

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3



Software para la manipulación directa de las imágenes provenientes de la lámpara de hendidura y su almacenamiento en diversos formatos de compresión.

Trabajo final presentado en opción al título de Ingeniero en Ciencias Informáticas

Autores: Alexis De Armas Francisco

Daimarys Laureiro González

Tutor: Ing. Michel Alvarez Cancio

La Habana, junio de 2018



*"En la tierra hace falta personas que trabajen más y critiquen menos,
que construyan más y destruyan menos, que esperen recibir menos y
den más, que digan mejor ahora que mañana".*

Che

Declaración jurada de autoría

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Autorizamos a dicho centro para que haga el uso que estime pertinente con este trabajo.

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

Alexis De Armas Francisco

Autor

Daimarys Laureiro González

Autora

Ing. Michel Alvarez Cancio

Tutor

Dedicatoria

A mis padres, Alexis y Xiomara, por el apoyo incondicional.

Alexis.

A mi abuela Cristina, todo este tiempo ha sido por ella. Te adoro.

A mi padre, que, desde cualquier lugar, sé que está muy orgulloso. Te quiero.

Daimarys.

Agradecimientos

A mis padres Alexis y Xiomara, por ser mis guías. Gracias por educarme, cuidarme y protegerme en cada momento y sobre todo por ser mis grandes ejemplos a seguir, los quiero.

A mi hermana, gracias por apoyarme.

A Ariandi, por acompañarme cada noche a tesiar, por ser mi cómplice, mi amigo.

A mi compañera de tesis, por estar siempre al pendiente y compartir juntos cada momento para poder cumplir con nuestro objetivo.

A Eileén, por soportar cada momento de locura de mi compañera de tesis y evitar que se acercara a mí.

A mi tutor por la confianza, siendo guía y fuerza durante todo el proceso.

A mi oponente, gracias por estar al pendiente del proceso.

A todos muchas gracias.

Alexis

Agradecimientos

A mi madre, la verdad no tengo palabras para expresar mis sentimientos hacia ella, es la única persona del mundo, siempre incondicional, me ha hecho reír, me ha secado las lágrimas, me ha visto triunfar, me ha regañado, pero siempre a mi lado. Cuando las demás puertas cierran, las de ellas siempre están abiertas, siempre ahí para mí. Es una bendición que me ha dado Dios, una verdadera amiga, ojalá y estuvieras a mi lado para siempre.

A mi hermano, que sé que debe estar pensando en mí en estos momentos, aunque estamos distanciados y vivamos discutiendo, sabes que te quiero con el alma y que en mí siempre puedes confiar.

A mi familia, mis tías, primos, primas, no los puedo mencionar a todos porque son muchos, pero miles de gracias por todos esos consejos que en su momento hicieron falta.

A Chichi, gracias por ser como una segunda madre para mí, gracias por el apoyo que me estás dando, ahora con todo lo que ha pasado y que en realidad es cuando más te necesito. Te quiero.

A mis niños grandes, Dayron y Jorgito, su tata los ama.

A la loca de mi tía Caridad, que con todo y su mal genio me adora y vive al pendiente mío.

A mi tía Mercedes, gracias por el apoyo y por tus cuidados.

A Guille, gracias por ser mi chofer, mi amigo y cuidar de mi mami en todo momento.

A la gente del barrio, Yaque, la China, Lisi, Nuria, gracias por preocuparse y estar a mi lado.

No por ser la última es menos importante, a mi abuela a esa viejuca que ha sabido acompañarme en cada etapa de mi vida, protegiéndome cada vez con más fuerza, ayudándome a ser cada día mejor, a ti miles de gracias, te quiero mucho.

A mi compañero de tesis, que, sin él, esto no hubiera sido posible. Gracias por escogerme para compartir este momento único de nuestras vidas.

A mi tutor por la confianza y el apoyo, que no solo supo guiarme, sino también me hizo sonreír cuando pasaba por tiempos de tensión y estrés.

Al tribunal y a mi oponente gracias por el apoyo.

A todos los profesores que contribuyeron a que este día llegara, en especial a Yorguy y Mailen.

A mis compañeros de aula, por soportar mis gritos, en especial a mi papi, Darlyn, Frank, Dayi, Ariandi, David, Raykof.

A mis amigos, y aquí me detengo porque tengo la dicha de poder decir que a mi lado tengo gente maravillosa que me apoya incondicionalmente. Muchas gracias por su complicidad, su respeto, su lealtad, su cariño y su confianza. Los quiero a todos.

¡A mis amigas locas!! Clau, Angi, Day, Ingrid, Grisell. Gracias por cada uno de esos momentos que pasé cerca de ustedes, las kierooooo

A Ei, que a pesar del poco tiempo que nos conocemos me has demostrado que puedo confiar plenamente en ti, gracias por la complicidad y los buenos consejos. Espero que con el paso del tiempo sigamos con esta linda amistad.

A mi titi, gracias por estresarme cada día con tus peleas, gracias por quererme siempre y por tus cariños. Te quiero.

A Elizabeth, por demostrarme que soy parte de su familia.

A Jose por adoptar una "hija boba", siempre aconsejándome y buscando la mejor solución para ayudarme cuando ando con el moño torcido. Miles de gracias

A una persona muy especial que ha sabido acogerme de una forma increíble, siempre aguantando mis malcriadeces y mis malhumores, apoyándose en cada momento y sin decir un solo día que no, a ti Alfre, Te quiero.

Ahora me detengo, aquí definitivamente las palabras no me fluyen, porque a pesar de estar nueva en el grupo, me han sabido acoger con amor y cariño, siempre incondicionales. Mi mamerto te quiero, gracias por escucharme y ser cómplice de mis trastadas. Dai y mi niño pequeño(Osvi), gracias por no dejarme sola en ninguna fiesta y cuidarme siempre. Miche y Luisi, gracias por tenerme como su malcriada. Yuyi, Lixan, David y mi Fer, los adoro, gracias por soportarme. Ive por pelearme tanto, siempre con su razón, pero bueno. Ori, Abelito y los flacos, gracias por tenerme presente.

A todas las personas que hicieron posible este momento gracias.

A todos muchas gracias.

Daimarys

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.

Uno de los procedimientos para el estudio y diagnóstico de esta anomalía es mediante las imágenes provenientes de la lámpara de hendidura. El software empleado en nuestro país para la manipulación de estas imágenes es antiguo, por lo que se hace necesario la creación de un software que satisfaga las necesidades de los especialistas, permita mejorar las imágenes que obtenga, guardar la imagen en un formato con compresión sin pérdida, extraer secuencias de imágenes y almacenarlas en forma de secuencia o video.

Para la validación de este software se aplicaron pruebas de caja blanca y caja negra, así como pruebas utilizando métricas de tamaño operacional de clase, demostrando la correcta realización del mismo y la satisfacción del cliente con el producto.

Palabras claves: opacidad de la cápsula posterior, catarata, complicación postoperatoria, lámpara de hendidura, compresión sin pérdida, secuencias de imágenes.

Índice General

Pensamiento.....	1
Declaración jurada de autoría	2
Dedicatoria	3
Agradecimientos.....	4
Agradecimientos.....	5
Resumen	7
Índice General	8
Índice de figuras	11
Índice de tablas	12
Introducción	14
Capítulo 1: Fundamentación teórica	18
1.1 Imagen Digital.....	18
1.1.1 Imagen Oblicua.....	19
1.1.2 Imagen en retroiluminación	19
1.1.3 Imágenes DICOM	20
1.2 Espacios de color	21
1.2.1 Espacio de color RGB	22
1.2.2 Espacio de color CMY	22
1.2.3 Espacio de color HSV.....	23
1.2.4 Espacio de color HSL	24
1.3 Calidad de la imagen.....	24
1.4 Filtrado de imágenes	25
1.5 Formatos de compresión.....	26
1.6 Lámpara de hendidura	27
1.6.1 Adaptador de cámara para la lámpara de hendidura	28
1.7 Comunicación	29
1.8 Conclusiones del capítulo	30
Capítulo 2: Herramientas y tecnologías.....	31
2.1 Lenguaje de programación.....	31
2.2 Entorno integrado de desarrollo.....	31
2.3 Herramienta de prueba	32
2.4 Herramienta para el modelado.....	32
2.5 Herramientas de análisis de datos USB	32
2.5.1 USB Monitor	33
2.5.2 USB Analyzer 4.0	33

2.5.3 Wireshark.....	34
2.5.4 Wireshark con USBPcap	34
2.6 Herramientas de análisis de datos Serial.....	34
2.6.1 Free serial port monitor	35
2.6.2 Serial monitor.....	35
2.7 Herramientas para el trabajo con DICOM.....	35
2.7.1 Biblioteca Pixelmed	35
2.7.2 Biblioteca ImageJ	36
2.8 Justificación de la selección de las herramientas de análisis.....	36
2.9 Metodología de desarrollo.....	36
2.10 Conclusiones del capítulo	37
Capítulo 3: Análisis y diseño de la solución.....	39
3.1 Descripción de la solución.....	39
3.2 Fase de planeación	42
3.2.1 Especificación de los requisitos	42
3.2.2 Descripción de historias de usuarios.....	43
3.2.3 Fase de Planificación de la Entrega.....	43
3.2.4 Plan de iteraciones	44
3.3 Diseño de la solución	45
3.3.1 Tarjetas CRC	45
3.3.2 Estándares de codificación.....	46
3.3.3 Patrones de diseño utilizados.....	47
3.4 Conclusiones del capítulo	49
Capítulo 4: Resultados y validación.....	50
4.1 Fase de implementación	50
4.1.1 Iteración 1	50
4.1.2 Iteración 2.....	52
4.1.3 Iteración 3.....	54
4.2 Interfaz del sistema	56
4.3 Validación del sistema.....	56
4.3.1 Pruebas de caja blanca	57
4.3.2 Pruebas de caja negra	58
4.3.3 Pruebas utilizando métricas de tamaño operacional de clase.....	59
4.4 Conclusiones del capítulo	61
Conclusiones generales.....	62
Recomendaciones.....	63
Referencias Bibliográficas.....	64
Anexos.....	68
Historias de usuarios	68

Índice de figuras

Ilustración 1: Espectro electromagnético. Tomada de Imágenes Google.....	18
Ilustración 2: Imagen oblicua tomada con la lámpara de hendidura.	19
Ilustración 3: Imagen en retroiluminación tomado de la lámpara de hendidura.....	20
Ilustración 4: Espacios de color. Tomada de Imágenes Google.	21
Ilustración 5: Subespacios de colores. Tomada de Imágenes Google.	22
Ilustración 6: Subespacio de color CMY. Tomada de Imágenes Google.....	23
Ilustración 7: Subespacio de color HSV. Tomada de Imágenes Google.	23
Ilustración 8: Subespacio de color HSL. Tomada de Imágenes Google.....	24
Ilustración 9: Imágenes filtradas.	26
Ilustración 10: Imagen con formato de compresión con pérdida y sin pérdida respectivamente.	27
Ilustración 11: Lámpara de hendidura. Tomado de imágenes Google.	28
Ilustración 12: Adaptador para la lámpara de hendidura. Tomado de imágenes Google.....	29
Ilustración 13: Flujo de datos de la comunicación USB.....	30
Ilustración 14: Diagrama de flujo de proceso de la interfaz Capturar.	39
Ilustración 15: Diagrama de flujo de proceso de la interfaz Principal.....	41
Ilustración 16: Interfaz principal.	56
Ilustración 17: Interfaz de captura.....	56

