

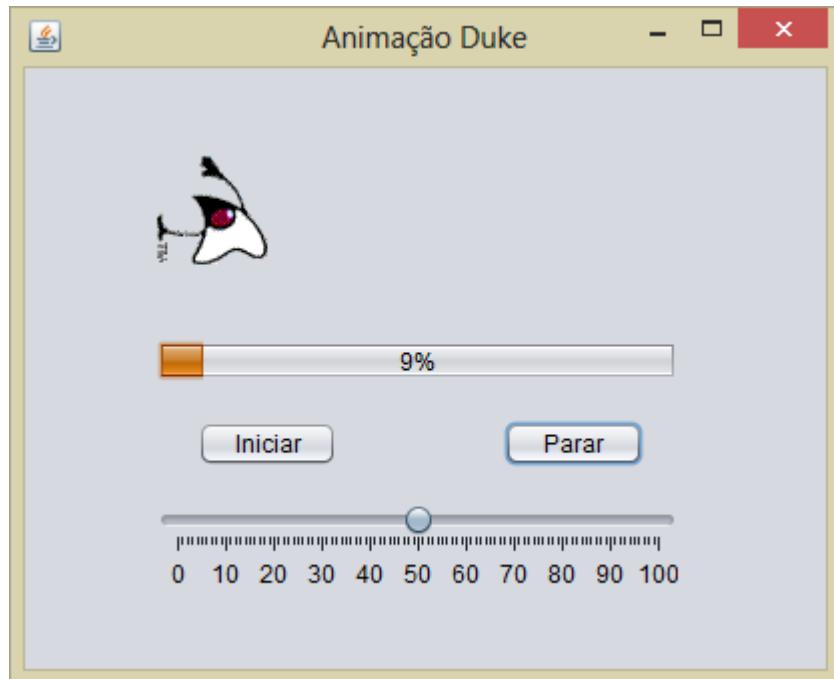
Atividade04

OBJETIVOS

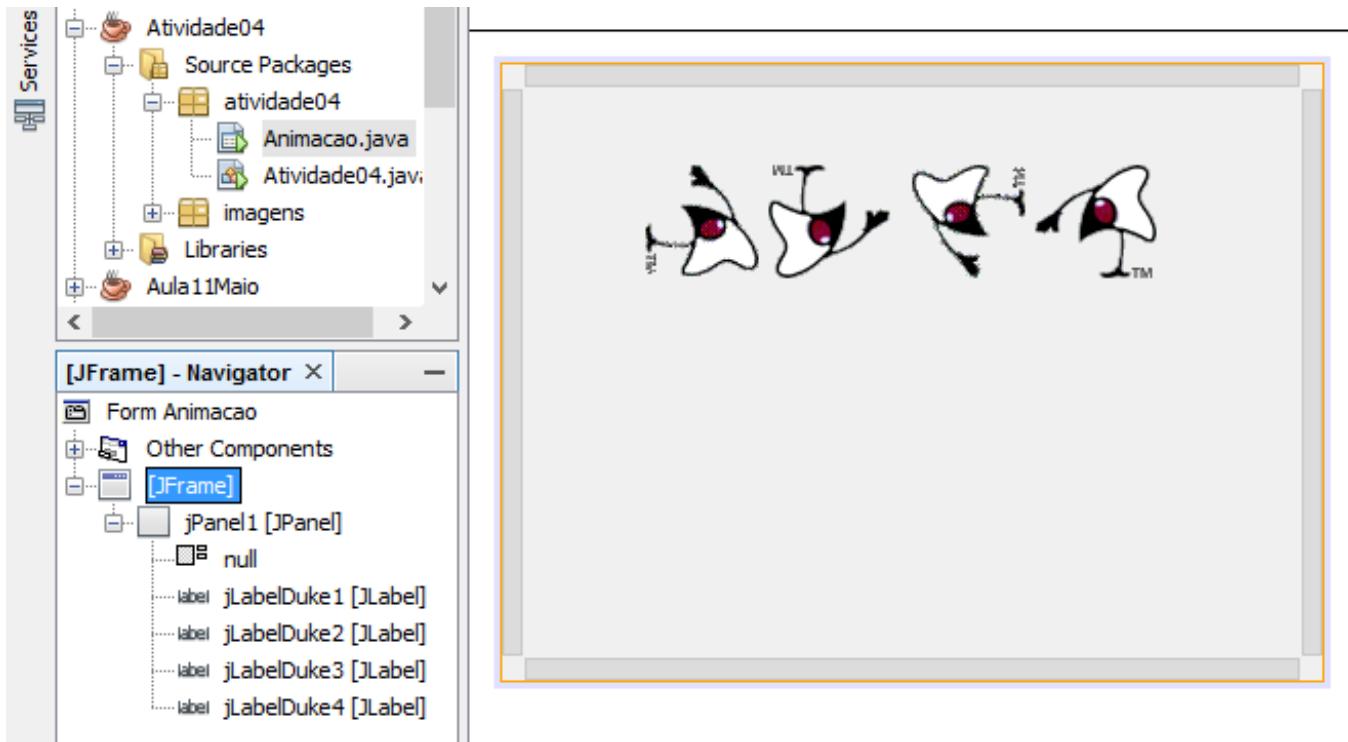
Manipulação de componentes Swing:

- JProgressBar
- JSlider
- ToolTipText
- Timer

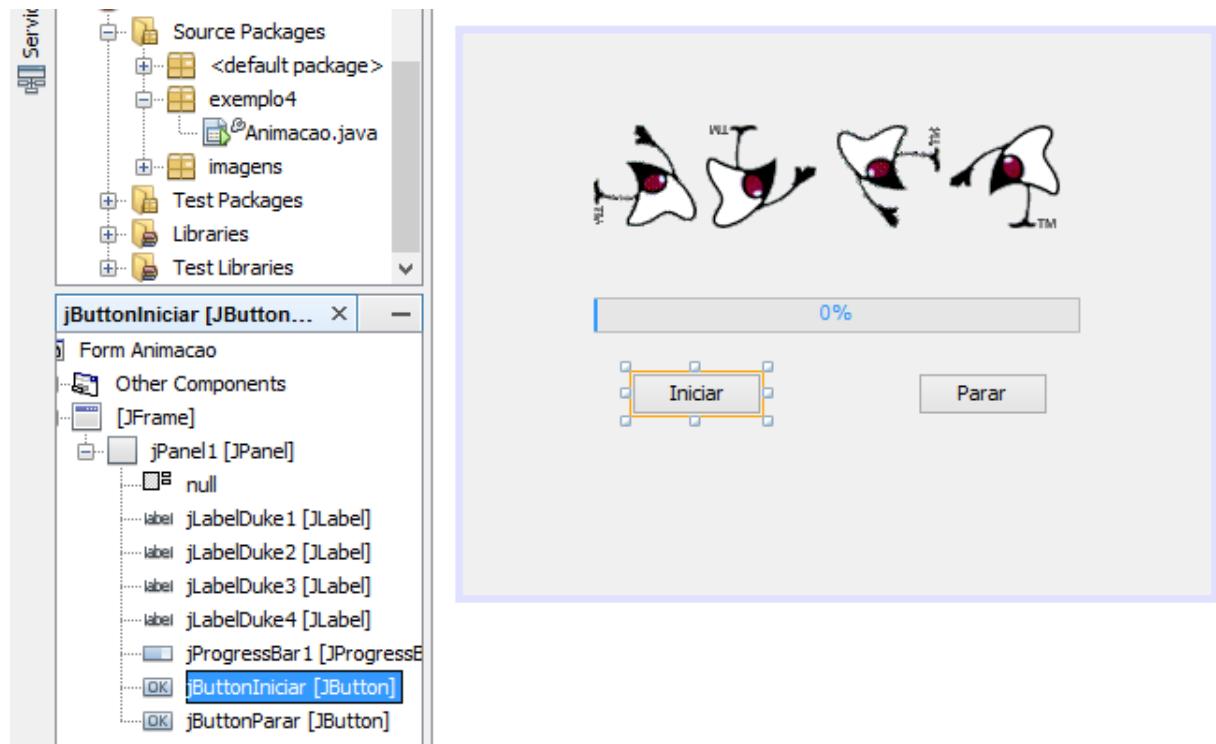
Aspetto final do projeto



1. Utilizando o IDE NetBeans, criar um novo projeto com o nome **Atividade04**.
2. Colocar a pasta das imagens dentro do projeto. Copiar a pasta com as imagens da plataforma: **imagensDuke.zip**.
3. Criar um novo JFrame Form de nome: **Animacao**.
4. Centrar a janela da aplicação JFrame no ecrã.
5. Colocar título na barra de título da aplicação: Animação Duke.
6. Colocar um JPanel a ocupar toda a área do JFrame.
7. Alterar o Layout do JPanel para null layout.
8. Colocar as imagens como na figura seguinte:



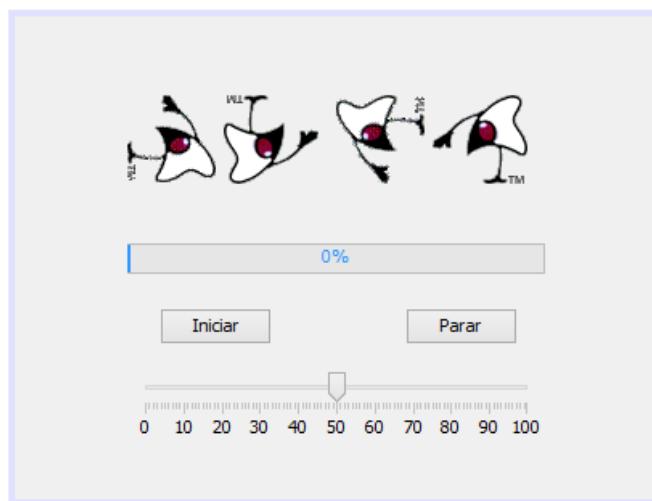
9. Alterar o nome das JLabel's para **jLabelDuke1**, **jLabelDuke2**, **jLabelDuke3**, **jLabelDuke4**, respectivamente.
10. Coloque uma **Progress Bar** por baixo das imagens.
11. Nas propriedades da Progress Bar, colocar o campo stringPainted: true.
12. Colocar dois botões como na figura e os respetivos nomes: **jButtonIniciar** e **jButtonParar**.



13. De seguida, colocar um Slider por baixo dos botões.

- 13.1. Colocar no campo majorTickSpacing: 10
- 13.2. Colocar no campo minorTickSpacing: 1
- 13.3. O campo paintLabels: true
- 13.4. O campo paintTicks: true

14. As operações realizadas devem resultar no ecrã com o aspeto seguinte:



15. Adicionar no campo ToolTipText, do botão Iniciar o texto: Clique para iniciar

ToolTipText é um balão de ajuda que é visualizado quando o cursor do rato passa por cima do swing control.

16. Adicionar o tooltipText no botão **Parar** como o texto: Clique para parar

17. Adicionar o tooltipText no Slider como o texto: Ajustar a velocidade

18. A aplicação deve iniciar com todas as imagens escondidas ou não visíveis.

Para tal deve colocar o respetivo código depois de **initComponents()**

```
public Animacao() {  
    initComponents();  
    // Colocar AQUI o código do passo 18  
}
```

19. Para implementar a animação, uma possibilidade, é a utilização de um Timer.

Timer é uma classe java que permite a execução repetida de uma tarefa.

20. Colocar a variável **timer** do Tipo: **Timer** e a variável **valor = 0** do Tipo inteiro, na área de definição de variáveis da classe *Animacao*.

21. Antes de “**public static void main(String args[])** {“ colocar o código seguinte:

```
private void criarTimer(){  
    timer = new javax.swing.Timer(1000, new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            tarefa();  
        }  
    });  
}
```

22. Depois do código anterior é necessário incluir as classes seguintes:

```
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;
```

23. Depois do método anterior coloque o seguinte código:

```
private void tarefa(){  
    jLabelDuke1.setVisible(valor % 4 == 0);  
    jLabelDuke2.setVisible(valor % 4 == 1);  
    jLabelDuke3.setVisible(valor % 4 == 2);
```

```
jLabelDuke4.setVisible(valor % 4 == 3);
valor++;
jProgressBar1.setValue(valor);
if (valor >= 100){
    valor = 0;
    timer.stop();
}
}
```

24. Depois do código anterior é necessário incluir as classes seguintes:

```
import javax.swing.Timer;
```

25. No evento associado ao botão **Iniciar** colocar o seguinte código:

```
if (timer == null){
    criarTimer();
    timer.start();
} else if (!timer.isRunning()) timer.restart();
```

26. No evento associado ao botão **Parar** colocar o seguinte código:

```
timer.stop();
```

27. Para ajustar a velocidade temos de ativar o evento **stateChanged** do swing control slider.

28. No método do evento anterior colocar e completar o seguinte código:

```
int valorSlider = jSlider1.getValue();
timer.setDelay(???);
```