

Se pretende realizar una aplicación multimedia que permita trabajar con **imágenes** para dibujar sobre ellas y procesarlas, así como **reproducir y capturar audio y vídeo**. Para ello se desarrollará un entorno multiventana en el que se distinguirán, al menos, los siguientes tipos de ventana:

1. Ventanas de imágenes
2. Ventanas de reproducción de sonido
3. Ventanas de reproducción de vídeo
4. Ventanas de grabación de audio
5. Ventana webcam

Además, la aplicación contará con un menú y, al menos, las siguientes barras de herramientas:

1. Barra de carácter general (nuevo, abrir, guardar, etc.)
2. Barra(s) de dibujo para formas y atributos
3. Barra de operaciones sobre imágenes
4. Barra de audio/vídeo

A continuación se indicará el contenido y objetivos de cada tipo de ventana, así como de las diferentes barras de herramientas. El diseño del interfaz queda abierto a criterio del estudiante, si bien la evaluación tendrá en cuenta la complejidad del diseño, por lo que se recomienda usar el mayor número posibles de elementos Swing.

Las especificaciones indicadas en este documento son los requisitos mínimos, si bien se podrán incorporar todas las alternativas que se consideren oportunas para mejorar la calidad de la aplicación.

## ■ Tipos de Ventanas

La aplicación tendrá un escritorio central en el que podrán alojarse ventanas internas de diferentes tipos. Éstas se irán creando en función de las acciones que lleve a cabo el usuario (abrir, grabar, capturar, etc.). A continuación se describen las principales características de estas ventanas, si bien se podrán introducir cuantas mejoras se deseen para optimizar la funcionalidad o el aspecto visual.

### ● Ventanas de imágenes<sup>1</sup>

Estas ventanas mostrarán imágenes, tanto las leídas de un fichero como las creadas nuevas por el usuario o generadas al capturar una imagen instantánea a partir de un vídeo o webcam. La imagen de una de estas ventanas también se podrá guardar en fichero. Cada ventana mostrará una única imagen (habiendo, por tanto, tantas ventanas como imágenes estemos tratando). Esta ventana deberá mostrar barras de desplazamiento en caso de que la imagen sea mayor que la zona visible. El título de la ventana corresponderá al nombre del fichero, si es una imagen abierta o guardada, “nueva”, si ha sido creada por el usuario, o “captura”, si es una instantánea captada de un vídeo o de la webcam.

Sobre las imágenes que aparecen en estas ventanas se podrá dibujar (usando las formas y atributos de la barra de dibujo) y hacer operaciones de procesamiento (las incluidas en la barra de operaciones sobre imagen). Sólo se podrá dibujar sobre el área de la imagen, por lo que estas ventanas tendrán definidas como área de visualización (área “clip”) el área rectangular correspondiente a la imagen.

---

<sup>1</sup> Véase práctica 8.

Al mover el ratón sobre la imagen, se ha de indicar en la barra de estado las coordenadas del pixel sobre el que se está situado, así como su valor (color o nivel de gris, según el tipo de imagen).

- **Ventanas de reproducción de sonido<sup>2</sup>**

Estas ventanas reproducirán sonidos leídos de fichero, teniendo para ello un conjunto de botones que permitan controlar la reproducción (comienzo, parada, etc.). El número y funcionalidad de estos botones, así como el modo de interacción con el usuario, se deja a elección del estudiante. La ventana tendrá como título el nombre del fichero que se está reproduciendo. Cada ventana reproducirá un único sonido (habrá, por tanto, tantas ventanas como sonidos abiertos).

*[Opcional: mostrar la evolución de la reproducción en una barra de progreso]*

- **Ventanas de reproducción de vídeo<sup>3</sup>**

Estas ventanas reproducirán vídeos leídos de un fichero. Para ello, dispondrá de una zona de visualización central y de un conjunto de botones para controlar la reproducción (comenzar, pausar, avanzar, etc.) situados en la parte inferior. La ventana tendrá como título el nombre del fichero que se está reproduciendo, existiendo tantas ventanas como vídeos tengamos abiertos.

- **Ventanas de grabación de audio<sup>4</sup>**

La ventana de grabación permitirá almacenar en un fichero el sonido captado a través del micrófono. Para ello, la ventana contará con botones para el inicio de la grabación y su posterior parada, dejando al estudiante la elección del número y funcionalidad de estos botones, así como el modo de interacción con el usuario.

