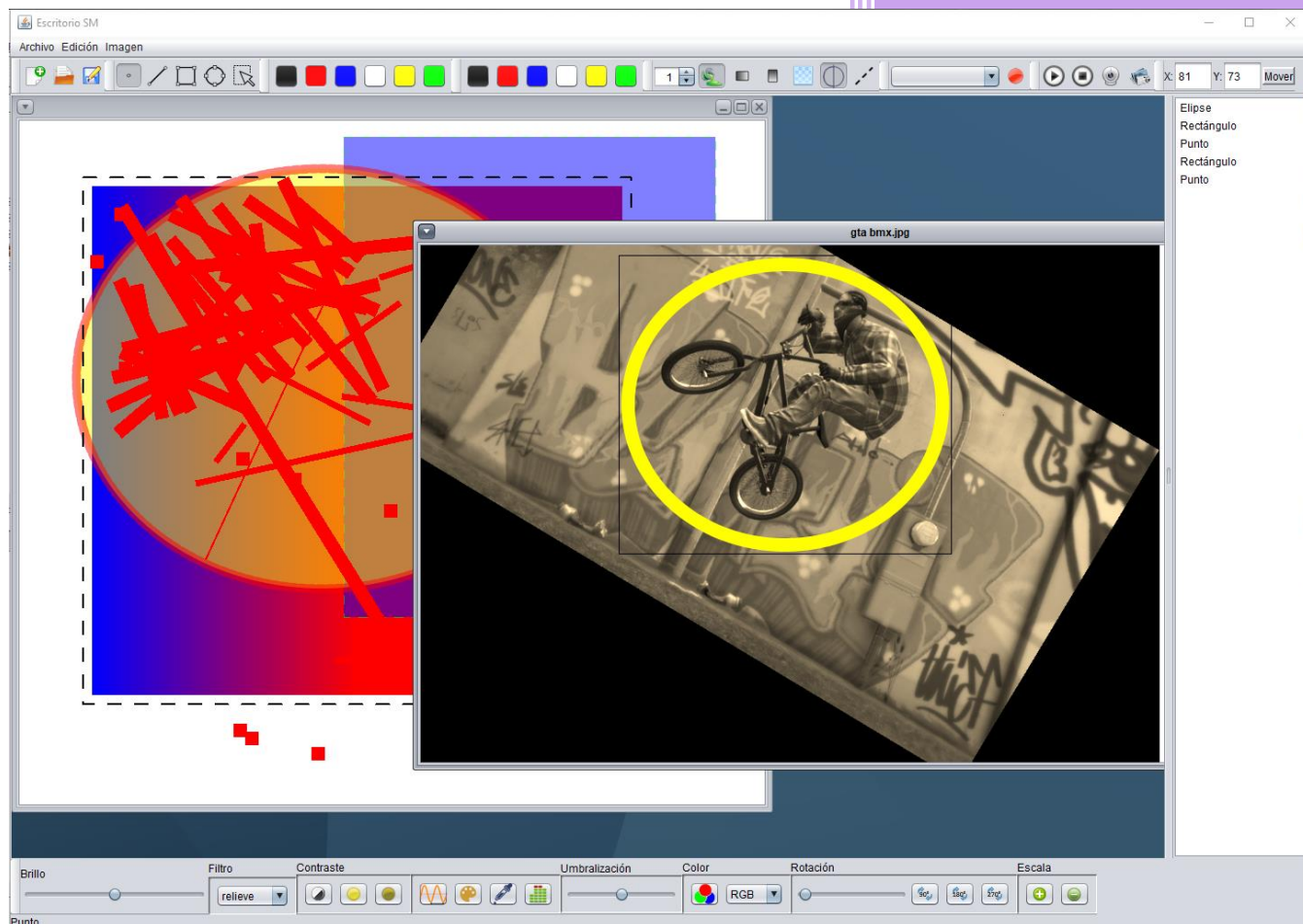


2019

# Documentación



Antonio Galdó Seiquer  
Sistemas Multimedia - UGR

## 1 INTRODUCCIÓN

---

Esta es la documentación de la aplicación desarrollada para la asignatura Sistemas Multimedia de la Universidad de Granada, así como la correspondiente biblioteca que la acompaña y le da capacidad para funcionar realmente.

Esta aplicación servirá en términos generales para edición de imágenes, así como reproducción de sonido y vídeo. Más concretamente será capaz de redimensionar imágenes, aplicar diferentes tipos de filtros, girar la imagen, pintar figuras sobre ella, separar sus capas (RGB o YCbCr) y cambiar el espacio de color. Las figuras podrán ser puntos, líneas, rectángulos y elipses, y cada una con propiedades como el color, el grosor de la línea o la transparencia.

Con respecto al vídeo se podrá reproducir un archivo de vídeo en una nueva ventana, o reproducir la *Webcam*, e incluso sacar una "captura" y editarla o tratarla como una imagen más.

Como he indicado, el proyecto consta de dos partes: La aplicación (El escritorio con las ventanas y los botones) y la biblioteca SM.AGS.Biblioteca, donde están las clases de los lienzos, los filtros y las figuras.

## 2 REQUISITOS FUNCIONALES DE LA APLICACIÓN

---

### 2.1 ASPECTOS GENERALES

En la barra superior se muestran una serie de menús con varias opciones que son las siguientes:

- Con el botón "Nuevo" se puede crear una nueva imagen con fondo blanco en una nueva ventana del escritorio.
- Con el botón "Guardar" se puede guardar una imagen con las todas las modificaciones que se hayan hecho sobre el lienzo a un archivo PNG.
- Con el botón "Abrir" se permite abrir tres tipos de archivos: Imagen, video o sonido. Al abrir una imagen se creará una ventana interna con las dimensiones de la imagen y tendrá la imagen como un objeto sobre el que se pueden realizar modificaciones. El sonido y video se explicará en la sección *Video y sonido*.
- La opción "Imagen" muestra una serie de opciones para aplicar sobre una imagen. La opción "Duplicar" crea otra ventana interna con una imagen igual a la seleccionada, y las demás opciones son distintos filtros que se aplican directamente al pulsar.
- Con la opción "Ver" se muestran tres opciones para mostrar o no las tres barras del programa (barra de herramientas superior, barra de herramientas inferior, y la barra de estado)
- La opción "Ayuda" tiene una única opción: "Acerca de" que muestra el programa, la versión y el nombre del autor.

Lo siguiente que hay es un grupo de botones que nos permiten seleccionar qué forma pintaremos. Esto cambia la variable que las selecciona en el lienzo de la ventana interna que haya seleccionada, y hará que se cree un objeto del tipo de la figura que le corresponda (el Punto es un objeto de tipo línea cuyo inicio y fin coinciden).

Tras esto se muestran dos grupos de botones con colores, que se usarán para seleccionar el color de las figuras y el filtro "tintar". El primer grupo de colores servirá para los márgenes de las figuras y para el filtro, y el segundo grupo para el relleno de las figuras.

Después tenemos opciones de personalización para las figuras, que son el grosor, si hay o no relleno, si el relleno es degradado, si el degradado es en dirección vertical u horizontal, si hay transparencia, si hay alisado de bordes y si el trazo es discontinuo o no.

A esto le sigue la lista de reproducción de sonidos mencionada anteriormente, y los botones para reproducir, parar, grabar, usar la webcam y capturar la pantalla, explicados en su sección correspondiente.

Por último, tenemos dos campos de coordenadas y un botón de mover, para poder modificar las coordenadas de una figura (previamente dibujada por el usuario) que seleccionemos en la lista lateral.

## 2.2 DIBUJO

Al lanzar una Ventana Interna se incluye un lienzo y en este se pueden dibujar todas las formas que se quieran (del tipo línea, rectángulo, elipse y punto), cada una con propiedades propias (color, trazo, transparencia, posición o alisado), las cuales se pueden modificar una vez creada ya la figura. Las formas, una vez creadas aparecerán en una lista a la derecha de la aplicación, que cambiará al cambiar de una ventana interna a otra, y nos permitirá seleccionar las formas para cambiar sus atributos.

Cada vez que se crea una figura se lanza un evento (de la clase propia de eventos creada para el lienzo, *LienzoEvent*) y esto hace que sea posible que la instancia de ese lienzo notifique al escritorio donde están todas las ventanas internas, y se actualice la lista de formas.

Al seleccionar una figura en la lista, se crea un rectángulo discontinuo que la rodea (a 10 píxeles de distancia) para indicar visualmente qué forma es.

