



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

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

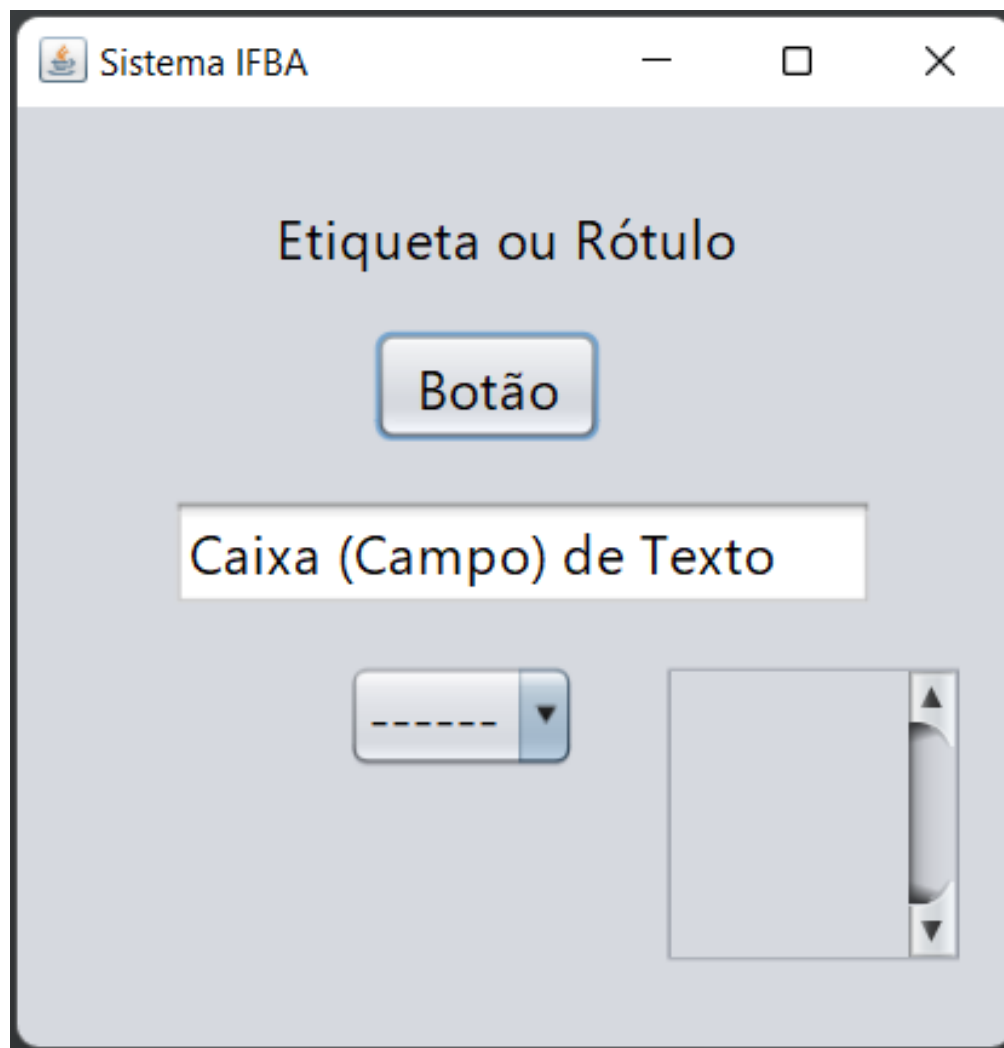


# Java Swing – Componentes

- 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



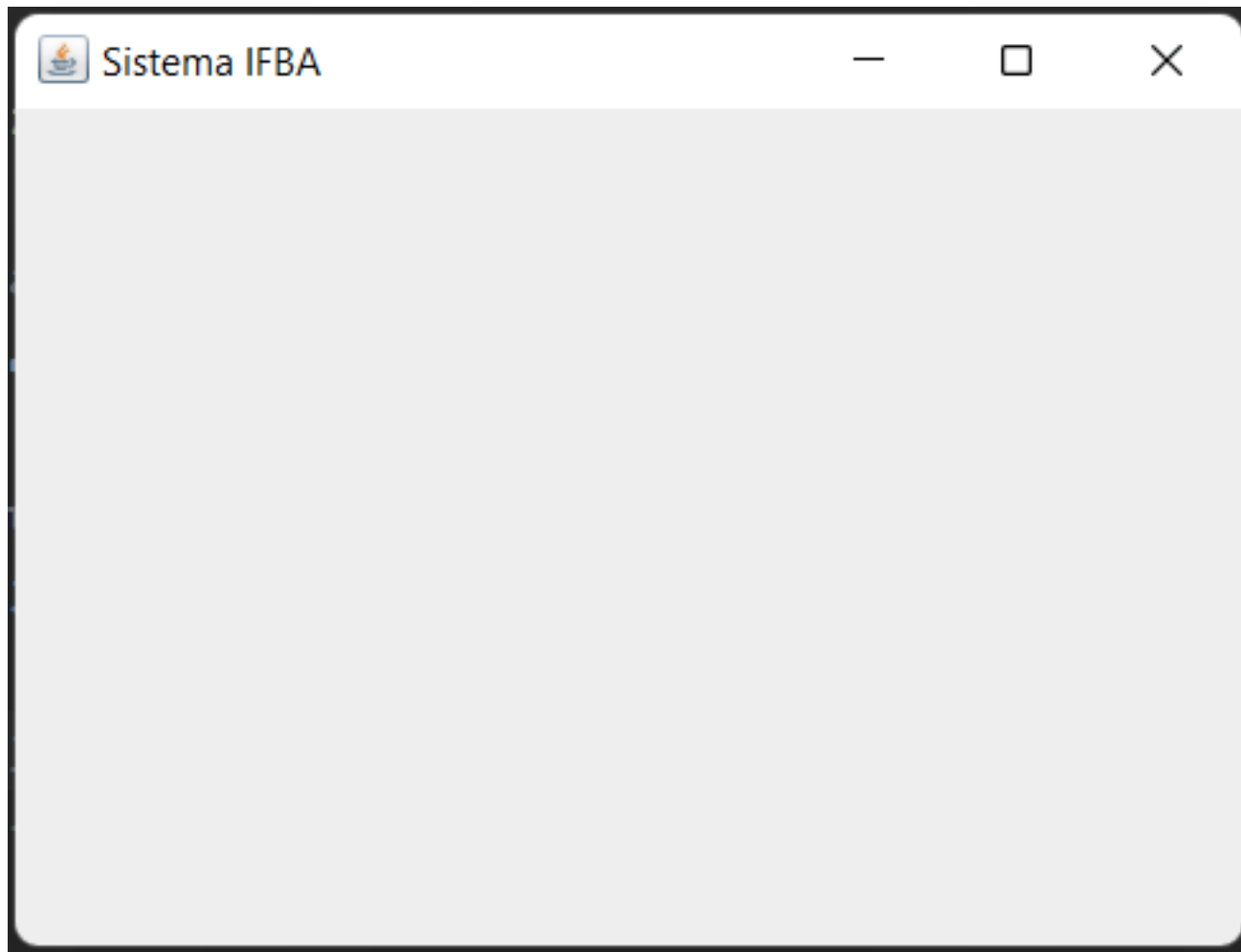
# Java Swing – JFrame

---

- 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**.

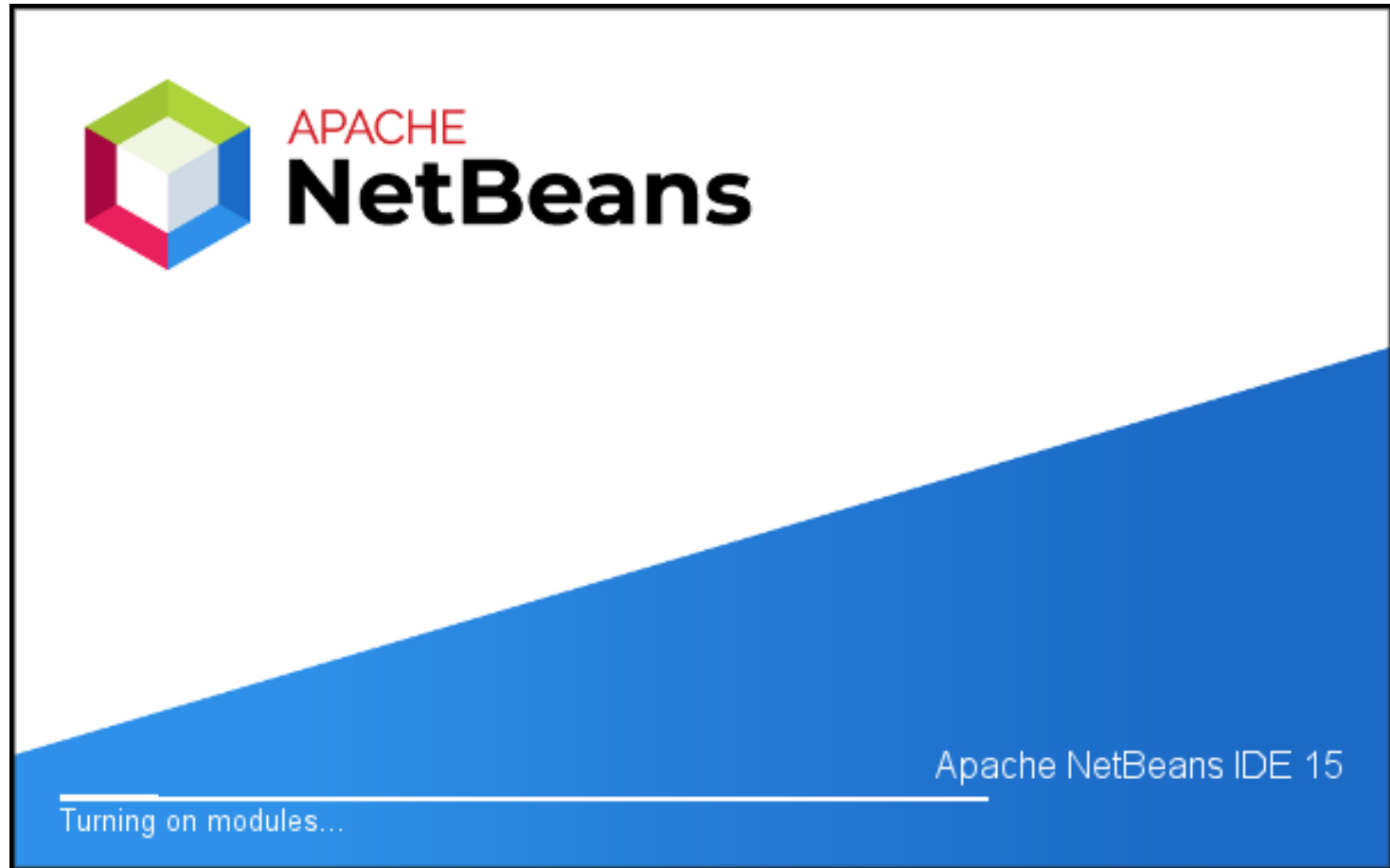


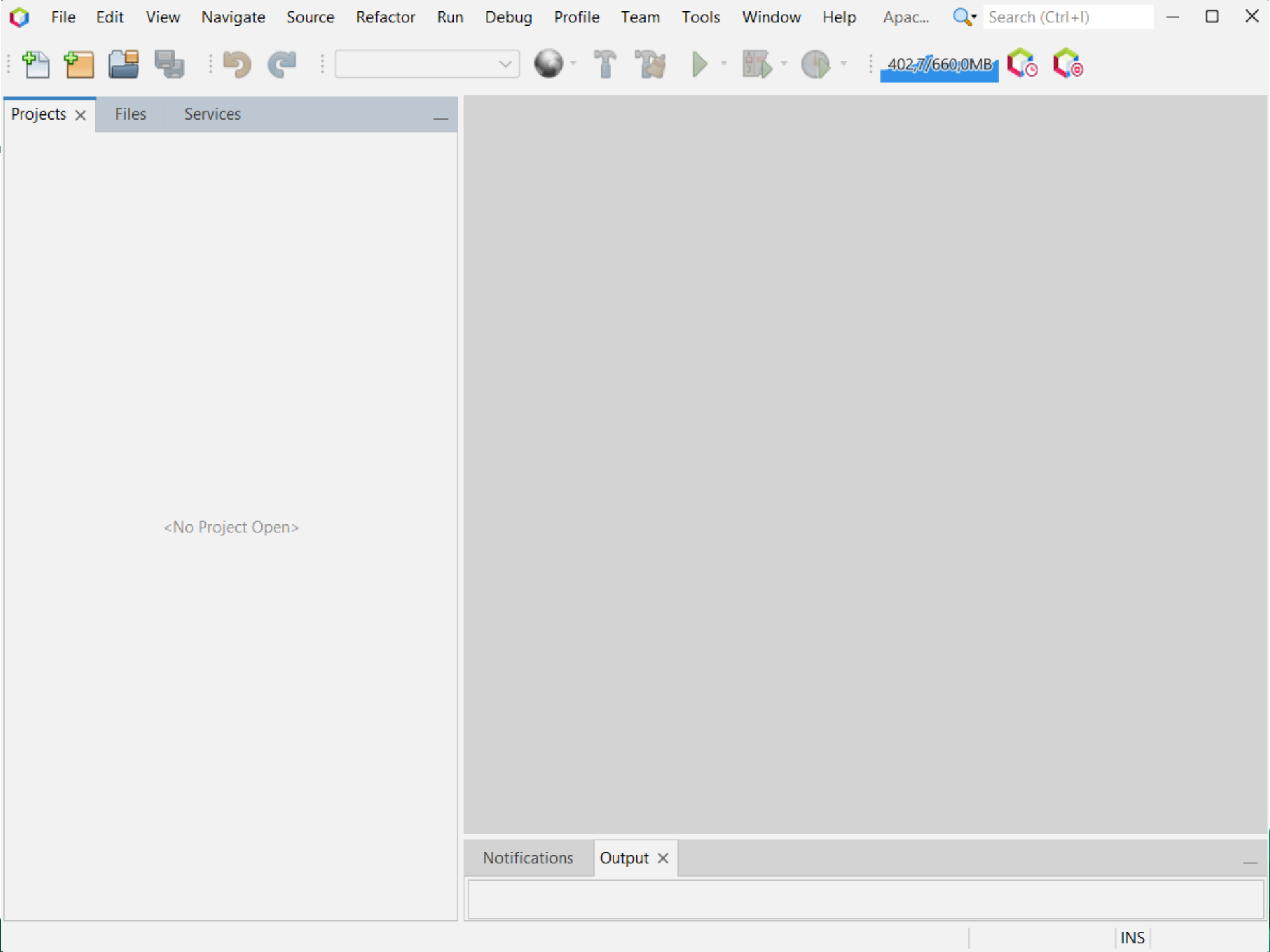
# Java Swing – JFrame



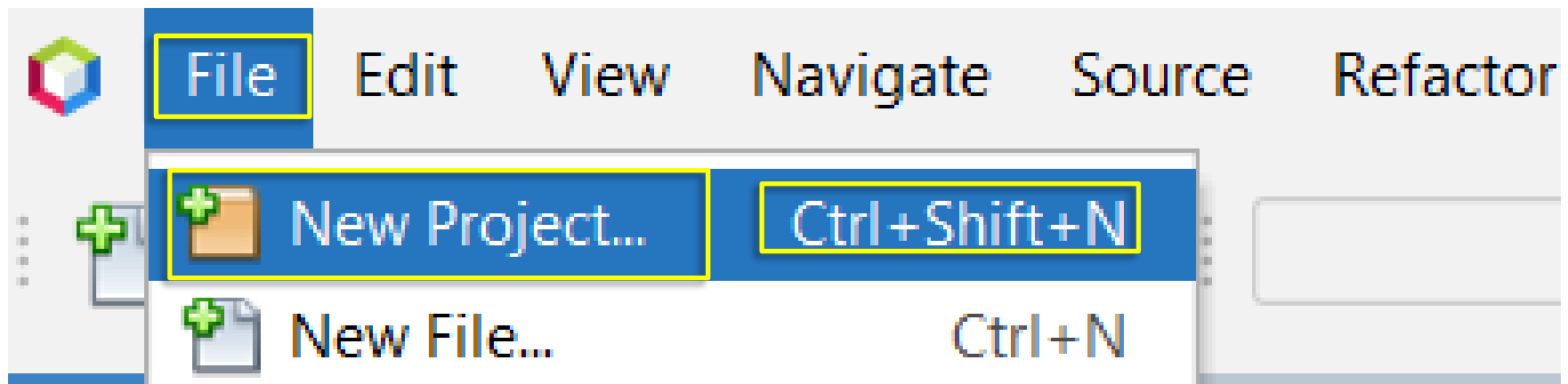


# Netbeans: IDE





# 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

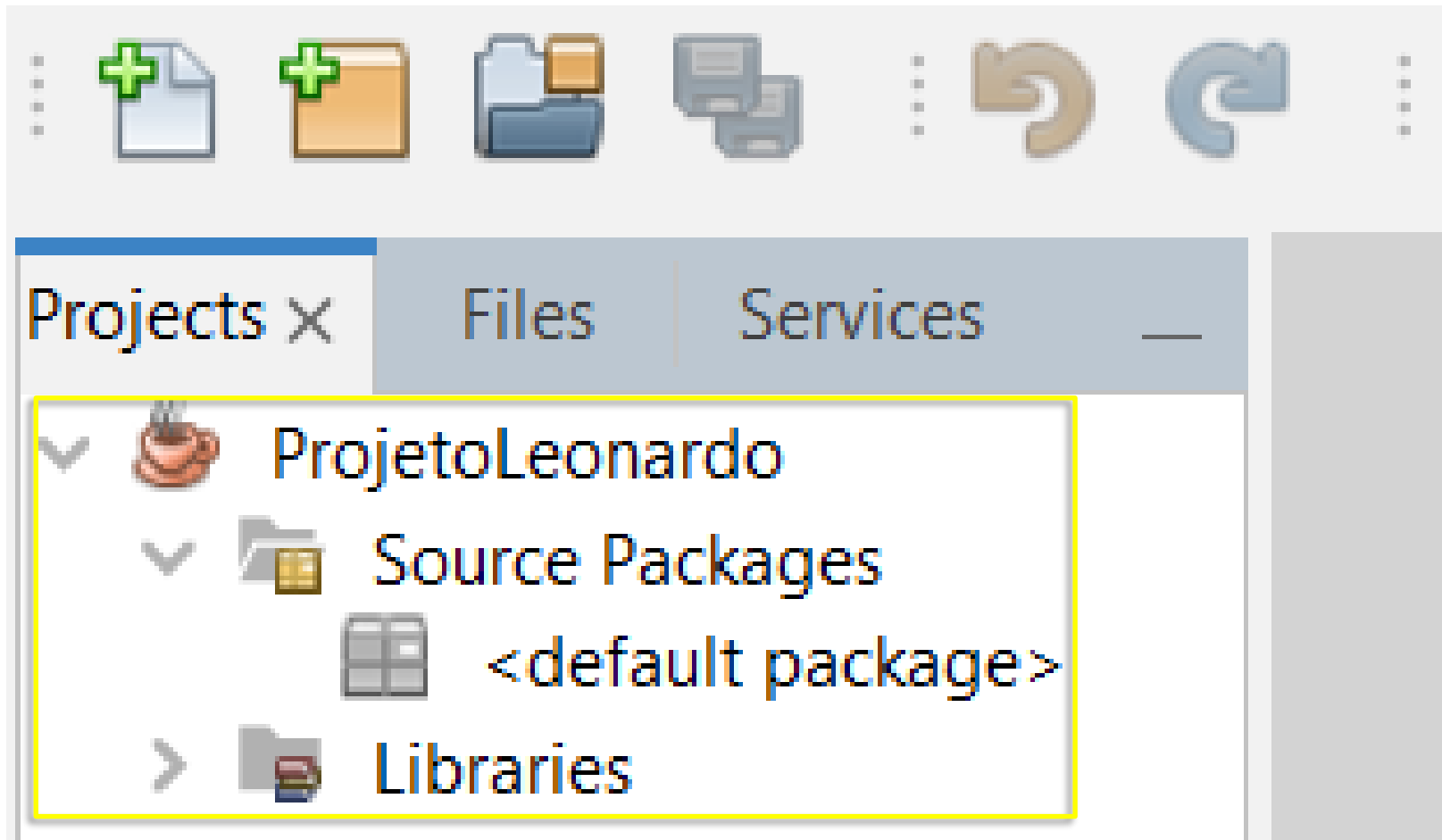
Libraries Folder:

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

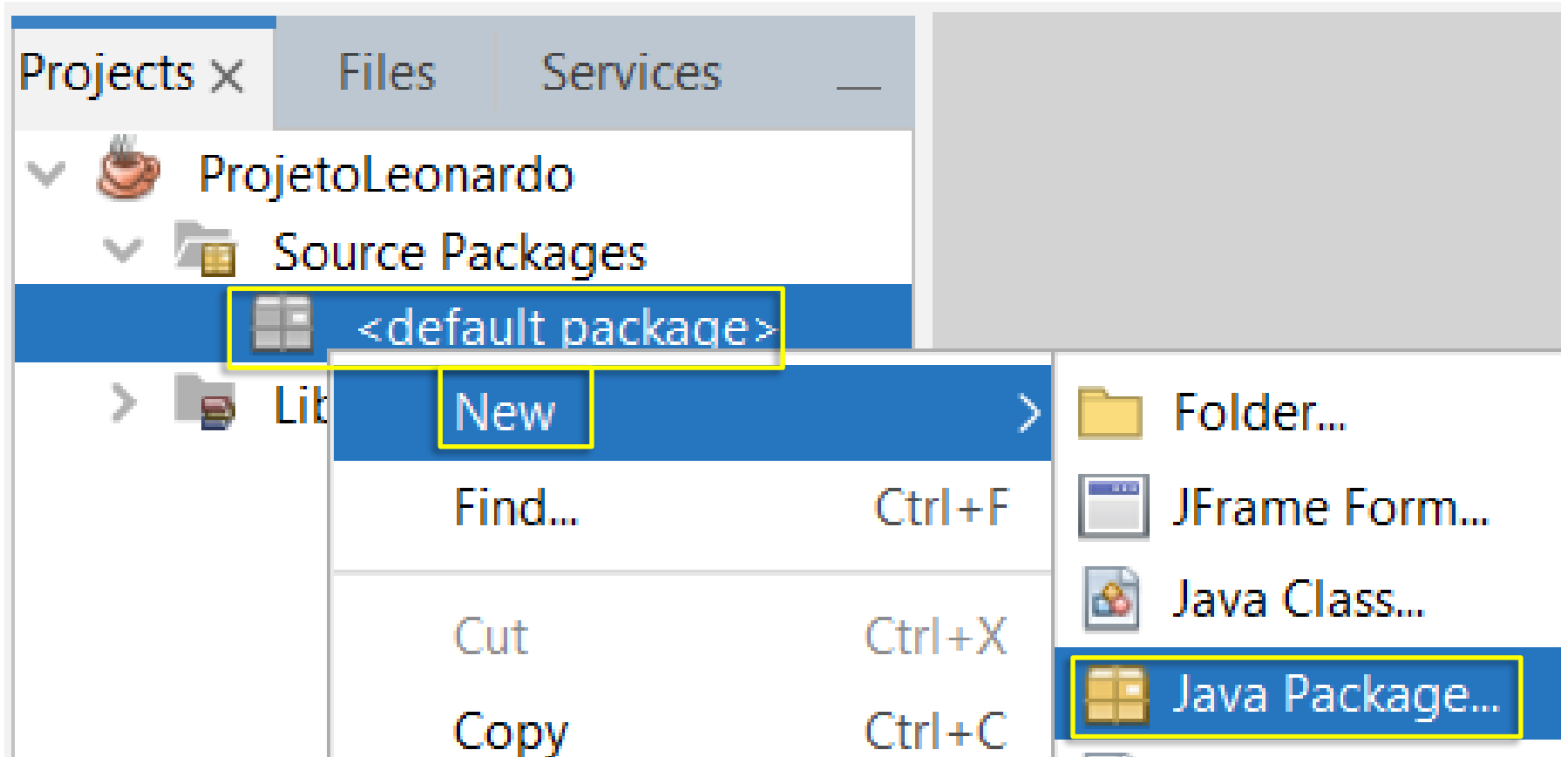
☐ Create Main Class



# Netbeans: Novo Projeto



# Netbeans: Novo Pacote



# Netbeans: Novo Pacote

## Name and Location

Package Name:

Principal

Project:

ProjetoLeonardo

Location:

Source Packages

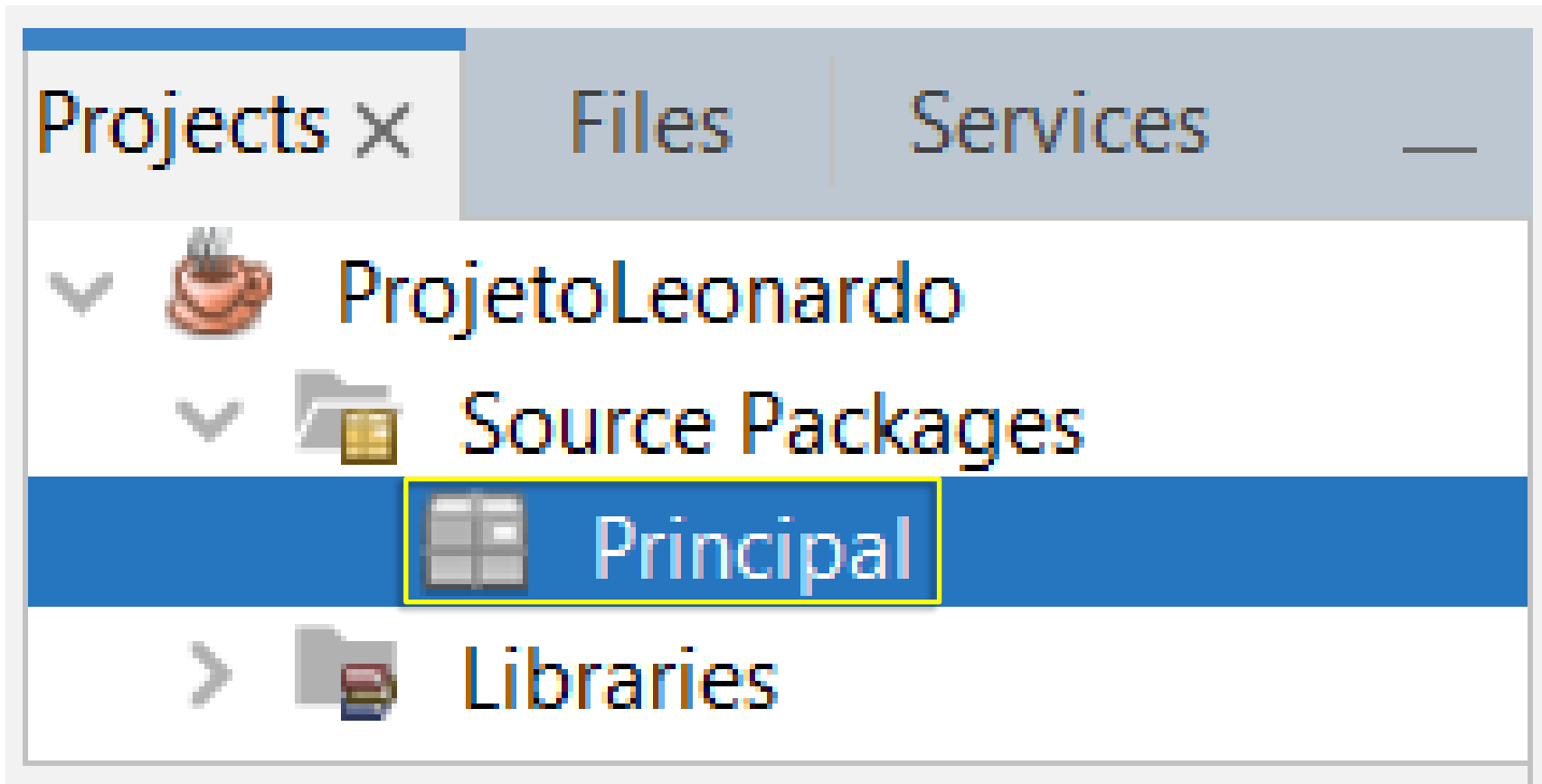
Created Folder:

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

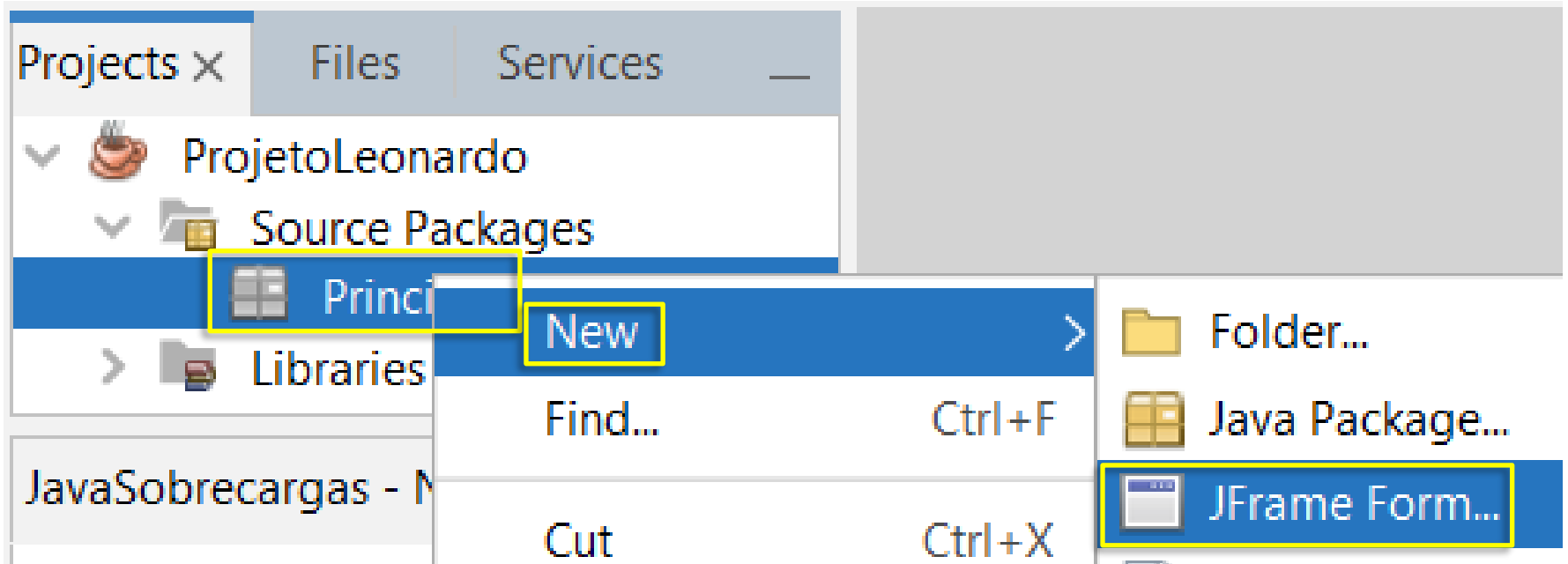




# Netbeans: Novo Pacote



# Netbeans: Nova Janela (JFrame)



# Netbeans: Nova Janela (Jframe)

## Name and Location

Class Name:

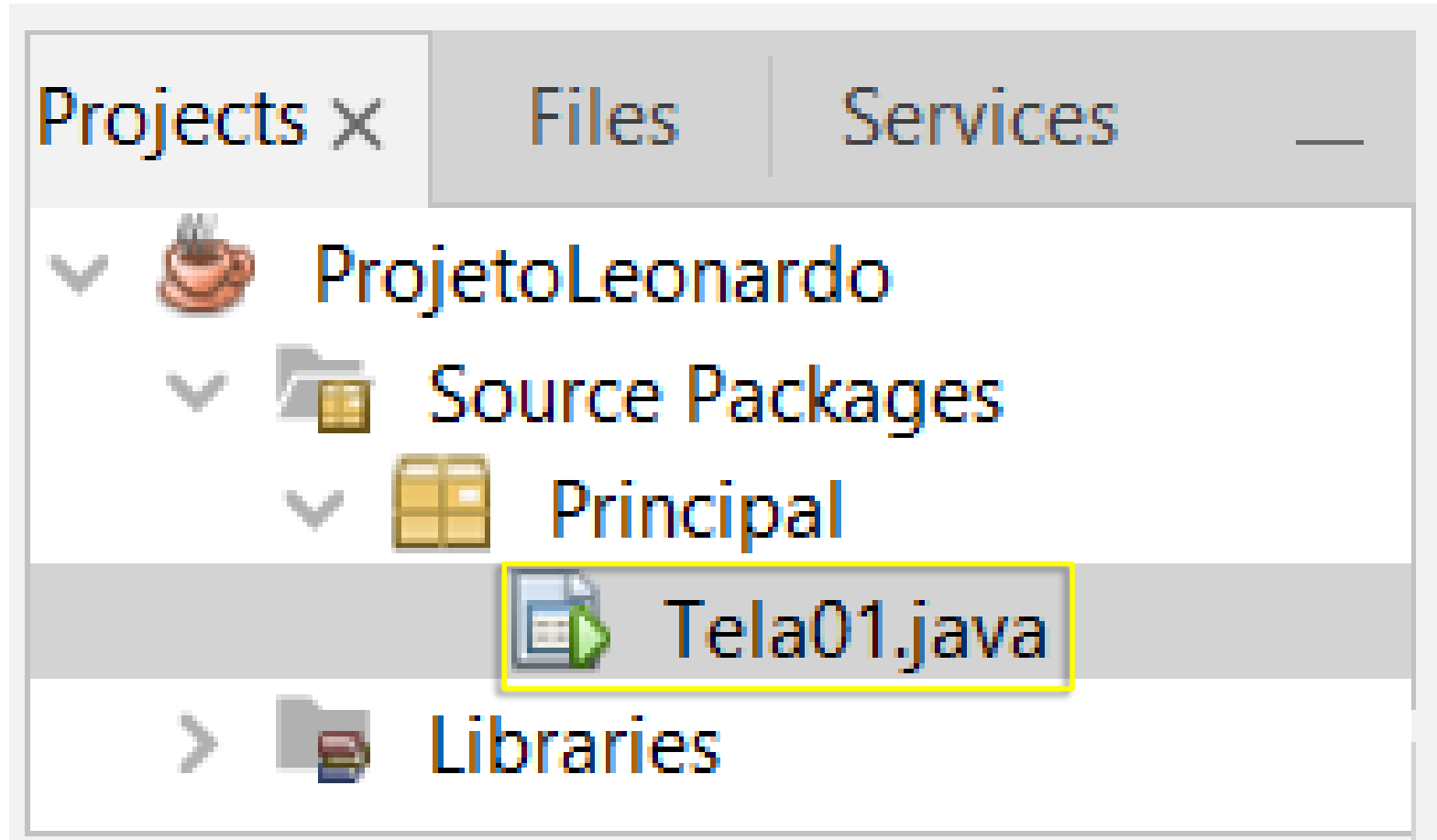
Project:

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		...

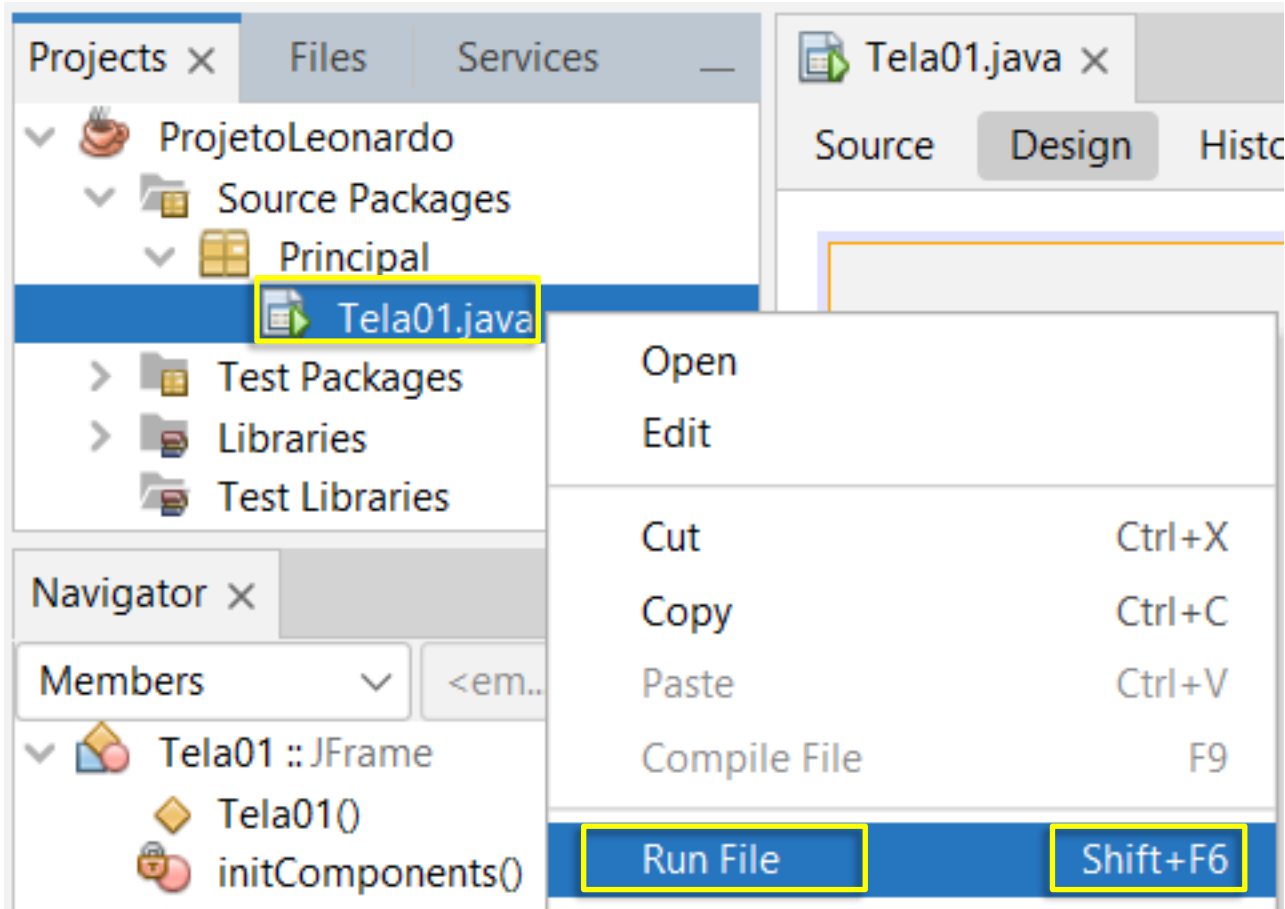
Other Properties

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

[JFrame]

