

Documentación de la práctica de evaluación realizada por Oleksandr Kudryavtsev

Especificación de requisitos

Analizando el documento de la evaluación de la practica se han podido extraer los siguientes requisitos, a su vez estos se han podido clasificar en distintos grupos según las funcionalidades en las que influyen.

Requisitos funcionales: son los requisitos referentes a las tareas que puede realizar el usuario utilizando el programa.

- Requisitos de carácter general.
 - El usuario podrá crear un lienzo nuevo pulsando el botón correspondiente.
 - El usuario podrá abrir cualquier tipo de archivo soportado por el programa con el mismo botón
 - El usuario podrá interactuar con las diferentes ventanas internas y activar la que desee.
- Requisitos de dibujo.
 - El usuario podrá seleccionar una figura del lienzo.
 - El usuario podrá modificar las características de la figura seleccionada.
 - El usuario podrá mover la figura seleccionada
 - El usuario podrá modificar la profundidad a la que se encuentran las figuras, es decir, enviar al fondo, traer al frente, enviar atrás o traer adelante.
 - El usuario podrá dibujar las siguientes figuras: linea, rectángulo, elipse, curva con un punto de control, trazo libre, semicírculo (área personalizada),
 - El usuario podrá gestionar los siguientes atributos de las figuras: color, trazo, relleno, alisado de bordes, transparencia, composición y transformaciones.
- Requisitos de procesamiento de imágenes.
 - El usuario podrá duplicar una imagen, es decir crear una copia en una ventana interna nueva de la imagen contenida en la ventana interna seleccionada.
 - El usuario podrá modificar el brillo de la imagen seleccionada.
 - El usuario podrá aplicar distintos filtros a la imagen, esos filtros serán emborronamiento, enfoque y relieve.
 - El usuario podrá variar el contraste eligiendo entre las opciones normal, iluminado u oscurecido.

- El usuario podrá realizar la operación negativo sobre la imagen seleccionada.
- El usuario podrá extraer las bandas de la imagen seleccionada.
- El usuario podrá cambiar el espacio de colores de la imagen seleccionada a los espacios: RGB; YCC y GRAY.
- El usuario podrá modificar el giro de la imagen.
- El usuario podrá modificar el tamaño de la imagen aumentandolo o reduciendolo.
- El usuario podrá aplicar el efecto sepia sobre la imagen seleccionada.
- Requisitos de sonido.2
 - El usuario podrá reproducir un sonido seleccionado.
 - El usuario podrá grabar sonidos.
- Requisitos de video.
 - El usuario podrá reproducir un video en formato valido reconocido por la aplicación.
 - El usuario podrá activar la webcam y mostrar la imagen en una ventana interna.
 - El usuario podrá gestionar la reproducción de un video empezando la reproducción, pausando o terminando de reproducirlo.
 - El usuario podrá realizar capturas del contenido que se esta reproduciendo sea un video abierto o el video proporcionado por la webcam

Requisitos no funcionales: son los requisitos generalmente relacionados con el diseño del programa y de la interfaz de usuario.

