

Graficar Histograma de Imagen RGB

En la siguiente actividad se utilizará para la transformación el promedio de R, G y B, y se graficarán sus valores en un histograma en base a una imagen de entrada.

Desarrollo

Cree un nuevo proyecto de Windows Forms en lenguaje C# y luego agregue las siguientes características, guiándose con la figura 1:

- a) En la ventana de diseño agregue dos botones:
 - a. "Cargar Imagen".
 - b. "Salir".
- b) Incorpore además un *PictureBox*, de tamaño adecuado para visualizar la imagen y modifique la propiedad *SizeMode* a "Zoom".
- c) Añada 3 charts.
- d) Añada por último un *openFileDialog*.

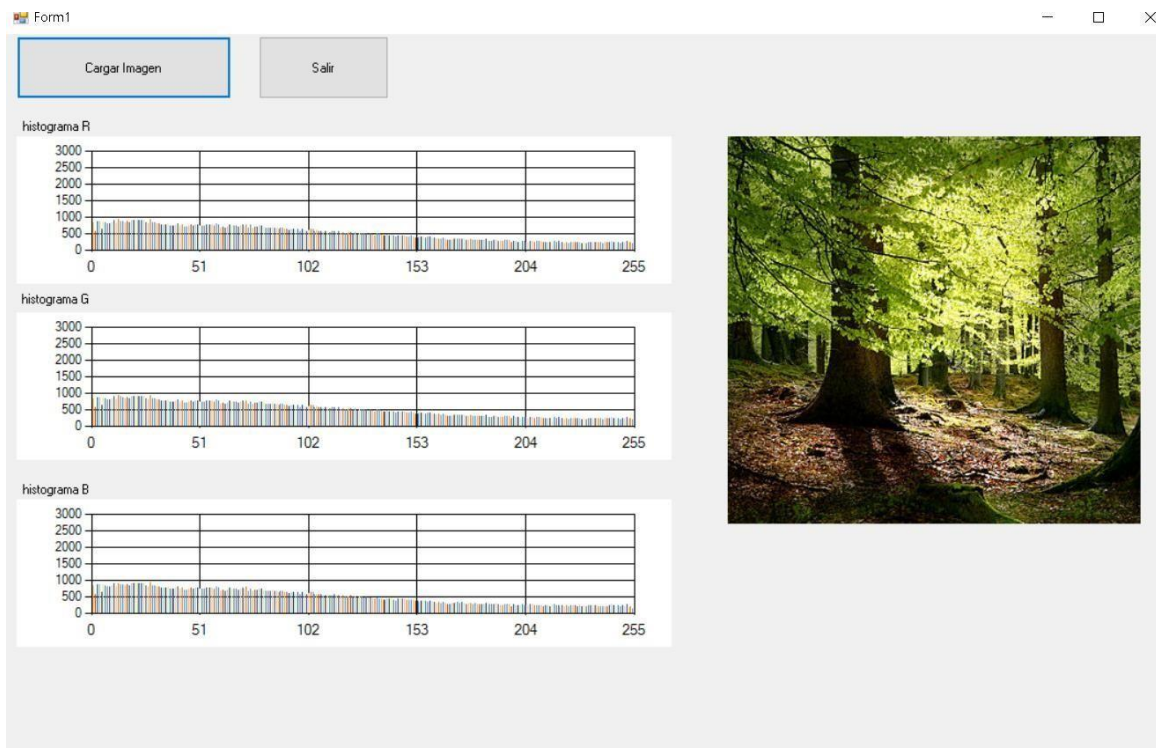


Figura 1

1. Bibliotecas que se van a utilizar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
```

2. En el evento del botón Salir escriba el siguiente código:

```
private void Button2_Click(object sender, EventArgs e)
{
    Close();
}
```

3. En el evento botón Cargar Imagen escriba el siguiente código:

```
try
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        string Ruta = openFileDialog1.FileName;
        Bitmap imagen = new Bitmap(Ruta);
        pictureBox1.Image = imagen;

        // crea histogramas rgb

        int[] histogramaR = new int[256];
        int[] histogramaG = new int[256];
        int[] histogramaB = new int[256];
        for (int f = 0; f < 256; f++)
        {
            histogramaR[f] = 0;
            histogramaG[f] = 0;
            histogramaB[f] = 0;
        }
        for (int f = 0; f < imagen.Width; f++)
```

```
for (int f = 0; f < imagen.Width; f++)
{
    for (int c = 0; c < imagen.Height; c++)
    {
        Color pixel = imagen.GetPixel(f, c);
        histogramaR[pixel.R] = histogramaR[pixel.R] + 1;
        histogramaG[pixel.G] = histogramaG[pixel.G] + 1;
        histogramaB[pixel.B] = histogramaB[pixel.B] + 1;
    }
}

for (int f = 0; f < 256; f++)
{
    chart1.ChartAreas[0].AxisX.Minimum = 0;
    chart1.ChartAreas[0].AxisX.Maximum = 256;

    chart2.ChartAreas[0].AxisX.Minimum = 0;
    chart2.ChartAreas[0].AxisX.Maximum = 256;

    chart3.ChartAreas[0].AxisX.Minimum = 0;
    chart3.ChartAreas[0].AxisX.Maximum = 256;

    Series intensidades = chart1.Series.Add(f.ToString());
    intensidades.Points.AddXY(f, histogramaR[f]);

    Series intensidades2 = chart2.Series.Add(f.ToString());
    intensidades2.Points.AddXY(f, histogramaG[f]);

    Series intensidades3 = chart3.Series.Add(f.ToString());
    intensidades3.Points.AddXY(f, histogramaB[f]);
}
}

catch (Exception ex)
{
    MessageBox.Show("Imagen no válida. Error: " + ex);
}
```

Desafío:

Realizar la misma aplicación, pero para una imagen en escala de grises