



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)



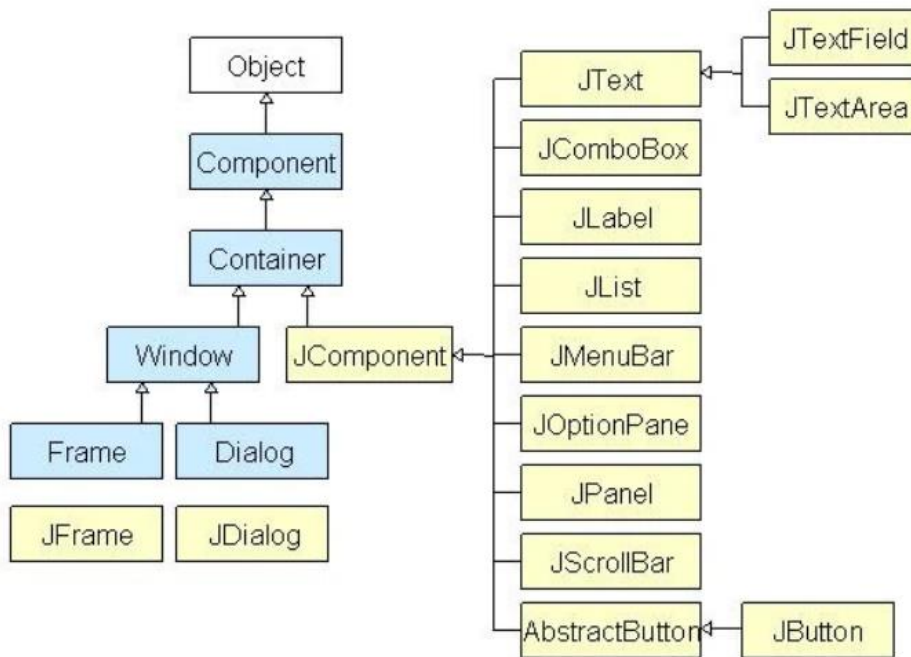
Roteiro

- 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)**.

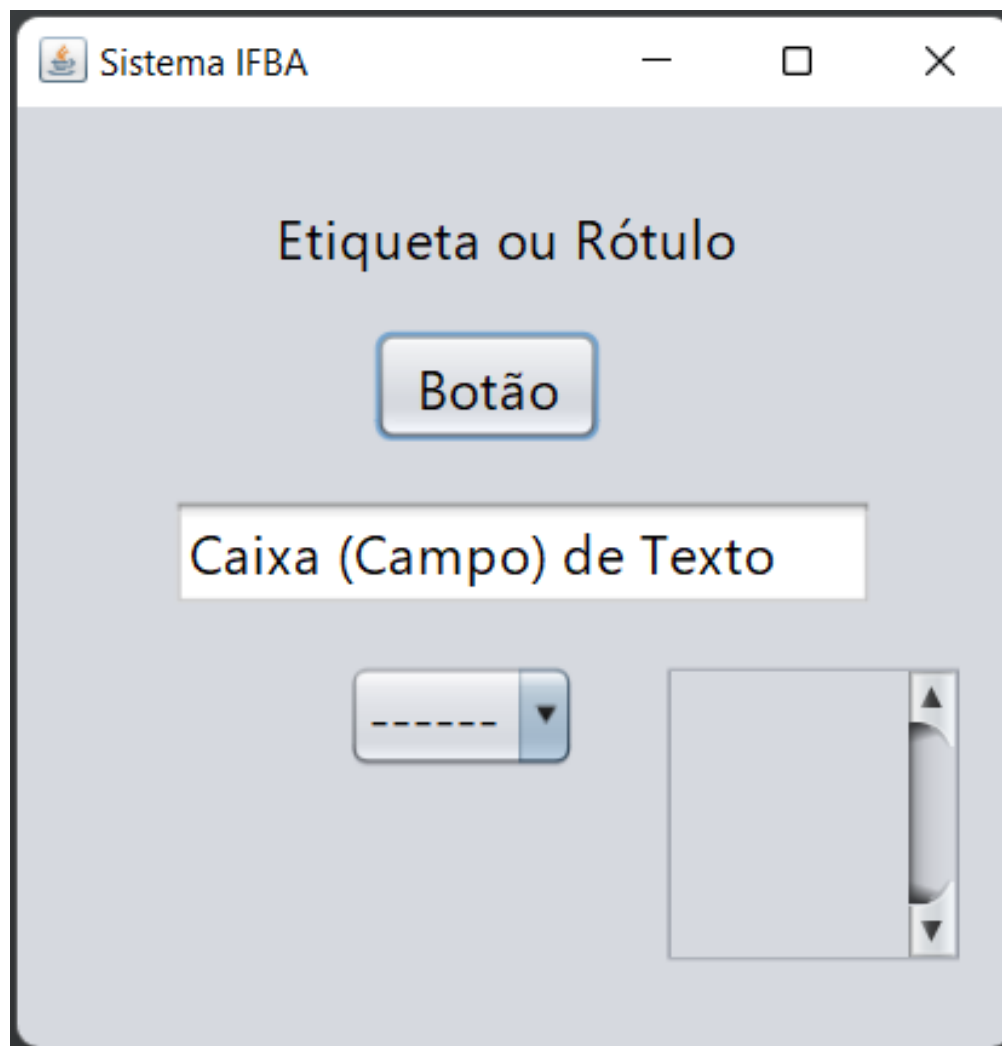


Java Swing – Componentes

- 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

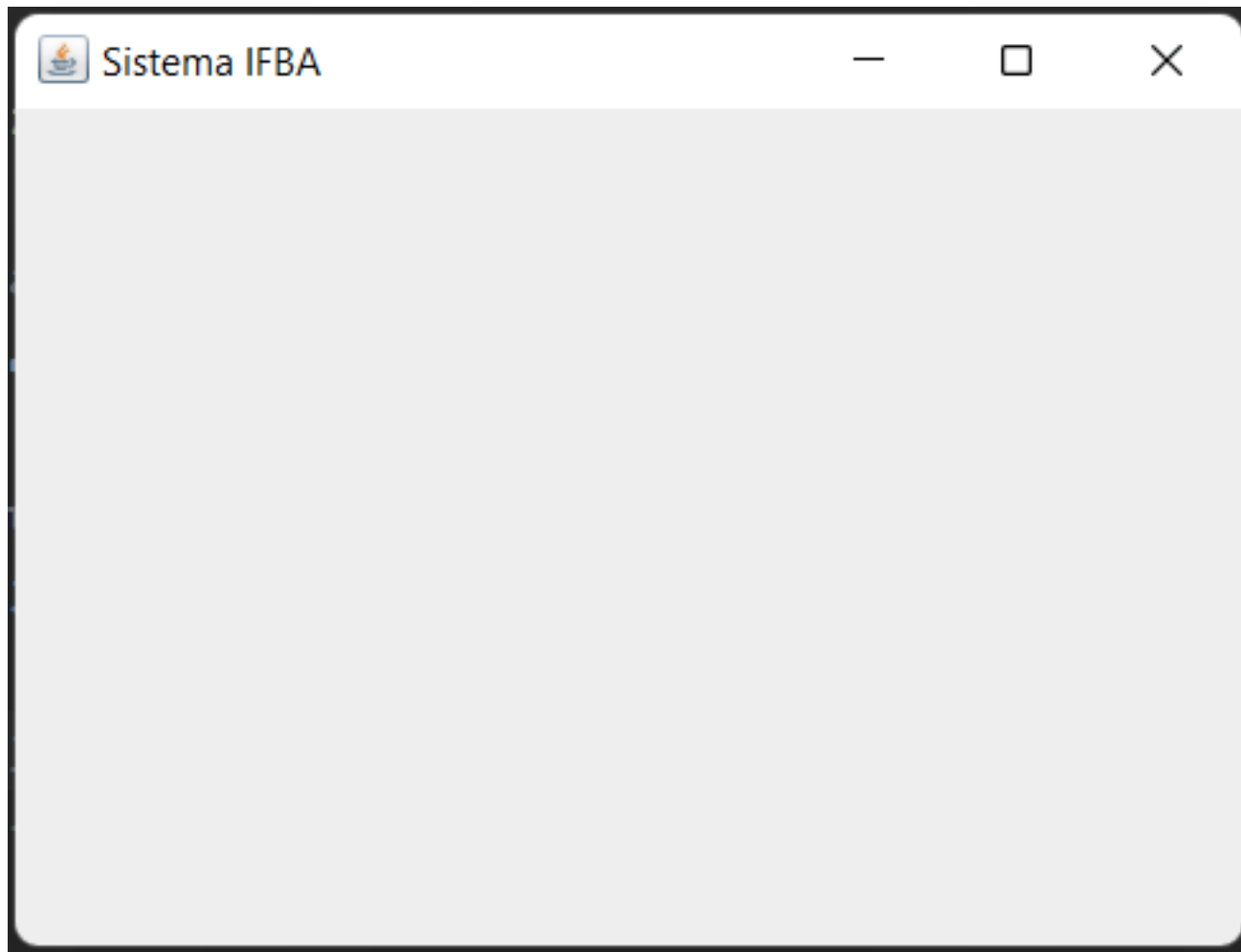


Java Swing – JFrame

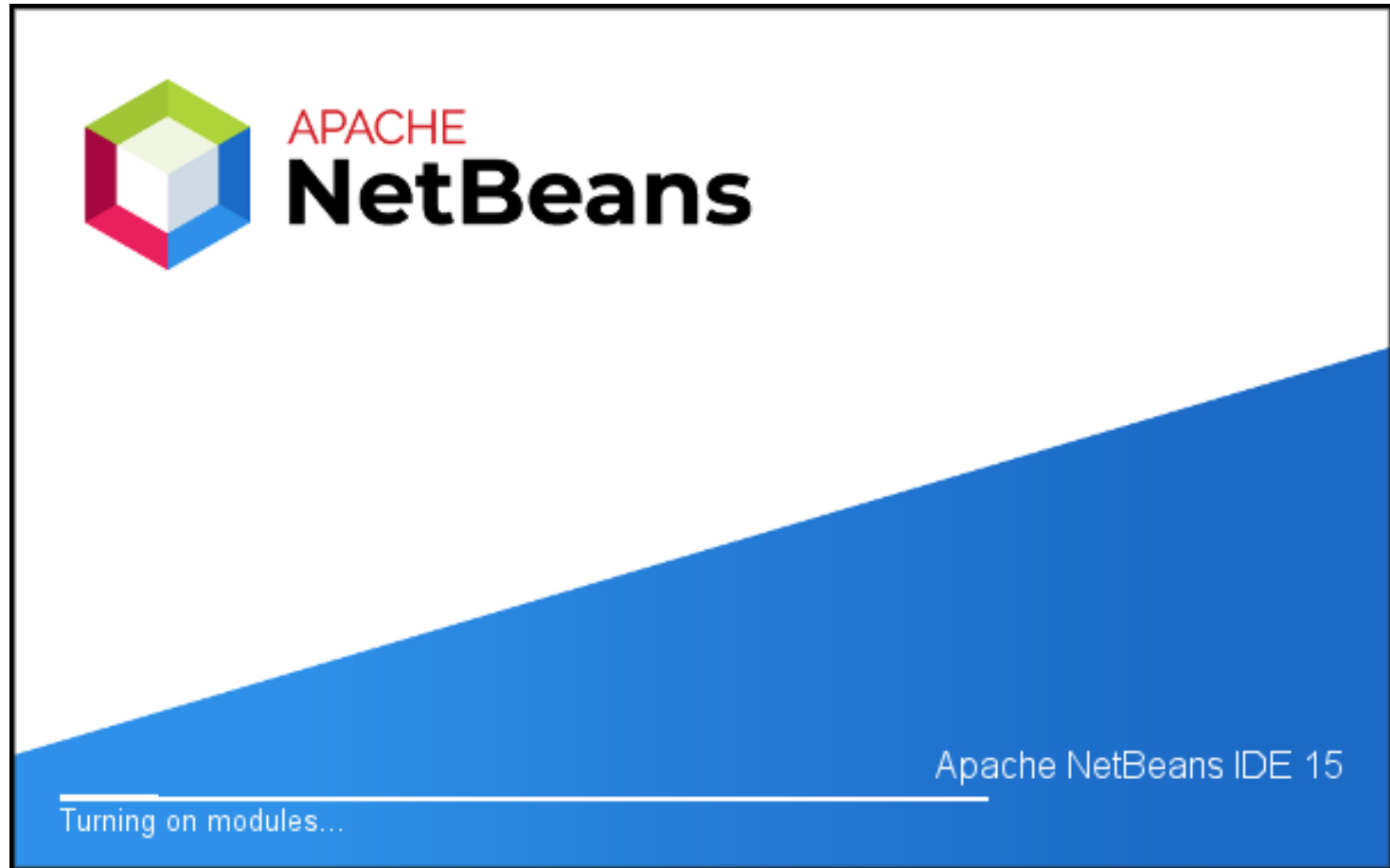
- Para criar **Janelas** utilizaremos a classe **JFrame** disponível no pacote **swing**, a qual gera uma **janela** com **barra de título**, **bordas** e eventualmente outros **componentes visuais**.

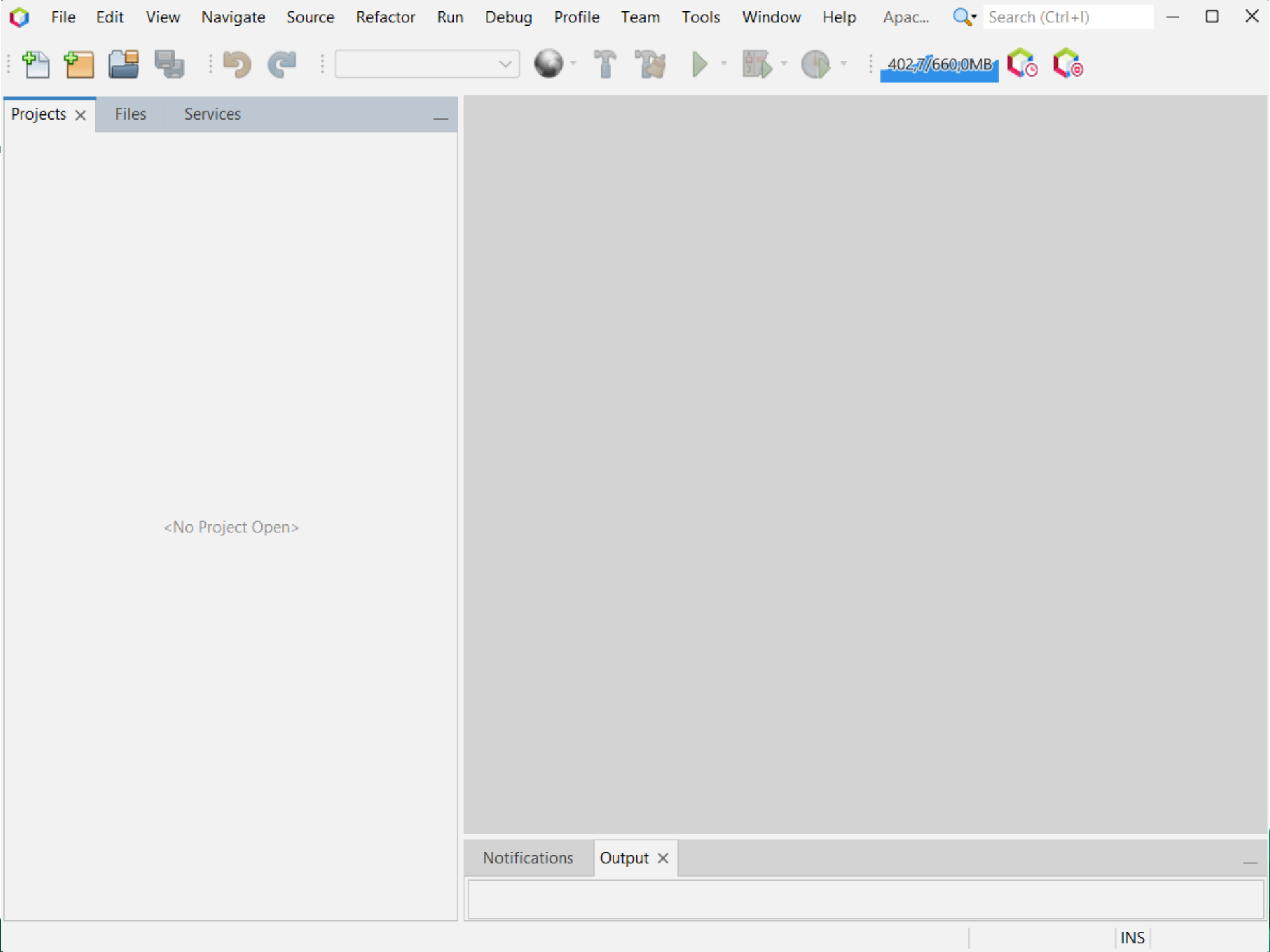


Java Swing – JFrame

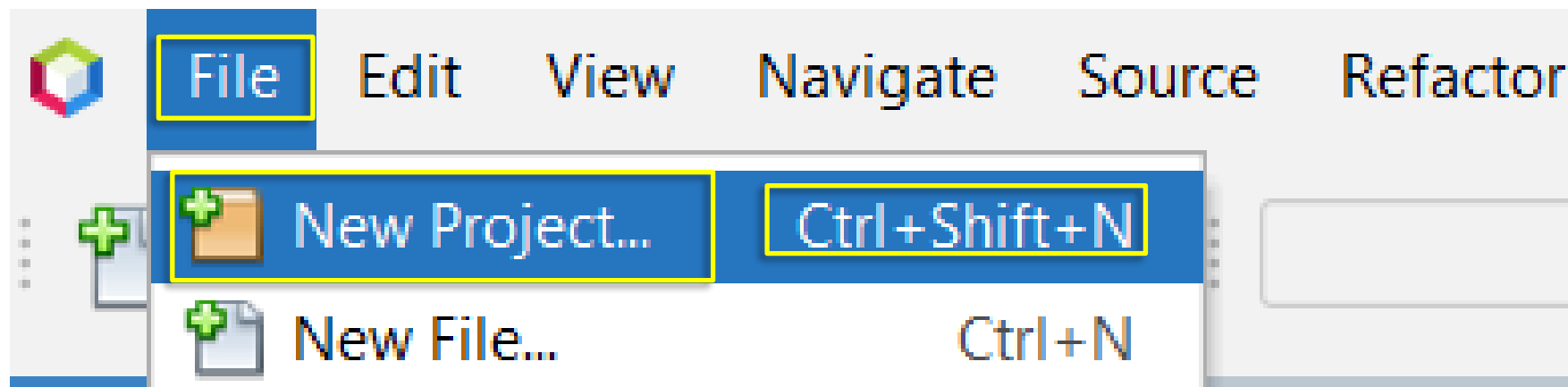


Netbeans: IDE





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

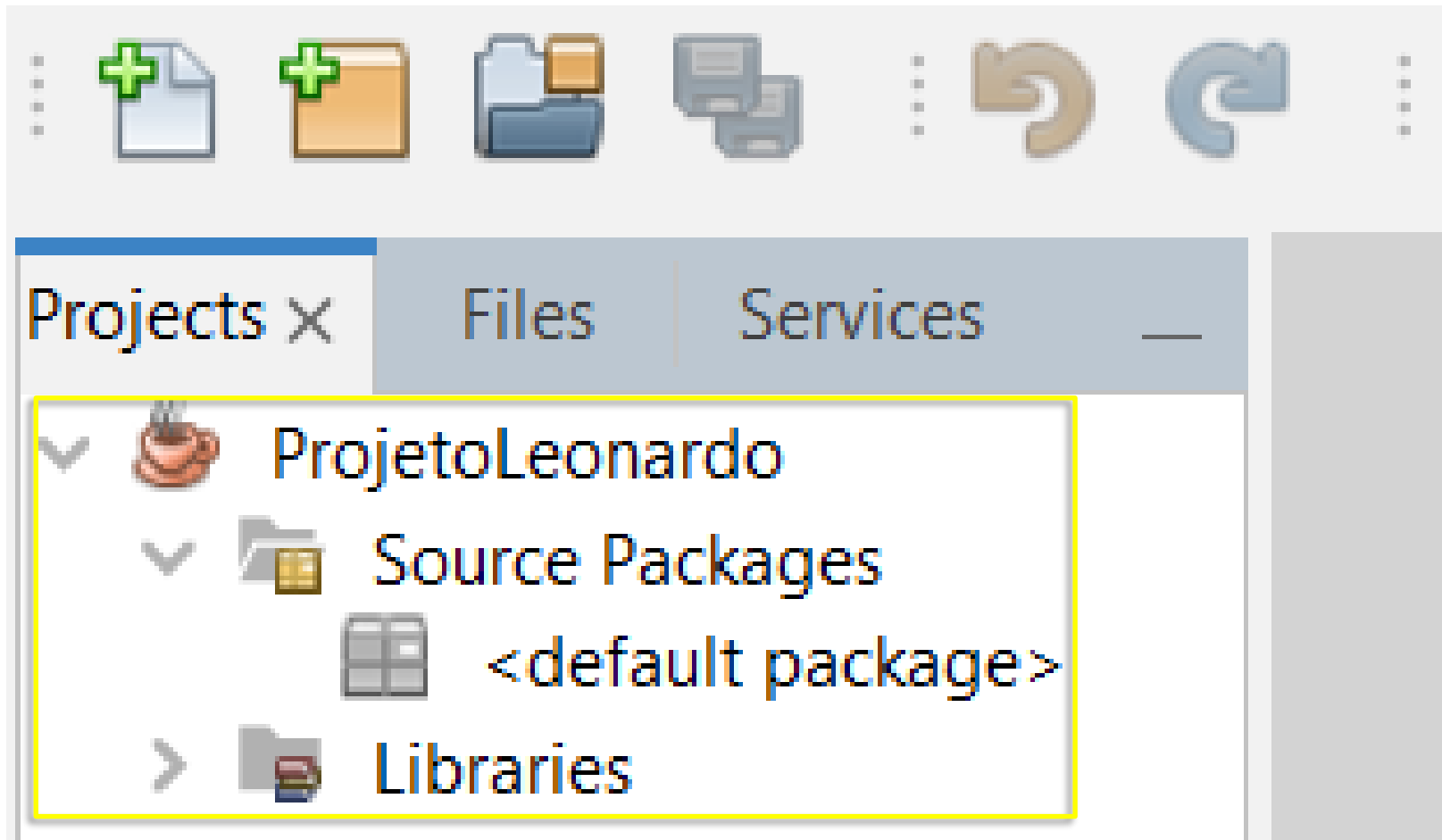
Libraries Folder:

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

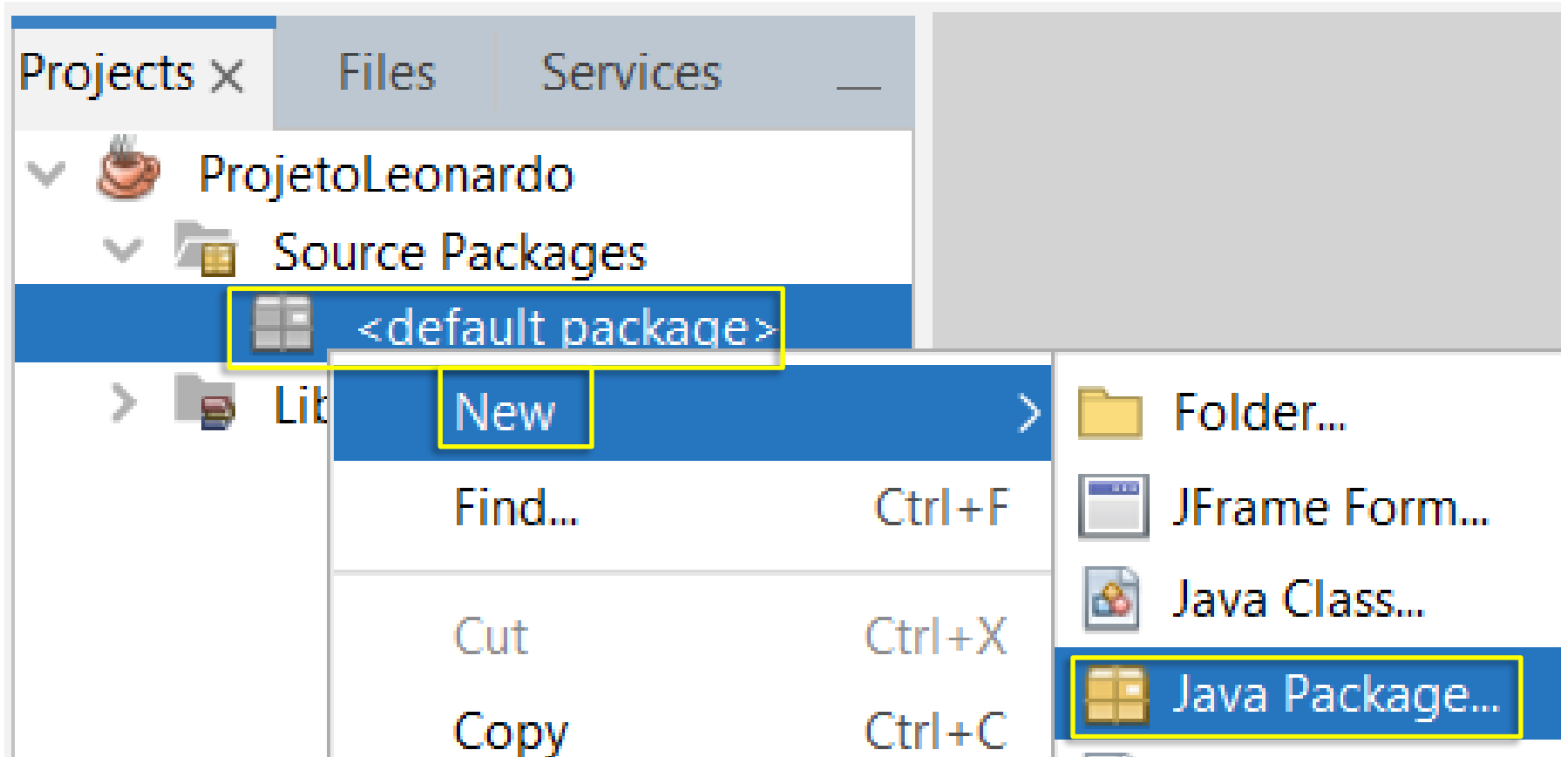
☐ Create Main Class



Netbeans: Novo Projeto



Netbeans: Novo Pacote



Netbeans: Novo Pacote

Name and Location

Package Name:

Principal

Project:

ProjetoLeonardo

Location:

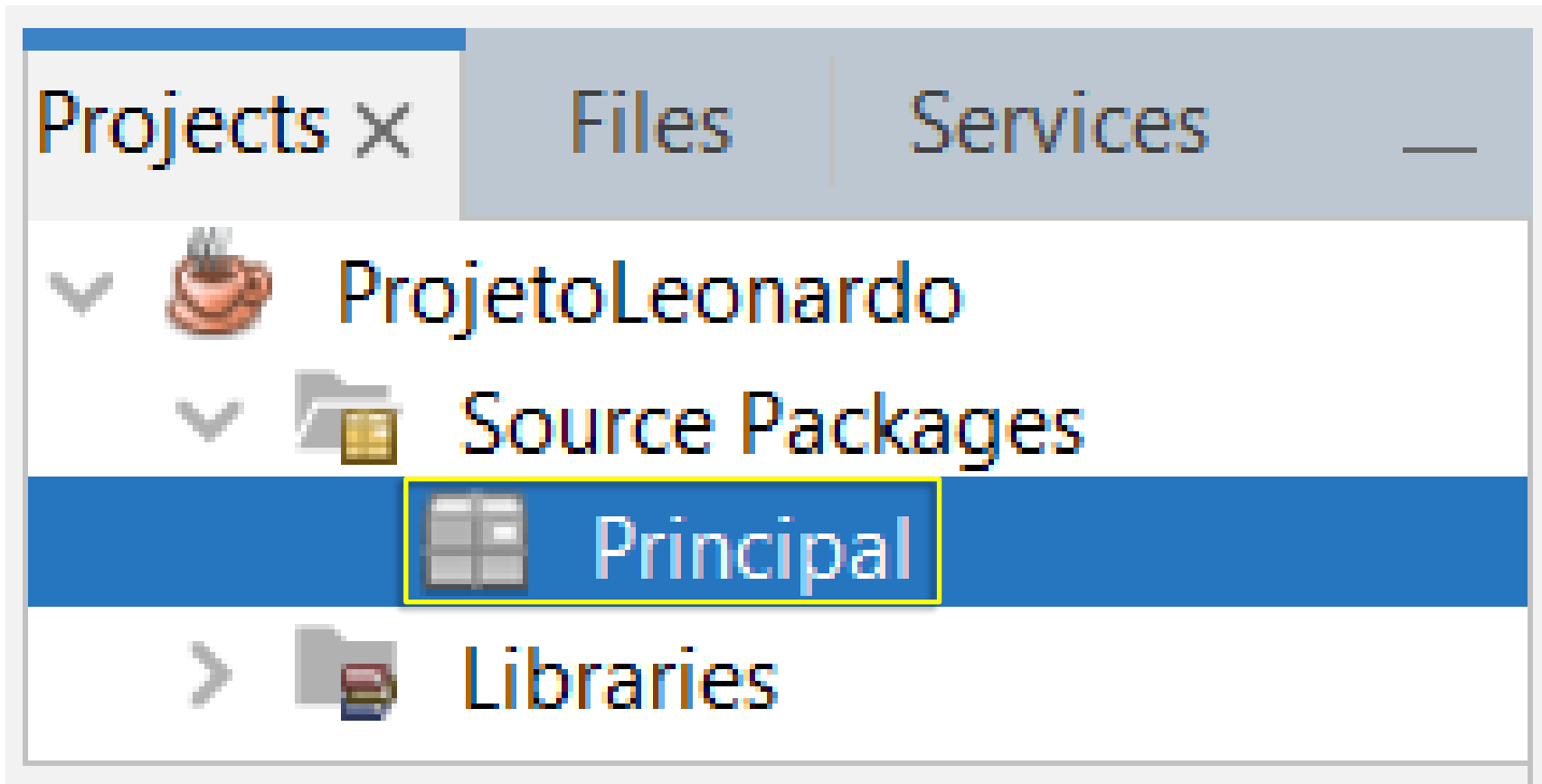
Source Packages

Created Folder:

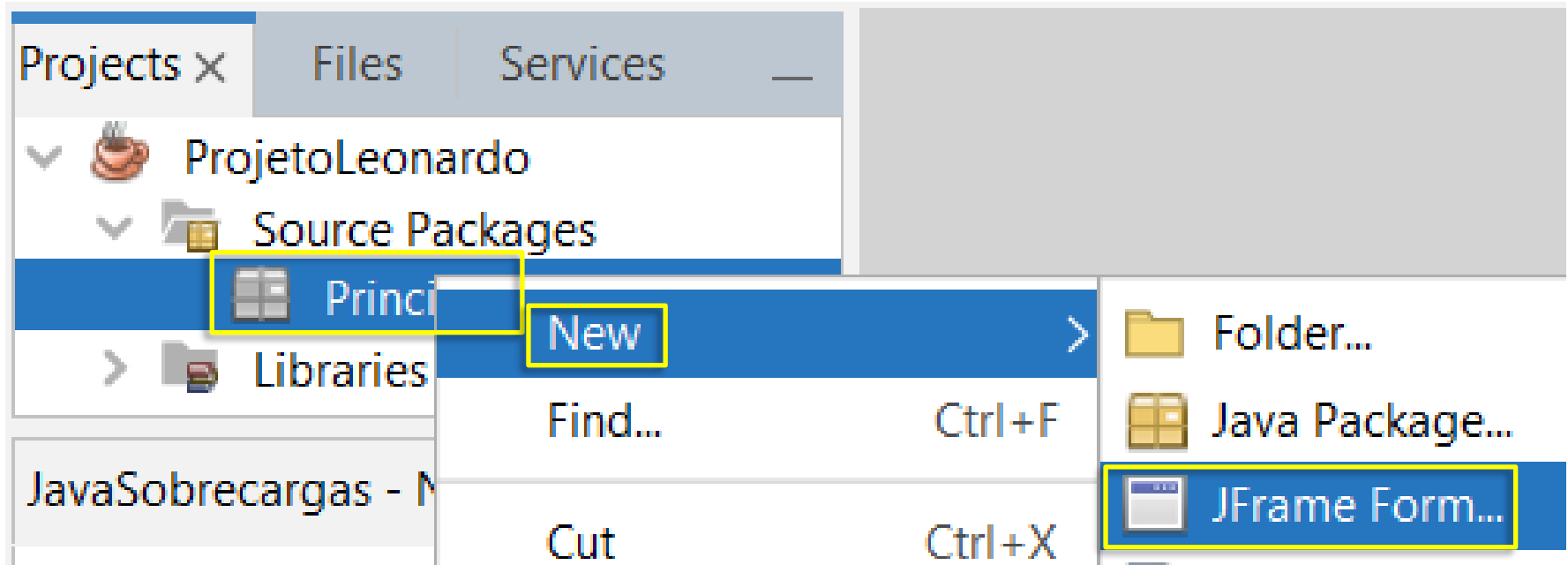
.:\\Users\\leomi\\Documents\\NetBeansProjects\\ProjetoLeonardo\\src\\Principal



Netbeans: Novo Pacote



Netbeans: Nova Janela (JFrame)



Netbeans: Nova Janela (Jframe)

Name and Location

Class Name:

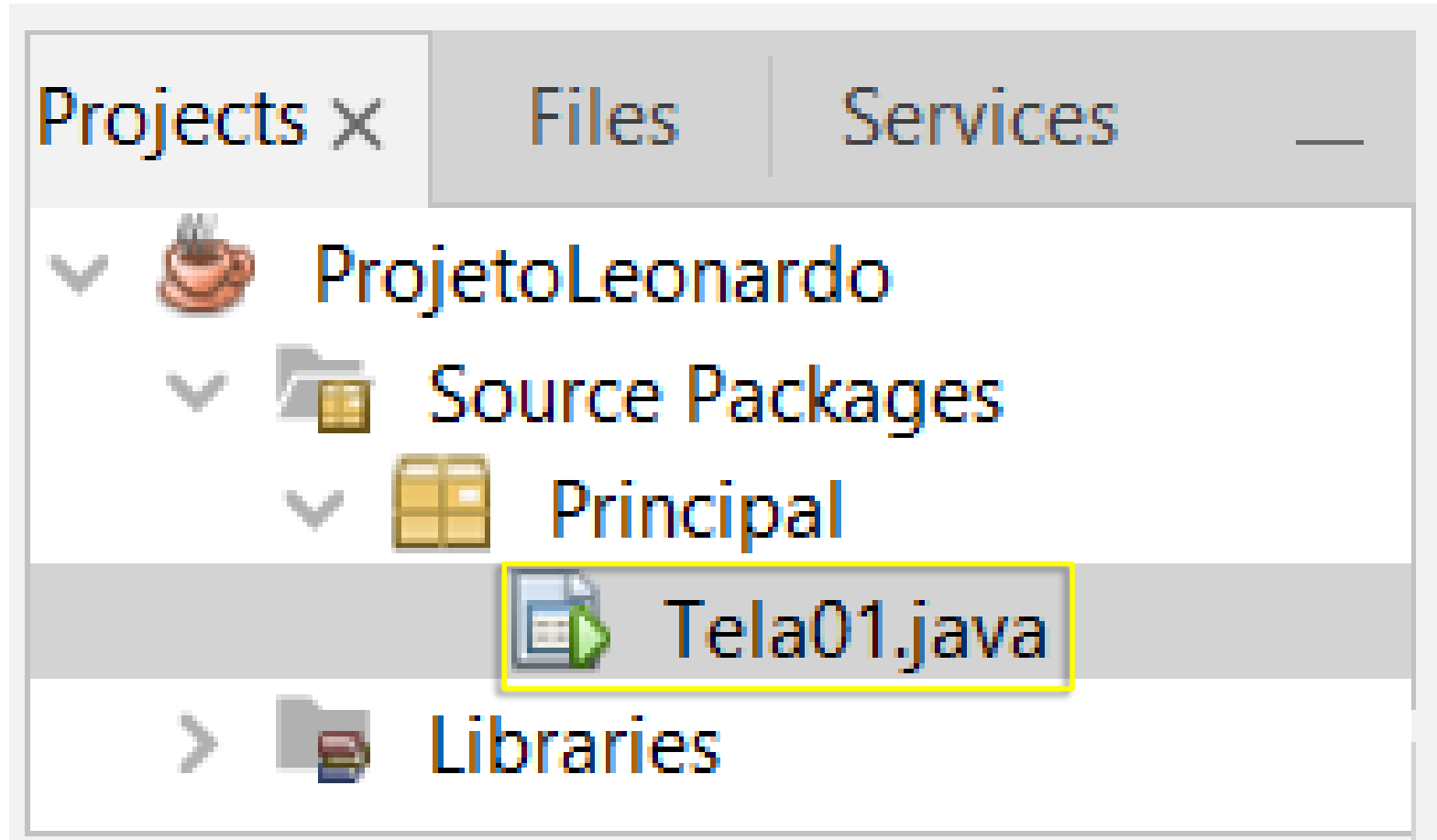
Project:

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		...

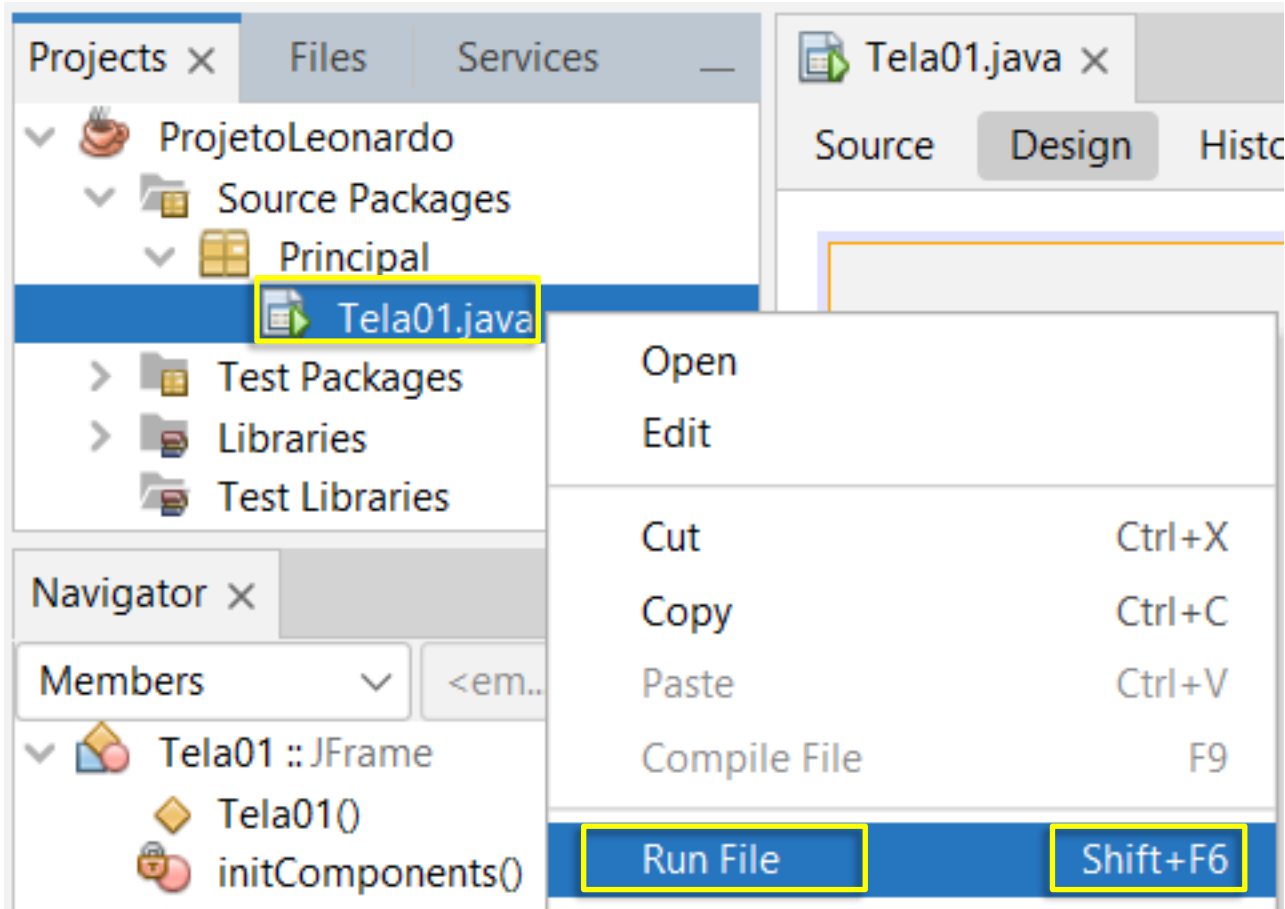
Other Properties

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

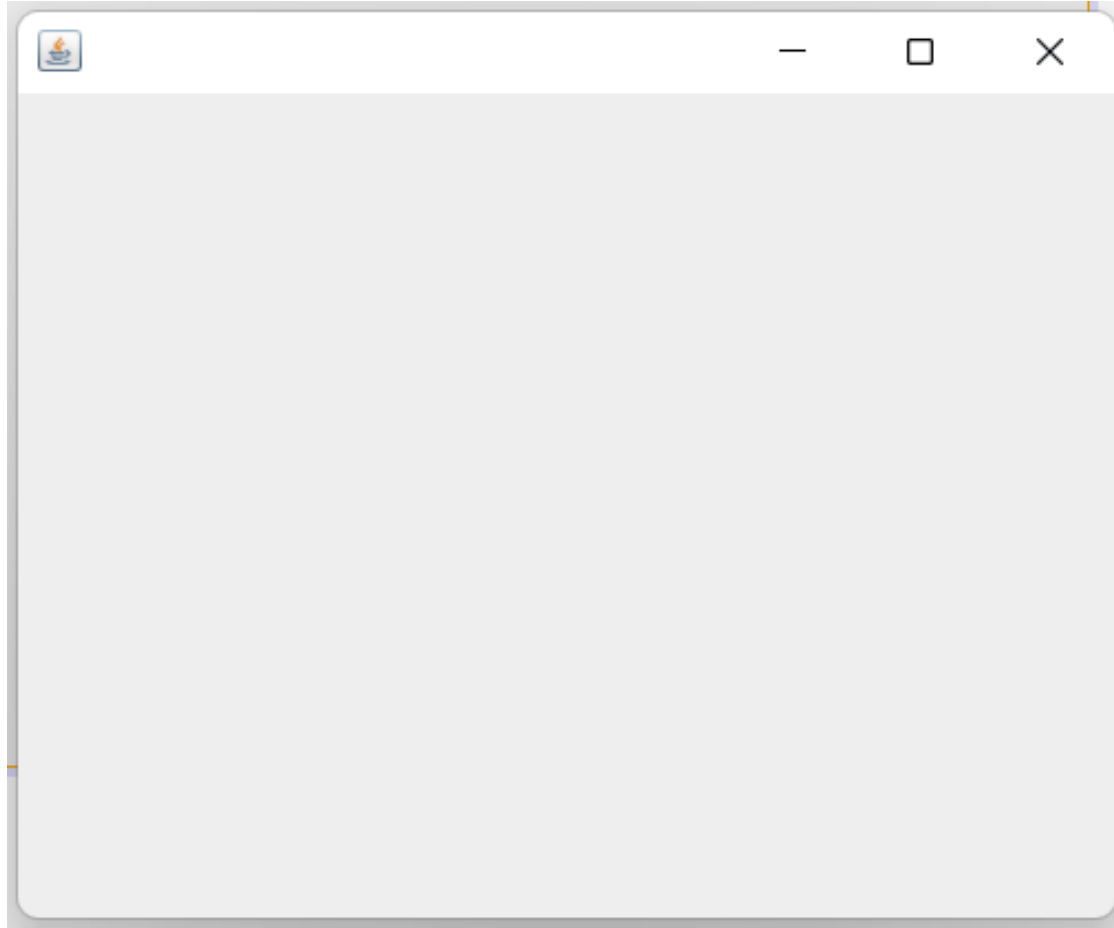
[JFrame]

