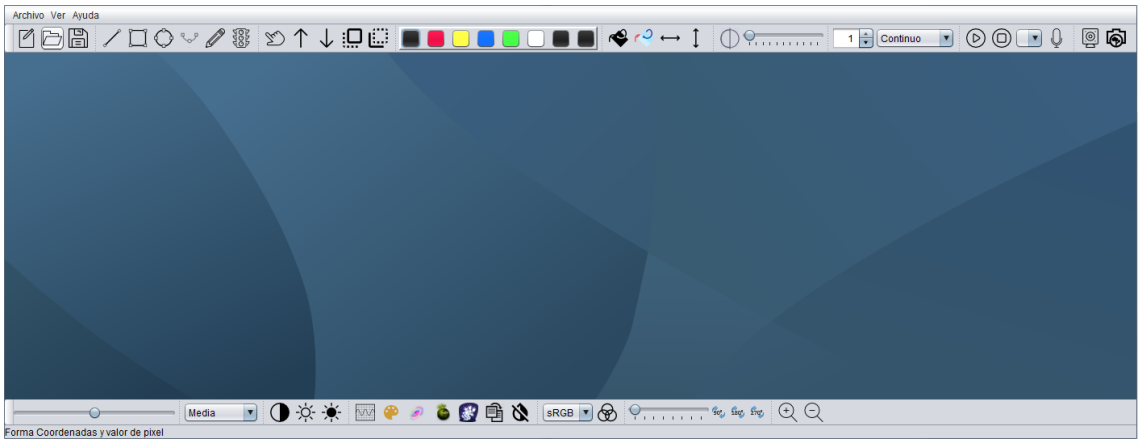
PROYECTO DE EVALUACIÓN

Sistemas multimedia

Carlos Ariza García - Curso 2017-2018



Contenido

Descripción de la aplicación

A lo largo del curso se ha realizado una serie de prácticas en la asignatura Sistemas Multimedia que engloban los conocimientos a adquirir de la asignatura.

Como evaluación final se realiza esta aplicación multimedia que recoge todas las funcionalidades realizadas en prácticas más algunas añadidas además del diseño propio de las clases necesarias para la realización de la misma.

Esta aplicación multimedia incluye un entorno basado en escritorio que incluye diferentes ventanas en las que se pondrán realizar diferentes funcionalidades cómo:

- Dibujo en un lienzo
- Procesamiento de imágenes
- Reproducción de sonido, video y WebCam

Todo el código realizado en este proyecto está incluido en una biblioteca que recopila todas las clases propias creadas por el estudiante, junto al proyecto que se encarga de ejecutar la aplicación. En la fase de diseño se explicarán a fondo todas las clases de la biblioteca.

Requisitos funcionales

La aplicación trata con cuatro medios, de los que se describirán los requisitos:

Gráficos

RF-1 Dibujo en lienzo

El usuario podrá dibujar sobre el lienzo utilizando las formas y atributos de la barra de tareas de la aplicación

> **RF-1.1 Persistencia de las formas**: El lienzo mantendrá todas las figuras que se vayan dibujando
>
> **RF-1.1.1 Atributos propios de una forma**: Cada figura tendrá sus propios atributos, independientes del resto de formas.

RF-1.2 Formas

> El usuario podrá dibujar usando las siguientes formas: línea, rectángulo, óvalo, curva con punto de control, trazo libre y área
>
> **RF.1.2.1 Atributos de la forma:** Los atributos que se podrán aplicar a la forma son: color, relleno, color del relleno, grandiente del relleno, tipo de grandiente (en caso de figuras rellenas), alisado de formas, transparencia, grosor y tipo de grosor. (SEPARA CADA ATRIBUTO EN UN REQUISITO ¿=?==?)

RF-1.3 Persistencia de los atributos

Cuando se dibuje una nueva figura se utilizarán los atributos activos en el momento actual en la barra de herramientas

RF-2 Selección en el lienzo

El usuario podrá seleccionar figuras mediante el botón de selección y moverlas.

RF-3 Edición en el lienzo

El usuario podrá editar las figuras dibujadas en el lienzo, seleccionando la figura.

RF-3.1 Edición de atributos: Se podrán modificar los atributos de la figura seleccionada, es decir cambiar sus propiedades, mediante la barra de herramientas.