Índice de tablas

Tabla 1: Historia de usuario 1 (HU1).....	43
Tabla 2: Estimación de tiempo por historia de usuario.....	44
Tabla 3: Estimación por tiempo de las historias de usuarios según las iteraciones.	45
Tabla 4: Tarjetas CRC de la clase MyImage.	46
Tabla 5: Tiempo de implementación de las HU de la iteración #1	50
Tabla 6: Tabla de ingeniería de la funcionalidad Cargar secuencia de imágenes.....	51
Tabla 7: Tabla de ingeniería de la funcionalidad Listar secuencia de imágenes.....	51
Tabla 8: Tabla de ingeniería de la funcionalidad Reproducir secuencia de imágenes.....	51
Tabla 9: Tabla de ingeniería de la funcionalidad Añadir metadatos.	52
Tabla 10: Tabla de ingeniería de la funcionalidad Visualizar metadatos.	52
Tabla 11: Tiempo de implementación de las HU de la iteración #2.	52
Tabla 12: Tabla de ingeniería de la funcionalidad Zoom de la imagen.....	53
Tabla 13: Tabla de ingeniería de la funcionalidad Ajuste de brillo y contraste.	53
Tabla 14: Tabla de ingeniería de la funcionalidad de Manipulación de valores RGB.....	53
Tabla 15: Tabla de ingeniería de la funcionalidad de Filtros de realce.....	54
Tabla 16: Tabla de ingeniería de la funcionalidad de Filtros suavizado.....	54
Tabla 17: Tiempo de implementación de las HU de la iteración #3.	54
Tabla 18: Tabla de ingeniería de la funcionalidad de Captura de tráfico USB.	55
Tabla 19: Tabla de ingeniería de la funcionalidad de Extracción de imágenes del tráfico USB.....	55
Tabla 20: Tabla de ingeniería de la funcionalidad de Grabar imágenes de la lámpara de hendidura. .	55
Tabla 21: Tabla de ingeniería de la funcionalidad de Exportar imágenes en formato DICOM.....	56
Tabla 22: Tabla de caso de prueba de caja blanca.....	58
Tabla 23: Gráfico de barras que muestra las no conformidades por iteraciones.....	59
Tabla 24: Gráfico de pastel que muestra la métrica Responsabilidad.	60
Tabla 25: Gráfico de pastel que muestra la métrica Complejidad.....	60
Tabla 26: Gráfico de pastel que muestra la métrica Reutilización.	61
Tabla 27: Historia de usuario 2 (HU2).....	68
Tabla 28: Historia de usuario 3 (HU3).....	69
Tabla 29: Historia de usuario 4 (HU4).....	69
Tabla 30: Historia de usuario 5 (HU5).....	70
Tabla 31: Historia de usuario 6 (HU6).....	70
Tabla 32: Historia de usuario 7 (HU7).....	71
Tabla 33: Historia de usuario 8 (HU8).....	71
Tabla 34: Historia de usuario 9 (HU9).....	72
Tabla 35: Historia de usuario 10 (HU10)	72
Tabla 36: Historia de usuario 11 (HU11)	73
Tabla 37: Historia de usuario 12 (HU12)	73
Tabla 38: Historia de usuario 13 (HU13)	74
Tabla 39: Historia de usuario 14 (HU14)	74

Tabla 40: Tarjeta CRC de la clase FormatConverter.	75
Tabla 41: Tarjeta CRC de la clase CreateADicomFile.	75
Tabla 42: Tarjeta CRC de la clase VentanaPrincipal.	75
Tabla 43: Tarjeta CRC de la clase MostrarMetadata.	75
Tabla 44: Tarjeta CRC de la clase VentanaCaptura.	76
Tabla 45: Tarjeta CRC de la clase AdicionarMetadata.....	76

Introducción

En la actualidad miles de personas de todo el mundo padecen de la enfermedad de cataratas, sufriendo como consecuencia la pérdida total o parcial de la visión [1] [2] [3]. La catarata es la opacidad en el cristalino del ojo, trayendo como consecuencia que la luz se disperse dentro del mismo y no sea posible enfocar en la retina, lo que provoca en el paciente la visualización de imágenes difusas [1].

Las cataratas generan problemas con la apreciación de cambios de colores y cambios de contrastes, siendo esta la causa más común de ceguera tratable con cirugía. 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. Esta enfermedad tiende a aparecer con mayor frecuencia en pacientes mayores de 50 años de edad. [2] [4]

Tras una intervención quirúrgica el paciente puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito a largo plazo, pues en muchos de ellos el paciente puede presentar complicaciones postoperatorias. La complicación postoperatoria más frecuente a largo o mediano plazo es la opacidad de la cápsula posterior (OCP) [3] [5]. La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata, para garantizar una buena recuperación visual del mismo es imprescindible la transparencia de esta estructura [3] [6]. La OCP trae consigo deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas [3], pudiendo aparecer meses e incluso años después de realizada la cirugía, siendo actualmente uno de los aspectos más importantes en la cirugía de catarata de los tiempos modernos [1] [6].

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Grupo de Investigación AIRI (*Artificial Intelligence: Research and Innovation*), el cual tiene como principal resultado y una de sus líneas de investigación la creación del software PANDOC (Programa Analizador de Opacidad Capsular), el cual fue realizado en colaboración con el Instituto Cubano de Oftalmología (ICO) "Ramón Pando Ferrer".

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, minimizando el sesgo de observación entre un médico y otro.

Para esto utiliza las imágenes provenientes de la lámpara de hendidura¹, también llamada biomicroscopio. Es un dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras de la porción anterior del ojo. Consiste en un microscopio binocular dotado de una potente fuente de luz que permite ver en tres dimensiones y con una amplificación que oscila entre 6 y

¹ Dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras de la porción anterior del ojo.

40 aumentos [7], las estructuras del polo anterior del ojo: párpado, conjuntiva, córnea, iris, cristalino y cámara anterior.

El software que permite la comunicación con las lámparas de hendidura usadas en nuestro país es antiguo y tras un análisis con los desarrolladores de PANDOC y especialistas del ICO se le detectaron las siguientes deficiencias:

- No graba secuencia de imágenes o video.
- Usa formato de compresión con pérdida JPG.
- No utiliza filtros para el mejoramiento o restauración de las imágenes.
- No genera imágenes DICOM, que son las utilizadas en la comunidad médica para el envío y almacenamiento de las imágenes.
- No permite la conexión a dispositivos Android para el envío y recepción de datos a través del adaptador para smartphones.

Por otra parte, la precisión del resultado de la cuantificación de PANDOC está sujeta a los parámetros de calidad de la imagen, ya que en varias ocasiones se ve afectada por elementos como desenfoque, *blur*² y formato de compresión propios del ambiente y las tecnologías de donde fue tomada la misma.

Es por ello que surge como necesidad la obtención de imágenes con mayor calidad para el análisis por parte del software, que permita además mejorar las imágenes que obtenga, guardar la imagen en un formato con compresión sin pérdida, extraer secuencias de imágenes y almacenarlas en forma de secuencia o video.

Una vez culminada la etapa de análisis se dio paso a la próxima etapa de desarrollo donde se identifica el siguiente **problema a resolver**: ¿Cómo obtener las imágenes provenientes de la lámpara de hendidura, de manera que se mejore la calidad de las imágenes obtenidas y permita su almacenamiento en diversos formatos?

Teniendo como objeto de estudio **objeto de estudio**: Procesamiento de imágenes digitales.

Enmarcándose en el **campo de acción**: Procesamiento de imágenes biomédicas.

Con el **objetivo general**: Desarrollar un software para la manipulación directa de las imágenes provenientes de la lámpara de hendidura que permita mejorar la calidad de las imágenes obtenidas y su almacenamiento en diversos formatos.

Teniendo como referencia el problema a resolver y el objetivo general se plantea la siguiente **idea a defender**: Con el desarrollo de un software para la manipulación directa de imágenes provenientes de la lámpara de hendidura, se mejorará la calidad de las imágenes obtenidas y permitirá su almacenamiento en diversos formatos.

Para dar cumplimiento al objetivo propuesto se han derivado un conjunto de **objetivos específicos** orientados fundamentalmente a proveer los elementos necesarios para la implementación de la solución, los cuales se listan a continuación:

² Elementos adversos que deterioran la calidad visual de la imagen, como desenfoque y ruido entre otros.

- ✓ Registrar y organizar información científico-técnica sobre los métodos de comunicación y almacenamiento en las imágenes biomédicas y su respectiva aplicación en el caso particular de las imágenes provenientes de la lámpara de hendidura.
- ✓ Implementar algoritmos para la captura, modificación y conversión a distintos formatos de las imágenes provenientes de la lámpara de hendidura.
- ✓ Evaluar la propuesta de solución mediante pruebas unitarias y funcionales.

Los **métodos científicos** utilizados para validar metodológicamente la investigación son:

Métodos empíricos:

- ✓ Observación: Como método consiste en la percepción directa del objeto de investigación, es el instrumento universal del científico. La observación permite conocer la realidad mediante la percepción directa de los objetos y fenómenos. Este método fue utilizado para la observación y estudio de los diferentes algoritmos en su aplicación a diferentes muestras, identificando además cuáles eran los más adecuados para la obtención de los mejores resultados.

Métodos teóricos:

- ✓ Analítico-Sintético: Se utiliza con el objetivo de analizar las teorías, documentos e información referente al tema en cuestión, para luego extraer los elementos más importantes que se relacionan con las imágenes provenientes de la lámpara de hendidura a través de una comunicación USB.
- ✓ Histórico-Lógico: Permite estudiar y analizar la trayectoria y evolución de la comunicación con la lámpara de hendidura, para poder mejorar el nivel de calidad y almacenarlas en diversos formatos, y con estas experiencias se podrá tener un mayor entendimiento y comprensión de este tipo de sistemas, basándose en las experiencias existentes.
- ✓ Inductivo-Deductivo: Se utiliza con el objetivo de inducir una serie de conocimientos referentes a comunicación USB con la lámpara de hendidura, para luego arribar a razonamientos que conlleven a la deducción de conocimientos que pueden ser aplicables al problema en cuestión.

El presente trabajo de investigación está estructurado en 4 capítulos, de los cuales se brinda una breve descripción a continuación:

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.

Capítulo 2. Herramientas y tecnologías: Se describen las herramientas y lenguajes, así como la metodología de desarrollo de software.

Capítulo 3. Propuesta de solución: En este capítulo se realiza una descripción detallada de la solución propuesta con los aspectos definidos en la fundamentación teórica. Esta solución contiene los artefactos necesarios para el desarrollo del sistema a construir.

Capítulo 4. Resultados y validación: En este capítulo se muestra el conjunto de pruebas realizadas a la herramienta y los resultados que arrojaron las mismas. Además, se valida que el diseño

realizado cumpla con la calidad requerida y el sistema implementado satisface las necesidades del cliente.

Capítulo 1: Fundamentación teórica

En este capítulo se realiza un estudio sobre los elementos que fundamentan la base teórica conceptual para la manipulación directa de las imágenes provenientes de la lámpara de hendidura y su almacenamiento en diversos formatos.

1.1 Imagen digital

La luz que detecta el objetivo de la cámara llega hasta el sensor de imagen, formado por multitudes de receptores fotosensibles denominados "fotodiodos".

La luz incidente genera una pequeña señal eléctrica a cada receptor, que posteriormente, esta señal se transformará en datos digitales por el conversor ADC, como una serie de cadenas de números ceros y unos, denominados dígitos binarios. Estos números binarios (0,1), se representan como pequeños cuadrados, en forma de mosaico individual denominados píxeles.

Lo que el ojo humano percibe es solo una pequeña zona del espectro electromagnético conocida como "espectro visible". Pero con el avance de la tecnología existe en la actualidad un amplio rango del espectro electromagnético que se puede detectar y registrar en imágenes, para posteriormente analizar, transmitir o almacenar [7].