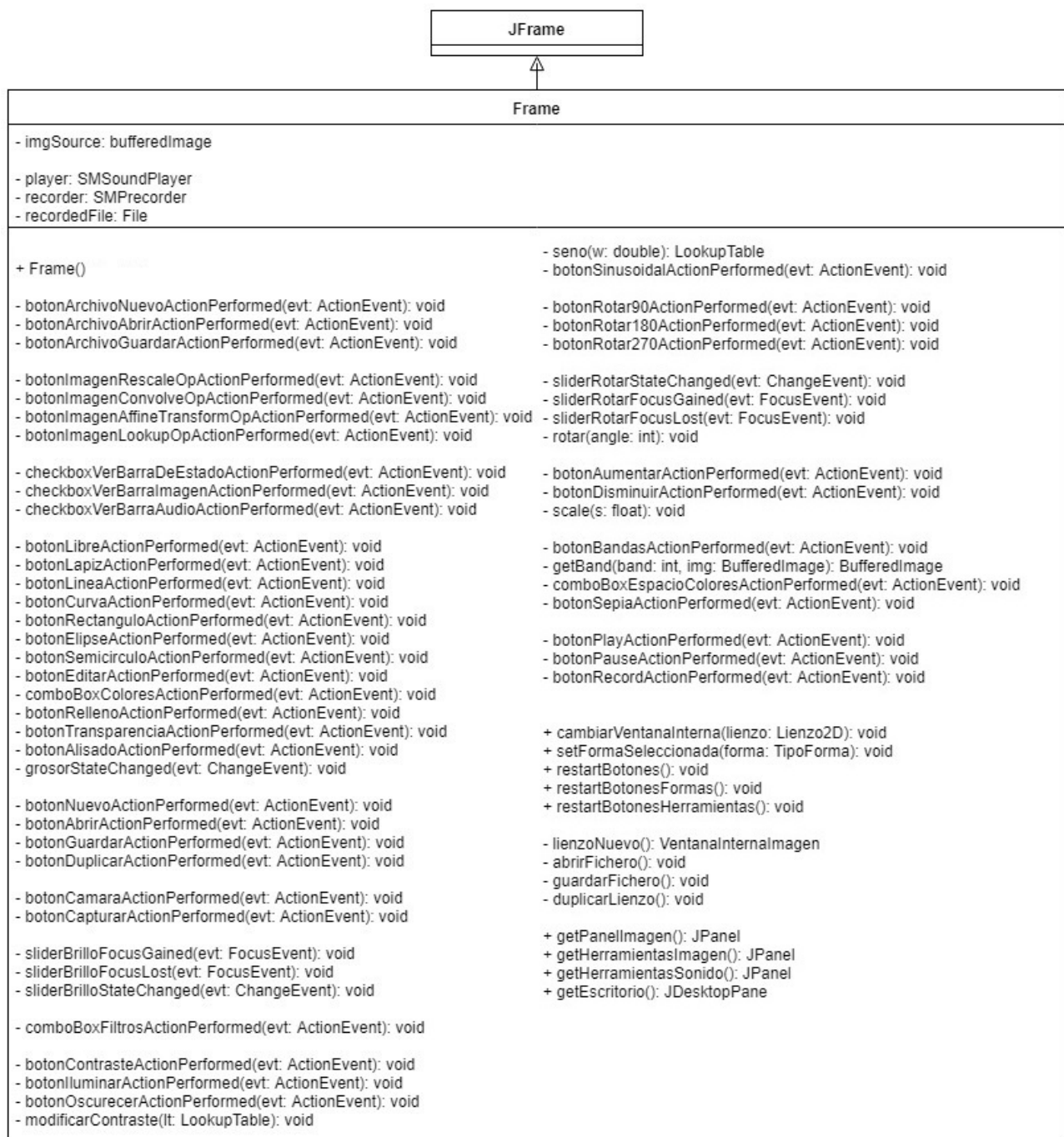
- Requisitos de carácter general
 - Tiene que existir una ventana principal que albergue las barras de herramientas y las ventanas internas que se vayan creando
 - La ventana interna sera de un tipo especifico según el tipo de archivo que se haya abierto.
 - Una ventana interna podrá contener una única imagen.
 - Las ventanas internas incluirán barras de desplazamiento si la imagen es mas grande que la ventana.
 - El nombre de una ventana interna será el nombre del archivo que se haya abierto y en el caso de ser una imagen nueva, sera “Lienzo nuevo”.
 - Los botones de la interfaz han de tener un “ToolTipText” asociado y un icono que los represente.

- Requisitos de dibujo.
 - Los botones destinados al dibujado y modificación de formas se agruparan en una barra de tareas.
 - La ventana interna almacenara las figuras dibujadas y los atributos seleccionados por el usuario.
 - Cada figura se tendrá sus propios atributos.
 - Una figura seleccionada se marcara con una “Bounding Box” discontinua alrededor suya.
 - La interfaz deberá mostrar al usuario la forma y los atributos seleccionados en la ventana interna seleccionada.
- Requisitos de procesamiento de imágenes.
 - Los botones destinado a la edición de la imagen se localizarán en una barra de herramientas.
 - El brillo de la imagen se modificara utilizando un deslizador.
 - El giro de la imagen se modificara con un deslizador.
- Requisitos de sonido.
 - Se usara una lista desplegable para seleccionar el audio a reproducir.
- Requisitos de video.
 - Se incluirá un conjunto de botones en las ventanas internas destinadas a la reproducción de video con el fin de controlar la reproducción y se situaran en una barra de herramientas.

