

**SISTEMAS MULTIMEDIA (2016-2017)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Documentación práctica final

---

Cristian Vélez Ruiz

4 de septiembre de 2017

## Índice

<b>1</b>	<b>Identificación del problema</b>	<b>4</b>
<b>2</b>	<b>Información</b>	<b>4</b>
<b>3</b>	<b>Análisis de requisitos funcionales</b>	<b>4</b>
<b>4</b>	<b>Casos de uso</b>	<b>6</b>
<b>5</b>	<b>Análisis y diseño</b>	<b>44</b>
5.1	Análisis y diseño modulo de gráficos . . . . .	44
5.2	Análisis de audio . . . . .	46
5.3	Análisis de vídeo . . . . .	46
<b>6</b>	<b>Codificación y empaquetado de la librería</b>	<b>47</b>
6.1	Paquete gráficos . . . . .	47
6.2	Paquete interfaz usuario . . . . .	49
6.3	Paquete imagen . . . . .	49
<b>7</b>	<b>Manual de usuario</b>	<b>51</b>
7.1	Menú principal . . . . .	51
7.2	Menú de herramientas . . . . .	52
7.3	Menú de efectos . . . . .	53

## Índice de figuras

4.1.	Gestión de datos . . . . .	6
4.2.	Gestión de dibujo . . . . .	10
4.3.	Gestión de efectos 1 . . . . .	17
4.4.	Gestión de efectos 2 . . . . .	24
4.5.	Gestión de efectos 3 . . . . .	31
4.6.	Gestión de sonido . . . . .	38
4.7.	Gestión de Vídeo . . . . .	41
5.1.	Diagrama de clases . . . . .	46
6.1.	Diagrama de clases . . . . .	50
7.1.	Vista principal . . . . .	51
7.2.	Vista principal . . . . .	51
7.3.	Zona 1 del menú de herramientas . . . . .	52
7.4.	Zona 2 del menú de herramientas . . . . .	52
7.5.	Zona 3 del menú de herramientas . . . . .	52
7.6.	Zona 4 del menú de herramientas . . . . .	52
7.7.	Brillo . . . . .	53
7.8.	Filtros de convolución . . . . .	53
7.9.	Efectos diversos . . . . .	53

7.10. Contraste . . . . .	54
7.11. Rotación . . . . .	54
7.12. Escalado . . . . .	54
7.13. Umbralizar . . . . .	55

## 1. Identificación del problema

Desarrollo de un software para la edición de contenido multimedia.

## 2. Información

Tenemos la necesidad de trabajar con imágenes, vídeos y sonido.

- **Imágenes:** Las imágenes se deben poder crear a partir de nuestro software desde cero, también debe ser posible que se puedan cargar imágenes desde fuentes externas al programa. A su vez las imágenes pueden ser tratadas mediante distinto tratamiento de imágenes, como son filtros, rotaciones, cambios de espacio de color, extracción de bandas, o simplemente pintar figuras sobre ellas. Debido al manejo de imágenes también todo el tratamiento debe ser guardado sobre la imagen para que esta se pueda guardar en fuentes externas.
- **Sonido:** Debe ser capaz de capturar sonido y reproducir sonido.
- **Vídeos:** En el apartado de vídeos, debemos ser capaces de poder reproducir vídeos, ya sea por fuente externa o a través de la webcam del ordenador, también debe dejar realizar capturas de la webcam y que se puedan manipular como una imagen.

## 3. Análisis de requisitos funcionales

Análisis de requisitos funcionales de cada uno de los apartados,

**RFI\_1: Gestión de Imágenes.**

- **RFI\_1.1: Gestión de datos**
  - **RFI\_1.1.1:** Crear una imagen en blanco
  - **RFI\_1.1.2:** Abrir imagen
  - **RFI\_1.1.3:** Guardar imagen
- **RFI\_1.2: Gestión de dibujo**
  - **RFI\_1.2.1:** Pintar líneas rectas
  - **RFI\_1.2.2:** Pintar rectángulos
  - **RFI\_1.2.3:** Pintar elipses
  - **RFI\_1.2.4:** Pintar curva con un punto de control
  - **RFI\_1.2.5:** Pintar texto formateado
  - **RFI\_1.3.6:** Pintar puntos

- **RFI\_1.3: Gestión de efectos de imagen**
  - **RFI\_1.3.1:** Duplicar imagen
  - **RFI\_1.3.2:** Cambiar brillo
  - **RFI\_1.3.3:** Emborronamiento
  - **RFI\_1.3.4:** Enfoque
  - **RFI\_1.3.5:** Relieve
  - **RFI\_1.3.6:** Contraste
  - **RFI\_1.3.7:** Iluminar
  - **RFI\_1.3.8:** Oscurecer
  - **RFI\_1.3.9:** Negativo de la imagen
  - **RFI\_1.3.10:** Extracción de las bandas
  - **RFI\_1.3.11:** Conversión de espacios de colores
  - **RFI\_1.3.12:** Giro de la imagen
  - **RFI\_1.3.13:** Escalado
  - **RFI\_1.3.14:** Efecto sepia
  - **RFI\_1.3.15:** Efecto tintado
  - **RFI\_1.3.16:** Ecualización
  - **RFI\_1.3.17:** Umbralización
  - **RFI\_1.3.18:** Cambio de colores entre R-G-B

#### **RFI\_2: Gestión de sonido.**

- **RFS\_2.1:** Grabar sonido webcam
- **RFS\_2.2:** Reproducir sonido

#### **RFI\_3: Gestión de vídeo.**

- **RFV\_3.1:** Reproducir video
- **RFV\_3.2:** Reproducir webcam
  - **RFV\_3.2.1:** Captura webcam

## 4. Casos de uso

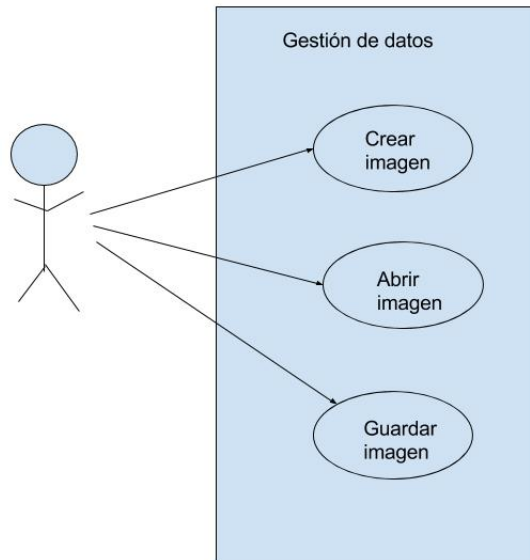


Figura 4.1: Gestión de datos

Caso de Uso	Crear imagen en blanco				CU_01	
Actores	Usuario					
Tipo	Primario					
Referencias	RF_1.1.1					
Precondición	Se debe conocer los valores de la imagen a crear					
Postcondición	Se almacena una imagen en blanco					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario crear una imagen en blanco para trabajar sobre ella.</i>

<b>Resumen</b>
<i>El usuario indica la creación de una imagen en blanco especificando las medidas.</i>

Caso de Uso	Abrir imagen				CU_02	
Actores	Usuario					
Tipo	Primario					
Referencias	RF_1.1.2					
Precondición	La imagen debe existir en el sistema					
Postcondición	El usuario tendrá la imagen disponible para poder editarla					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario abrir una imagen de su sistema</i>

<b>Resumen</b>
<i>El usuario seleccionará que imagen desea abrir, una vez realizado, el programa <u>cargara</u> la imagen, se la mostrará y el usuario podrá editarla.</i>



Caso de Uso	Guardar imagen				CU_03
Actores	Usuario				
Tipo	Primario				
Referencias	RF_1.1.3				
Precondición	El usuario tiene que tener abierta una imagen.				
Postcondición	En su sistema de ficheros se le creará la imagen.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Poder guardar las imágenes que se crean en el programa.</i>

<b>Resumen</b>
<i>Una vez tenga cargada en el programa una imagen, el usuario podrá guardar la imagen en <u>sus sistema</u>.</i>

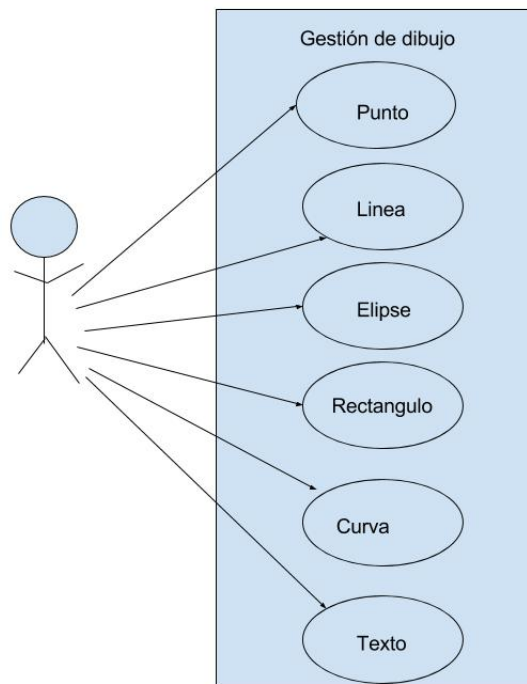


Figura 4.2: Gestión de dibujo

Caso de Uso	Pintar línea recta				CU_04
Actores	Usuario				
Tipo	Primario				
Referencias	RF1_1.2.1				
Precondición	El usuario debe tener una imagen activa.				
Postcondición	Se almacena una recta en la imagen.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario dibujar una línea recta en la imagen.</i>

<b>Resumen</b>
<i>Una vez el usuario tenga abierta una imagen en el programa, podrá pintar una línea recta en ella, con las propiedades que deseé.</i>

Caso de Uso	Pintar rectángulo				CU_05
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.2.2				
Precondición	El usuario debe tener una imagen activa.				
Postcondición	Se creará un rectángulo en la imagen <u>activa</u> .				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>El usuario podrá dibujar un rectángulo en la imagen.</i>

<b>Resumen</b>
<i>El usuario podrá dibujar un rectángulo en la imagen con las propiedades que deseé.</i>

Caso de Uso	Pintar elipse				CU_06
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.2.3				
Precondición	El usuario debe tener abierta una imagen donde pintar.				
Postcondición	Se creará una elipse en la imagen.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Pintar elipses en las imágenes.</i>

<b>Resumen</b>
<i>El usuario puede pintar elipses con las propiedades que deseé en la imagen activa.</i>

Caso de Uso	Pintar curva con punto de control				CU_07	
Actores	Usuario					
Tipo	Primario					
Referencias	RFI_1.2.4					
Precondición	El usuario debe tener una imagen activa.					
Postcondición	Se creará una curva en la imagen.					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Pintar curvas con punto de control en la imagen.</i>



<b>Resumen</b>
<i>El usuario puede pintar una curva con punto de control en la imagen activa y con las propiedades que deseé.</i>

Caso de Uso	Pintar texto formateado				CU_08
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.2.5				
Precondición	El usuario debe haber facilitado el texto y su formato.				
Postcondición	Se crea un texto en la imagen como indicó el usuario				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Pintar texto formateado en la imagen</i>

<b>Resumen</b>
<i>El usuario puede escribir texto en la imagen, con las propiedades que le indicó.</i>

Caso de Uso	Pintar puntos				CU_09	
Actores	Usuario					
Tipo	Primario					
Referencias	RFI_1.2.6					
Precondición	Debe haber una imagen activa_					
Postcondición	Se crea un punto dentro de la imagen.					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Dibujar un punto en la imagen.</i>

<b>Resumen</b>
<i>El usuario puede dibujar puntos dentro de la imagen activa, con las propiedades seleccionadas.</i>



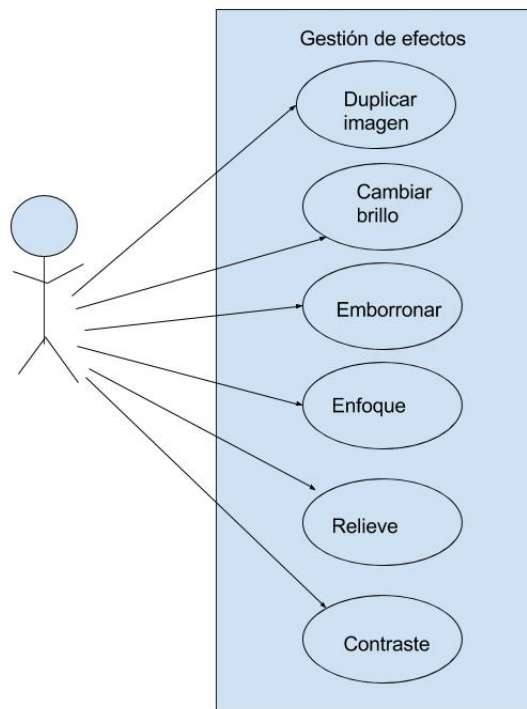


Figura 4.3: Gestión de efectos 1

Caso de Uso	Duplicar imagen				CU_10	
Actores	Usuario					
Tipo	Primario					
Referencias	RFI_1.3.1					
Precondición	El usuario tiene que tener una imagen activa.					
Postcondición	Al usuario se le crea una nueva ventana con la imagen					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario duplicar la imagen activa.</i>

3

<b>Resumen</b>
<i>Tras tener activa una imagen en el programa, el usuario puede duplicarla, una vez realizado podrá editarla./</i>

Caso de Uso	Cambiar brillo			CU_11	
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.2				
Precondición	El usuario tiene que tener abierta una imagen.				
Postcondición	La imagen queda con el brillo deseado por el usuario.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario cambiar el brillo de la imagen activa.</i>

<b>Resumen</b>
<i>El usuario elige mediante un deslizador el valor del brillo y este se le aplicará a la imagen.</i>

Caso de Uso	Cancelar reserva				CU_12
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.3				
Precondición	El usuario tiene que tener una imagen activa.				
Postcondición	Obtiene una imagen emborronada.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario realizar el emborronamiento de la imagen activa.</i>



<b>Resumen</b>
<i>El usuario, tras tener activa una imagen, puede emborronarla.</i>

Caso de Uso	Enfoque				CU_13
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.4				
Precondición	El usuario tiene que tener abierta una imagen.				
Postcondición	Al usuario se le proporcionará la imagen más enfocada.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario que pueda enfocar una imagen.</i>

<b>Resumen</b>
<i>El usuario con una imagen activa, puede aplicarle un filtro para mejorar el enfoque.</i>

Caso de Uso	Relieve				CU_14
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.5				
Precondición	El usuario debe tener una imagen activa.				
Postcondición	Obtiene una imagen el relieve destacado.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario aplicar a su imagen el efecto de relieve.</i>

<b>Resumen</b>
<i>El usuario con una imagen activa, puede aplicar el efecto relieve, una vez realizado, obtiene la imagen con el relieve más/destacado.</i>

Caso de Uso	Contraste				CU_15
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.6				
Precondición	El usuario tiene que tener activa una imagen.				
Postcondición	El usuario obtiene la imagen con el efecto de contraste aplicado.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

Propósito
<i>Permitir al usuario aplicar un efecto de contraste a la imagen.</i>

Resumen
<i>Cuando el usuario tiene activa una imagen, puede aplicarle un efecto de contraste.</i>

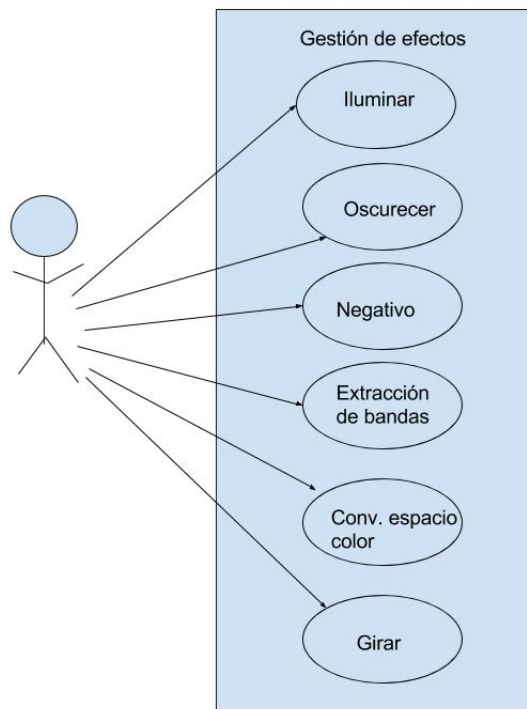


Figura 4.4: Gestión de efectos 2



Caso de Uso	Iluminar	CU_16			
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.7				
Precondición	El usuario tiene que tener activa una imagen.				
Postcondición	El usuario obtiene una imagen más iluminada.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario iluminar una imagen.</i>

<b>Resumen</b>
<i>El usuario tiene activa una imagen, aplica este efecto y obtiene una imagen más iluminada.</i>

Caso de Uso	Oscurecer			CU_17	
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.8				
Precondición	El usuario tiene que tener activa una imagen.				
Postcondición	Obtiene una imagen más oscurecida.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario oscurecer una imagen.</i>

<b>Resumen</b>
<i>El usuario al tener una imagen activa, si se aplica este efecto, obtiene la misma imagen pero más oscurecida.</i>

Caso de Uso	Negativo de la imagen				CU_18	
Actores	Usuario					
Tipo	Primario					
Referencias	RFI_1.3.9					
Precondición	El usuario debe tener activa una imagen.					
Postcondición	Obtiene el negativo de la imagen activa.					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario obtener el negativo de la imagen activa.</i>

<b>Resumen</b>
<i>El usuario solicita aplicar el filtro a la imagen activa, una vez aplicado se obtiene el negativo en la imagen posteriormente activa.</i>

Caso de Uso	Extracción de bandas				CU_19
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.10				
Precondición	Debe haber una imagen activa.				
Postcondición	El usuario obtiene tantas imágenes como bandas tenga la imagen.				
Autor	Cristian Vélez Ruiz	Fecha	01-04-16	Versión	1.0

<b>Propósito</b>
<i>Permitir obtener las bandas de la imagen para poder editarlas.</i>

<b>Resumen</b>
<i>El usuario obtiene las bandas de una imagen para poder editarlas, si se edita la banda, también se aplica en la imagen “madre”.</i>

Caso de Uso	Conversión de espacio de color				CU_20	
Actores	Usuario					
Tipo	Primario					
Referencias	RFI_1.3.11					
Precondición	El usuario debe tener una imagen activa.					
Postcondición	Obtiene una imagen con el espacio de color deseado.					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario cambiar el espacio de color de la imagen.</i>

<b>Resumen</b>
<i>El usuario puede cambiar el espacio de color de la imagen activa.</i>

Caso de Uso	Giro de la imagen				CU_21
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.12				
Precondición	El usuario tiene que tener activa una imagen.				
Postcondición	Obtendrá la imagen rotada los grados que desee.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario girar los grados que deseé la imagen.</i>

<b>Resumen</b>
<i>Tras tener activa una imagen, permite al usuario realizar un giro libre a la imagen.</i>

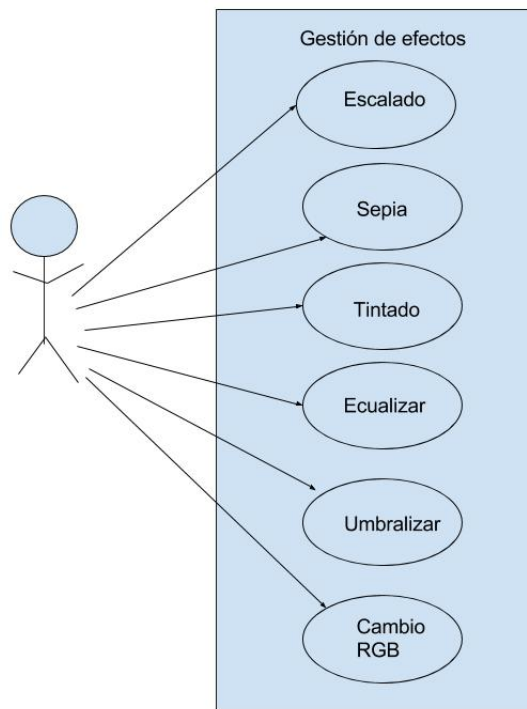


Figura 4.5: Gestión de efectos 3

Caso de Uso	Escalado				CU_22
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.13				
Precondición	Debe haber una imagen activa.				
Postcondición	Obtiene la imagen ampliada o reducida.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario ampliar/reducir la imagen.</i>

<b>Resumen</b>
<i>Tras tener una imagen activa, el usuario podrá ampliar/reducir la imagen.</i>



Caso de Uso	Efecto sepia				CU_23
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.14				
Precondición	El actor tiene que tener una imagen activa.				
Postcondición	El usuario obtiene la imagen con el efecto sepia aplicado.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

Propósito
<i>Aplica el efecto sepia a la imagen activa.</i>

Resumen
<i>Una vez el usuario tenga una imagen activa, se le aplicará el efecto sepia</i>

Caso de Uso	Efecto tintado			CU_24	
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.15				
Precondición	El actor tiene que tener una imagen activa.				
Postcondición	El usuario obtiene la imagen con el efecto tinte aplicado.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Aplica el efecto tinte a la imagen activa con el color deseado.</i>

<b>Resumen</b>
<i>Una vez el usuario tenga una imagen activa y el color seleccionado, se le aplicará el efecto tinte.</i>

Caso de Uso	Ecualización				CU_25
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.16				
Precondición	El usuario tiene que tener activa una imagen.				
Postcondición	Obtiene una imagen ecualizada				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Ecualiza la imagen</i>

<b>Resumen</b>
<i>Tras tener activa una imagen, se le aplica el filtro para ecualizar la imagen, el usuario obtiene sobre la imagen anterior la nueva imagen.</i>

Caso de Uso	Umbralización				CU_26
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.17				
Precondición	El usuario tiene que tener una imagen activa.				
Postcondición	Genera una nueva imagen umbralizada				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario umbralizar una imagen con en el valor deseado.</i>

<b>Resumen</b>
<i>Tras tener la imagen activa, el usuario puede realizar la Umbralización con el valor deseado.</i>

Caso de Uso	Cambio de colores entre R-G-B				CU_27
Actores	Usuario				
Tipo	Primario				
Referencias	RFI_1.3.18				
Precondición	Tiene que haber activa una imagen.				
Postcondición	Obtiene la imagen con el valor de los colores cambiado				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Cambiar los valores entre RGB</i>

<b>Resumen</b>
<i>Tras tener una imagen activa, el usuario puede rotar los valores de R, G Y B.</i>

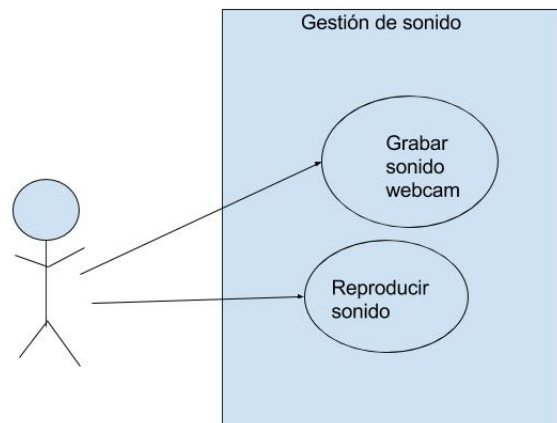


Figura 4.6: Gestión de sonido

Caso de Uso	Grabar sonido webcam				CU_28	
Actores	Usuario					
Tipo	Primario					
Referencias	RFS_2.1					
Precondición						
Postcondición	Obtiene un fichero de audio en el sistema					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0	

<b>Propósito</b>
<i>Permitir al usuario grabar audio a través de la webcam.</i>

<b>Resumen</b>
<i>Permitir al usuario grabar audio a través de la webcam en formato sin comprimir.</i>

Caso de Uso	Reproducir sonido			CU_29	
Actores	Usuario				
Tipo	Primario				
Referencias	RFS_2.1				
Precondición	El fichero de sonido debe existir.				
Postcondición					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario reproducir sonido.</i>

3.

<b>Resumen</b>
<i>El usuario seleccionará el fichero de audio que quiere reproducir, una vez realizado se reproducirá./</i>



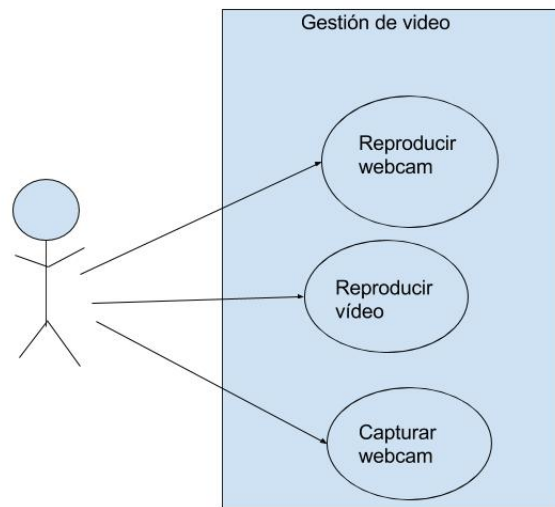


Figura 4.7: Gestión de Vídeo

Caso de Uso	Reproducir video				CU_30
Actores	Usuario				
Tipo	Primario				
Referencias	RFV_3.1				
Precondición	El fichero de video debe existir.				
Postcondición					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario reproducir videos de su sistema de ficheros.</i>



<b>Resumen</b>
<i>Permitir al usuario reproducir videos de su sistema de ficheros en el formato que sea./</i>

Caso de Uso	Reproducir webcam				CU_31
Actores	Usuario				
Tipo	Primario				
Referencias	RFV_3.2				
Precondición	Debe haber conectada e instalada una webcam.				
Postcondición					
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario reproducir su webcam.</i>

<b>Resumen</b>
<i>Permitir al usuario reproducir la webcam de su ordenador.</i>

Caso de Uso	Captura webcam				CU_32
Actores	Usuario				
Tipo	Primario				
Referencias	RFV_3.2.1				
Precondición	Debe estar reproduciendo la webcam.				
Postcondición	Se crea una nueva ventana con la imagen.				
Autor	Cristian Vélez Ruiz	Fecha	01-09-17	Versión	1.0

<b>Propósito</b>
<i>Permitir al usuario realizar capturas mientras visualiza la webcam.</i>

<b>Resumen</b>
<i>Permitir al usuario realizar capturas de la webcam, mientras la visualiza, una vez realizada, se crea una nueva ventana con la captura./</i>

## 5. Análisis y diseño

Tras determinar los requisitos que debe cumplir el software, hemos visto claramente que están divididos en 3 apartados, gráficos, sonido y vídeo.

### 5.1. Análisis y diseño modulo de gráficos

Lo mas importante que necesitamos es que cada figura pueda tener sus propiedades, tales como un grosor variable, borde variable, color propio, una determinada transparencia, y estos se puedan modificar, pero tras conocer la clase `graphics` y `graphic2d` de Java, nos damos cuenta que para cada figura tenga sus propiedades y puedan cumplir sus requisitos funcionales es imposible realizarlo solo con esas clases, por lo que necesitamos crear nuestras propias clases, a continuación analizaré diferentes formas de hacerlo.

- **Opción 1:** Crear una clase para cada uno de los objetos, similar al `graphic2d` de java, pero añadiendo sus propiedades gráficas, de esta manera podremos heredar de las clases de `graphic2d` y añadir nuevas funcionalidades a las clases, con lo que reduciremos la complejidad de la clase, solo tendríamos que añadir nuevos métodos y redefinir el método de pintar.

**Positivo:** Se reutiliza la mayoría del código de la geometría para los objetos, de manera que podemos seguir usándolos tal y como se usan en `graphic2d`.

**Negativo:** Dificultad para instanciar a todos los objetos en la misma variable tal y como se pide, por tanto esta opción no es viable.

- **Opción 2:** Crear una jerarquía de clases, con la que tendremos solucionado el problema de que todos los objetos se puedan instanciar en la misma variable, también cada uno de esos objetos tenga como propiedad el objeto al que defina. También creando aparte clases auxiliares para definir las propiedades de los objetos, como pueden ser clases que definan el borde y el relleno, debido a que muchos objetos comparten esos atributos.

**Positivo:** Se instancia en una sola variable y re-utilización de código

**Negativo:** Hay que definir los métodos de la geometría para que sean accesibles desde fuera de la clase.

Por tanto para conseguir un buen diseño y que todas las figuras se puedan pintar llamando a la misma función, diseñare las clases siguiendo la opción 2, donde habrá una clase `MyShape` de la que todos deben heredar y obligatoriamente tiene que tener el método `draw`, para que cuando se llame a `draw()`, dibuje sea cual sea la figura que este instanciada. También realizaré una pequeña variación para facilitar el uso de las clases, crearé dos clases abstractas (`Bordeable` y `Fillerable`), donde cada una tendrán sus métodos y atributos que las definan, así sus hijos las heredarán.

- **MyShape:** Clase principal abstracta de la que todas las figuras heredan, su única propiedad es la transparencia, esta clase se usara para tener instanciada la última figura antes de ser pintada.
- **Bordeable:** Clase abstracta de la que heredaran los objetos que tengan borde y ella misma hereda de `shape`, por tanto su propiedad es un objeto de la clase `Border`.
- **Border:** Clase que define el borde de la figura, sus propiedades son grosor, color, tipo de borde y si tiene suavizado el borde.
- **Fillerable:** Clase abstracta para las figuras que tienen relleno, una figura que tiene relleno también tiene borde, por lo cual hereda de `Bordeable`, su propiedad principal es un objeto de la clase `Filler`.

- **Filler**: Clase que define las propiedades del interior de la figura como son el color (Color) y si el relleno esta activo o no.
- **MyText**: Clase que permite el uso de escritura en imagenes, hereda de MyShape y sus propiedades son texto, su fuente, su localizacion y su color.
- **MyPoint**: Clase para pintar puntos, he decidido que para mi si tiene borde (aun que no puede ser discontinuo) y por tanto hereda de Bordeable, es una linea que tiene mismo punto de inicio y fin.
- **MyLine**: Clase para pintar lineas, al tener solo borde hereda de la clase abstracta Bordeable.
- **MyCurveLine**: Clase para pintar lineas curvas, es igual que MyLine, hereda de Bordeable pero lleva a demás necesita un punto adicional como punto de control.
- **MyEllipse**: Clase para pintar elipses, es una figura geometrica que puede tener relleno, por tanto hereda de la clase abstracta Fillerable.
- **MyRectangle**: Clase para pintar rectangulos, es una figura que tambien puede tener relleno por tanto hereda de Fillerable.

Tras definir todas las clases nos lleva a la siguiente jerarquía

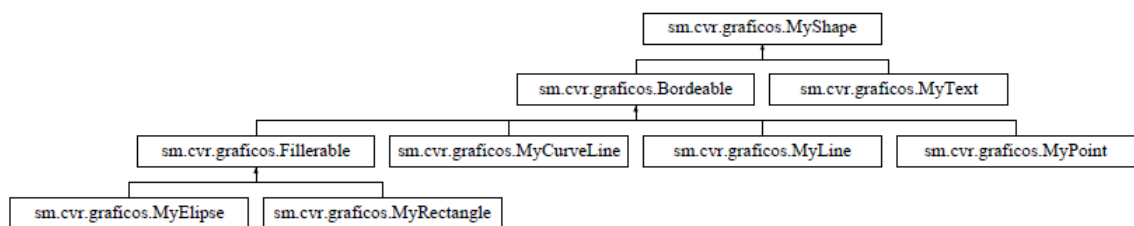


Figura 5.1: Diagrama de clases

## 5.2. Análisis de audio

Para cumplir con los requisitos funcionales se ha comprobado que con las clases que nos facilita java para trabajar con audio tenemos suficiente, por lo que no tenemos que implementar clases adicionales para trabajar con él.

Pero en este caso trabajamos con la clase que nos facilitó en la librería SMClipPlayer.

## 5.3. Análisis de vídeo

En el caso de poder reproducir vídeos y grabar desde la webcam, Java no nos da las clases necesarias para poder realizarlo directamente, por lo que para la reproducción de

vídeo he optado por usar la librería `vlcj`, la librería para Java del reproductor VLC [2], y para la visualización de la webcam opto por utilizar la clase `Webcam` de la librería de Sarxos [1], por lo que no tenemos que implementar clases adicionales para cumplir con la funcionalidad .

## 6. Codificación y empaquetado de la librería

Para organizar mejor las clases de la librería se han empaquetado en 3 paquetes,

- **sm.cvr.graficos**: En este paquete podemos encontrar las clases referentes a gráficos, principalmente encontramos las clases de nuestra jerarquía y además clases auxiliares para el manejo de las herramientas y los colores.
- **sm.cvr.iu**: En este paquete están las clases, que muestran interacción con el usuario, así que tenemos la clase principal del lienzo y también una clase con su interfaz que nos pregunta como queremos el texto, su formato y su tamaño.
- **sm.cvr.imagen**: Encontramos las clases de los efectos propios para las imágenes.

### 6.1. Paquete gráficos

- **MyShape**: Clase principal abstracta
  - Atributos:
    - transparencia (`AlphaComposite`): Utilizado para poder usar la transparencia.
  - Métodos importantes:
    - `draw(Graphic2d)`: Es el método mas importante, en esta clase es abstracto.
- **Bordeable**: Clase abstracta de la que heredan los objetos que tengan borde.
  - Hereda de: `MyShape`
  - Atributos:
    - `border (Border)`: Atributo que mantiene las propiedades del borde.
  - Métodos importantes:
    - `setBorder`: Nos permite modificar el borde.
- **Border**: Clase que define las propiedades del borde.
  - Hereda de: -
  - Atributos:
    - `continuidad (int)`: Identificador de la continuidad.
    - `grosor (float)`: Tamaño del grosor.

- borde (BasicStroke): Tipo de basic stroke a aplicar al borde.
  - color (Color): Color del borde.
  - suavizado (RenderingHints): Suavizado para evitar la pixelación del borde.
- **Fillerable**: Clase abstracta para los objetos con relleno.
  - Hereda de: Bordeable
  - Atributos:
    - filler (Filler): Atributo que almacena las propiedades del relleno.
  - Métodos importantes:
    - setFiller: Modificador del relleno.
- **Filler**: Clase para definir las propiedades del relleno.
  - Hereda de: -
  - Atributos:
    - color (Color): Color del relleno
    - activo (boolean): Determina si el relleno esta activo.
- **MyText**: Define el texto
- Hereda de: MyShape
  - Atributos:
    - font (String): Determina la fuente del texto.
    - location (Point2d): Determina la posición.
    - text (String): Texto
    - color (Color): Color del texto.
- **MyPoint**: Clase para pintar puntos.
  - Hereda de: Bordeable
  - Atributos:
    - line (Line2D.Double): Define el punto como una linea con mismo punto de inicio y fin.
- **MyLine**: Clase para pintar lineas.
  - Hereda de: Bordeable
  - Atributos:
    - line (Line2D.Double): Define la linea con inicio y fin.



- **MyCurveLine:**
  - Hereda de: Bordeable
  - Atributos:
    - curve (QuadCurve2D.Double): Define la linea curva con inicio, fin y punto de control.
- **MyEllipse:**
  - Hereda de: Fillerable
  - Atributos:
    - ellipse (Ellipse2D): Define la geometría de la elipse.
- **MyRectangle:**
  - Hereda de: Fillerable
  - Atributos:
    - rectangle (Rectangle): Define geometría del rectángulo.
- **EnumColor:** Clase auxiliar para tener un listado de los colores disponibles en el programa.
- **EnumHerramienta:** Clase auxiliar para tener un listado de las herramientas disponibles en el programa, esto nos facilita que podamos añadir mas herramientas.

## 6.2. Paquete interfaz usuario

- **AskText:** Clase para configurar las propiedades del texto.
- **Lienzo2d:** Clase principal de la librería, donde se dibuja y se maneja todo.

## 6.3. Paquete imagen

- **NegativoOp:** Clase para aplicar el efecto de negativo de la imagen.
  - Hereda de: BufferedImageOpAdapter
- **RotaColorOp:** Clase para aplicar el efecto de rotar los colores.
  - Hereda de: BufferedImageOpAdapter
- **SepiaOp:** Clase para aplicar el efecto sepia.
  - Hereda de: BufferedImageOpAdapter
- **UmbralizacionOp:** Clase para aplicar la umbralización.
  - Hereda de: BufferedImageOpAdapter

[illegible]

Para mas información sobre la librería se puede consultar el Javadoc.

## 7. Manual de usuario

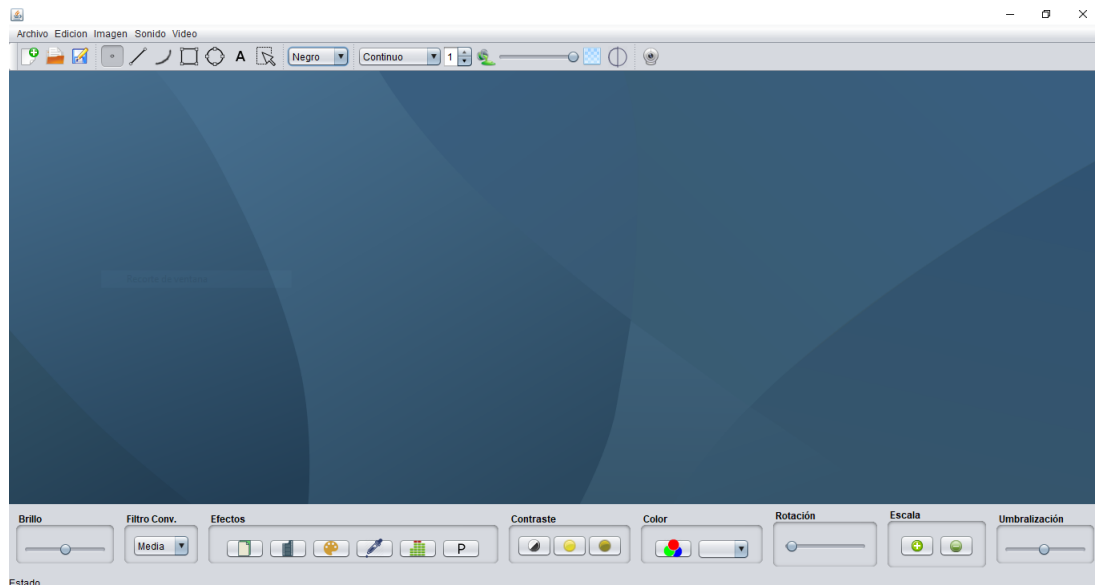


Figura 7.1: Vista principal

A continuación se detallan las funcionalidades del software.

### 7.1. Menú principal

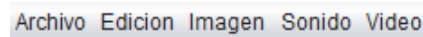


Figura 7.2: Vista principal

- **Archivo:** Nos da las tres opciones básicas para trabajar con imágenes, opción de abrir una imagen, crear una en blanco y guardar en un fichero externo.
- **Edición:** Nos permite la edición de la barra de herramientas, podemos ocultar/vissualizar la barra de estados, barra de formas y la barra de atributos.
- **Imagen:** Permite crear una imagen con las medidas especificadas.
- **Sonido:** Tenemos dos opciones, una para reproducir un fichero y otra para grabar audio a través del micrófono de la webcam.
- **Vídeo:** Tiene una opción para reproducir un fichero de vídeo.

## 7.2. Menú de herramientas

- Parte 1:



Figura 7.3: Zona 1 del menú de herramientas

Nos da las tres opciones básicas para trabajar con imágenes, opción de abrir una imagen, crear una en blanco y guardar en un fichero externo.

- Parte 2:



Figura 7.4: Zona 2 del menú de herramientas

En esta parte podemos seleccionar la herramienta con la que queremos pintar, tenemos puntos, línea, curvas, rectángulos, elipses, texto y una herramienta para poder editar figuras.

- Parte 3:



Figura 7.5: Zona 3 del menú de herramientas

Podemos editar las propiedades visuales de las figuras a pintar, desde el color con el que pintar, su trazo, su grosor, si la figura tendrá relleno (si ella lo permite), la transparencia y el suavizado del borde para evitar que se vean los píxeles.

- Parte 4:



Figura 7.6: Zona 4 del menú de herramientas

En esta opción es para poder activar la webcam y poder vernos, una vez activa se activa a su lado un botón para obtener instantáneas de la vista.

### 7.3. Menú de efectos

En esta sección principalmente encontramos los efectos que se pueden aplicar a las imágenes.

- **Brillo:**

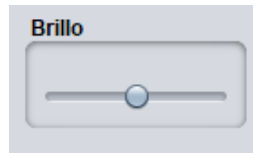


Figura 7.7: Brillo

Podemos modificar el brillo de la imagen.

- **Convoluciones:**

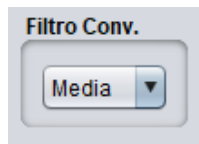


Figura 7.8: Filtros de convolución

Tenemos diferentes filtros para poder aplicar, desde detector de bordes a filtros de enfoque.

- **Efectos diversos:**

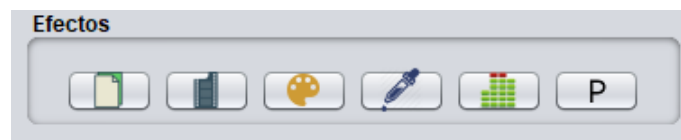


Figura 7.9: Efectos diversos

En el mismo orden de izquierda a derecha tenemos.

- Duplicado de imagen
- Negativo de la imagen
- Efecto sepia
- Tintado con el color activo
- Ecualización

- Efecto de rotación de colores

■ **Contraste:**

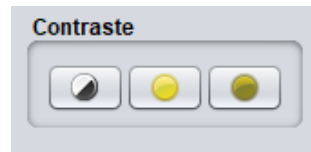


Figura 7.10: Contraste

En el mismo orden de izquierda a derecha tenemos.

- Contraste
- Efecto iluminar
- Efecto oscurecer

■ **Rotación:**

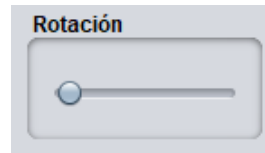


Figura 7.11: Rotación

Podemos rotar la imagen libremente desde 0 a 360°

■ **Escalado:**

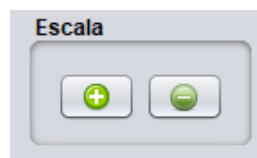


Figura 7.12: Escalado

Puedes ampliar o reducir el tamaño de la imagen.

■ **Umbralización:**

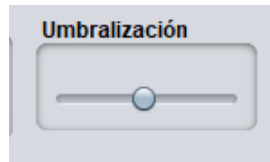


Figura 7.13: Umbralizar

Sirve para umbralizar la imagen activa.

## Referencias

- [1] Sarxos webcam proyect. <https://github.com/sarxos/webcam-capture>, Septiembre de 2017.
- [2] Vlcj reproductor. <https://github.com/caprica/vlcj>, Septiembre de 2017.