## 2.3 PROCESAMIENTO DE IMÁGENES

Se puede realizar una serie de acciones sobre la imagen que haya en uno de los lienzos. Están repartidas entre el menú superior de "Imagen" y la barra de herramientas inferior.

### 2.3.1 Menú superior

- **"Duplicar"**: duplica la imagen creando otra ventana interna igual.
- **"Negativo"**: aplica el filtro negativo que usa un objeto "LookUpOp".
- **"LookUpOp"**: aplica un filtro creado por mí que inserta mucho ruido a la imagen oscureciéndola. La función usada es  $0.5 * I^2$  siendo  $I$  un valor entre 0 y 255.
- **"CompACompOp"**: crea un objeto de la clase "CompACompOp" y aplica su filtro sobre la imagen (se explica más adelante).
- **"PixelAPIxelOp"**: crea un objeto de la clase "PixelAPIxelOp" y aplica su filtro sobre la imagen (se explica más adelante).

### 2.3.2 Barra inferior

Diferentes opciones que afectarán únicamente a la imagen de la ventana interna seleccionada:

- Una *slider* para cambiar el brillo de la imagen.
- Una lista desplegable de filtros que se aplicarán definitivamente al seleccionar otra opción.
- Tres opciones para contrastar la imagen, iluminarla u oscurecerla, cambiando el nivel de intensidad.
- Cuatro filtros predeterminados:
  - Sinusoidal: se aplicará una función sinusoidal con "LookUpOp".
  - Sepia: se aplicará el filtro de un objeto "SepiaOp" (explicado más adelante).
  - Tintado: se aplicará el filtro de un objeto "TintOp".
  - Ecuilizado: se aplicará el filtro de un objeto "EqualizationOp".

## 2.4 SONIDO

La aplicación permite reproducir sonidos con formatos soportados por Java, y permite grabar el sonido del micrófono en un archivo. Primero hay que indicar la ubicación y el nombre del archivo, y después se empieza a grabar hasta que damos al botón de stop y se termina de guardar el archivo.

## 2.5 VÍDEO Y SONIDO

Al abrir un archivo de vídeo se abre una ventana interna nueva que inicialmente está en negro, pero con los botones de "Reproducir" y "Parar" podremos reproducirlo. Esto es posible gracias al reproductor embebido de VLC. Si se abre un archivo de sonido se incluirá en la lista de reproducción desplegable y se podrá reproducir o parar con los mismos botones que un vídeo.

La aplicación permite grabar sonido entrante del micrófono con el botón "Grabar". Primero hay que indicar la ubicación y el nombre del archivo (tendremos que incluir la extensión), y después capturará el sonido del micrófono hasta que pulsemos "Parar", y lo guardará en ese momento. Adicionalmente tenemos un botón de "Webcam", que abrirá una ventana interna en la que reproducirá la webcam del sistema. Y un botón de "Captura", con el cual podremos obtener una imagen del momento actual mientras se está reproduciendo un archivo de vídeo o la webcam (como sacar una foto).

# 3 BIBLIOTECA SM.AGS.BIBLIOTECA

---

La biblioteca que he desarrollado se divide en tres paquetes: "sm.agsgraficos", "sm.agsimagen" y "sm.agsiu".

## 3.1 PAQUETE DE GRÁFICOS

En este paquete se encuentran las clases que representan las formas que se podrán dibujar en la aplicación (requisito de dibujo). Hay una clase abstracta de la cual hereda el resto, llamada "Forma", que contiene todos los atributos (excepto los puntos de la forma, que son personalizados para cada una) métodos de "get" y set, así como los métodos principales "Paint" y "Aplicar". Excepto los métodos simples de "get" y "set", la mayoría de estos métodos están sobrescritos en cada una de las otras clases, que son "Linea2D", "Linea", "Rectangulo" y "Elipse". "Linea2D" es usada para la geometría (coordenadas) de la línea, y las otras tres son en sí las figuras que podremos dibujar.

