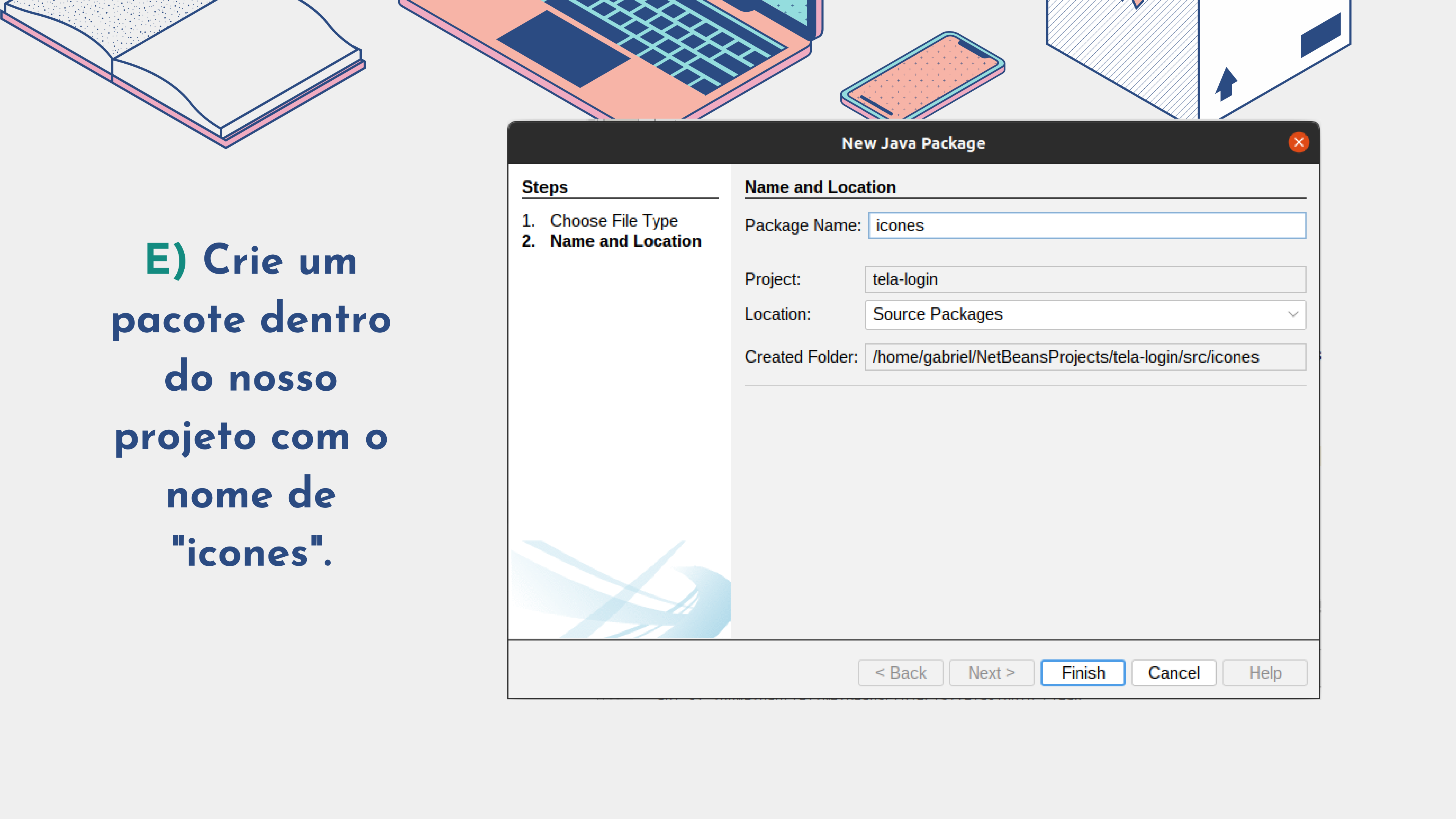
Project:

Location:

Created Folder:

< Back Next > **Finish** Cancel Help

E) Crie um pacote dentro do nosso projeto com o nome de "ícones".



New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name:

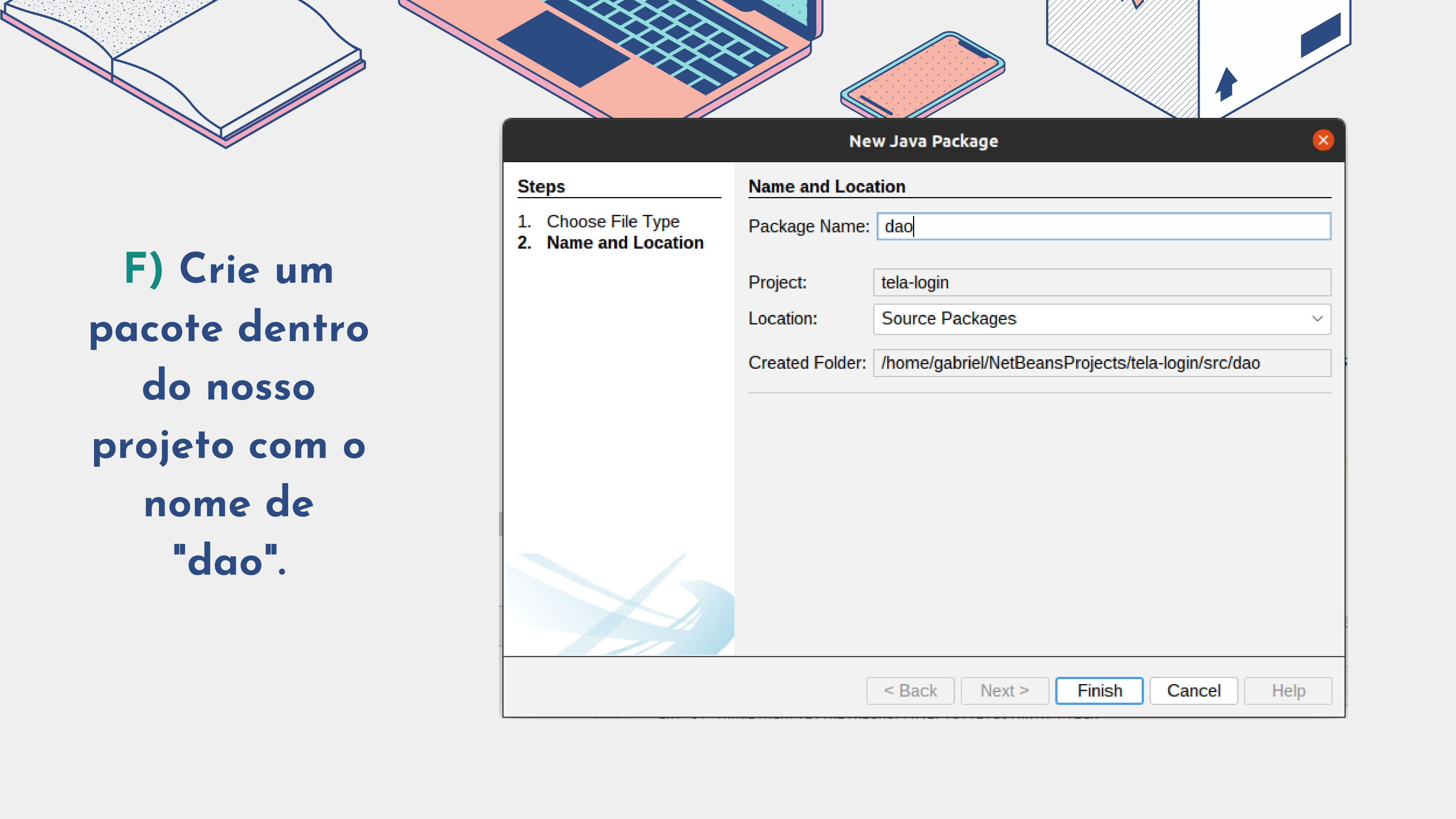
Project:

Location:

Created Folder:

< Back Next > **Finish** Cancel Help

F) Crie um pacote dentro do nosso projeto com o nome de "dao".



New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name:

Project:

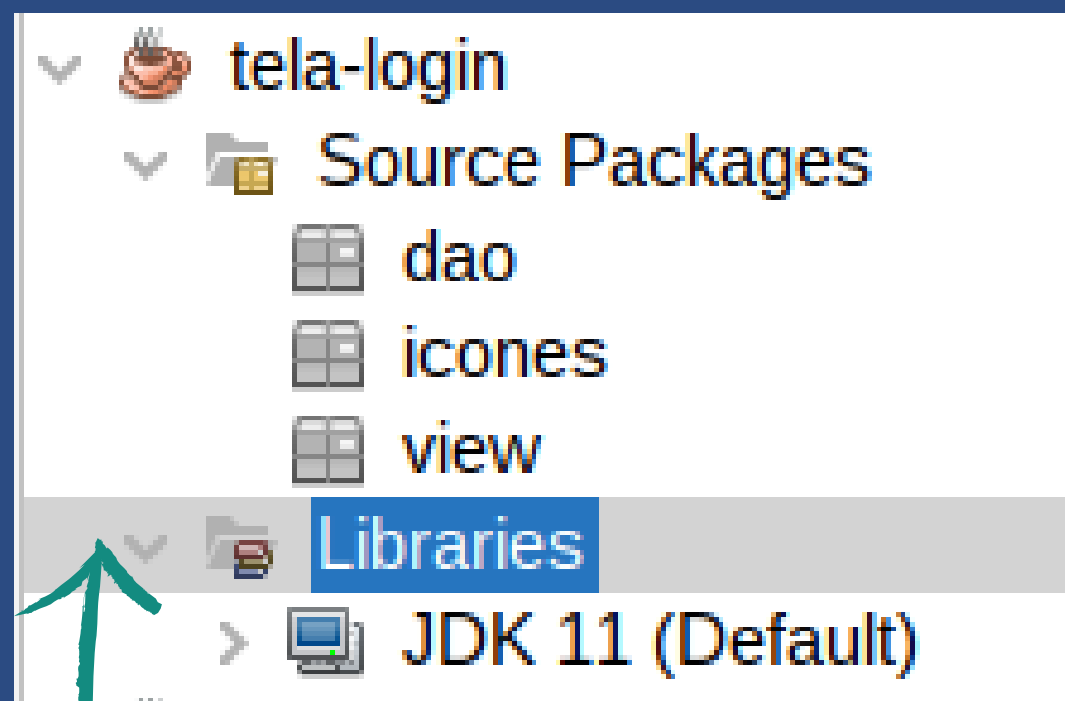
Location:

Created Folder:

< Back Next > **Finish** Cancel Help

Para finalizar essa etapa

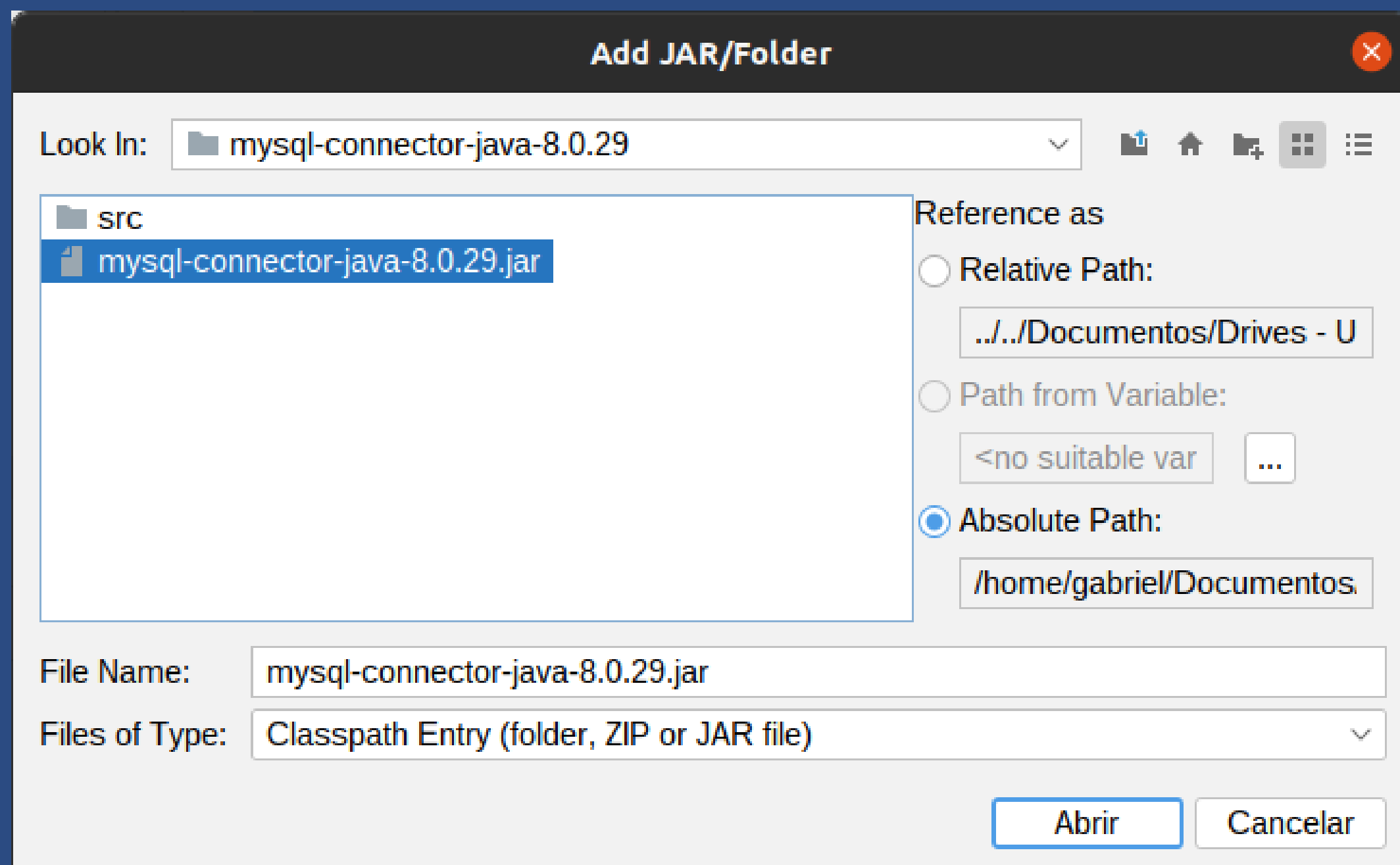
Precisamos importar o driver de conexão que baixamos lá etapa 1º.



a) Clique com o botão direito dentro de "Libraries". Escolha a opção "ADD JAR/Folder."

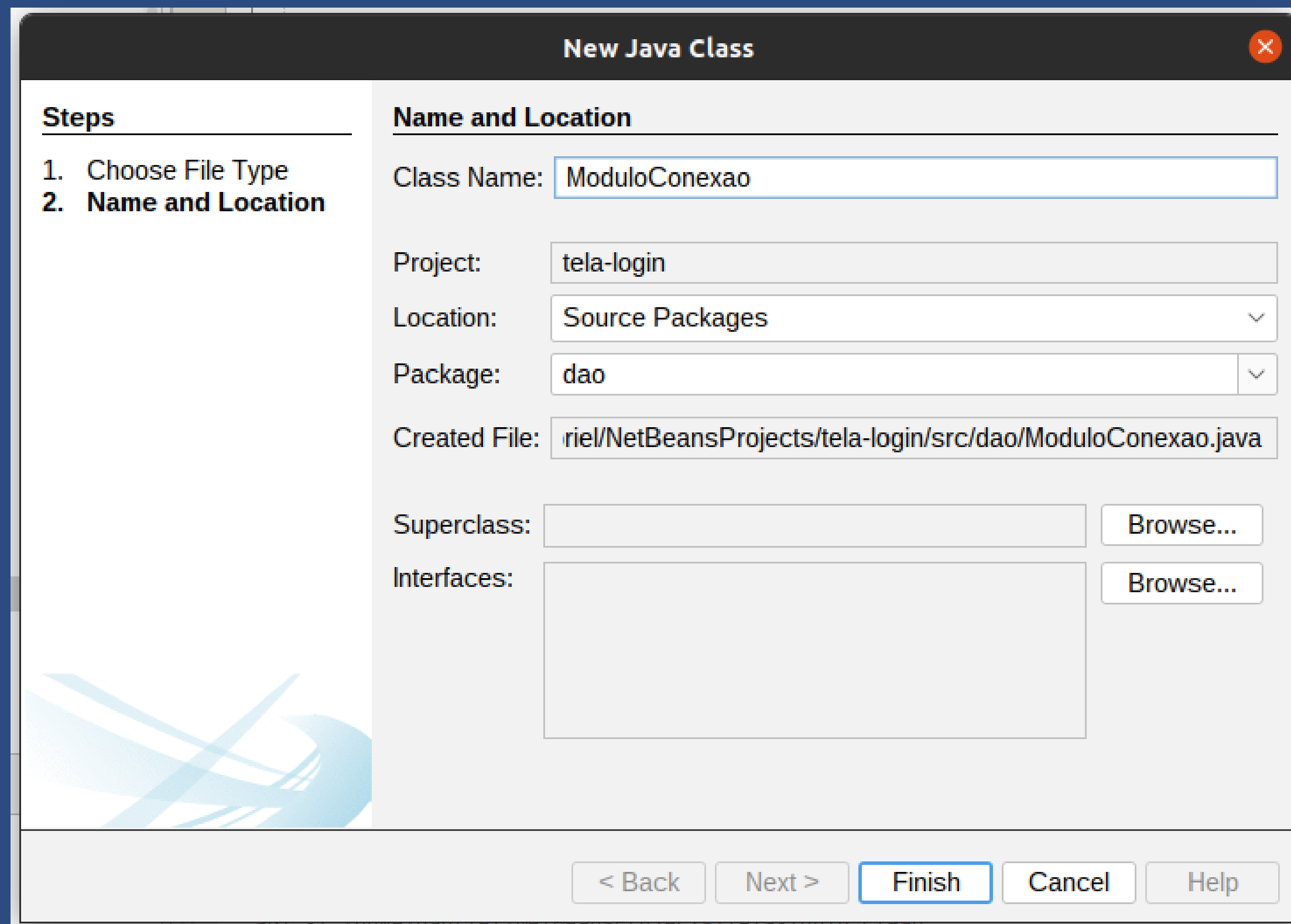
b)

Vá até a pasta onde foi extraído o download do conector e procure pelo arquivo .jar (como mostra abaixo). Clique em abrir.



c)

Crie uma nova classe dentro do pacote "dao" e a nomeie como "ModuloConexao".