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

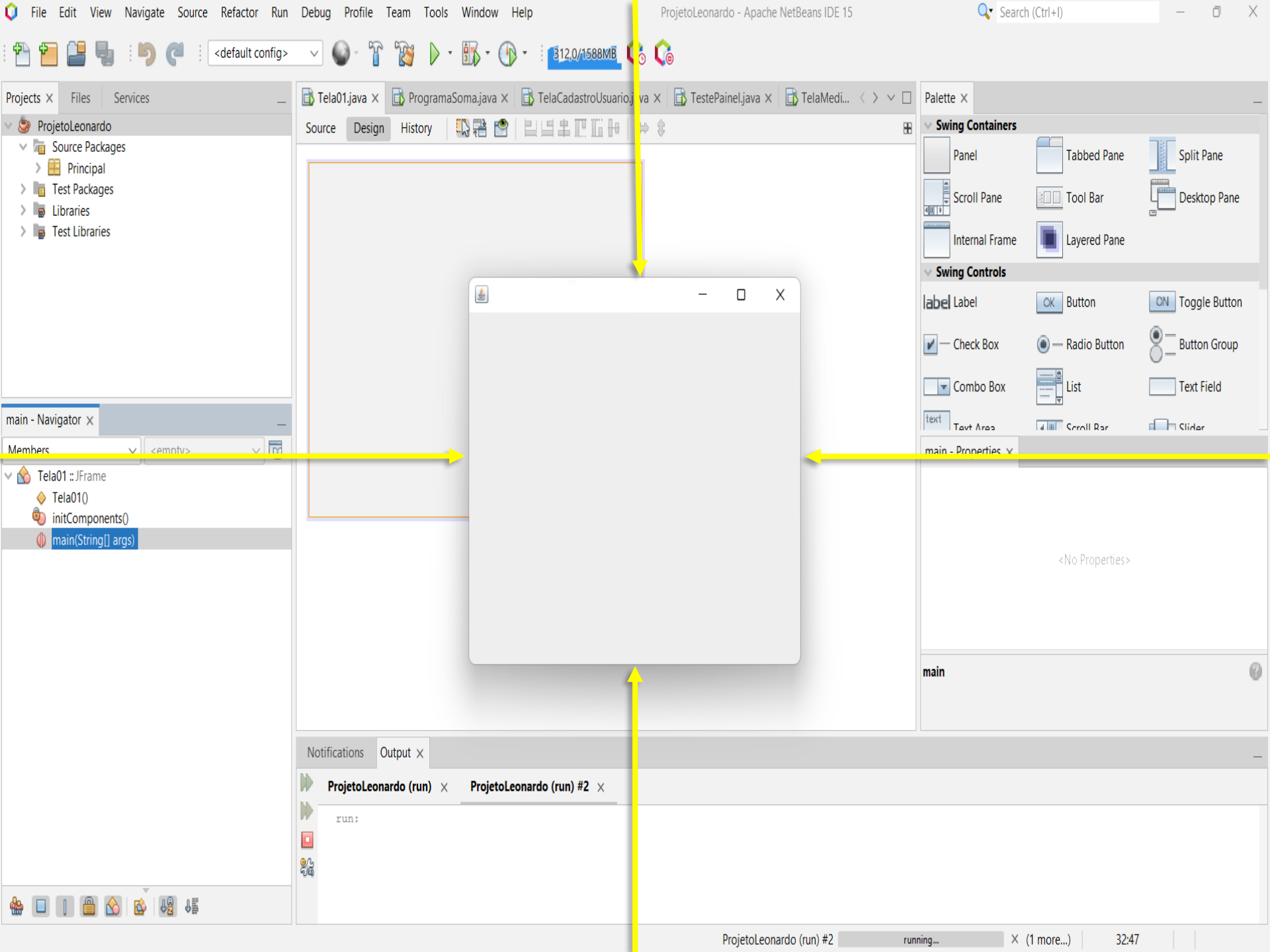
Configurações de Layout do JFrame

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

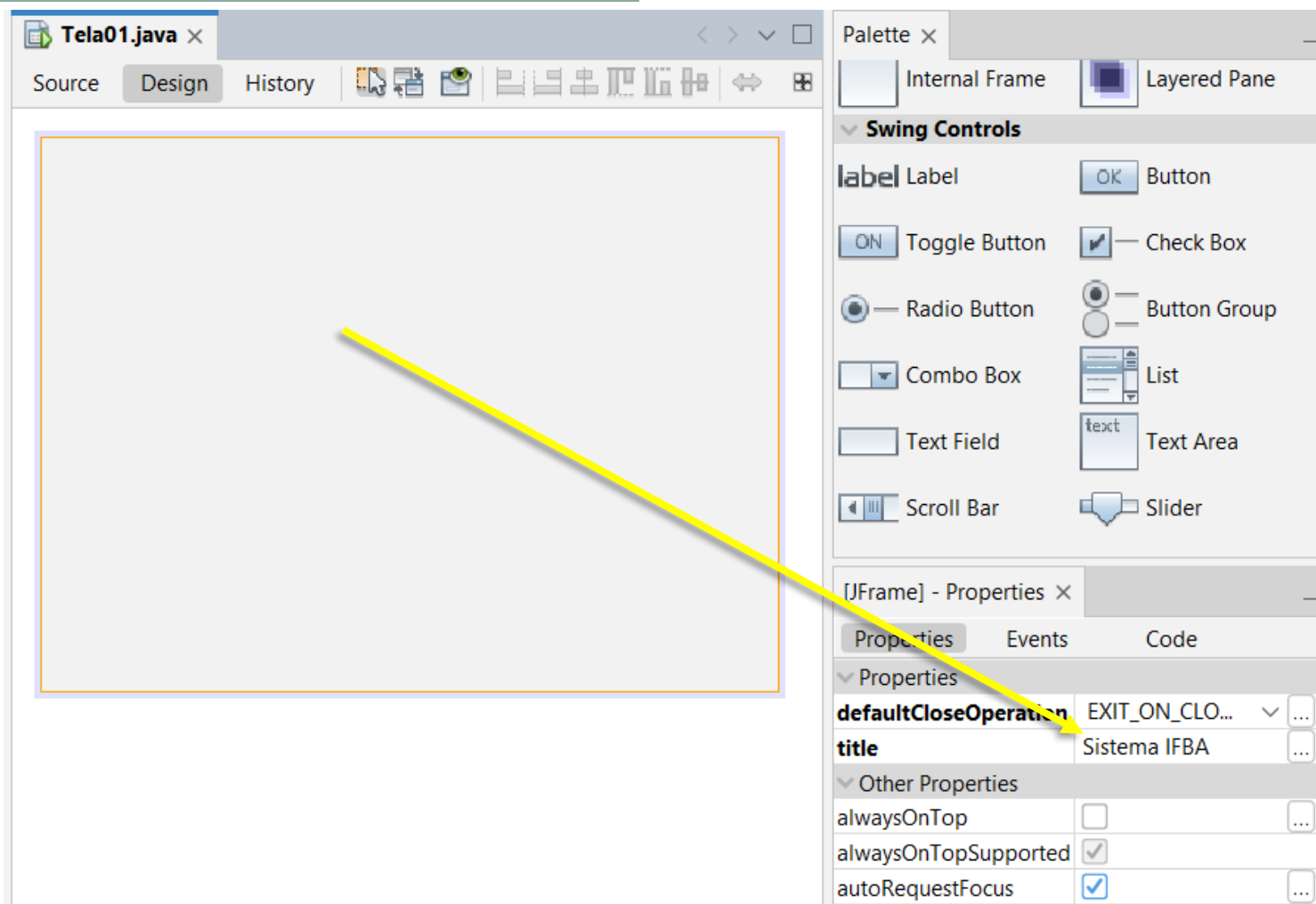
Netbeans: Centralizar JFrame

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

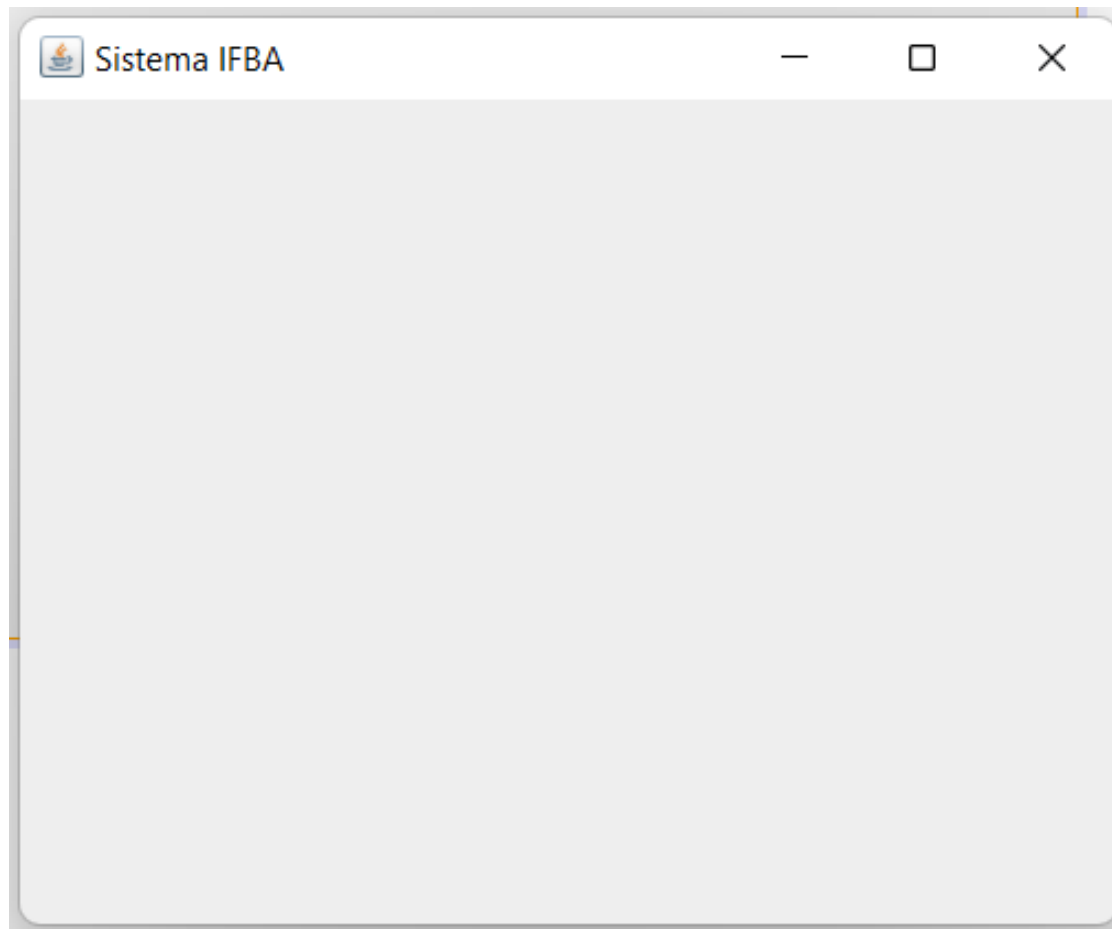




Netbeans: Título da Janela (JFrame)



Netbeans: Título da Janela (JFrame)



Netbeans: Desabilitar Redimensionar

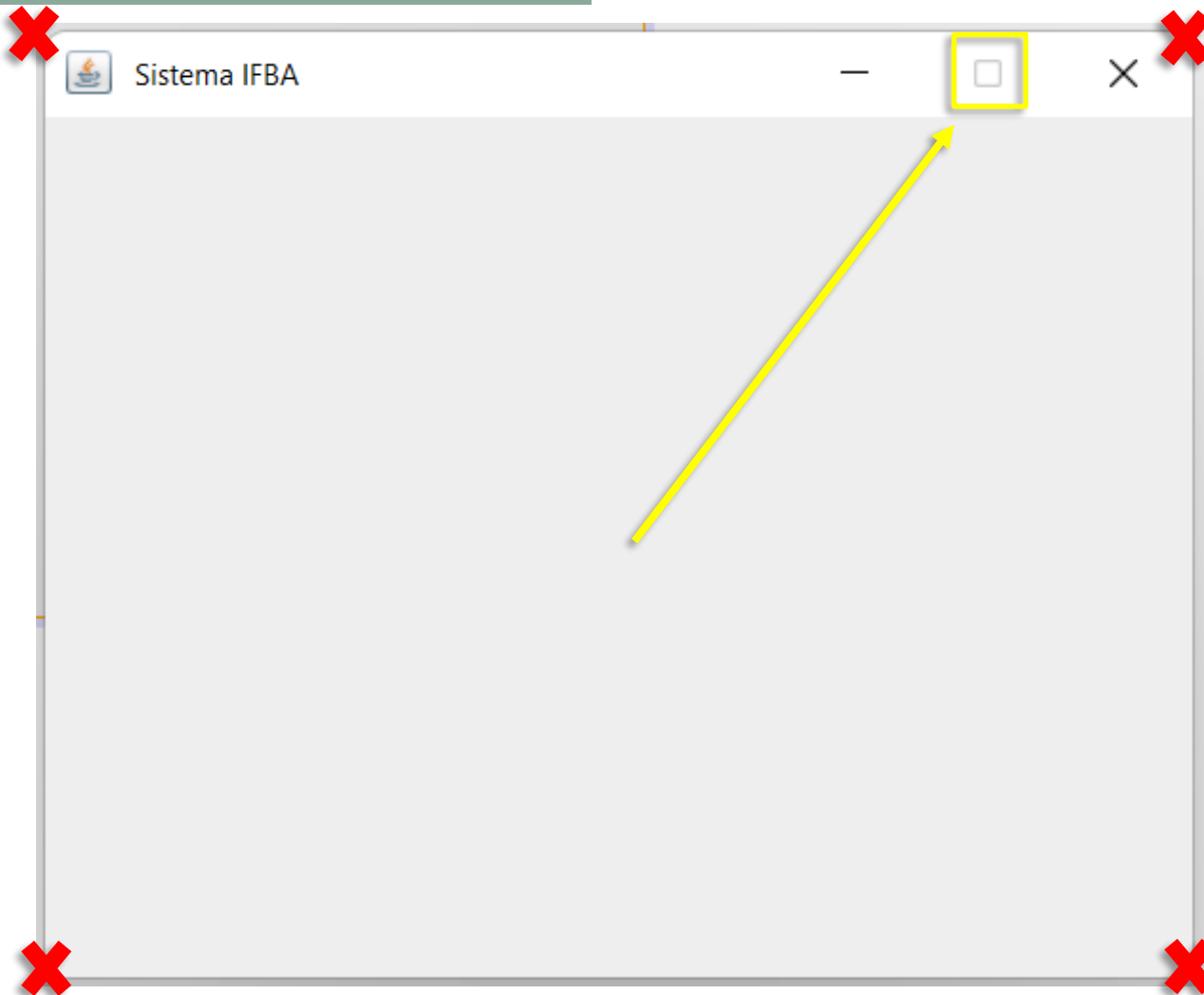
[JFrame] - Properties X

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

False



Netbeans: Desabilitar Redimensionar

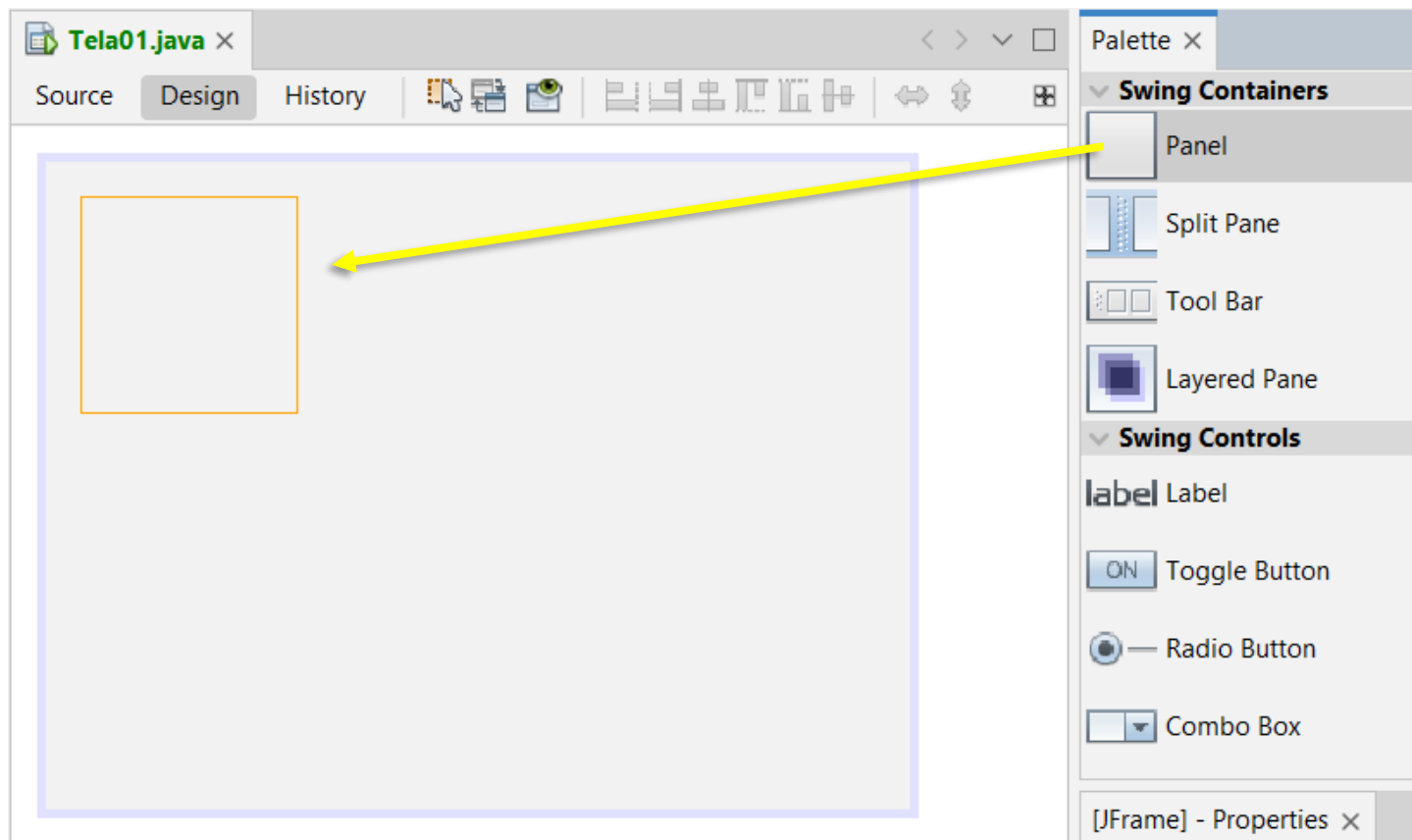


Java Swing – JPanel

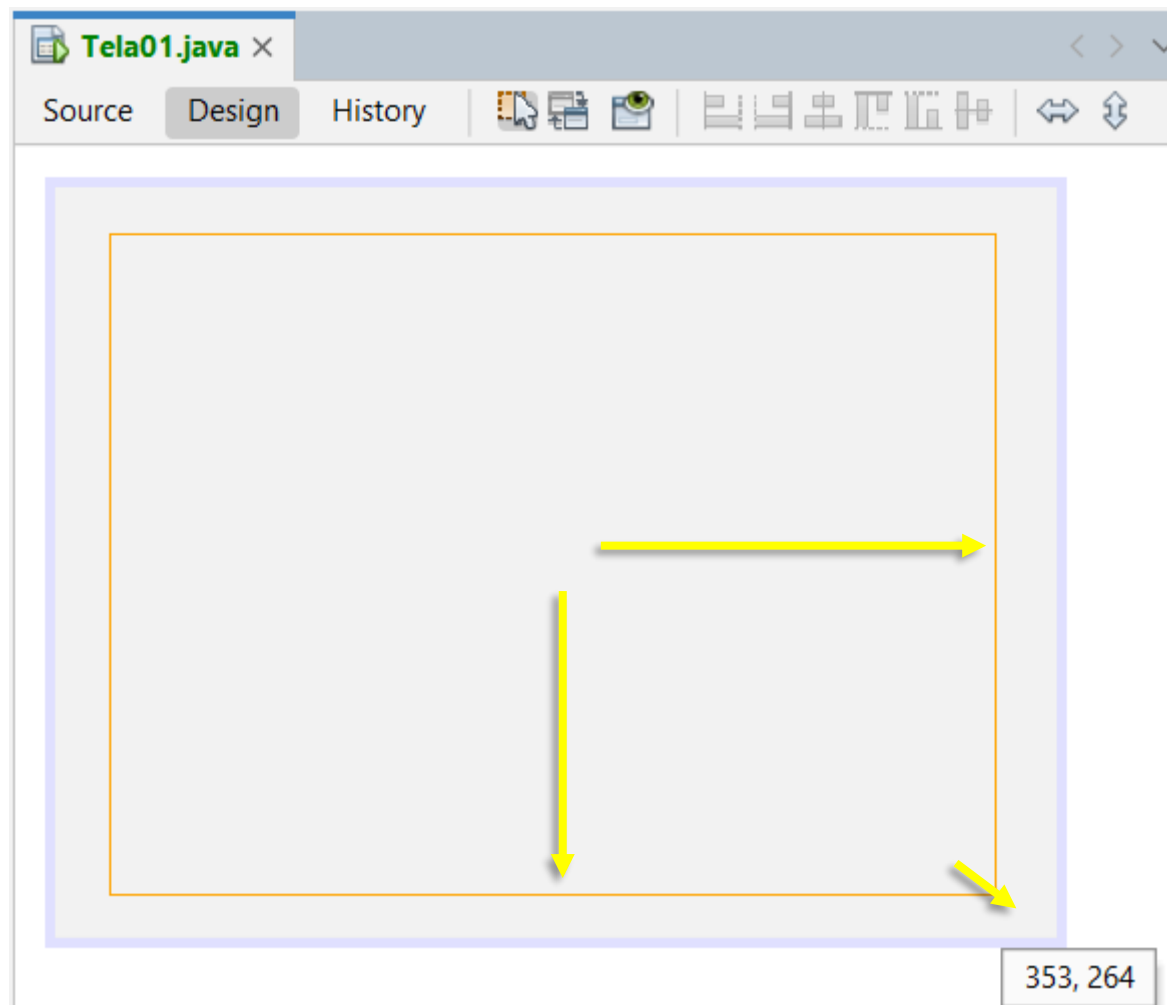
- O **JPanel** é um **contêiner flexível** no Java Swing, utilizado para **agrupar e organizar componentes** em interfaces gráficas.
- Ele **oferece opções de personalização**, como **layout e cor de fundo**, permitindo criar **interfaces atraentes e adaptáveis**.