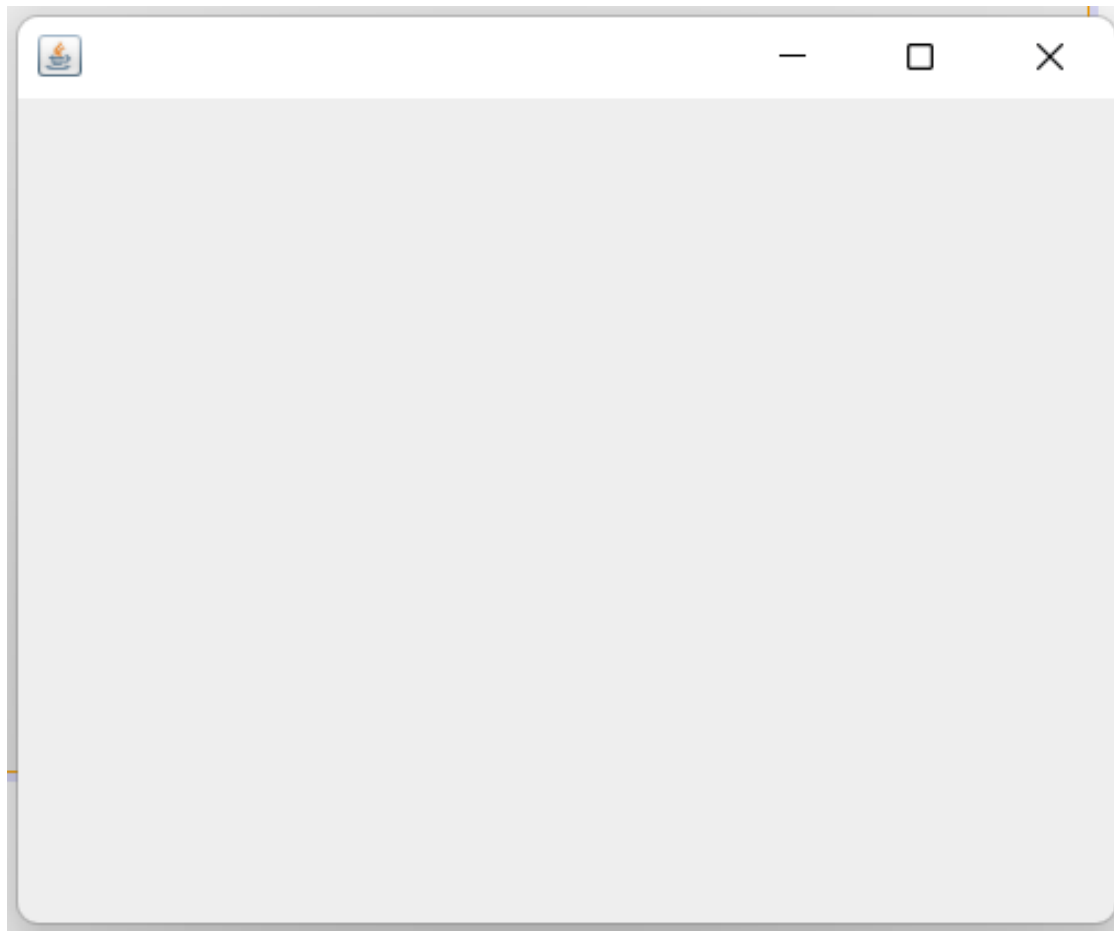
Notifications Output x

# 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

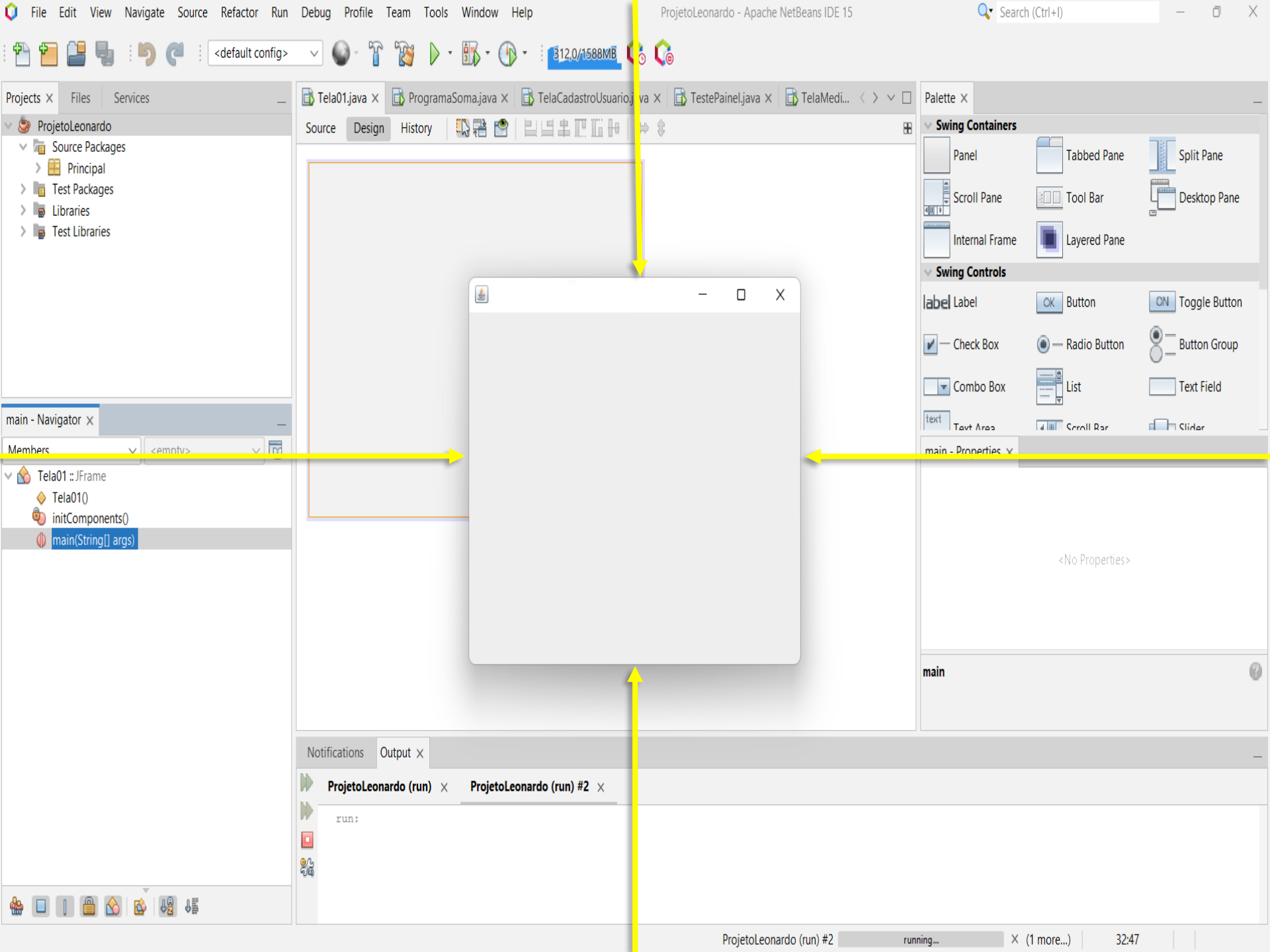
Configurações de Layout do JFrame

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

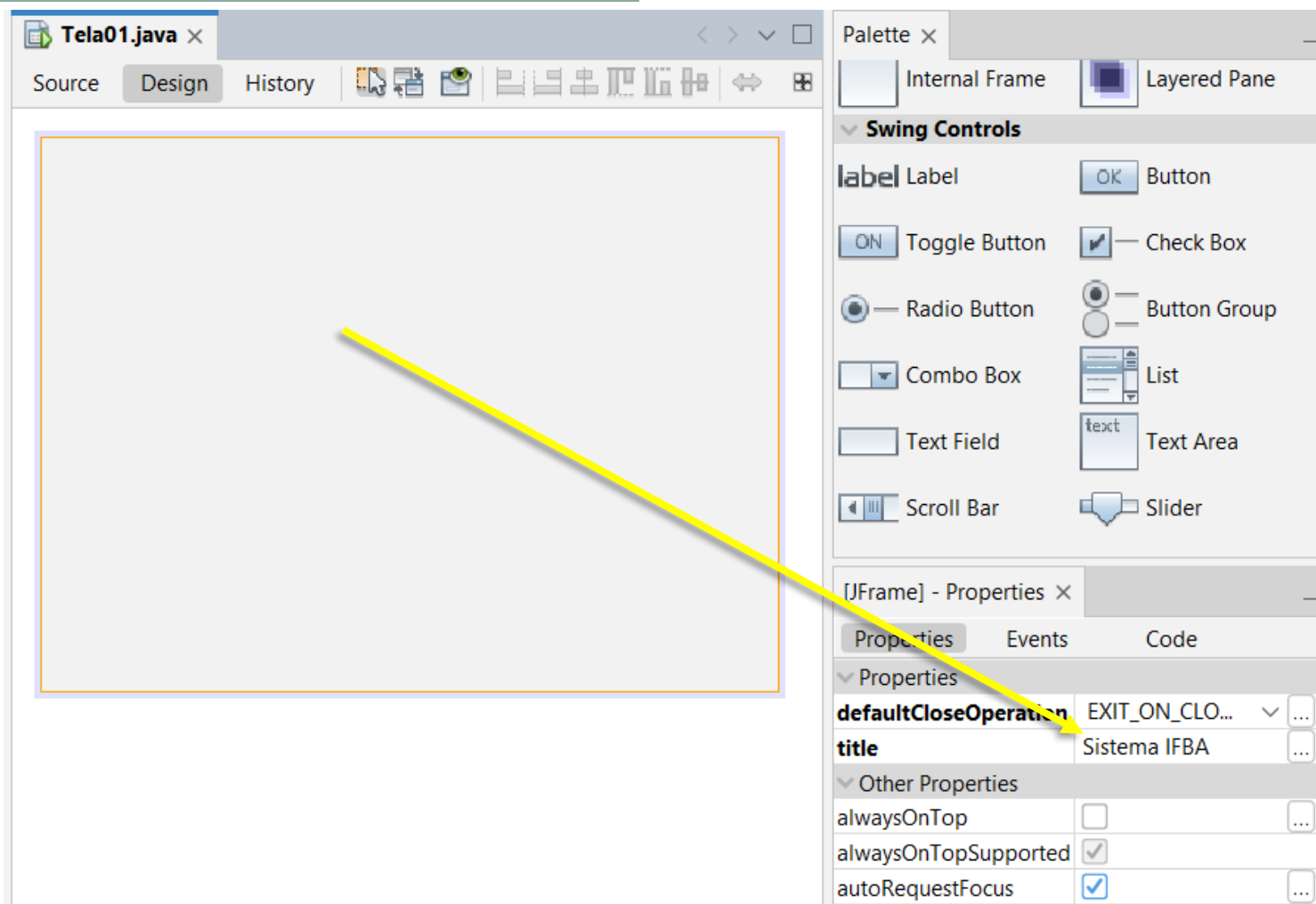
# Netbeans: Centralizar JFrame

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

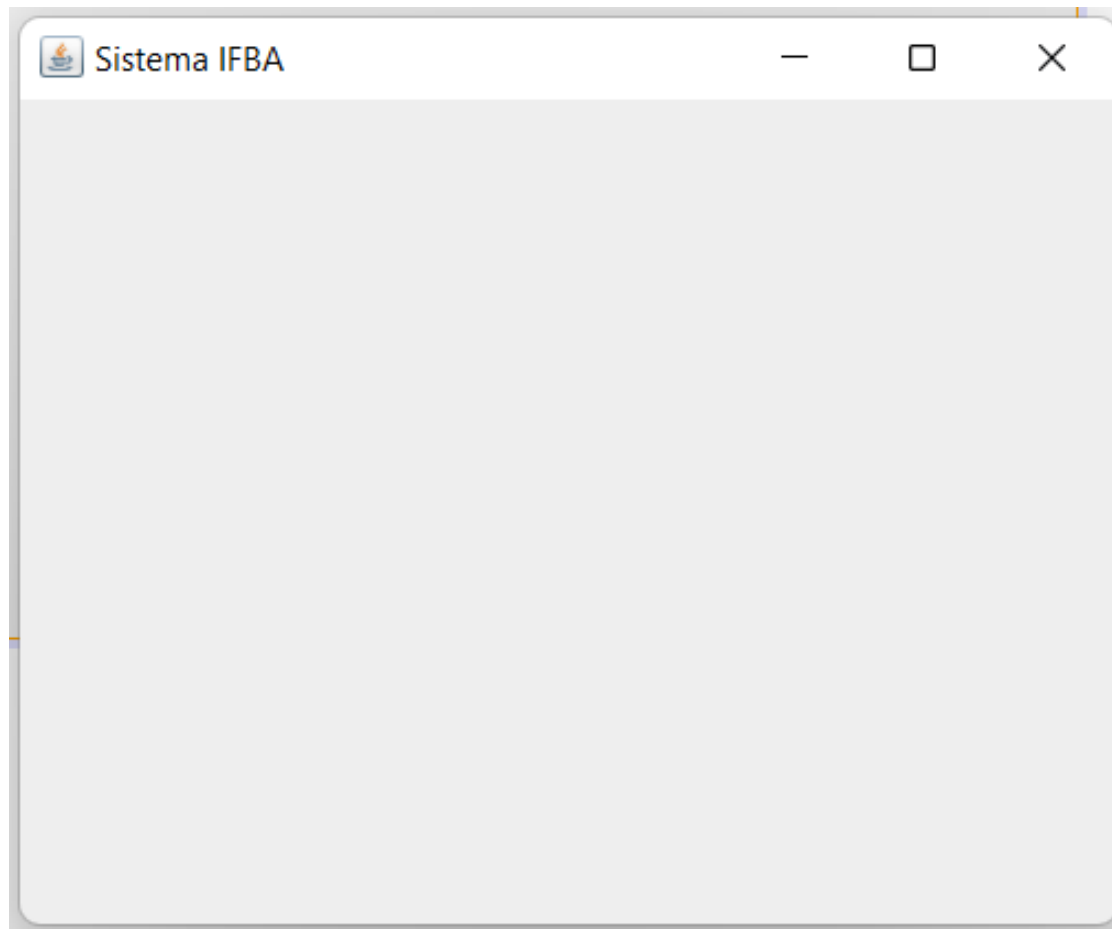




# Netbeans: Título da Janela (JFrame)



# Netbeans: Título da Janela (JFrame)



# Netbeans: Desabilitar Redimensionar

[JFrame] - Properties X

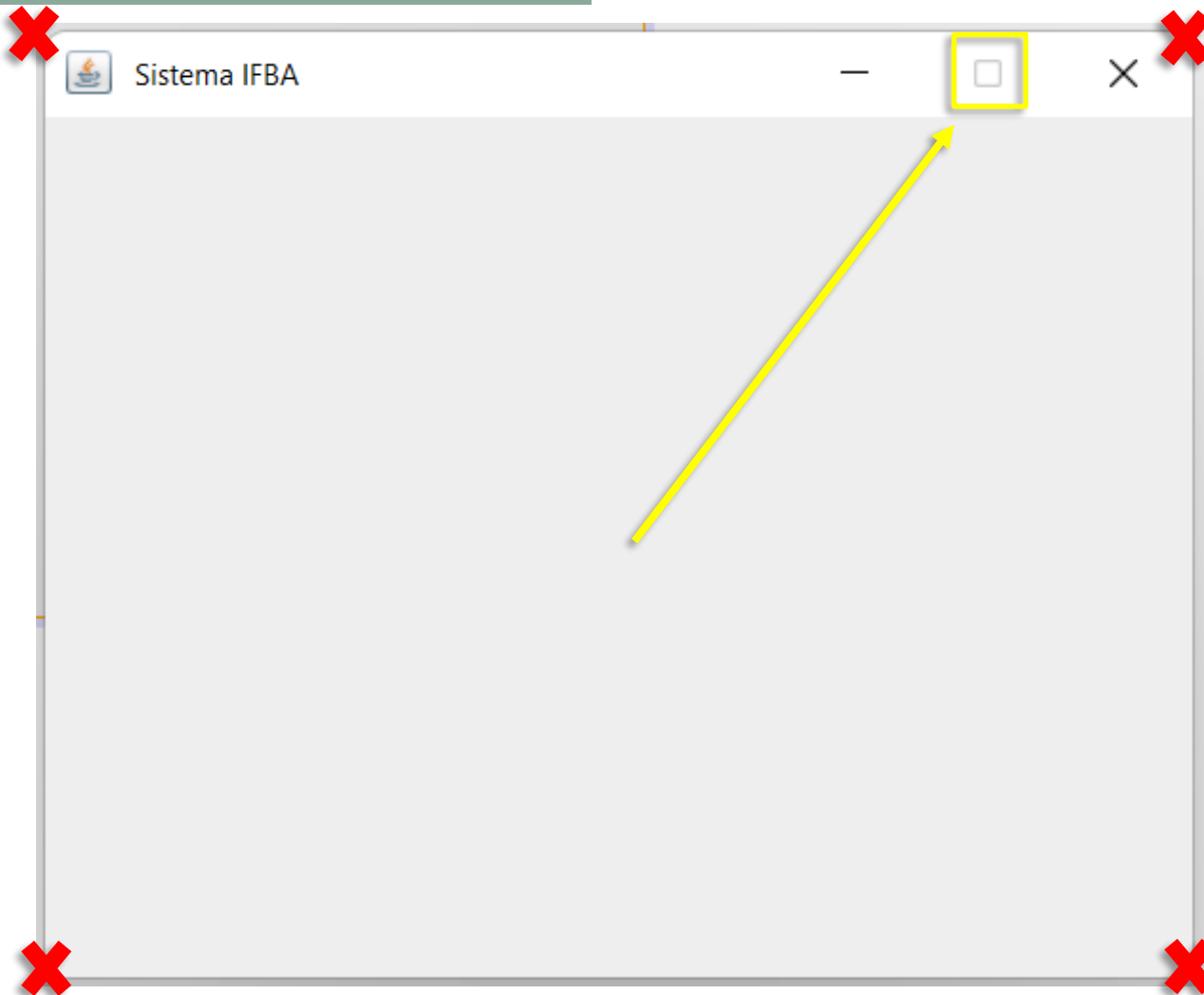
Properties	Events	Code
preferredSize		[395, 300]
<b>resizable</b>	<input type="checkbox"/>	
shape		<none>
size		<Not Set>
state		0
type		NORMAL

False





# Netbeans: Desabilitar Redimensionar

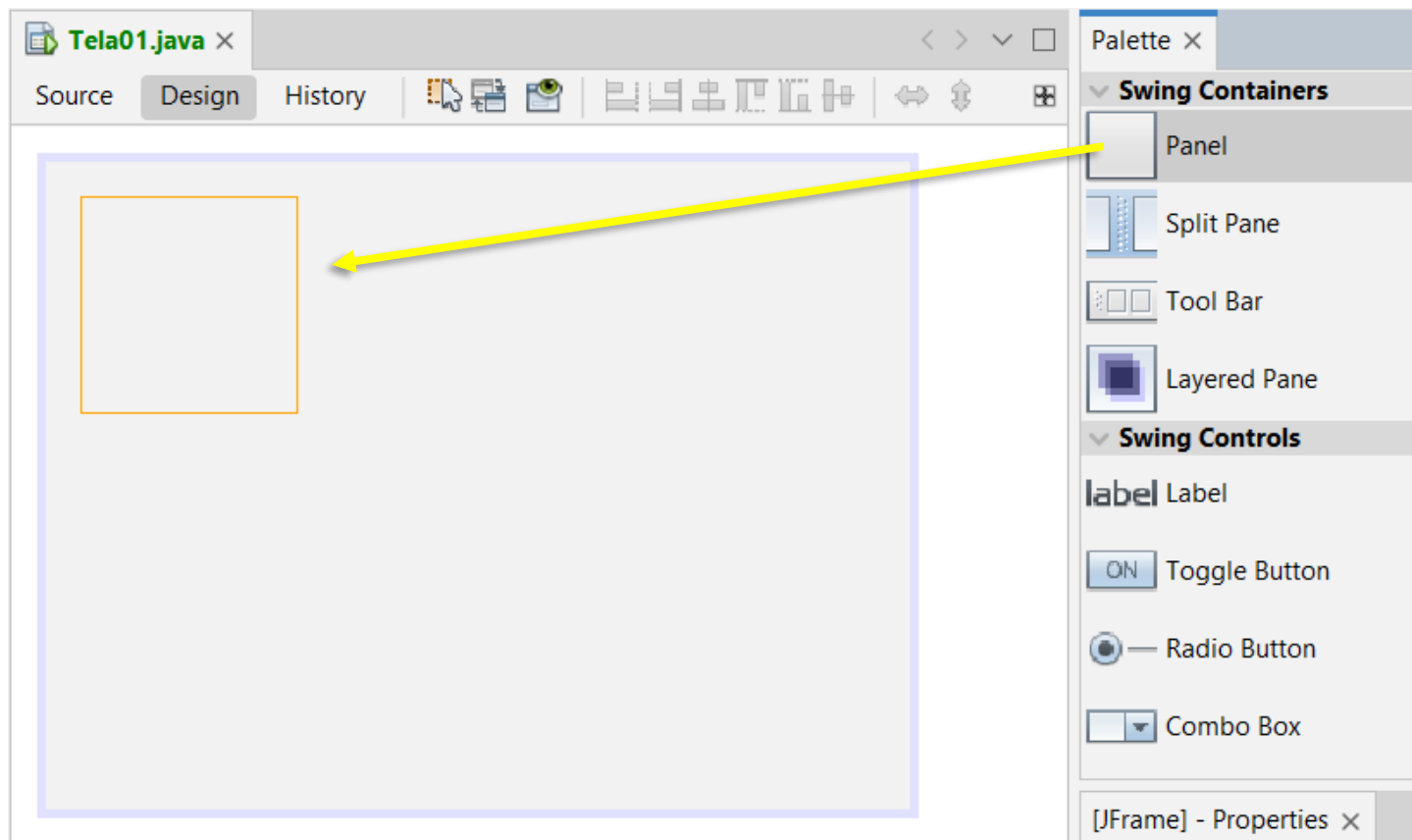


# Java Swing – JPanel

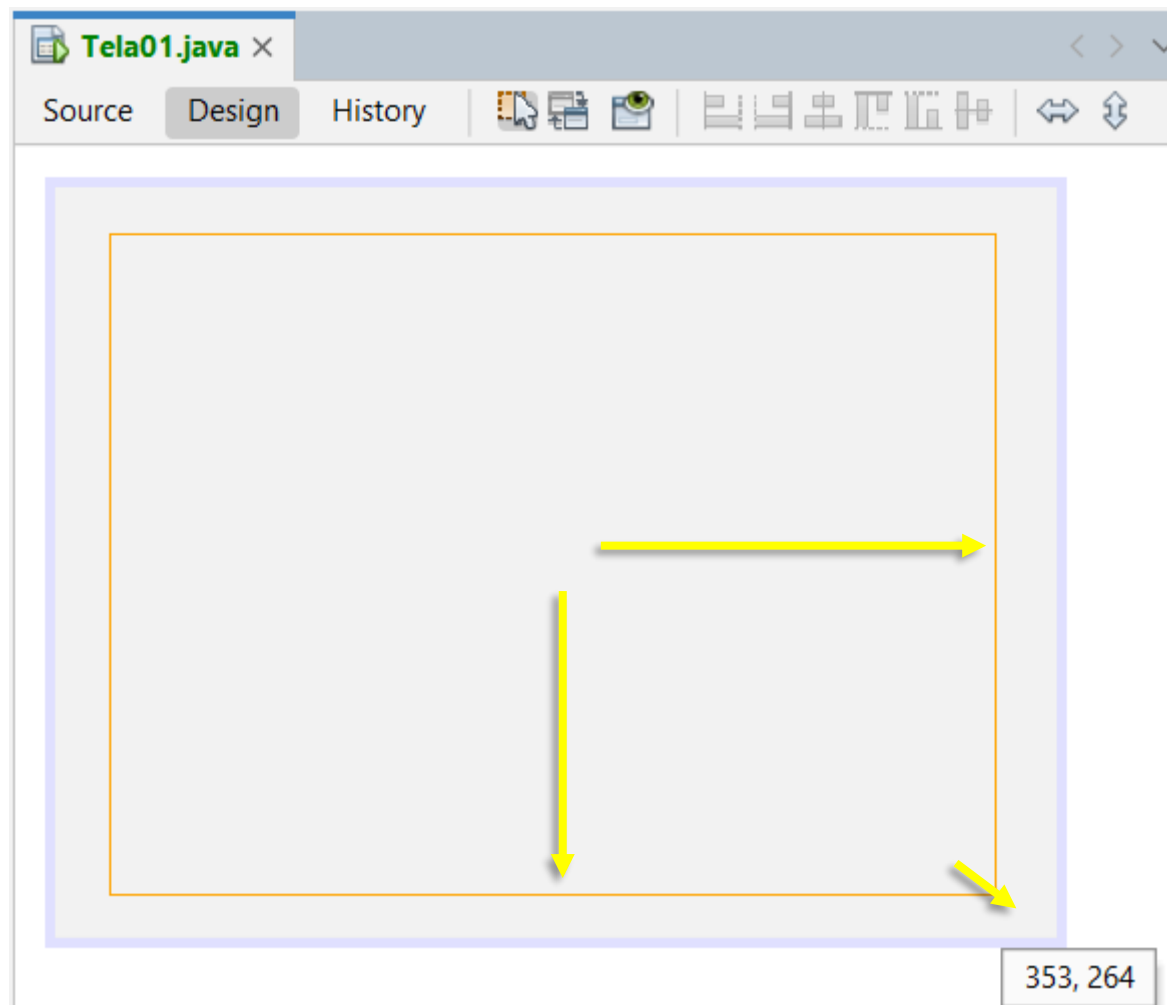
- O **JPanel** é um **contêiner flexível** no Java Swing, utilizado para **agrupar e organizar componentes** em interfaces gráficas.
- Ele **oferece opções de personalização**, como **layout e cor de fundo**, permitindo criar **interfaces atraentes e adaptáveis**.



