



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA Y
TELECOMUNICACIONES

SISTEMAS MULTIMEDIA
PRÁCTICA FINAL

Realizado por:
Pedro Miguel Parrilla Navarro

Contents

1	Introducción	2
2	Entorno General	3
2.1	Requisitos Funcionales	3
2.2	Diseño	4
3	Gráficos	5
3.1	Requisitos funcionales	5
3.2	Diseño	7
3.3	Explicación de la interfáz general de gráficos	8
4	Imágenes	10
4.1	Requisitos funcionales	10
4.2	Diseño	11
4.3	Muestra de los diferentes efectos	12
4.4	Funciones propias	13
5	Sonido y Vídeo	14
5.1	Requisitos funcionales	14
6	Codificación	15
7	Bibliografía	15

1 Introducción

El trabajo a realizar consiste en una mezcla de diferentes aplicaciones cómo un paint, un editor de imágenes básico orientado a la aplicación de filtros y un reproductor de audio.

Este se ha realizado con el lenguaje Java, utilizando Swing para la interfaz, y otras bibliotecas que se mencionarán conforme se van explicando las diferentes funciones disponibles.

Para tener una imagen de como es el programa, a continuación adjunto una captura del programa a primera vista.



Algunas funcionalidades pueden deducirse de que van a realizar, pero en esta documentación voy a explicar que hace cada cosa, analizando los requisitos que cumple y parte de su diseño.

2 Entorno General

El entorno de trabajo ha sido diseñado utilizando Java Swing con el programa Netbeans por su facilidad de arrastrar y colocar, ya que gracias a ello se realiza un aprendizaje más rápido de lo que se está haciendo.

Como se ha visto en la imagen de la introducción, la ventana principal contiene todas las herramientas de trabajo para la parte de gráficos, imágenes y sonido.

2.1 Requisitos Funcionales

Los requisitos funcionales del entorno son:

- RF1. En la ventana general se podrán gestionar las diferentes ventanas internas.
- RF2. Según que archivo sea abierto, se asignará un tipo de ventana diferente.
- RF3. Se puede añadir nuevos lienzos de dibujo, indicando el tamaño de estos en la ventana mostrada.
- RF4. Al pulsar el botón de abrir se mostrará una ventana para seleccionar el archivo que se desea abrir.
- RF5. Al pulsar el botón de guardar se mostrará una ventana para guardar el archivo activo en ese momento.
- RF6. Se pueden abrir imagenes para su visualización y edición, que serán mostradas en una ventana nueva.
- RF7. Se pueden abrir archivos de sonido para su reproducción, que serán añadidos a la lista del apartado de sonido.
- RF8. Se pueden ocultar las diferentes barras de estado seleccionando en los items del apartado Ver en la barra superior.
- RF9. Al darle a Ayuda se visualizará una ventana con la información del creador del proyecto.

3 Gráficos

Esta parte engloba a todo lo relacionado con dibujar, es decir, desde pintar una línea con sus atributos de color del trazo, a seleccionar una de las figuras mostradas en la lista y modificar sus propiedades.

Para la realización de esta parte he utilizado bibliotecas de gráficos implementadas en Java como Shape y Graphics como principales.

Como tarea para el proyecto, había que realizar clases propias para las figuras, debido a que las bibliotecas que tiene Java no permiten de una manera cómoda el uso de definir una forma (un rectángulo) y sus propiedades (color rojo, relleno, degradado...) independientemente.

Para ello, he definido 2 clases principales:

- TrazoPPN. Agrupa todas las propiedades de dibujo comunes de los diferentes objetos
- ShapePPN. Engloba todo lo relacionado con las figuras. Es una clase abstracta de la cual heredan las demás formas. Incluye a la clase anterior para manejarlo como una única cosa cada objeto.