Diseño del programa

A continuación se a explicar como se ha llegado a una solución que permita satisfacer los requisitos mencionados anteriormente. Al principio se mostrará la forma de abordar el problema y diseñar las clases y se explicara la razón por la que se ha decidido proceder de esa manera. Después de eso se explicara la interfaz de usuario usada, su funcionamiento y ademas se explicara la forma de organizar las opciones.

Para empezar tenemos una ventana principal de clase “Frame” propia que hereda de la clase existente “JFrame”. Esta clase es la clase principal de la practica ya que contiene la gran mayoría de los controles de la aplicación y dentro de esta ventana se se encuentra el escritorio y sus ventanas secundarias que se explicarán posteriormente.



La clase “Frame” es la mas extensa de la practica y contiene el código responsable de la ejecución casi todas las acciones disponibles para el usuario.

Los atributos de la clase que cabe destacar son “imgSource” de la clase “BufferedImage”, se utiliza para almacenar la imagen a la que el usuario quiera aplicar los efectos implementados. Los otros tres atributos son necesarios para la gestión de audio, tanto de su grabación como de su reproducción. Son “player” de la clase “SMSoundPlayer”, “recorder” de la clase “SMRecorder” y “recordedFile” de la clase “File” ese ultimo se utiliza para almacenar temporalmente el archivo de audio que se crea al grabar un audio.

La mayoría de las funciones de esta clase son las funciones llamadas en el caso de un evento que puede ser la pulsación de un botón, deslizamiento de un deslizador etc, a su vez estas funciones llaman a otras funciones que evitan repetir el código y en otros casos llaman a funciones localizadas en otras clases que realicen las operaciones correspondientes.

Esta clase contiene numerosos botones y controles cuyo funcionamiento se explicara en el apartado “Manual de usuario” que se encuentra después del “Diseño del programa” .

Esta ventana principal tiene un componente “escritorio” de clase “JdesktopPane” que es un contenedor en el que se pueden almacenar otras ventanas internas. Estas ventanas internas se crean y se añaden al escritorio para visualizarse y mostrar su contenido, cuando el usuario abre un fichero o crea un lienzo nuevo.

