

Processamento de Imagens - Aula02

Uniube
2023-2

Prof. Dr. Marcos Lopes

(34) 9 9878 0925

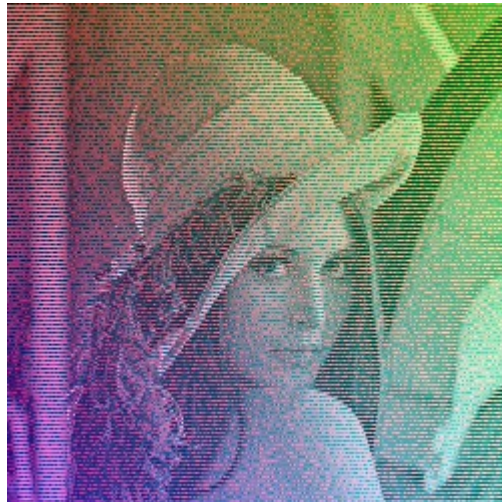
malopes21@gmail.com

<https://sites.google.com/view/malopes21/>



Primeiro exemplo prático:

Vamos montar um exemplo bem simples de manipulação de uma imagem utilizando um algoritmo bem simples pra obter um efeito “especial” na imagem carregada.













A imagem que iremos usar é bem conhecida no contexto de processamento de imagens.

Usaremos a imagem “lena_gray_256.png”



Será útil montar um projeto numa IDE: eclipse, intellij ou vscode, etc:

- ▼  AulasPDI
 - ▶  JRE System Library [JavaSE-1.8]
 - ▼  src
 - ▼  cap01
 - ▶  LeImagem01.java
 - ▶  LeImagem02.java
 - ▶  LeImagem03.java
 - ▶  LeImagem04.java
 - ▶  LeImagem05.java
 - ▶  lena_gray_256.png

Iremos utilizar Java, mas praticamente qualquer linguagem de programação dá suporte para as operações necessárias.

LeImagem01:

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class LeImagem01 {

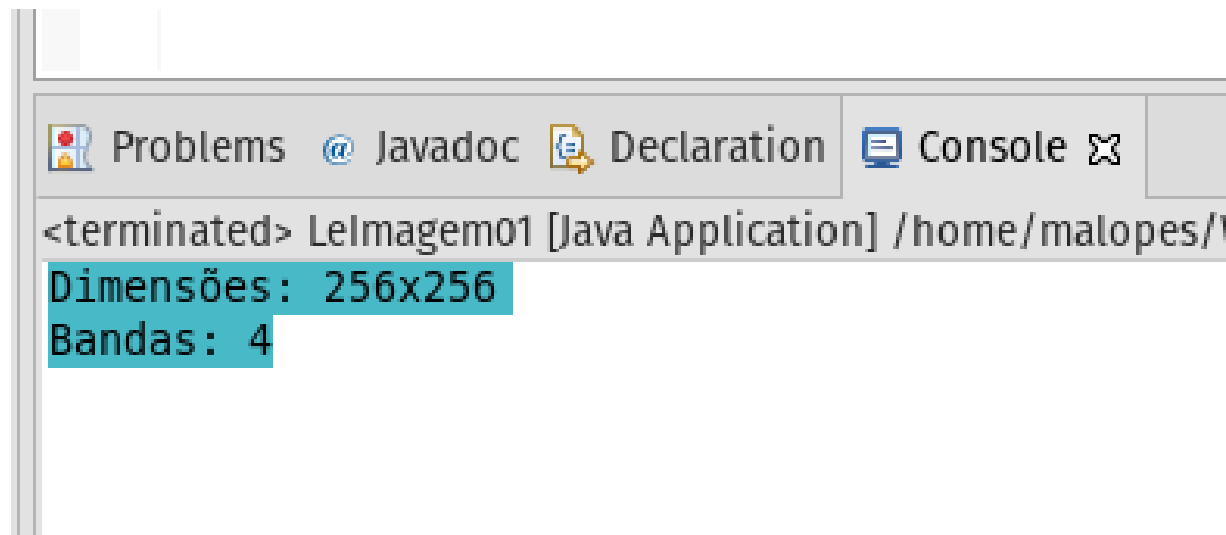
    public static void main(String[] args) throws IOException {

        String path = "lena_gray_256.png";
        File file = new File(path);
        BufferedImage imagem = ImageIO.read(file);

