Menus em Java Swing

Aula 03 · JMenu, JPopupMenu, Look and Feel e JInternalFrame

Práticas de POO

Ciência da Computação · Graduação

`JMenuBar` · `JMenu` · `JMenuItem` · `JPopupMenu` · `UIManager` · `JDesktopPane`

AULA 03

O que você vai aprender hoje

01

Menus em Java

Construir barras de menu com JMenuBar, JMenu e JMenuItem.

02

Itens especiais

Usar JCheckBoxMenuItem, JRadioButtonMenuItem e mnemônicos.

03

Submenus e separadores

Organizar menus em hierarquia com separadores lógicos.

04

Menus de contexto

Criar menus pop-up sensíveis ao botão direito com JPopupMenu.

05

Look and Feel

Personalizar a aparência com UIManager.

06

Janelas internas

Montar interfaces MDI com JDesktopPane e JInternalFrame.

CONCEITO

O que são menus?

Menus
permitem que o usuário realize ações sem poluir a interface com botões e componentes extras. Eles agrupam opções de forma organizada e só ocupam espaço quando são abertos.

VANTAGENS

- Interface mais limpa
- Ações agrupadas por categoria
- Padrão conhecido pelo usuário
- Suporte a atalhos de teclado

ONDE SÃO USADOS

- Barra de menus (topo da janela)
- Menus de contexto (botão direito)
- Submenus em cascata

ESTRUTURA

Os componentes de menu do Swing

`JMenuBar`

A barra que abriga os menus

`JMenu`

Cada menu individual (ex.: Arquivo)

`JMenuItem`

Item clicável dentro do menu

`JCheckBoxMenuItem`

Item que liga/desliga

`JRadioButtonMenuItem`

Item de escolha única

`JPopupMenu`

Menu de contexto flutuante

COMPONENTES

JMenuBar: a barra de menus

- A `JMenuBar` é o contêiner que fica no topo do JFrame e abriga todos os menus.
- Ela deve ser anexada com o método `setJMenuBar()`

.

Criando e anexando a barra

```
01  // cria a barra de menus
02  JMenuBar barraMenus = new JMenuBar();
03
04  // anexa a barra ao JFrame
05  setJMenuBar(barraMenus);
```

Erro comum

Esquecer de chamar setJMenuBar() faz a barra não aparecer na janela.

COMPONENTES

JMenu: cada menu da barra

- Cada **JMenu** é um item da barra que, ao ser clicado, abre uma lista de opções.
- Um JMenu também pode ser adicionado dentro de outro JMenu, criando um submenu.

Criando um menu

```
01  // cria o menu Arquivo
02  JMenu menuArquivo = new JMenu("Arquivo");
03
04  // adiciona o menu à barra
05  barraMenus.add(menuArquivo);
```

Submenus
Um submenu é criado adicionando um JMenu dentro de outro JMenu com o método add().

COMPONENTES

JMenuItem: a ação do menu

- O **JMenuItem** é o elemento clicável que dispara uma ação.
- É ele que recebe o ActionListener, assim como um JButton.

Criando um item com ação

```
01  // cria o item de menu Sair
02  JMenuItem itemSair = new JMenuItem("Sair");
03
04  // adiciona o item ao menu
05  menuArquivo.add(itemSair);
06
07  // registra a ação (encerra o aplicativo)
08  itemSair.addActionListener(e -> System.exit(0));
```

COMPONENTES

Itens com seleção: CheckBox e Radio

JCheckBoxMenuItem

Item que pode estar ligado ou desligado, de forma independente.

```
JCheckBoxMenuItem itemNegrito =
    new JCheckBoxMenuItem("Negrito");
menuFormato.add(itemNegrito);
```

