

```

1  using Mundo;
2  using Brasil.Unip.Anchieta;
3  using Brasil.Unip.Anchieta.Alunos;
4
5  namespace EqualizadorDelImagem.CapaTrabalho
6  {
7      public class IdentificacaoAluno
8      {
9          public EnMateria Materia = Materia.PI;
10         public string Turma = "CC6P39";
11         public string Nome = "Nataly Noria Karol Sousa Goncalves";
12         public string RA = "N3336A7";
13         public string LinkGitHub = "https://github.com/Nanokaso/EqualizadorImagem2020";
14     }
15 }
16
17 using System;
18 using System.Collections.Generic;
19 using System.Linq;
20 using System.Text;
21 using System.Threading.Tasks;
22 using System.Windows.Forms;
23
24 namespace EqualizadorDelImagem
25 {
26     public class ExercicioProfessor
27     {
28         public static void MainFake()
29         {
30             var resultado = Equalizar(new List<int>()
31             { 0, 0, 4, 6, 8, 8, 4, 7, 8, 9, 9, 4, 3, 2, 3, 8, 2, 2, 1, 0});
32         }
33
34         public static List<int> Equalizar(List<int> valores)
35         {
36             var N = valores.Count;
37
38             var L = 0;
39
40             List<int> contagem = new List<int>();
41             for (int i = 0; i < valores.Count; i++)
42             {
43                 if (!contagem.Contains(valores[i]))
44                     contagem.Add(valores[i]);
45
46                 if (contagem.Count > 0)
47                     L = contagem.OrderByDescending(x => x).First() + 1;
48             }
49
50             List<int> Hk = new List<int>();
51             for (int i = 0; i < 256; i++) { Hk.Add(0); }
52
53             for (int i = 0; i < valores.Count; i++)
54             {
55                 for (int lugark = 0; lugark < Hk.Count; lugark++)
56                 {
57                     if (valores[i] == lugark)
58                     {
59                         Hk[lugark] = Hk[lugark] + 1;
60                     }
61                 }
62             }
63
64             List<int> HaK = new List<int>();
65             for (int i = 0; i < 256; i++) { HaK.Add(0); }
66

```

```

67     for (int lugark = 0; lugark < HaK.Count; lugark++)
68     {
69         if (lugark == 0)
70             HaK[0] = Hk[0];
71
72         if (lugark > 0)
73             HaK[lugark] = HaK[lugark - 1] + Hk[lugark];
74     }
75
76     List<decimal> Pk = new List<decimal>();
77     for (int i = 0; i < 256; i++) { Pk.Add(0); }
78
79     for (int i = 0; i < Pk.Count; i++)
80     {
81         int valorlugar = HaK[i];
82         decimal valorzinho = Convert.ToDecimal(valorlugar) / Convert.ToDecimal(N);
83         Pk[i] = valorzinho;
84     }
85
86     List<int> k2 = new List<int>();
87     for (int i = 0; i < 256; i++) { k2.Add(0); }
88
89     for (int i = 0; i < k2.Count; i++)
90     {
91         int aa = (L - 1);
92         var resultadoMulti = aa * Pk[i];
93         var item = Math.Round(resultadoMulti, MidpointRounding.ToEven);
94         k2[i] = Convert.ToInt32(item);
95     }
96
97     List<int> g = new List<int>();
98     for (int i = 0; i < valores.Count; i++) { g.Add(0); }
99
100    for (int i = 0; i < valores.Count; i++)
101    {
102        for (int ij = 0; ij < k2.Count; ij++)
103        {
104            int item = valores[i];
105            if (item == ij)
106            {
107                g[i] = k2[ij];
108            }
109        }
110    }
111
112    return g;
113 }
114 }
115 }
116
117 using System;
118 using System.Collections.Generic;
119 using System.Diagnostics;
120 using System.Drawing;
121 using System.IO;
122 using System.Linq;
123 using System.Text;
124 using System.Threading.Tasks;
125 using System.Windows.Forms;
126
127 namespace EqualizadorDeImagem
128 {
129     public class ExercicioNataly
130     {
131         public static void Equalizar(string caminholImagem, bool comCor = false)
132         {
133             Bitmap imagem = new Bitmap(caminholImagem);

```

```

134 Bitmap imagemEqualizada = new Bitmap(caminhoImagem);
135
136 List<ImagemModel> imagemObj = new List<ImagemModel>();
137
138 int contador = 0;
139 for (int i = 0; i < imagem.Width; i++)
140 {
141     for (int j = 0; j < imagem.Height; j++)
142     {
143         ImagemModel a = new ImagemModel();
144         Color pixel = imagem.GetPixel(i, j);
145
146         a.corOriginal = pixel;
147         a.identificador = contador;
148         if (comCor)
149         {
150             a.K = Utils.CorGetLuminosidade(pixel);
151             a.KAzul = pixel.B;
152             a.KVerde = pixel.G;
153             a.KVermelho = pixel.R;
154         }
155         else
156         {
157             a.K = Utils.CorGetLuminosidade(pixel);
158         }
159
160         a.localizacaoPixel = new Point(i, j);
161         imagemObj.Add(a);
162         contador++;
163     }
164 }
165
166 var listinha = imagemObj.OrderBy(x => x.identificador).ToList();
167 List<int> K = new List<int>();
168 List<int> KAzul = new List<int>();
169 List<int> KVerde = new List<int>();
170 List<int> KVermelho = new List<int>();
171 if (comCor)
172 {
173     KAzul = ExercicoProfessor.Equalizar(listinha.Select(x => x.KAzul).ToList());
174     KVerde = ExercicoProfessor.Equalizar(listinha.Select(x => x.KVerde).ToList());
175     KVermelho = ExercicoProfessor.Equalizar(listinha.Select(x => x.KVermelho).ToList());
176 }
177 else
178 {
179     K = ExercicoProfessor.Equalizar(listinha.Select(x => x.K).ToList());
180 }
181
182 foreach (ImagemModel item in listinha)
183 {
184     Color cor = new Color();
185     var i = item.identificador;
186     if (comCor)
187         cor = Color.FromArgb(KVermelho[i], KVerde[i], KAzul[i]);
188     else
189         cor = Color.FromArgb(K[i], K[i], K[i]);
190
191     imagemEqualizada.SetPixel(item.localizacaoPixel.X, item.localizacaoPixel.Y, cor);
192 }
193
194 SaveFileDialog save = new SaveFileDialog();
195 save.FileName = caminhoImagem.Replace(".jpg", "") + Guid.NewGuid().ToString() + ".jpg";
196 save.DefaultExt = Path.GetExtension(caminhoImagem);
197 if (save.ShowDialog() == DialogResult.OK)
198 {
199     imagemEqualizada.Save(save.FileName);
200     Process.Start(Path.GetDirectoryPath(save.FileName));

```

```
201     }
202 }
203 }
204 }
205 using System;
206 using System.Collections.Generic;
207 using System.ComponentModel;
208 using System.Data;
209 using System.Diagnostics;
210 using System.Drawing;
211 using System.IO;
212 using System.Linq;
213 using System.Text;
214 using System.Threading.Tasks;
215 using System.Windows.Forms;
216
217 namespace EqualizadorDeImagem
218 {
219     public partial class Form1 : Form
220     {
221         const int tamanhoArray = 20;
222
223         public Form1()
224         {
225             InitializeComponent();
226         }
227         private void Form1_Load(object sender, EventArgs e)
228         {
229             this.FormBorderStyle = FormBorderStyle.FixedSingle;
230             ShowHidePainel(0);
231             btnVetorResetar_Click(sender, e);
232         }
233
234         #region Metodo Designs
235
236         private void btnImagem_Click(object sender, EventArgs e)
237         {
238             ShowHidePainel(1);
239             btnVetorResetar_Click(sender, e);
240         }
241
242         private void btnVetor_Click(object sender, EventArgs e)
243         {
244             ShowHidePainel(2);
245             btnVetorResetar_Click(sender, e);
246         }
247
248         public void ShowHidePainel(int tipo = 0)
249         {
250             pnlImagem.Dock = DockStyle.Fill;
251             pnlVetor.Dock = DockStyle.Fill;
252
253             pnlImagem.Visible = false;
254             pnlVetor.Visible = false;
255
256             // imagem
257             if (tipo == 1)
258                 pnlImagem.Visible = true;
259
260             // vetor
261             if (tipo == 2)
262                 pnlVetor.Visible = true;
263         }
264         #endregion
265
266         #region Vetor
267         private void btnVetorResetar_Click(object sender, EventArgs e)
```

```

268 {
269     int quantidade = tamanhoArray;
270
271     Random random = new Random();
272
273     List<int> lsitadenumeros = new List<int>();
274     for (int i = 0; i < quantidade; i++)
275         lsitadenumeros.Add(0);
276
277     txtVetorEntrada.Text = String.Join("-", lsitadenumeros);
278     txtVetorSaida.Text = "";
279
280 }
281 private void btnVetorAutoPreencher_Click(object sender, EventArgs e)
282 {
283     int quantidade = tamanhoArray;
284
285     Random random = new Random();
286
287     List<int> lsitadenumeros = new List<int>();
288     for (int i = 0; i < quantidade; i++)
289         lsitadenumeros.Add(random.Next(0, 255));
290
291     txtVetorEntrada.Text = String.Join("-", lsitadenumeros);
292 }
293 private void btnVetorConfirmar_Click(object sender, EventArgs e)
294 {
295     try
296     {
297         string valor = txtVetorEntrada.Text;
298         if (!Utils.TextoContemSomenteNumeros(valor, new List<string>() { "-", " " }))
299             throw new Exception("Texto inválido ou incorreto! Verifique os valores novamente!");
300         List<int> valores = valor.Split('-').Select(x => Convert.ToInt32(x)).ToList();
301         List<int> valoresEqualizados = ExercicioProfessor.Equalizar(valores);
302         txtVetorSaida.Text = String.Join("-", valoresEqualizados);
303     }
304     catch (Exception ex)
305     {
306         MessageBox.Show("Não foi possível equalizar: " + ex.Message);
307     }
308 }
309
310 #endregion
311
312 #region Imagem
313 private void btnEscolherImagem_Click(object sender, EventArgs e)
314 {
315     try
316     {
317         OpenFileDialog open = new OpenFileDialog();
318         open.Filter = "Arquivos de Imagem (*.jpg) | *.jpg";
319
320         if (open.ShowDialog() == DialogResult.OK)
321         {
322             string imagemCaminho = open.FileName;
323             ExercicioNataly.Equalizar(imagemCaminho);
324         }
325     }
326     catch (Exception ex)
327     {
328         MessageBox.Show("Não foi possível equalizar: " + ex.Message);
329     }
330 }
331 #endregion
332
333 private void btnExemploProf_Click(object sender, EventArgs e)
334 {

```

```

335         var listinha = new List<int>()
336         { 0, 0, 4, 6, 8, 8, 4, 7, 8, 9, 9, 4, 3, 2, 3, 8, 2, 2, 1, 0};
337
338         txtVetorEntrada.Text = String.Join("-", listinha);
339         txtVetorSaida.Text = "";
340     }
341 }
342 }
343
344 using System;
345 using System.Collections.Generic;
346 using System.Drawing;
347 using System.Linq;
348 using System.Text;
349 using System.Threading.Tasks;
350
351 namespace EqualizadorDeImagem
352 {
353     public class ImagemModel
354     {
355         public int identificador { get; set; }
356         public Point localizacaoPixel { get; set; }
357         public Color corOriginal { get; set; }
358         public int K { get; set; }
359         public int KVermelho { get; set; }
360         public int KAzul { get; set; }
361         public int KVerde { get; set; }
362     }
363 }
364 }
365
366 using System;
367 using System.Collections.Generic;
368 using System.Drawing;
369 using System.Linq;
370 using System.Text;
371 using System.Threading.Tasks;
372
373 namespace EqualizadorDeImagem
374 {
375     public class Utils
376     {
377         public static bool TextoContemSomenteNumeros(string text, List<string> ignore = null)
378         {
379             if (text == null)
380                 return false;
381
382             if (ignore != null && ignore.Count > 0)
383             {
384                 foreach (var a in ignore)
385                 {
386                     text = text.Replace(a, "");
387                 }
388             }
389
390             List<int> valuesnumber = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
391
392             foreach (var a in valuesnumber)
393             {
394                 text = text.Replace(a.ToString(), "");
395             }
396
397             return string.IsNullOrWhiteSpace(text);
398         }
399         public static int CorGetLuminosidade(Color cor)
400         {
401             return Convert.ToInt32(0.299 * cor.R + 0.587 * cor.G + 0.114 * cor.B);

```

```
402     }  
403   }  
404 }
```