



Universidad de las Ciencias Informáticas

AIRI, Artificial Intelligence: Research and Innovation, Facultad 3

Módulo para la identificación del espacio de color más adecuado para la manipulación de imágenes de opacidad capsular provenientes de la lámpara de hendidura.

**Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas**

Autor:

Selma Hernández Romero

Tutores:

Ing. Michel Álvarez Cancio

Ing. Michel Sariol Fernández

La Habana. Junio, 2017.

Declaratoria de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Selma Hernández Romero

Autor

Ing. Michel Álvarez Cancio

Tutor

Ing. Michel Sariol Fernández

Tutor

Agradecimientos

A mis tutores de tesis (Los Micheles) que no se parecen en el físico pero si en la grandeza de corazón, sin ustedes no hubiese sido posible esta investigación para la culminación de mis estudios como Ingeniera en Ciencias Informáticas, gracias por permitirme ser parte del grupo de investigación AIRI y compartir lindos momentos con todos sus integrantes, por el incondicional apoyo profesional y personal de ambos, aportando en todo momento sus experiencias como profesionales e inestimable calidad humana.

A ti mami, por amarme tanto desde que me concebiste y ser la luz eterna que me guía.

A mi papi, que extraño todos los días de mi vida, y si estuviese físicamente conmigo hoy, brotarían lágrimas de alegría de sus ojazos por ver a su Makuki graduada.

A mi hermano, por ser mi complemento y mi compañero de vida.

A mi tío Rafael, por ser como un padre para mí y darme su amor incondicional.

A mi prima Annia, por estar siempre presente en mi corazón y amarme como a una hermana.

A mi amiga Giselle, por estar en los buenos y malos momentos cuidándonos mutuamente, por brindarme su plena confianza y enseñarme que no importa la distancia que nos separe, porque siempre seremos amigas.

A mi querido Lechu, que a pesar de que me pone como loca con sus temas, lo quiero montonazo y no lo quiero perder nunca, gracias por guiarme a tomar decisiones correctas cuando me sentía perdida, por ser mi compañía favorita y por sacarme siempre la sonrisa y el brillo de mis ojos.

A todos mis familiares que siempre han estado pendiente y hoy están felices por este logro.

A los estupendos amigos de mi mami y de mi hermano, que conforman la otra parte de la familia y los amamos por su incondicionalidad.

A todos mis profesores que influyeron en mi preparación integral como estudiante.

A todas las amistades que conocí en el transcurso y que influyeron de algún modo en mi paso por la universidad, en especial a Elena, Michel, Dalili y Keiger.

Gracias a ti mamá. TE AMO.

Dedicatoria

Le dedico el presente trabajo, a la Revolución Cubana y a nuestro líder histórico Fidel Castro, que gracias a su inmensa labor revolucionaria, se concibió la Universidad de las Ciencias Informáticas (UCI), con el objetivo de formar jóvenes comprometidos con la revolución capaces de contribuir al proceso de informatización de la sociedad cubana.

Contenido

Declaratoria de autoría	II
Agradecimientos	III
Dedicatoria	IV
Resumen	VII
Introducción	5
Capítulo 1 Fundamentación Teórica	10
1.1 Imagen Digital	10
1.2 Imágenes a color	11
1.3 Espacio de color	11
1.3.1 Espacio de color RGB	12
1.3.2 Espacio de color CMY	13
1.3.3 Espacio de color CIEL*a*b*	14
1.3.4 Espacio de color HSL	15
1.3.5 Espacio de color HSB	16
1.3.6 Espacio de color YCbCr	17
1.4 Ruido y Filtrado Digital de imágenes	18
1.4.1 Ruido en imágenes digitales	19
1.4.2 Filtros lineales espaciales	19
1.4.3 Filtrado no lineal espacial	20
1.4.4 Filtros no lineales de paso bajo (RO)	22
1.4.5 Filtros de la mediana	22
1.5 Ambiente de desarrollo	22
1.5.1 Herramienta de desarrollo	23
1.5.2 Herramienta para el modelado	23
1.5.3 Programa Analizador de Opacidad Capsular: PANDOC	24
1.5.4 Metodología de desarrollo	25
1.6 Métricas de validación	26
1.7 Conclusiones del capítulo	28
Capítulo 2 Propuesta de solución	29
2.1 Descripción de la solución	29
2.2 Fase de planificación	35
2.2.1 Requisitos del módulo	35
2.2.2 Historias de usuario (HU)	36
2.2.3 Estimación de esfuerzo por historias de usuario	38
2.2.4 Plan de iteraciones	39
2.2.5 Plan de entrega	39

2.3 Diseño de la solución	39
2.3.1 Tarjetas CRC	40
2.3.2 Estándar de codificación	41
2.4 Conclusiones del capítulo.....	42
Capítulo 3 Resultados y validación	43
3.1 Fase de implementación	43
3.2 Interfaz gráfica.....	43
3.3 Resultados	45
3.4 Pruebas.....	53
3.4.1 Pruebas de caja blanca.....	53
3.4.2 Pruebas de aceptación	55
3.4 Conclusiones del capítulo.....	57
Conclusiones generales.....	58
Referencias bibliográficas	59
Glosario de Términos.....	63
Anexos	64

Resumen

La opacidad de la cápsula posterior es actualmente uno de los aspectos más importantes en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación postoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas. El software PANDOC provee al oftalmólogo de una herramienta por medio de la cual este es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, lo cual se realiza de manera manual y depende mucho el resultado de la experticia del especialista. Dentro de las imágenes usadas en los pacientes operados de cataratas se encuentran las imágenes provenientes de la lámpara de hendidura, su tratamiento digital depende del espacio en cual hayan sido generadas. Es por ello que surge como necesidad la identificación del espacio de color idóneo para el tratamiento de las mismas. Por tal motivo la presente investigación se enmarca en la creación de un módulo para elevar la calidad de la imagen para su manipulación mediante la selección de espacio de color y técnicas de filtrado más adecuadas, a fin de propiciar un mejor diagnóstico al especialista.

Palabras claves: catarata, espacio de color, lámpara de hendidura, opacidad de la cápsula posterior, técnicas de filtrado.

Introducción

En el área de la oftalmología, la opacidad de la cápsula posterior (OPC) es un elemento fundamental de análisis en la actualidad, es uno de los aspectos más importante en la cirugía de catarata de los tiempos modernos [2]. Siendo la complicación postoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan a repercusiones sociales, médicas y económicas, siendo una preocupación a nivel mundial y que fundamentalmente interesan a nuestro país, los aspectos y procedimientos de la cirugía de cataratas, siendo las cataratas la causa que más repercute en los problemas de ceguera en todo el mundo [1] [3].

La catarata corresponde a una opacificación del cristalino que interfiere en forma progresiva con la agudeza visual, lo cual trae como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, dando lugar a una serie de imágenes difusas. Tiene diversas causas, pero se le atribuye mayormente a la edad, acelerando este proceso si el paciente padece de enfermedades como la diabetes o hipertensión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad [3] [4]. La catarata continúa siendo un problema de salud pública en la mayoría de los países subdesarrollados o en vías de desarrollo, debido a la carencia de especialistas y a un incremento de la expectativa de vida de la población, lo que lleva a un aumento de la prevalencia de esta enfermedad, ligada al envejecimiento [3] [4].

En el contexto del procesamiento de imágenes, los espacios de color más utilizados son el RGB (rojo, verde y azul por sus siglas en inglés), para monitores y videocámaras; el CMY (cyan, magenta y amarillo por sus siglas en inglés) y el HSI (tono, saturación e intensidad por sus siglas en inglés) que está estrechamente relacionado con la forma en que los humanos describen e interpretan el color [5]. También se han usado otros como: CIE $L^*a^*b^*$ (Estándar CIE de colores, luminancia y cromaticidad) [6][7][8][9][10], HSL (tono, saturación y luminancia por sus siglas en inglés) [11], HSB (tono, saturación y brillo por sus siglas en inglés) [12], YCbCr (Luminancia, componente azul y un valor de referencia, componente roja y un valor de referencia) [13][14][15][16][17][18].

En la Facultad 3 de la Universidad de las Ciencias Informáticas, el grupo de investigación AIRI, entre otras líneas, investiga el tema de procesamiento de señales digitales y en particular el de imágenes digitales. Dicho grupo tiene una estrecha vinculación con el Instituto Cubano de Oftalmología (ICO) "Ramón Pando Ferrer", el cual

tiene la necesidad de cuantificar la incidencia de la OCP en los pacientes operados de catarata.

Para la identificación de la opacidad de la cápsula posterior se utilizan las imágenes provenientes de la lámpara de hendidura, entre otros diagnósticos [4], pero es una imagen con alta presencia de ruido, con mucha incidencia de luz en ciertas regiones y escasa en otras, lo cual dificulta al especialista poder realizar un diagnóstico adecuado. Para el procesamiento de estas es necesario modificar la conformación del espacio de color de las imágenes.

La imagen que se observa desde una lámpara de hendidura corresponde a las estructuras del polo anterior del ojo: párpado, conjuntiva, córnea, iris, cristalino y cámara anterior [3]. De cada una de las estructuras del ojo mencionadas, en la investigación se hace referencia específicamente al cristalino y su cámara posterior [1].

De la cooperación con el ICO surge un proyecto para identificar y cuantificar diferencias de opacidad a veces imperceptibles para el ojo humano.

Para realizar esta acción, uno de los principales problemas que enfrenta el grupo de investigación, es la identificación del espacio de color más adecuado para tratar las imágenes y obtener con la mayor precisión los resultados que se buscan en los análisis [2].

La mayor parte de las imágenes con las que se trabaja provienen de la lámpara de hendidura¹, y estas son generadas en el espacio de color RGB, que a pesar de ser el más difundido, tiene desventajas [19]:

- No se cuenta con un canal directo de intensidad que permite realizar transformaciones sobre el mismo.
- Los valores de 3 estímulos dependen de la luminancia, que es una transformación lineal de las componentes R, G y B del color.
- Para obtener la matriz de intensidad se realiza un promedio de los valores en cada canal, no siendo posible recuperar nuevamente la composición de los colores al modificarse este valor de intensidad.
- Debido a que los espacios de color RGB son dependientes del dispositivo, hay una multitud de espacios de color RGB con diferentes características.

¹ Lámpara de hendidura: Instrumento oftalmológico que permite tomar imágenes de estructura ocular.

En este momento, es crítico para el grupo, que se determine el mejor espacio de color para el tratamiento de un conjunto de 24 imágenes, provenientes de la lámpara de hendidura, 12 tomadas en retroiluminación y 12 tomadas en iluminación oblicua. La criticidad viene dada por 13 investigaciones defendidas como trabajos de diploma en el curso 2015-2016, donde en cada una de estas, ya fuera por los procesos de segmentación, mejoramiento² o restauración³, se necesitó manipular la matriz de intensidades (suma promediada de los 3 valores de ese espacio de color) [5], siendo necesario, en todos los casos, transformar del espacio de color nativo de las imágenes con las que se cuenta (RGB) a otro, haciéndose esta selección diferente en cada caso, y realizada la misma a conveniencia del investigador, de cuál sería el espacio de color más conveniente, sin tener en cuenta en cuál de estos se elimina mejor el ruido, ni en cual se disminuye el tiempo de ejecución de dicha transformación.

Dada la situación problemática planteada anteriormente se identifica el siguiente **problema a resolver**: ¿Cuál es el espacio de color más adecuado para la manipulación de las imágenes provenientes de la lámpara de hendidura?, definiéndose como **objeto de estudio**: el procesamiento digital de imágenes, enmarcando el **campo de acción**: espacios de color en imágenes de la lámpara de hendidura, persiguiendo como **objetivo general**: desarrollar un módulo para identificar el espacio de color más adecuado en cuanto a eliminación de ruido y disminución de tiempo de ejecución mediante el análisis de los resultados de experimentos con 24 imágenes provenientes de la lámpara de hendidura, planteando **como idea a defender** que: Con la identificación del espacio de color más adecuado para la manipulación de las imágenes de la lámpara de hendidura en cuanto a eliminación de ruido y tiempo de ejecución, se obtendrán mejores resultados de calidad en la manipulación de las mismas.

Objetivos específicos:

- Elaborar el marco teórico de la investigación mediante el estudio de las técnicas actuales de procesamiento digital de imágenes.
- Identificar los requisitos funcionales y no funcionales para el desarrollo de la propuesta de solución.

² Mejoramiento: Proceso de mejorar características de interés de una imagen.

³ Restauración: Proceso que permite recuperar una parte deteriorada de la imagen o que tiene algún objeto que la oculta, con el objetivo de mejorar su calidad.

- Diseñar un módulo para la transformación de espacio de color en imágenes provenientes de la lámpara de hendidura.
- Implementar los elementos definidos en la etapa del diseño para cumplir con la propuesta de solución.
- Validar los resultados comprobando la efectividad de los espacios de color en cuanto a la eliminación de ruido y tiempo de ejecución.

Para darle cumplimiento a estos objetivos se han definido las siguientes **tareas de la investigación**:

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de transformación de espacio de color.
3. Revisión de tipos de ruido existentes en imágenes médicas.
4. Revisión de las técnicas de filtrado y mejora en imágenes médicas.
5. Revisión de métricas de calidad del filtrado en imágenes médicas.
6. Descripción de los pasos a seguir para el uso del módulo propuesto.
7. Desarrollo de una propuesta de solución en función de los requisitos especificados.
8. Implementación de las técnicas de transformación de espacio de color.
9. Implementación de las técnicas de adición de ruido.
10. Implementación de las técnicas de filtrado y mejora de la imagen.
11. Validación de la solución propuesta a partir de la comparación de los resultados por métricas (PSNR, SNR, EMC, SSIM) y tiempo de ejecución.

La lógica de la investigación se desarrolla a partir de métodos de investigación científica esta se hace concreta en las diversas etapas o pasos del proceso de la investigación, para dar solución al problema investigado. El método de investigación es como una especie de brújula en la que no se produce auténticamente el saber [54].

Métodos teóricos:

- El método **Histórico-Lógico** se empleó con el fin de conocer la evolución y desarrollo de las técnicas actuales utilizadas para la detección y transformación de espacios de color.
- El método **Analítico-Sintético** se empleó para realizar una búsqueda y análisis profundo acerca de los módulos, teorías y documentos relacionados con el tema y a partir de esto se seleccionaron los elementos más útiles para el desarrollo del presente trabajo.

Métodos empíricos

- El método de **Experimentación** sirvió para probar las distintas métricas y transformaciones hacia diferentes espacios de color en cada una de las imágenes.

La presente investigación está estructurada en tres capítulos; a continuación se describe el objetivo principal de cada uno de ellos:

Capítulo 1: Fundamentación Teórica

En este capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación, también se realiza un estudio del arte de los algoritmos y técnicas para el procesamiento digital de imágenes. Se describen las herramientas y tecnologías a utilizar durante el desarrollo de la solución. Son expuestos los argumentos que conllevan a la selección de dichas herramientas y tecnologías. Se explican las principales características de la metodología de software.

Capítulo 2: Propuesta de solución.

Se describen las principales características del módulo propuesto, guiando todo el proceso por las fases definidas por la metodología Programación Extrema (XP). Se analizan elementos necesarios para garantizar el éxito en el proceso de desarrollo, tales como: los requisitos del sistema, las historias de usuario y las tarjetas CRC (Clase, Responsabilidad y Colaborador).

Capítulo 3: Resultados y validación.

Se comparan los espacios de color en cuanto a ruido eliminado y tiempo de ejecución. Se realizan un conjunto de pruebas al sistema, divididas en pruebas unitarias y aceptación para conocer si el módulo implementado satisface las necesidades del cliente. Además, se describen los resultados de la aplicación de las métricas para validar el diseño y las técnicas para la validación de los requisitos.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se realizará un estudio sobre los fundamentos teóricos de la investigación con el fin de obtener un mejor entendimiento del procesamiento digital de imágenes. Se realizará énfasis dentro de la etapa de preprocesamiento, para poder obtener los conocimientos necesarios, y llevar a cabo la propuesta de un módulo que dé solución al problema planteado. Además de fundamentarse la necesidad de la realización de esta tarea.

1.1 Imagen Digital

En numerosas ocasiones se piensa en "lo que se ve" como en "lo que está ahí", cuando en realidad el ojo humano solo percibe una pequeña zona del espectro electro-magnético (E-M) conocida como "espectro visible". Con el avance de la tecnología existe un amplio rango de la radiación E-M que se puede detectar y registrar como imágenes, para posteriormente analizar, transmitir o almacenar [20].