JRadioButtonMenuItem

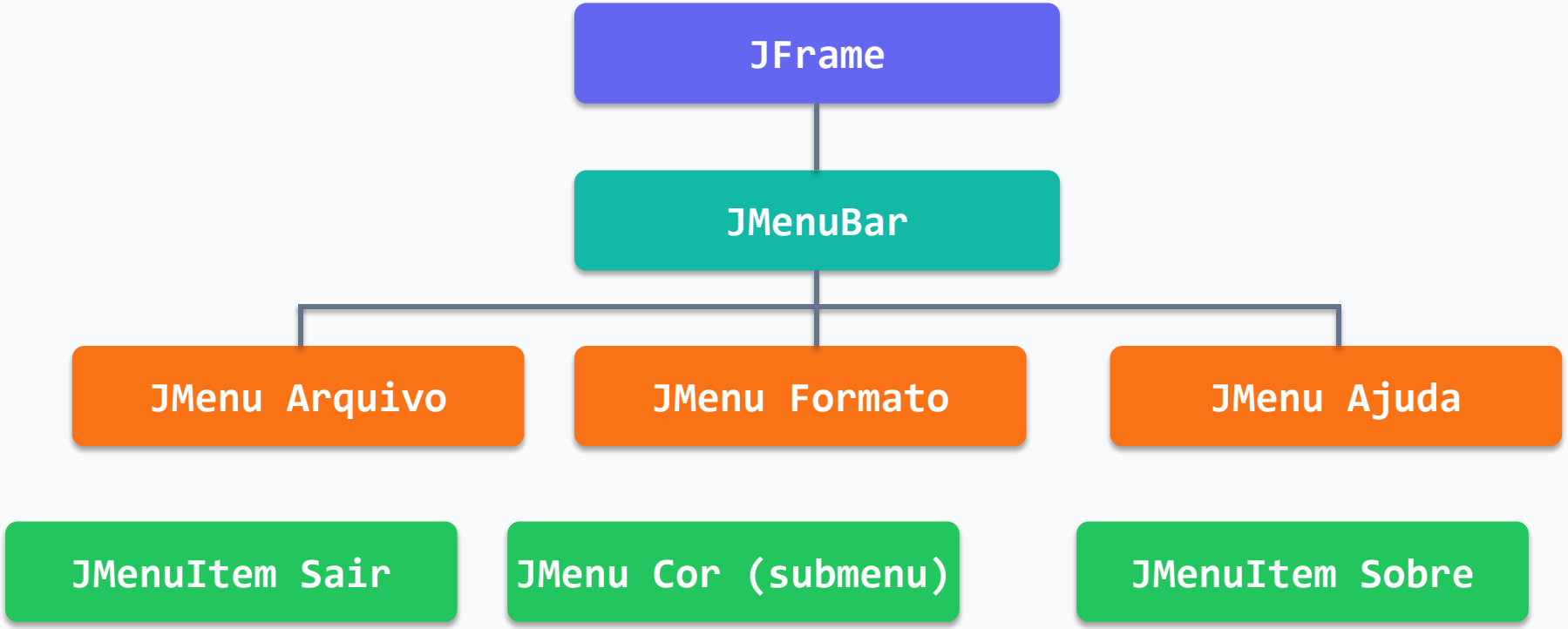
Itens de escolha única. Agrupe com ButtonGroup para que apenas um fique selecionado.

```
JRadioButtonMenuItem itemAzul =
    new JRadioButtonMenuItem("Azul");
JRadioButtonMenuItem itemAmarelo =
    new JRadioButtonMenuItem("Amarelo");

ButtonGroup grupo = new ButtonGroup();
grupo.add(itemAzul);
grupo.add(itemAmarelo);
```

ESTRUTURA

Montando a hierarquia de menus



Regra de montagem
JMenuItem e JMenu são adicionados a um JMenu; JMenu é adicionado à JMenuBar; JMenuBar é anexada ao JFrame.

RECURSO

Mnemônicos: atalhos pelo teclado

- **Mnemônicos** são teclas de atalho que abrem menus ou ativam itens pressionando **Alt + letra.**
- A letra aparece sublinhada no rótulo.

Definindo mnemônicos

```
01  // configura o mnemônico A para abrir o menu Arquivo
02  menuArquivo.setMnemonic('A');
03
04  // configura o mnemônico S para o item Sair
05  itemSair.setMnemonic('S');
```

Boa prática
Use mnemônicos diferentes para cada menu e item.
Normalmente usa-se a primeira letra do rótulo.