A la hora de seleccionar el fichero donde almacenar el sonido grabado, existen dos opciones: que se elija el fichero antes de comenzar la grabación (en cuyo caso la ventana estará asociada a dicho fichero y tendremos tantas ventanas como ficheros hayamos creado), o que dicho fichero se elija al finalizar el proceso de grabación (en cuyo caso sólo tendríamos una ventana de grabación que iría generando todos los ficheros).

*[Opcional: permitir al usuario definir los parámetros de digitalización (codificación, resolución, frecuencia de muestreo, etc.) y formato de fichero]*

- **Ventana webcam<sup>3</sup> [Opcional]**

La ventana webcam mostrará la secuencia que esté captando la cámara (siendo el título de la misma “webcam”).

## **Barras de herramientas general**

La aplicación contará con una barra de herramientas de carácter general que incluirá, al menos, los siguientes botones asociados a las clásicas opciones de archivo (todos ellos tendrán que tener asociado un icono y un “*ToolTipText*”):

---

<sup>2</sup> Véanse prácticas 12 y 13. Nótese que la práctica 12 sólo trabajaba con la *Java Sound API* y, por tanto, con los formatos y códecs soportados por dicha API; para la evaluación, se puede ampliar la funcionalidad con la incorporación del *JMF* (práctica 13) y los códecs que ésta soporta.

<sup>3</sup> Véase práctica 13. Se pueden incorporar mejoras basadas en soluciones diferentes al JMF.

<sup>4</sup> Véase práctica 12. En este caso, se pueden introducir mejoras a lo visto en dicha práctica.

- **Nuevo.** Permite crear una nueva imagen (que aparecerá en una nueva ventana). El usuario deberá indicar el tamaño de la imagen (a través de un diálogo que se lance en el momento de la creación o asociado a un menú general de opciones).
- **Abrir:** Abre el dialogo "Abrir fichero" y permite seleccionar un fichero de imagen, sonido o vídeo. Dependiendo del tipo de fichero, éste se mostrará en un tipo de ventana u otra. Los formatos reconocidos serán los estándares manejados por Java.

Se usará el mismo botón para abrir cualquier medio, teniendo el diálogo abrir asociado filtros para los tipos de ficheros reconocidos. Si se produce algún error al abrir el medio (formato desconocido, etc.) se lanzará un diálogo que informe del problema.

Asociado al diálogo abrir definiremos filtros para que sólo muestre extensiones correspondientes a ficheros de formatos admitidos.

- **Guardar:** Lanza el dialogo "Guardar fichero" y permite guardar la imagen de la ventana que esté seleccionada (esta opción estará desactivada para el caso de sonidos y vídeos). Se podrá almacenar en cualquiera de los formatos reconocidos por Java (JPG, PNG, etc.), obteniendo dicho formato a partir de la extensión indicada por el usuario. Asociado al diálogo guardar definiremos filtros para que sólo muestre extensiones correspondientes a formatos admitidos.

## ■ Barra(s) de dibujo

Esta(s) barra(s) dará(n) acceso a todos los elementos necesarios para poder dibujar sobre una imagen, incluyendo formas y atributos. Todos los elementos de esta barra tendrán que tener asociado un "ToolTipText".

### • Formas de dibujo

En la barra aparecerá un botón (con icono) por cada forma de dibujo disponible. Se usarán botones de dos posiciones agrupados, de forma que siempre aparezca pulsada la forma seleccionada. Al menos, deberán incluirse las siguientes formas:

- Punto
- Línea recta
- Rectángulo
- Rectángulo redondeado [opcional]
- Elipse
- Arco [opcional]
- Curva con un punto de control
- Curva con dos puntos de control [opcional]
- Trazo libre [opcional]
- Polígono
- Formas personalizadas (áreas) [opcional]
- Texto formateado<sup>5</sup> [opcional]

Nótese que, de las formas anteriores, las seis primeras y la novena corresponden a un dibujo de un solo paso (entendido un paso como una secuencia *pressed-dragged-released*<sup>6</sup>), la séptima de dos pasos, la octava de tres pasos, y la décima de tantos pasos como lados tenga el polígono (en este caso, se usará el doble clic para indicar el último vértice [opcional: la posibilidad de que sea cerrado o no]).

Adicionalmente, se incluirá un botón "selección" que, si está pulsado, indicará que el usuario va a seleccionar una forma para su posible edición.

<sup>5</sup> La escritura del texto se podrá hacer directamente sobre el área de dibujo o bien utilizando un diálogo previo en el que introducir la cadena. Independientemente de la forma de introducir el texto, éste deberá de aparecer en el punto de la imagen donde se haga el clic y con el formato indicado.