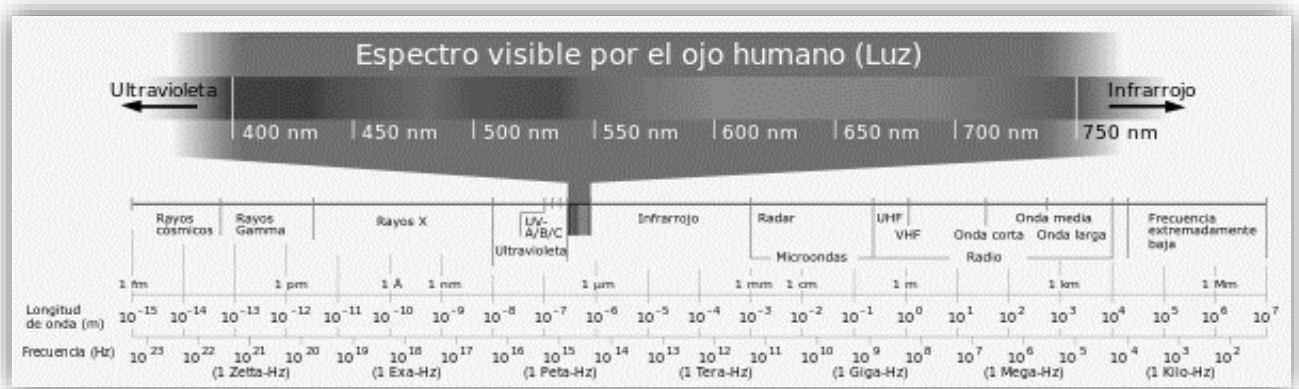


Ilustración 1: Espectro electromagnético. Tomada de Imágenes Google.

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 modelos consiste en que a partir de la combinación de tres colores primarios se puede obtener cualquier color de la naturaleza[8].

Las imágenes digitales se pueden modificar mediante filtros, añadir o suprimir elementos, modificar su tamaño, etc. y almacenarse en un dispositivo de grabación de datos.

1.1.1 Imagen oblicua

Imagen con un haz de luz que se proyecta como una franja de hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos y que se detecten e identifiquen las regiones que presentan opacidad capsular. Estas son obtenidas a través de la lámpara de hendidura, y presentan las siguientes características:

1. Se encuentran en un espacio de color RGB, por lo que está compuesta de los canales de Rojo, Verde y Azul.
2. Tienen como formato JPG, lo que significa que la imagen obtenida no es la misma que la deseada, esto es provocado por el algoritmo de compresión que utiliza dicho formato, que es un algoritmo de reducción con pérdida para disminuir el tamaño del archivo.
3. Tiene una profundidad de 24 bits (8 por cada uno de los canales del espacio de color), por lo que presenta variaciones de intensidad entre 0 y 255 colores.
4. Una resolución de 1024 x 768 píxeles.
5. Presenta una luz provocada por el mismo equipo que toma las imágenes que puede dificultar la detección de la opacidad (esto provoca irregularidad en las intensidades y que existan bordes que no estén bien definidos).

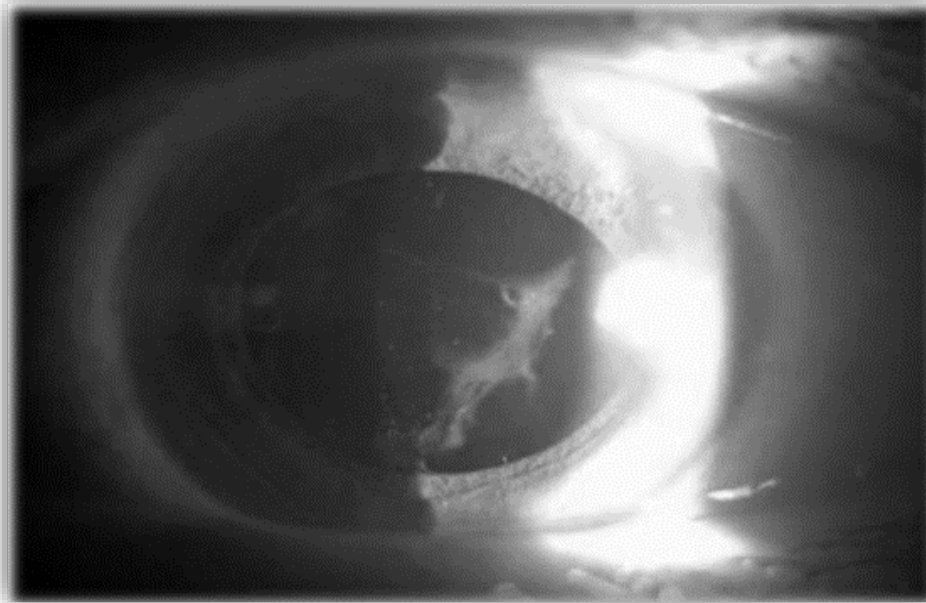


Ilustración 2: Imagen oblicua tomada con la lámpara de hendidura.

1.1.2 Imagen en retroiluminación

Estas son obtenidas a través de la lámpara de hendidura, las cuales presentan las siguientes características:

1. Se encuentran en un espacio de color RGB, por lo que está compuesta de los canales de Rojo, Verde y Azul.

2. Tienen como formato JPG, lo que significa que la imagen obtenida no es la misma que la deseada, esto es provocado por el algoritmo de compresión que utiliza dicho formato, que es un algoritmo de reducción con pérdida para disminuir el tamaño del archivo.
3. Tiene una profundidad de 24 bits (8 por cada uno de los canales del espacio de color), por lo que presenta variaciones de intensidad entre 0 y 255 colores.
4. Una resolución de 483 x 333 píxeles.
5. Se refleja la luz en los elementos profundos del globo ocular para examinar estructuras anteriores. Puede ser directa, reflejando la luz directamente sobre la estructura a examinar, o indirecta, donde la estructura a observar se ve contra un fondo oscuro.

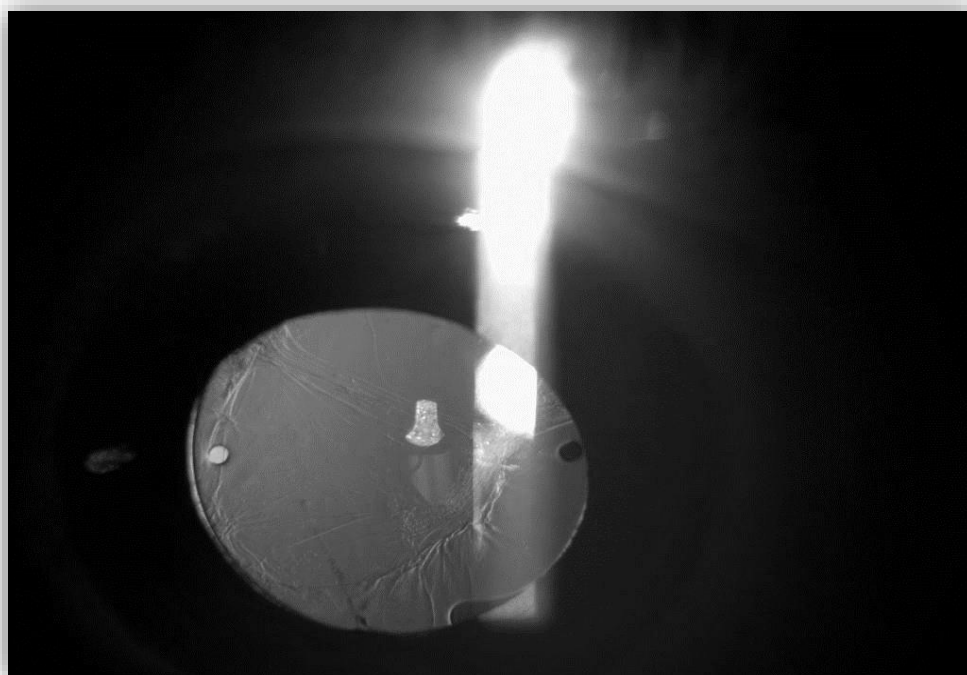


Ilustración 3: Imagen en retroiluminación tomado de la lámpara de hendidura.

1.1.3 Imágenes DICOM

DICOM (*Digital Imaging and Communication in Medicine*) es el estándar reconocido mundialmente para el intercambio de pruebas médicas, pensado para su manejo, visualización, almacenamiento, impresión y transmisión. Incluye la definición de un formato de fichero y de un protocolo de comunicación de red. El protocolo de comunicación es un protocolo de aplicación que usa TCP/IP para la comunicación entre sistemas. Los ficheros DICOM pueden intercambiarse entre dos entidades que tengan capacidad de recibir imágenes y datos de pacientes en formato DICOM [25].

DICOM permite la integración de escáneres, servidores, estaciones de trabajo, impresoras y hardware de red de múltiples proveedores dentro de un sistema de almacenamiento y comunicación de imágenes. Las diferentes máquinas, servidores y estaciones de trabajo tienen una declaración de conformidad DICOM (conformance statements) que establece claramente las clases DICOM que soportan. DICOM

ha sido adoptado ampliamente por hospitales y está haciendo incursión en pequeñas aplicaciones de oficinas de dentistas y de médicos.

1.2 Espacios de color

Un espacio de color es un rango de colores definido. Los espacios de colores bien documentados incluyen sRGB, AdobeRGB y ProPhotoRGB. El sistema visual humano no es un sensor RGB simple, pero se puede aproximar cómo responde el ojo humano. En la visión humana se detectan muchos más tonos de verde que de azul o rojo. Con un espacio de color tricromático como el RGB, se representan estos colores en el equipo usando tres valores, lo que se restringe codificar un triángulo de colores.

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 [9] [10].

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 [9].

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

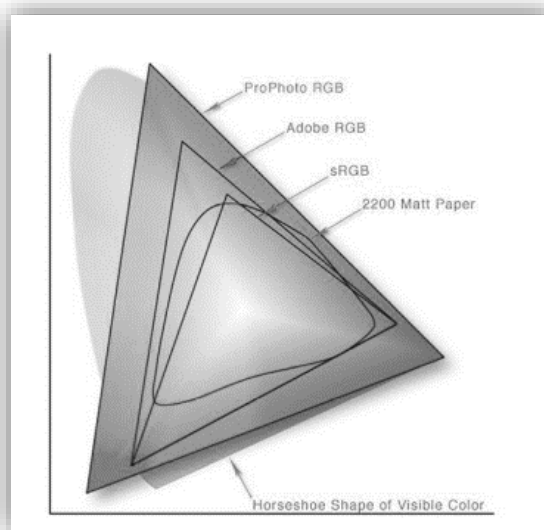


Ilustración 4: Espacios de color. Tomada de Imágenes Google.

1.2.1 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.

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 [9].

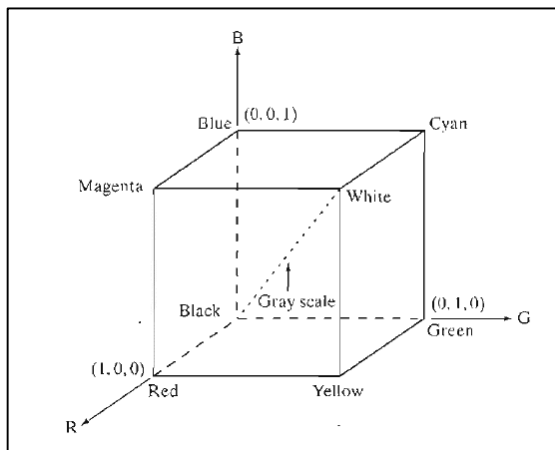


Ilustración 5: Subespacios de colores. Tomada de Imágenes Google.

1.2.2 Espacio de color CMY

El cyan, 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:

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (1)$$

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 [8] [9].