EXEMPLO

Menu completo — 1/6: estrutura

MenuExemplo.java — parte 1

```
01  import javax.swing.*;
02  import java.awt.*;
03
04  public class MenuExemplo extends JFrame {
05
06      private JMenuBar barraMenus;
07      private JMenu menuArquivo, menuFormato, menuCor;
08      private JMenuItem itemSobre, itemSair;
09      private JRadioButtonMenuItem itemAzul, itemVermelho;
10      private ButtonGroup grupoCores;
11      private JLabel lblExemplo;
12
```

EXEMPLO

Menu completo — 2/6: estrutura

MenuExemplo.java — parte 1

```
12
13      public MenuExemplo() {
14          super("Exemplo de Menus");
15
16          // rótulo central que será modificado pelos menus
17          lblExemplo = new JLabel("Texto de exemplo", SwingConstants.CENTER);
18          lblExemplo.setFont(new Font("Serif", Font.PLAIN, 48));
19          add(lblExemplo, BorderLayout.CENTER);
20
21          criarMenus();   // monta toda a barra de menus
22      }
23
```

EXEMPLO

Menu completo — 3/6: montando os menus

MenuExemplo.java — parte 2

```
23
24      private void criarMenus() {
25          barraMenus = new JMenuBar();
26          setJMenuBar(barraMenus);            // anexa a barra ao JFrame
27
28          // ----- Menu Arquivo -----
29          menuArquivo = new JMenu("Arquivo");
30          menuArquivo.setMnemonic('A');        // atalho Alt+A
31          barraMenus.add(menuArquivo);
32
33          itemSobre = new JMenuItem("Sobre...");
34          itemSobre.setMnemonic('S');
35          menuArquivo.add(itemSobre);
36
```

EXEMPLO

Menu completo — 4/6: montando os menus

MenuExemplo.java — parte 2

```
36
37          menuArquivo.addSeparator();          // linha separadora
38
39          itemSair = new JMenuItem("Sair");
40          itemSair.setMnemonic('x');
41          itemSair.addActionListener(e -> System.exit(0));
42          menuArquivo.add(itemSair);
43
```

EXEMPLO

Menu completo — 5/6: submenu e main

MenuExemplo.java — parte 3

```
43
44          // ----- Menu Formato com submenu Cor -----
45          menuFormato = new JMenu("Formato");
46          menuFormato.setMnemonic('F');
47          barraMenus.add(menuFormato);
48
49          menuCor = new JMenu("Cor");          // submenu dentro de Formato
50          menuFormato.add(menuCor);
51
```