3.1 Requisitos funcionales

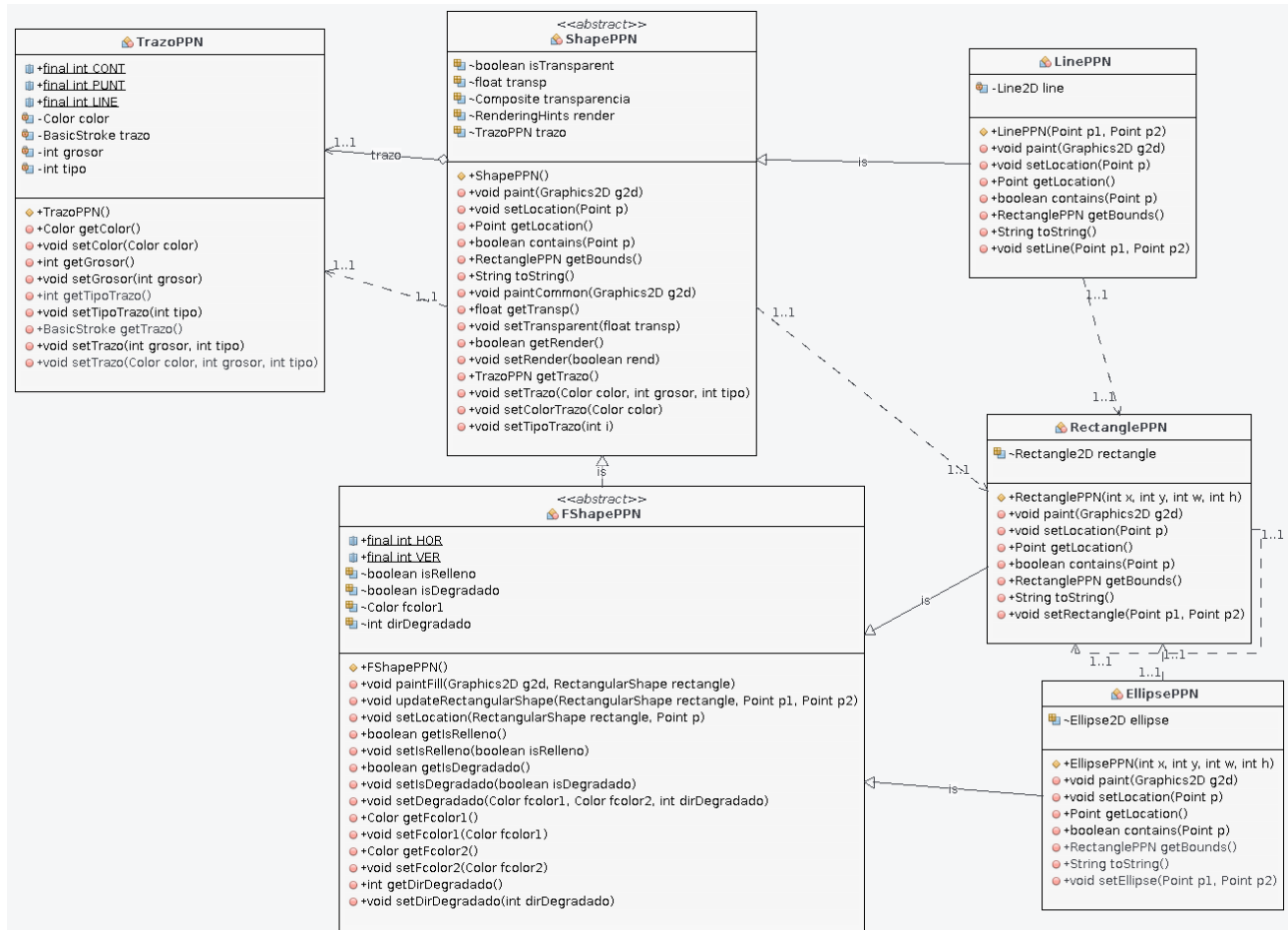
Los requisitos funcionales de gráficos son los siguientes:

- RF1. En el lienzo se dibujará la figura correspondiente según el botón seleccionado
- RF2. En el lienzo cuando se pinta la figura, esta debe visualizarse con los parámetros seleccionados, como el color del trazo, si tiene relleno, renderizado...
- RF3. La figura al terminar de dibujarla se añadirá a la lista de la ventana principal para poder seleccionarla.
- RF4. Al seleccionar una figura de la lista, debe visualizarse un rectángulo que la rodeará para que al usuario le sea más fácil. Además deberá deseleccionarse el botón de figura, como que en la barra de herramientas se activen las propiedades correspondientes a esta.
- RF5. La figura podrá cambiarse de atributos una vez seleccionada, como la traslación, el color del trazo, la transparencia...
- RF6. Al seleccionar algún botón de color aparecerá una ventana desplegable para seleccionar el color, que este será asignado a la figura seleccionada, y si no la hubiese, sería el color para la siguiente figura a dibujar.
- RF7. Al seleccionar el botón de translación cuando una figura de la lista esté seleccionada, se mostrará una ventana para poder asignarle una nueva posición a la figura
- RF8. Al seleccionar el panel de figuras, se mostrará una lista con las que aparecen en el lienzo.

- RF9. Al seleccionar el panel de tipo de trazo, se nos mostrará las 3 opciones disponibles: continuo, punto y línea
- RF10. Al seleccionar o asignar render, relleno, grosor, transparencia o dirección de degradado, se establecerá para dibujar la siguiente figura. Si en cambio hay ya una seleccionada, se le asignarán los parámetros a esa.
- RF11. Al cambiar entre ventanas de dibujo, en la barra de herramientas se activarán los parámetros usados por última vez en ellos, además de la actualización de la lista de figuras disponibles.