<sup>6</sup> Dependiendo de la figura, un paso podría estar asociado a solo un clic (sin *dragged*) simple o doble.

## • Atributos de dibujo

El usuario podrá elegir los atributos con los que se pintarán las formas (el diseño y organización del interfaz queda abierto al criterio del estudiante, si bien la complejidad del mismo se tendrá en cuenta en la evaluación). Al menos, deberán incluirse los siguientes atributos:

- **Color.** El usuario podrá elegir el color del trazo y el de relleno. Deberán aparecer una serie de colores predeterminados y, además, la posibilidad de lanzar un diálogo de selección de colores. En este apartado deberá de haber una referencia al color de trazo y de relleno actuales.
- **Trazo.** Se podrán modificar el grosor y el tipo de discontinuidad del trazo. En este último caso, se podrán dibujar, al menos, líneas continuas o líneas punteadas.

*[Opcional: estilos final y de unión de línea, más tipos de discontinuidad]*

- **Relleno.** El usuario podrá elegir entre tres opciones a la hora de rellenar: no rellenar, rellenar con un color liso o rellenar con un degradado. En el caso del degradado, éste se aplicará utilizando los dos colores (frente y fondo) seleccionados en ese momento (para la dirección del degradado, se ofrecerá al menos dos posibilidades: horizontal y vertical).

*[Opcional: más direcciones de degradado, relleno mediante imágenes predeterminadas, relleno radial]*

- **Alisado de bordes.** El usuario podrá activar/desactivar la mejora en el proceso de renderizado correspondiente al alisado de bordes.

*[Opcional: incluir otras mejoras del renderizado]<sup>7</sup>*

- **Transparencia.** Se podrá establecer un grado de transparencia asociado a la forma (por ejemplo, mediante un deslizador)<sup>8</sup>

y opcionalmente:

- **Reglas en la composición.** Se podrán definir reglas para combinar las nuevas formas con las ya existentes.
- **Transformaciones sobre la forma.** Se podrán aplicar traslaciones, escalados, rotaciones y deformaciones sobre la forma a dibujar.
- **Fuente del texto** (en caso de que se incluya el texto como forma). Se podrá establecer la fuente, tamaño y estilo del texto.

Cuando se cambie de una ventana interna a otra, los botones de forma y atributos de la barra de herramientas deberán activarse conforme a la forma y atributos de la ventana activa.

## • A la hora de dibujar...

El usuario podrá dibujar sobre cualquier imagen utilizando la forma y atributos seleccionados. Para ello, hay que tener en cuenta los siguientes requisitos:

- El lienzo mantendrá todas las figuras que se vayan dibujando
- Cada figura tendrá sus **propios atributos**<sup>9</sup> independientes del resto de formas (es decir, no compartirán los mismos valores). Cuando se dibuje la forma por

<sup>7</sup> Cada mejora podrá ser activada/desactivada por el usuario de forma individual.

<sup>8</sup> Nótese que no es sólo la semitransparencia (como en la práctica 7), sino que el usuario podrá definir el grado de transparencia que vaya desde opaco hasta totalmente transparente.

primera vez, ésta usará los atributos que estén activos en ese momento (que no tienen por qué coincidir con los de las figuras que ya estén en el lienzo).

## ● Edición

El usuario podrá editar las figuras ya dibujadas. Para ello:

- Podrá seleccionar la figura haciendo clic sobre ella, para lo cual tendrá que tener activo el botón “seleccionar” de la barra de herramientas. La forma seleccionada deberá identificarse mediante un rectángulo (*boundingbox*) tipo selección.

*[Opcional: selección múltiple, es decir, de varias formas simultáneamente]*

- Para la figura seleccionada, se podrán **editar sus atributos**, esto es, se podrá modificar cualquiera de sus propiedades (color, trazo, relleno, etc.) sin más que cambiarla en la barra de herramientas
- El usuario podrá **mover las figuras** que ya estén dibujadas.