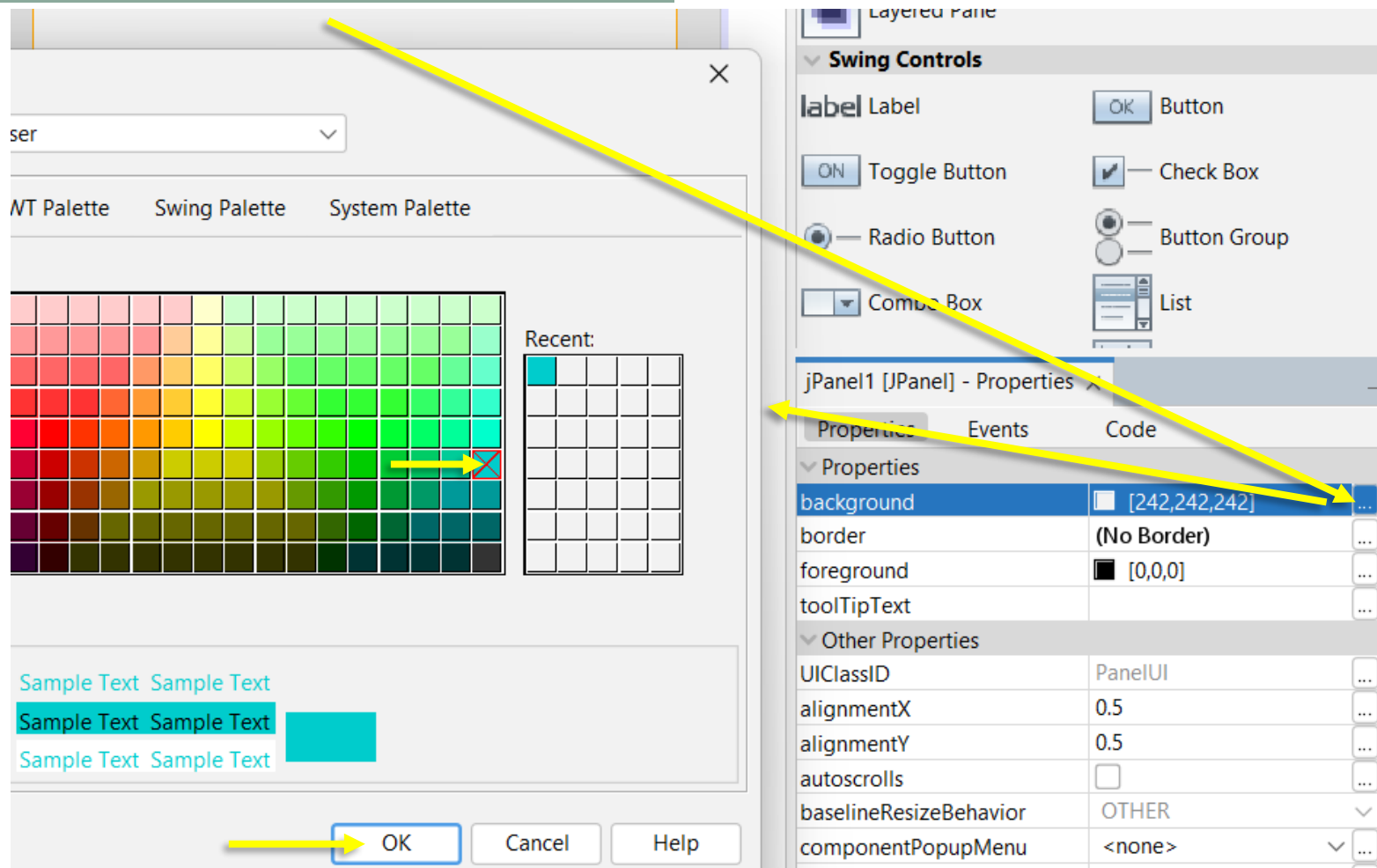
# Java Swing – JPanel - Adicionar



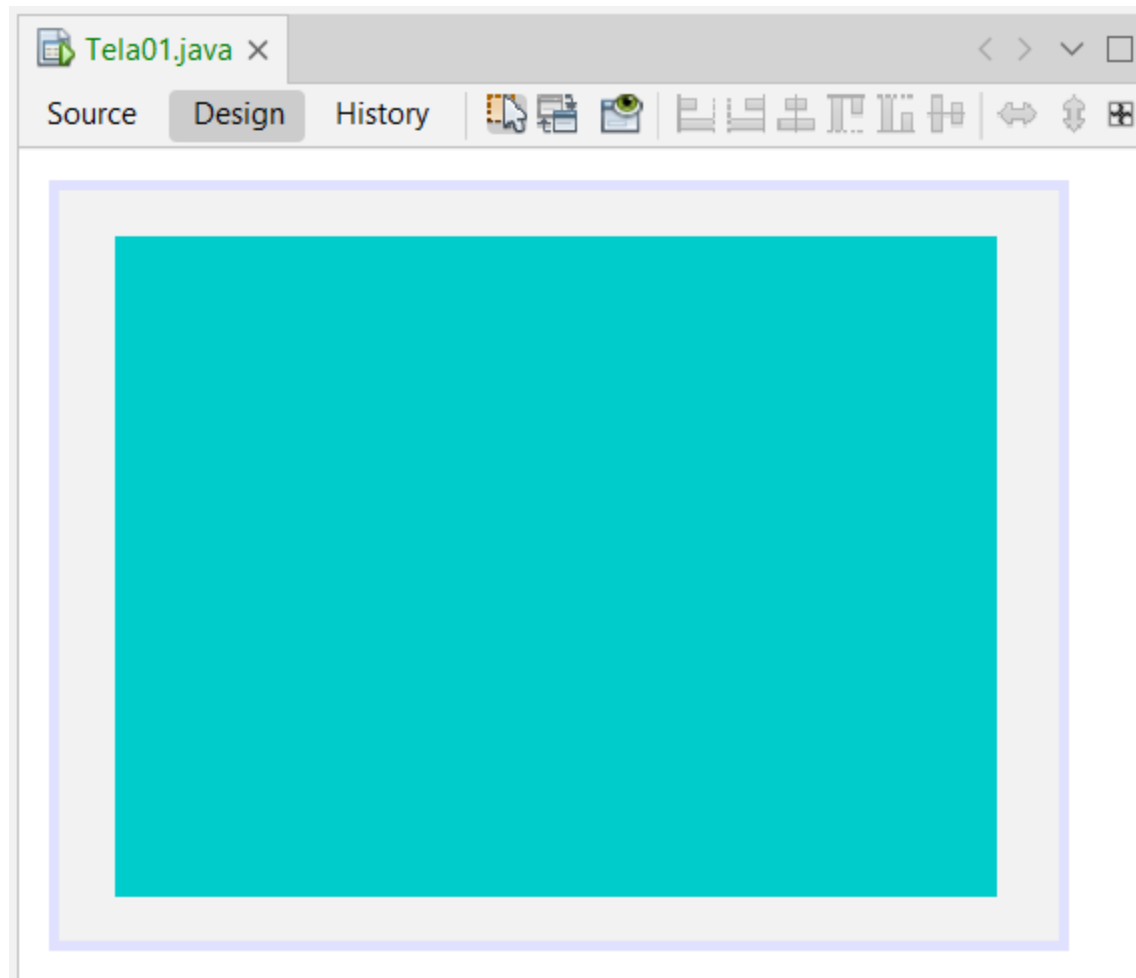
# Java Swing – JPanel - Redimensionar



# Java Swing – JPanel – Mudar Cor Fundo



# Java Swing – JPanel – Mudar Cor Fundo

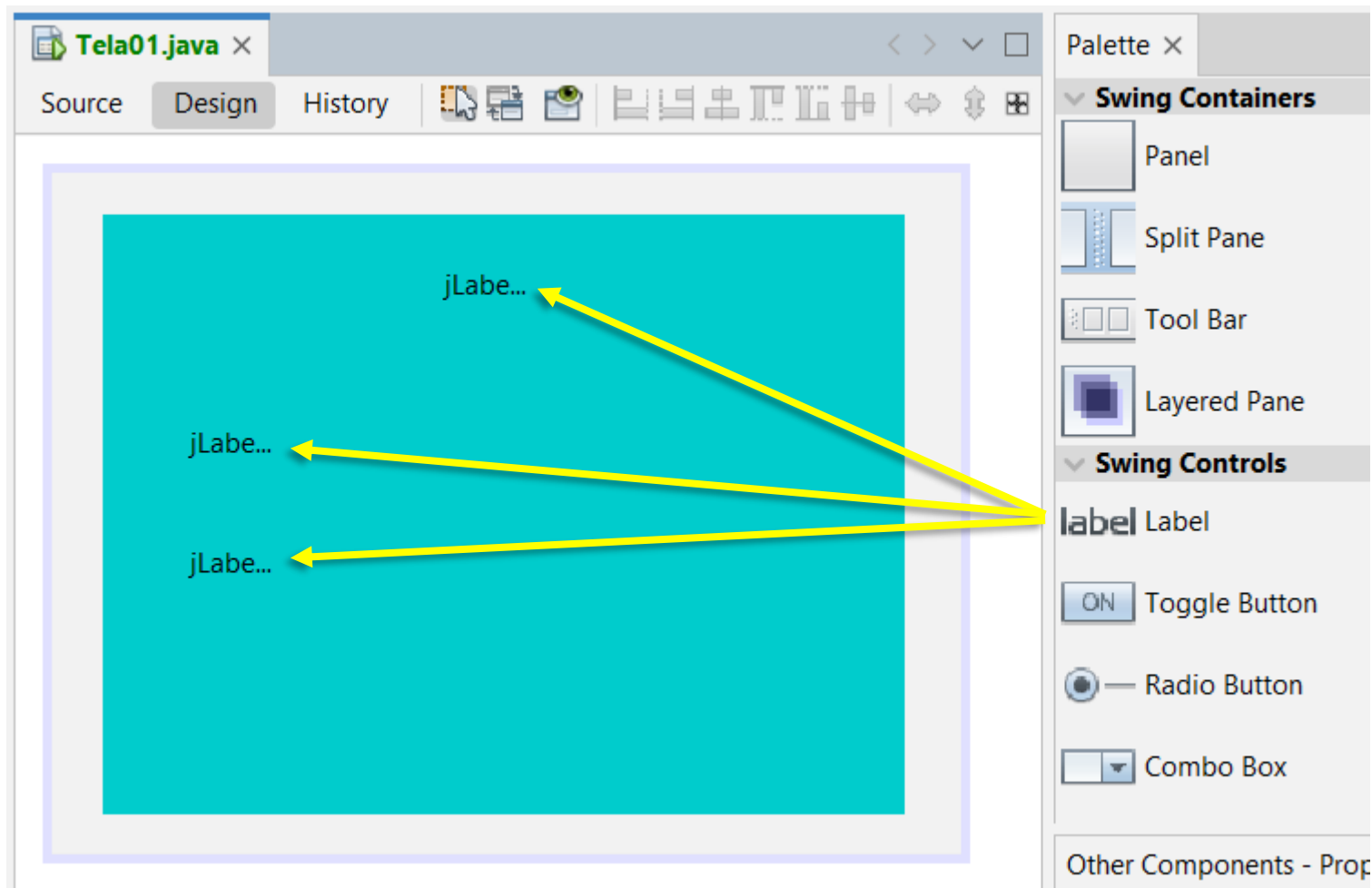


# Java Swing – JLabel

- 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 shows the Java Swing IDE with the following components:

- Design View:** A teal rectangular area representing a window. Inside, there is a grey rectangular component labeled "Tela 01" at the top center. Below it, there are two labels, each labeled "jLabe...".
- Palette:** A panel on the right side of the IDE showing various Swing components. It is divided into two sections: "Swing Containers" and "Swing Controls".
  - Swing Containers:** Panel, Split Pane, Tool Bar, Layered Pane, Tabbed Pane, Scroll Pane, Internal Frame.
  - Swing Controls:** label Label, OK Button.
- Properties Window:** A window titled "jLabel1 [JLabel] - Properties" is open. It has three tabs: "Properties", "Events", and "Code". The "Properties" tab is selected, showing a table of properties for the JLabel component.

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

A yellow arrow points from the "Tela 01" text in the Properties window to the "Tela 01" component in the Design view.



# 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 window 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 Label, OK Button.
- Properties Window:** A window titled "jLabel2 [JLabel] - Properties" showing the properties of the selected component. It has three tabs: Properties, Events, and Code. The Properties tab is active, showing the following properties:

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



# Java Swing – JLabel – Editor Texto

The screenshot shows an IDE window titled 'Tela01.java' with tabs for 'Source', 'Design', and 'History'. The 'Design' tab is active, showing a visual representation of a Java Swing window. The window has a cyan background and a grey panel labeled 'Tela 0..'. Inside this panel is 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**

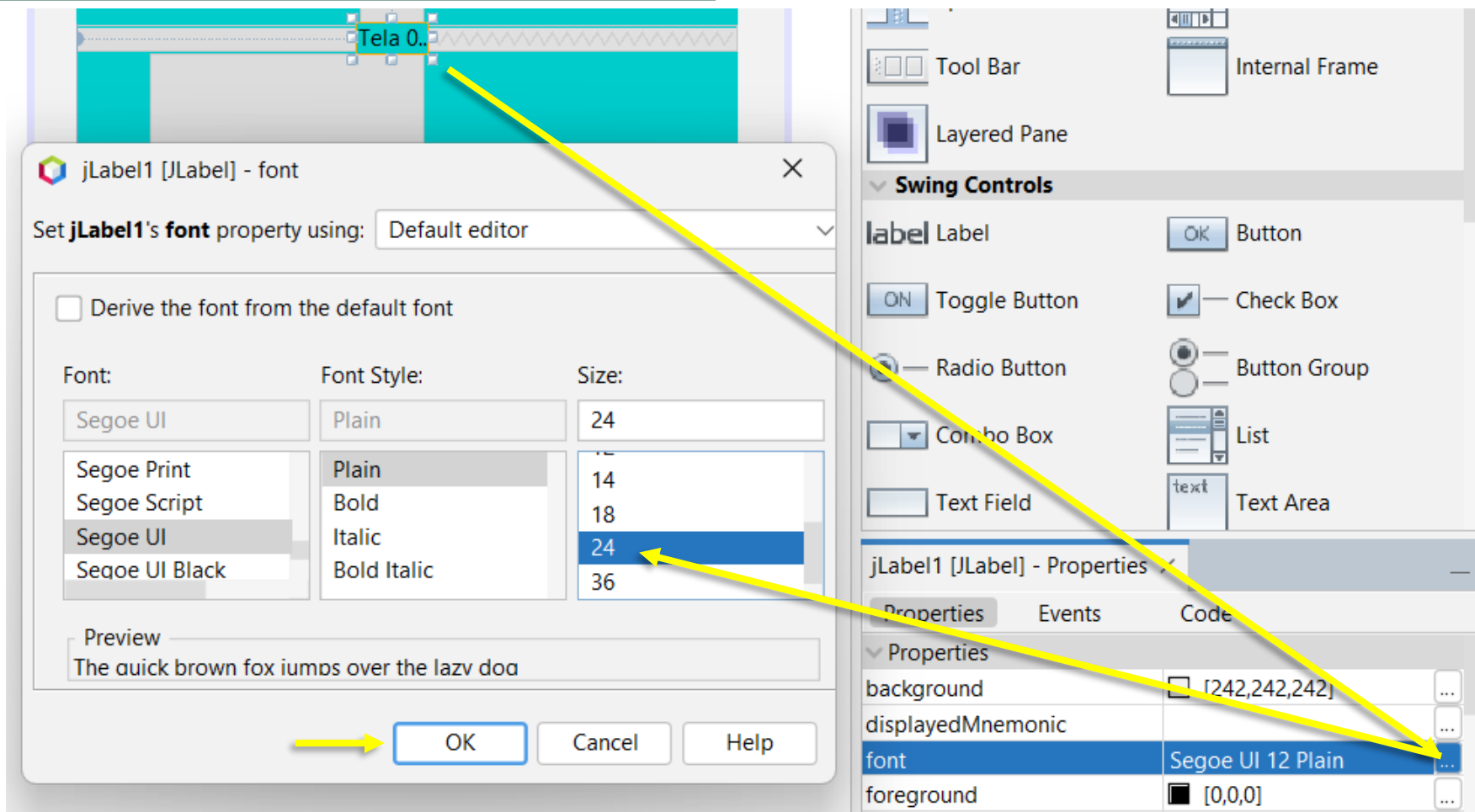
- label Label
- OK Button

**jLabel3 [JLabel] - Properties**

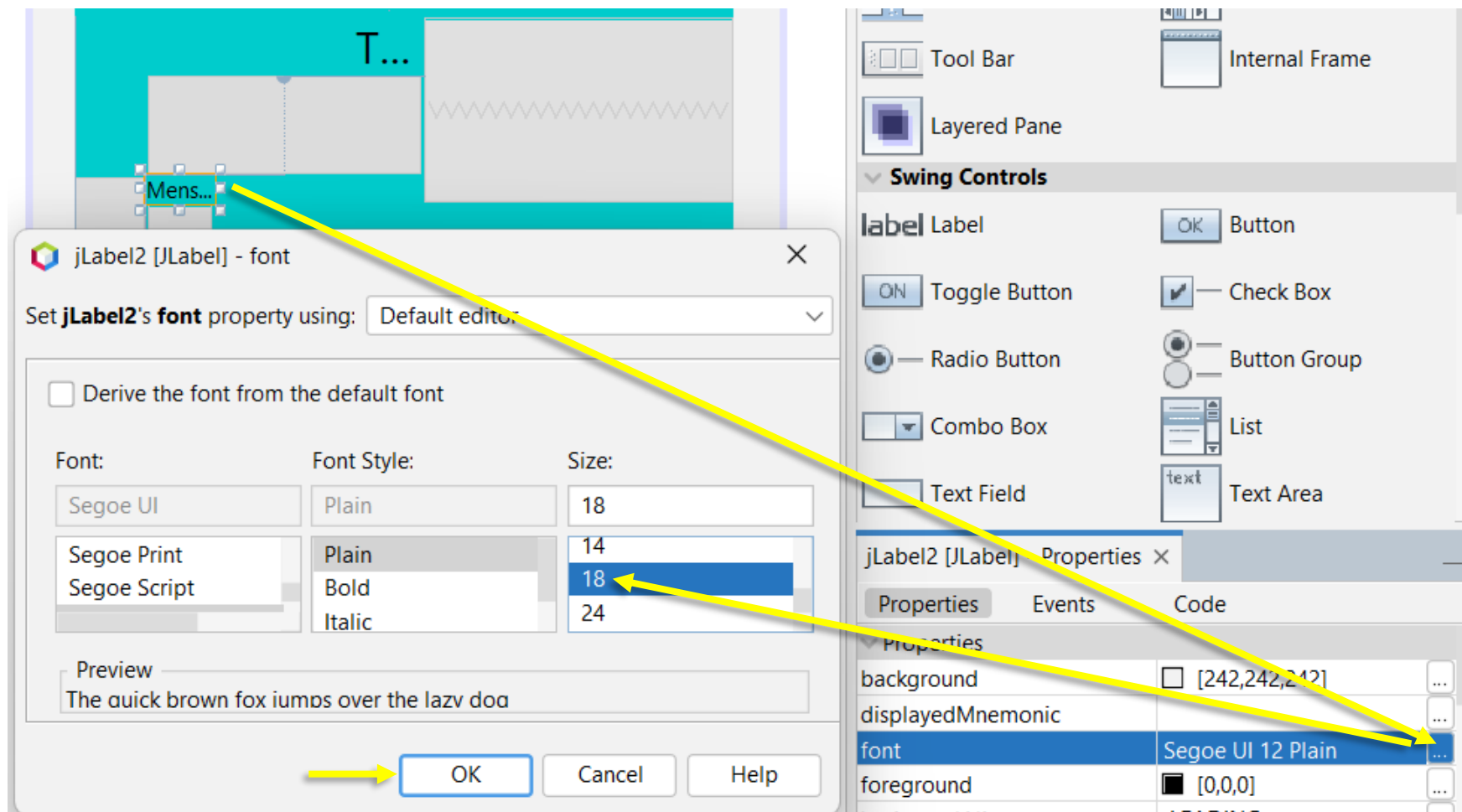
Properties	Events	Code
icon		<none>
labelFor		<none>
<b>text</b>		Mensagem 2:
toolTipText		



# Java Swing – JLabel – Aumentar Fonte



# 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 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" and "Segoe Print". The "Font Style:" column has a list with "Plain" and "Bold". The "Size:" column has a list with "18" and "24". The "18" option is selected. Below these columns is a "Preview" section with the text "The quick brown fox jumps over the lazy dog". At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help". A yellow arrow points from the "OK" button to the "font" property in the "jLabel3 [JLabel] - Properties" window. The "jLabel3 [JLabel] - Properties" window has three tabs: "Properties", "Events", and "Code". The "Properties" tab is active. It shows a list of properties: "background", "DisplayedMnemonic", "font", "foreground", "horizontalAlignment", and "icon". The "font" property is selected and its value is "Segoe UI 12 Plain". A yellow arrow points from the "font" property in the "jLabel3 [JLabel] - Properties" window to the "font" property in the "jLabel3 [JLabel] - font" dialog box.

Font: Segoe UI, Segoe Print

Font Style: Plain, Bold

Size: 18, 24

Preview: The quick brown fox jumps over the lazy dog

OK, Cancel, Help

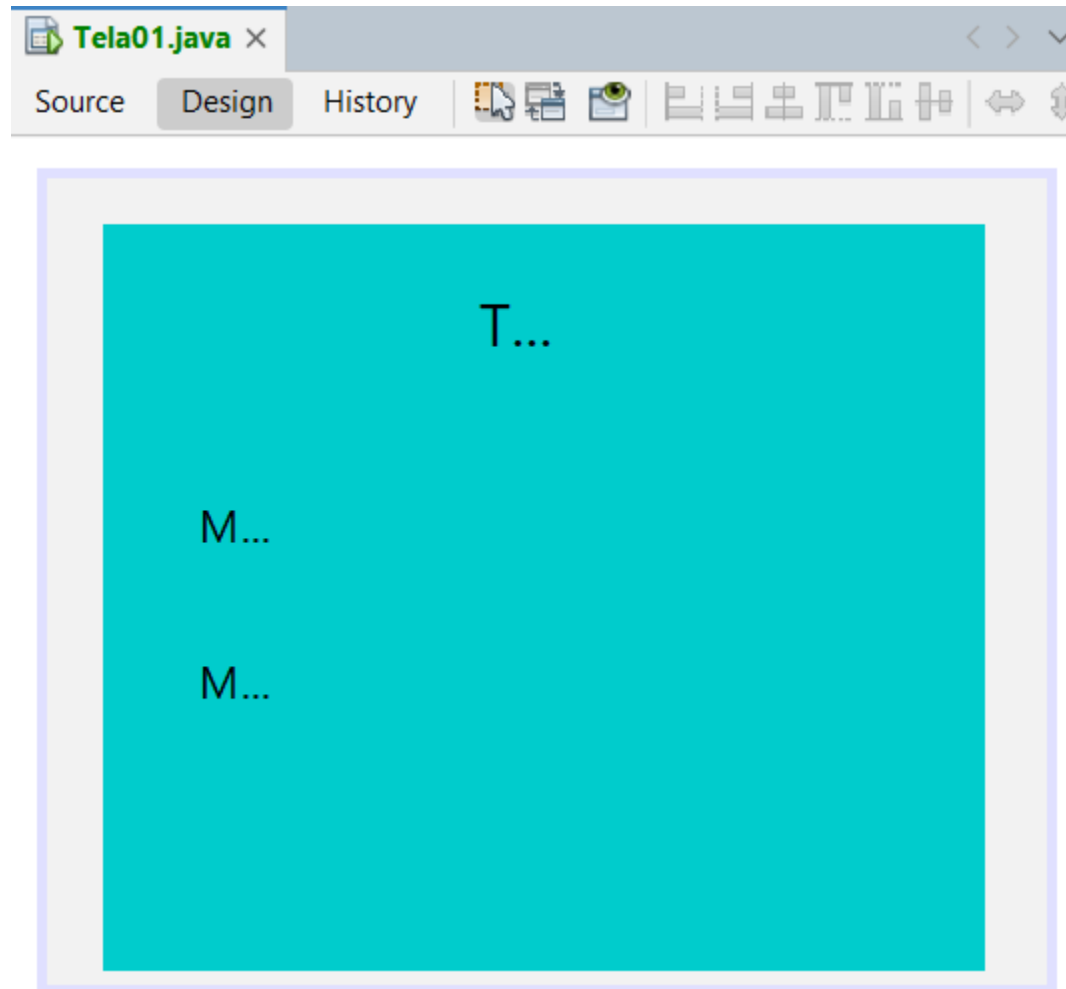
Swing Controls: Label, Toggle Button, Radio Button, Combo Box, Text Field, Button, Check Box, Button Group, List, Text Area

jLabel3 [JLabel] - Properties

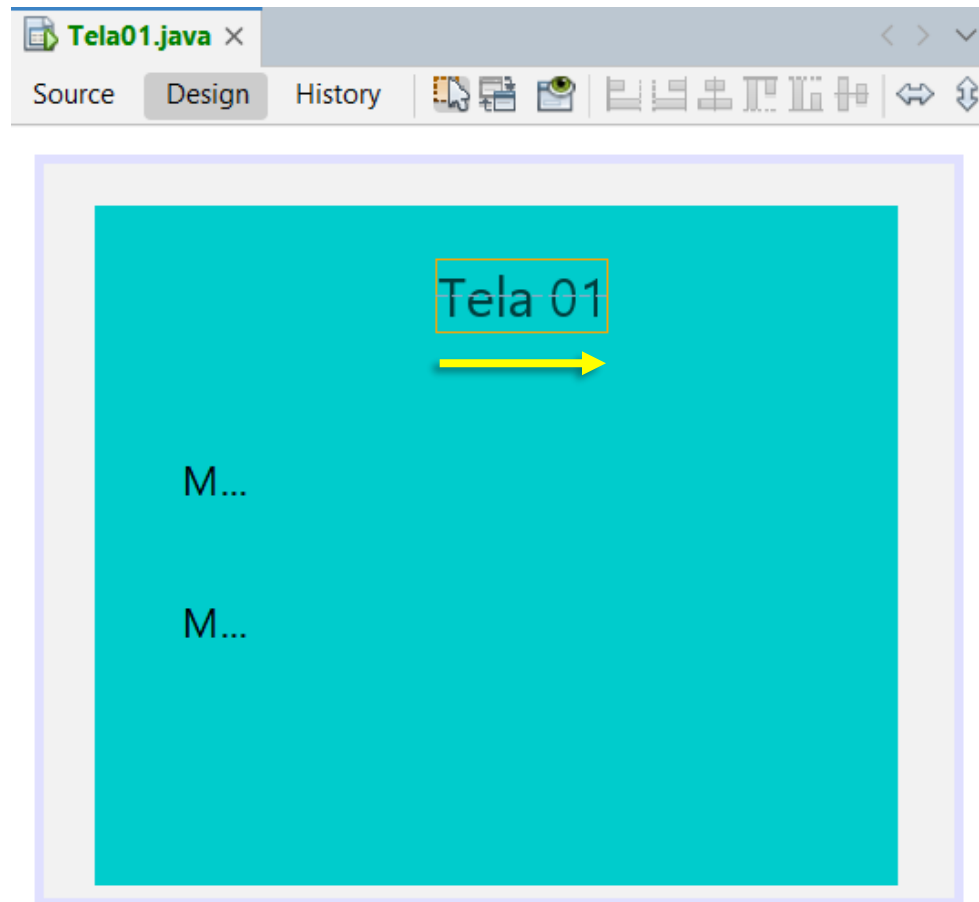
Properties	Events	Code
background		[242,242,242]
DisplayedMnemonic		
font		Segoe UI 12 Plain
foreground		[0,0,0]
horizontalAlignment		LEADING
icon		<none>