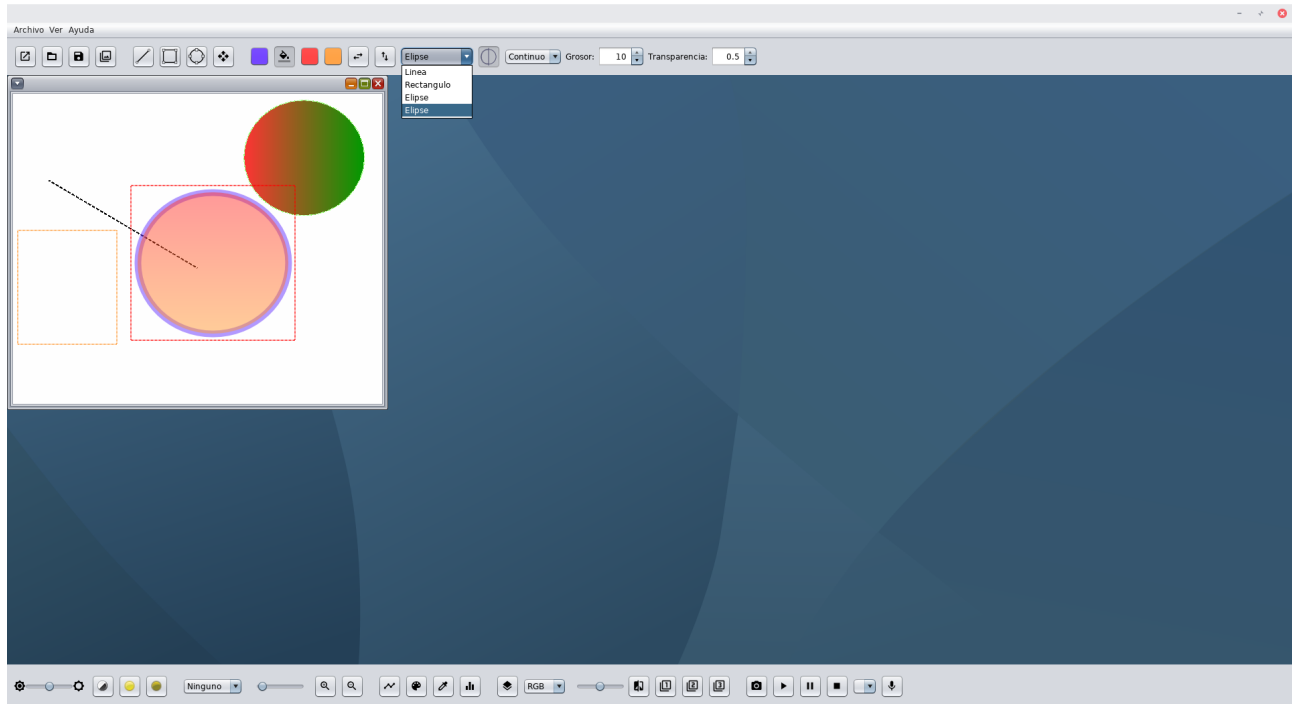
3.2 Diseño

A continuación adjunto los diagramas correspondientes a las diferentes clases de la biblioteca propia de figuras.



3.3 Explicación de la interfáz general de gráficos

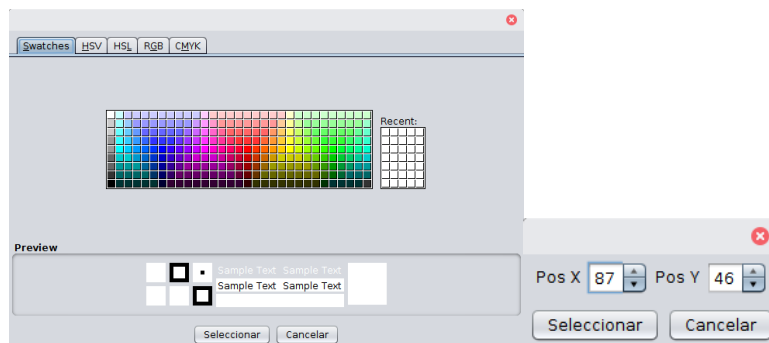
En este apartado voy a situar las acciones de las que disponemos. En la siguiente imagen podemos observar la interfaz:



Para esto, solo vamos a fijarnos en los botones que aparecen arriba, los cuales comentaré de izquierda a derecha:

- Nueva ventana. Crea un nuevo lienzo donde dibujar.
- Abrir archivo. Permite abrir un documento compatible con el programa, como imágenes, sonido o vídeo.
- Guardar archivo. Guarda el archivo seleccionado activado.
- Duplicar imagen. Permite hacer una copia de la imagen actual.
- Línea. Establece la línea como figura.
- Rectángulo. Establece el rectángulo como figura.
- Elipse. Establece la elipse como figura.
- Traslación. Abre una ventana para asignar una nueva posición a la figura seleccionada.
- Color Trazo. Abre una ventana para asignar el color del trazo.

- Relleno. Activa o desactiva el relleno para pintar.
- Color Degradado 1. Abre una ventana para asignar el color del degradado 1.
- Color Degradado 2. Abre una ventana para asignar el color del degradado 2.
- Degradado Horizontal. Establece el degradado en horizontal.
- Degradado Vertical. Establece el degradado en vertical.
- Lista Figuras. Muestra las figuras dibujadas en el lienzo.
- Render. Activa o desactiva el render para pintar.
- Lista trazo. Muestra los 3 trazos disponibles: continuo, punto y línea.
- Grosor. Establece el grosor de la figura.
- Transparencia. Establece la transparencia. 1 = sólido, 0 = transparente.



Las imágenes anteriormente mostradas son las correspondientes a la ventana de colores y a la ventana de traslación.

4 Imágenes

El programa realizado además de la parte de gráficos ya vista tiene otras dos más, una de ellas imágenes que es la que se va a exponer en esta sección. Esto abarca todos los efectos disponibles a utilizar con las imágenes, como las diferentes operaciones de rotación u obtención de capas.

