

Programação Visual com Java Swing

Aula 01 · Frames e componentes básicos com layout absoluto

Práticas de POO

Ciência da Computação · Graduação

JFrame · setBounds · JLabel · JTextField · JPasswordField · JCheckBox · JRadioButton · JComboBox · JButton

Prof. Me. Célio Ricardo Castelano

O que você vai aprender hoje

01

Conceitos de GUI

Entender o que é uma interface gráfica e por que usamos Swing no Java.

02

Hierarquia e JFrame

Conhecer as superclasses dos componentes e criar a janela principal.

03

Layout absoluto

Posicionar componentes com precisão usando `setBounds(x, y, w, h)`.

04

7 componentes básicos

`JLabel`, `JTextField`, `JPasswordField`, `JCheckBox`, `JRadioButton`, `JComboBox` e `JButton`.

05

As 4 etapas

Declarar, instanciar, posicionar e adicionar componentes no frame.

06

Boas práticas

EDT, constantes, nomes significativos e organização limpa do código.

O que é uma GUI?

GUI – Graphical User Interface

É o mecanismo visual que permite ao usuário interagir com uma aplicação de forma amigável, sem digitar comandos em um terminal.

CONTÉM TIPICAMENTE

- Barra de título
- Barra de menus
- Botões
- Caixas de combinação
- Campos de texto
- Barras de rolagem

POR QUE USAR GUI?

- Mais amigável e intuitiva
- Feedback visual imediato
- Padrão em apps modernos
- Facilita validação de dados

Swing vs AWT: leves vs pesados

AWT

Componentes pesados

- Pacote java.awt.*
- Atrelados ao SO (peer nativo)
- Aparência varia por plataforma
- Precursor do Swing

SWING

Componentes leves

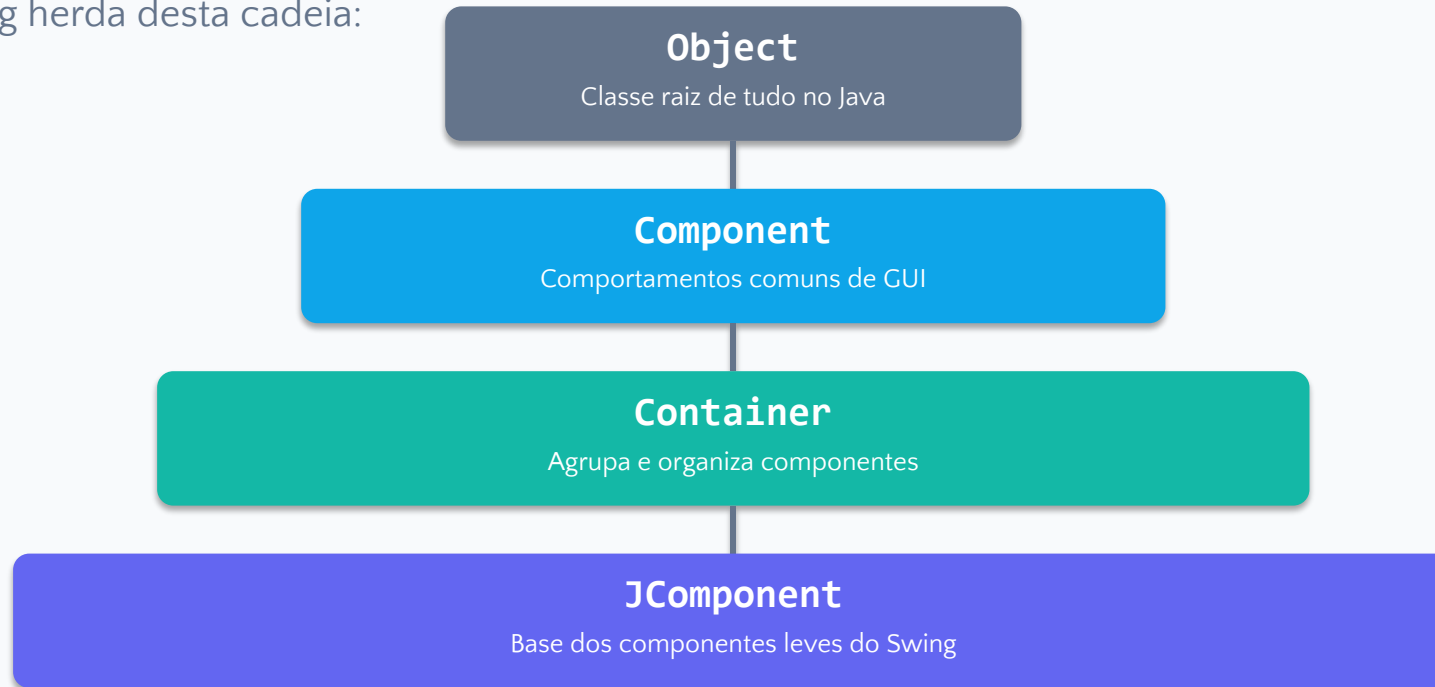
- Pacote javax.swing.*
- Desenhados em Java puro
- Aparência consistente (Look & Feel)
- Muito mais recursos