Los métodos que se sobrescriben en cada una de las figuras son los siguientes:

- "Paint": llama a aplicar y pinta la figura con relleno o sin ello, excepto si es una línea.
- "aplicar" (en Rectangulo y Elipse): se le pasa como argumento el objeto Graphics2D y aplica sobre él las propiedades que lleva la figura (color, trazo...).
- "setLocation" y "getLocation": según el tipo de figura tratan con el punto de la esquina superior izquierda, o un punto cualquiera en caso de la línea.
- "toString": devuelve la palabra que corresponda a la figura.
- "getSelection": devuelve un objeto rectángulo con trazo discontinuo, que rodea a la figura con 10 píxeles de margen.
- "setPoints": al igual que los constructores, sirve para indicar las coordenadas de la figura.
- "contains": indica si un punto pertenece o no a la figura (no usado en la aplicación).
- Como la clase Forma no tiene constructor, cada clase personalizada que hereda de ella sí tiene varios personalizados, con argumentos para las coordenadas de la figura.

Esta clase es abstracta porque no está creada para que se defina una instancia de ella, si no para usar las clases que heredan de ella.

### 3.2 PAQUETE DE IU

En este paquete se encuentran los distintos tipos de lienzos que se usan en el proyecto. Todos vienen de "javax.swing.JPanel" por lo que son lo que en el proyecto se incluirá en las ventanas internas que contenga nuestro escritorio. Nos encontramos con las siguientes clases:

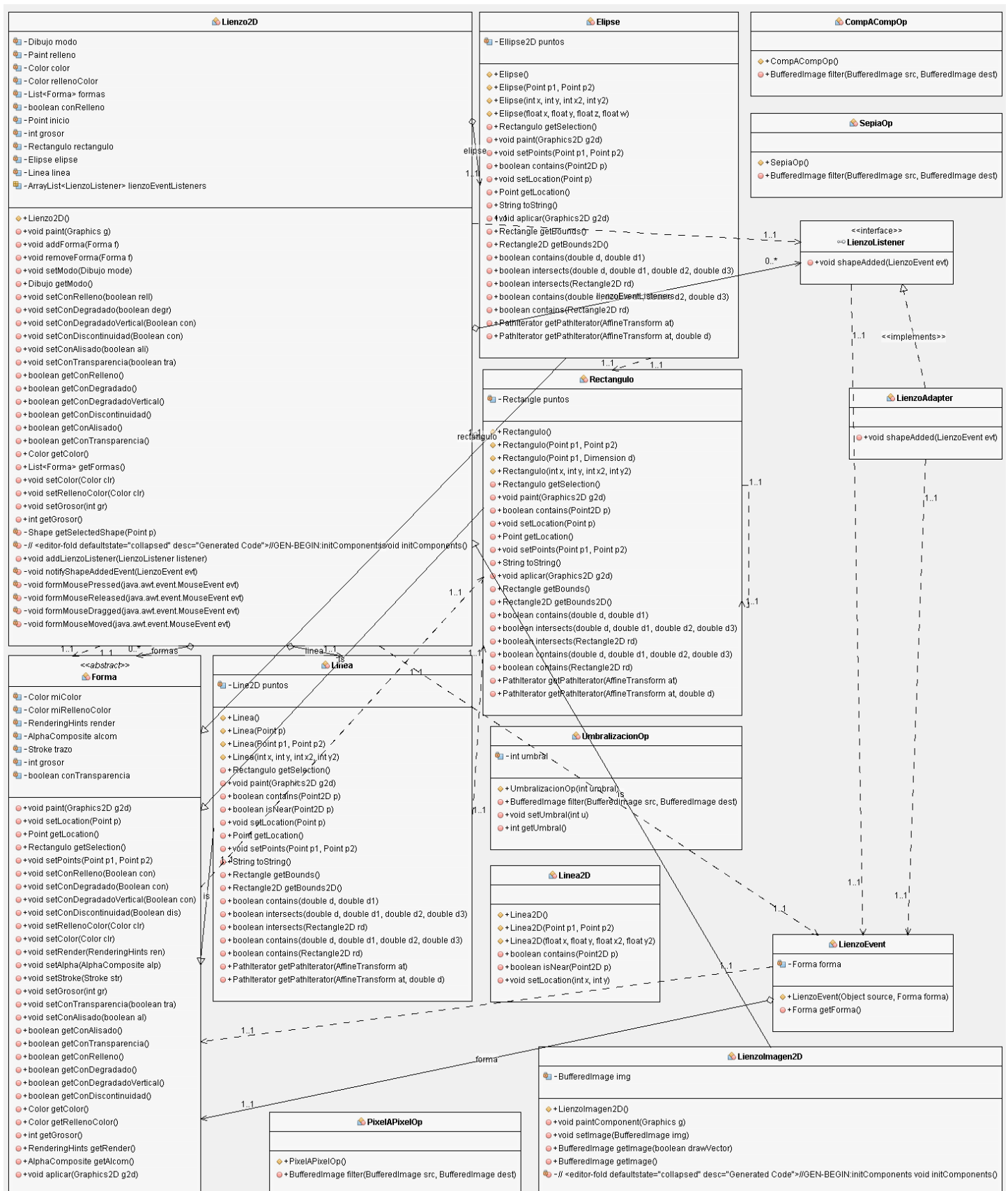
- **"Lienzo2D"**: clase que hereda directamente de "JPanel" y que es la que más se ha usado en la aplicación. Tendrá los mismos atributos que tiene la clase Forma, esto es porque cuando se cree una figura se le traspasarán estos atributos y serán propios de la figura. También hay 3 variables temporales que son las posibles formas que se dibujarán (una línea, rectángulo o elipse) y la variable de tipo "enum" que indica qué figura se quiere dibujar. Los métodos que contiene, a parte de los básicos "get" y set son para notificar un evento (que se lanzará cuando se cree una figura, y en este caso servirá para actualizar la lista de la aplicación) y los manejadores de eventos al hacer *click* en el lienzo, que según las variables definidas (en este caso) por la ventana principal, realizarán ciertas funciones como dibujar una figura.
- **"LienzoEvent"**: clase evento cuya funcionalidad es avisar de cuándo una figura es creada o modificada. Se incluye en ella información de la figura en particular.
- **"LienzoListener"**: clase que escuchará al evento para poder gestionarlo.
- **"LienzoAdapter"**: clase que implemente la anterior.
- **"LienzoImagen2D"**: hereda directamente de Lienzo2D con funciones especiales para cargar una imagen de fondo y poder aplicar filtros sobre ella. Simplemente tiene una variable "BufferedImage" y, en el método "paint", se pinta al fondo del lienzo. Tiene lógicamente los métodos para obtener la imagen y para establecerla. El método "getImage" con parámetro true devuelve una copia de la imagen, en vez de la original.

