

INTERFACE GRÁFICA DO USUÁRIO

Alexandre Mendonça Fava
alexandre.fava@udesc.br

Universidade do Estado de Santa Catarina – UDESC
Programa de Pós-graduação em Computação Aplicada – PPGCA

GRAPHICAL USER INTERFACE

Alexandre Mendonça Fava
alexandre.fava@udesc.br

Universidade do Estado de Santa Catarina – UDESC
Programa de Pós-graduação em Computação Aplicada – PPGCA

GUI

Alexandre Mendonça Fava
alexandre.fava@udesc.br

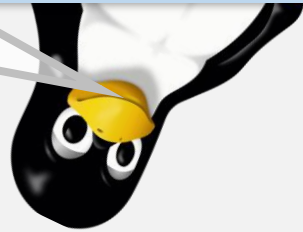
Universidade do Estado de Santa Catarina – UDESC
Programa de Pós-graduação em Computação Aplicada – PPGCA

Roteiro

- ☐ Definição
 - ☐ Classe
 - ☐ Objeto
- ☐ Construtor
- ☐ Classes Invólucro
- ☐ String
- ☐ Scanner
- ☐ Resumo
- ☐ Referências

Definição

Eu prefiro Interface de Linha de Comandos



O que é uma Interface Gráfica do Usuário?

VOZ

GESTOS

CLIQUE

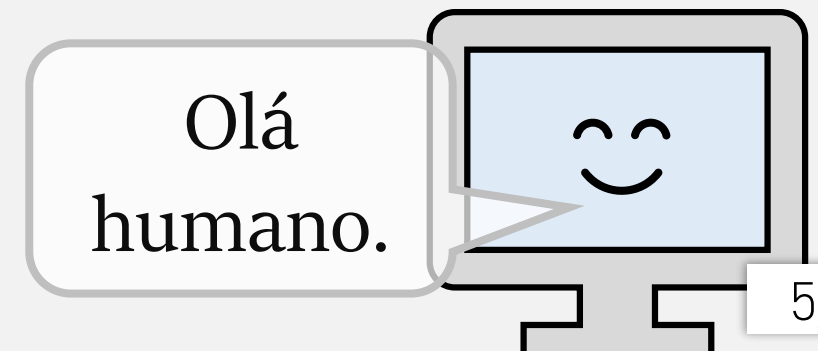
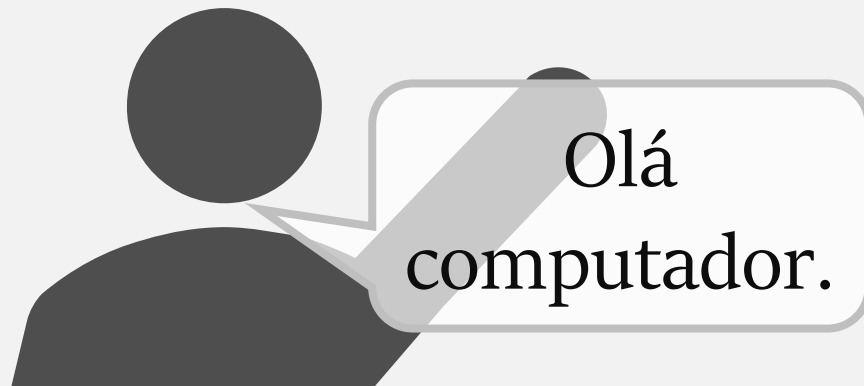
É uma forma de interação
entre o usuário e o programa
baseada em **elementos visuais***

*mas não somente

BOTÕES

ÍCONES

TEXTOS



Definição

SCHILD, Herbert. Java para iniciantes.

Embora os **linha de comando** sejam ótimos no ensino dos aspectos básicos de Java e em alguns tipos de programas, como em **códigos no lado do servidor**, na vida real a maioria dos aplicativos é baseada em GUI. Quando este texto foi escrito, a GUI Java mais amplamente usada era baseada em **Swing**.

Definição



Definição

SCHILD, Herbert. Java para iniciantes.

Swing define uma coleção de classes e interfaces que dá suporte a um rico conjunto de componentes visuais, como botões, campos de texto, painéis de rolagem, caixas de seleção, árvores e tabelas, para citar alguns. Coletivamente, esses controles podem ser usados na construção de interfaces gráficas poderosas e ainda assim fáceis de usar. Devido ao seu uso disseminado, Swing é algo que todos os programadores de Java devem conhecer. Logo, este capítulo fornece uma introdução a esse importante framework de GUI.

Definição

Antes de prosseguirmos, é preciso mencionar que um novo framework de GUI chamado **JavaFX** foi criado recentemente para Java. JavaFX fornece uma abordagem **poderosa, otimizada e flexível que simplifica a criação de GUIs** visualmente atraentes. Como tal, ele está sendo visto como a plataforma do futuro. Devido à sua importância, uma introdução é fornecida no Capítulo 17. É claro que **Swing** ainda **continuará sendo usado por muito tempo**, em parte por causa da grande quantidade de **código legado** que existe para ele. Logo, tanto Swing quanto JavaFX devem fazer parte das tarefas dos programadores de Java de agora em diante.

História

Biblioteca para criar interfaces gráficas

AWT

(Abstract Window Toolkit)

Java 1.0

1ª Geração (GUI)

Delega a elaboração das interfaces ao sistema operacional

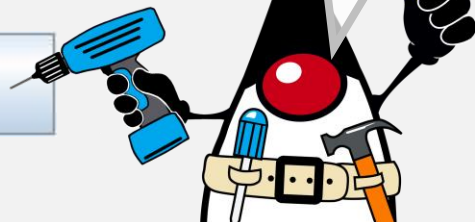
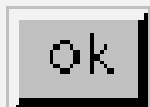
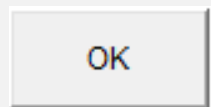
SWING

Java 1.2

2ª Geração (GUI)

Já vou resolver isso.

Eu quero que os botões sejam iguais!
E aquele *slogan*:
“**Write once, run anywhere**”?



História

Biblioteca para criar interfaces gráficas

AWT

Java 1.0

SWING

Java 1.2

Eu quero que os
botões sejam iguais!

SCHILDT, Herbert. **Java para iniciantes.**

O uso de pares nativos gera vários problemas. Em primeiro lugar, devido às diferenças entre os sistemas operacionais, **um componente pode aparecer, ou até mesmo agir, diferentemente em plataformas distintas.** Essa variabilidade potencial ameaçava a filosofia de abrangência total de Java: **escreva uma vez, execute em qualquer local.** Em segundo lugar, a aparência de cada componente era fixa (porque é definida pela plataforma) e não podia ser (facilmente) alterada.

História

Biblioteca para criar interfaces gráficas

AWT

(Abstract Window Toolkit)

Java 1.0

1ª Geração (GUI)

Delega a elaboração das interfaces ao sistema operacional

SWING

Java 1.2

2ª Geração (GUI)

Se encarrega da elaboração das interfaces*

*apenas desktop

Java 7.0

JavaFX

3ª Geração (GUI)

Além de elaborar as interfaces agrega mais recursos

(compatível com vários dispositivos)

História

SCHILD, Herbert. Java para iniciantes.

Não, Swing não substitui AWT. Swing se baseia nos fundamentos fornecidos por AWT. Logo, o AWT ainda é uma parte crucial de Java.

AWT

(Abstract Window Toolkit)

1ª Geração (GUI)

Delega a elaboração das interfaces ao sistema operacional

SWING

Java 1.2

2ª Geração (GUI)

Não tem o objetivo de substituir o AWT

JavaFX

3ª Geração (GUI)

Tem o objetivo de substituir o SWING

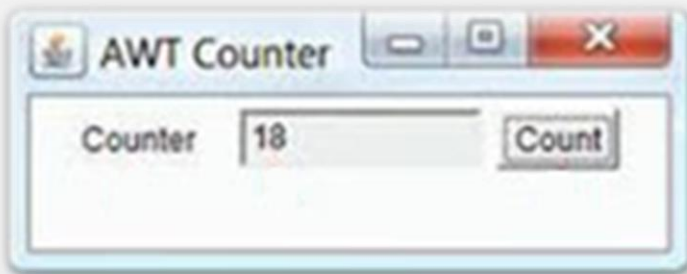
SCHILD, Herbert. Java para iniciantes.

JavaFX foi projetado como substituto de Swing?
A resposta é, basicamente, *sim*.

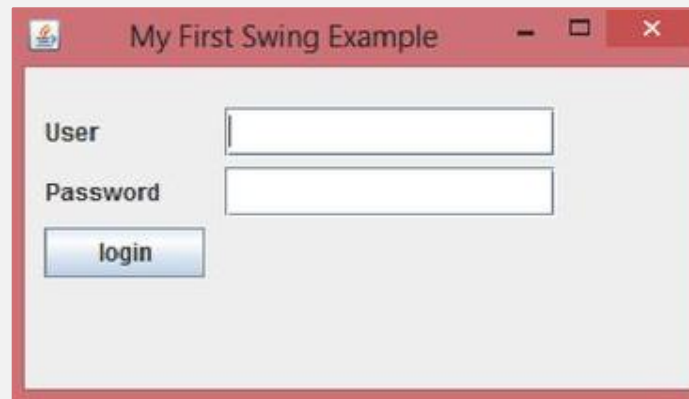
História

AWT

(Abstract Window Toolkit)



SWING



JavaFX



DEITEL, Paul. Java: como programar.



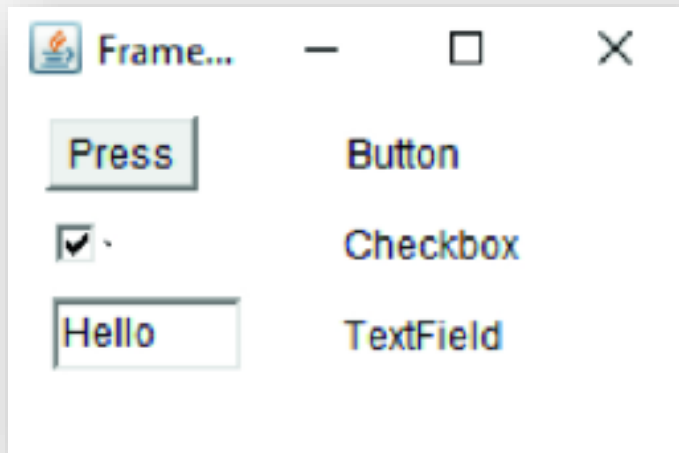
Dica de portabilidade 22.1

Componentes GUI muitas vezes têm uma aparência diferente em plataformas distintas (fontes, tamanhos de fonte, bordas de componente etc.) e podem exigir diferentes quantidades de espaço para que sejam exibidos. Isso poderia alterar os layouts e alinhamentos da GUI.

História

AWT

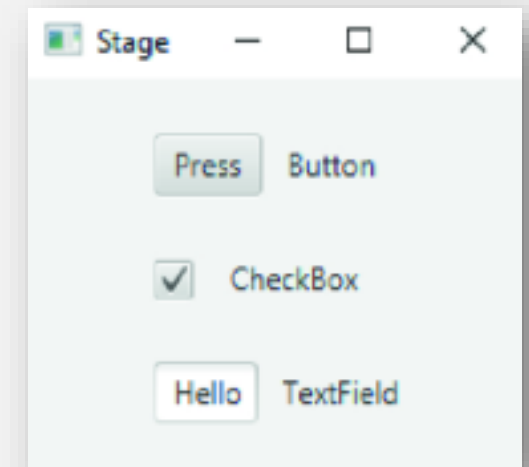
(Abstract Window Toolkit)



SWING



JavaFX



DEITEL, Paul. Java: como programar.



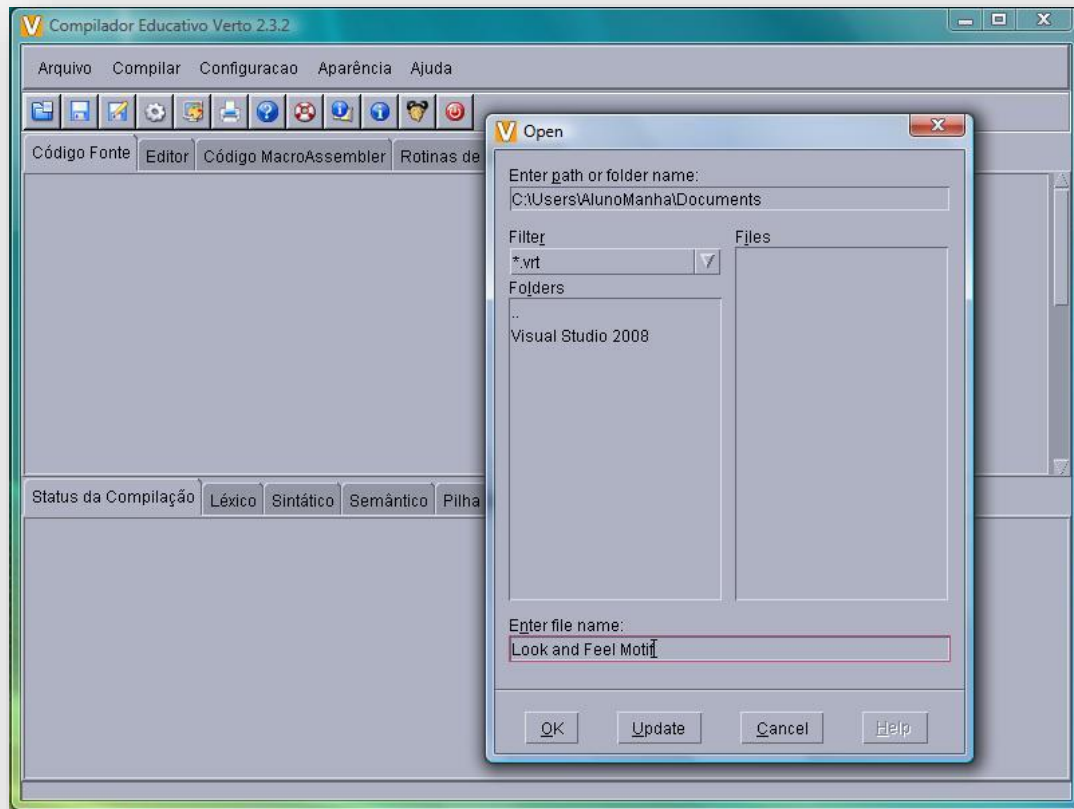
Dica de portabilidade 22.2

Componentes GUI em plataformas diferentes têm funcionalidades-padrão diferentes, por exemplo, nem todas as plataformas permitem que um botão com o foco seja “pressionado” com a barra de espaço.

