

```
1 using Mundo;
2 using Brasil.Unip.Anchieta;
3 using Brasil.Unip.Anchieta.Alunos;
4
5 namespace 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    }
14 }
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Windows.Forms;
7
8  namespace EqualizadorDelmagem
9  {
10     public class ExercicioProfessor
11     {
12         public static void MainFake()
13         {
14             var resultado = Equalizar(new List<int>()
15             { 0, 0, 4, 6, 8, 8, 4, 7, 8, 9, 9, 4, 3, 2, 3, 8, 2, 2, 1, 0});
16         }
17
18         public static List<int> Equalizar(List<int> valores)
19         {
20             var N = valores.Count;
21
22             var L = 0;
23
24             List<int> contagem = new List<int>();
25             for (int i = 0; i < valores.Count; i++)
26             {
27                 if (!contagem.Contains(valores[i]))
28                     contagem.Add(valores[i]);
29
30                 if (contagem.Count > 0)
31                     L = contagem.OrderByDescending(x => x).First() + 1;
32             }
33
34             List<int> Hk = new List<int>();
35             for (int i = 0; i < 256; i++) { Hk.Add(0); }
36
37             for (int i = 0; i < valores.Count; i++)
38             {
39                 for (int lugark = 0; lugark < Hk.Count; lugark++)
40                 {
41                     if (valores[i] == lugark)
42                     {
43                         Hk[lugark] = Hk[lugark] + 1;
44                     }
45                 }
46             }
47
48             List<int> HaK = new List<int>();
49             for (int i = 0; i < 256; i++) { HaK.Add(0); }
50
51             for (int lugark = 0; lugark < HaK.Count; lugark++)
52             {
53                 if (lugark == 0)
54                     HaK[0] = Hk[0];
55
56                 if (lugark > 0)
57                     HaK[lugark] = HaK[lugark - 1] + Hk[lugark];
58             }
59
60             List<decimal> Pk = new List<decimal>();
61             for (int i = 0; i < 256; i++) { Pk.Add(0); }
62
63             for (int i = 0; i < Pk.Count; i++)
64             {
65                 int valorlugar = HaK[i];
66                 decimal valorzinho = Convert.ToDecimal(valorlugar) / Convert.ToDecimal(N);

```

```
67     Pk[i] = valorzinho;
68 }
69
70 List<int> k2 = new List<int>();
71 for (int i = 0; i < 256; i++) { k2.Add(0); }
72
73 for (int i = 0; i < k2.Count; i++)
74 {
75     int aa = (L - 1);
76     var resultadoMulti = aa * Pk[i];
77     var item = Math.Round(resultadoMulti, MidpointRounding.ToEven);
78     k2[i] = Convert.ToInt32(item);
79 }
80
81 List<int> g = new List<int>();
82 for (int i = 0; i < valores.Count; i++) { g.Add(0); }
83
84 for (int i = 0; i < valores.Count; i++)
85 {
86     for (int ij = 0; ij < k2.Count; ij++)
87     {
88         int item = valores[i];
89         if (item == ij)
90         {
91             g[i] = k2[ij];
92         }
93     }
94 }
95
96 return g;
97 }
98 }
99 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Diagnostics;
6 using System.Drawing;
7 using System.IO;
8 using System.Linq;
9 using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12
13 namespace EqualizadorDeImagem
14 {
15     public partial class Form1 : Form
16     {
17         const int tamanhoArray = 20;
18
19         public Form1()
20         {
21             InitializeComponent();
22         }
23         private void Form1_Load(object sender, EventArgs e)
24         {
25             this.FormBorderStyle = FormBorderStyle.FixedSingle;
26             ShowHidePainel(0);
27             btnVetorResetar_Click(sender, e);
28         }
29
30         #region Metodo Designs
31
32         private void btnImagem_Click(object sender, EventArgs e)
33         {
34             ShowHidePainel(1);
35             btnVetorResetar_Click(sender, e);
36         }
37
38         private void btnVetor_Click(object sender, EventArgs e)