EXEMPLO

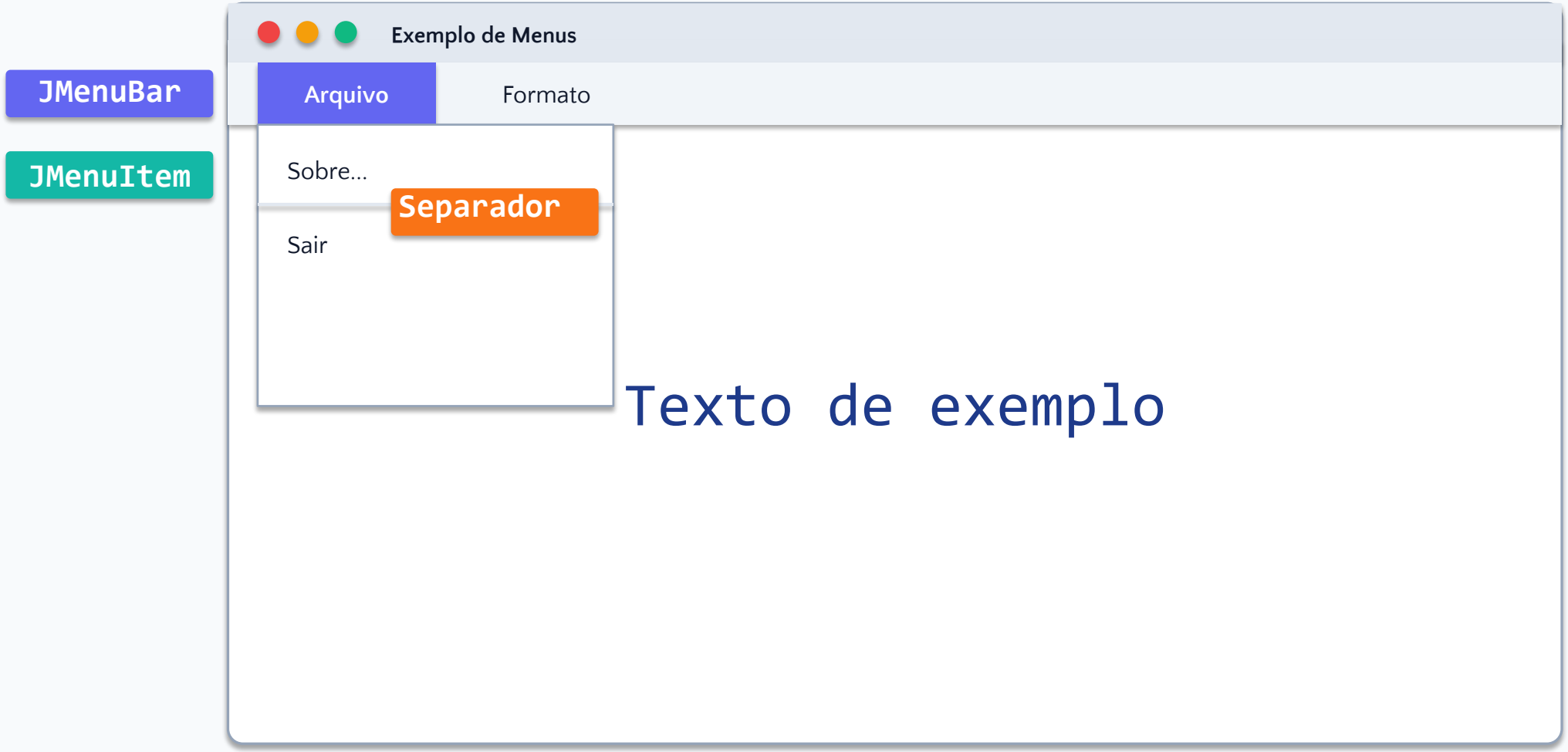
Menu completo — 6/6: submenu e main

MenuExemplo.java — parte 3

```
51
52          grupoCores = new ButtonGroup();
53          itemAzul = new JRadioButtonMenuItem("Azul", true);   // selecionado
54          itemVermelho = new JRadioButtonMenuItem("Vermelho");
55          grupoCores.add(itemAzul);
56          grupoCores.add(itemVermelho);
57          menuCor.add(itemAzul);
58          menuCor.add(itemVermelho);
59      }
60
61      public static void main(String[] args) {
62          SwingUtilities.invokeLater(() -> new MenuExemplo().setVisible(true));
63      }
64  }
```

RESULTADO

O menu em execução



ORGANIZAÇÃO

Separadores e submenus

SEPARADORES

Agrupam itens logicamente dentro de um menu, criando uma linha horizontal entre eles.

```
// adiciona uma linha separadora
menuArquivo.addSeparator();
```

Boa prática

Use separadores para dividir grupos de itens relacionados, como "Abrir/Salvar" de "Sair".

SUBMENUS

Um submenu é criado adicionando um JMenu dentro de outro JMenu. Ele expande ao passar o mouse.

```
// submenu Cor dentro de Formato
menuFormato.add(menuCor);
```

Organização

Submenus evitam menus muito longos, agrupando opções por categoria.

CONTEXTO

JPopupMenu: menu de contexto

- O **JPopupMenu** é um menu flutuante sensível ao contexto.
- Ele aparece quando o usuário clica com o **botão direito do mouse** sobre um componente.

COMO FUNCIONA

- Usuário clica com o botão direito
- Verifica-se isPopupTrigger()
- Exibe-se com popupMenu.show()
- Aparece na posição do clique

POR QUE NOS DOIS EVENTOS?

O gatilho varia por plataforma: em algumas o pop-up abre no **mousePressed**, em outras no **mouseReleased**. Por isso verificamos nos dois.