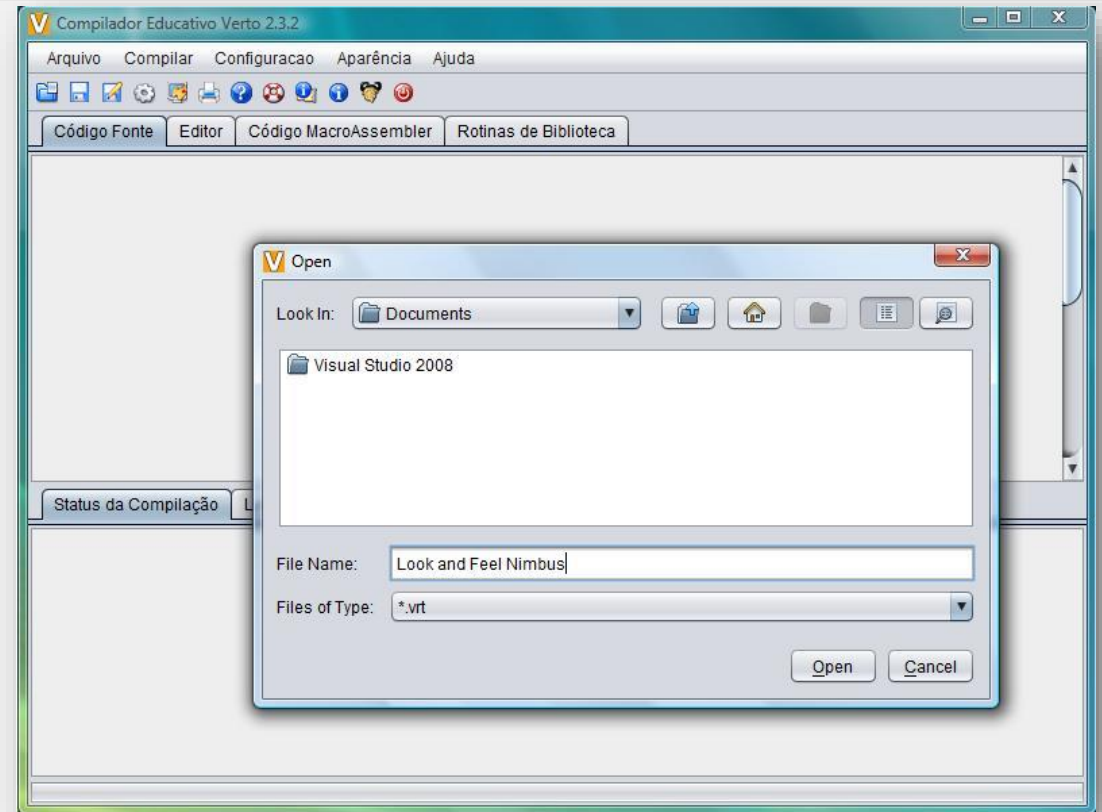
História

SWING

(Fornece a flexibilidade para personalizar a aparência e comportamento em vários estilos, o padrão é o *Nimbus*)



Aparência
Motif

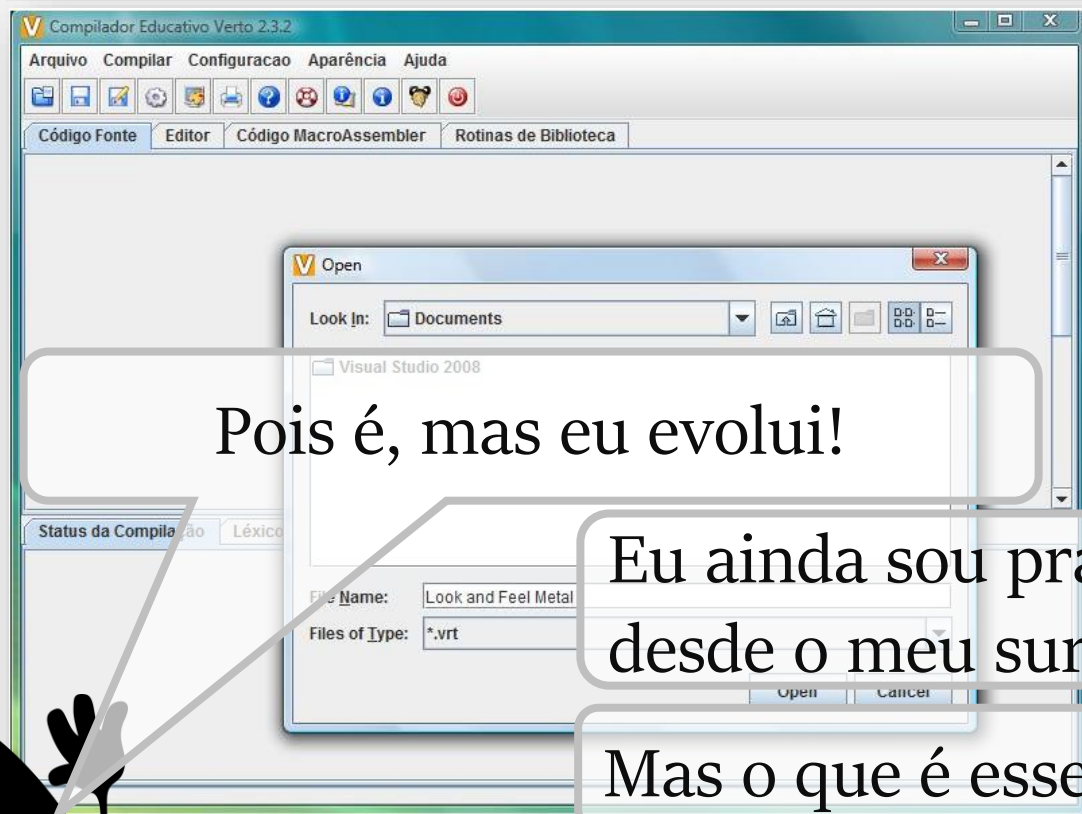


Aparência
Nimbus

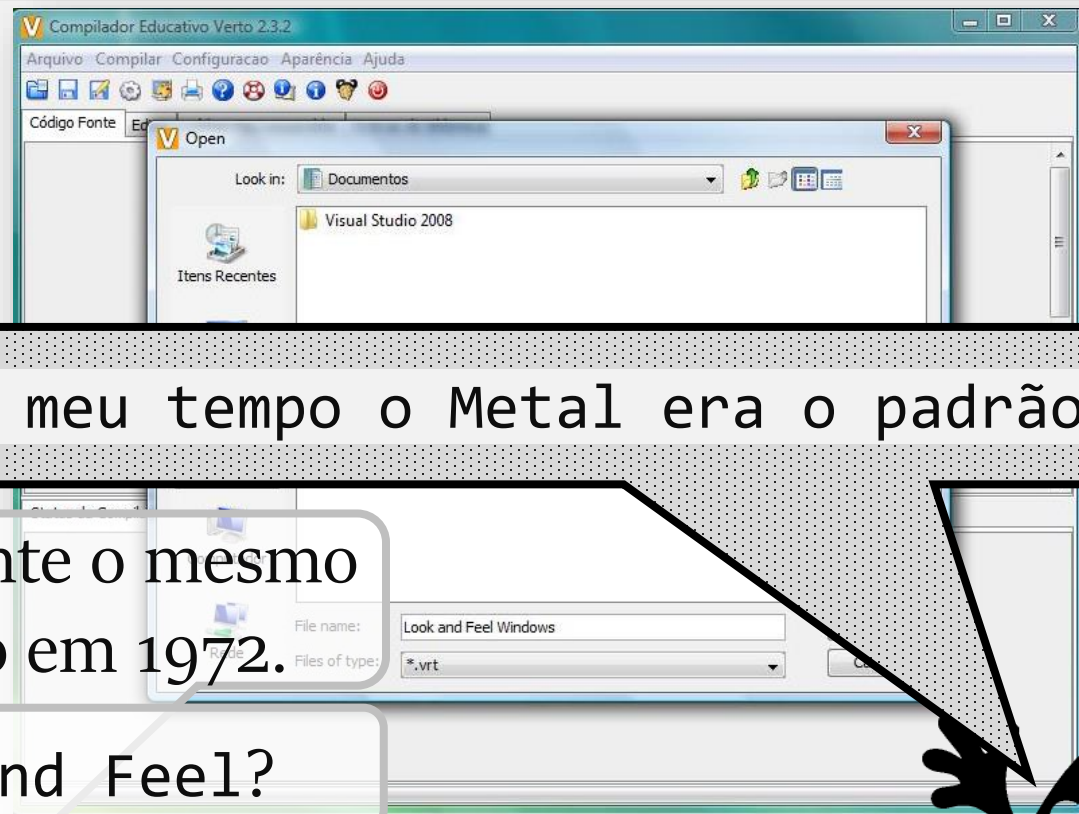
História

SWING

(Fornece a flexibilidade para personalizar a aparência e comportamento em vários estilos, o padrão é o *Nimbus*)



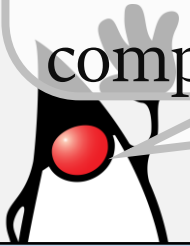
Aparência
Metal



Aparência
Windows

História

Conforme minha documentação oficial: "Look" refere-se à aparência dos componentes e "Feel" refere-se à maneira como os componentes da interface se comportam.



<https://docs.oracle.com/>

Look and Feel

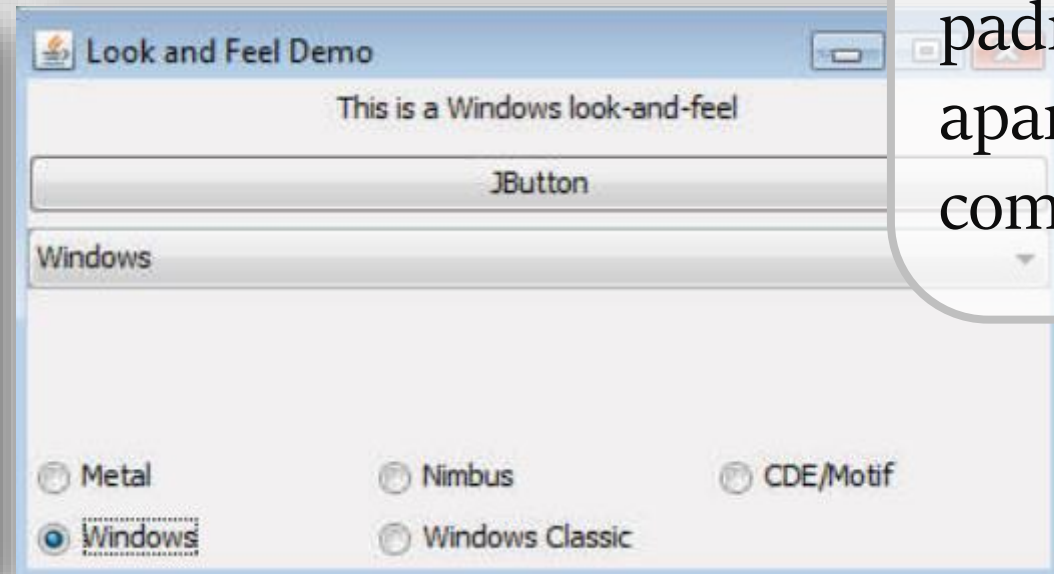
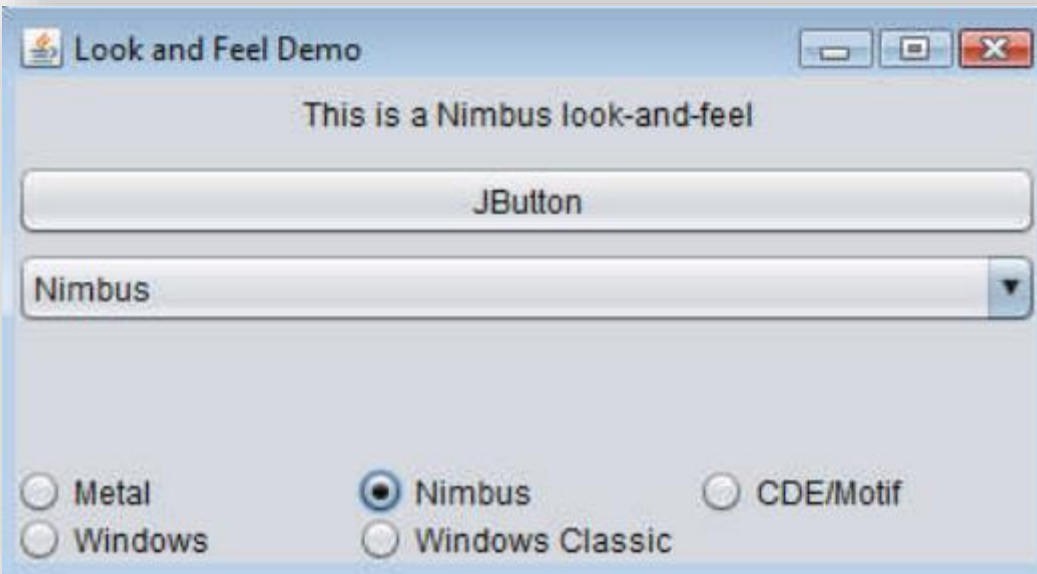
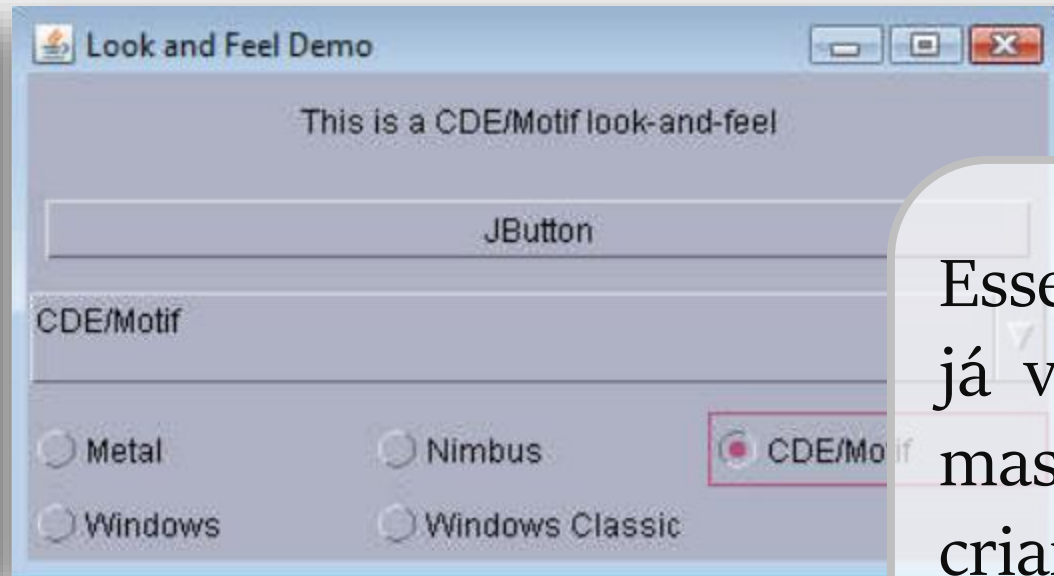
The architecture of Swing is designed so that you may change the "look and feel" (L&F) of your application's GUI (see [A Swing Architecture Overview](#)). "Look" refers to the appearance of GUI widgets (more formally, `JComponents`) and "feel" refers to the way the widgets behave.

Look and Feel, significa olhar e sentir.

Ah, entendi. Parece um pouco o meu OpenGL...



História



Esses são os que já vem prontos, mas você pode criar os seus padrões de aparência e comportamento.



História

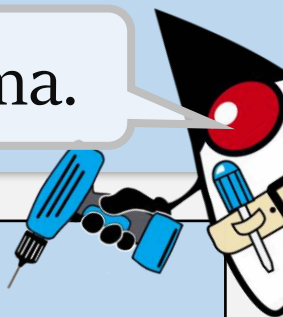
SCHILD, Herbert. Java para iniciantes.

Java fornece aparências, como metal e Nimbus, que estão disponíveis para todos os usuários de Swing. A aparência metal também é chamada de *aparência Java*. Ela é uma aparência independente da plataforma que está disponível em todos os ambientes de execução Java. Também é a aparência padrão. Portanto, a aparência padrão Java (metal) será usada pelos exemplos deste capítulo.

Swing pode fornecer uma aparência adaptável porque usa uma versão modificada da arquitetura clássica *modelo-exibição-controlador* (*MVC, model-view-controller*). Na terminologia MVC, *modelo* são as informações de estado associadas ao componente. Por exemplo, no caso de uma caixa de seleção, o *modelo* contém um campo que indica se a caixa está marcada ou desmarcada. *Exibição determina como o componente será exibido na tela*, incluindo qualquer aspecto que seja afetado pelo estado atual do modelo. *Controlador determina como o componente reagirá ao usuário*. Por exemplo, quando o usuário clicar em uma caixa de seleção, o controlador reagirá alterando o modelo para refletir a escolha (marcada ou desmarcada).

Swing

Resolvi o problema da interface, o resto não tinha problema.



SCHILD, Herbert. Java para iniciantes.

Todos os componentes de Swing são representados por classes definidas dentro do pacote **javax.swing**. O pacote **javax.swing** contém os componentes e modelos definidos por Swing. Por exemplo, ele define **classes que implementam rótulos, botões, controles de edição e menus.**

SWING

Agrega o conjunto de componentes responsáveis pelas interfaces (*e.g.* botão)

```
import javax.swing
```

AWT

Agrega o conjunto de componentes responsáveis pelos eventos (*e.g.* clicar)

```
import java.awt
```

Swing

SCHILDT, Herbert. Java para iniciantes.

JApplet	JButton	JCheckBox	JCheckBoxMenuItem
JColorChooser	JComboBox	JComponent	JDesktopPane
JDialog	JEditorPane	JFileChooser	JFormattedTextField
JFrame	JInternalFrame	JLabel	JLayer
JLayeredPane	JList	JMenu	JMenuBar
JMenuItem	JOptionPane	JPanel	JPasswordField
JPopupMenu	JProgressBar	JRadioButton	JRadioButtonMenuItem
JRootPane	JScrollBar	JScrollPane	JSeparator
JSlider	JSpinner	JSplitPane	JTabbedPane
JTable	JTextArea	TextField	JTextPane
JToggleButton	JToolBar	JToolTip	JTree
JViewport	JWindow		

Descrição:

Janela da aplicação, podendo agrupar outros componentes dentro de si

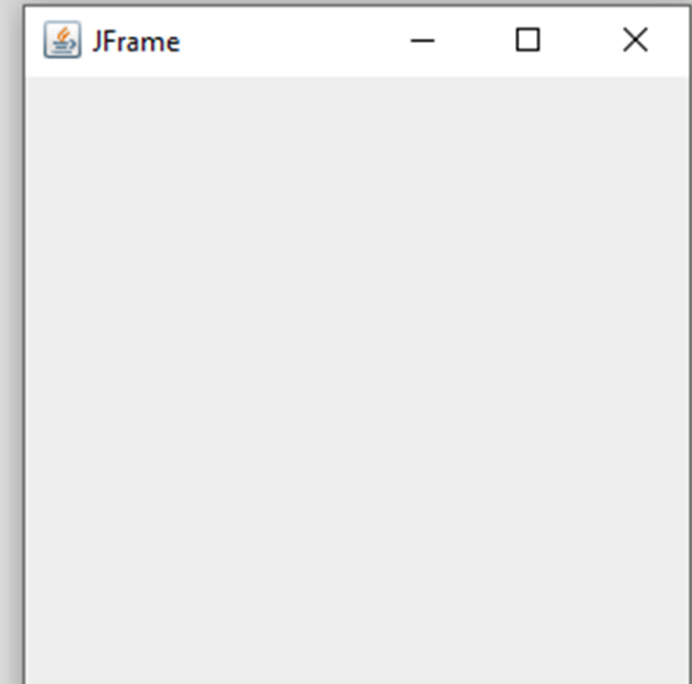


Código (Janela.java)

```
1 import javax.swing.JFrame;
2
3 public class Janela extends JFrame{
4     public Janela(){
5         setTitle("JFrame");
6         setBounds(50, 50, 300, 300);
7     }
8
9     public static void main(String[] args){
10         Janela objeto = new Janela();
11         objeto.setVisible(true);
12     }
13 }
```

Herança de JFrame

Resultado

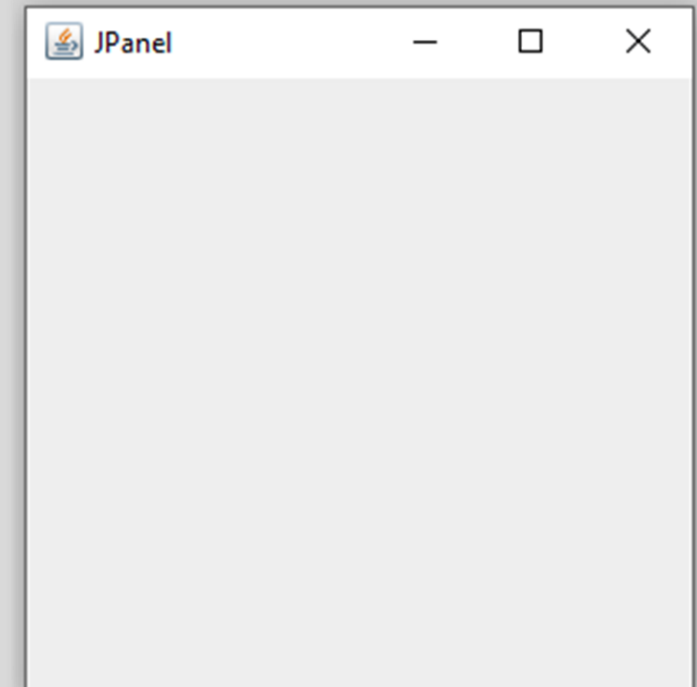


Descrição: Contêiner responsável por agrupar componentes dentro de uma área

Código (Janela.java)

```
1 import javax.swing.JFrame;  
2 import javax.swing.JPanel;  
3  
4 class Painel extends JPanel{  
5     public Painel(){  
6     }  
7 }  
8  
9 public class Janela extends JFrame{  
10     public Janela(){  
11         setTitle("JPanel");  
12         setBounds(50, 50, 300, 300);  
13     }  
14 }
```

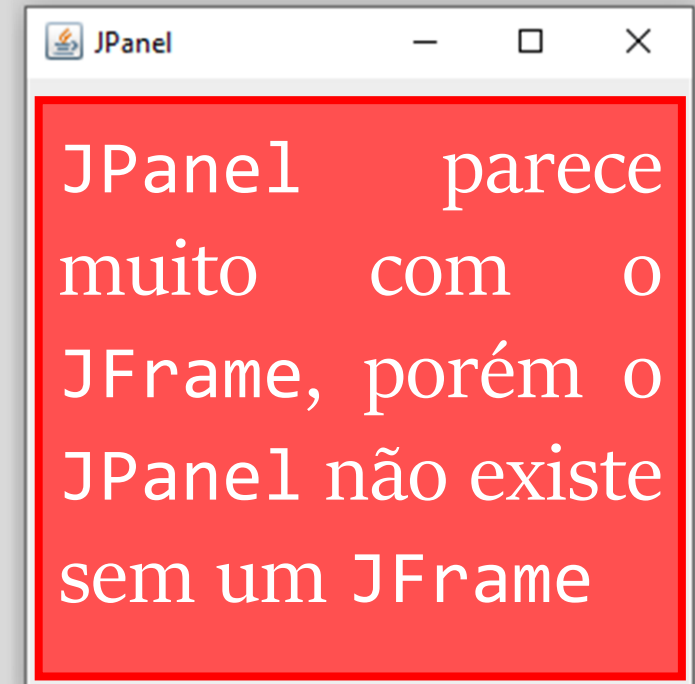
Resultado




```
2 import javax.swing.JPanel;
3
4 class Painel extends JPanel{
5     public Painel(){
6     }
7 }
8
9 public class Janela extends JFrame{
10     public Janela(){
11         setTitle("JPanel");
12         setBounds(50, 50, 300, 300);
13     }
14
15     public static void main(String[] args){
16         Janela objeto = new Janela();
17         objeto.add(new Painel());
18         objeto.setVisible(true);
19     }
20 }
```

grupar componentes dentro

Resultado

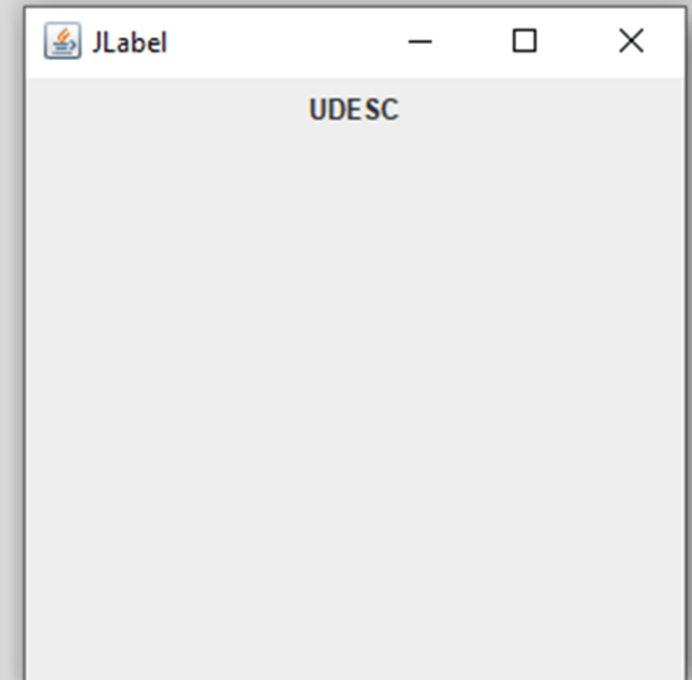


Descrição: Componente responsável por exibir texto ou imagens

Código (Janela.java)

```
1 import javax.swing.JFrame;  
2 import javax.swing.JPanel;  
3 import javax.swing.JLabel;  
4 class Painel extends JPanel{  
5     public Painel(){  
6         JLabel texto = new JLabel("UDESC");  
7         add(texto);  
8     }  
9 }  
10  
11 public class Janela extends JFrame{  
12     public Janela(){  
13         setTitle("JLabel");  
14         setBounds(50, 50, 300, 300);
```

Resultado

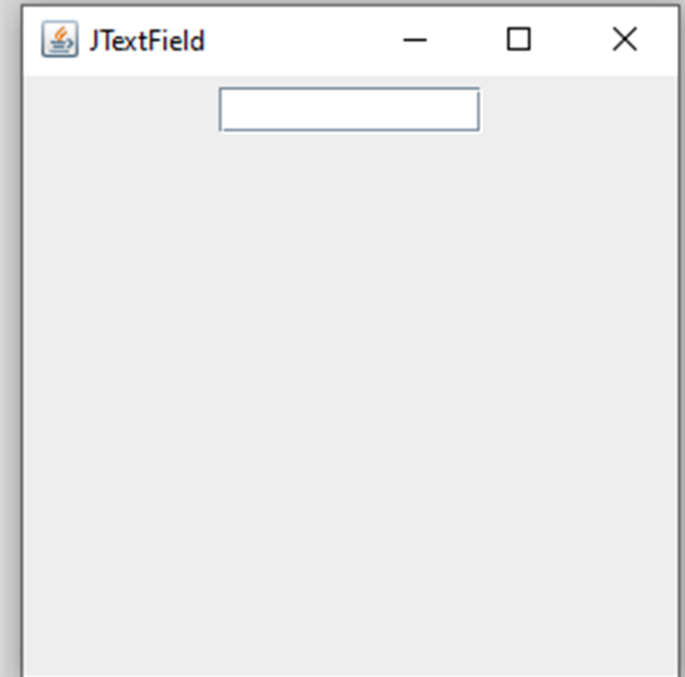


Descrição: Campo de entrada de texto

Código (Janela.java)

```
1 import javax.swing.JFrame;
2 import javax.swing.JPanel;
3 import javax.swing.JTextField;
4 class Painel extends JPanel{
5     public Painel(){
6         JTextField campo = new JTextField();
7         campo.setColumns(10);
8         add(campo);
9     }
10 }
11
12 public class Janela extends JFrame{
13     public Janela(){
14         setTitle("JTextField");
```

Resultado



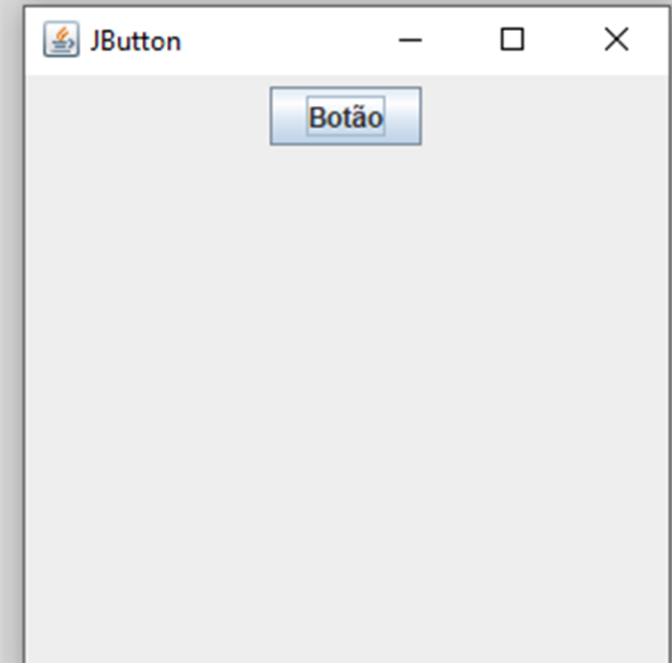
Descrição:

Botão responsável por disparar uma ação quando pressionado (ação deve ser programada)

Código (Janela.java)

```
1 import javax.swing.JFrame;  
2 import javax.swing.JPanel;  
3 import javax.swing.JButton;  
4 class Painel extends JPanel{  
5     public Painel(){  
6         JButton controle = new JButton("Botão");  
7         add(controle);  
8     }  
9 }  
10  
11 public class Janela extends JFrame{  
12     public Janela(){  
13         setTitle("JTextField");  
14         setBounds(50, 50, 300, 300);
```

Resultado

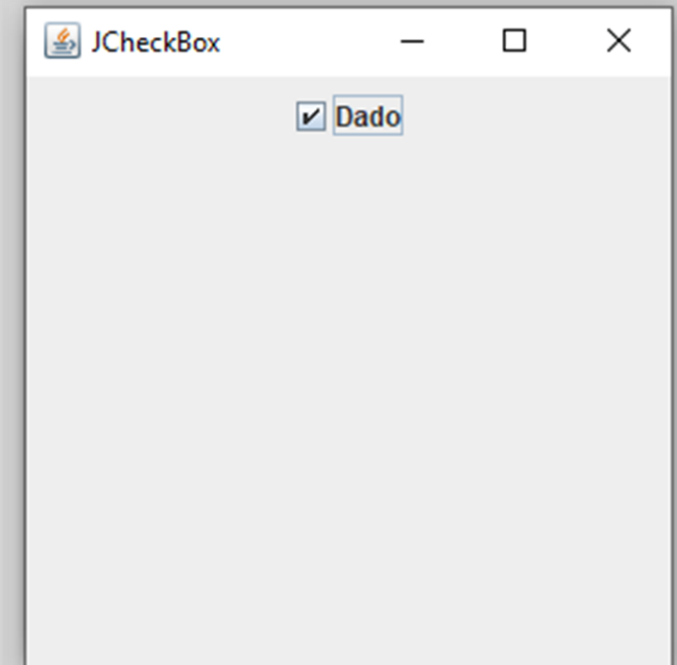


Descrição: Caixa de seleção responsável por marcar ou desmarcar opções (true/false)

Código (Janela.java)

```
1 import javax.swing.JFrame;
2 import javax.swing.JPanel;
3 import javax.swing.JCheckBox;
4 class Painel extends JPanel{
5     public Painel(){
6         JCheckBox caixa = new JCheckBox("Dado");
7         add(caixa);
8     }
9 }
10
11 public class Janela extends JFrame{
12     public Janela(){
13         setTitle("JCheckBox");
14         setBounds(50, 50, 300, 300);
```

Resultado

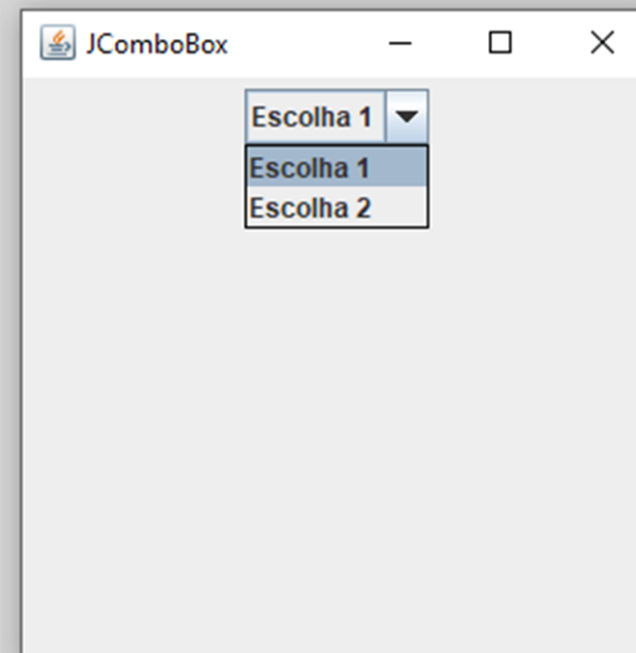


Descrição: Lista suspensa para a seleção de objetos (seleção única)

Código (Janela.java)

```
1 import javax.swing.JFrame;
2 import javax.swing.JPanel;
3 import javax.swing.JComboBox;
4 class Painei extends JPanel{
5     public Painei(){
6         String[] itens = new String[]{
7             "Escolha 1", "Escolha 2" };
8         JComboBox<String> obj = new JComboBox<String>(itens);
9         add(obj);
10    }
11 }
12
13 public class Janela extends JFrame{
14     public Janela(){
```

Resultado

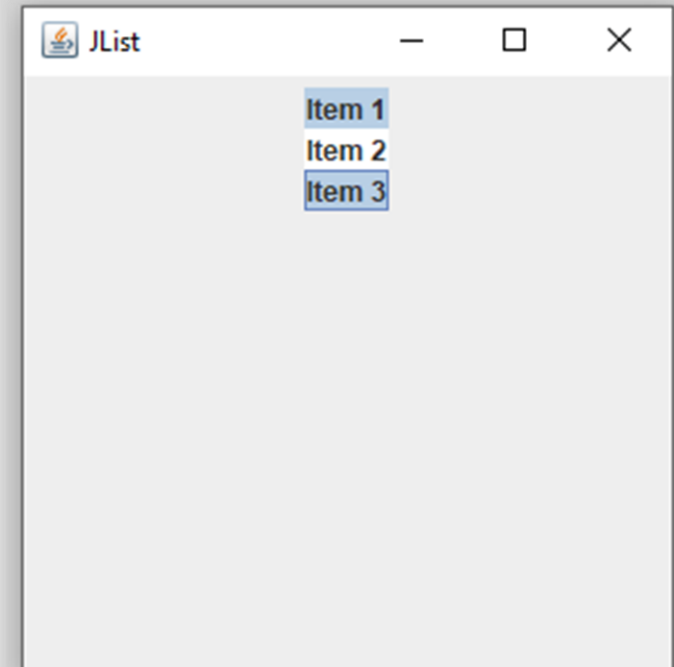


Descrição: Lista suspensa para a seleção de vários objetos (seleção múltipla)

Código (Janela.java)

```
1 import javax.swing.JFrame;
2 import javax.swing.JPanel;
3 import javax.swing.JList;
4 class Painei extends JPanel{
5     public Painei(){
6         String[] i = new String[]{
7             "Item 1", "Item 2", "Item 3" };
8         JList<String> L = new JList<String>(i);
9         add(L);
10    }
11 }
12
13 public class Janela extends JFrame{
14     public Janela(){
```

Resultado

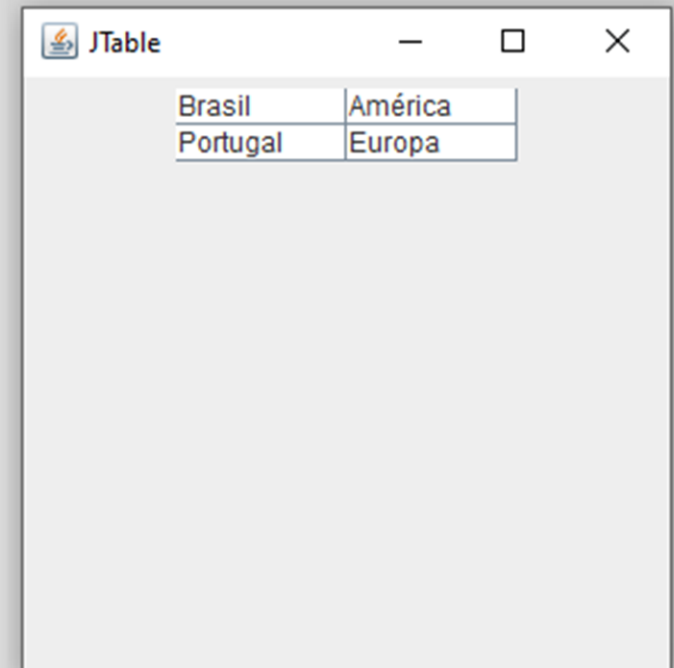


Descrição: Tabela de dados editáveis

Código (Janela.java)

```
1 import javax.swing.JFrame;
2 import javax.swing.JPanel;
3 import javax.swing.JTable;
4 class Painel extends JPanel{
5     public Painel(){
6         Integer[] coluna = {1,2};
7         Object[][] obj = {
8             {"Brasil", "América"},
9             {"Portugal", "Europa"},};
10        JTable tabela = new JTable(obj, coluna);
11        add(tabela);
12    }
13 }
14
```

Resultado



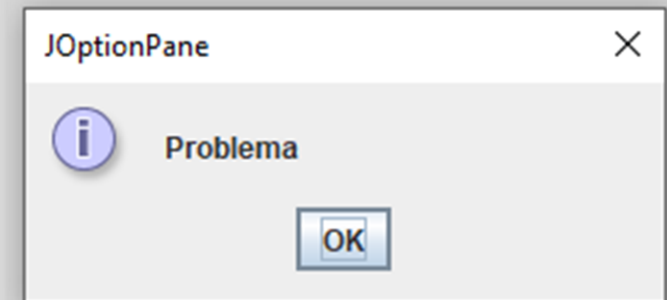
Brasil	América
Portugal	Europa

Descrição: Janela de alerta

Código (Janela.java)

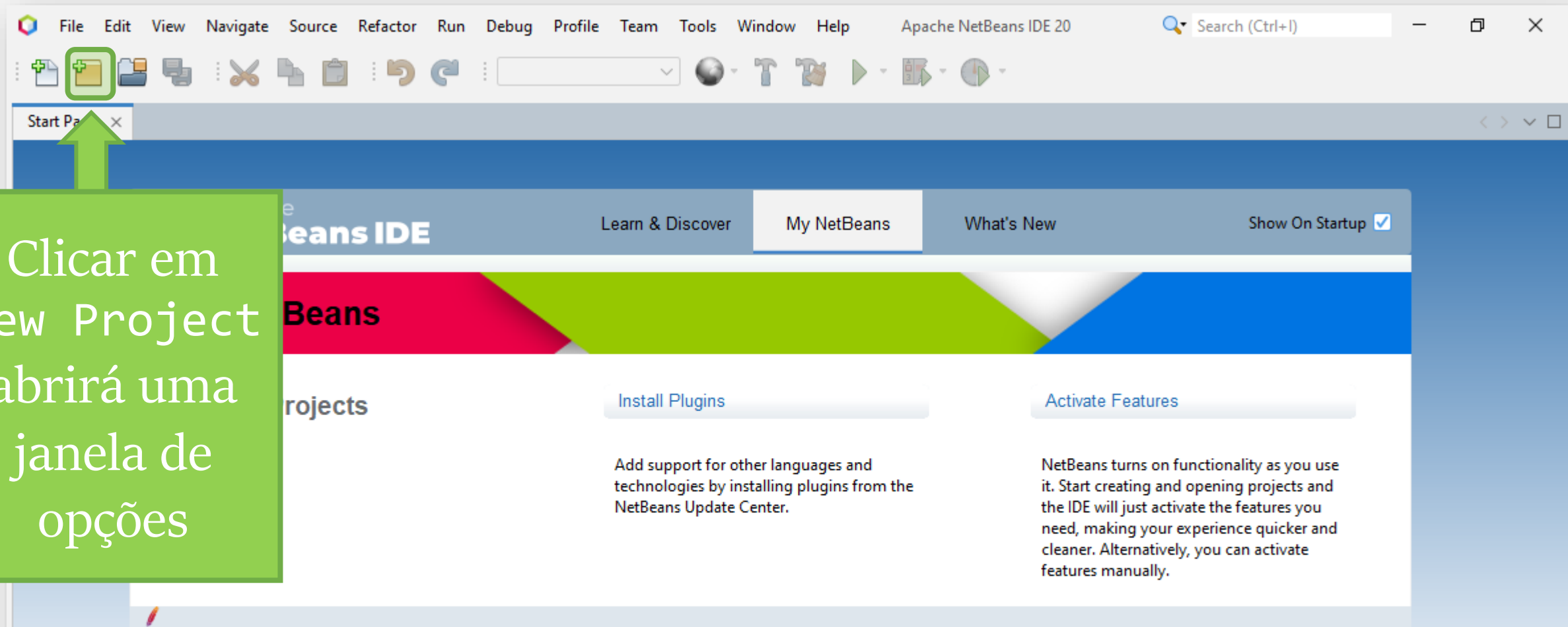
```
1 import javax.swing.JFrame;  
2 import javax.swing.JPanel;  
3 import javax.swing.JOptionPane;  
4 class Painei extends JPanel{  
5     public Painei(){  
6         JOptionPane.showMessageDialog(  
7             null,  
8             "Problema",  
9             "JOptionPane",  
10            JOptionPane.INFORMATION_MESSAGE);  
11     }  
12 }  
13  
14 public class Janela extends JFrame{
```

Resultado



NetBeans

Vamos começar, clique em **New Project** para abrir uma janela de opções.



Clicar em
New Project
abrirá uma
janela de
opções

NetBeans

Selecione a opção **Java Application**
e depois clique no botão **Next**.



Start Page x

Apache NetBeans

My NetBeans

Recent Projects

New Project

Steps

1. Choose Project
2. ...

Choose Project

Filter:

Categories:

- Java with Maven
- Java with Gradle
- Java with Ant
- JavaFX
- Java Web
- Java Enterprise
- NetBeans Modules
- HTML5/JavaScript
- C/C++
- PHP
- Samples

Projects:

- Java Application
- Java Class Library
- Java Project with Existing Sources
- Java Regular Project
- Java Web Form Project

Description:

Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use an IDE-generated Ant build script to build, run, and debug your project.