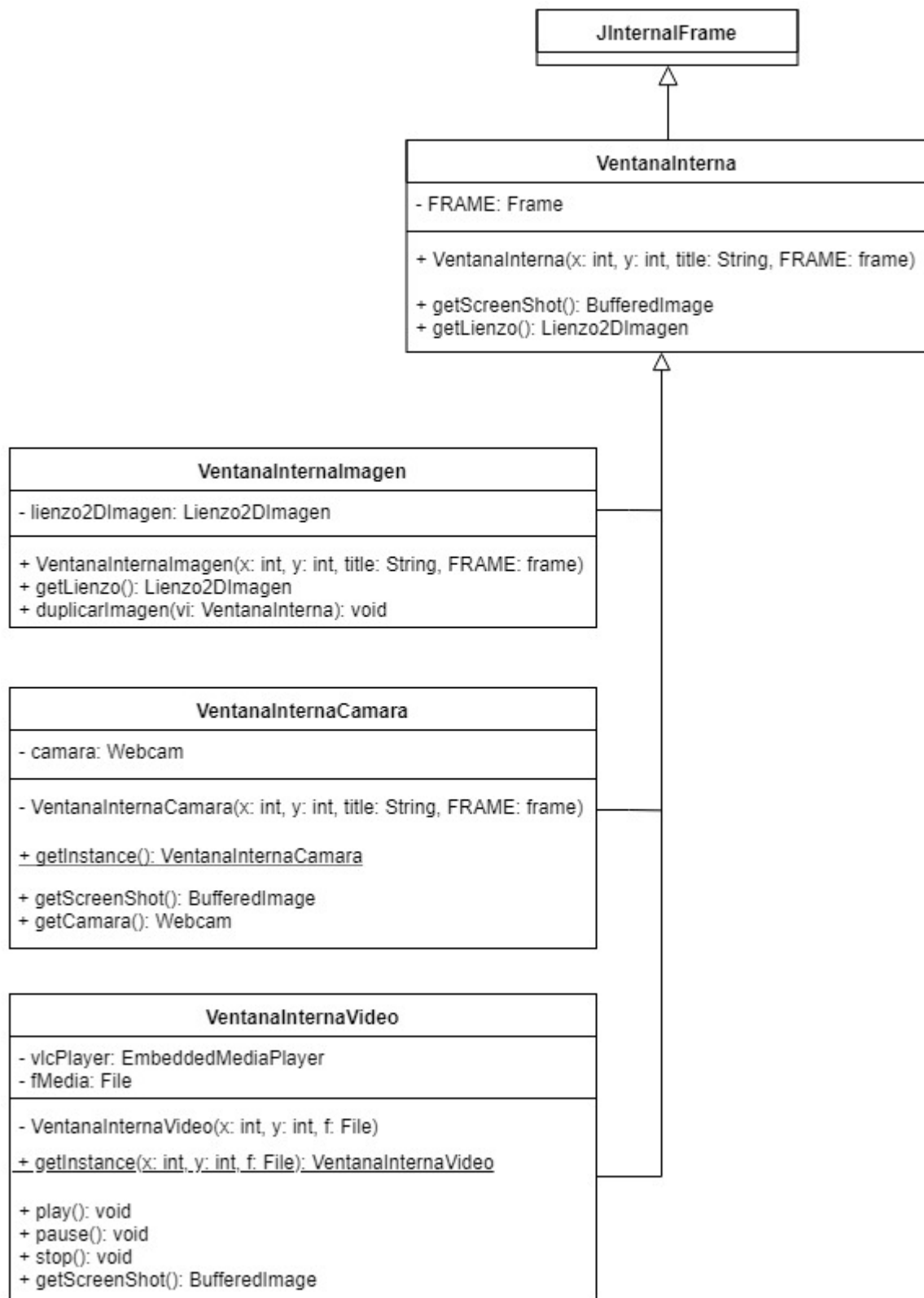
Como se ha mencionado en los requisitos del programa el usuario podrá abrir distintos tipos de archivos. Dependiendo del tipo de archivo abierto se podrán realizar acciones diferentes sobre este y utilizar una misma ventana interna para representar y gestionar distintos tipos de archivos no resulta nada eficiente.

Para solucionar este problema he creado una clase básica llamada “VentanaInterna” que hereda de la clase “JinternalFrame”. Teniendo esta clase básica he creado otras clases que heredan de ella, una clase distinta para cada tipo de archivo que necesitemos utilizar. “VentanaInternaImagen” para imágenes y dibujos, “VentanaInternaCamara” para gestionar la salida de la webcam, “VentanaInternaVideo” para la reproducción de archivos de video.

La clase “VentanaInterna” contiene el atributo FRAME que hace referencia a la ventana principal “Frame”, se utiliza cuando se cambia la ventana seleccionada y necesitamos cambiar el estado de los botones encontrados en la ventana principal. También tiene un constructor en el que se pasa la posición en la que se quiere crear la ventana, el título que se le quiere poner y la referencia a la ventana principal.

En cuanto a las clases que heredan de “Ventana Interna” tenemos:

La clase “VentanaInternaImagen” que contiene el atributo “lienzo2DImagen” de la clase propia “Lienzo2DImagen” que explicaremos mas adelante y funciones para devolver el atributo “lienzo2DImagen” y la función necesaria para duplicar la imagen almacenada en una ventana interna a partir de la imagen que tiene almacenada otra ventana interna dada.



La clase “VentanaInternaCamara” se usa para gestionar la salida de la webcam y tiene las funciones necesarias para la gestión de este recurso, haciendo posible hacer capturas del video emitido.

Por ultimo tenemos la clase “VentanaInternaVideo” utilizada para la reproducción de archivos de video. Esta ventana difiere visualmente de las otras dos ya que tiene un área con botones para reproducir pausar y parar el video visualizado. También tiene implementada la posibilidad de hacer capturas del video reproducido.

