

# Practica Final Sistemas Multimedia



Manuel Ortiz Hita

14/16/2019

## Indice:

---

Planteamiento:.....	3
Requisitos Funcionales:.....	3
Mi Aplicación:.....	4
Dibujo.....	7
Imagen .....	7
Mi Funcion LopkupOp:.....	8
Negativo:.....	9
Umbralizacion: .....	10
Sepia: .....	11
Filtro Random(componente a componente) .....	12
Filtro Bosque (Pixel a pixel) .....	13
Extracción de bandas: .....	14
Audio y Video:.....	15
Archivos: .....	16
Sistema Propio de clases:.....	17
Bibliografía .....	18

## Planteamiento:

---

Para esta práctica se elaborará una aplicación en Java que nos permita hacer dibujos, trazos, aplicar diversos tipos de filtros, reproducir clips de audio o video, crear clips de audio e imágenes a través de la cámara de nuestro dispositivo.

Dicha aplicación se hará poniendo en practica los conocimientos aprendido en este curso.

Dicha aplicación debe cumplir una serie de requisitos básicos:

## Requisitos Funcionales:

---

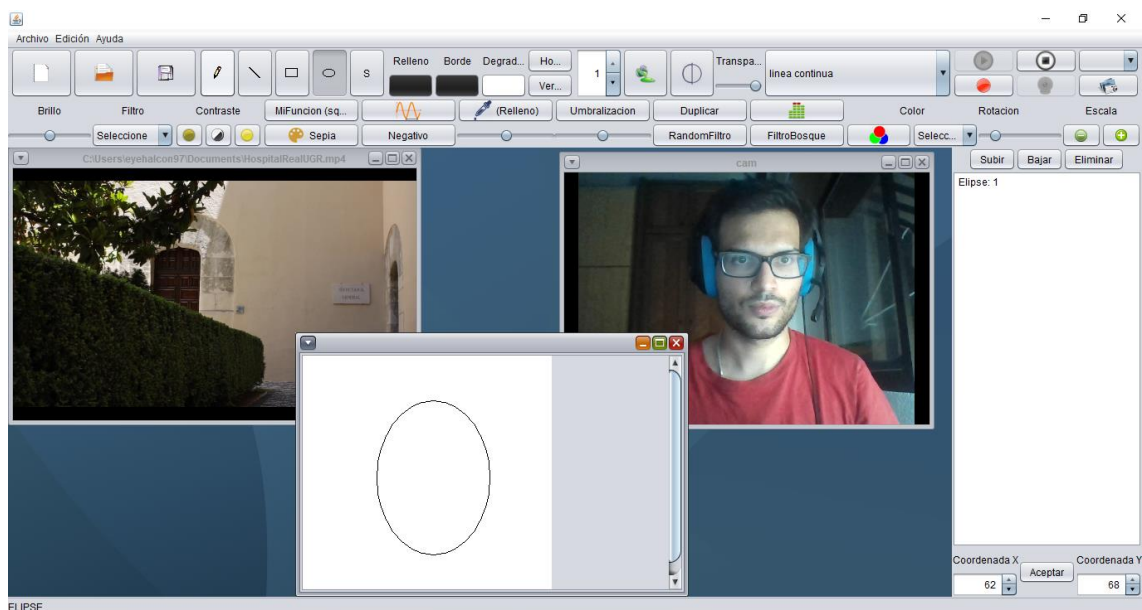
- RF1: Crear Ventana Principal del programa en la que nos proporcionara las herramientas de creación y de edición.
- RF2: Crear nuevas ventanas dentro de nuestra Ventana Principal, las cuales cada una tendrá sus propias propiedades y en las que se incluirá un lienzo en el cual nosotros podremos seleccionar su tamaño.
- RF3: Dichas ventanas puedes ser lienzos en blanco o un archivo imagen en el cual nosotros podremos trabajar encima de él.
- RF4: Podremos dibujar figuras en el lienzo (sea tanto blanco como imagen)
- RF5: Podremos modificar los atributos de cada figura como su trazo, grosor, así como eliminarlas o crear nuevas
- RF6: Permitir usar diversos tipos de filtros a las imágenes, así como tinto o umbralización.
- RF7: Permitir posicionar, escalar o rotar dichas imágenes.

- RF8: Permitir utilizar múltiples ventanas de lienzos, así como duplicarlas.
- RF9: Utilizar una lista de figuras en la cual podemos modificar la prioridad de estas para dibujar unas encima de otras, también podremos eliminar figuras.
- RF10: Guardar nuestras propias imágenes creadas.
- RF11: Crear clips de audio y guardarlos como archivo.
- RF12: Reproducir clips de audio, así como pausarlos y reanudarlos.
- RF13: Reproducir formatos de audio como mpg o mp4.
- RF14: Utilizar la cámara del dispositivo para crear imágenes.
- RF15: Crear imágenes de clip de video.

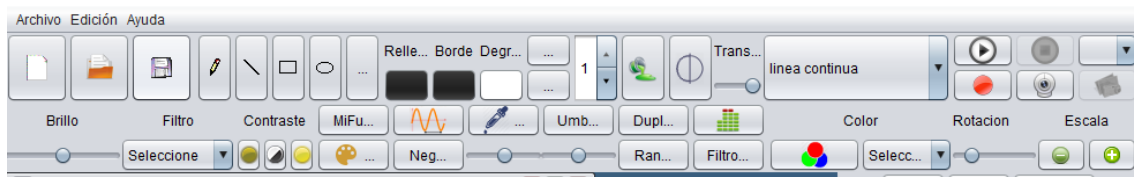
## Mi Aplicación:

---

En este apartado explicare mas a fondo nuestro programa:



Aquí tenemos una imagen nuestro software, su interfaz grafica se divide en dos zonas.