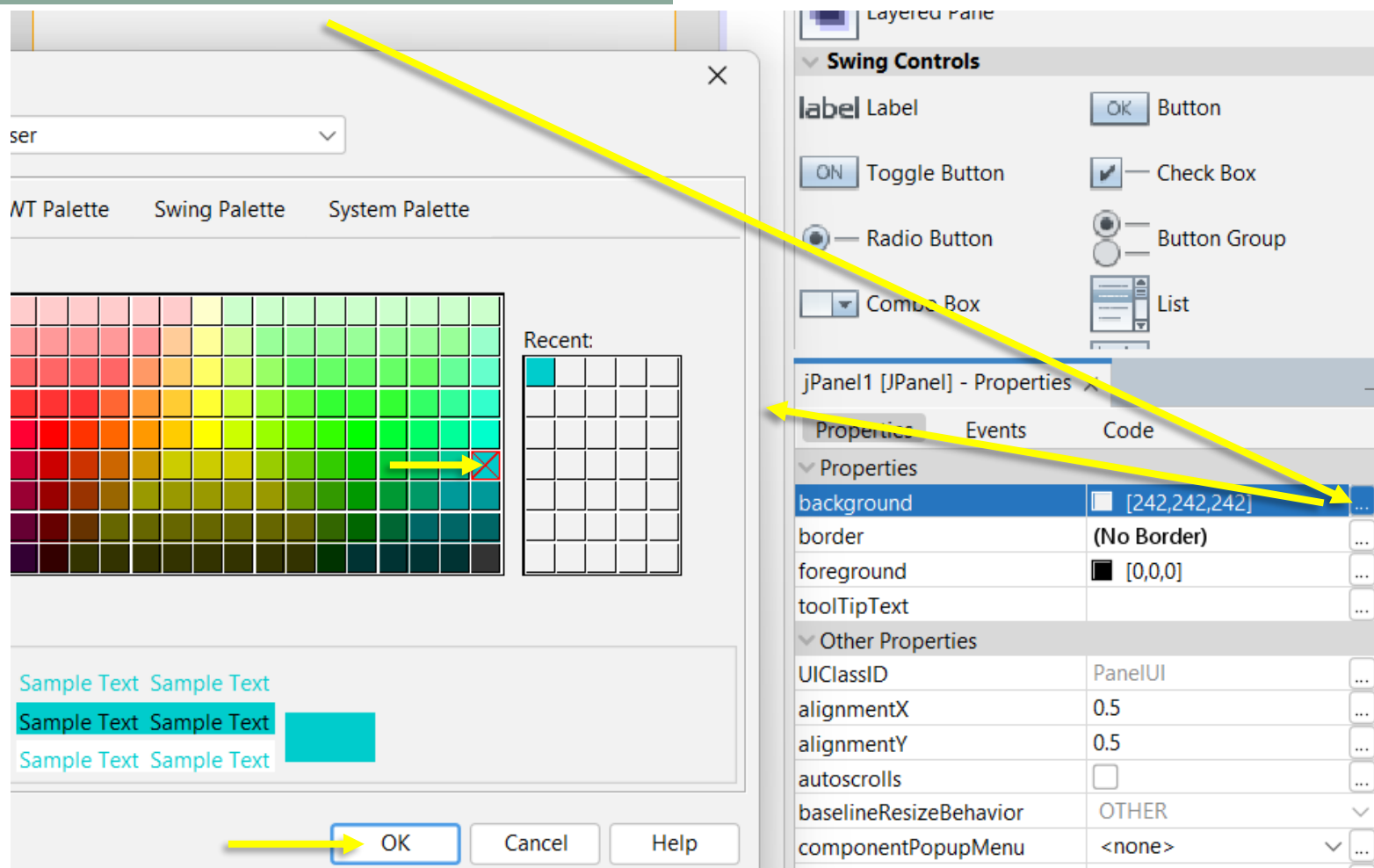
Java Swing – JPanel - Adicionar



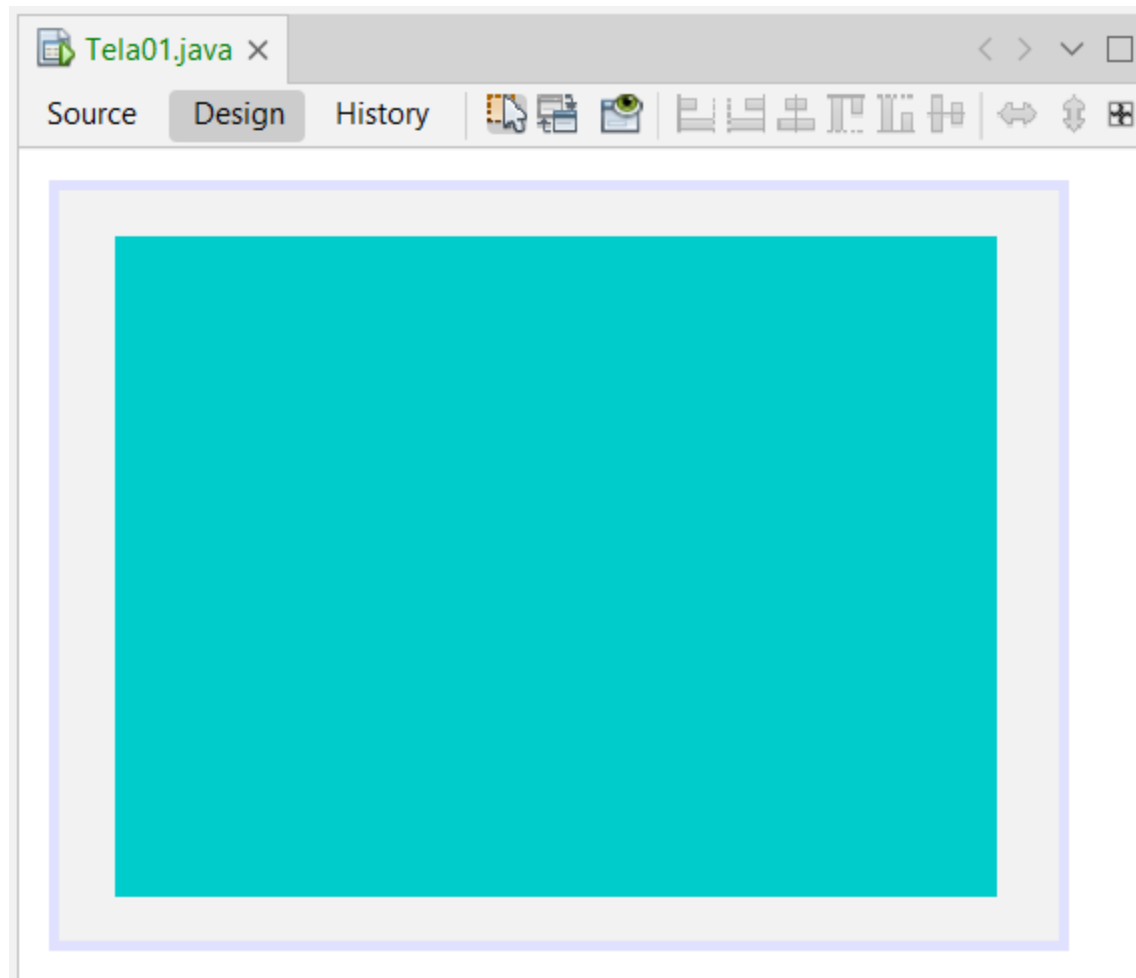
Java Swing – JPanel - Redimensionar



Java Swing – JPanel – Mudar Cor Fundo



Java Swing – JPanel – Mudar Cor Fundo

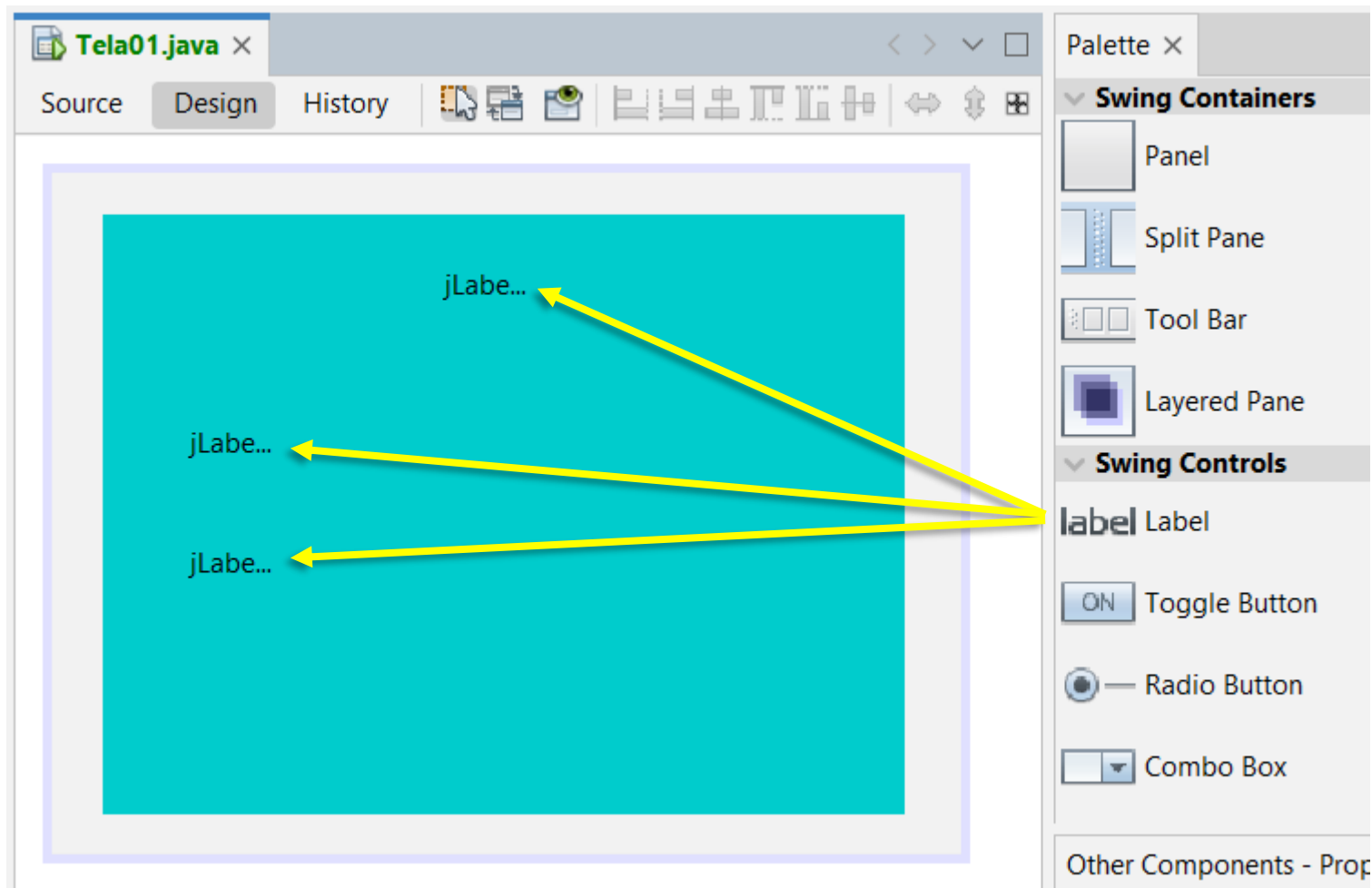


Java Swing – JLabel

- 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 light blue rectangular area representing a window. Inside, there is a grey rectangular component labeled "Tela 01". Below it, there are two labels "jLabe..." (partially visible).
- 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 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 the following properties:

Properties	Events	Code
icon		<none>
labelFor		<none>
text		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 labeled "Mens" is visible. A yellow arrow points from the "text" property in the Properties window to this "Mens" component.
- 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' tab is active, showing a visual representation of a Java Swing window. The window has a cyan background and a grey rectangular area. Inside this area, there are two labels, both labeled 'Mens...', one above the other. A yellow arrow points from the second 'Mens...' label to the 'text' property field in the 'jLabel3 [JLabel] - Properties' window. The 'Palette' window on the right shows 'Swing Containers' and 'Swing Controls'. The 'Swing Controls' section includes 'Label' and 'Button'.

Swing Containers

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

Swing Controls

- Label
- 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 design view and a properties view. A yellow arrow points from the 'Tela 0...' label in the design view to the 'font' property in the Properties window, which is set to 'Segoe UI 12 Plain'. Another yellow arrow points from the 'font' property to the 'font' dialog box, which shows 'Segoe UI' selected for font, 'Plain' for style, and '24' for size. The 'OK' button in the dialog is also highlighted with a yellow arrow.

Font Dialog Box:

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

Buttons: OK, Cancel, Help

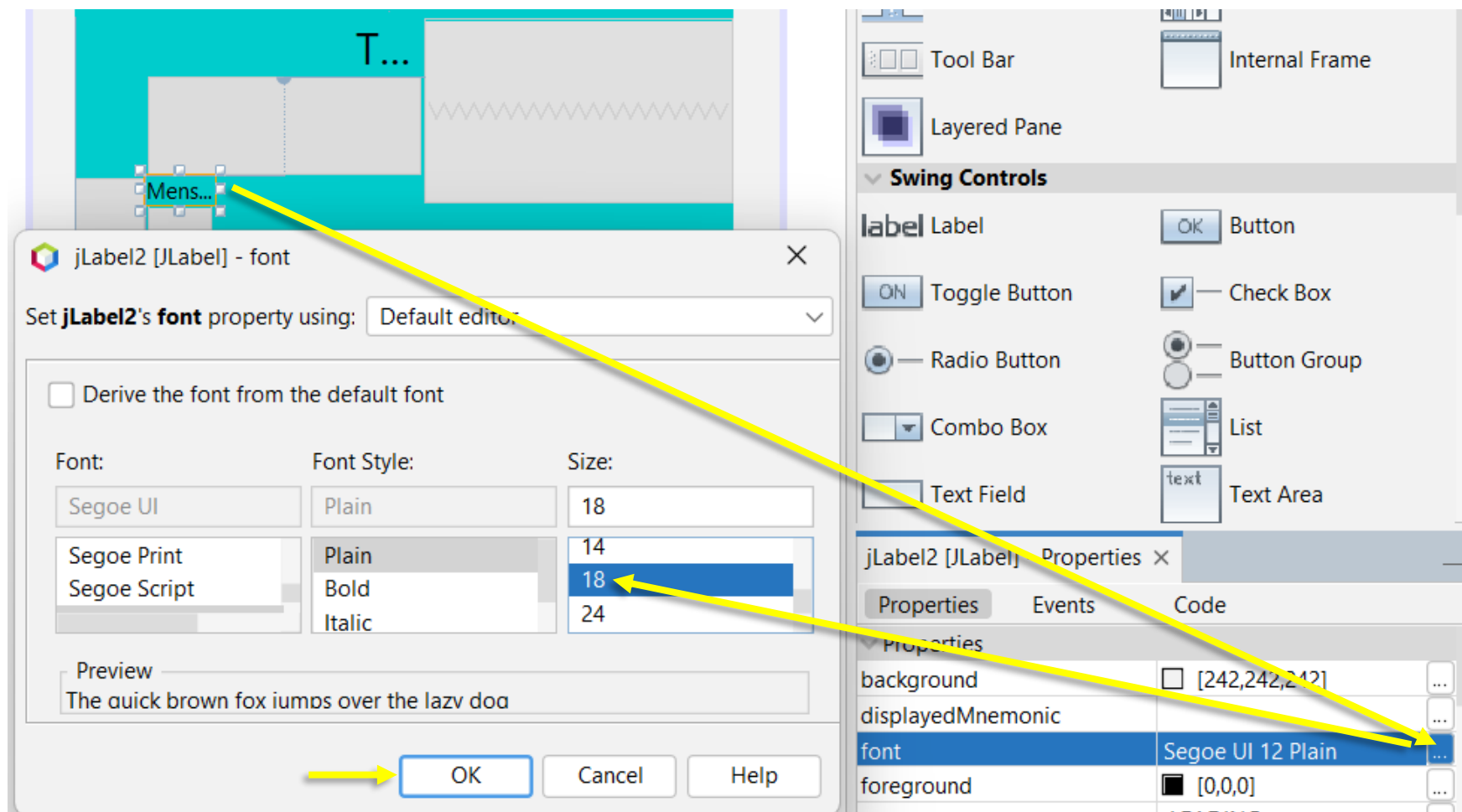
Properties Window:

jLabel1 [JLabel] - Properties

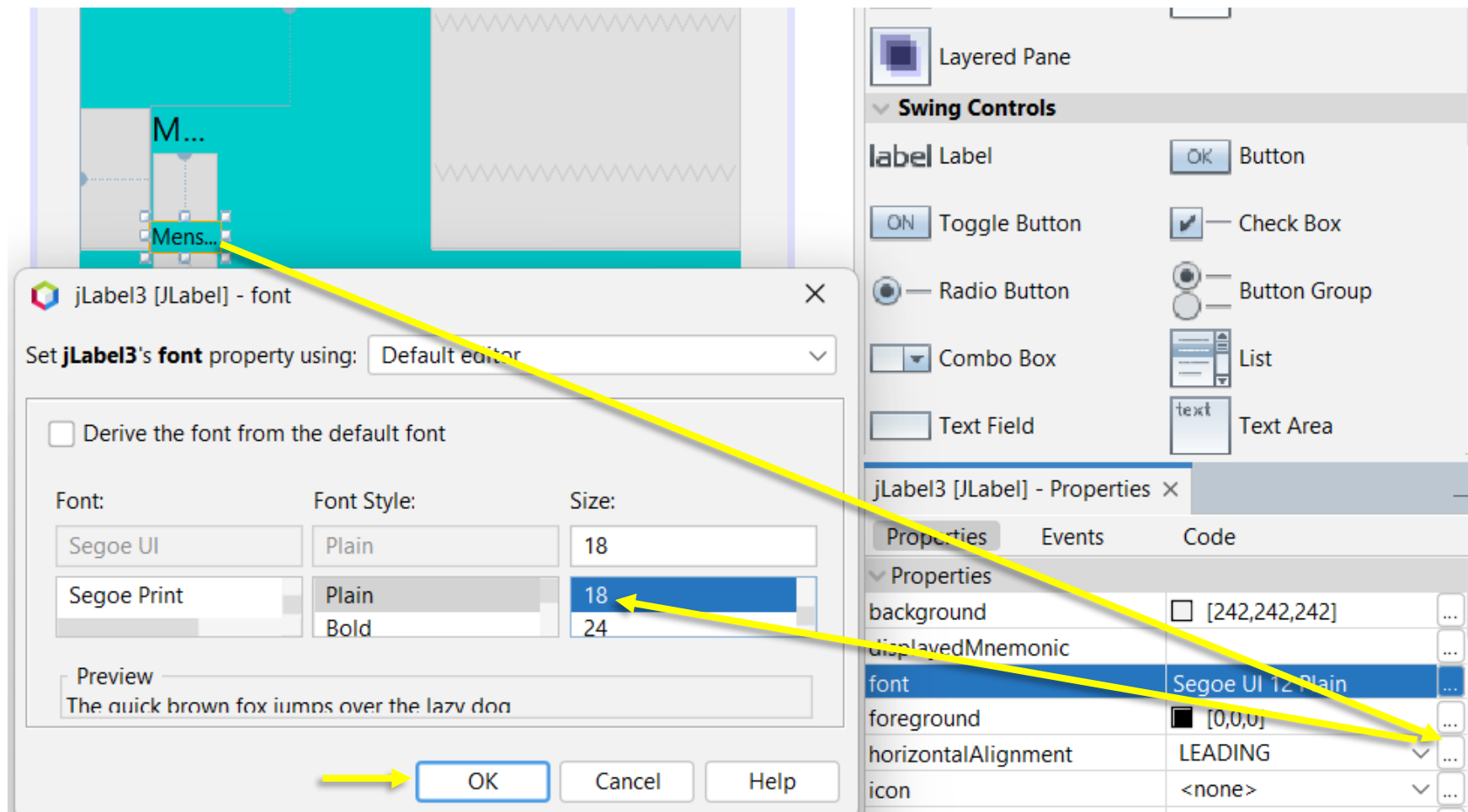
Properties	Events	Code
background		[242,242,242]
displayedMnemonic		
font		Segoe UI 12 Plain
foreground		[0,0,0]



