



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

Linguagem de Programação II

Introdução às Interfaces Gráficas de
Usuário (GUI) em Java: Pacote Swing
(JFrame, JPanel, JLabel, JTextField,
JPasswordField e JButton)



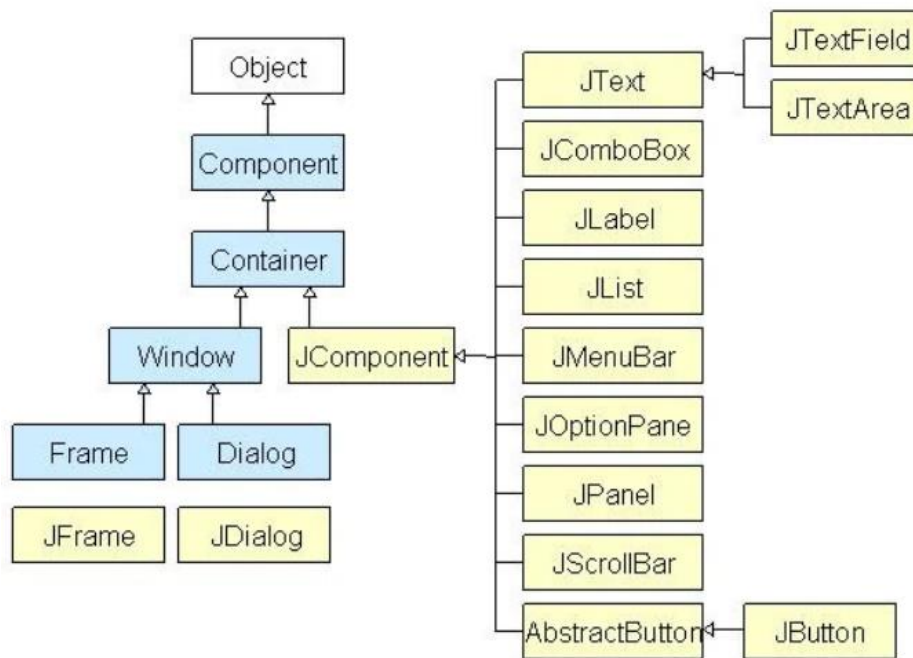
Roteiro

- Introdução às Interfaces Gráficas de Usuário (GUI) em Java (Swing):
 - Introdução;
 - Componentes;
 - JFrame (Janela);
 - Netbeans e Swing
 - JButton (Botão)
 - JLabel (Rótulo ou Etiquetas);
 - JTextField (Campo de Texto);
 - JPasswordField (Campo de Senha);



Java Swing - Introdução

- **Java Swing** é um conjunto classes disponíveis no Java para o desenvolvimento de interfaces gráficas.



Java Swing - Introdução

- Na **Interface Gráfica** é necessário definir quais **componentes (objetos)** serão utilizados e a **disposição** que **eles** terão na **janela (objeto)**.

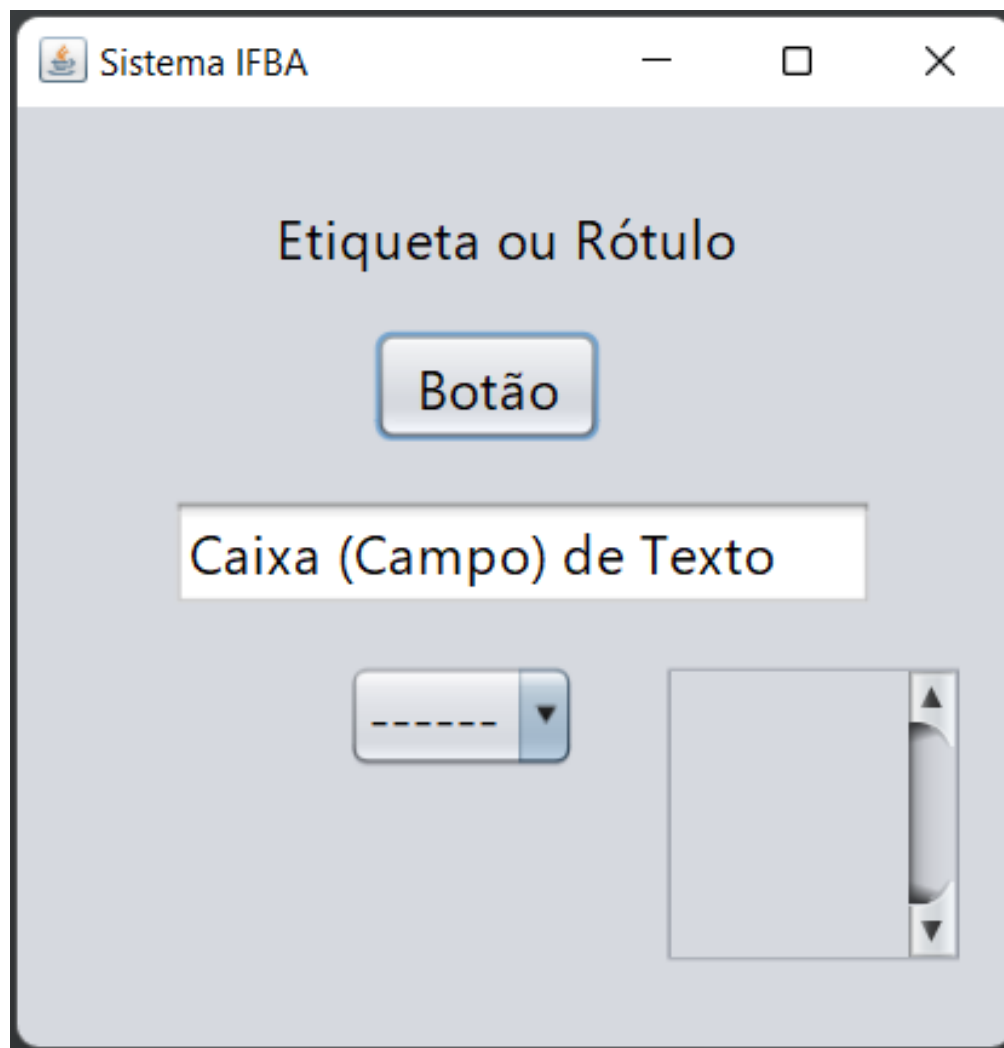


Java Swing – Componentes

- Um **componente** da **Interface Gráfica** é um **objeto visual (Java)** que possibilita realizar a **interação com o programa** por meio do **mouse** e do **teclado**.
- Ex.: **etiquetas, botões, caixas de texto, painéis de rolagem, menus, objetos de múltipla escolha**, entre outros.



Java Swing – Componentes

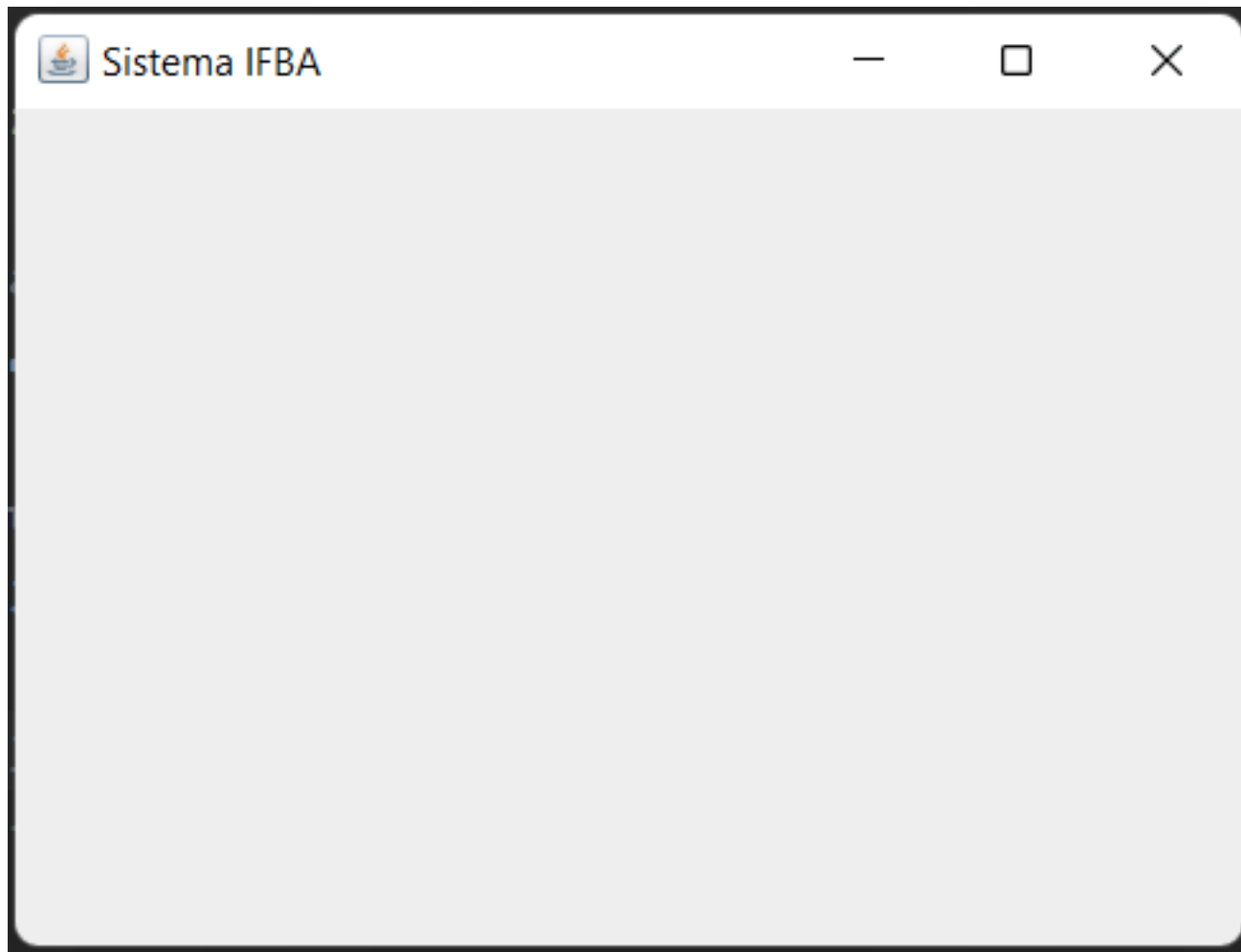


Java Swing – JFrame

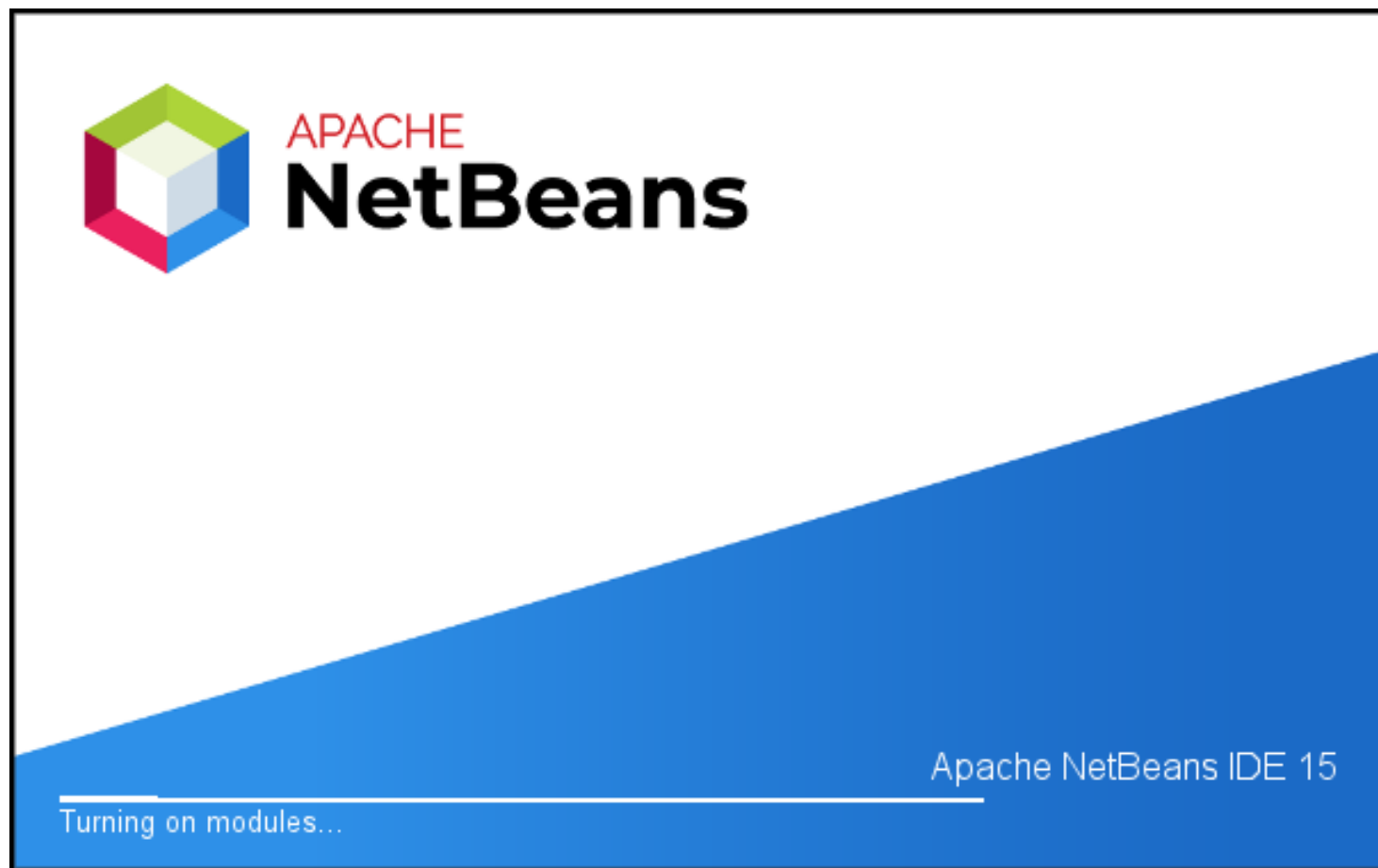
- Para criar **Janelas** utilizaremos a classe **JFrame** disponível no pacote **swing**, a qual gera uma **janela** com **barra de título**, **bordas** e eventualmente outros **componentes visuais**.

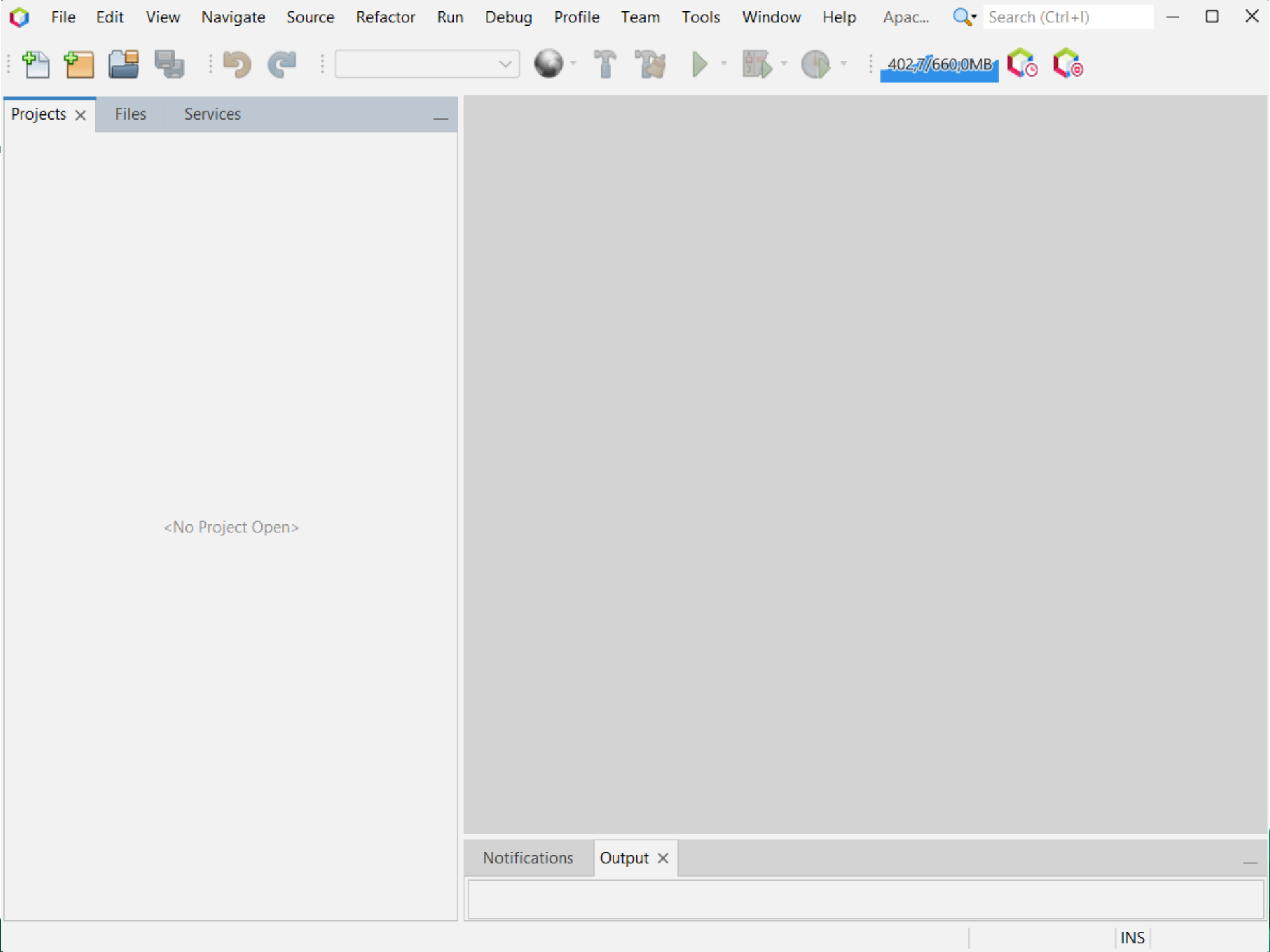


Java Swing – JFrame

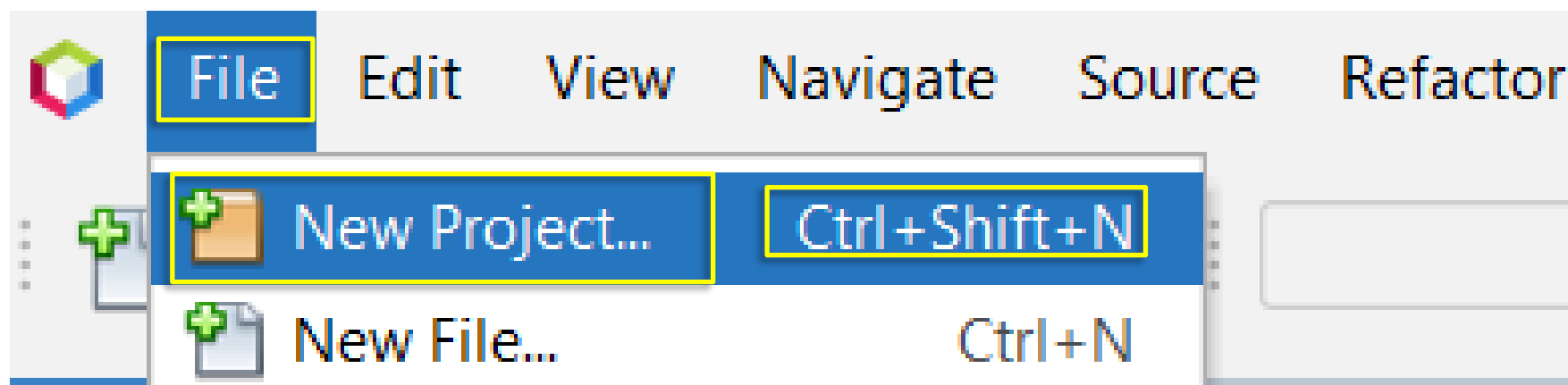


Netbeans: IDE





Netbeans: Novo Projeto



Netbeans: Novo Projeto

Choose Project



Filter:

Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
- JavaFX
- Java Web

Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Modular Project
- Java Free-Form Project



Netbeans: Novo Projeto

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

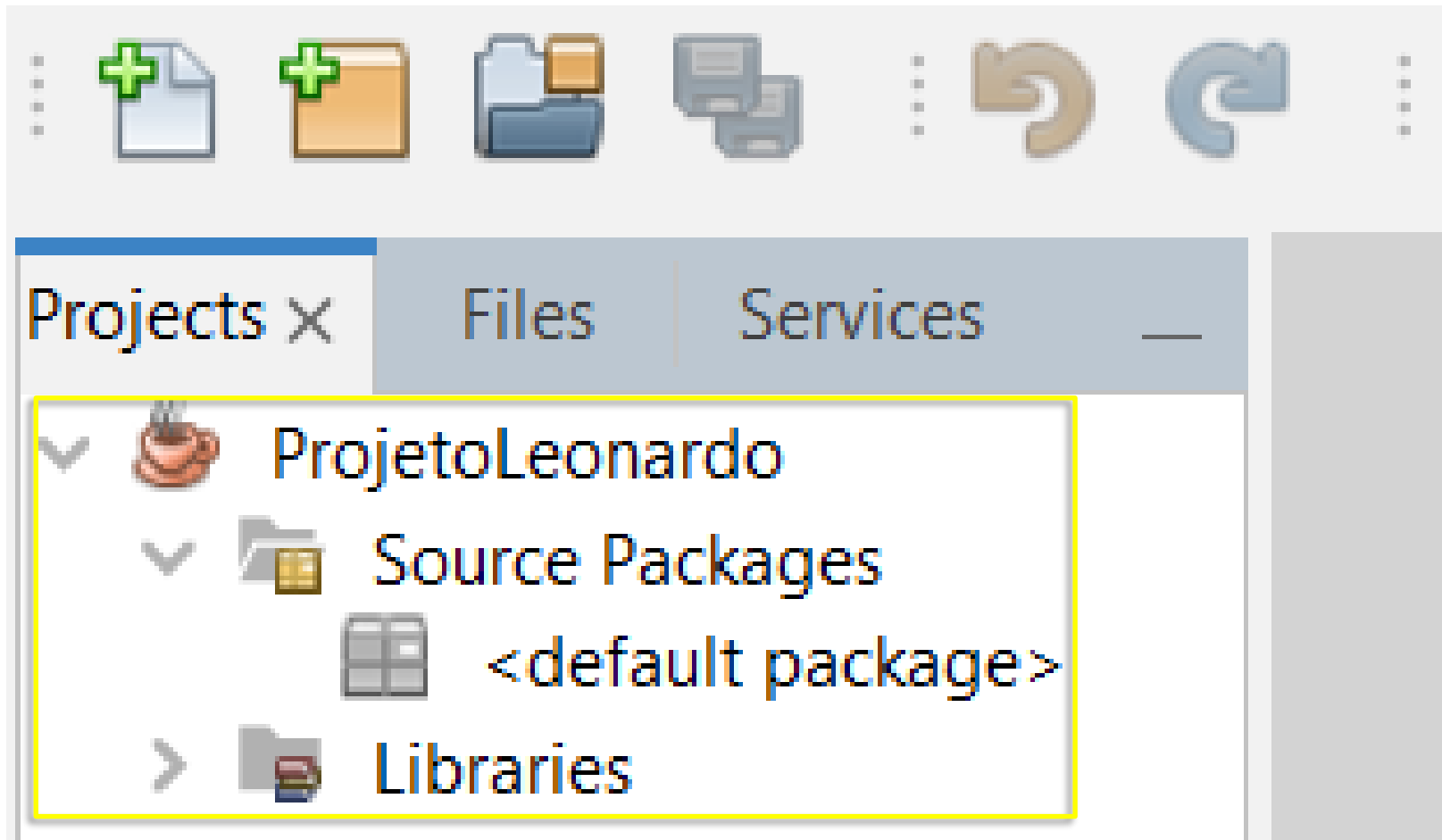
Libraries Folder:

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

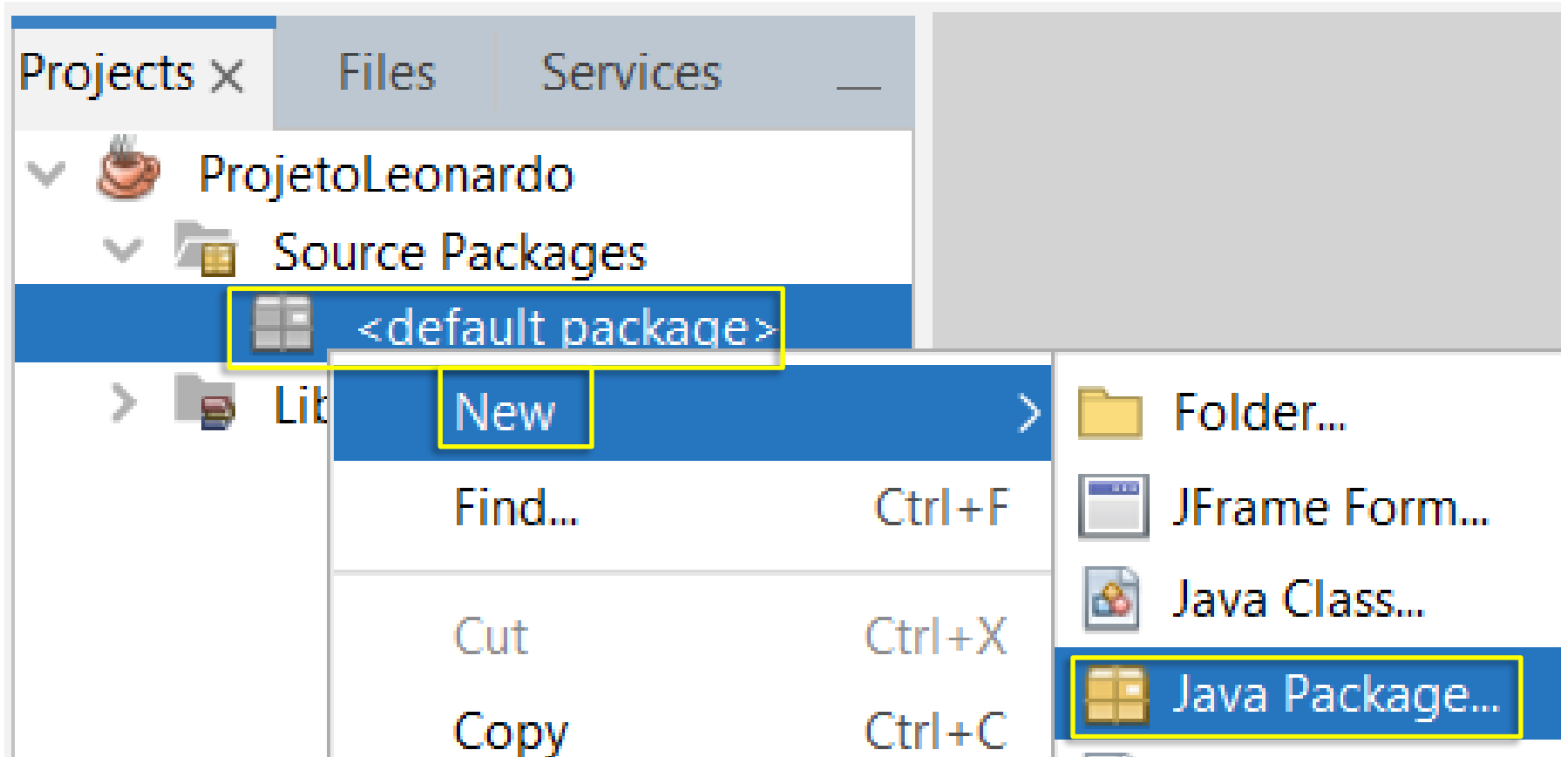
☐ Create Main Class



Netbeans: Novo Projeto



Netbeans: Novo Pacote



Netbeans: Novo Pacote

Name and Location

Package Name:

Principal

Project:

ProjetoLeonardo

Location:

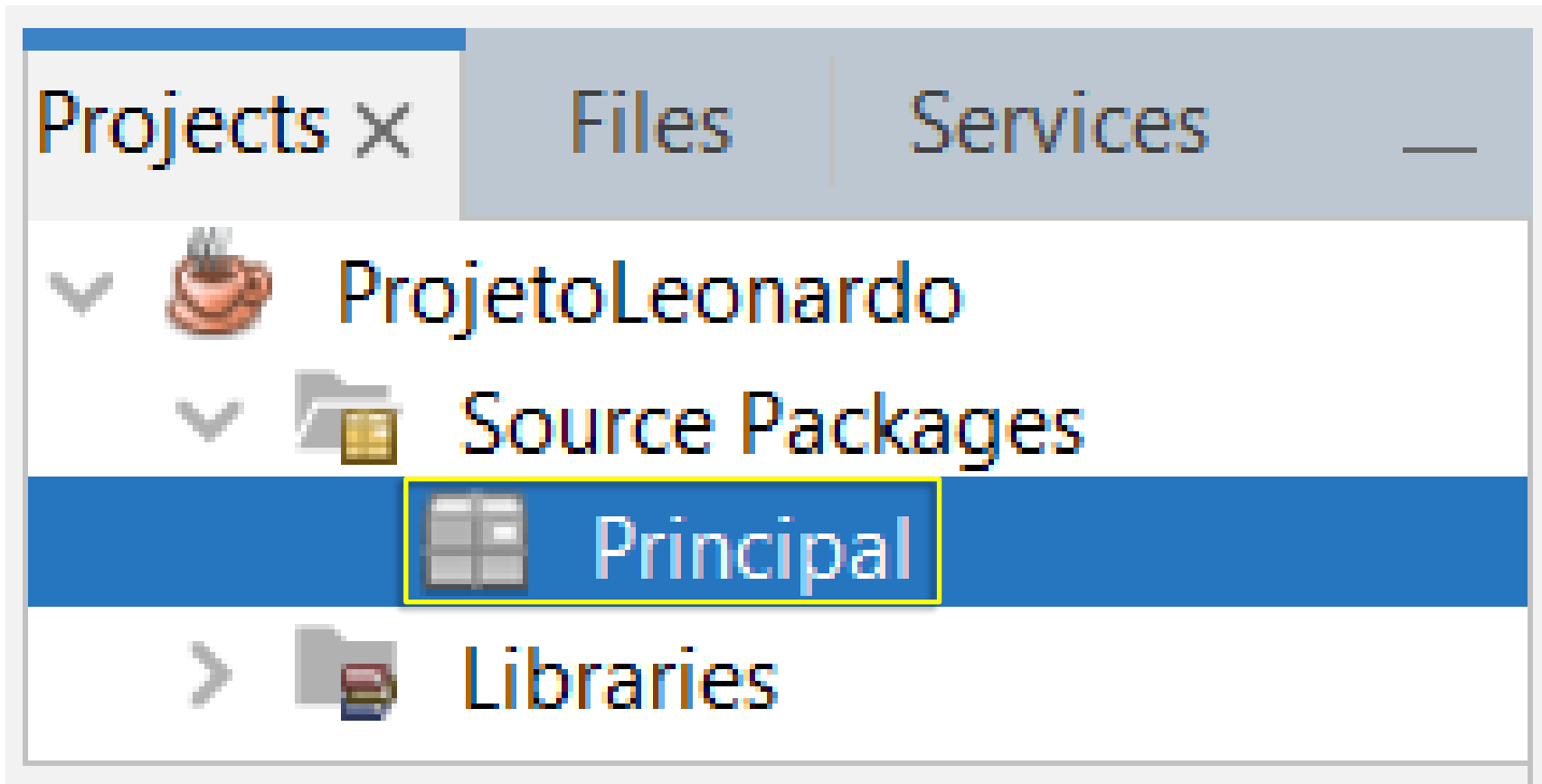
Source Packages

Created Folder:

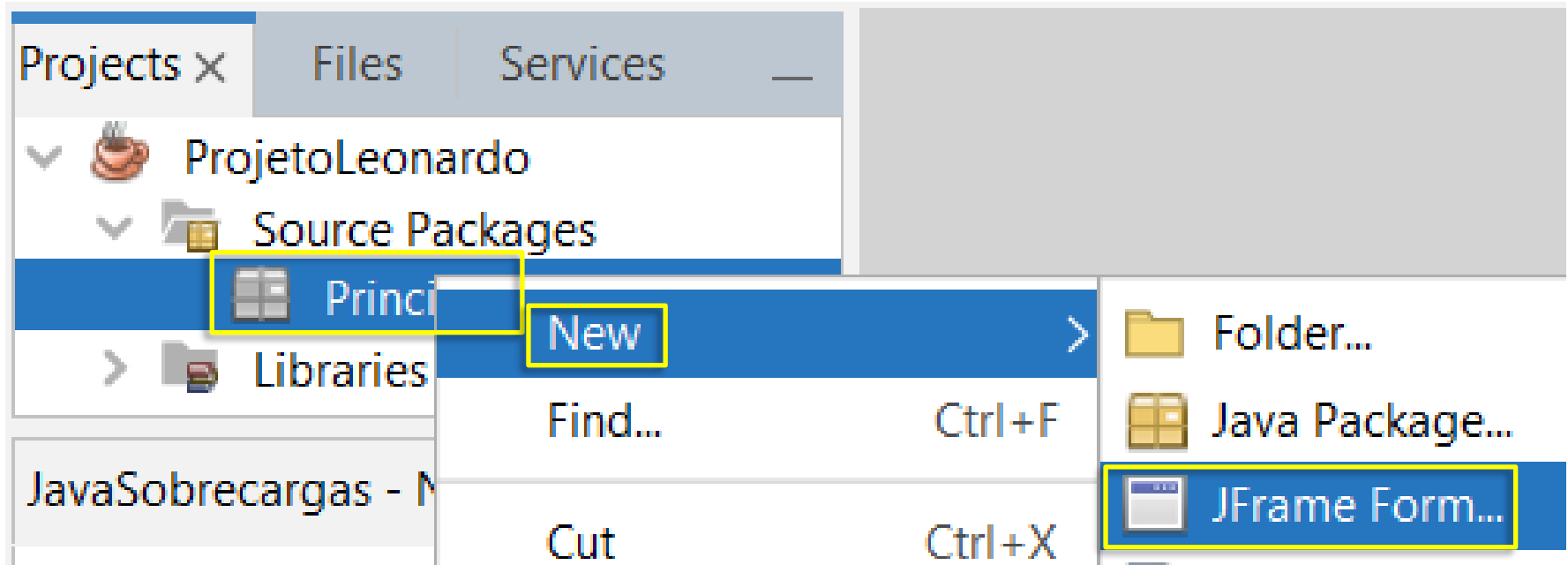
.:\\Users\\leomi\\Documents\\NetBeansProjects\\ProjetoLeonardo\\src\\Principal



Netbeans: Novo Pacote



Netbeans: Nova Janela (JFrame)



Netbeans: Nova Janela (Jframe)

Name and Location

Class Name:

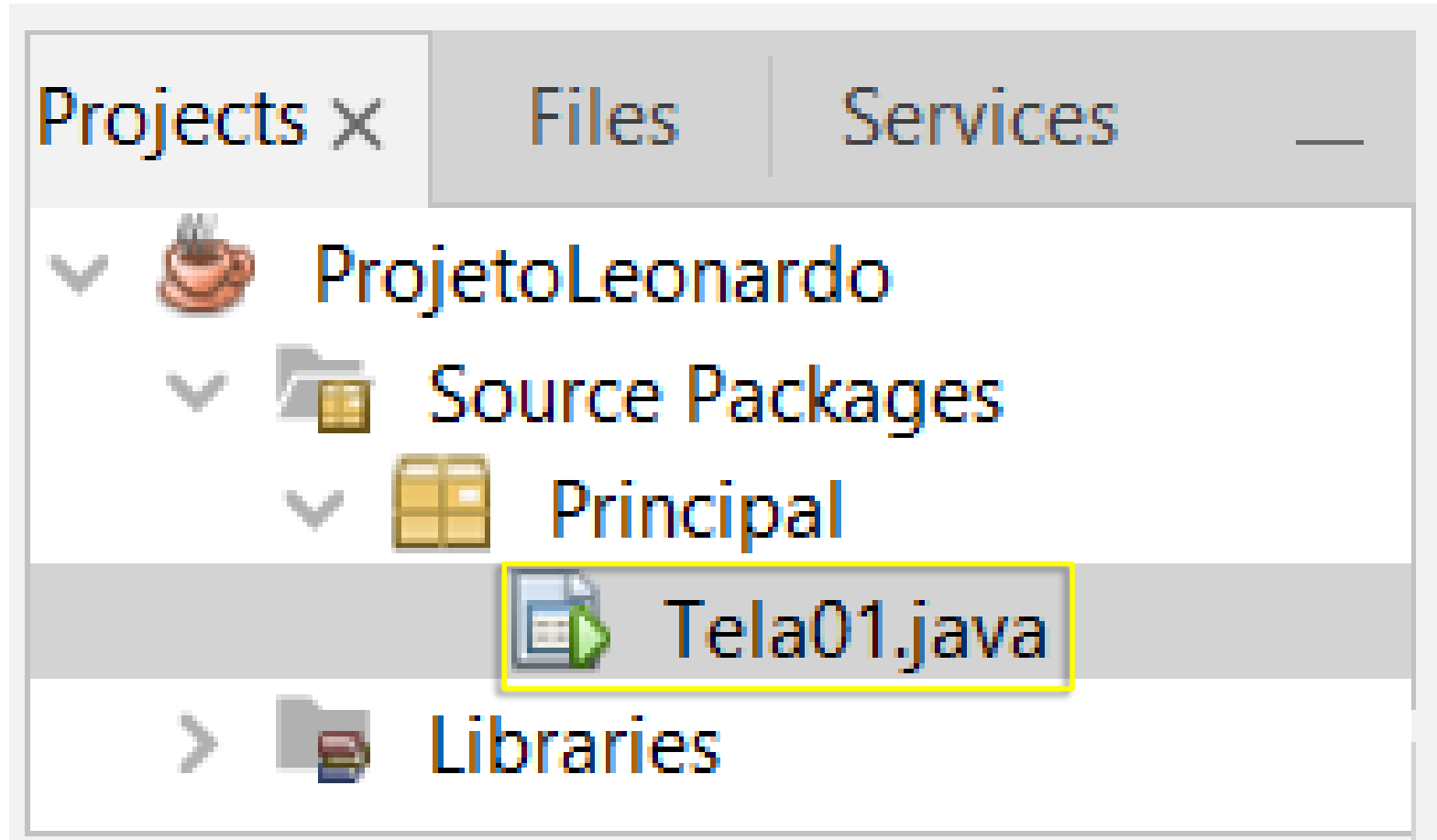
Project:

Location:

Package:



Netbeans: Nova Janela (JFrame)



Projects x Files Services


- ProjetoLeonardo
 - Source Packages
 - Principal
 - Tela01.java
 - Libraries

[JFrame] - Navigator x

- Form Tela01
 - Other Components
 - [JFrame]

Tela01.java x

Source Design History



Palette x

Swing Containers

- Panel
- Tabbed Pane
- Split Pane
- Scroll Pane
- Tool Bar
- Desktop Pane
- Internal Frame

[JFrame] - Properties x

Properties Events

Code

Properties

defaultCloseOperation	EXIT_ON...	...
title		...

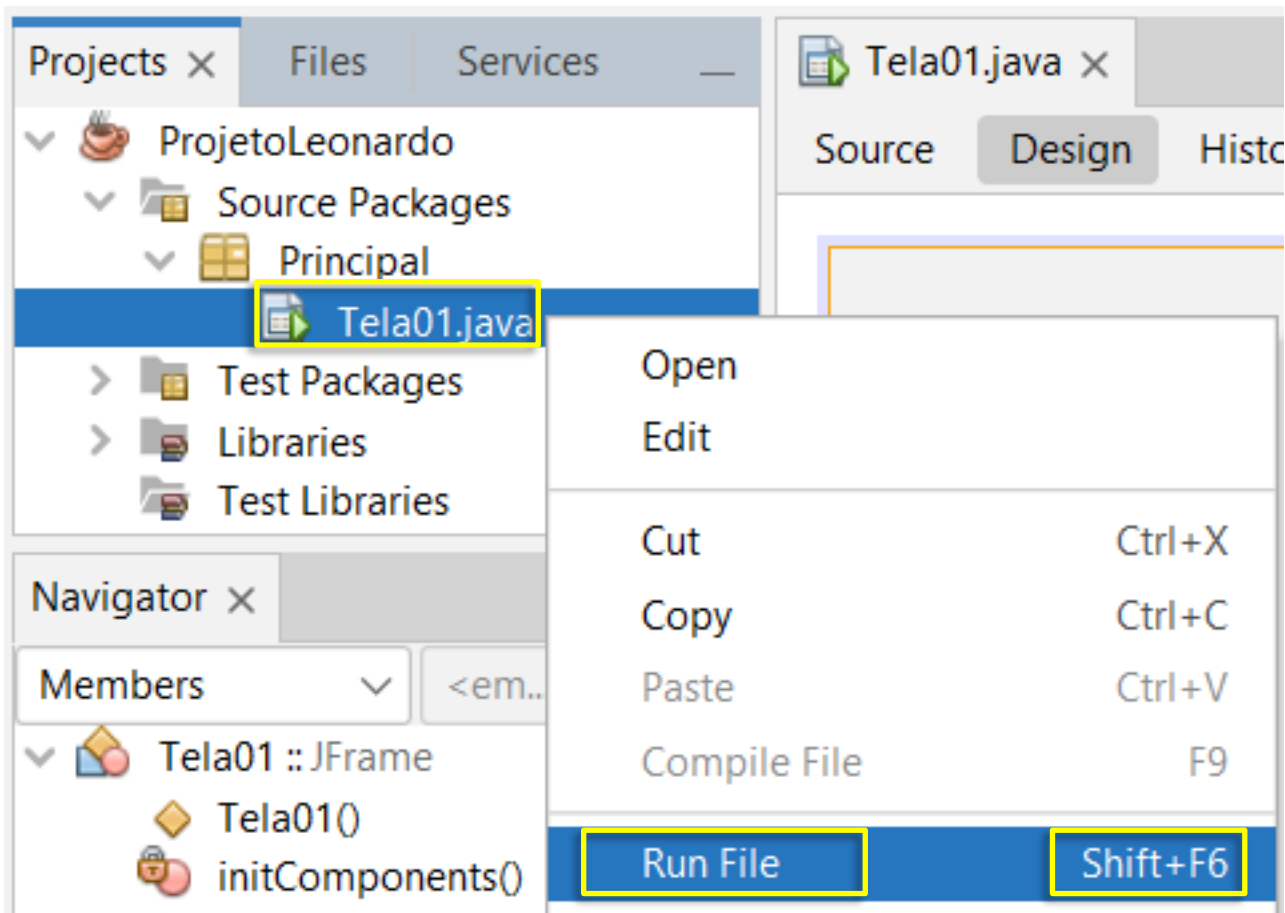
Other Properties

alwaysOnTop	<input type="checkbox"/>	...
alwaysOnTopSupport	<input checked="" type="checkbox"/>	...

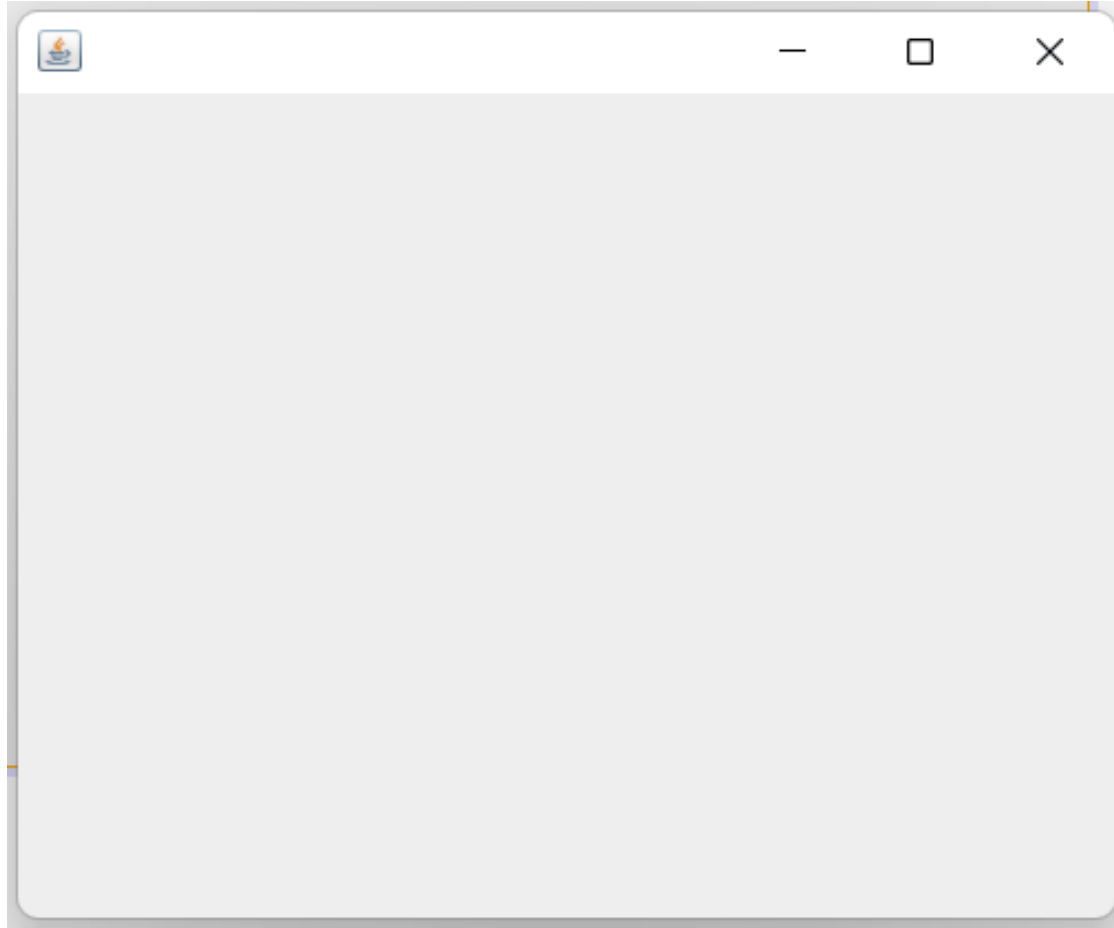
[JFrame] ?

Notifications Output x

Netbeans: Executando Janela (JFrame)



Netbeans: Executando Janela (JFrame)





```
1 package Principal;
```

```
2  
3 public class Tela01 extends javax.swing.JFrame {
```

```
4  
5     public Tela01() {  
6         initComponents();  
7     }  
8
```

```
9 + Generated Code
```

```
27  
28     public static void main(String args[]) {  
29         new Tela01().setVisible(true);  
30     }  
31
```

```
32 // Variables declaration - do not modify  
33 // End of variables declaration  
34
```

```
}
```




```
1 package Principal;
```

Pacote

```
2  
3 public class Tela01 extends javax.swing.JFrame {
```

Construtor

Classe

Herança

```
4 public Tela01() {  
5     initComponents();  
6 }  
7
```

Chamada de método que
inicializa componentes

Generated Code

Inicialização e configuração de
componentes

Método estático "main"

```
27  
28 public static void main(String args[]) {  
29     new Tela01().setVisible(true);  
30 }  
31
```

Declaração
de variáveis

Cria objeto (componente) Tela01 e o torna visível

```
32 // Variables declaration - do not modify  
33 // End of variables declaration  
34 }
```

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

```
private void initComponents() {
```

```
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

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

```
    getContentPane().setLayout(layout);
```

```
    layout.setHorizontalGroup(
```

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

```
            .addGap(0, 400, Short.MAX_VALUE)
```

```
    );
```

```
    layout.setVerticalGroup(
```

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

```
            .addGap(0, 300, Short.MAX_VALUE)
```

```
    );
```

```
    pack();
```

```
}// </editor-fold>
```

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

```
private void initComponents() {
```

Método

Comportamento do ícone (botão) de fechar o JFrame

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

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

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

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

```
        .addGap(0, 400, Short.MAX_VALUE)
```

```
);
```

```
layout.setVerticalGroup(
```

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

```
        .addGap(0, 300, Short.MAX_VALUE)
```

```
);
```

```
pack();
```

Layout Manager do JFrame ajusta o tamanho da tela (Frame) com base em seus componentes internos

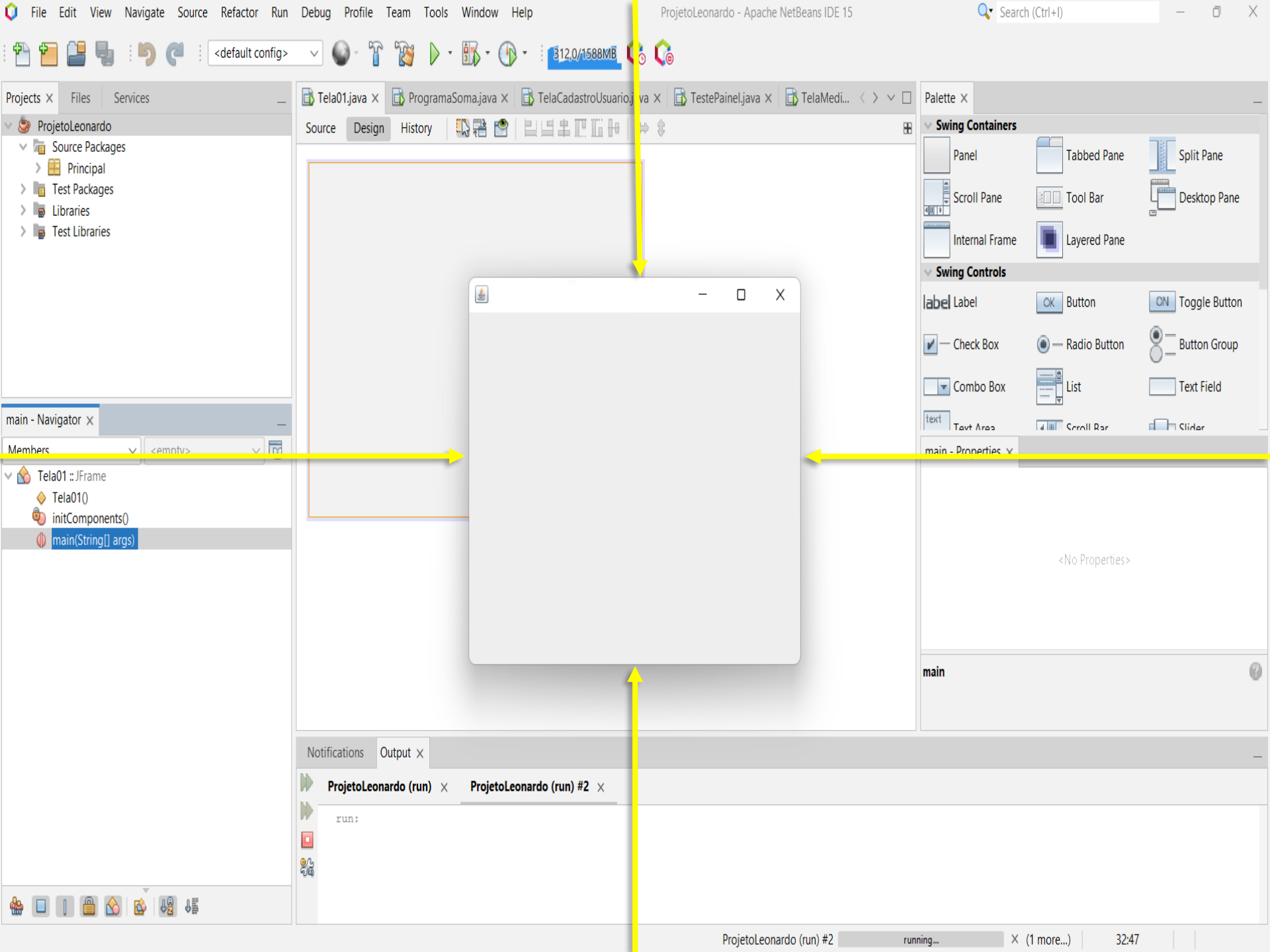
Configurações de Layout do JFrame