1. Arriba tenemos las herramientas para poder:

- Crear un lienzo, cargar un archivo y guardar un archivo.
- Crear distintos tipos de figuras, poner diferentes tipos de colores, diferente grosor o incluso diferentes transparencias.
- Ajustar el brillo de la imagen o un gran número de filtros o tintonarla.
- Rotar la imagen como más nos guste.
- Utilizar diversos tipos de contrastes.
- Escalar imágenes.
- Utilizar diversos tipos de bandas de colores.
- Crear clip de audio y reproducir una lista de clips tanto recién creados como cargados.
- Iniciar la Reproducción de clips de video con formato mpg o mp4, podremos iniciar o detener dicho clip.
- Abrir una ventana en la cual se muestra la imagen en tiempo real de la imagen que tiene la cámara.
- Podremos hacer capturas de una imagen ya sea de una ventana de clip de video o de una ventana de nuestra propia cámara.

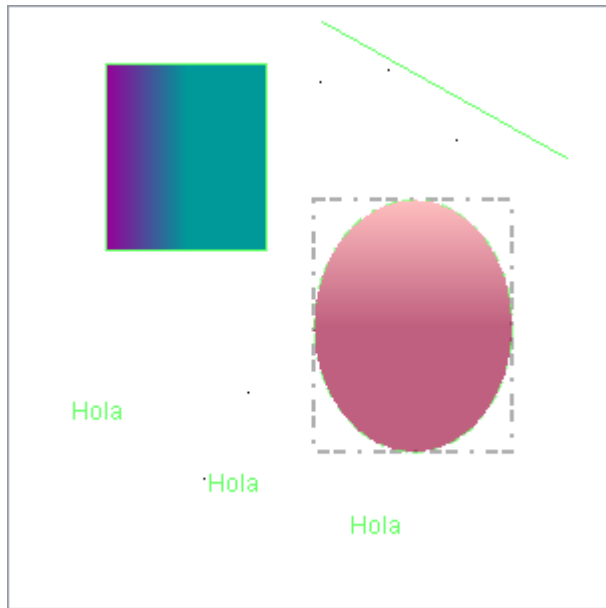
The image shows a software interface window. At the top, there is a horizontal bar containing three buttons: "Subir", "Bajar", and "Eliminar". Below this bar is a large, empty rectangular canvas. At the bottom of the window, there is a control panel. On the left, it is labeled "Coordenada X" and contains a text input field with the number "287" and a small vertical spinner. In the center of the control panel is a button labeled "Aceptar". To the right of the "Aceptar" button is another text input field labeled "Coordenada Y" containing the number "8", also with a small vertical spinner.

2. A nuestra derecha tenemos:

- Una lista en la que podemos seleccionar figuras ya creadas en el lienzo.
- Podemos cambiar la prioridad entre las figuras para saber que dibujar sobre que figura, también podemos eliminar figuras o incluso colocarla en las coordenadas que se deseen.

## Dibujo

---



En Este apartado explicare las cosas que podemos personalizar:

- Podemos crear elipses, rectángulos, puntos, líneas y string
- Podemos personalizar el color del borde, relleno o degradado
- Podemos seleccionar degradado horizontal o vertical
- Podemos seleccionar el tipo de trazado de las figuras
- Podemos elegir el nivel de grosor del borde de la figura
- Podemos establecer diferentes grados de transparencia mediante un slider.
- Podemos dibujar figuras y hacer que una se vean una u otra encima dependiendo de la prioridad de dicha figura

## Imagen

---

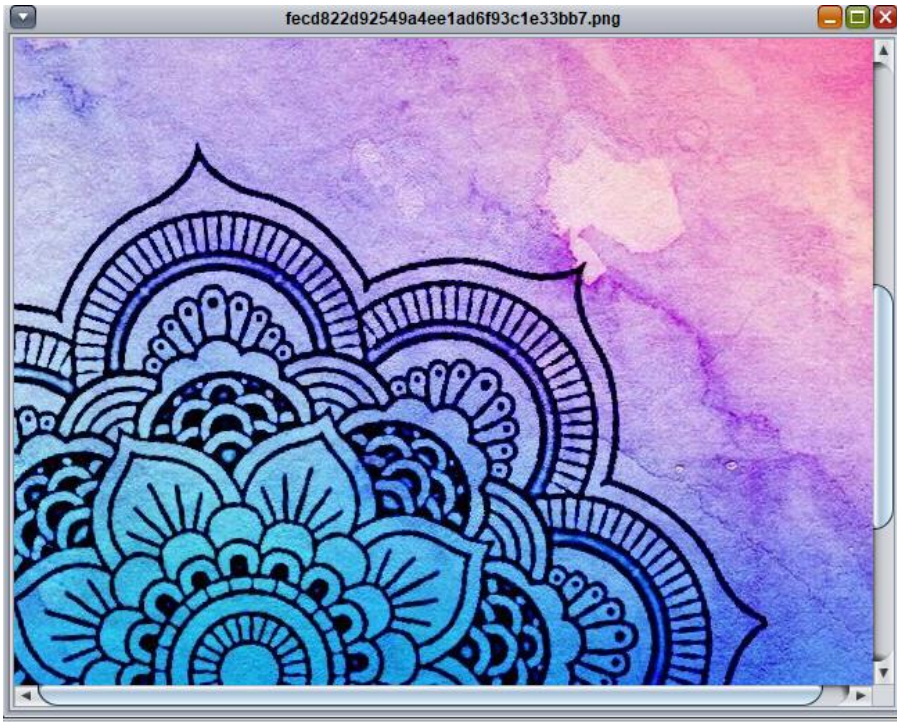
En este apartado explicaremos los diferentes tipos de filtros que podemos utilizar en nuestra aplicación:



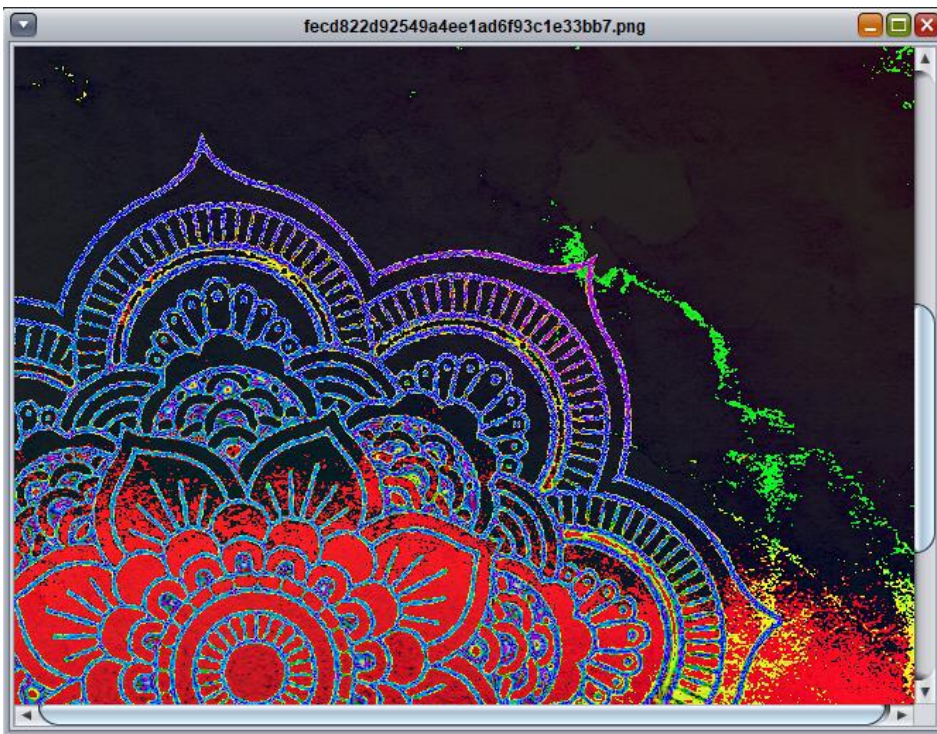
## Mi Funcion LopkupOp:

Esta función consiste en aplicar La raíz del seno a dicha figura:

Antes:



Después:

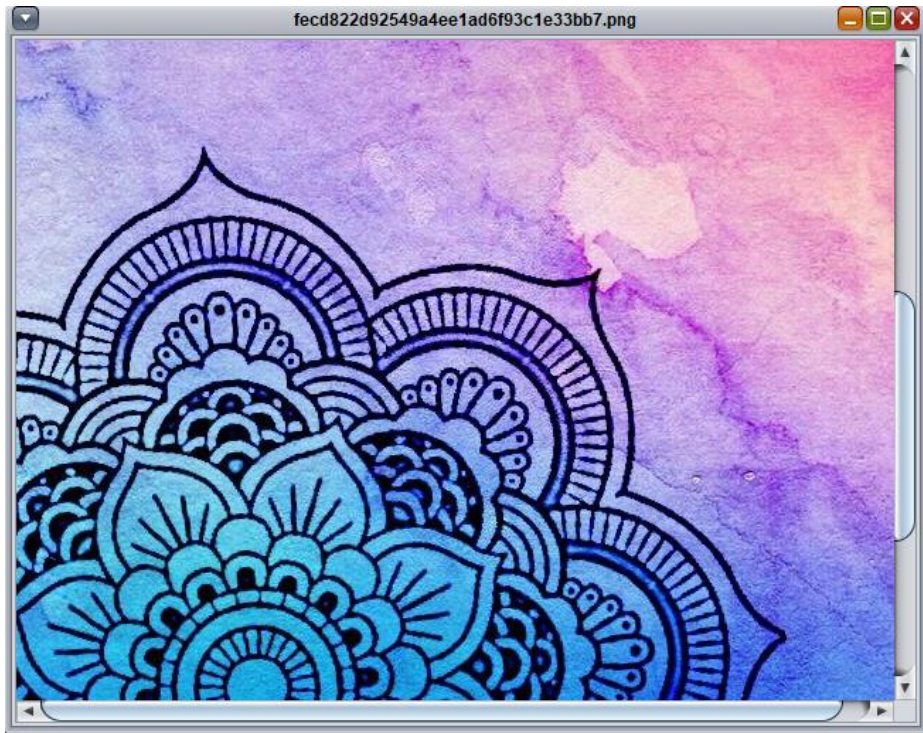




**Negativo:**

Este filtro cambia el color a su complementario en toda la imagen:

Antes:



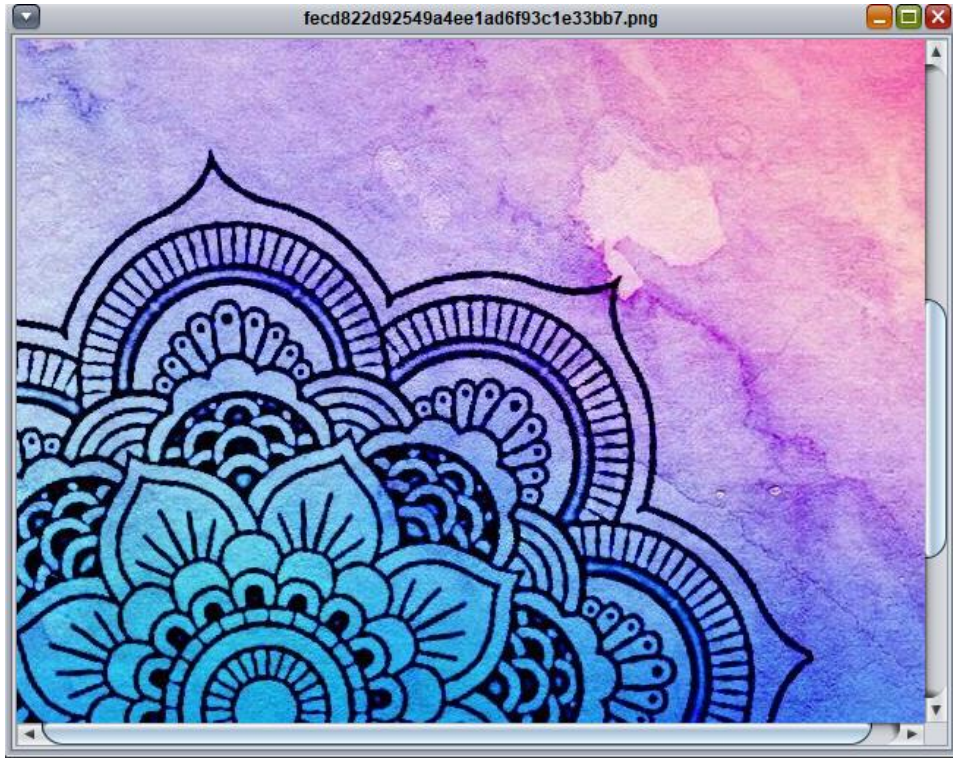
Después:



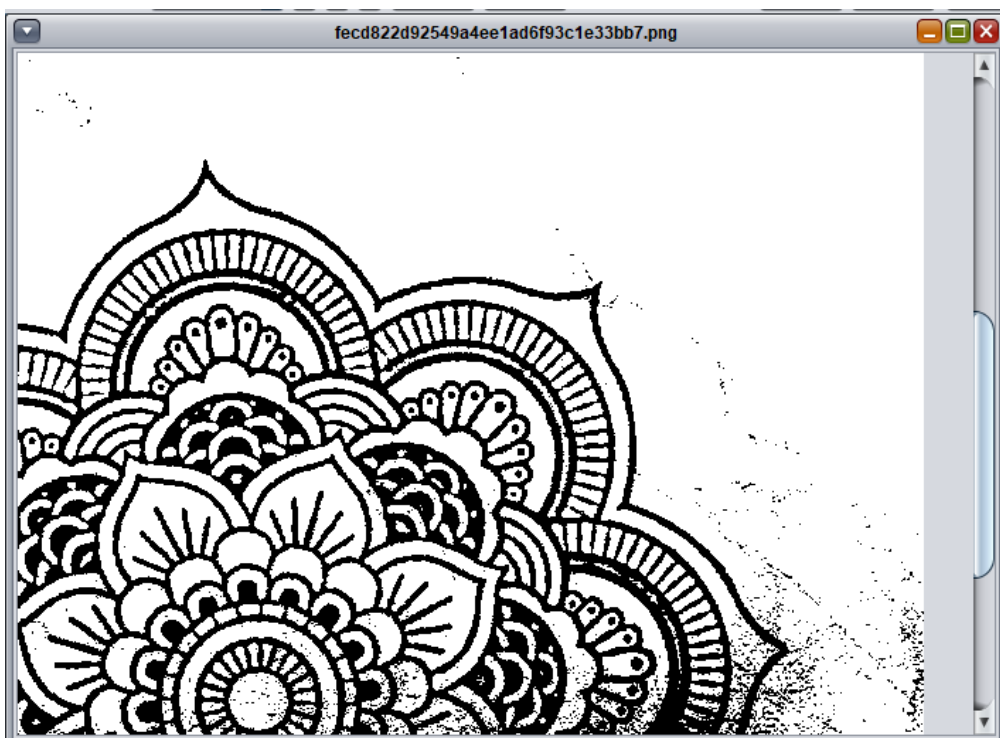
### Umbralizacion:

Dicha función utiliza un valor umbral, compara el valor con la imagen y sustituye todos los colores en blanco y negro dependiendo del umbral.

