

## Explicación del Código para Manejo de Imágenes en el Proyecto

### 1. Introducción

En este proyecto desarrollado con Java Swing en NetBeans, se implementó el uso de imágenes para representar visualmente los productos en la interfaz. El objetivo fue enriquecer la experiencia del usuario integrando imágenes dentro de tablas, etiquetas y vistas detalladas de cada producto. A continuación, se explica el código utilizado para lograr esto, dividido en dos partes: carga de imágenes y visualización dentro de los JFrames.

### 2. Código para Cargar Imágenes (JFileChooser)

Se utiliza un botón que abre un JFileChooser para seleccionar una imagen desde el sistema de archivos. El resultado se muestra en un JLabel y se guarda la ruta en un campo de texto oculto o variable.

```
JFileChooser fileChooser = new JFileChooser();
FileNameExtensionFilter filter = new
FileNameExtensionFilter("Imágenes", "jpg", "jpeg", "png");
fileChooser.setFileFilter(filter);

int resultado = fileChooser.showOpenDialog(null);
if (resultado == JFileChooser.APPROVE_OPTION) {
    File archivoSeleccionado = fileChooser.getSelectedFile();
    String ruta = archivoSeleccionado.getAbsolutePath();
    txtRutaImagen.setText(ruta);
    ImageIcon icon = new ImageIcon(ruta);
    Icon iconoEscalado = new
    ImageIcon(icon.getImage().getScaledInstance(lblPreview.getWidth(
    ), lblPreview.getHeight(), Image.SCALE_SMOOTH));
    lblPreview.setIcon(iconoEscalado);
}
```

---

### 3. Visualización de Imágenes en JTable

Para mostrar las imágenes dentro de una tabla, se cargan con ImageIcon y se redimensionan antes de agregarlas al modelo de la tabla:

```
ImageIcon icono = new ImageIcon(producto.getRutaImagen());
Image imagen = icono.getImage().getScaledInstance(60, 60,
Image.SCALE_SMOOTH);
model.addRow(new Object[] { producto.getNombre(),
```

```
producto.getPrecio(), new ImageIcon(imagen) });
```

---

#### **4. Uso de Imágenes en DetalleProductoFrame**

En la vista detallada del producto, la imagen se carga usando ImageIcon directamente en un JLabel:

```
ImageIcon icon = new ImageIcon(rutaImagen);  
Image scaledImage =  
icon.getImage().getScaledInstance(lblImagen.getWidth(),  
lblImagen.getHeight(), Image.SCALE_SMOOTH);  
lblImagen.setIcon(new ImageIcon(scaledImage));
```

---

#### **5. Conclusión**

El manejo de imágenes en este proyecto permite una mejor presentación visual de los productos y contribuye a una experiencia de usuario más completa. Se implementó utilizando clases estándar de Java como ImageIcon, JFileChooser y JLabel, con una lógica clara y modular.