Relatório Trabalho Prático PDI - 2020

Nome: Giulia Campos
RA: 161255752

1. IMPLEMENTAÇÃO

O trabalho foi desenvolvido em Java, versão 13, com a IDE Apache NetBeans 12.0, e controle de versionamento pelo GitHub (link do repositório do projeto: https://github.com/GiuliaCampos/TrabalhoPDI).

O projeto possui 2 pacotes principais:

a. projetopdi - Contém as classes responsáveis pela manipulação das imagens pgm (MetodosCinza.java) e ppm (MetodosColorido.java e Rgb.java) alem da classe principal (ProjetoPDI.java).

b. projetopdi.ui - Contém a interface gráfica do projeto e ativação dos métodos.

Quando uma imagem é lida, seja pelo processo sequencial ou comandos em arquivos de texto, descritos mais à frente, ela é armazenada na variável 'matriz' na classe MetodosCinza, essa matriz é responsável por armazenar a imagem original e após cada transformação, a nova imagem é armazenada em 'matrizResultado' para que a imagem original não seja perdida.

Ainda na classe MetodosCinza, todos os atributos da imagem são armazenados, como tipo do arquivo, número de linhas e colunas além dos métodos disponíveis para serem executados em arquivos do tipo PGM. Os métodos disponíveis nesta classe são:

a. public MetodosCinza(String nomeArquivo) - controlador, responsável pela leitura da imagem e armazenamento, recebe o arquivo que será lido;

b. public void printMatriz() - função auxiliar na classe, que imprime no console a imagem original e resultado;

c. public void matrizResultadoNovaMatriz() - função responsável por trocar a imagem resultado para a original, no caso de mais de um processamento na imagem;

d. public void negativoMatriz() - método que transforma a imagem armazenada em negativa;

e. public void clarearMatriz(int valor) - método responsável por clarear a imagem armazenada, necessita de um valor inteiro para ser somado a imagem;

f. public void escurecerMatriz(int valor) - método responsável por escurecer a imagem armazenada, necessita de um valor para ser subtraído da imagem;

g. public void fatiamentoImagem(int a, int b, int novaTonalidade) - método responsável por realizar o fatiamento da imagem, recebe um intervalo de ‘a’ até ‘b’ e uma nova tonalidade que esse intervalo receberá;

h. public void fatiamento2Imagem(int a, int b, int novaTonalidadeIntervalo, int novaTonalidadeFora) - método responsável pelo fatiamento, recebe um intervalo de ‘a’ até ‘b’, uma nova tonalidade para o intervalo definido e uma nova tonalidade para os pixels de fora do intervalo;

i. public void transfGama(float c, float gama) - método responsável por aplicar a transformação gama na imagem, necessita de um valor para o atributo ‘c’ e ‘gama’;

j. public void flipHorizontal() - método responsável por realizar um flip horizontal na imagem armazenada;

k. public void girarMatriz(int graus) - método responsável por girar a imagem, necessita de um valor em graus (aceita apenas 90º, -90º e 180º);

l. public void histograma() - método responsável pela equalização do histograma da imagem armazenada como original;

m. public void laplaciano1() - método responsável por aplicar o filtro laplaciano, com o valor 4 no elemento central;

n. public void laplaciano2() - método responsável por aplicar o filtro laplaciano, com o valor 8 no elemento central;

o. public void somarImagens() - método que soma a imagem original com a imagem resultado;

p. public void subtrairImagens() - método que subtrai a imagem resultado da imagem original;

q. public void media(int filtro) - método que aplica o filtro da média, necessita de um valor para a dimensão da janela do filtro;

r. public void mediana(int filtro) - método que aplica o filtro da mediana, necessita de um valor para a dimensão da janela do filtro;

s. public void binarizacao(int valor) - método responsável por realizar a binarização da imagem, necessário um valor para ser o limite da binarização;

t. public void multiplicarValor(int valor) - método que multiplica um valor a imagem armazenada, é necessário um valor para ser multiplicado.

