

**INSTITUTO FEDERAL DE MINAS GERAIS
CAMPUS SÃO JOÃO EVANGELISTA
SISTEMAS DE INFORMAÇÃO**

VALDIR DE SOUZA CARVALHO NETO

Linguagem de Programação I - SI 241
Trabalho Prático 02

SUMÁRIO

1. Exercício 1 - Calculadora de IMC com Interface Gráfica.....	4
2. Exercício 2 - Conversor de moedas.....	14

LISTA DE FIGURAS

Figura 1. Captura de tela do Exercício 01.....	11
Figura 2. Captura de tela do Exercício 01.....	12
Figura 3. Captura de tela do Exercício 01.....	12
Figura 4. Captura de tela do Exercício 01.....	13
Figura 5. Captura de tela do Exercício 01.....	13
Figura 6. Captura de tela do Exercício 02.....	20
Figura 7. Captura de tela do Exercício 02.....	20
Figura 8. Captura de tela do Exercício 02.....	21
Figura 9. Captura de tela do Exercício 02.....	21
Figura 10. Captura de tela do Exercício 02.....	22

1. Exercício 1 - Calculadora de IMC com Interface Gráfica

Crie uma aplicação gráfica em Java Swing que funcione como uma calculadora de IMC (Índice de Massa Corporal). O programa deve permitir que o usuário digite o peso (em quilogramas) e a altura (em metros) em campos de texto (JTextField). Além disso, o usuário deverá selecionar o sexo por meio de um grupo de botões do tipo JRadioButton (masculino ou feminino).

Ao clicar em um botão (JButton) com o rótulo “Calcular IMC”, o programa deverá calcular o valor do IMC utilizando a fórmula:

$$IMC = \frac{peso}{altura^2}$$

Em seguida, o resultado numérico do IMC e uma classificação qualitativa do peso deverão ser exibidos em um JLabel ou em um JOptionPane. A classificação deve considerar o sexo informado, por exemplo:

- **Masculino:** abaixo do peso ($IMC < 20.7$), peso normal ($20.7 \leq IMC < 26.4$), sobrepeso ($26.4 \leq IMC < 27.8$), obesidade leve ($27.8 \leq IMC < 31.1$) e obesidade ($IMC \geq 31.1$).
- **Feminino:** abaixo do peso ($IMC < 19.1$), peso normal ($19.1 \leq IMC < 25.8$), sobrepeso ($25.8 \leq IMC < 27.3$), obesidade leve ($27.3 \leq IMC < 32.3$) e obesidade ($IMC \geq 32.3$).

Certifique-se de que o programa valide os dados informados (por exemplo, não permitir campos vazios ou valores inválidos) e que apenas um dos sexos possa ser selecionado por vez, utilizando um ButtonGroup. O layout e a organização dos componentes devem ser claros e bem distribuídos na janela da aplicação.

Códigos:

Arquivo Ex01 (main):

```
package com.valdir.ex01;

import forms.Main;

/**
 *
 * @author valdi
 */
public class Ex01 {
```

```

    public static void main(String[] args) {
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(() -> new
Main().setVisible(true));
    }
}

```

Arquivo Main (*form*):

```

package forms;

import javax.swing.ButtonModel;
import javax.swing.JOptionPane;

/**
 *
 * @author valdi
 */
public class Main extends javax.swing.JFrame {

    private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(Main.class.getName());

    /**
     * Creates new form Main
     */
    public Main() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize
the form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">
    private void initComponents() {

```

```

        bgSexo = new javax.swing.ButtonGroup();
        lbTitulo = new javax.swing.JLabel();
        lbPeso = new javax.swing.JLabel();
        lbAltura = new javax.swing.JLabel();
        lbSexo = new javax.swing.JLabel();
        tfPeso = new javax.swing.JTextField();
        tfAltura = new javax.swing.JTextField();
        rbMasculino = new javax.swing.JRadioButton();
        rbFeminino = new javax.swing.JRadioButton();
        btOK = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Calculadora de IMC");

        lbTitulo.setFont(new java.awt.Font("Times New Roman", 1, 18));
// NOI18N

lbTitulo.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        lbTitulo.setText("Calculadora de IMC (Índice de Massa
Corporal)");

        lbPeso.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
        lbPeso.setText("Digite seu peso (quilogramas):");

        lbAltura.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
        lbAltura.setText("Digite sua altura (metros):");

        lbSexo.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
        lbSexo.setText("Escolha seu sexo:");

        bgSexo.add(rbMasculino);
        rbMasculino.setFont(new java.awt.Font("Times New Roman", 0,
14)); // NOI18N
        rbMasculino.setText("Masculino");
        rbMasculino.setActionCommand("M");

        bgSexo.add(rbFeminino);
        rbFeminino.setFont(new java.awt.Font("Times New Roman", 0,