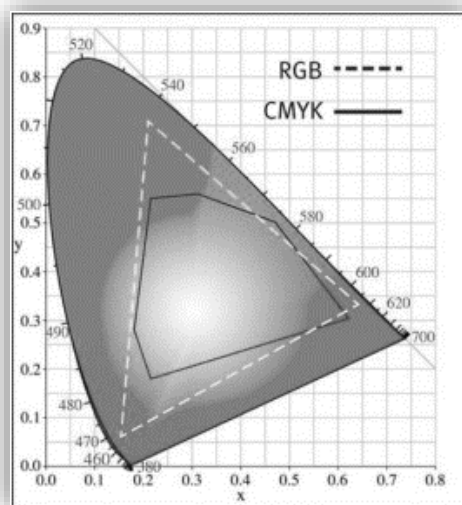


Ilustración 6: Subespacio de color CMY. Tomada de Imágenes Google.

1.2.3 Espacio de color HSV

Es común que se desee elegir un color adecuado para alguna de nuestras aplicaciones, cuando es así resulta muy útil usar la ruleta de color HSV. En ella el matiz se representa por una región circular; una región triangular separada, puede ser usada para representar la saturación y el valor del color. Normalmente, el eje horizontal del triángulo denota la saturación, mientras que el eje vertical corresponde al valor del color. De este modo, un color puede ser elegido al tomar primero el matiz de una región circular, y después seleccionar la saturación y el valor del color deseados de la región triangular.

El valor representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100%. 0 siempre es negro. Dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado.

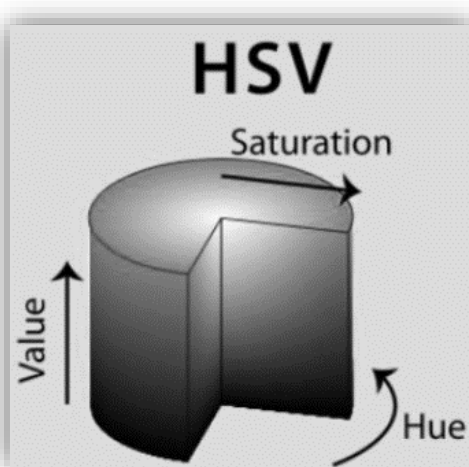


Ilustración 7: Subespacio de color HSV. Tomada de Imágenes Google.

1.2.4 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 trasmite la luz. Por ejemplo, un carro azul refleja un tono azul [18] [19].

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 [19].

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 [19].

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 [19].

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 [19].

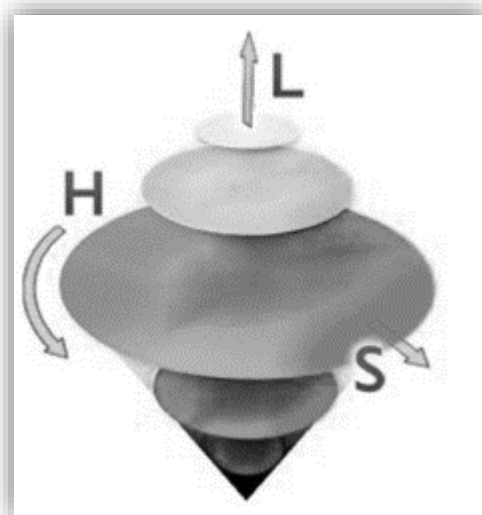


Ilustración 8: Subespacio de color HSL. Tomada de Imágenes Google.

1.3 Calidad de la imagen

La resolución de una imagen indica la cantidad de detalles que puede observarse en esta. El término es comúnmente utilizado en relación a imágenes de fotografía digital, pero también se utiliza para describir cuán nítida (como antónimo de granular) es una imagen de fotografía convencional (o fotografía química). Tener mayor resolución se traduce en obtener una imagen con más detalle o calidad visual.

Los dos parámetros que influyen en la calidad de una imagen por mapa de puntos son la resolución y la profundidad del color.

La resolución: Es el número de puntos por pulgadas (ppp) que representa la imagen. Para representar una imagen en el ordenador. Es mejor utilizar una cuadrícula con cuadrados pequeños para que parezca más real. La resolución se mide normalmente en píxeles, existen diferentes resoluciones como: 72, 1540, 300, 600 puntos por pulgadas o incluso más.

La profundidad del color: Es el número de colores que muestra. A mayor número de colores, más se parece la imagen a la realidad. El color se construye según el número de bits que usan para codificarlo. Si se utiliza un bit se tendrán dos colores (negro y blanco) o si se utilizan 24 bits se obtendrán más de 16 millones de colores.

1.4 Filtrado de imágenes

Es el conjunto de técnicas englobadas dentro del preprocesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen original, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma, lo cual posibilita efectuar operaciones del procesado sobre ella.

Los principales objetivos que se persiguen con la aplicación de filtros son:

Suavizar la imagen: Reducir la cantidad de variaciones de intensidad entre píxeles vecinos.

Eliminar ruido: Eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.

Realzar bordes: Destacar los bordes que se localizan en una imagen.

Detectar bordes: Detectar los píxeles donde se produce un cambio brusco en la función intensidad.

Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella.

El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio.

El Realce y Mejora o Filtrado es un método para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos. Este caso en ocasiones es crucial para lograr los objetivos propuestos sobre la imagen. El rol fundamental de este paso es la transformación más adecuada de la imagen en aras de facilitar los posteriores procedimientos en el análisis de la escena. Esta etapa comprende varios tipos de procedimientos como son el mejoramiento y restauración.

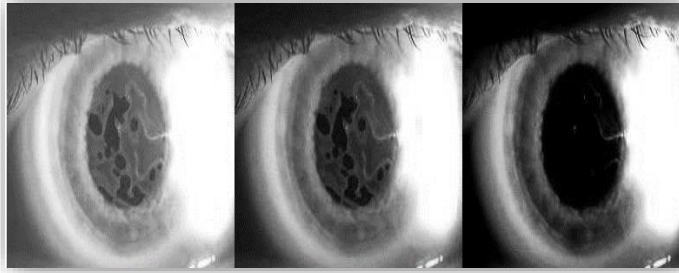


Ilustración 9: Imágenes filtradas.

1.5 Formatos de compresión

Se han desarrollado diferentes técnicas de compresión debido a que los archivos de imagen pueden ocupar mucho espacio, provocando que muchas veces sean imposibles de manejar con máquinas comunes. Estas técnicas tratan de reducir, mediante algoritmos matemáticos, el volumen del archivo para así disminuir los recursos que consume y abreviar el tiempo de transferencia. Estos complejos algoritmos matemáticos reducen de diversos modos los 0 y 1 que conforman una imagen digital. Asimismo, como con los formatos de imagen, las técnicas de compresión son de dominio público o pertenecen a la empresa que las desarrolló. Su división más común es la compresión sin pérdida y la compresión con pérdida, lo cual radica en que tanta información de la imagen se pierde al ser comprimida.

Compresión con pérdida:

La compresión con pérdida hace que los algoritmos usados, para reducir las cadenas del código, desechen información redundante de la imagen. Así, los archivos comprimidos con este método pierden parte de los datos de la imagen. Algunos formatos, como el jpg, compensan esta pérdida con técnicas que suavizan los bordes y áreas que tienen un color similar, haciendo que la falta de información sea invisible a simple vista. Este método permite un alto grado de compresión con pérdidas en la imagen que, muchas veces, sólo es visible si se realiza un fuerte acercamiento –zoom–. El grupo JPEG (Joint Photographic Experts Group) incluye este método de compresión en los archivos jpg y éste es, por mucho, el formato más difundido en el diseño para Internet. Se utiliza usualmente para almacenar fotografías y otras imágenes de tono continuo. Gracias a que utiliza un sistema de compresión que de forma eficiente reduce el tamaño de los archivos.

Compresión sin pérdida:

Esta técnica condensa las cadenas de código sin despreciar nada de la información que forma la imagen, por lo que ésta se regenera intacta al ser descomprimida. Sin embargo, es menor la capacidad de compresión que provee este tipo de técnicas; dado que su fin es permitir una impresión de calidad, además de una exacta visualización de la imagen. A continuación, se describen algunos de estos formatos.

- ✓ PNG son las siglas del grupo que lo desarrolló (*Portable Networks Graphics*) pensando en un formato ideal para su distribución en Internet. Posee ventajas respecto a los otros formatos, ya que fue desarrollado especialmente para su distribución en red, permite altos niveles de

comprensión, además, permite utilizar la técnica de la indexación para crear colores transparentes, semitransparencias o transparencias degradadas. Finalmente, no está limitado a una paleta de 256 colores, sino que puede utilizar millones de colores. Su única limitación es que no podemos crear ficheros animados.

- ✓ El formato TIF (*Tag Image File Format*) se utiliza para imágenes de mapa de bits y es admitido prácticamente por todas las aplicaciones de autoedición y tratamiento de imágenes. Lo reconocen casi todos los programas. Su uso es de los más extendidos en la industria gráfica por la calidad de imagen y de impresión que presenta.
- ✓ DICOM (*Digital Imaging and Communication in Medicine*) es el estándar para el intercambio de imágenes médicas, pensado para su manejo, visualización, almacenamiento, impresión y transmisión. DICOM se diferencia de otros ficheros de datos en que agrupa la información dentro de un conjunto de datos. Los ficheros DICOM consisten en una cabecera con campos estandarizados y de forma libre, y un cuerpo con datos de imagen. Un objeto DICOM simple puede contener solamente una imagen, pero esta imagen puede tener múltiples "fotogramas" (*frames*), permitiendo el almacenamiento de bloques de cine o cualquier otro dato con varios fotogramas.

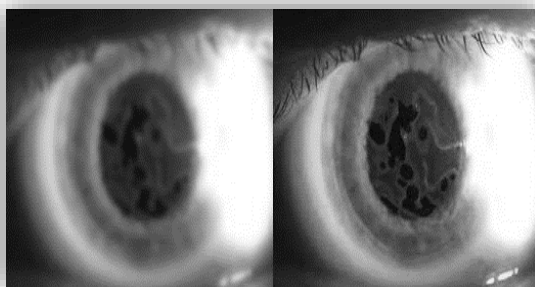


Ilustración 10: Imagen con formato de compresión con pérdida y sin pérdida respectivamente.

1.6 Lámpara de hendidura

La lámpara de hendidura, también llamada biomicroscopio, es un dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras de la porción anterior del ojo.

La lámpara de hendidura es uno de los instrumentos de diagnóstico más comúnmente usados por un oftalmólogo de hoy. Proporciona iluminación y magnificación para examinar varias partes del ojo. Una serie de accesorios se pueden añadir a una lámpara de hendidura para convertirlo en un instrumento de medida. Uno puede medir la presión intraocular, la curvatura de la córnea, el espesor de la córnea, la distancia entre la córnea y el lente, el volumen de la cámara anterior, la opacidad, etcétera, utilizando accesorios diferentes [20]. Algunas lámparas de hendidura tienen accesorios para conectar una cámara fotográfica. Las lámparas de hendidura también se utilizan para proporcionar energía láser en cualquier lugar del ojo para el tratamiento.

A continuación, se describen los componentes principales del equipo y una descripción de los subsistemas principales: una lámpara de hendidura moderna consta de tres componentes principales [21]:

- ✓ Un sistema de iluminación - fuente de luz, espejos y prismas.
- ✓ Un sistema de magnificación - el biomicroscopio.
- ✓ Un sistema mecánico que une el sistema de magnificación con el sistema de iluminación y proporciona movimientos verticales y laterales para enfocar la luz en la parte deseada del ojo.



Ilustración 11: Lámpara de hendidura. Tomado de imágenes Google.

1.6.1 Adaptador de cámara para la lámpara de hendidura

El adaptador de *smartphone* para lámpara de hendidura le permitirá tomar fotografías de alta calidad con rapidez usando cualquier Smartphone en el mercado.