# Java Swing – JLabel – Aumentar Fonte

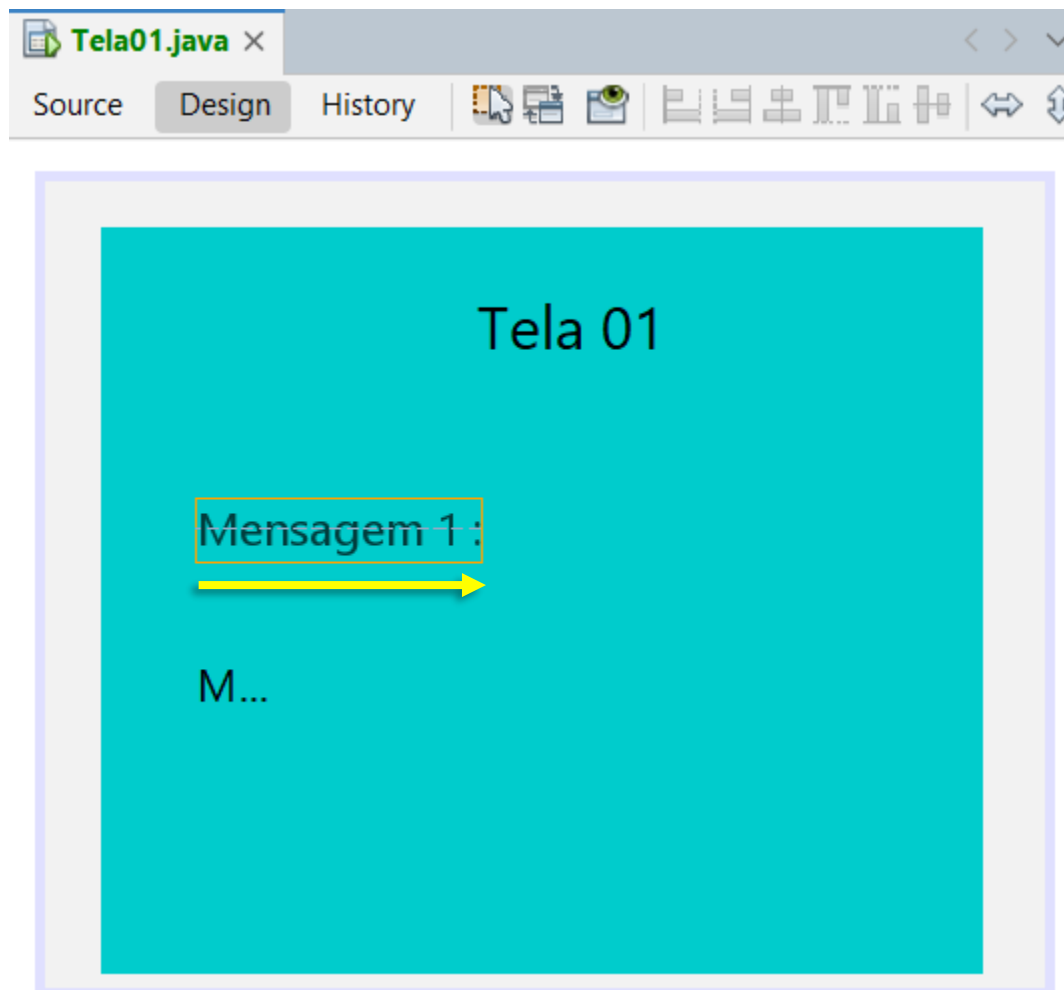


# Java Swing – JLabel – Redimensionar

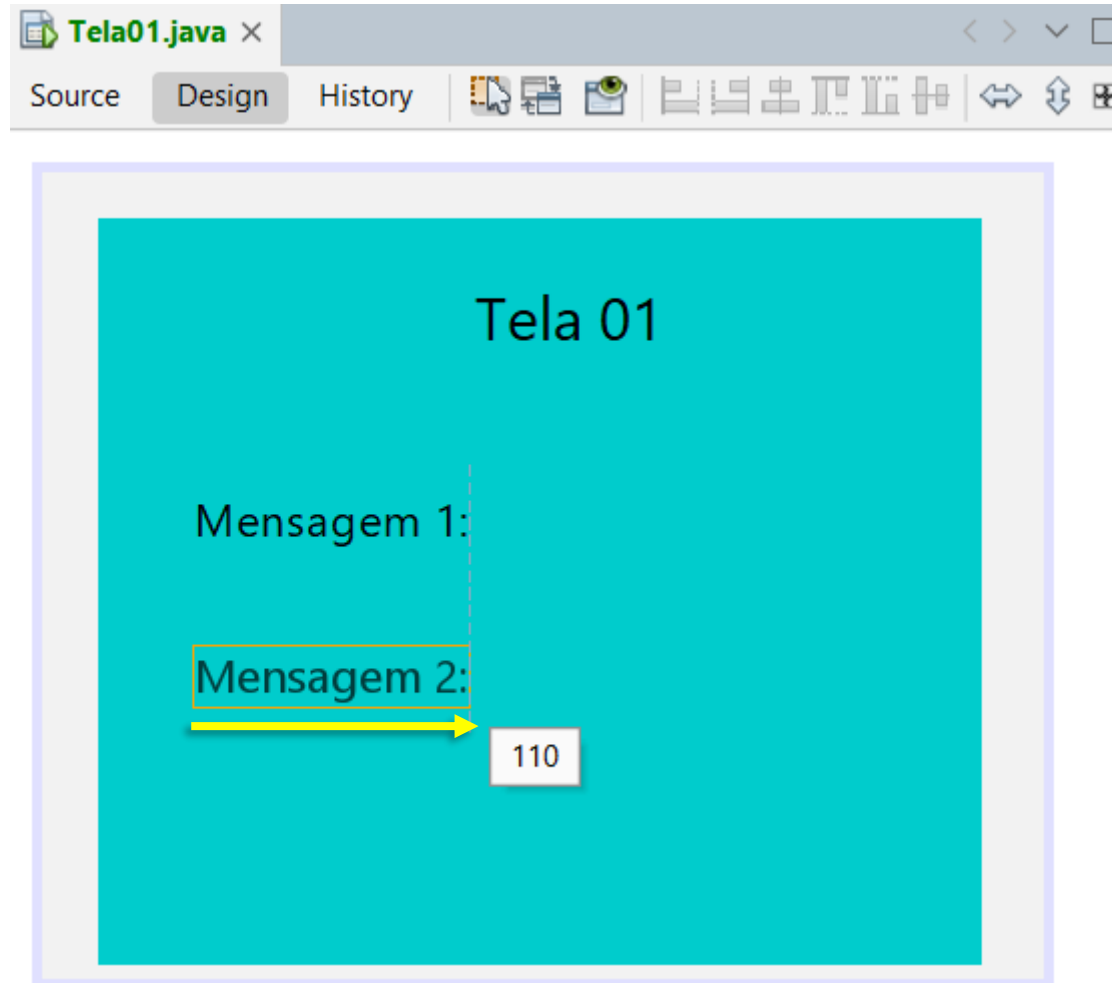




# Java Swing – JLabel – Redimensionar



# Java Swing – JLabel – Redimensionar



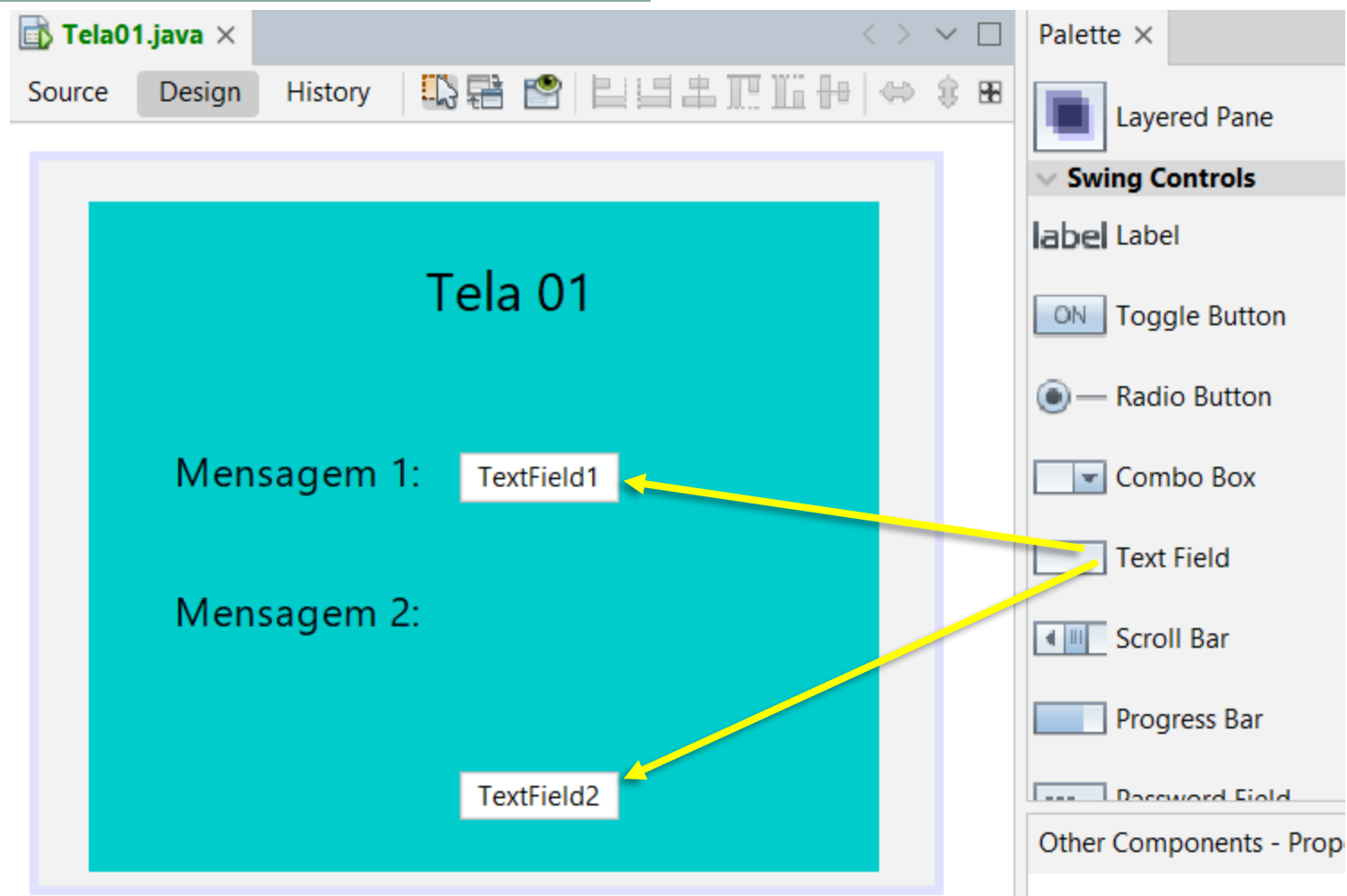
# Java Swing – JTextField

---

- 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 image shows a screenshot of an IDE window titled 'Tela01.java' in Design mode. The main window contains a cyan background with the text 'Tela 01' at the top. Below it, 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 'text' property in the 'jTextField1 [JTextField] - Properties' panel on the right.

**Palette**

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

**jTextField1 [JTextField] - Properties**

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



# Java Swing – JTextField – Editor Texto

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:', and a text field. A yellow arrow points from the text field to the 'text' property in the Properties window.

**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		
<b>Other Properties</b>		
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 (likely NetBeans) with a Java Swing window titled "Tela 01" in the Design view. The window has a cyan background and contains two labels, "Mensagem 1:" and "Mensagem 2:", each followed by a text field. A yellow arrow points from the text field under "Mensagem 2:" to the "Code" tab of the "jTextField2 [JTextField] - Properties" window. In the "Code" tab, the "Variable Name" is set to "txtMensagemFinal".

**Swing Controls Palette:**

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

**jTextField2 [JTextField] - Properties**

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





# 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 for configuring the font of the 'txtMensagem1' component. It has 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. The 'Font Style:' section shows 'Plain' selected. The 'Size:' section shows '18' selected. A preview of the text 'The quick brown fox jumps over the lazy dog' is shown. The 'OK' button is highlighted with a yellow arrow. The right window is titled 'txtMensagem1 [JTextField] - Properties' and shows the 'Properties' tab. It lists various properties: 'columns' (0), 'document' (<default>), 'font' (Segoe UI 12 Plain), 'foreground' (black [0.0.0]), 'horizontalAlignment' (LEADING), and 'text'. The 'font' property is highlighted with a yellow arrow. A yellow arrow also points from the 'font' property in the Properties 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.

Mensagem 1:

txtMensagem1 [JTextField] - font

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

OK Cancel Help

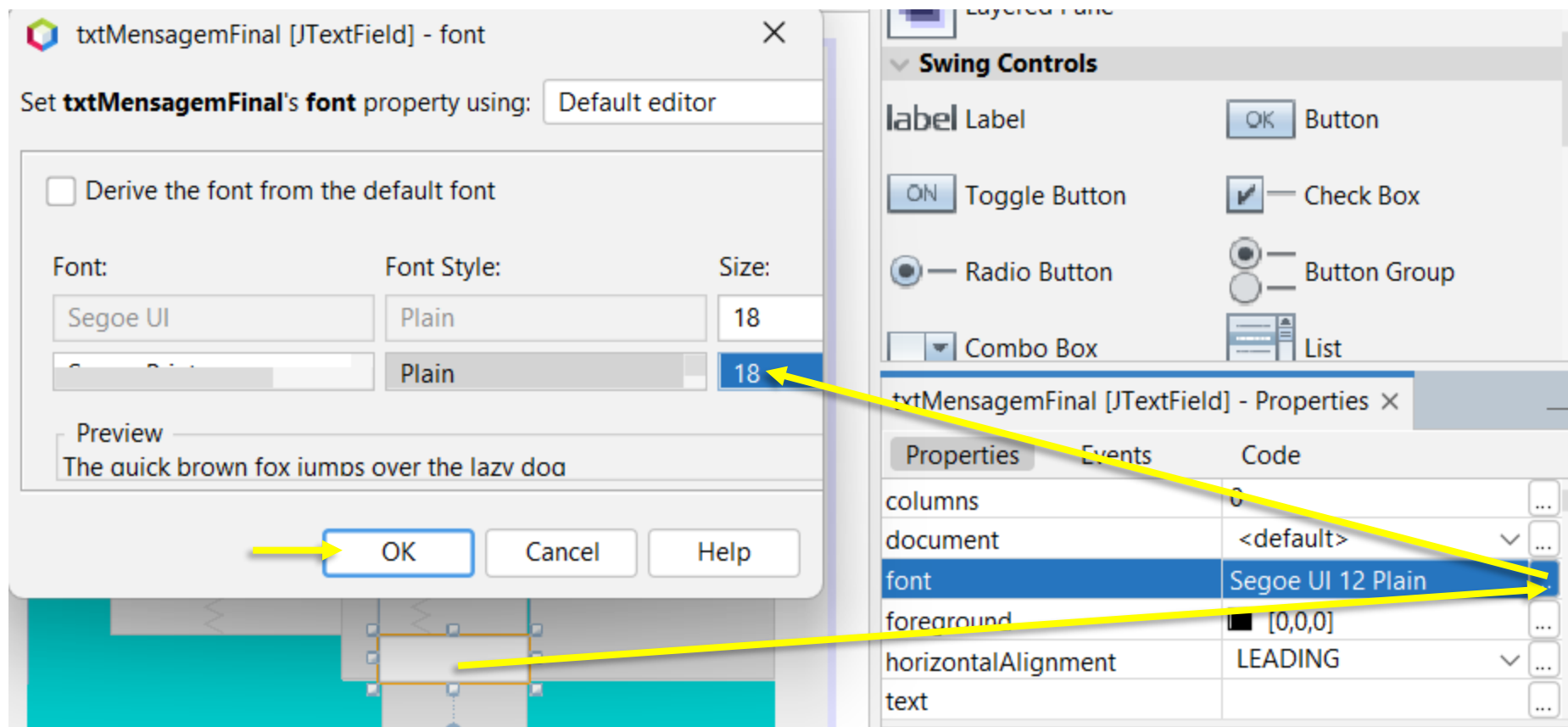
txtMensagem1 [JTextField] - Properties

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

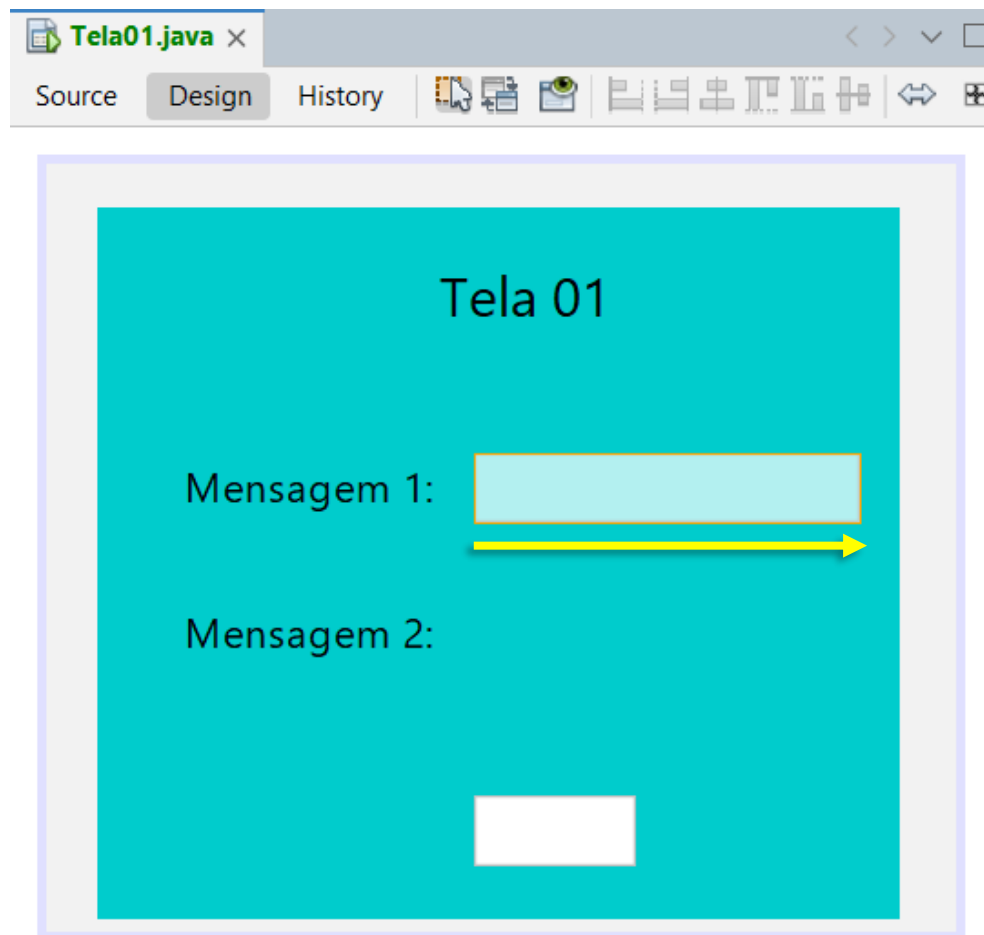
**font**  
(java.awt.Font) The font for the component.



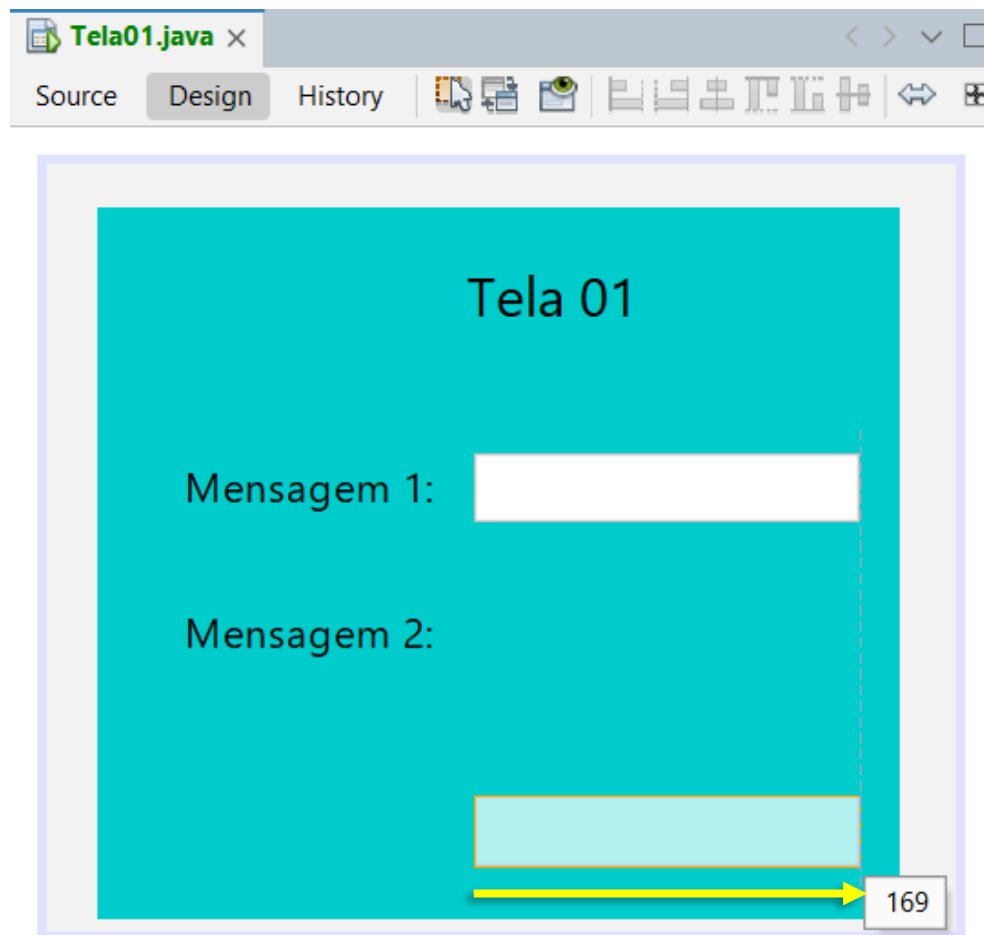
# Java Swing – JTextField – Aument. Fonte



# 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



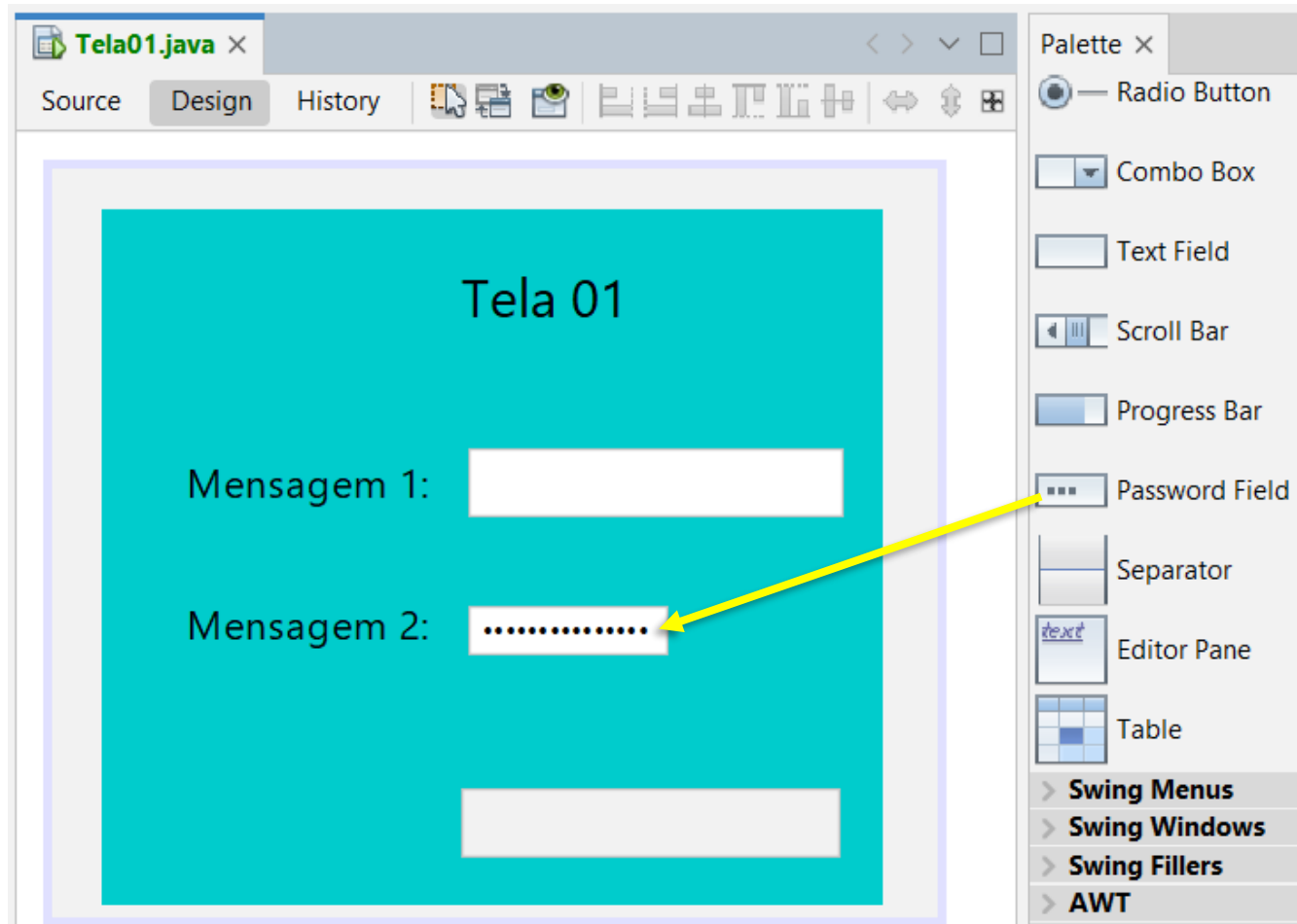
# Java Swing – JPasswordField

---

- 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. The main window, titled 'Tela01.java', is in the 'Design' tab. It shows a graphical user interface with a cyan background and the title 'Tela 01'. There are two labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text input field. A yellow arrow points from the 'Mensagem 2:' input field to the 'Properties' tab of the 'jPasswordField1 [JPasswordField]' component in the right-hand 'Properties' window.