```

```

14)); // NOI18N
    rbFeminino.setText("Feminino");
    rbFeminino.setActionCommand("F");

    btOK.setFont(new java.awt.Font("Times New Roman", 1, 18)); //
NOI18N
    btOK.setText("Calcular IMC");
    btOK.addActionListener(this::btOKActionPerformed);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(29, 29, 29)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(lbPeso)
                        .addComponent(lbAltura)
                        .addGap(3, 3, 3)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(tfAltura,
javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(tfPeso,
javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(43, 43, 43)
                        .addComponent(lbTitulo))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(29, 29, 29)
                        .addComponent(lbSexo)

```

```

        .addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)

        .addComponent(btOK)
        .addGroup(layout.createSequentialGroup()
            .addComponent(rbMasculino)
            .addGap(41, 41, 41)
            .addComponent(rbFeminino))))
        .addContainerGap(53, Short.MAX_VALUE))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(lbTitulo)
        .addGap(27, 27, 27)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)

        .addComponent(lbPeso)
        .addComponent(tfPeso,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(25, 25, 25)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)

        .addComponent(lbAltura)
        .addComponent(tfAltura,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(26, 26, 26)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)

        .addComponent(lbSexo)
        .addComponent(rbMasculino)
        .addComponent(rbFeminino))

```



```

        .addGap(45, 45, 45)
        .addComponent(btOK)
        .addContainerGap(56, Short.MAX_VALUE))

    );

    pack();
    setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents

    private void btOKActionPerformed(java.awt.event.ActionEvent evt)
{ // GEN-FIRST: event_btOKActionPerformed

        if (tfPeso.getText().trim().isEmpty() ||
tfAltura.getText().trim().isEmpty()) {
            JOptionPane.showMessageDialog(this, "Por favor, preencha
os campos de Peso e Altura.", "Campo Vazio",
JOptionPane.WARNING_MESSAGE);
            return;
        }

        if (bgSexo.getSelection() == null) {
            JOptionPane.showMessageDialog(this, "Por favor, selecione
o Sexo (Masculino ou Feminino).", "Sexo não selecionado",
JOptionPane.WARNING_MESSAGE);
            return;
        }

        try {
            String textoPeso = tfPeso.getText().replace(",", ".");
            String textoAltura = tfAltura.getText().replace(",", ".");

            float peso = Float.parseFloat(textoPeso);
            float altura = Float.parseFloat(textoAltura);

            if (peso <= 0 || altura <= 0) {
                JOptionPane.showMessageDialog(this, "Peso e Altura
devem ser valores maiores que zero.", "Valor Inválido",
JOptionPane.ERROR_MESSAGE);
                return;
            }

            float imc = peso / (altura * altura);

```

```

String classificacao = "";

String op = bgSexo.getSelection().getActionCommand();

if ("M".equals(op)) {
    if (imc < 20.7) {
        classificacao = "Abaixo do Peso";
    } else if (imc < 26.4) {
        classificacao = "Peso Normal";
    } else if (imc < 27.8) {
        classificacao = "Sobrepeso";
    } else if (imc < 31.1) {
        classificacao = "Obesidade Leve";
    } else {
        classificacao = "Obesidade";
    }
} else {
    if (imc < 19.1) {
        classificacao = "Abaixo do Peso";
    } else if (imc < 25.8) {
        classificacao = "Peso Normal";
    } else if (imc < 27.3) {
        classificacao = "Sobrepeso";
    } else if (imc < 32.3) {
        classificacao = "Obesidade Leve";
    } else {
        classificacao = "Obesidade";
    }
}

String resultadoFormatado = String.format("Classificação:
%s\nIMC= %.2f", classificacao, imc);

JOptionPane.showMessageDialog(this, resultadoFormatado,
"Resultado do IMC", JOptionPane.INFORMATION_MESSAGE);

} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(this, "Digite apenas números
válidos para Peso e Altura.", "Erro de Formato",
JOptionPane.ERROR_MESSAGE);
}

} //GEN-LAST:event_btOKActionPerformed

