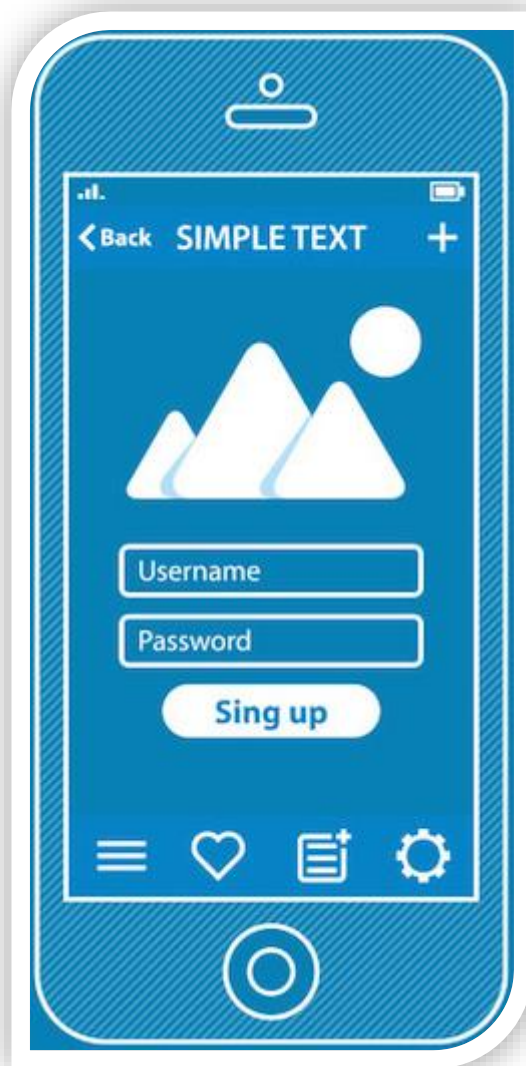


# Diseño de Interfaces de Usuario



**Nombre:** Zabai Armas Herrera

**Curso:** 4°

## Índice:

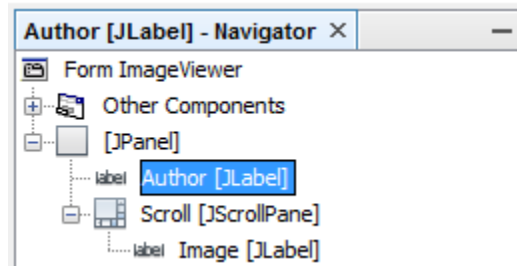
1. Práctica 7. Diseño de patrones de usuario
  - 1.1. Visor de imagen
  - 1.2. Contenedor de pestañas
  - 1.3. Escritorio con ventanas interiores
  - 1.4. Generalización
2. Capturas
3. Bibliografía

# 1. Práctica 7. Diseño de patrones de usuario

## 1.1 Visor de imagen

Importación de la clase “ImageFilter.java” al proyecto.

Creación de un Form de JPanel para el “Visor Imagen” con sus correspondientes elementos y métodos.

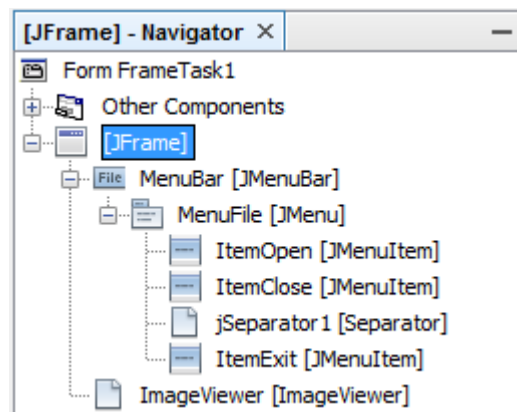


*Ilustración 1 Componentes del Visor Imagen*

```
public void setImage(File file){  
    if(file != null) Image.setIcon(new javax.swing.ImageIcon(file.getAbsolutePath()));  
    else Image.setIcon(null);  
}
```

*Ilustración 2 Método para asignar imagen*

En el JFrame principal creamos la barra de menú pedida e insertamos el objeto anterior mediante Beans.