### 3.3 PAQUETE DE IMAGEN

Aquí se encuentran las clases que he creado para los diferentes filtros que podremos aplicar a los píxeles de una imagen de tipo "BufferedImage". Todas tienen un constructor vacío y un método "filter" al que se le pasan dos imágenes (origen y destino), aplica el filtro sobre la imagen destino y también la devuelve. Estas son las diferentes clases:

- **"SepiaOp"**: filtro que, a través de una función, mezclando las tres capas de color RGB, le da un aspecto desgastado a la imagen.
- **"UmbralizacionOp"**: este filtro hace que la imagen pase a ser binaria (teniendo solamente color blanco o negro en todos sus píxeles. Según el umbral de tolerancia, necesitará más o menos intensidad para pasar los píxeles a blanco o negro.
- **"CompACompOp"**: filtro creado por mí que tiene un bucle que recorre los 3 componentes de la imagen y realiza una operación muy sencilla. En este caso multiplica el color por 0.7 para quitar algo de color e intensidad y que la imagen se vuelva más gris.
- **"PixelAPixelOp"**: filtro creado por mí que opera sobre los 3 componentes de cada píxel en una misma iteración. Lo que hace es rotar los colores RGB, y los intercambia para que el resultado sea GBR. Si se realiza este filtro 3 veces se vuelve a la imagen original.

4



PracticaFinal
• static void main(String args)

VentanaInternaVLCPlayer
<ul style="list-style-type: none"> <li>- EmbeddedMediaPlayer vlcplayer</li> <li>- File Media</li> </ul>
<ul style="list-style-type: none"> <li>+ VentanaInternaVLCPlayer()</li> <li>- VentanaInternaVLCPlayer(File f)</li> <li>+ static VentanaInternaVLCPlayer getInstance(File f)</li> <li>+ void play()</li> <li>+ void stop()</li> <li>+ void addMediaPlayerEventListener(MediaPlayerEventListener ml)</li> <li>+ BufferedImage getImage()</li> <li>- // &lt;editor-fold defaultstate="collapsed" desc="Generated Code"&gt; //GEN-BEGIN initComponents void initComponents()</li> <li>- void formInternalFrameClosing(javax.swing.event.InternalFrameEvent evt)</li> </ul>

