Relatório Atividade 3 - PDI

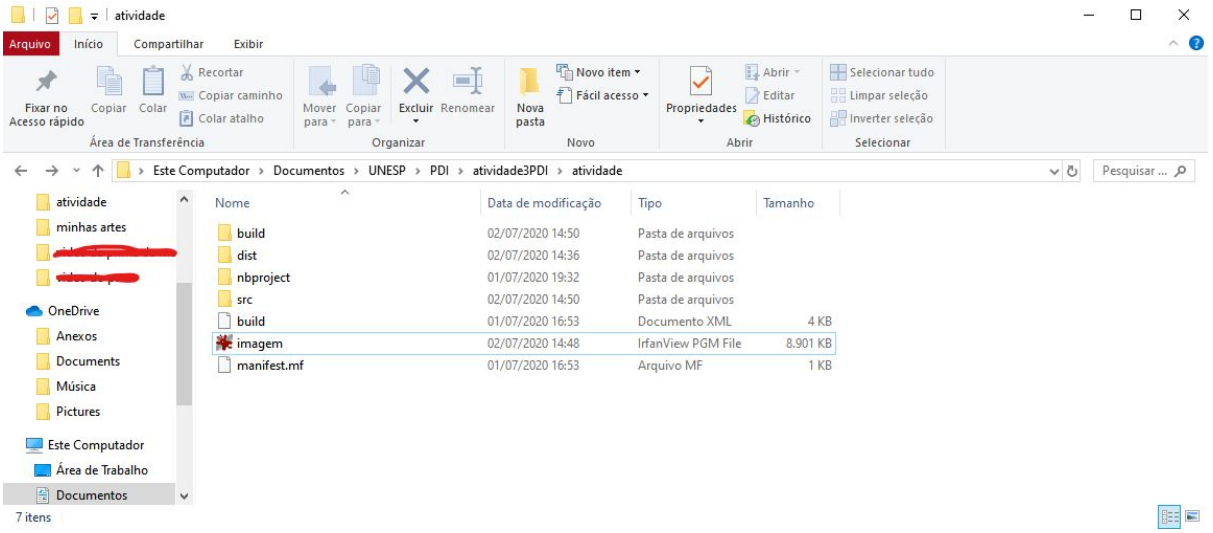
Autor: Guilherme de Aguiar Pacianotto

LINGUAGEM DE PROGRAMAÇÃO:

A linguagem de programação utilizada nesta atividade foi o Java, juntamente com a IDE NetBeans.

COMO UTILIZAR A APLICAÇÃO

Para utilizar a aplicação é simples, basta copiar na pasta do projeto a imagem PGM que deseja utilizar e renomeá-la para “imagem.pgm”, como está mostrando a captura de tela:



O próximo passo é abrir o NetBeans (ou alguma outra IDE de Java), abrir o projeto “atividade” e executar a função main.

Depois de executado a pasta irá se encontrar com outras imagens pgm nela:

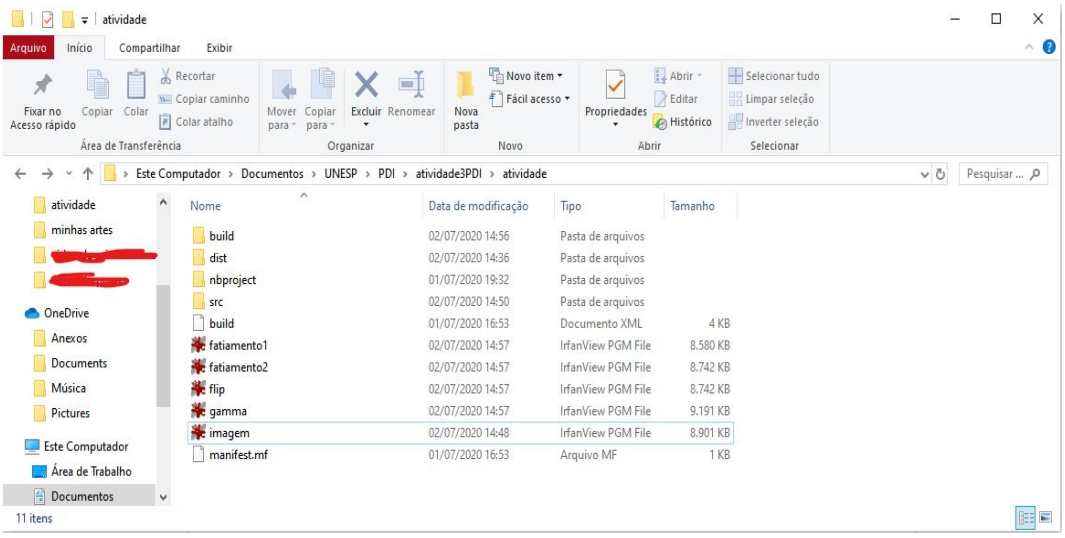


IMAGEM DE TESTE (ORIGINAL):