*Ilustración 3 Elementos del JFrame principal*

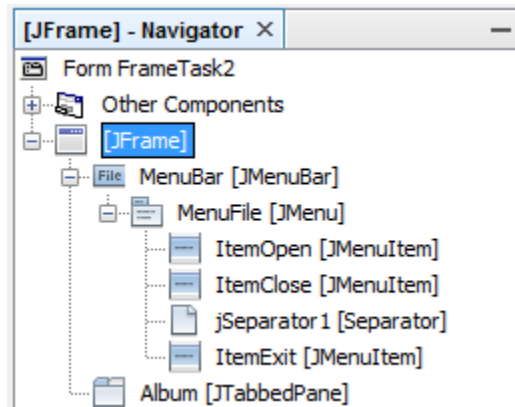
Queda la configuración de cada elemento del menú.

```
private void ItemOpenActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JFileChooser fileChooser = new JFileChooser();  
    fileChooser.setCurrentDirectory(null);  
    ImageFilter filter = new ImageFilter();  
    fileChooser.addChoosableFileFilter(filter);  
    fileChooser.setFileFilter(filter);  
  
    int returned = fileChooser.showOpenDialog(this);  
    if(returned == JFileChooser.APPROVE_OPTION){  
        File file = fileChooser.getSelectedFile();  
        ImageViewer.setImage(file);  
    }  
}  
  
private void ItemCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    ImageViewer.setImage(null);  
}  
  
private void ItemExitActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

*Ilustración 4 Eventos del menú*

## 1.2 Contenedor de pestañas

Creamos los elementos necesarios para la interfaz.



*Ilustración 5 Elementos del contenedor de pestañas*

Configuración de las acciones de cada elemento del menú, con la mejora de que, al abrir una nueva imagen, la pestaña cambie a esta.

```
private void ItemOpenActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JFileChooser fileChooser = new JFileChooser();  
    fileChooser.setCurrentDirectory(null);  
    ImageFilter filter = new ImageFilter();  
    fileChooser.addChoosableFileFilter(filter);  
    fileChooser.setFileFilter(filter);  
  
    int returned = fileChooser.showOpenDialog(this);  
    if(returned == JFileChooser.APPROVE_OPTION){  
        File file = fileChooser.getSelectedFile();  
        ImageViewer viewer = new ImageViewer();  
        viewer.setImage(file);  
        Album.add(file.getName(), viewer);  
        // Mejora, cambia de panel cuando abres una imagen nueva  
        Album.setSelectedIndex(Album.getSelectedIndex()+1);  
    }  
}  
  
private void ItemCloseActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int index = Album.getSelectedIndex();  
    if(index != -1) Album.removeTabAt(index);  
}  
  
private void ItemExitActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

*Ilustración 6 Funciones del menú*

## 1.3 Escritorio con ventanas interiores

Creación de los elementos necesarios para la interfaz.

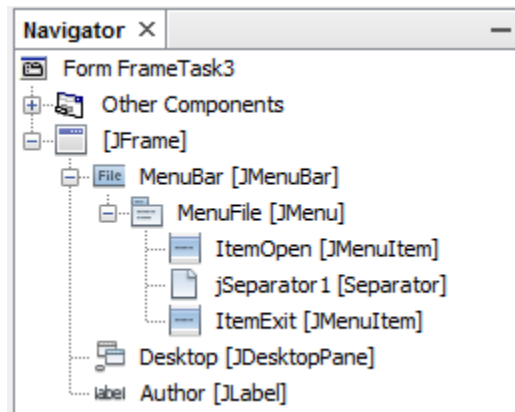


Ilustración 7 Componentes de la interfaz

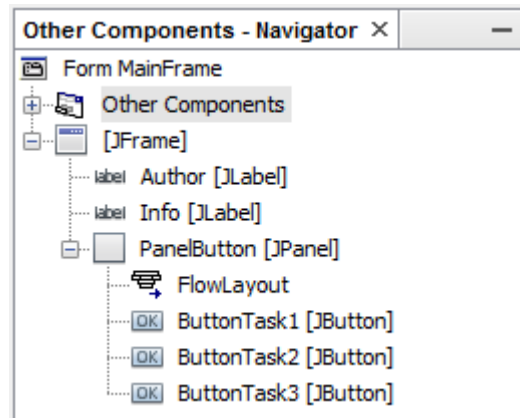
Configuración de las acciones de cada elemento del menú.

```
private void ItemOpenActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JFileChooser fileChooser = new JFileChooser();  
    fileChooser.setCurrentDirectory(null);  
    ImageFilter filter = new ImageFilter();  
    fileChooser.addChoosableFileFilter(filter);  
    fileChooser.setFileFilter(filter);  
  
    int returned = fileChooser.showOpenDialog(this);  
    if(returned == JFileChooser.APPROVE_OPTION){  
        File file = fileChooser.getSelectedFile();  
        if(file != null){  
            ImageViewer viewer = new ImageViewer();  
            viewer.setImage(file);  
  
            JInternalFrame window = new JInternalFrame();  
            window.setTitle(file.getName());  
            window.setSize(400, 200);  
            window.setClosable(true);  
            window.setIconifiable(true);  
            window.setResizable(true);  
            window.setMaximizable(true);  
            window.add(viewer);  
            window.setVisible(true);  
  
            Desktop.add(window);  
        }  
    }  
}  
  
private void ItemExitActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    System.exit(0);  
}
```

Ilustración 8 Configuración del menú

## 1.4 Generalización

Para no tener todas las tareas en varios proyectos, he creado un selector muy simple.



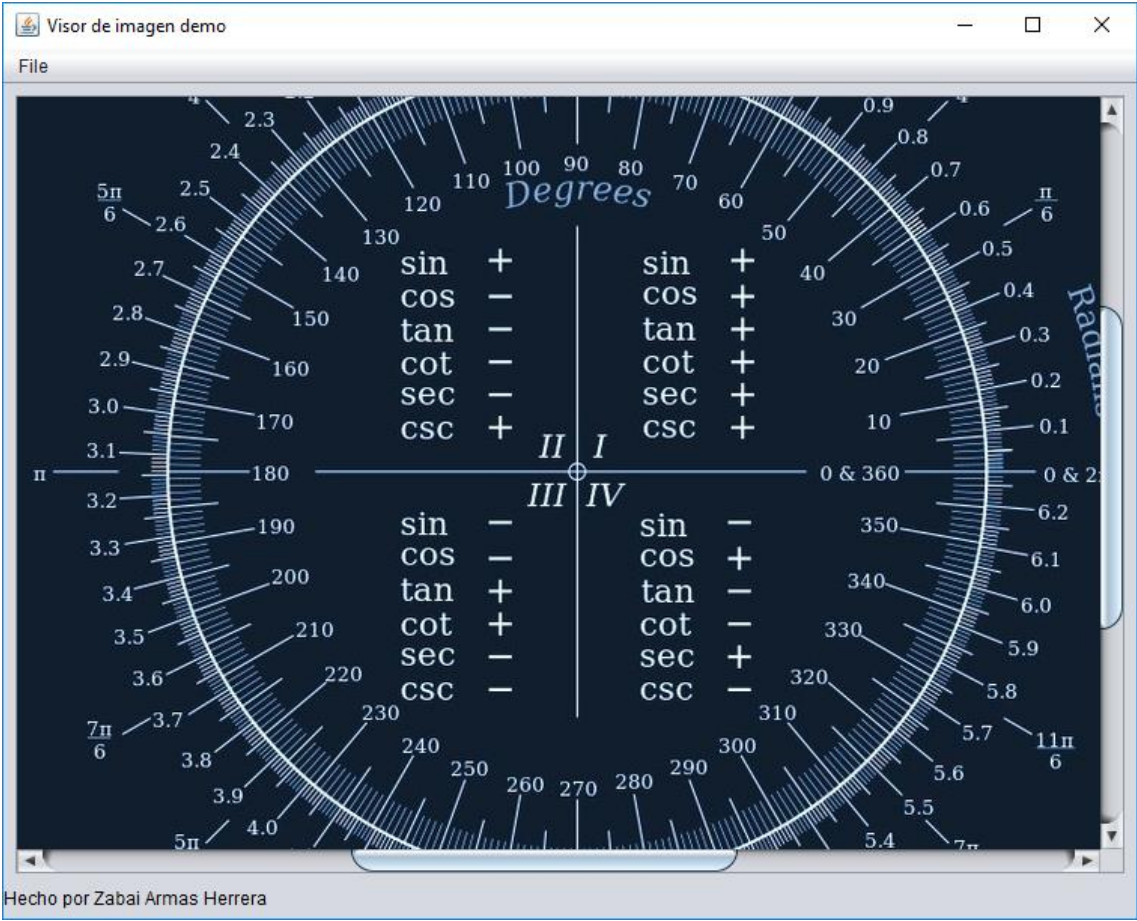
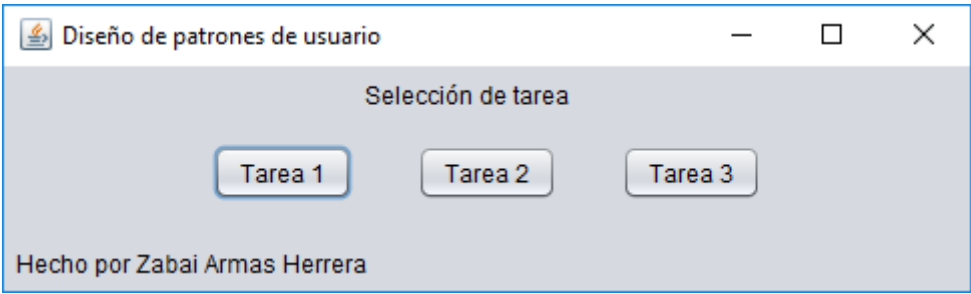
*Ilustración 9 Elementos del selector*

Con la función de crear una nueva ventana con respecto a la tarea elegida.

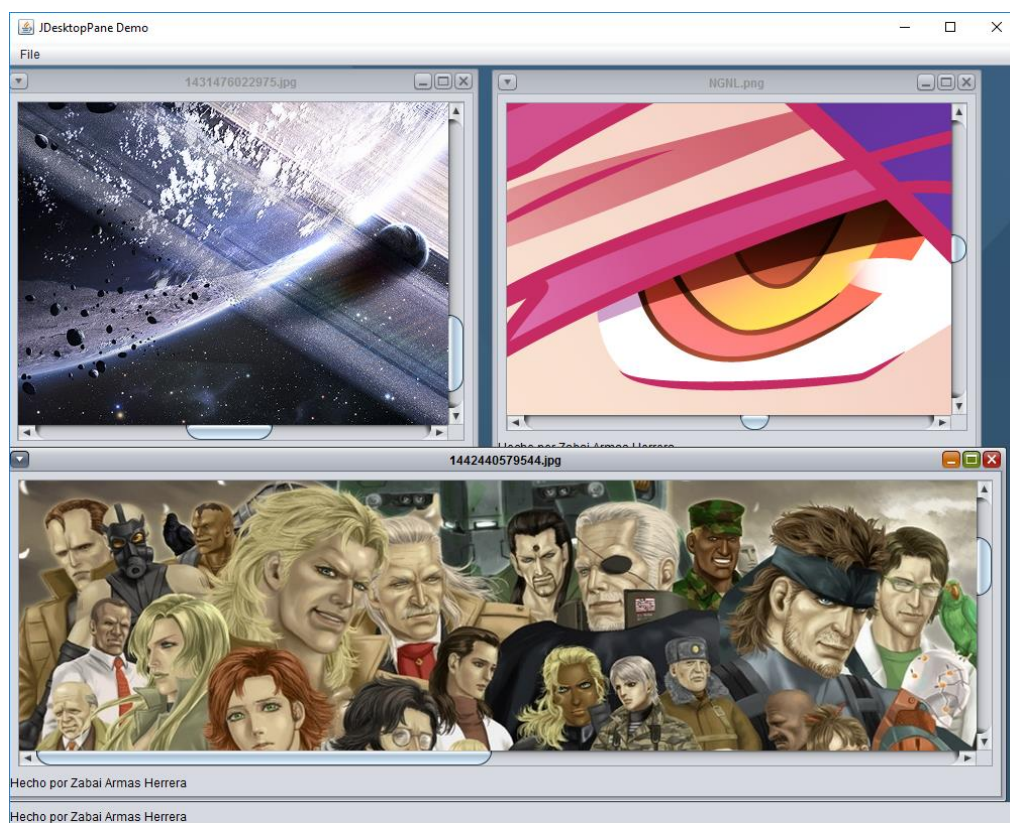
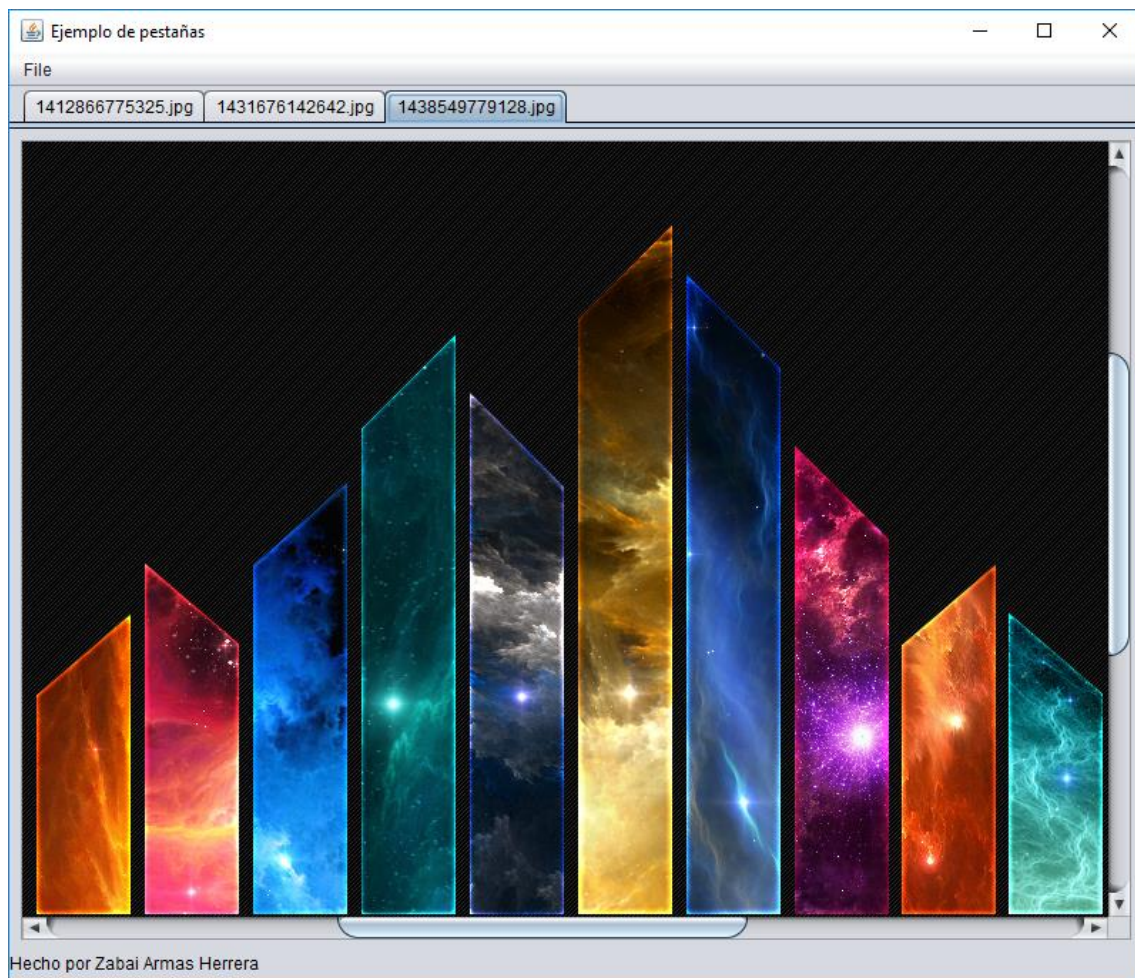
```
private void ButtonTask1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    new FrameTask1().setVisible(true);  
}  
  
private void ButtonTask2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    new FrameTask2().setVisible(true);  
}  
  
private void ButtonTask3ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    new FrameTask3().setVisible(true);  
}
```

*Ilustración 10 Función de selección*

## 2. Capturas







### 3. Bibliografía

- [PDF de la práctica](#)
- [Stack Overflow](#)