Leve vs pesado

Componentes leves não dependem de recursos nativos do sistema operacional, permitindo aparência e comportamento uniformes em qualquer plataforma.

Hierarquia dos componentes Swing

Todo componente Swing herda desta cadeia:



Por que isso importa?

Como todos herdam de JComponent, métodos como `setBounds()`, `setVisible()` e `add()` funcionam de forma parecida em quase todos os componentes.

JFrame: a janela principal

- Todo aplicativo Swing começa com um **JFrame**.
- Ele é o contêiner raiz que abriga os demais componentes.



Barra de título

Identifica a janela; contém minimizar, maximizar e fechar.

Área de conteúdo

Recebe os componentes via método `add()`.

Botões de controle

Minimizar, maximizar/restaurar e fechar a aplicação.

JFrame: métodos essenciais

```
MinhaJanela.java

01 import javax.swing.*;
02
03 public class MinhaJanela extends JFrame {
04
05     public MinhaJanela() {
06         setTitle("Minha primeira janela");           // título da barra
07         setSize(500, 350);                           // largura e altura (px)
08         setLocationRelativeTo(null);                 // centraliza na tela
09         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // fecha o app
10         setVisible(true);                           // torna a janela visível
11     }
12
13     public static void main(String[] args) {
14         new MinhaJanela();
15     }
16 }
```

Layout absoluto: controle total

- No **layout absoluto** você define manualmente a posição e o tamanho de cada componente, sem um gerenciador automático.

COMO ATIVAR

```
setLayout(null);
```

Chame **setLayout(null)** no JFrame antes de posicionar os componentes.
Isso desativa o gerenciador padrão.

QUANDO USAR

- Aprendizado e protótipos
- Telas com posições fixas
- Controle pixel a pixel

CUIDADO

- Não se adapta a redimensionamento

setBounds(x, y, largura, altura)

```
Assinatura do método  
01 // posiciona e dimensiona um componente  
02 componente.setBounds(x, y, largura, altura);
```

x

Distância da borda ESQUERDA do frame (pixels)

y

Distância da borda SUPERIOR do frame (pixels)