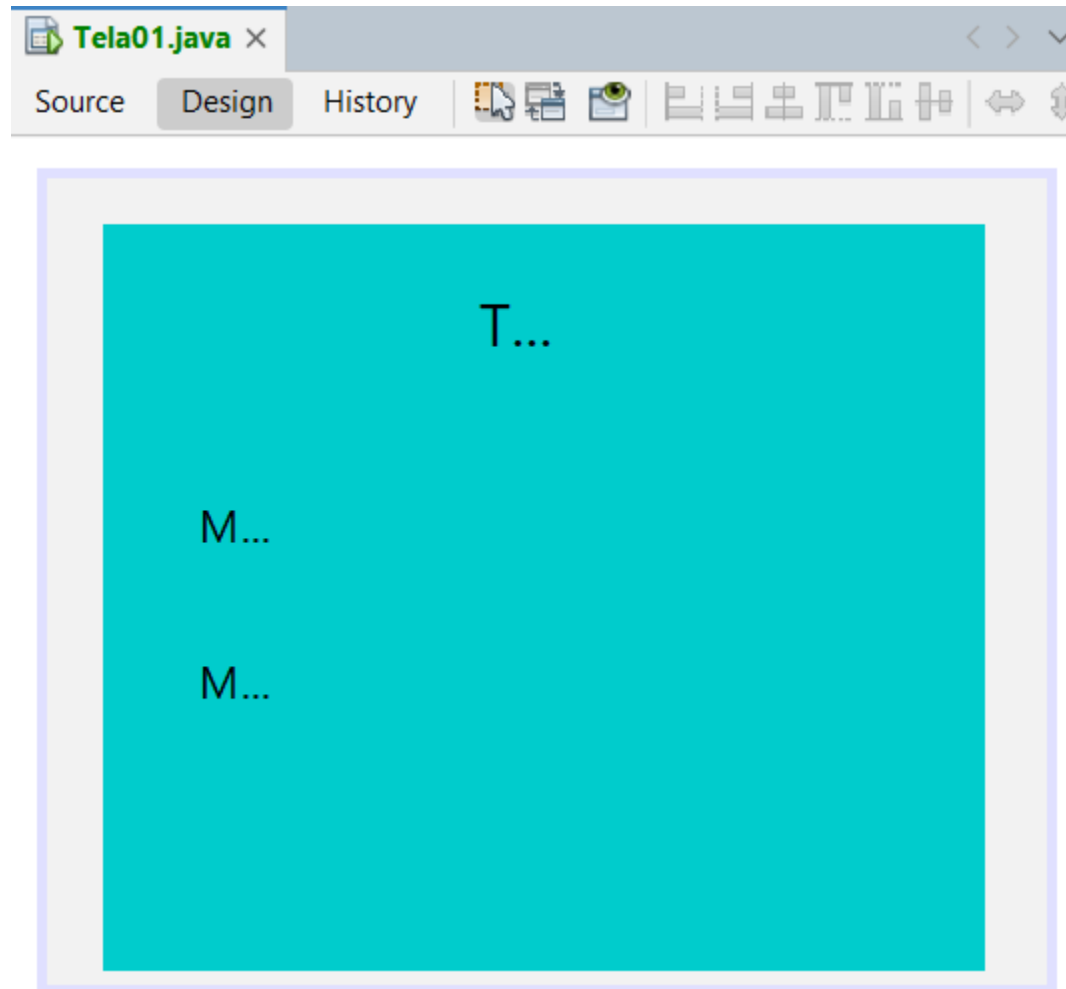
Java Swing – JLabel – Aumentar Fonte



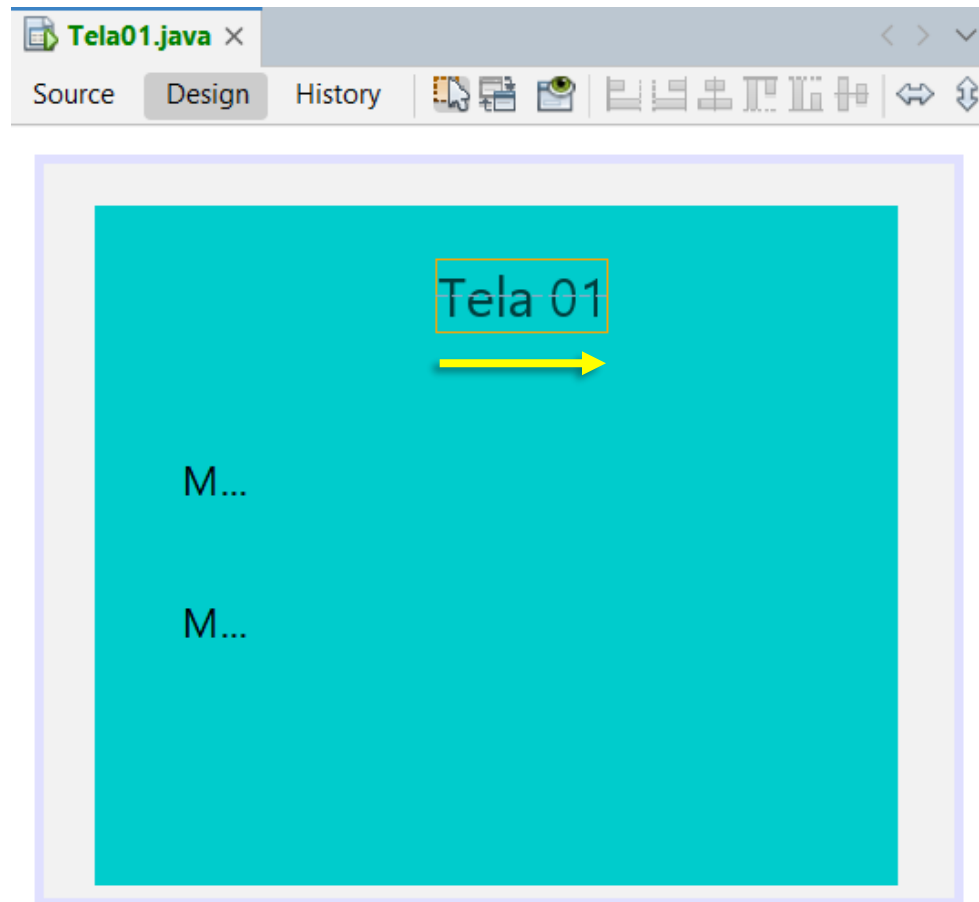
Java Swing – JLabel – Aumentar Fonte



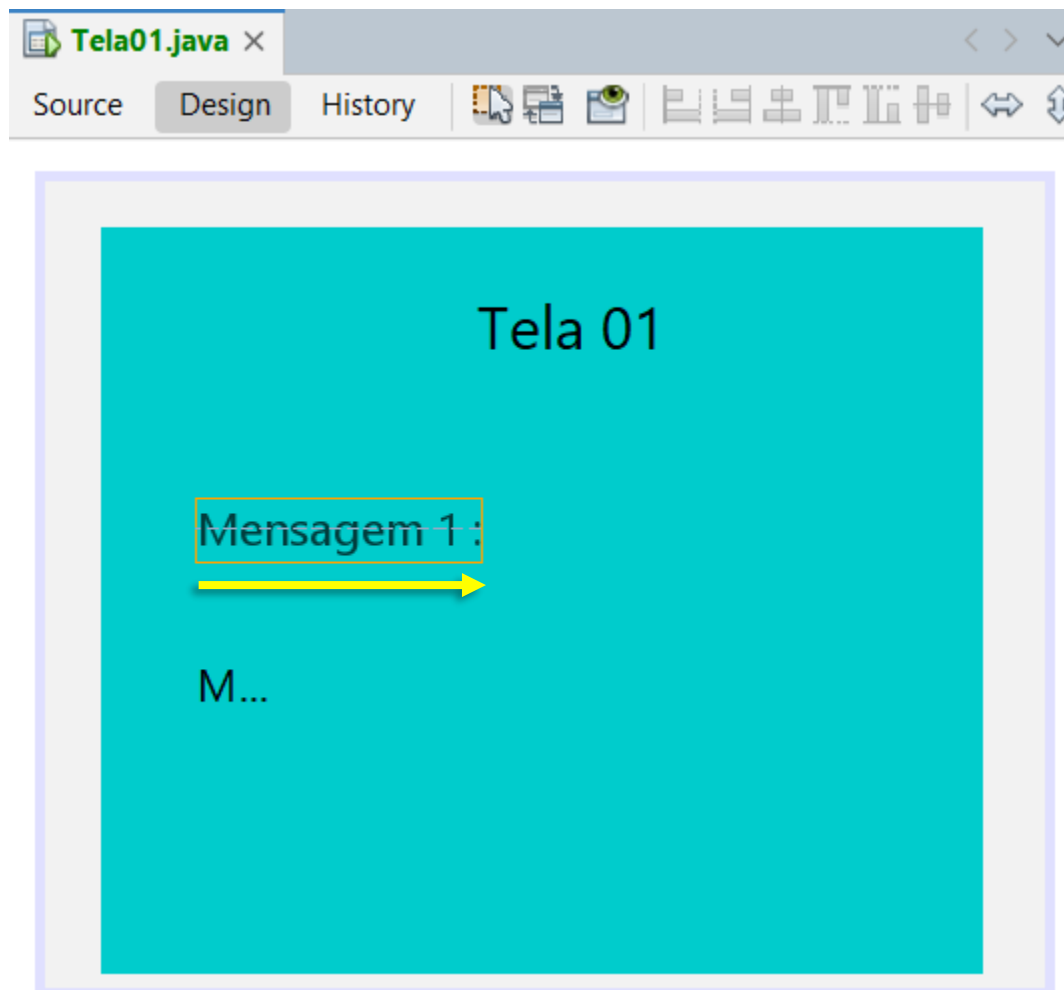
Java Swing – JLabel – Aumentar Fonte



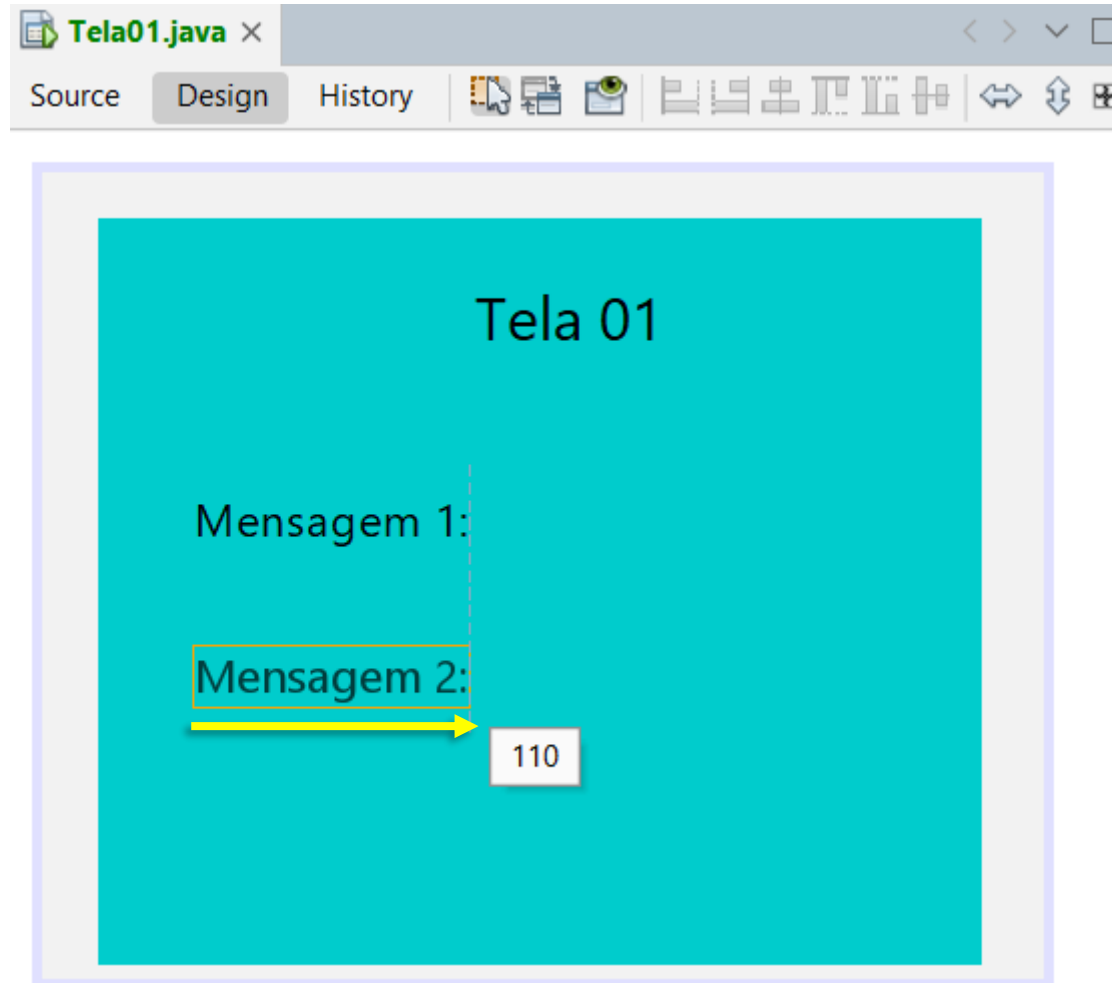
Java Swing – JLabel – Redimensionar



Java Swing – JLabel – Redimensionar



Java Swing – JLabel – Redimensionar

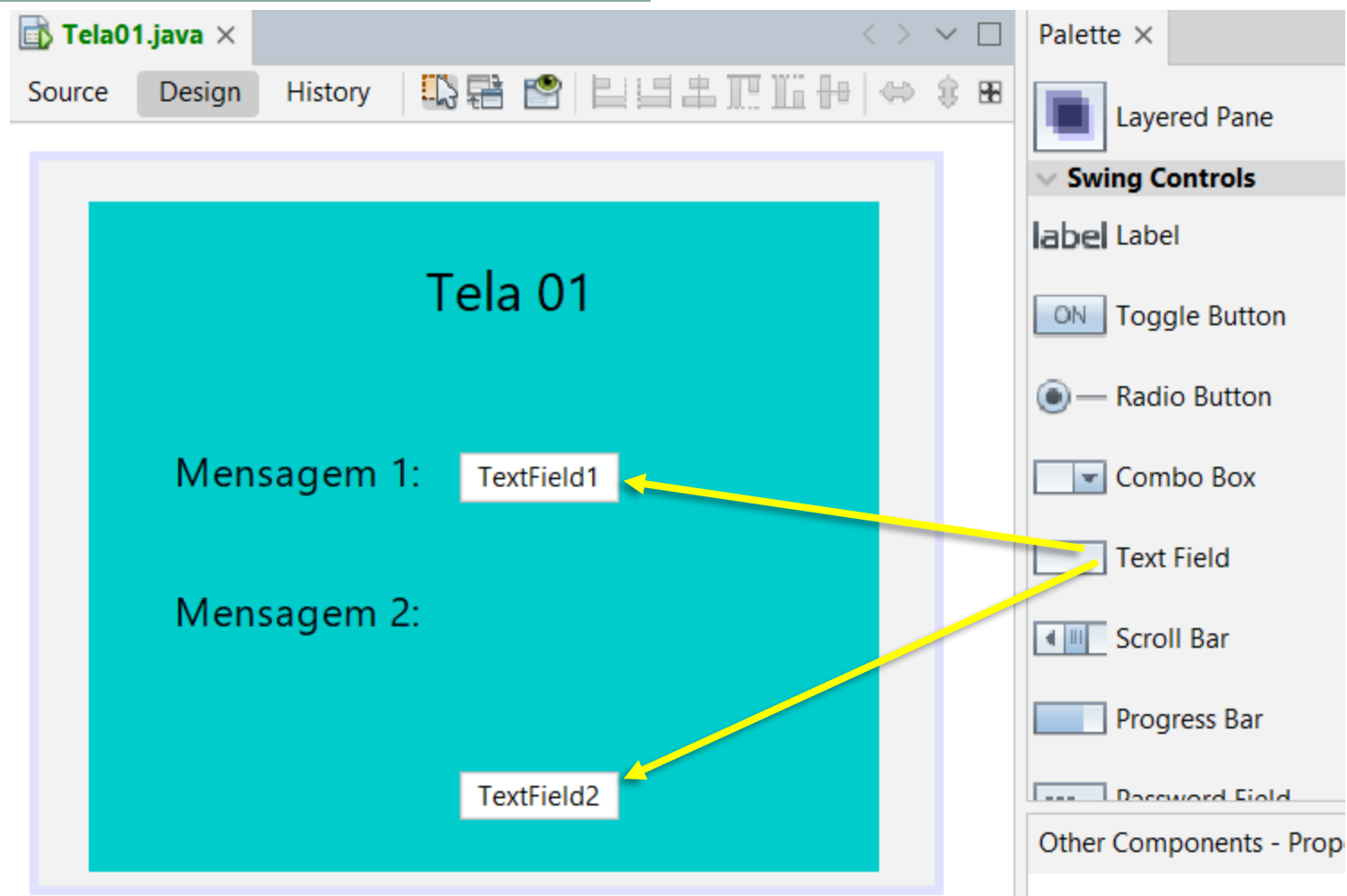


Java Swing – JTextField

- 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', 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
foreground	[0,0,0]
horizontalAlignment	LEADING
text	
toolTipText	
Other Properties	
UI	<default>



Java Swing – JTextField – Editor Texto

The image shows a screenshot of an IDE (Integrated Development Environment) with a Java Swing window titled "Tela 01". The window has a cyan background and contains two text 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.

The Properties window shows the following properties for the selected component:

Properties	Events	Code
foreground		■ [0,0,0] ...
horizontalAlignment		LEADING ...
text		... (highlighted by a yellow arrow)
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 'Variable Name' set to 'txtMensagemFinal'.

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 to configure the font for the selected component. It includes a checkbox for "Derive the font from the default font", fields for "Font" (Segoe UI), "Font Style" (Plain), and "Size" (18). 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 "font" property set to "Segoe UI 12 Plain". A yellow arrow points from the "font" property in the Properties window to the "font" property in the Properties window. Another yellow arrow 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.



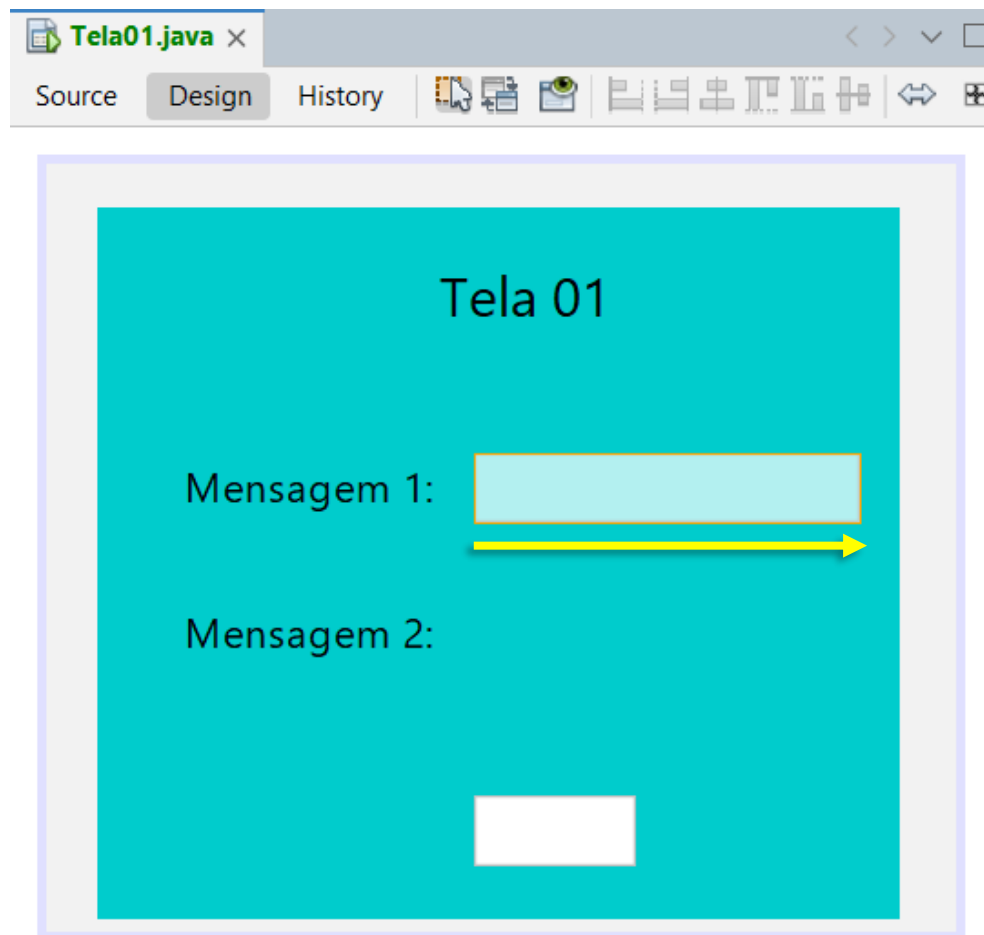
Java Swing – JTextField – Aument. Fonte

The image shows two windows from a Java Swing IDE. The left window, titled 'txtMensagemFinal [JTextField] - font', is used to set the font for a specific text field. It has a 'Set txtMensagemFinal's font property using:' dropdown set to 'Default editor'. Below this, there's a checkbox 'Derive the font from the default font' which is unchecked. The 'Font' section has 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 right window. The 'Preview' section shows the text 'The quick brown fox jumps over the lazy dog'. At the bottom are 'OK', 'Cancel', and 'Help' buttons. A yellow arrow points to the 'OK' button. The right window, titled 'txtMensagemFinal [JTextField] - Properties', shows a list of properties for the text field. The 'font' property is highlighted in blue, and its value is 'Segoe UI 12 Plain'. A yellow arrow points from this property to the 'font' property in the left window. The 'Properties' tab is active, showing a table of properties and their values.