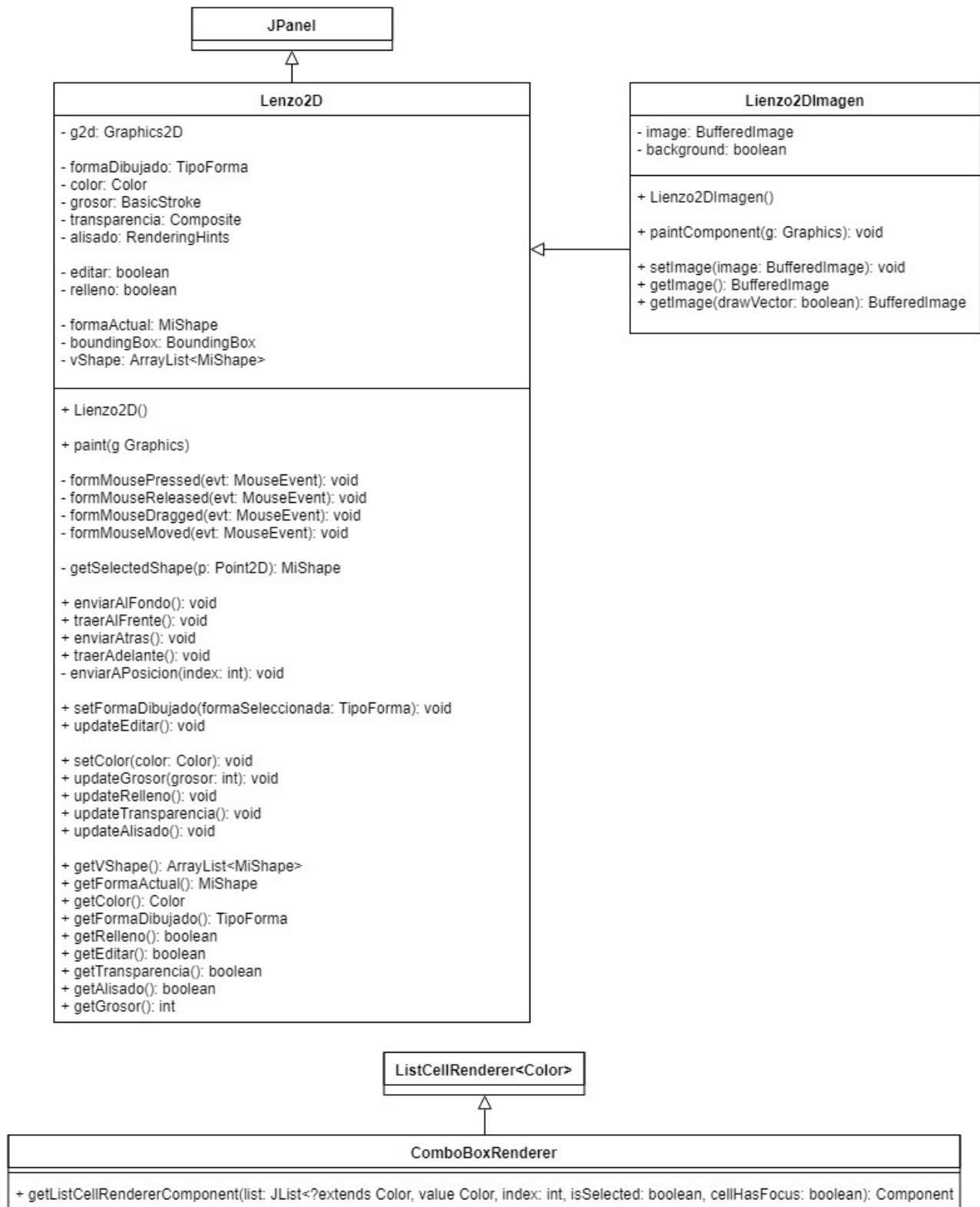
De los tres tipos de ventana interna el tipo de ventana que mayor complejidad presenta es la clase “VentanaInternaImagen” que permite dibujar formas con distintos atributos, modificarlas y realizar distintas operaciones sobre una imagen. A continuación se va a explicar como este tipo de ventana interna puede realizar todas estas acciones.