Na classe MetodosColoridos.java estão os métodos disponíveis para as imagens PPM. Quando uma imagem colorida é lida, os valores do canal RGB são armazenados seguindo o tipo RGB, disponível na classe Rgb.java. A classe Rgb.java é responsável por armazenar os valores de vermelho, verde e azul de cada pixel. Assim, cada imagem do tipo PPM, é salva na variavel ‘matriz’ da classe MetodosColoridos, cada posição dessa matriz é do tipo Rgb. Como nos

processamentos de imagens do tipo PGM, a classe armazena todos os atributos da imagem e ainda uma ‘matrizResultado’ que guarda cada processamento realizado na imagem original. Os métodos disponíveis na classe MetodosColoridos.java, são:

a. public MetodosColorido(String nomeArquivo) - método contrutor que lê a imagem e a armazena, assim como os atributos;
b. public void separarRGB() - método responsável por extrair os canais vermelho, verde e azul da imagem e armazena-los separadamente;
c. public void separaCMY() - Método responsável por extrair os canais ciano, magenta e amarelo da imagem, que antes seguia os padrões RGB, e armazenar cada canal separadamente.

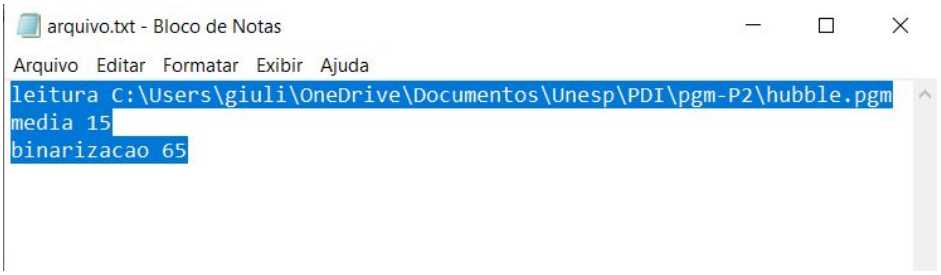
No pacote projetopdi.ui encontra-se a classe UIPrincipal.java, responsável por instanciar os métodos disponíveis para as imagens do tipo PGM e PPM, além de controlar as imagens abertas, leitura de arquivos e salvar as imagens novas após as transformações.

2. SEQUÊNCIAS DE PROCESSAMENTO

a. Descrita em arquivo texto

A aplicação permite a leitura de um arquivo de texto, do tipo ’txt’, que contém os comandos a serem aplicados na imagem. No caso de não conter a leitura de uma imagem para iniciar as transformações a aplicação indica que uma imagem deve ser lida antes, no caso de comandos de transformações com parâmetros errados, a aplicação passa para o próximo comando.

O arquivo que será lido deve conter cada comando em uma linha, seguido dos seus parâmetros, não deve haver espaços vazios no final de cada linha e no final do arquivo. Como é possível observar na imagem abaixo:



Os comandos podem ser escritos em caracteres maiúsculos e/ou minúsculos. Abaixo encontra-se uma lista dos comandos disponíveis, mas esses estão disponíveis no arquivo ‘Métodos Disponíveis em Arquivo’, enviado junto com esse relatório, lá os comandos encontram-se listados, com os parâmetros descritos e exemplos.

Método	Exemplo Comando
Leitura de arquivo	leitura C:\Users\strawberries.ppm

Escrever em arquivo	escrever C:\Users\ imagemTeste
Escurecer/Subtrair valor	escurecer 100
Clarear/Somar valor	clarear 100
Girar	girar -90
Negativo	negativo
Fatiamento - fora não há alteração	fatiamento 100 150 255
Fatiamento - fora recebe outro valor	fatiamento2 100 150 255 0
Transformação Gama	gama 0.1 1
Flip Horizontal	flip
Equalização de Histograma	histograma
Filtro Laplaciano elemento central 4	laplaciano 4
Filtro Laplaciano elemento central 8	laplaciano 8
Média	media 5
Binarização	binarizacao 90
Soma de imagem original com nova	soma
Multiplicação de um valor	multiplicacao
Subtração imagem nova e original	subtracao
Mediana	mediana 3
Extrair canais RGB	C:\Users\ testeR testeG testeB
Compor os canais RGB	C:\Users\testeR.pgm C:\Users\testeG.pgm C:\Users\testeB.pgm
Extrair canis CMY	C:\Users\ testeC testeM testeY