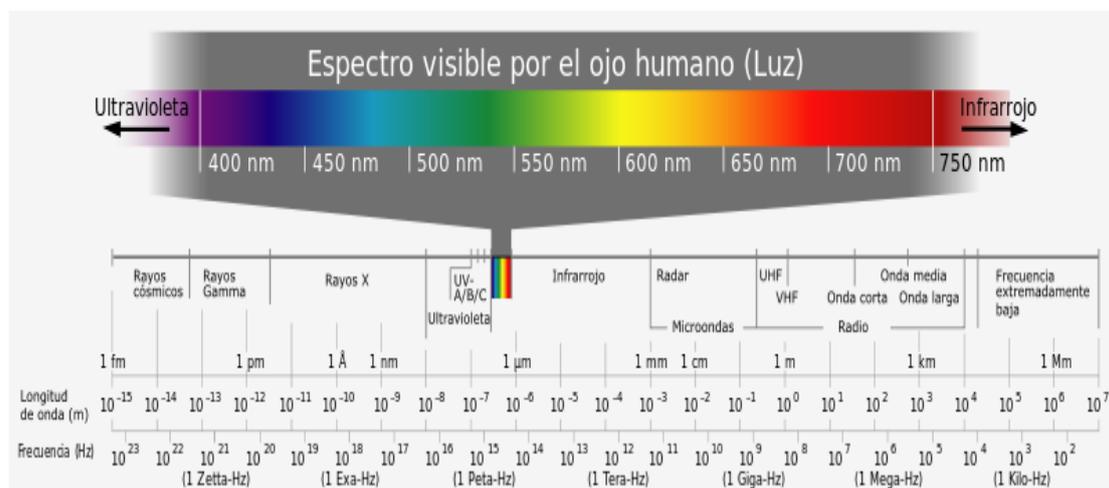


Figura 1. Espectro Electro-Magnético tomada de [20].

Una imagen digital, suponiendo que se trate de una imagen monocromática, puede considerarse como una matriz cuyos índices de filas y columnas identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Esto es lo que se conoce como píxel, abreviatura del término en inglés "picture elements". Si por el contrario se tratara de una imagen en color, la información de ese punto se codificaría según el espacio de color que se fuera a utilizar, creando una imagen multicanal. Estos espacios de color proporcionan un método para representar, ordenar y manipular los colores. El principio en el que se basan la mayoría de estos espacio de color consiste en que a partir de la combinación de tres colores primarios se puede obtener cualquier color de la naturaleza [20] [21].

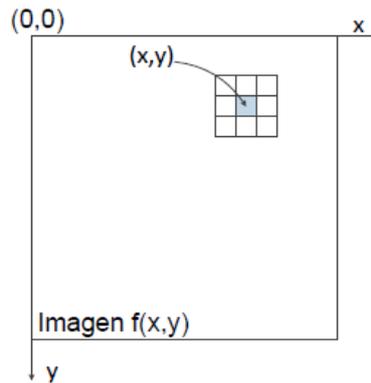


Figura 2. Imagen Digital, tomada de [21].

1.2 Imágenes a color

Las imágenes digitales a color están gobernadas por los mismos conceptos de muestreo, cuantificación y resolución que las imágenes en escala de grises. Sin embargo, en lugar de un único valor de intensidad que expresa el nivel de gris, los píxeles de las imágenes a color están cuantificados usando tres componentes, que tendrán distinto significado según el modelo o espacio de color utilizado.

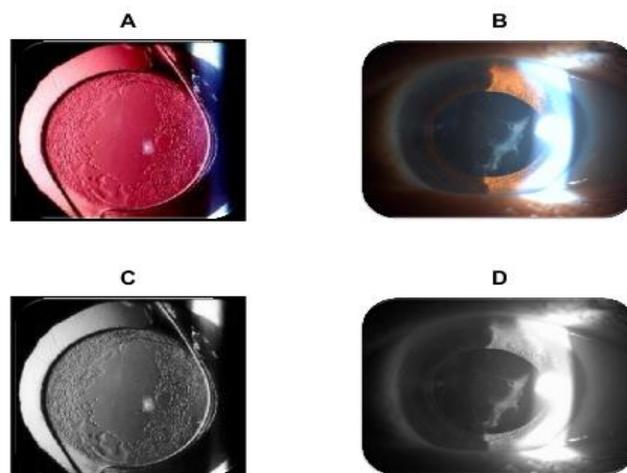


Figura 3: Imágenes de opacidad a color (A y B) y en escala de grises (C y D).

1.3 Espacio de color

El propósito de un espacio de color (también denominado modelo de color o sistema de color) es facilitar la especificación de colores en algún estándar. Es una especificación de un sistema de coordenadas y un subespacio dentro del sistema donde cada color es representado por un punto único [5] [11].

La mayoría de los espacios de color utilizados en la práctica son orientados a hardware o hacia aplicaciones que manipulan el color, como las que crean gráficos de color para

la animación. En el contexto del procesamiento de imágenes, los espacios de color más utilizados son el RGB (rojo, verde y azul por sus siglas en inglés), para monitores y videocámaras; el CMY (cyan, magenta y amarillo por sus siglas en inglés) y el HSI (tono, saturación e intensidad por sus siglas en inglés) que está estrechamente relacionado con la forma en que los humanos describen e interpretan el color [5].

También se han usado otros como: CIE $L^*a^*b^*$ (Estándar CIE de colores, luminancia y cromaticidad) [6][7][8][9][10], HSL (tono, saturación y luminancia por sus siglas en inglés) [11], HSB (tono, saturación y brillo por sus siglas en inglés) [12], YCbCr (Luminancia, componente azul y un valor de referencia, componente roja y un valor de referencia) [13][14][15][16][17][18].

1.3.1 Espacio de color RGB

En el espacio de color **RGB**, cada color aparece descompuesto en sus tres componentes espectrales primarias de **rojo, verde y azul**. Combinando distintas intensidades de estos tres colores primarios, se puede obtener todos los colores visibles.

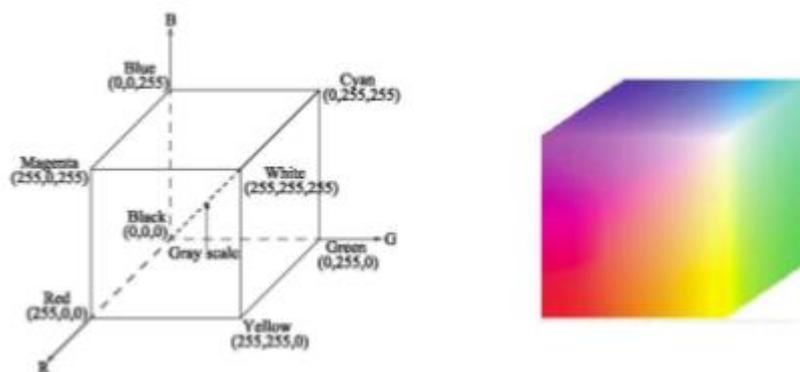


Figura 4. Espacio de color RGB [49].

En el espacio de color RGB, cada color aparece en sus componentes primarios de rojo, verde y azul. Este espacio de color se basa en un sistema de coordenadas cartesianas. El número de bits que se utilizan para representar cada pixel en el espacio RGB constituye la profundidad del pixel. El término “imagen a todo color” es usado frecuentemente para denotar una imagen RGB de 24 bits de profundidad, lo que significa que el número total de colores que permite es $(2^8)^3$ [5].

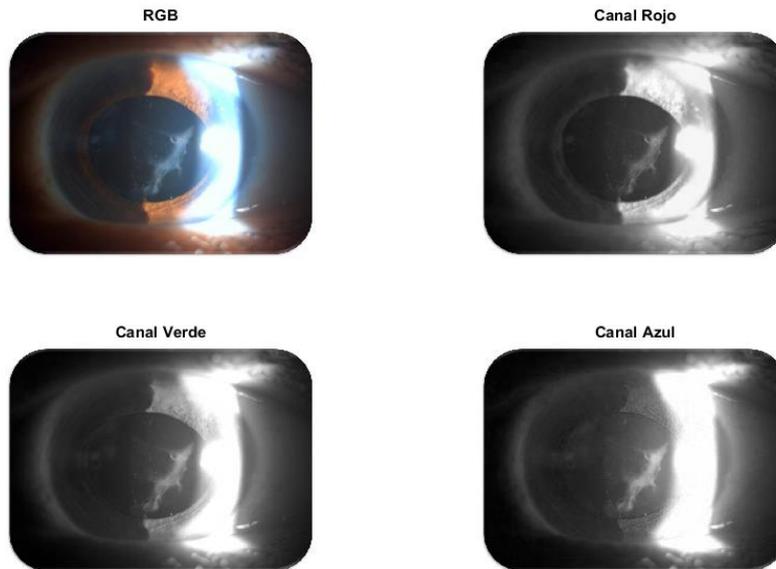


Figura 5: Canales del espacio de color RGB en imagen oblicua de la lámpara de hendidura.

1.3.2 Espacio de color CMY

El cian, el magenta y el amarillo son los colores secundarios de la luz o, alternativamente, los colores primarios de los pigmentos. La mayoría de los dispositivos que depositan pigmentos de colores en el papel, tales como las impresoras y fotocopiadoras, requieren datos CMY o realizan una conversión interna de RGB a CMY. La conversión se realiza de la siguiente forma:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Figura 6: Espacio de color CMY tomado de [5].

bajo la hipótesis de que todos los valores de color están normalizados en el rango [0,1].

En la práctica es fácil comprobar que, a iguales cantidades de rojo (R), verde (G) y azul (B) obtendremos los siguientes como resultantes:

R + G = Amarillo (Y, de yellow en inglés)

G + B = Cian (C)

B + R = Magenta (M)

Actualmente las impresoras de cuatro colores utilizan un cartucho negro además de los colores primarios de este espacio, lo cual genera un mejor contraste. Sin embargo, el

color que una persona ve en una pantalla de computador difiere del mismo color en una impresora, debido a que los RGB y CMY son distintos. El color RGB está hecho por la reflexión o emisión de luz, mientras que el CMY, mediante la absorción de esta [19] [20].

1.3.3 Espacio de color CIEL*a*b*

Este fue introducido en 1976 por CIE (Comisión Internacional de Iluminación) como un dispositivo de espacio de color independiente para que tuviera una percepción visual uniforme aproximada, bajo el precepto de que los colores que son visualmente similares están cerca en el espacio de color. Fue adoptado en esa época como un estándar internacional [23]. El espacio CIEL*a*b* consiste en un componente de luminancia L*, el componente de cromaticidad a* indica donde el color cae a lo largo del eje rojo-verde, y el componente de cromaticidad b* indica donde el color cae a lo largo del eje azul-amarillo [19].

El valor L viene dado por [22]:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16 \quad (1)$$

y las transformaciones a* y b* están dadas por las ecuaciones:

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \quad (2)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \quad (3)$$

Donde

$$f(x) = \begin{cases} x^{\frac{1}{3}} & x > 0.008856 \\ 7.787x + \frac{16}{116} & x \leq 0.008856 \end{cases} \quad (4)$$

Y X_n, Y_n, Z_n son el triple estímulo del estímulo blanco, el cuál es típicamente el estímulo de brillo en el campo de visión. El espacio de color CIEL*a*b* es ampliamente usado en imageneología, transformaciones hacia y desde el espacio CIEL*a*b* a otros espacios de color son comúnmente empleados [22].

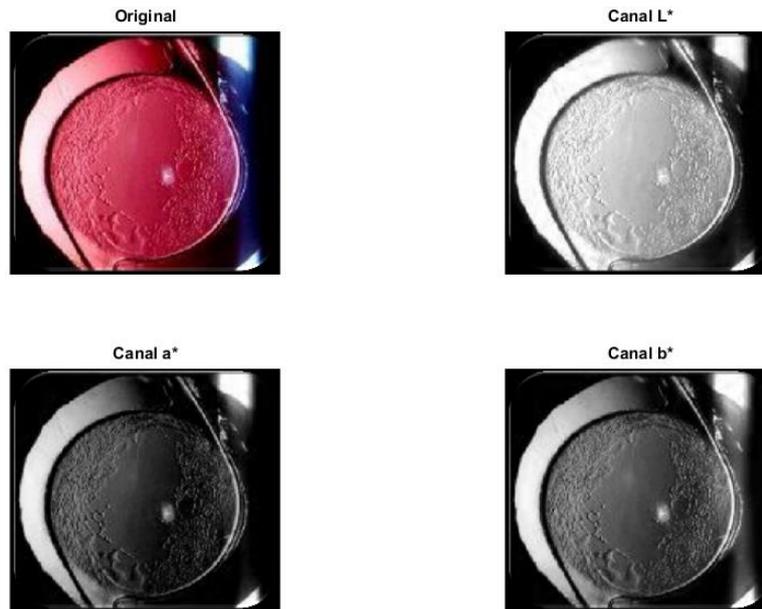


Figura 7: Canales del espacio de color CIE L*a*b* de imagen en retroiluminación.

1.3.4 Espacio de color HSL

El espacio de color HSL o HSI describe los colores como son percibidos por el ser humano. HSI o HSL está dado por H (tono), S (saturación) e L (luminancia) o I (intensidad). El tono viene dado como una propiedad del color de la luz. Este puede ser también concebido como una propiedad de la superficie que refleja o transmite la luz. Por ejemplo, un carro azul refleja un tono azul [19] [23].

El tono que es esencialmente el componente cromático de nuestra percepción puede ser nuevamente considerado como un tono débil o un tono fuerte. La coloración está descrita por el componente de saturación. Por ejemplo, el color de una simple monocromática fuente de luz, la cual produce color de una simple onda, es altamente saturada [23].

La luminosidad o intensidad esencialmente provee una medida de la brillantez de los colores. Esto da una medida de cuanta luz es reflejada desde un objeto o cuanta luz es emitida desde una región. La luminosidad (o intensidad) esencialmente ayuda al ojo humano a percibir el color [23].

La saturación se refiere a la cantidad del color o a la pureza de éste. Va de un color claro a un color más vivo (azul cielo – azul oscuro). También se puede considerar como la mezcla de un color con blanco o gris [49].

El espacio de color HSI tiene la ventaja de dissociar la información de los colores y de la escala de grises en una imagen, lo que es muy útil para la aplicación de técnicas de procesamiento en escala de grises [5].

Conversión de RGB a HSL.

Los valores (R, G, B) deben ser expresados como números del 0 al 1. MAX equivale al máximo de los valores (R, G, B), y MIN equivale al mínimo de esos valores. La fórmula puede ser escrita como:

$$H = \begin{cases} 0, & \text{si } MAX = MIN \\ (60 \times \frac{G-B}{MAX-MIN} + 360) \bmod 360, & \text{si } MAX = R \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{si } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{si } MAX = B \end{cases} \quad (5)$$

$$L = \frac{1}{2}(MAX + MIN) \quad (6)$$

$$S = \begin{cases} 0, & \text{si } MAX = MIN \\ \frac{MAX-MIN}{MAX+MIN} = \frac{MAX-MIN}{2L}, & \text{si } L \leq \frac{1}{2} \\ \frac{MAX-MIN}{2-(MAX+MIN)} = \frac{MAX-MIN}{2-2L}, & \text{si } L > \frac{1}{2} \end{cases} \quad (7)$$

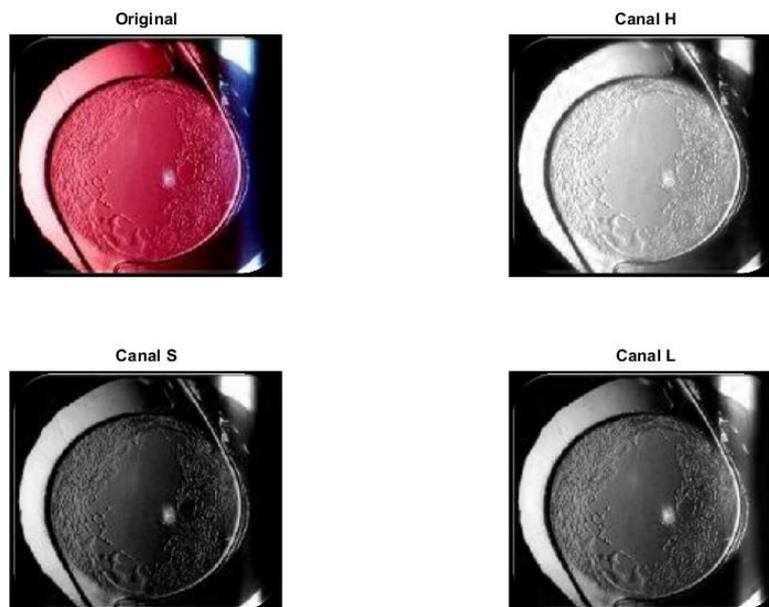


Figura 8: Canales del espacio de color HSL de imagen en retroiluminación.

1.3.5 Espacio de color HSB

El espacio de color HSB (Tono, Saturación y Brillo), fue creado en 1978 como una forma generalizada del espacio de color HSL, y es frecuentemente usado en procesamiento digital de imágenes [49]. Este espacio de color orientado al usuario está basado en el atractivo de las tintas de los artistas. Los tres componentes que lo constituyen son Tono (con el mismo significado que en el espacio de color HSL), la saturación (similar a HSL), y el brillo del color B (con rango desde 0 para el negro y 1 para el brillo máximo) [19].

Brevemente el tono, saturación y valor indica la calidad por la cual distinguimos una familia de color de otra, un color fuerte de uno débil, un color claro de uno oscuro respectivamente [22].

