

Aprendizaje Automático (2018-2019)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Evaluación Sistemas Multimedia

Montserrat Rodríguez Zamorano

3 de julio de 2019

Índice

1. Descripción del sistema	1
2. Requisitos	1
2.1. Requisitos funcionales	1
2.1.1. Carácter general	1
2.1.2. Dibujo	1
2.1.3. Procesamiento de imágenes	2
2.1.4. Sonido	3
2.1.5. Vídeo	3
2.2. Requisitos no funcionales	3
3. Análisis	4
4. Diseño	4
4.1. Figura	4
4.2. Ventana multimedia	4
4.3. Eventos	4
4.4. Operaciones de diseño propio	4
4.4.1. Componente a componente	4
4.4.2. Pixel a pixel	5
5. Implementación	5
6. Validación	5
6.1. Gráficos	5
6.2. Procesamiento de imágenes	5
6.2.1. Duplicar	5
6.2.2. Modificar brillo	6
6.2.3. Filtro de emborronamiento	6
6.2.4. Filtro de enfoque	7
6.2.5. Filtro de relieve	7
6.2.6. Contraste normal	7
6.2.7. Iluminado	8
6.2.8. Oscurecido	8
6.2.9. Invertir colores	8
6.2.10. Conversión a espacios RGB, YCC, GRAY	8
6.2.11. Giro libre	9
6.2.12. Escalado	9
6.2.13. Tintado	9
6.2.14. Ecualización	10
6.2.15. Filtro sepia	10
6.2.16. Umbralización	10
6.2.17. Operador <i>LookupOp</i> basado en una función propia	11
6.2.18. Operación de diseño propio: filtro violeta	11
6.2.19. Operación de diseño propio: media de las bandas	11
6.3. Sonido	11
6.4. Vídeo	11
6.4.1. WebCam	11

1. Descripción del sistema

Para la evaluación de la asignatura de *Sistemas multimedia* se quiere realizar una aplicación multimedia que permita gestionar gráficos, imágenes, sonido y vídeo. Se podrán crear, editar, procesar y visualizar contenido multimedia de distintos tipos.

Esta aplicación tendrá una ventana principal con diferentes barras de herramientas y botones que permitirán realizar las distintas operaciones sobre el contenido multimedia. Se intentará que la interfaz sea lo más intuitiva posible y facilitar así el uso de la aplicación al usuario final.

Con este fin, se realiza una aplicación que se llamará *Paint*.

2. Requisitos

2.1. Requisitos funcionales

2.1.1. Carácter general

- [RFCG-1] Creación de una nueva imagen en una nueva ventana.
- [RFCG-2] Abrir un fichero de imagen.
- [RFCG-3] Abrir un fichero de sonido.
- [RFCG-4] Abrir un fichero de vídeo.
- [RFCG-5] Guardar una imagen y sus figuras dibujadas.
- [RFCG-6] Ocultar las barras de herramientas.
- [RFCG-7] Visualizar las barras de herramientas.
- [RFCG-8] Consultar el nombre del programa, versión y autor.

2.1.2. Dibujo

- [RFD-1] Dibujar las siguientes formas geométricas con sus propios atributos independientes.
 - Línea.
 - Rectángulo.
 - Elipse.
 - Rectángulo con esquinas redondeadas.
- [RFD-2] Mantener todas las figuras que se vayan dibujando.
- [RFD-3] Elegir el color de trazo de dibujo.
 - Rojo.
 - Azul.
 - Negro.
 - Blanco.
 - Verde.
- [RFD-4] No rellenar la imagen.
- [RFD-5] Rellenar con color el dibujo.

- Rojo.
- Azul.
- Negro.
- Blanco.
- Verde.
- [RFD-6] Seleccionar una figura dibujada.
- [RFD-7] Editar una figura dibujada.
- [RFD-8] Mover una figura dibujada.
- [RFD-9] Consultar los atributos de una figura dibujada.
- [RFD-10] Asociar un grado de transparencia a una figura.
- [RFD-11] Activar alisado de bordes de una figura.
- [RFD-12] Desactivar alisado de bordes de una figura.