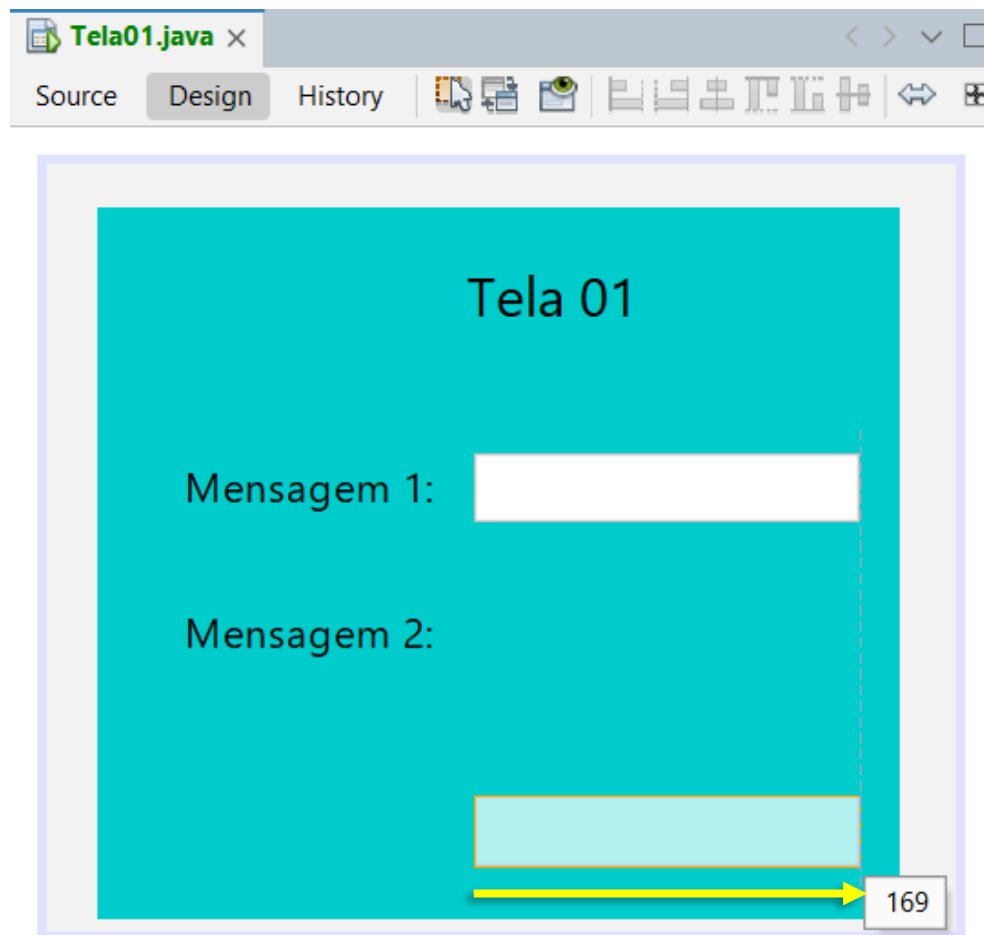
Properties	Events	Code
columns		0
document		<default>
font		Segoe UI 12 Plain
foreground		[0,0,0]
horizontalAlignment		LEADING
text		



Java Swing – JTextField – Redimens.



Java Swing – JTextField – Redimens.



Java Swing – JTextField – Não Editável

The image shows a Java Swing application 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, and a yellow arrow points from it 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

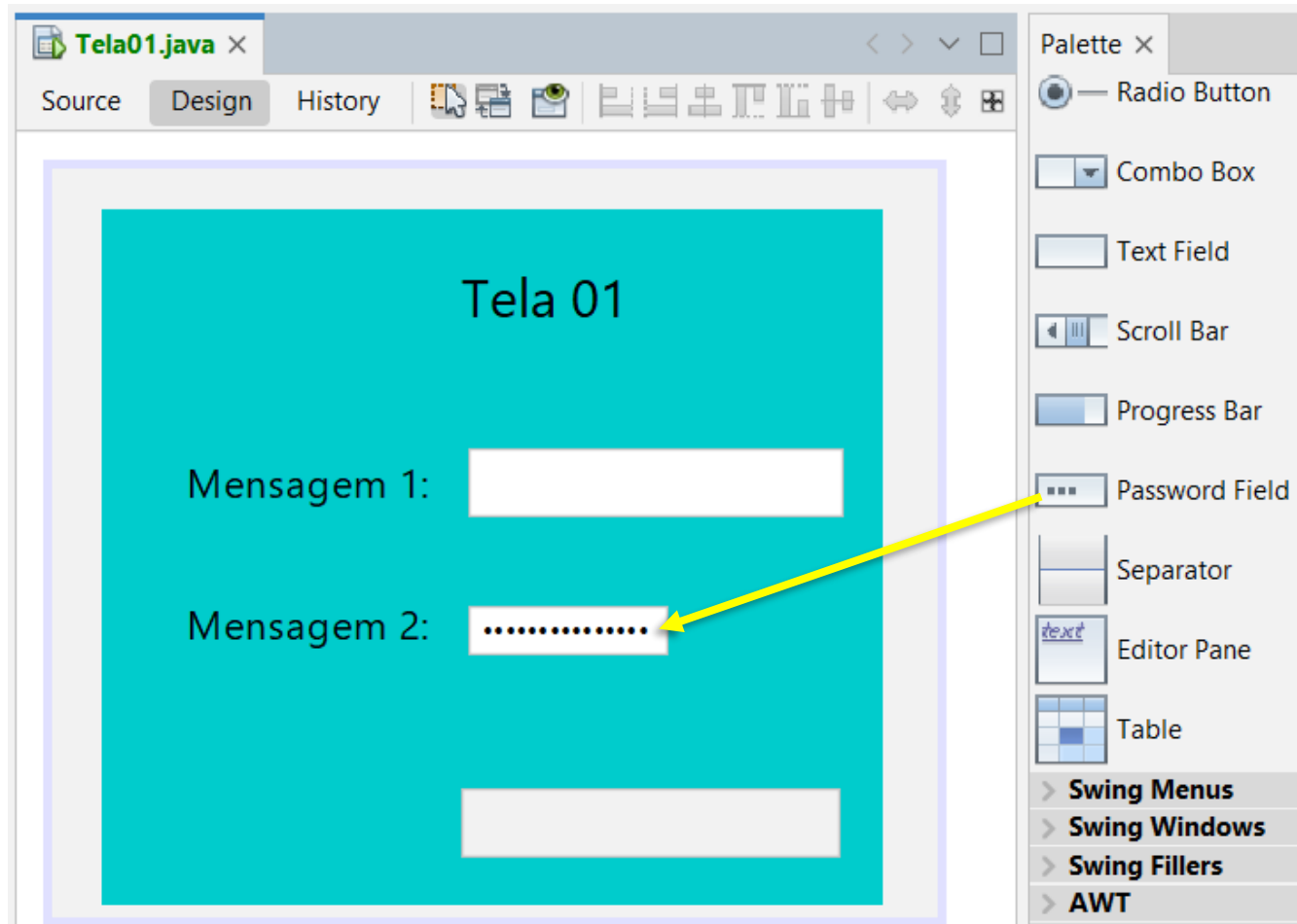


Java Swing – JPasswordField

- 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 image shows a Java Swing IDE window titled 'Tela01.java'. The 'Design' tab is active, displaying a visual representation of a Java Swing window titled 'Tela 01'. The window has a cyan background and contains two labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text input field. The 'Mensagem 2:' field is a `JPasswordField`, indicated by the masked characters. A yellow arrow points from the `JPasswordField` component in the design view to the 'jPasswordField1 [JPasswordField] - Properties' window. The Properties window shows the following properties:

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

The 'jPasswordField1 [JPasswordField]' component is selected, and the Properties window is open.



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 labels, 'Mensagem 1:' and 'Mensagem 2:', each followed by a text input field. The 'Mensagem 2:' field is a JPasswordField, indicated by the small dots on its right side. A yellow arrow points from this JPasswordField to the 'txpMensagem2' property in the Properties window.

The Properties window on the right shows the 'txpMensagem2 [JPasswordField] - Properties' tab. The 'Code Generation' section is expanded, showing the following properties:

Property	Value
Bean Class	class javax.swing.JPassword...
Variable Name	txpMensagem2
Variable Modifiers	private

The bottom of the Properties window shows the name 'txpMensagem2 [JPasswordField]'.



Java Swing – JPasswordField – Aum. F.

The image shows a Java Swing IDE with a window titled 'Tela01.java'. A dialog box titled 'txpMensagem2 [JPasswordField] - font' is open, allowing the user to set the font for the 'txpMensagem2' JPasswordField. The dialog has a 'Default editor' button and a checkbox 'Derive the font from the default font' which is unchecked. The 'Font' field is set to 'Segoe UI', the 'Font Style' is 'Plain', and the 'Size' is '18'. Below these fields are 'OK', 'Cancel', and 'Help' buttons. A yellow arrow points from the 'OK' button to the 'txpMensagem2 [JPasswordField] - Properties' window. The 'Properties' window shows the 'font' property set to 'Segoe UI 12 Plain'. Another yellow arrow points from the 'font' property to the 'txpMensagem2 [JPasswordField] - font' dialog. The background shows a Java Swing window with a label 'Mensagem 2:' and a JPasswordField.

Set **txpMensagem2**'s **font** property using: Default editor

☐ Derive the font from the default font

Font: Segoe UI Font Style: Plain Size: 18

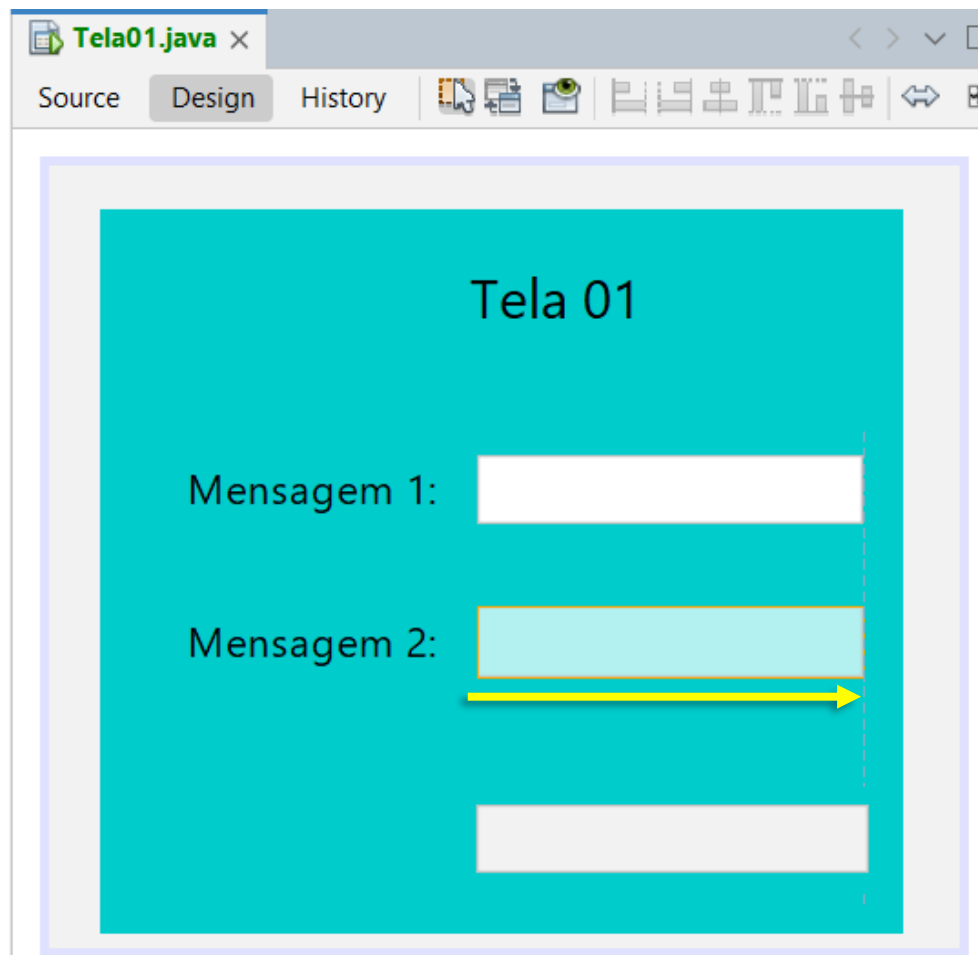
OK Cancel Help

txpMensagem2 [JPasswordField] - Properties

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



Java Swing – JPasswordField – Redimen.

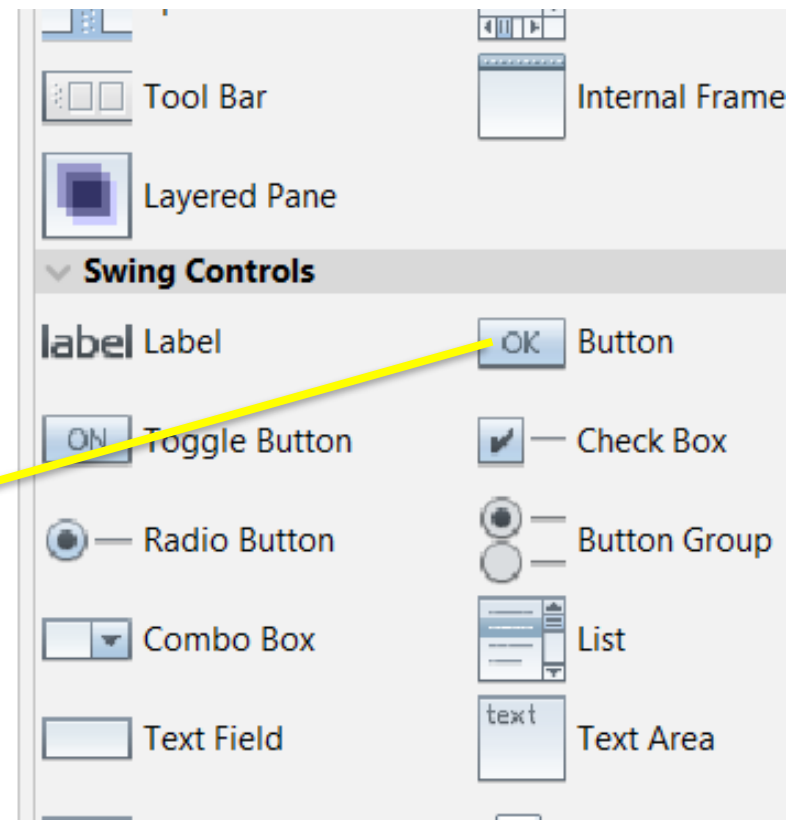
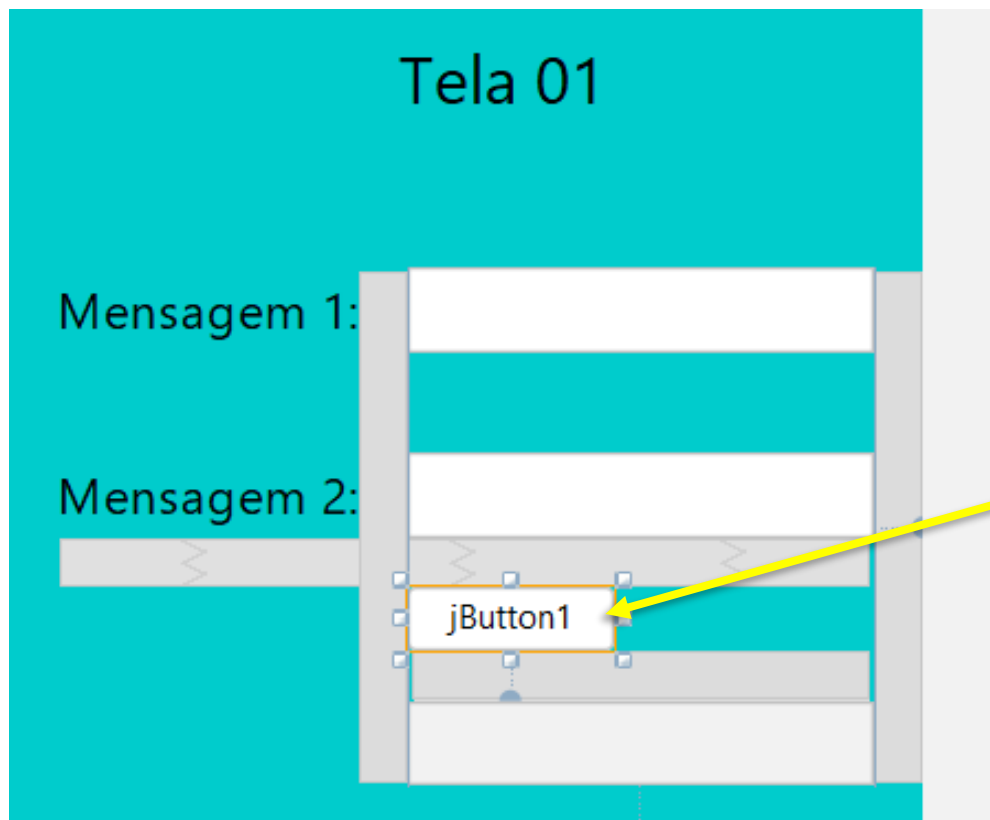


Java Swing – JButton

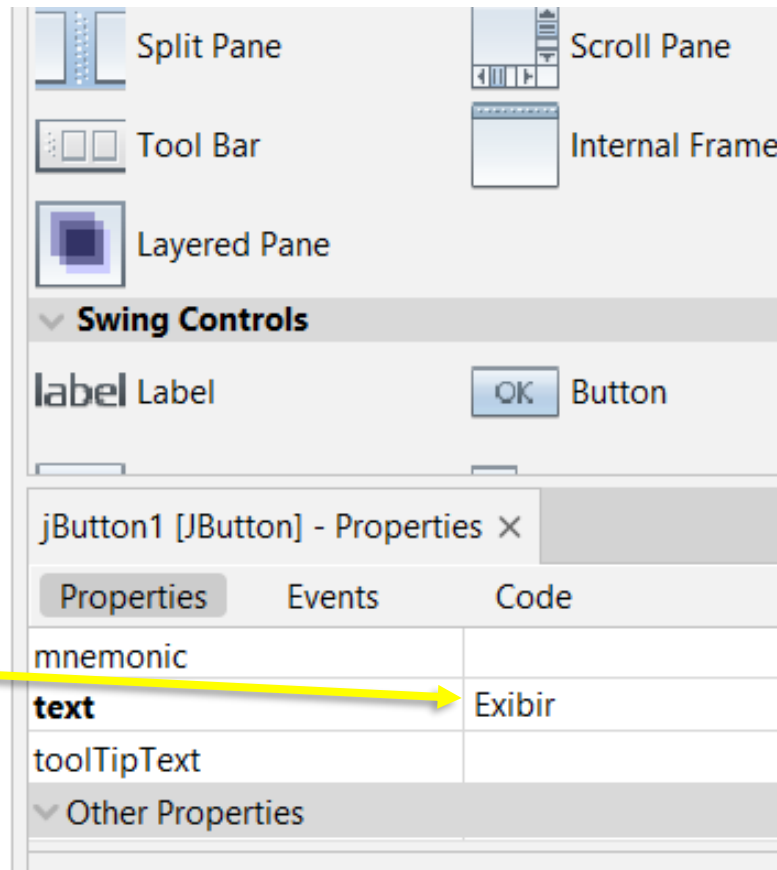
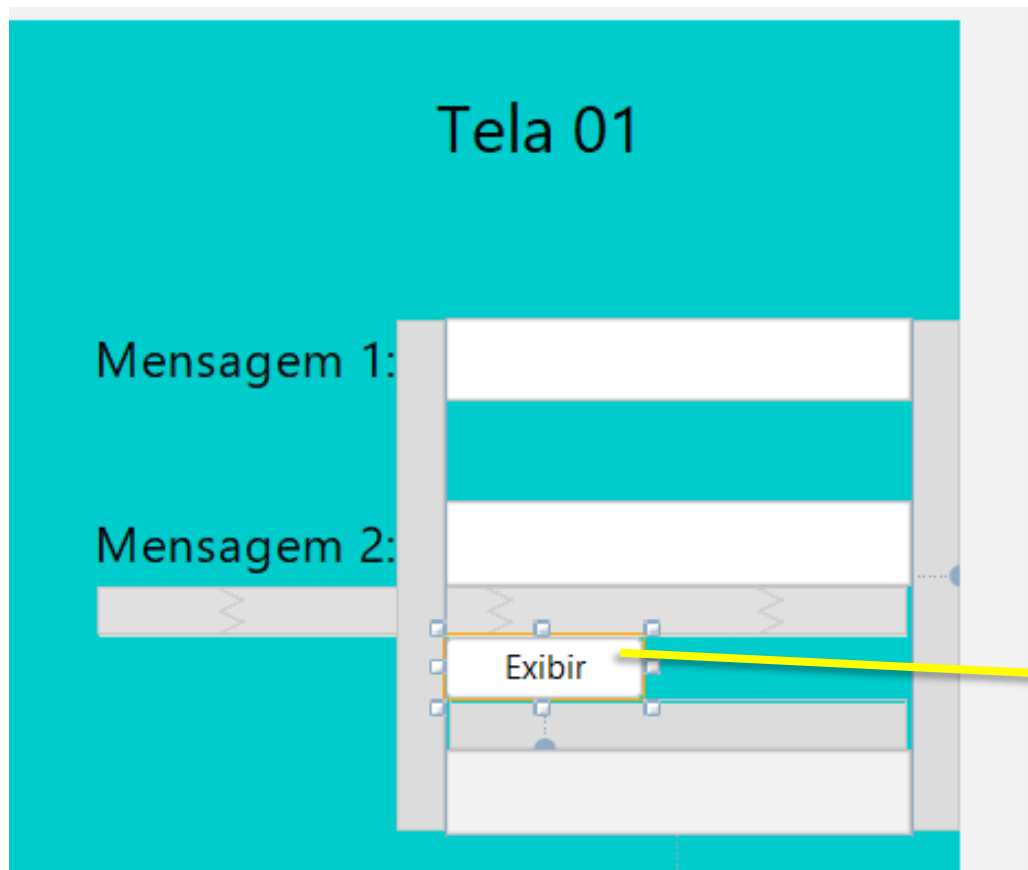
- A classe **JButton** permite **definir botões gráficos** que **pode ser adicionados** a um **outro componente** (frame, painel etc.).
- Um **botão** pode **ser criado** com apenas um **texto e/ou com ícones** para tornar o ambiente mais intuitivo.