```

```

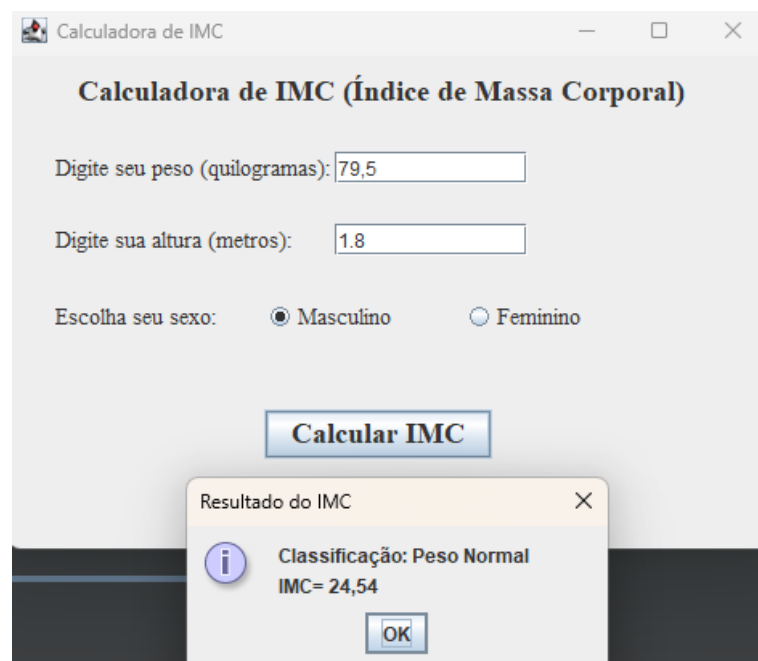
/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.ButtonGroup bgSexo;
private javax.swing.JButton btOK;
private javax.swing.JLabel lbAltura;
private javax.swing.JLabel lbPeso;
private javax.swing.JLabel lbSexo;
private javax.swing.JLabel lbTitulo;
private javax.swing.JRadioButton rbFeminino;
private javax.swing.JRadioButton rbMasculino;
private javax.swing.JTextField tfAltura;
private javax.swing.JTextField tfPeso;
// End of variables declaration//GEN-END:variables
}

```

Imagens:

Figura 1. Captura de tela do Exercício 01.



Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2026.

Figura 2. Captura de tela do Exercício 01.

The screenshot shows a Windows application titled "Calculadora de IMC". The main window has a title bar with standard minimize, maximize, and close buttons. The content area is titled "Calculadora de IMC (Índice de Massa Corporal)". It contains three input fields: "Digite seu peso (quilogramas):" with the value "62", "Digite sua altura (metros):" with the value "1,65", and "Escolha seu sexo:" with radio buttons for "Masculino" and "Feminino" (the latter is selected). Below these fields is a blue button labeled "Calcular IMC". A modal dialog box titled "Resultado do IMC" is open in the foreground. It features an information icon (i) and displays the text "Classificação: Peso Normal" and "IMC= 22,77". At the bottom of the dialog is an "OK" button.

Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2026.

Figura 3. Captura de tela do Exercício 01.

This screenshot is similar to the previous one, showing the same "Calculadora de IMC" application. However, the weight input field now contains the value "72". The "Calcular IMC" button remains visible. The "Resultado do IMC" dialog box is still open, but it now displays "Classificação: Sobrepeso" and "IMC= 26,45". The "OK" button is still present at the bottom of the dialog.

Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2026.

Figura 4. Captura de tela do Exercício 01.