New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

Superclass:

Interfaces:

d)

Dentro da classe "ModuloConexao", digite o código ao lado. Em "root" e "password", substitua por suas credenciais de acesso MySQL.

```
import java.sql.*;

public class ModuloConexao {
    public static Connection conector() {
        Connection conexao;
        String driver = "com.mysql.cj.jdbc.Driver";
        String url = "jdbc:mysql://localhost:3306/login";
        String user = "root";
        String password = "24017878";
        try {
            Class.forName(driver);
            conexao = DriverManager.getConnection(url, user, password);
            return conexao;
        } catch (Exception e) {
            System.out.println(e);
            return null;
        }
    }
}
```

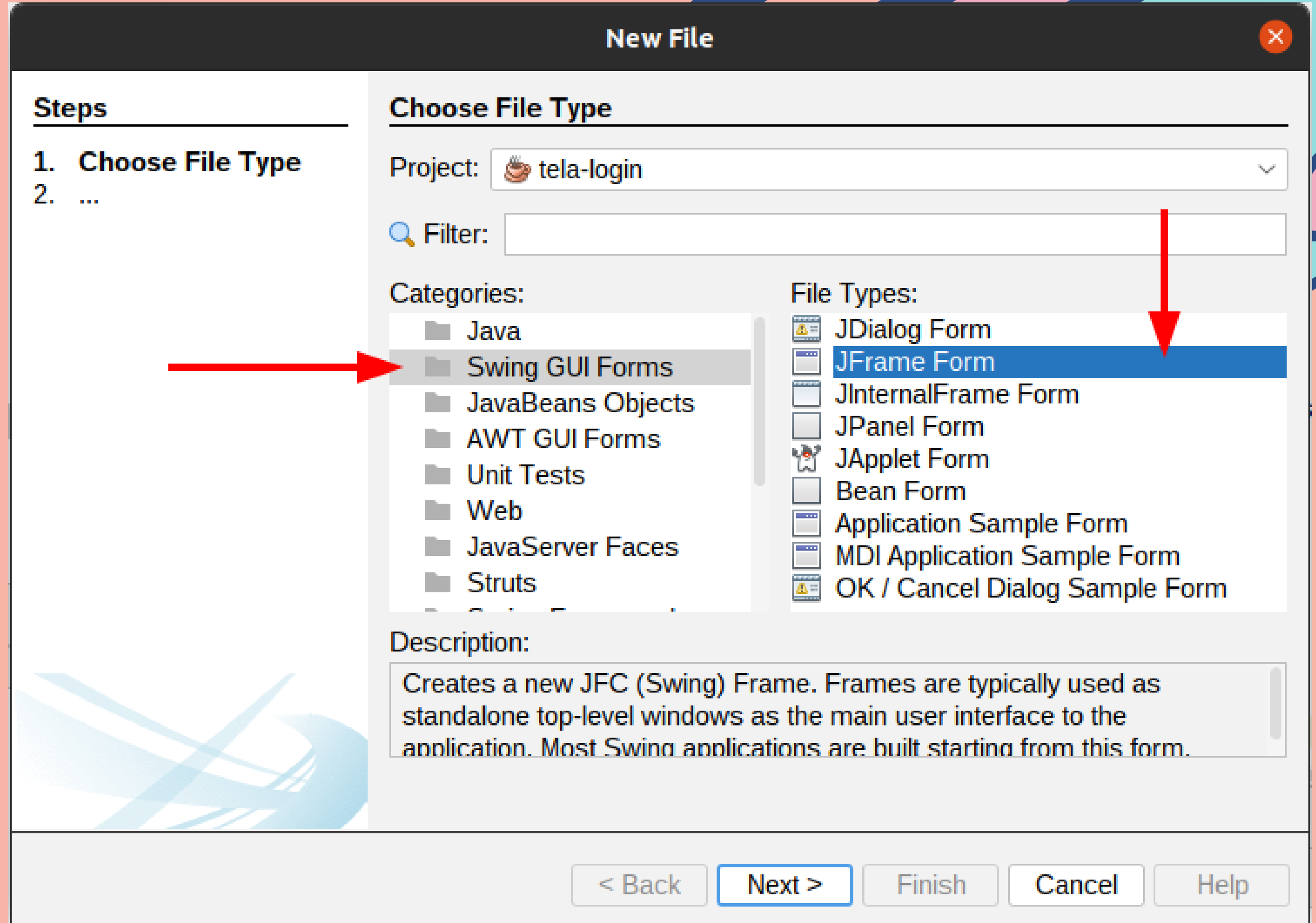
nome do banco de dados que criamos

4a

**Desenvolvendo
nossa interface
gráfica**

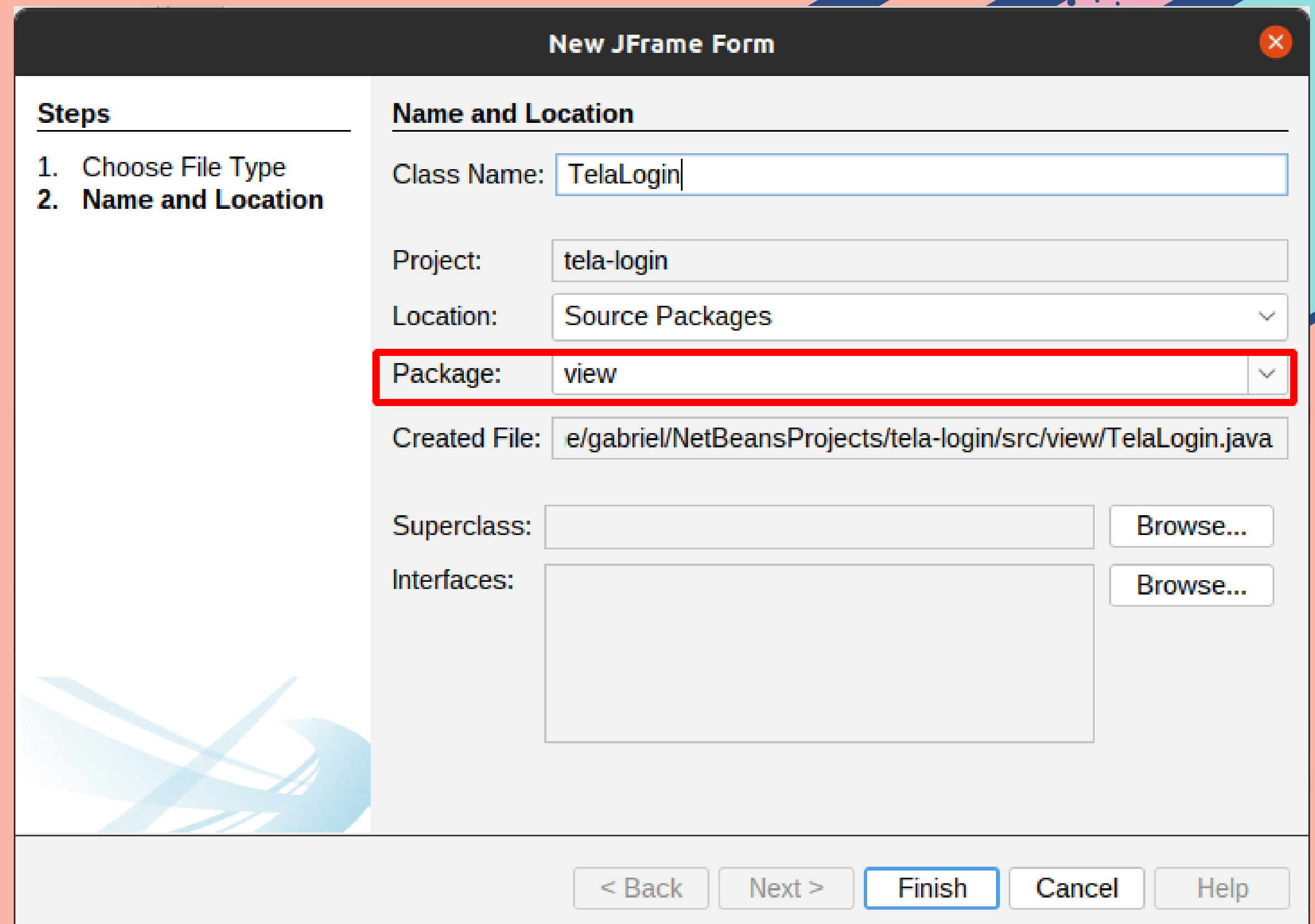


**A) Crie um
novo arquivo
Jframe Form**



**B) Dê ao Form
o nome de
"TelaLogin".**

**Certifique-se
de que ele vai
estar no
pacote "view".**

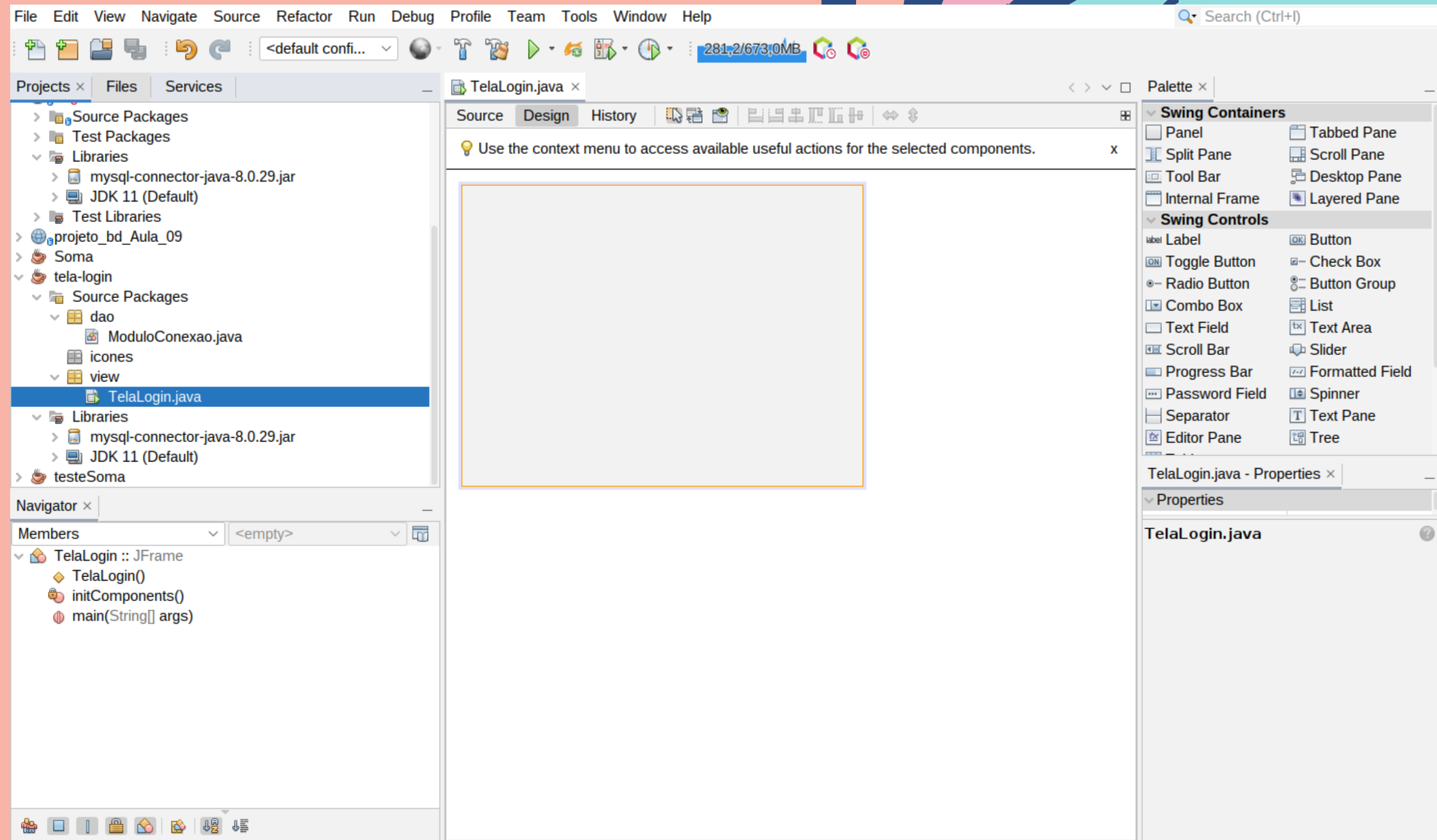


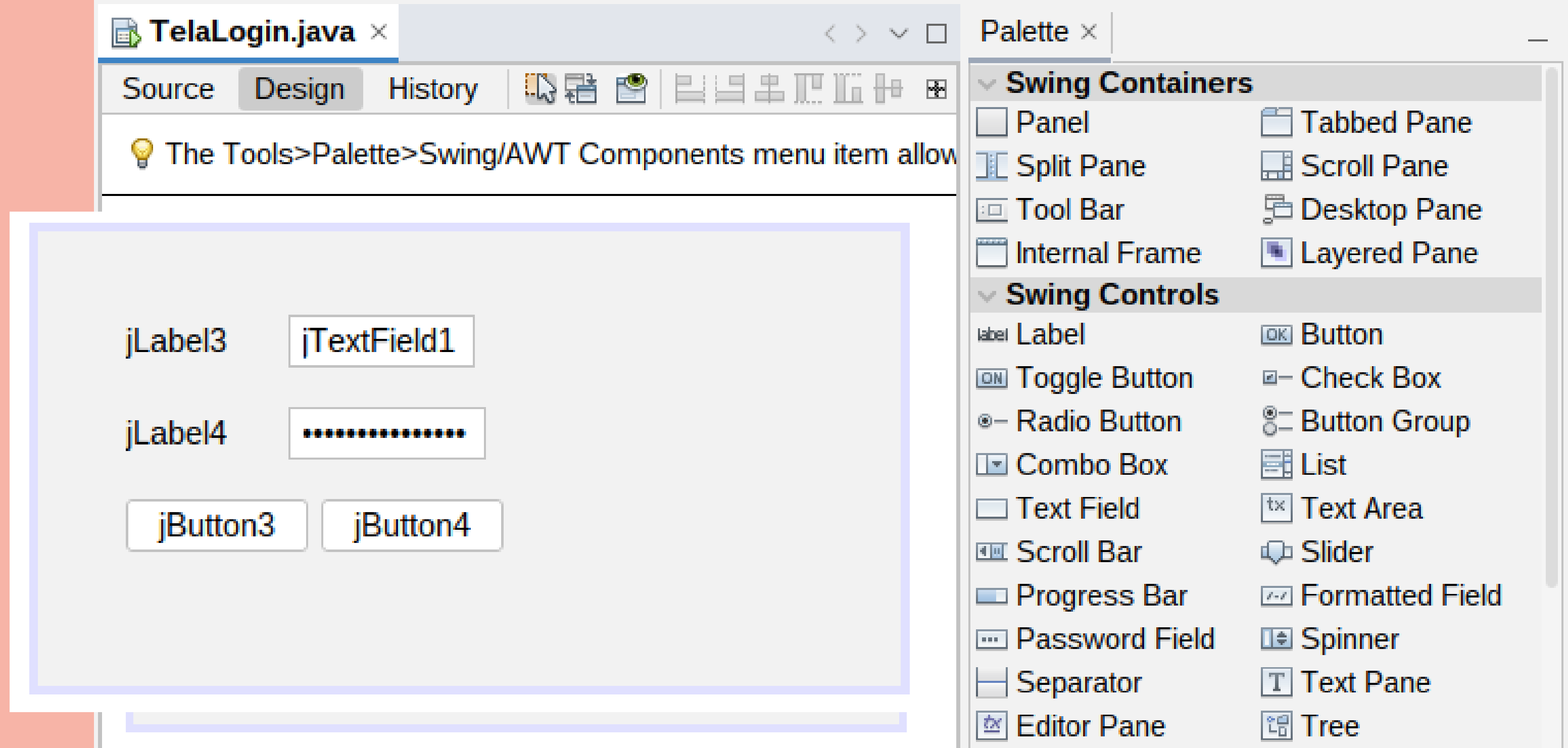
The screenshot shows the 'New JFrame Form' dialog box in the NetBeans IDE. The dialog is divided into two main sections: 'Steps' and 'Name and Location'. The 'Steps' section on the left lists two steps: '1. Choose File Type' and '2. Name and Location'. The 'Name and Location' section on the right contains several input fields. The 'Class Name' field is filled with 'TelaLogin'. The 'Project' field is filled with 'tela-login'. The 'Location' field is a dropdown menu showing 'Source Packages'. The 'Package' field is a dropdown menu showing 'view', and this entire row is highlighted with a red rectangular border. Below the 'Package' field, the 'Created File' field shows the path 'e/gabriel/NetBeansProjects/tela-login/src/view/TelaLogin.java'. At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), 'Cancel', and 'Help'.

