

GUIA JAVA SWING

Guia Completo de Personalização Java Swing

1. Cores e Fundos

Mudando Cores de Fundo

```
// JFrame/JPanel
frame.getContentPane().setBackground(Color.DARK_GRAY);
panel.setBackground(new Color(45, 45, 45)); // RGB
panel.setBackground(Color.decode("#2D2D2D")); // Hexadecimal

// Cores com transparência
panel.setBackground(new Color(255, 0, 0, 128)); // RGBA (alpha = transparência)
```

Gradientes de Fundo

```
public class PainelGradiente extends JPanel {
    @Override
    protected void paintComponent(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        Color cor1 = new Color(135, 206, 250); // Azul claro
        Color cor2 = new Color(25, 25, 112); // Azul escuro

        GradientPaint gradient = new GradientPaint(
            0, 0, cor1,
            0, getHeight(), cor2
        );
        g2d.setPaint(gradient);
        g2d.fillRect(0, 0, getWidth(), getHeight());
    }
}
```

Cores de Componentes

```
// Botões
botao.setBackground(Color.BLUE);
botao.setForeground(Color.WHITE);

// Campos de texto
campo.setBackground(Color.LIGHT_GRAY);
campo.setForeground(Color.BLACK);
campo.setCaretColor(Color.RED); // Cursor

// Labels
label.setForeground(Color.GREEN);
label.setOpaque(true);
label.setBackground(Color.YELLOW);
```

2. Fontes e Texto

Personalizando Fontes

```
// Fonte básica
Font fonte = new Font("Arial", Font.BOLD, 16);
label.setFont(fonte);

// Fontes do sistema
Font fonteSystem = new Font(Font.SANS_SERIF, Font.ITALIC, 12);

// Múltiplos estilos
Font fonteCompleta = new Font("Times New Roman",
    Font.BOLD | Font.ITALIC, 18);

// Listar fontes disponíveis
GraphicsEnvironment ge = GraphicsEnvironment.getLocalGraphicsEnvironment();
String[] fontNames = ge.getAvailableFontFamilyNames();
```

Efeitos de Texto

```
// Texto com sombra
public void paintComponent(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

    // Sombra
    g2d.setColor(Color.GRAY);
    g2d.drawString("Texto", 12, 22);

    // Texto principal
    g2d.setColor(Color.BLACK);
    g2d.drawString("Texto", 10, 20);
}
```

3. Bordas

Tipos de Bordas

```
// Borda simples
panel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2));

// Borda com título
panel.setBorder(BorderFactory.createTitledBorder("Título"));

// Borda elevada/abaixada
panel.setBorder(BorderFactory.createRaisedBevelBorder());
panel.setBorder(BorderFactory.createLoweredBevelBorder());

// Borda composta
Border bordaExterna = BorderFactory.createLineBorder(Color.BLACK, 1);
Border bordaInterna = BorderFactory.createEmptyBorder(10, 10, 10, 10);
panel.setBorder(BorderFactory.createCompoundBorder(bordaExterna, bordaInterna));

// Borda arredondada personalizada
public class BordaArredondada implements Border {
```

```

private int radius;

public BordaArredondada(int radius) {
    this.radius = radius;
}

@Override
public void paintBorder(Component c, Graphics g, int x, int y, int width, int height) {
    Graphics2D g2d = (Graphics2D) g;
    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
    g2d.setColor(Color.GRAY);
    g2d.drawRoundRect(x, y, width-1, height-1, radius, radius);
}

@Override
public Insets getBorderInsets(Component c) {
    return new Insets(radius/2, radius/2, radius/2, radius/2);
}

@Override
public boolean isBorderOpaque() {
    return false;
}
}

```

4. Layout e Espaçamento

Margens e Padding

```

// Margens externas
panel.setBorder(BorderFactory.createEmptyBorder(10, 20, 10, 20)); // top, left, bottom, right

// Espaçamento em GridBagLayout
GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(5, 5, 5, 5); // espaçamento entre componentes

```

```
// Espaçamento em FlowLayout
FlowLayout layout = new FlowLayout(FlowLayout.CENTER, 15, 10); // hgap,
vgap
```