Para la realización de esta parte en su mayoría he utilizado bibliotecas que nos ha facilitado el profesor de la asignatura para la realización del trabajo. Pero el extra es que tenemos que definir 3 efectos propios utilizando el operador Lookup y dos clases, una de ellas para operaciones pixel a pixel y la otra componente a componente.

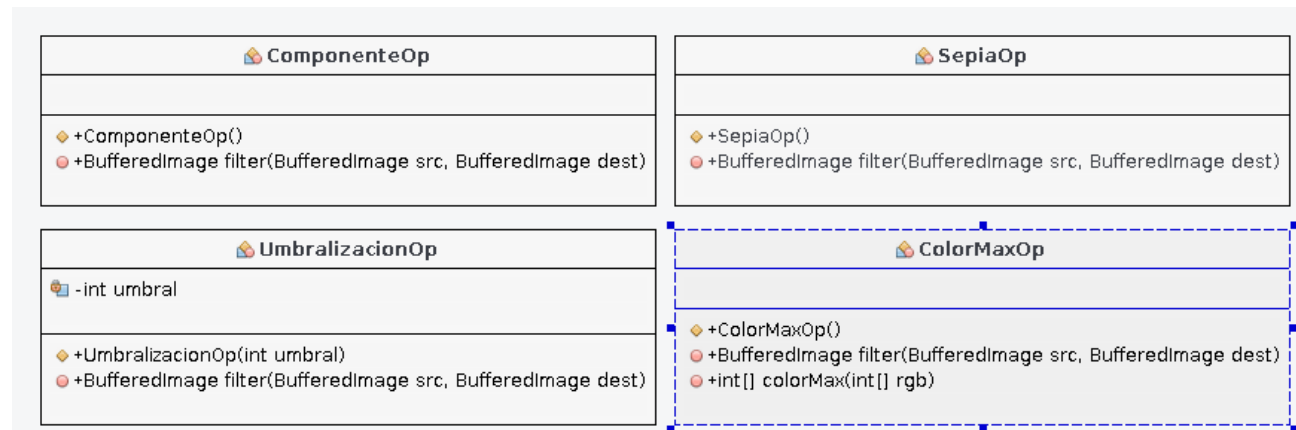
4.1 Requisitos funcionales

Los requisitos funcionales de las imágenes son los siguientes:

- RF1. Al desplazar el deslizador de brillo, se cambiará el brillo de la imagen actual.
- RF2. Al seleccionar uno de los 3 botones de contraste, a la imagen se le aplicará la función S, las partes oscuras aumentaran su brillo o las partes claras disminuirán la suya.
- RF3. Al seleccionar uno de los distintos efectos (Media, Binomial, Enfoque, Relieve, Fronteras o Ninguno) se aplicará a la imagen. Si se pulsa ninguno la imagen vuelve al estado anterior antes de seleccionar alguno de esos filtros. Eso si, si no ha perdido el foco, es decir, que no se haya realizado otra operación.
- RF4. Al desplazar el segundo deslizador la imagen rotará en el panel.
- RF5. La imagen se podrá escalar al pulsar cualquiera de los dos botones de lupa
- RF6. Se podrá aplicar la función seno a la imagen.
- RF7. Se podrá aplicar el efecto sepia a la imagen.
- RF8. Se podrá aplicar un tinte de color a la imagen según el color de trazo seleccionado.
- RF9. Se podrá aplicar una ecualización de color a la imagen.
- RF10. Al pulsar un botón se obtendrán las diferentes bandas.
- RF11. Se podrá cambiar la imagen entre RGB, YCC y GREY.
- RF12. Se dispondrá de un deslizador para aplicar una umbralización a la imagen.
- RF13. Se podrá aplicar el efecto negativo a la imagen.
- RF14. Se podran aplicar 3 efectos realizados por el alumno.