Antes:



Después:

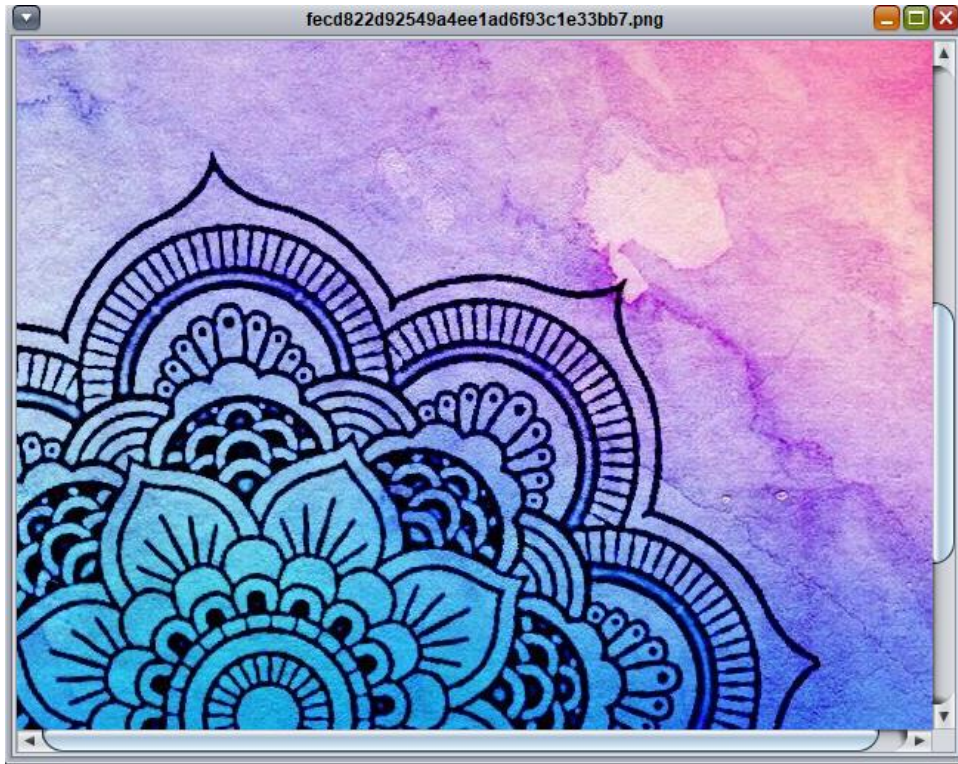




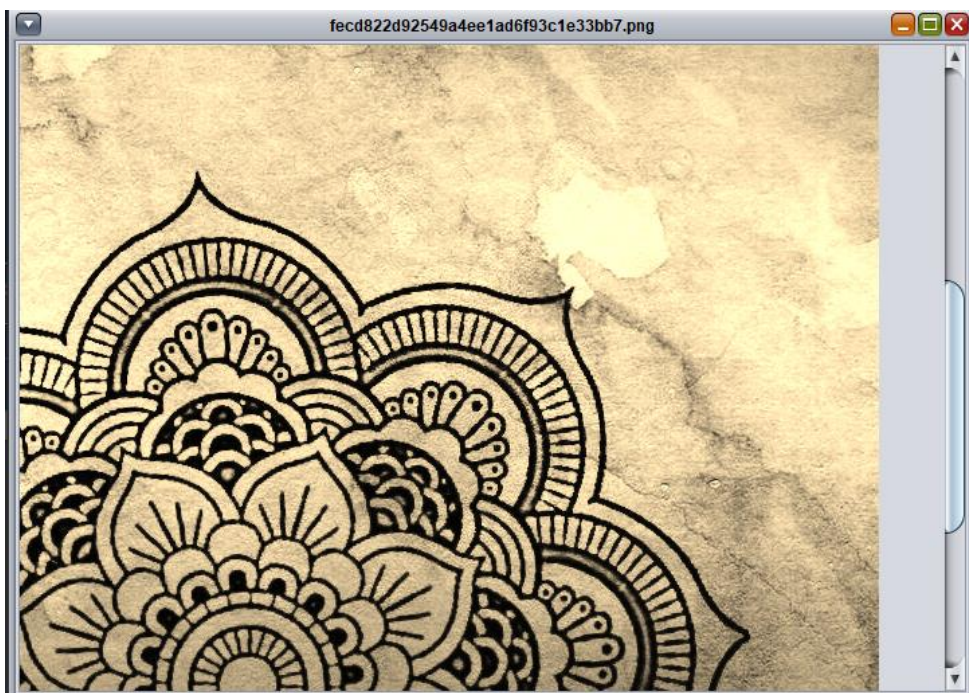
Sepia:

Dicho filtro consiste en multiplicar cada componente del pixel por un valor en concreto y su suma, guardarla en cada una de los componentes correspondientes:

Antes:



Después:

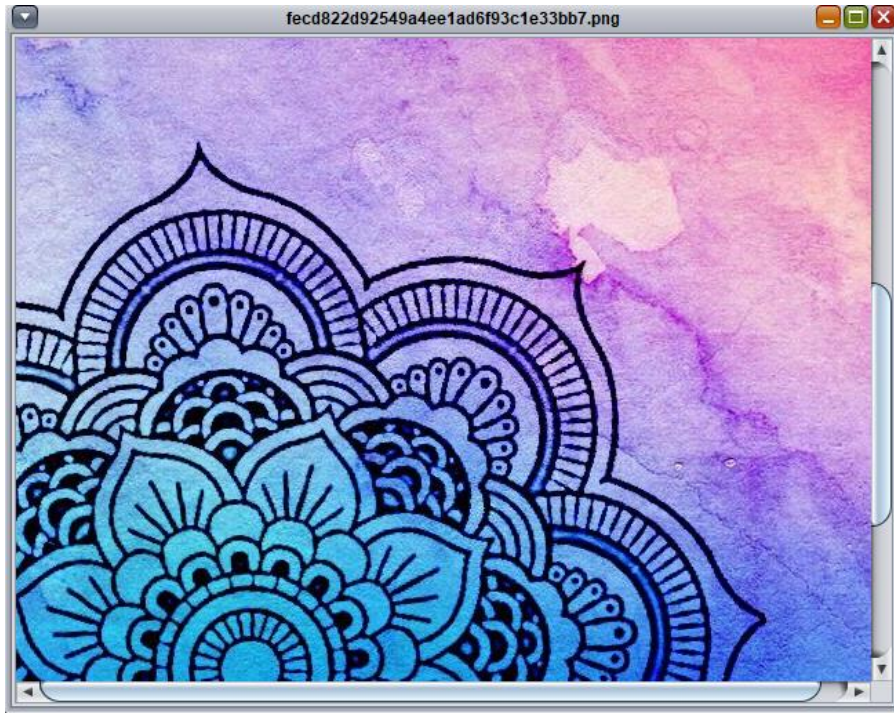




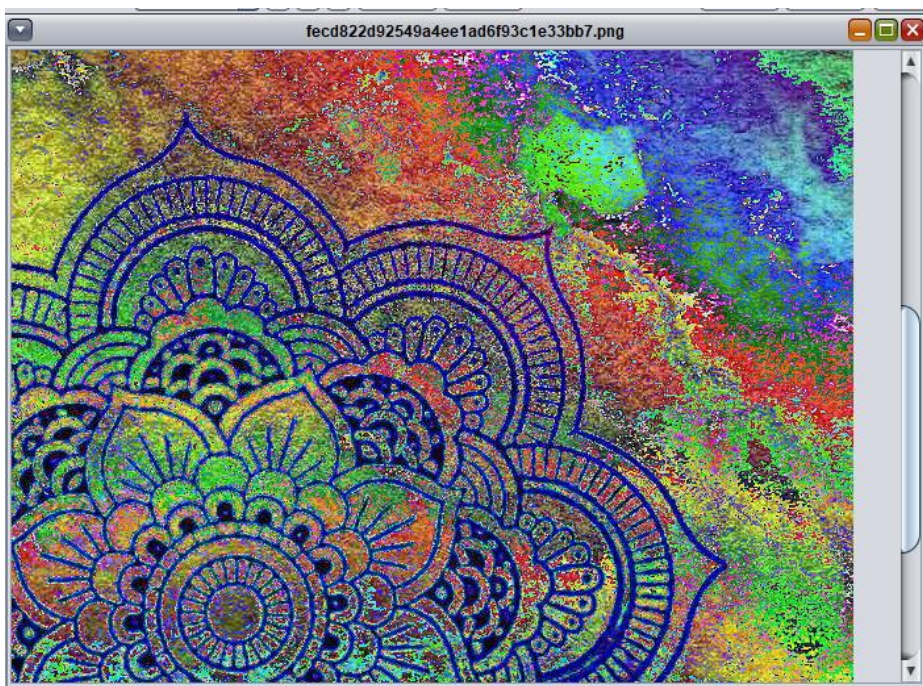
## Filtro Random(componente a componente)

Este filtro es uno de los propuestos por mí, en el cual utilizamos un numero aleatorios y lo multiplicamos por 3.5, a continuación se le suma el valor del pixel.

Antes:



Después:

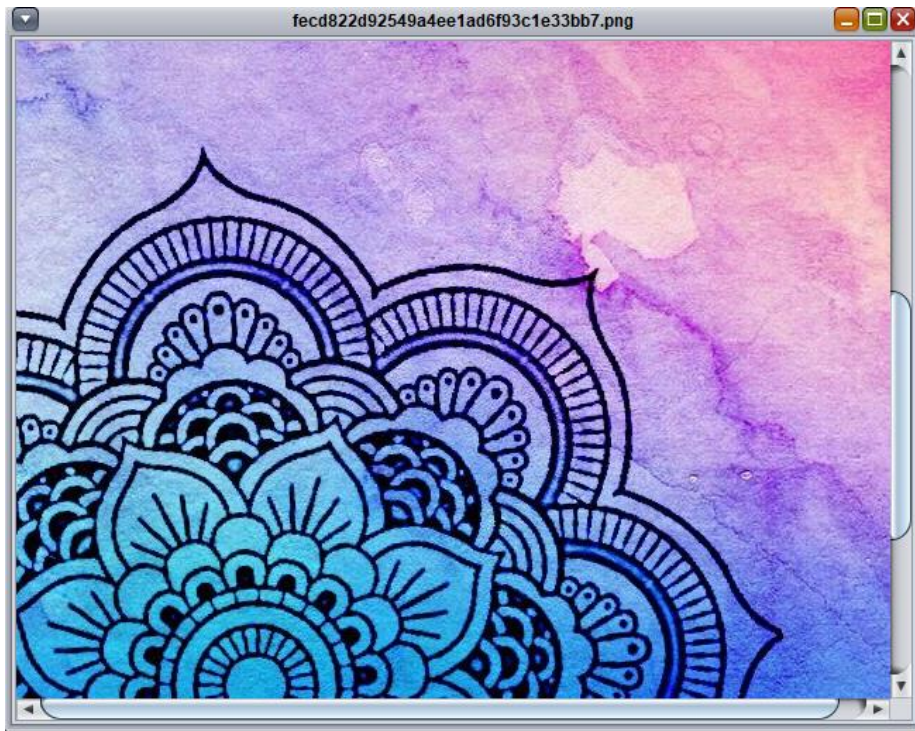




### Filtro Bosque (Pixel a pixel)

Este filtro es el segundo propuesto por mi, en el cual se busca el valor mas alto de todos los componentes de un pixel y se intercambian por el valor del Green.