5. 🎯 Componentes Customizados

Botões Personalizados

```
public class BotaoPersonalizado extends JButton {
    private Color corNormal = Color.BLUE;
    private Color corHover = Color.CYAN;
    private Color corPressionado = Color.DARK_GRAY;

    public BotaoPersonalizado(String texto) {
        super(texto);
        setContentAreaFilled(false);
        setFocusPainted(false);
        setBorderPainted(false);

        addMouseListener(new MouseAdapter() {
            @Override
            public void mouseEntered(MouseEvent e) {
                repaint();
            }

            @Override
            public void mouseExited(MouseEvent e) {
                repaint();
            }
        });
    }

    @Override
    protected void paintComponent(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, Rendering
Hints.VALUE_ANTIALIAS_ON);
    }
}
```

```

        Color cor = corNormal;
        if (getModel().isPressed()) {
            cor = corPressionado;
        } else if (getModel().isRollover()) {
            cor = corHover;
        }

        g2d.setColor(cor);
        g2d.fillRoundRect(0, 0, getWidth(), getHeight(), 20, 20);

        super.paintComponent(g);
    }
}

```

Campos de Texto Estilizados

```

public class CampoPersonalizado extends JTextField {
    private String placeholder;

    public CampoPersonalizado(String placeholder) {
        this.placeholder = placeholder;
        setBorder(BorderFactory.createCompoundBorder(
            BorderFactory.createLineBorder(Color.GRAY, 1),
            BorderFactory.createEmptyBorder(5, 10, 5, 10)
        ));
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        if (getText().isEmpty() && !hasFocus()) {
            Graphics2D g2d = (Graphics2D) g;
            g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
            g2d.setColor(Color.GRAY);
            g2d.drawString(placeholder, getInsets().left,

```

```

        g.getFontMetrics().getMaxAscent() + getInsets().top);
    }
}
}

```

6. 🤖 Look and Feel

Alterando o Look and Feel

```

// Look and Feel do sistema
UIManager.setLookAndFeel(UIManager.getSystemLookAndFeel());

// Nimbus (moderno)
UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");

// Metal (padrão Java)
UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

// Listar L&Fs disponíveis
UIManager.LookAndFeelInfo[] infos = UIManager.getInstalledLookAndFeels();
for (UIManager.LookAndFeelInfo info : infos) {
    System.out.println(info.getName() + " - " + info.getClassName());
}

```

Personalizando UIManager

```

// Mudar cores globais
UIManager.put("Panel.background", Color.DARK_GRAY);
UIManager.put("Button.background", Color.BLUE);
UIManager.put("Button.foreground", Color.WHITE);
UIManager.put("TextField.background", Color.LIGHT_GRAY);

// Aplicar mudanças
SwingUtilities.updateComponentTreeUI(frame);

```

7. Ícones e Imagens

Adicionando Ícones

```
// Ícones em botões
Imagelcon icone = new Imagelcon("caminho/para/icone.png");
botao.setIcon(icone);

// Redimensionar ícone
Image img = icone.getImage().getScaledInstance(32, 32, Image.SCALE_SMOOTH);
botao.setIcon(new Imagelcon(img));

// Ícones de estados diferentes
botao.setIcon(iconeNormal);
botao.setRolloverIcon(iconeHover);
botao.setPressedIcon(iconePressed);
botao.setDisabledIcon(iconeDisabled);
```

Imagem de Fundo

```
public class PaineComImagem extends JPanel {
    private Image imagemFundo;

    public PaineComImagem(String caminhaImagem) {
        try {
            imagemFundo = ImageIO.read(new File(caminhaImagem));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (imagemFundo != null) {
            g.drawImage(imagemFundo, 0, 0, getWidth(), getHeight(), this);
        }
    }
}
```



```
}  
}
```

8. ✨ Animações e Efeitos

Animação de Fade

```
public class AnimacaoFade {  
    private Timer timer;  
    private float alpha = 0.0f;  
    private JComponent componente;  
  
    public AnimacaoFade(JComponent comp) {  
        this.componente = comp;  
        timer = new Timer(50, e → {  
            alpha += 0.05f;  
            if (alpha >= 1.0f) {  
                alpha = 1.0f;  
                timer.stop();  
            }  
            componente.repaint();  
        });  
    }  
  
    public void iniciar() {  
        timer.start();  
    }  
  
    // No paintComponent do componente:  
    @Override  
    protected void paintComponent(Graphics g) {  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC  
_OVER, alpha));  
        super.paintComponent(g2d);  
    }  
}
```

Animação de Movimento

```
public void animarMovimento(JComponent comp, Point destino) {
    Timer timer = new Timer(10, null);
    Point inicio = comp.getLocation();
    long tempoInicio = System.currentTimeMillis();
    int duracao = 1000; // 1 segundo

    timer.addActionListener(e → {
        long tempoDecorrido = System.currentTimeMillis() - tempoInicio;
        float progresso = Math.min(1.0f, (float) tempoDecorrido / duracao);

        int x = (int) (inicio.x + (destino.x - inicio.x) * progresso);
        int y = (int) (inicio.y + (destino.y - inicio.y) * progresso);

        comp.setLocation(x, y);

        if (progresso >= 1.0f) {
            timer.stop();
        }
    });

    timer.start();
}
```