EXEMPLO

JPopupMenu — 1/4: construção

PopupExemplo.java — parte 1

```
01  import java.awt.Color;
02  import java.awt.event.*;
03  import javax.swing.*;
04
05  public class PopupExemplo extends JFrame {
06
07      private JPopupMenu popupMenu;
08      private JRadioButtonMenuItem items[];
09      private final Color cores[] = { Color.BLUE, Color.YELLOW, Color.RED };
10
```

EXEMPLO

JPopupMenu — 2/4: construção

PopupExemplo.java — parte 1

```
10
11      public PopupExemplo() {
12          super("Usando JPopupMenu");
13          setBackground(Color.WHITE);
14
15          String nomes[] = { "Azul", "Amarelo", "Vermelho" };
16          ButtonGroup grupo = new ButtonGroup();
17          popupMenu = new JPopupMenu();              // cria o menu pop-up
18          items = new JRadioButtonMenuItem[3];
19
```

EXEMPLO

JPopupMenu — 3/4: gatilho e exibição

PopupExemplo.java — parte 2

```
19
20          for (int i = 0; i < items.length; i++) {
21              items[i] = new JRadioButtonMenuItem(nomes[i]);
22              popupMenu.add(items[i]);
23              grupo.add(items[i]);
24          }
25
26          addMouseListener(new MouseAdapter() {
27              public void mousePressed(MouseEvent e)  { verificarGatilho(e); }
28              public void mouseReleased(MouseEvent e) { verificarGatilho(e); }
29          });
30      }
31
```

EXEMPLO

JPopupMenu — 4/4: gatilho e exibição

PopupExemplo.java — parte 2

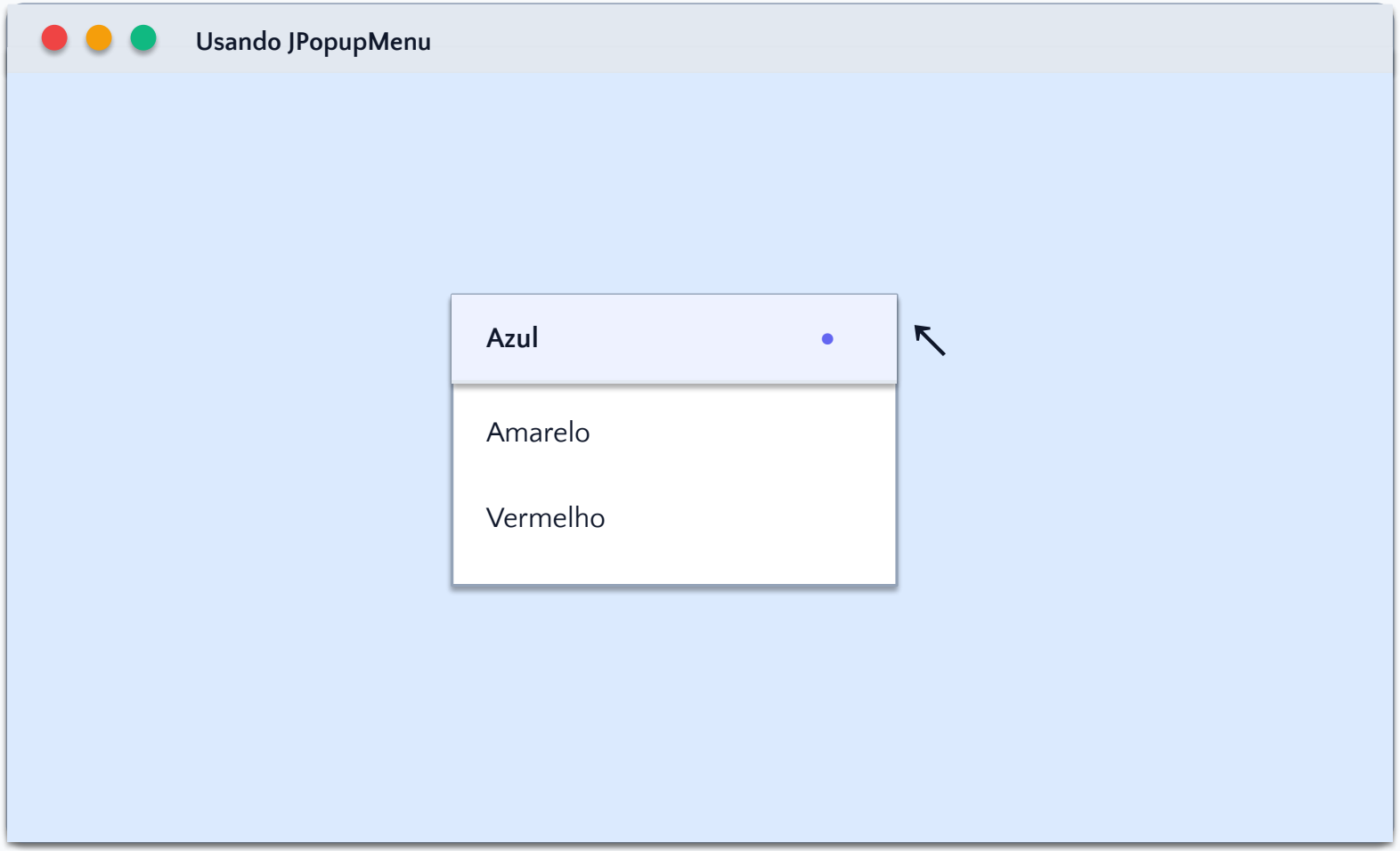
```
31
32      // verifica se o evento deve acionar o menu pop-up
33      private void verificarGatilho(MouseEvent e) {
34          if (e.isPopupTrigger())
35              popupMenu.show(e.getComponent(), e.getX(), e.getY());
36      }
37
38      public static void main(String[] args) {
39          SwingUtilities.invokeLater(() -> new PopupExemplo().setVisible(true));
40      }
41  }
```

