

Practica 2

Nombre: Franco Lopez Angel Sebastian Grupo: M5

Metodos nuevos

Cambios en metodos anteriormente hechos

```
public ControladorAplicacion(Principal Ventana) {
    this.ventanaPrincipal = Ventana;
    this.ventanaPrincipal.AgregarImagen.addActionListener(this);
    this.ventanaPrincipal.ConvertirGrises.addActionListener(this);
}

public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals("Abrir imagen")) {
        System.out.println("Seleccionaste Abrir imagen");

        if (this.ventanaPrincipal.jLabel1.getIcon() == null) {
            this.cargarRGB();
        }

        this.ventanaPrincipal.ConvertirGrises.setEnabled(true);
        this.ventanaPrincipal.ConvertirGrises.setVisible(true);
    }

    if (e.getActionCommand().equals("Convertir a escala de grises")) {
        System.out.println("Seleccionaste Convertir a escala de grises");
        // como paso las matrices?
        this.cargarGris(this.imagenOriginal.getMatrizR(),
            this.imagenOriginal.getMatrizG(), this.imagenOriginal.getMatrizB());

        this.imagenOriginal.convierteMatrizEnBuffered(this.imagenOriginal.getMatrizR());
        this.ventanaPrincipal.jLabel1.setIcon(new
            ImageIcon(this.imagenOriginal.getBufferImagen().getScaledInstance(100, 200,
                Image.SCALE_DEFAULT)));
    }
}
```

Cargar gris

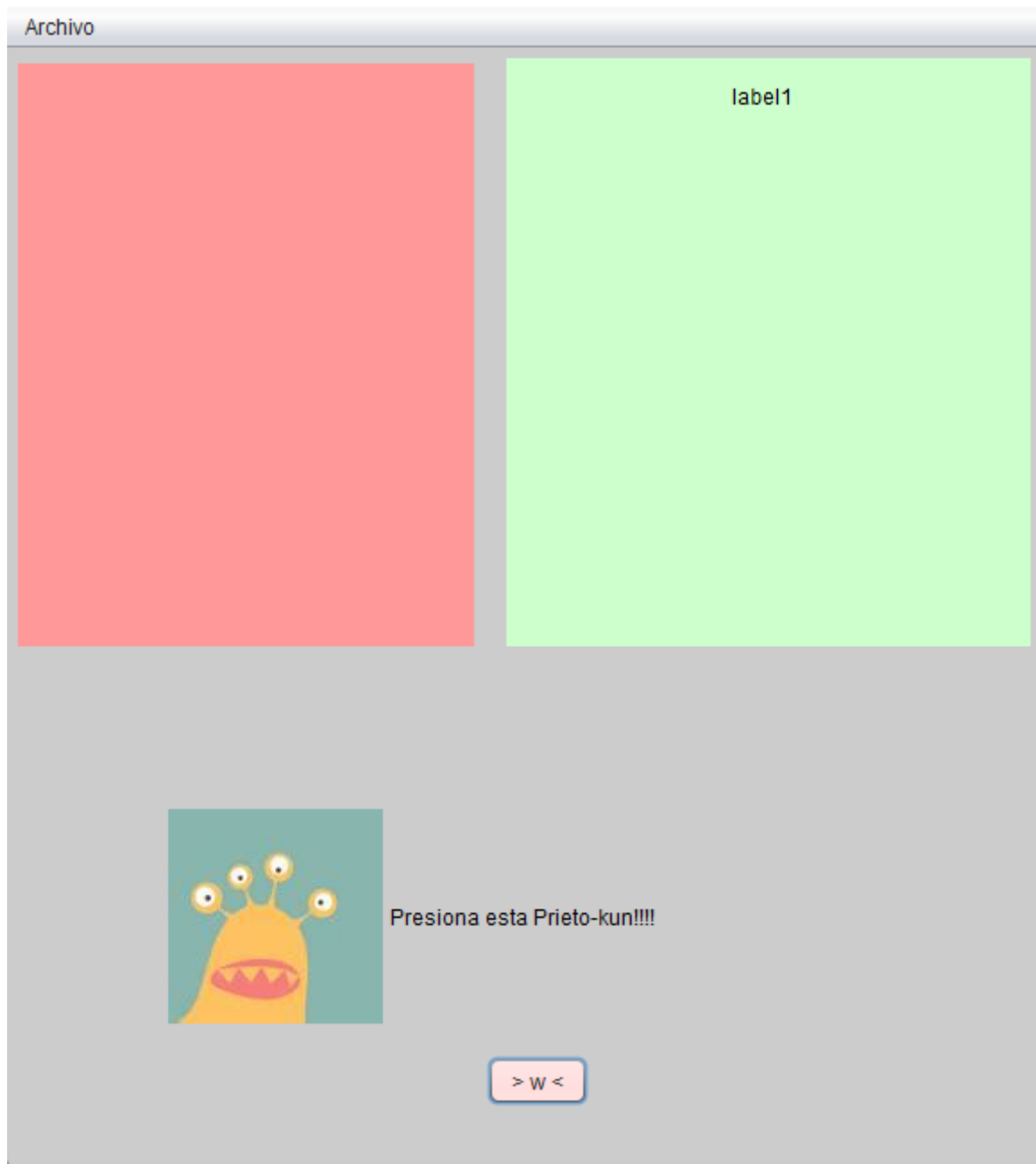
```
private void cargarGris(short matrizR[][], short matrizG[][], short matrizB[][]) {
    short matrizResultante[][] = new short[matrizR.length][matrizR[0].length];

    for (int i = 0; i < matrizR.length; i++) {
        for (int j = 0; j < matrizR[0].length; j++) {
            // operacion
            int suma = (matrizR[i][j] + matrizG[i][j] + matrizB[i][j]);
        }
    }
}
```