```
} // </editor-fold>
```

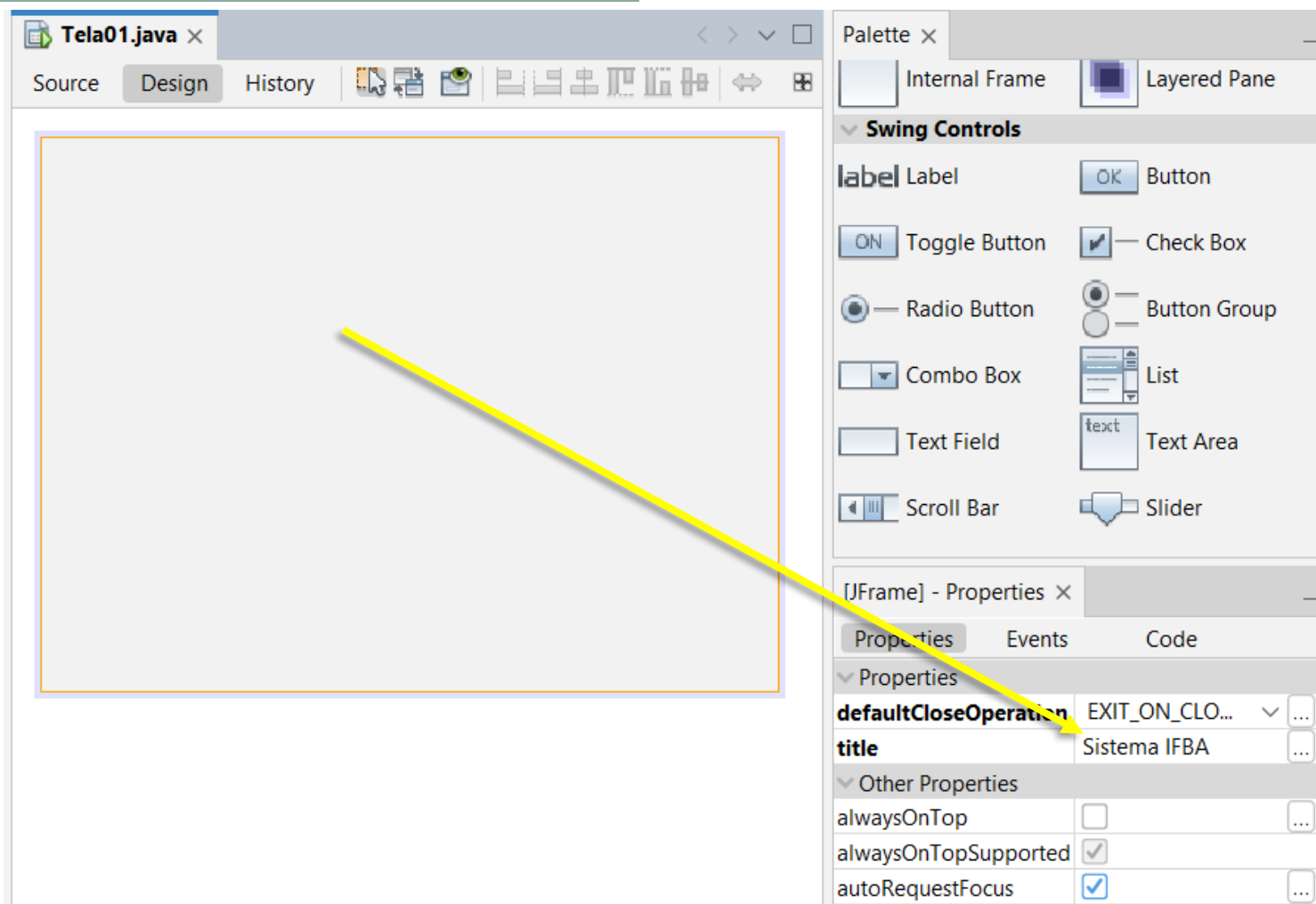
Netbeans: Centralizar JFrame

```
public Tela01 () {  
    initComponents () ;  
    this.setLocationRelativeTo (null) ;  
}
```

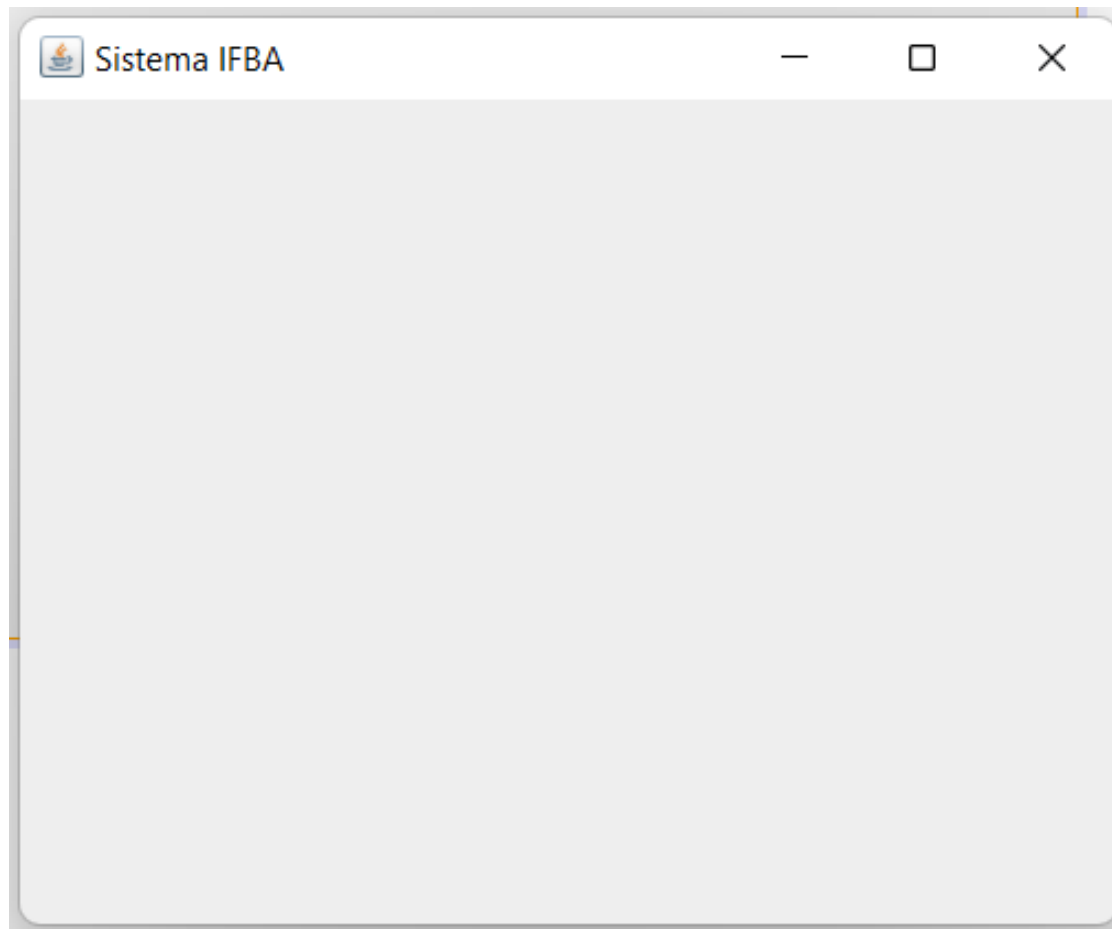




Netbeans: Título da Janela (JFrame)



Netbeans: Título da Janela (JFrame)



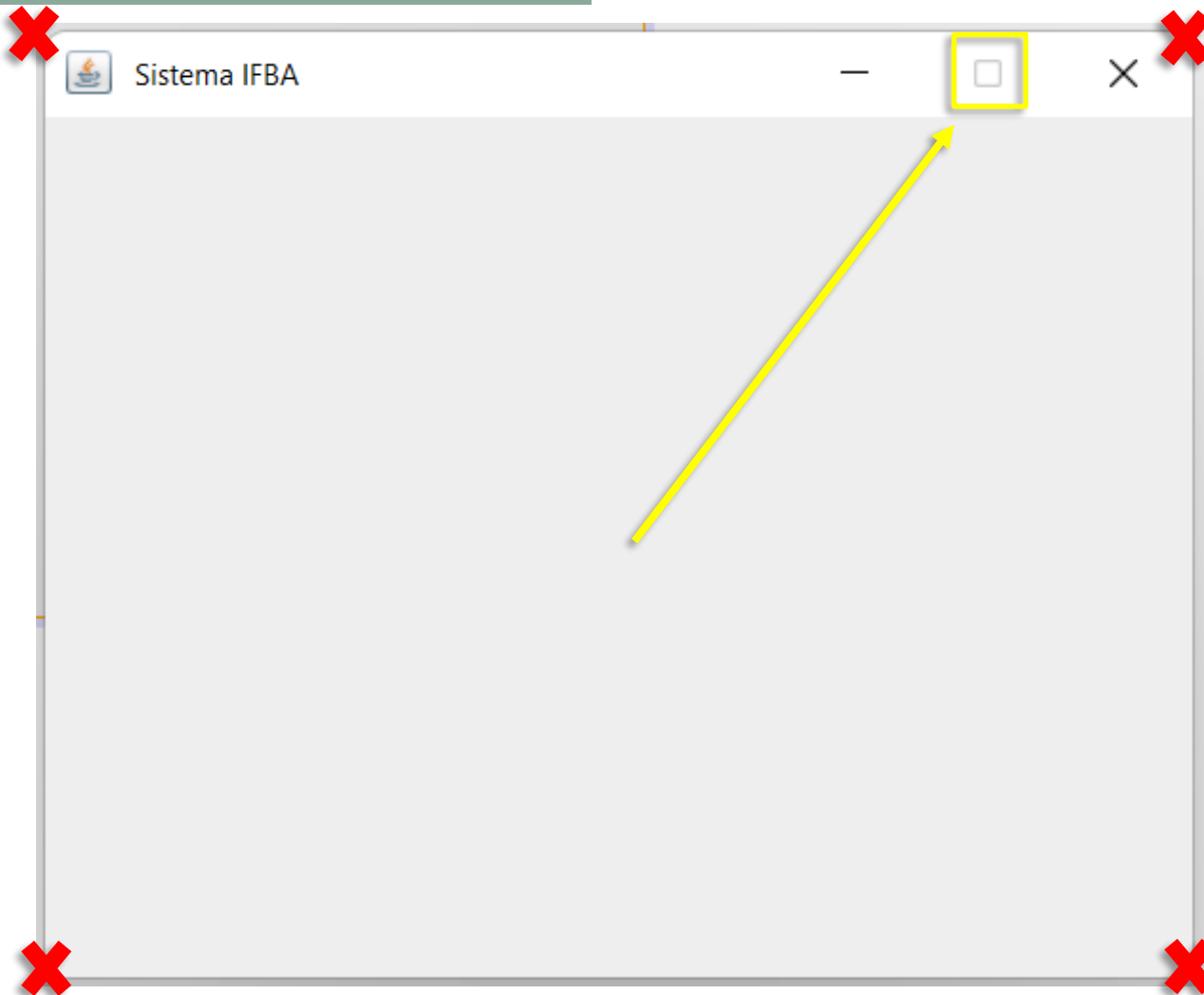
Netbeans: Desabilitar Redimensionar

[JFrame] - Properties X

Properties	Events	Code
preferredSize	[395, 300]	...
resizable	<input type="checkbox"/>	...
shape	<none>	False
size	<Not Set>	...
state	0	...
type	NORMAL	...



Netbeans: Desabilitar Redimensionar

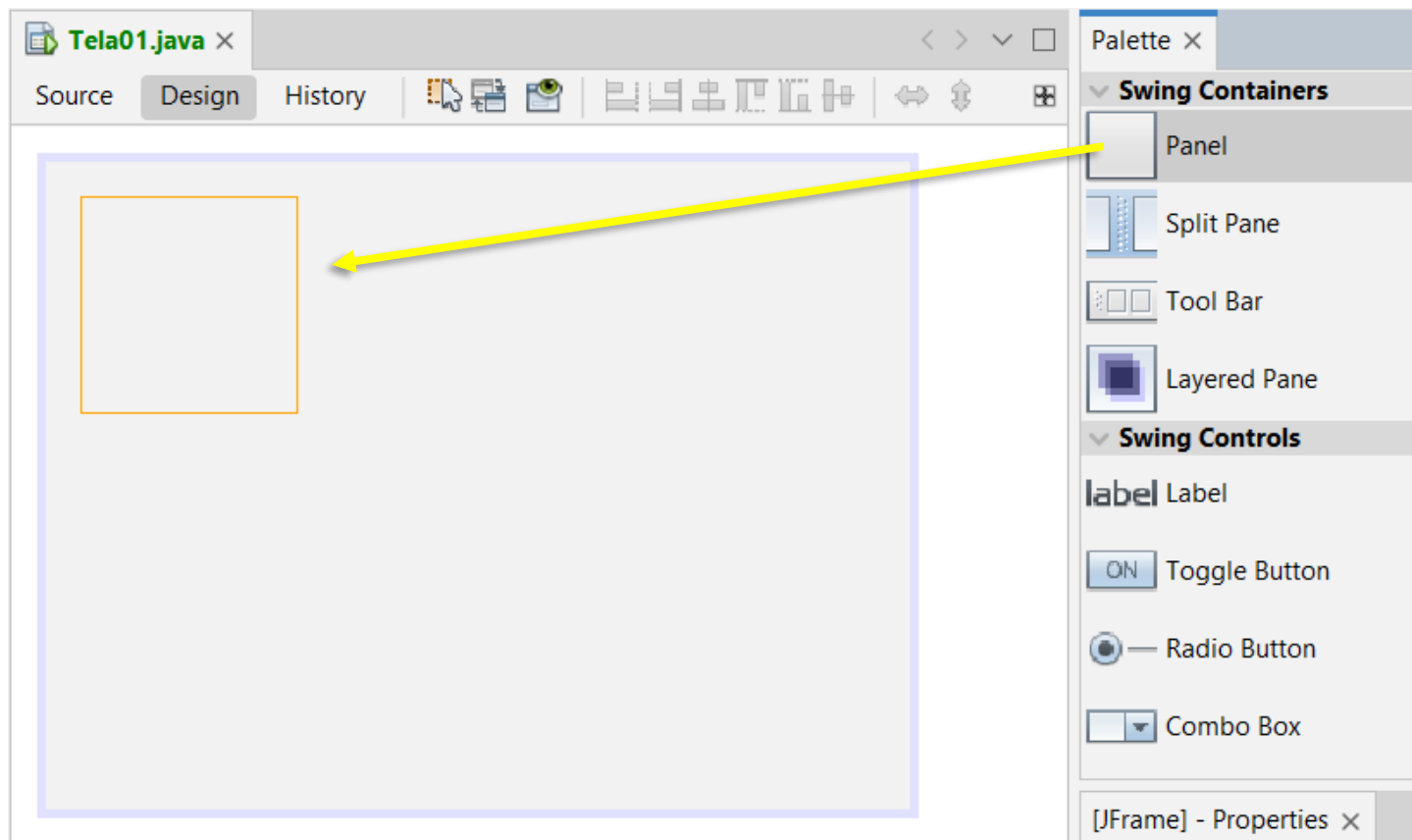


Java Swing – JPanel

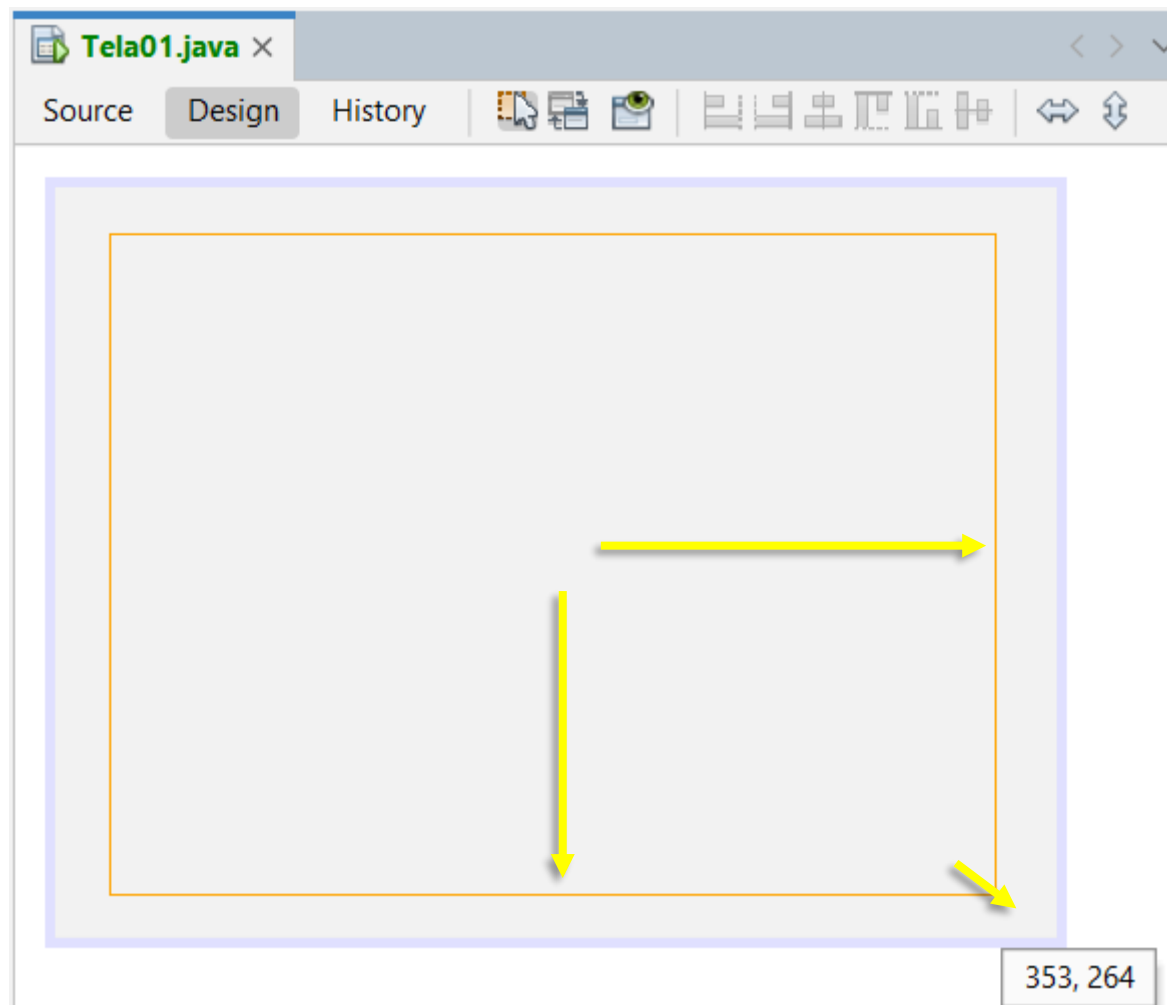
- O **JPanel** é um **contêiner flexível** no Java Swing, utilizado para **agrupar e organizar componentes** em interfaces gráficas.
- Ele **oferece opções de personalização**, como **layout e cor de fundo**, permitindo criar **interfaces atraentes e adaptáveis**.



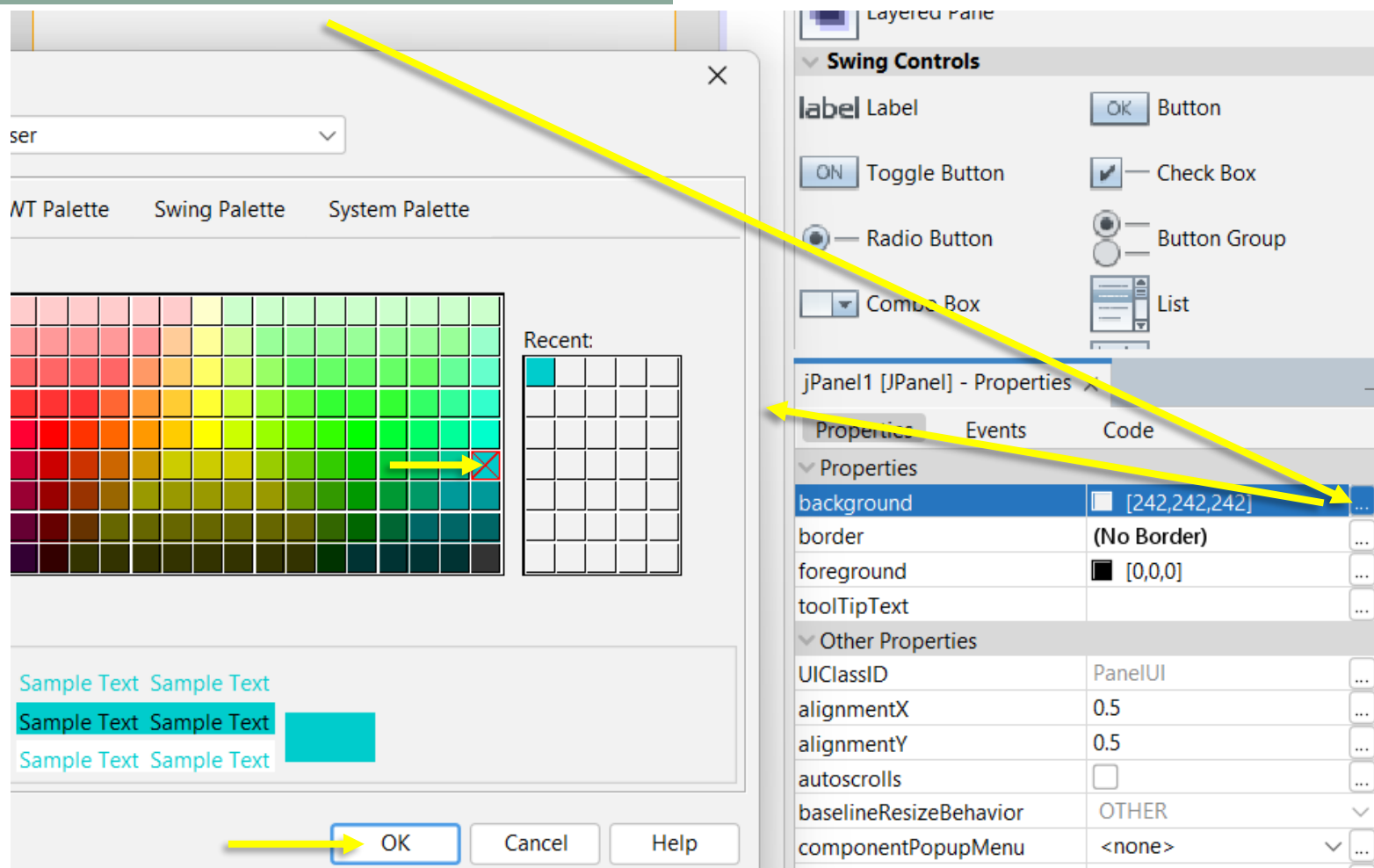
Java Swing – JPanel - Adicionar



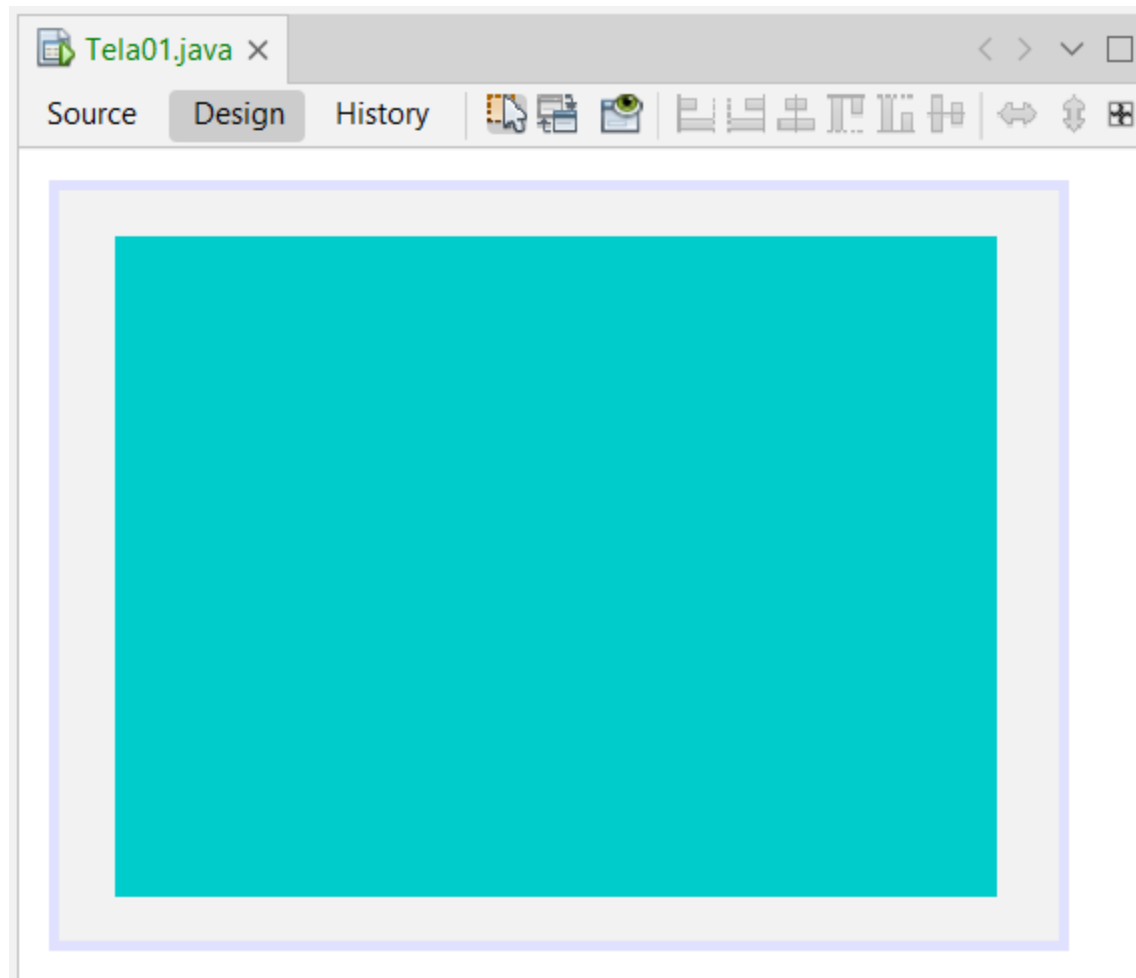
Java Swing – JPanel - Redimensionar



Java Swing – JPanel – Mudar Cor Fundo



Java Swing – JPanel – Mudar Cor Fundo

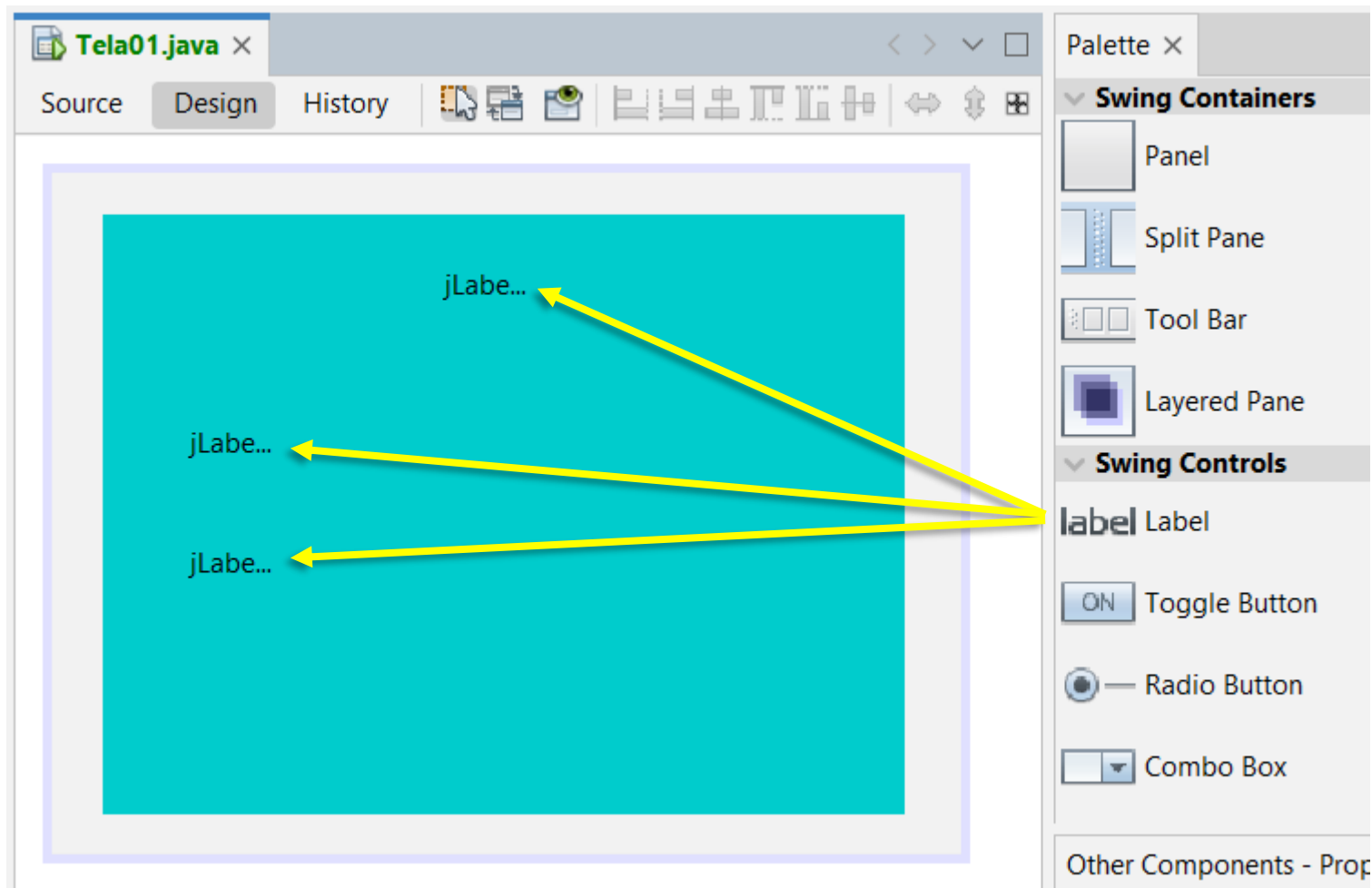


Java Swing – JLabel

- A classe **JLabel** permite **definir um texto** que **pode ser adicionado** a um **outro componente** (frame, painel etc.).
- Podem ser definidas várias **propriedades** para esse texto, tais como **alinhamento**, **tipo de letra**, **tamanho**, **cor** etc.



Java Swing – JLabel – Adicionar



Java Swing – JLabel – Editor Texto

The screenshot displays the Java Swing IDE interface. The main window shows a design view of a Java Swing component named 'Tela01.java'. The component is a light blue rectangle with a darker blue header bar. A yellow arrow points from the 'Tela 01' text in the header bar to the 'text' property in the 'jLabel1 [JLabel] - Properties' window. The 'Properties' window shows the following properties:

Properties	Events	Code
icon		<none>
labelFor		<none>
text		Tela 01
toolTipText		

The 'Palette' window on the right shows the 'Swing Containers' and 'Swing Controls' sections. The 'Swing Controls' section includes 'Label' and 'Button'.



Java Swing – JLabel – Editor Texto

The screenshot shows the Java Swing IDE with the following components:

- Design View:** A cyan rectangular area representing the GUI. Inside, there is a grey rectangular component labeled "Tela 0...". Below it, a smaller grey component is labeled "Mens" and "jLabe...". A yellow arrow points from the "text" property in the Properties window to the "Mens" label.
- Palette:** A panel on the right side of the IDE showing various Swing components. It is divided into two sections:
 - Swing Containers:** Panel, Split Pane, Tool Bar, Layered Pane, Tabbed Pane, Scroll Pane, Internal Frame.
 - Swing Controls:** Label, Button.
- Properties Window:** A window titled "jLabel2 [JLabel] - Properties" with three tabs: Properties, Events, and Code. The Properties tab is active, showing the following properties:

Properties	Events	Code
icon		<none>
labelFor		<none>
text		Mensagem 1:
toolTipText		



Java Swing – JLabel – Editor Texto

The screenshot shows an IDE window titled 'Tela01.java' with tabs for Source, Design, and History. The Design view displays a graphical user interface with a cyan background. A grey panel labeled 'Tela 0..' contains a vertical grey bar with two labels 'Mens...' and a wavy line. A yellow arrow points from the 'Mens...' label to the 'jLabel3 [JLabel] - Properties' window. The 'Palette' window on the right shows 'Swing Containers' and 'Swing Controls'. The 'Properties' window shows the 'text' property set to 'Mensagem 2:'.

Swing Containers

- Panel
- Split Pane
- Tool Bar
- Layered Pane
- Tabbed Pane
- Scroll Pane
- Internal Frame

Swing Controls

- label Label
- OK Button

jLabel3 [JLabel] - Properties

Properties	Events	Code
icon		<none>
labelFor		<none>
text		Mensagem 2:
toolTipText		



Java Swing – JLabel – Aumentar Fonte

The image shows a Java Swing IDE interface with a yellow arrow pointing from a JLabel component in a window titled 'Tela 0...' to the 'jLabel1 [JLabel] - font' dialog box. The dialog box has a title bar with a close button and a subtitle 'Set jLabel1's font property using: Default editor'. It contains a checkbox 'Derive the font from the default font' which is unchecked. Below this are three columns: 'Font', 'Font Style', and 'Size'. The 'Font' column has a list with 'Segoe UI' selected. The 'Font Style' column has a list with 'Plain' selected. The 'Size' column has a list with '24' selected. At the bottom of the dialog is a 'Preview' section with the text 'The quick brown fox jumps over the lazy dog'. Below the preview are three buttons: 'OK', 'Cancel', and 'Help'. A yellow arrow points from the 'OK' button to the 'jLabel1 [JLabel] - Properties' panel on the right. The 'Properties' panel has tabs for 'Properties', 'Events', and 'Code'. The 'Properties' tab is active, showing a table of properties. The 'font' property is highlighted in blue, and its value is 'Segoe UI 12 Plain'. A yellow arrow points from the 'font' property value to the 'font' property value in the 'jLabel1 [JLabel] - font' dialog box.

jLabel1 [JLabel] - font

Set **jLabel1's font** property using: Default editor

☐ Derive the font from the default font

Font:	Font Style:	Size:
Segoe UI	Plain	24
Segoe Print	Plain	14
Segoe Script	Bold	18
Segoe UI	Italic	24
Segoe UI Black	Bold Italic	36

Preview: The quick brown fox jumps over the lazy dog

OK **Cancel** **Help**

jLabel1 [JLabel] - Properties

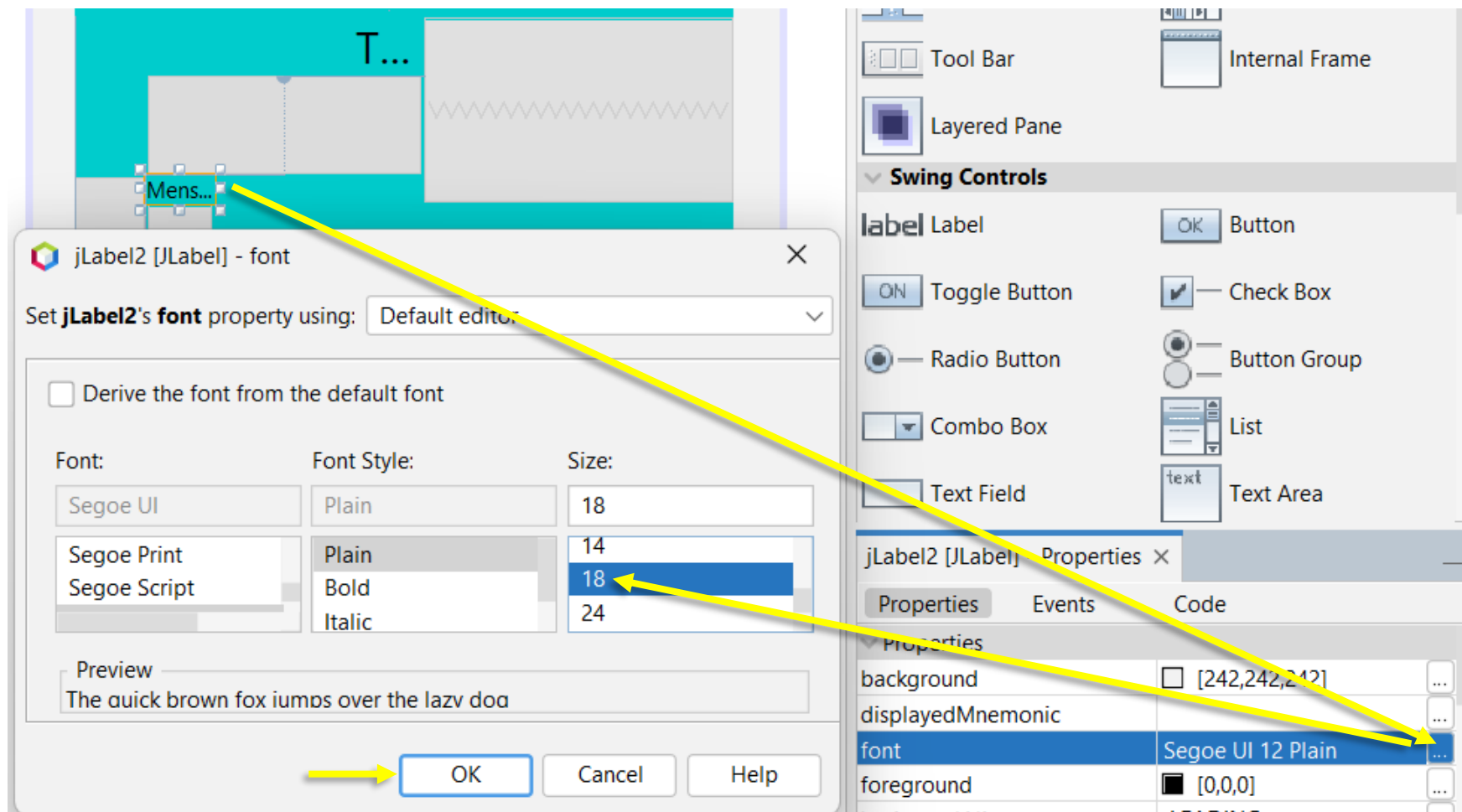
Properties Events Code

Properties

background	[242,242,242]
displayedMnemonic	
font	Segoe UI 12 Plain
foreground	[0,0,0]



Java Swing – JLabel – Aumentar Fonte



Java Swing – JLabel – Aumentar Fonte

The image shows a Java Swing IDE interface with two main windows. The top window is a preview of a JLabel component with the text "Mens...". Below it is a dialog box titled "jLabel3 [JLabel] - font". The dialog box has a dropdown menu set to "Default editor". It contains three sections: "Font:" with a list box showing "Segoe UI" and "Segoe Print"; "Font Style:" with a list box showing "Plain" and "Bold"; and "Size:" with a list box showing "18" and "24". A yellow arrow points from the "18" in the "Size:" list box to the "font" property in the "Properties" tab of the "jLabel3 [JLabel] - Properties" window. The "font" property is currently set to "Segoe UI 12 Plain". Another yellow arrow points from the "OK" button in the font dialog to the "font" property in the properties window. The "Properties" tab also shows other properties like "background", "displayedMnemonic", "foreground", "horizontalAlignment", and "icon".

Font settings for JLabel3:

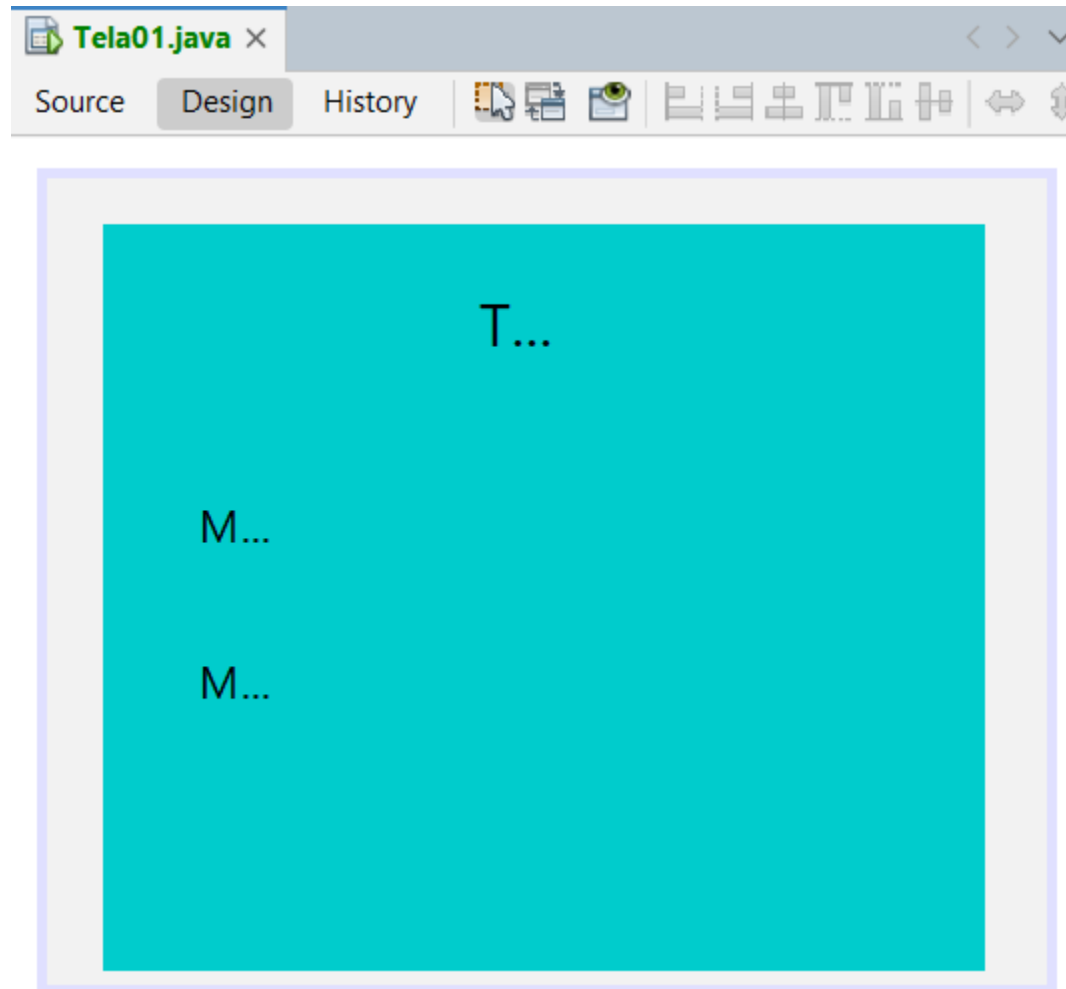
- Font: Segoe UI
- Font Style: Plain
- Size: 18

Properties for JLabel3 [JLabel]:

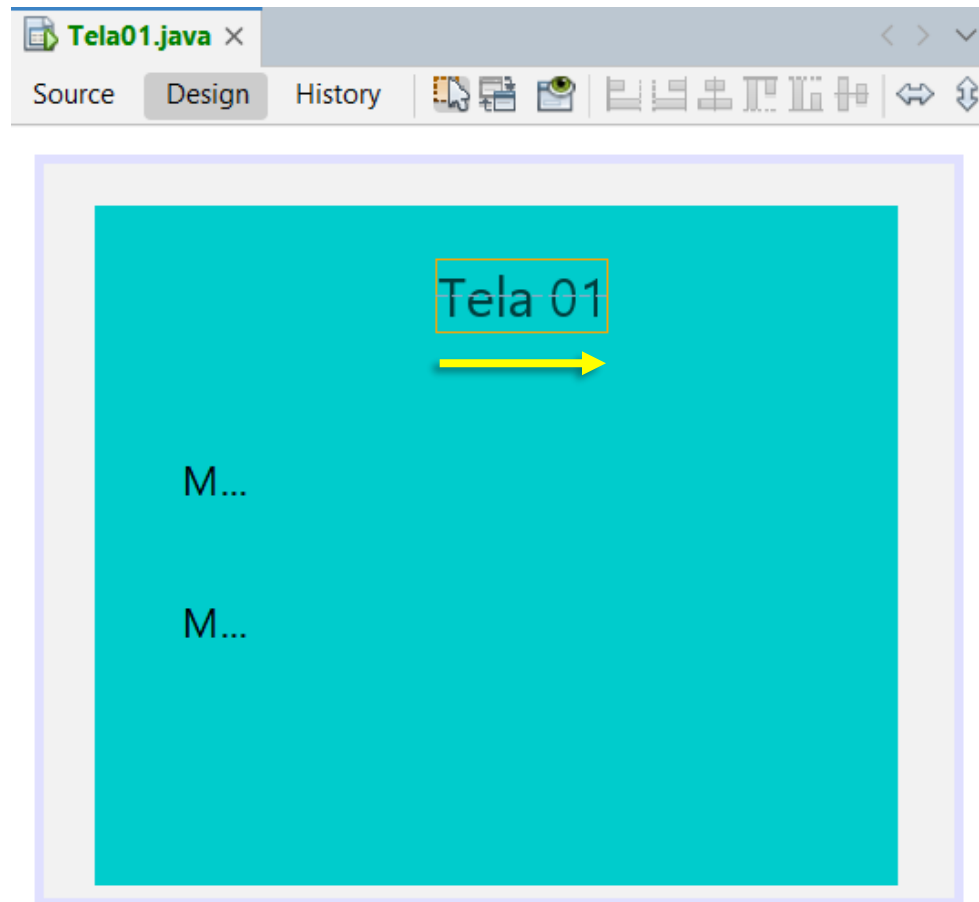
Property	Value
background	[242,242,242]
displayedMnemonic	
font	Segoe UI 12 Plain
foreground	[0,0,0]
horizontalAlignment	LEADING
icon	<none>



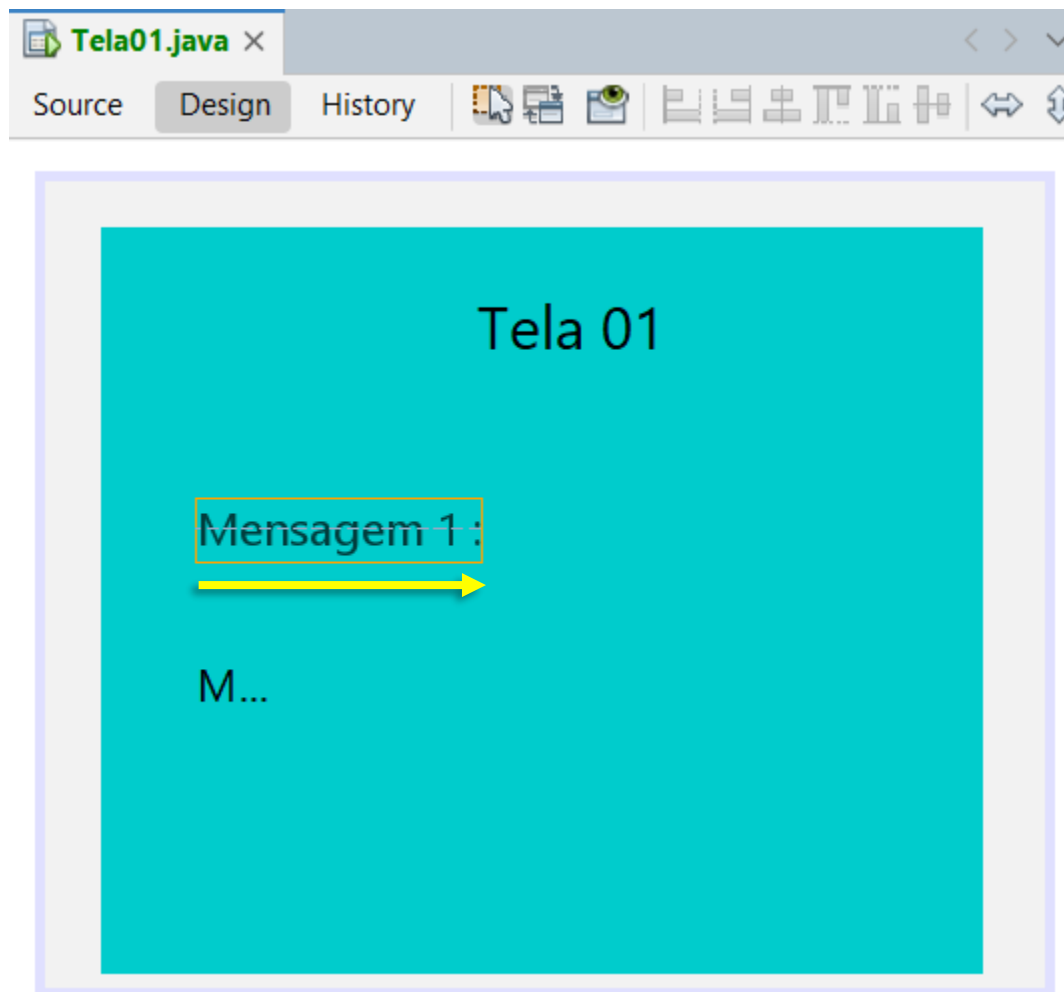
Java Swing – JLabel – Aumentar Fonte



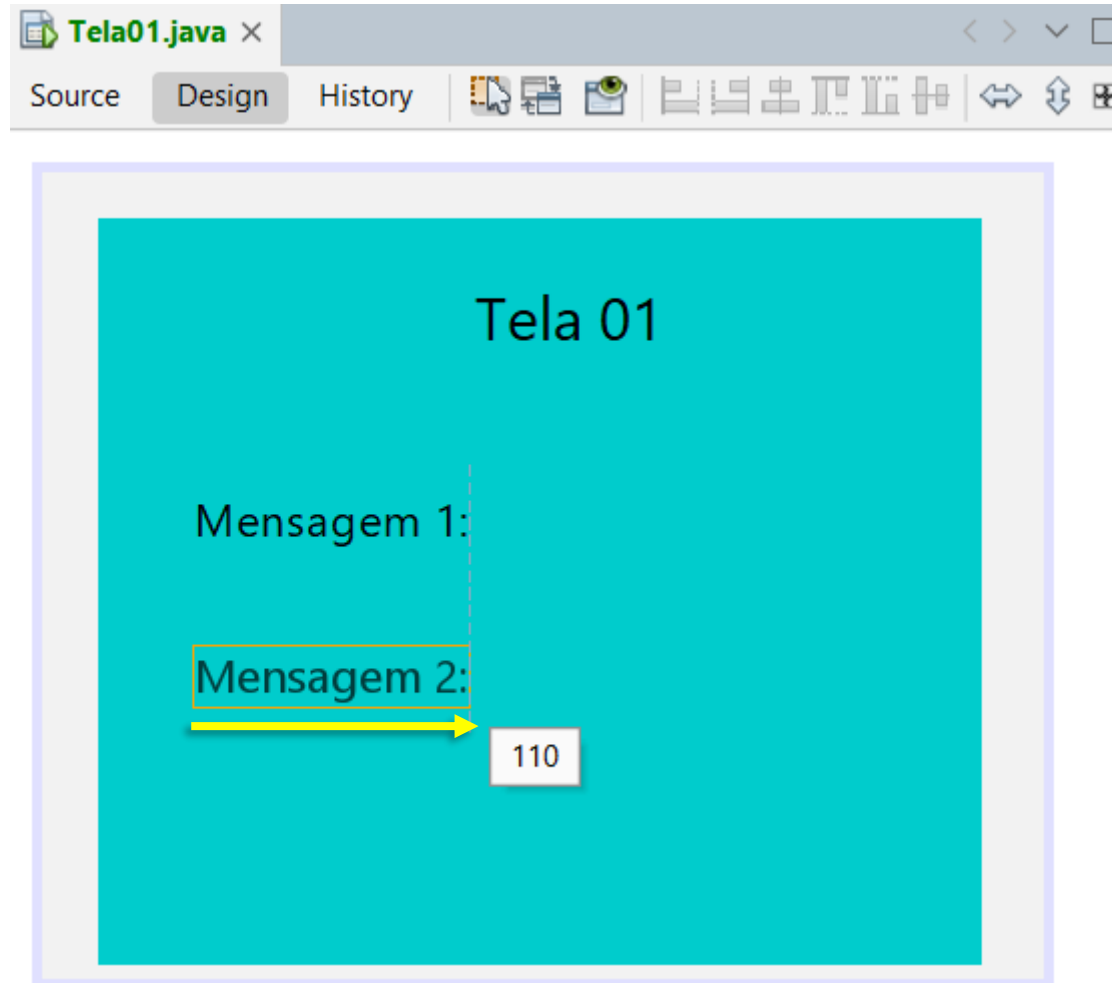
Java Swing – JLabel – Redimensionar



Java Swing – JLabel – Redimensionar



Java Swing – JLabel – Redimensionar

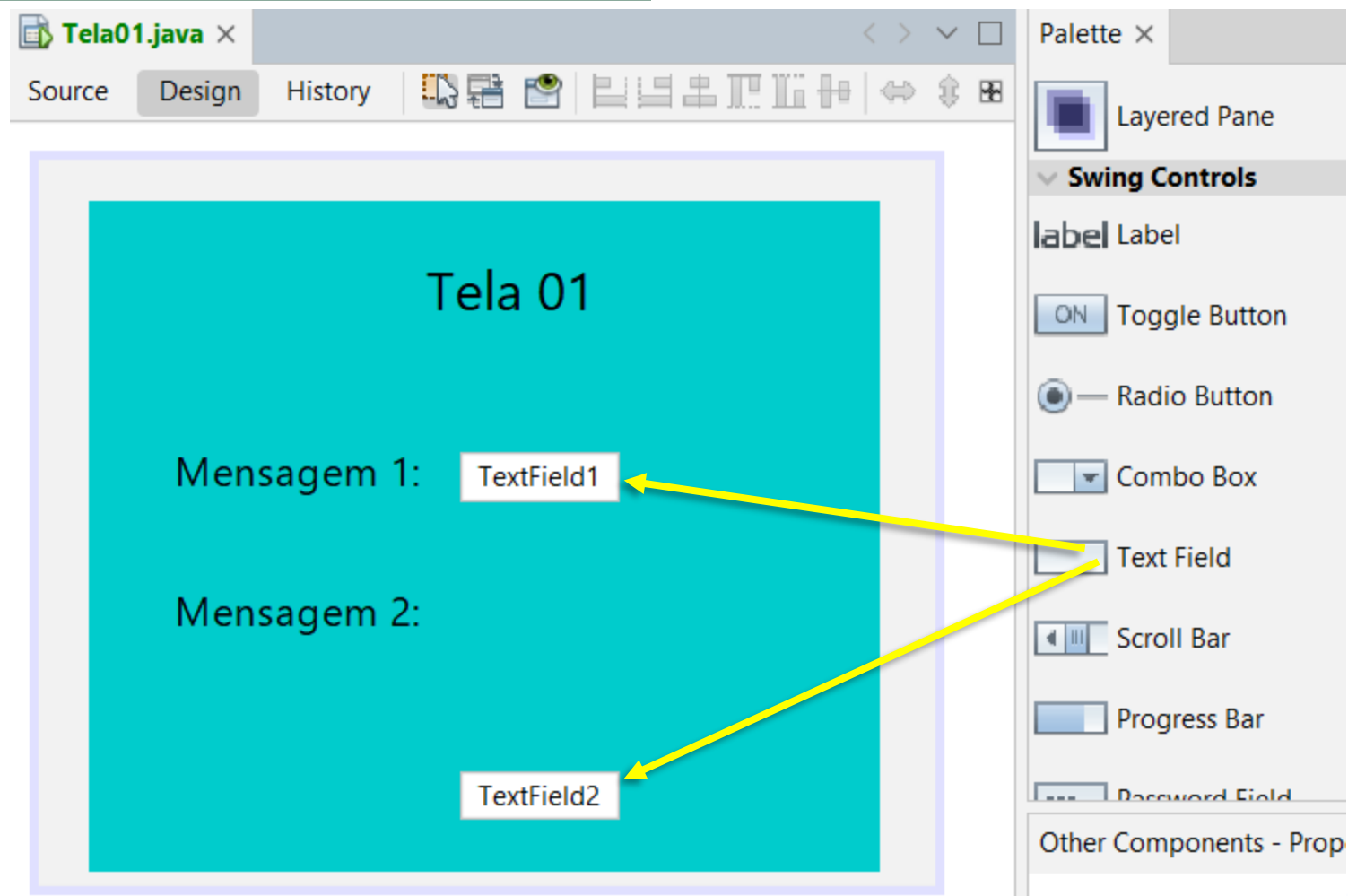


Java Swing – JTextField

- A classe **JTextField** permite criar uma **caixa de texto gráfica** na qual o usuário pode **digitar dados**.
- Existem **diversas propriedades** cujos **conteúdos podem ser modificados**.



Java Swing – JTextField – Adicionar



Java Swing – JTextField – Editor Texto

The screenshot displays an IDE window titled 'Tela01.java' with tabs for Source, Design, and History. The Design view shows a window titled 'Tela 01' with a cyan background. Inside, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. A text field, labeled 'TextField2' at its bottom, is positioned to the right of 'Mensagem 1:'. A yellow arrow points from the text field to the 'jTextField1 [JTextField] - Properties' panel on the right.

The 'jTextField1 [JTextField] - Properties' panel has three tabs: Properties, Events, and Code. The Properties tab is active, showing the following properties:

Property	Value	Action
foreground	[0,0,0]	...
horizontalAlignment	LEADING	...
text		...
toolTipText		...
Other Properties		
UI	<default>	...



Java Swing – JTextField – Editor Texto

The image shows a Java Swing IDE interface. The main window, titled 'Tela 01', has a cyan background. It contains two labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text field. The text fields are white with a gray border. A yellow arrow points from the 'text' property in the 'jTextField2 [JTextField] - Properties' panel to the second text field in the design view.

Palette

- Layered Pane
- Swing Controls**
 - Label
 - Toggle Button
 - Radio Button
 - Combo Box
 - Button
 - Check Box
 - Button Group
 - List

jTextField2 [JTextField] - Properties

Properties	Events	Code
foreground		[0,0,0]
horizontalAlignment		LEADING
text		
toolTipText		
Other Properties		
UI		<default>



Java Swing – JTextField – Renomear Var.

Properties Events Code

Code Generation	
Bean Class	class javax.swing.JTextField
Variable Name	txtMensagem1
Variable Modifiers	private
Type Parameters	
Use Local Variable	<input type="checkbox"/>



Java Swing – JTextField – Renomear Var.

The image shows a screenshot of an IDE interface. On the left, a window titled 'Tela01.java' is in 'Design' mode. It displays a window titled 'Tela 01' with a cyan background. Inside, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. Below 'Mensagem 1:', there is a text field. A yellow arrow points from this text field to the 'Code' tab of the 'jTextField2 [JTextField] - Properties' panel on the right. The 'Code' tab shows the following properties:

Code Generation	
Bean Class	class javax.swing.JTextField
Variable Name	txtMensagemFinal
Variable Modifiers	private
Type Parameters	
Use Local Variable	<input type="checkbox"/>

The 'Swing Controls' palette on the right lists various UI components: Layered Pane, Label, Button, Toggle Button, Check Box, Radio Button, Button Group, Combo Box, and List.



