

EVENTOS BÁSICOS GUI JAVA

**PROF. ME.
HÉLIO
ESPERIDIÃO**

JOPTIONPANE

CAIXA DE MENSAGEM SIMPLES

```
JOptionPane.showInputDialog("Qual é o seu nome?");
```

SHOWINPUTDIALOG

```
String nome = null;
```

```
nome = JOptionPane.showInputDialog("Qual é o seu nome?");
```

SHOWCONFIRMDIALOG

```
int resposta =  
JOptionPane.showConfirmDialog(null, "Deseja  
realmente excluir?");
```

INTERFACE GRÁFICA X INTERFACE

- Para evitar confusões na definição dos termos, chamaremos de interface os elementos da orientação a objetos que descrevem um conjunto de métodos, e de interface gráfica os elementos como botões, janelas, campos e outros. Ao se construir uma interface gráfica usando Java Swing, acrescentamos as interfaces para definir qual método será utilizado no momento de chamada que um evento é iniciado

INTERFACE

```
public interface AcessoElementos {  
    public int getElemento (int index);  
    public void setElemento (int index);  
}
```

USO DE UMA INTERFACE

```
import javax.swing.*;

public class Aluno extends JFrame implements AcessoElementos{

    @Override
    public int getElemento(int index) {
        return 0;
    }

    @Override
    public void setElemento(int index) {
    }

}
```

TRATAMENTO DE EVENTOS

Evento	Descrição
ActionListener	Evento gerado pelo clique com o mouse em um botão.
ItemListener	Evento gerado quando um item em uma lista é selecionado.
FocusListener	Evento gerado quando um elemento ganha ou perde foco.
WindowListener	Evento gerado quando ocorre uma mudança na janela.


```
import java.awt.Container;
import javax.swing.*;
public class PrimeiraTela extends JFrame{
    private JButton btnok;
    private JTextField txtNome;
    private JLabel lblNome;
    private Container ctn;
    public PrimeiraTela() {
        setSize(300,140);
        setTitle("Eventos em Java Swing");
    }
}
```

```
ctn = getContentPane();
ctn.setLayout(null);
btnok = new JButton("Enviar");
lblNome = new JLabel("Nome");
txtNome = new JTextField();
lblNome.setBounds(10,10,100,25);
txtNome.setBounds(70,10,200,25);
btnok.setBounds(90,50,80,40);
ctn.add(lblNome);
ctn.add(txtNome);
ctn.add(btnok);
```

```
}
```

```
public static void main(String[] args) {  
    PrimeiraTela tela = new PrimeiraTela();  
}  
}
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class PrimeiraTela Tela extends JFrame implements ActionListener{
    btnok.addActionListener(this);
    @Override
    public void actionPerformed(ActionEvent e) {
        // caso seja necessário tratar eventos de mais de um botão
        if(e.getActionCommand().equals("Enviar")) {
            txtNome.setText("Botão clicado");
        }
    }
}
```

UTILIZANDO CLASSE ANÔNIMA

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Tela extends JFrame {
    public Tela() {
        btnok.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                trataBotaoOk();
            }
        });
    }
    public void trataBotaoOk() {
        txtNome.setText("Botão clicado");
    }
}
```

TRATAR EVENTO DE MUDANÇA DE TEXTO EM UM JTEXTFIELD

```
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;

txtNome.getDocument().addDocumentListener(new DocumentListener() {
    public void removeUpdate(DocumentEvent e) {
        // ações quando texto for apagado }

    public void insertUpdate(DocumentEvent e) {
        // ações quando texto for inserido
    }

    public void changedUpdate(DocumentEvent e) {
        // ações quando texto for alterado
    }

});
```

TRATAMENTO DE SELEÇÃO DE UM JCOMBOBOX

```
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

jmbTipos.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        if (e.getStateChange() == ItemEvent.SELECTED) {
            trataJmbTipos();
        }
    }
});

public void trataJmbTipos() {
    JOptionPane.showMessageDialog(this, "Item selecionado: "+ jmbTipos.
getSelectedItem());
}
```

EVENTO DE PERDA OU GANHO DE FOCO SOB UM COMPONENTE

```
txtNome.addFocusListener(new FocusListener() {  
    public void focusLost(FocusEvent e) {  
  
    }  
    public void focusGained(FocusEvent e) {  
    }  
});
```


ATIVIDADE