*[Opcional: redimensionar la figura seleccionada]*

## ■ Barra de operaciones sobre imágenes<sup>10</sup>

Esta barra incluirá el conjunto de operaciones que se podrán aplicar sobre cualquier imagen (de una “ventana imagen”)<sup>11</sup>. Todos los elementos de esta barra tendrán que tener asociado un “*ToolTipText*”. Al menos, se deberán de poder realizar las siguientes operaciones:

- Duplicar, que creará una nueva “ventana imagen” con una copia de la imagen
- Modificar el brillo mediante un deslizador
- Filtros de emborronamiento, enfoque y relieve
- Contraste normal, iluminado y oscurecido
- Negativo (invertir colores)
- Transformación a niveles de gris
- Giro libre mediante deslizador
- Escalado (aumentar y disminuir)
- Sepia<sup>12</sup>
- Suma y resta de imágenes
- Mezcla de imágenes mediante deslizador
- Umbralización<sup>12</sup> con deslizador para modificar umbral
- Una nueva operación de diseño propio

y opcionalmente:

- Tintado (con el color de frente seleccionado en ese momento)
- Detección de fronteras *Sobel*
- Ecuilización
- Mostrar el histograma
- Cualquier otra operación de diseño propio

Las operaciones se irán aplicando de forma concatenada, es decir, una operación se aplicará sobre el resultado de operaciones aplicadas anteriormente. En el caso del **brillo**, el

---

<sup>9</sup> Nótese que, a diferencia de la práctica 7, en la que todas las formas se mostraban con los mismos atributos, en este caso cada forma tendrá asociado un conjunto de atributos propio. Esto implica, necesariamente, el diseño de nuevas clases que respondan a este requisito.

<sup>10</sup> Véanse prácticas 9-11.

<sup>11</sup> No es necesario aplicar el efecto sobre las formas dibujadas

<sup>12</sup> Necesario crear clase propia

deslizador permitirá ir variando el brillo sobre la imagen que haya en ese momento, no sobre el resultado del cambio de brillo (es decir, si deslizamos el brillo a su máximo valor –lo que implicaría ver la imagen en blanco–, si después reducimos dicho valor se tiene que volver a ver la imagen inicial). Una vez que se elija otra operación, el brillo se aplicará de forma definitiva (y se concatenará con el resto de operaciones).

De igual manera, la **mezcla de imágenes** usará un deslizador para seleccionar qué proporción de cada imagen es mezclada. Se deja a elección del estudiante la forma en la que se seleccionan las imágenes a mezclar, así como la forma de determinar el fin de la operación (p.e., al elegir otra operación).

*[Opcional: introducir operaciones de tres o más imágenes]*

Para la **umbralización**, aplicaremos la basada en niveles de gris. En este caso, el deslizador permitirá ir variando el umbral e ir viendo el resultado parcial (aplicado sobre la imagen inicial). Una vez que se elija otra operación, la umbralización se aplicará de forma definitiva.

*[Opcional: umbralizar sobre el espacio de color]<sup>13</sup>*

Por último, se deberá de crear una **operación de diseño propio**<sup>14</sup>. El estudiante deberá indicar en la documentación qué operación ha implementado y la correspondiente formulación (si la hubiera). Deberá especificarse qué parámetros tiene la operación e incluir en el interfaz los elementos necesarios para su selección y aplicación.

## ■ Barra de audio/video

Esta barra contará con botones asociados a operaciones con audio y vídeo (todos ellos tendrán que tener asociado un icono y un “*ToolTipText*”). Al menos, contará con el siguiente botón:

- **Capturar:** Permitirá al usuario capturar imágenes de la cámara o del vídeo que se esté reproduciendo; concretamente, lo hará de la ventana que esté activa, siempre y cuando sea una ventana de tipo “reproducción de vídeo” o “webcam”.

y opcionalmente

- **Webcam.** Lanza la ventana tipo webcam que mostrará la secuencia que esté captando la webcam
- Botones **play** y **stop** que centralicen reproducción de audio y/o vídeo
- Botón de **grabación/stop** de audio (en este caso, no sería necesaria la ventana de grabación).

## ■ Menú principal

En el menú (al igual que en las barras de herramientas) el estudiante pondrá aquellas opciones que considere más adecuadas para el diseño que haya realizado. En cualquier caso, al menos deberán aparecer las opciones **Archivo**, **Ver** (que permitirá ocultar/visualizar las diferentes barra de herramientas) y **Ayuda** (que tendrá la opción “Acerca de” que lanzará un diálogo con el nombre del programa, versión y autor). Dentro de la opción Archivo habrá tres opciones: **Nuevo**, **Abrir** y **Guardar** (cuyas acciones serán las mismas definidas para los botones de la barra de herramientas).