Java Swing – JTextField – Aument. Fonte

The image shows two windows from a Java Swing IDE. The left window is titled 'txtMensagem1 [JTextField] - font' and is used to configure the font for the selected component. It includes a 'Set txtMensagem1's font property using:' dropdown set to 'Default editor'. Below this, there is a checkbox 'Derive the font from the default font' which is unchecked. The 'Font' section shows 'Segoe UI' selected, 'Plain' for 'Font Style', and '18' for 'Size'. A 'Preview' section shows the text 'The quick brown fox jumps over the lazy dog' in the selected font. At the bottom are 'OK', 'Cancel', and 'Help' buttons. A yellow arrow points from the 'font' property in the right window to the 'font' dialog box. The right window is titled 'txtMensagem1 [JTextField] - Properties' and shows a list of properties. The 'font' property is selected, showing 'Segoe UI 12 Plain' as the current value. A yellow arrow points from the 'font' property in the right window to the 'font' property in the Properties window. A third yellow arrow points from the 'font' property in the Properties window to the 'font' property in the Properties window.

Font configuration dialog for txtMensagem1 [JTextField]:

- Set txtMensagem1's font property using: Default editor
- ☐ Derive the font from the default font
- Font: Segoe UI
- Font Style: Plain
- Size: 18
- Preview: The quick brown fox jumps over the lazy dog
- Buttons: OK, Cancel, Help

Properties window for txtMensagem1 [JTextField]:

Properties	Events	Code
columns		0
document		<default>
font		Segoe UI 12 Plain
foreground		■ [0.0.0]
horizontalAlignment		LEADING
text		

font (java.awt.Font) The font for the component.



Java Swing – JTextField – Aument. Fonte

The image shows two overlapping windows from a Java Swing IDE. The foreground window is titled "txtMensagemFinal [JTextField] - font" and contains settings for the font of the text field. It has a "Set txtMensagemFinal's font property using:" dropdown set to "Default editor". Below this, there are three input fields: "Font:" (Segoe UI), "Font Style:" (Plain), and "Size:" (18). A yellow arrow points from the "Size:" field to the "font" property in the background window. The background window is titled "txtMensagemFinal [JTextField] - Properties" and shows a table of properties. The "font" property is highlighted in blue, and its value is "Segoe UI 12 Plain". A yellow arrow points from the "font" property in the background window to the "font" property in the foreground window. The foreground window also has a "Preview" section showing the text "The quick brown fox jumps over the lazy dog" and buttons for "OK", "Cancel", and "Help".

txtMensagemFinal [JTextField] - font

Set **txtMensagemFinal's font** property using: Default editor

☐ Derive the font from the default font

Font: Segoe UI Font Style: Plain Size: 18

Preview
The quick brown fox jumps over the lazy dog

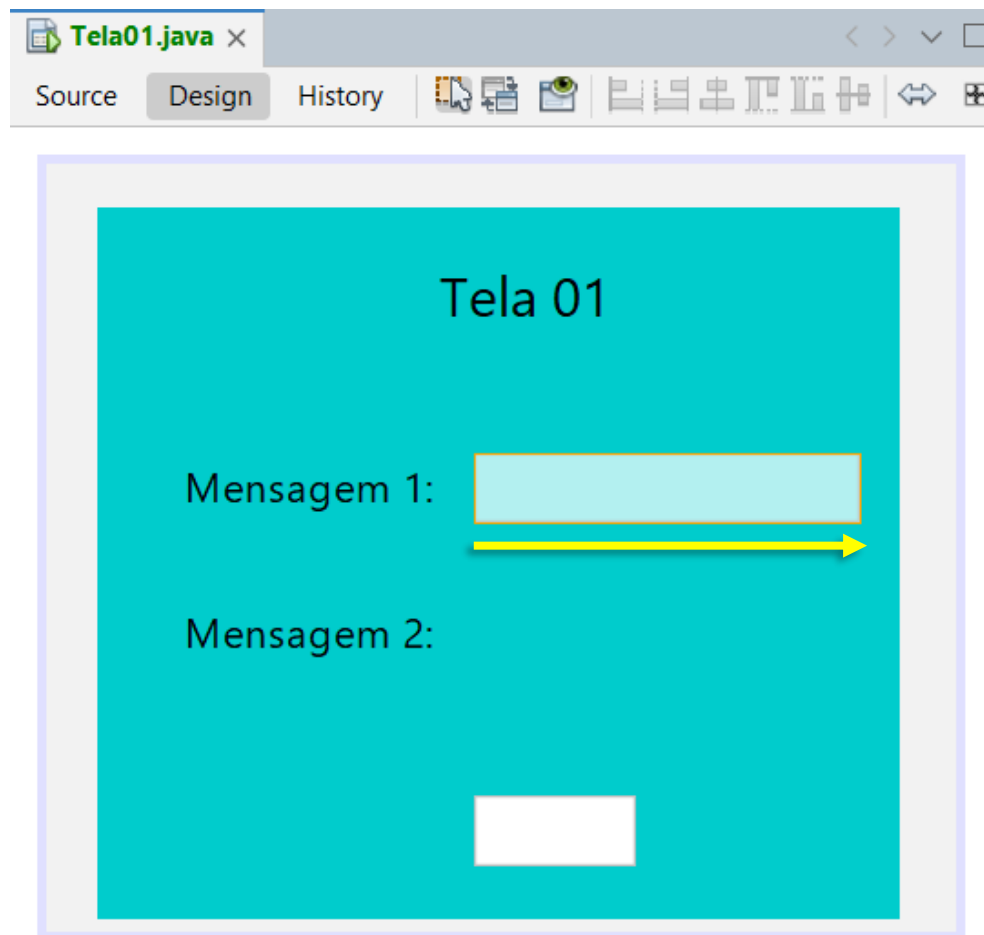
OK Cancel Help

txtMensagemFinal [JTextField] - Properties

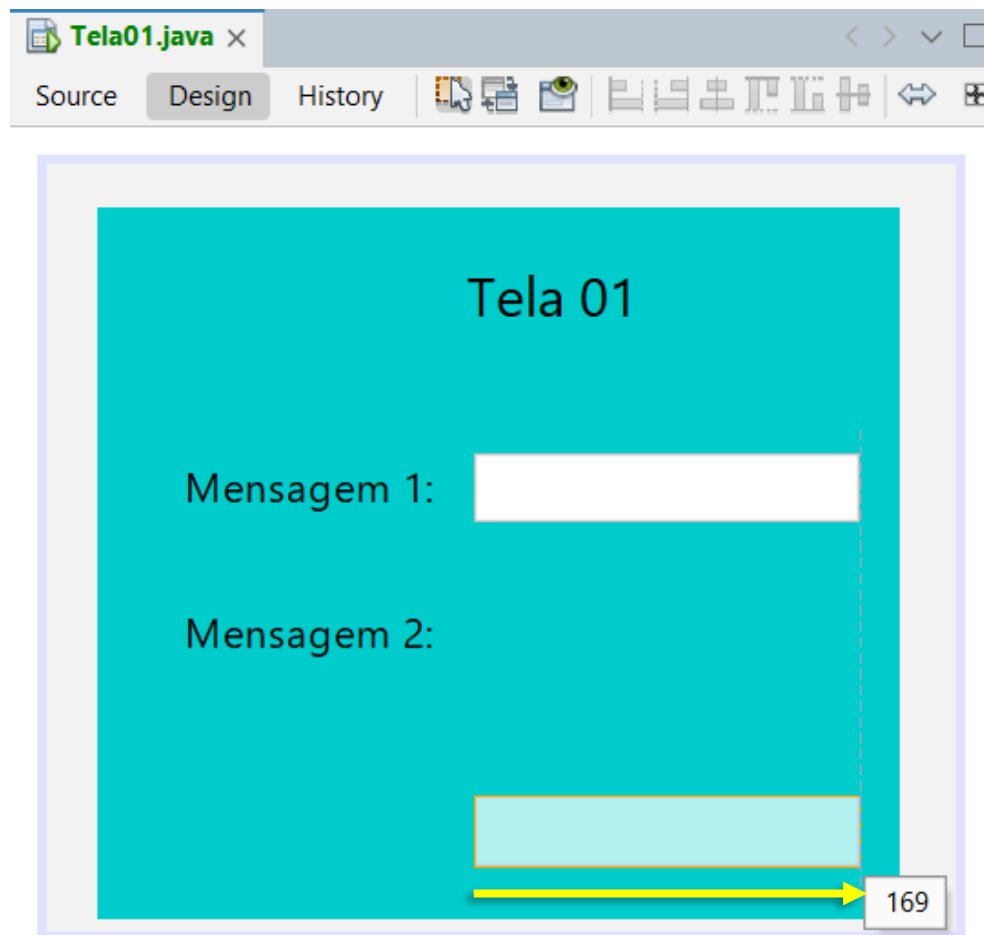
Properties	Events	Code
columns		0
document		<default>
font		Segoe UI 12 Plain
foreground		[0,0,0]
horizontalAlignment		LEADING
text		



Java Swing – JTextField – Redimens.



Java Swing – JTextField – Redimens.



Java Swing – JTextField – Não Editável

The screenshot shows an IDE window titled 'Tela01.java' with tabs for Source, Design, and History. The Design view displays a window titled 'Tela 01' with a cyan background. It contains two labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text field. The text field for 'Mensagem 2:' is highlighted with a yellow box. A yellow arrow points from this text field to the 'editable' property in the Properties panel, which is set to false.

Palette ×

- Radio Button
- Button Group
- Combo Box
- List
- Text Field
- Text Area
- Scroll Bar
- Slider
- Progress Bar
- Formatted Field
- Password Field
- Spinner
- Separator
- Text Pane

txtMensagemFinal [JTextField] - Properties ×

Properties	Events	Code
editable		<input type="checkbox"/>
background		<input type="checkbox"/> [242,242,242]
columns		0

editable

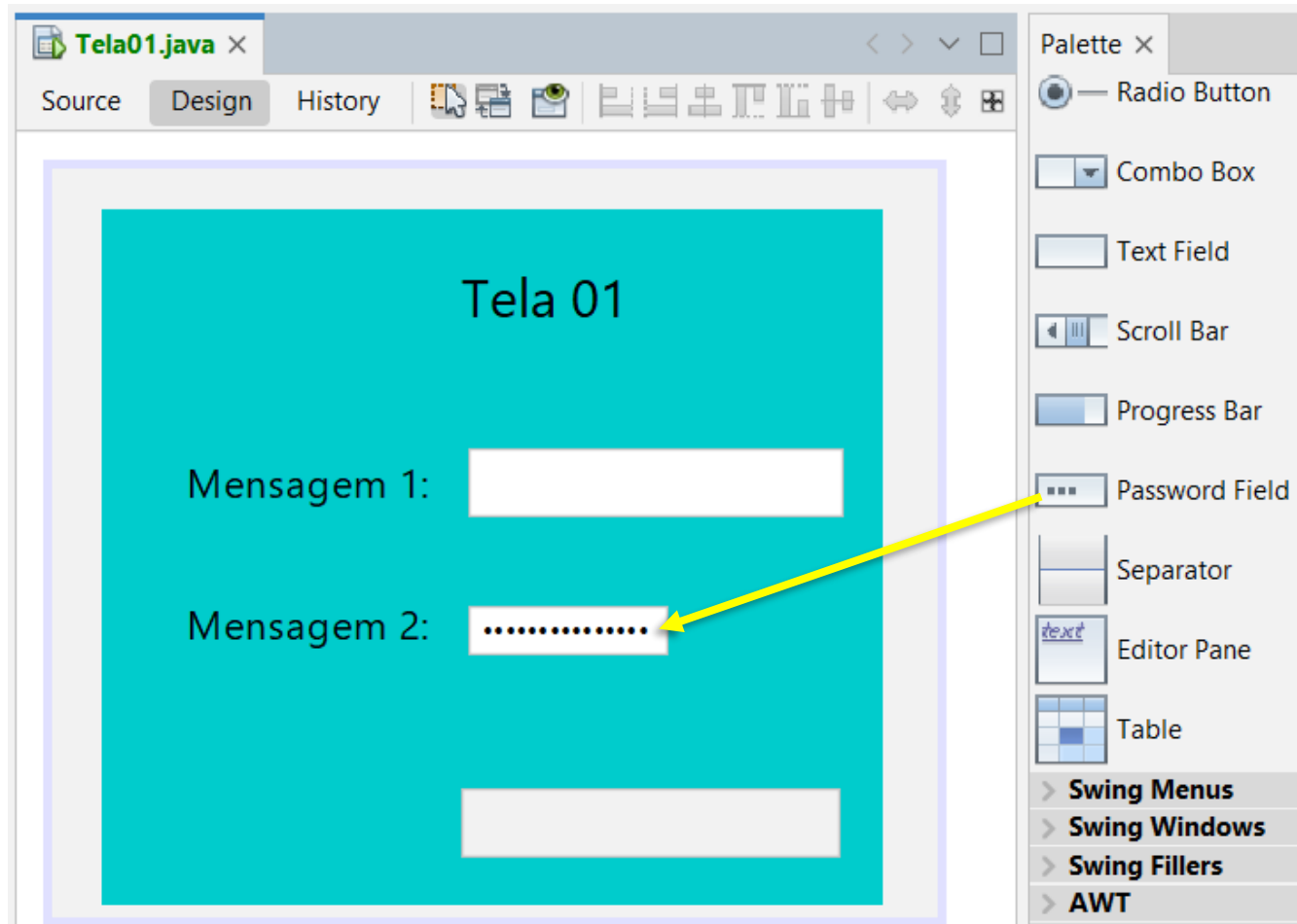


Java Swing – JPasswordField

- A classe **JPasswordField** permite criar uma **caixa de texto gráfica** na qual o usuário pode **digitar dados**, porém eles são substituídos (visualmente) por outro caractere (*).



Java Swing – JPasswordField – Adic.



Java Swing – JPasswordField – Editor T.

The screenshot displays the NetBeans IDE interface for a Java Swing application. The main window, titled 'Tela 01', is in the 'Design' view. It features a cyan background with two labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text field. The second text field is a JPasswordField, highlighted with a yellow box and a yellow arrow pointing to its properties. The 'Properties' tab of the 'jPasswordField1 [JPasswordField] - Properties' window is open, showing the 'text' property set to an empty string. The 'Palette' window on the right shows various Swing components, including the JPasswordField.

Palette

- Radio Button
- Button Group
- Combo Box
- List
- Text Field
- Text Area
- Scroll Bar
- Slider
- Progress Bar
- Formatted Field
- Password Field
- Spinner
- Separator
- Text Pane

jPasswordField1 [JPasswordField] - Properties

Properties	Events	Code
foreground		[0,0,0]
horizontalAlignment		LEADING
text		
toolTipText		

jPasswordField1 [JPasswordField]



Java Swing – JPasswordField – Ren. Var.

The screenshot shows an IDE window titled 'Tela01.java' with tabs for Source, Design, and History. The Design view displays a window titled 'Tela 01' with a cyan background. It contains two password fields, 'Mensagem 1:' and 'Mensagem 2:'. A yellow arrow points from the 'Mensagem 2:' password field to the 'txpMensagem2' property in the Properties window.

Palette

- Radio Button
- Button Group
- Combo Box
- List
- Text Field
- Text Area
- Scroll Bar
- Slider
- Progress Bar
- Formatted Field
- Password Field
- Spinner
- Separator
- Text Pane

txpMensagem2 [JPasswordField] - Properties

Properties	Events	Code
Code Generation		
Bean Class	class javax.swing.JPassword...	
Variable Name	txpMensagem2	
Variable Modifiers	private	

txpMensagem2 [JPasswordField]



Java Swing – JPasswordField – Aum. F.

The screenshot displays the Java Swing IDE interface. A dialog box titled "txpMensagem2 [JPasswordField] - font" is open, showing the font configuration for the selected component. The dialog has a "Font" section with "Segoe UI" selected, a "Font Style" section with "Plain" selected, and a "Size" section with "18" selected. The "OK" button is highlighted with a yellow arrow. Below the dialog, a preview of the component shows the text "Mensagem 2:" and a password field. A yellow arrow points from the "OK" button to the "font" property in the "txpMensagem2 [JPasswordField] - Properties" window. The "font" property is set to "Segoe UI 12 Plain".

Font configuration details:

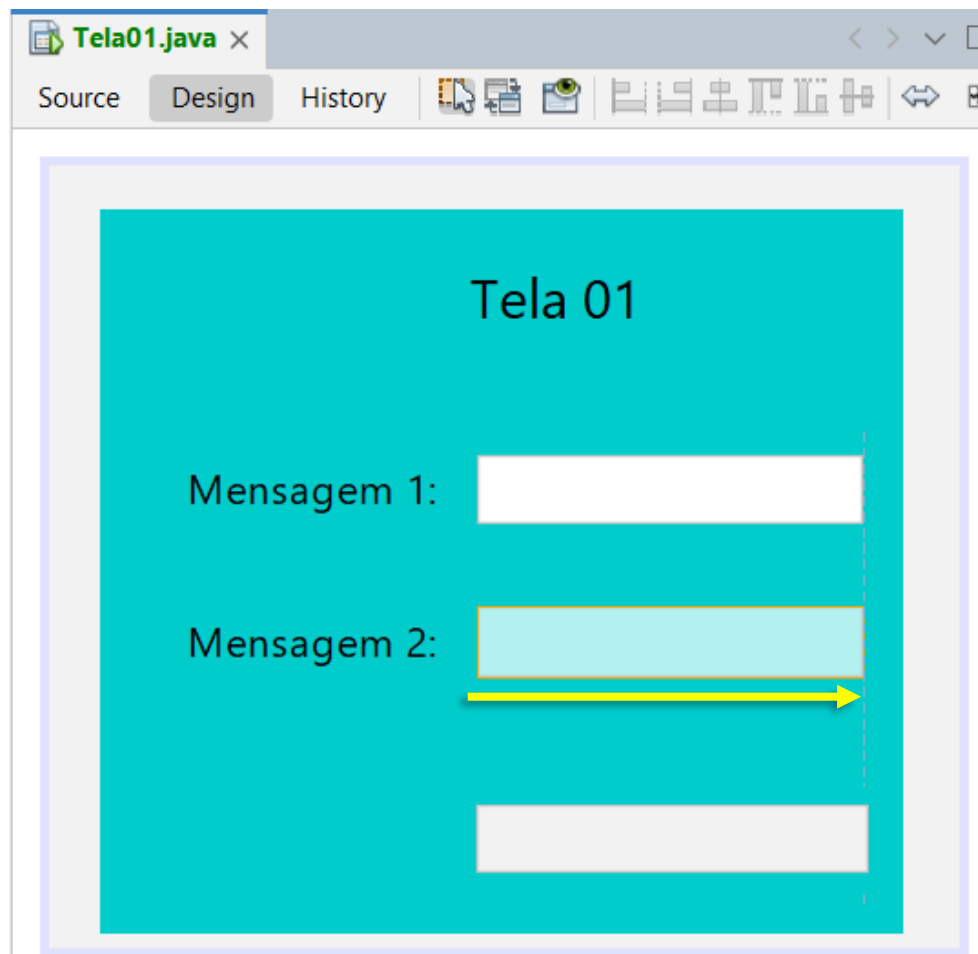
Font	Font Style	Size
Segoe UI	Plain	18

Properties window details:

Property	Value
columns	0
document	<default>
font	Segoe UI 12 Plain
foreground	[0,0,0]



Java Swing – JPasswordField – Redimen.

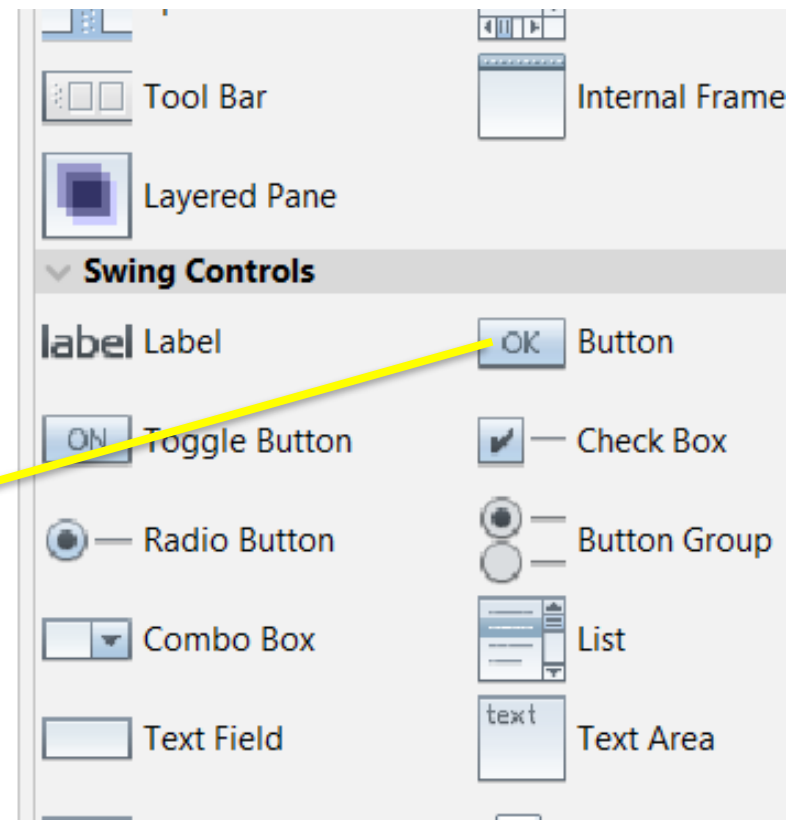
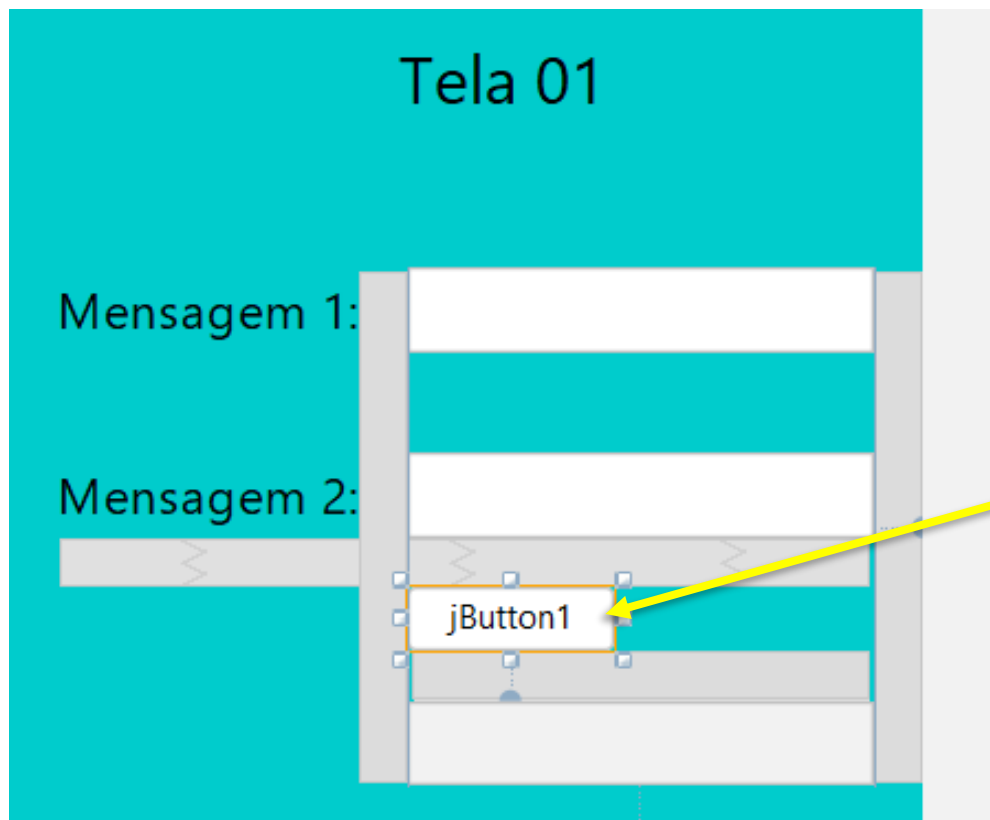


Java Swing – JButton

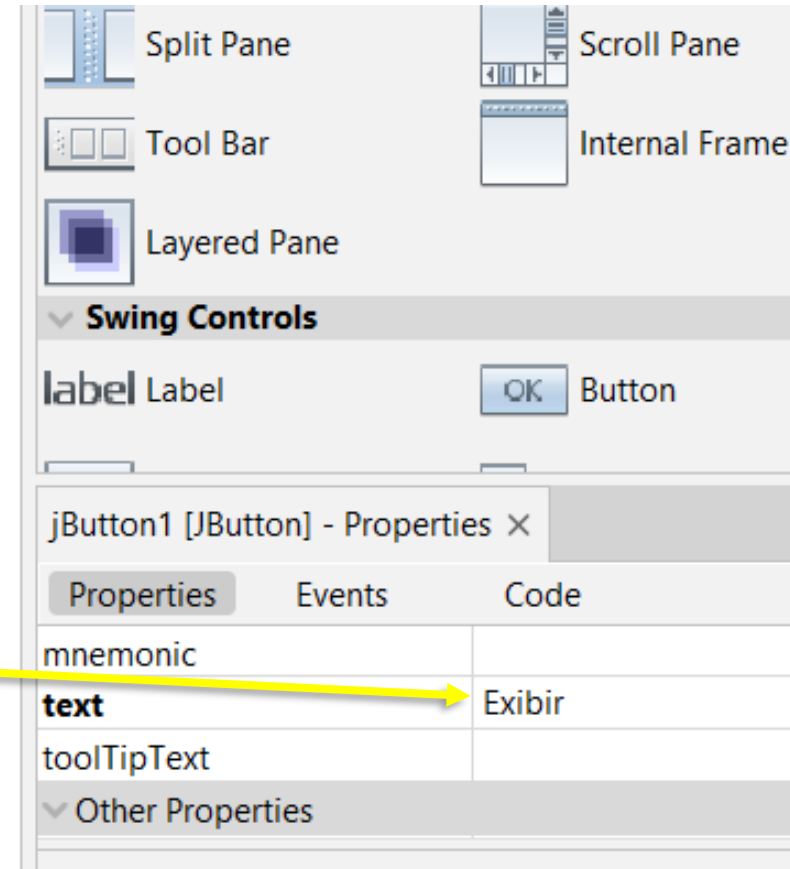
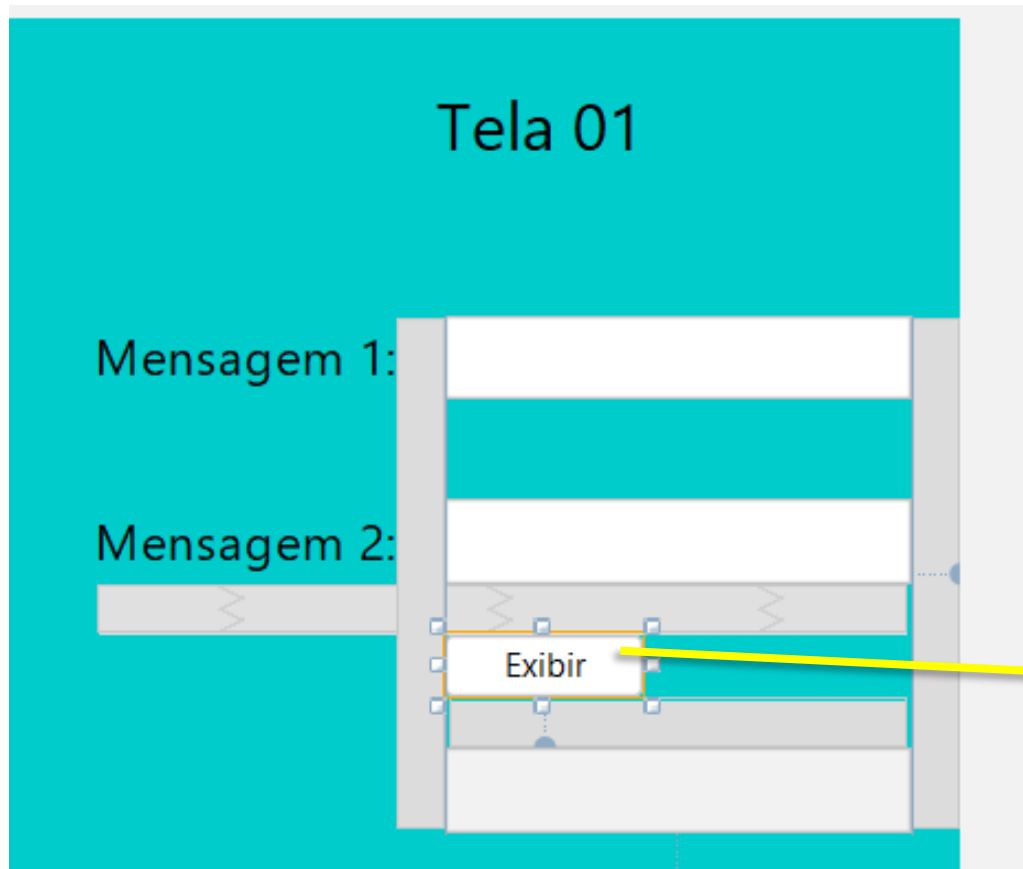
- A classe **JButton** permite **definir botões gráficos** que **pode ser adicionados** a um **outro componente** (frame, painel etc.).
- Um **botão** pode **ser criado** com apenas um **texto e/ou com ícones** para tornar o ambiente mais intuitivo.



Java Swing – JButton – Adicionar



Java Swing – JButton – Editor Texto



Java Swing – JButton – Aumentar Fonte

The image shows a Java Swing IDE interface with two main windows. The left window, titled 'jButton1 [JButton] - font', is used for configuring the font of a JButton. It includes a 'Set jButton1's font property using:' dropdown set to 'Default editor'. Below this is a checkbox 'Derive the font from the default font' which is unchecked. The 'Font:' field is set to 'Segoe UI', 'Font Style:' is 'Plain', and 'Size:' is '18'. A 'Preview' section shows the text 'The quick brown fox jumps over the lazy dog' in the selected font. At the bottom are 'OK', 'Cancel', and 'Help' buttons. A yellow arrow points from the 'OK' button to the 'font' property in the right window. The right window, titled 'jButton1 [JButton] - Properties', shows a tree view of Swing Containers and Controls. Under 'Swing Controls', the 'font' property is highlighted, showing 'Segoe UI 12 Plain'. A yellow arrow points from the 'font' property in the right window to the 'font' property in the left window's preview section.

Set **jButton1**'s font property using: Default editor

☐ Derive the font from the default font

Font: Segoe UI Font Style: Plain Size: 18

Preview
The quick brown fox jumps over the lazy dog

OK Cancel Help

Swing Containers

- Panel
- Split Pane
- Tool Bar
- Layered Pane
- Tabbed Pane
- Scroll Pane
- Internal Frame

Swing Controls

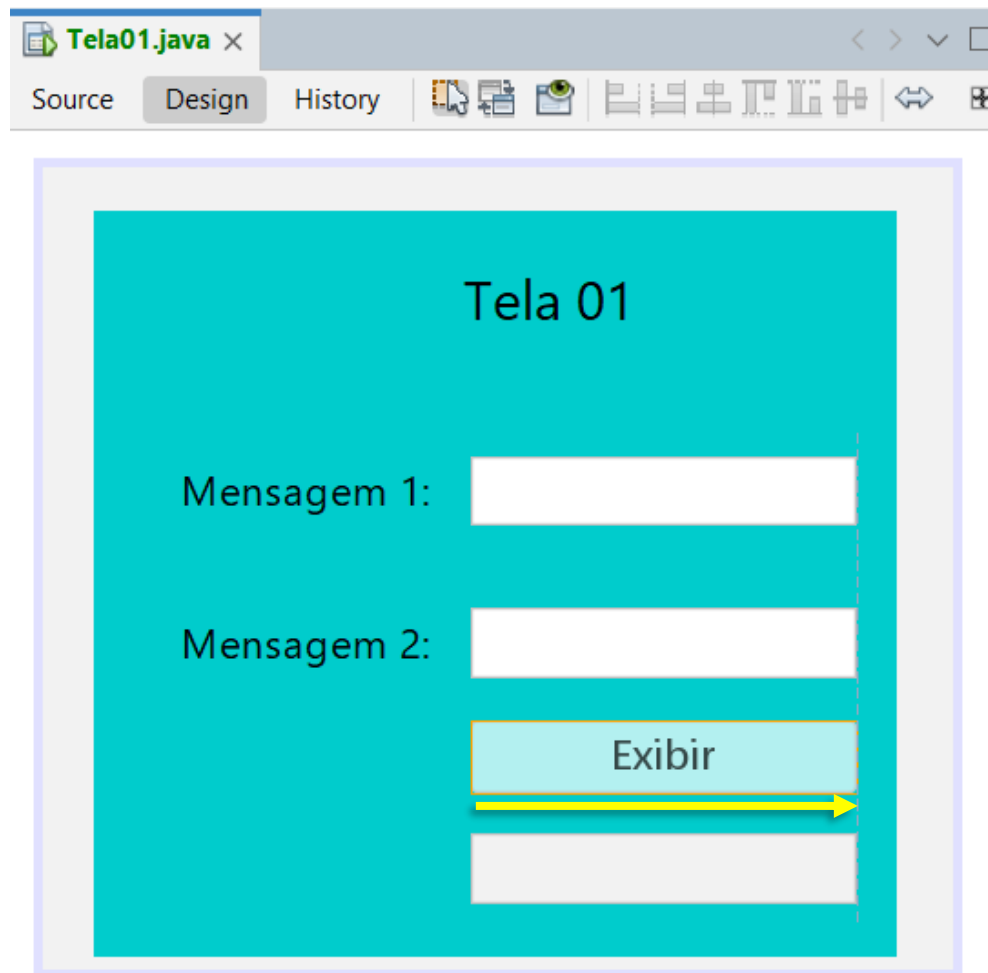
- Label
- Button

jButton1 [JButton] - Properties

Properties	Events	Code
font		Segoe UI 12 Plain
foreground		[0,0,0]
icon		<none>



Java Swing – JButton – Redimensionar



Java Swing – JButton – Renomear Var.

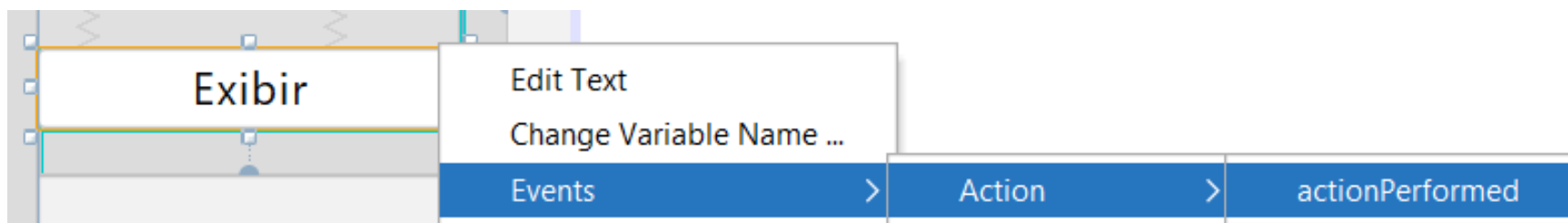
The image shows a Java Swing IDE interface. The main window, titled 'Tela 01', is in Design mode. It contains a cyan background with the text 'Tela 01'. Below this, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. To the right of these labels is a vertical stack of components. The bottom component is a JButton labeled 'Exibir'. A yellow arrow points from the 'Exibir' button to the 'Properties' window.

The 'Properties' window for 'btnExibir [JButton]' is open. It shows the 'Code' tab. The 'Code Generation' section displays the following information:

Property	Value
Bean Class	class javax.swing.JButton
Variable Name	btnExibir
Variable Modifiers	private



Java Swing – JButton – Adicionar Ação



Java Swing – JButton – Adicionar Ação