RF-4 Ordenación de formas

Se podrán cambiar la ordenación de las figuras mediante diferentes opciones

RF-4.1 Enviar al fondo: Situará la figura seleccionada detrás del resto de figuras.

RF-4.2 Traer al frente: Traerá la figura seleccionada delante del resto de figuras, de forma que ninguna parte quede oculta detrás de otra figura.

RF-4.3 Enviar atrás: Llevará la figura seleccionada una posición hacia atrás respecto a su ordenación actual, de forma que quede oculta por más figuras.

RF-4.4 Traer adelante: Traerá la figura seleccionada una posición hacia delante, de forma que quede oculta por menos figuras

Imágenes

RF-5 Visualización de imágenes

Visualizar imágenes en el lienzo sobre los formatos reconocidos por Java (png, jpg, jpeg), mediante el botón abrir.

RF-6 Aplicar operaciones sobre imagen

Se podrá aplicar un conjunto de operaciones sobre las imágenes establecido en la barra de herramientas inferior

RF-6.1 Duplicar: Crear una nueva ventana de imagen con una copia de la imagen actual del lienzo.

RF-6.2 Modificar el brillo Cambiar el brillo de la imagen actual del lienzo mediante un deslizador.

FR-6.2.1 Aplicación del brillo: El brillo se aplicará de forma definitiva cuando el deslizador de brillo pierda el foco.

RF-6.3 Filtros: Aplicar filtros de emborronamiento, enfoque y relieve a través de una lista desplegable.

RF-6.4 Contraste: Aplicar contraste normal, iluminado y oscurecido mediante los botones establecidos.

RF-6.5 Negativo: Aplicar el filtro negativo (invertir colores).

RF-6.6 Extracción de bandas: Extracción de las bandas correspondientes al espacio de color de la imagen actual del lienzo.

RF-6.7 Conversión de espacio de color: Convertir el espacio de color de la imagen actual del lienzo a los espacios RGB, YCC y GRAY.

RF-6.8 Giro: Girar la imagen de manera libre mediante deslizador o con botones de giro de 90º, 180º o 270º.

RF-6.9 Escalado: Aumentar y disminuir la imagen con dos botones establecidos.

RF-6.10 Sepia: Aplicar el efecto sepia.

RF-6.11 Aplicar operadores personalizados: Aplicar los operadores personalizados del estudiante.

RF-6.11.1 Operación componente a componente: Una nueva operación de diseño propio aplicada componente a componente: Operación Spooky.

RF-6.11.2 Operación pixel a pixel: Una nueva operación de diseño propio aplicada pixel a pixel: Operación Musgo y Operación SimplYRGB

RF-7 Aplicación concatenada de operaciones

Las operaciones se irán aplicando de forma concatenada.

RF-8 Dibujo sobre imagen

Dibujar con todos los medios del apartado de gráficos sobre la imagen actual del lienzo.

Sonido

RF-9 Reproducción de sonidos

Reproducir sonidos mediante el botón abrir.

RF-9.1 Filtros: Se podrán reproducir sonidos sobre los formatos y códecs reconocidos por Java (mp3, wav).

RF-10 Grabación

Grabar sonidos a través del micrófono del ordenador. Al iniciar se pedirá un nombre para el fichero y este se añadirá a la lista desplegable de sonidos una vez pulsado el botón de stop.

Video

RF-11 Reproducción de videos

Reproducir videos mediante el botón abrir.

> **RF-11.1 Filtros:** Se podrán reproducir videos sobre los formatos y códecs reconocidos por Java (mp4, avi).
>
> **RF-11.2 Capturas de Video:** Realizar una captura de la imagen actual del video.

RF-12 Visualizar Cámara Web

Ejecución de la cámara web del equipo y visualización de la misma en una ventana de video.

> **RF-12.1 Capturas de Cámara Web:** Realizar una captura de la imagen actual de la Cámara Web.

RF 13 Edición de capturas

Editar las capturas realizadas por la Cámara Web o el video con los medios del apartado de imagen y gráficos.

Interfaz

RF-14 Abrir lienzo

Abrir un nuevo lienzo sobre el que se puede dibujar.

RF-15 Abrir medios

Abrir un nuevo medio.

> **RF-15.1 Abrir imagen:** Abrir una nueva imagen en una ventana de imagen.
>
> **RF-15.2 Abrir Cámara Web:** Abrir la Cámara Web en una ventana de Cámara Web.
>
> **RF-15.3 Abrir video:** Abrir un nuevo video en una ventana de video.