b. Sequencial - via interface

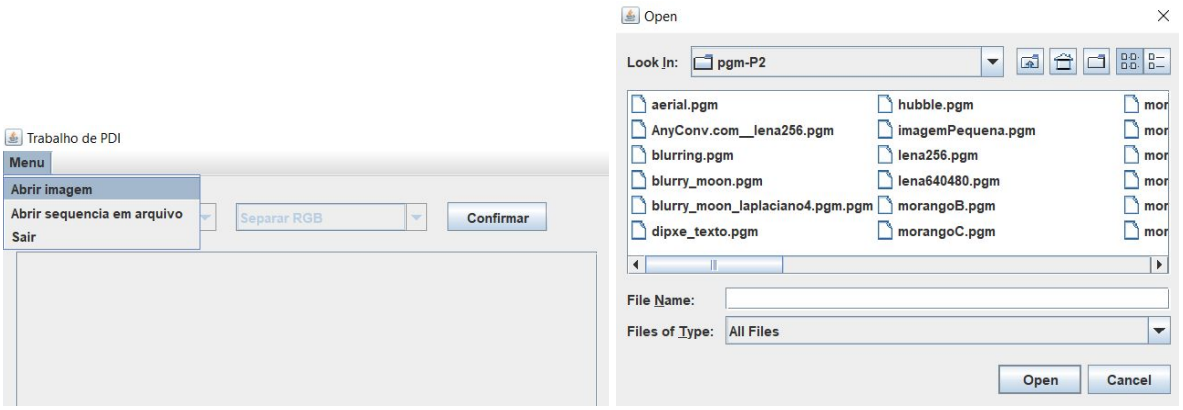
Os mesmos processamentos estão disponíveis para serem executados via interface gráfica.

3. MANIPULAÇÃO DA INTERFACE GRÁFICA

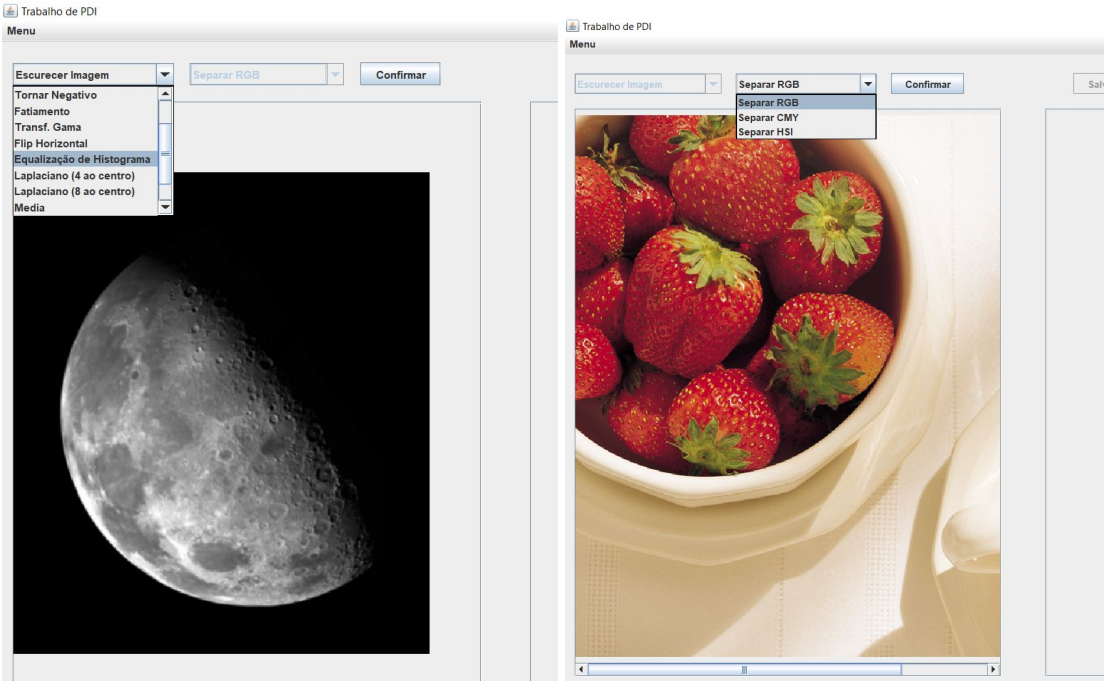
O arquivo no formato ‘Jar’, encontra-se na pasta raíz do projeto.

a. Processamento Sequencial

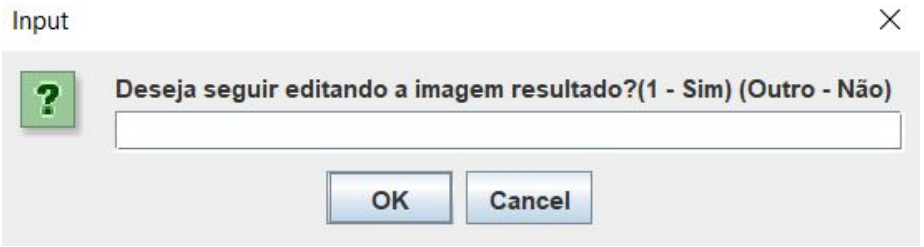
No canto superior esquerdo, é necessário selecionar no menu a opção ‘Abrir Imagem’ e selecionar no computador uma imagem do tipo PPM ou PGM para ser manipulada.