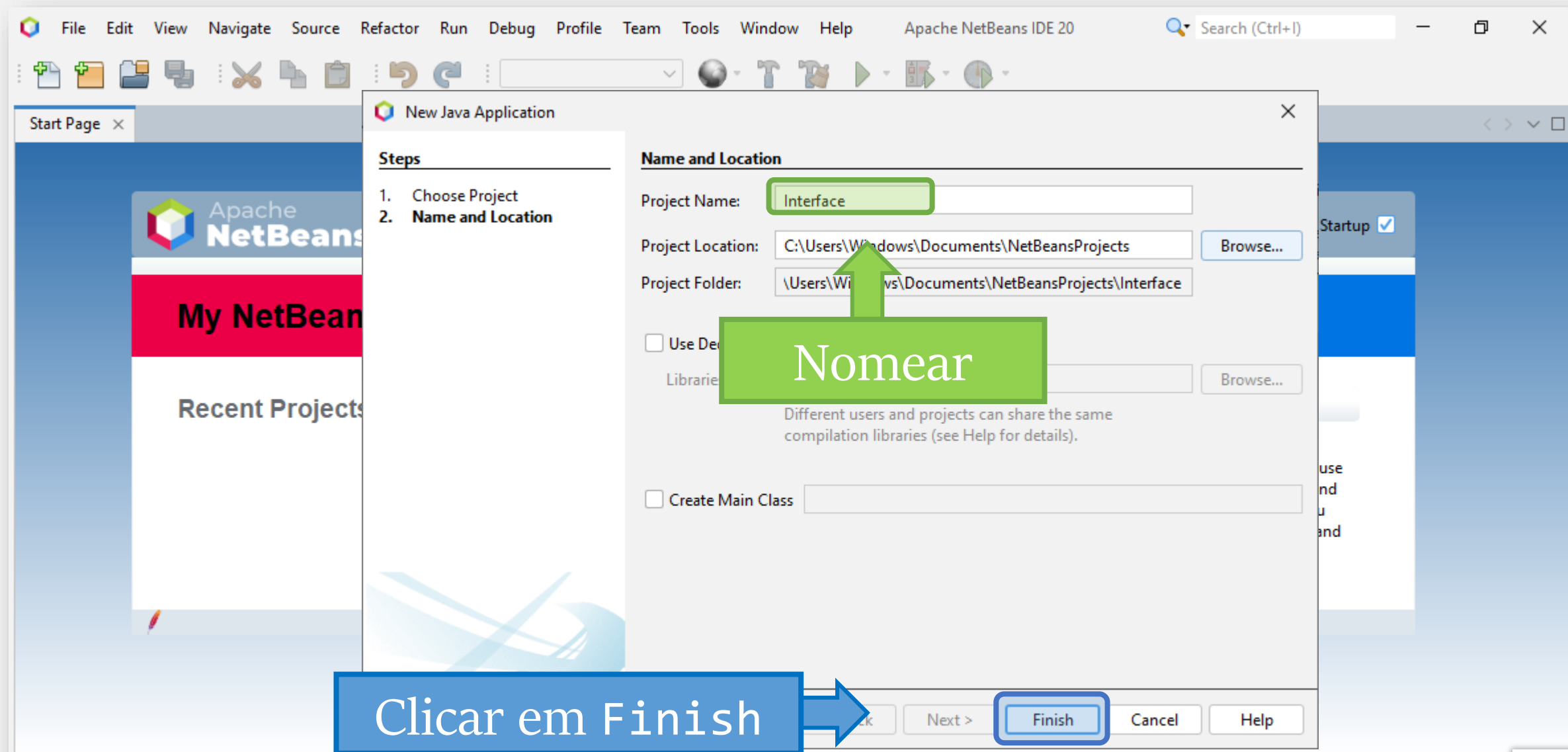
Next > Finish Cancel Help

Selecionar Java Application

Clicar em Next

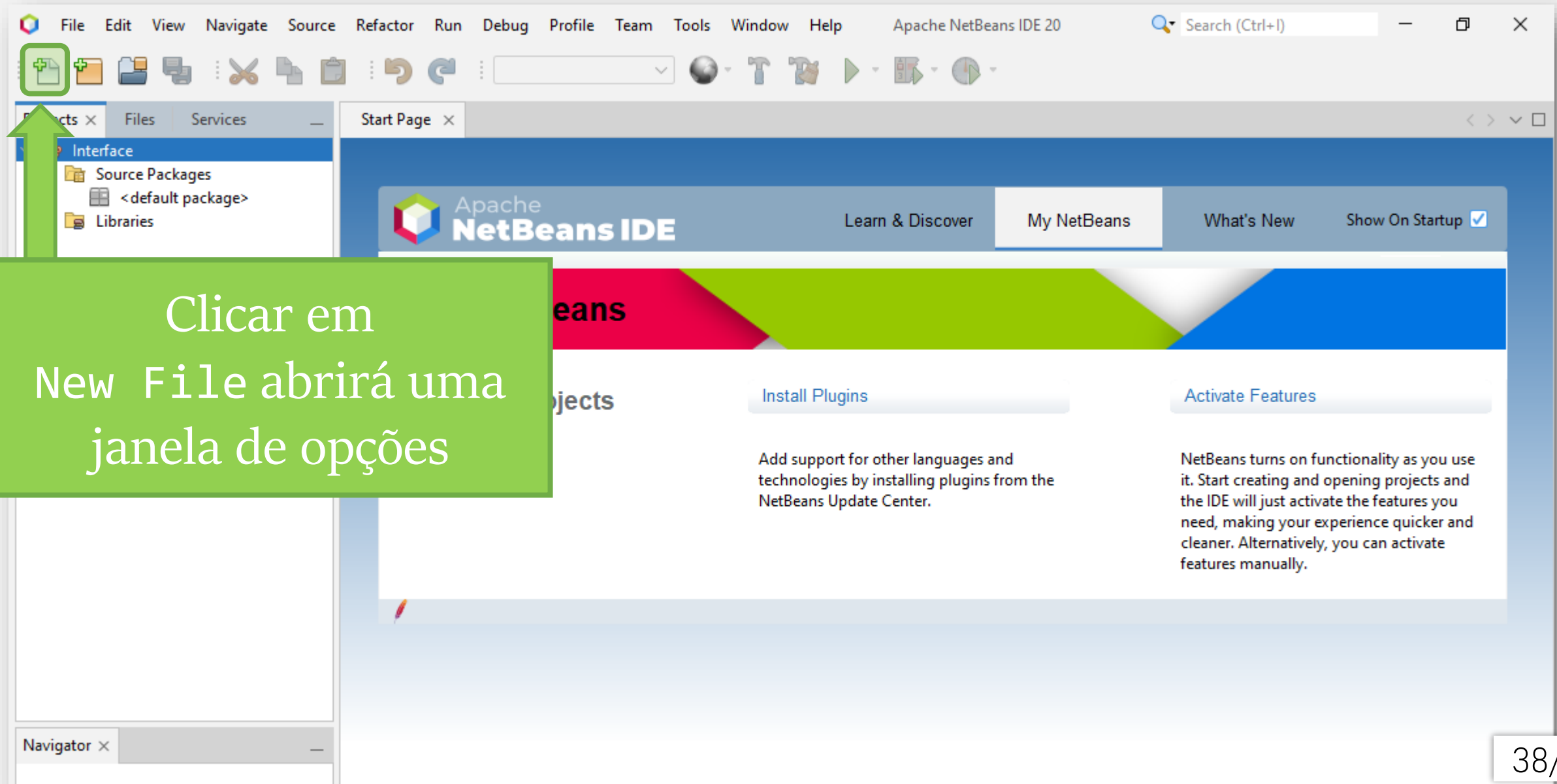
NetBeans

Dê um nome ao Projeto e depois clique no botão **Finish**.



NetBeans

Clique em **New File**
para abrir outra janela de opções.



Clicar em
New File abrirá uma
janela de opções

NetBeans

Praticamente toda a aplicação de interface em Java necessita de um **JFrame**.



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Apache NetBeans IDE 20 Search (Ctrl+I)

Projects Files Services

Interface

- Source Packages
 - <default package>
- Libraries

New File

Steps

1. Choose File Type
2. ...

Choose File Type

Project: Interface

Filter:

Categories:

- Java
- Swing GUI Forms
- JavaBeans Objects
- AWT GUI Forms
- Unit Tests
- Web
- Struts
- Spring Framework
- Micronaut
- Persistence

File Types:

- JDialog Form
- JFrame Form**
- JInternalFrame Form
- JPanel Form
- JApplet Form
- Bean Form
- Application Sample Form
- MDI Application Sample Form
- OK / Cancel Dialog Sample Form

Description:

Creates a new JFC (Swing) Frame. Frames are typically used as standalone top-level windows as the main user interface to the application. Most Swing applications are built starting from this form.

Show On Startup ☒

Selecionar JFrame Form

Olha eu ali.

Clicar em Next

Next > Finish Cancel Help



Apache NetBeans IDE 20

Search (Ctrl+I)

Projects × Files Services

Interface

Source Packages

<default package>

Libraries

New JFrame Form

Steps

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

Name and Location

Class Name: Programa

Project: Interface

Location: Source Packages

Package: udesc.poo

Created File: indows\Documents\NetBeansProjects\Interface\src\udesc\poo\Programa.java

Superclass: Browse...

Interfaces: Browse...

Show On Startup ☒

es

n functionality as you use
nd opening projects and
ctivate the features you
ur experience quicker and
vely, you can activate
/.

Clicar em Finish

Next > Finish Cancel Help

Navigador ×

NetBeans

Você pode programar a interface clicando em Source ou desenhar clicando em Desing.



Vamos desenhar!
Clique, segure e arraste Panel para a área de Desing depois solte...

Projects × Files Services — Start Page × Programa.java ×
Source Design History
Use the context menu to access available useful actions for the selected

Palette ×
Swing Containers
Panel
Tabbed Pane
Split Pane
Scroll Pane
Desktop Pane
Internal Frame
Layered Pane
Swing
Label
Text Field
Progress Bar
Separator
Table
Button
Button Group
Text Area
Formatted Field
Text Pane
Toggle Button
Combo Box
Scroll Bar
Password Field
Editor Pane
Tree
Swing Menus
Swing Windows
Swing Fillers
AWT
Beans

[JFrame] - Properties ×
Properties Events Code
Properties
defaultCloseOperation EXIT_ON_CLOSE
title
Other Properties
alwaysOnTop
alwaysOnTopSupported
autoRequestFocus
background [242,242,242]

NetBeans



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help Apache NetBeans IDE 20 Search (Ctrl+I)

Projects Files Services Start Page Programa.java Source Design History

Use the context menu to access available useful actions for the selected

Interface Source Packages udesc.poo Programa.java Libraries

Swing Containers Panel Tabbed Pane Split Pane Scroll Pane Tool Bar Desktop Pane Internal Frame Layered Pane

Swing Controls Label OK Button ON Toggle Button Check Box Radio Button Button Group Combo Box List Text Field Text Area Scroll Bar Slider Progress Bar Formatted Field Password Field Spinner Separator Text Pane Editor Pane Tree Table

Swing Menus Swing Windows Swing Fillers AWT Beans

[JFrame] - Properties Properties Events Code

Properties

defaultCloseOperation EXIT_ON_CLOSE

title

Other Properties

alwaysOnTop alwaysOnTopSupported autoRequestFocus background

Prontinho, agora que temos um Painel podemos adicionar outro componentes

jPanel1 [JPanel] - Navigator Programa Other Components [JPanel] [JPanel]

42/78

NetBeans



Apache NetBeans IDE 20

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

Interface

Source Packages

udesc.poo

Programa.java

Libraries

Start Page Programa.java

Source Design History

Use the context menu to access available useful actions for the selected

Palette

Swing Containers

Panel Tabbed Pane Split Pane Scroll Pane

Tool Bar Desktop Pane Internal Frame Layered Pane

Swing Controls

Label Label OK Button ON Toggle Button Check Box

Radio Button Button Group Combo Box List

Text Field Text Area Scroll Bar Slider

Progress Bar Formatted Field Password Field Spinner

Separator Text Pane Editor Pane Tree

Table

Swing Menus

Swing Windows

Swing Fillers

AWT

Beans

[JFrame] - Properties

Properties Events Code

Properties

defaultCloseOperation EXIT_ON_CLOSE

title

Other Properties

alwaysOnTop

alwaysOnTopSupported

autoRequestFocus

background [242,242,242]

É possível alterar algumas **propriedades** do painel da melhor forma que convier.

jPanel1 [JPanel] - Navigator

Programa

Other Components

Panel1 [JPanel]

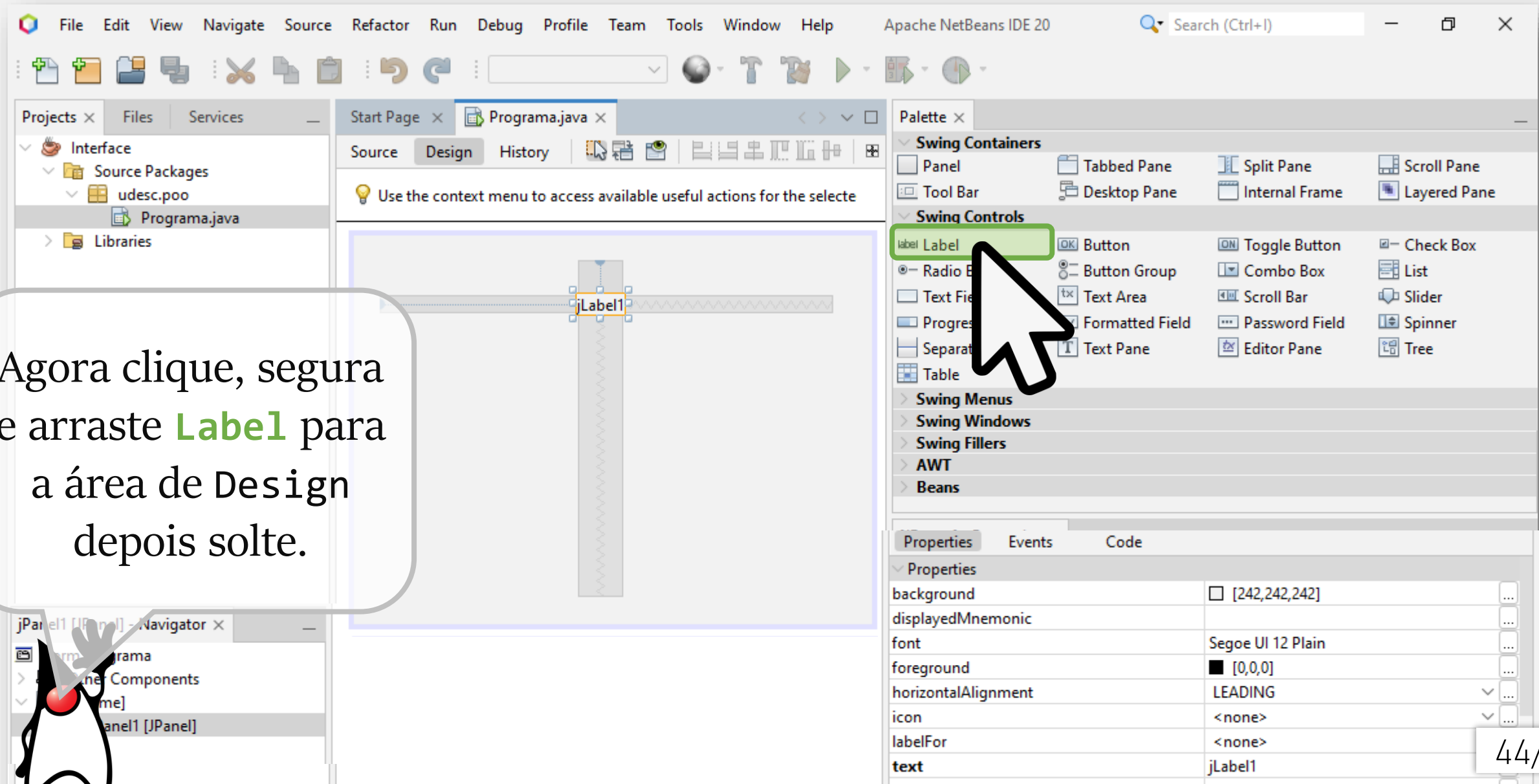
43/78

NetBeans

Eu vou definir por padrão os nomes, mas depois você pode alterar...



Agora clique, segura e arraste **Label** para a área de Design depois solte.



NetBeans



Apache NetBeans IDE 20

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

Interface

Source Packages

udesc.poo

Programa.java

Libraries

Start Page Programa.java

Source Design History

Use the context menu to access available useful actions for the selected

Palette

Swing Containers

Panel Tabbed Pane Split Pane Scroll Pane

Tool Bar Desktop Pane Internal Frame Layered Pane

Swing Controls

Label Label OK Button ON Toggle Button Check Box

Radio Button Button Group Combo Box List

Text Field Text Area Scroll Bar Slider

Progress Bar Formatted Field Password Field Spinner

Separator Text Pane Editor Pane Tree

Table

Swing Menus

Swing Windows

Swing Fillers

AWT

jLabel1 [JLabel] - Properties

Properties Events Code

Properties

background	[242,242,242]
displayedMnemonic	
font	Segoe UI 12 Plain
foreground	[0,0,0]
horizontalAlignment	LEADING
icon	<none>
labelFor	<none>
text	jLabel1

É possível ir em **text** e alterar o conteúdo de `jLabel1`.

jPanel1 [JPanel] - Navigator

Programa

Other Components

[JPanel]

Panel1 [JPanel]

45/78

NetBeans

Agora basta clicar em **Run Project (F6)**
para ver o resultado.