Conversión de RGB a HSB.

$$H = \begin{cases} \text{no definido, si } MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, \text{ si } MAX = R \text{ y } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, \text{ si } MAX = R \text{ y } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, \text{ si } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, \text{ si } MAX = B \end{cases} \quad (8)$$

$$S = \begin{cases} 0, \text{ si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, \text{ en otro caso} \end{cases} \quad (9)$$

$$B = MAX \quad (10)$$

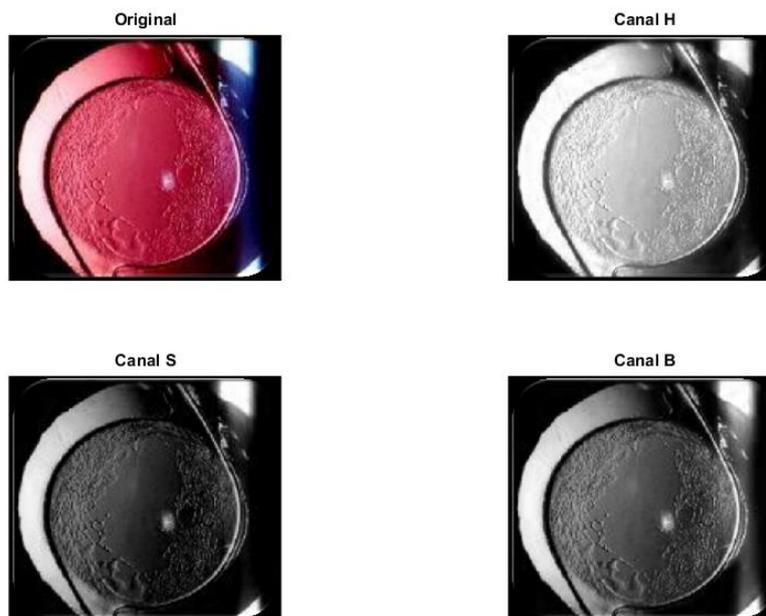


Figura 9: Canales del espacio de color HSB de imagen en retroiluminación.

1.3.6 Espacio de color YCbCr

El espacio de color YCbCr de espacio de color, algunas veces abreviado a YCC [19], es ampliamente usado en video y fotografía digital [23]. En este espacio de color, Y es el componente lumínico, y es obtenido por la suma pesada de los valores de R, G y B [19] [23]. La información del color es almacenada en 2 componentes diferentes Cb y Cr. Cb es la diferencia entre el componente azul y un valor de referencia, y Cr la diferencia entre el componente rojo y un valor de referencia. EL componente de luminancia Y está

definido para tener un rango nominal de $[\frac{16}{255}, \frac{235}{255}]$. La diferencia azul Cb y la diferencia roja Cr están definidas para tener un rango nominal de $[\frac{16}{255}, \frac{240}{255}]$.

Conversión de RGB a YCbCr.

$$Y = 16 + (65.481 \times R + 128.553 \times G + 24.966 \times B) \quad (11)$$

$$Cb = 128 + (-37.797 \times R - 74.203 \times G + 112.0 \times B) \quad (12)$$

$$Cr = 128 + (112.0 \times R - 93.786 \times G + 18.214 \times B) \quad (13)$$

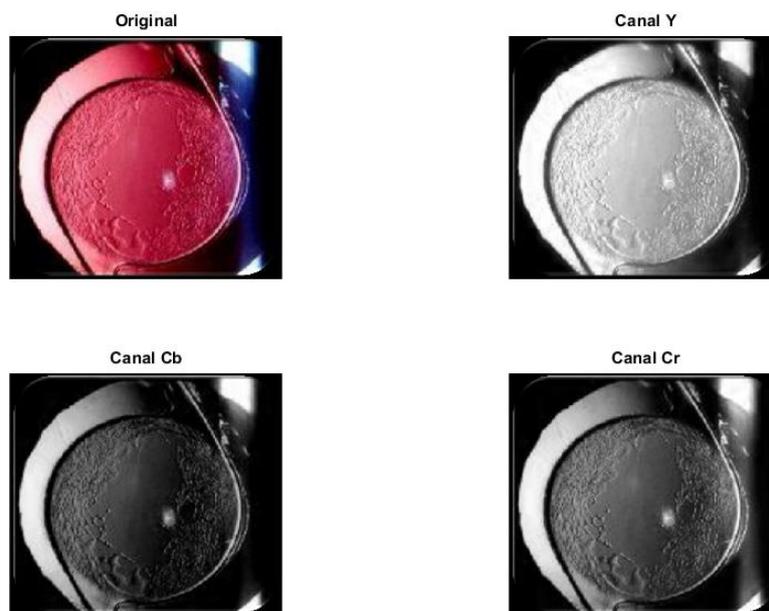


Figura 10: Canales del espacio de color HSB de imagen en retroiluminación.

1.4 Ruido y Filtrado Digital de imágenes

En este apartado se presentan las mejoras que se pueden realizar en la imagen. Todo ello enfocado a que la imagen resultante facilite etapas posteriores de procesado. Las técnicas están en gran medida orientadas al problema concreto, o bien respecto a la eliminación del ruido introducido en el proceso de captura, o bien respecto al aumento del contraste debido a deficiencias en la iluminación [24].

La imagen adquirida puede tener varias fuentes de degradación, dando lugar también a algunos tipos de ruidos. Como regla general, cualquier algoritmo o sistema de tratamiento que busque atenuar estas alteraciones, puede interpretarse como un filtro [25]. En este trabajo se entiende por filtro aquel sistema lineal e invariante que permite el paso de las componentes de la señal existentes en un determinado intervalo (o dominio) frecuencial o espacial.

1.4.1 Ruido en imágenes digitales

Todas las imágenes tienen cierta cantidad de ruido que se manifiesta generalmente en píxeles aislados cuyo valor de gris es diferente al de sus vecinos. El ruido según [26] puede clasificarse en cuatro tipos:

Gaussiano: Produce pequeñas variaciones en la imagen. Siempre consideramos que el valor final de un píxel sería el ideal más una cantidad correspondiente al error, que puede describirse como una variable gaussiana.

Impulsivo (conocido como sal y pimienta): El valor que toma el píxel no tiene relación con el valor ideal sino con el valor del ruido que toma valores muy altos o muy bajos. Se caracteriza porque el píxel toma un valor máximo, o un valor mínimo si se ha perdido la señal que lo define.

Frecuencia: La imagen obtenida es la suma entre la imagen ideal y otra señal, la interferencia se caracteriza por ser una senoide de frecuencia determinada.

Multiplicativo: La imagen obtenida es fruto de la multiplicación de dos señales.

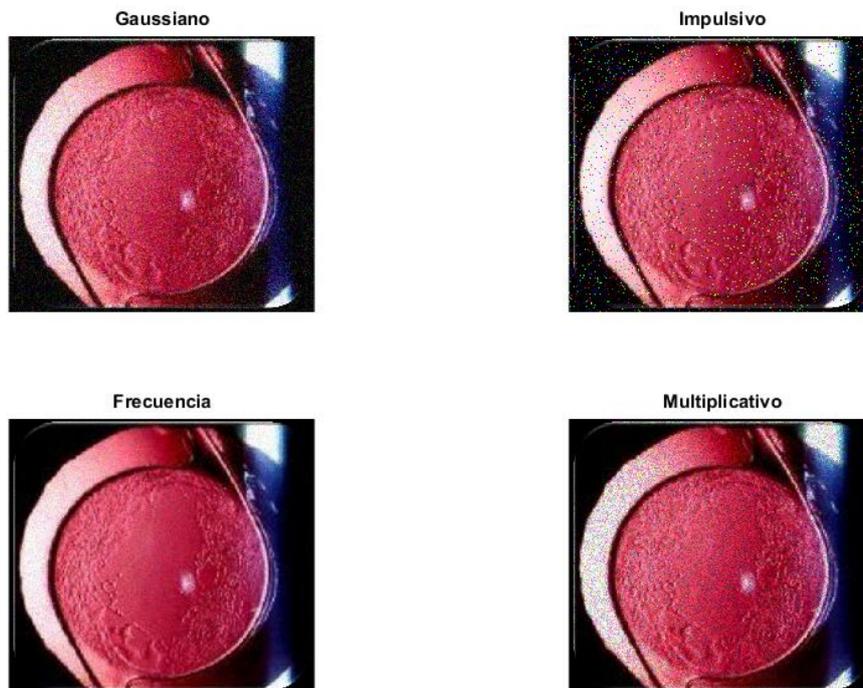


Figura 11: Modelos de ruido en imagen de lámpara de hendidura.

1.4.2 Filtros lineales espaciales

El procesamiento espacial lo constituyen aquellas técnicas que operan directamente sobre los valores de los píxeles de una imagen. Los algoritmos de filtrado se emplean para reducir el ruido de una imagen o para prepararla para una segmentación. La

diferencia entre los *algoritmos lineales* y *no-lineales* es que los primeros se pueden analizar en el dominio de Fourier y los segundos no. Un *filtro lineal* produce salidas que son combinaciones lineales de la secuencia de entrada. Podemos definirlo matemáticamente de la siguiente forma [26]:

$$y(n) = L_n a \quad (14)$$

donde a es la imagen de entrada, $y(n)$ es la salida en el tiempo n , y L_n es el filtro expresado como el valor real de la función para cada n .

1.4.3 Filtrado no lineal espacial

Los algoritmos de filtrado son métodos cuyo fin es favorecer alguna característica presente en una imagen o eliminar otra que esté ocultando lo que se busca. Para llegar al objetivo se utilizan diversas operaciones y transformaciones matemáticas. Las **operaciones** que se pueden aplicar a una imagen digital para transformarla en otra (o a otra representación) según [26] [27] se clasifican en tres categorías, presentadas en la Tabla 1.

Tabla 1 : Operaciones aplicables a una imagen.

Operación	Descripción
Punto	El valor de salida de una coordenada específica; solo depende del valor de entrada de la misma coordenada.
Local	El valor de salida de una coordenada específica, depende de los valores de salida de los vecinos de la misma coordenada.
Global	El valor de salida de una coordenada específica, depende de todos los valores de la imagen de entrada.

Las transformaciones matemáticas son de dos tipos y pueden llevarse a cabo en el dominio del espacio: convolución y correlación, y en el dominio de la frecuencia: transformada de Fourier. La convolución, denotada comúnmente por \otimes , de una función $b(m, n)$ respecto a la función $a(m, n)$ genera una nueva función $c(m, n)$ tal que:

$$c[m, n] = a[m, n] \otimes b[m, n] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a[j, k] b[m - j, n - k] \quad (15)$$

Lo más normal es usar convoluciones de kernels de 3x3 elementos. Entonces la expresión anterior puede concretarse en [26]:

$$c[m, n] = a[m, n] \otimes b[m, n] = \sum_{j=0}^2 \sum_{k=0}^2 a[j, k] b[m - j, n - k] \quad (16)$$

La correlación \circ de $c(m, n)$ entre dos funciones $b(m, n)$ y $a(m, n)$ es:

$$c[m, n] = a[m, n] \circ b[m, n] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a^* [j, k] b[m + j, n + k] \quad (17)$$

donde a^* es el complejo conjugado, sin embargo, en las imágenes digitales, al ser números reales a^* es igual a a .

La *transformada de Fourier* es una extensión de las *series de Fourier* para señales no periódicas. Para ello se toma un nuevo coeficiente c_k que es igual a:

$$c_k = T_0 c_k = \int_0^{T_0} f(t) e^{jk\omega_0 T_0} d(t) \quad (18)$$

por otro lado:

$$f(t) = \frac{1}{T_0} \sum_{k=-\infty}^{\infty} C_k e^{jk\omega_0 T_0} = \frac{\omega_0}{2\pi} \sum_{k=-\infty}^{\infty} C_k e^{jk\omega_0 T_0} \quad (19)$$

ya que $\omega_0 T_0 = 2\pi$.

Como las señales no son periódicas, su período abarca desde $-\infty$ hasta ∞ por lo que:

$$c_k = F(\omega) = \int_{-\infty}^{\infty} f(t) e^{j\omega t} d(t) \quad (20)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega) e^{j\omega t} d(\omega) \quad (21)$$

considerando que:

$$k\omega_0 = \omega \quad (22)$$

$$\omega_0 \rightarrow d\omega \text{ cuando } T_0 \rightarrow \infty \quad (23)$$

para el caso bidimensional discreto se define como:

$$F(u, v) = \frac{1}{n} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-\frac{j2\pi(ux+vy)}{N}} \quad (24)$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) e^{\frac{j2\pi(ux+vy)}{N}} \quad (25)$$

Este tipo de operaciones son utilizadas por los algoritmos de filtrado y aunque existe una gran variedad de algoritmos, podemos hacer una clasificación general de la siguiente manera [26]:

Filtros no lineales:

- RO (de orden no lineal)
- Mediana (FM)

1.4.4 Filtros no lineales de paso bajo (RO)

Los *filtros no lineales* para imágenes basados en un orden estático, los cuales se denominan *filtros RO* (de Orden no Lineal), han sido desarrollados para aplicaciones donde el filtrado lineal es inadecuado [28] [29] [30]. Los filtros lineales suelen fallar en la preservación de los contornos y suprimen impulsivamente los componentes de ruido. De cualquier forma, los filtros basados en orden estático emplean solamente un orden lineal de la secuencia de entrada para producir las salidas. Los filtros RO son robustos porque preservan la información de contornos.

1.4.5 Filtros de la mediana

El filtro que calcula la mediana FM, es aún mejor que los filtros RO. Pero independientemente de que es fácil implementarlo y preserva bien los contornos, tiene limitantes en cuanto al manejo del ruido que puede tener repercusiones [32]. Para corregir esta limitante en [28] se presentan filtros L basados en combinaciones lineales. Un filtro L puede designarse como un criterio óptimo para una gran variedad de distribuciones de ruido. Como caso concreto de filtros L, el movimiento promedio de un filtro es óptimo para el ruido Gaussiano y el FM es cercano al óptimo de un filtro L para el ruido de Laplace. Una de las desventajas principales de los filtros L es que no consideran la correlación temporal de una secuencia de entrada. Como resultado de un filtrado L, la señal a filtrar se supone como una constante representativa que se corrompe mediante la agregación de ruido y no pueden retenerse los contenidos de espectro de la señal de entrada. Se han realizado múltiples trabajos para poder combinar las características de los filtros lineales y los filtros no lineales basados en ordenes estáticos para restablecer señales de frecuencias selectivas de señales no estacionarias [31] [32] [33] [34].

1.5 Ambiente de desarrollo

En el presente apartado se describen las herramientas y tecnologías a utilizar durante el desarrollo del sistema, el análisis realizado, llevará a conocer cuáles son las más adecuadas para el desarrollo.

1.5.1 Herramienta de desarrollo

Para la realización de pruebas de diversos algoritmos y técnicas sobre la imagen oblicua y en retroiluminación, se utilizó la herramienta de software MATLAB (abreviatura de **MAT**rix **LAB**oratory, "laboratorio de matrices"). MATLAB, en su contenido básico, es un entorno integrado de trabajo que permite el análisis y la computación matemáticos interactivos de forma sencilla con más de 500 funciones matemáticas, estadísticas y técnicas implementadas, así como la visualización a través de todo tipo de gráficos de ingeniería y científico [35]. Además, el entorno básico de MATLAB se complementa con una amplia colección de cajas de herramientas que contienen funciones específicas para determinadas aplicaciones en diferentes ramas de las ciencias y la ingeniería. Se hizo énfasis específicamente en el área del procesamiento de imágenes MATLAB.

Específicamente se trabajó con la Toolbox de procesamiento digital de imágenes de MATLAB, Image Processing Toolbox (IPT) le proporciona a MATLAB un conjunto completo de algoritmos y herramientas gráficas para el procesamiento, análisis y visualización de imágenes y para el desarrollo de aplicaciones y de nuevos algoritmos en el campo del procesamiento y análisis de imágenes digitales. Además, el entorno matemático sobre el que se sustenta MATLAB, es ideal para este procesamiento digital de imágenes ya que las imágenes digitales son matrices [36].

Entre las funciones principales de la IPT destacan [36].

- Mejora y filtrado de imágenes.
- Enfoque de imágenes borrosas.
- Análisis de imágenes, incluyendo segmentación, morfología, extracción de funciones y medición.
- Transformaciones geométricas y métodos de registro de imágenes basados en intensidad.

1.5.2 Herramienta para el modelado

La herramienta de modelado seleccionada es el Visual Paradigm para UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo necesario para la obtención de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos.

Ofrece interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo.

1.5.3 Programa Analizador de Opacidad Capsular: PANDOC

El software PANDOC es una herramienta para la cuantificación y análisis de la OCP. Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía postoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto toma las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas orientadas solo al uso de imágenes de tomogramas del Pentacam, lo cual hace más intuitivo la interfaz actual y la focaliza en su propósito principal. Con la presente investigación se pretende que el software PANDOC sea capaz de detectar la OCP no solo en las imágenes del Pentacam, sino que también lo realice con las imágenes provenientes de la lámpara de hendidura. PANDOC es un software que forma parte del grupo de investigación que engloba a esta investigación por lo que es de suma importancia su fortalecimiento para el crecimiento del grupo [37].

Funcionalidades del PANDOC [37].