La clase “VentanaInternaImagen” se puede interpretar como un contenedor dedicado a almacenar y gestionar el atributo lienzo2DImagen de la clase propia “Lienzo2DImagen”. Este atributo en realidad es un “JPanel” extendido por la clase “Lienzo2D” y esta a su vez extendida por “Lienzo2DImagen”. Por lo tanto “VentanaInternaImagen” actúa de contenedor de para “Lienzo2DImagen” dentro del cual es donde se realizan todas las operaciones de dibujo y edición.

“Lienzo2DImagen” extiende a la clase “Lienzo2D” por lo tanto empezaremos explicando como funciona esta clase.

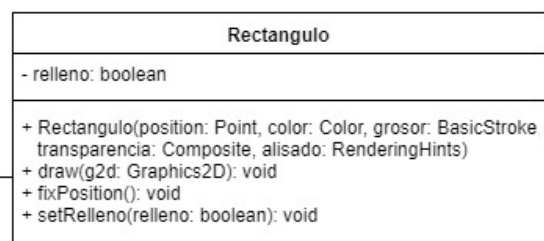
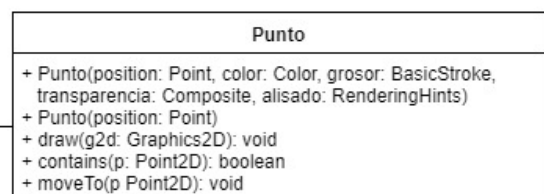
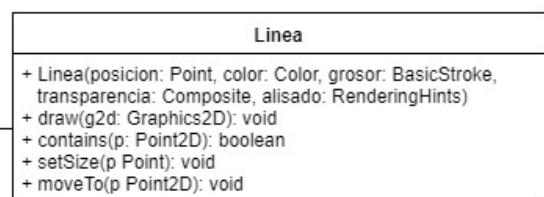
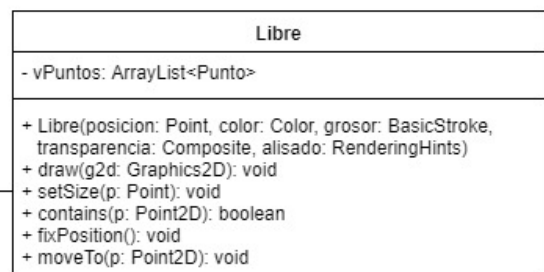
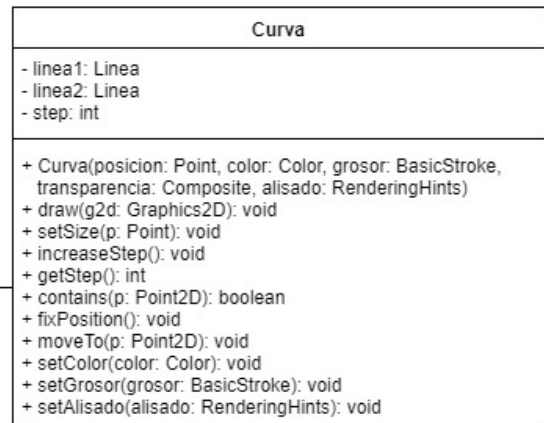
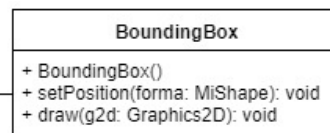
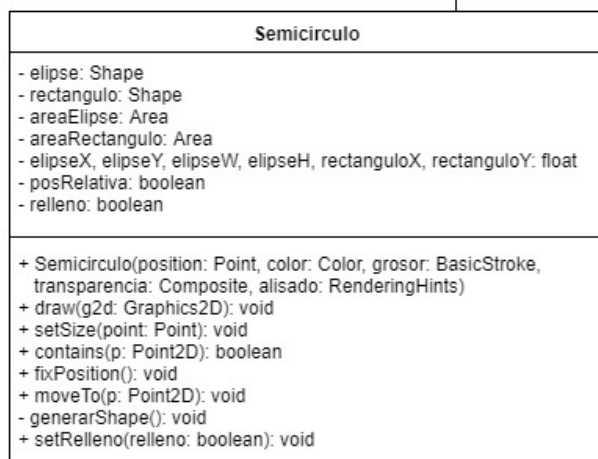
“Lienzo2D” es una clase que extiende de la clase “JPanel” y añade la funcionalidad necesaria para dibujar, almacenar y modificar formas. Para almacenar las formas dibujadas tiene un contenedor donde almacena las distintas formas “vShape” y para dibujarlas tiene atributos para todos los atributos seleccionados en la ventana principal y una forma auxiliar. De esta forma cuando el usuario dibuje una forma se utiliza esta forma auxiliar a la que se le aplican los atributos almacenados, finalmente cuando se ha terminado de dibujar esta forma se añade a la lista de formas y el atributo de forma auxiliar deja de referenciar a la forma creada. Para la visualización se utiliza el atributo “g2d” de la clase “Graphics2D” y la función “paint” donde se recorre el vector de formas “vShape” y se dibuja cada forma. Las demás funciones de la clase son responsables de gestionar los atributos y operaciones seleccionados desde la ventana principal.