A continuación, se detallan las principales ventajas del adaptador de *smartphone* para la lámpara de Hendidura [22]:

- ✓ Muy buena calidad de fotos.
- ✓ También puede tomar fotos de fondo de ojo usando un lente de diagnóstico. (Se recomienda usar vídeo para fondo de ojo)
- ✓ Varias opciones de aplicaciones para compartir las fotos, ya sea con colegas o su computadora.
- ✓ Los tamaños compatibles son de: 4" a 5.5" diagonal de pantalla. Ocular: 25 a 45 mm de diámetro.
- ✓ Compatible con todas las lámparas y microscopios en el mercado. Bandas de adaptación elásticas incluidas para oculares pequeños.
- ✓ Toma segundos configurarlo. No es necesario sacar el Smartphone de su estuche.
- ✓ Posibilidad de conectar a la TV para mostrar las fotos y video (Aplicaciones para enseñanza)

Un sistema novedoso, simple y eficaz que permite de forma económica crear un completo sistema de video. Es el complemento perfecto para una atención profesional y personalizada de sus pacientes. El novedoso sistema adapta la mayoría de los móviles convencionales a su lámpara de hendidura, es válido para todos los modelos de lámparas de hendidura [22].



Ilustración 12: Adaptador para la lámpara de hendidura. Tomado de imágenes Google.

Este adaptador presenta gran utilidad en nuestro país, debido a que en la actualidad los smartphones presentan disímiles prestaciones, específicamente en términos de calidad de imagen, por lo que en la actualidad los médicos lo obtienen para utilizar sus dispositivos móviles, reemplazando la antigua DC-1 y sus deficiencias.

1.7 Comunicación

Topcon presenta una serie de lámparas de hendidura digital que combina alto rendimiento óptico con una excelente versatilidad óptica. El concepto de alta calidad digital se fundamenta en el extenso rango de dispositivos de captura digital opcionales que pueden acoplarse en estos instrumentos clínicos tan valiosos. [22]

Se especializan en una gama completa de lámparas de hendidura digitales disponible para cubrir las necesidades clínicas de cada especialista del cuidado de la visión. Las lámparas de hendidura proveen una clara definición de imagen, fidelidad de color y transmisión de luz. [22]

Estas vienen acopladas con un cable de conexión Universal Serial Bus (USB), que permite la interconexión con los equipos de cómputo mediante esta vía. Las versiones DC-4, SL-D2, SL-2G, SL-D301, SL-D4 y SL-D701 de los dispositivos de lámparas de hendidura Topcon presentan esta conexión USB, que permite ganancias como [23]:

- ✓ Un solo tipo de conector para todos los periféricos.
- ✓ Habilidad para conectar varios dispositivos periféricos al mismo conector.
- ✓ Un método para facilitar los conflictos por recursos.
- ✓ Conexión en caliente.
- ✓ Detección y configuración automática de los periféricos.

- ✓ USB soporta tres tipos de velocidades.
- ✓ Bajo precio para la implementación del sistema y los periféricos.
- ✓ Aumento en la capacidad de performance.
- ✓ Soporte para hardware y software legado.
- ✓ Implementación de bajo consumo energético

Este tipo de conexión soporta velocidades de [24]:

- ✓ 1.5 Mbps (*low speed*)
- ✓ 12 Mbps (*full speed*)
- ✓ 480 Mbps (*high speed*)

Para un mejor entendimiento se describe a continuación el flujo de datos en la comunicación USB:

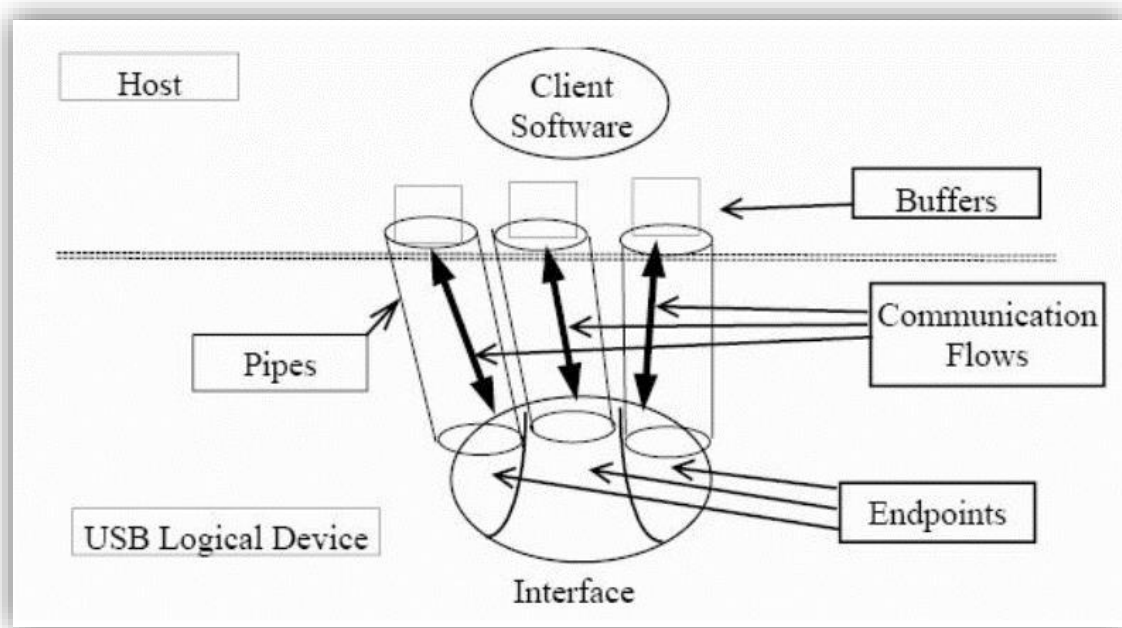


Ilustración 13: Flujo de datos de la comunicación USB.

1.8 Conclusiones del capítulo

Luego de haber realizado un análisis de los conceptos fundamentales relacionados con las imágenes digitales, se puede arribar a las siguientes conclusiones:

- ✓ El software utilizado actualmente para la comunicación con la lámpara de hendidura utiliza el formato de compresión jpg, formato de compresión con pérdida, lo que provoca que se pierdan partes de los datos de la imagen, aunque esto en ocasiones sea invisible a simple vista. Por otra parte, las imágenes DICOM posibilitan almacenar los datos del paciente y del estudio realizado, además de almacenar secuencia de imágenes; lo cual lo hacen ideal para el almacenamiento de las imágenes provenientes de la lámpara de hendidura.

Capítulo 2: Herramientas y tecnologías

En el presente capítulo 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.

2.1 Lenguaje de programación

Un lenguaje de programación es un idioma artificial diseñado para expresar instrucciones que pueden ser llevadas a cabo por un ordenador. Puede usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana. Permiten especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una gran cantidad de opciones posibles. Todo esto, a través de un lenguaje que intenta ser relativamente próximo al lenguaje humano o natural [26]. Para el desarrollo de esta aplicación se escogió como lenguaje de programación a Java, por ser un lenguaje orientado a objetos, lo que le proporciona una gran reusabilidad, independencia de la plataforma además de ser el lenguaje en que está desarrollado el software, al cual se le agregará esta propuesta de solución. Dentro de las características fundamentales de Java se encuentran [27]:

- ✓ Simple: elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.
- ✓ Robusto: maneja la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución.
- ✓ Portable: un programa compilado de Java puede ser utilizado por cualquier computadora que tenga instalado el intérprete de Java, ya que su código compilado es interpretado.
- ✓ Multiproceso: puede ejecutar diferentes líneas de código al mismo tiempo.
- ✓ Dinámico: no es necesario que compile todas las clases de un programa para que este funcione. Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases.
- ✓ Interpretado: corre en máquina virtual.

2.2 Entorno integrado de desarrollo

Para el desarrollo de la solución se seleccionó como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) el NetBeans 8.2. Es libre y de código abierto; posee todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente [28].

2.3 Herramienta de prueba

Para la realización de pruebas de diversos algoritmos y técnicas sobre la imagen oblicua, 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 [29]. 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 MATLAB, el entorno matemático sobre el que se sustenta, es ideal para este procesamiento digital de imágenes ya que las imágenes digitales son matrices al fin y al cabo [30].

Entre las funciones principales de la IPT destacan [30]:

- ✓ Mejora y filtrado de imágenes y 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.

2.4 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. Hace posible la generación de código Java desde los diagramas [31] [32].

2.5 Herramientas de análisis de datos USB

En el presente epígrafe se detallan las herramientas de análisis de datos USB, estas representan un proceso consistente en inspeccionar, limpiar y transformar datos con el objetivo de resaltar información útil, para analizar los datos USB [33].

2.5.1 USB Monitor

Advanced USB Port Monitor es el bus USB, el dispositivo USB y el software analizador de protocolo. Advanced USB Port Monitor incluye la sólida funcionalidad que los clientes han capturado, visto y procesado el tráfico USB. Ofrece una visualización sofisticada y una búsqueda inteligente para depurar y probar de manera precisa y eficiente dispositivos y software USB de velocidad alta (480 Mbps), completa (12 Mbps) y baja (1,5 Mbps).

El software Advanced USB Port Monitor le permite visualizar los paquetes enviados, decodificar los descriptores, detectar errores en periféricos o controladores y medir el rendimiento del dispositivo y del controlador. Al capturar los paquetes, la pantalla estadística en tiempo real le permite verificar el estado del bus, incluso antes de mirar los paquetes que se han leído. Estos se muestran en una lista cronológica junto con la dirección del periférico y el número de punto final. Para facilitar la identificación de los paquetes que le interesan, el software proporciona funciones de filtrado y codificación de color de paquetes. También le da al usuario la opción de varios niveles de detalle, y la información tomada del tráfico contiene referencias a la especificación USB. Le ayudará a desarrollar dispositivos USB y analizar su tráfico en todas las etapas de un proceso de desarrollo. Advanced USB Port Monitor puede ampliar su entorno de laboratorio o una lista de herramientas personales de desarrolladores USB y usuarios avanzados de dispositivos USB [34].

2.5.2 USB Analyzer 4.0

USB Analyzer es un analizador de protocolo, puede capturar y visualizar cualquier tráfico que pase por las conexiones USB de su computadora. El formato binario bruto de los datos de comunicación capturados se analiza y decodifica en una forma legible por el ser humano, lo que le permite revisar los datos intercambiados y realizar un análisis forense eficaz de los paquetes de datos transmitidos [35].

La característica clave de este analizador lógico USB es su capacidad para procesar paquetes monitoreados en tiempo real incluso bajo una alta carga de tráfico. Junto con otras poderosas características de esta utilidad gratuita de explorador y analizador USB, le permite explorar y rastrear las solicitudes de USB de todos los dispositivos y aplicaciones modernos.

A continuación, se detallan algunas ventajas del analizador USB [35]:

- ✓ Controla el tráfico del bus, decodifica y muestra los datos USB, filtra el tráfico del bus.
- ✓ Intercepta y ve transferencias de control USB, interrupción de transferencias, transferencias masivas y transferencias isócronas.
- ✓ Protocolos USB de análisis e ingeniería inversa, dispositivos USB de ingeniería inversa, controladores y aplicaciones.
- ✓ Capture datos transmitidos a través de alta velocidad (*Hi-Speed*) USB directamente a su computador.
- ✓ Capture tráfico de datos en el objetivo Dispositivos USB 2.0 de alta velocidad.
- ✓ Controle los datos de las unidades externas que fluyen entre la interfaz del dispositivo y el puerto de interfaz USB.
- ✓ Analizar los protocolos de transferencia de datos entre computadoras locales o remotas y dispositivos USB. [35]

2.5.3 Wireshark

La herramienta Wireshark, antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos de forma únicamente hueca [36].

La funcionalidad que provee es añadir una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo³.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP [36].

2.5.4 Wireshark con USBPcap

Wireshark es un analizador de protocolos de fuente abierta diseñado por Gerald Combs y que actualmente está disponible para múltiples plataformas. Conocido originalmente como Ethereal, su principal objetivo es el análisis de tráfico, además de ser una excelente aplicación didáctica para el estudio de las comunicaciones y para la resolución de problemas de red [36].

