1.5. Separación de componentes de frecuencia según el modelo RGB

Ahora los componentes de una imagen RGB, si tenemos $f(x,y)$ la imagen en color, sus componentes serán:

- $f_R(x,y) = f(x,y)\ \&\ 0x00ff0000 >> 16$ es el componente rojo.
- $f_G(x,y) = f(x,y)\ \&\ 0x0000ff00 >> 8$ es el componente verde.
- $f_B(x,y) = f(x,y)\ \&\ 0\text{x}000000\text{ff}$ es el componente azul.

Así todos los componentes

$$f(x,y) = f_R(x,y)\ + f_G(x,y)\ + f_B(x,y) \tag{1.60}$$

Para ver la separación de los componentes hemos creado una clase que se llama FrameComponenteRGB la cual se muestra en la figura 1.20.

```
1   package vista;
2
3   import control.ControlImagen;
4   import java.awt.Container;
5   import java.awt.event.WindowEvent;
6   import java.awt.event.WindowListener;
7   import javax.swing.JFrame;
8
9   /**
10   *
11   * @author sdelaot
12   */
13  public class FrameComponenteRGB extends JFrame {
14      private PanelDeImagen panelRojo;
15      private PanelDeImagen panelVerde;
16      private PanelDeImagen panelAzul;
17      private ControlImagen controlImagen;
18      public FrameComponenteRGB(String nombreArchivo) {
19          super("Visor de imagen");
20          initComponents(nombreArchivo);
21      }
22      private void initComponents(String nombreArchivo) {
23          Container contenedor = this.getContentPane();
24          contenedor.setLayout(null);
25          controlImagen = new ControlImagen(nombreArchivo);
26          panelRojo = new PanelDeImagen(controlImagen.getImagen(1));
27          panelRojo.setBounds(0, 0,
28                              controlImagen.getAncho(),
29                              controlImagen.getAlto());
30          contenedor.add(panelRojo);
31          panelVerde = new PanelDeImagen(controlImagen.getImagen(2));
32          panelVerde.setBounds(controlImagen.getAncho(), 0,
33                              controlImagen.getAncho(),
34                              controlImagen.getAlto());
35          contenedor.add(panelVerde);
36          panelAzul = new PanelDeImagen(controlImagen.getImagen(3));
37          panelAzul.setBounds(controlImagen.getAncho()*2, 0,
38                              controlImagen.getAncho(),
39                              controlImagen.getAlto());
40          contenedor.add(panelAzul);
41          this.setSize(controlImagen.getAncho()*3, controlImagen.getAlto()+40);
42          this.setVisible(true);
43          this.addWindowListener(new SalidaFrame());
44      }
```

```
45      private class SalidaFrame extends Object implements WindowListener {
46          @Override
            public void windowOpened(WindowEvent e) { }
48          @Override
            public void windowClosing(WindowEvent e) {
50              System.exit(0);
51          }
52          @Override
            public void windowClosed(WindowEvent e) {
54              System.exit(0);
55          }
56          @Override
            public void windowIconified(WindowEvent e) { }
58          @Override
            public void windowDeiconified(WindowEvent e) { }
60          @Override
            public void windowActivated(WindowEvent e) { }
62          @Override
            public void windowDeactivated(WindowEvent e) { }
64      }
65  }
```

Figura 1.20 Clase FrameComponenteRGB.

Ahora para ejecutarlo se modificó la clase principal ImageAnalysis como se muestra en la figura 1.21.

```
1   package imageanalysis;
2
3   import vista.FrameComponenteRGB;
4   //import vista.FrameVisorImagen;
5
6   /**
7    *
8    * @author sdelaot
9    */
10  public class ImageAnalysis {
11      /**
12       * @param args the command line arguments
13       */
14      public static void main(String[] args) {
15          //FrameVisorImagen visorI = new FrameVisorImagen("aguila.jpg");
16          FrameComponenteRGB visor = new FrameComponenteRGB("aguila.jpg");
17      }
18
19  }
```

Figura 1.21 Clase principal ImageAnalysis.

Ahora en la figura 1.22 se muestra el resultado de la ejecución del programa para mostrar los componentes de la imagen RGB.

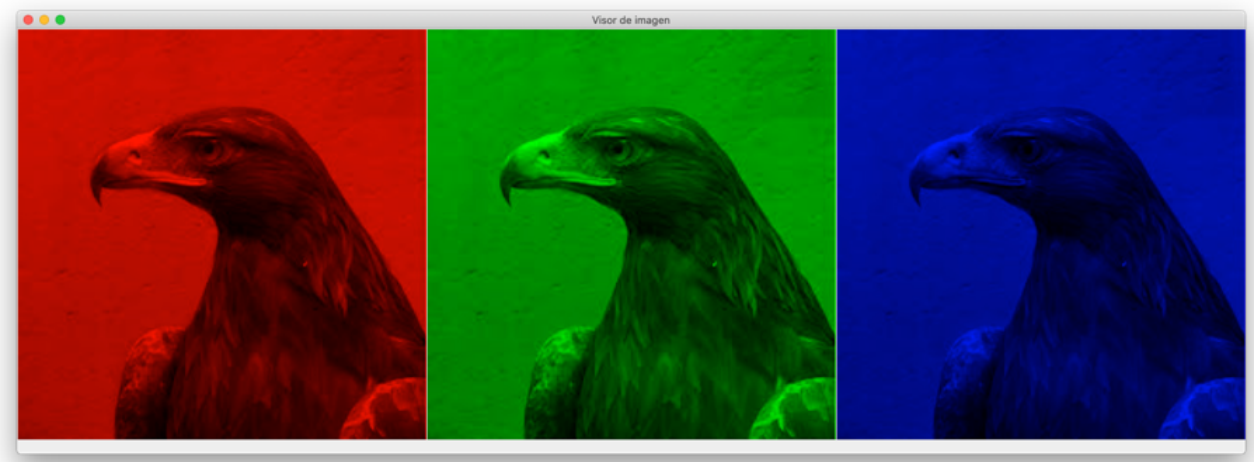

Figura 1.22 Componentes RGB de la imagen de la figura 1.19.