Steps	Name and Location
1. Choose File Type	Class Name: TelaLogin
2. Name and Location	Project: tela-login
	Location: Source Packages
	Package: view
	Created File: e/gabriel/NetBeansProjects/tela-login/src/view/TelaLogin.java
	Superclass: <input type="text"/> Browse...
	Interfaces: <input type="text"/> Browse...

< Back Next > **Finish** Cancel Help

Vamos nos
deparar com
essa tela.



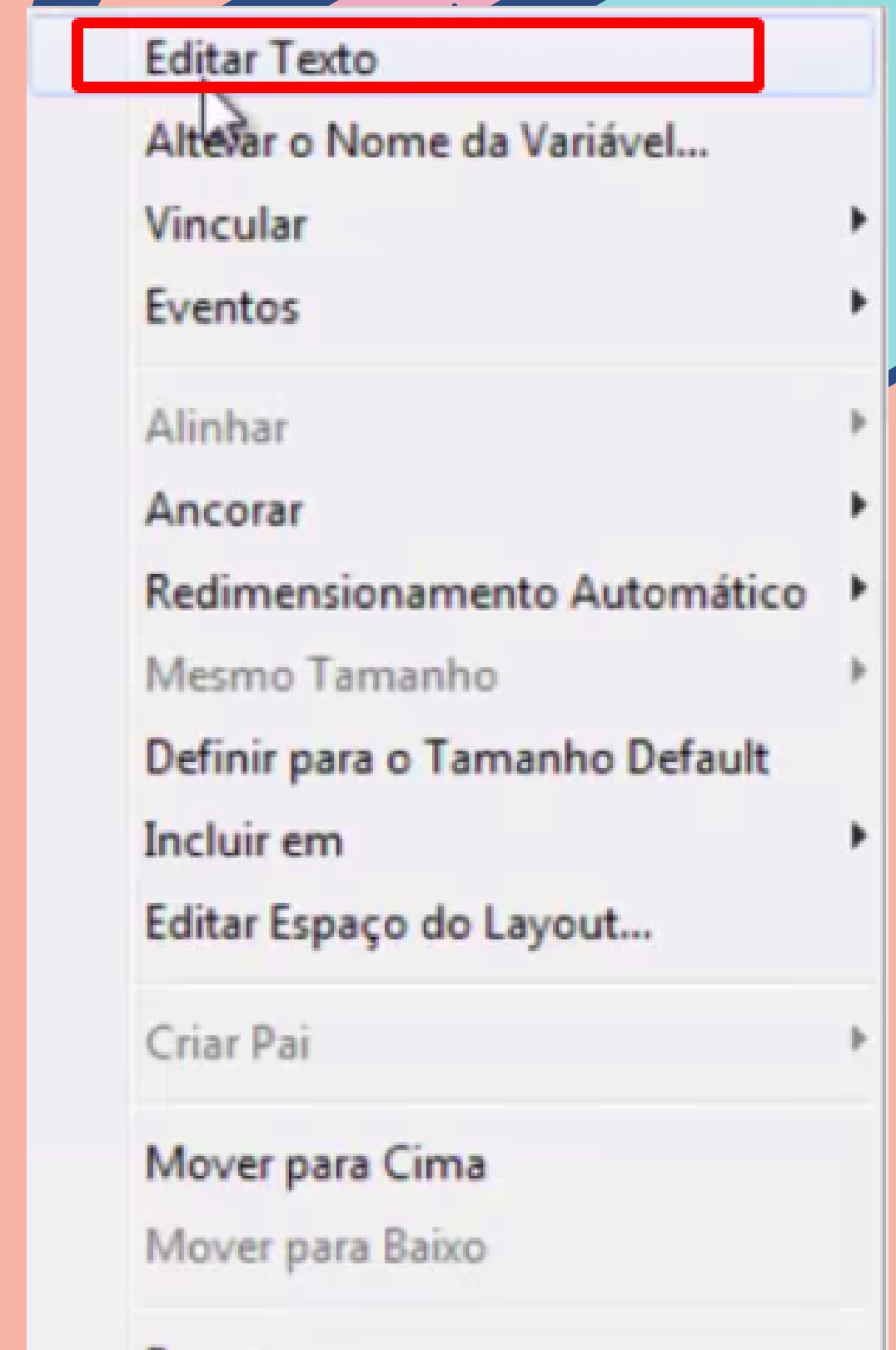
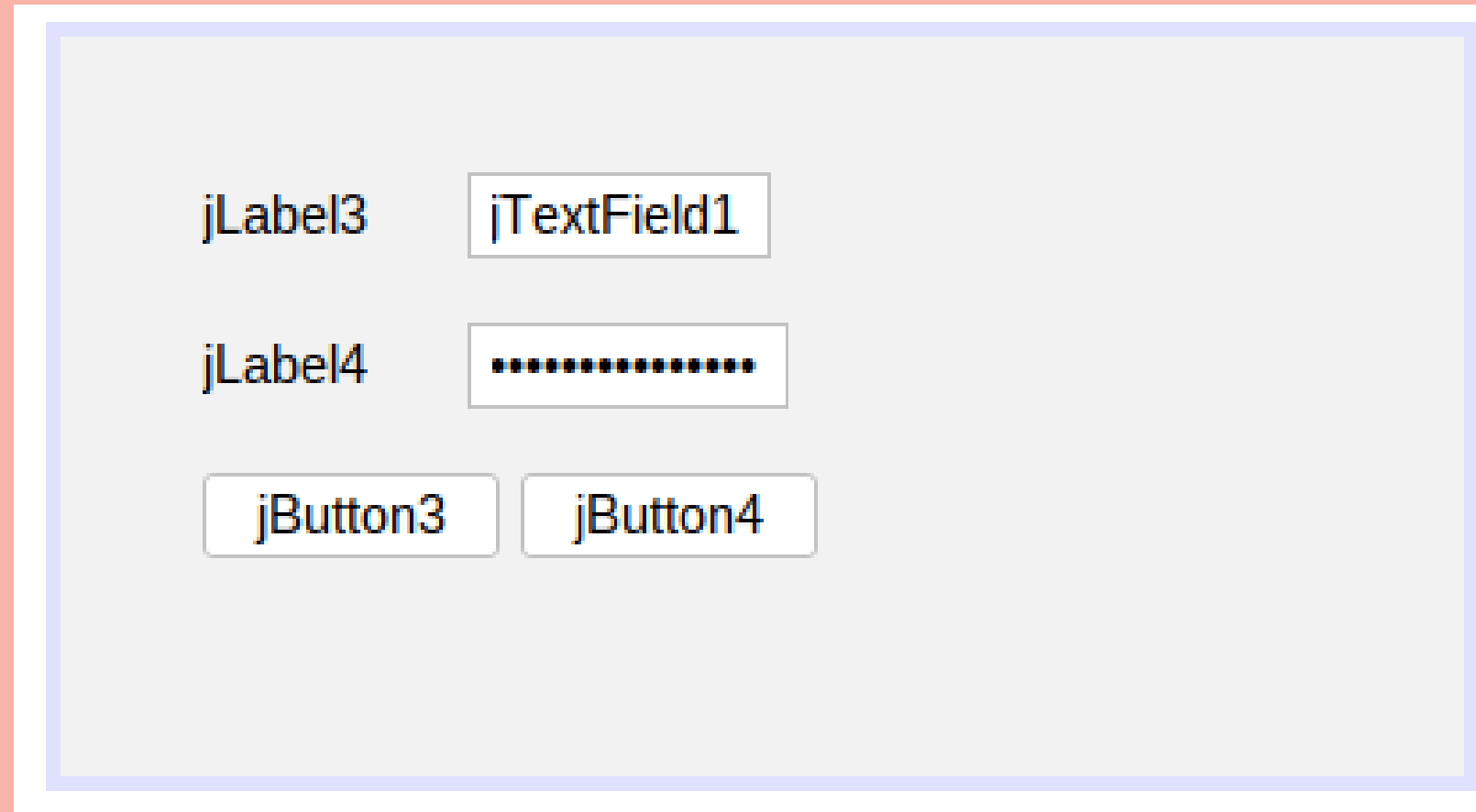


C) Vamos precisar de 2 cada (Label, Text Field, Button) e 1 Password Field.

Basta procurar cada componente na paleta à direita, clicar e arrastar para a nossa tela.

D) Vamos
editar o texto
de cada um dos
elementos

**Clique com o botão
direito no componente
desejado e edite o
texto.**



Depois de editar o texto de todos os componentes esse é o resultado.

Usuário:

Senha:

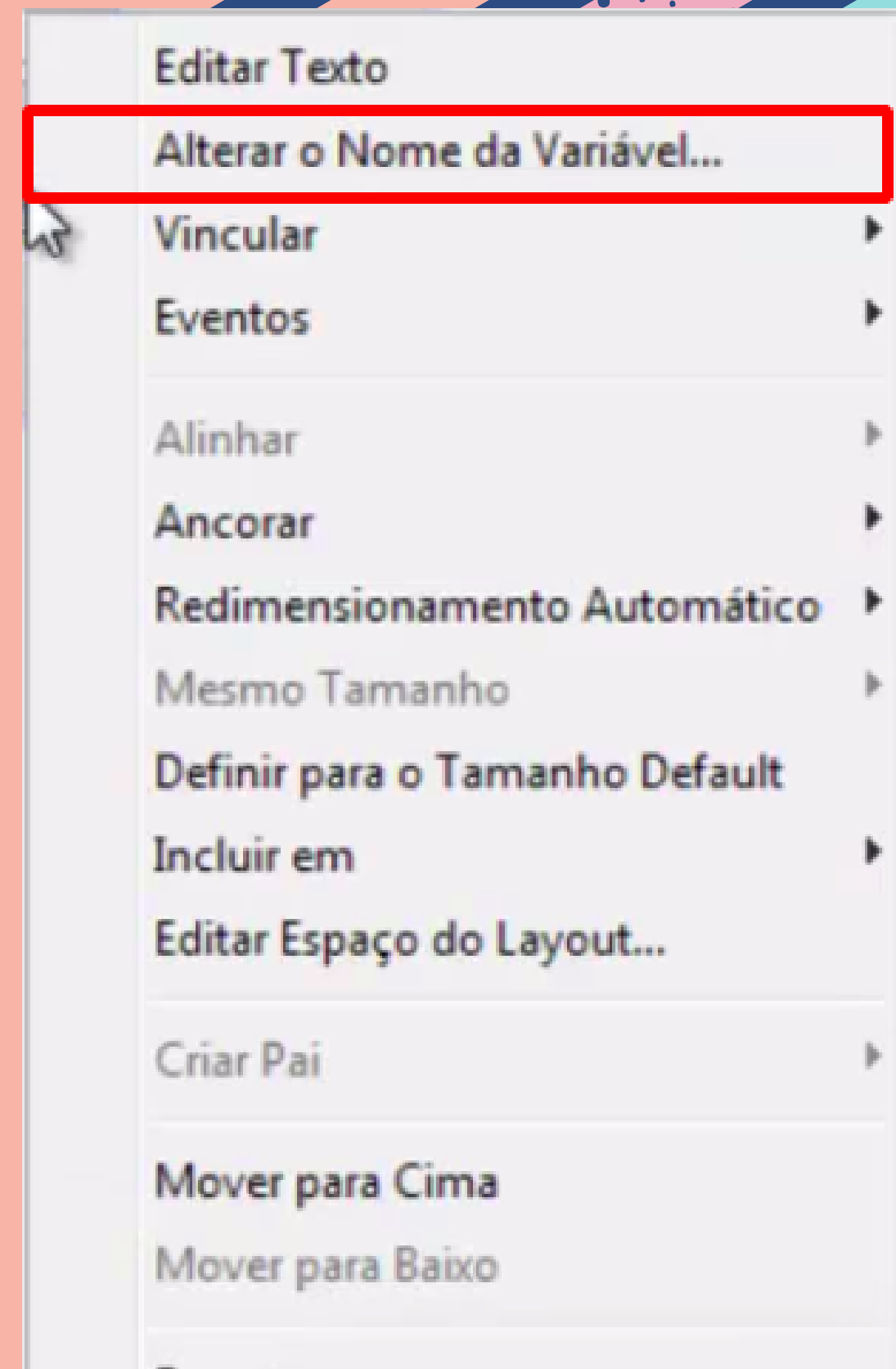
Sair

Entrar

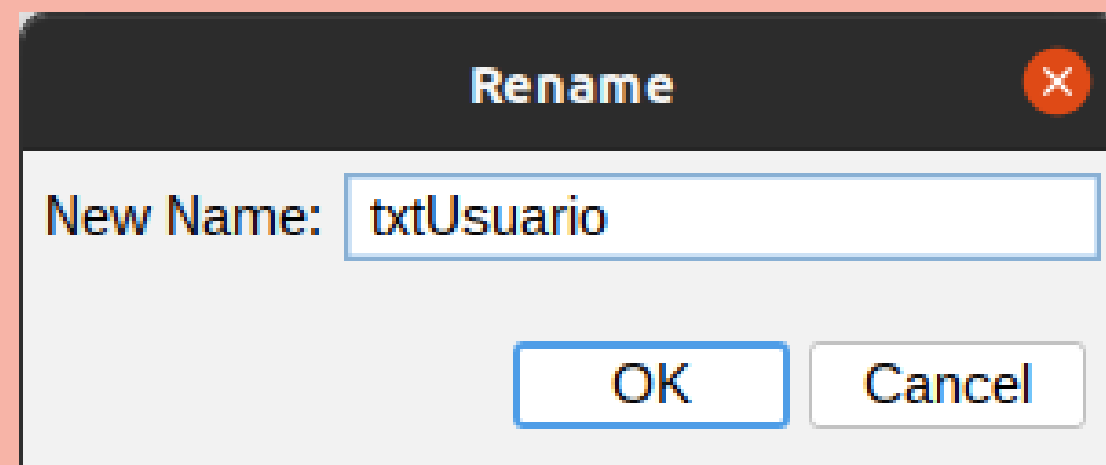
E) Vamos editar o nome da variável dos campos de usuário e senha. Clique com o botão direito no campo e escolha "Altere o Nome da Variável".



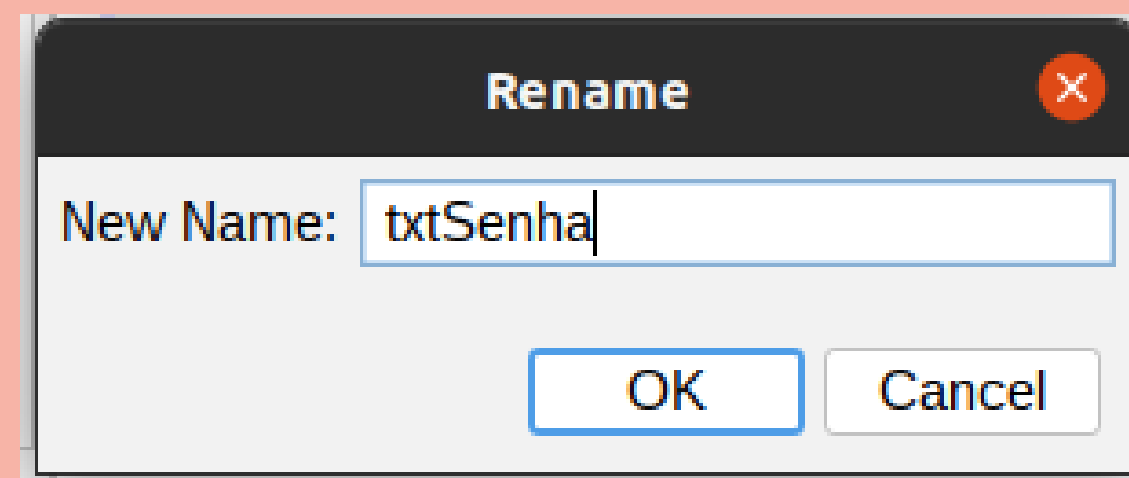
A login form with a light blue border. It contains two input fields: "Usuário:" and "Senha:". Both input fields are highlighted with a red rectangular border. Below the input fields are two buttons: "Sair" and "Entrar".