39         {
40             ShowHidePainel(2);
41             btnVetorResetar_Click(sender, e);
42         }
43
44         public void ShowHidePainel(int tipo = 0)
45         {
46             pnlImagem.Dock = DockStyle.Fill;
47             pnlVetor.Dock = DockStyle.Fill;
48
49             pnlImagem.Visible = false;
50             pnlVetor.Visible = false;
51
52             // imagem
53             if (tipo == 1)
54                 pnlImagem.Visible = true;
55
56             // vetor
57             if (tipo == 2)
58                 pnlVetor.Visible = true;
59         }
60         #endregion
61
62         #region Vetor
63         private void btnVetorResetar_Click(object sender, EventArgs e)
64         {
65             int quantidade = tamanhoArray;
66
```

```

67 Random random = new Random();
68
69 List<int> lsitadenumeros = new List<int>();
70 for (int i = 0; i < quantidade; i++)
71     lsitadenumeros.Add(0);
72
73 txtVetorEntrada.Text = String.Join("-", lsitadenumeros);
74 txtVetorSaida.Text = "";
75
76 }
77 private void btnVetorAutoPreencher_Click(object sender, EventArgs e)
78 {
79     int quantidade = tamanhoArray;
80
81     Random random = new Random();
82
83     List<int> lsitadenumeros = new List<int>();
84     for (int i = 0; i < quantidade; i++)
85         lsitadenumeros.Add(random.Next(0, 255));
86
87     txtVetorEntrada.Text = String.Join("-", lsitadenumeros);
88 }
89 private void btnVetorConfirmar_Click(object sender, EventArgs e)
90 {
91     try
92     {
93         string valor = txtVetorEntrada.Text;
94         if (!Utils.TextoContemSomenteNumeros(valor, new List<string>() { "-", " " }))
95             throw new Exception("Texto inválido ou incorreto! Verifique os valores novamente!");
96         List<int> valores = valor.Split('-').Select(x => Convert.ToInt32(x)).ToList();
97         List<int> valoresEqualizados = ExercicioProfessor.Equalizar(valores);
98         txtVetorSaida.Text = String.Join("-", valoresEqualizados);
99     }
100     catch (Exception ex)
101     {
102         MessageBox.Show("Não foi possível equalizar: " + ex.Message);
103     }
104 }
105
106 #endregion
107
108 #region Imagem
109 private void btnEscolherImagem_Click(object sender, EventArgs e)
110 {
111     try
112     {
113         OpenFileDialog open = new OpenFileDialog();
114         open.Filter = "Arquivos de Imagem (*.jpg) | *.jpg";
115
116         if (open.ShowDialog() == DialogResult.OK)
117         {
118             string imagemCaminho = open.FileName;
119             ExercicioNataly.Equalizar(imagemCaminho);
120         }
121     }
122     catch (Exception ex)
123     {
124         MessageBox.Show("Não foi possível equalizar: " + ex.Message);
125     }
126 }
127 #endregion
128
129 private void btnExemploProf_Click(object sender, EventArgs e)
130 {
131     var listinha = new List<int>()
132     { 0, 0, 4, 6, 8, 8, 4, 7, 8, 9, 9, 4, 3, 2, 3, 8, 2, 2, 1, 0 };
133

```

```
134         txtVetorEntrada.Text = String.Join("-", listinha);
135         txtVetorSaida.Text = "";
136     }
137 }
138 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Diagnostics;
4 using System.Drawing;
5 using System.IO;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace EqualizadorDelImagem
12 {
13     public class ExercicioNataly
14     {
15         public static void Equalizar(string caminholImagem, bool comCor = false)
16         {
17             Bitmap imagem = new Bitmap(caminholImagem);
18             Bitmap imagemEqualizada = new Bitmap(caminholImagem);
19
20             List<ImagemModel> imagemObj = new List<ImagemModel>();
21
22             int contador = 0;
23             for (int i = 0; i < imagem.Width; i++)
24             {
25                 for (int j = 0; j < imagem.Height; j++)
26                 {
27                     ImagemModel a = new ImagemModel();
28                     Color pixel = imagem.GetPixel(i, j);
29