```
private void btnExibirActionPerformed(java.awt.event.ActionEvent evt) {
```

}



Java Swing – JButton – Adicionar Ação

```
import javax.swing.JOptionPane;
```



Java Swing – JButton – Adicionar Ação

```
private void btnExibirActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String mensagem1 = txtMensagem1.getText();  
    String mensagem2 = String.valueOf(txpMensagem2.getPassword());  
  
    String mensagemFinal = mensagem1 + " " + mensagem2;  
  
    txtMensagemFinal.setText(mensagemFinal);  
  
    JOptionPane.showMessageDialog(this, mensagemFinal);  
  
}
```



Java Swing – JButton – Execução



Sistema IFBA

Tela 01

Mensagem 1:

Mensagem 2:



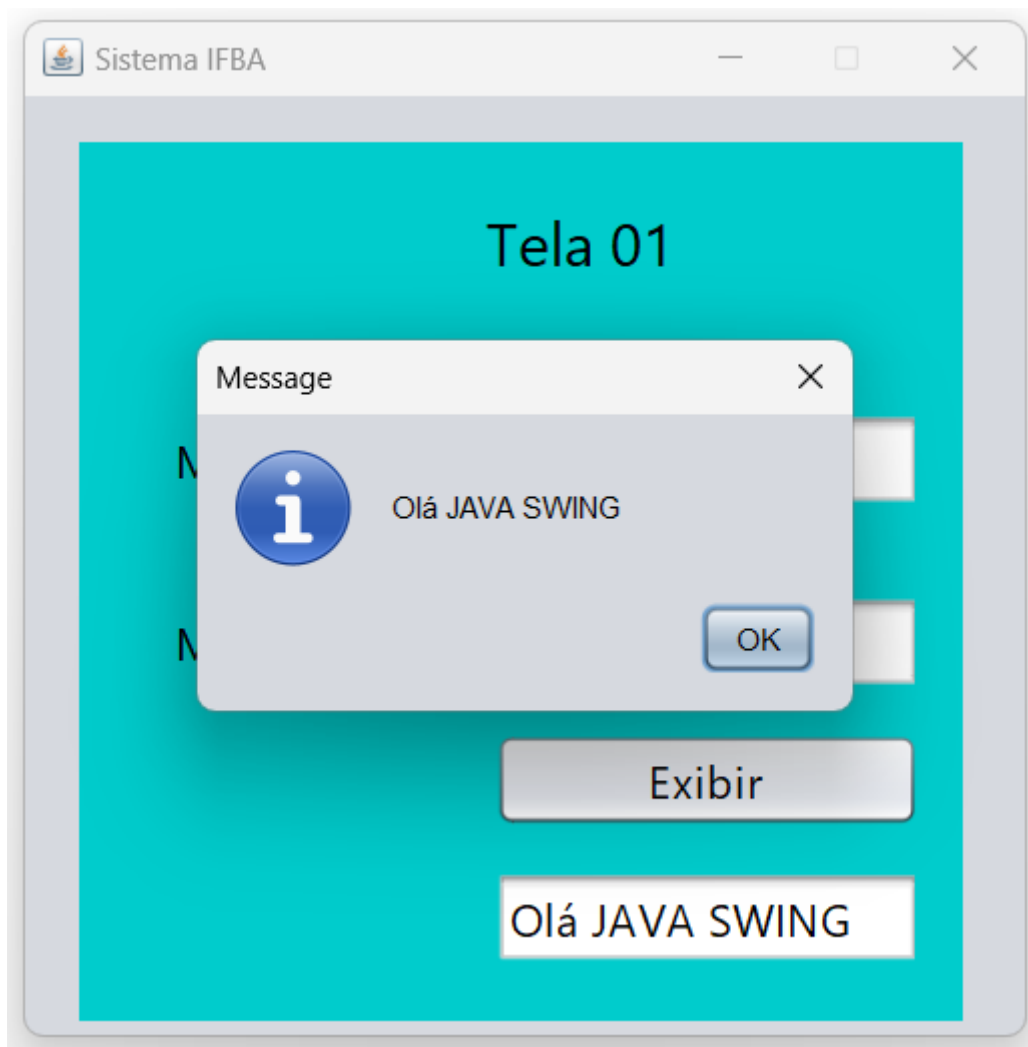
Java Swing – JButton – Execução



The screenshot shows a Java Swing window titled "Sistema IFBA". The window has a light gray title bar with standard minimize, maximize, and close buttons. The main content area has a cyan background. At the top center, the text "Tela 01" is displayed. Below this, there are two text input fields. The first field is labeled "Mensagem 1:" and contains the text "Olá". The second field is labeled "Mensagem 2:" and contains a series of asterisks "*****". Below the second field is a button labeled "Exibir". At the bottom of the cyan area, there is an empty text input field.



Java Swing – JButton – Execução



Java Swing – JButton – Execução



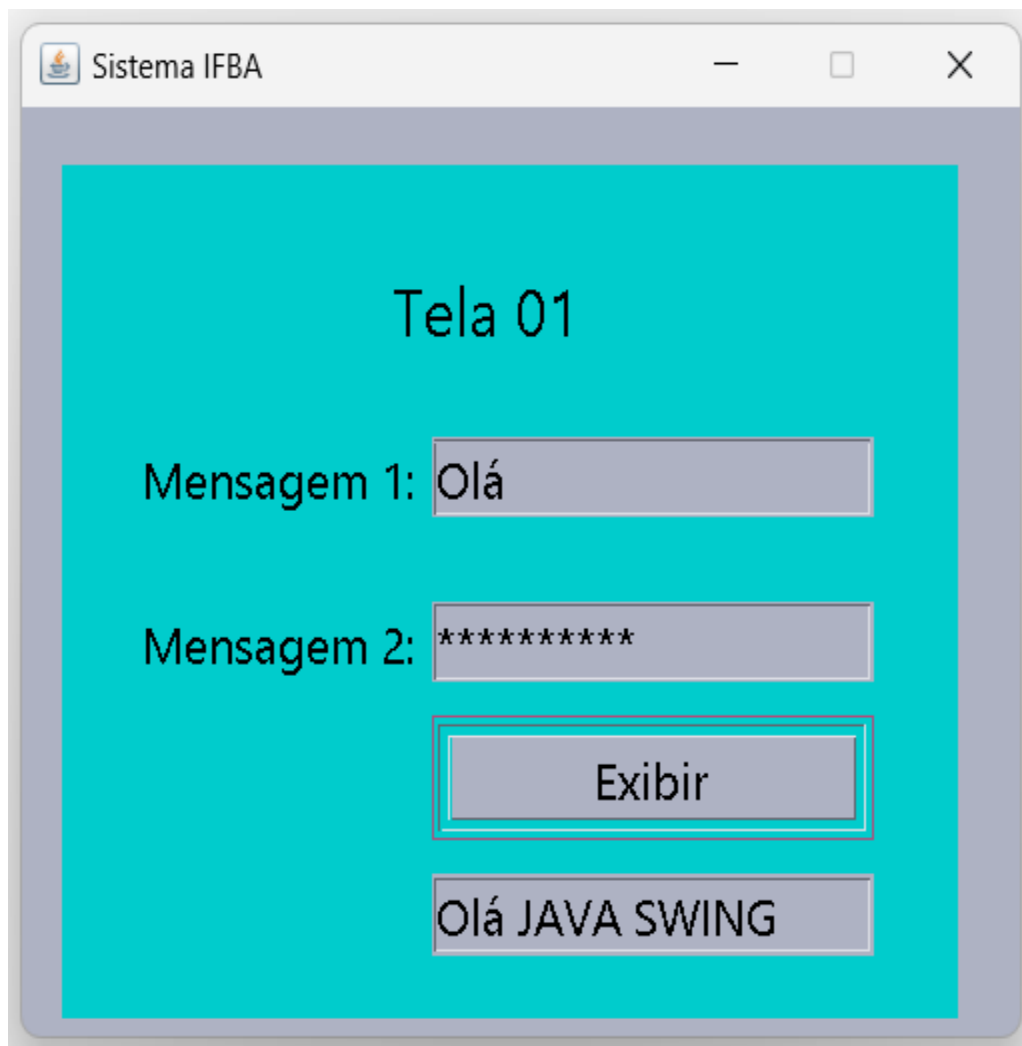
Java Swing – Troca de Tema (Aparência)

```
if ("Nimbus".equals(info.getName())) {  
    javax.swing.UIManager.setLookAndFeel(info.getClassName());  
    break;  
}
```