VentanaInternaJMFPlayer
<ul style="list-style-type: none"> <li>- Player player</li> <li>- File Media</li> </ul>
<ul style="list-style-type: none"> <li>- VentanaInternaJMFPlayer(File f)</li> <li>+ static VentanaInternaJMFPlayer getInstance(File f)</li> <li>+ void play()</li> <li>+ void stop()</li> <li>+ void close()</li> <li>- // &lt;editor-fold defaultstate="collapsed" desc="Generated Code"&gt; //GEN-BEGIN initComponents void initComponents()</li> <li>- void formInternalFrameClosing(javax.swing.event.InternalFrameEvent evt)</li> </ul>

VentanaInternaCamara
<ul style="list-style-type: none"> <li>- Webcam camara</li> </ul>
<ul style="list-style-type: none"> <li>- VentanaInternaCamara()</li> <li>+ static VentanaInternaCamara getInstance()</li> <li>+ void close()</li> <li>+ BufferedImage getImage()</li> <li>- // &lt;editor-fold defaultstate="collapsed" desc="Generated Code"&gt; //GEN-BEGIN initComponents void initComponents()</li> <li>- void formInternalFrameClosing(javax.swing.event.InternalFrameEvent evt)</li> </ul>

VentanaInterna
<ul style="list-style-type: none"> <li>- static int fx</li> <li>- static char modo</li> <li>- javax.swing.JScrollPane jScrollPane1</li> <li>- sm.ags.lu.LienzoImagen2D lienzo</li> </ul>
<ul style="list-style-type: none"> <li>+ VentanaInterna()</li> <li>+ Lienzo2D getLienzo()</li> <li>- // &lt;editor-fold defaultstate="collapsed" desc="Generated Code"&gt; //GEN-BEGIN initComponents void initComponents()</li> </ul>

+ VentanaPrincipal()
- // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN initComponents void initComponents()
+ void enlazarEventoVentana(VentanaInterna v)
- void nuevoMousePressed(javax.awt.event.MouseEvent evt)
- void verBarraOpActionPerformed(javax.awt.event.ActionEvent evt)
- void lapizActionPerformed(javax.awt.event.ActionEvent evt)
- void lineaActionPerformed(javax.awt.event.ActionEvent evt)
- void rectanguloActionPerformed(javax.awt.event.ActionEvent evt)
- void elipseActionPerformed(javax.awt.event.ActionEvent evt)
- void abrirActionPerformed(javax.awt.event.ActionEvent evt)
- void guardarActionPerformed(javax.awt.event.ActionEvent evt)
- void escritorioMousePressed(javax.awt.event.MouseEvent evt)
- void grosorSpinnerStateChanged(javax.swing.event.ChangeEvent evt)
- void rescaleMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void conRellenoActionPerformed(javax.awt.event.ActionEvent evt)
- void conTransparenciaActionPerformed(javax.awt.event.ActionEvent evt)
- void conMoverActionPerformed(javax.awt.event.ActionEvent evt)
- void brilloSliderFocusGained(javax.awt.event.FocusEvent evt)
- void brilloSliderFocusLost(javax.awt.event.FocusEvent evt)
- void brilloSliderStateChanged(javax.swing.event.ChangeEvent evt)
- void negativoMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void filtrosCBoxActionPerformed(javax.awt.event.ActionEvent evt)
- void convolveMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void contrasteBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rotarSliderStateChanged(javax.swing.event.ChangeEvent evt)
- void iluminarBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void oscurecerBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void sinuosidadBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rotar180BotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rotar90BotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rotar270BotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rotarSliderFocusGained(javax.awt.event.FocusEvent evt)
- void rotarSliderFocusLost(javax.awt.event.FocusEvent evt)
- void aumentarBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void disminuirBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void bandasBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void cambioDeColorCBoxActionPerformed(javax.awt.event.ActionEvent evt)
- void abrirBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void guardarBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void nuevoBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void escritorioMouseMoved(javax.awt.event.MouseEvent evt)
- void duplicarMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void tntarBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void negroBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void rojoBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void azulBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void blancoBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void amarilloBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void verdeBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void equalizarBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void playBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void stopBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void recordBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void CamaraBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void capturaCamaraBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void listaFigurasValueChanged(javax.swing.event.ListSelectionEvent evt)
- void moverBotonActionPerformed(javax.awt.event.ActionEvent evt)
- void listaFigurasFocusLost(javax.awt.event.FocusEvent evt)
- void conAlisadoActionPerformed(javax.awt.event.ActionEvent evt)
- void LookUpMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void negroBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void rojoBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void azulBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void blancoBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void amarilloBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void verdeBoton1ActionPerformed(javax.awt.event.ActionEvent evt)
- void conDegradadoActionPerformed(javax.awt.event.ActionEvent evt)
- void sepiabotonActionPerformed(javax.awt.event.ActionEvent evt)
- void umbralSliderFocusGained(javax.awt.event.FocusEvent evt)
- void umbralSliderStateChanged(javax.swing.event.ChangeEvent evt)
- void umbralSliderFocusLost(javax.awt.event.FocusEvent evt)
- void PixelAPIxelMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void CompACompMenuActionPerformed(javax.awt.event.ActionEvent evt)
- void conDegradadoVerticalActionPerformed(javax.awt.event.ActionEvent evt)
- void conDiscontinuidadActionPerformed(javax.awt.event.ActionEvent evt)