RESULTADO

O menu pop-up em ação

JPopupMenu

botão direito



APARÊNCIA

Look and Feel: personalizando o visual

- **Look and Feel**
- Define a aparência e o comportamento visual dos componentes.
- O Swing permite trocar esse estilo em tempo de execução.

Metal

Padrão do Java, igual em qualquer SO

Motif

Estilo clássico do UNIX

Windows

Imita o visual do Windows

Vantagem do Swing

Como os componentes são leves (desenhados em Java), o Swing oferece uma aparência uniforme entre plataformas — ou pode imitar o sistema operacional, se preferir.

EXEMPLO

Look and Feel — trocando a aparência

Alterando o Look and Feel em tempo de execução

```
01  // obtém todas as aparências instaladas no sistema
02  UIManager.LookAndFeelInfo looks[] =
03          UIManager.getInstalledLookAndFeels();
04
05  try {
06      // aplica a aparência escolhida (ex.: posição 0 = Metal)
07      UIManager.setLookAndFeel(looks[0].getClassName());
08
09      // atualiza todos os componentes da janela
10      SwingUtilities.updateComponentTreeUI(minhaJanela);
11  } catch (Exception e) {
12      e.printStackTrace();
13  }
```

Desempenho

getInstalledLookAndFeels() apenas lista os nomes das classes disponíveis — elas só são carregadas quando você realmente aplica uma delas.

MDI

JDesktopPane e JInternalFrame

Interface de Múltiplos Documentos (MDI)
: uma janela principal (janela-pai) contém várias janelas internas (janelas-filhas), gerenciando documentos abertos em paralelo.

JDESKTOPPANE

- A "área de trabalho" da aplicação
- Contêiner das janelas internas
- Adicionado ao JFrame com add()

JINTERNALFRAME

- A "janela-filha" dentro da área
- Pode ser redimensionada e minimizada
- Requer setVisible(true) para aparecer

EXEMPLO

MDI — 1/2: estrutura da área de trabalho

MDIExemplo.java — parte 1

```java
import java.awt.BorderLayout;
import javax.swing.*;

public class MDIExemplo extends JFrame {

    private JDesktopPane desktop;   // área de trabalho
    private int contador = 0;

```

EXEMPLO

MDI — 1/2: estrutura da área de trabalho

MDIExemplo.java — parte 1

```
09      public MDIExemplo() {
10          super("Interface de Múltiplos Documentos");
11
12          // barra de menus com um item para criar janelas
13          JMenuBar barra = new JMenuBar();
14          JMenu menuJanela = new JMenu("Janela");
15          JMenuItem itemNova = new JMenuItem("Nova Janela");
16          itemNova.addActionListener(e -> criarJanelaInterna());
17          menuJanela.add(itemNova);
18          barra.add(menuJanela);
19          setJMenuBar(barra);
20
21          desktop = new JDesktopPane();   // cria a área de trabalho
22          add(desktop);
23      }
```