Wireshark implementa una amplia gama de filtros que facilitan la definición de criterios de búsqueda para más de 1100 protocolos soportados actualmente y una interfaz sencilla e intuitiva que permite desglosar por capas cada uno de los paquetes capturados. Gracias a que Wireshark entiende la estructura de los protocolos, se pueden visualizar los campos de cada una de las cabeceras y capas que componen los paquetes monitorizados, proporcionando un gran abanico de posibilidades al administrador de redes a la hora de abordar ciertas tareas en el análisis de tráfico [37].

USBPcap es un completo sniffer de código abierto y que funciona en los sistemas operativos Windows XP, Vista, Windows 7, Windows 8 y Windows 8.1. Esta herramienta permite capturar toda la actividad de cualquier dispositivo USB en formato pcap para posteriormente analizarlo con el popular programa Wireshark [38].

2.6 Herramientas de análisis de datos Serial

En el análisis de datos serial se utilizan varias herramientas, las cuales se usan para conectar un equipo a una amplia gama de equipos que no exigen velocidad de intercambio de datos, para esto

³ Modo promiscuo: es aquel en el que una computadora conectada a una red compartida, tanto la basada en cable de cobre como la basada en tecnología inalámbrica, captura todo el tráfico que circula por ella.

es necesario capturar y almacenar datos serie para un uso y análisis posterior. Además, algunas de estas herramientas tienen soporte para el análisis y captura del puerto USB.

2.6.1 Free serial port monitor

El software libre monitor de puerto serie, succionador de COM Rs232 con la comunicación analizador de paquetes de datos. Esta utilidad de monitorización puede espiar, capturar, ver, analizar, probar la actividad portuaria de realizar la conexión del puerto y el análisis de tráfico con adquisición de datos y control. Este sistema se puede utilizar como herramienta de prueba de interfaz de dispositivo, visor de transferencia de datos del módem y así sucesivamente [39].

Serial Monitor puede ser utilizado con éxito para controlar el flujo de datos entre los dispositivos de serie y la aplicación de Windows, la depuración de las conexiones del puerto, el desarrollo, la ingeniería inversa y la implementación de protocolo serie, el desarrollo de controladores de dispositivos serie, el desarrollo de hardware de serie, la investigación de la funcionalidad de cualquier software de terceros y el hardware, el software de pruebas y el hardware.

Serial Port Monitor puede ser utilizado con éxito por los programadores, desarrolladores de software, ingenieros de hardware, especialistas de apoyo, personal de control industrial, ingenieros de software, técnicos de la ciencia, integradores de sistemas, consultores, expertos de laboratorio, estudiantes de la universidad [39].

2.6.2 Serial monitor

Free Serial Port Monitor le permite interceptar, visualizar y analizar todos los datos intercambiados entre la aplicación de Windows y el dispositivo serie. Se puede utilizar con éxito en el desarrollo de aplicaciones, el controlador de dispositivos o el desarrollo de hardware en serie y ofrece una potente plataforma para una codificación, prueba y optimización efectivas [40].

2.7 Herramientas para el trabajo con DICOM

En el trabajo con imágenes DICOM se utilizan varias bibliotecas, las cuales se utilizan tanto para crear una imagen DICOM, así como para consultar los pixeldata y metadata de una imagen DICOM.

2.7.1 Biblioteca Pixelmed

Este es un juego de herramientas DICOM independiente que implementa código para leer y crear datos DICOM, compatibilidad con archivos y redes DICOM, una base de datos de objetos DICOM, soporte para visualización de directorios, imágenes, informes y espectros, y validación de objetos DICOM.

El kit de herramientas es una implementación completamente nueva, que no depende de ninguna otra herramienta DICOM, comercial o gratuita. Hace uso de otras herramientas de Java puro de libre disposición para compresión y compatibilidad con XML y bases de datos.

El kit de herramientas es de código abierto y está disponible de forma gratuita tanto para uso comercial como no comercial. Esta biblioteca se utiliza específicamente para la creación de una imagen DICOM.

2.7.2 Biblioteca ImageJ

ImageJ es un programa de procesamiento de imagen digital de dominio público programado en Java.

Fue diseñado con una arquitectura abierta que proporciona extensibilidad vía *plugins* Java y macros (macroinstrucciones) grabables. Se pueden desarrollar *plugins* de escaneo personalizado, análisis y procesamiento usando el editor incluido en ImageJ y un compilador Java.

Los plug-ins escritos por usuarios hacen posible resolver muchos problemas de procesado y análisis de imágenes, desde imágenes en vivo de las células en tres dimensiones, procesado de imágenes radiológicas, comparaciones de múltiples datos de sistema de imagen hasta sistemas automáticos de hematología. La arquitectura de plugins y entorno de desarrollo integrados de ImageJ lo han convertido en una plataforma popular para enseñar procesamiento de imagen. Esta biblioteca se utiliza específicamente para la consulta de pixeldata y metadata de una imagen DICOM.

2.8 Justificación de la selección de las herramientas de análisis

Free serial port monitor y serial monitor se utilizaron inicialmente para el análisis del flujo de datos provenientes de la lámpara de hendidura, siendo el mismo por el protocolo USB, a pesar de tener soporte para este protocolo su enfoque principal es para los puertos RS232 o puerto de serie, por ello, presentaban algunas deficiencias en cuanto a identificación de información clave y del filtrado de la misma.

Posteriormente, se utilizó USB monitor superando las deficiencias anteriormente mencionadas, además de contar con una interfaz un poco mejorada y mayor información para el trabajo con el protocolo USB. Aun así, la documentación no fue la más explícita y acorde a las necesidades para el análisis necesario.

Con USB Analyzer es un software muy actual con una interfaz intuitiva y fácil de trabajar además de amplia documentación y ejemplos de trabajos con este tipo de protocolo (USB). Se utilizó para los primeros pasos de análisis de la información proveniente de la lámpara de hendidura pero que, por estar bajo una licencia de prueba de solo 30 días, por lo que se redujo su uso y solo se utilizó en los primeros análisis que sentaron las bases para mejor entendimiento de los datos provenientes de la lámpara de hendidura.

Finalmente, Wireshark, una herramienta muy utilizada y potente para capturar el tráfico de los protocolos existentes en la actualidad, fue la seleccionada para este propósito contando con amplia documentación, siendo libre además y con amplias opciones de filtrado de datos y formas de representación de la misma. Con esta herramienta el proceso se divide en dos partes fundamentales la captura de los datos mediante el driver USBPcap creando un fichero con extensión .pcap proveniente de una interfaz USB seleccionada y mediante la carga del mismo por el software Wireshark además de poder realizar este proceso de manera directa o indirecta.

2.9 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 [41].

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 [42]. El cliente es parte del equipo de desarrollo, el equipo de desarrollo es de dos personas, además de la dificultad para un equipo de desarrollo pequeño, adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo [43] [41], lo cual permitió identificar la Programación Extrema (XP) como una alternativa acertada, además de ser la metodología de desarrollo empleada en el grupo de investigación, AIRI (*Artificial Intelligence: Research and Innovation*) para los sistemas desarrollados hasta la actualidad .

La metodología XP presenta varias ventajas, entre ellas [41]:

- ✓ 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 [44]:

- ✓ 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 se desarrolla en función de cada historia de usuario, además de ser una fase donde se definen las tareas de la ingeniería y los tiempos reales en 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.

2.10 Conclusiones del capítulo

Luego de haber realizado un estudio bibliográfico de las herramientas y tecnologías, se puede arribar a las siguientes conclusiones:

- ✓ Las herramientas y tecnologías empleadas en el grupo de investigación AIRI para el desarrollo de sistemas similares (a Java como lenguaje de programación, NetBeans como IDE, Visual

Paradigm como herramienta para el modelado, Matlab como herramienta de prueba, XP como metodología de desarrollo) se adecuan a las necesidades de desarrollo de la propuesta de solución.

Capítulo 3: Análisis y diseño de la solución

Este capítulo tiene como objetivo describir las actividades desarrolladas durante todo el proceso de análisis y diseño. Se detalla la propuesta de solución y se describe su arquitectura, los requisitos funcionales y no funcionales, conjuntamente con los artefactos que plantea la metodología escogida que contribuyeron al desarrollo de las nuevas funcionalidades agregadas.

Se describe la fase inicial de la metodología XP: planificación y se obtienen los artefactos importantes de estas fases, tales como, las Historias de Usuarios, Plan de Iteraciones, Plan de Duración de Iteraciones y Plan de Entregas.

3.1 Descripción de la solución

A continuación, se muestra un diagrama de flujo de proceso donde se describen las actividades a realizar en la interfaz de captura del sistema.

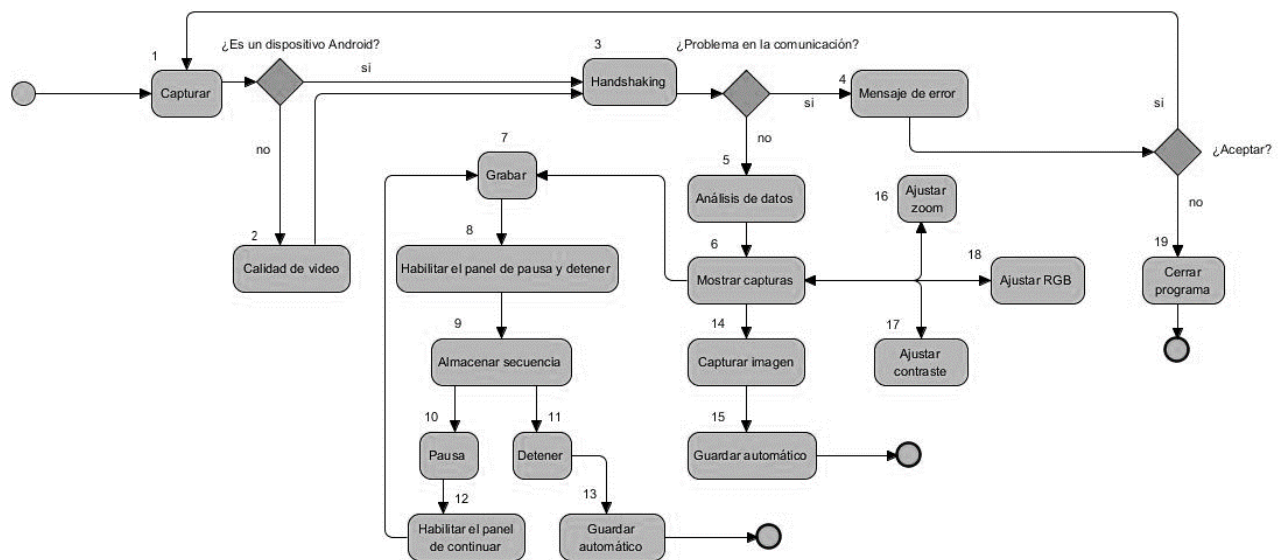


Ilustración 14: Diagrama de flujo de proceso de la interfaz Capturar.

1. Seleccionar el dispositivo que va a realizar la captura de los datos provenientes de la lámpara de hendidura, ya sea un dispositivo Android o la DC-1.
2. Se selecciona la calidad de video que utilizará la DC-1.
3. Realizar el proceso Handshaking⁴ entre el software y la DC-1⁵ o el dispositivo Android.

En caso que fuera una DC-1, se comprueba que la comunicación se esté efectuando correctamente para comenzar a enviar los datos, a partir de esto se obtiene el rootHUB (Controlador raíz), el cual es un bus virtual al cual están conectadas todas las interfaces usb de la PC, a partir de VID (Vendor ID) y el PID (Product ID) correspondiente a la DC-1, se

⁴ Establecimiento de comunicación

⁵ Cámara digital utilizada en la lámpara de hendidura para captura de imágenes, obteniendo un análisis detallado y diagnóstico preciso de las estructuras oculares.

obtiene la interfaz donde está conectada y se comprueba que haya transcurrido la fase de setup⁶, en caso contrario se le envía la solicitud pertinente para que transcurra la misma. Una vez transcurrida la DC-1 está lista para recibir solicitudes y se envía una para comprobar que esté lista para comunicarse con el software.