9. 🎨 Temas Personalizados

Tema Escuro

```
public class TemaEscuro {
    public static void aplicar() {
        UIManager.put("Panel.background", new Color(45, 45, 45));
        UIManager.put("Label.foreground", Color.WHITE);
        UIManager.put("Button.background", new Color(70, 70, 70));
        UIManager.put("Button.foreground", Color.WHITE);
        UIManager.put("TextField.background", new Color(60, 60, 60));
        UIManager.put("TextField.foreground", Color.WHITE);
        UIManager.put("TextField.caretForeground", Color.WHITE);
    }
}
```

```
}  
}
```

Tema Colorido

```
public class TemaColorido {  
    public static void aplicar() {  
        UIManager.put("Panel.background", new Color(240, 248, 255));  
        UIManager.put("Button.background", new Color(100, 149, 237));  
        UIManager.put("Button.foreground", Color.WHITE);  
        UIManager.put("TextField.background", Color.WHITE);  
        UIManager.put("TextField.selectionBackground", new Color(173, 216, 230));  
    }  
}
```

10. Exemplos Práticos para sua Tela de Login

Login com Tema Escuro

```
public class LoginTemaEscuro extends JFrame {  
    public LoginTemaEscuro() {  
        // Configurações da janela  
        setTitle("Login - Tema Escuro");  
        setSize(400, 300);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLocationRelativeTo(null);  
  
        // Painel principal com gradiente  
        JPanel painelPrincipal = new JPanel() {  
            @Override  
            protected void paintComponent(Graphics g) {  
                Graphics2D g2d = (Graphics2D) g;  
                Color cor1 = new Color(30, 30, 30);  
                Color cor2 = new Color(60, 60, 60);  
                GradientPaint gradient = new GradientPaint(  
                    0, 0, cor1, 0, getHeight(), cor2);  
            }  
        };  
    }  
}
```

```

        g2d.setPaint(gradient);
        g2d.fillRect(0, 0, getWidth(), getHeight());
    }
};

painelPrincipal.setLayout(new BorderLayout());

// Painel do formulário
JPanel painelForm = new JPanel(new GridBagLayout());
painelForm.setOpaque(false);

// Componentes estilizados
JLabel lblTitulo = new JLabel("Sistema de Frequência");
lblTitulo.setFont(new Font("Arial", Font.BOLD, 24));
lblTitulo.setForeground(Color.WHITE);
lblTitulo.setHorizontalAlignment(SwingConstants.CENTER);

JTextField txtEmail = new CampoPersonalizado("Digite seu email");
JPasswordField txtSenha = new JPasswordField();
txtSenha.setBorder(BorderFactory.createCompoundBorder(
    BorderFactory.createLineBorder(Color.GRAY),
    BorderFactory.createEmptyBorder(5, 10, 5, 10)
));

JButton btnEntrar = new BotaoPersonalizado("Entrar");

// Layout
GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10);

gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;
painelForm.add(lblTitulo, gbc);

gbc.gridwidth = 1; gbc.gridy = 1;
painelForm.add(new JLabel("Email:") { { setForeground(Color.WHITE);
}}, gbc);
gbc.gridx = 1;
painelForm.add(txtEmail, gbc);

```

```
gbc.gridx = 0; gbc.gridy = 2;
painelForm.add(new JLabel("Senha:") {{ setForeground(Color.WHITE);
}}, gbc);
gbc.gridx = 1;
painelForm.add(txtSenha, gbc);

gbc.gridx = 0; gbc.gridy = 3; gbc.gridwidth = 2;
painelForm.add(btnEntrar, gbc);

painelPrincipal.add(painelForm, BorderLayout.CENTER);
add(painelPrincipal);
}
}
```

Dicas Importantes

1. **Performance:** Evite `repaint()` desnecessários em animações
2. **Consistência:** Mantenha um padrão visual em toda aplicação
3. **Acessibilidade:** Considere contraste de cores adequado
4. **Responsividade:** Teste em diferentes resoluções
5. **Look and Feel:** Defina antes de criar componentes
6. **Recursos:** Gerencie imagens e recursos adequadamente

Recursos Adicionais

- Use `Graphics2D` para desenhos avançados
- `RenderingHints` para melhor qualidade visual
- `Timer` para animações suaves
- `ComponentListener` para redimensionamento dinâmico
- `MouseListener/MouseMotionListener` para interações customizadas