30                     a.corOriginal = pixel;
31                     a.identificador = contador;
32                     if (comCor)
33                     {
34                         a.K = Utils.CorGetLuminosidade(pixel);
35                         a.KAzul = pixel.B;
36                         a.KVerde = pixel.G;
37                         a.KVermelho = pixel.R;
38                     }
39                     else
40                     {
41                         a.K = Utils.CorGetLuminosidade(pixel);
42                     }
43
44                     a.localizacaoPixel = new Point(i, j);
45                     imagemObj.Add(a);
46                     contador++;
47                 }
48             }
49
50             var listinha = imagemObj.OrderBy(x => x.identificador).ToList();
51             List<int> K = new List<int>();
52             List<int> KAzul = new List<int>();
53             List<int> KVerde = new List<int>();
54             List<int> KVermelho = new List<int>();
55             if (comCor)
56             {
57                 KAzul = ExercicioProfessor.Equalizar(listinha.Select(x => x.KAzul).ToList());
58                 KVerde = ExercicioProfessor.Equalizar(listinha.Select(x => x.KVerde).ToList());
59                 KVermelho = ExercicioProfessor.Equalizar(listinha.Select(x => x.KVermelho).ToList());
60             }
61             else
62             {
63                 K = ExercicioProfessor.Equalizar(listinha.Select(x => x.K).ToList());
64             }
65
66             foreach (ImagemModel item in listinha)
```

```
67     {
68         Color cor = new Color();
69         var i = item.identificador;
70         if (comCor)
71             cor = Color.FromArgb(KVermelho[i], KVerde[i], KAzul[i]);
72         else
73             cor = Color.FromArgb(K[i], K[i], K[i]);
74
75         imagemEqualizada.SetPixel(item.localizacaoPixel.X, item.localizacaoPixel.Y, cor);
76     }
77
78     SaveFileDialog save = new SaveFileDialog();
79     save.FileName = caminholImagem.Replace(".jpg", "") + Guid.NewGuid().ToString() + ".jpg";
80     save.DefaultExt = Path.GetExtension(caminholImagem);
81     if (save.ShowDialog() == DialogResult.OK)
82     {
83         imagemEqualizada.Save(save.FileName);
84         Process.Start(Path.GetDirectoryName(save.FileName));
85     }
86 }
87 }
88 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace EqualizadorDeImagem
9 {
10     public class ImagemModel
11     {
12         public int identificador { get; set; }
13         public Point localizacaoPixel { get; set; }
14         public Color corOriginal { get; set; }
15         public int K { get; set; }
16         public int KVermelho { get; set; }
17         public int KAzul { get; set; }
18         public int KVerde { get; set; }
19     }
20 }
21 }
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Drawing;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace EqualizadorDeImagem
9 {
10     public class Utils
11     {
12         public static bool TextoContemSomenteNumeros(string text, List<string> ignore = null)
13         {
14             if (text == null)
15                 return false;
16
17             if (ignore != null && ignore.Count > 0)
18             {
19                 foreach (var a in ignore)
20                 {
21                     text = text.Replace(a, "");
22                 }
23             }
24
25             List<int> valuesnumber = new List<int>() { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
26
27             foreach (var a in valuesnumber)
28             {
29                 text = text.Replace(a.ToString(), "");
30             }
31
32             return string.IsNullOrEmpty(text);
33         }
34         public static int CorGetLuminosidade(Color cor)
35         {
36             return Convert.ToInt32(0.299 * cor.R + 0.587 * cor.G + 0.114 * cor.B);
37         }
38     }
39 }
```