Antes:

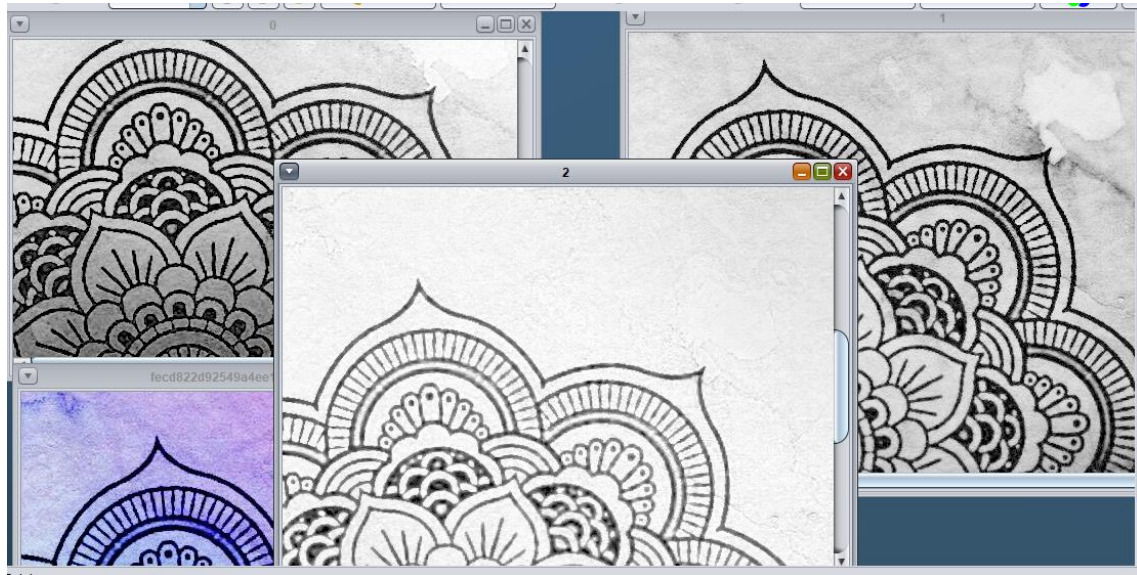


Despues:



### Extracción de bandas:

Con dicha función extraemos las bandas de color de una imagen y creamos una nueva imagen con niveles de gris de cada respectiva banda.



Adicionalmente:

Tenemos un Slider de tintado (1-100) que nos deja seleccionar el nivel de inclusión del color seleccionado en la imagen seleccionada una vez aplicado nuestro filtro de tintado mediante el color relleno.

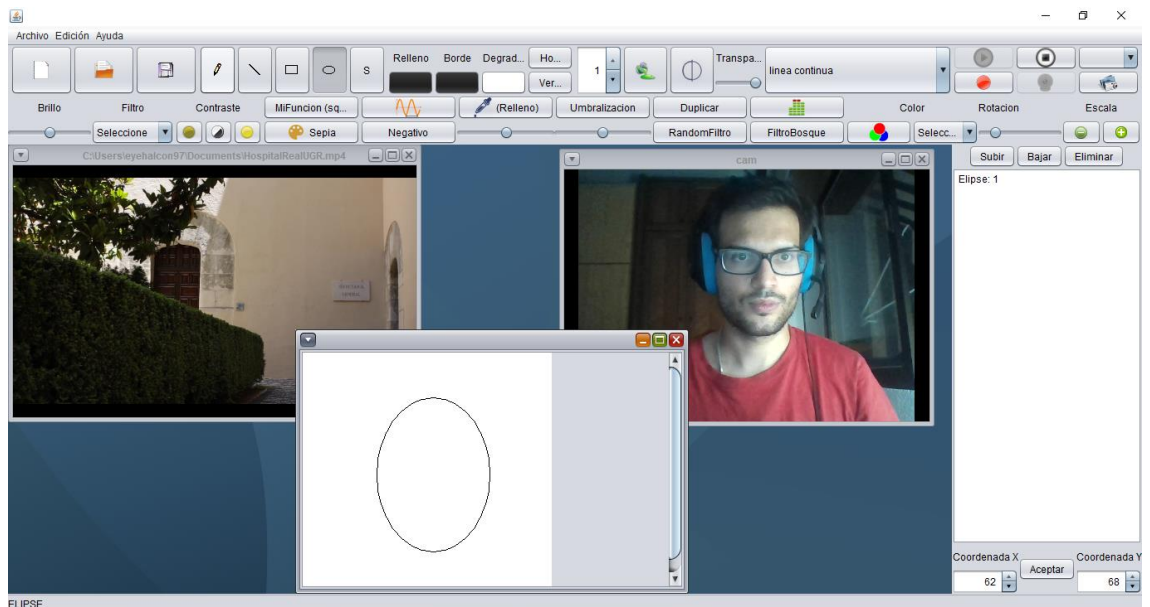
## Audio y Video:

---



Esta imagen es nuestro controlador de audio y video, con dichos botones podemos gestionar:

- Reproducir un audio o video seleccionado.
- Grabar un nuevo clip de audio.
- Detener la reproducción tanto de video como de audio
- Iniciar la webcam
- Hacer una captura de la imagen tanto del video como de la Cámara.