The screenshot shows a window titled "Calculadora de IMC" with a subtitle "Calculadora de IMC (Índice de Massa Corporal)". It contains two input fields: "Digite seu peso (quilogramas):" with the value "100" and "Digite sua altura (metros):" with the value "1,8". Below these is a label "Escolha seu sexo:" with two radio buttons: "Masculino" (unselected) and "Feminino" (selected). A blue button labeled "Calcular IMC" is positioned above a smaller dialog box titled "Resultado do IMC". This dialog box displays an information icon, the text "Classificação: Obesidade Leve", "IMC= 30,86", and an "OK" button.

Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2026.

Figura 5. Captura de tela do Exercício 01.

This screenshot is similar to the previous one, showing the same "Calculadora de IMC" window. However, the weight input field now contains the value "120". The "Calcular IMC" button is again above the "Resultado do IMC" dialog box, which now displays "Classificação: Obesidade" and "IMC= 37,04".

Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2026.

2. Exercício 2 - Conversor de moedas

Crie uma aplicação gráfica em Java Swing para simular um conversor de moedas. A interface deve permitir que o usuário digite um valor em reais (R\$) em um campo de texto (JTextField) e selecione, por meio de uma ComboBox (JComboBox), a moeda para a qual deseja converter o valor. As opções disponíveis na JComboBox devem ser, por exemplo: Dólar (USD), Euro (EUR), Libra (GBP) e Peso Argentino (ARS).

Ao clicar em um botão (JButton) com o rótulo “Converter”, o programa deverá exibir o resultado da conversão em um JLabel ou em um JOptionPane, mostrando o valor convertido com duas casas decimais. Utilize taxas de câmbio fixas para simplificar o exercício (por exemplo: 1 USD = 5,60 BRL; 1 EUR = 6,10 BRL; 1 GBP = 7,00 BRL; 1 ARS = 0,03 BRL).

O programa deve validar a entrada, impedindo o cálculo se o valor informado estiver vazio, for negativo ou contiver caracteres inválidos. A interface deve ser clara e organizada, com rótulos apropriados para cada componente, e deve permitir que o usuário altere a moeda facilmente pela JComboBox antes de realizar novas conversões.

Códigos:

Arquivo Ex02 (*main*):

```
package com.valdir.ex02;

import forms.Main;

/**
 *
 * @author valdi
 */
public class Ex02 {

    public static void main(String[] args) {
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(() -> new
Main().setVisible(true));
    }
}
```

Arquivo Main (*form*):

```

package forms;

import javax.swing.JOptionPane;

/**
 *
 * @author valdi
 */
public class Main extends javax.swing.JFrame {

    private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(Main.class.getName());

    /**
     * Creates new form Main
     */
    public Main() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize
the form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">//GEN-BEGIN: initComponents
    private void initComponents() {

        lbTitulo = new javax.swing.JLabel();
        lbValor = new javax.swing.JLabel();
        tfValor = new javax.swing.JTextField();
        cbMoedaConversao = new javax.swing.JComboBox<>();
        lbMoedaConversao = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Conversor de Moedas");
    }
}

```

```

        lbTitulo.setFont(new java.awt.Font("Times New Roman", 1, 18));
// NOI18N
        lbTitulo.setText("Conversor de Moedas");

        lbValor.setFont(new java.awt.Font("Times New Roman", 0, 14));
// NOI18N
        lbValor.setText("Digite o valor em reais (R$):");

        cbMoedaConversao.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Dólar (USD)", "Euro
(EUR)", "Libra (GBP)", "Peso Argentino (ARS)", "Iene Japonês (JPY)"
}));

        lbMoedaConversao.setFont(new java.awt.Font("Times New Roman",
0, 14)); // NOI18N
        lbMoedaConversao.setText("Moeda para conversão:");

        jButton1.setFont(new java.awt.Font("Times New Roman", 1, 18));
// NOI18N
        jButton1.setText("Converter");
        jButton1.addActionListener(this::jButton1ActionPerformed);

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(lbTitulo)
                    .addComponent(lbValor)
                    .addComponent(cbMoedaConversao)
                    .addComponent(lbMoedaConversao)
                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE)
                )
            )
        );
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
}