        String infoImagem = "Dimensões: " + imagem.getWidth()
            + "x" + imagem.getHeight() + "\nBandas: "
            + imagem.getRaster().getNumBands();
        System.out.println(infoImagem);
    }
}
```

O programa do slide anterior, apenas lê uma imagem (seu arquivo) referenciada pelo caminho relativo no projeto.

Após isso, printa na tela algumas informações sobre o arquivo da imagem, teste!



O próximo programa, na realidade não lida com imagens, apenas demonstra a criação de um Frame Java:

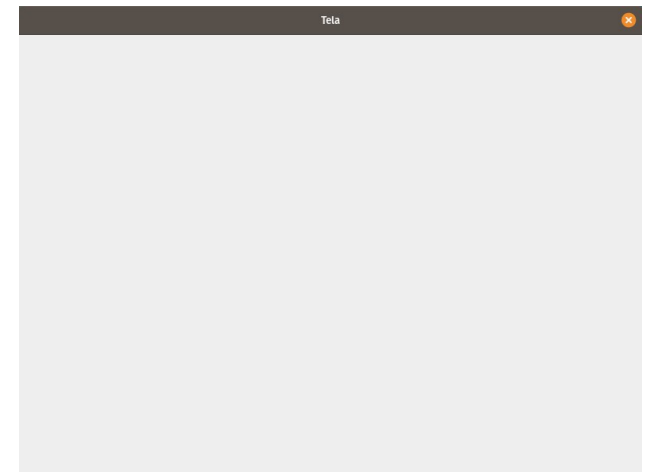
Lelimagem02:

```
import javax.swing.JFrame;

public class LeImagem02 {

    public static void main(String[] args) {

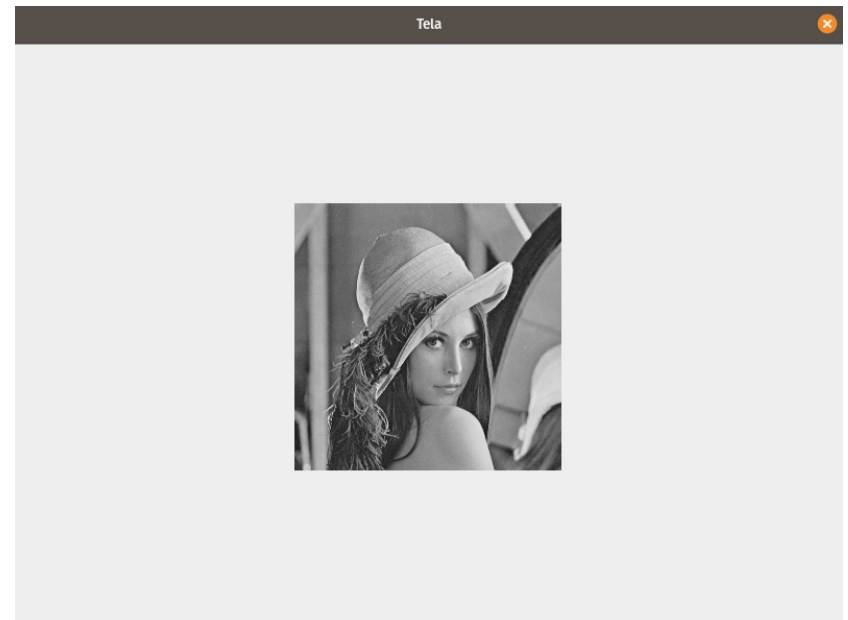
        JFrame principal = new JFrame("Tela");
        principal.setSize(800, 600);
        principal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        principal.setLocationRelativeTo(null);
        principal.setVisible(true);
    }
}
```



Lelimagem03:

Carregando uma imagem num JFrame Java:

```
import java.awt.BorderLayout;  
import java.awt.Container;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import java.io.IOException;  
  
import javax.imageio.ImageIO;  
import javax.swing.ImageIcon;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JScrollPane;
```




```
public class LeImagem03 {  
  
    public static void main(String[] args) throws IOException {  
  
        JFrame principal = new JFrame("Tela");  
        principal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        String path = "lena_gray_256.png";  
        File file = new File(path);  
        BufferedImage imagem = ImageIO.read(file);  
  
        ImageIcon icone = new ImageIcon(imagem);  
        JLabel labImagem = new JLabel(icone);  
  
        Container contentPane = principal.getContentPane();  
        contentPane.setLayout(new BorderLayout());  
        contentPane.add(new JScrollPane(labImagem), BorderLayout.CENTER);  
  
        principal.setSize(800, 600);  
        principal.setVisible(true);  
        principal.setLocationRelativeTo(null);  
    }  
}
```

Lelimagem04: ajustando melhor no JFrame

```
public class LeImagem04 {  
  
    public static void main(String[] args) throws IOException {  
  
        JFrame principal = new JFrame("Tela");  
        principal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        String path = "lena_gray_256.png";  
        File file = new File(path);  
        BufferedImage imagem = ImageIO.read(file);  
  
        ImageIcon icone = new ImageIcon(imagem);  
        JLabel labImagem = new JLabel(icone);  
  
        Container contentPane = principal.getContentPane();  
        contentPane.setLayout(new BorderLayout());  
        contentPane.add(new JScrollPane(labImagem), BorderLayout.CENTER);  
  