FATIAMENTO 1

Código fonte:

```
53      public static int[][] fatiamentoImagem(int[][] imagemOriginal){
54          int coluna, linha;
55
56
57          coluna = imagemOriginal.length;
58          linha = imagemOriginal[0].length;
59
60          //System.out.println("Colunas: "+coluna+" Linhas: "+linha);
61
62          int[][] imagemFatiada = new int[coluna][linha];
63
64
65          for(int x = 0; x < coluna; x++)
66          {
67              for(int y = 0; y < linha; y++)
68              {
69                  imagemFatiada[x][y] = imagemOriginal[x][y];
70              }
71          }
72          for(int x = 0; x < coluna; x++)
73          {
74              for(int y = 0; y < linha; y++)
75              {
76
77                  if((imagemFatiada[x][y] < 120) || (imagemFatiada[x][y] > 200))
78                  {
79                      imagemFatiada[x][y] = 10;
80                  }
81                  else
82                  {
83                      imagemFatiada[x][y] = 250;
84                  }
85
86              }
87          }
88          return imagemFatiada;
89      }
```

Resultado:

FATIAMENTO 2

Código Fonte:

```
91      public static int[][] fatiamentoImagem2(int[][] imagemOriginal){
92          int coluna, linha;
93
94          coluna = imagemOriginal.length;
95          linha = imagemOriginal[0].length;
96
97          //System.out.println("Colunas: "+coluna+" Linhas: "+linha);
98
99          int[][] imagemFatiada = new int[coluna][linha];
100
101
102         for(int x = 0; x < coluna; x++)
103         {
105             for(int y = 0; y < linha; y++)
106             {
107                 imagemFatiada[x][y] = imagemOriginal[x][y];
108             }
109         }
110         for(int x = 0; x < coluna; x++)
111         {
112             for(int y = 0; y < linha; y++)
113             {
114
115                 if((imagemFatiada[x][y] < 150) || (imagemFatiada[x][y] > 250))
116                 {
117
118                 }
119                 else
120                 {
121                     imagemFatiada[x][y] = 200;
122                 }
123             }
124         }
125
126         return imagemFatiada;
        }
```

Resultado:

GAMMA

Código Fonte:

```
128     public static int[][] transformacaoGamma(int[][] imagemOriginal, double gamma){
129         int coluna, linha;
130 
131 
132         coluna = imagemOriginal.length;
133         linha = imagemOriginal[0].length;
134 
135         //System.out.println("Colunas: "+coluna+" Linhas: "+linha);
136 
137         int[][] imagemGamma = new int[coluna][linha];
138 
139 
140         for(int x = 0; x < coluna; x++)
141         {
143             for(int y = 0; y < linha; y++)
144             {
145                 imagemGamma[x][y] = imagemOriginal[x][y];
146             }
147         }
148 
149         for(int x = 0; x < coluna; x++)
150         {
151             for(int y = 0; y < linha; y++)
152             {
153 
154                 imagemGamma[x][y] = (int) Math.pow((imagemOriginal[x][y]), gamma);
155                 if(imagemGamma[x][y] > 255)
156                 {
157                     imagemGamma[x][y] = 255;
158                 }
159 
160             }
161         }
162 
163         return imagemGamma;
164     }
```

Resultado com gamma valendo 1,125:

O gamma pode ser mudado se “descomentar” as linhas 240 e 242 no código fonte na função “MAIN”.

FLIPPING

Nessa parte da atividade encontrei certa dificuldade pois aparentemente as linhas e as colunas da matriz não correspondem exatamente às linhas e colunas da imagem. Fiz um teste com outra imagem que “comprova” a minha “hipótese”.

Teste:

```
for(int x = 0; x < linha; x++)
{
    for(int y = 0; y < coluna; y++)
    {
        //imagemFlip[y][x] = imagemOriginal[y][linha - x-1];

        if(y == 125)
        {
            imagemFlip[y][x] = 0;

        }
        else
        {
            imagemFlip[y][x] = imagemOriginal[y][x];
        }
    }

}
```

O resultado do teste criou um “risco” preto na imagem, que, teoricamente, deveria ser uma linha inteira da imagem, mas pegou apenas uma parte.

Tendo isso em vista, elaborei um código de flipping que, **teoricamente** funcionaria, porém na prática ele não funciona. Inclusive era uma dúvida que gostaria de tirar posteriormente.

Código fonte:

```
165     public static int[][] flipping(int[][] imagemOriginal){
166         int coluna, linha;
167         int auxLinha, auxColuna;
168         int z;
169
170
171         coluna = imagemOriginal.length ;
172         linha = imagemOriginal[0].length;
173
174         auxLinha = 0;
175         auxColuna = coluna - 1;
176
177         int aux[] = new int[coluna];
178
179         int[][] imagemFlip = new int[coluna][linha];
180
181         for(int x = 0; x < linha; x++)
182         {
183             for(int y = 0; y < coluna; y++)
184             {
185
186                 imagemFlip[y][x] = imagemOriginal[coluna - y - 1][x];
187
188             }
189
190         }
191
192         return imagemFlip;
193     }
```

Resultado: