

## TEMA 5: HERENCIA

### PRÁCTICA 8. FILTROS DE IMAGEN<sup>1</sup>

El proyecto *ImageViewer* proporciona la funcionalidad básica para crear una pequeña aplicación basada en menus que permita cargar imágenes, aplicar filtros, visualizar y salvar el resultado. A modo de ejemplo, el proyecto incluye la clase `SaltPepperFilter` que permite añadir a una imagen un cierto porcentaje de ruido de sal y pimienta. La práctica consiste en añadir las clases necesarias para que la aplicación incluya la posibilidad de aplicar sendos filtros de media y mediana a las imágenes.

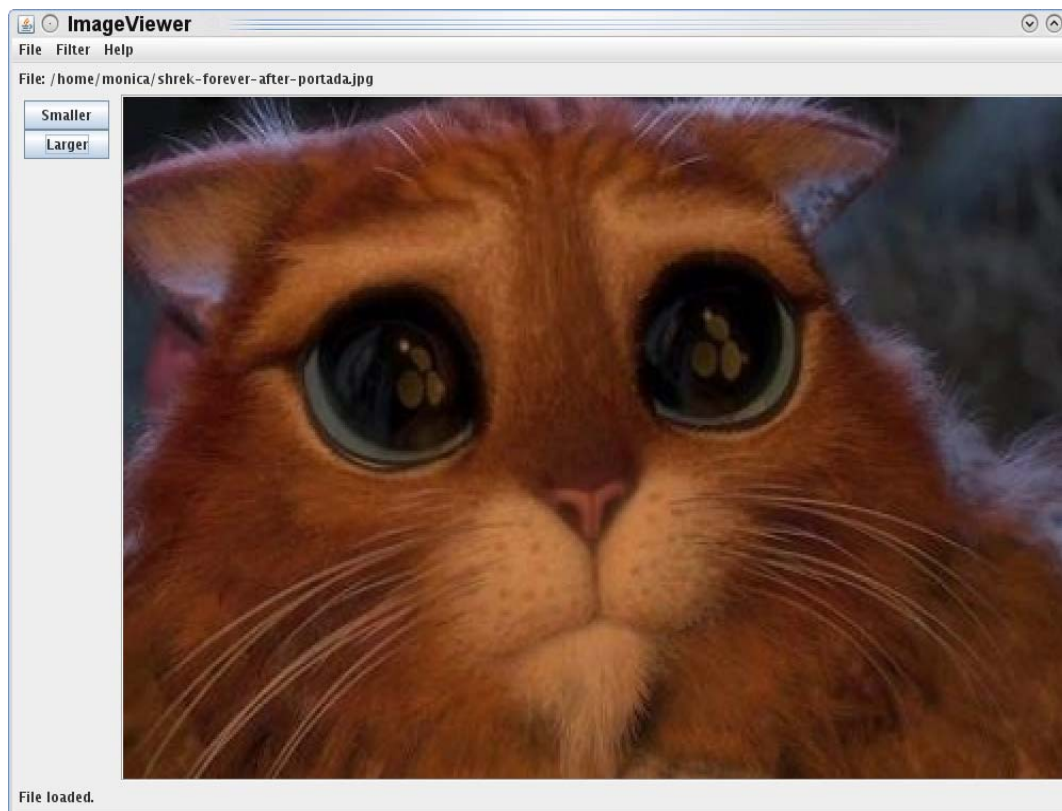


Figura 1. Aplicación implementada en el proyecto *ImageViewer*

<sup>1</sup> Ejercicio del libro: Barnes David J., Kölling Michael, “*Programación orientada a objetos con Java: una introducción práctica usando BlueJ*”, Ed. 3ª Pearson Education

Los filtros de media y mediana pertenecen a una de las familias de filtros de suavizado de imagen más populares en visión por computador ya que suelen ser necesarios como paso previo para infinidad de algoritmos. El filtro de media consiste en reemplazar el valor de los píxeles de la imagen original por la media de los valores calculada en una máscara cuadrada con un número impar de píxeles centrada en dicho píxel. El filtro de mediana, por su parte, consiste en reemplazar el valor de los píxeles de la imagen original por la mediana de los valores de la máscara. Un ejemplo de máscara 3 x 3 se muestra en la Figura 2.

0 (-1, -1)	1 (0, -1)	2 (1, -1)
3 (-1, 0)	4 (0, 0)	5 (1, 0)
6 (-1, 1)	7 (0, 1)	8 (1, 1)

radius = [1,1]  
size = [3,3]

*Figura 2. Ejemplo de máscara de tamaño 3 x 3 con centro en el píxel 4.*

Los pasos a seguir para el cálculo de estos filtros de suavizado son los siguientes:

1. Crear una imagen donde almacenar el resultado del filtro.
2. Para todos los píxeles de la imagen rellenar la máscara con los valores de intensidad de la imagen de modo que el centro de la máscara coincida con el píxel. Si la máscara cubre zonas fuera de la imagen, la intensidad añadida a la máscara será de RGB (0,0,0).
3. Calcular el estadístico (media o mediana) de las intensidades encontradas en la máscara.
4. Asignar dicho valor al píxel de la imagen que almacena el resultado del filtro.

De los pasos anteriores, se puede deducir que la mejor decisión de diseño para resolver el problema planteado es definir una clase madre `StatisticsFilter` que se encargue de realizar los pasos 1 – 4 y sendas clases hijas `MeanFilter` y `MedianFilter` que se encarguen de calcular el estadístico correspondiente (media o mediana), tal y como se indica en el diagrama de clases de la Figura 3.

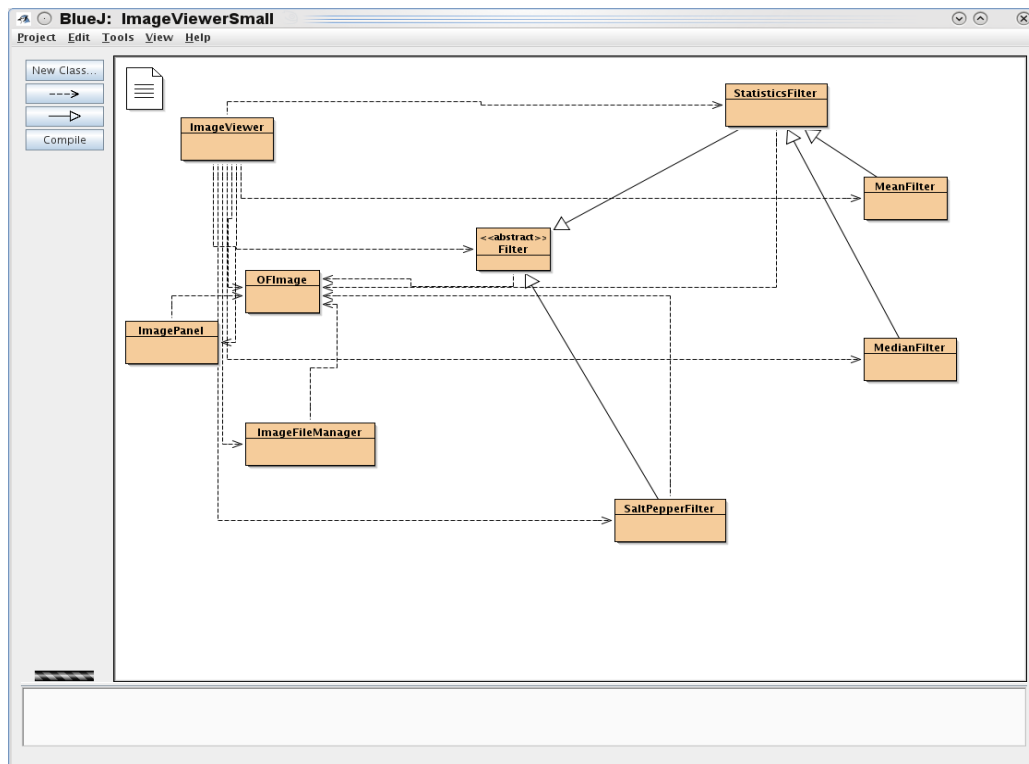


Figura 3. Diagrama de clases propuesto.

Para la creación de la estructura de los filtros de suavizado, toma como modelo la clase `SaltPepperFilter`. Finalmente, para incorporar los filtros de suavizado al menú de la aplicación, estudia el método `private List<Filter> createFilters()` donde se genera el menú `Filter` con la llamada a la ejecución de los filtros que incorpora.