EXEMPLO

MDI — 2/2: criando janelas internas

MDIExemplo.java — parte 2

```
24
25      private void criarJanelaInterna() {
26          contador++;
27          // título, redimensionável, fechável, maximizável, minimizável
28          JInternalFrame interna = new JInternalFrame(
29                  "Documento " + contador, true, true, true, true);
30
31          interna.setSize(250, 150);
32          interna.setLocation(30 * contador, 30 * contador);  // em cascata
33          interna.setVisible(true);     // obrigatório para aparecer
34          desktop.add(interna);         // anexa à área de trabalho
35      }
36
```

EXEMPLO

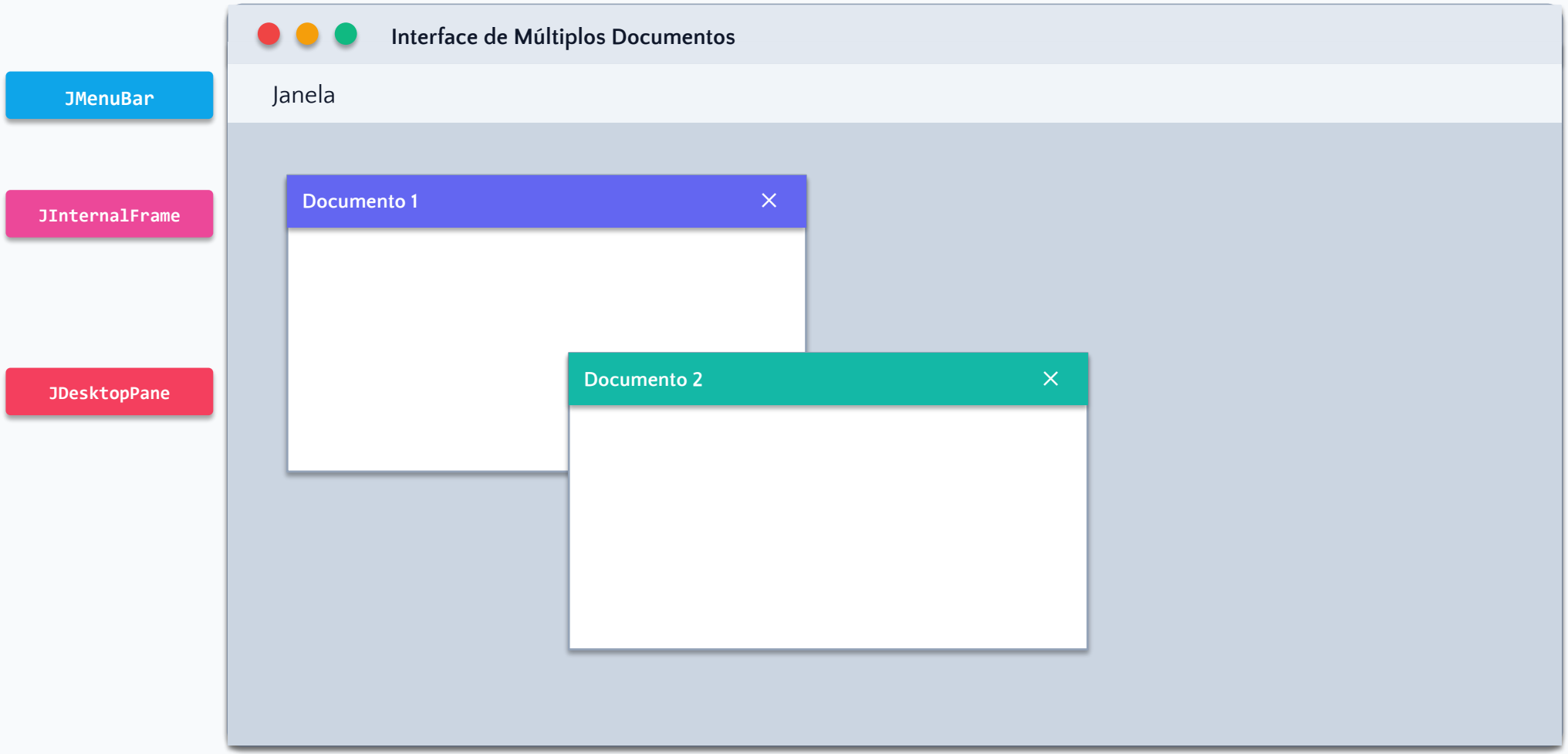
MDI — 2/2: criando janelas internas

MDIExemplo.java — parte 2

```
36
37      public static void main(String[] args) {
38          SwingUtilities.invokeLater(() -> {
39              MDIExemplo janela = new MDIExemplo();
40              janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41              janela.setSize(600, 480);
42              janela.setLocationRelativeTo(null);
43              janela.setVisible(true);
44          });
45      }
46  }
```

RESULTADO

A aplicação MDI em execução



BOAS PRÁTICAS

Escrevendo menus com qualidade

Mnemônicos únicos

Cada menu e item deve ter um mnemônico diferente; use a primeira letra do rótulo.

```
menuArquivo.setMnemonic('A');
```

Crie a GUI na EDT

Sempre use SwingUtilities.invokeLater() para evitar travamentos.

```
SwingUtilities.invokeLater(() -> ...);
```

Organize com separadores

Agrupe itens relacionados usando addSeparator().

```
menuArquivo.addSeparator();
```

Agrupe itens de escolha única

Use ButtonGroup com JRadioButtonMenuItem.

```
grupo.add(itemAzul);
```

ATENÇÃO

Erros comuns ao trabalhar com menus

Esquecer setJMenuBar()

Se não anexar a barra ao JFrame, ela não aparece na janela.

Mnemônicos repetidos

Dois itens com o mesmo mnemônico causam comportamento imprevisível.