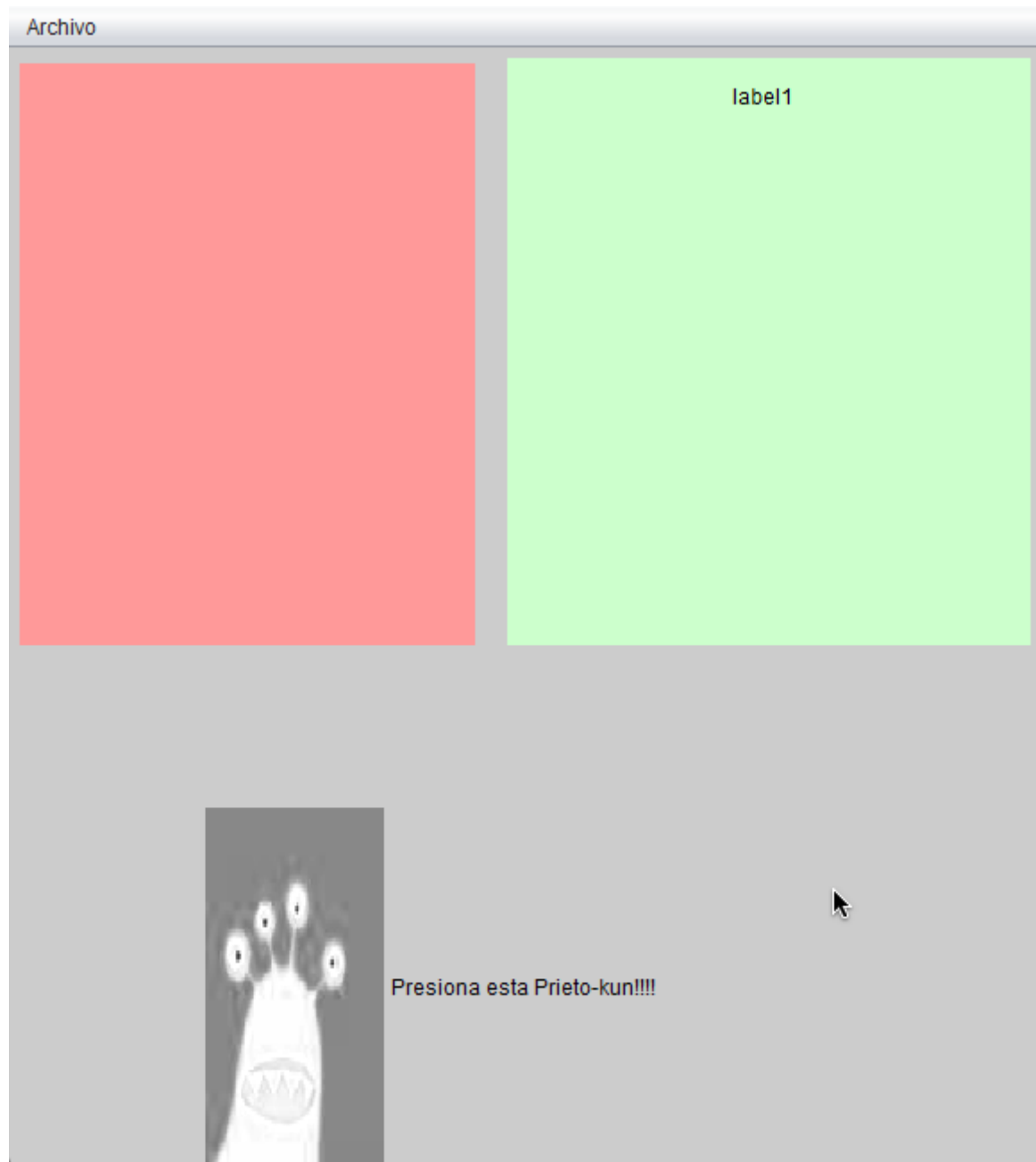
```
        int resultado = (suma / 3);  
        matrizResultante[i][j] = (short) resultado;  
    }  
}  
this.imagenOriginal.setMatrizGris(matrizResultante);  
}
```