2.1.3. Procesamiento de imágenes

- [RFPI-1] Duplicar una imagen.
- [RFPI-2] Modificar el brillo de una imagen.
- [RFPI-3] Aplicar filtro para emborronar una imagen.
- [RFPI-4] Aplicar filtro para enfocar una imagen.
- [RFPI-5] Aplicar filtro de relieve a una imagen.
- [RFPI-6] Aplicar contraste a una imagen.
- [RFPI-7] Iluminar una imagen.
- [RFPI-8] Oscurecer una imagen.
- [RFPI-9] Extraer las bandas de una imagen.
- [RFPI-10] Invertir los colores de una imagen.
- [RFPI-11] Convertir una imagen a los siguientes espacios:
 - RGB
 - YCC
 - GRAY
- [RFPI-12] Girar una imagen a cualquier ángulo.
- [RFPI-13] Tintar una imagen.
- [RFPI-14] Escoger nivel de tintado de una imagen.
- [RFPI-15] Ecualizar una imagen.
- [RFPI-16] Umbralizar una imagen en niveles de gris.
- [RFPI-17] Escoger nivel de umbralización.

- [RFPI-18] Aplicar filtro sepia a una imagen.
- [RFPI-19] Aplicar filtro violeta a una imagen.
- [RFPI-20] Aplicar filtro de media entre las bandas de colores.
- [RFPI-21] Aplicar filtro cosinosoidal.

2.1.4. Sonido

- [RFS-1] Reproducir audios.
- [RFS-2] Grabar sonidos.
- [RFS-3] Pausar la reproducción.
- [RFS-4] Parar la grabación.
- [RFS-5] Parar la reproducción.

2.1.5. Vídeo

- [RFV-1] Mostrar la secuencia que capte la Webcam.
- [RFV-2] Capturar una imagen desde la Webcam.

2.2. Requisitos no funcionales

- [RNF-1] Se mostrará en la barra de estado el pixel en el que está situado el ratón.
- [RNF-2] Se habilitarán en cada momento sólo los botones que pueden utilizarse. Por ejemplo, si se abre una ventana de vídeo se deshabilitarán aquellos correspondientes a las imágenes.
- [RNF-3] Los botones tendrán asociados un *ToolTipText* para facilitar su uso.
- [RNF-4] Al seleccionar una figura se activarán sus propiedades en la barra de atributos.
- [RNF-5] Si hay una figura seleccionada, al pulsar el ratón sobre en otro punto, deberá deseleccionarse la figura.
- [RNF-6] El título de una nueva ventana abierta será el nombre del fichero si se trata de una imagen abierta o guardada.
- [RNF-7] El título de una nueva ventana abierta será *Nueva* si se trata de una imagen creada por el usuario.
- [RNF-8] El título de una nueva ventana abierta será *Captura* si se trata de una captura captada de un vídeo o de la WebCam.
- [RNF-9] La *BoundingBox* será un rectángulo de color azul, con línea discontinua.
- [RNF-10] Cuando se cree una imagen se lanzará un diálogo que permita elegir las dimensiones de la imagen.

3. Análisis

Se analizará qué requisitos son cubiertos por las bibliotecas disponibles y para cuáles necesitamos desarrollar soluciones alternativas.

En primer lugar, se desarrollará una clase *Lienzo* que constituirá el área de dibujo y que almacenará las figuras dibujadas en una misma ventana. La clase *Lienzo* será la encargada de pintar las figuras.

La solución planteada durante el desarrollo de las prácticas no es muy flexible: todas las figuras de un mismo lienzo tienen los mismos atributos. Para cumplir [RFD-1]-[RFD-5] y [RFD-9]-[RFD-12] se desarrollará una clase propia *Figura*, que almacenará cada uno de los atributos de la figura y permitirá que cada figura tenga los suyos propios. El uso de esta clase permitirá además modificar estos atributos [RFD-7]. Sin embargo, la línea, por ejemplo, no puede tener relleno, por lo que no tiene sentido guardar este atributo.

Para solucionarlo, se desarrollará una clase propia para cada una de las figuras que pueden dibujarse: Línea, rectángulo, elipse, rectángulo redondeado. El uso de estas clases permitirá mover las figuras dibujadas [RFD-8], implementando en cada caso la edición de la posición.

Esta solución completa los requisitos del módulo de figuras. Para el caso del procesamiento de imágenes, se tendrán que desarrollar clases propias para cumplir con [RFPI-16]-[RFPI-20]. El resto de los requisitos funcionales están cubiertos por las funciones ofrecidas por Java y las bibliotecas proporcionadas por el profesor. Para el caso de cosinusoidal [RFPI-21] se hará uso de la clase *LookUpTable* con una operación propia (coseno).

En cuanto al manejo de eventos, se usará en muchas ocasiones las funciones que ofrece Java para la gestión de eventos (por ejemplo, para el uso de botones). Sin embargo, para la comunicación entre el lienzo y la ventana principal será necesario crear una clase manejadora para la gestión de eventos relacionados con el lienzo. La existencia de esta clase permitirá también informar de la falta de atributos necesarios para el dibujo de las figuras y poder lanzar mensajes de error desde la ventana principal (por ejemplo: no se ha seleccionado una forma de dibujo).