JRadioButtonMenuItem sem ButtonGroup

Sem agrupar, o usuário consegue marcar várias opções ao mesmo tempo.

JInternalFrame invisível

É obrigatório chamar setVisible(true) na janela interna.

Pop-up em um evento só

Verifique isPopupTrigger() em mousePressed E mouseReleased.

Trocar Look and Feel fora da EDT

A troca de aparência deve ocorrer na thread de eventos do Swing.

PRÁTICA

Exercícios: construa seus menus

EXERCÍCIO 1 — Fácil

Barra de menus básica

Crie um frame com uma JMenuBar contendo os menus Arquivo (itens Novo, Abrir, separador e Sair), Editar e Ajuda. Configure um mnemônico para cada menu e faça o item Sair encerrar a aplicação.

EXERCÍCIO 2 — Médio

Editor de formato

Crie um frame com um JLabel central e um menu Formato contendo o submenu Cor (JRadioButtonMenuItem: Azul, Vermelho, Verde em um ButtonGroup) e o submenu Estilo (JCheckBoxMenuItem: Negrito e Itálico). Os itens devem alterar a cor e o estilo do rótulo.

EXERCÍCIO 3 — Desafio

Sistema com MDI

Monte o menu corporativo Cadastros (Funcionários, Alunos, Professores), Movimentos (Matrícula, Mensalidades, Fluxo de Caixa), Financeiro (Contas a Receber, Contas a Pagar) e Sair. Ao clicar em um item de cadastro, abra uma JInternalFrame correspondente dentro de um JDesktopPane.

Dica: comece montando a estrutura dos menus vazia, depois adicione os itens e, por último, o tratamento de eventos.

Resumo da aula

Sua aplicação agora tem menus profissionais

- JMenuBar, JMenu e JMenuItem
- JCheckBoxMenuItem e JRadioButtonMenuItem
- Mnemônicos e separadores
- JPopupMenu e menu de contexto
- Look and Feel com UIManager
- MDI com JDesktopPane e JInternalFrame

PRÓXIMA AULA

Gerenciadores de Layout

Vamos abandonar o layout absoluto e usar FlowLayout, BorderLayout e GridLayout para criar interfaces que se adaptam ao redimensionamento.

```
new BorderLayout()
```

Obrigado!

Aula 03 · Menus em Java Swing

Práticas de POO

Prof. Me. Célio Ricardo Castelano · Ciência da Computação