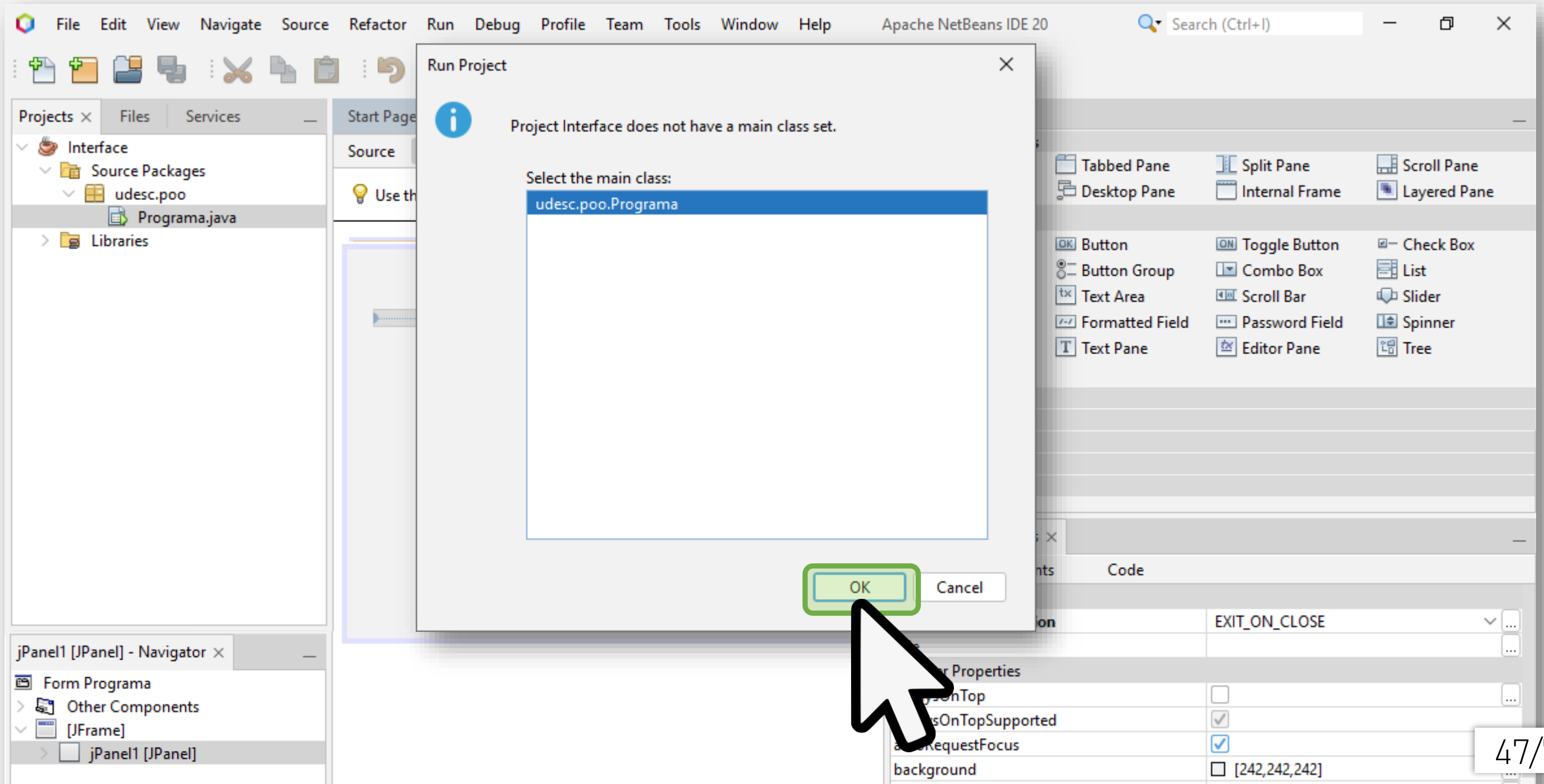


Prontinho, você criou o seu primeiro programa “Olá, Mundo!” em interface gráfica.

The screenshot shows the NetBeans IDE 20 interface. The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar contains various icons, including a green 'Run Project' button (F6) which is highlighted with a green box and a mouse cursor. The project explorer on the left shows a project named 'Interface' with a sub-project 'udesc.poo' containing a file 'Programa.java'. The source editor in the center shows the 'Programa.java' file with the text 'Olá, Mundo!' in a text field. The properties window on the right shows the 'jLabel1 [JLabel] - Properties' tab, displaying various properties like background, displayedMnemonic, font, foreground, horizontalAlignment, icon, and labelFor.

NetBeans

Como não definimos uma classe principal, precisamos fazer isso, mas é só na primeira vez.



NetBeans

Agora feche a janela e volte a programar!



Eu chamo sim isso
de programar, só
que quem está
gerando os códigos é
o **NetBeans**.

Olá, Mundo!

Você chama isso de
“programar”?

Panel1 [JPanel]
Panel1 [JPanel]
Panel1 [JPanel]

displayedMnemonic
font Segoe UI 12 Plain
foreground [0,0,0]
horizontalAlignment LEADING
icon <none>
labelFor <none>
text Olá, Mundo!



NetBeans



Vou mostrar a programação. Primeiro coloque o componente **Button** na área de Design

Apache NetBeans IDE 20

Search (Ctrl+I)

Projects | Files | Services

Start Page | Programa.java

Source | Design | History

Use the context menu to access available useful actions for the selected

Swing Containers

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

Swing Controls

- Label
- Radio Button
- Text Field
- Progress Bar
- Separator
- Table
- OK Button
- Button Group
- Text Area
- Form
- Text Password Field
- Toggle Button
- Combo Box
- Scroll Bar
- Password Field
- Editor Pane
- Check Box
- List
- Slider
- Spinner
- Tree

jLabel1 [JLabel] - Properties

Properties | Events | Code

Properties

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

49/78

NetBeans

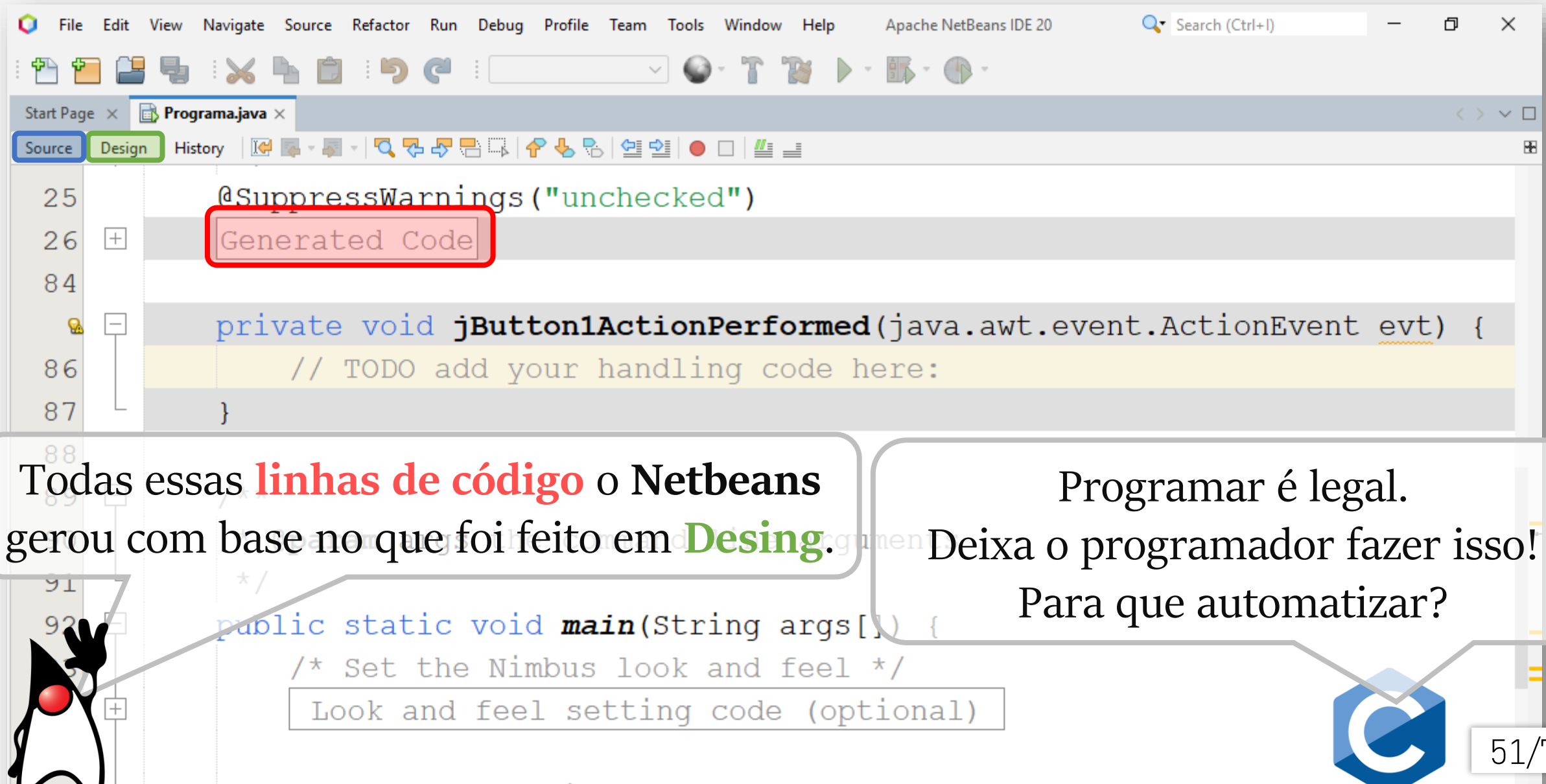


Agora clique em Button com o botão direito e selecione actionPerformed

NetBeans IDE 20 interface showing the design view of a Java Swing window. The window contains a text area with the text "Olá, Mundo!" and a button labeled "jButton1". The "Palette" on the right lists various Swing components, including "Swing Containers" (Panel, Tabbed Pane, Split Pane, Scroll Pane, Tool Bar, Desktop Pane, Internal Frame, Layered Pane) and "Swing Controls" (Label, Button, Toggle Button, Check Box, Radio Button, Button Group, Combo Box, List, Text Field, Text Area, Scroll Bar, Slider, Progress Bar, Formatted Field, Password Field, Spinner, Separator, Text Pane, Editor Pane, Tree, Table). The "Properties" window for "jLabel1 [JLabel]" is also visible, showing the "Properties" tab with various attributes like "text", "font", "background color", and "border".

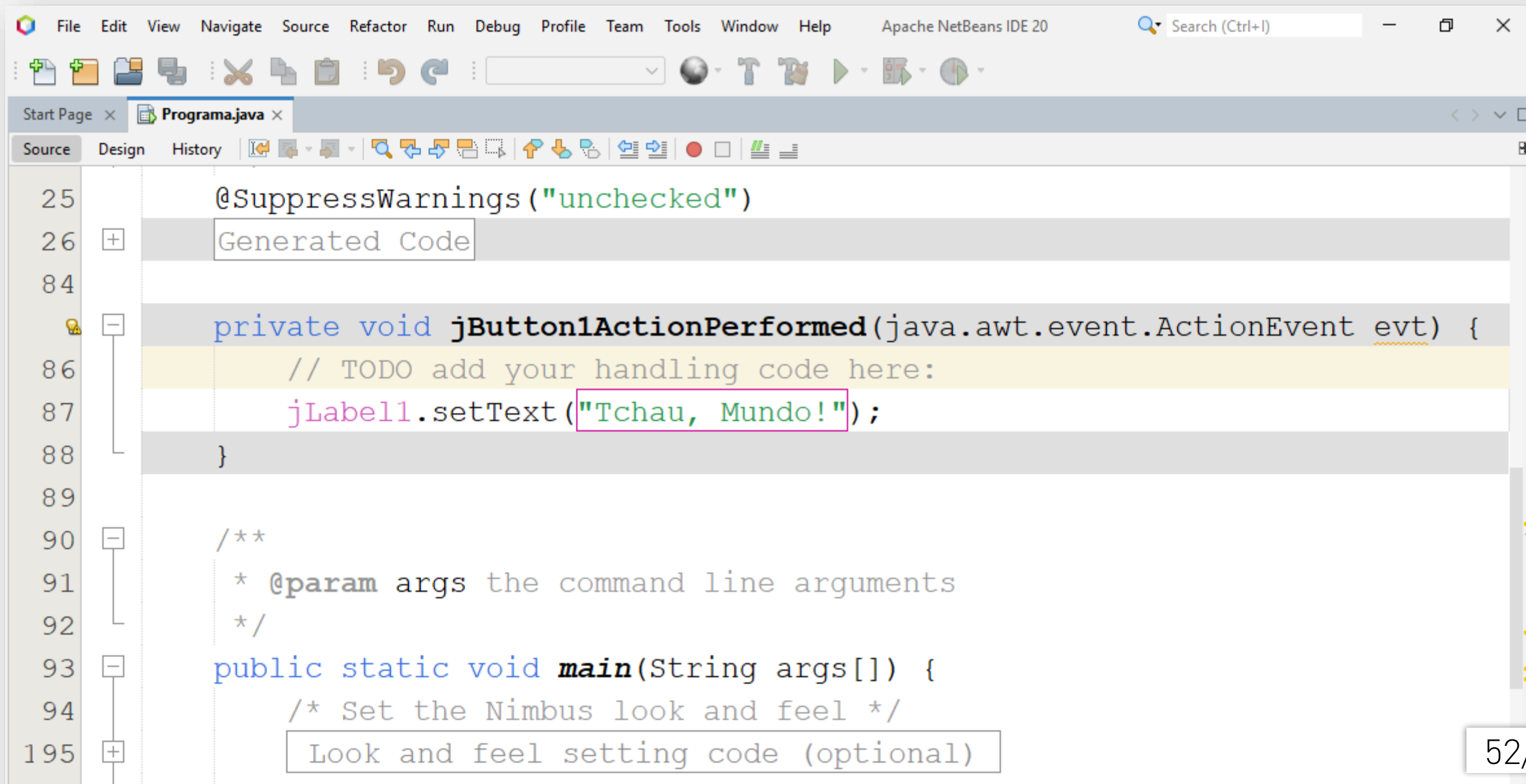
NetBeans

Lhe redirecionei para o seu código.
Eu sai de **Desing** e fui para **Source**.