```



```

        .addComponent(cbMoedaConversao,
javax.swing.GroupLayout.PREFERRED_SIZE, 150,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(lbValor)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(tfValor,
javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGroup(layout.createSequentialGroup()
            .addGap(160, 160, 160)
            .addComponent(lbTitulo)))
        .addGap(166, 166, 166))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addComponent(jButton1)
            .addGap(200, 200, 200))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(lbTitulo)
            .addGap(38, 38, 38)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASILINE)
            .addComponent(lbValor)
            .addComponent(tfValor,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addComponent(lbMoedaConversao)
            .addComponent(cbMoedaConversao,
javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
125, Short.MAX_VALUE)
    .addComponent(jButton1)
    .addGap(26, 26, 26)
);

pack();
setLocationRelativeTo(null);
} // </editor-fold> // GEN-END: initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) { // GEN-FIRST: event_jButton1ActionPerformed
    if (tfValor.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Por favor, preencha o
campo de Valor.", "Campo Vazio", JOptionPane.WARNING_MESSAGE);
        return;
    }
    // Acesso Combo Box
    String moeda = cbMoedaConversao.getSelectedItem().toString();

    try {
        String textoValor = tfValor.getText().replace(",", ".");
        float valorEmReais = Float.parseFloat(textoValor);

        if (valorEmReais <= 0) {
            JOptionPane.showMessageDialog(this, "Valor deve ser
maior que zero.", "Valor Inválido", JOptionPane.ERROR_MESSAGE);
            return;
        }

        double valorConvertido = 0;
        if ("Dólar (USD)".equals(moeda)) {
            valorConvertido = valorEmReais / 5.6;
        } else if ("Euro (EUR)".equals(moeda)) {
            valorConvertido = valorEmReais / 6.1;
        } else if ("Libra (GBP)".equals(moeda)) {
            valorConvertido = valorEmReais / 7;
        } else if ("Peso Argentino (ARS)".equals(moeda)) {
            valorConvertido = valorEmReais / 0.03;
        }
    }
}

```

```

        } else if ("Iene Japonês (JPY)".equals(moeda)) {
            valorConvertido = valorEmReais * 29.36;
        }

        String resultadoFormatado = String.format("Valor em Reais (R$): %.2f\nValor em %s: %.2f", valorEmReais, moeda, valorConvertido);

        JOptionPane.showMessageDialog(this, resultadoFormatado, "Resultado da Conversão", JOptionPane.INFORMATION_MESSAGE);

    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(this, "Digite apenas número válido para Valor.", "Erro de Formato", JOptionPane.ERROR_MESSAGE);
    }
} //GEN-LAST:event_jButton1ActionPerformed

/**
 * @param args the command line arguments
 */

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JComboBox<String> cbMoedaConversao;
private javax.swing.JButton jButton1;
private javax.swing.JLabel lbMoedaConversao;
private javax.swing.JLabel lbTitulo;
private javax.swing.JLabel lbValor;
private javax.swing.JTextField tfValor;
// End of variables declaration//GEN-END:variables
}

```

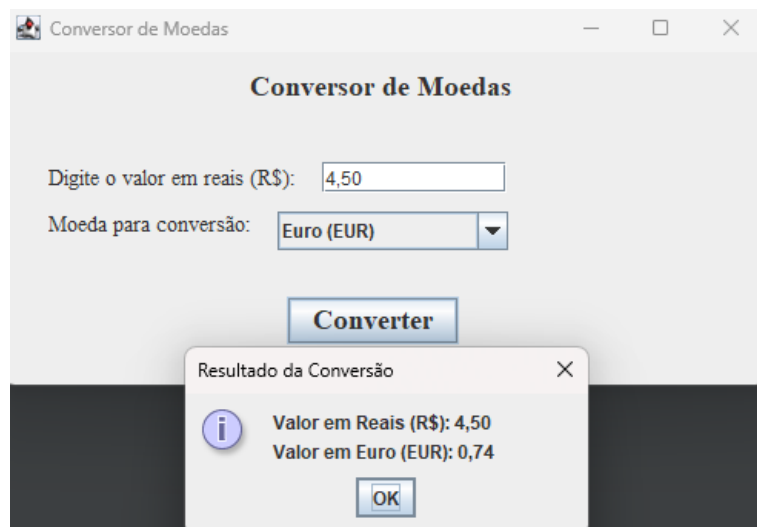
Imagens:

Figura 6. Captura de tela do Exercício 02.