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, and its value is '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

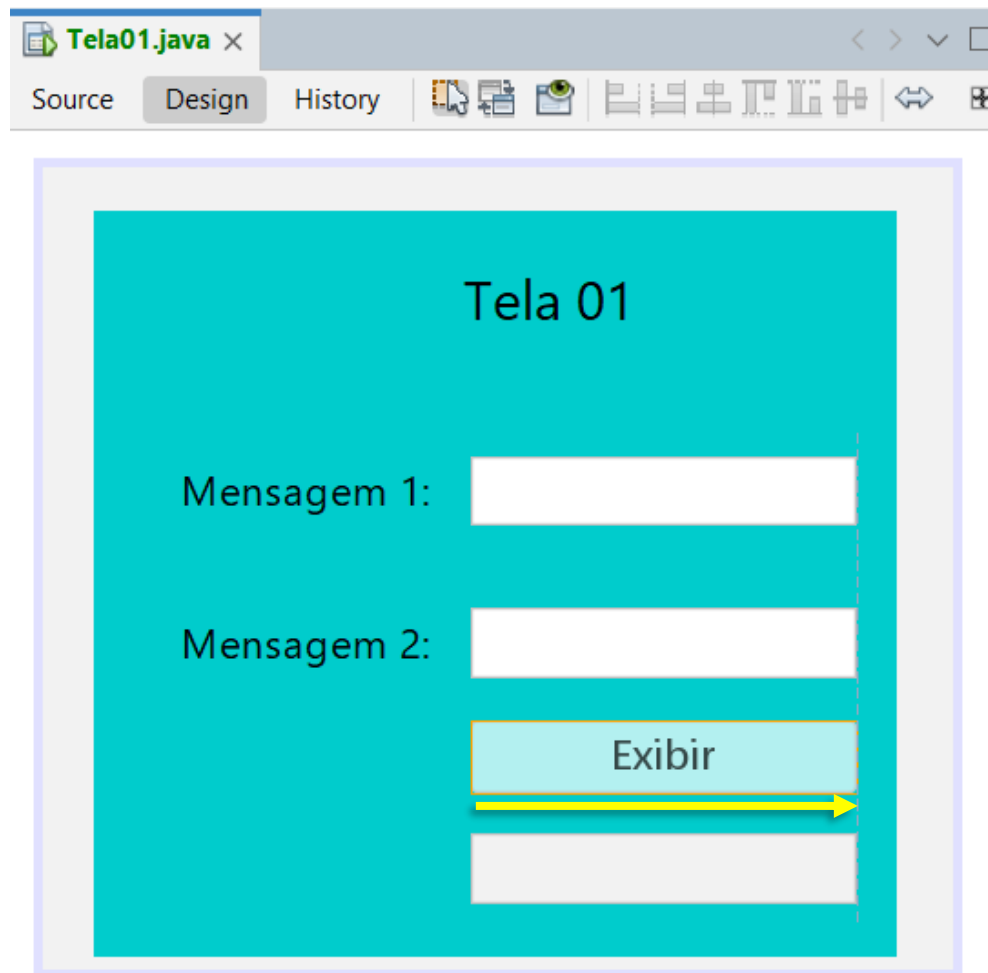
- Label
- Button

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.

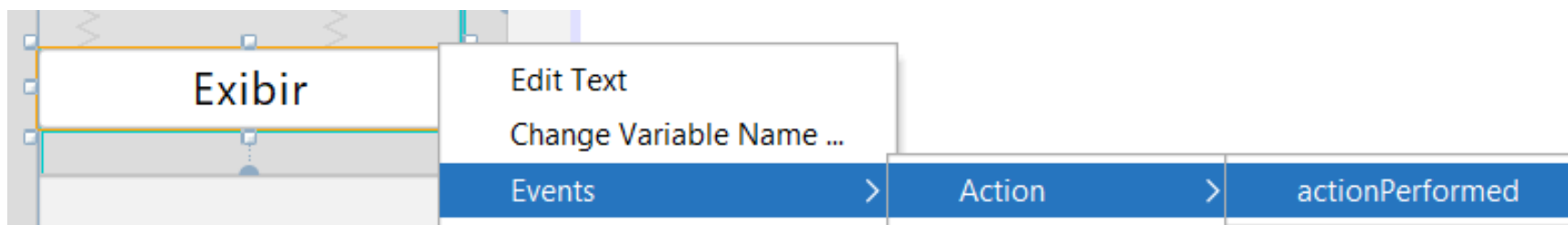
The image shows a Java Swing IDE interface. The main window, titled 'Tela 01', is in Design mode. It contains a cyan background with the text 'Tela 01'. Below this, there are two labels: 'Mensagem 1:' and 'Mensagem 2:'. To the right of these labels is a vertical stack of components. The bottom component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



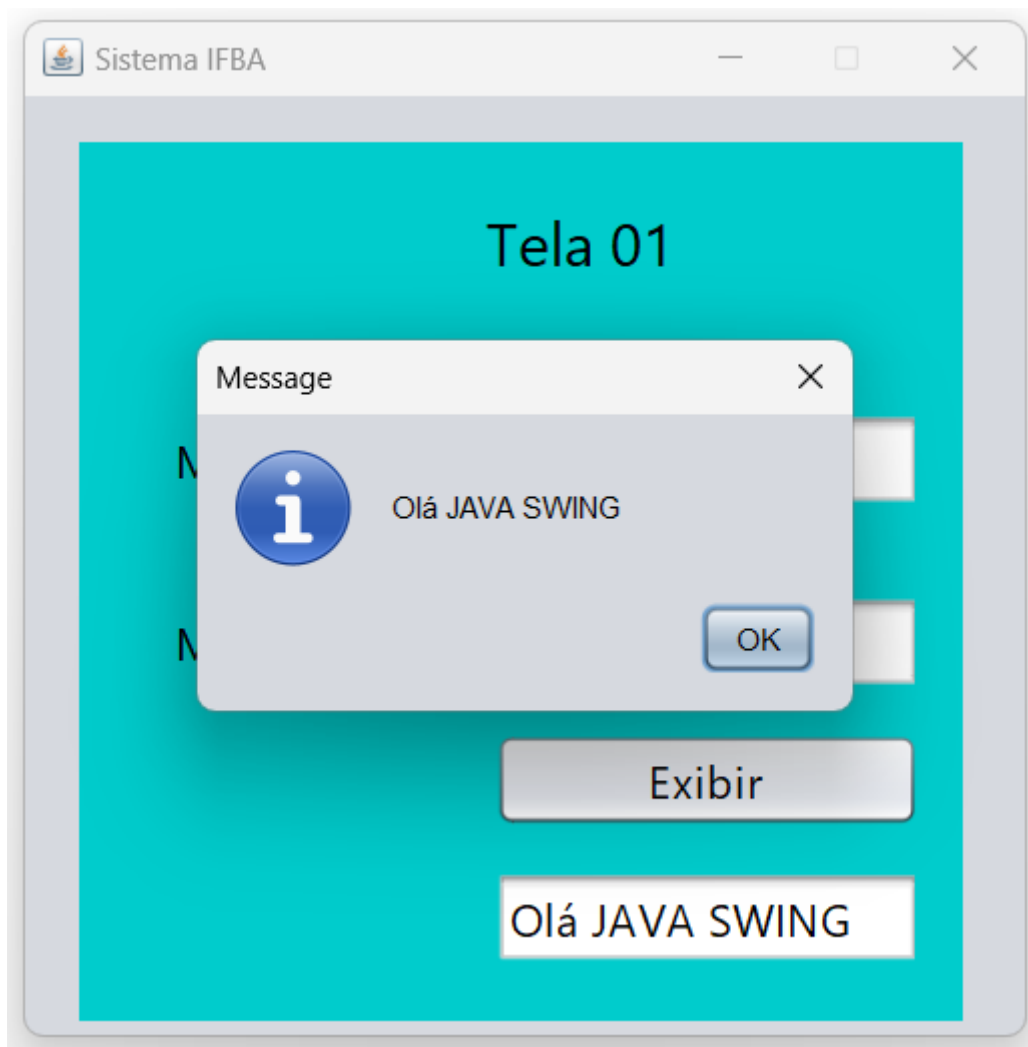
Java Swing – JButton – Execução



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.