        String infoImagem = "Dimensões: " + imagem.getWidth() + "x" +  
        imagem.getHeight() + "Bandas: " + imagem.getRaster().getNumBands();  
        contentPane.add(new JLabel(infoImagem), BorderLayout.NORTH);  
  
        principal.setSize(imagem.getWidth() + 40, imagem.getHeight() + 100);  
        principal.setVisible(true);  
        principal.setLocationRelativeTo(null);  
    }  
}
```



Finalmente, temos:

LeImagem05:

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.Random;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.UIManager;
import javax.swing.UIManager.LookAndFeelInfo;

public class LeImagem05 implements ActionListener {

    private JButton btnAcao = null;
    private JButton btnAcao2 = null;
    private BufferedImage imagem = null;
    private JFrame principal = null;
```

```
public void run() throws IOException {

    checkNimbus();

    principal = new JFrame("Lendo uma imagem PNG");
    principal.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    String path = "lena_gray_256.png";
    File file = new File(path);
    imagem = ImageIO.read(file);

    ImageIcon icone = new ImageIcon(imagem);
    JLabel labImagem = new JLabel(icone);

    String infoImagem = "Dimensões: " + imagem.getWidth() + "x" +
    imagem.getHeight() + "Bandas: " + imagem.getRaster().getNumBands();

    Container contentPane = principal.getContentPane();
    contentPane.setLayout(new BorderLayout());
    contentPane.add(new JScrollPane(labImagem), BorderLayout.CENTER);
    contentPane.add(new JLabel(infoImagem), BorderLayout.NORTH);

    JPanel painel = new JPanel();
    btnAcao = new JButton("Efeito Especial");
    btnAcao.addActionListener(this);
    btnAcao2 = new JButton("Salvar Imagem");
    painel.add(btnAcao);
    painel.add(btnAcao2);
    contentPane.add(painel, BorderLayout.SOUTH);

    principal.setSize(imagem.getWidth() + 40, imagem.getHeight() + 100);
    principal.setVisible(true);
    principal.setLocationRelativeTo(null);

}
```

```
private static void checkNimbus() {  
    try {  
        for (LookAndFeelInfo info :  
            UIManager.getInstalledLookAndFeels()) {  
            if ("Nimbus".equals(info.getName())) {  
                UIManager.setLookAndFeel(info.getClassName());  
                break;  
            }  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, "Nimbus not available!");  
    }  
}
```

@Override

```
public void actionPerformed(ActionEvent e) {
```

```
    if(e.getSource() == btnAcao) {
```

```
        int w = imagem.getWidth();
```

```
        int h = imagem.getHeight();
```

```
        int[] pixels = imagem.getRGB(0, 0, w, h, null, 0, w);
```

```
        Random r = new Random();
```

```
        for (int col = 0; col < w; col++) {
```

```
            for (int lin = 0; lin < h; lin++) {
```

```
                if (lin % 2 == 0)
```

```
                    pixels[w * lin + col] = new Color(r.nextInt(255),  
col % 255, lin % 255).getRGB();  
            }  
        }
```

```
        imagem.setRGB(0, 0, w, h, pixels, 0, w);  
        principal.repaint();
```

```
        System.out.println("OK!");
```

```
    } else if(e.getSource() == btnAcao2) {
```

```
    }
```

```
}
```

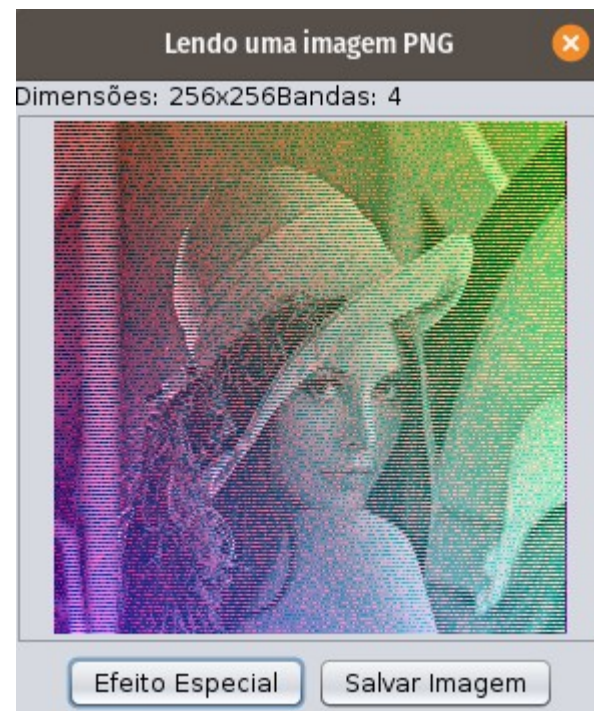
```
public static void main(String[] args) throws IOException {
```

```
    new LeImagem05().run();
```

```
}
```

```
}
```

Resultado:



Exercício:

Mudar o código que manipula os pixels da imagem, de forma que a imagem tenha seus tons de cinza com a amplitude invertida, exemplo:

