

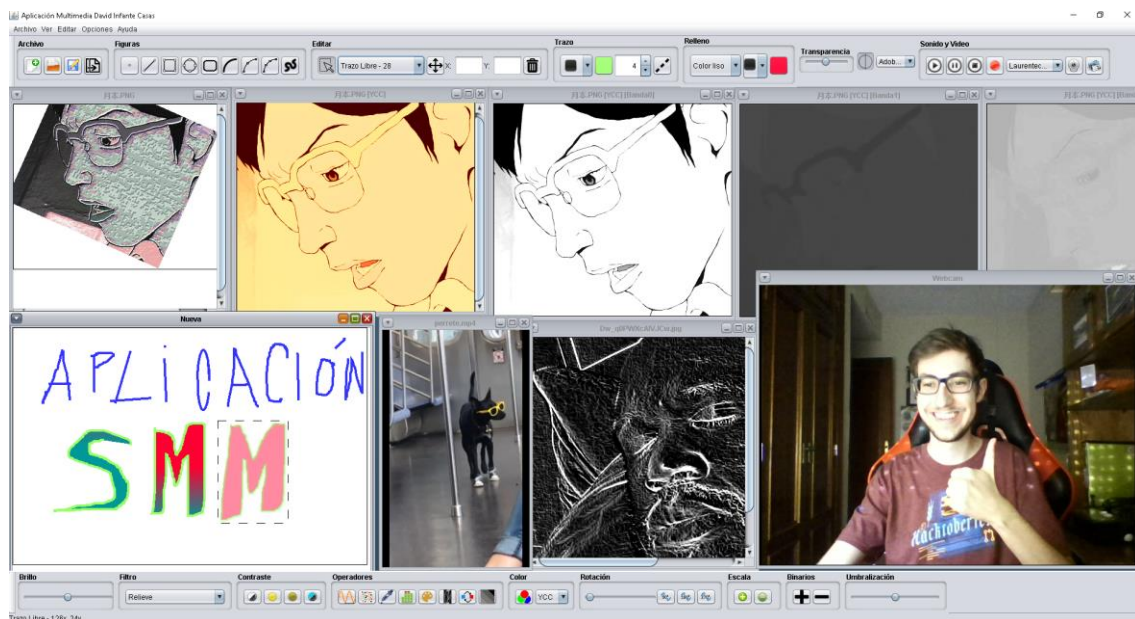
# Documentación

## Aplicación Sistemas Multimedia

David Infante

### Índice:

- 1.- Requisitos
- 2.- Análisis
- 3.- Diseño
- 4.- Codificación
- 5.- Manual de usuario
- 6.- Bibliografía



Aspecto final de la aplicación con algunos ejemplos

# **Requisitos**

## **Funcionales:**

La aplicación podrá realizar las mismas tareas básicas que una aplicación multimedia simple, lo cual incluye:

- Crear lienzos vacíos sobre los que dibujar formas simples.
- Modificar los atributos básicos con los que se dibujan las formas.
- Seleccionar, mover y borrar formas de un lienzo.
- Abrir todo tipo de archivos multimedia, de imagen, audio o video.
- Modificar el aspecto de las imágenes abiertas, mediante el dibujo de formas sobre ellas, cambio en su espacio de color, filtros, u otro tipo de operaciones.
- Crear ventanas duplicadas a partir de lienzos o imágenes abiertas.
- Guardar el resultado de las modificaciones en una nueva imagen.
- Reproducir ficheros de audio.
- Grabar audio a partir de un micrófono.
- Reproducir ficheros de video.
- Reproducir el video que está siendo capturado por una webcam.
- Tomar instantáneas de lo que se reproduce en un video o en la webcam.

## **No funcionales:**

- Estabilidad de la aplicación. Aunque ocurra un error, la funcionalidad debe de ser transparente al usuario, de modo que pueda seguir usando la aplicación sin problemas.
- La aplicación debe de ser accesible, simplificando su uso mediante iconos y añadiendo toolTipText a todos los elementos en caso de que el usuario no comprenda lo que representa un icono.
- El programa debe de ser portable y poder ejecutarse en distintos entornos de forma transparente.
- Será obligatorio tener instalado el reproductor VLC en el directorio C:\Program Files\VideoLAN\VLC o Program Files (x86) para que se pueda reproducir video.

# **Análisis**

En la actualidad, en el ámbito de dibujo de gráficos y administración de imágenes en Java, existen dos clases principales: Graphics y Graphics2D.

Graphics ofrece la posibilidad de dibujar figuras simples e imágenes, siendo la clase abstracta que el resto de clases Graphics extienden.

Por otro lado, Graphics2D ofrece algunas operaciones más avanzadas sobre la geometría, el control de color o texto.

Sin embargo estas soluciones pueden no ser óptimas dado que las opciones que ofrecen son limitadas.

Lo que se propone con el diseño e implementación de la biblioteca SM.dic.biblioteca es crear una biblioteca que unifique las opciones básicas y avanzadas de Graphics y Graphics2D y además las extienda añadiendo más funcionalidades.

Para los gráficos, se ha implementado un sistema de clases que ofrece la posibilidad de dibujo de varias formas y modificación de sus atributos, los cuales no se limitan solo a color y trazo sino que añaden nuevos como transparencia, relleno o degradado entre otros.

Para el control de imagen, además de poder abrir las imágenes, se han añadido todo tipo de filtros y operaciones para modificar el aspecto de la imagen, además de poder extraer las bandas o cambiar su espacio de color.

Por último, con el fin de unificar en una sola biblioteca todo tipo de funcionalidades multimedia, se ha incluido la reproducción de video, captura de video mediante una webcam, grabación de audio y reproducción de archivos de audio.

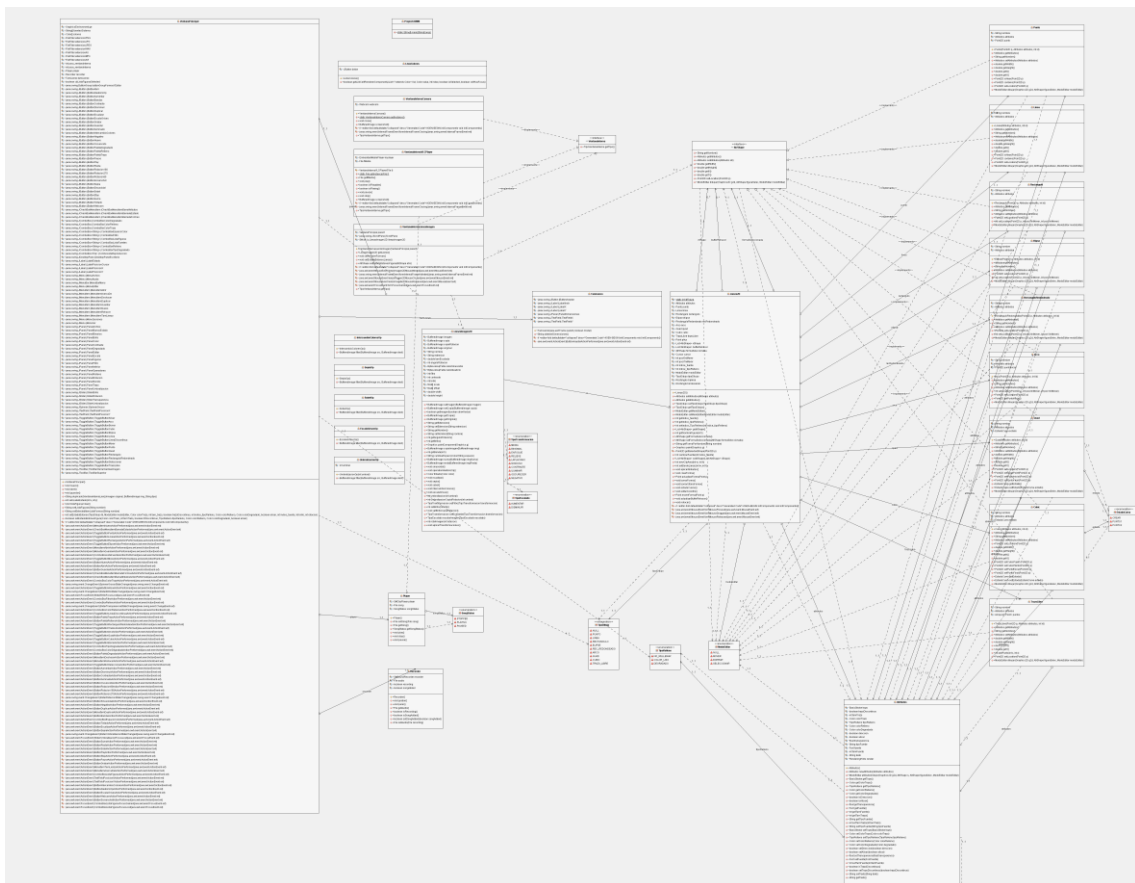
Gracias al desarrollo de una biblioteca multimedia unificada, se conseguirá simplificar otros desarrollos multimedia.

# Diseño

La aplicación está compuesta por el Proyecto “Proyecto-Final” (1.), que contiene la Ventana Principal del programa, así como las ventanas internas que se irán abriendo conforme abramos archivos o creamos lienzos nuevos.

Para administrar los gráficos, imágenes, lienzos y sonidos, se hace uso de la biblioteca “SM.dic.biblioteca” (2.).

El diagrama de clases de ambos proyectos se muestra en la siguiente imagen:



En la raíz del zip entregado se incluye el fichero DiagramaClases.png para poder ver el diagrama con una mayor resolución

## 1. Proyecto-Final

El proyecto cuenta con un paquete principal, “**proyectosmm**” en el cual se incluyen las siguientes clases:

- **ProyectoSMM**: contiene el main y pone en marcha el programa, también incluye un pequeño código que intenta cargar las bibliotecas de VLC, en caso de no conseguirlo el programa continúa pero no podrá reproducir video.
- **VentanaPrincipal**: vista principal del programa con las distintas opciones, en el jDesktopPane central, se irán añadiendo las distintas ventanas internas al programa. Cuando sea necesario escoger un color, se ofrecerán colores básicos en un comboBox y justo a su lado una paleta más variada. Cuando se actualice el color al de una forma seleccionada, solo se actualizará el color de la paleta.
- **TipoVentanaInterna**: enumerado con los distintos tipos de ventana interna que puede haber.
- **VentanaInterna**: interfaz que implementan las ventanas internas, incluye operaciones básicas para una ventana interna, siendo la principal getTipo(), la cual devuelve el tipo de ventana interna que es. Esta función es vital a la hora de controlar sobre qué tipo de ventana interna se realizan los eventos en la ventana principal.
- **TamLienzo**: JDialog que se lanzará cuando vayamos a crear un lienzo vacío o queramos cambiar los valores por defecto de tamaño de creación de un lienzo.
- **VentanaInternaCamara**: tipo de ventana que se puede abrir en la VentanaPrincipal y que muestra el video que está siendo capturado por la webcam. Se pueden tomar instantáneas del video.
- **VentanaInternaVLCPlayer**: segundo tipo de ventana que se puede abrir en la VentanaPrincipal, reproduce un video. Como se ha

mencionado, en caso de que VLC Player no esté instalado en uno de los directorios propuestos, este tipo de ventana no funcionará. También se pueden tomar instantáneas.

- **VentanaInternaLienzoImagen:** En función de la llamada desde VentanaPrincipal, crea una ventana interna con un lienzo vacío o con una imagen. Controla la posición en la que se encuentran los botones de dibujo, edición y atributos, así como la lista de formas de la VentanaPrincipal mediante el manejo de eventos.

El paquete “**iconos**”, contiene los iconos que se muestran en la VentanaPrincipal.

Por otro lado el paquete “**lib**” contiene las bibliotecas necesarias para el funcionamiento de la aplicación.

## 2. SM.dic.biblioteca

La biblioteca cuenta con varios paquetes, diferenciados según el campo multimedia sobre el que actúan.

El paquete “**SM.dic.graficos**” contiene todos los tipos de formas que podrán ser dibujadas, sus atributos y los distintos modos que tienen.

- **AtrShape:** interfaz que hereda de Shape. Todas las formas que se pueden dibujar en un lienzo la implementan. Se ha optado por el uso de una interfaz antes que de una clase abstracta por dos motivos principales:
  - Dado que Java no permite la herencia múltiple, no sería posible implementar una forma que heredase de AtrShape y de otra clase tipo java.awt.geom.forma.
  - Por simpleza de código, todas las formas implementan una interfaz que tiene las operaciones básicas que necesita una forma.
- **Atributos:** contiene todos los atributos que puede tener una forma, desde color de trazo, tamaño, transparencia, etcétera. Contiene el método “atributosDibujo” el cual define los parámetros con los que la forma se dibujará en el lienzo.
- **TipoRelleno:** enumerado que delimita el tipo de relleno de una forma.
- **Punto, Linea, Rectangulo, Elipse, RectanguloRedondeado, Arco, Quad, Cubic y TrazoLibre:** clases con las distintas formas que se pueden dibujar, todas implementan AtrShape. Además cuentan con un atributo de tipo Atributos para que puedan tener los suyos propios y un nombre, al cual se le añade un identificador único para cada forma para poder identificarlas mejor. Todas implementan un método setLocation para poder desplazarlas y un método dibujar para poder pintarlas sobre un lienzo.

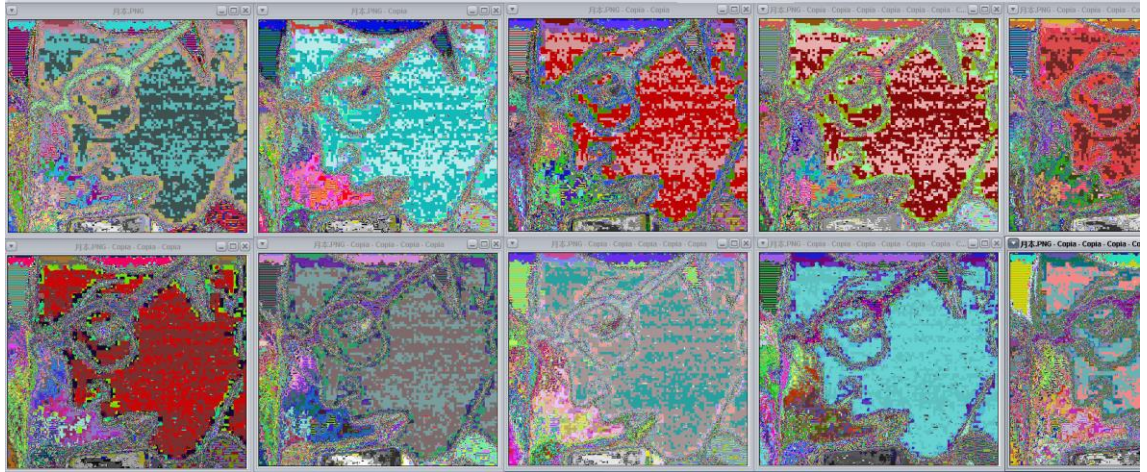
- **EstadoCurva:** enumerado que establece en qué paso de dibujado se encuentra una forma que tenga puntos de control. Se aplica a Quad y Cubic.
- **TipoDibujo:** enumerado que define el tipo de forma que se está dibujando en ese momento.
- **ModoEditar:** enumerado que define el tipo de edición que se está realizando: Ninguno, Seleccionar, Mover o Borrar.

El paquete “**SM.dic.iu**” incluye las clases que hacen referencia a la interfaz de usuario; destacan:

- **Lienzo2D:** lienzo básico sobre el que dibujar formas AtrShape, mencionadas en el bloque anterior. Contiene un vector de formas AtrShape sobre el que ir añadiendo formas según se dibujen, o eliminando si se borran. El método paint recorre dicho vector y dibuja cada AtrShape llamando a su método propio dibujar. La clase contiene todo tipo de atributos y métodos relacionados con el dibujo, control del mismo, creación de formas y su edición. También maneja eventos de entrada de ratón para poder controlar estos parámetros. Por último se ha añadido un “bufferRehacer” (tipo Ctrl+Y, Ctrl+Z) para poder restaurar o borrar formas del lienzo de forma sencilla.
- **LienzoImagen2D:** subclase de Lienzo2D especializada en el procesamiento de imágenes, con todo tipo de operaciones sobre ellas y control de las mismas. El método copiaImagen crea una imagen copia nueva a partir de otra imagen. También cabe destacar el atributo “copiaRotacion” al cual se le aplican todas las operaciones al igual que a la imagen del lienzo, menos las de rotación y escalado, de forma que nunca perdamos la calidad o las esquinas de la imagen al rotarla o escalarla. Esta clase también implementa el operador de tipo “LookupOp”, “**operadorAleatorioOp**” el cual devuelve un valor aleatorio entre 0 y 255 mediante Math.Random() 256 veces y genera un valor



ByteLookupTable que se pasa al operador LookupOp que se usa para “filtrar” la imagen. Dado que el resultado es siempre aleatorio, no se puede saber con certeza cómo será el resultado de aplicar el operador.



Resultado de aplicar el operadorAleatorioOp a diez imágenes iguales

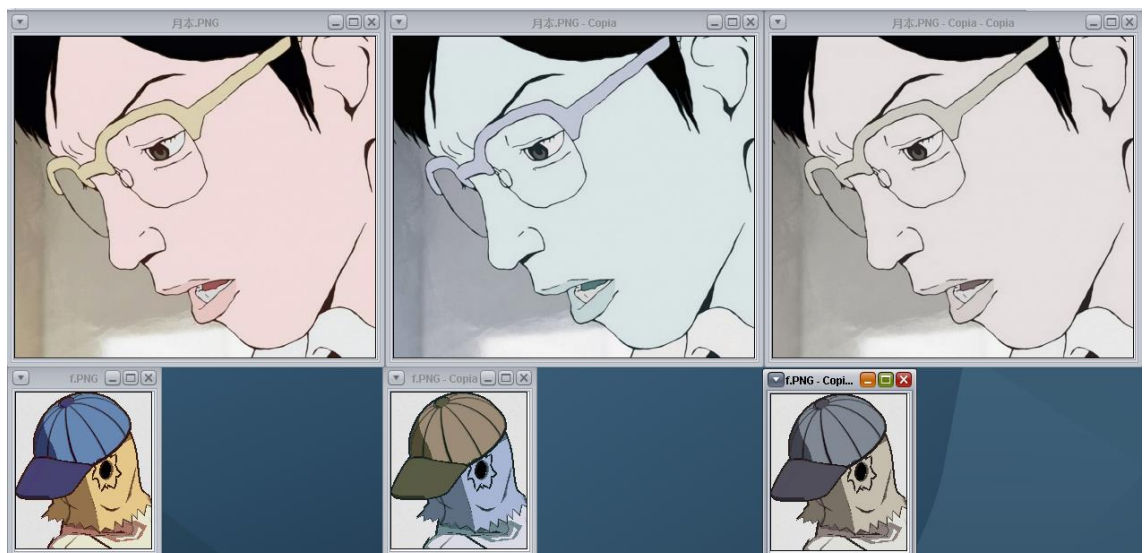
- **ListaColores**: clase que implementa un ListCellRenderer de colores para el comboBox de colores básicos de la Ventana Principal.

En el paquete “**SM.dic.imagen**” se encuentran todas las operaciones para el tratado de imágenes:

- **TipoEscalado**: enumerado que establece el tipo de escalado de la imagen, aumentar o disminuir.
- **TipoTransformación**: enumerado que establece distintos tipos de transformaciones que se pueden realizar a una imagen incluyendo tipos de filtros y contrastes.
- **SepiaOp**: operación que aplica el filtro sepia a una imagen.
- **SobelOp**: operación que aplica el filtro de detección de bordes “sobel” a una imagen.

- **UmbralizacionOp**: operación que aplica umbralización a una imagen.
- **IntercambioColoresOp**: operación componente a componente que suma para cada componente RGB del píxel el valor multiplicado por 0.5 de las otras dos componentes, es decir:
  - Rojo =  $\text{Mínimo}(255, \text{Verde} * 0.5 + \text{Azul} * 0.5)$
  - Verde =  $\text{Mínimo}(255, \text{Rojo} * 0.5 + \text{Azul} * 0.5)$
  - Azul =  $\text{Mínimo}(255, \text{Rojo} * 0.5 + \text{Verde} * 0.5)$

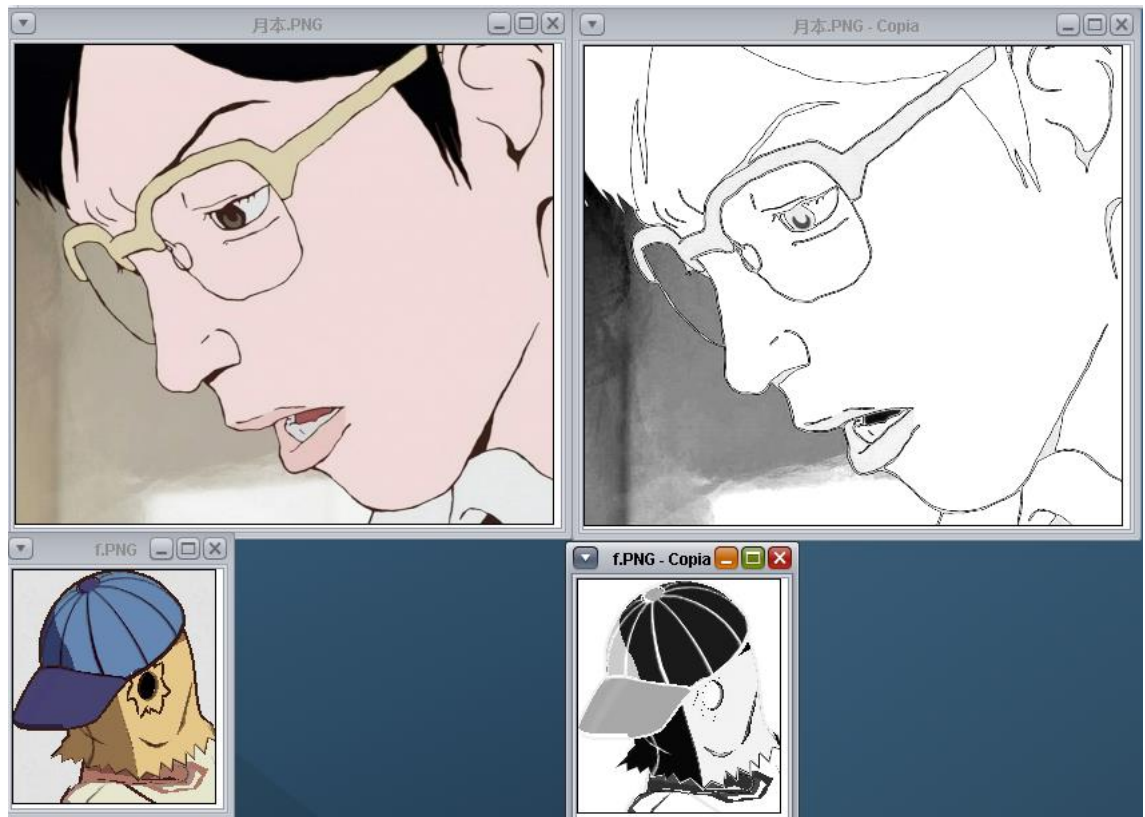
De este modo se obtiene un efecto de “Intercambio de colores” y “enfriamiento” de los mismos, ya que las componentes se van multiplicando por 0.5.



El primer par de imágenes en vertical son las originales, el segundo par es tras recibir la operación una vez y el tercer par es tras recibir la operación dos veces consecutivas. Como se puede apreciar, conforme más aplicamos la operación más se “enfrian” los colores y tras aplicar el operador dos veces, obtenemos la imagen con unos colores similares a la original pero más fríos.

- **EscalaGrisesOp**: operación pixel a pixel en la cual se calcula la hipotenusa mediante `Math.hypot()` del valor absoluto de la suma de los samples 0 y 1 del píxel – 255 y de nuevo este valor. Este valor se aplica a las tres componentes del píxel.  
Tras tomar la suma de los samples 0 y 1 del píxel, la fórmula sería:

Hipotenusa(valor absoluto(sumaSamples – 255), valor absoluto(sumaSamples – 255));



Como se aprecia, al aplicar la operación, se obtiene un efecto blanco-gris-negro, cambiando cada color por una tonalidad distinta y los negros por blancos.

El paquete “**SM.dic.sonido**” contiene los elementos multimedia relacionados con el sonido:

- **SongStatus**: enumerado que muestra en qué estado se encuentra el fichero de audio, reproduciéndose, parado o pausado.
- **Player**: clase que crea un reproductor de audio al que se le pasa un fichero para su reproducción. Implementa métodos imprescindibles como play, stop, pause, setSong o getSongStatus.
- **Recorder**: crea un grabador con el que grabar un fichero de audio mediante el sonido captado por un micro. Implementa los métodos básicos grabar y parar entre otros.

El paquete “**lib**” contiene las bibliotecas necesarias para el funcionamiento de la biblioteca.

# Codificación

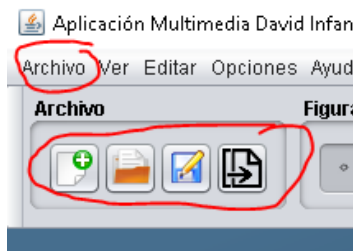
Todo el código (clases, atributos y métodos) ha sido documentado usando javadoc. Las APIs generadas por javadoc de ambos proyectos se pueden encontrar en la carpeta Javadoc de la raíz del zip.

## Manual de Usuario

En este apartado repasaremos las distintas opciones que ofrece el programa, subdividiéndolas en cinco apartados.

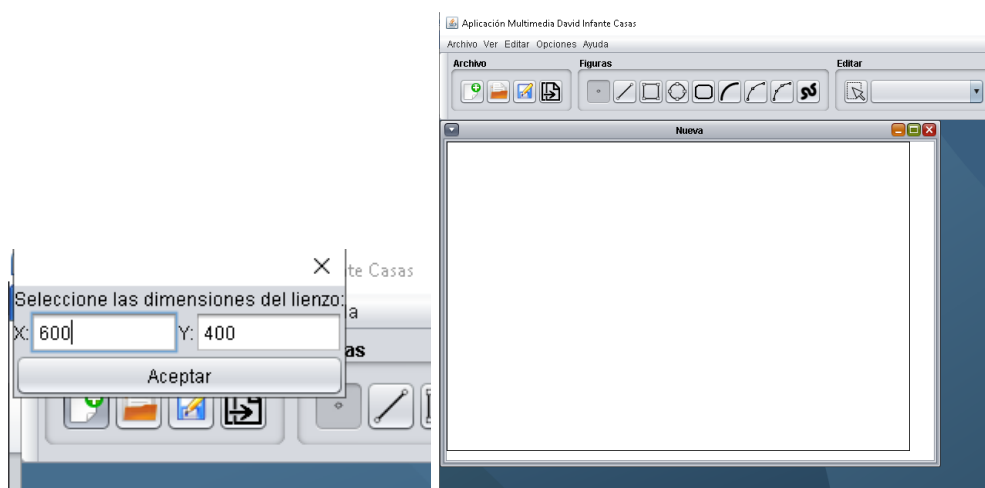
### Archivo:

Las opciones de “archivo” se pueden acceder desde los iconos del panel archivo de la parte superior como desde la barra de menú:



Encontramos:

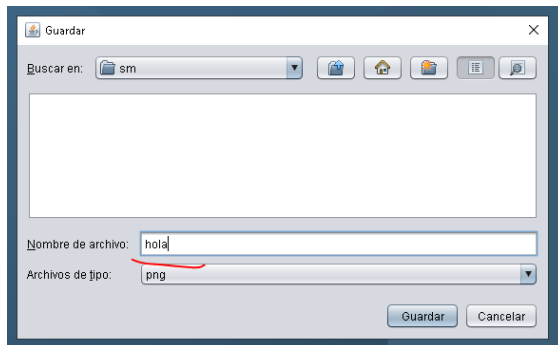
- **Nuevo** (Alt+N): abrirá un diálogo que nos pedirá las dimensiones del lienzo a crear, una vez introducidas el lienzo se añadirá al programa.



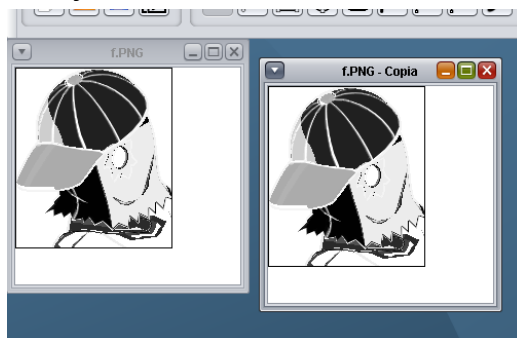
- **Abrir (Alt+A):** Abre un nuevo fichero de imagen, audio o video. Podemos elegir la extensión del fichero gracias a “Archivos de tipo”. Al seleccionar un fichero, si es de imagen, este se abrirá en una ventana de su clase. En el caso de audio y video, se añadirá a la lista de reproducción.



- **Guardar (Alt+G):** Guarda una imagen o lienzo dibujado en el directorio seleccionado con el nombre escrito.



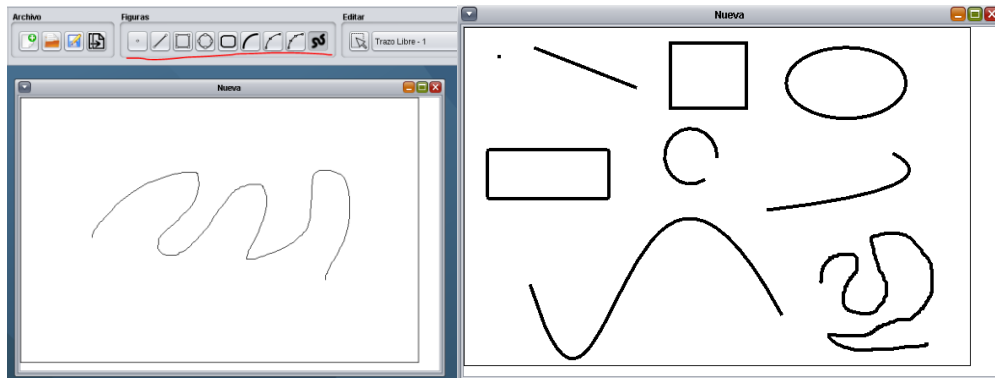
- **Duplicar (Alt+D):** Crea una nueva ventana copiando la imagen o dibujo de otra.



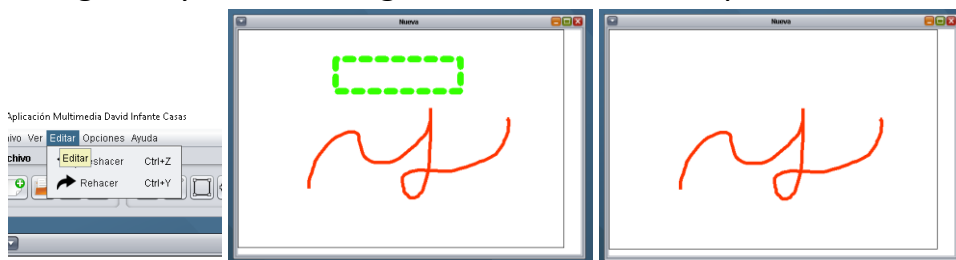
## Gráficos:

A partir de un lienzo o una imagen:

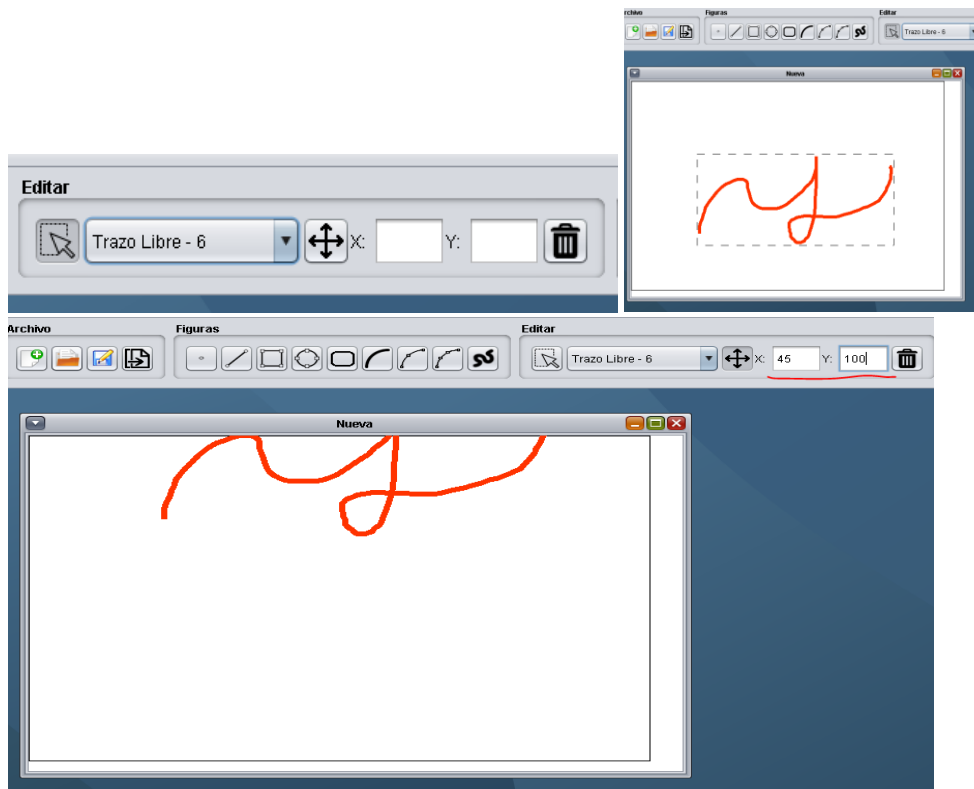
- **Panel figuras:** seleccione una y dibújela usando el click del ratón. Para todas las figuras excepto el punto será necesario arrastrar el ratón al menos una vez. Para la figura quad será necesario hacerlo dos veces y para cubic tres. En la segunda imagen se muestran las distintas formas que se pueden dibujar.



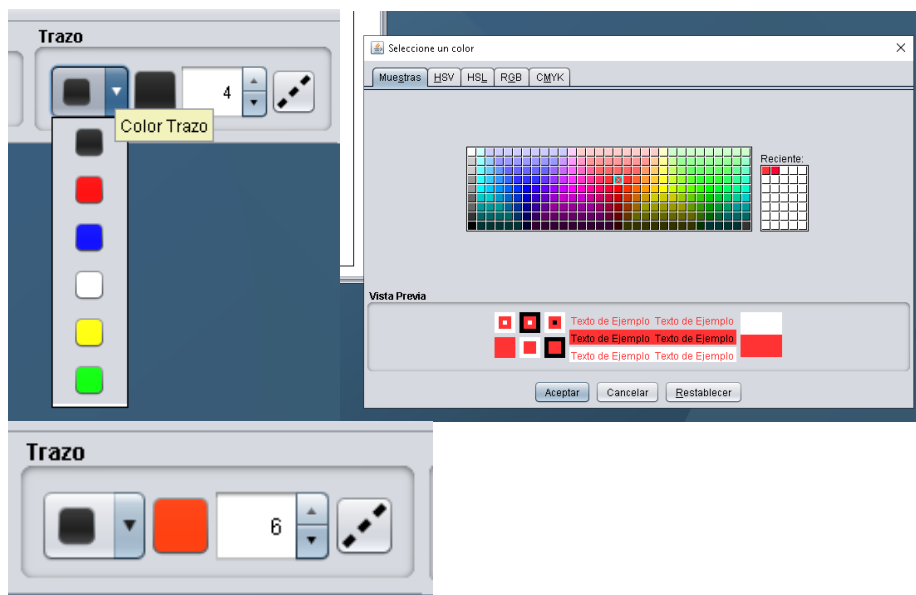
- **Opciones de deshacer (Ctrl+Z) y rehacer (Ctrl+Y):** puede deshacer y rehacer figuras dibujadas mediante el uso de estas opciones. Entre la segunda y tercera imagen se ha utilizado la operación deshacer.



- **Panel Editor:** seleccione una forma de la lista o pulse el botón seleccionar y clique sobre ella para seleccionarla (mientras está seleccionada puede modificar sus atributos). Modifique la posición de una figura usando el botón mover y arrastrándola o selecciónela e introduzca las nuevas coordenadas en los campos de texto X: e Y:. Por último puede borrar formas usando el botón borrar y clicando sobre una. En la segunda imagen se muestra una forma seleccionada, y en la tercera una forma desplazada usando los campos de texto.

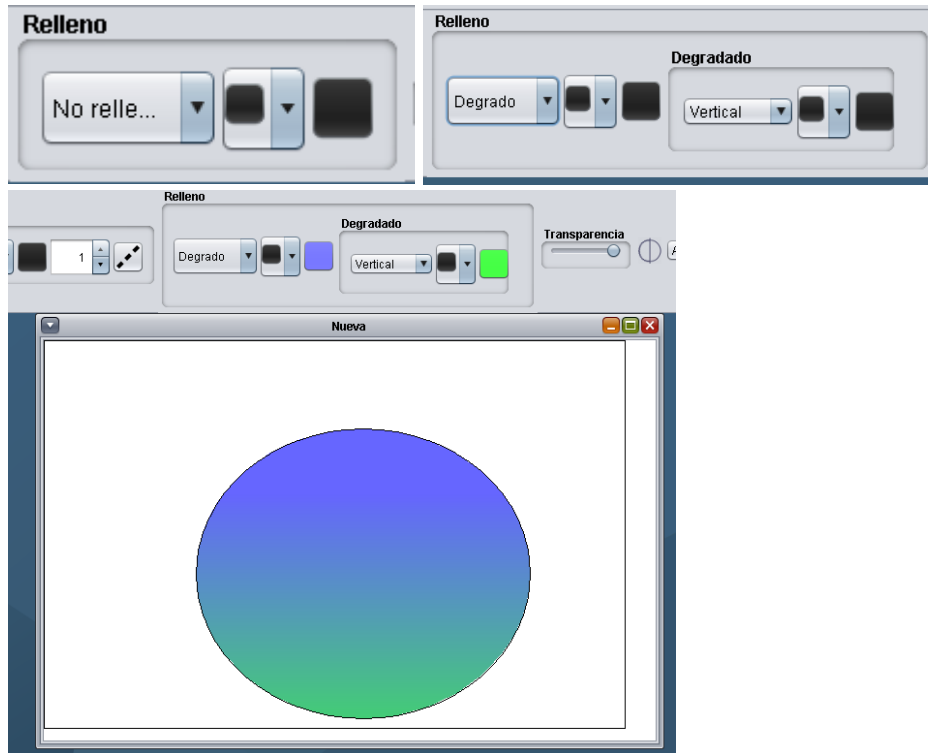


- **Panel Trazo:** puede cambiar el color del trazo usando los colores básicos que se ofrecen en el primer desplegable o usar una paleta de colores que abre el segundo botón. El número del tercer elemento representa el grosor del trazo y el cuarto botón si el trazo es discontinuo o no. En la primera imagen se muestra la lista de colores básicos y en la segunda la paleta de colores.

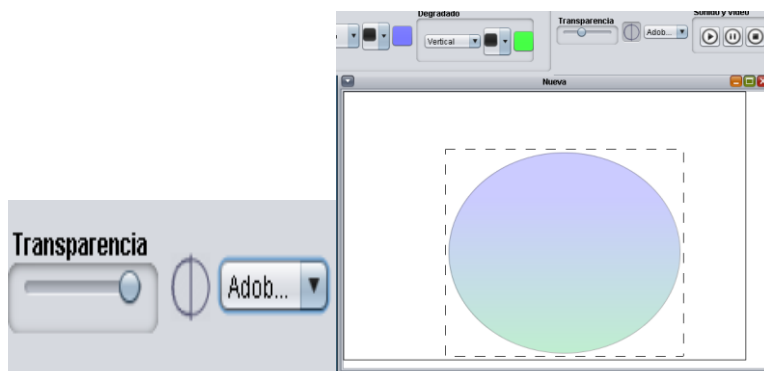




- **Panel Relleno:** ofrece distintas opciones de relleno de figura así como su color. Si seleccionamos la opción de degradado, se abrirá otro panel con las opciones de color y dirección del degradado. En la primera imagen se muestra el panel relleno sin seleccionar la opción degradado, en la segunda el panel extendido con el degradado y en la tercera imagen se muestra una figura con relleno degradado.



- **Transparencia, Alisado y Fuente:** use el deslizador para modificar la transparencia de la figura a dibujar, seleccione el botón de alisado para dibujar con bordes más suaves y el desplegable para seleccionar la fuente de texto (entrada de texto no disponible en esta versión). En la imagen se muestra la figura dibujada en el apartado anterior con una transparencia del 30% y alisado de bordes.





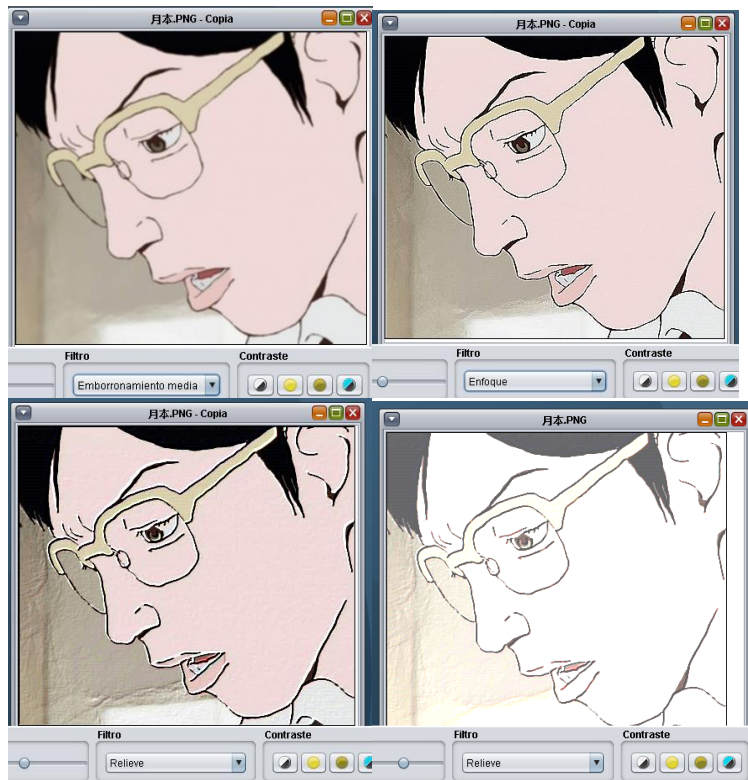
## Imagen:

Las opciones de imagen se encuentran en la barra de herramientas inferior.

- **Brillo:** mueva el deslizador para modificar el brillo de una imagen abierta.



- **Filtro:** seleccione uno de los filtros para aplicarlo a una imagen abierta. La cuarta imagen muestra la aplicación del filtro “Relieve” a la imagen con brillo modificado del apartado anterior.



- **Contraste:** al pulsar los botones sobre una imagen abierta se aplica el contraste relacionado al botón seleccionado. Las imágenes muestran los contrastes aplicados sobre la misma imagen en el orden de los botones de la sección.

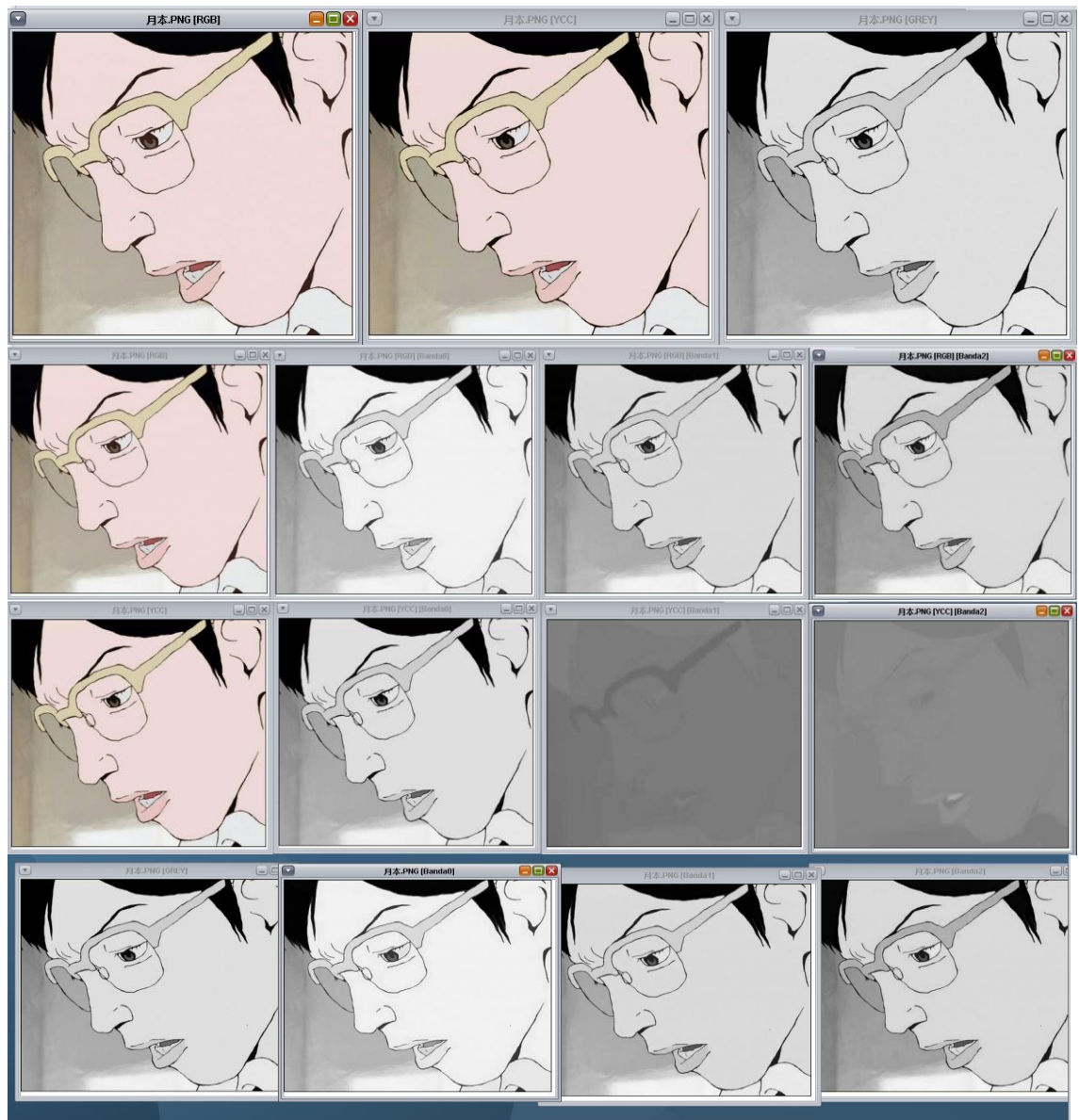


- **Operadores:** al igual que con el contraste, cada operador modifica la imagen abierta según la función asociada. Las imágenes muestran los operadores aplicados sobre la misma imagen, el orden de las imágenes es igual al de los operadores de la sección operadores.

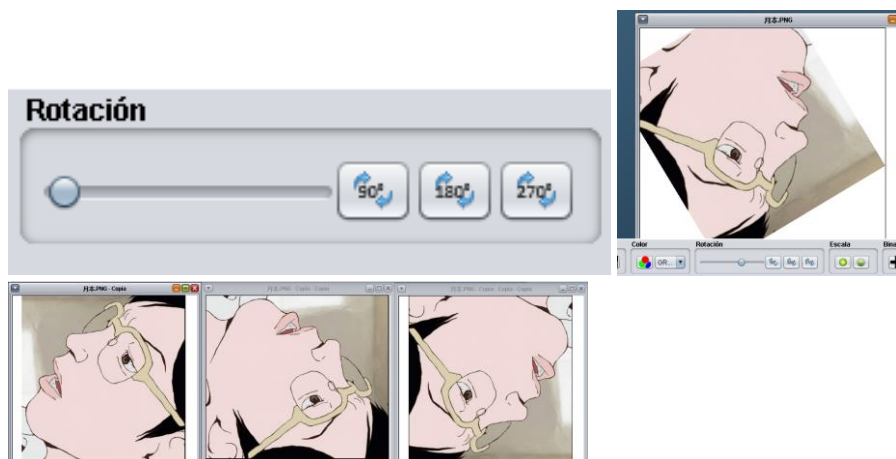


- **Panel Color:** el botón extrae las bandas de la imagen abierta y el desplegable cambia el espacio de color de la imagen al seleccionado creando una nueva ventana. El set de tres imágenes muestra la imagen original en cada uno de los espacios de color. Los siguientes tres sets de cuatro imágenes muestra cada espacio de color con sus tres bandas.

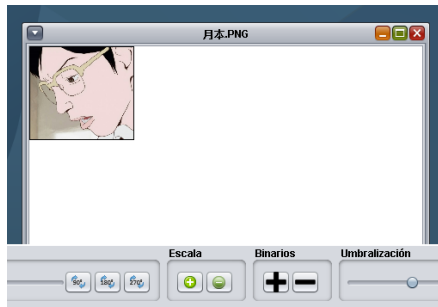




- **Panel Rotación:** rota la imagen abierta usando el deslizador o los botones con ángulos prefijados. El set de tres imágenes muestra la imagen modificada con los botones con ángulos prefijados.



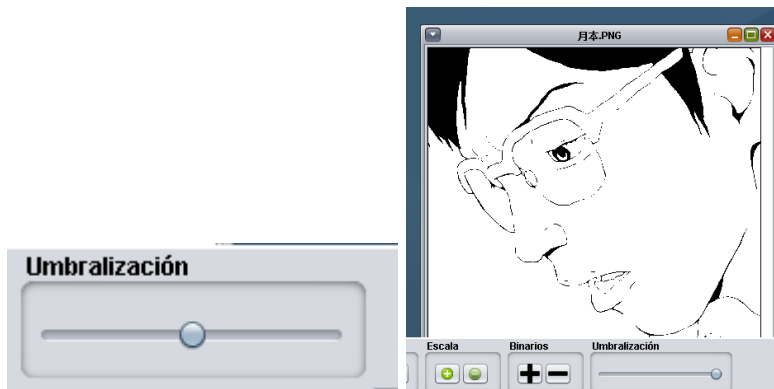
- **Panel Escala:** escala el tamaño de la imagen con los botones aumentar y disminuir.



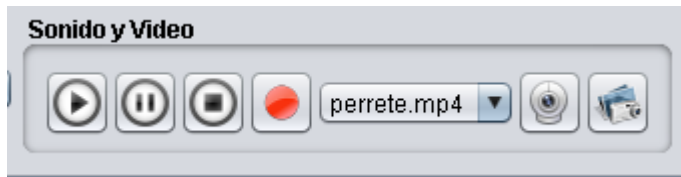
- **Panel Binarios:** utilice las operaciones suma y resta mediante dos imágenes abiertas. La imagen muestra un ejemplo de la operación suma.



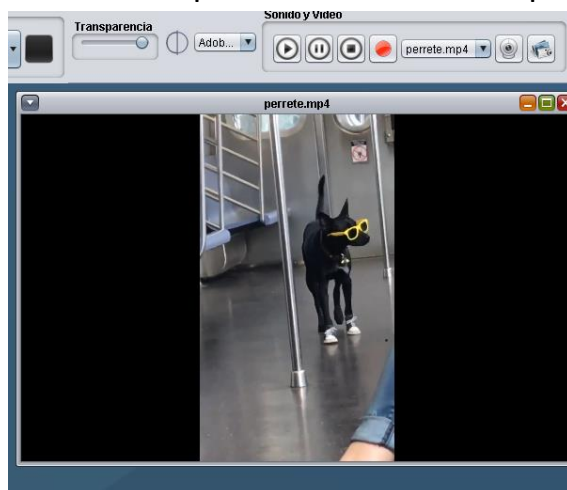
- **Panel Umbralización:** cambie la umbralización de la imagen abierta arrastrando el deslizador.



## Sonido y Video:



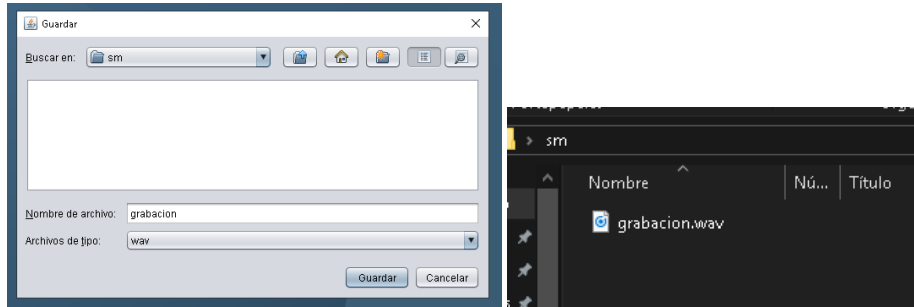
- **Lista de reproducción:** desplegable en el centro del panel. El elemento seleccionado es sobre el que actuarán los botones Play, Pausa y Stop. Como se ha mencionado en el apartado “Archivo”, al abrir un archivo de sonido o video, se añadirá a esta lista.
- **Botón Play:** Reproduce el audio seleccionado en la lista de reproducción. En el caso de un video, abre una ventana y reproduce el video. La imagen muestra la reproducción de un video cargado en la lista de reproducción al darle al play.



- **Botón Pausa:** para la reproducción del elemento seleccionado en la lista de reproducción, pudiendo reanudarlo pulsando de nuevo el botón Play.
- **Botón Stop:** detiene la reproducción del elemento seleccionado en la lista de reproducción. En el caso de pulsar el botón Play, comenzará la reproducción desde el principio.



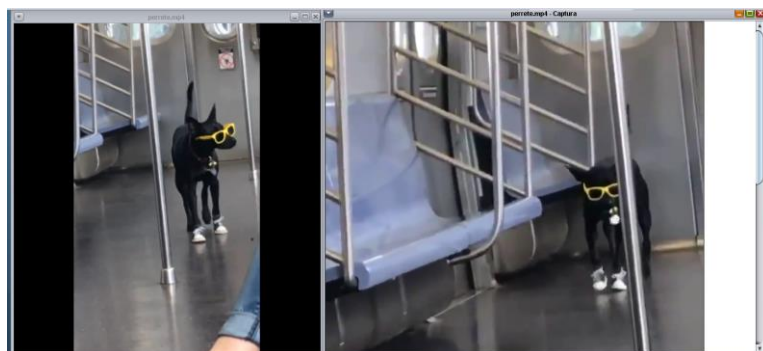
- **Botón Grabar:** al seleccionarlo abre un diálogo donde debemos elegir un directorio y nombre para el archivo que se creará. Una vez hecho procedemos a grabar usando el micrófono y acabaremos la grabación pulsando el botón Stop. La primera imagen muestra el diálogo de creación de archivo grabado y la segunda el archivo creado en el directorio seleccionado al terminar de grabar.



- **Botón Webcam:** abre una ventana con lo que esté siendo grabado por la webcam.

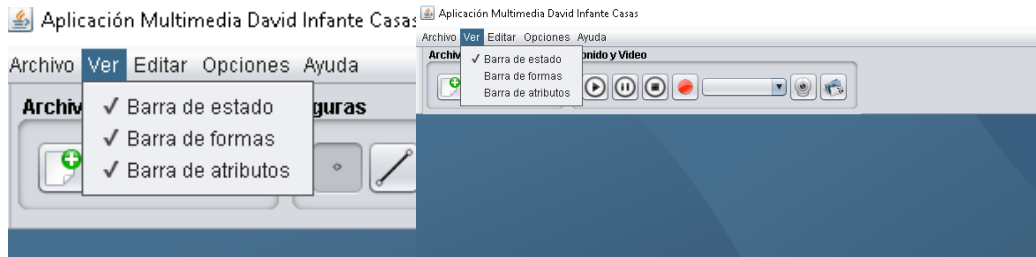


- **Botón Screenshot:** toma una captura de pantalla de una ventana tipo webcam o reproducción de video y la abre en una nueva ventana. La primera imagen muestra el video del que se ha tomado una captura (segunda imagen).

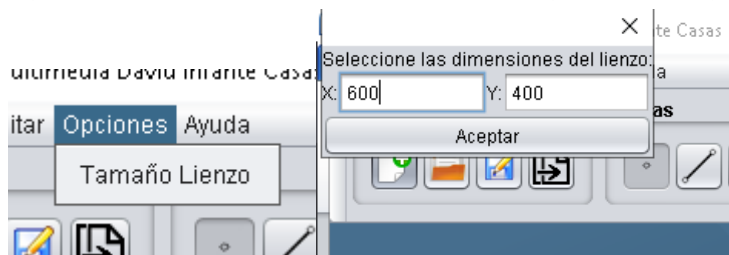


## Otro:

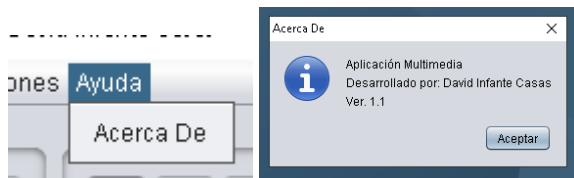
- **Menú Ver:** ofrece la posibilidad de ocultar algunas barras de herramientas. La segunda imagen muestra el menú al deseleccionar la barra de formas y la barra de atributos.



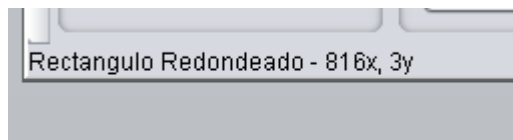
- **Menú Opciones:** contiene la opción de poner modificar los parámetros por defecto del tamaño de creación de los lienzos. La opción Tamaño Lienzo abre el diálogo de la segunda imagen.



- **Menú Ayuda:** muestra información acerca del programa.



- **Barra de Estado:** muestra el tipo de forma que se está dibujando y la posición del cursor en el lienzo.



# **Bibliografía**

Documentación oficial de Oracle: <https://docs.oracle.com/javase/>

Guiones de prácticas de la asignatura

Material de teoría de la asignatura

Obtención de componentes RGB de un píxel:

<https://alvinalexander.com/blog/post/java/getting-rgb-values-for-each-pixel-in-image-using-java-bufferedimage>