VentanaPrincipal
<ul style="list-style-type: none"> <li>- BufferedImage imgSource</li> <li>- String estado</li> <li>- Kernel k</li> <li>- Forma formaAux</li> <li>- Lienzo2D millienzo</li> <li>- SMPPlayer player</li> <li>- SMRecorder recorder</li> <li>- MIManejadorLienzo manejador</li> <li>- javax.swing.JButton CamaraBoton</li> <li>- javax.swing.JMenuItem CompACompMenu</li> <li>- javax.swing.JMenuItem LookUpMenu</li> <li>- javax.swing.JMenuItem PixelAPIxelMenu</li> <li>- javax.swing.JMenuItem abrir</li> <li>- javax.swing.JButton abrirBoton</li> <li>- javax.swing.JToggleButton amarilloBoton</li> <li>- javax.swing.JButton aumentarBoton</li> <li>- javax.swing.JToggleButton azulBoton</li> <li>- javax.swing.JToggleButton azulBoton1</li> <li>- javax.swing.JButton bandasBoton</li> <li>- javax.swing.JLabel barraEstado</li> <li>- javax.swing.JToolBar barraSonido</li> <li>- javax.swing.JToggleButton blancoBoton</li> <li>- javax.swing.JToggleButton blancoBoton1</li> <li>- javax.swing.JSlider brilloSlider</li> <li>- javax.swing.JComboBox&lt;String&gt; cambioDeColorCBox</li> <li>- javax.swing.JButton capturaCamaraBoton</li> <li>- javax.swing.ButtonGroup colores</li> <li>- javax.swing.JToggleButton conAlisado</li> <li>- javax.swing.JToggleButton conDegradado</li> <li>- javax.swing.JToggleButton conDegradadoVertical</li> <li>- javax.swing.JToggleButton conDiscontinuidad</li> <li>- javax.swing.JToggleButton conMover</li> <li>- javax.swing.JToggleButton conRelleno</li> <li>- javax.swing.JToggleButton conTransparencia</li> <li>- javax.swing.JButton contrasteBoton</li> <li>- javax.swing.JMenuItem convolveMenu</li> <li>- javax.swing.JButton disminuirBoton</li> <li>- javax.swing.JMenuItem duplicarMenu</li> <li>- javax.swing.JButton equalizarBoton</li> <li>- javax.swing.JToggleButton elipse</li> <li>- javax.swing.JDesktopPane escritorio</li> <li>- javax.swing.JComboBox&lt;String&gt; filtrosCBox</li> <li>- javax.swing.ButtonGroup formas</li> <li>- javax.swing.JSpinner grosorSpinner</li> <li>- javax.swing.JMenuItem guardar</li> <li>- javax.swing.JButton guardarBoton</li> <li>- javax.swing.JButton iluminarBoton</li> <li>- javax.swing.JLabel jLabel1</li> <li>- javax.swing.JLabel jLabel11</li> <li>- javax.swing.JLabel jLabel12</li> <li>- javax.swing.JLabel jLabel13</li> <li>- javax.swing.JLabel jLabel14</li> <li>- javax.swing.JLabel jLabel15</li> <li>- javax.swing.JLabel jLabel16</li> <li>- javax.swing.JLabel jLabel17</li> <li>- javax.swing.JLabel jLabel18</li> <li>- javax.swing.JLabel jLabel19</li> <li>- javax.swing.JMenu jMenu1</li> <li>- javax.swing.JMenu jMenu2</li> <li>- javax.swing.JMenuBar jMenuBar1</li> <li>- javax.swing.JPanel jPanel1</li> <li>- javax.swing.JPanel jPanel10</li> <li>- javax.swing.JPanel jPanel13</li> <li>- javax.swing.JPanel jPanel14</li> <li>- javax.swing.JPanel jPanel15</li> <li>- javax.swing.JPanel jPanel16</li> <li>- javax.swing.JPanel jPanel17</li> <li>- javax.swing.JPanel jPanel18</li> <li>- javax.swing.JPanel jPanel19</li> <li>- javax.swing.JPanel jPanel2</li> <li>- javax.swing.JPanel jPanel20</li> <li>- javax.swing.JPanel jPanel21</li> <li>- javax.swing.JPanel jPanel22</li> <li>- javax.swing.JPanel jPanel23</li> <li>- javax.swing.JPanel jPanel24</li> <li>- javax.swing.JPanel jPanel25</li> <li>- javax.swing.JPanel jPanel26</li> <li>- javax.swing.JPanel jPanel3</li> <li>- javax.swing.JPanel jPanel30</li> <li>- javax.swing.JPanel jPanel31</li> <li>- javax.swing.JPanel jPanel4</li> <li>- javax.swing.JPanel jPanel5</li> <li>- javax.swing.JPanel jPanel6</li> <li>- javax.swing.JPanel jPanel7</li> <li>- javax.swing.JPanel jPanel8</li> <li>- javax.swing.JScrollPane jScrollPane1</li> <li>- javax.swing.SplitPane jSplitPane1</li> <li>- javax.swing.JToolBar jToolBar1</li> <li>- javax.swing.JToolBar jToolBar3</li> <li>- javax.swing.JToolBar jToolBar4</li> <li>- javax.swing.JToolBar jToolBar5</li> <li>- javax.swing.JToolBar jToolBar6</li> <li>- javax.swing.JToolBar jToolBar7</li> <li>- javax.swing.JToggleButton lapiz</li> <li>- javax.swing.JToggleButton linea</li> <li>- javax.swing.JComboBox&lt;File&gt; listaDeReproduccion</li> <li>- javax.swing.JList&lt;Forma&gt; listaFiguras</li> <li>- javax.swing.JMenu menuImagen</li> <li>- javax.swing.JButton moverBoton</li> <li>- javax.swing.JMenuItem negativoMenu</li> <li>- javax.swing.JToggleButton negroBoton</li> <li>- javax.swing.JToggleButton negroBoton1</li> <li>- javax.swing.JMenuItem nuevo</li> <li>- javax.swing.JButton nuevoBoton</li> <li>- javax.swing.JButton oscurecerBoton</li> <li>- javax.swing.JButton playBoton</li> <li>- javax.swing.JButton recordBoton</li> <li>- javax.swing.JToggleButton rectangulo</li> <li>- javax.swing.ButtonGroup relleno</li> <li>- javax.swing.JMenuItem rescaleMenu</li> <li>- javax.swing.JToggleButton rojoBoton</li> <li>- javax.swing.JToggleButton rojoBoton1</li> <li>- javax.swing.JButton rotar180Boton</li> <li>- javax.swing.JButton rotar270Boton</li> <li>- javax.swing.JButton rotar90Boton</li> <li>- javax.swing.JSlider rotarSlider</li> <li>- javax.swing.JButton sepiaboton</li> <li>- javax.swing.JButton sinuosidadBoton</li> <li>- javax.swing.JButton stopBoton</li> <li>- javax.swing.JTextField textoLocationX</li> <li>- javax.swing.JTextField textoLocationY</li> <li>- javax.swing.JButton tntarBoton</li> <li>- javax.swing.JToolBar toolbarDeColores</li> <li>- javax.swing.JToolBar toolbarDeColores1</li> <li>- javax.swing.JSlider umbralSlider</li> <li>- javax.swing.JCheckBoxMenuItem verBarraOp</li> <li>- javax.swing.JToggleButton verdeBoton</li> <li>- javax.swing.JToggleButton verdeBoton1</li> </ul>