Observacion de los resultados

- imagen a color:



- imagen a escala de grises:



Explicacion de la linea

El metodo es un bucle anidado que recorre todos los píxeles de una imagen. En cada iteración del bucle interno, se realiza lo siguiente:

- Se obtiene el valor del píxel actual de la matriz de grises.
- Se convierte el valor del píxel de grises a un valor SRGB.
- Se establece el valor del píxel SRGB en la imagen de destino.

La línea específica que has pedido que se explique, `pixelSRGB = pixelNuevo << 16 | pixelNuevo << 8 | pixelNuevo;`, convierte el valor del píxel de grises a un valor SRGB.

Para entender cómo funciona esta línea, primero debemos entender cómo se almacenan los píxeles de grises y los píxeles SRGB en imágenes.

Los píxeles de grises se almacenan como un solo valor, que representa el nivel de gris del píxel. Este valor puede variar de 0 a 255, donde 0 representa el negro y 255 representa el blanco.

Los píxeles SRGB se almacenan como tres valores, que representan los niveles de rojo, verde y azul del píxel. Cada valor puede variar de 0 a 255, donde 0 representa el color negro y 255 representa el color más intenso.

Para convertir un píxel de grises a un píxel SRGB, primero debemos multiplicar el valor del píxel de grises por 255. Esto nos dará un valor que puede usarse como el nivel de rojo, verde o azul del píxel SRGB.

Una vez que tenemos el valor de rojo, verde o azul, debemos desplazarlo a la izquierda por un número de bits determinado. El número de bits que se desplazan depende del canal de color. El canal rojo se desplaza 16 bits a la izquierda, el canal verde se desplaza 8 bits a la izquierda y el canal azul se desplaza 0 bits a la izquierda.

Por ejemplo, si el valor del píxel de grises es 128, el valor del rojo del píxel SRGB será 128. Para obtener este valor, multiplicamos 128 por 255, lo que nos da 32768. Luego, desplazamos 32768 16 bits a la izquierda, lo que nos da 16711680.

El código `pixelSRGB = pixelNuevo << 16 | pixelNuevo << 8 | pixelNuevo;` realiza estos dos pasos para convertir un píxel de grises a un píxel SRGB. El operador `|` se utiliza para combinar los tres valores de rojo, verde y azul en un único valor de píxel SRGB.

En resumen, el código que has proporcionado recorre todos los píxeles de una imagen y convierte cada píxel de grises a un píxel SRGB.