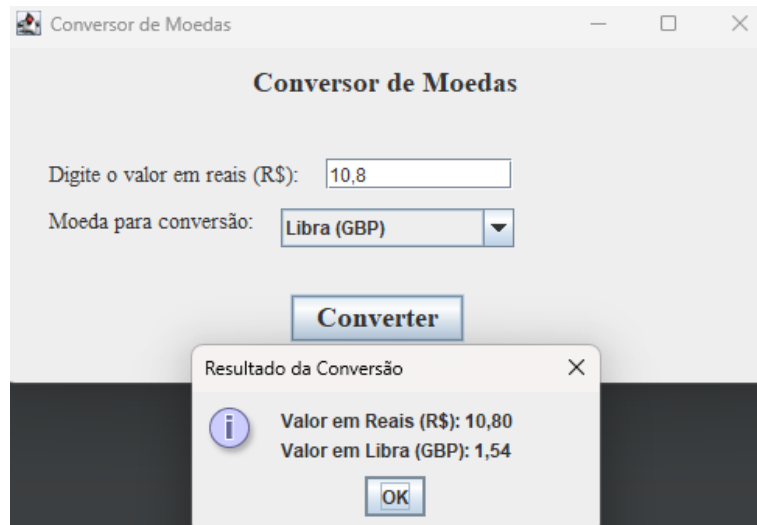
Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2025.

Figura 7. Captura de tela do Exercício 02.



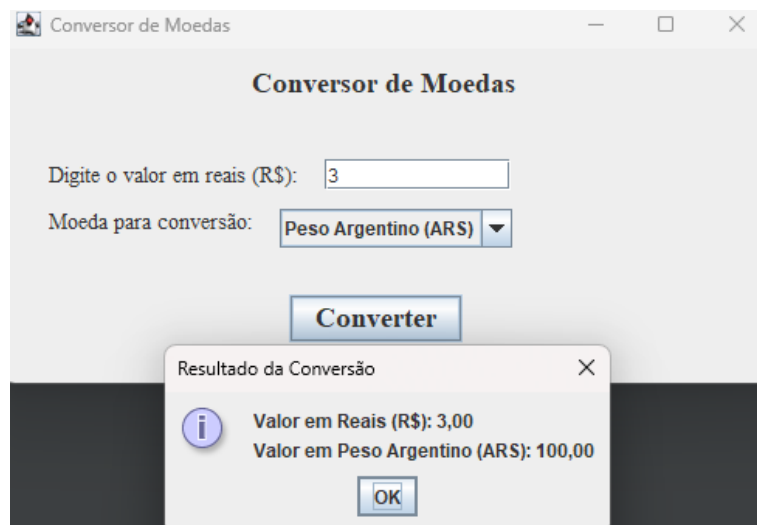
Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2025.

Figura 8. Captura de tela do Exercício 02.



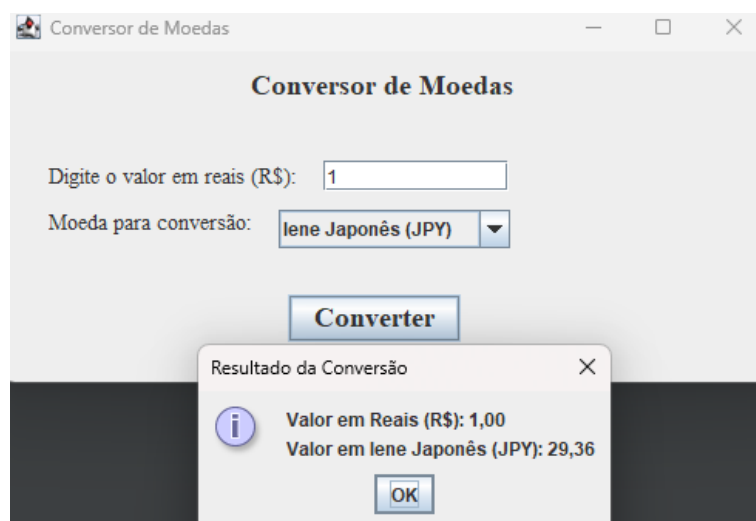
Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2025.

Figura 9. Captura de tela do Exercício 02.



Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2025.

Figura 10. Captura de tela do Exercício 02.



Fonte: Captura de tela retirada pelo autor em 14 de janeiro de 2025.