RF-16 Guardar medios

Guardar la imagen actual que se encuentre en el lienzo de la ventana activa.

RF-17 Seleccionar figuras

Seleccionar la figura a dibujar en el lienzo.

RF-17.1 Seleccionar atributos de figuras: Seleccionar el atributo a aplicar a la forma seleccionada o dibujada en el lienzo.

RF-18 Reproducir medios

Reproducir los sonidos agregados a la lista desplegable de sonido

RF-19 Seleccionar filtros

Seleccionar el filtro que se aplicará sobre la imagen actual del lienzo.

RF-20: Mostrar estados

Se mostrará el estado de ciertas características de la aplicación en el panel inferior.

RF-20.1 Mostrar forma: Mostrar la forma actual de dibujo

RF-20.2 Mostrar coordenadas del pixel: Mostrar las coordenadas del pixel sobre el que está el ratón.

RF-20.3: Mostrar valor del pixel: Mostrar el valor del pixel sobre el que está el ratón.

RF-21 Esconder paneles

Esconder la barra de herramientas, barra de filtros o barra de estado mediante el menú "Ver".

RF-22 Información de la aplicación

Conocer información de la aplicación mediante el menú "Acerca de".

RF-23 Cambio de lienzo

Al cambiar de un lienzo a otro los correspondientes botones de atributos se ajustarán a las propiedades de dicho lienzo

Análisis

En esta fase se va a realizar un análisis de lo que necesita cada apartado de la aplicación, de que disponemos y cómo vamos a solucionar las deficiencias:

Gráficos

En cuanto al apartado de gráficos, para realizar el dibujo de las formas seguimos utilizando los atributos que incluyen las clases de propiedades geométricas que nos proporciona Java, cómo son: Line2D, Rectangle y las demás que se verán en la etapa de diseño.

Al igual que para el dibujo, para el movimiento de las formas seguimos utilizando los métodos que nos proporcionan las propias formas de Java.

Para cambiar los atributos de las figuras seguimos utilizando también los que nos proporciona Java (Stroke, Color, etc..), que aplicaremos a los gráficos del lienzo

Para poder cumplir el requisito de independencia de los atributos entre las diferentes formas dibujadas en el lienzo, como Java solo nos permite cambiar los atributos de todos los gráficos del lienzo, lo que cambiaría los atributos de todas las formas dibujadas en este a la vez, por lo que es necesario crear una jerarquía de clases de formas propia:

Se desarrolla una jerarquía de clases propia en la cual encontramos una súper clase, que contiene los atributos comunes a todas las formas, de la que heredarán todas las clases propias de formas, además existirá una clase intermedia para diferenciar las figuras que tienen relleno y las que no.

Estas clases propias implementan un método propio de dibujo, que les permitirá aplicar cuando se vayan a dibujar sus propios atributos, de manera que cada clase podrá mantener sus atributos independientemente de las demás formas dibujadas.

En la etapa de diseño veremos en profundidad el desarrollo de la jerarquía de clases y de las clases.

Imágenes

En cuanto al apartado de imágenes, para la creación, lectura y escritura de imágenes utilizamos la clase que nos proporciona el paquete de imagen de Java, *BufferedImage*, con la que podremos visualizar todas las imágenes compatibles en formato y códec de Java.

Para el procesamiento de imágenes utilizamos las operaciones que nos proporciona java para actuar sobre la imagen.

- Para las operaciones de brillo se utiliza la operación *RescaleOp*
- Para las operaciones de filtros se utiliza la operación *ConvolveOp*
- Para las transformaciones de escalado y rotación utilizamos *AffineTransformOp*
- Para las operaciones de contraste y seno utilizamos la operación LookupOp
- La operación Sepia se realiza con una nueva operación realizada en prácticas.
- Para las operaciones de diseño propio se han diseñado clases propias que definen la operación, que se describirán en la fase de diseño.

Sonido

Para el apartado de sonido, se utiliza la *Java Sound API*, que incluye todo lo necesario para cumplir los requisitos de la aplicación, es decir, reproducción de sonidos. Todas las clases necesarios para la realización de este apartado están incluidas en la biblioteca *sm.sound* proporcionada por la asignatura, incluida en la biblioteca propia del estudiante.

Video

Para la realización del apartado de video, tenemos dos aproximaciones, podríamos realizarlo con la API del *Java Multimedia Framework*, pero esta API no es muy fácil de gestionar, por lo que nos decidimos por usar bibliotecas de código abierto de funcionamiento más fácil y funcional, como son:

- *Webcam Capture API:* Para la visualización de la cámara web
- *VLCj*: para la reproducción de videos

Diseño

Diseño de clases

En cuanto al diseño de la jerarquía propia de clases, se ha realizado una súper clase llamada *MiForma,* de la que heredan todas las formas propias. Esta clase contiene los atributos comunes a todas las formas: color, trazo, transparencia y alisado. Además, para las formas con relleno existe una clase intermedia, *MiFormaConRelleno*, contiene los atributos de las formas con relleno, como son el color del relleno, el grandiente, el color del grandiente y el tipo de grandiente.

Cuatro clases heredan directamente de MiForma, las que no tienen relleno:

- *MiLinea*
- *MiCurva*
- *MiTrazoLibre*
- *MiArea*

Las dos formas restantes tienen relleno:

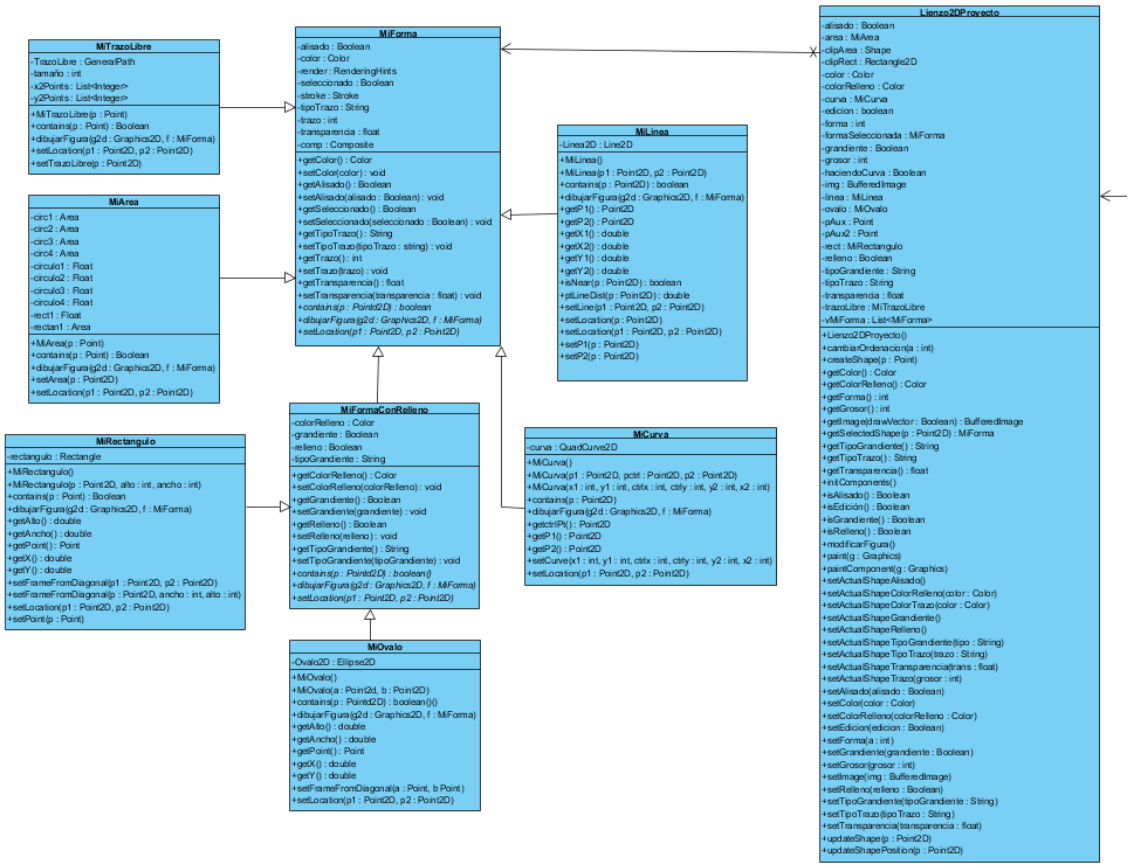
- *MiRectangulo*
- *MiOvalo*

La clase MiForma tiene tres métodos abstractos: dibujarFigura, contains y setLocation, que estas clases hijas sobrecargan cada una a su manera. Sobre estas clases, para poder utilizar la geometría que nos proporciona Java, tienen como atributo un objeto del tipo de geometría de Java correspondiente:

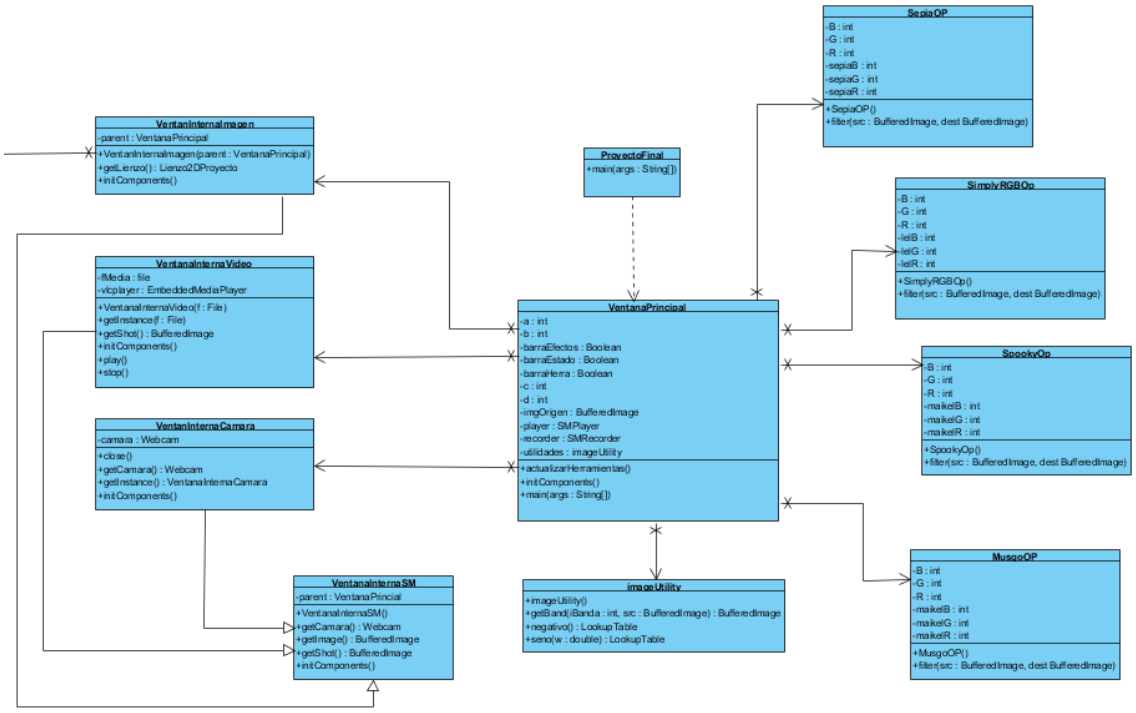
- *MiLinea* tiene un objeto *Line2D*
- *MiRectangulo* tiene un objeto *Rectangle*

- *MiOvalo* tiene un objeto *Ellipse2D*
- *MiCurva* tiene un objeto *QuadCurve2D*
- *MiTrazoLibre* tiene un objeto *GeneralPath*
- *MiArea* tiene varios objetos *Area* y las correspondientes geometrías

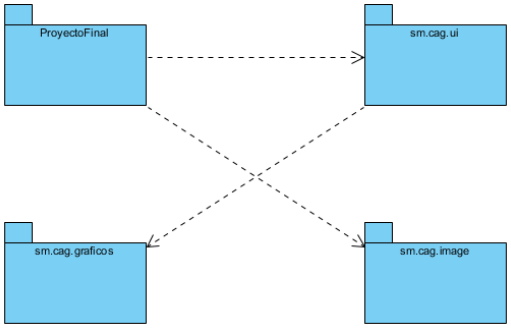
Ahora se presenta el correspondiente diagrama de clases del proyecto (segunda parte en la siguiente página):



Se aclara que existe una relación entre Lienzo2DProyecto y VentanaInternaImagen.