La clase “Lienzo2DImagen” extiende la funcionalidad de “Lienzo2D” añadiendo la posibilidad de almacenar una imagen. Gracias a eso obtenemos la clase “Lienzo2DImagen” con la que podemos dibujar y editar formas combinado con la visualización de una imagen y posibilidad de aplicarle distintas modificaciones como filtros, escalados, rotaciones etc.



Aquí se incluye el diagrama de la clase **ComboBoxRenderer** utilizado para renderizar las celdas del combo box de la selección de los colores. Cabe mencionar que

“**Lienzo2D**” es capaz de almacenar en un mismo vector distintos tipos de formas y el funcionamiento esto es lo que vamos a explicar a continuación.



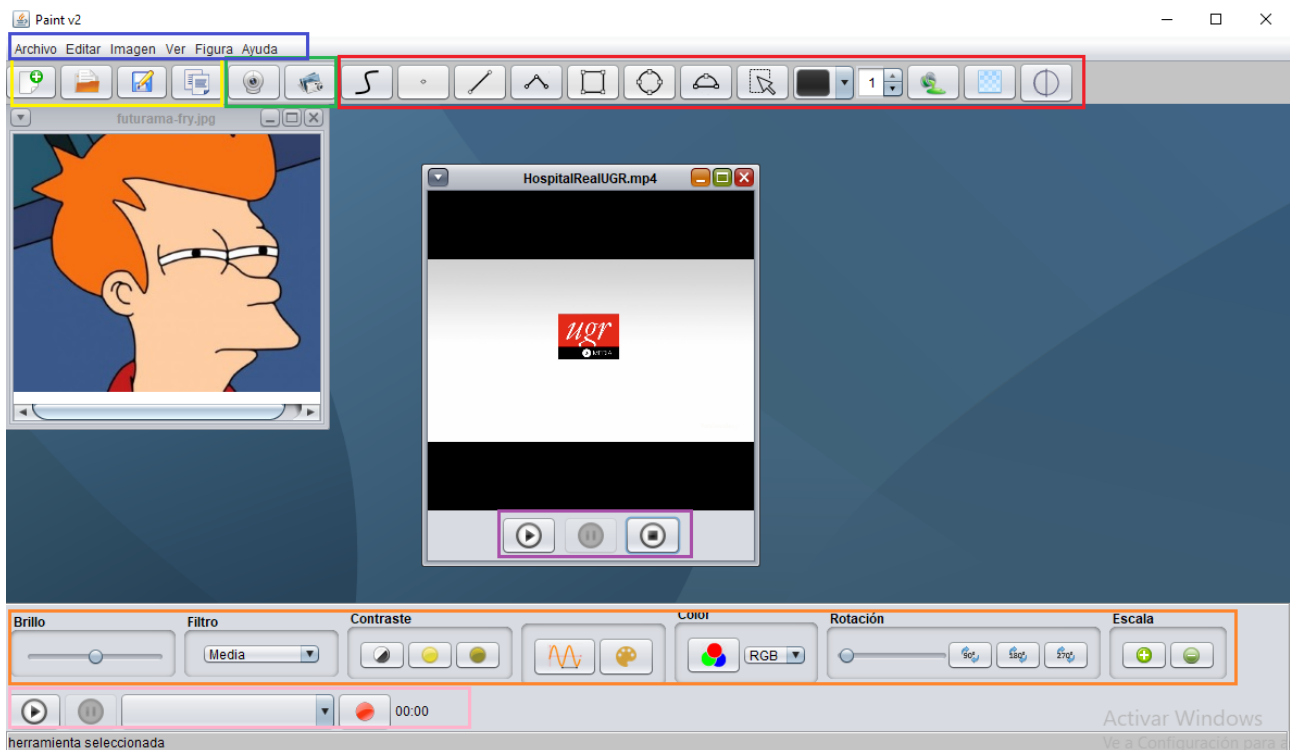
Como se ve en la imagen anterior para optimizar lo mas posible el funcionamiento del programa he optado por crear una clase básica “MiShape” que almacena todas las características comunes de las formas posibles en forma de atributos. También es donde se define toda la funcionalidad necesaria para las formas.