The 'Properties' window for 'jPasswordField1 [JPasswordField]' shows the following properties:

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

The 'Palette' window on the right lists various Swing components, including Radio Button, Button Group, Combo Box, Text Field, Scroll Bar, Progress Bar, Password Field, Separator, Button Group, List, Text Area, Slider, Formatted Field, Spinner, and Text Pane.





# 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. Inside the window, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. Below 'Mensagem 2:', there is a JPasswordField component. A yellow arrow points from the JPasswordField to the 'txpMensagem2 [JPasswordField] - Properties' window. The Properties window shows the 'Code' tab with the following information:

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

Below the table, the text 'txpMensagem2 [JPasswordField]' is displayed.



# Java Swing – JPasswordField – Aum. F.

The image shows a Java Swing application window titled "Tela01.java" with a component named "txpMensagem2 [JPasswordField]". A dialog box titled "txpMensagem2 [JPasswordField] - font" is open, allowing the user to set the font for the component. The dialog has a "Default editor" button and a checkbox "Derive the font from the default font" which is unchecked. The font settings are as follows:

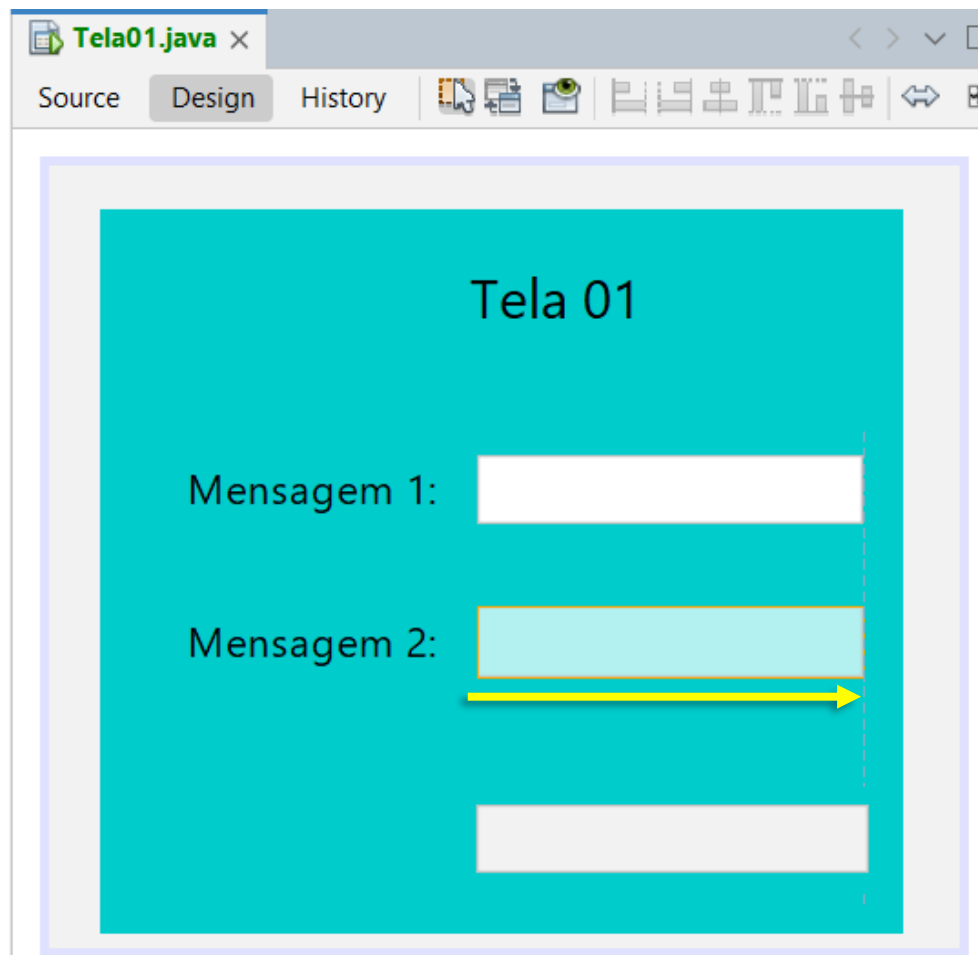
Font:	Font Style:	Size:
Segoe UI	Plain	18
	Plain	18

Yellow arrows point from the "OK" button in the dialog to the "font" property in the "txpMensagem2 [JPasswordField] - Properties" window. The properties window shows the following settings:

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



# Java Swing – JPasswordField – Redimen.



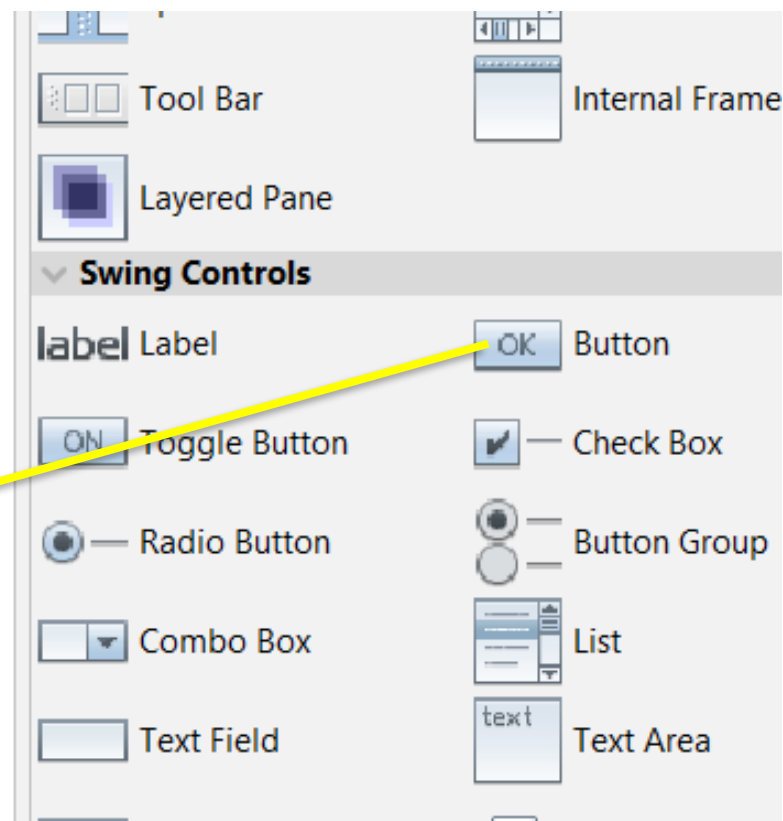
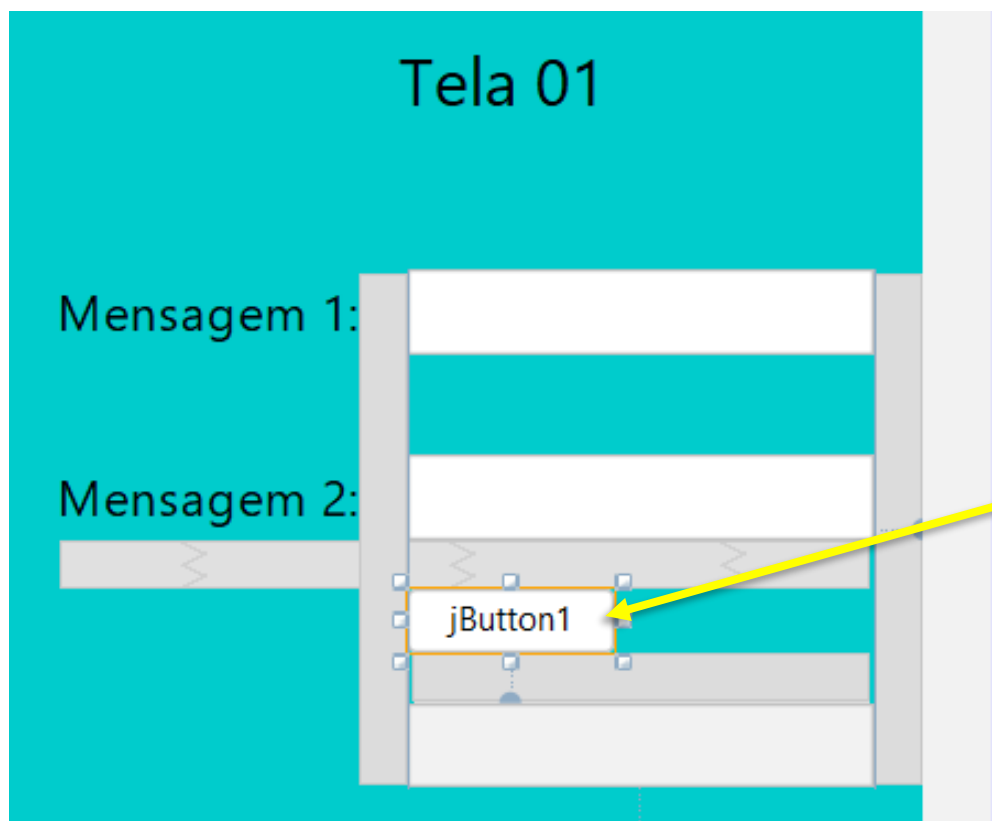
# Java Swing – JButton

---

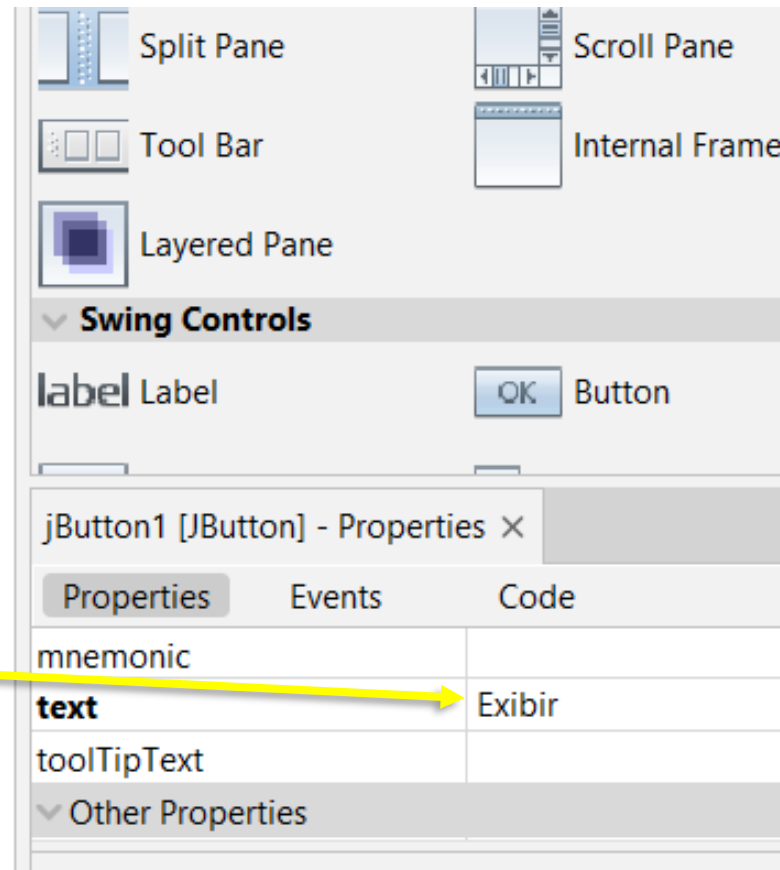
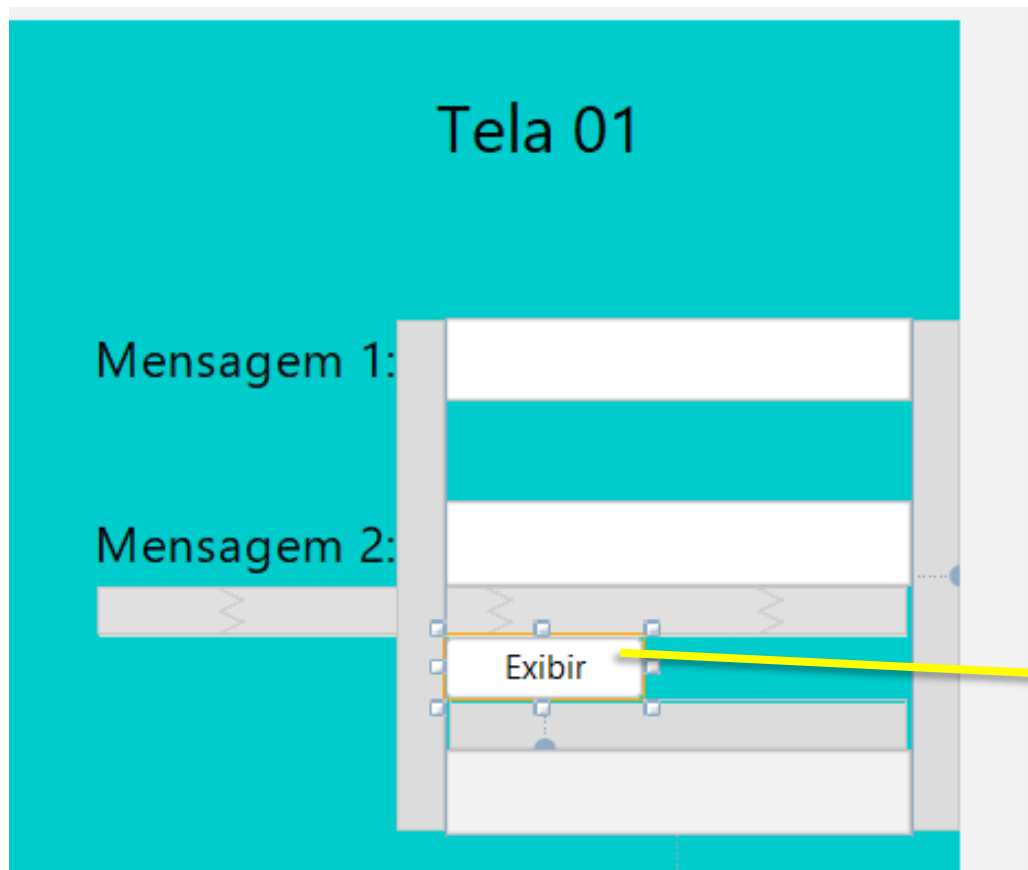
- A classe **JButton** permite **definir botões gráficos** que **pode ser adicionados** a um **outro componente** (frame, painel etc.).
- Um **botão** pode **ser criado** com apenas um **texto e/ou com ícones** para tornar o ambiente mais intuitivo.



# 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 is titled "jButton1 [JButton] - font" and contains a "Default editor" tab. It has a checkbox "Derive the font from the default font" which is unchecked. Below this are three fields: "Font:" (Segoe UI), "Font Style:" (Plain), and "Size:" (18). A preview area shows the text "The quick brown fox jumps over the lazy dog" in the selected font and size. 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 is titled "jButton1 [JButton] - Properties" and has three tabs: "Properties", "Events", and "Code". The "Properties" tab is active, showing a list of properties. The "font" property is highlighted, and its value is "Segoe UI 12 Plain". A yellow arrow points from the "Size:" field in the left window to the "font" property in the right window. Another yellow arrow points from the "font" property in the right window to the "font" property in the left window.

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

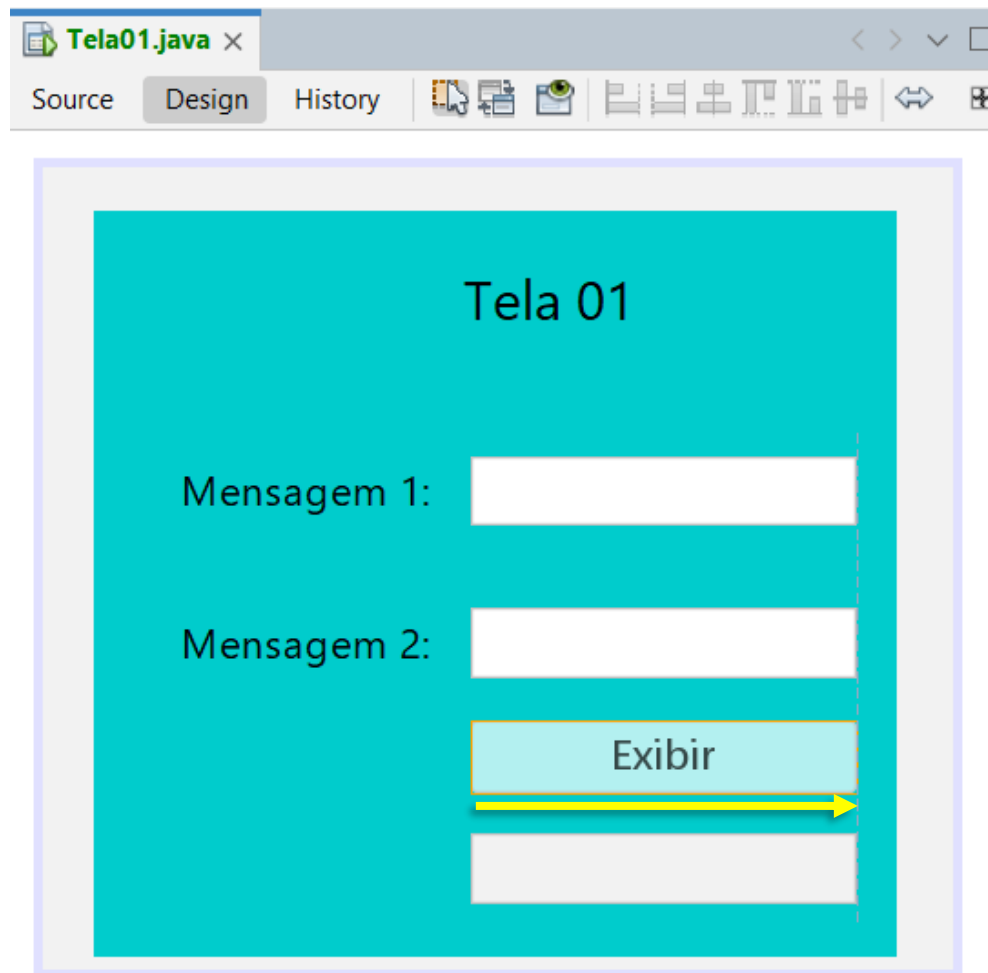
- Label
- Button

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.

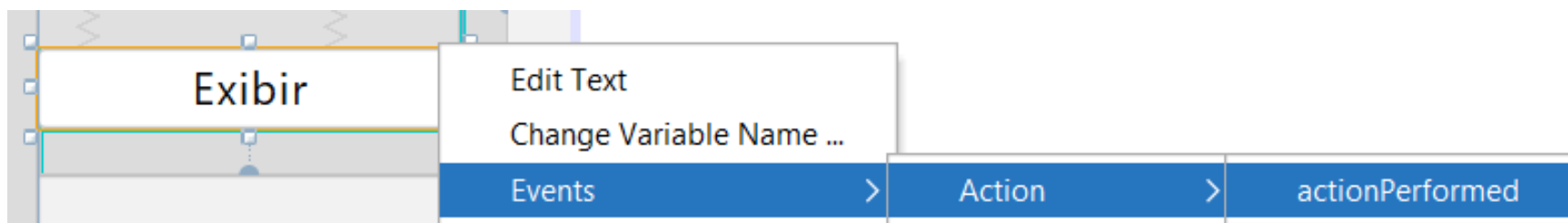
The image shows a Java Swing IDE interface. The main window, titled 'Tela 01', is in Design mode. It contains a cyan background with the text 'Tela 01'. Below this, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. To the right of these labels is a vertical stack of components. The top component is a text area. Below it is a button 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:



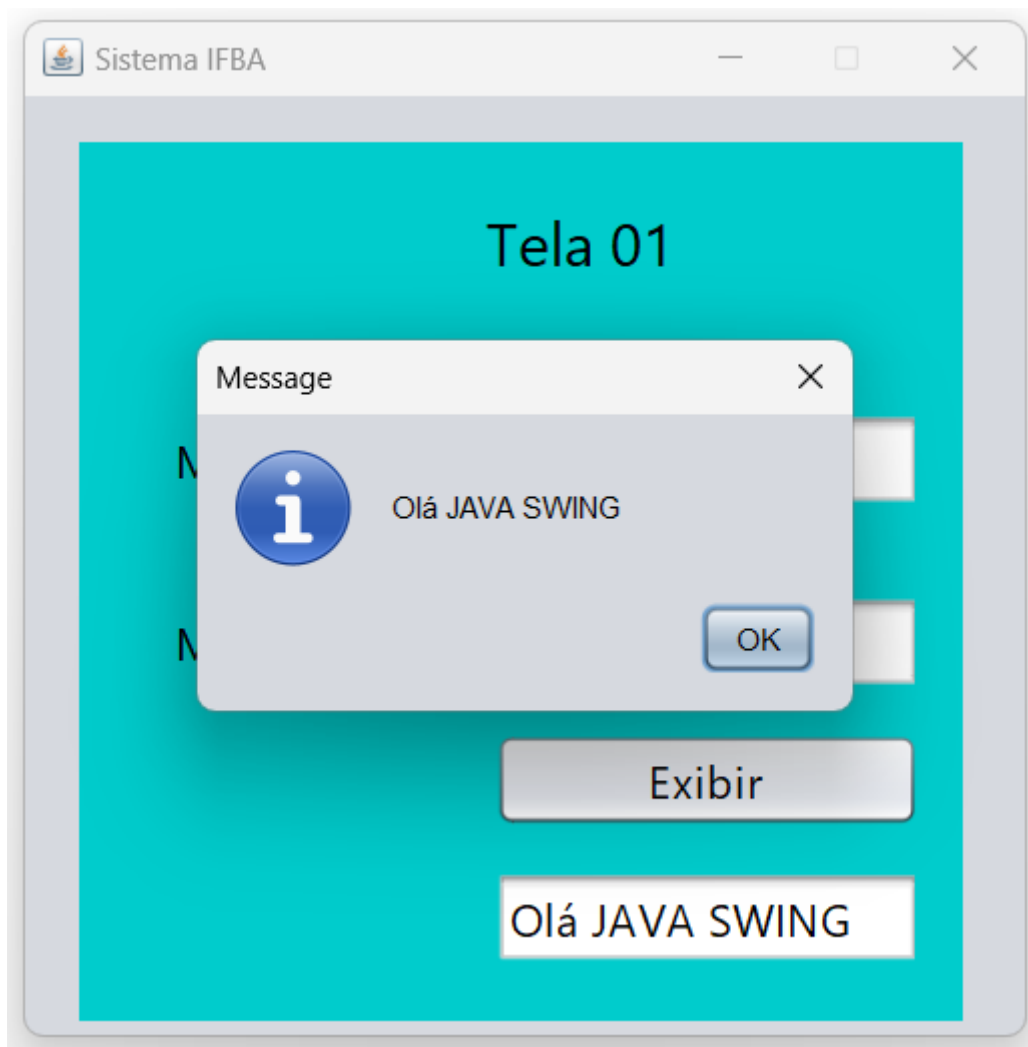
# Java Swing – JButton – Execução



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 labels: "Mensagem 1:" and "Mensagem 2:". Next to "Mensagem 1:" is a text input field containing the text "Olá". Next to "Mensagem 2:" is a text input field containing ten asterisks "\*\*\*\*\*". Below these fields is a gray button with the text "Exibir". At the bottom of the cyan area, there is an empty white rectangular box.



# Java Swing – JButton – Execução





# Java Swing – JButton – Execução



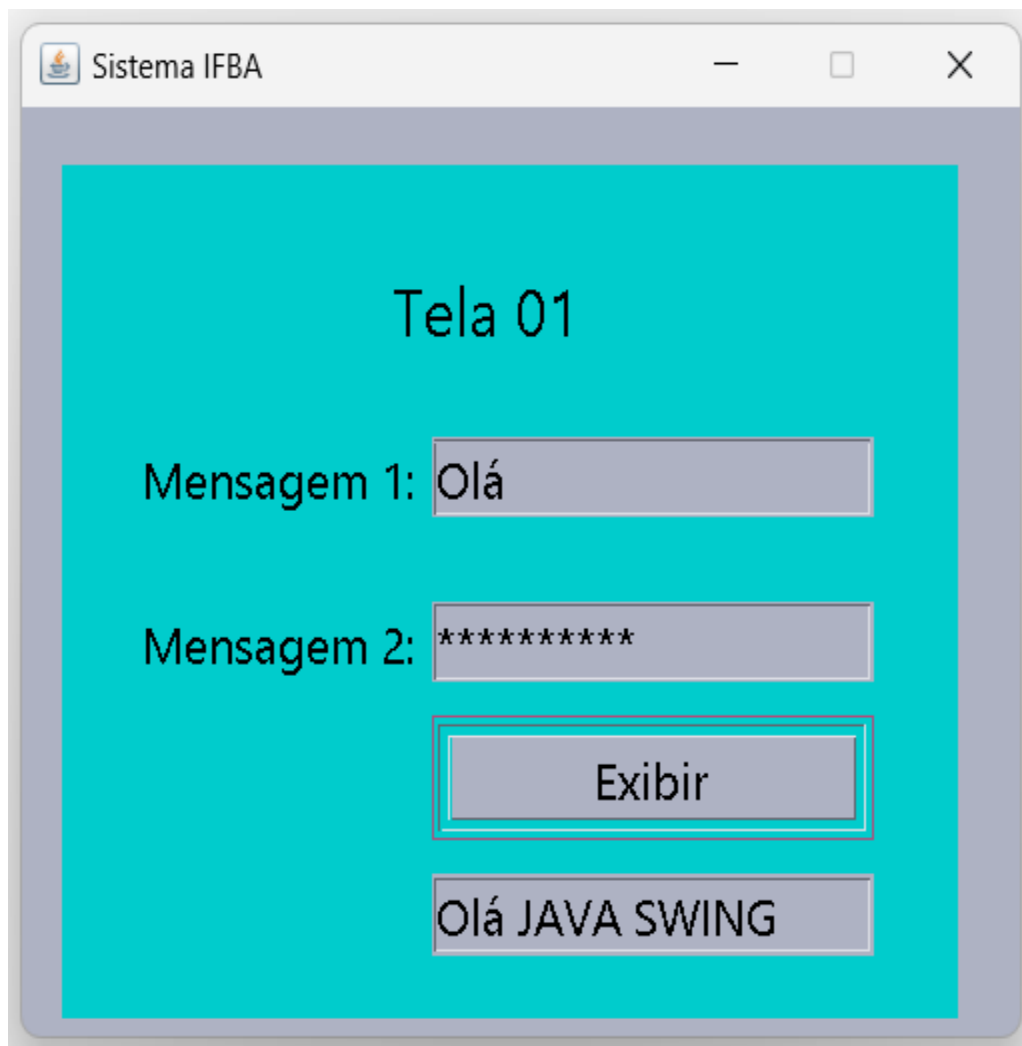
# Java Swing – Troca de Tema (Aparência)

```
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)

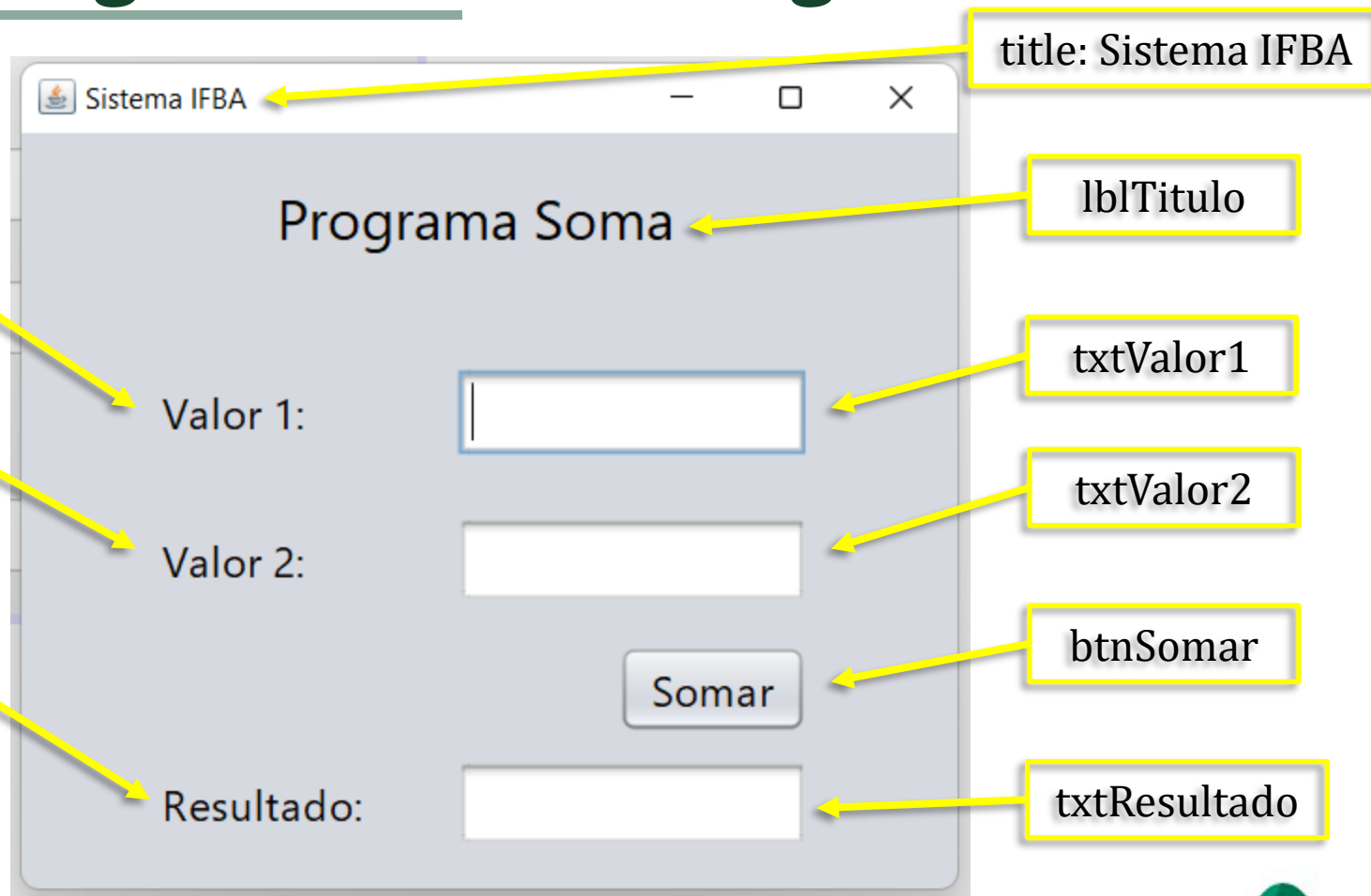


# Java Swing – Lista de Tema (Aparência)

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



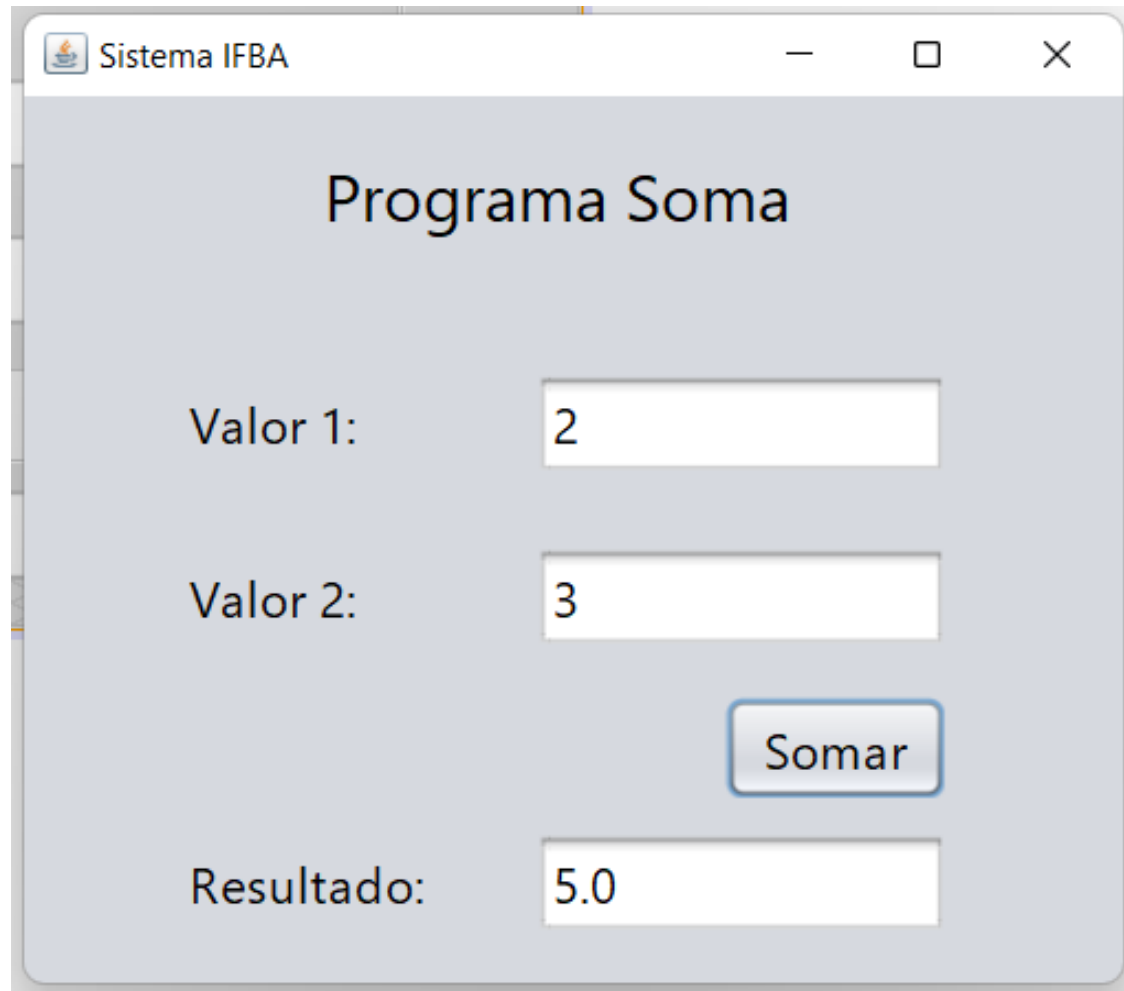
# Java Swing – Exercício – Prog. Soma

```
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: 2

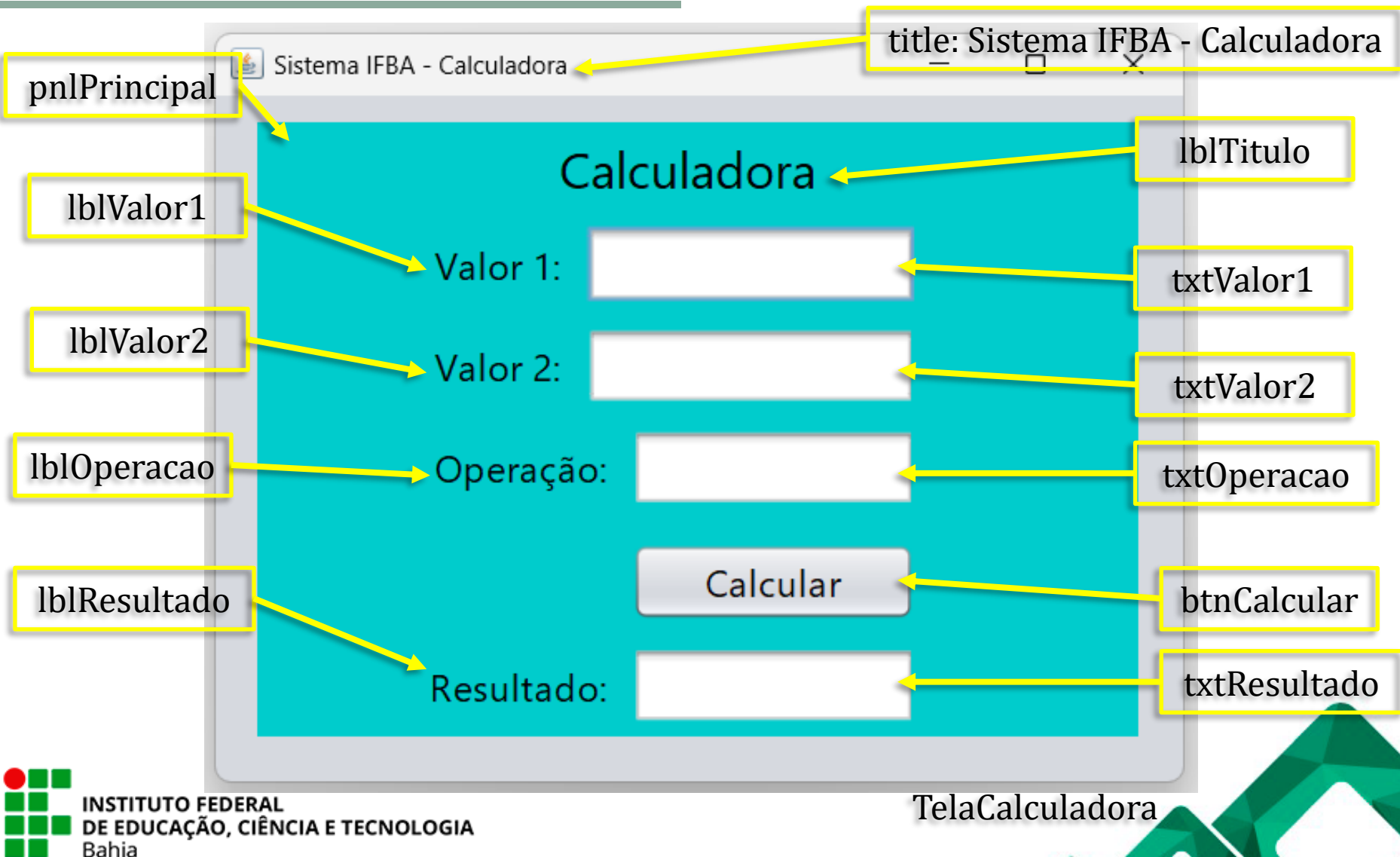
Valor 2: 3

Somar

Resultado: 5.0



# Java Swing – Exercício – Prog. Calc.





# 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.

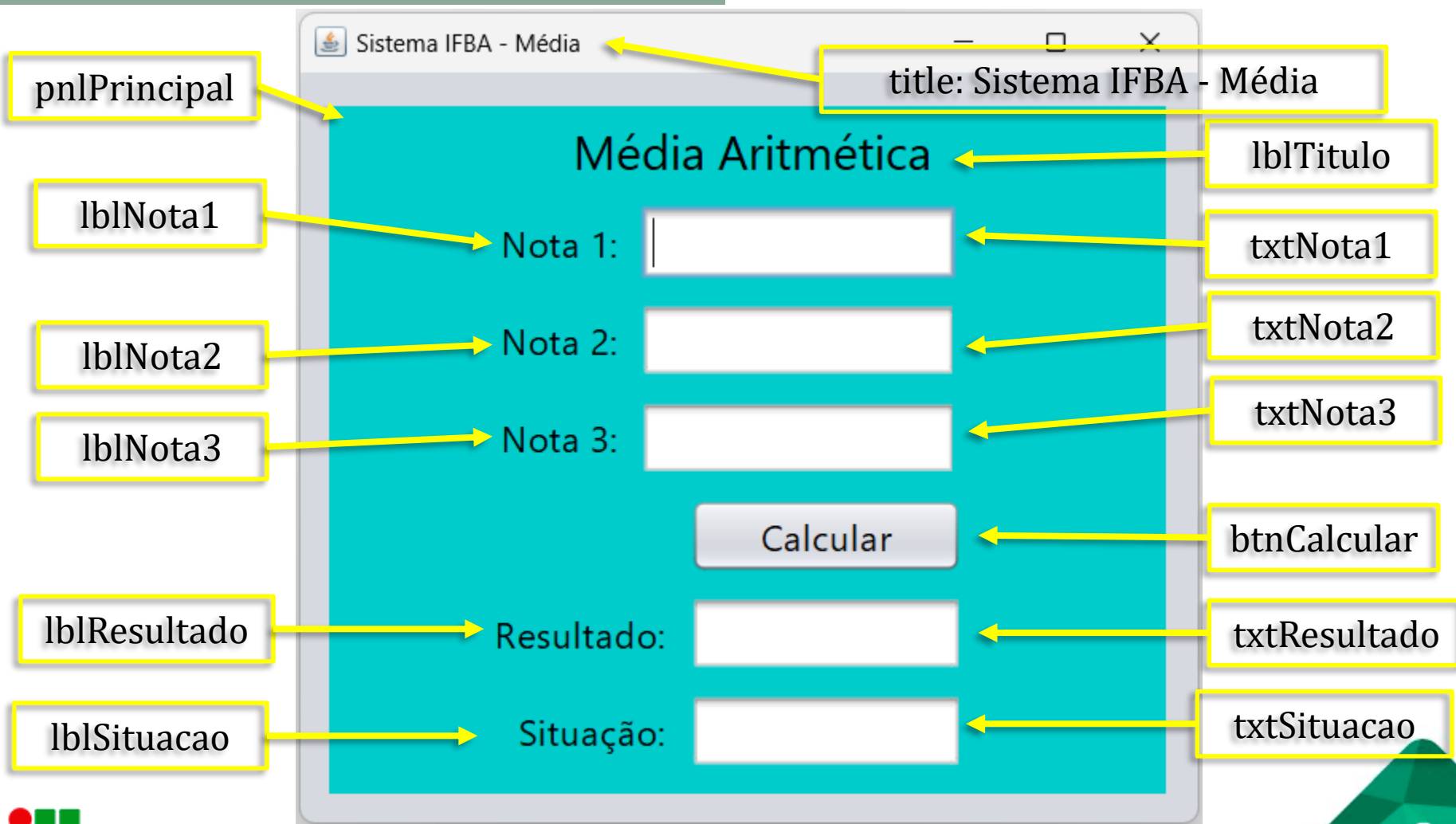
Execução

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 dropdown menu showing the multiplication symbol (\*). A yellow box highlights the operation menu, and a yellow arrow points to the selected \* option. The other visible options are +, -, and /.
- Calcular:** A button with the text "Calcular".
- Resultado:** A text input field containing the result "50.0".