Y además se proporciona el diagrama de paquetes:



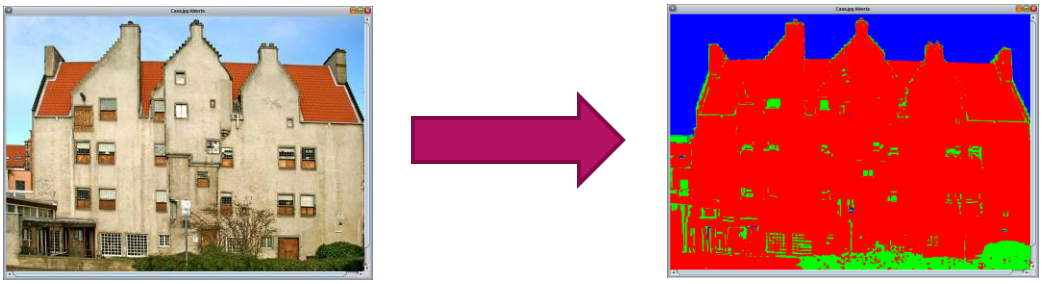
Para la realización de los diagramas se ha utilizado Visual Paradigm.

Operaciones propias

- Operación SimplyRGB: Esta operación recorre pixel a pixel la imagen, y en cada pixel identifica que componente es mayor, y establece el valor de ese componente al máximo y el de los demás al mínimo (0).
  La formulación matemática de la función sería la siguiente, siendo R, G y B los componentes del pixel:

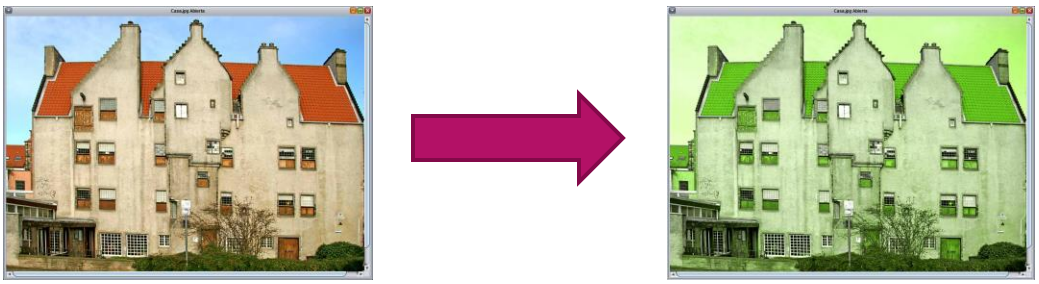
  Para R > G y R > B ⟶ R = 255 B = 0 G = 0
  Para R > G y R > B ⟶ R = 0 B = 255 G = 0
  Para R > G y R > B ⟶ R = 0 B = 0 G = 255

El efecto sería el siguiente:



- Operación Musgo: Esta operación recorre pixel a pixel la imagen, y en cada pixel aplica la siguiente ecuación:
  R = (R + G + B) / 3
  G = máximo(máximo(R, G), máximo(R, B))
  B = mínimo(mínimo(R, G), mínimo(R, B))
  De manera que al aplicarse esta ecuación se obtendrán tonos verdosos, razón por la que la operación se titula Musgo. Los tonos verdosos se deben a que la componente roja siempre divide entre tres y la componente azul siempre escoge el mínimo, a diferencia de la componente verde que siempre escoge el máximo, por eso se notan tonos verdosos.

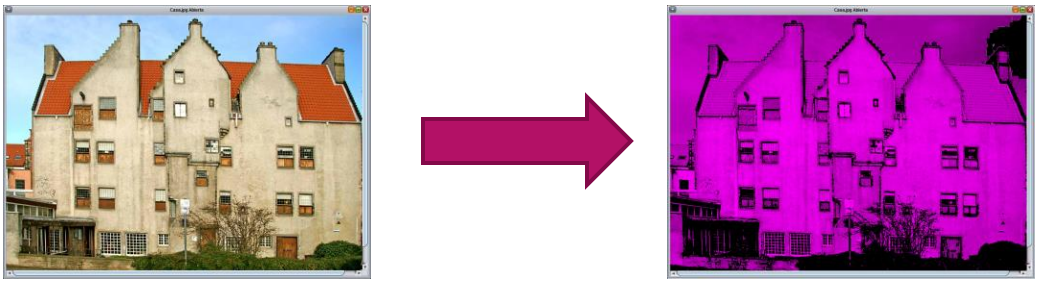
  El efecto sería el siguiente:



- Operación Spooky: Esta operación recorre cada pixel de la imagen, componente a componente, aplicando en cada componente respectivamente estas operaciones:
  Para la componente roja:
  Si R < 127 → R = arcoCoseno(R)
  Si G > 127 → G = arcoTangente(G)
  Si B < 127 → B = raizCúbica(B)
  En el caso de la componente roja, si se aplica, si el valor es 0 aplicará 90, y si el valor es 1 aplicará 0, en otro caso 0.
  En el caso de la componente verde, si se aplica, si el valor es 0 aplicará 0, y si el valor es 1 aplicará 45.
  En el caso de la variable azul, se realiza la raíz cúbica, por lo que como mucho con valor 255 se aplicaría 6.
  Viendo los resultados, se entiendo el por qué de los oscuros colores que se generan, ya que se obtienen valores muy bajos de los componentes. Y la razón del tono púrpura es debida a que este lo encontramos cuando tenemos bastante componente roja y azul y baja verde, que es el caso que

ocurre con esta operación, ya que el arco tangente nos va a dar valores de verde bajos.

El efecto sería el siguiente:

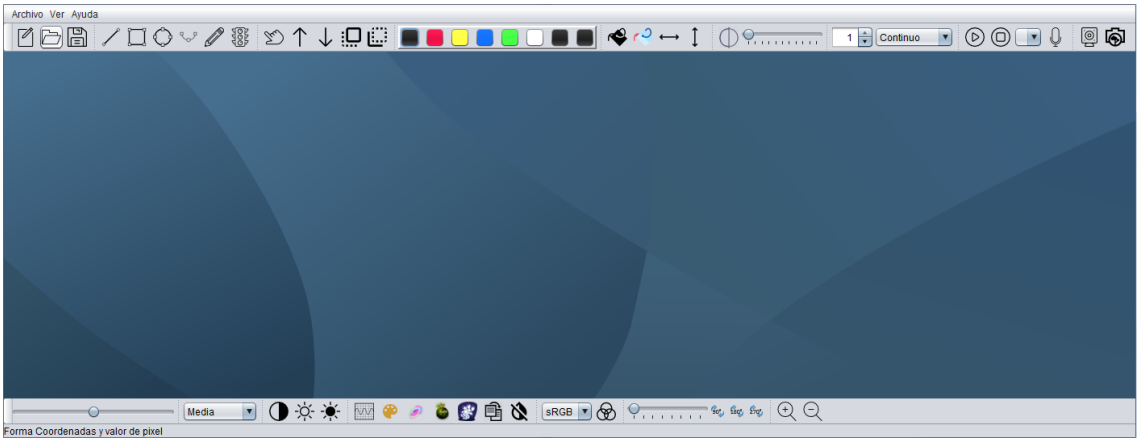


Codificación

Para esta fase se ha realizado la documentación del código con javadoc y generado su correspondiente API entregados en la primera entrega.

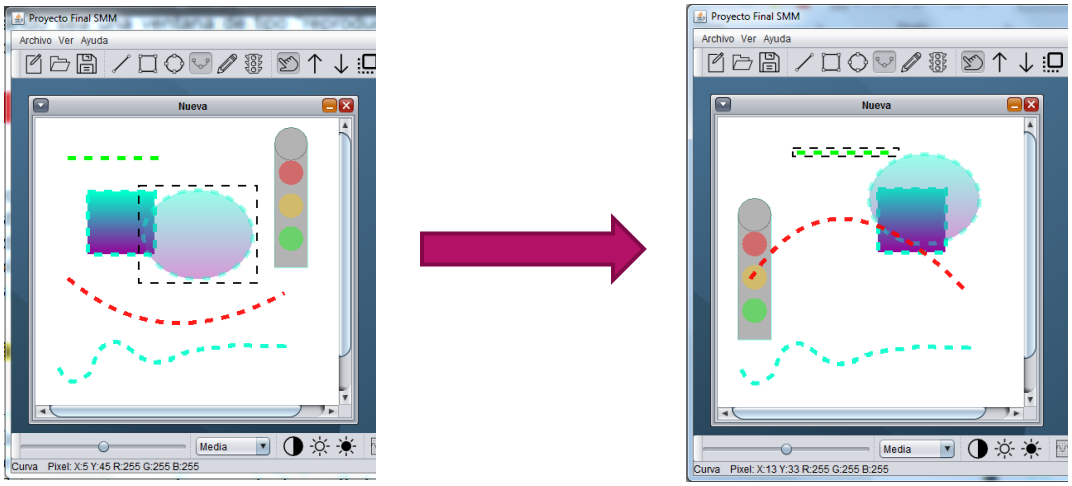
Demostración

En esta última fase se realiza una demostración de las capacidades de la aplicación, el aspecto de la aplicación es el siguiente:

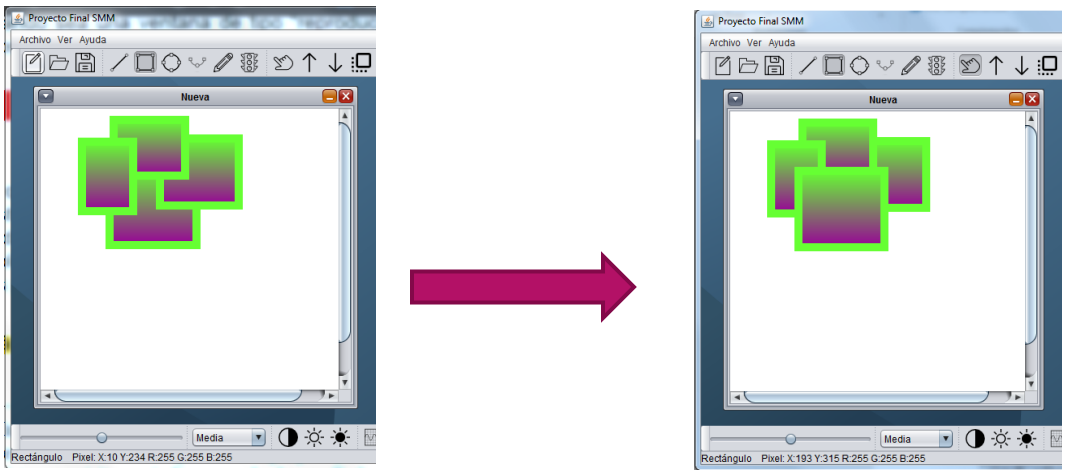


Gráficos

Para este apartado se muestra unas capturas donde se puede ver todas las formas con diferentes atributos y trasladadas, también se puede apreciar el bounding box de la selección:

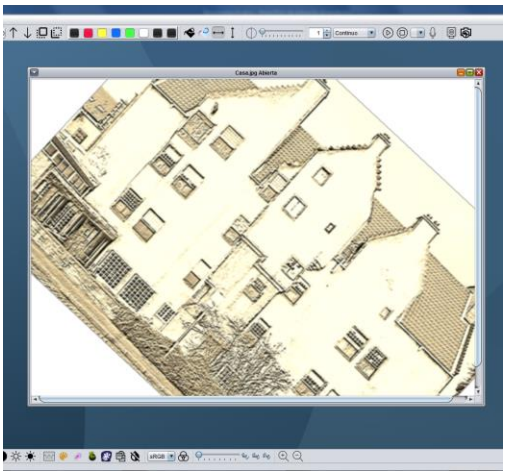


Aquí se puede ver cómo se trae al frente la figura del fondo:



Imagen

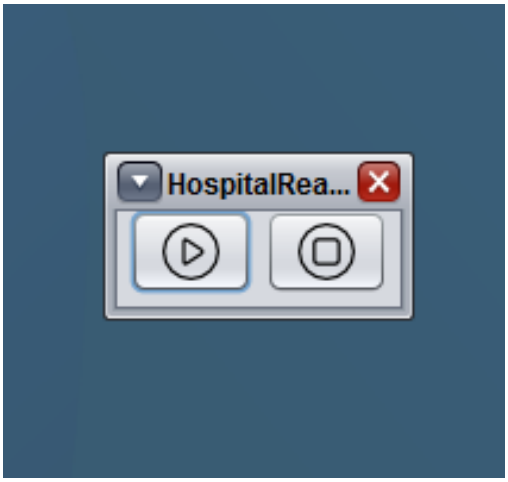
El apartado de imagen incluye todos los efectos realizados en prácticas, más los efectos de diseño propio desarrollados en el diseño:



Sonido y Video

Este apartado incluye al igual que el anterior lo realizado en prácticas, que se puede probar en la aplicación.

Para la parte del video solo se ha conseguido reproducir el audio, como se observa en la imagen, el video no se muestra, pero con los botones se puede iniciar y parar el video:



Bibliografía

Se han usado las siguientes fuentes:

- Documentación de Java: http://docs.oracle.com/javase/tutorial/2d/index.html