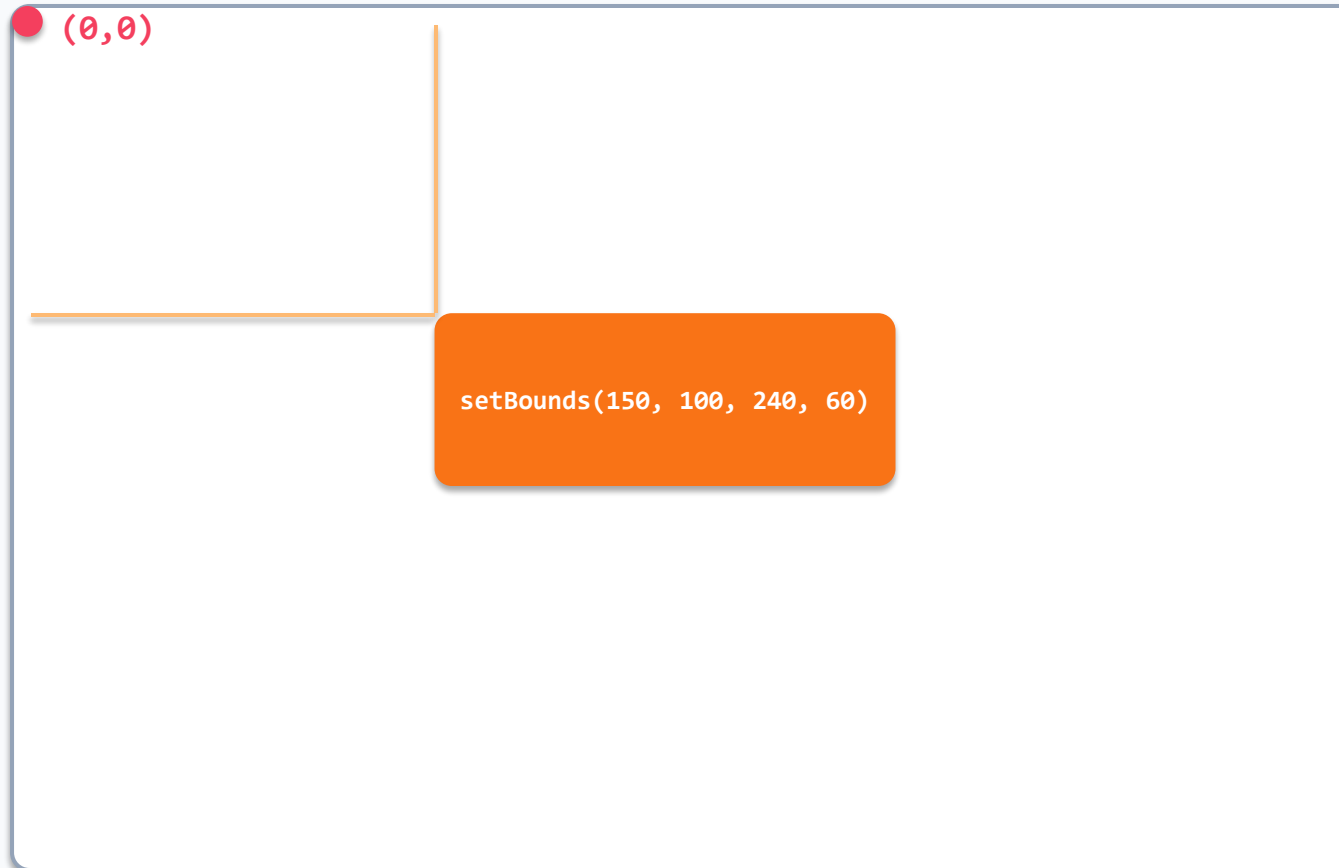
largura

Largura do componente (pixels)

altura

Altura do componente (pixels)

Sistema de coordenadas do frame



ATENÇÃO

- Origem no canto superior esquerdo
- y aumenta para baixo
- Unidade: pixels

Dica

- Diferente da matemática, aqui o eixo y aponta para baixo.

As 4 etapas para usar um componente

1

Declarar

Crie a variável do componente como atributo da classe.

```
private JLabel lblNome;
```

2

Instanciar

Crie o objeto com new.

```
lblNome = new JLabel("Nome");
```

3

Posicionar

Defina posição e tamanho com setBounds.

```
lblNome.setBounds(15, 90, 100, 25);
```

4

Adicionar

Inclua o componente no frame com add().

```
add(lblNome);
```

Você repete essas 4 etapas para CADA componente da sua tela.

As 4 etapas na prática

```
Aplicando as 4 etapas

01  // ETAPA 1 – declarar (como atributo da classe)
02  private JLabel lblNome;
03
04  // ETAPA 2 – instanciar
05  lblNome = new JLabel("Nome:");
06
07  // ETAPA 3 – posicionar (x, y, largura, altura)
08  lblNome.setBounds(15, 90, 100, 25);
09
10  // ETAPA 4 – adicionar no frame
11  add(lblNome);
```

Os 7 componentes desta aula

JLabel

Rótulo de texto

TextField

Campo de texto

JPasswordField

Campo de senha

JCheckBox

Caixa de seleção

JRadioButton

Seleção única

JComboBox

Lista suspensa

JButton

Botão de ação

JLabel

Rótulo de texto — exibe informação

- Exibe texto ou imagem na tela.
- Não é editável pelo usuário. Usado para títulos, nomes de campos e mensagens.

MÉTODOS PRINCIPAIS

- **setText(txt)** Altera o texto exibido
- **getText()** Retorna o texto atual
- **setIcon(icone)** Mostra uma imagem

Cadastro de Alunos

Nome:

```
JLabel.java  
  
01 // cria um rótulo com texto  
02 JLabel lblNome = new JLabel("Nome:");  
03 // posiciona: x, y, largura, altura  
04 lblNome.setBounds(20, 50, 80, 25);  
05 add(lblNome); // adiciona no frame
```

TextField

Campo de texto de uma linha

- Permite ao usuário digitar uma linha de texto.
- É o campo de entrada mais comum em formulários.