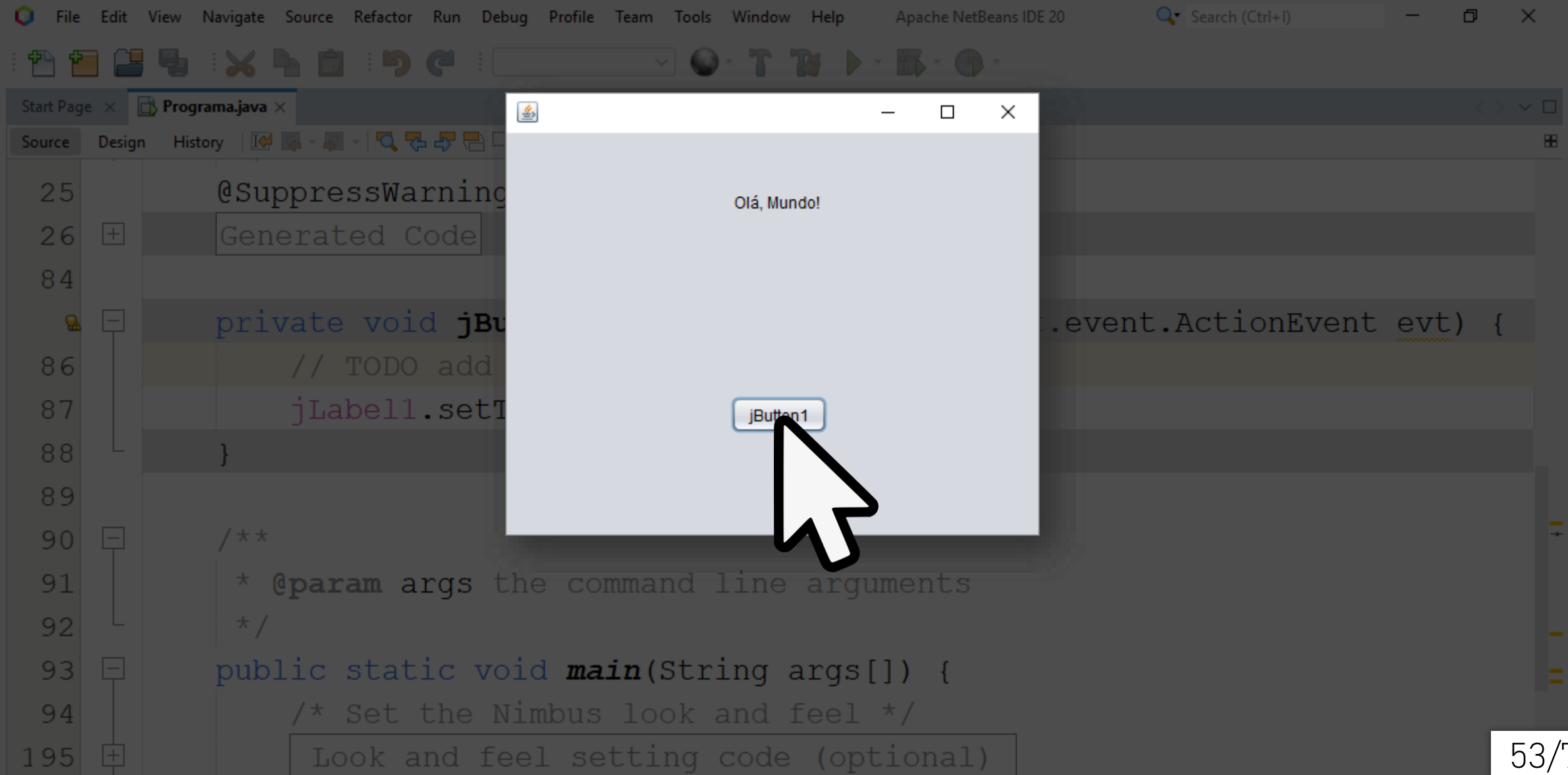
NetBeans

Coloque seu código em `jButton1ActionPerformed` para ser executado quando o botão for pressionado.



NetBeans

Ao executar o código e clicar no botão
você verá o resultado



NetBeans

Clique e arraste o controle **TextField** para o **JPanel1**.
Depois volte para o seu código clicando em **Source**.



The screenshot shows the NetBeans IDE 20 interface. The main window is titled 'Programa.java' and is in the 'Design' view. The design area shows a JPanel1 containing a JLabel with the text 'Olá, Mundo!' and a JButton labeled 'jButton1'. A JTextField control is being dragged from the Palette to the JPanel1. The Palette shows the following controls:

- Swing Containers: Panel, Tabbed Pane, Split Pane, Scroll Pane, Tool Bar, Desktop Pane, Internal Frame, Layered Pane
- Swing Controls: Label, OK Button, Toggle Button, Check Box, Radio Button, Button Group, Combo Box, List, Text Area, Scroll Bar, Slider, Formatted Field, Password Field, Spinner, Text Pane, Editor Pane, Tree

The Properties window for JLabel1 [JLabel] is visible at the bottom right, showing the following properties:

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

NetBeans



Apache NetBeans IDE 20

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects x Files Services

- Interface
 - Source Packages
 - udesc.poo
 - Programa.java
 - Libraries

Start Page x Programa.java x

Source Design History

Use the context menu to access available useful actions for the selected

Olá, Mundo!

jTextField1

jCheckBox1

jButton1

Palette x

- Swing Containers
 - Panel
 - Tabbed Pane
 - Split Pane
 - Scroll Pane
 - Tool Bar
 - Desktop Pane
 - Internal Frame
 - Layered Pane
- Swing Controls
 - Label
 - Radio Button
 - Text Field
 - Progress Bar
 - Separator
 - Table
 - OK Button
 - Button Group
 - Text Area
 - Formatted Field
 - Text Pane
 - Toggle Button
 - Combo Box
 - Scroll Bar
 - Password Field
 - Editor Pane
 - Checkbox**
 - List
 - Slider
 - Spinner
 - Tree
- Swing Menus
- Swing Windows
- Swing Fillers
- AWT
- Beans

jLabel1 [JLabel] - Properties x

Properties Events Code

Properties

background	[242,242,242]
displayedMnemonic	
font	Segoe UI 12 Plain
foreground	[0,0,0]
horizontalAlignment	LEADING
icon	<none>
labelFor	<none>
text	Olá, Mundo!

jPanel1 [JPanel] - Navigator x

- Form Programa
 - Other Components
 - [JFrame]
 - jPanel1 [JPanel]



```
25 @SuppressWarnings("unchecked")
26 Generated Code
84
86 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
87     // TODO add your handling code here:
88     if(jCheckBox1.isSelected()){
89         jLabel1.setText(jTextField1.getText());
90     }else{
91         jLabel1.setText("Falso");
92     }
93 }
94 /**
195  * @param args the command line arguments
```

NetBeans

Modifique `jButton1ActionPerformed` para executar com base no `jCheckBox1` .



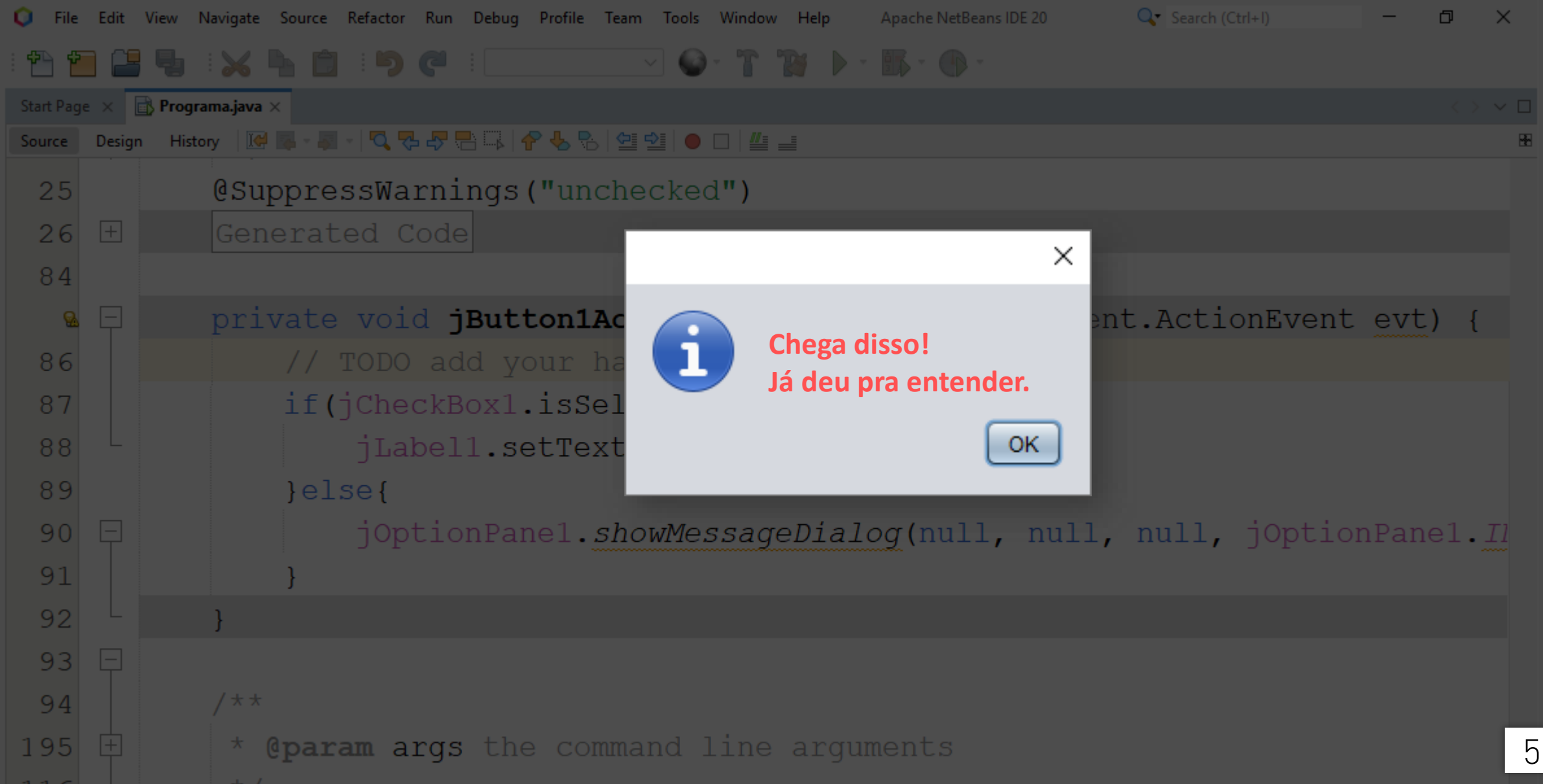
The screenshot displays the NetBeans IDE interface. At the top, the menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The toolbar below contains icons for file operations, running, and debugging. The main workspace is divided into three windows, each showing a different state of a Java Swing application:

- Left Window:** Displays a window titled "Olá, Mundo!" with a text field labeled `jTextField1`, an unchecked checkbox labeled `jCheckBox1`, and a button labeled `jButton1`.
- Middle Window:** Displays the same window, but the checkbox `jCheckBox1` is now checked, and the text field contains the word "Verdadeiro".
- Right Window:** Displays the same window, but the text field is empty, and the checkbox `jCheckBox1` remains checked.

Below the windows, a portion of the `Programa.java` source code is visible, showing a JavaDoc comment for the `main` method:

```
91     }
92 }
93
94 /**
195  * @param args the command line arguments
116  */
```

NetBeans



Métodos

JFrame

setLayout():

Monta o layout padrão de um JFrame, por exemplo, 'null' faz com que os componentes, não se sobreponham e não se conflitem com outros componentes

setDefaultCloseOperation():

Método de como será fechado o frame, por exemplo, 'EXIT_ON_CLOSE' sai totalmente do sistema e da RAM, já o 'DISPOSE_ON_CLOSE' só sai da janela atual

setTitle():

Muda o título do JFrame

setResizable():

Deixa a janela, redimensionável (true) ou não (false)

setBounds():

definir tanto a posição quanto as dimensões de um componente.

add():

adiciona um componente (botão, um painel, etc.), a

Métodos

JFrame

setTitle():

Muda o título do JFrame

setResizable():

Deixa a janela, redimensionável (true) ou não (false)

setBounds():

definir tanto a posição quanto as dimensões de um componente.

add():

adiciona um componente (botão, um painel, etc.) a janela;

setVisible():

Deixando a janela visível (true), ou invisível (false)

setSize():

definir apenas as dimensões (largura e altura) de um componente.

`setTitle():` Muda o título do JFrame

SCHILD, Herbert. Java para iniciantes.

A forma geral de `setDefaultCloseOperation()` é mostrada aqui:

```
void setDefaultCloseOperation(int o que se deseja)
```

O valor passado em *o que se deseja* determina o que ocorrerá quando a janela for fechada. Há várias outras opções além de **JFrame.EXIT_ON_CLOSE**. Eles são mostrados abaixo:

`JFrame.DISPOSE_ON_CLOSE`

Destrói a janela (aplicação ainda roda)

`JFrame.HIDE_ON_CLOSE`

Fecha a janela (aplicação ainda roda)

`JFrame.DO_NOTHING_ON_CLOSE`

Desabilita cliques

Seus nomes refletem suas ações. Essas constantes foram declaradas em **WindowConstants**, uma interface declarada no pacote **javax.swing** que é implementada por **JFrame**.

setBorder():

Define uma borda para o painel, veremos mais a frente sobre bordas;

setBackground():

Define a cor de fundo do painel

setBounds():

Define onde o painel estará dentro do componente que o abriga, mesmo método do JFrame

add():

Adiciona um componente (botão, um painel, etc.) a janela;

Métodos

JLabel

setText():

Altera o texto da JLabel

getText():

Pega o texto atual do JLabel

**set
Foreground():**

Muda a cor da letra do texto

setIcon():

Define o ícone exibido

**setHorizontal
Alignment():**

Define o alinhamento horizontal

**setVerticalAl
ignment():**

Define o alinhamento vertical

setText():

Altera o texto da JLabel

- **setHorizontalAlignment(int arg0):** define o alinhamento horizontal:
 - **JLabel.LEFT:** texto começa na esquerda;
 - **JLabel.CENTER:** texto começa no centro;
 - **JLabel.RIGHT:** texto começa na direita;
- **setVerticalAlignment(int arg0):** define o alinhamento vertical:
 - **JLabel.TOP:** texto começa na topo;
 - **JLabel.CENTER:** texto começa no centro;
 - **JLabel.BOTTOM:** texto começa em baixo;

setHorizontalAlignment():

Define o alinhamento horizontal

setVerticalAlignment():

Define o alinhamento vertical

setText():

Altera o texto do JTextField

getText():

Retorna a String que está dentro da caixa

grabFocus():

Muda o foco atual para o componente JTextField

setEditable():

Define se o usuário pode ou não alterar seu texto

setColumns():

Define a quantidade de colunas que a caixa vai ter
(não é a quantidade de caracteres!)