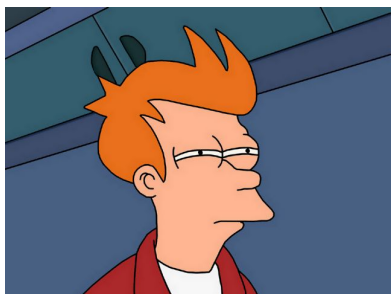
4.2 Diseño

A continuación adjunto los diagramas correspondientes a las diferentes clases propias:



4.3 Muestra de los diferentes efectos

En las siguientes imágenes se podrán observar los diferentes efectos disponibles con aplicados en nuestro gran modelo Fry:



(a) Fry



(b) Seno



(c) Sepia



(a) Tinte Rojo



(b) Ecuilización



(c) Negativo



(a) Operación 1



(b) Operación 2



(c) Operación 3

4.4 Funciones propias

Para la primera función con el operador LookUpOp, he realizado un vector el cual se aplicará a todos los componentes de cada pixel. Para ello he usado una operación sencilla que es la siguiente:

$$i = (i * 2) \% 256 \quad (1)$$

Para la segunda hay que crear una clase y realizar una operación componente a componente, es decir, a cada elemento de cada pixel de manera independiente. Tras la primera, por probar a establecer como quedaría con otro producto de i , he escrito esta:

$$i = (i * 3) \% 256 \quad (2)$$

Para la tercera y última operación propia, tiene que aplicarse pixel a pixel, pudiendo combinar el valor de sus pixeles. En mi caso he calculado el máximo de los 3, y en caso de que dos o los tres coincidiesen como máximos se le aplicaba. Para ello una vez obtenido el valor del máximo, se aplicaba la siguiente condición: Si el valor es mayor de 128, se iguala a 255, si no, se iguala a 0.

Con este último efecto obtenemos imágenes con un máximo de 8 colores disponibles, que sería una imagen formada por 3 bits de colores.

5 Sonido y Vídeo

Por último, en las últimas sesiones de prácticas de la asignatura se explicó la inserción de audio y vídeo en la aplicación.

Para la realización del audio he utilizado las clases que nos facilitó el profesor: SMClipPlayer para el reproductor y SMSoundRecorder para grabar el audio.

Para la realización del vídeo se ha tenido que crear una nueva clase VentanaInterna con la biblioteca vlcj. Así que cuando se abre un vídeo se crea una nueva ventana donde se reproduce. Además de esas 2, había que añadir 3 más para la reproducción de vídeo, adjuntas con el programa.

5.1 Requisitos funcionales

Los requisitos funcionales de sonido y vídeo son los siguientes:

- RF1. Al abrir un archivo de sonido, se añadirá a la lista de sonido del programa
- RF2. Al abrir un archivo de vídeo, se creará una ventana donde se visualice.
- RF3. Al pulsar el play, si hay una ventana de vídeo activa este comenzará a reproducirse. Si no la hay, se reproducirá el audio de la lista si existe.
- RF4. Al pulsar el stop, detendrá la reproducción del video si está activo. Si no, la del audio si lo está.
- RF5. Al pulsar el botón de grabación, el dispositivo comenzará a grabar el audio.
- RF6. Al pulsar el botón de detener grabación, mostrará una ventana donde guardar el audio grabado. Después se añadirá a la lista de sonido.
- RF7. Al pulsar el botón de cámara, se activará la cámara del dispositivo y se visualizará en una nueva ventana interna.

6 Codificación

Respecto al código, va incluido junto a este documento, además he utilizado javadoc para documentar las clases propias situadas en SM.PPN.Biblioteca, exponiendo como funcionan y todos los métodos.

7 Bibliografía

He consultado de diferentes páginas webs, como información facilitada por el profesor de la asignatura y los guiones de prácticas, pero principalmente ha sido:

- [Documentación de Oracle](#)
- [JavatPoint](#)
- [StackOverFlow](#)