En caso que sea un dispositivo Android, se inicializa el servicio ADB (*Android Debug Bridge*), el cual realiza el port forwarding para la utilización del puerto USB como medio de comunicación entre la PC y el dispositivo móvil.

4. Mostrar un mensaje de error si el proceso de Handshaking no se realiza correctamente con una descripción del mismo.
5. Realizar un análisis de datos donde se decodifica el streaming⁷ obtenido directamente del dispositivo hacia la PC para convertirlo en un formato de imagen el cual es la base para las funcionalidades del software.
6. Mostrar las capturas realizadas en el momento deseado.
7. Realizar la acción de grabar, permitiendo grabar una secuencia de video.

En la actividad 5 y 6 se van a guardar automáticamente en un directorio en la ruta donde se encuentra la aplicación, y se guardan en formato DICOM que es el formato de comunicación médica estandarizado.

8. Habilitar el panel de pausa y detener.
9. Almacenar la secuencia grabada.
10. Habilitar la pausa.
11. Habilitar el detener.
12. Habilitar el panel de continuar.
13. Guardar automático.
14. Capturar la imagen.
15. Guardar automático.
16. Ajustar el zoom para determinar un mayor o menor tamaño para cada uno de los pixeles.
17. Ajustar el contraste de la imagen.
18. Ajustar cada uno de los canales RGB (rojo, verde y azul), donde se pueden modificar los valores de cada uno de los pixeles.
19. Cerrar el programa luego de ser verificado.

⁶ Setup también se utiliza como sinónimo de BIOS (Basic Input-Output System). Este es el Sistema Básico de Entrada-Salida, un software que reconoce los dispositivos necesarios para cargar el sistema operativo en la memoria ROM de la computadora.

⁷ Streaming, es la distribución digital de contenido multimedia de manera que el usuario pueda utilizar el producto a la vez que se descarga. La palabra *retransmisión* se refiere a una corriente continua que fluye sin interrupción.

A continuación, se muestra un diagrama de flujo de proceso donde se describen las actividades a realizar en la interfaz principal del sistema.

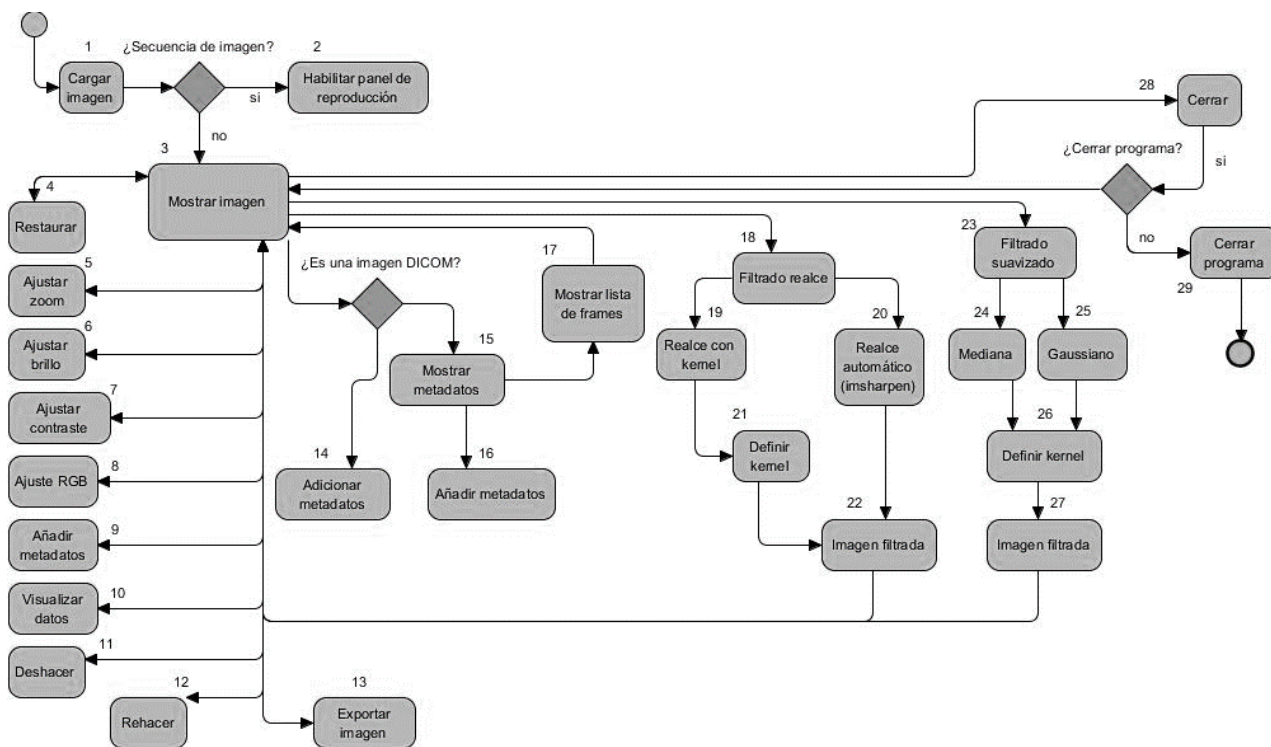


Ilustración 15: Diagrama de flujo de proceso de la interfaz Principal.

1. Cargar la imagen almacenada en la PC.
2. Habilitar el panel de reproducción si es una secuencia de imágenes lo que proviene de la PC.
3. Mostrar la imagen cargada.
4. Restaurar la imagen luego de realizarle las modificaciones que más adelante se detallan.
5. Ajustar el zoom para determinar un mayor o menor tamaño para cada uno de los pixeles.
6. Ajustar el brillo donde se aumenta o disminuye la intensidad de cada uno de los pixeles de la imagen.
7. Ajustar el contraste de la imagen.
8. Ajustar cada uno de los canales RGB (rojo, verde y azul), donde se pueden modificar los valores de cada uno de los pixeles.
9. Añadir metadatos permite agregar los datos del estudio a la imagen cargada.
10. Mostrar la lista de frames de la imagen DICOM cargada.
11. Deshacer cambios en la imagen.
12. Rehacer cambios en la imagen.
13. Exportar la imagen a los diferentes formatos (jpg, png, tiff, DICOM).
14. Añadir metadatos permite agregar los datos del estudio a la imagen cargada.
15. Mostrar los metadatos.
16. Añadir metadatos permite agregar los datos del estudio a la imagen cargada.
17. Mostrar lista de frames para seleccionar el deseado.
18. Aplicar el método de filtrado de realce con el objetivo que la imagen se observe con mayor nivel de detalle.
19. Realizar un filtrado de realce convolucionando la imagen con un kernel definido por el usuario.

20. Realizar un filtrado automático(imsharpen).
21. Definir el kernel a utilizar.
22. Guardar automático la imagen filtrada y se muestra con los cambios realizados.
23. Aplicar el método de filtrado suavizado, donde este se utiliza para eliminar el ruido de la imagen.
24. Aplicar el filtro de la mediana, donde se sustituyen los valores dentro de un kernel ya previamente definido por el valor de la media de la imagen.
25. Aplicar el filtro gaussiano, donde se utiliza un kernel gaussiano del tamaño que el usuario defina para filtrar la imagen.
26. Definir el kernel a utilizar.
27. Guardar automático la imagen filtrada y se muestra con los cambios realizados.
28. Cerrar el programa.
29. Cerrar el programa luego de ser verificado.

3.2 Fase de planeación

La fase de planeación es la etapa inicial del desarrollo de software de la metodología XP. En este punto comienza a interactuar con el cliente para identificar cuáles son las historias de usuario (HU). Donde se definen el número y tamaño de las historias de usuario, en donde se plantean los ajustes necesarios a la metodología según las características del proyecto y el cliente definen el nivel de prioridad de las historias de usuario, como el tiempo, el esfuerzo que conllevarán su desarrollo.

3.2.1 Especificación de los requisitos

La ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales.

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema de software es llamado Ingeniería de Requerimientos. La meta de esta es entregar una especificación de requerimientos de software correcta y completa. La misma apunta a mejorar la forma en que se comprenden y definen sistemas de software complejos [45], trata los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora de forma sistemática y repetible [46].

Requisitos funcionales (RF): son declaraciones de las funcionalidades que debe proporcionar el sistema. Definen la manera en que el software debe reaccionar a determinadas entradas. Especifican cómo debe comportarse el sistema en situaciones particulares. Pueden declarar explícitamente lo que el sistema no debe hacer [47]. Los requisitos funcionales de la aplicación son los siguientes:

- ✓ **RF.1** Cargar secuencia de imágenes
- ✓ **RF.2** Reproducir secuencia de imágenes
- ✓ **RF.3** Gestionar metadatos
- ✓ **RF.4** Ajustes de valores de los pixeles
- ✓ **RF.5** Filtrado de la imagen
- ✓ **RF.6** Captura de datos de la lámpara de hendidura
- ✓ **RF.7** Exportar a diversos formatos

3.2.2 Descripción de historias de usuarios

Los requisitos funcionales describen lo que debe cumplir el sistema en un lenguaje técnico. Una HU es una representación de un requisito de software escrito en una o dos frases al utilizar el lenguaje común del usuario, son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos, permiten responder rápidamente a los requisitos cambiantes, es una manera simple de describir una tarea concisa que aporta valor al usuario o al negocio.

Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Cargar secuencia de imágenes
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	
Descripción: la HU comienza cuando el usuario carga una secuencia de imágenes de diversos formatos, esta operación podrá realizarla al hacer <i>click</i> a través de su selección en el menú de la aplicación.	
Observaciones: La secuencia puede ser de una sola imagen y los formatos permitidos son (jpg, DICOM, png, tiff).	

Tabla 1: Historia de usuario 1 (HU1)

3.2.3 Fase de Planificación de la Entrega

En la fase de la planificación de entrega se definen las prioridades de cada historia de usuario, y consecuentemente se realiza una estimación del esfuerzo necesario de cada una de ellas. Se definen las entregas con el cliente donde se tiene por norma que una entrega debe adecuarse a tres meses como máximo.

En la siguiente tabla se muestran cada una de las historias de usuario, así como la estimación del tiempo en que se cumplirá, donde las tareas de máxima duración son: “reproducir secuencia de imágenes”, “filtros de realce”, “filtros de suavizado”, “extracción de imágenes del tráfico USB” y “exportar imágenes en formato DICOM”, obteniéndose una duración total estimada de 29 semanas.

No	Historias de usuario	Puntos de estimación(semanas)
1.	Cargar secuencia de imágenes	2
2.	Listar secuencia de imágenes	1
3.	Reproducir secuencia de imágenes	3
4.	Añadir metadatos	1
5.	Visualizar metadatos	1
6.	Zoom de la imagen	2
7.	Ajuste de brillo y contraste	2
8.	Manipulación de valores RGB	1
9.	Filtros de realce	3
10.	Filtros de suavizado	3
11.	Captura de tráfico USB	2
12.	Extracción de imágenes del tráfico USB	3
13.	Grabar imágenes de la lámpara de hendidura	2
14.	Exportar imágenes en formato DICOM	3

Tabla 2: Estimación de tiempo por historia de usuario.

3.2.4 Plan de iteraciones

Una vez definidas las historias de usuarios e identificar el tiempo para su implementación, se diseña un plan de iteraciones donde las historias de usuario están contenidas, por tanto, se pretenden realizar el desarrollo en 3 iteraciones, referidas a continuación:

Iteración 1

La iteración tiene como fin cargar las secuencias de imágenes, listarlas y reproducirlas, así como añadir y visualizar los metadatos que serán necesarios para conformar la imagen DICOM.