```
if ("CDE/Motif".equals(info.getName())) {  
    javax.swing.UIManager.setLookAndFeel(info.getClassName());  
    break;  
}
```



Java Swing – Troca de Tema (Aparência)

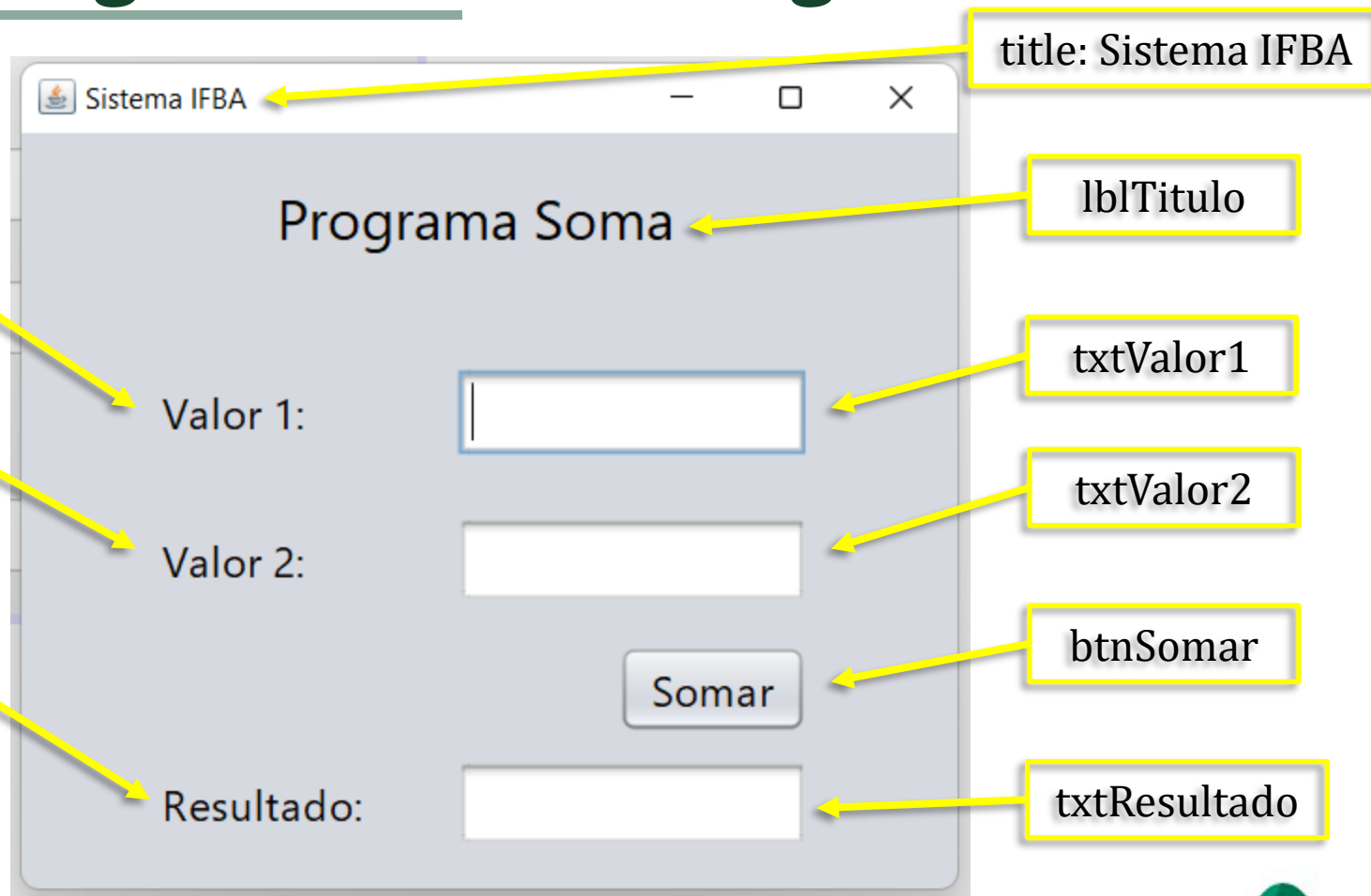


Java Swing – Lista de Tema (Aparência)

1. Nimbus
2. Metal
3. CDE/Motif (também conhecido como Motif)
4. Windows
5. Windows Classic
6. GTK (disponível apenas em sistemas Linux)
7. Aqua (disponível apenas em sistemas macOS)
8. Synth
9. Mac
10. Mac OS X
11. Ocean
12. Plastic
13. System



Java Swing – Exercício – Prog. Soma



TelaSoma.java



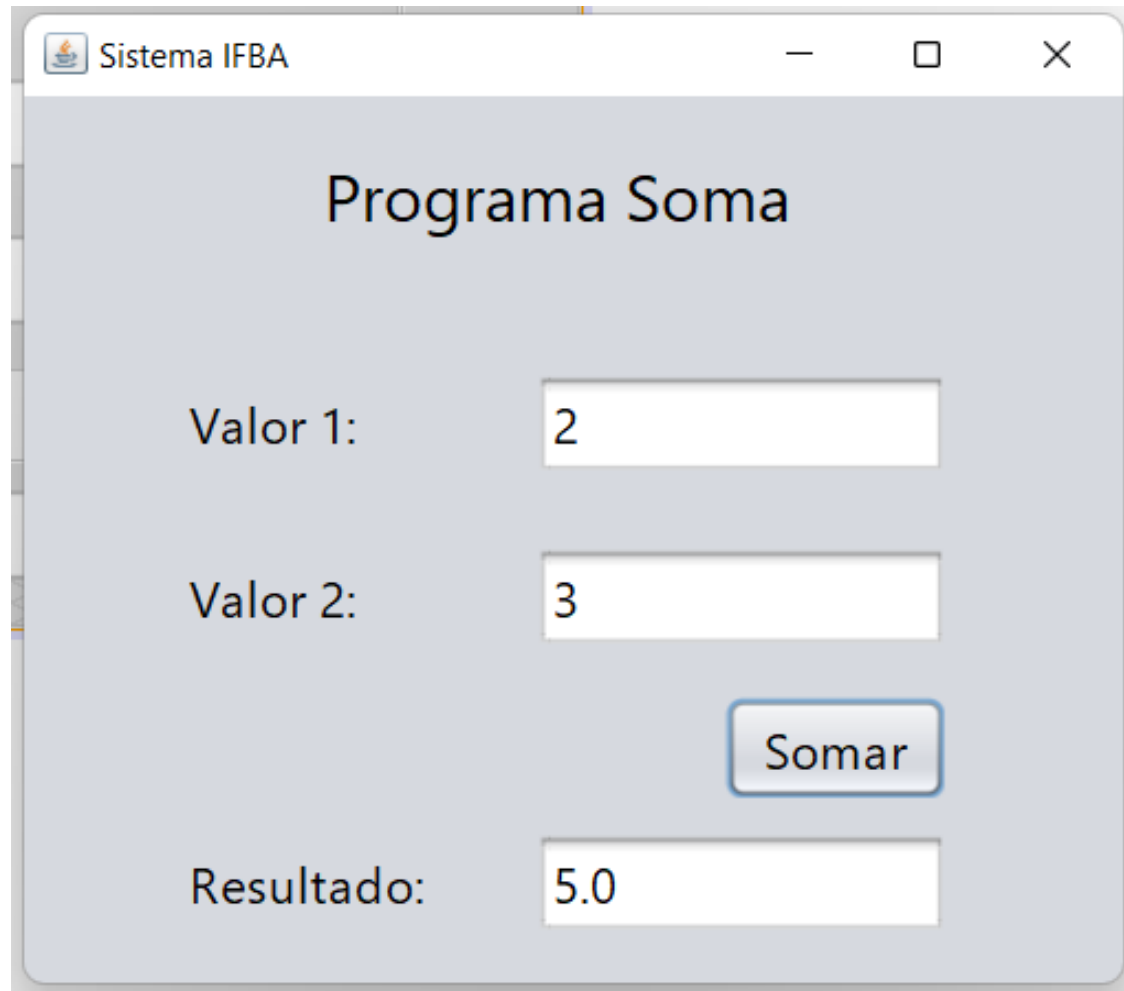
Java Swing – Exercício – Prog. Soma

```
private void btnSomarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    double valor1 = Double.parseDouble(txtValor1.getText());  
  
    double valor2 = Double.parseDouble(txtValor2.getText());  
  
    double resultado = valor1 + valor2;  
  
    String strResultado = String.valueOf(resultado);  
  
    txtResultado.setText(strResultado);  
  
}
```



Java Swing – Exercício – Prog. Soma

Execução



Sistema IFBA

Programa Soma

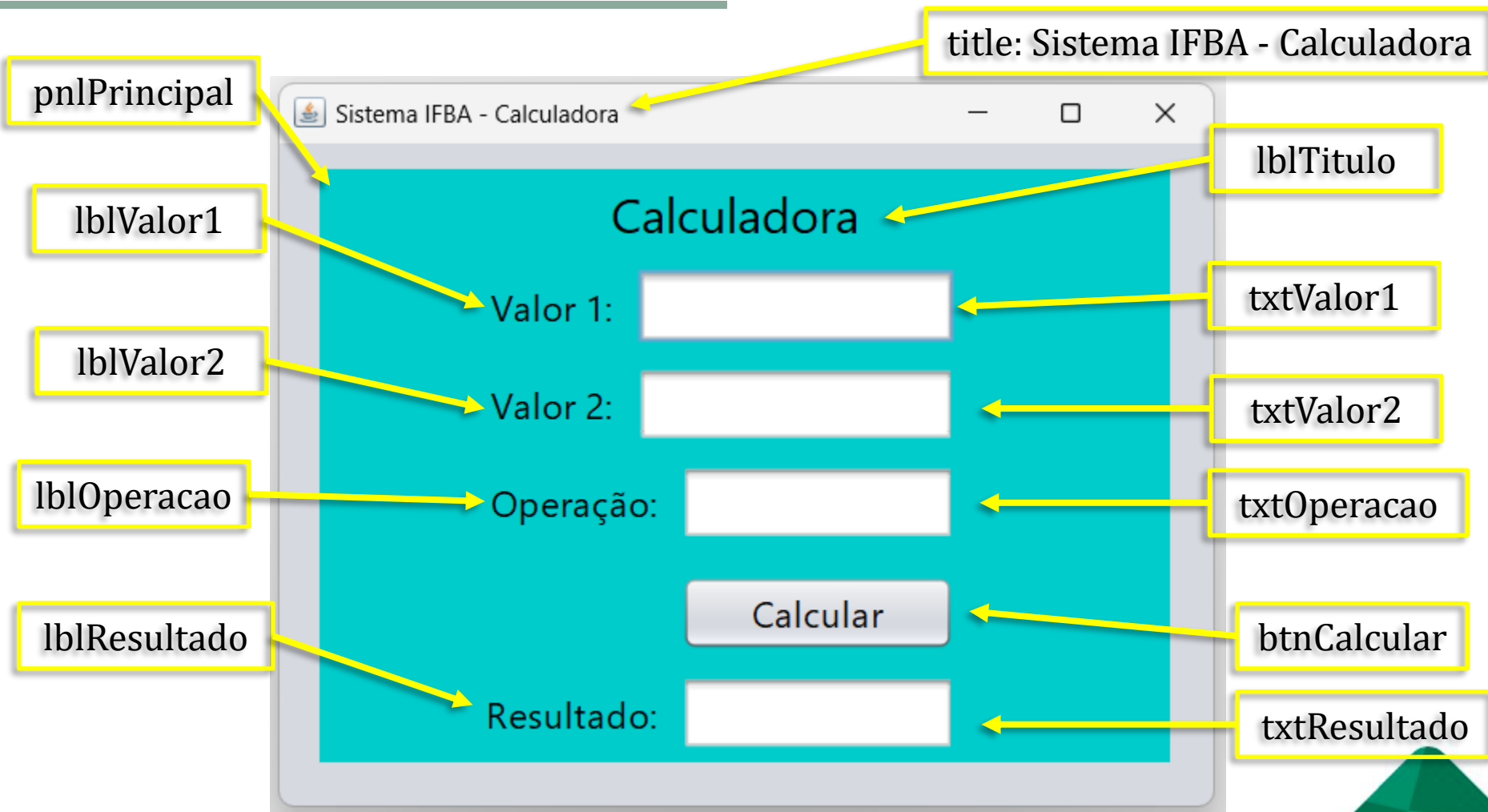
Valor 1:

Valor 2:

Resultado:



Java Swing – Exercício – Prog. Calc.



TelaCalculadora



Java Swing – Exercício – Prog. Calc.

Qual é o código executado ao acionar o botão “Calcular”?



Java Swing – Exercício – Prog. Calc.

Leia valor1 como um número decimal do campo de texto txtValor1.

Leia valor2 como um número decimal do campo de texto txtValor2.

Leia operacao como uma string do campo de texto txtOperacao.

Declare resultado como um número decimal e inicialize com 0.0.

Se operacao for igual a "+":

atribua a resultado a soma de valor1 e valor2.

Senão, se operacao for igual a "-":

atribua a resultado a diferença entre valor1 e valor2.

Senão, se operacao for igual a "*":

atribua a resultado o produto de valor1 e valor2.

Senão, se operacao for igual a "/":

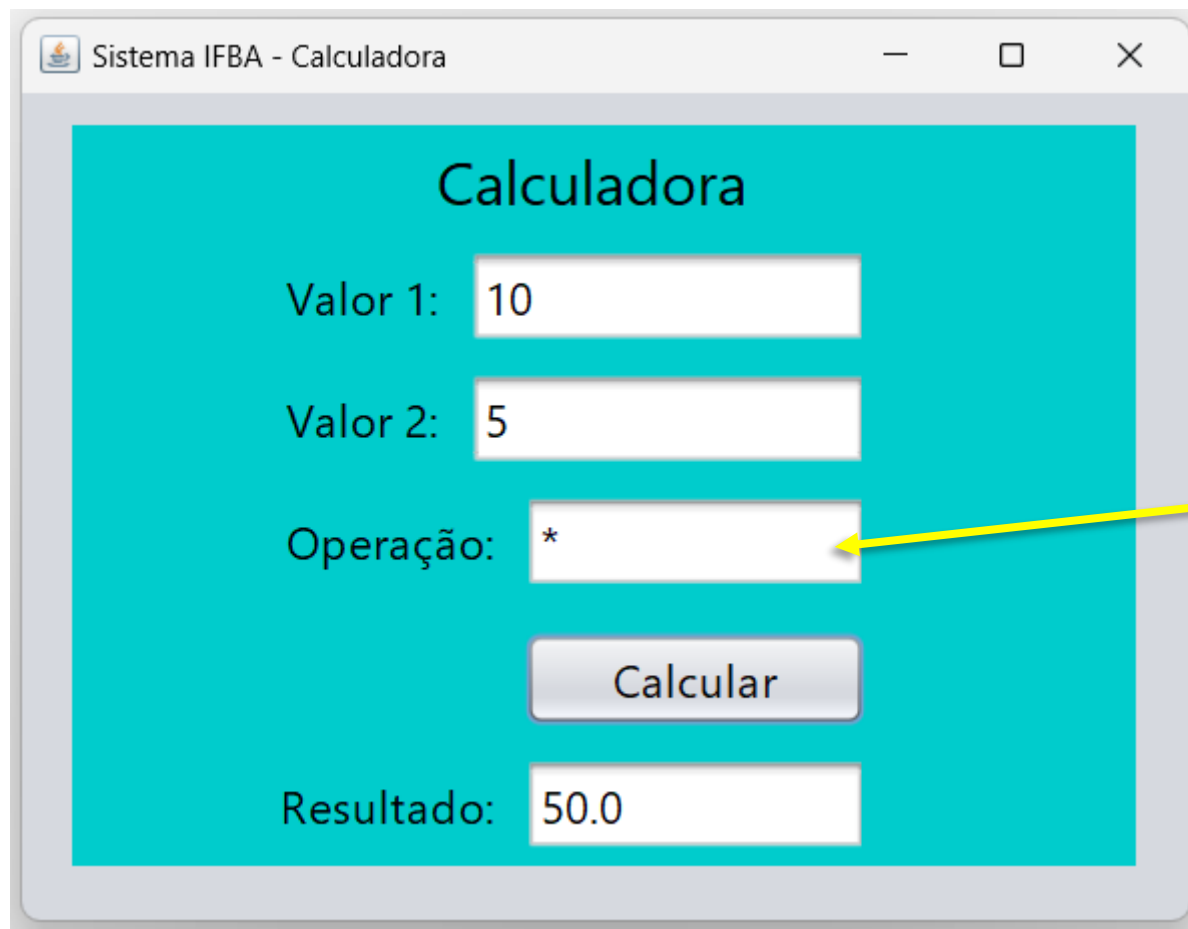
atribua a resultado a divisão de valor1 por valor2.

Converta resultado para uma string e atribua ao campo de texto txtResultado.



Java Swing – Exercício – Prog. Calc.

Execução



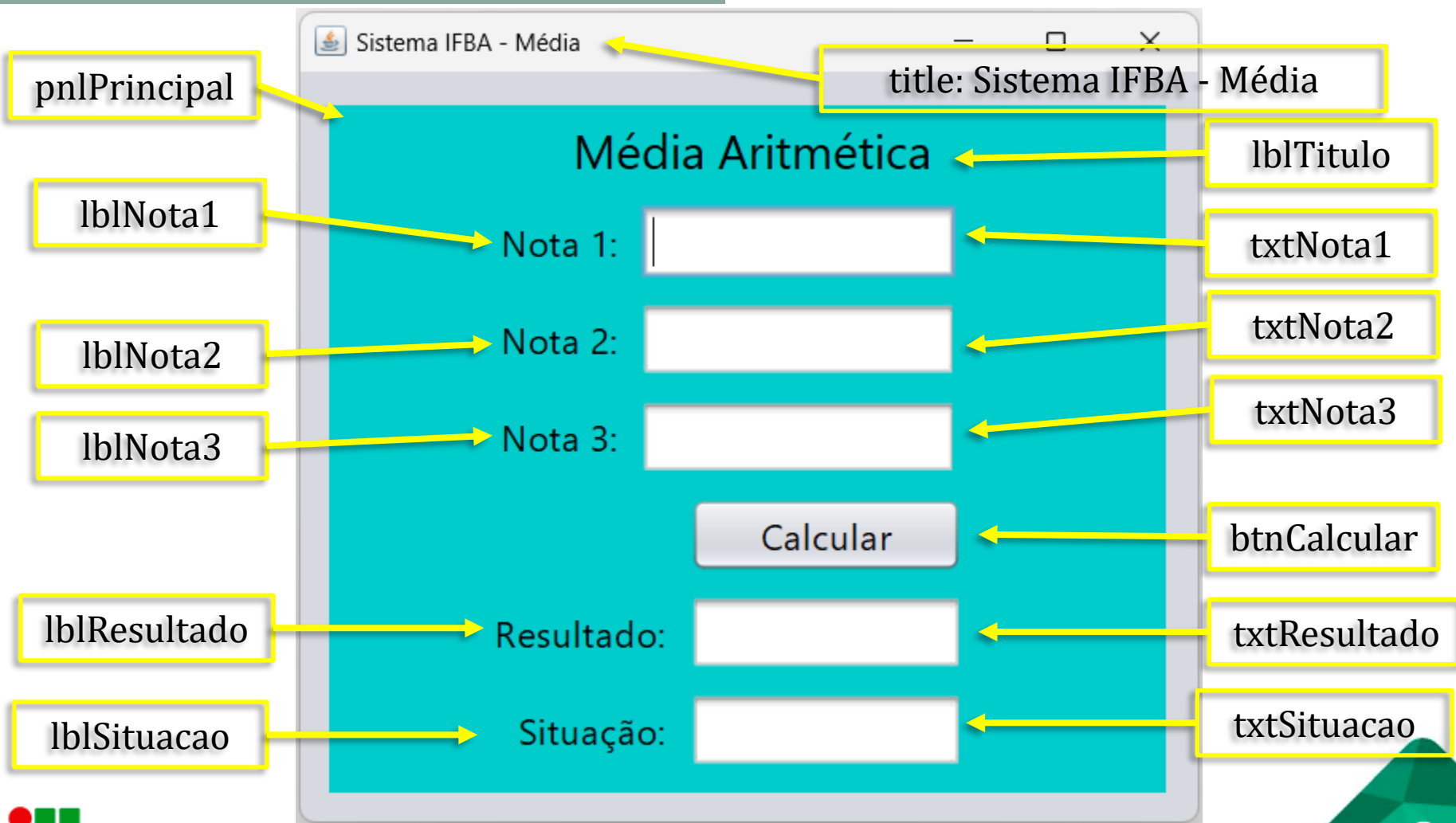
The screenshot shows a Java Swing window titled "Sistema IFBA - Calculadora". The window has a light blue background and contains the following elements:

- Valor 1:** A text input field containing the number "10".
- Valor 2:** A text input field containing the number "5".
- Operação:** A text input field containing the multiplication symbol "*".
- Calcular:** A button with the text "Calcular".
- Resultado:** A text input field containing the result "50.0".

A yellow arrow points from a yellow box containing the operators "+", "-", "*", and "/" to the "*" operator in the "Operação" field.



Java Swing – Exercício – Prog. Média



TelaMedia



Java Swing – Exercício – Prog. Média

Qual é o código executado ao acionar o botão “Calcular”?



Java Swing – Exercício – Prog. Média

Leia nota1 como um número decimal do campo de texto txtNota1.

Leia nota2 como um número decimal do campo de texto txtNota2.

Leia nota3 como um número decimal do campo de texto txtNota3.

Calcule a média como a soma de nota1, nota2 e nota3 dividido por 3.

Declare situacao como uma string.

Se a média for maior ou igual a 6.0:

atribua "Aprovação" a situacao.

Senão:

atribua "Recuperação" a situacao.

Arredonde a média para duas casas decimais.

Converta a média para uma string e atribua ao campo de texto txtResultado.

Atribua a situacao ao campo de texto txtSituacao.



Java Swing – Exercício – Prog. Média

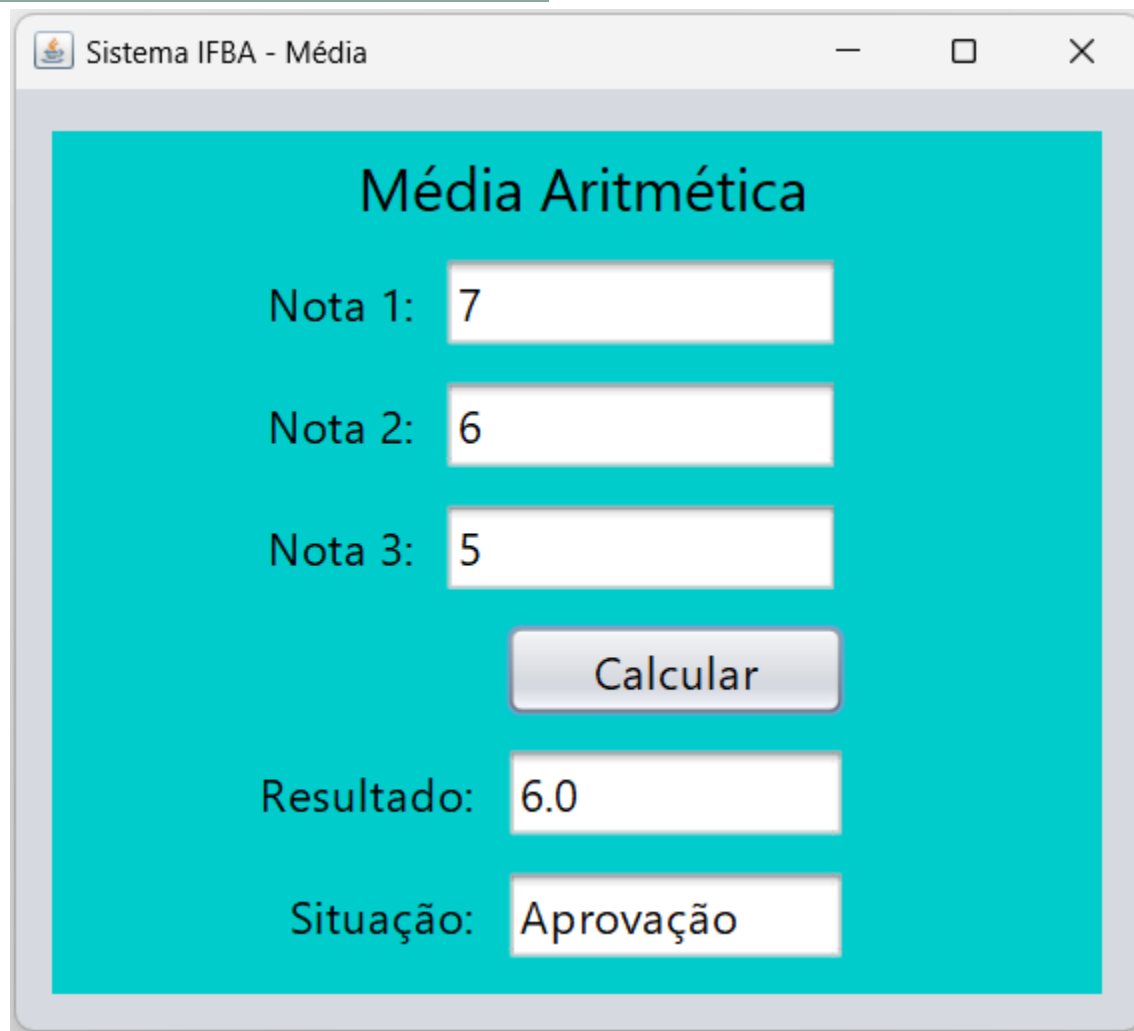
Arredondar o valor da variável “media” do tipo double para duas casas decimais:

```
media = Math.floor(media * 100) / 100;
```



Java Swing – Exercício – Prog. Média

Execução



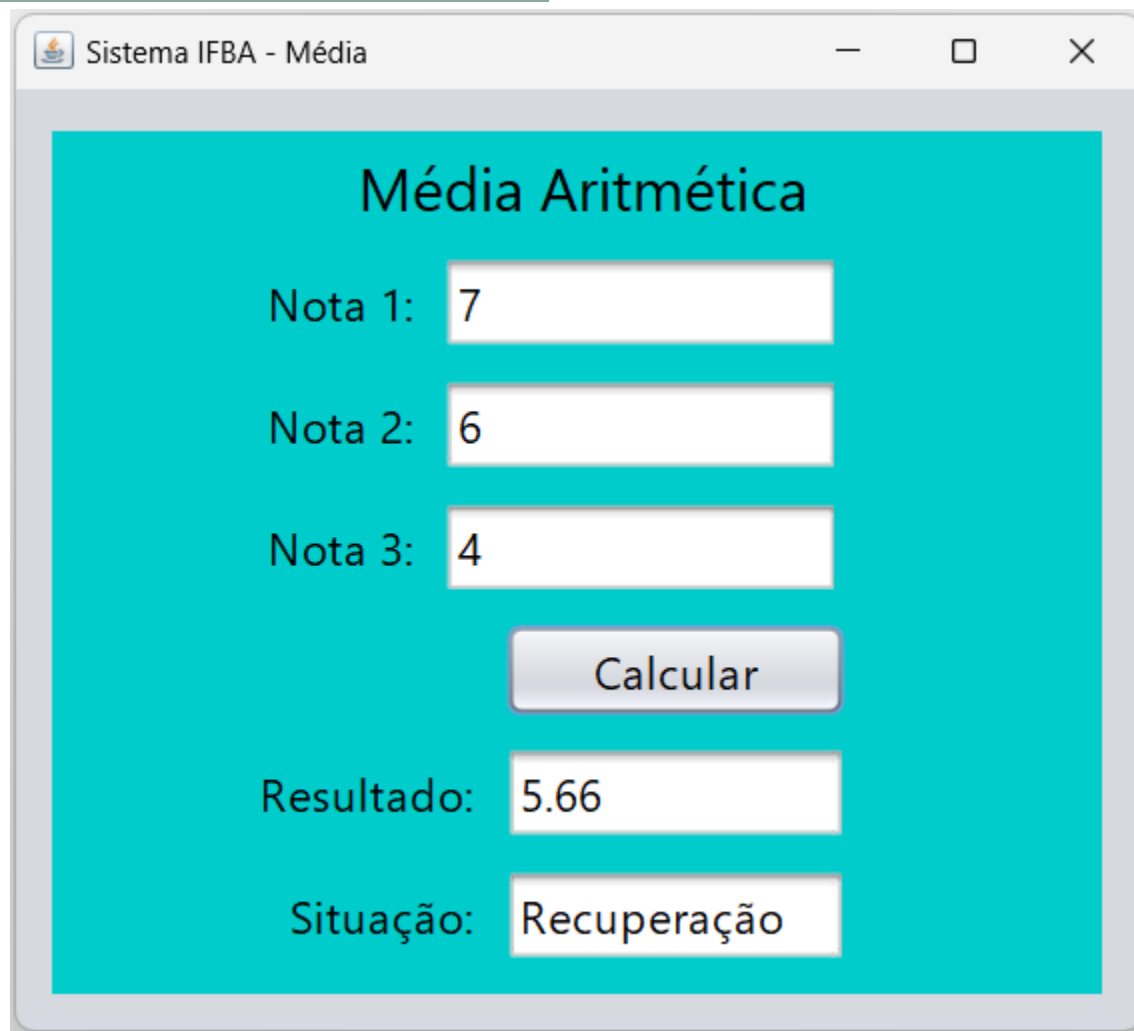
The screenshot shows a Java Swing window titled "Sistema IFBA - Média". The window has a light blue background and contains the following elements:

- Média Aritmética**: The title of the application.
- Nota 1:** A text label followed by a text input field containing the value "7".
- Nota 2:** A text label followed by a text input field containing the value "6".
- Nota 3:** A text label followed by a text input field containing the value "5".
- Calcular**: A button with a light blue background and a dark blue border.
- Resultado:** A text label followed by a text input field containing the value "6.0".
- Situação:** A text label followed by a text input field containing the value "Aprovação".



Java Swing – Exercício – Prog. Média

Execução

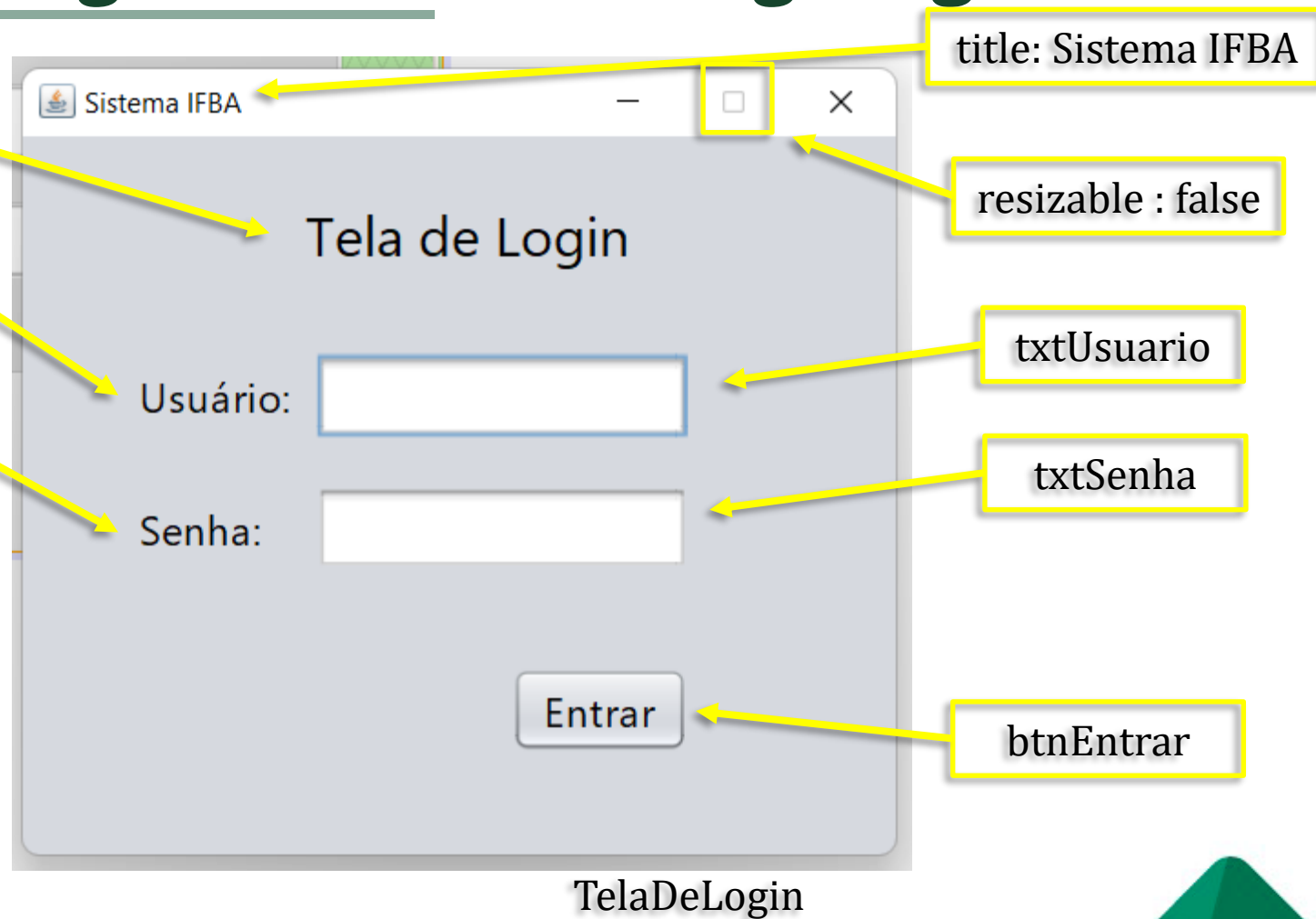


The screenshot shows a Java Swing window titled "Sistema IFBA - Média". The window has a light blue background and contains the following elements:

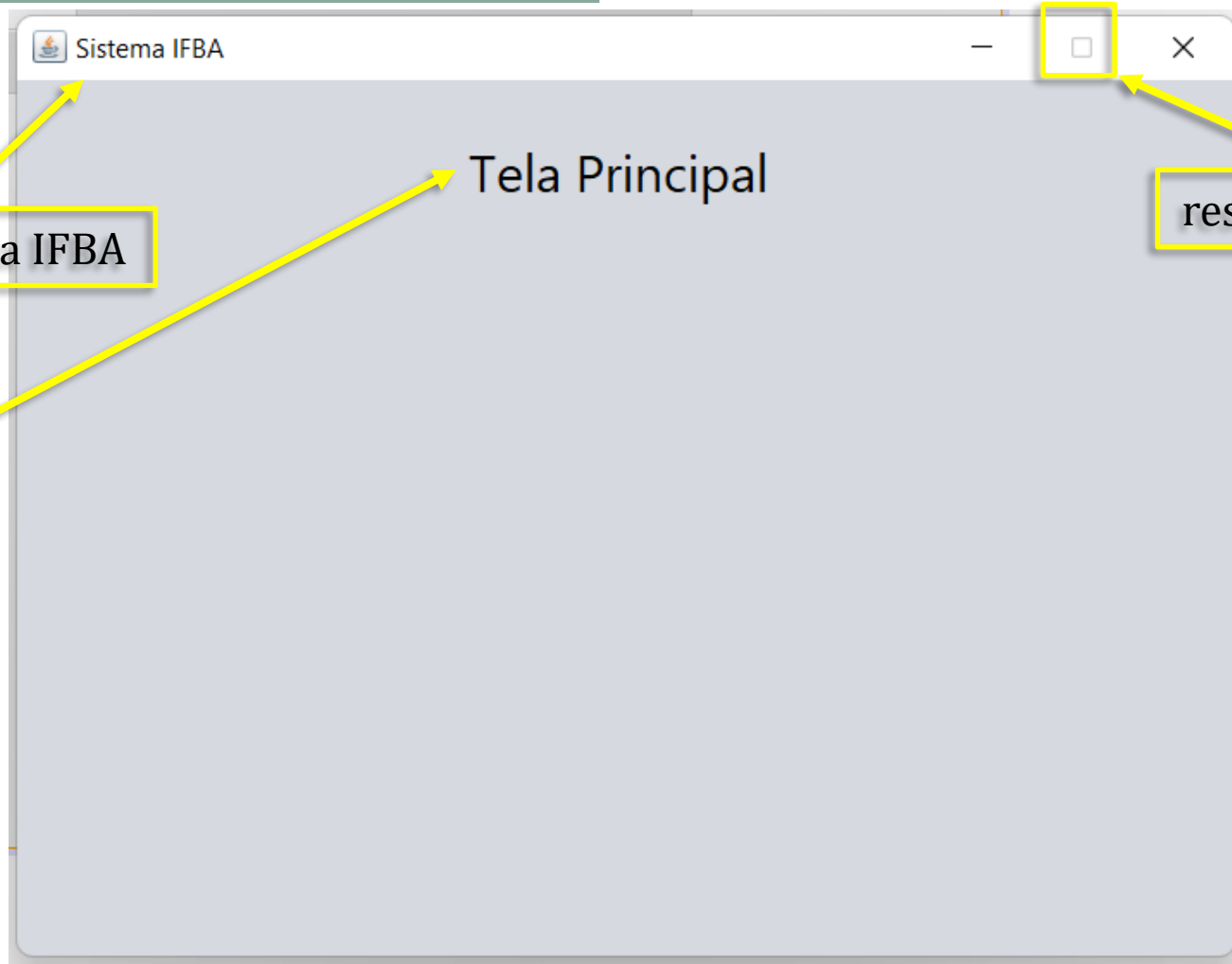
- Média Aritmética**: The title of the application.
- Nota 1:** A text label followed by a text field containing the value "7".
- Nota 2:** A text label followed by a text field containing the value "6".
- Nota 3:** A text label followed by a text field containing the value "4".
- Calcular**: A button with a light blue background and a dark blue border.
- Resultado:** A text label followed by a text field containing the value "5.66".
- Situação:** A text label followed by a text field containing the value "Recuperação".



Java Swing – Exercício – Prog. Login



Java Swing – Exercício – Prog. Principal



title: Sistema IFBA

lblTitulo

resizable : false

TelaPrincipal



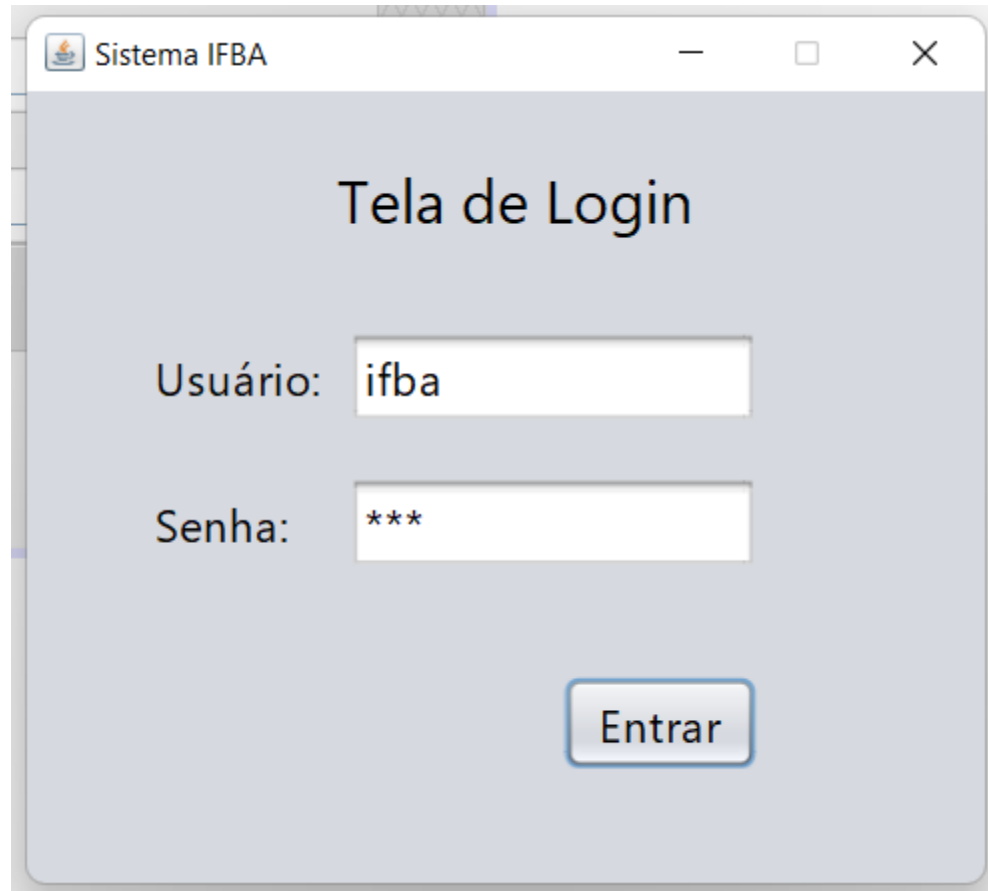
Java Swing – Exercício – Prog. Login

```
private void btnEntrarActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String usuario = txtUsuario.getText();  
  
    String senha = String.valueOf(txpSenha.getPassword());  
  
    if(usuario.equals("ifba") && senha.equals("1234"))  
    {  
        new TelaPrincipal().setVisible(true);  
  
        this.dispose();  
    }  
    else  
    {  
        JOptionPane.showMessageDialog(this, "Acesso negado!");  
    }  
}
```



Java Swing – Exercício – Prog. Login

Execução

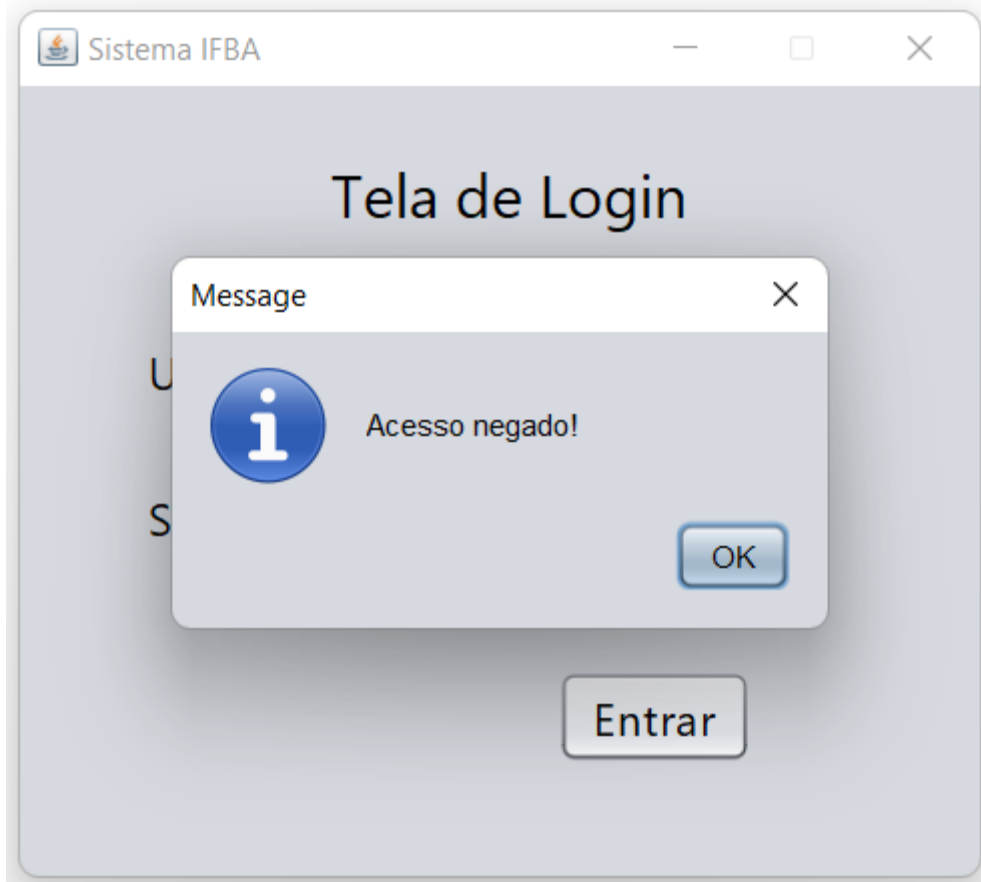


The screenshot shows a Java Swing window titled "Sistema IFBA" with standard window controls (minimize, maximize, close). The window contains a login form titled "Tela de Login". It features two text input fields: one for the username labeled "Usuário:" containing the text "ifba", and another for the password labeled "Senha:" containing three asterisks "***". Below the password field is a button labeled "Entrar".



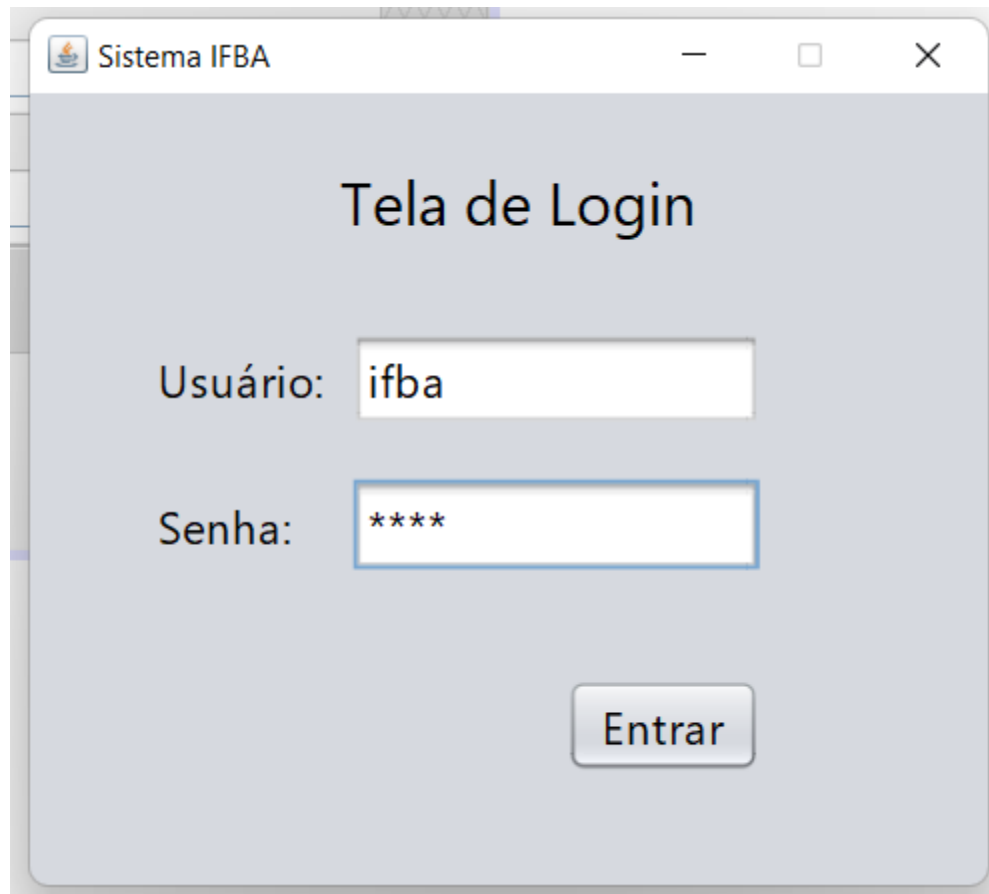
Java Swing – Exercício – Prog. Login

Execução



Java Swing – Exercício – Prog. Login

Execução

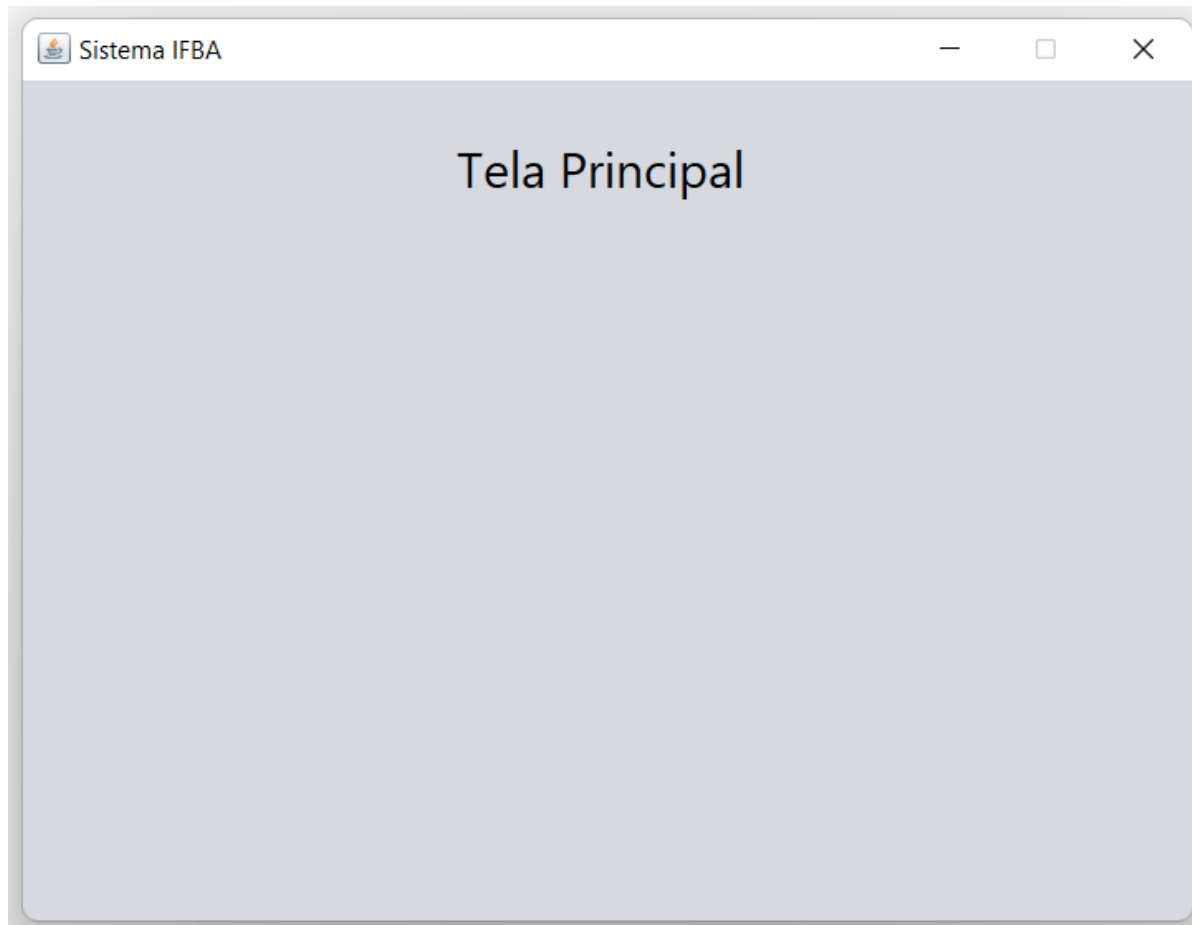


The screenshot shows a Java Swing window titled "Sistema IFBA". Inside the window, the text "Tela de Login" is centered. Below this, there are two input fields. The first is labeled "Usuário:" and contains the text "ifba". The second is labeled "Senha:" and contains four asterisks "****". Below the input fields is a button labeled "Entrar".



Java Swing – Exercício – Prog. Login

Execução



Referências

- Junior, Peter Jandl. Java Guia do Programador - 4ª Edição. Novatec Editora.
- SÉRGIO FURGERI. Java Ensino Didático: Desenvolvimento e Implementação de Aplicações. Editora Érica.
- <https://www.youtube.com/playlist?list=PLwQkYMetu0OYFOU71txhtvSYTVUCHjJYD>
- https://www.youtube.com/playlist?list=PLWd_VnthxxLfeEUK08gB7G3316OS5xIT3



Obrigado!

- Canais de Comunicação;
- Horário de Atendimento.