# 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

---

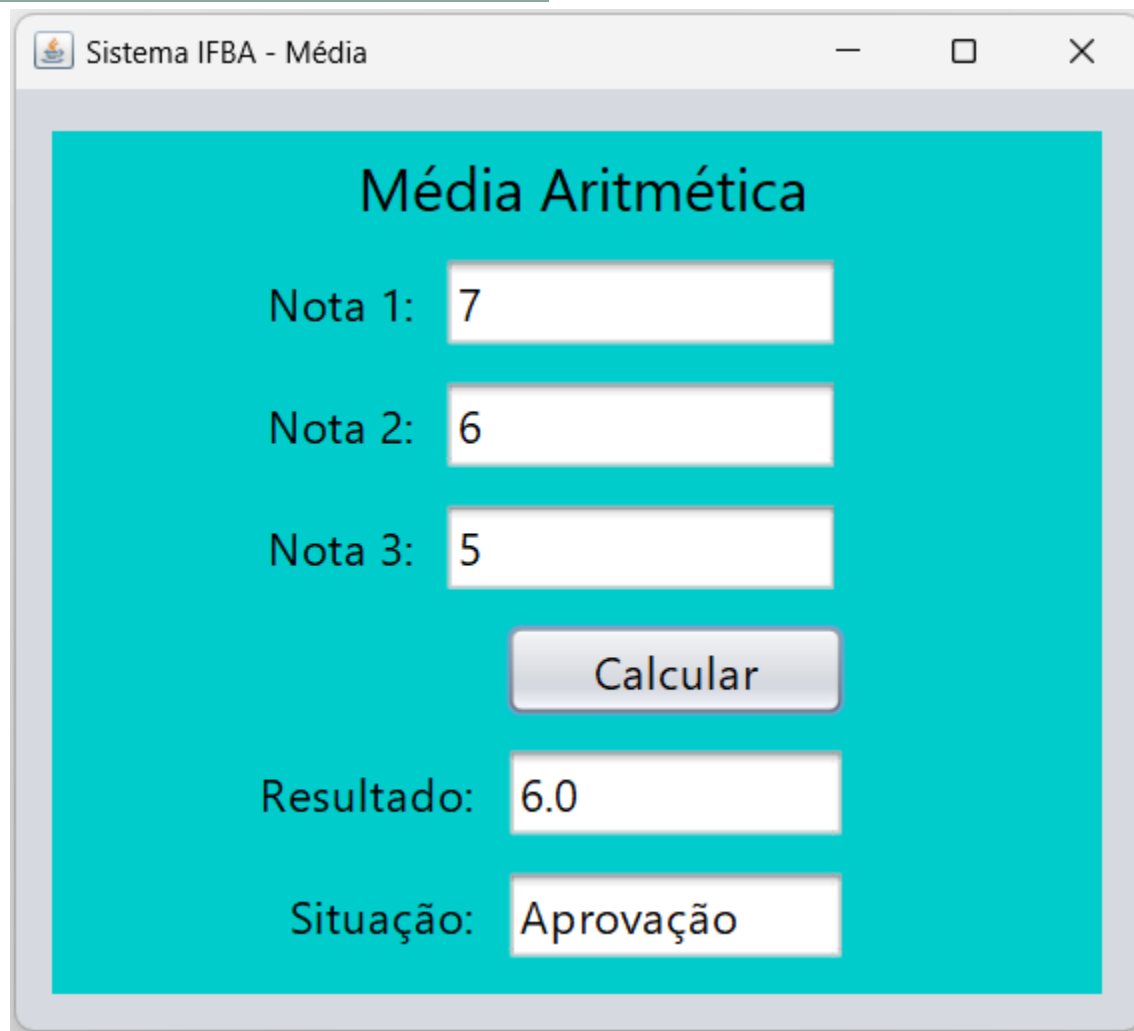
Arredondar o valor da variável “media” do tipo double para duas casas decimais:

```
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:

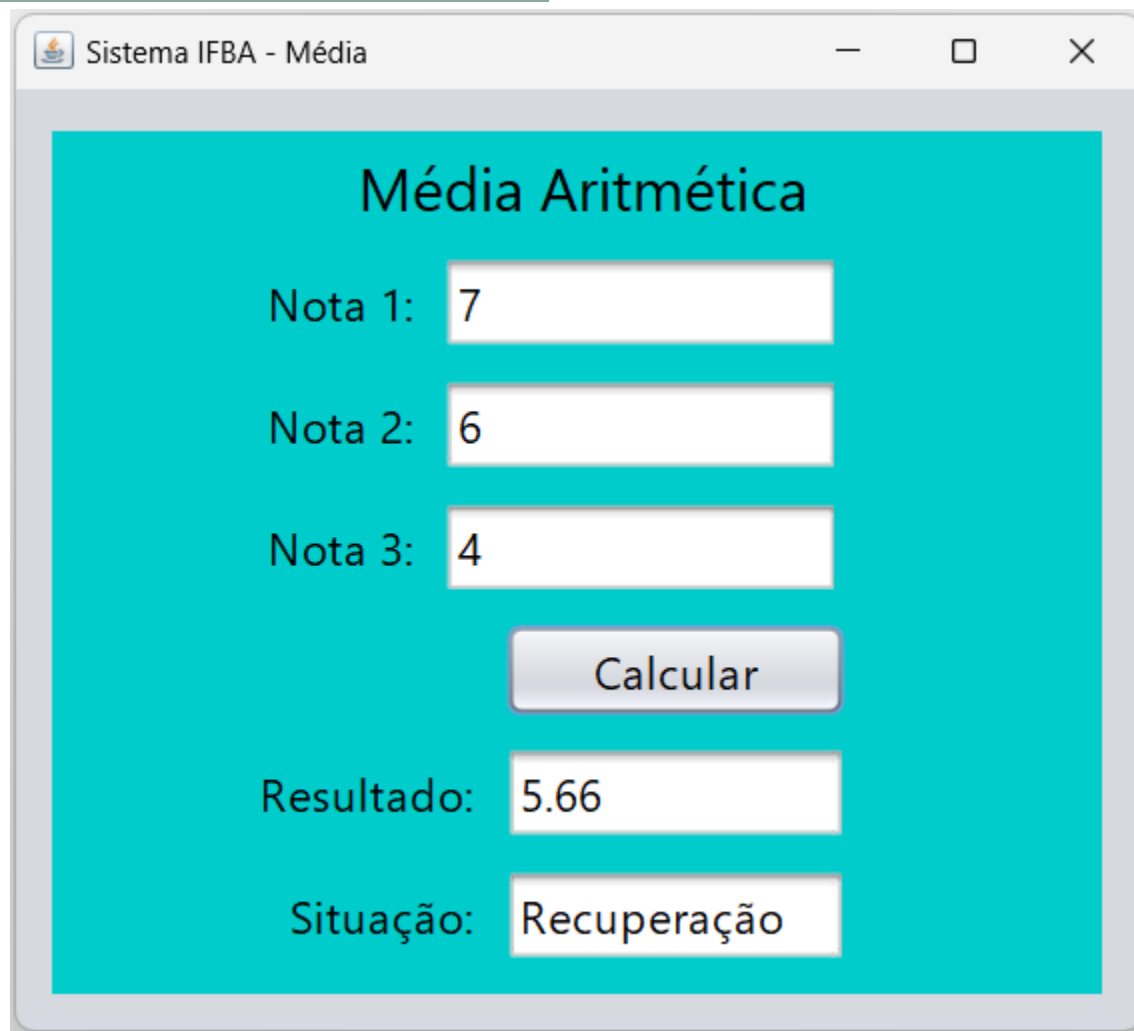
- Média Aritmética**: The title of the application.
- Nota 1:** A text label followed by a text input field containing the value "7".
- Nota 2:** A text label followed by a text input field containing the value "6".
- Nota 3:** A text label followed by a text input field containing the value "5".
- Calcular**: 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

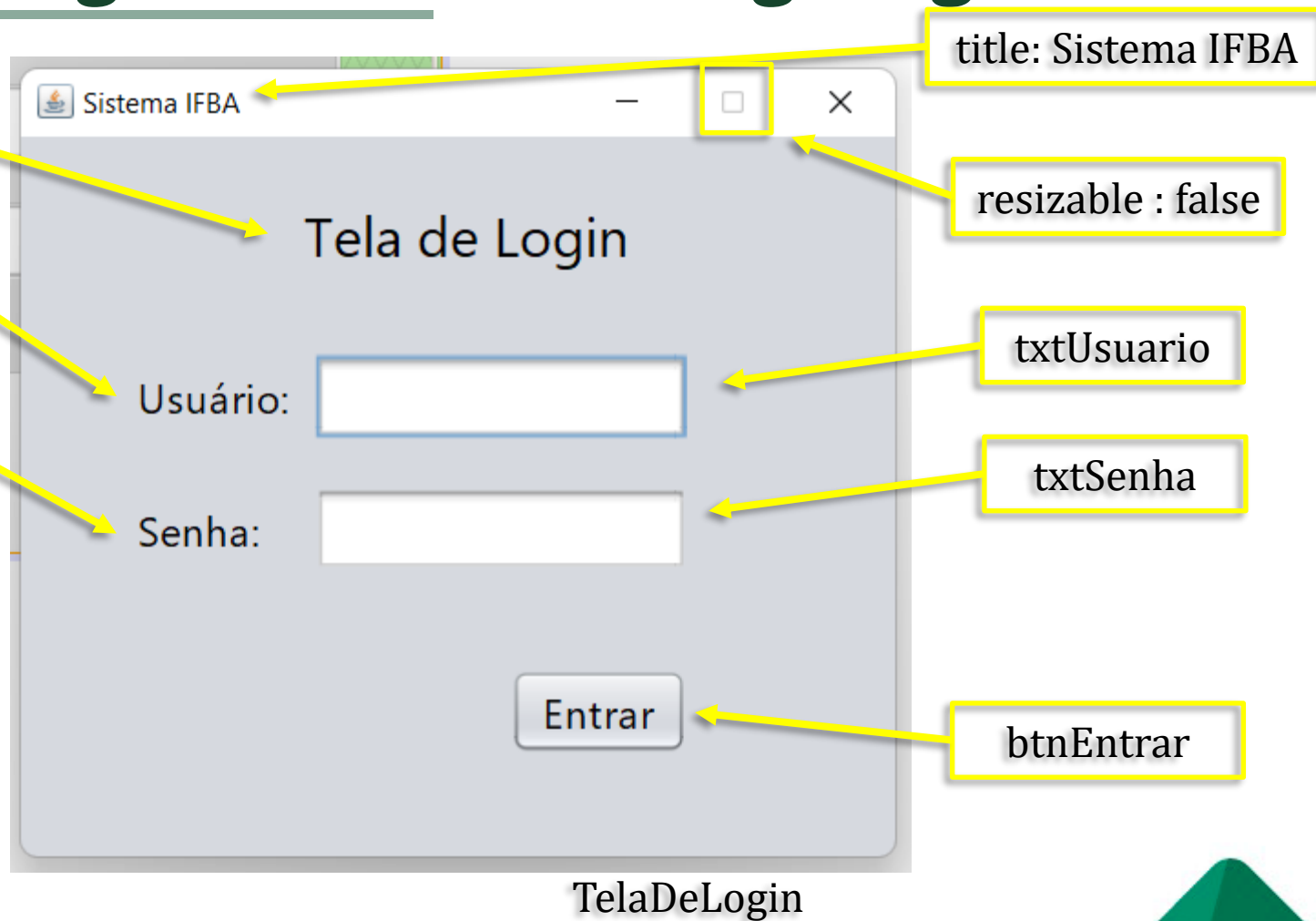


The screenshot shows a Java Swing window titled "Sistema IFBA - Média". The window has a light blue background and contains the following elements:

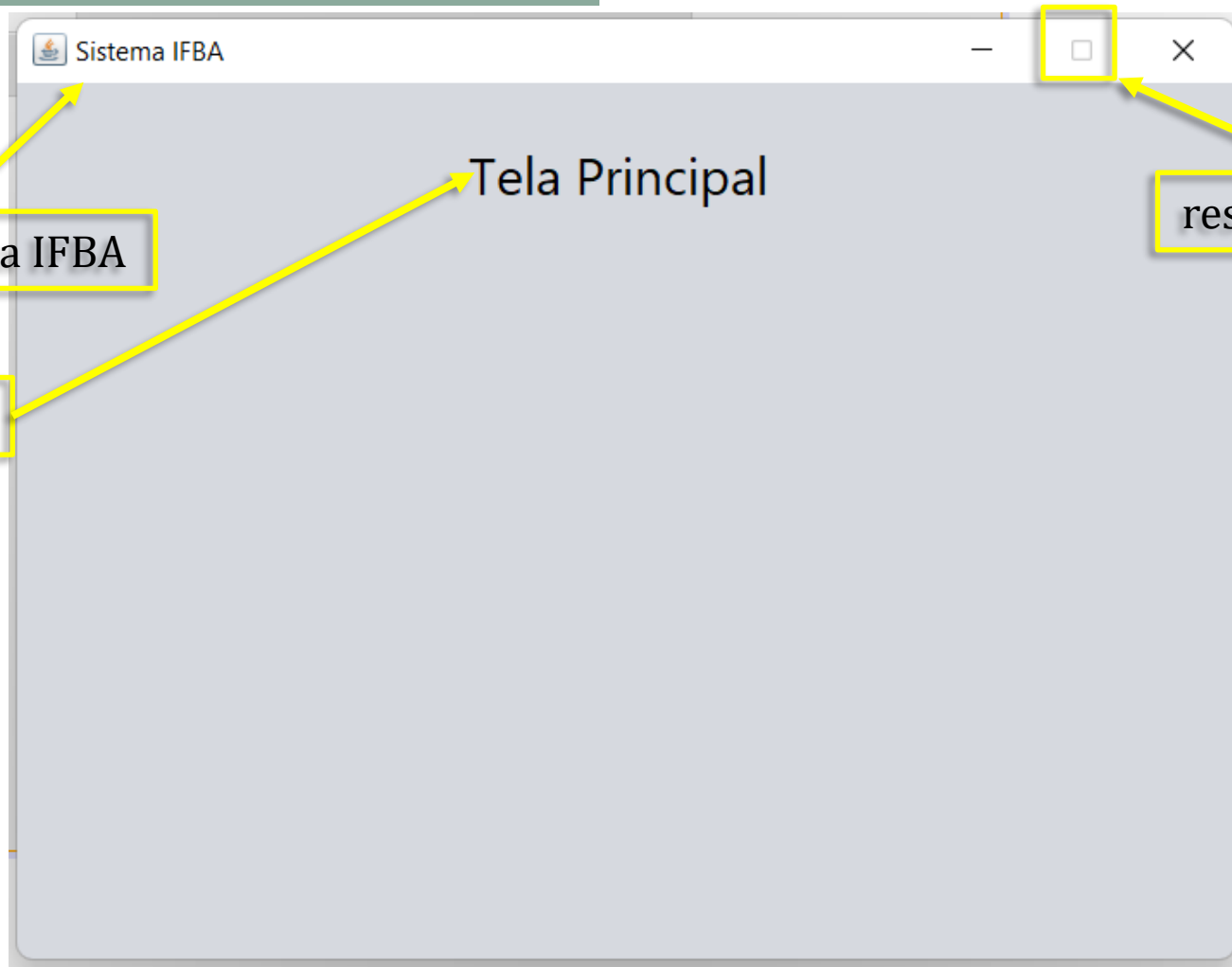
- Média Aritmética**: The title of the form.
- Nota 1:** A text label followed by a text field containing the value "7".
- Nota 2:** A text label followed by a text field containing the value "6".
- Nota 3:** A text label followed by a text field containing the value "4".
- Calcular**: A button with 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



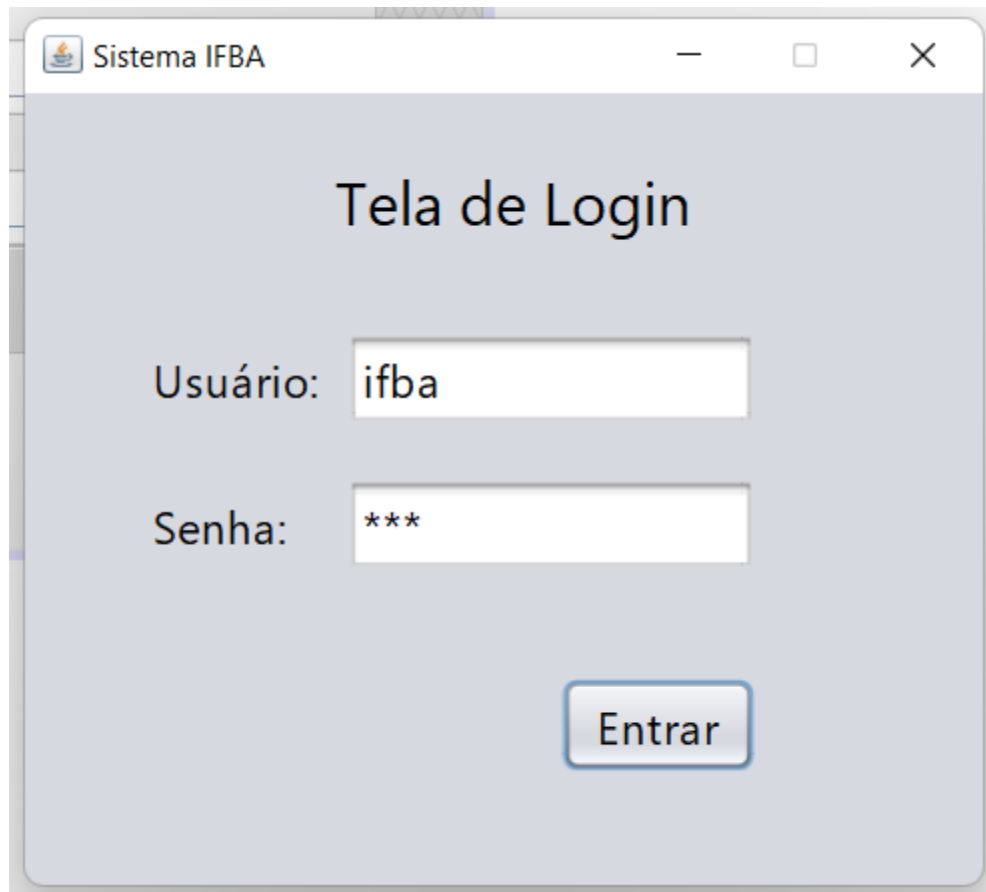
# Java Swing – Exercício – Prog. Login

```
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

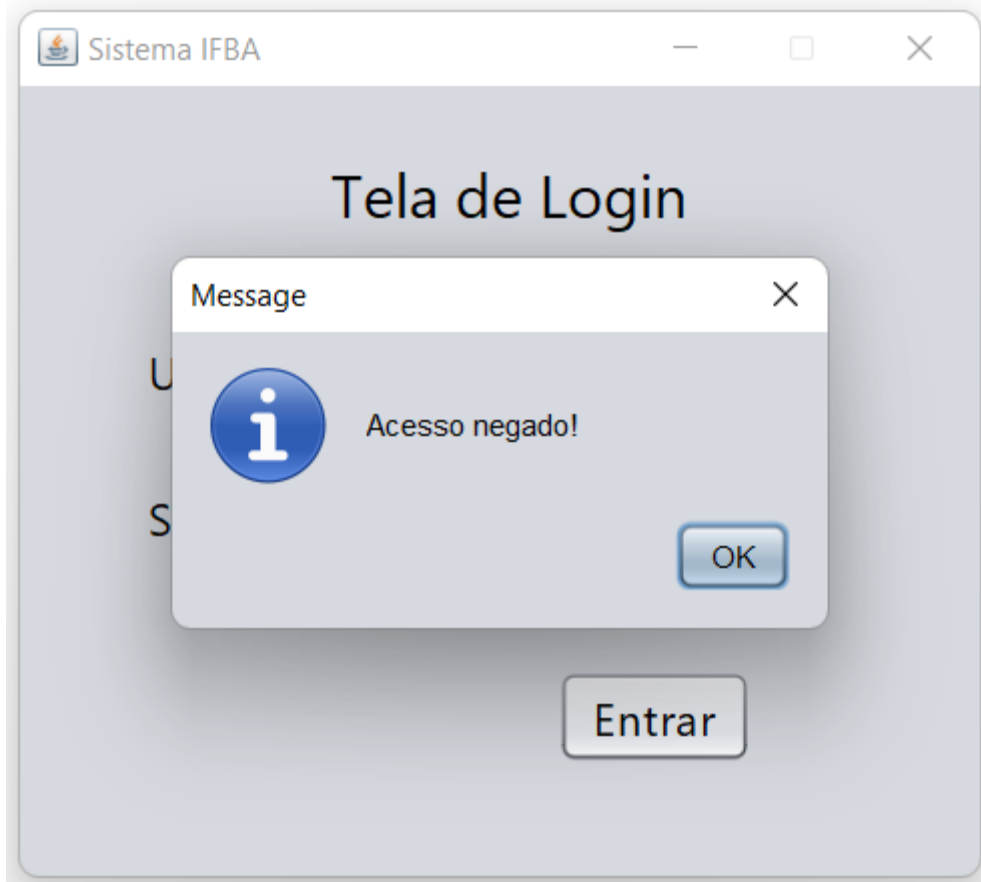


The screenshot shows a Java Swing window titled "Sistema IFBA". Inside the window, the text "Tela de Login" is centered. Below this, there are two input fields. The first is labeled "Usuário:" and contains the text "ifba". The second is labeled "Senha:" and contains three asterisks "\*\*\*". At the bottom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



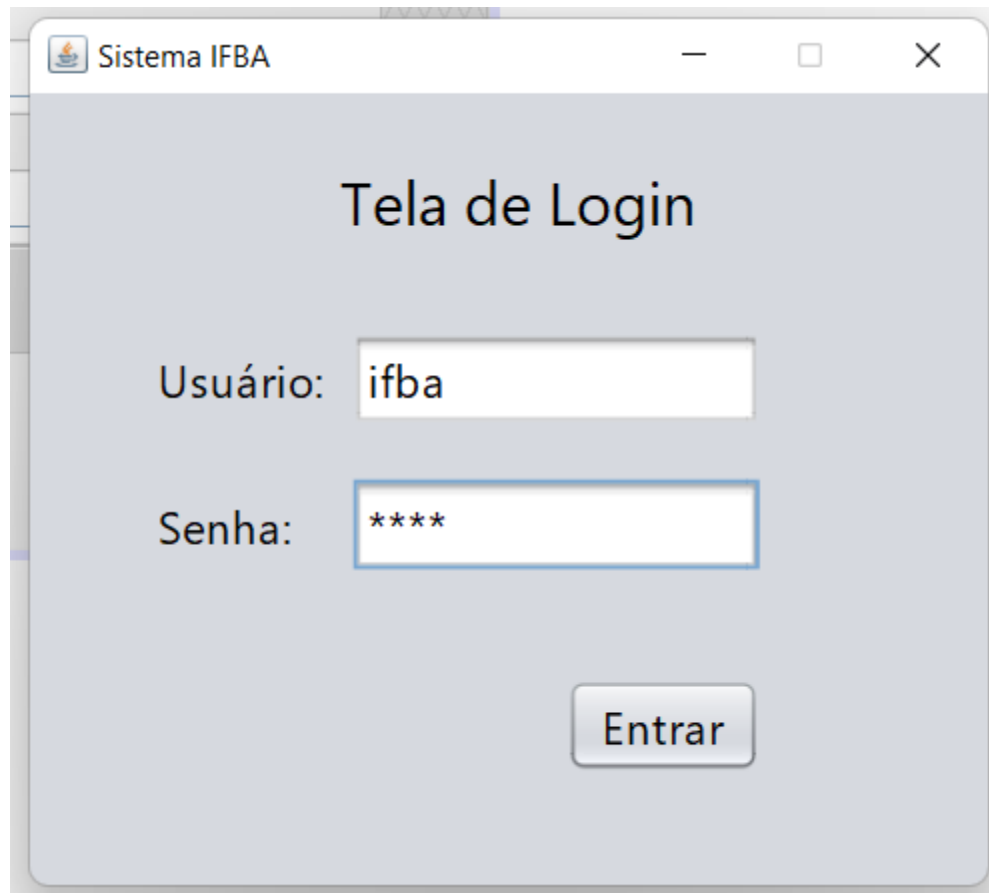
# Java Swing – Exercício – Prog. Login

Execução



# Java Swing – Exercício – Prog. Login

Execução

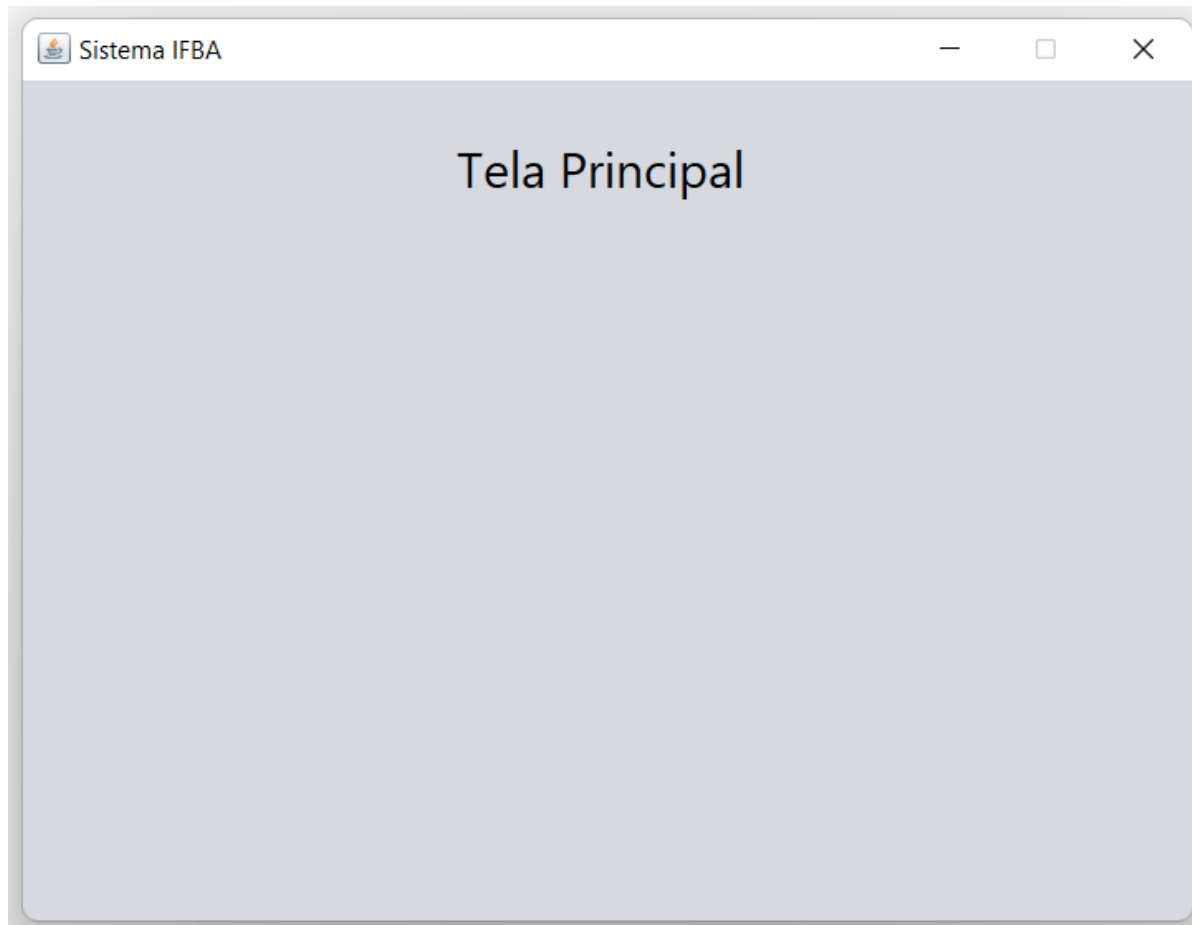


The screenshot shows a Java Swing window titled "Sistema IFBA". Inside the window, the text "Tela de Login" is centered. Below this, there are two input fields. The first is labeled "Usuário:" and contains the text "ifba". The second is labeled "Senha:" and contains four asterisks "\*\*\*\*". Below these 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](https://www.youtube.com/playlist?list=PLWd_VnthxxLfeEUK08gB7G3316OS5xIT3)



# Obrigado!

- Canais de Comunicação;
- Horário de Atendimento.