MÉTODOS PRINCIPAIS

- `getText()` Lê o texto digitado
- `setText(txt)` Preenche o campo
- `setEditable(bool)` Trava ou libera edição

```
JTextField.java
01 // campo com 20 colunas de largura
02 JTextField edtNome = new JTextField(20);
03 edtNome.setBounds(100, 50, 250, 25);
04 add(edtNome);
```

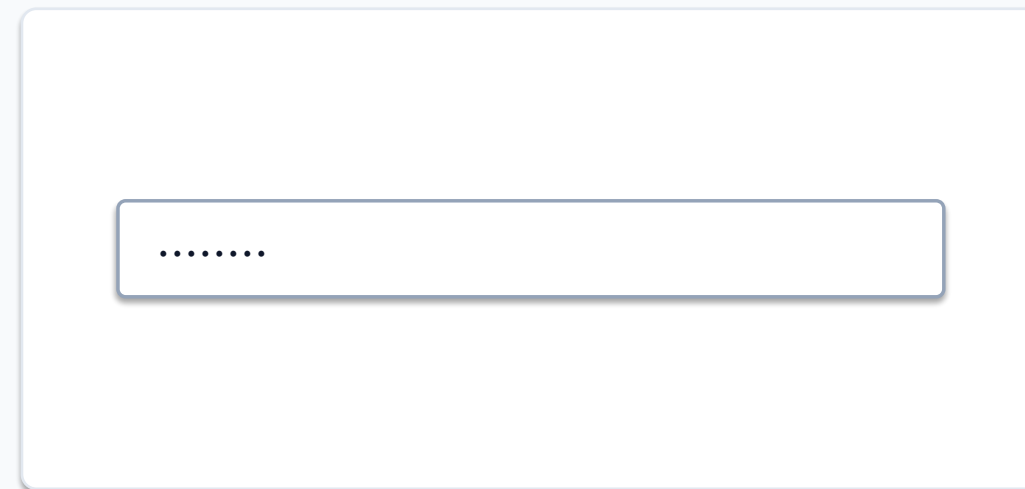
JPasswordField

Campo de senha (texto oculto)

- Campo de texto que esconde os caracteres digitados com um símbolo (.).
- Ideal para senhas.

MÉTODOS PRINCIPAIS

- **getPassword()** Retorna char[] com a senha
- **setEchoChar(c)** Troca o caractere de máscara
- **setText(txt)** Preenche o campo



```
JPASSWORDFIELD.java
01 // campo de senha com 10 colunas
02 JPasswordField pwdSenha = new JPasswordField(10);
03 pwdSenha.setBounds(100, 90, 150, 25);
04 add(pwdSenha);
```

JCheckBox

Caixa de seleção — liga/desliga

- Opção que pode estar marcada ou desmarcada.
- Vários checkboxes podem estar marcados ao mesmo tempo.

MÉTODOS PRINCIPAIS

- `isSelected()` Retorna true se marcado
- `setSelected(bool)` Marca ou desmarca
- `getText()` Retorna o rótulo



Aluno ativo

```
JCheckBox.java
01 // caixa de seleção com rótulo
02 JCheckBox chkAtivo = new JCheckBox("Ativo");
03 chkAtivo.setBounds(100, 130, 120, 25);
04 add(chkAtivo);
```

JRadioButton

Seleção única em um grupo

- Botão de rádio: apenas UMA opção do grupo pode estar selecionada.
- Use `ButtonGroup` para agrupá-los.

MÉTODOS PRINCIPAIS

- `isSelected()` Retorna true se escolhido
- `setSelected(bool)` Seleciona o botão
- `ButtonGroup.add()` Agrupa os botões



```
JRadioButton.java

01  JRadioButton rdbMasc = new JRadioButton("Masculino");
02  JRadioButton rdbFem  = new JRadioButton("Feminino");
03  ButtonGroup grupo = new ButtonGroup();
04  grupo.add(rdbMasc);  // só um pode ser escolhido
05  grupo.add(rdbFem);
```

JComboBox

Lista suspensa de opções

- Mostra uma lista de opções em um menu suspenso.
- O usuário escolhe apenas uma entre várias.

MÉTODOS PRINCIPAIS

- `getSelectedItem()` Retorna o item escolhido
- `setSelectedIndex(i)` Seleciona pela posição
- `addItem(item)` Adiciona uma opção



```
JComboBox.java
01 String[] semestres = {"1º", "2º", "3º", "4º"};
02 JComboBox<String> cboSem = new JComboBox<>(semestres);
03 cboSem.setBounds(100, 170, 100, 25);
04 add(cboSem);
```

JButton

Botão de ação

- Botão clicável que dispara uma ação.
- É o componente mais usado para confirmar ou cancelar operações.

MÉTODOS PRINCIPAIS

- **setText(txt)** Altera o texto do botão
- **setEnabled(bool)** Ativa ou desativa
- **addActionListener(l)** Define a ação (próx. aula)



```
01 JButton btnGravar = new JButton("Gravar");
02 btnGravar.setBounds(150, 220, 110, 30);
03 add(btnGravar);
```

Cadastro de Aluno – 1/4: estrutura e declaração

```
CadastroAluno.java – parte 1

01 import javax.swing.*;
02 import java.awt.*;
03
04 public class CadastroAluno extends JFrame {
05
06     // ETAPA 1 – declarar os componentes (atributos da classe)
07     private JLabel lblTitulo, lblNome, lblSenha;
08     private JTextField edtNome;
09     private JPasswordField pwdSenha;
10     private JCheckBox chkAtivo;
11     private JRadioButton rdbMasculino, rdbFeminino;
12     private ButtonGroup grpSexo;
13     private JComboBox<String> cboSemestre;
14     private JButton btnGravar, btnLimpar;
15     private String[] semestres = {"1º", "2º", "3º", "4º"};
```

Cadastro de Aluno – 1/4: estrutura e declaração

```
CadastroAluno.java – parte 1

17     public CadastroAluno() {
18         setTitle("Cadastro de Aluno");
19         setPreferredSize(new Dimension(520, 420));
20         setLayout(null);           // layout absoluto
21         instanciar(); adicionar(); posicionar();
22         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
23         pack();
24         setVisible(true);
25     }
```

Cadastro de Aluno – 2/4: instanciar

```
CadastroAluno.java – parte 2

01     private void instanciar() {
02         // ETAPA 2 – instanciar cada componente com new
03         lblTitulo = new JLabel("Cadastro de Aluno");
04         lblNome    = new JLabel("Nome");
05         lblSenha   = new JLabel("Senha");
06         edtNome    = new JTextField(30);
07         pwdSenha   = new JPasswordField(12);
08         chkAtivo   = new JCheckBox("Aluno ativo");
09         rdbMasculino = new JRadioButton("Masculino");
10         rdbFeminino = new JRadioButton("Feminino");
11         grpSexo     = new ButtonGroup();
12         grpSexo.add(rdbMasculino); // agrupa: só um pode ser escolhido
13         grpSexo.add(rdbFeminino);
14         cboSemestre = new JComboBox<>(semestres);
15         btnGravar   = new JButton("Gravar");
16         btnLimpar   = new JButton("Limpar");
17     }
```

Cadastro de Aluno – 3/4: posicionar

```
CadastroAluno.java – parte 3

01     private void posicionar() {
02         // ETAPA 3 – posicionar com setBounds(x, y, largura, altura)
03         lblTitulo.setBounds(170, 15, 200, 25);
04         lblNome.setBounds(30, 60, 60, 25);
05         edtNome.setBounds(100, 60, 350, 25);
06         lblSenha.setBounds(30, 100, 60, 25);
07         pwdSenha.setBounds(100, 100, 150, 25);
08         chkAtivo.setBounds(100, 140, 150, 25);
09         rdbMasculino.setBounds(100, 180, 110, 25);
10         rdbFeminino.setBounds(220, 180, 110, 25);
11         cboSemestre.setBounds(100, 220, 120, 25);
12         btnGravar.setBounds(130, 280, 110, 30);
13         btnLimpar.setBounds(260, 280, 110, 30);
14     }
```