4. Diseño

Siguiendo la propuesta que se ha planteado en el análisis, se plantea las siguientes jerarquías de clases.

4.1. Figura

4.2. Ventana multimedia

4.3. Eventos

4.4. Operaciones de diseño propio

4.4.1. Componente a componente

La función componente a componente modificará cada una de las componentes (R, G, B) de forma separada.

En este caso se ha implementado un filtro violeta. Para aplicar un tono morado a la imagen, teniendo en cuenta que la mezcla de rojo y azul es la que da lugar al morado, se reducirá el valor de la componente verde y se aumentará el valor de la componente roja y azul.

```
for(int x=0; x<src.getWidth(); x++){  
    for(int y=0; y<src.getHeight(); y++){
```

```

    int destR, destG, destB;
    int srcR, srcG, srcB;
    Color colorSrc = new Color(src.getRGB(x, y));
    Color colorDest;
    //color del pixel
    srcR = colorSrc.getRed();
    srcG = colorSrc.getGreen();
    srcB = colorSrc.getBlue();
    //operacion componente a componente
    destR = (int) Math.min(255, 1.3*srcR);
    destG = (int) Math.min(255, 0.2*srcG);
    destB = (int) Math.min(255, 1.5*srcB);

    colorDest = new Color(destR, destG, destB);
    dest.setRGB(x, y, colorDest.getRGB());
}
}

```

Listing 1: Operación componente a componente

El resultado de aplicar el filtro puede verse a continuación.



Figura 4.1: Aplicación filtro violeta

4.4.2. Pixel a pixel

5. Implementación

6. Validación

En este apartado se comprobará que el programa implementado en base a la solución propuesta cumple con los requisitos enumerados en el primer apartado.

6.1. Gráficos

6.2. Procesamiento de imágenes

6.2.1. Duplicar

Se comprobará si la duplicación de la imagen se ha realizado correctamente, de forma que si se modifica una, los cambios no se aplican en la otra.



Figura 6.1: Duplicar imagen

6.2.2. Modificar brillo

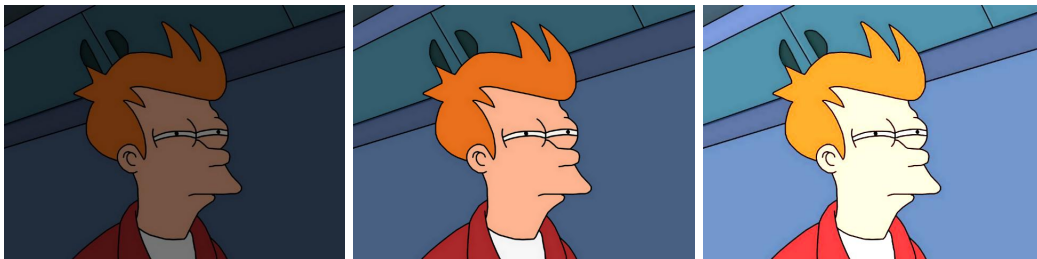


Figura 6.2: Modificación del brillo

6.2.3. Filtro de emborronamiento

Se ha aplicado varias veces el filtro para que sea visible.