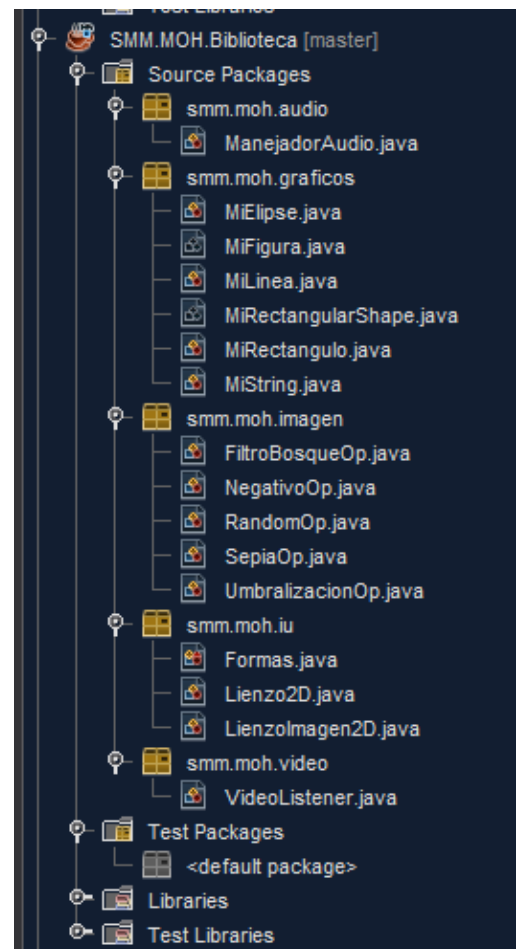
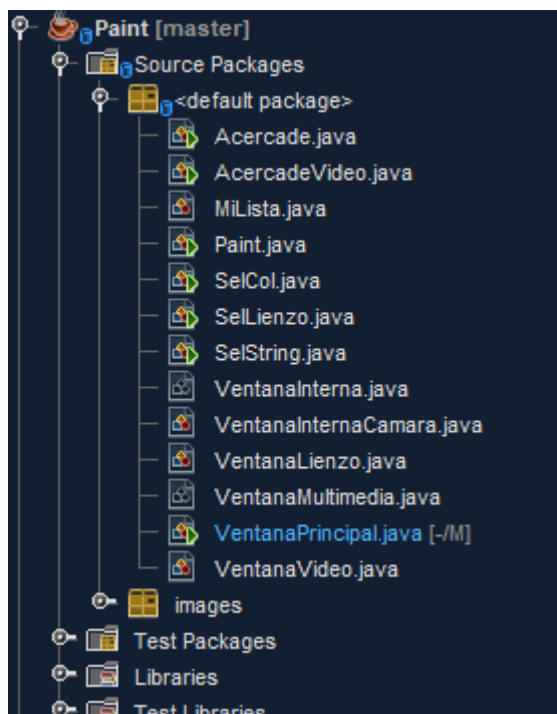


## Archivos:

---

En nuestra aplicación, los ficheros están divididos en dos grandes paquetes:

- el paquete Paint contiene la ventana principal y las ventanas internas, así como todo el resto de ventanas utilizadas para facilitar el uso de nuestro programa como el selector de color o el selector del tamaño del lienzo.
- Diferentes tipos de ventanas para gestionar el video y la cámara.
- el paquete SMM.MOH.biblioteca contiene una gran lista de clases propias creadas para la implementación de diferentes tipos de figuras, diferentes tipos de filtros, el manejador de nuestro audio y el manejador de video. En dicha biblioteca también viene el apartado del lienzo con el que podremos dibujar o cargar imágenes y editarlas.



## Sistema Propio de clases:

---

En nuestro programa hemos implementado nuestro propio sistema de clases, empezaremos describiendo nuestra clase Mifigura:

### Class MiRectangulo

```
java.lang.Object
  smm.moh.graficos.MiFigura
    smm.moh.graficos.MiRectangularShape
      smm.moh.graficos.MiRectangulo
```

Con este dibujo podremos ver mejor como está formada, para el caso de los rectángulos y las elipses tenemos una clase abstracta intermedia llamada MiRectangularShape en cuya clase están las propiedades en común, a su vez esta clase hereda de MiFigura en cuya clase abstracta contiene todos los atributos y métodos que tienen en común todas las figuras.

También hemos implementado una jerarquía de clases de diferentes tipos de ventana interna de la aplicación.

### Class VentanaInternaCamara

```
java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JInternalFrame
          VentanaInterna
            VentanaMultimedia
              VentanaInternaCamara
```

All Implemented Interfaces:

Tenemos una clase abstracta ventana interna de la que heredan todas las demás ventanas.

Tenemos otra clase abstracta ventana multimedia que implementa las cosas en común entre la ventana interna del clip de video y la ventana interna de la webcam.

Para acabar tenemos la clase Lienzolmagen2D:

## **Class Lienzolmagen2D**

```
java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.JPanel
          smm.moh.iu.Lienzo2D
            smm.moh.iu.Lienzolmagen2D
```

Dicha clase hereda de Lienzo 2D, en lienzo2D tenemos todo lo necesario para gestionar el lienzo simple y las figuras, en lienzolmagen2D tenemos todo lo necesario para gestionar las imágenes y los filtros o modificaciones de imágenes.

## **Bibliografía**

---

- Documentación de Java:

<https://docs.oracle.com/javase/7/docs/api/>

- Algunos ejemplos o solución de errores:

<https://es.stackoverflow.com/questions/195350/qu%C3%A9-es-el-stackoverflowerror-en-java>