Cadastro de Aluno – 4/4: adicionar e main

```
CadastroAluno.java – parte 4

01     private void adicionar() {
02         // ETAPA 4 – adicionar cada componente no frame
03         add(lblTitulo);
04         add(lblNome);      add(edtNome);
05         add(lblSenha);     add(pwdSenha);
06         add(chkAtivo);
07         add(rdbMasculino); add(rdbFeminino);
08         add(cboSemestre);
09         add(btnGravar);    add(btnLimpar);
10     }
11
12     public static void main(String[] args) {
13         // Boa prática: cria a GUI na Event Dispatch Thread
14         SwingUtilities.invokeLater(() -> new CadastroAluno());
15     }
16 }
```

Como a janela ficará ao executar

JTextField

JPasswordField

JCheckBox

JRadioButton

JComboBox

The screenshot shows a Java Swing window titled "Cadastro de Aluno". The window contains a form with the following elements:

- Nome:** A text input field.
- Senha:** A password input field with masked characters (dots).
- Aluno ativo:** A checked checkbox.
- Sexo:** Two radio buttons, "Masculino" (selected) and "Feminino".
- Curso:** A dropdown menu showing "2°".
- Buttons:** "Gravar" (Save) and "Limpar" (Clear).

Crie a GUI na Event Dispatch Thread

- O Swing **não é thread-safe**.
- Criar ou modificar componentes fora da thread dedicada (EDT) pode causar travamentos e comportamento imprevisível.

```
01 public static void main(String[] args) {  
02  
03     // ERRADO – cria a GUI na thread principal  
04     // new CadastroAluno();  
05  
06     // CERTO – agenda a criação na Event Dispatch Thread  
07     SwingUtilities.invokeLater(() -> new CadastroAluno());  
08 }
```

invokeLater não bloqueia

- Ele coloca a criação da janela na fila de eventos do Swing e retorna imediatamente.
 - Assim, toda a interface é construída na thread correta.

Código limpo desde a primeira aula

Nomes com prefixo

lbl, txt, pwd, chk, rdb, cbo, btn indicam o tipo do componente.

```
JLabel lblNome; JButton btnGravar;
```

Separe em métodos

instanciar(), posicionar() e adicionar() deixam o código legível.

```
private void posicionar() { ... }
```

Use constantes

Evite "números mágicos" de posição e tamanho espalhados.

```
private static final int LARGURA = 520;
```

Generics no combo

JComboBox<String> evita erros de tipo em tempo de compilação.

```
JComboBox<String> cboSemestre;
```

Erros comuns para evitar



Esquecer o add()

Se não adicionar o componente ao frame, ele não aparece na tela.



Esquecer setVisible(true)

A janela só fica visível depois dessa chamada.



Esquecer setLayout(null)

Sem isso, o setBounds é ignorado pelo gerenciador padrão.



setBounds fora da área

Posições além do tamanho do frame ficam invisíveis.



JRadioButton sem ButtonGroup

Sem agrupar, o usuário consegue marcar vários ao mesmo tempo.



Chamar pack() antes de posicionar

O pack() recalcula tamanhos; use depois de adicionar os componentes.

Exercícios: construa telas com os componentes

EXERCÍCIO 1

Fácil

Perfil básico

Monte um frame 450x320 com um JLabel de título, 3 JLabel + 3 JTextField (nome, idade, e-mail) e um JButton "OK". Posicione tudo com setBounds.

EXERCÍCIO 2

Médio

Cadastro de produto

Monte um frame 500x380 com JLabels para JTextField (nome), JComboBox (categoria), JCheckBox (disponível), 2 JRadioButton com ButtonGroup (nacional/importado) e botões "Gravar" e "Limpar".

EXERCÍCIO 3

Desafio

Tela de login

Monte um frame 420x300 com JTextField (usuário), JPasswordField (senha), JCheckBox (lembrar-me) e botões "Entrar"/"Cancelar". Use constantes para as posições. Bônus: inserir uma imagem, deixando o layout mais profissional, consegue????

Não é necessário tratar eventos (cliques) — foque em criar e posicionar os componentes corretamente.

Resumo da aula

Você construiu sua primeira GUI em Swing

- GUI e componentes leves do Swing
- JFrame como janela principal
- Layout absoluto com setBounds
- As 4 etapas: declarar, instanciar, posicionar, adicionar
- 7 componentes: JLabel a JButton
- Boas práticas: EDT, nomes e organização

PRÓXIMA AULA

Eventos e ActionListener

Vamos fazer os botões responderem ao clique: tratamento de eventos, leitura dos campos e validação.

```
btnGravar.addActionListener(...)
```