1. Extraer de manera automática la porción relevante de las imágenes brindadas por los tomogramas del Pentacam. Esta funcionalidad permite abstraer al especialista de realizar dicha operación manualmente según su experiencia.
2. Calcular el grado de la OCP sin incluir los píxeles del fondo. Esta funcionalidad permite que los píxeles de fondo no influyan de forma negativa en la obtención de la intensidad media de los píxeles.
3. Extracción de la intensidad media de píxeles de áreas irregulares de la imagen. Esta funcionalidad amplía la flexibilidad de la aplicación y le da una mayor maniobrabilidad al usuario a la hora de extraer la información que sea relevante al especialista.
4. Mostrar al usuario el porcentaje del área que fue seleccionada considerándose opacidad. Con esta funcionalidad se muestra una variación del color de fondo en las zonas que se consideran opacidad.
5. Determinar el centro de la estructura ocular, la cual mediante la identificación de variación de colores en los píxeles y trilateración celular, define automáticamente el centro de la estructura para funcionalidades posteriores.

6. Calcular el valor de la OCP. Con esta funcionalidad se calcula el valor de la OCP teniendo en cuenta la intensidad de píxeles de esa área multiplicado por el porcentaje de opacidad de la misma, esta expresa dicho valor también con respecto al centro y a la periferia de la estructura ocular.

1.5.4 Metodología de desarrollo

El proceso de desarrollo de software, es definido como el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, tiene como finalidad la obtención de un producto que cumpla con las expectativas del cliente [38].

Según la filosofía de desarrollo se pueden clasificar las metodologías en dos grupos. Las metodologías tradicionales, que se basan en una fuerte planificación durante todo el desarrollo, y las metodologías ágiles, en las que el desarrollo de software es incremental, cooperativo, sencillo y adaptado.

Luego de la evaluación de las metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software [39].

El cliente es parte del equipo de desarrollo, el equipo de desarrollo es de solo una persona, además de la dificultad para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo [40] [38], lo cual permitió identificar la Programación Extrema (XP) como una alternativa acertada, además de ser la metodología de desarrollo del grupo de investigación, AIRI(Artificial Intelligence: Research and Innovation).

Ventajas de XP (Sánchez Mendoza, 2004) [38]:

- Comienza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- El cliente o el usuario se convierte en parte del equipo.

XP consta de 4 fases [41]:

Planificación: Es la fase donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada iteración.

Diseño: La metodología XP hace especial énfasis en los diseños simples y claros. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Codificación: En la fase de codificación de desarrolla en función de cada historia de usuario, además de ser fase donde se definen las tareas de la ingeniería y los tiempos reales en que se realizaron cada una de las funcionalidades especificadas, en la cual la implementación, debe realizarse de acuerdo los estándares de codificación.

Pruebas: Estas pruebas se realizan al final del ciclo en el que se desarrollan, para verificar que las iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

1.6 Métricas de validación

Para validar cuál será el espacio de color más adecuado para el tratamiento de las imágenes provenientes de la lámpara de hendidura en cuanto a ruido y tiempo de ejecución, es necesario aplicar métricas de calidad al filtrado de ruido. Para esto se deciden utilizar las siguientes métricas:

Índice de Similitud Estructural (**SSIM**) se basa en el cálculo de tres términos, a saber, el término de luminancia, el término de contraste y el término estructural. El índice general es una combinación multiplicativa de los tres términos [47].

$$SSIM(x, y) = [l(x, y)^\alpha] * [c(x, y)^\beta] * [s(x, y)^\gamma] \quad (26)$$

donde

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (27)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (28)$$

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (29)$$

Donde μ_x , μ_y , σ_x , σ_y y σ_{xy} son los medios locales, las desviaciones estándar y la covarianza cruzada para las imágenes x, y. Si $\alpha = \beta = \gamma = 1$ (el valor por defecto para los exponentes) y $C_3 = C_2/2$ (selección por defecto de C_3) el índice se simplifica a:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (30)$$

Esta métrica se expresa en el rango de 0 a 1, a medida que el valor *ssim* se acerque más a 1, más semejanza tiene la imagen restaurada con respecto a la imagen original.

Relación señal ruido (**SNR**) y Pico de la relación señal ruido (**PSNR**)

El PSNR calcula la relación pico de señal a ruido, en decibelios, entre dos imágenes. Esta relación se utiliza a menudo como una medida de calidad entre la imagen original y una imagen comprimida. Cuanto mayor sea el PSNR, mejor será la calidad de la imagen comprimida o reconstruida [53].

$$SNR = \frac{PotenciaImagen}{PotenciaRuido} \quad (31)$$

$$PSNR = 10\log_{10}\left(\frac{R^2}{SNR}\right) \quad (32)$$

Donde:

$$PotenciaImagen = x_{rms}^2 \quad (31.1)$$

$$PotenciaRuido = n_{rms}^2 \quad (31.2)$$

Donde *rms* se corresponde a la desviación estándar de la señal y se puede definir como:

$$rms = \left[\frac{1}{N} \sum_{n=1}^N S_n^2 \right]^{\frac{1}{2}} \quad (31.3)$$

Siendo $n=1 \dots N$ el número de muestras de la señal y *S*, la señal que se esté analizando, en el caso de x_{rms}^2 es la señal original y n_{rms}^2 es la señal ruidosa.

En la ecuación 32, *R* es la fluctuación máxima en el tipo de datos de imagen de entrada. Por ejemplo, si la imagen de entrada tiene un tipo de datos de coma flotante de doble precisión, entonces *R* es 1. Si tiene un tipo de datos enteros sin signo de 8 bits, *R* es 255.

Error Cuadrático Medio (**EMC**) [52], mientras menor sea el error, mejor será la calidad de la imagen resultante.

$$EMC = \frac{\sum_{i=1}^M \sum_{j=1}^N \|O_{i,j} - O_{i,j}\|}{MN} \quad (33)$$

M y *N* indican las dimensiones de la imagen. $O_{i,j}$ y $O_{i,j}$, muestra los píxeles originales y los píxeles restaurados localizados en (*i*, *j*), respectivamente.

1.7 Conclusiones del capítulo

El espacio de color RGB presente en las imágenes provenientes de la lámpara de hendidura no es eficiente para el tratado de las mismas debido a que no presenta componentes tales como luminosidad/intensidad.

Luego de un análisis de las investigaciones sobre la línea del procesamiento digital de imágenes en el grupo de investigación AIRI, se identificó que los espacios de color a implementar son (CMY, YCbCr, CIEL*a*b*, HSB, HSL) ya que son los utilizados en los trabajos del grupo.

Para la identificación del espacio de color más adecuado para el tratamiento de estas imágenes, se añadió ruido (Sal y pimienta) y filtrado (filtro de la mediana) a cada una de ellas después de transformarlas a cada uno de los espacio de color analizados, para luego aplicarles las métricas (PSNR, SNR, SSIM, EMC) de calidad al filtrado de ruido y ver en qué espacio de color se comportó mejor estos resultados.

Se seleccionó la metodología de desarrollo XP, por ser la definida por el grupo de investigación (AIRI) y por las bondades que brinda al proceso de desarrollo.

Capítulo 2 Propuesta de solución

En el presente capítulo se describen los principales elementos que se tuvieron en cuenta en el desarrollo de la solución propuesta, así como las técnicas de conversión de espacio de color empleadas. Se describe la ejecución de las fases que define la metodología XP en la presente investigación y además se hace referencia a los artefactos generados en cada una de ellas.

2.1 Descripción de la solución

Para establecer la propuesta de solución se seleccionaron diferentes métodos de procesamiento de imágenes, los cuales se encuentran contenidos en las etapas de preprocesamiento, con el objetivo de transformar el espacio de color de las imágenes provenientes de la lámpara de hendidura, de tal forma que permita al usuario una visión más amplia y con diferentes niveles de detalles sobre la representación de las imágenes de OCP en los pacientes operados de cataratas en otros espacios de color.

Por lo que se propone una secuencia de pasos, que permitirá a los miembros del grupo de investigación AIRI, obtener resultados más precisos en cuanto al comportamiento del ruido en este tipo de imagen y puedan realizar de manera sencilla la conversión entre espacios de color. Estos pasos se describen en la Figura 12.

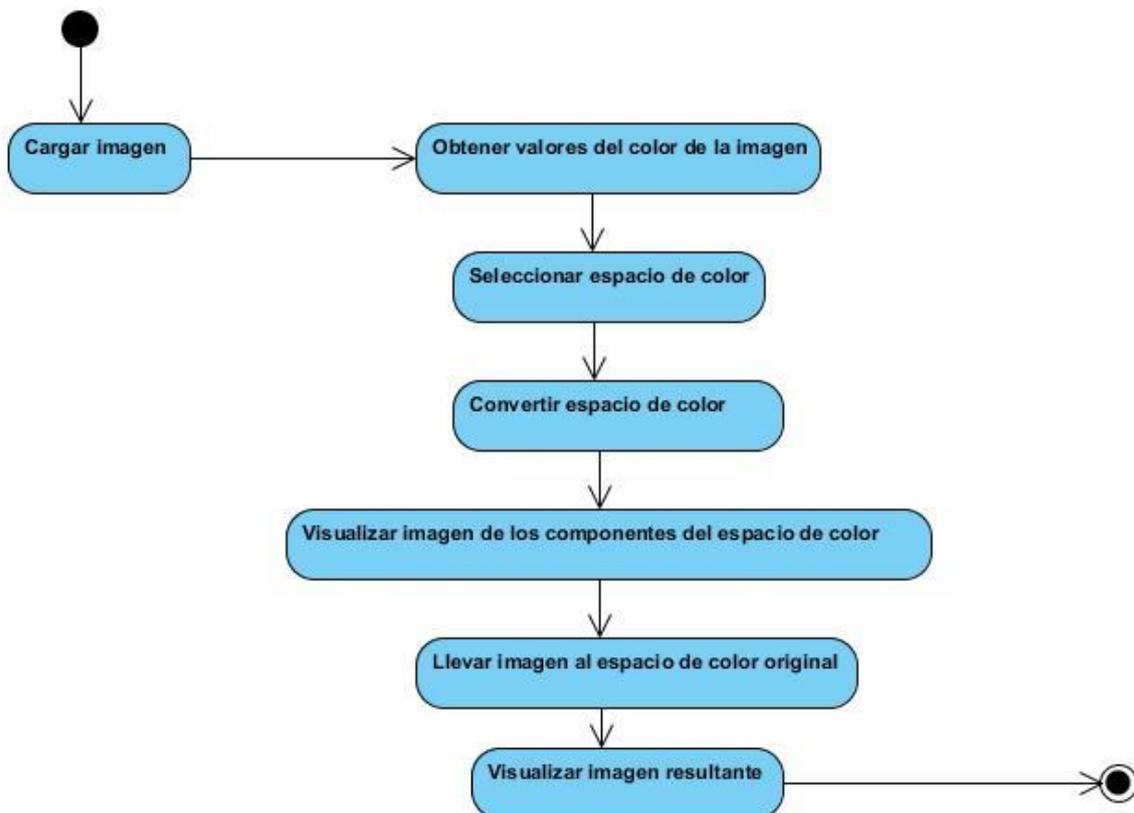


Figura 12: Propuesta de solución representada.

Paso 1. Cargar imagen

Para comenzar la transformación, la imagen debe estar ubicada en el directorio activo de MatLab. Para completar la operación de captura de la imagen para su posterior análisis se emplea la función *imread*, especificando como único parámetro, en este caso, el nombre de la imagen.

Paso 2. Obtener valores del color de la imagen

La imagen se divide en una matriz de $M \times N \times 3$, donde cada una representa la intensidad de los 3 componentes del espacio de color RGB (rojo, verde, azul), a partir de esta se calcula la matriz de intensidades de la matriz principal (suma promediada de los 3 valores de ese espacio de color), tal y como se observa en la Figura 13.

Imágenes RGB (color)

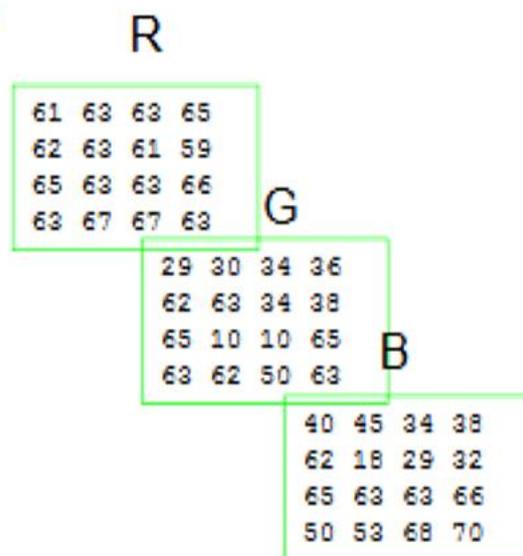


Figura 13: Representación de la matriz $M \times N \times 3$.

Para manipular cualquier característica de la imagen cargada, en este caso la intensidad de color rojo, azul o verde, se debe tener en cuenta que MatLab representa, este y cualquier objeto, como una matriz.

Paso 3. Seleccionar espacio de color

El usuario puede seleccionar el espacio de color al que desea transformar la imagen entre los que se encuentran (CMY, CIEL*a*b*, HSL, HSB, YCbCr).

El sistema interpreta de la siguiente forma las transformaciones seleccionadas por el usuario. A la clase **colorspace** se le envía como parámetro la imagen a transformar, y

una cadena de texto con el siguiente formato 'ECorigen->ECdestino'. Donde ECorigen, es el espacio de color nativo que presenta la imagen cargada, y ECdestino, es el espacio de color al que se desea convertir. Dentro de esta clase se encuentra el método **parse**, el cual se encarga de identificar el espacio de color de origen y el de destino. Para ello este método ubica la cadena '->' la cual indica la conversión, y los valores que se encuentren antes de esta cadena los compara con los que tiene implementados el **colorspace** y si está vacío, entiende es RGB el espacio de color de origen, los valores después de la cadena también son comparados y si se encuentra vacío también se asume se va a convertir hacia RGB.

Después se procede a aplicar las transformaciones para convertir de espacio de color.

Paso 4. Convertir espacio de color

Dentro del procesamiento digital de imágenes existen varias etapas por las que debe transcurrir la imagen para su tratamiento con cualquier tipo de fin. En este caso se transformará el espacio o modelo de color RGB de las imágenes provenientes de la lámpara de hendidura hacia otro espacio de color seleccionado, mediante los modelos matemáticos de conversión referidos en el Capítulo 1 del presente trabajo. Para resolver el problema fue necesario el estudio de funciones nativas de MatLab que apoyaran el desarrollo del módulo propuesto. Dando solución a este paso se produjo la implementación de las funciones:

- cmy: Empleada para transformar el espacio de color de la imagen de RGB a CMY.

Conversión de RGB a CMY.

La conversión se realiza de la siguiente forma:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

bajo la hipótesis de que todos los valores de color están normalizados en el rango [0,1].

- hsl: Empleada para transformar el espacio de color de la imagen de RGB a HSL.

Conversión de RGB a HSL.

Los valores (R, G, B) deben ser expresados como números del 0 al 1. MAX equivale al máximo de los valores (R, G, B), y MIN equivale al mínimo de esos valores. La fórmula puede ser escrita como:

$$H = \begin{cases} 0, si MAX = MIN \\ (60 \times \frac{G-B}{MAX-MIN} + 360) mod 360, si MAX = R \\ 60 \times \frac{B-R}{MAX-MIN} + 120, si MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, si MAX = B \end{cases}$$

$$L = \frac{1}{2}(MAX + MIN)$$

$$S = \begin{cases} 0, si MAX = MIN \\ \frac{MAX-MIN}{MAX+MIN} = \frac{MAX-MIN}{2L}, si L \leq \frac{1}{2} \\ \frac{MAX-MIN}{2-(MAX+MIN)} = \frac{MAX-MIN}{2-2L}, si L > \frac{1}{2} \end{cases}$$

- hsb: Empleada para transformar el espacio de color de la imagen de RGB a HSB.

Conversión de RGB a HSB.

Los valores (R, G, B) deben ser expresados como números del 0 al 1. MAX equivale al máximo de los valores (R, G, B), y MIN equivale al mínimo de esos valores. La fórmula puede ser escrita como:

$$H = \begin{cases} no\ definido, si\ MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, si\ MAX = R\ y\ G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, si\ MAX = R\ y\ G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, si\ MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, si\ MAX = B \end{cases}$$

$$S = \begin{cases} 0, si\ MAX = 0 \\ 1 - \frac{MIN}{MAX}, en\ otro\ caso \end{cases}$$

$B = MAX$

- lab: Empleada para transformar el espacio de color de la imagen de RGB a CIEL*a*b*.

Conversión de RGB a CIEL*a*b*.

El espacio CIEL*a*b* consiste en un componente de luminancia L^* , el componente de cromaticidad a^* indica donde el color cae a lo largo del eje rojo-verde, y el componente de cromaticidad b^* indica donde el color cae a lo largo del eje azul-amarillo.

El valor L viene dado por:

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16$$

y las transformaciones a^* y b^* están dadas por las ecuaciones:

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right)$$

Donde

$$f(x) = \begin{cases} x^{\frac{1}{3}} & x > 0.008856 \\ 7.787x + \frac{16}{116} & x \leq 0.008856 \end{cases}$$

Y X_n, Y_n, Z_n son el triple estímulo del estímulo blanco, el cuál es típicamente el estímulo de brillo en el campo de visión.

- ycbcr: Empleada para transformar el espacio de color de la imagen de RGB a YCbCr.

Conversión de RGB a YCbCr.