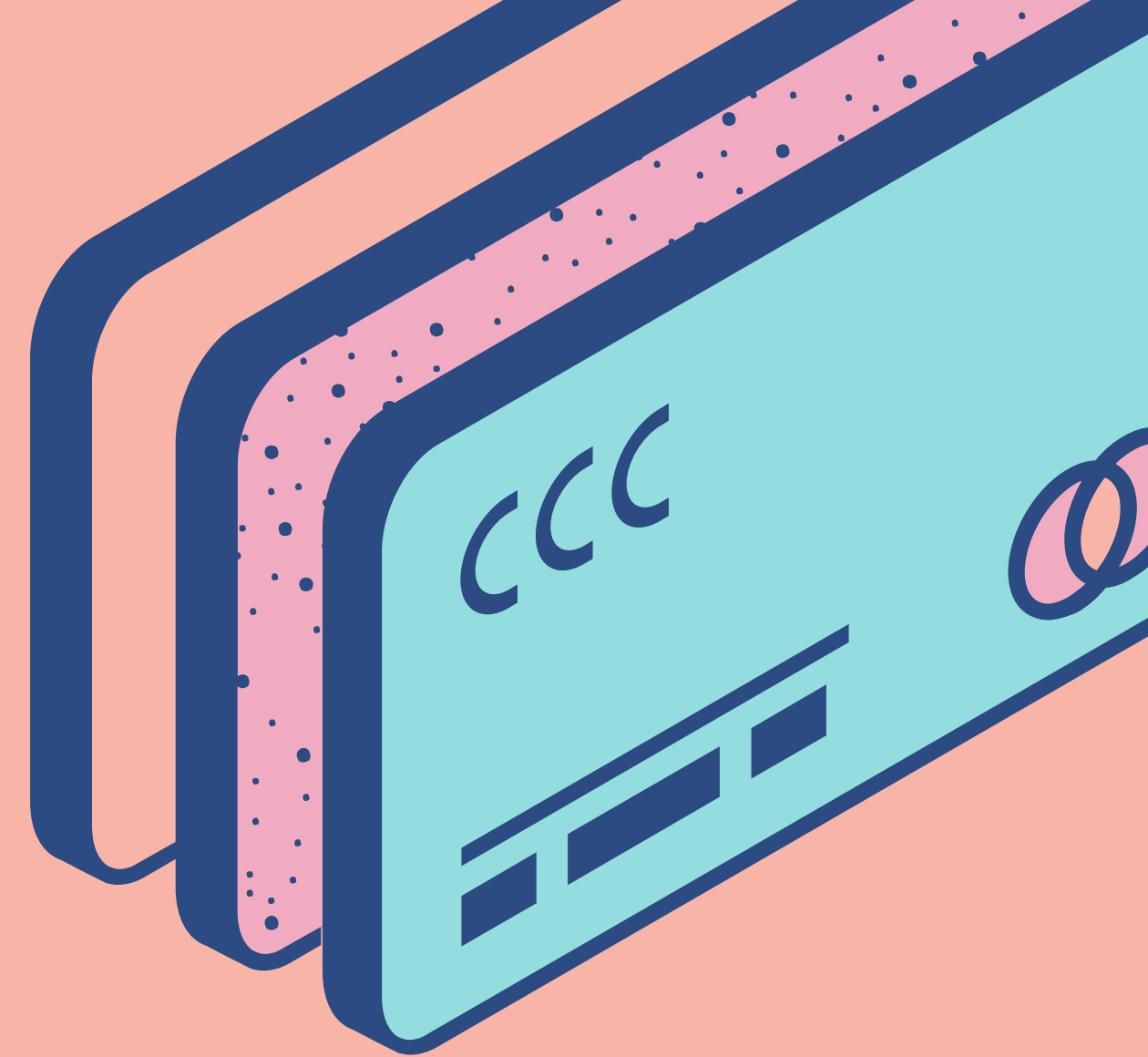
Dê ao campo de usuário o nome de variável `txtUsuario`. E ao campo de senha `txtSenha`.



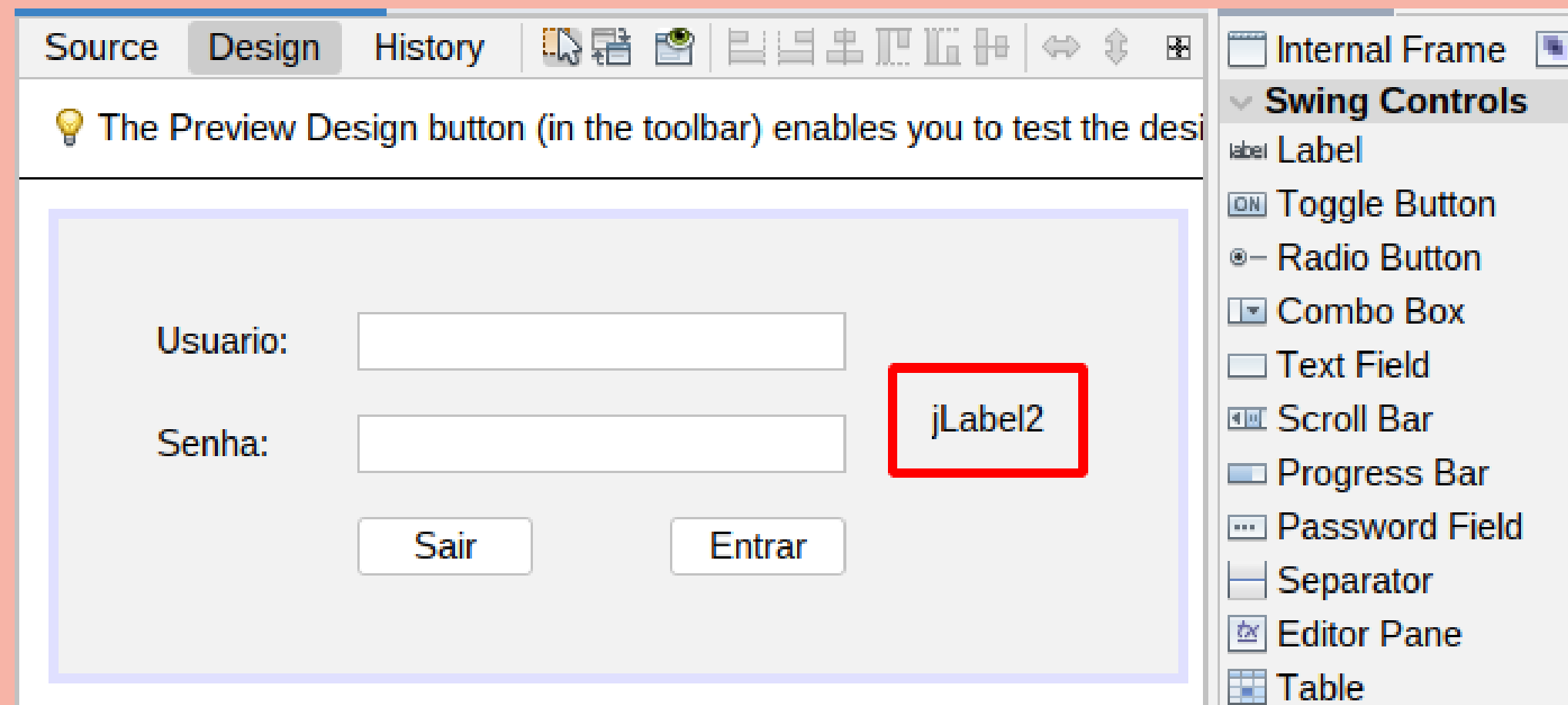
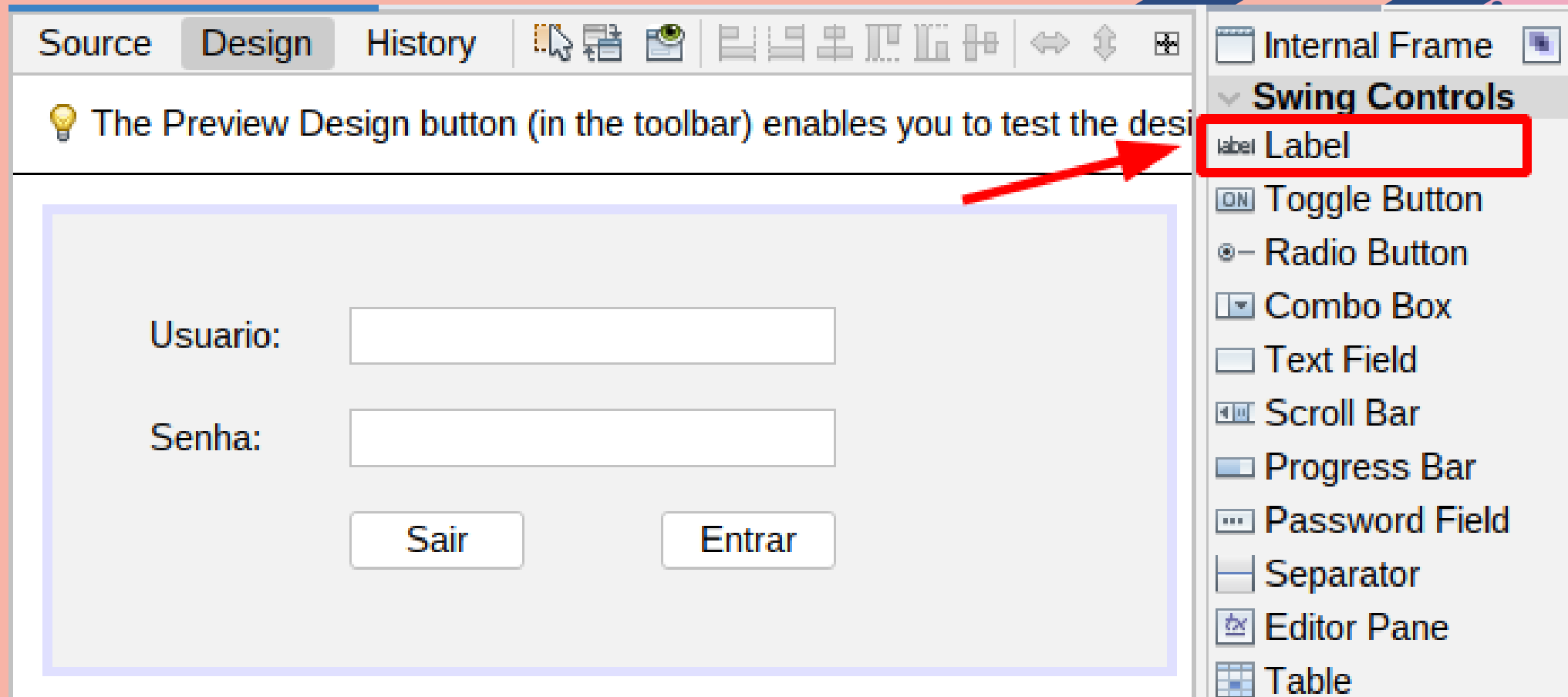
A screenshot of a 'Rename' dialog box. The title bar is dark gray with the word 'Rename' in white and a red close button. The main area has a light gray background. It contains a text input field with the label 'New Name:' and the text 'txtUsuario'. Below the input field are two buttons: 'OK' and 'Cancel'.



A screenshot of a 'Rename' dialog box. The title bar is dark gray with the word 'Rename' in white and a red close button. The main area has a light gray background. It contains a text input field with the label 'New Name:' and the text 'txtSenha'. Below the input field are two buttons: 'OK' and 'Cancel'.



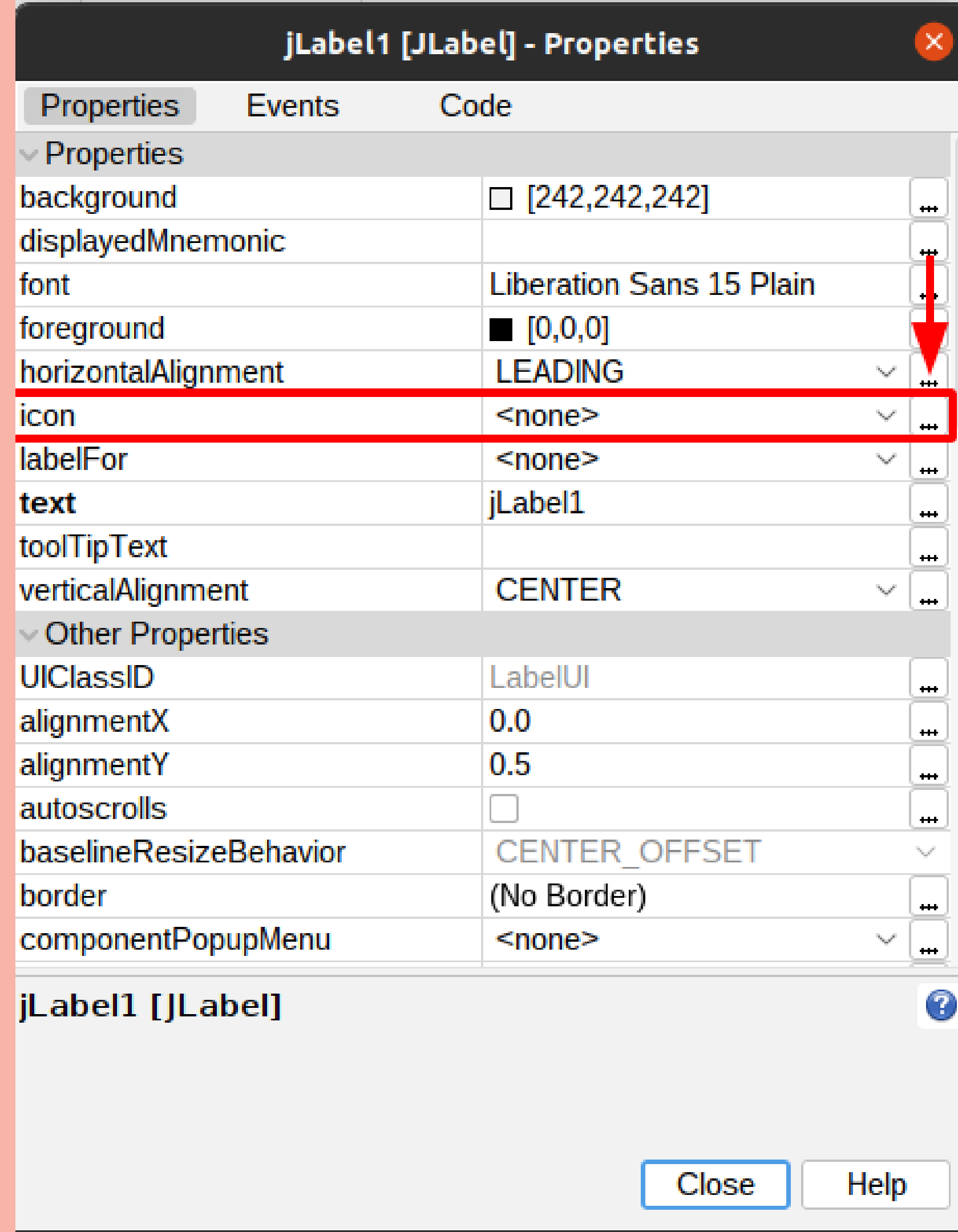
F) Adicione mais uma label para o projeto





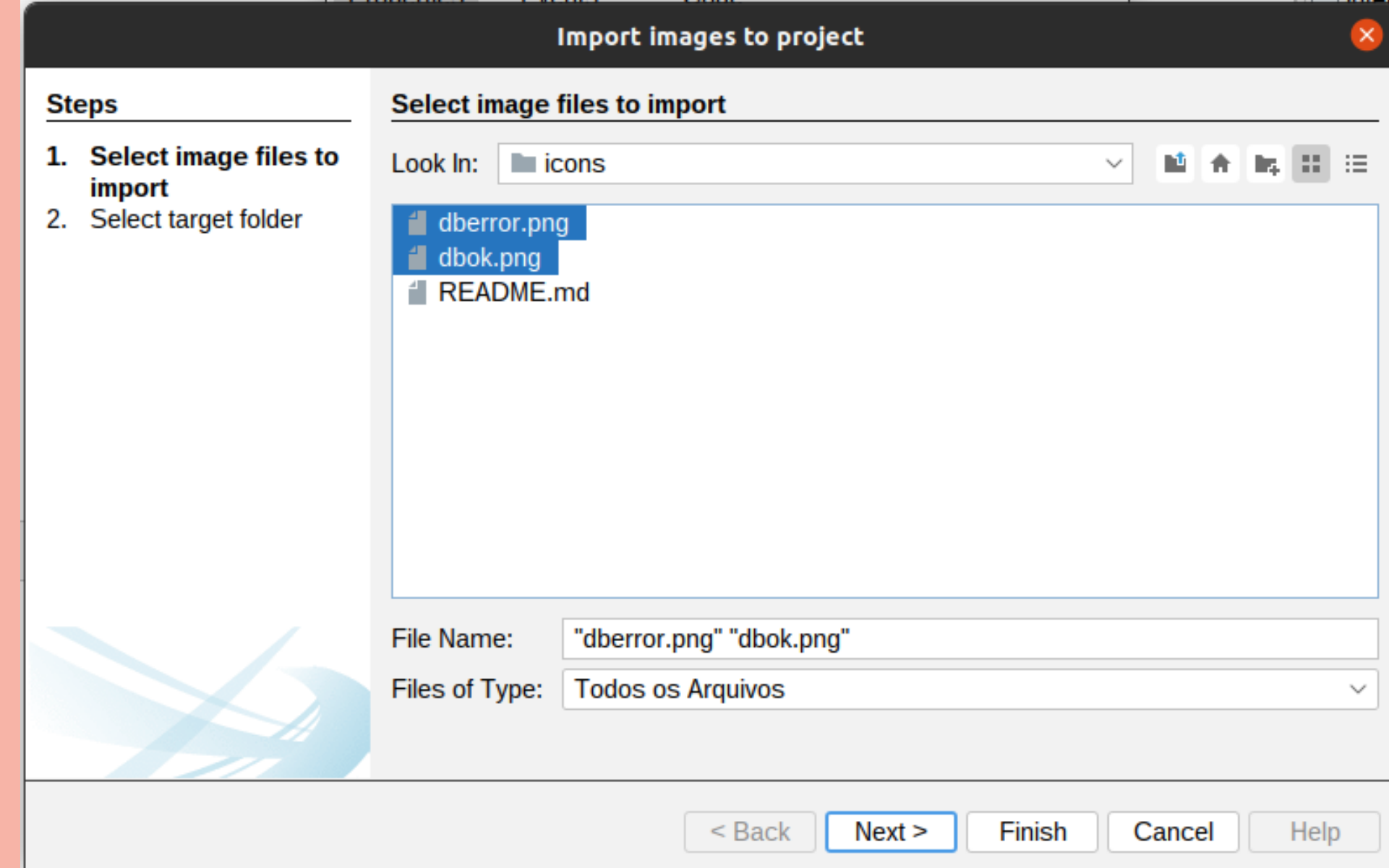
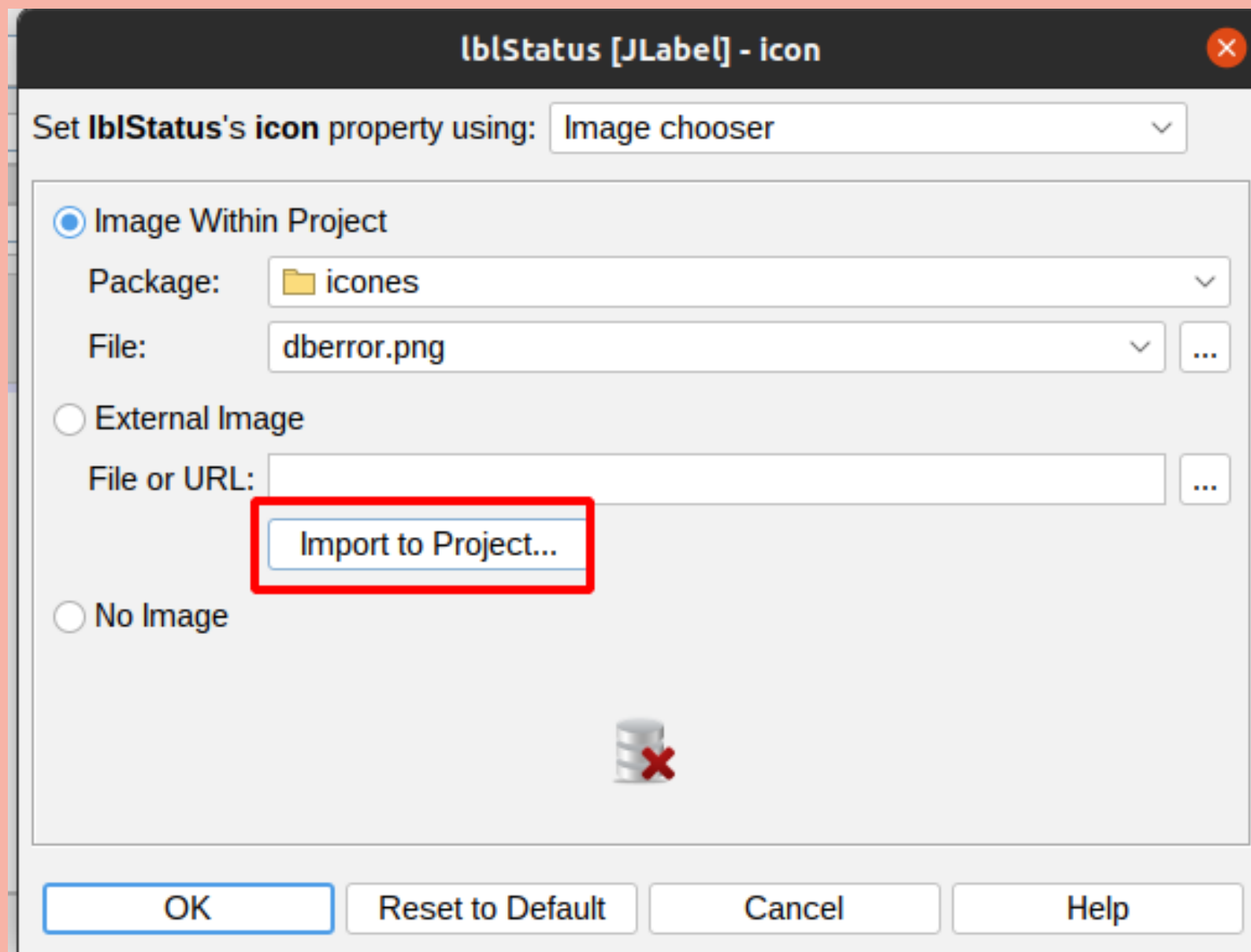
G) Clique com o botão direito do mouse sobre a label e vá em "Propriedades."

H) Vá nos três pontinhos em "Icon".

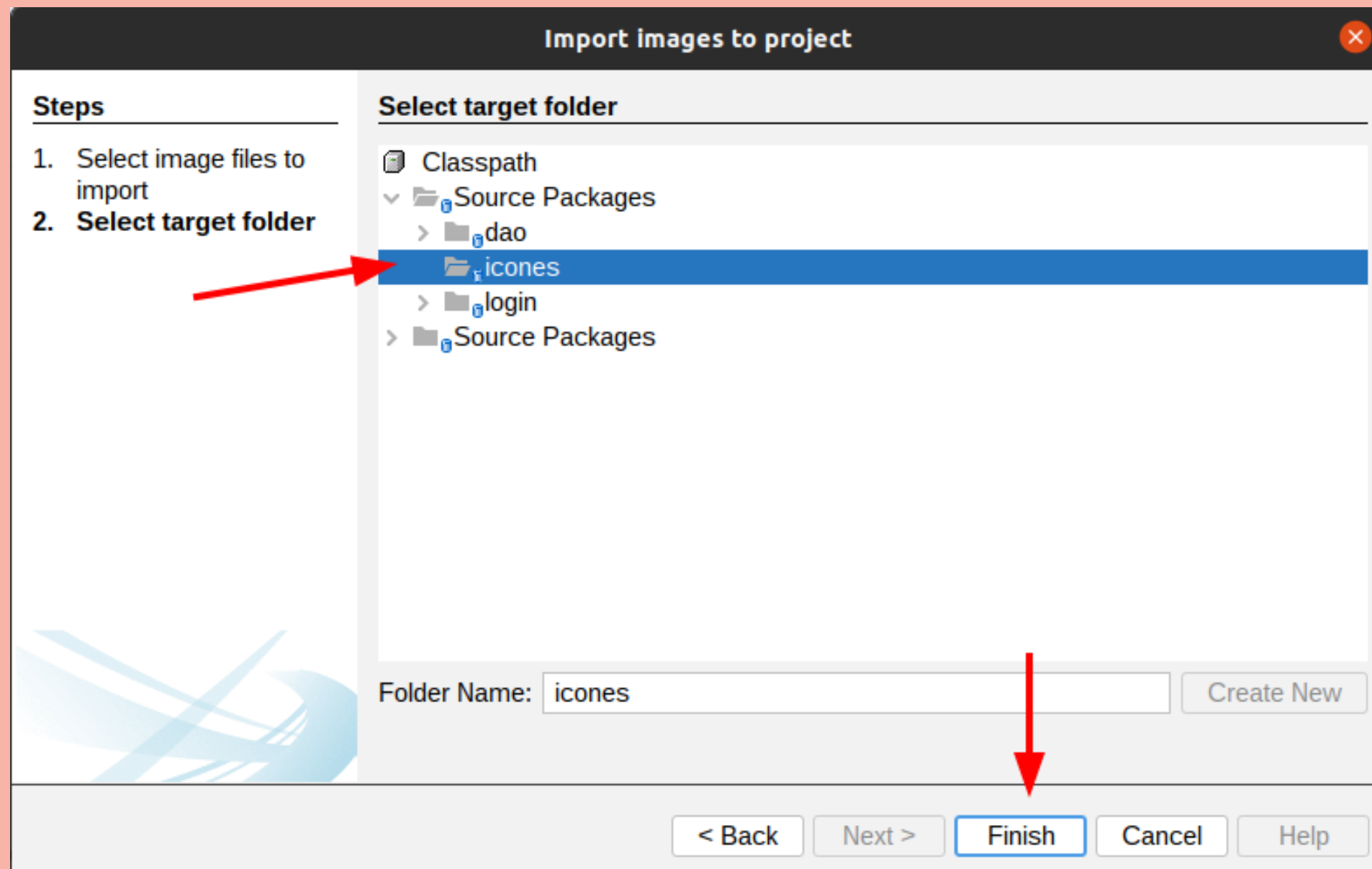


I) Lembra daquele segundo arquivo que baixamos do GitHub na primeira etapa? Precisamos importá-lo. São duas imagens png que serão nosso ícones de status.

Certifique-se de que o "Pacote" ícones foi selecionado. E clique em importar.

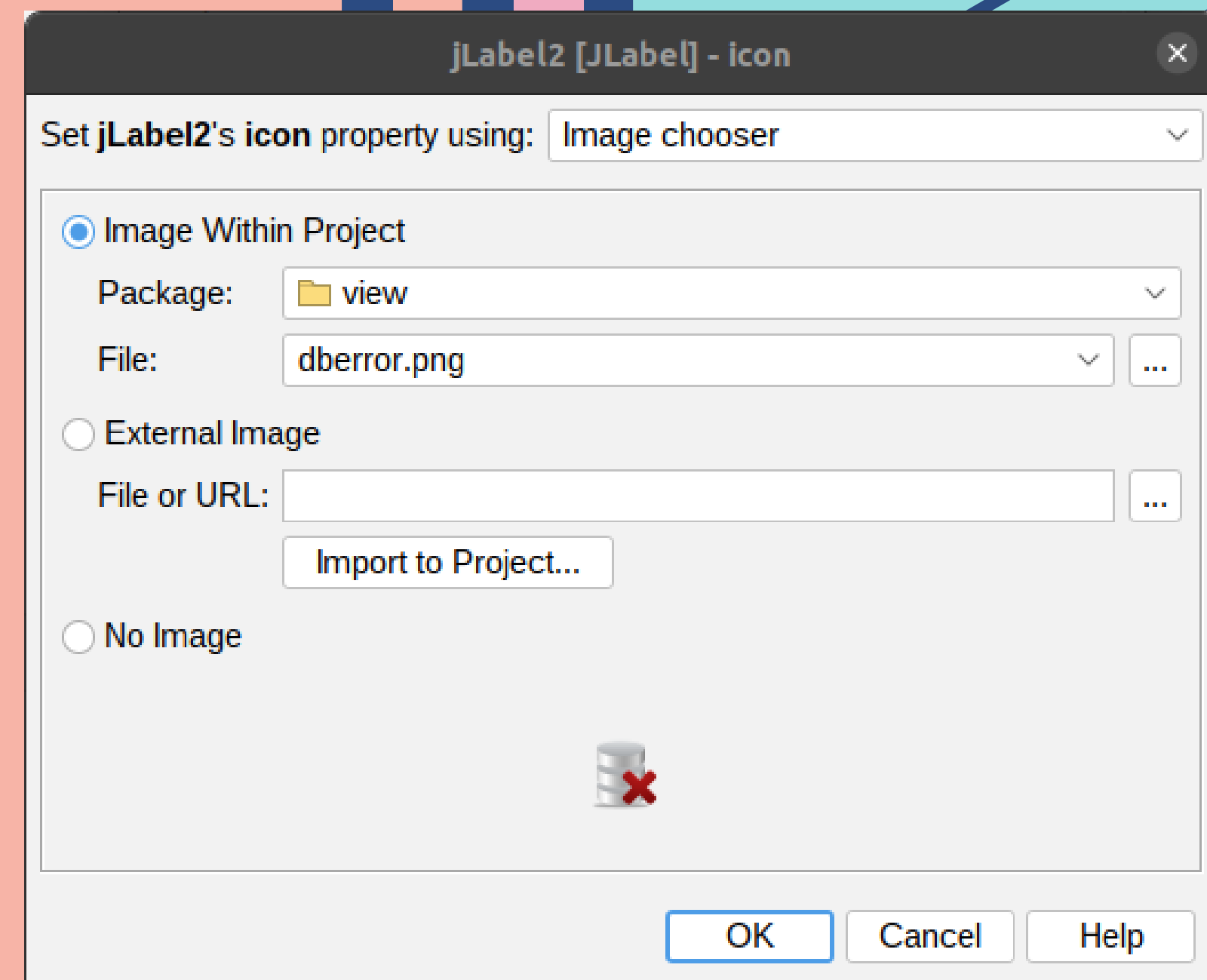


Apoós clicar em importar, busque o local onde foi extraído o arquivo e selecione as duas imagens png. Dê um next.



Clique em finish.

Clique em ok.



Usuario:

Senha:

Sair

Entrar



jLabel2

J) Clique com botão direito no ícone e vá em "Editar Texto". Apague o texto jLabel

Usuario:

Senha:

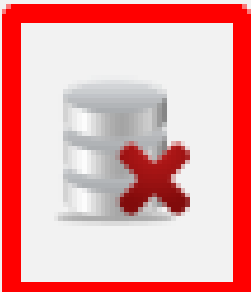
Sair

Entrar




Usuario:

Senha:





K) Clique com o botão direito no ícone e vá em "Editar Nome da Variável", mude para "lblStatus".

Rename 

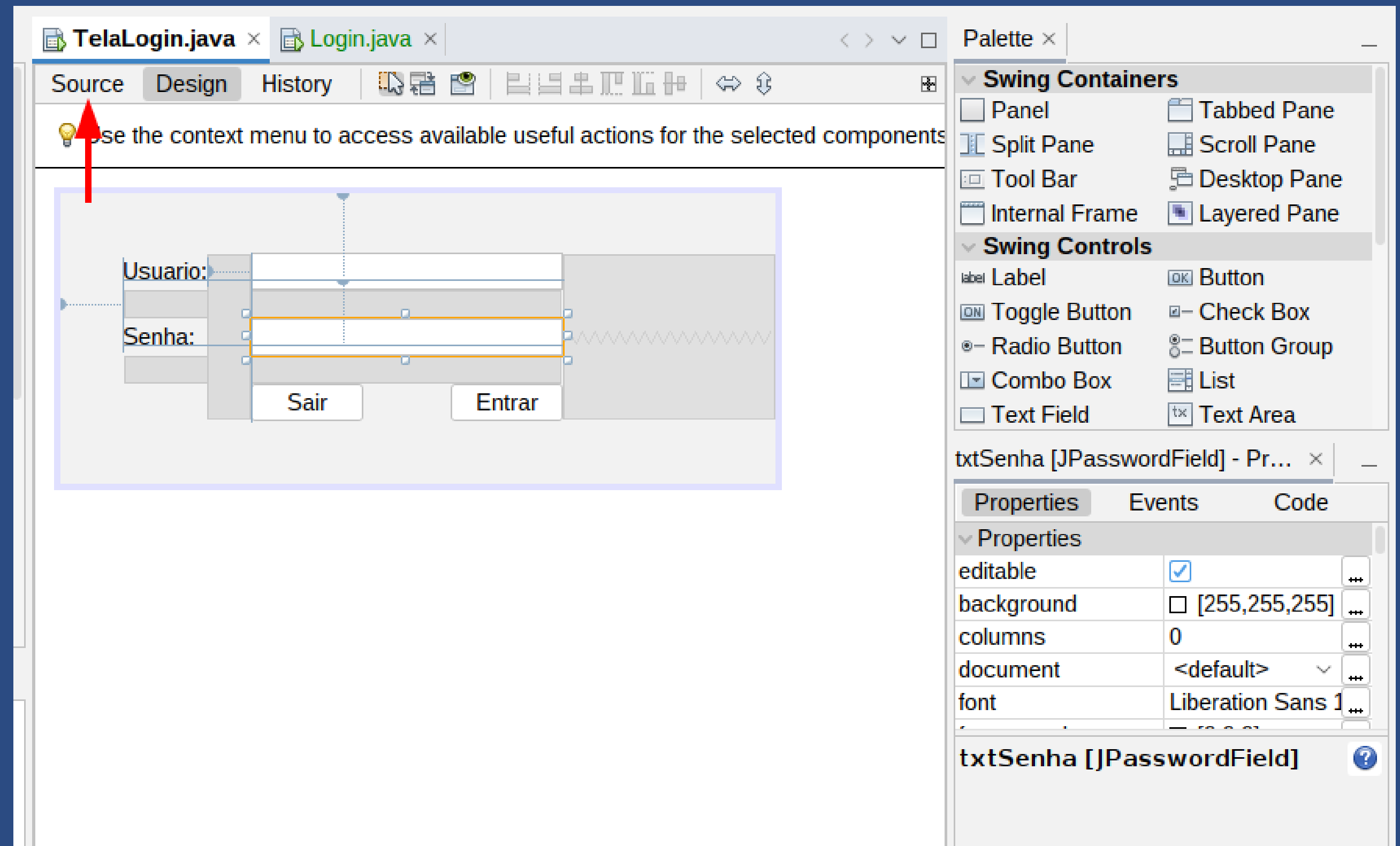
New Name:



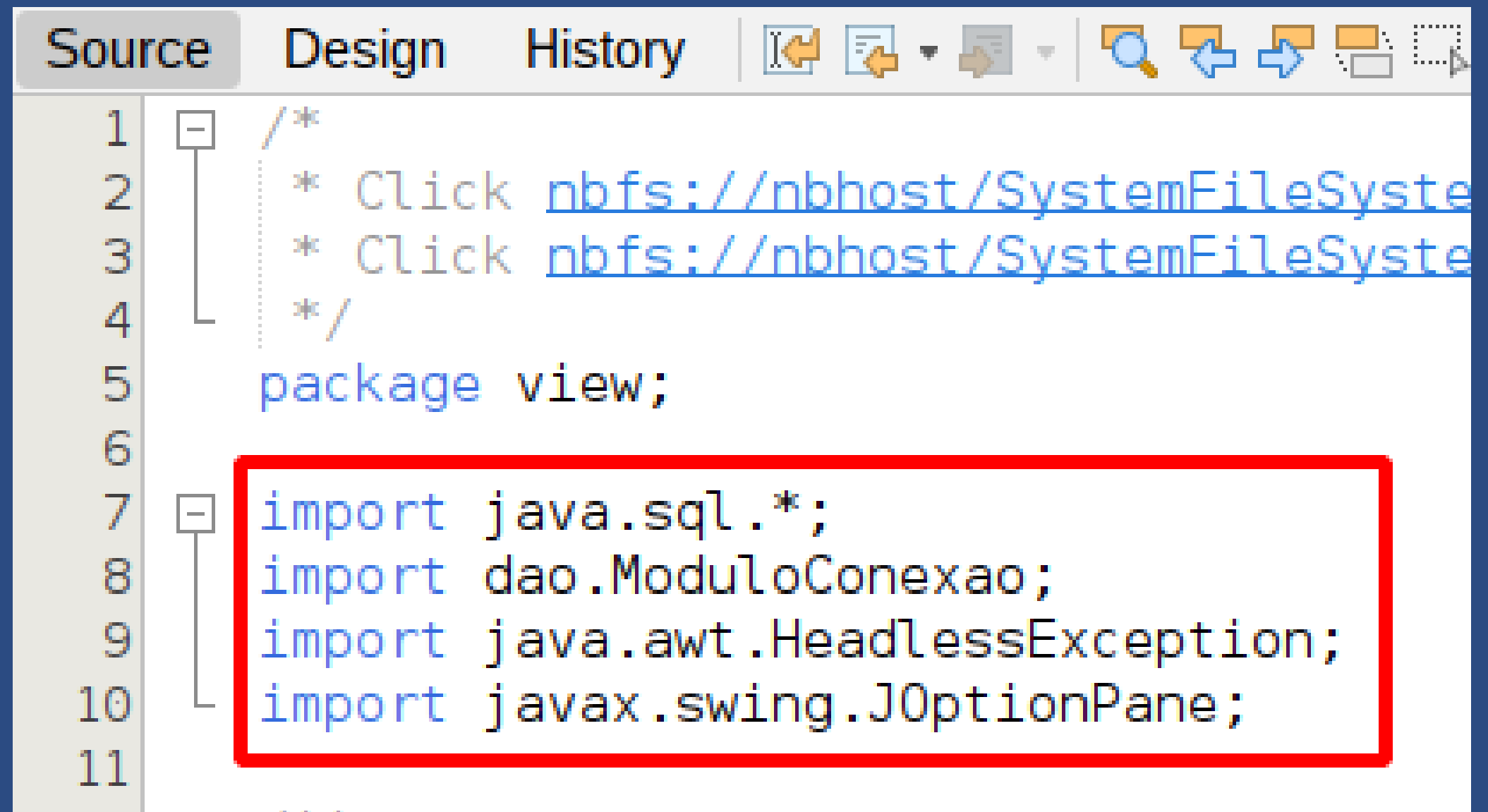
5ª

**Realizando a
validação de login**

A) Vá na aba
"Source" ou
"Código".



B) Vai
aparecer o
código fonte do
formulário que
criamos. Depois
de "package
view",
digitamos as
importações ao
lado.



The screenshot shows an IDE window with tabs for 'Source', 'Design', and 'History'. The 'Source' tab is active, displaying Java code. The code includes a multi-line comment on lines 1-4, followed by 'package view;' on line 5. Lines 7-10 contain four import statements, which are enclosed in a red rectangular box. The code is as follows:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem
3  * Click nbfs://nbhost/SystemFileSystem
4  */
5  package view;
6
7  import java.sql.*;
8  import dao.ModuloConexao;
9  import java.awt.HeadlessException;
10 import javax.swing.JOptionPane;
11
```

C) Adicione os seguintes atributos na classe "TelaLogin"

```
14 public class TelaLogin extends javax.swing.JFrame {
15     Connection conexao;
16     PreparedStatement pst;
17     ResultSet rs;
18
19     public TelaLogin() {
20         initComponents();
21     }
22
23     /**
24      * This method is called from within the constructor to initialize the form.
25      * WARNING: Do NOT modify this code. The content of this method is always
26      * regenerated by the Form Editor.
27      */
28     @SuppressWarnings("unchecked")
```

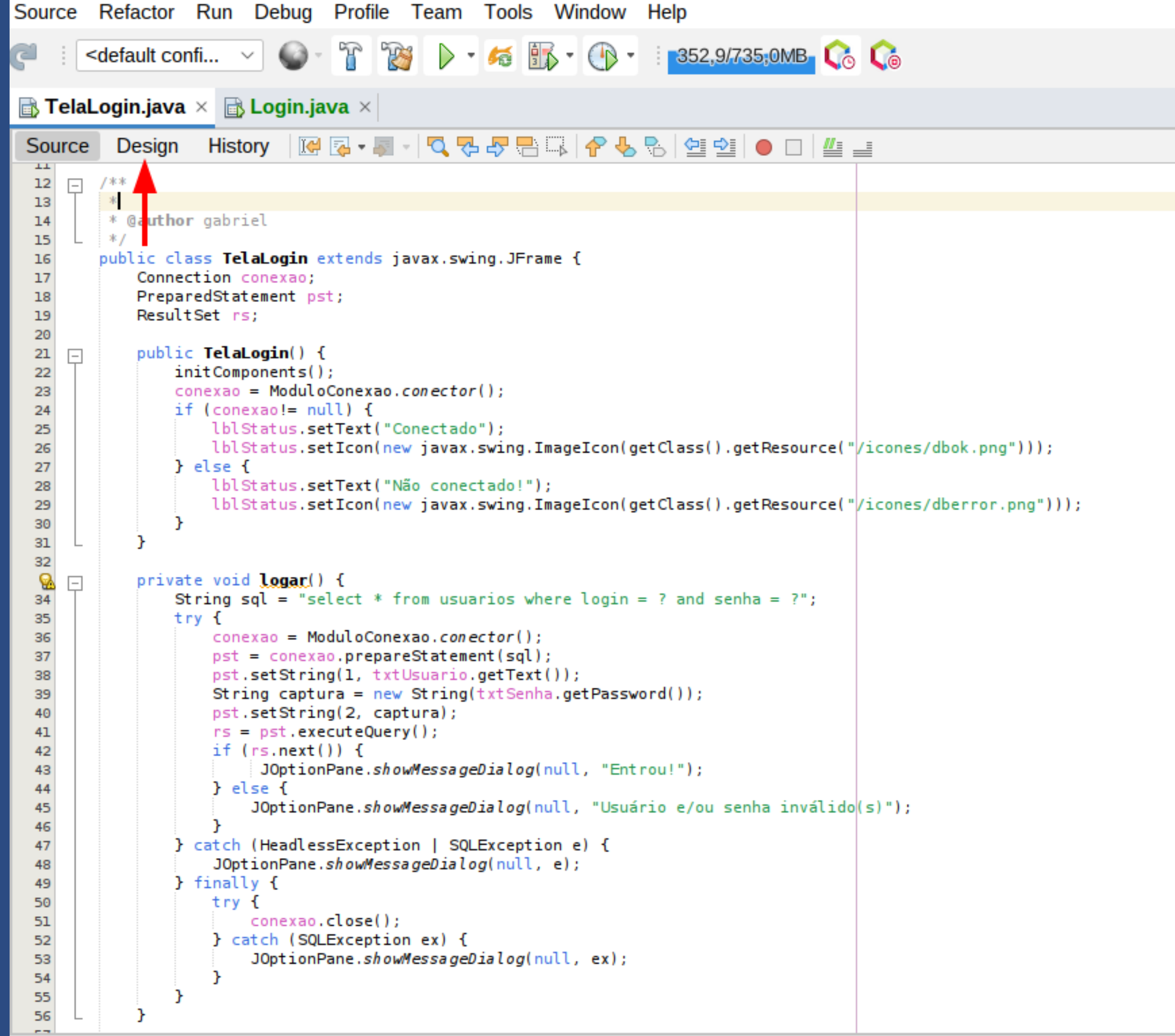
D) Dentro do construtor TelaLogin adicionamos o código que verifica o status da conexão.

```
19 public TelaLogin() {
20     initComponents();
21     conexao = ModuloConexao.conector();
22     if (conexao != null) {
23         lblStatus.setText("Conectado");
24         lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icones/dbok.png")));
25     } else {
26         lblStatus.setText("Não conectado!");
27         lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icones/dberror.png")));
28     }
29 }
```


E) Abaixo do construtor digite este método. Ela faz a autenticação do login do usuário, verificando se tem no banco de dados."

```
private void logar() {  
    String sql = "select * from usuarios where login = ? and senha = ?";  
    try {  
        conexao = ModuloConexao.conector();  
        pst = conexao.prepareStatement(sql);  
        pst.setString(1, txtUsuario.getText());  
        String captura = new String(txtSenha.getPassword());  
        pst.setString(2, captura);  
        rs = pst.executeQuery();  
        if (rs.next()) {  
            JOptionPane.showMessageDialog(null, "Entrou!");  
        } else {  
            JOptionPane.showMessageDialog(null, "Usuário e/ou senha inválido(s)");  
        }  
    } catch (HeadlessException | SQLException e) {  
        JOptionPane.showMessageDialog(null, e);  
    } finally {  
        try {  
            conexao.close();  
        } catch (SQLException ex) {  
            JOptionPane.showMessageDialog(null, ex);  
        }  
    }  
}
```

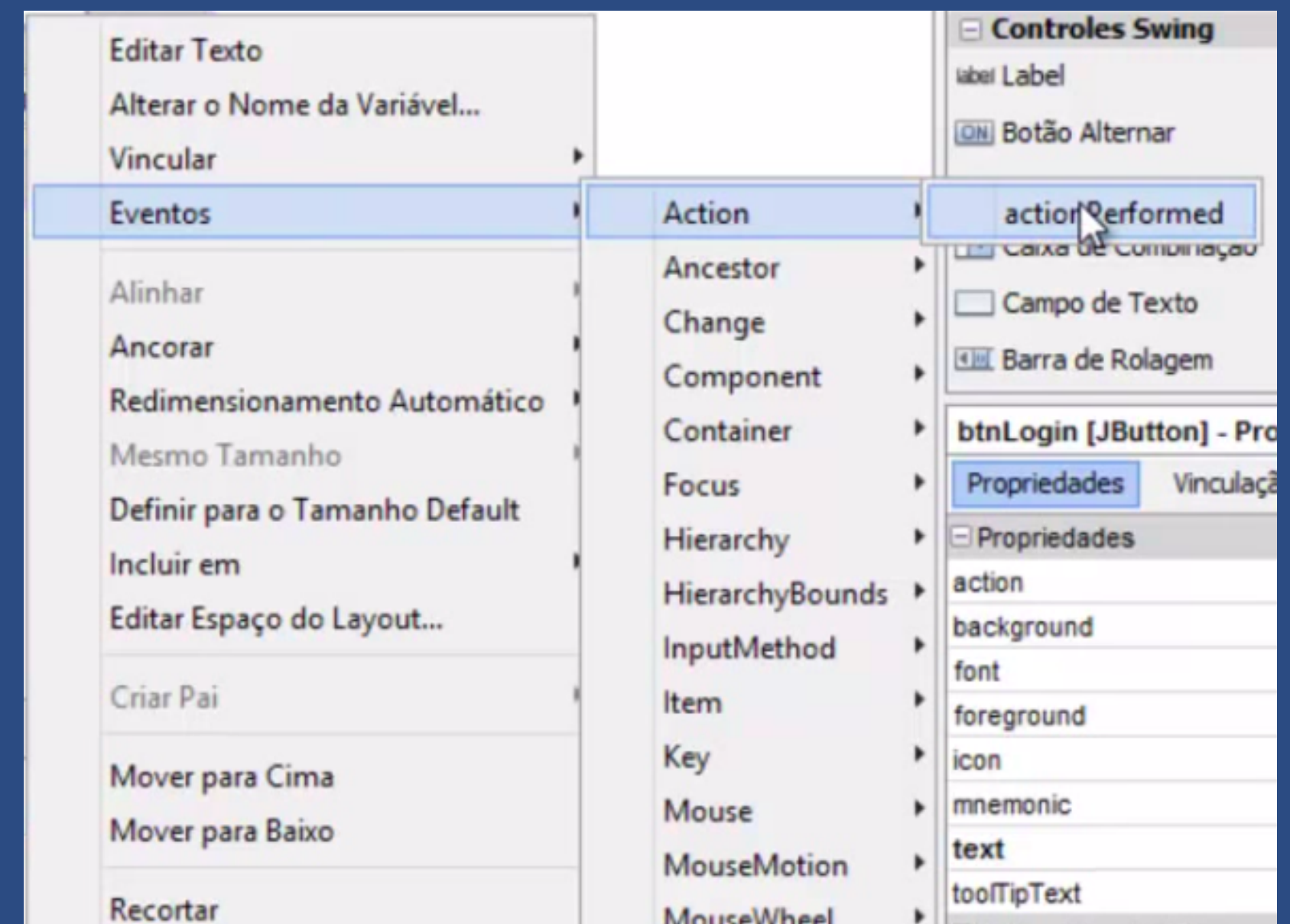
F) Vamos voltar ao nosso Design



```
Source Refactor Run Debug Profile Team Tools Window Help
<default confi... 352,9/735,0MB
TelaLogin.java x Login.java x
Source Design History
11 /**
12  *
13  * @author gabriel
14  */
15
16 public class TelaLogin extends javax.swing.JFrame {
17     Connection conexao;
18     PreparedStatement pst;
19     ResultSet rs;
20
21     public TelaLogin() {
22         initComponents();
23         conexao = ModuloConexao.conector();
24         if (conexao != null) {
25             lblStatus.setText("Conectado");
26             lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icones/dbok.png")));
27         } else {
28             lblStatus.setText("Não conectado!");
29             lblStatus.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icones/dberror.png")));
30         }
31     }
32
33     private void logar() {
34         String sql = "select * from usuarios where login = ? and senha = ?";
35         try {
36             conexao = ModuloConexao.conector();
37             pst = conexao.prepareStatement(sql);
38             pst.setString(1, txtUsuario.getText());
39             String captura = new String(txtSenha.getPassword());
40             pst.setString(2, captura);
41             rs = pst.executeQuery();
42             if (rs.next()) {
43                 JOptionPane.showMessageDialog(null, "Entrou!");
44             } else {
45                 JOptionPane.showMessageDialog(null, "Usuário e/ou senha inválido(s)");
46             }
47         } catch (HeadlessException | SQLException e) {
48             JOptionPane.showMessageDialog(null, e);
49         } finally {
50             try {
51                 conexao.close();
52             } catch (SQLException ex) {
53                 JOptionPane.showMessageDialog(null, ex);
54             }
55         }
56     }
57 }
```



G) Clique com o botão direito em "Entrar" e vá em Eventos -> Ação -> ActionPerformed.

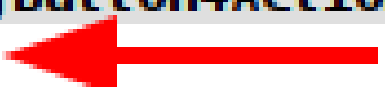


H) Ele retorna automaticamente ao código fonte.

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

No evento criado, fazemos a chamada do método que criamos "logar()";

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    logar();  
}
```

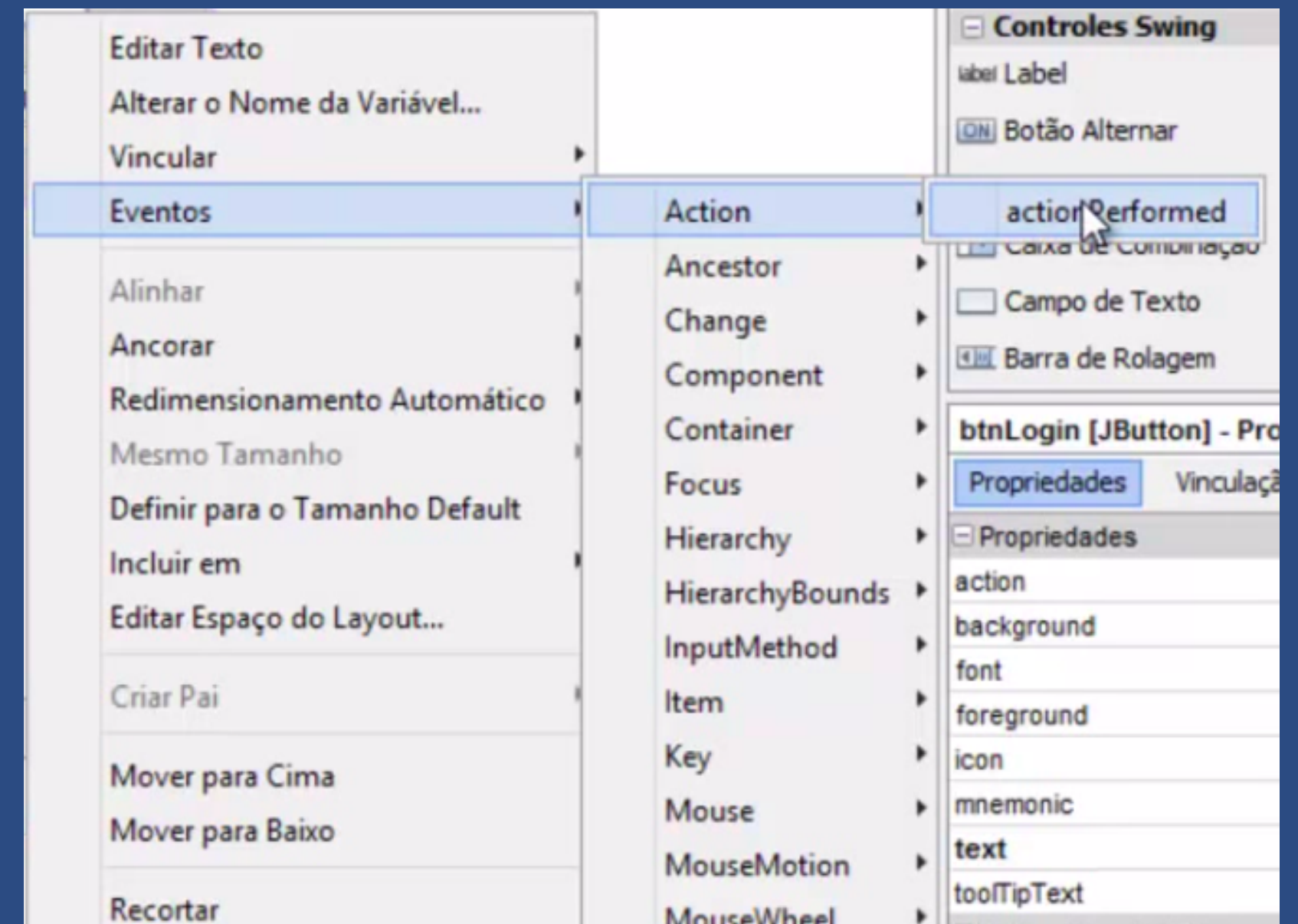


Usuario:

Senha:



I) Aqui nós fazemos a mesma coisa para o botão sair.



J) Ele retorna automaticamente ao código fonte.

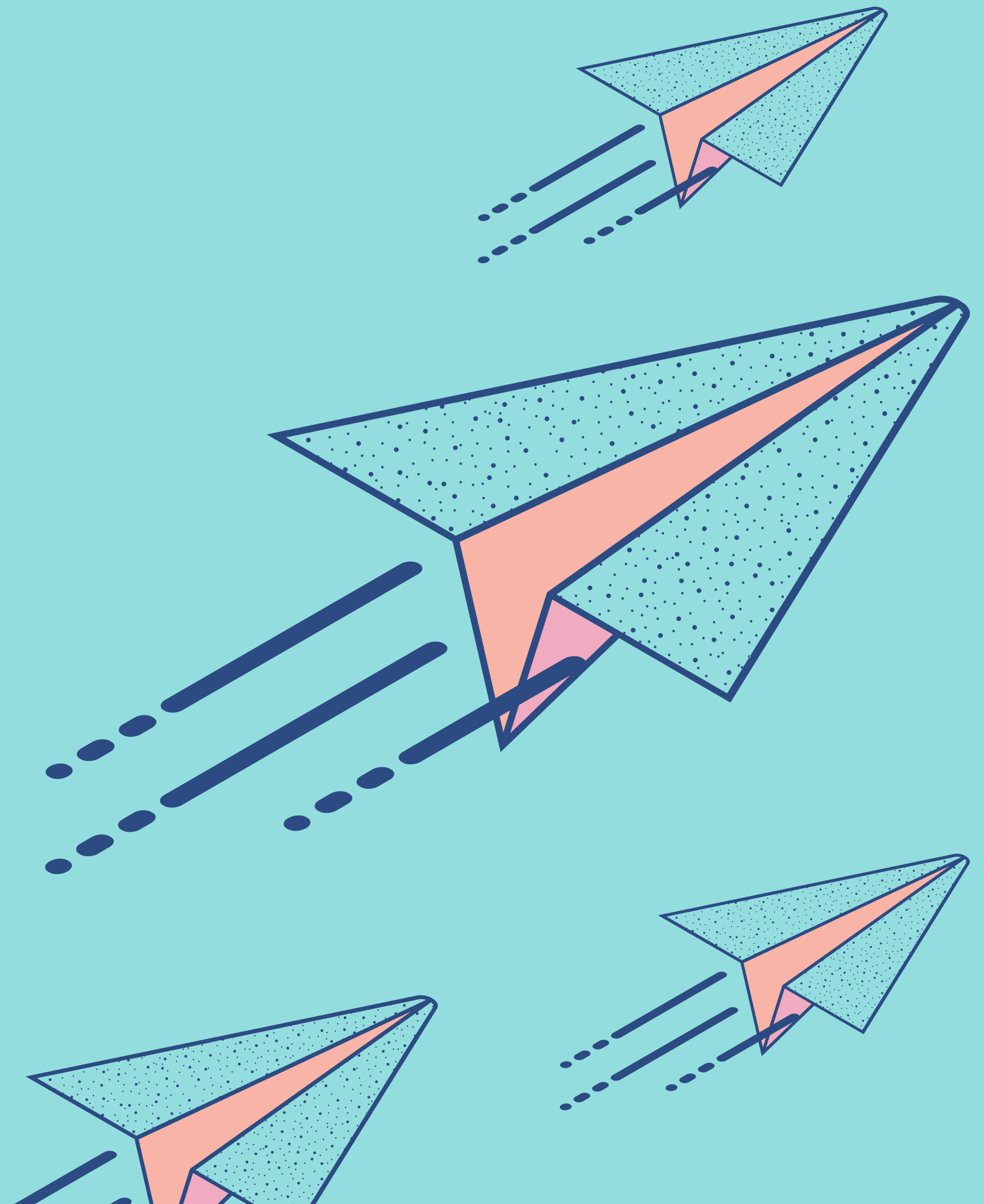
```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

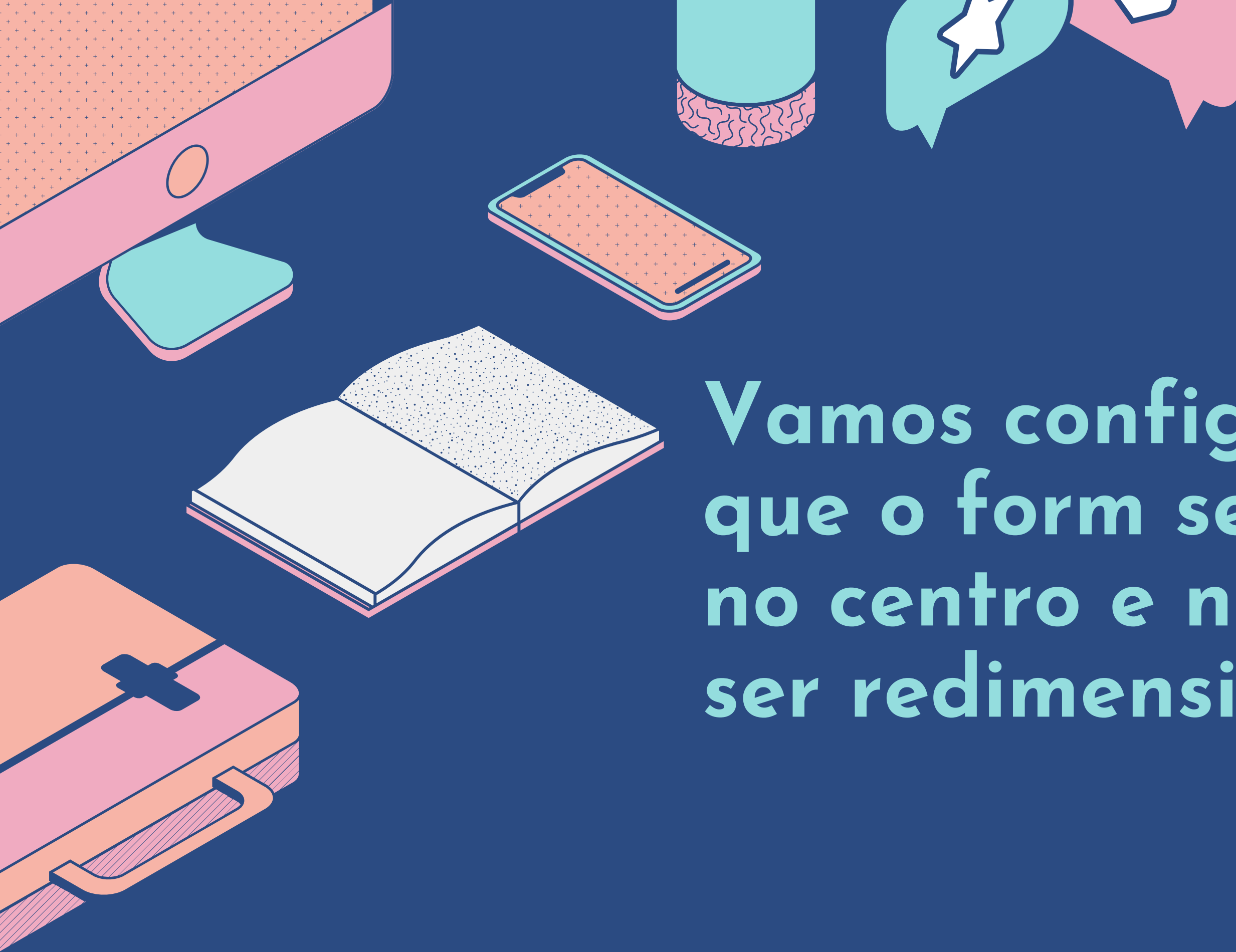
No evento criado, adicionamos o código `System.exit(0);`.

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Quase lá...


Só falta uns pequenos ajustes





Vamos configurar para
que o form seja gerado
no centro e não possa
ser redimensionado.

A) Clique com o botão direito dentro do form e vá em "Propriedades".



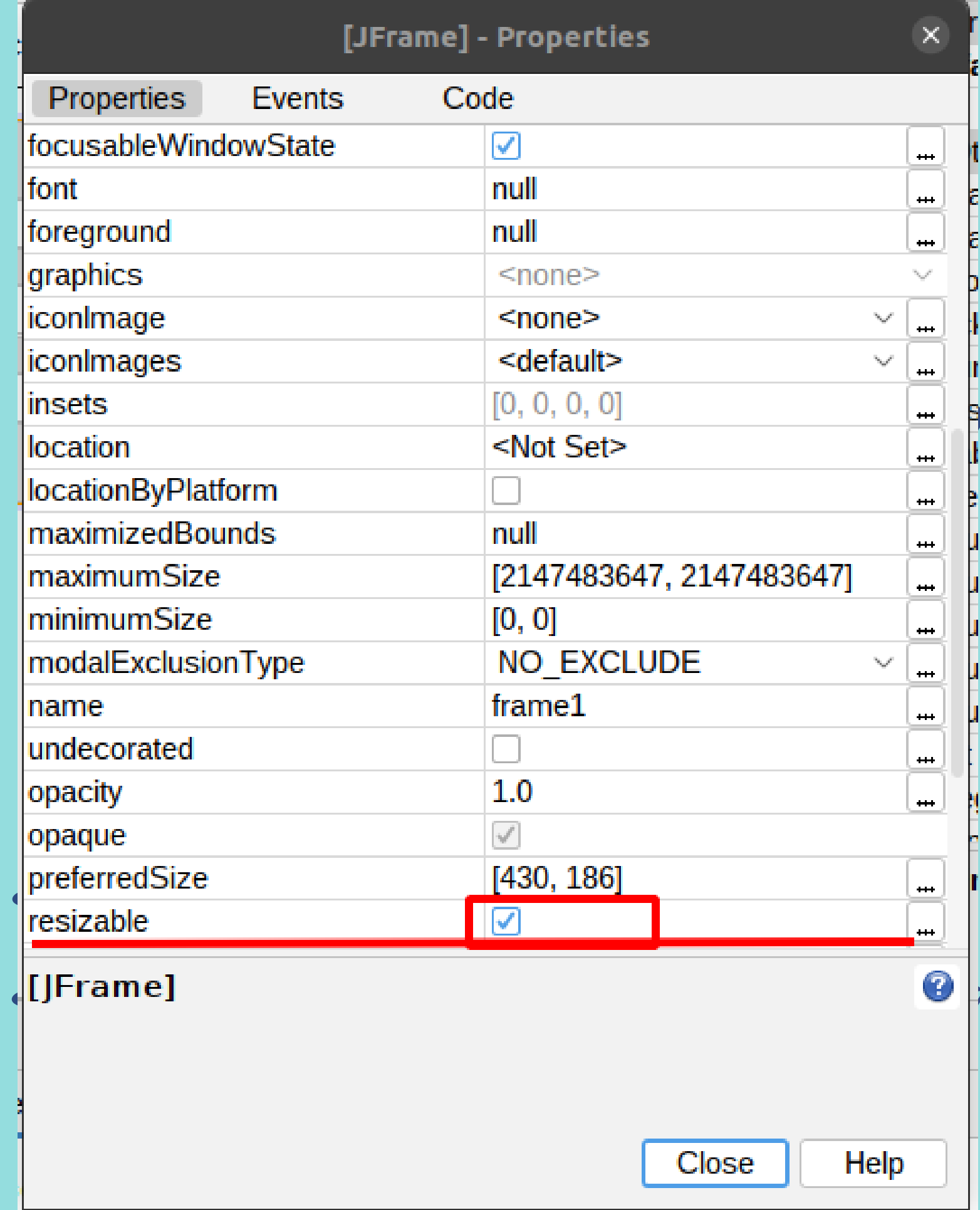
Usuario:

Senha:

A red arrow points to the bottom right corner of the form, indicating a right-click action.

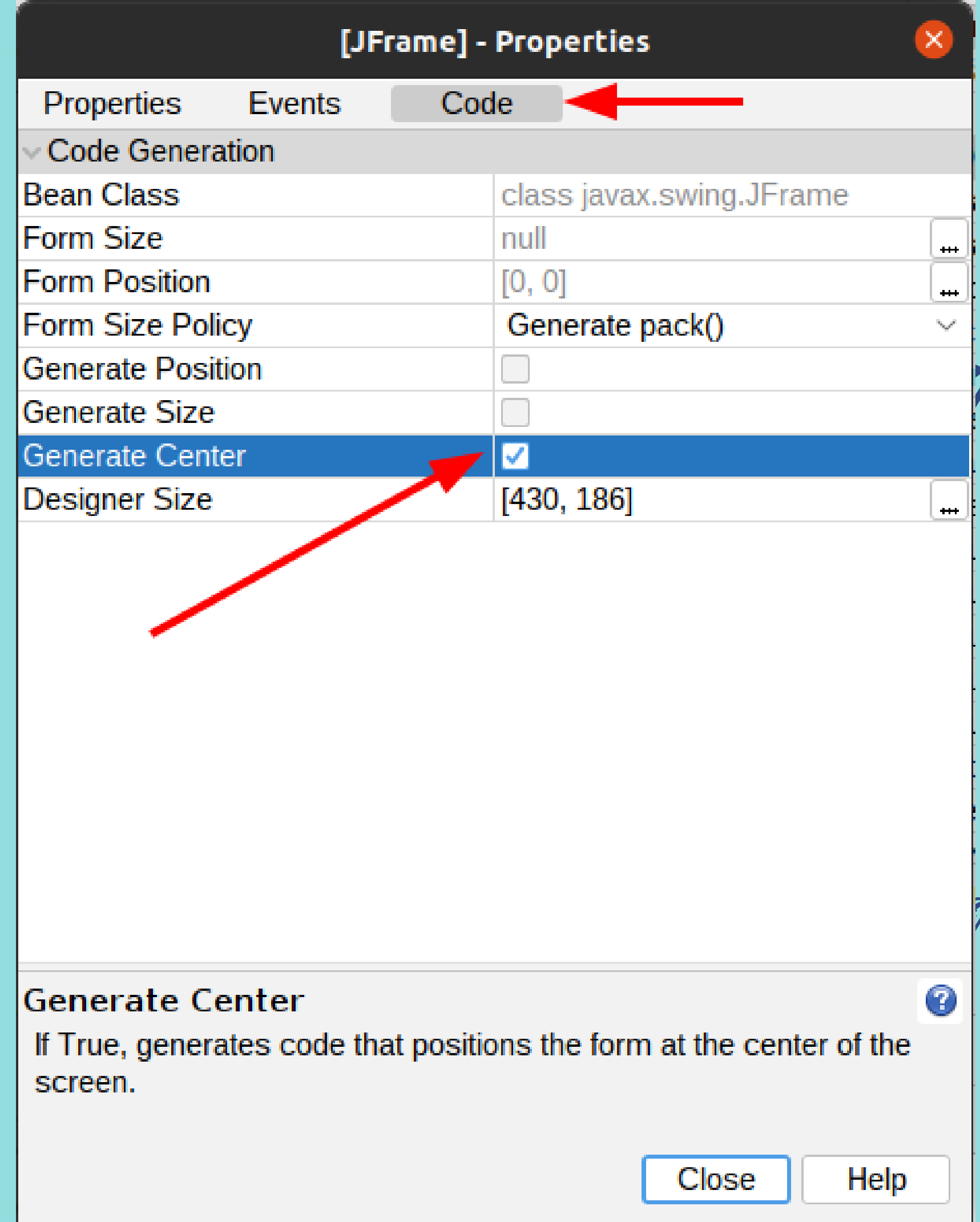


B) Desmarque a opção em "resizable".



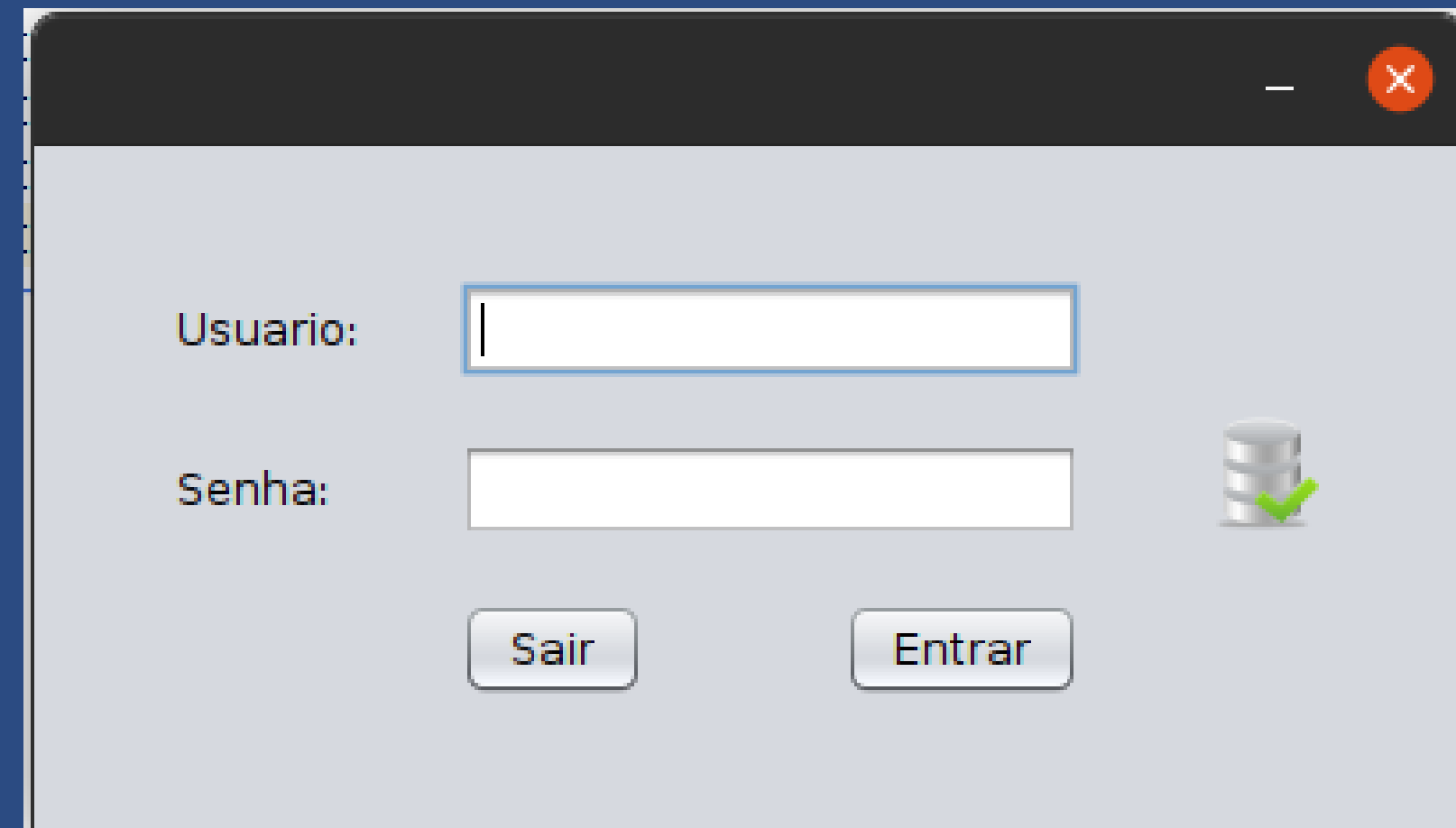
C) Vamos na aba "Code" e em "Generate Center", deixamos a caixa selecionada.

Feito isso é só fechar.

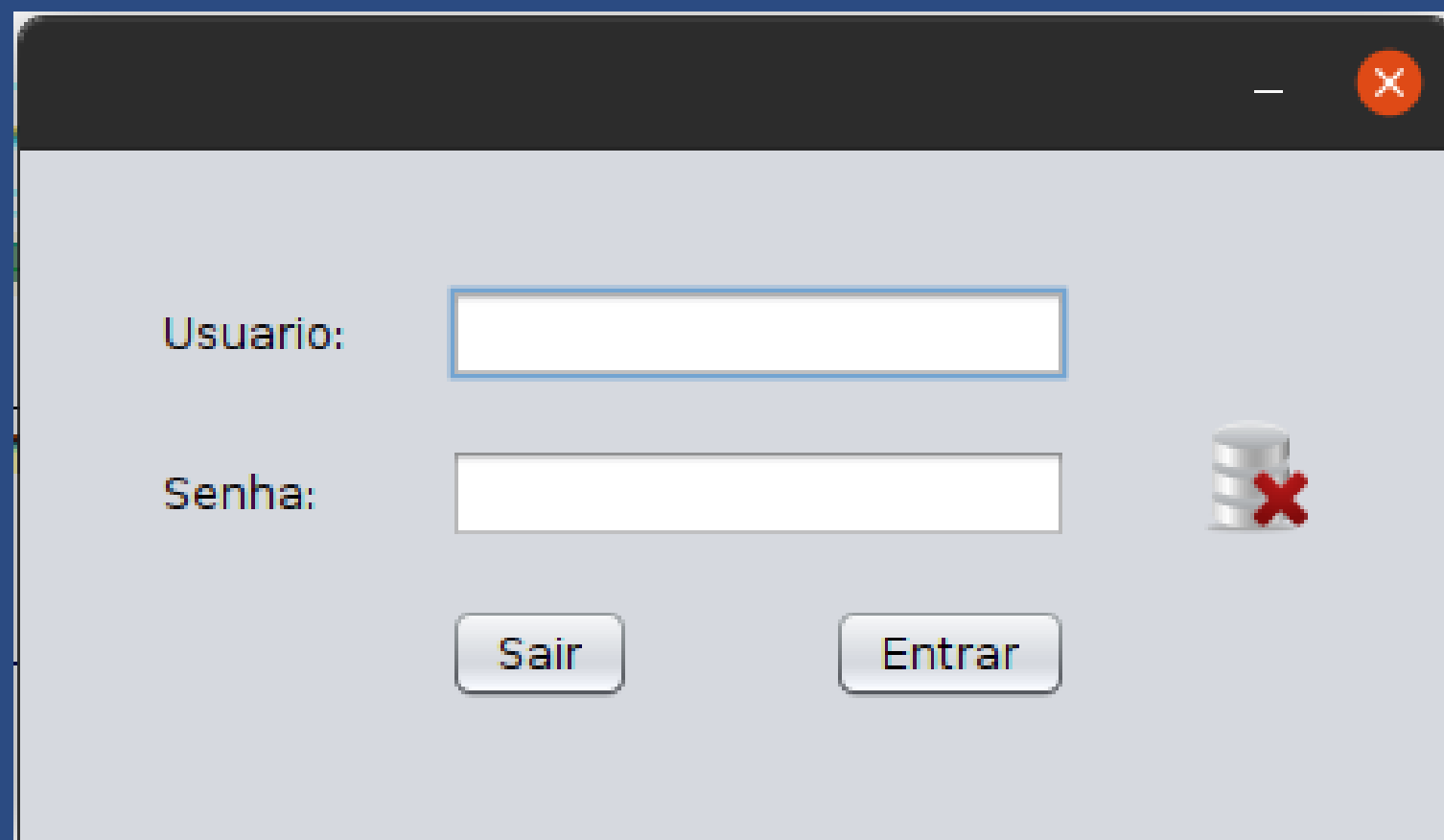


Resultado:

Quando a conexão com o banco de dado é bem-sucedida.



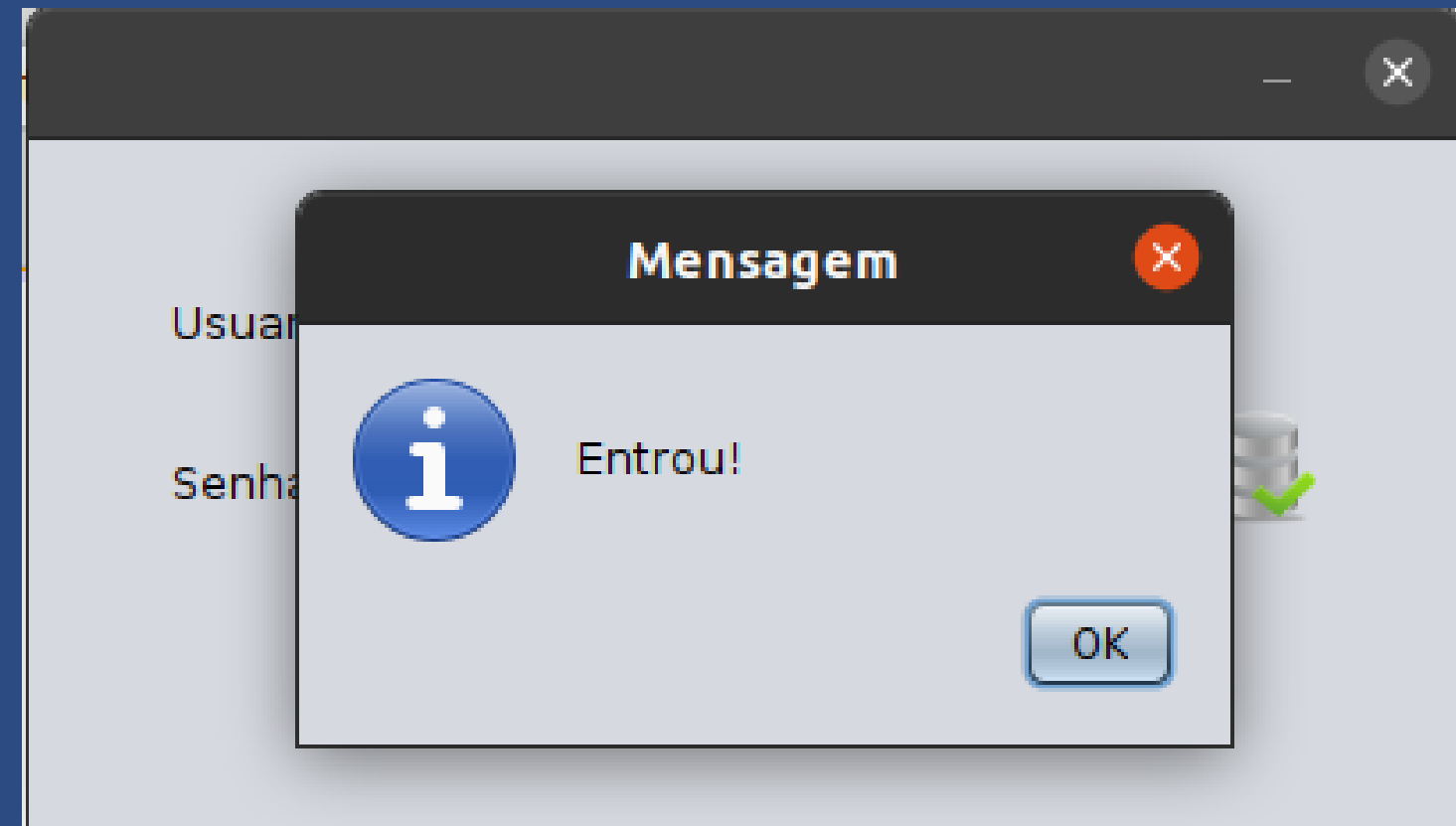
A login form window with a dark title bar containing a close button (red circle with a white 'X'). The form has a light gray background. It contains two input fields: 'Usuario:' and 'Senha:'. To the right of the 'Senha:' field is a database icon (cylinder) with a green checkmark, indicating a successful connection. Below the input fields are two buttons: 'Sair' and 'Entrar'.



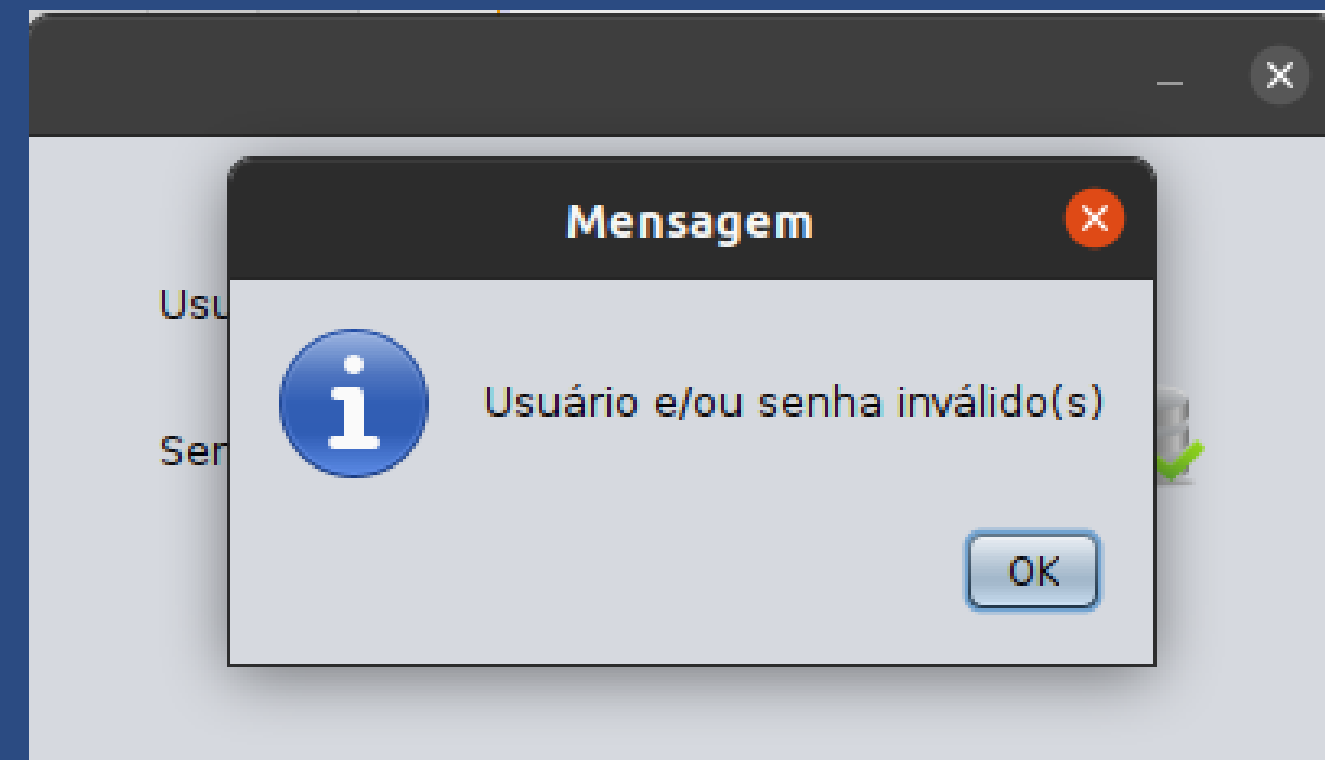
A login form window with a dark title bar containing a close button (red circle with a white 'X'). The form has a light gray background. It contains two input fields: 'Usuario:' and 'Senha:'. To the right of the 'Senha:' field is a database icon (cylinder) with a red 'X', indicating a failed connection. Below the input fields are two buttons: 'Sair' and 'Entrar'.

Quando existe alguma falha.

Quando existe usuário e senha correspondente cadastrados.



Quando há algum erro na digitação do login ou o usuário não existe.





Obrigado!

Veja o projeto feito no link

https://github.com/amorim66/TelaLogin_atividade03TP