Figura 6.3: Aplicación filtro emborronamiento

6.2.4. Filtro de enfoque



Figura 6.4: Aplicación filtro enfoque

6.2.5. Filtro de relieve



Figura 6.5: Aplicación filtro relieve

6.2.6. Contraste normal

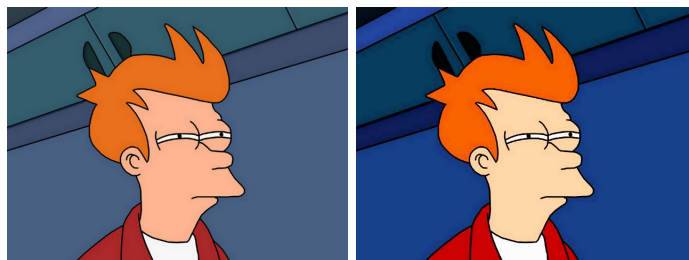


Figura 6.6: Aplicación contraste

6.2.7. Iluminado

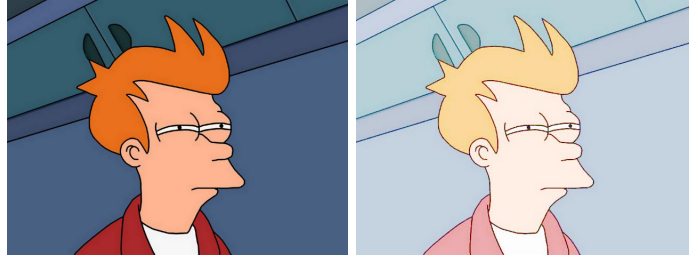


Figura 6.7: Aplicación iluminación

6.2.8. Oscurecido



Figura 6.8: Aplicación oscurecido

6.2.9. Invertir colores

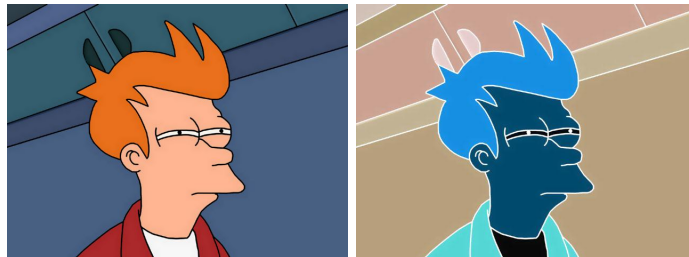


Figura 6.9: Aplicación filtro negativo

6.2.10. Conversión a espacios RGB, YCC, GRAY

Para comprobar que la conversión se ha realizado correctamente, se extraen las bandas de colores y se comparan. En la imagen se pueden ver la tercera banda de una imagen en espacio de color RGB y YCC, respectivamente.



Figura 6.10:

6.2.11. Giro libre

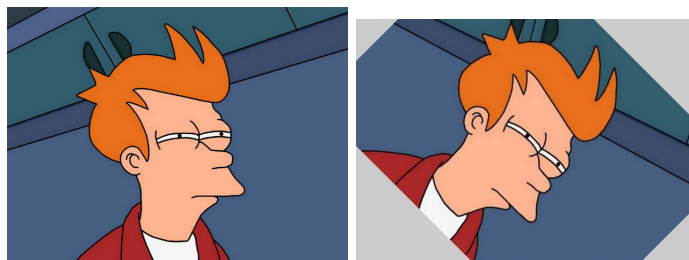


Figura 6.11: Giro libre

6.2.12. Escalado

6.2.13. Tintado

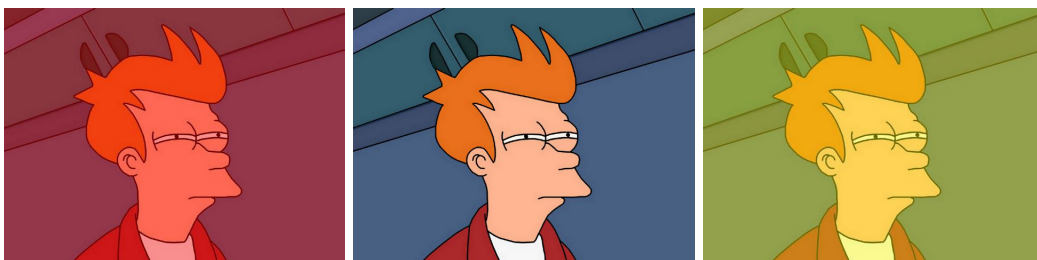


Figura 6.12: Tintado de la imagen

6.2.14. Ecualización

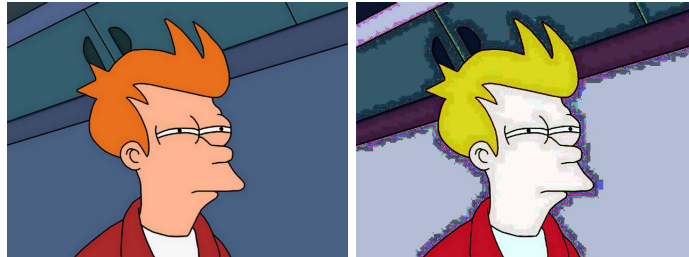


Figura 6.13: Aplicación ecualización

6.2.15. Filtro sepia



Figura 6.14: Aplicación filtro sepia

6.2.16. Umbralización

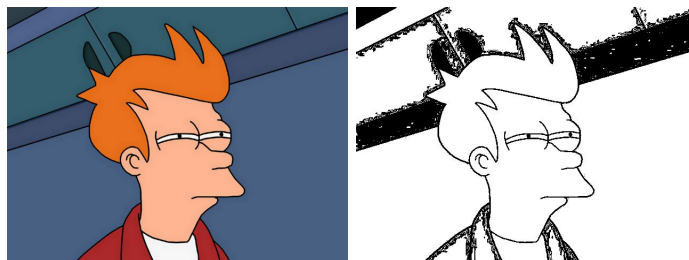


Figura 6.15: Umbralizar imagen

6.2.17. Operador *LookupOp* basado en una función propia



Figura 6.16: Operación cosinusoide

6.2.18. Operación de diseño propio: filtro violeta



Figura 6.17: Aplicación filtro violeta

6.2.19. Operación de diseño propio: media de las bandas



Figura 6.18: Aplicación filtro media de colores

6.3. Sonido

6.4. Vídeo

A continuación se muestran las funcionalidades del vídeo y la cámara.

6.4.1. WebCam

En primer lugar, comprobamos que puede abrirse correctamente la cámara. Una vez hecho esto, se comprobará si se realizan las capturas de pantalla correctamente. Si no hay ventana de cámara abierta, no se podrá sacar ninguna captura de pantalla.

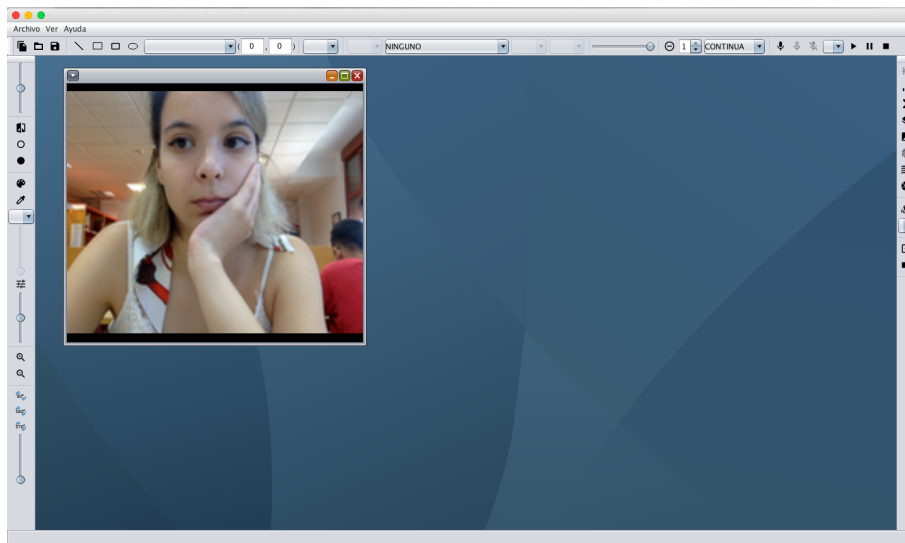


Figura 6.19: Webcam abierta

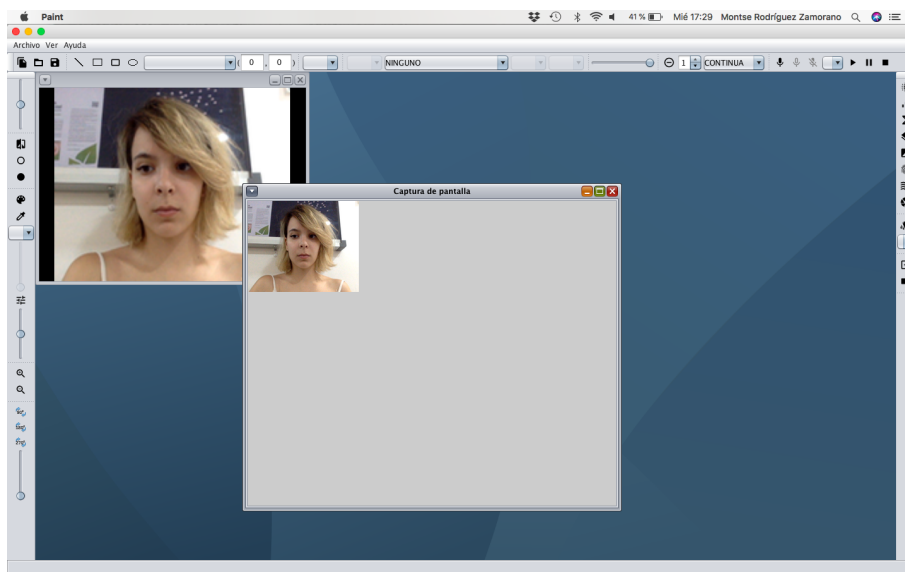


Figura 6.20: Captura de pantalla