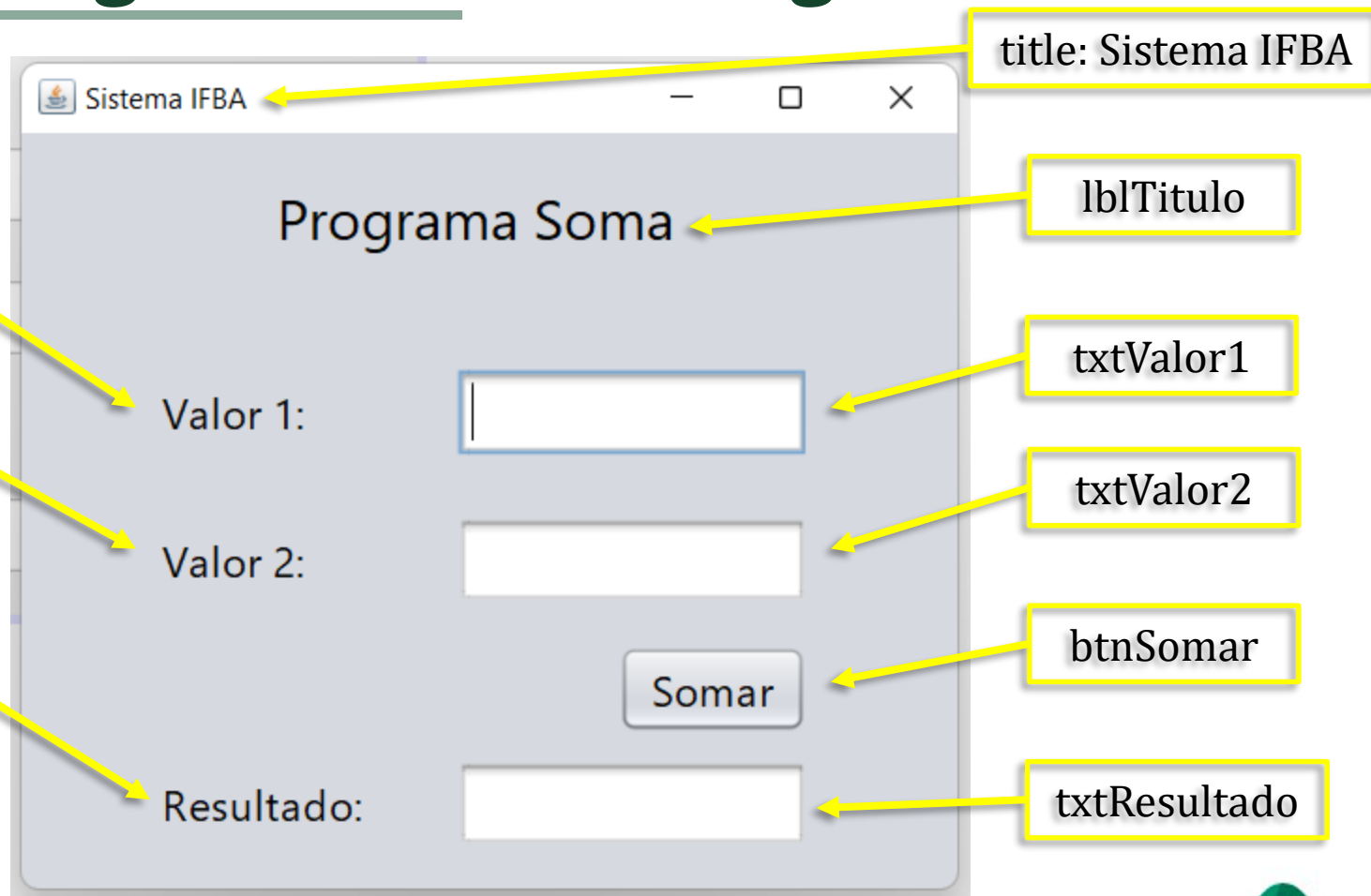
Java Swing – JButton – Execução



Java Swing – JButton – Execução



Java Swing – Exercício – Prog. Soma



TelaSoma.java



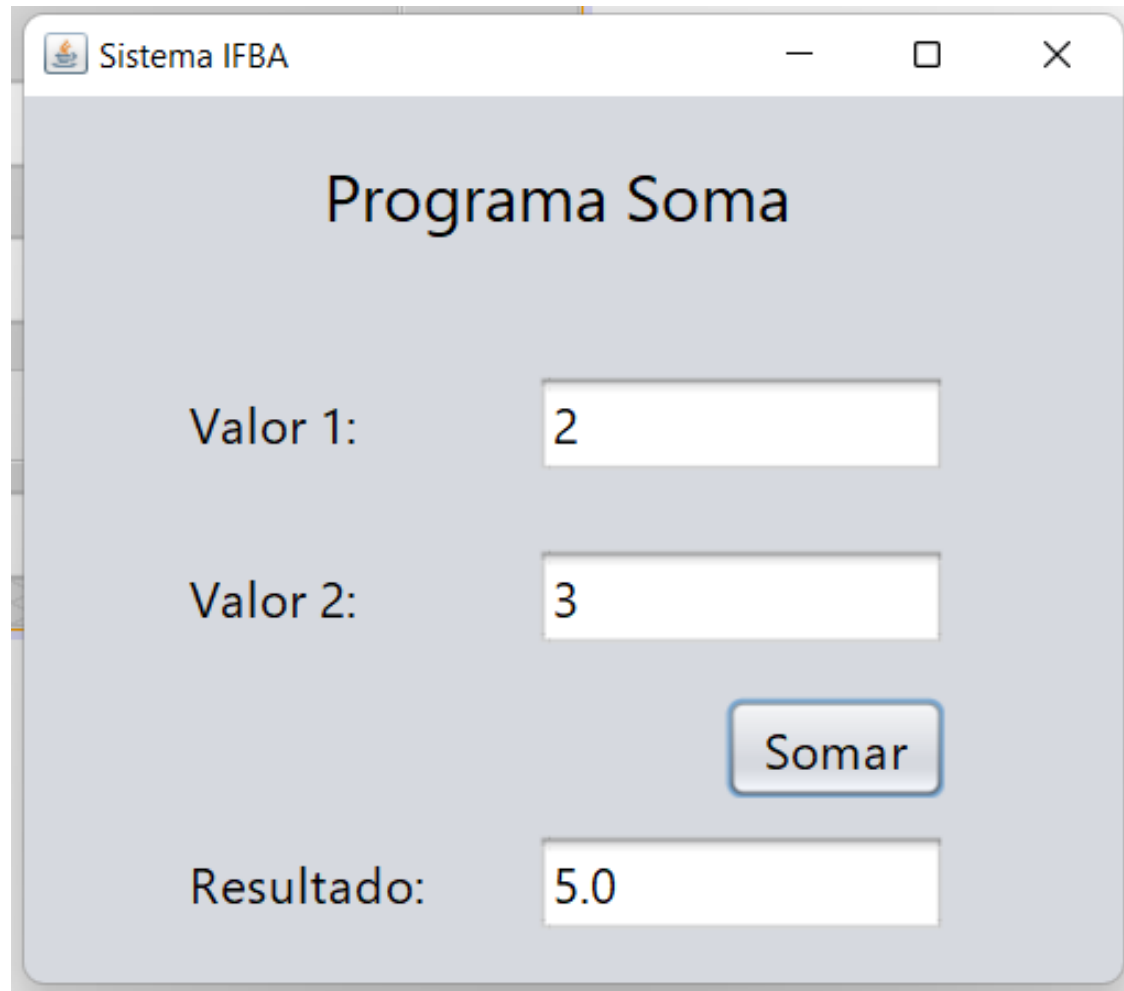
Java Swing – Exercício – Prog. Soma

```
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

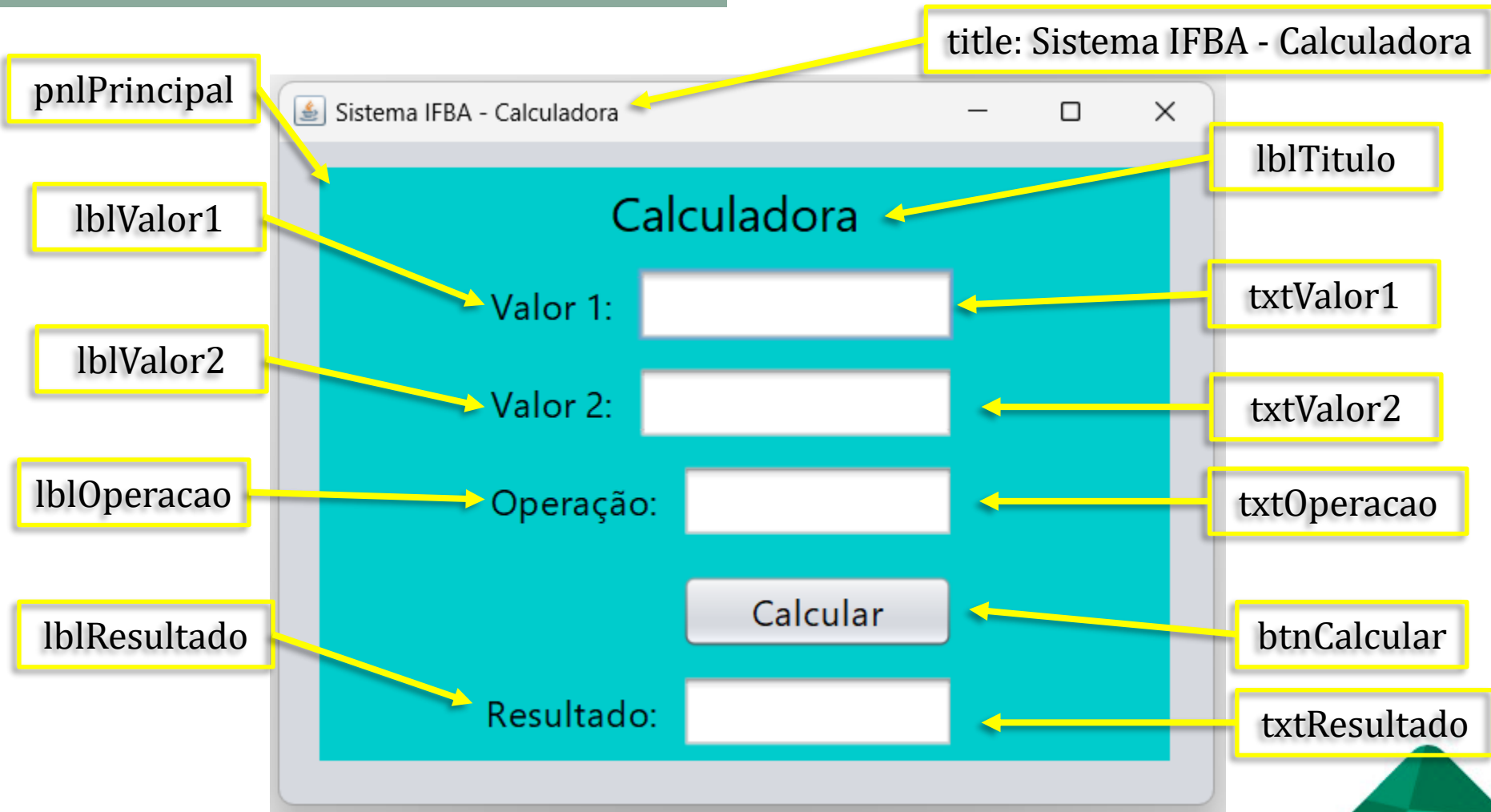
Valor 2: 3

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.

Execução

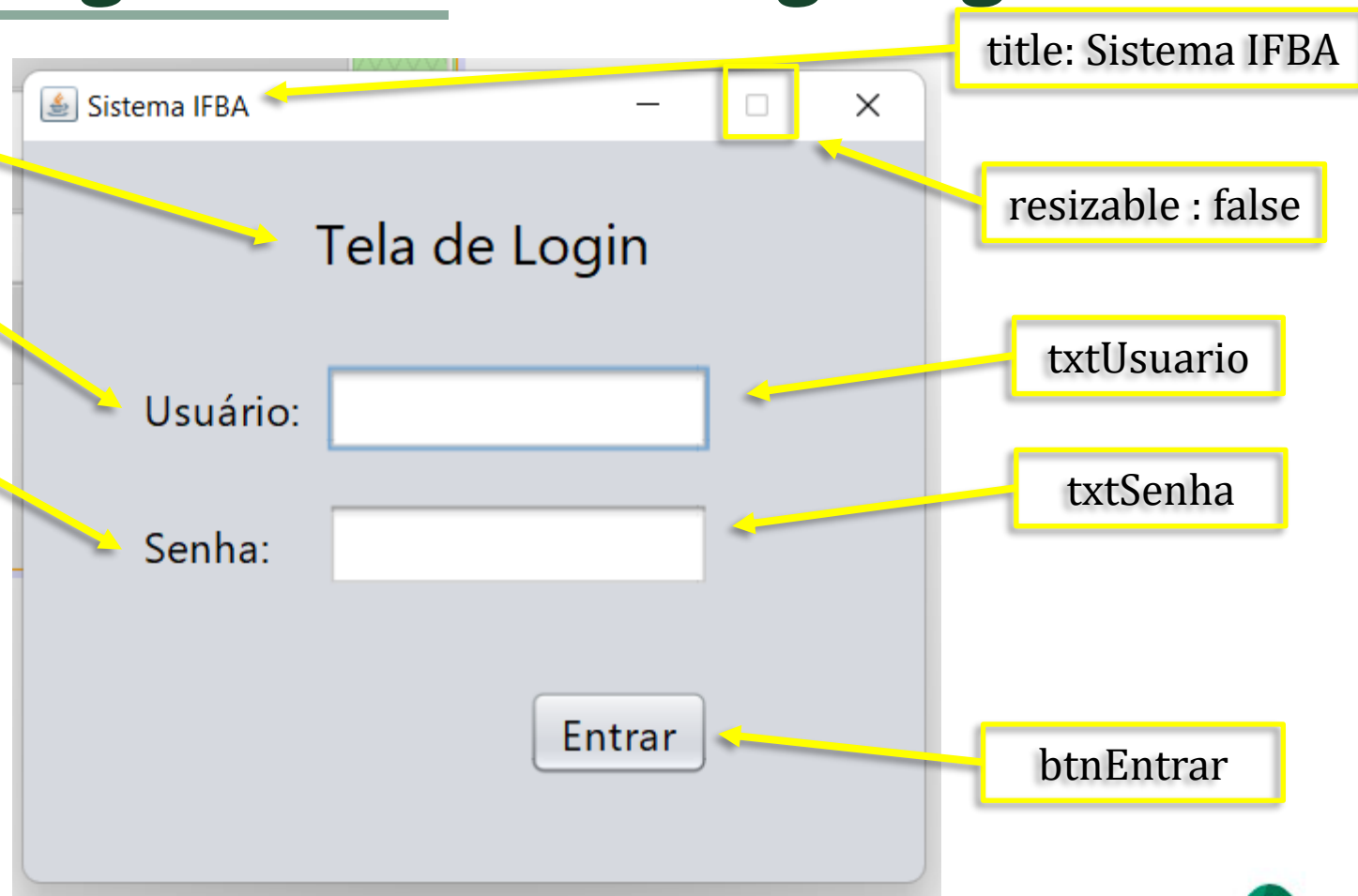
The screenshot shows a Java Swing window titled "Sistema IFBA - Calculadora". The window has a light blue background and contains the following elements:

- 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 separate box containing the operators "+", "-", "*", and "/" to the "Operação" input field.



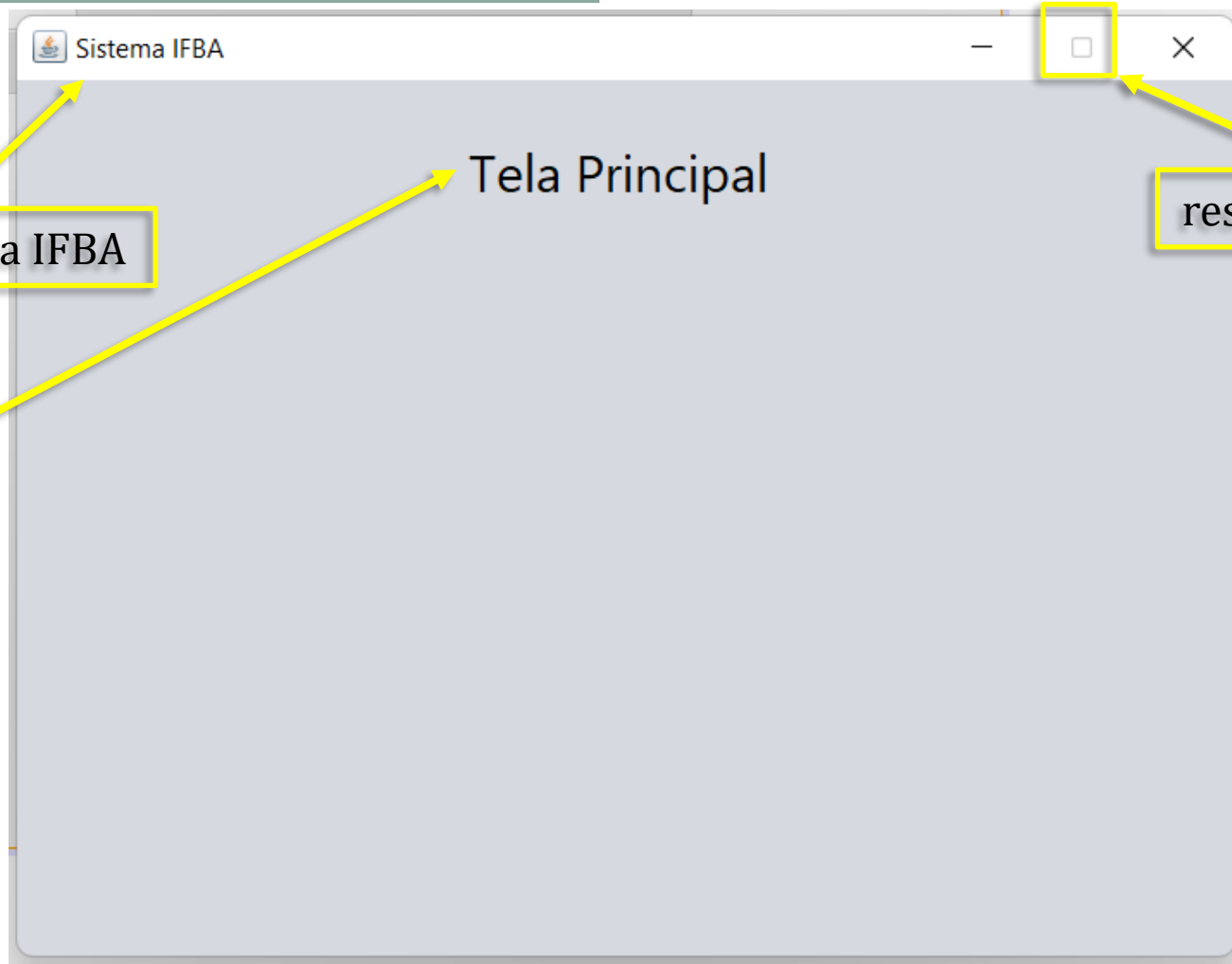
Java Swing – Exercício – Prog. Login



TelaDeLogin.java



Java Swing – Exercício – Prog. Principal



title: Sistema IFBA

lblTitulo

resizable : false

TelaPrincipal.java



INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Bahia

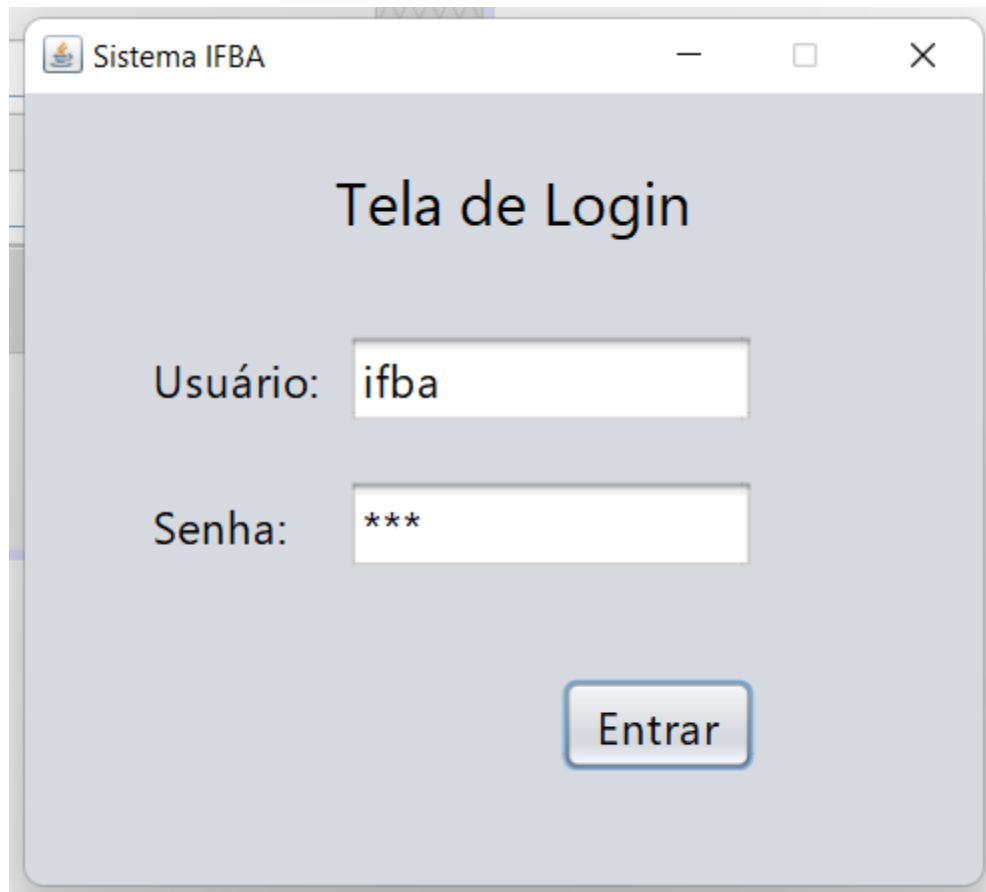
Java Swing – Exercício – Prog. Login

```
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

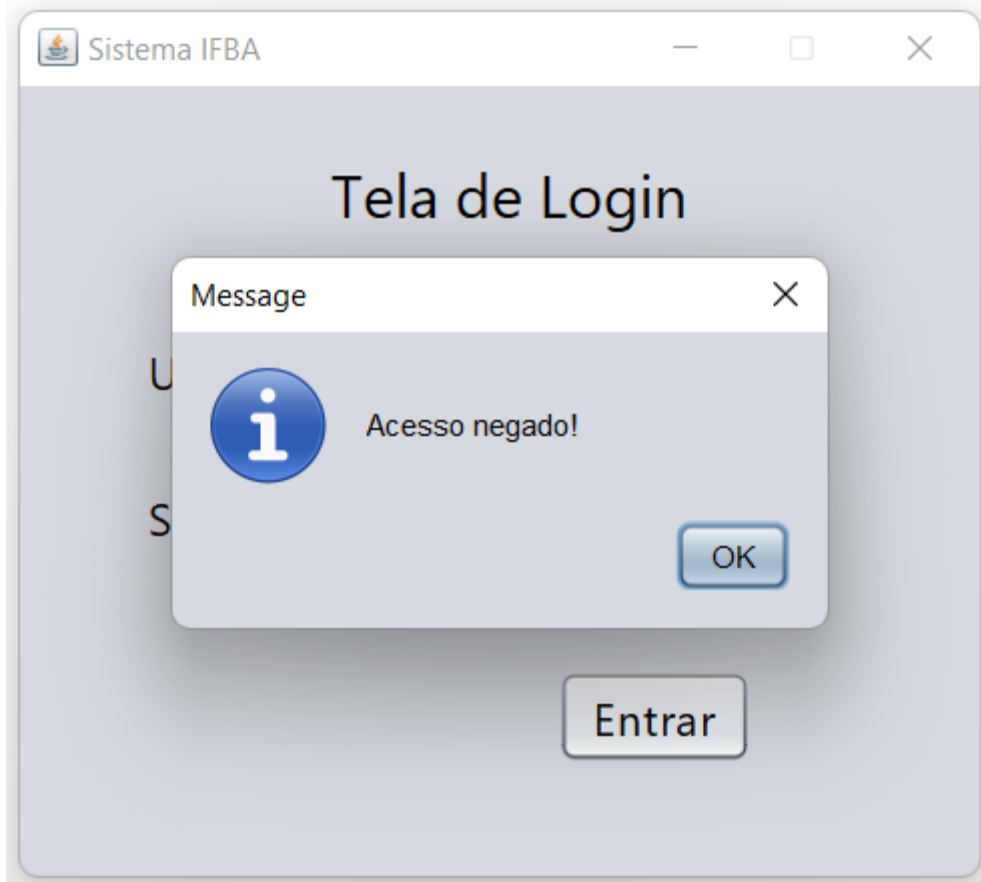


The screenshot shows a Java Swing window titled "Sistema IFBA" with standard window controls (minimize, maximize, close). The window contains a login form titled "Tela de Login". It features two text input fields: one for the username labeled "Usuário:" with the text "ifba" entered, and another for the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



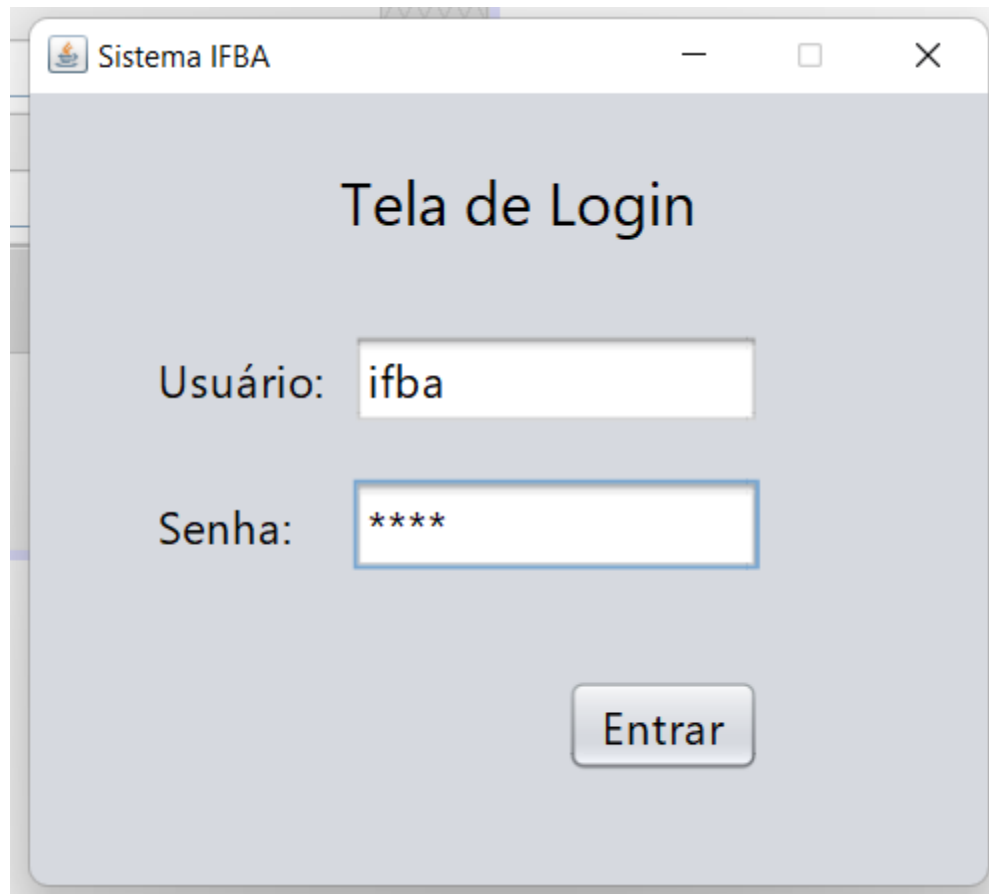
Java Swing – Exercício – Prog. Login

Execução



Java Swing – Exercício – Prog. Login

Execução

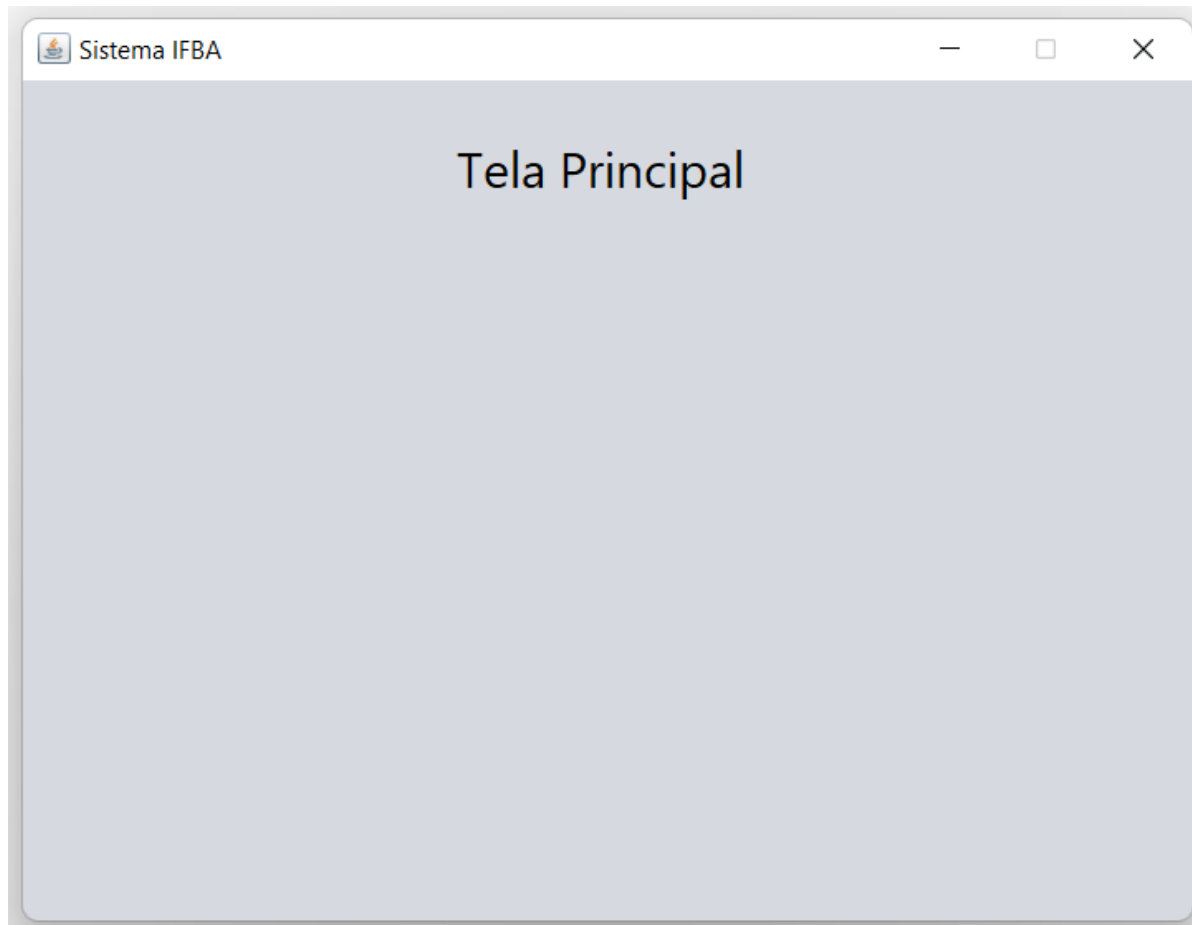


The screenshot shows a Java Swing window titled "Sistema IFBA" with a standard Mac OS-style title bar (minimize, maximize, close buttons). The window content has a light gray background and is titled "Tela de Login" in a large, black, sans-serif font. Below the title, there are two input fields. The first is labeled "Usuário:" and contains the text "ifba". The second is labeled "Senha:" and contains four asterisks "****". Both labels are in a black, sans-serif font. Below the input fields, there is a single button labeled "Entrar" in a black, sans-serif font. The button has a light gray background and a thin black border.



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.