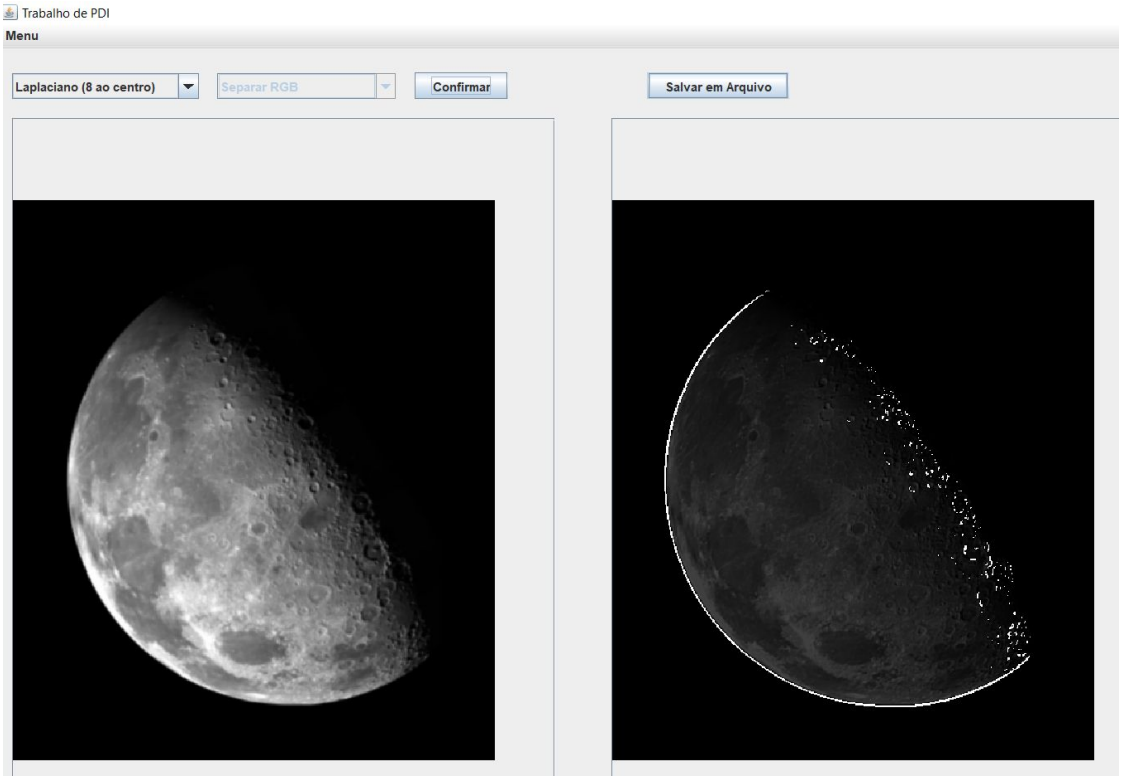


A imagem selecionada será exibida no primeiro frame, e de acordo com o tipo de imagem aberta, é possível selecionar nas opções as transformações disponíveis. Ao selecionar uma transformação, clique em ‘confirmar’.



Após cada transformação é necessário informar se deseja continuar editando a nova imagem ou a imagem nova. E também é possível selecionar o botão ‘Salvar em Arquivo’ que armazena a nova imagem em um arquivo.

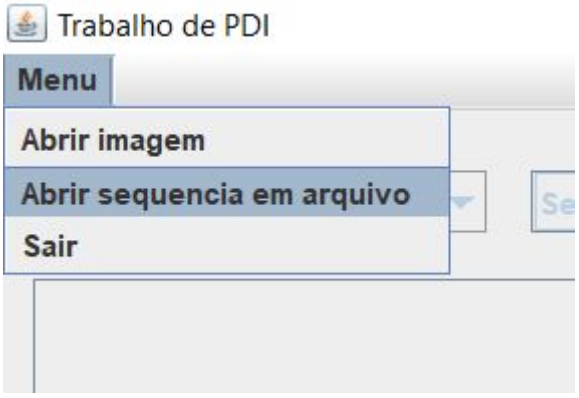




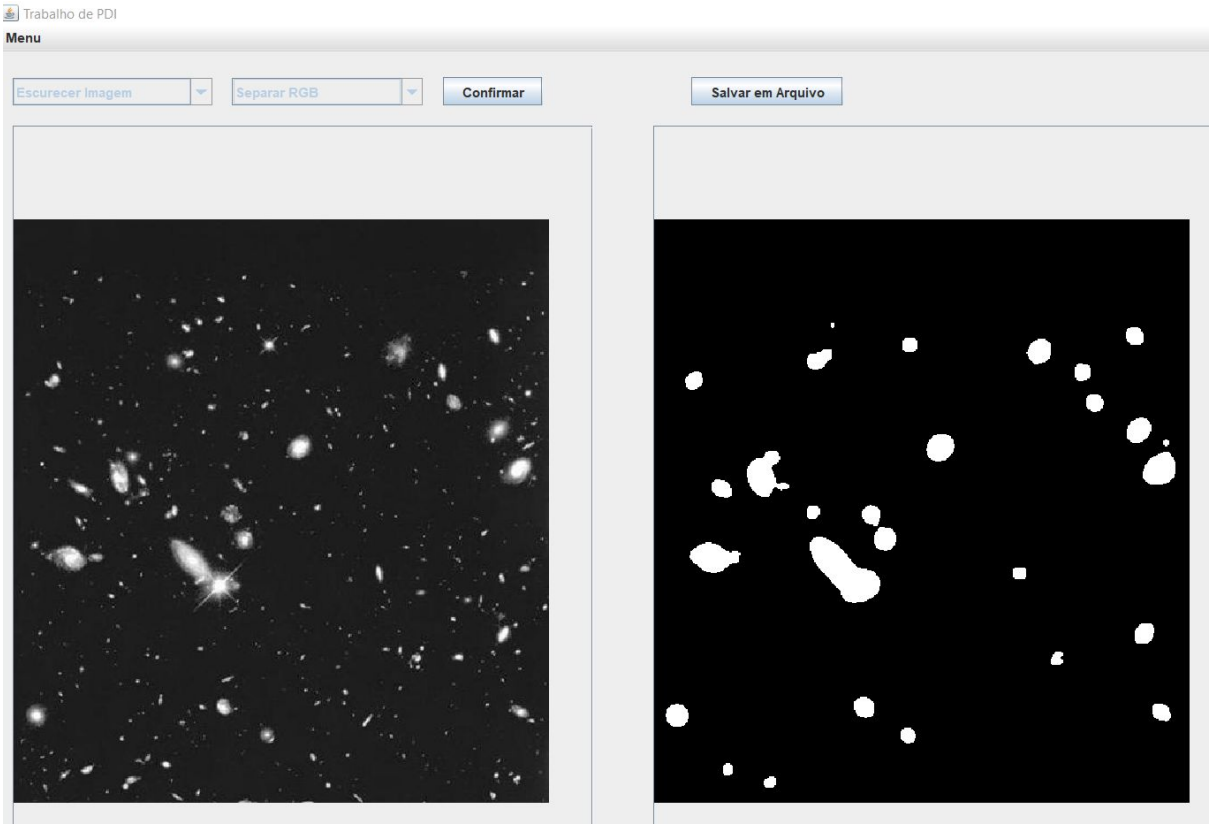
Não é necessário reiniciar a aplicação para alterar a imagem ou o tipo de processamento.

b. Processamento por Arquivo

Para o processamento por arquivo de texto, selecione no menu superior a esquerda, a opção ‘Abrir Sequência em Arquivo’, como é possível observar abaixo.



Selecione um arquivo do tipo ‘txt’ em seu computador, e o final do processamento do arquivo, assim como a imagem original serão exibidos nos frames lado a lado.



O arquivo utilizado neste exemplo de processamento possui os comandos:
leitura C:\Users\giuli\OneDrive\Documentos\Unesp\PDI\pgm-P2\hubble.pgm
media 15
binarizacao 65