EL componente de luminancia Y es obtenido de la suma pesada de los valores R , G y B y su valor está definido para tener un rango nominal de $[\frac{16}{255}, \frac{235}{255}]$. La información del color es almacenada en 2 componentes diferentes Cb y Cr . Cb es la diferencia entre el componente azul y un valor de referencia, y Cr la diferencia entre el componente rojo y un valor de referencia y sus valores están definidos para tener un rango nominal de $[\frac{16}{255}, \frac{240}{255}]$.

$$Y = 16 + (65.481 \times R + 128.553 \times G + 24.966 \times B)$$

$$Cb = 128 + (-37.797 \times R - 74.203 \times G + 112.0 \times B)$$

$$Cr = 128 + (112.0 \times R - 93.786 \times G + 18.214 \times B)$$

Para todos los casos se siguió estrictamente el modelo matemático descrito, para cada espacio de color, en el Capítulo 1 de la presente investigación.

Paso 5. Visualizar imagen de los componentes del espacio de color

Según el espacio del color seleccionado se visualizará la imagen correspondiente a cada uno de los canales del mismo, para ello fueron empleadas las funciones **subplot** e **imshow**, nativas de MatLab.

Paso 6. Llevar imagen al espacio de color original

Se utiliza el modelo matemático inverso para regresar de los espacios de color (CMY, CIEL*a*b*, HSL, HSB, YCbCr) al espacio de color RGB, debido a que desde 1965 se demostró que el aparato óptico humano muestra mayor sensibilidad a tres longitudes de onda emitidas por la luz, comprendidas dentro del espectro visible por el ojo humano que va desde los 400 hasta los 700 nanómetros (nm), donde la vista se hace más sensible al color azul en la longitud de onda de los 445nm, sensible al color verde en la longitud de onda de los 535nm y al color rojo en la longitud de onda de los 575nm, respectivamente. Por lo que se puede afirmar que el ser humano tiene una visión RGB.

Por otra parte, el diseño de los medios digitales de transmisión de señales, particularmente, los que tienen el poder de representar imágenes, lo hacen hasta la fecha en RGB, empleándose otros espacios de color para el trabajo a nivel lógico en determinados procesos, producto a las deficiencias anteriormente planteadas del espacio de color RGB. Estos son unos ejemplos de espacios de color para el trabajo a nivel lógico en determinados procesos:

- CMY es la conversión interna usada en la mayoría de los dispositivos que depositan pigmentos de colores en el papel, tales como las impresoras y fotocopadoras.
- HSL es muy útil para la aplicación de técnicas de procesamiento en escala de grises.
- YCbCr es la conversión interna que se aplica ampliamente en la producción de videos y fotografías digitales.

Además, en este momento ningún medio de cómputo es capaz de visualizar una imagen en un espacio de color diferente al RGB, empleando como alternativa la visualización de las imágenes en escala de grises, perdiéndose detalles de la imagen que, en la gran mayoría de los casos son de mucho interés. Por lo que la transformación inversa viene dada por la incapacidad de las computadoras de representar un espacio de color distinto al RGB y a la imposibilidad de los seres humanos de percibirlo, si lo anterior llegara a ser posible.

Paso 7. Visualizar imagen resultante

Para finalizar con esta secuencia de pasos, apoyándose en la nativa función de MatLab *imshow* se muestra la imagen resultante.

2.2 Fase de planificación

La fase planificación es la que da comienzo al ciclo de vida de un proyecto realizado con la metodología XP. En esta fase los clientes plantean a grandes rasgos las HU y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Además, se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Se deben incluir varias iteraciones para lograr una entrega. El resultado de esta fase es un Plan de Entregas.

2.2.1 Requisitos del módulo

Los requisitos cumplen un papel primordial en el proceso de producción de software y se enfocan en la definición de lo que se desea producir, se clasifican en requisitos funcionales y no funcionales. Los requisitos funcionales indican lo que debe hacer la solución propuesta, es decir, lo que el sistema hace para el usuario. Los requisitos no funcionales, por su parte, se refieren a propiedades que debe cumplir el software, tales como capacidad de almacenamiento, fiabilidad o mantenibilidad. A continuación, se describen los requisitos del módulo propuesto.

Requisitos funcionales

- RF1: Cargar imagen.
- RF2: Obtener valores de la imagen de entrada.
- RF3: Conversión de RGB a CMY.
- RF4: Conversión de RGB a CIEL*a*b*.
- RF5: Conversión de RGB a HSL.
- RF6: Conversión de RGB a HSB.
- RF7: Conversión de RGB a YCbCr.
- RF8: Visualizar imagen de cada uno de los componentes del espacio de color.

Requisitos no funcionales

Los requisitos no funcionales de hardware se eligieron en función de los prerrequisitos de MatLab y también teniendo que se estará trabajando con imágenes digitales que ocupan espacio en memoria.

▪ Requisito de software

RNF 1: Se debe tener instalada en la computadora la herramienta MATLAB en su versión 8.3.0.532.

- **Requisitos de hardware**

RNF 2: Para ejecutar el módulo la computadora debe contar con las siguientes características:

- 1 GB de memoria RAM.
- Más de un núcleo físico con frecuencia mayor de 1.6 GHz.
- Tarjeta gráfica de 512 MB o superior.
- 4 GB de espacio libre en el disco duro.
- Windows XP en adelante.

2.2.2 Historias de usuario (HU)

La HU es la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas [42].

Las HU se clasifican según [46]:

La prioridad en el negocio:

- Alta: Se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.
- Media: Se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
- Baja: Se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo:

- Alta: Cuando en la implementación de las HU se consideran la posible existencia de errores que conlleven a la inoperatividad del código.
- Media: Cuando pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.

- Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por las siguientes secciones:

- Número: esta sección representa el número, incremental en el tiempo, de la historia de usuario que se describe.
- Nombre de Historia de Usuario: identifica la HU que se describe entre los desarrolladores y el cliente.
- Modificación de Historia de Usuario Número: sección que representa si la HU se le realizó alguna modificación con respecto al estado anterior.
- Usuario: Cliente que realiza la especificación de requisitos, para la posterior conformación de las historia de usuario.
- Programador responsable: Los programadores responsables de la historia de usuario.
- Iteración asignada: número de la iteración donde va a desarrollarse la HU.
- Prioridad en negocio: se le otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.
- Riesgo en Desarrollo: se le otorga una medida de (Alto, Medio, Bajo), a la ocurrencia de errores en el proceso de desarrollo de la HU.
- Puntos Estimados: es el tiempo estimado en semanas que se demorará el desarrollo de la HU.
- Puntos Reales: representa el tiempo que se demoró en realidad el desarrollo de la HU.
- Descripción: breve descripción de la HU.
- Observaciones: señalamiento o advertencia del sistema.

A continuación, se describe en la Tabla2 una de las HU generadas en la presente investigación.

Tabla2 HU "Conversión de RGB a CMY".

Historia de Usuario	
Número: 3	Nombre de la Historia de Usuario: Conversión de RGB a CMY
Modificación a la Historia de Usuario: Ninguna	

Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Alto	Puntos reales: 1 semanas
Programador responsable: Selma Hernández Romero	
Descripción: RGB-CMY: Realiza la conversión del espacio RGB al espacio CMY.	

2.2.3 Estimación de esfuerzo por historias de usuario

La estimación de esfuerzo asociado a la implementación de las historias de usuario la establece los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Si la estimación supera las 3 semanas, la HU deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. En caso de que el esfuerzo sea menor que una semana, la HU será combinada por otra [42].

Tabla 3 Estimación de esfuerzo por HU.

No	Historias de usuario	Puntos de estimación
1	Cargar imagen	1
2	Obtener valores de la imagen de entrada	2
3	Conversión de RGB a CMY	2
4	Conversión de RGB a CIEL*a*b*	3
5	Conversión de RGB a HSL	3
6	Conversión de RGB a HSB	3
7	Conversión de RGB a YCbCr	2
8	Visualizar imagen de cada uno de los componentes del espacio de color	2

2.2.4 Plan de iteraciones

El plan de iteraciones incluye varias iteraciones sobre el módulo antes de ser entregada y se define exactamente qué HU serán implementadas en cada iteración. A continuación, se describen cada una de las iteraciones propuestas, donde la duración total de iteraciones en días, se obtiene a partir del esfuerzo en días estimado por el desarrollador para implementar cada HU:

Tabla 4 Plan de iteraciones.

Iteraciones	Historias de usuario a implementar	Duración
1	HU 1	3
	HU 2	
2	HU 3	15
	HU 4	
	HU 5	
	HU 6	
	HU 7	
	HU 8	

2.2.5 Plan de entrega

El plan de entrega tiene como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una.

Tabla 5 Plan de entrega.

	Iteración 1	Iteración 2
Cantidad de HU	3 (3 semanas)	6 (15 semanas)
Fecha de entrega	14/1/2017	29/5/2017

2.3 Diseño de la solución

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.

Define que utilizar una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código [41] [42].

2.3.1 Tarjetas CRC

La metodología XP en lugar de utilizar diagramas para desarrollar modelos representa las clases mediante tarjetas. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan a definir actividades durante el diseño del sistema.

Estas tarjetas representan una entidad del sistema, a la cual asignar responsabilidades y colaboraciones. El formato físico de las tarjetas CRC facilita la interacción entre los participantes del proyecto, en sesiones en las que se aplican técnicas de grupos como tormenta de ideas o juego de roles, y se ejecutan escenarios a partir la de especificación de requisitos, historias de usuarios o casos de uso. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones [43].

Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

A continuación se muestra en la Tabla 6, una de las tarjetas CRC de la clase colorspace que presenta como responsabilidades 16 métodos y 20 clases que colaboradoras.

colorspace	
Description: Transforma el espacio de color de las imágenes	
Attributes:	
Name	Description
Imagen	Es la imagen de entrada que va a ser transformada de un espacio de color a otro.
Conversion	Es un string que contiene la configuración de espacio de color a realizar. Ej: rgb->lab
Responsibilities:	
Name	Collaborator
parse	lower,strep, size,error,alias
rgb	strep,hue_torgb,gammacorrection
xyz	invf,rgb,invgammacorrection
hsv	rgb,rgbtohue
hsl	rgb,rgbtohue
lab	xyz,f
huertorgb	round,reshape
rgbtohue	sort,zeros
gammacorrection	real
invgammacorrection	real
f	real
invf	fY
gettransform	strep
alias	error
cmy	rgb
ycbcr	rgb2ycbcr

Tabla 6 Tarjeta CRC colorspace.

2.3.2 Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez [50].

El estándar que se ha usado en este caso, es el CamelCase. El mismo se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. El nombre CamelCase se podría traducir como (Mayúsculas/Minúsculas Camello). El término case se traduce como "caja tipográfica", que a su vez implica si una letra es mayúscula o minúscula [50].

Existen dos tipos de CamelCase [50]:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: `SrcSpace`, `DestSpace`.
- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: `invGammaCorrection`

2.4 Conclusiones del capítulo

Se utiliza la metodología XP para realizar el análisis y diseño de la aplicación, generando como artefactos un total de 8 HU que describen los aspectos principales a tener en cuenta para el desarrollo de la solución. El cliente estableció la prioridad de cada una de las HU que describen las funcionalidades del sistema, puntualizando el orden de su implementación y las iteraciones en que serán implementadas. También se logra traducir los requerimientos a funcionalidades a implementar, usando las tarjetas CRC que propone la fase de Diseño de la metodología.

Se definen 7 pasos lógicos, los cuales describen el comportamiento que debe seguir el sistema con vista a dar solución al módulo de transformación (valor agregado de la presente investigación).

Capítulo 3 Resultados y validación

En el desarrollo del presente capítulo, se describen las fases de implementación y prueba que propone la metodología XP, generando como artefacto, en el caso de la fase de implementación, las tareas de desarrollo que dan solución a cada una de las historias de usuarios identificadas en la fase de planificación. Se comparan los espacios de color en cuanto métricas de filtrado de ruido y tiempo de ejecución. También se realizan un conjunto de pruebas al sistema, divididas en pruebas de caja blanca, pruebas de caja negra y pruebas de aceptación.

3.1 Fase de implementación

Se realiza, en esta fase, la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Una vez identificadas las HU, los programadores proceden a descomponer cada una en tareas específicas, las denominadas tareas de la programación para de esta forma poder organizar la implementación. Estas tareas, al contrario de las HU escritas en el lenguaje del cliente, son escritas en un lenguaje técnico. A continuación, se describe la tarea de ingeniería correspondiente a la historia de usuario 3, tomada como ejemplo en el capítulo anterior que tiene como nombre: Conversión de RGB a CMY.

Tarea de ingeniería	
Número de tarea: 3	Historia de Usuario (No.3): Conversión de RGB a CMY
Nombre de tarea: Convertir de RGB a CMY	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha de inicio: 09/01/2017	Fecha de fin: 14/01/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB a CMY	

Tabla 7 Tarea de ingeniería 3.

3.2 Interfaz gráfica

A continuación, se muestra la interfaz que permite visualizar los resultados de la transformación a los espacios de color referidos en la investigación. La misma cuenta

con un conjunto de funcionalidades para realizar dicha transformación, mostrándose para cada espacio de color, sus respectivos canales.

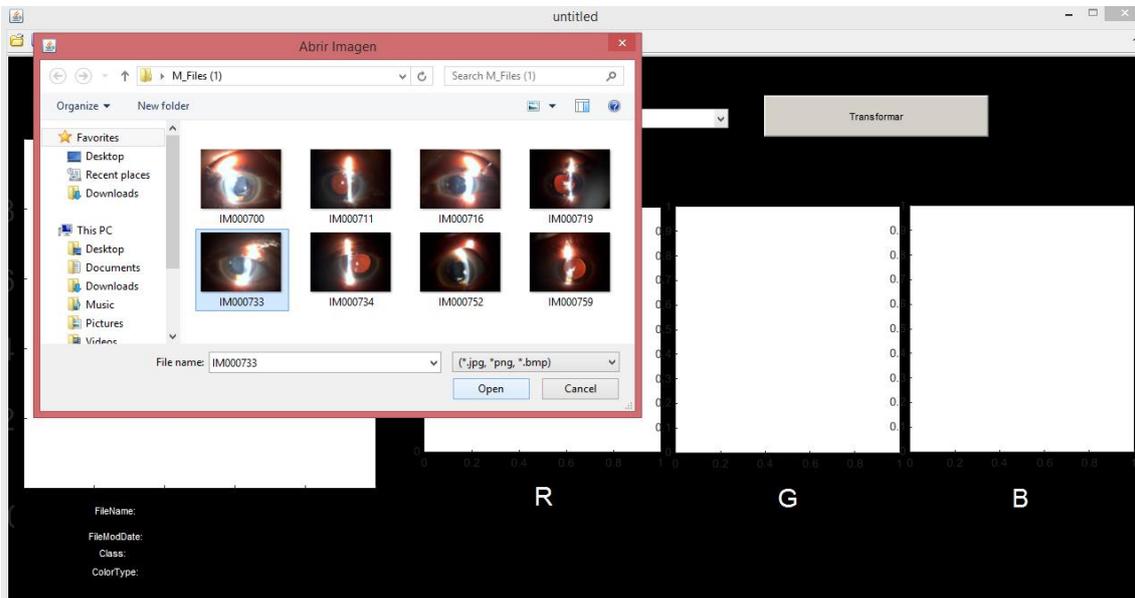


Figura 14: Abrir imagen.

En la Figura 14, se muestra el procedimiento a seguir para cargar la imagen, para esto se busca la misma y se selecciona, siendo posible analizar las imagenes en retroiluminación o en iluminación oblicua del ojo del paciente.

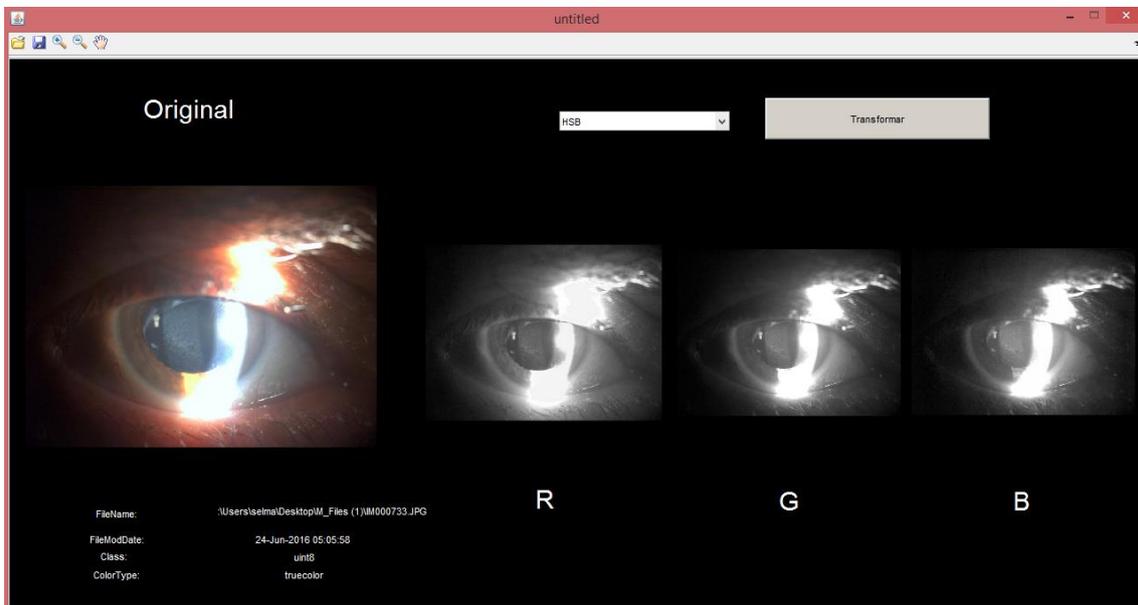


Figura 15: Presentación de la imagen cargada.

Una vez cargada, esta se puede ver en la interfaz principal, como se muestra en la Figura 15.

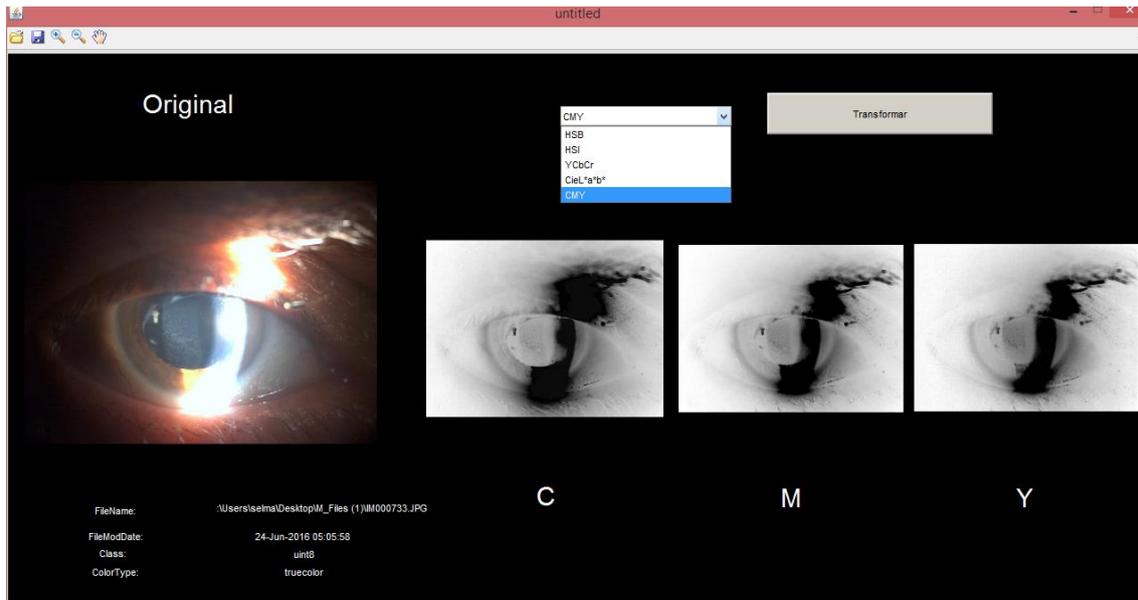


Figura 16: Presentación de los canales del espacio de color seleccionado.

Luego, se selecciona el espacio de color al cual se desea transformar, mostrándose automáticamente, las imágenes de las tres componentes (Figura 16) del espacio de color seleccionado para dicha transformación.

3.3 Resultados

El módulo fue probado en 24 imágenes provenientes de la lámpara de hendidura, 12 de ellas en retroiluminación y 12 en iluminación oblicua. A continuación, se muestran los resultados obtenidos de realizar la transformación a cada uno de los espacios de color por cada tipo de estas imágenes en retroiluminación y en iluminación oblicua.

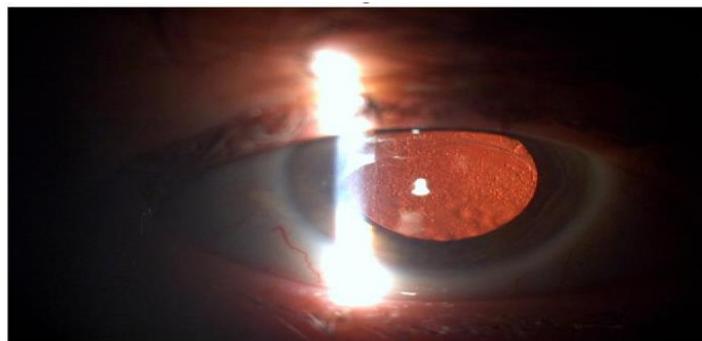


Figura 17: Imagen original en retroiluminación.

En la Figura 18 se muestran tres imágenes, representando, de izquierda a derecha, los componentes del espacio de color CIEL*a*b* (L^* , a^* , b^*) respectivamente.

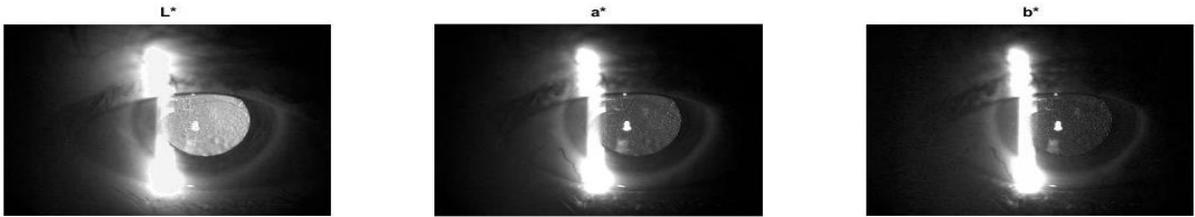


Figura 18: Canales del espacio de color CIEL*a*b* de la imagen en retroiluminación.

Por otra parte, la Figura 19 contiene de igual forma tres imágenes, representando, de izquierda a derecha, los componentes del espacio de color CMY.

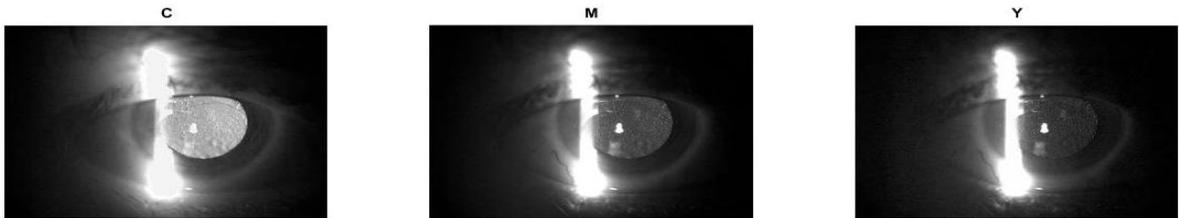


Figura 19: Canales del espacio de color CMY de la imagen en retroiluminación.

En la Figura 20, también se muestran tres imágenes, representando, de izquierda a derecha, los componentes H, S, B, pertenecientes al espacio de color HSB.

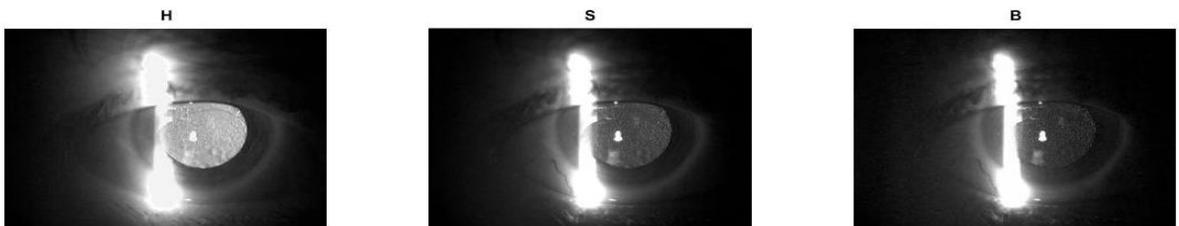


Figura 20: Canales del espacio de color HSB de la imagen en retroiluminación.

También podemos ver que en la Figura 21, se muestran las tres imágenes, representando, de izquierda a derecha, los componentes de dicho espacio de color (H, S, L).

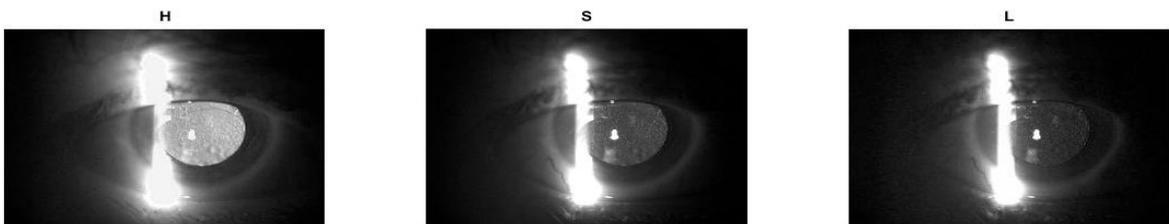


Figura 21: Canales del espacio de color HSL de la imagen en retroiluminación.

En la Figura 22, se muestran de izquierda a derecha, los componentes (Y, Cb, Cr), correspondientes al espacio de color.

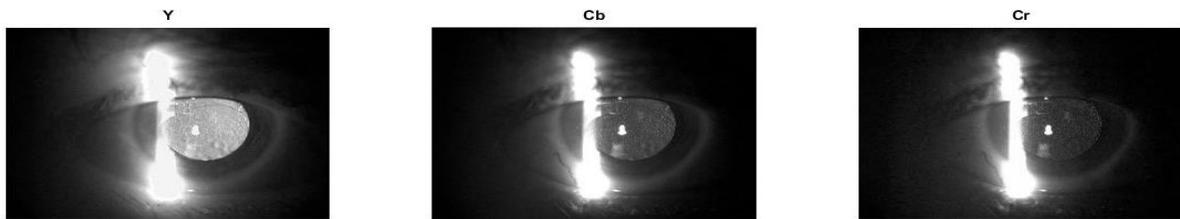


Figura 22: Canales del espacio de color YCbCr de la imagen en retroiluminación.

De los filtros estudiados, el más conocido en el procesamiento digital de imágenes es el filtro de la mediana, el cual es particularmente muy eficiente para la reducción del ruido “Salt & Pepper” [51]. Es por ello que se ha seleccionado esta combinación de ruido y filtrado, para ser empleado como apoyo en la presente investigación.

Una vez que se han filtrado las imágenes, se procede a evaluar los resultados de la eliminación de ruido con las métricas que a continuación se describen:

- **SNR:** Describe cómo se comporta el ruido en la imagen, mientras mayor sea el valor, menor será la cantidad de ondas transmitidas con ruido.

$$SNR = \frac{PotenciaImagen}{PotenciaRuido}$$

$$PotenciaImagen = x_{rms}^2$$

$$PotenciaRuido = n_{rms}^2$$

Donde *rms* se corresponde a la desviación estándar de la señal y se puede definir como:

$$rms = \left[\frac{1}{N} \sum_{n=1}^N S_n^2 \right]^{\frac{1}{2}}$$

Siendo $n= 1 \dots N$ el número de muestras de la señal y *S*, la señal que se esté analizando, en el caso de x_{rms}^2 es la señal original y n_{rms}^2 es la señal ruidosa.

- **PSNR:** El PSNR calcula la relación pico de señal a ruido, en decibelios, entre dos imágenes. Esta relación se utiliza a menudo como una medida de calidad entre la imagen original y una imagen comprimida. Cuanto mayor sea el PSNR, mejor será la calidad de la imagen comprimida o reconstruida.

$$PSNR = 10 \log_{10} \left(\frac{R^2}{SNR} \right)$$

En la ecuación anterior, R es la fluctuación máxima en el tipo de datos de imagen de entrada. Por ejemplo, si la imagen de entrada tiene un tipo de datos de coma flotante de doble precisión, entonces R es 1. Si tiene un tipo de datos enteros sin signo de 8 bits, R es 255.

- **SSIM**: Esta métrica expresa el valor del índice de similitud estructural en el rango de 0 a 1, a medida que el valor ssim se acerque más a 1, más semejanza tiene la imagen restaurada con respecto a la imagen original. Se basa en el cálculo de tres términos, a saber, el término de luminancia, el término de contraste y el término estructural. El índice general es una combinación multiplicativa de los tres términos.

$$SSIM(x, y) = [l(x, y)^\alpha] * [c(x, y)^\beta] * [s(x, y)^y]$$

donde

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$c(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Donde μ_x , μ_y , σ_x , σ_y y σ_{xy} son los medios locales, las desviaciones estándar y la covarianza cruzada para las imágenes x, y. Si $\alpha = \beta = \gamma = 1$ (el valor por defecto para los exponentes) y $C_3 = C_2/2$ (selección por defecto de C_3) el índice se simplifica a:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

- **EMC**: En el error cuadrático medio mientras menor sea el valor error, mejor será la calidad de la imagen resultante, esta medida se calcula como describe la siguiente fórmula:

$$EMC = \frac{\sum_{i=1}^M \sum_{j=1}^N \| O_{i,j} - \hat{O}_{i,j} \|^2}{MN}$$

M y N indican las dimensiones de la imagen. $O_{i,j}$ y $\hat{O}_{i,j}$, muestra los píxeles originales y los píxeles restaurados localizados en (i, j), respectivamente.

El otro parámetro que se ha evaluado para medir el la elección del espacio de color más adecuado es el tiempo de ejecución, para hacerlo se emplean las funciones de MatLab que a continuación se describen:

Tic inicia un cronómetro para medir el rendimiento. La función registra el tiempo interno en la ejecución del comando tic. Muestra el tiempo transcurrido con la función toc.

Toc lee el tiempo transcurrido desde el temporizador del cronómetro iniciado por la función tic. La función lee el tiempo interno en la ejecución del comando toc, y muestra el tiempo transcurrido desde la llamada más reciente a la función tic que no tuvo salida, en segundos.

Para el análisis de la selección del espacio de color más adecuado se procede a conformar la tabla 8. Para llenar el valor correspondiente a la intercepción de cada fila (espacio de color) con cada columna (métricas de calidad de filtrado y tiempo de ejecución) se procede a determinar el valor promedio obtenido por cada parámetro (columna) al ser aplicado a las 12 imágenes por cada uno de los espacio de color.

Espacio de color	Tipo de imagen	Tiempo promedio	Promedio por métricas			
			PSNR	SNR	EMC	SSIM
CMY	Retroiluminación	124.767225	28.202675	17.653225	98.400075	0.6789
CIEL*a*b*	Retroiluminación	126.778575	28.217175	17.668625	98.0841	0.67905
HSB	Retroiluminación	139.1305	28.2364	17.687025	97.63135	0.6796
HSL	Retroiluminación	127.52225	28.2339	17.684225	97.678575	0.67945
YCbCr	Retroiluminación	131.61185	28.214125	17.664825	98.142275	0.6789

Tabla 8 Análisis de los resultados promedio de la transformación de los espacios de color en imágenes en retroiluminación (rojo: peor resultado, amarillo: mejor resultado).

Del análisis de la tabla 8 se puede concluir que el espacio de color más desfavorable a elegir es el CMY y a pesar de este hecho es el de mejores resultados en cuanto a tiempo de ejecución, proponiéndose a investigaciones donde se pueda prescindir de los detalles que pudieran omitirse en la transformación, siendo lo más importante la rapidez de procesamiento de la misma.

También se puede apreciar que, por el contrario, el espacio de color más adecuado para el trabajo con imágenes tomadas en retroiluminación empleando la lámpara de hendidura sería el HSB, siendo este el de peor resultado en cuanto a tiempo de ejecución.

Una vez concluido el análisis a las imágenes tomadas en retroiluminación, se procede a contrastar los resultados con los que sean obtenidos del análisis de las imágenes tomadas en iluminación oblicua. A continuación, en la figura 23, se muestra un ejemplo de este tipo de imágenes.

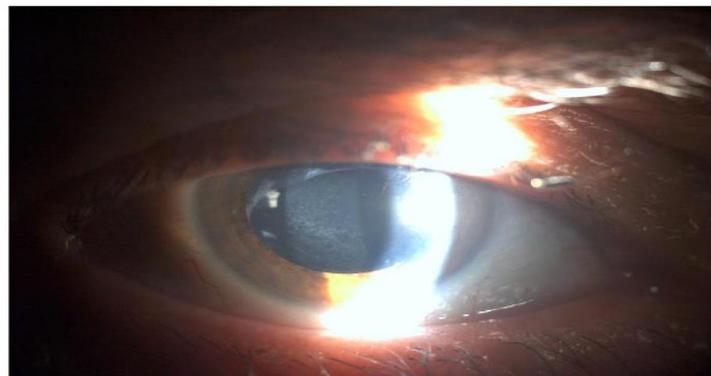


Figura 23: Imagen original en iluminación oblicua.

En la Figura 24, se puede observar los tres componentes (L^* , a^* , b^*), correspondientes al espacio de color CIEL*a*b*, pero en este caso en imágenes de iluminación oblicua.

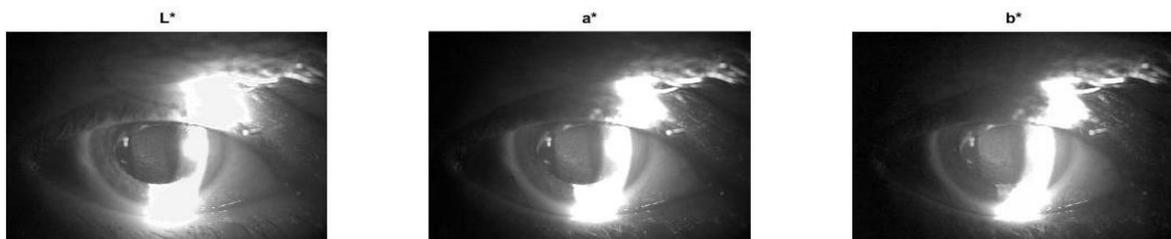


Figura 24: Canales del espacio de color CIEL*a*b* de la imagen en iluminación oblicua.

Ahora bien, en la Figura 25, se muestran los tres componentes (C, M, Y), correspondientes al espacio de color CMY.

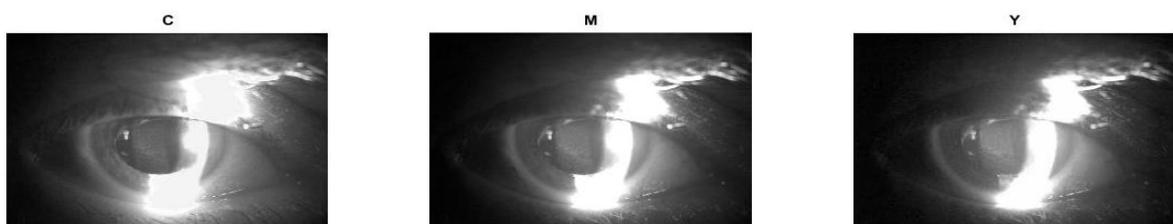


Figura 25: Canales del espacio de color CMY de la imagen en iluminación oblicua.

También se muestra en la Figura 26, los tres componentes (H, S, L), correspondientes al espacio de color HSL, distribuidos en estas tres imágenes.

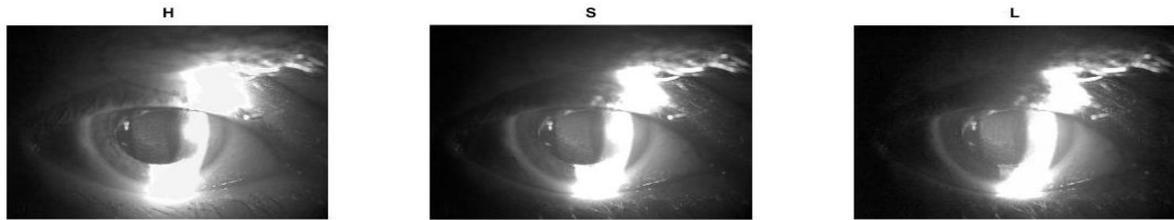


Figura 26: Canales del espacio de color HSL de la imagen en iluminación oblicua.

Se puede ver en la Figura 27, las imágenes de los tres componentes (H, S, B), en imágenes de iluminación oblicua.

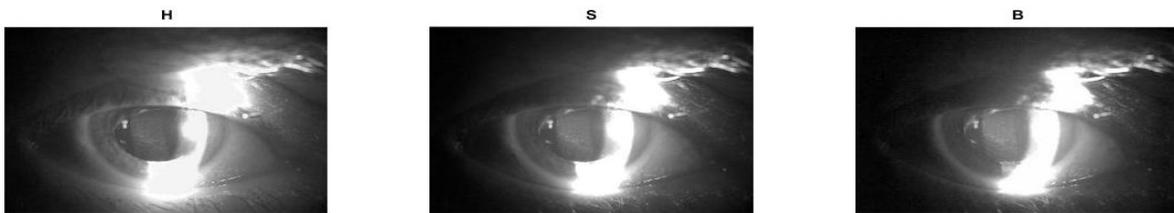


Figura 27: Canales del espacio de color HSB de la imagen en iluminación oblicua.

Por último, en la Figura 28, se muestran los componentes por cada canal del espacio de color (Y, Cb, Cr).

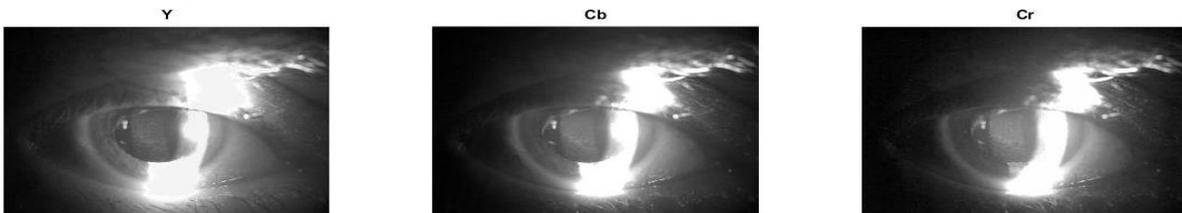


Figura 28: Canales del espacio de color YCbCr de la imagen en iluminación oblicua.

La elaboración y análisis de la Tabla 9 es muy similar al de la Tabla 8, tratándose en este caso con las 12 imágenes en iluminación oblicua.

Espacio de color	Tipo de imagen	Tiempo promedio	Promedio por métricas			
			PSNR	SNR	EMC	SSIM
CMY	Iluminación oblicua	137.444375	28.66615	20.80095	88.72815	0.763125
CIEL*a*b*	Iluminación oblicua	136.635425	28.648575	20.78315	89.09435	0.762675

HSB	Iluminación oblicua	125.279525	28.662975	20.797375	88.83665	0.762425
HSL	Iluminación oblicua	128.28105	28.6352	20.77055	89.36045	0.761375
YCbCr	Iluminación oblicua	123.925025	28.6333	20.768025	89.42885	0.7616

Tabla 9 Análisis de los resultados por promedio de la transformación de los espacios de color en imágenes en iluminación oblicua (rojo: peor resultado, amarillo: mejor resultado).

En la Tabla 9 se puede evidenciar que los resultados obtenidos, indican que el mejor espacio de color para el tratamiento de este tipo de imágenes en cuanto a las métricas es el CMY y el peor es el YCbCr. Y todo lo contrario en cuanto al tiempo de ejecución.

Resulta evidente notar que los resultados, del análisis hecho, difieren en cuanto a la elección del espacio de color más adecuado en cuanto a los parámetros establecidos en la presente investigación. Por otra parte, es de interés para el grupo de investigación, conocer el espacio de color con mayor índice de elegibilidad, sin importar el tipo de imagen del que se trate el análisis que será realizado, por lo que se procede a la confección de la tabla 10, la cual a continuación se describe.

La tabla 10 muestra un análisis más general y similar al de las Tablas 8 y 9, realizándose el análisis en este caso para las 24 imágenes (12 retroiluminación, 12 iluminación oblicua). Para su confección se sigue la filosofía descrita previamente, aunque con las 24 imágenes con las que se cuenta.

Espacio de color	Tiempo promedio	Promedio por métricas			
		PSNR	SNR	EMC	SSIM
CMY	131.1058	28.4344125	19.2270875	93.5641125	0.7210125
CIEL*a*b*	131.707	28.432875	19.2258875	93.589225	0.7208625
HSB	131.623076	28.4416007	19.2341993	93.4087654	0.7210125
HSL	127.90165	28.43455	19.2273875	93.5195125	0.7204125
YCbCr	127.768438	28.4237125	19.216425	93.7855625	0.72025

Tabla 10 Análisis de los resultados por promedio de la transformación de los espacios de color en ambas imágenes (rojo: peor resultado, amarillo: mejor resultado).

Al observar los resultados de los promedios de las métricas por cada espacio de color y el promedio de sus respectivos tiempos de ejecución, se pudo apreciar que el más adecuado para manipulación de estas imágenes provenientes de la lámpara de hendidura, es HSB. De esta forma, en cualquier investigación del grupo, la elección de este espacio de color queda respaldada, siempre que los análisis que se hagan dependan del índice de eliminación de ruido y bajos tiempos de ejecución en el procesamiento.

3.4 Pruebas

Uno de los pilares de la Programación Extrema (XP) es el proceso de pruebas [44]. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. XP divide las pruebas del sistema en dos grupos [44]: pruebas de caja blanca las cuales se concentran en los componentes individuales asegurándose que funcionen de manera apropiada como unidad, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.4.1 Pruebas de caja blanca

Este tipo de prueba se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones, bucles o ambos [45]. En la metodología XP los programadores deben definir las pruebas de caja blanca para cada módulo antes de escribir el código, y sólo para aquellas unidades de código donde exista la posibilidad de que puedan fallar es necesario escribir casos de prueba [46].

A continuación, se describirá la confección de las pruebas de caja blanca realizadas, desglosándola en sus elementos fundamentales: confección del grafo de flujo, cálculo de la complejidad ciclomática, extracción de los caminos independientes, realización de los casos de pruebas y análisis de los resultados obtenidos.

En la Figura 29, se muestra un ejemplo de la función *parse* para determinar los elementos fundamentales de este tipo de prueba.

```

function [SrcSpace, DestSpace] = parse(Str)
% identifico si los parámetros (espacio de color origen y destino) están correctos para luego realizar la conversión
if ischar(Str)
    Str = lower(strrep(strrep(Str, '-', ''), '=')); %aquí elimino el guion y me quedo con todo en minúsculas con el método lower
    k = find(Str == '>');
    if length(k) == 1 % aquí identifico el origen y el destino 'src->dest'
        SrcSpace = Str(1:k-1);
        DestSpace = Str(k+1:end);
    else
        k = find(Str == '<');
        if length(k) == 1 % aquí identifico el origen y el destino 'dest<-src'
            DestSpace = Str(1:k-1);
            SrcSpace = Str(k+1:end);
        else
            error(['Invalid conversion, ', Str, '.']);
        end
    end
end
SrcSpace = alias(SrcSpace);
DestSpace = alias(DestSpace);
else %aquí no me entran cadena de destino
    SrcSpace = 1;
    DestSpace = Conversion;
end
return;

```

Figura 29: Código del método parse.

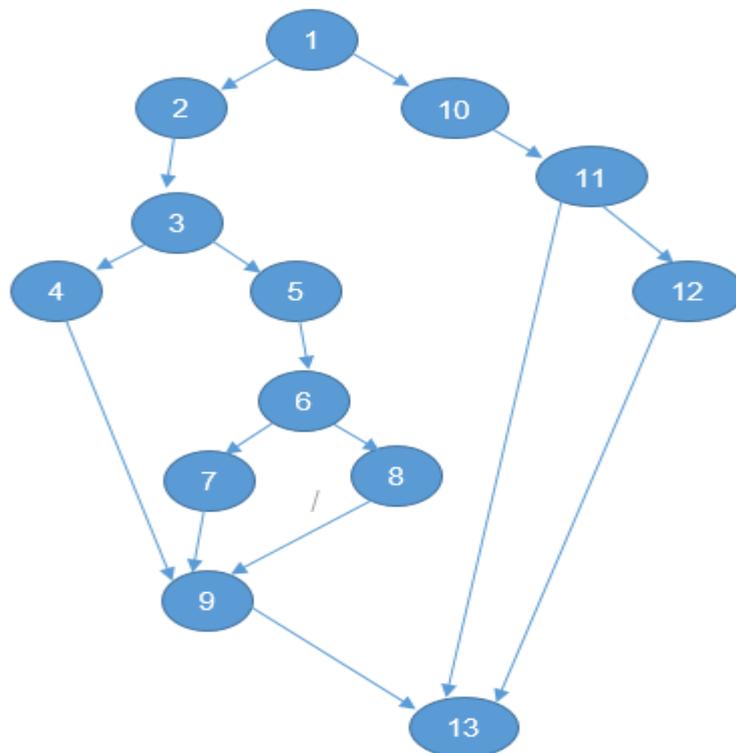


Figura 30: Representación del grafo de flujo de camino básico del método parse.

A partir del código fuente que representa la función parse, la cual permite determinar a través de la cadena de entrada hacia cuál espacio de color se quiere transformar. Se diseñó el grafo de flujo de camino básico como se muestra en la Figura 30, luego se procede a calcular la complejidad ciclomática $V(G)$, la cual define el número de caminos independientes del conjunto básico de un programa para asegurar que se ejecute cada sentencia al menos una vez.

$$V(G) = \text{Aristas}(A) - \text{Nodos}(N) + 2 \quad (34)$$

$$V(G) = 16 - 13 + 2$$

$$V(G) = 5$$

$$V(G) = 1 + \text{nodos predicados} \quad (35)$$

$$V(G) = 1 + 4$$

$$V(G) = 5$$

$$V(G) = \text{cantidad de regiones del grafo} \quad (36)$$

$$V(G) = 5$$

El valor $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 5 caminos:

Caminobásico1:1-2- 3- 4- 9- 13

Caminobásico2:1-2-3-5-6-7-9-13

Caminobásico3:1-2-3-5-6-8-9-13

Caminobásico4:1-10-11-12-13

Caminobásico5:1-10-11-13

Cada camino independiente es un caso de prueba a realizar, de forma que se garantiza que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. En el caso anterior se calcularon cinco caminos básicos, por tanto surge la necesidad de hacer igual número de casos de prueba. A continuación se muestra uno de los casos de pruebas realizados.

Entradas	Cualquier cadena de caracteres
Resultados esperados	Se asigna espacio de color de destino, pues se asume que el de origen es RGB
Condiciones	El espacio de origen en la cadena debe estar vacío

Tabla 11 Caso de prueba para el camino básico 5.

3.4.2 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las Historias de Usuarios, en cada ciclo de la iteración del desarrollo. Las pruebas de aceptación son de gran importancia, dado que miden el grado de satisfacción del cliente con el producto desarrollado [46].

Por lo tanto, son los clientes los responsables de verificar que los datos de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, son ellos los encargados de indicar el orden de resolución de los fallos [46]. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de dichas pruebas, de manera que todo el equipo esté al tanto de esta información. A continuación se presenta el diseño de caso de prueba de la HU3.

Historia de usuario: 3
Nombre: Conversión de RGB a CMY.
Descripción: Realiza la conversión del espacio RGB al espacio CMY.
Entrada/Pasos de Ejecución:
<ul style="list-style-type: none"> • Entrada: imagen RGB. • Se separa cada canal de la imagen • Se normalizan los resultados • Se resta uno a el valor de intensidad
Resultado esperado: se muestran los canales del espacio de color al cual se transformó (C, M, Y).
Evaluación de la prueba: prueba satisfactoria.

Tabla 12 Caso de prueba HU 3.

Por cada HU se realizó una prueba de aceptación. Fueron detectadas un total de 7 no conformidades en las 3 iteraciones realizadas. En la primera iteración se detectaron 2 no conformidades. En una segunda iteración se detectaron 5 y en la tercera iteración ninguna no conformidad. Las no conformidades fueron solucionadas en cada iteración, quedando el producto listo para la aceptación del cliente. A continuación se representan en la Figura 31 las no conformidades por iteraciones.

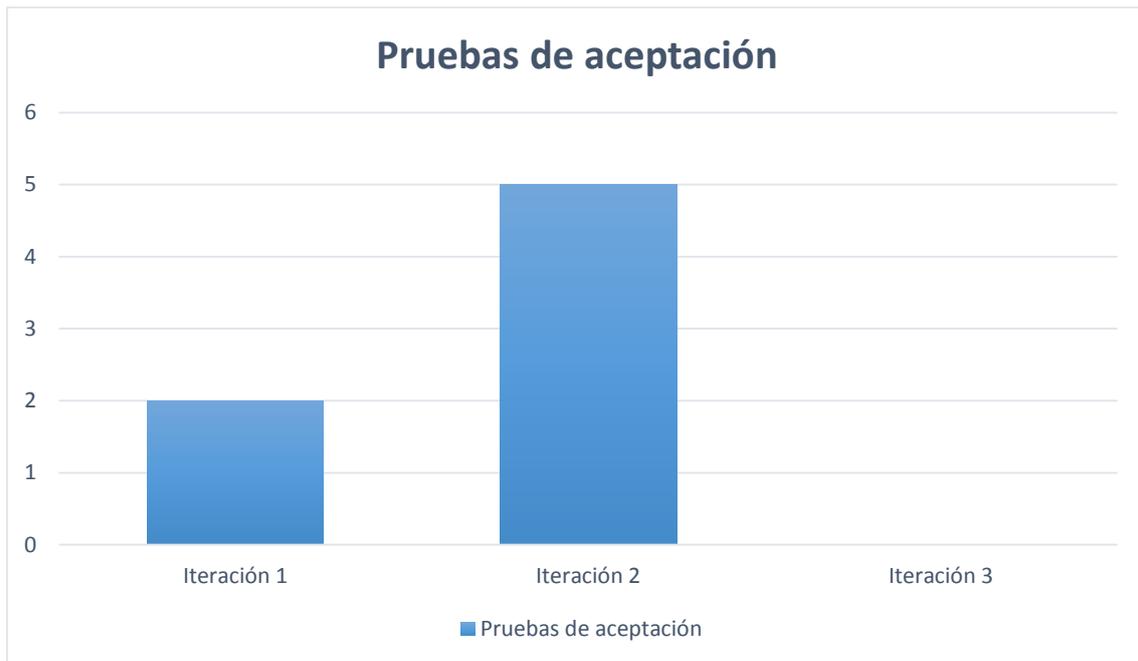


Figura 31: Tabla de no conformidades.

3.4 Conclusiones del capítulo

En el desarrollo del capítulo se abordaron los temas referentes a la implementación de la solución y la estrategia de pruebas a seguir. Se describieron las tareas de desarrollo para darle solución a las HU y se llevaron a cabo las pruebas realizadas a la solución, mostrando resultados satisfactorios que aseguran la funcionalidad del código y que el módulo quedó libre de no conformidades. También se realizó la validación del sistema haciendo uso las métricas PSNR, SNR, EMC y SSIM, además del cálculo del tiempo de ejecución, lo cual permitió demostrar que el espacio de color HSB es el más adecuado para el tratamiento de este conjunto de imágenes que son de interés para el grupo de investigación.

Conclusiones generales

Al finalizar el presente trabajo de diploma se arribó a las siguientes conclusiones:

- El espacio de color RGB presente en las imágenes provenientes de la lámpara de hendidura no es eficiente para el tratado de las mismas debido a que no presenta componentes tales como luminosidad/intensidad.
- Se implementaron las transformaciones hacia los espacios de color (CMY, YCbCr, CIEL*a*b*, HSB, HSL) y se definieron las métricas PSNR, SNR, EMC y SSIM además del cálculo del tiempo de ejecución como parámetros a medir para la elección del espacio de color más adecuado para la manipulación de las imágenes de interés para el grupo de investigación.
- Se comprobó, con los parámetros definidos, que el espacio de color más adecuado para la manipulación de las imágenes provenientes de la lámpara de hendidura en cuanto a eliminación de ruido, es el HSB.
- Como valor agregado de la investigación se realizó un módulo para la transformación del espacio de color de las imágenes hacia los utilizados por el grupo AIRI.

Referencias bibliográficas

- 1)Guerrero Úbeda, L. (2011). Separación de agregados en imágenes de microscopíacelular. Centro de Estudios de Electrónica y Tecnologías de la Información (CEETI). Las Villas, UCLV. Tesis para optar por el grado de Ingeniero Biomédico.
- 2) Alvarez Cancio, M. C. d. I. O., Reyder;Hernández López Iván (2016). "Aplicación de un sistema basado en casos para la identificación de opacidad en la cápsula posterior mediante imágenes del pentacam. ." Revista Cubana de Informática Médica 8(1): 545-558.
- 3)Hernández López, I., J. R. Hernández Silva, et al. (2010). "Estrategias de prevención de la opacidad de la cápsula posterior." Revista Cubana de Oftalmología 23: 608-623.
- 4)Acosta, R., L. Hoffmeister, et al. (2006). "Revisión sistemática de estudios poblacionales de prevalencia de catarata." Archivos de la Sociedad Española de oftalmología 81(9): 509-516.
- 5)González, R. R., Woods (2002). Digital Image Processing, Prentice Hall.
- 6)Chen, Y., P. Hao, et al. (2004). Optimal transform in perceptually uniform color space and its application in image coding. International Conference Image Analysis and Recognition, Springer.
- 7)Shih, P. and C. Liu (2005). "Comparative assessment of content-based face image retrieval in different color spaces." International Journal of Pattern Recognition and Artificial Intelligence 19(07): 873-893.
- 8)Cui, M., J. Hu, et al. (2010). "Color-to-gray conversion using ISOMAP." The Visual Computer 26(11): 1349-1360.
- 9)Phansalkar, N., S. More, et al. (2011). Adaptive local thresholding for detection of nuclei in diversity stained cytology images. Communications and Signal Processing (ICCSP), 2011 International Conference on, IEEE.
- 10)Wu, J., X. Shen, et al. (2012). "Interactive two-scale color-to-gray." The Visual Computer 28(6-8): 723-731.
- 11)Plataniotis, K. and A. N. Venetsanopoulos (2013). Color image processing and applications, Springer Science & Business Media.

- 12)Khairunniza-Bejo, S. and S. Kamarudin (2011). Chokanan Mango Sweetness Determination Using HSB Color Space. Computational Intelligence, Modelling and Simulation (CIMSIM), 2011 Third International Conference on, IEEE.
- 13)Braun, K. M. and R. L. de Queiroz (2005). Color to gray and back: Color embedding into textured gray images. Color and Imaging Conference, Society for Imaging Science and Technology.
- 14)Lee, T.-H., B.-K. Kim, et al. (2010). "Converting color images to grayscale images by reducing dimensions." Optical Engineering 49(5): 057006-057006-7.
- 15)Kaur, A. and B. Kranthi (2012). "Comparison between YCbCr color space and CIE Lab color space for skin color segmentation." IJAIS 3(4): 30-33.
- 16)Abbas, N. and D. Mohamad (2013). "Microscopic RGB color images enhancement for blood cells segmentation in YCBCR color space for k-means clustering." Journal of Theoretical and Applied Information Technology 55(1): 117-125.
- 17)Aimi Salihah, A.-N., M. Yusoff, et al. (2013). "Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering."
- 18)HARITHA, M. and R. S. REDDY (2013). COLOR CONVERSION AND WATER SHED SEGMENTATION FOR RGB IMAGES. International Conference on Electrical and Electronics Engineering.
- 19)Orozco-Morales, R. (2016). "Image Color Dimension Reduction. A comparative study of state-of-the-art methods."
- 20)Peguero Núñez, P. D. (2013). Realce y Restauración de imagen, Laboratorio de Procesamiento digital de imágenes.
- 21)Ramírez, B. E. (2006). Procesamiento Digital de Imágenes. Curso de Procesamiento Digital de Imágenes en la Universidad Autónoma de México. A. d. c. i. p. e. D. B. E. Ramírez. México.
- 22)Gaurav, S. (2003). Digital Color Imaging. New York, CRC PRESS.
- 23)Acharya, T. K. R., Ajoy (2005). Image Processing Principles and Applications Tempe, Arizona JOHN WILEY & SONS, MC.
- 24)González, R. C. W., P. (1996) Procesamiento Digital de Imágenes. Tema 3 y 4.pp. 89-269 Volume, pp. 89-269 DOI:

- 25)Bautista Cuellar, R. V. (2005). "Matlab y el diseño de filtros digitales. ." Revista digital "Investigación y Educación" 2(19): 1-23.
- 26)Romo González, A. E. (2005). Reconocimiento de imágenes mediante redes neuronales artificiales aplicado a la iridología. . Centro de Investigación y Estudios Avanzados del I.P.N. Departamento de Ingeniería Eléctrica. . México, IPN. Tesis para obtener el grado de Maestro en Ciencias en Ingeniería Eléctrica con opción en Computación.
- 27)Jain, A. K. (1989). "Fundamentos del procesamiento de imágenes digitales." Prentice Hall.
- 28)A.C. Botik, T. S. H. D. C. M. (1983). "A generalization of order statistics." IEEE Trans. Acoust., Speech, Signal Processing ASSP-31: 1342-1349.
- 29)Kassam, Y. H. L. a. S. A. (1985). "Generalized median filterin and related non-linear filtering techniques." IEEE Trans. Acoust., Speech, Signal Processing vol. ASSP-33: 672-683.
- 30)Barner, R. C. H. a. K. E. (1994). "Rank Conditioner Rank Selection Filters for Signal Restoration." IEEE Trans. Image Processing 3(2): 192-206.
- 31)Park, Y. H. L. a. S. Y. (1985). A frequency selective nonlinear filter based on a generalized median filter. Proc. Conf. Inform. Sci. Syst, (Baltimore, MD).
- 32)Lee, S. Y. P. a. Y. H. (1989). "Double Smoothing of Images Using Median and Weiner Filters." IEEE Trans. Acoust., Speech, Signal Processing 37(6): 943-946.
- 33)Kassam, P. P. G. a. S. A. (1991). "Design and performace of combination filters for signal restoration." IEEE Trans. Acoust., Speech, Signal Processing 39: 1524-1540.
- 34)Papoulis, A. (1984). Papoulis, Probability, Random variables and stochastic Processes, 2nd Ed. McGraw-Hill.
- 35)López, C. P. (2002). "Matlab y sus aplicaciones en las Ciencias y la Ingeniería."
- 36)Fiter, E. L. (2012). Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesado digital de imágenes . Madrid, Escuela Universitaria de Ingeniería.
- 37)Cancio, M. A. (2014). "PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata."
- 38)Sánchez Mendoza, M. A. (2004). "Metodologías del desarrollo de software."

- 39)Juristo, N., Moreno, A. M., & Vegas, S. (2006). "Técnicas de evaluación de software."
- 40)Joskowicz, J. (2008). "Reglas y prácticas en eXtreme Programming."
- 41)Bustamante, D., & Rodriguez, J. (2014). "Metodología Actual:Metodología XP."
- 42)Beck, K., Ed. (2002). Una explicación de la programación extrema: aceptar el cambio., Addison Wesley.
- 43)Casas, S., & Reinaga, H. (2009). " Aspectos tempranos : un enfoque basado en Tarjetas CRC. Argentina."
- 44)BECK, Kent. (1999). Extreme Programming Explained.
- 45)Malfará, Dayvis, y otros. (2006). Testing en eXtreme Programming.
- 46)Pressman, R. (2005). Ingeniería del Software un enfoque práctico.
- 47)Wang, Zhou, Alan C Bovik, Hamid R Sheikh & Eero P Simoncelli. (2004). Image quality assess- ment: from error visibility to structural similarity. IEEE transactions on image proces- sing, tomo 13(4):600–612.
- 48)JPEG (1983). Standard compression of digital images, ITU.
- 49) Coronado Martín, J. Á. (2012). Influencia de los productos hidrofugantes en las propiedades superficiales de las cerámicas porosas utilizadas en cerramientos exteriores cara vista. Madrid Universidad Politécnica 304.
- 50) Lungu, M. K. J. (2013). "On planning an evaluation of the impact of identifier names on the readability and quality of smalltalk programs." 2nd International Workshop on User Evaluations for Software Engineering Researchers (USER) vol. 00: pp. 13-15.
- 51) Osio, J., Rapallini, J., Quijano, A. A., & Ocampo, J. (2010). Implementación de un Algoritmo para procesamiento de imágenes en una FPGA. In *Congreso de Microelectrónica Aplicada 2010* (pp. 38-42).
- 52) Quarteroni, A., Sacco, R., & Saleri, F. (2010). Numerical mathematics (Vol. 37). Springer Science & Business Media.
- 53) SAHOO, A. & PRATAP, M. (2009). Fuzzy Weighted Adaptive Linear Filter for Color Image Restoration Using Morphological Detectors, International Journal on Computer Science and Engineering Vol.1 (3), 009, 217-221.
- 54) Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2003). Metodología de la investigación. La Habana: Editorial Félix Varela, 2.

Glosario de Términos

Mezcla aditiva del color

La mezcla o síntesis aditiva es un sistema de formación del color en el que los colores se consiguen sumando luces con distintas longitudes de onda, es decir, en el que la suma de luces forma el color. Los tres colores primarios de la síntesis aditiva son rojo, verde y azul. La mezcla aditiva es la base de dispositivos como los monitores de ordenador, los televisores y, en general, todos aquellos que forman imágenes usando luces.

Croma

Grado de diferencia existente entre un color y un gris de su misma luminosidad y claridad, que se corresponde con la saturación del color percibido. También llamado cromaticidad.

Anexos

Tabla 13 HU "Cargar imagen".

Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Cargar imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Alto	Puntos reales: 1 semanas
Programador responsable: Selma Hernández Romero	
Descripción: se carga la imagen en disco seleccionada por el usuario.	
Observaciones:	

Tabla 14 HU "Obtener valores de la imagen de entrada".

Historia de Usuario	
Número: 2	Nombre de la Historia de Usuario: Obtener valores de la imagen de entrada.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas

Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Selma Hernández Romero	
Descripción: Obtener componentes de imagen de entrada: permite extraer de la imagen de entrada los componentes RGB para su procesamiento posterior.	
Observaciones:	

Tabla 15 HU "Conversión de RGB a CMY".

Historia de Usuario	
Número: 3	Nombre de la Historia de Usuario: Conversión de RGB a CMY
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Selma Hernández Romero	
Descripción: RGB-CMY: Realiza la conversión del espacio RGB al espacio CMY.	
Observaciones:	

*Tabla 16 HU "Conversión de RGB a CIEL*a*b**".*

Historia de Usuario	
Número: 4	Nombre de la Historia de Usuario: Conversión de RGB a CIEL *a*b*
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Selma Hernández Romero	
Descripción:	
RGB-CIEL *a*b*: Realiza la conversión del espacio RGB al espacio CIEL *a*b*.	
Observaciones:	

Tabla 17 HU "Conversión de RGB a HSL".

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Conversión de RGB a HSL
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Selma Hernández Romero	

Descripción: RGB-HSL: Realiza la conversión del espacio RGB al espacio HSL.
Observaciones:

Tabla 18 HU "Conversión de RGB a HSB".

Historia de Usuario	
Número: 6	Nombre de la Historia de Usuario: Conversión de RGB a HSB
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Alto	Puntos reales: 3 semanas
Programador responsable: Selma Hernández Romero	
Descripción: RGB-HSB: Realiza la conversión del espacio RGB al espacio HSB.	
Observaciones:	

Tabla 19 HU "Conversión de RGB a YCbCr".

Historia de Usuario	
Número: 7	Nombre de la Historia de Usuario: Conversión de RGB a YCbCr
Modificación a la Historia de Usuario: Ninguna	

Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Selma Hernández Romero	
Descripción: RGB-YCbCr: Realiza la conversión del espacio RGB al espacio YCbCr.	
Observaciones:	

Tabla 20 HU “Visualizar imagen de cada uno de los componentes del espacio de color”.

Historia de Usuario	
Número: 8	Nombre de la Historia de Usuario: Visualizar imagen de cada uno de los componentes del espacio de color.
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 2
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Selma Hernández Romero	
Descripción: Visualiza la imagen de cada uno de los canales del espacio de color seleccionado.	
Observaciones:	

Tareas de ingeniería

Iteración 1

Tarea de ingeniería	
Número de tarea: 1	Historia de Usuario (No.1): Cargar imagen
Nombre de tarea: Cargar imagen	
Tipo de tarea: Desarrollo	Puntos estimados: 1 semana
Fecha de inicio: 19/12/2016	Fecha de fin: 23/12/2016
Programador responsable: Selma Hernández Romero	
Descripción: se carga la imagen en disco seleccionada por el usuario.	

Tabla 21 Tarea de ingeniería 1.

Tarea de ingeniería	
Número de tarea: 2	Historia de Usuario (No.2): Obtener valores de la imagen de entrada.
Nombre de tarea: Obtener valores de la imagen de entrada.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semana
Fecha de inicio: 02/01/2017	Fecha de fin: 14/01/2017
Programador responsable: Selma Hernández Romero	
Descripción: permite extraer de la imagen de entrada los componentes RGB para su procesamiento posterior.	

Tabla 22 Tarea de ingeniería 2.

Iteración 2

Tarea de ingeniería	
Número de tarea: 3	Historia de Usuario (No.3): Conversión de RGB a CMY

Nombre de tarea: Convertir de RGB a CMY	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semana
Fecha de inicio: 23/01/2017	Fecha de fin: 04/02/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB a CMY	

Tabla 23 Tarea de ingeniería 3.

Tarea de ingeniería	
Número de tarea: 4	Historia de Usuario (No.4): Conversión de RGB a CIEL*a*b*
Nombre de tarea: Convertir de RGB a CIEL*a*b*	
Tipo de tarea: Desarrollo	Puntos estimados: 3 semana
Fecha de inicio: 06/02/2017	Fecha de fin: 25/02/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB al espacio CIEL*a*b*.	

Tabla 24 Tarea de ingeniería 4.

Tarea de ingeniería	
Número de tarea: 5	Historia de Usuario (No.5): Conversión de RGB a HSL
Nombre de tarea: Convertir de RGB a HSL	
Tipo de tarea: Desarrollo	Puntos estimados: 3 semana
Fecha de inicio: 25/03/2017	Fecha de fin: 15/04/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB al espacio HSL.	

Tabla 25 Tarea de ingeniería 5.

Tarea de ingeniería	
Número de tarea: 6	Historia de Usuario (No.6): Conversión de RGB a HSB
Nombre de tarea: Convertir de RGB a HSB	
Tipo de tarea: Desarrollo	Puntos estimados: 3 semana
Fecha de inicio: 17/04/2017	Fecha de fin: 6/05/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB al espacio HSB.	

Tabla 26 Tarea de ingeniería 6.

Tarea de ingeniería	
Número de tarea: 7	Historia de Usuario (No.7): Conversión de RGB a YCbCr
Nombre de tarea: Convertir de RGB a YCbCr	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semana
Fecha de inicio: 08/05/2017	Fecha de fin: 20/05/2017
Programador responsable: Selma Hernández Romero	
Descripción: se transforma a través del modelo matemático de conversión del espacio de color RGB al espacio YCbCr.	

Tabla 27 Tarea de ingeniería 7.

Tarea de ingeniería	
Número de tarea: 8	Historia de Usuario (No.8): Visualizar imagen de cada uno de los componentes del espacio de color.
Nombre de tarea: Visualizar imagen de cada uno de los componentes del espacio de color.	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semana
Fecha de inicio: 20/05/2017	Fecha de fin: 29/05/2017
Programador responsable: Selma Hernández Romero	

Descripción: Visualiza el imagen de cada uno de los canales del espacio de color seleccionado.

Tabla 28 Tarea de ingeniería 8.