Teniendo esta clase básica “MiShape” he creado una clase para cada forma necesaria, incluida la clase “BoundingBox” que comparte bastante funcionalidad con las demás formas. Teniendo la funcionalidad y atributos básicos no es suficiente para muchas de las figuras por lo tanto necesitan atributos y funciones propios. En el caso de algunas figuras como por ejemplo “Curva” o “Semicírculo” almacenan y gestionan a su vez otras figuras de clases propias para en su conjunto formar la figura deseada. Una curva esta formada de dos lineas por lo tanto utilizo dos objetos de clase “Linea” para representarla y en el caso del semicírculo utilizo una elipse y un rectángulo haciendo uso de las clases “Elipse” y “Rectangulo” respectivamente. Aunque el caso de la clase “Libre” es el mas notable ya que es usada para el trazo libre en el que utilizo un array de “Punto” para hacerlo funcionar.

Manual de usuario

Para diferenciar los distintos controles y opciones del programa utilizaremos un código de colores.

- Azul: Barra de menú, en esta barra se agrupan las opciones para abrir un nuevo archivo, guardar un lienzo... ademas de la posibilidad de aplicar algunos filtros en el menú “Imagen” y cambiar el orden de las formas en el menú “Figura”
- Amarillo: contiene los botones para crear un lienzo nuevo, abrir un archivo, guardar un archivo y duplicar una imagen.
- Verde: contiene el botón para activar la cámara y el botón para hacer capturas sea al video de la cámara o a un video abierto como archivo.
- Rojo: Al principio tiene los botones para la selección de las formas posibles a dibujar, a continuación están los botones relacionados con la modificación de las formas, primero esta el de activar modo de edición posibilitando seleccionar formas y a continuación modificarlas y al final están los botones para seleccionar los atributos de la forma a dibujar o editar.
- Morado: en las ventanas de tipo video hay están los botones necesarios para poder comenzar la reproducción, pausarla o terminarla.
- Naranja: Es el panel que contiene todas las operaciones que podemos realizar sobre la imagen contenida en el lienzo.
- Rosa: Es el panel que contiene los botones necesarios para la gestión de audios, al abrir un audio este se añadirá al combobox de este panel, se podrá seleccionar y reproducir con los botones correspondientes. También esta disponible la posibilidad de grabar un audio, pulsando el botón correspondiente.



Notas y comentarios

Problemas en la practica que no se han podido resolver:

- La primera vez que se selecciona una forma de tipo “Semicirculo” da un salto en el eje y en sentido positivo, si se vuelve a seleccionar una segunda vez el error no se repite.
- Después de abrir la cámara y hacer una captura los filtros dejan de funcionar pero si funcionan con capturas de videos.

Tras realizar muchas comprobaciones solo he encontrado estos dos problemas graves que por falta de tiempo no he podido solucionar sin embargo creo que no debería de haber mas fallos.