<sup>13</sup> En este caso, el usuario podrá elegir entre dos opciones: umbralizar sobre los niveles de gris o umbralizar sobre el espacio de color. En el segundo caso, además del umbral, el usuario deberá elegir el color central de la umbralización que, por defecto, será el color de frente actualmente seleccionado (el deslizador será válido en ambos tipos de umbralizaciones).

<sup>14</sup> Implica crear una clase propia heredando de *BufferedImageOp* en la línea vista en la práctica 11. En este caso, se tratará de una operación propia y distinta de las implementadas en prácticas.

## ■ Documentación

La documentación tendrá que seguir las pautas propias de la Ingeniería del Software (requisitos, análisis, diseño y codificación), haciendo especial hincapié en la descripción y justificación de las clases de diseño propio. Para la codificación, es obligatorio documentar el código usando *javadoc* y generar la correspondiente API (especialmente importante en el caso de bibliotecas propias). Indicar las fuentes (bibliográficas o de código) utilizadas en la elaboración de las prácticas.

Una documentación **suficiente y adecuada** será **imprescindible** para la corrección de las prácticas, siendo el primer criterio de corte para su evaluación<sup>15</sup>.

### ● Javadoc

Todas las clases de diseño propio deberán estar documentadas usando Javadoc<sup>16</sup>. Debe incluirse tanto la descripción de la clase, como la de sus variables y métodos miembro (en éste último caso, incluyendo tanto parámetros como información devuelta). A modo de ejemplo, pueden verse las descripciones de las clases relativas a gráficos, imágenes, etc. usadas a lo largo de la asignatura<sup>17</sup>.

## ■ Entrega

La entrega se realizará a través de la página web de decsai.ugr.es. Se deberá entregar un fichero comprimido (.zip o .rar) que incluya:

- Fichero ejecutable .jar de la aplicación. Este fichero deberá estar en la raíz del fichero comprimido y deberá empaquetar todas las librerías<sup>18</sup>.
- Código fuente (proyectos Netbean completos) incluyendo bibliotecas propias.
- Documentación en PDF (localizable en la raíz del fichero comprimido)
- API generada usando Javadoc (localizable en la raíz del fichero comprimido)

La fecha de entrega será la fijada en el calendario de exámenes, esto es, el día **22 de junio de 2016**. Una vez pasada esta fecha, se abrirá una entrega adicional, activa hasta el **12 de julio**, mediante la cual el estudiante podrá entregar mejoras a la práctica presentada en primera instancia. **Estas mejoras sólo se evaluarán si la entrega previa está aprobada**, es decir, esta segunda entrega está pensada para poder “rematar” detalles que quedasen pendientes, arreglar errores menores o incorporar nuevas funcionalidades de las marcadas como opcionales<sup>19</sup>; en ningún caso esta segunda entrega podrá usarse para abordar aspectos sustanciales y de alto calado<sup>20</sup>. En un documento aparte y específico, se deberá de indicar la lista de mejoras incorporadas en esta segunda entrega.

En caso de que sea necesaria defensa de la práctica, se le notificará al estudiante por correo electrónico el día y hora de dicha defensa. La defensa podrá incluir examen escrito relativo al contenido de la práctica y los conocimientos adquiridos a lo largo del curso.

---

<sup>15</sup> La evaluación comienza con la lectura de la documentación; si ésta no explica en detalle lo realizado, justificando el porqué de los diseños y soluciones adoptadas, no procederá la ejecución de la aplicación (y, consecuentemente, implicará el suspenso en la evaluación).

<sup>16</sup> Para quien no conozca su uso, pueden consultarse los siguientes enlaces de la [Wikipedia](#) o de [Oracle](#).

<sup>17</sup> Recuérdese que, además de la [API](#) oficial, desde NetBeans se puede acceder al código fuente de las clases y ver el uso de Javadoc en la documentación de las mismas.

<sup>18</sup> Véase anexo de la práctica 7.

<sup>19</sup> La documentación puede entregarse en la segunda fase. También podría incluirse en esta segunda fase alguna forma/atributo pendiente de los exigidos en el guion, o alguna operación de imagen.

<sup>20</sup> Por ejemplo, la parte de dibujo, incluida la edición, se considera parte sustancial de la evaluación (por lo que tendría que abordarse en la primera entrega y no podría considerarse “mejora”). Por este motivo, no sería válida como entrega inicial aquella correspondiente a la unión de las prácticas realizadas durante el año, ya que, aunque recogen gran parte de los requisitos, carecen de elementos sustanciales de la evaluación final (como puede ser el diseño de clases propias).