Métodos

JButton

**set
Foreground():**

Muda a cor do texto do componente

**set
Background():**

Muda a cor de fundo do componente

**addAction
Listener():**

Adiciona um observador que observa quando o botão é pressionado e executa uma ação

**setTool
TipText():**

Mostra um texto quando o cursor do mouse estiver sobre o componente

setText():

Define o texto do botão

setIcon():

Adiciona um ícone ao botão

Métodos

JButton

setForeground(): Muda a cor do texto do componente

set Muda a cor de fundo do componente

- De fato, faz sentido ficar “observando” algo e disparar um evento quando algo acontecer;
- Para isso, os botões em Java observam eventos pré-programados em instâncias de classes que implementam a interface `ActionListener`;
- Para uma classe implementar essa interface, ela só precisa implementar o método **`actionPerformed()`**;

setText(): Define o texto do botão

setIcon(): Adiciona um ícone ao botão

Métodos

JCheckBox

setText():

Define o texto que ficará ao lado da caixa de marcação

setSelected():

Define se a caixa está marcada ou não

isSelected():

Retorna true caso a caixa esteja marcada e false caso contrário

setIcon():

Se você preferir uma imagem ao invés de um texto, utilize esse método

**set
Foreground():**

Muda a cor do texto do componente

**Set
Background():**

Muda a cor de fundo do componente

getSelectedItem():

Retorna um Object contendo o objeto selecionado

getSelectedItemIndex():

Retorna a posição do array em que se encontra o item atualmente selecionado

insertItemAt():

Adiciona um item no index especificado

removeItem():

Remove o item passado como parâmetro

removeItemAt():

Remove o item na posição passada como parâmetro

getItemAt():

Métodos

JComboBox

insertItemAt():

Adiciona um item no index especificado

removeItem():

Remove o item passado como parâmetro

**removeItem
At():**

Remove o item na posição passada como parâmetro

getItemAt():

Retorna o item na posição tal

**set
Foreground():**

Muda a cor do texto do componente

**Set
Background():**

Muda a cor de fundo do componente

Métodos

JList

**getSelected
Indices():**

Retorna um array de inteiros contendo os indexes selecionados

**getSelected
Values():**

Retorna os itens selecionados na forma de um array de Generics T

**getSelected
ValuesList():**

Retorna os itens selecionados na forma de uma List de T

**getSelected
Index():**

Retorna o primeiro index selecionado

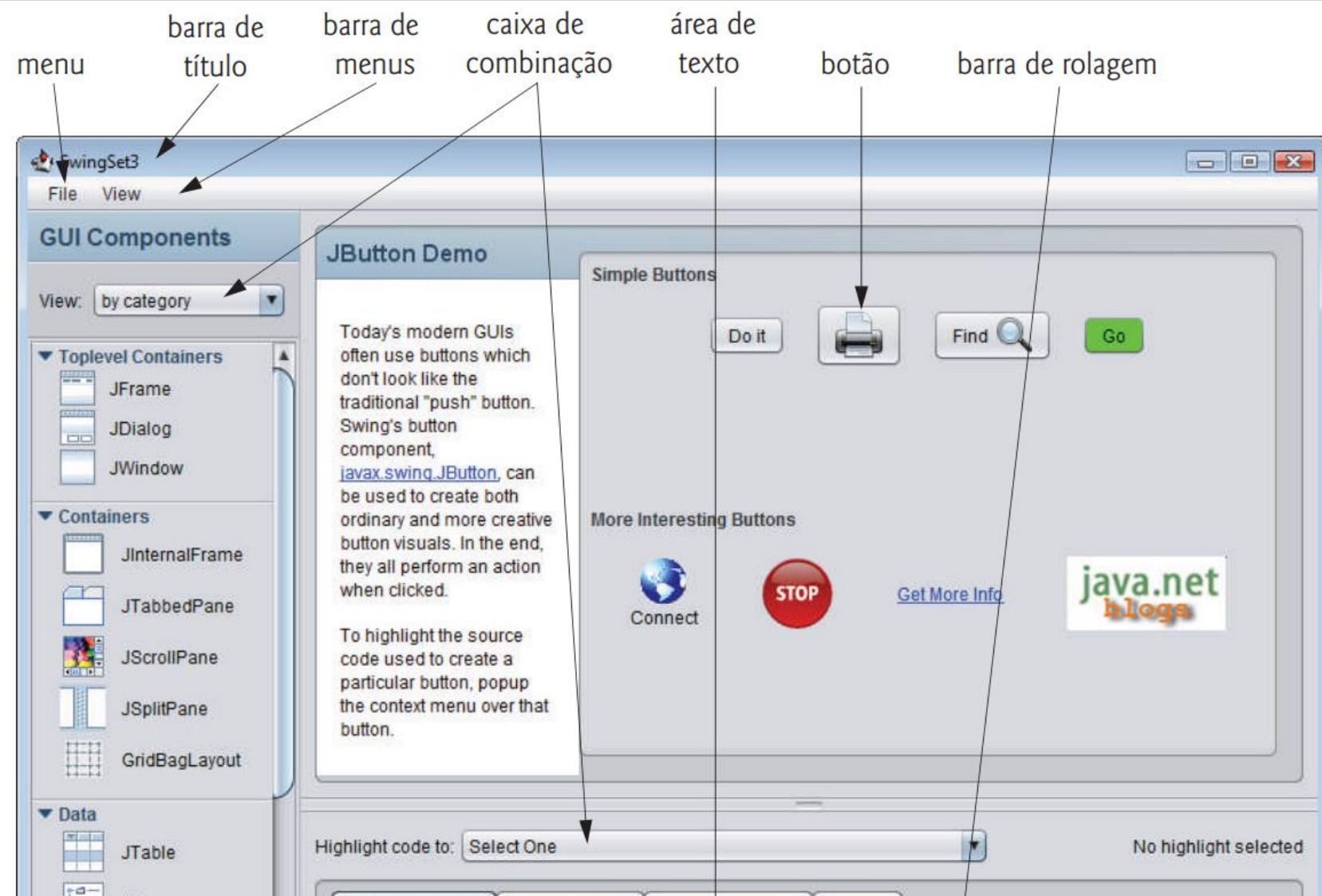
**getSelected
Value():**

Retorna o primeiro item selecionado

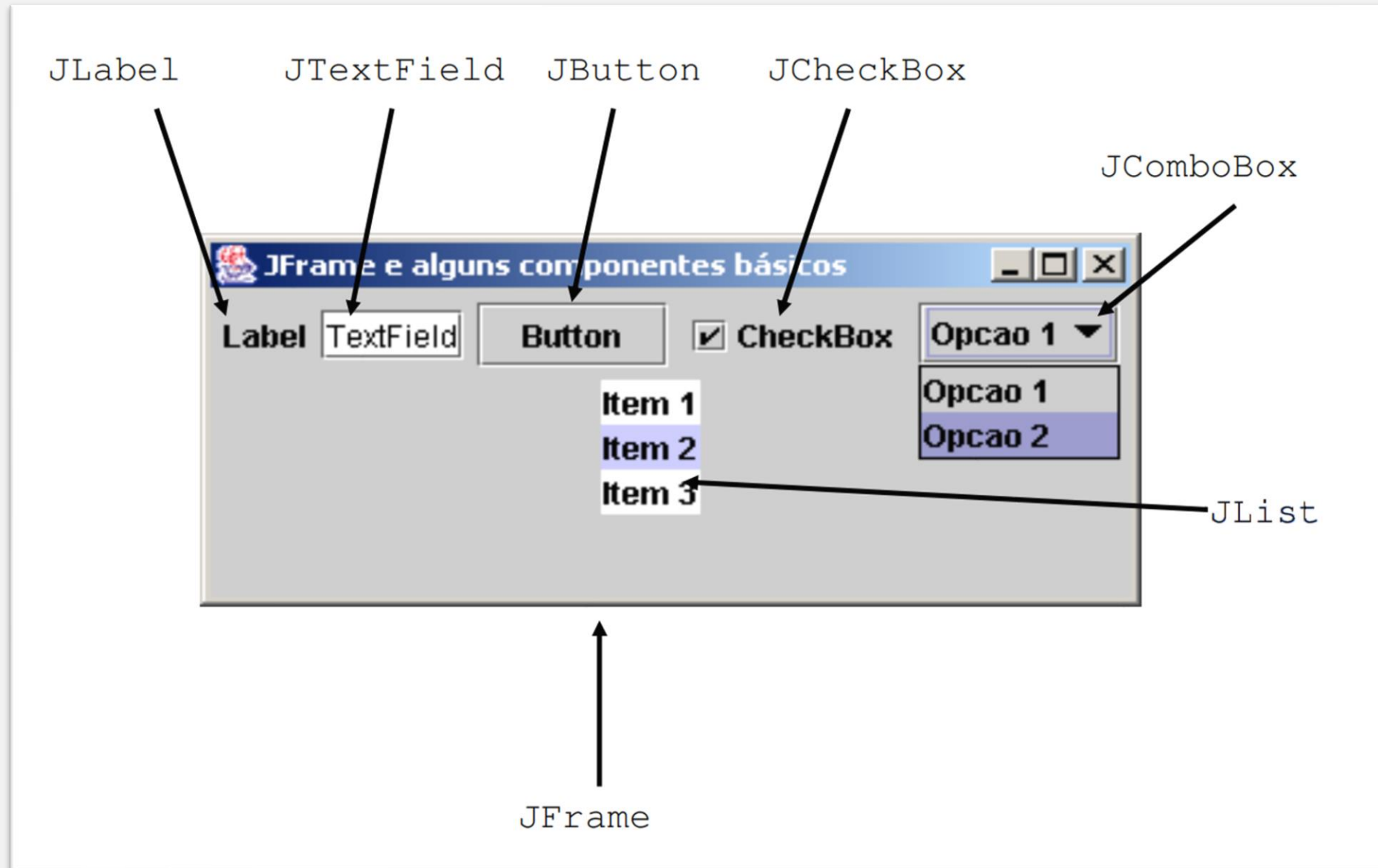
removeSelecti

Componentes básicos

DEITEL, Paul. Java: como programar.



Componentes básicos



Definição

DEITEL, Paul. Java: como programar.

Componente	Descrição
JLabel	Exibe <i>texto</i> e/ou ícones <i>não editáveis</i> .
TextField	Normalmente <i>recebe entrada</i> do usuário.
Button	Dispara um evento quando o usuário clicar nele com o mouse.
CheckBox	Especifica uma opção que pode <i>ser</i> ou <i>não selecionada</i> .
ComboBox	Uma <i>lista drop-down dos itens</i> a partir dos quais o <i>usuário</i> pode fazer uma <i>seleção</i> .
List	Uma <i>lista dos itens</i> a partir dos quais o usuário pode fazer uma <i>seleção clicando</i> em <i>qualquer um</i> deles. <i>Múltiplos</i> elementos <i>podem</i> ser selecionados.
Panel	Uma área em que os <i>componentes</i> podem ser <i>colocados</i> e <i>organizados</i> .

Definição

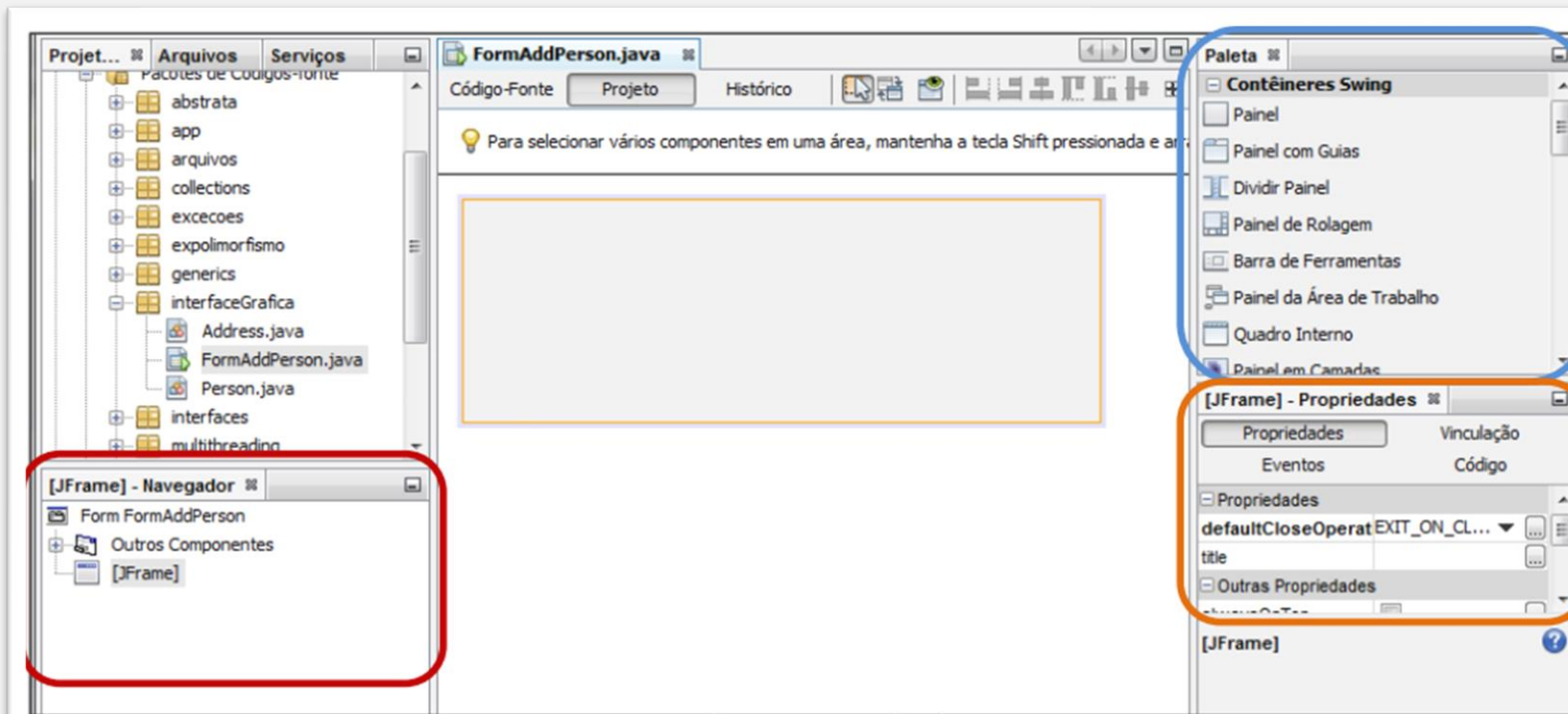


FIGURA 3 – Edição de um Formulário em Java: o retângulo vermelho demarca a região que exibe os objetos vinculados a janela que foram criados, o retângulo azul demarca os principais contêineres disponíveis na GUI Swing, e o retângulo laranja demarca região de propriedades do elemento selecionado.



Resumo

- ❑ **Interface gráfica:** Ambiente de interação visual baseado em uma série de elementos gráficos, textuais e/ou sonoros.
- ❑ **AWT:** 1º pacote Java para desenvolvimento de interfaces. Delega a elaboração das interfaces ao sistema operacional.
- ❑ **Swing:** 2º pacote Java para desenvolvimento de interfaces, focado apenas em *desktop*. Fica a cargo da elaboração das interfaces. **Não** tem o objetivo de substituir o **1º pacote**.
- ❑ **JavaFX:** 3º pacote Java para desenvolvimento de interfaces de vários dispositivos. **Tem** objetivo de substituir o **2º pacote**.
- ❑ **NetBeans:** Ambiente de desenvolvimento integrado da linguagem Java com uma série de recursos voltados a facilitar o desenvolvimento de interfaces gráficas, sem a necessidade de programação textual.

Referências

SCHILDT, Herbert. **Java para iniciantes: crie, compile e execute**. 6. ed. Porto Alegre: Bookman, 2015. Disponível em: <<https://app.minhabiblioteca.com.br/reader/books/9788582603376>>.

SILVA, Fabricio; LEITE, Márcia; OLIVEIRA, Diego. **Paradigmas de Programação**. Porto Alegre: SAGAH, 2019. Disponível em: <<https://app.minhabiblioteca.com.br/reader/books/9788533500426>>.

DEITEL, Paul. **Java: como programar**. 4. ed. Porto Alegre: Bookman, 2003.

DEITEL, Paul. **Java: como programar**. 10. ed. São Paulo: Pearson Education do Brasil, 2017.

INTERFACE GRÁFICA DO USUÁRIO

Alexandre Mendonça Fava
alexandre.fava@udesc.br

Universidade do Estado de Santa Catarina – UDESC
Programa de Pós-graduação em Computação Aplicada – PPGCA