Iteración 2

Esta iteración se encarga de aplicar zoom y filtros de mejora de imágenes, realce de borde y suavizado, así como ajustar el brillo y contraste.

Iteración 3

Esta iteración captura el tráfico USB, extrae las imágenes, graba imágenes y las exporta luego de realizadas las técnicas anteriormente mencionadas.

Iteraciones	Historias de usuario	Duración
Iteración 1	Cargar secuencia de imágenes	8
	Listar secuencia de imágenes	
	Reproducir secuencia de imágenes	
	Añadir metadatos	
	Visualizar metadatos	
Iteración 2	Zoom de la imagen	11
	Ajuste de brillo y contraste	
	Manipulación de valores RGB	
	Filtros de realce	
	Filtros de suavizado	
Iteración 3	Captura de tráfico USB	10
	Extracción de imágenes del tráfico USB	
	Grabar imágenes de la lámpara de hendidura	
	Exportar imágenes en formato DICOM	

Tabla 3: Estimación por tiempo de las historias de usuarios según las iteraciones.

3.3 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 y fácil de implementar, que a la larga costará menos tiempo y esfuerzo desarrollar.

3.3.1 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son utilizadas para representar las responsabilidades de las clases y sus interacciones. Estas tarjetas permiten trabajar con una metodología basada en objetos [48].

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 [49]. A continuación, se muestra la tarjeta CRC de la clase MyImage, las otras clases se pueden encontrar en los anexos.

Clase: MyImage	
Responsabilidades	Colaboradores
imageToBufferedImage Convert2BufferImage GetWidth getHeigth setOffset generatePixels aplicar_filtros Reset populate_ARGB_Matrix getPixelAround MoveRed MoveBlue MoveGreen Move_Brighth Move_Contrast ChangeBf Negative Noise_Reduction getProm	ImageIO BufferedImage

Tabla 4: Tarjetas CRC de la clase MyImage.

3.3.2 Estándares 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. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada [50]. Para la realización del presente trabajo se establecen tres estándares de codificación:

Variable:

- ✓ Los nombres de las variables deben ser cortos y significativos.
- ✓ La elección de un nombre de variable debe ser mnemotécnica⁸, esto es, diseñado para demostrar el propósito de su uso a cualquier observador.

⁸ Una regla mnemotécnica es una oración corta y fácil de recordar que ayuda de manera artificiosa a relacionar palabras, con el objetivo de memorizar conceptos con más facilidad.

Funciones:

- ✓ Los nombres de los métodos deben ser verbos o palabras que identifiquen de manera general el objetivo del método.
- ✓ Los nombres de los métodos no pueden contener espacios ni caracteres especiales, excepto guión bajo, sólo son permitidas las letras de la “a” a la “z” y los números del 0 al 9.
- ✓ Si el nombre de método requiere estar compuesto por más de una palabra, cada palabra adicional debe empezar con mayúscula.

3.3.3 Patrones de diseño utilizados

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares [51].

Un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. Para ello se aíslan los aspectos comunes y su solución y se añaden cuantos comentarios y ejemplos sean oportunos. Los patrones ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexibles, elegante y en algunos casos reusable [52].

Los patrones generales de software para asignación de responsabilidades (*GRASP por sus siglas en inglés, General Responsibility Assignment Software Patterns*) son los que ofrecen orientación de cómo asignar estas a los objetos ante determinada categoría de problemas, describiendo los principios fundamentales de la asignación. Los que se evidencian en el diseño del sistema son: Creador, Experto, Alta cohesión, Bajo acoplamiento y Controlador [53].

- ✓ **Creador:** El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento promoviendo el bajo acoplamiento.

```
private void jButton_cargarActionPerformed(java.awt.event.ActionEvent evt) {  
    int r = jFileChooser1.showOpenDialog(this);  
  
    if (r == JFileChooser.APPROVE_OPTION) {  
        Inicializar();  
        jButton_AddMetadata.setEnabled(true);  
        if (worker != null) {  
            worker.step();  
        }  
  
        String file = jFileChooser1.getSelectedFile().getAbsolutePath();  
        if (file.endsWith(".dcm")) {  
            jButton_play.setEnabled(true);  
            jButton_mostrarMetadata.setEnabled(true);  
            DefaultListModel<String> list_frames = new DefaultListModel<>();  
  
            dcm.open(file);  
  
            DefaultListModel<String> ls = new DefaultListModel<>();  
            for (int j = 1; j <= dcm.getStackSize(); j++) {  
                ls.addElement("Frame " + j);  
            }  
  
            imagen_actual = new MyImage(dcm.getBufferedImage());  
  
            jList_frames.setSelectedIndex(0);  
        }  
    }  
}
```


- ✓ **Experto:** Asigna la responsabilidad al experto en la información, es decir, a la clase que cuenta con la información necesaria para cumplir con la responsabilidad.

```
public class MyImage implements Serializable {  
  
    private BufferedImage br;  
    private final float delta_zoom = 1;  
}  
public int getWidth() {  
  
    return (int) (this.br.getWidth() * delta_zoom);  
}  
  
public int getHigh() {  
  
    return (int) (this.br.getHeight() * delta_zoom);  
}
```

- ✓ **Alta cohesión:** La información que almacena una clase debe ser coherente y en la medida de lo posible relacionada con ella.

```
public void aplicar_filtros(List<String> filters, int cdad) {  
  
    String[] filter_arg;  
    int kernel = 3;  
    String filter;  
    for (int i = 0; i < cdad; i++) {  
        filter_arg = filters.get(i).split("-");  
        filter = filter_arg[0];  
        kernel = Integer.parseInt(filter_arg[1]);  
        if (filter.equals("Noise Reduct")) {  
            Noise_Reduction(kernel);  
        }  
  
        if (filter.equals("Negative")) {  
            Negative();  
        }  
    }  
}
```

- ✓ **Bajo acoplamiento:** El propósito del patrón es tener las clases lo menos ligadas posible y de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en el resto. Al no asociar las clases de la capa de presentación con la de negocio o la de datos, la dependencia entre las clases se mantiene baja.
- ✓ **Controlador:** Mantiene el control actuando como intermediario entre una determinada interfaz y el algoritmo que la implementa. Además, recibe los datos del usuario y los envía a las distintas clases según el método llamado y permite dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Otros patrones de diseño de gran utilidad durante la fase de diseño son los GOF (*Gang Of Four*), conocidos como patrones de la Pandilla de los Cuatro, dado que el libro que los explica fue escrito por cuatro autores. Caracterizan las formas en que las clases o los objetos interactúan y distribuyen la responsabilidad. Son los encargados de las opciones de comportamiento de la aplicación, permitiendo que el comportamiento varíe en tiempo de ejecución, sin estos patrones cada comportamiento tendría que diseñarse e implementarse por separado [54].

- ✓ **Fachada (Facade):** Provee una interfaz unificada y simple para acceder a una interfaz o grupo de interfaces de un subsistema.

- ✓ **Orden** (Command): Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma.

```
public void Reset() {  
  
    int[] p = new int[3];  
    for (int i = 0; i < this.buffered_image.getWidth(); i++) {  
        for (int k = 0; k < this.buffered_image.getHeight(); k++) {  
            p = getPixel(this.buffered_image.getRGB(i, k));  
            this.RR[i][k] = p[0];  
            this.GG[i][k] = p[1];  
            this.BB[i][k] = p[2];  
        }  
    }  
}
```

- ✓ **Recuerdo** (Memento): Permite volver a los estados anteriores del sistema.

```
public void Inicializar() {  
    filters_list = new ArrayList<>();  
    atribute_list = new String[5];  
  
    play_itrator = 1;  
    initial_x = 0;  
    initial_y = 0;  
    indice_historial = 0;  
    imagen_actual = null;  
    imagen_original = null;  
    atribute_list[0] = "Anonimyzed";  
    atribute_list[1] = "None";  
    atribute_list[2] = "None";  
    atribute_list[3] = "None";  
    atribute_list[4] = "None";  
    dcm = new DICOM();  
    i = 0;  
}
```

- ✓ **Módulo** (Module): Agrupa varios elementos relacionados, como clases, singletons y métodos utilizados globalmente en una entidad única.

3.4 Conclusiones del capítulo

Luego de haber realizado un amplio análisis para diseñar la solución del sistema, se puede arribar a las siguientes conclusiones:

- ✓ El desarrollo estuvo guiado por tres iteraciones, donde la primera iteración trabaja con el manejo de las imágenes, la segunda iteración se encarga de aplicar los diferentes filtros a las imágenes, y la tercera iteración se encarga de captura el tráfico USB, grabar y exportar luego de realizadas las técnicas anteriormente mencionadas.
- ✓ Se realizaron dos interfaces (principal y captura), lo que permitió separar la obtención de las imágenes (dispositivos móviles o cámara de la lámpara de hendidura) del almacenamiento, modificación y visualización de las mismas.

Capítulo 4: Resultados y validación

Durante el desarrollo del presente capítulo se estará ilustrando 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. También se realizan un conjunto de pruebas al sistema, divididas en pruebas unitarias y de aceptación como propone la metodología, finalmente se realiza la validación a los resultados obtenidos aplicando las distintas técnicas de filtrado y mejora a las imágenes provenientes de la lámpara de hendidura.

4.1 Fase de implementación

Dentro del proceso de desarrollo de la metodología del desarrollo del software, la parte más importante es la implementación. Se especifica en esta fase, la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Lo primero es hacer un chequeo de cada HU, en conjunto con el plan de iteraciones y se modifica en caso de ser necesario, para esto se crean tareas de desarrollo 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. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones de desarrollo sobre el sistema.

4.1.1 Iteración 1

En esta iteración se implementaron las funcionalidades que tiene como fin cargar las secuencias de imágenes, listarlas y reproducirlas, así como añadir y visualizar los metadatos que serán necesarios para conformar la imagen DICOM.

Historias de usuarios (HU)	Tiempo de implementación	
	Estimación	Real
Cargar secuencia de imágenes	2	2
Listar secuencia de imágenes	1	1
Reproducir secuencia de imágenes	3	3
Añadir metadatos	1	1
Visualizar metadatos	1	1

Tabla 5: Tiempo de implementación de las HU de la iteración #1

A continuación, se muestran las tareas de ingeniería efectuadas para las funcionalidades implementadas en esta iteración:

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre: Cargar secuencia de imágenes	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Programador responsable: Daimarys Laureiro González	
Descripción: La HU comienza cuando el usuario carga una secuencia de imágenes de diversos formatos, esta operación podrá realizarla al hacer <i>clic</i> a través de su selección en el menú de la aplicación.	

Tabla 6: Tabla de ingeniería de la funcionalidad Cargar secuencia de imágenes.

Tarea	
Número de tarea: 2	Número de HU: 2
Nombre: Listar secuencia de imágenes	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando el usuario carga una imagen DICOM multiframe, quedan listadas los diferentes frames o la secuencia puede ser de una sola imagen.	

Tabla 7: Tabla de ingeniería de la funcionalidad Listar secuencia de imágenes.

Tarea	
Número de tarea: 3	Número de HU: 3
Nombre: Reproducir secuencia de imágenes	
Tipo de tarea: Desarrollo	Puntos de estimación: 3
Programador responsable: Daimarys Laureiro González	
Descripción: La HU comienza cuando quedan listadas los diferentes frames de la imagen, se da en el botón reproducir y se muestra la secuencia de imágenes.	

Tabla 8: Tabla de ingeniería de la funcionalidad Reproducir secuencia de imágenes.

Tarea	
Número de tarea: 4	Número de HU: 4
Nombre: Añadir metadatos	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Programador responsable: Alexis De Armas Francisco	
Descripción: La HU comienza cuando el usuario carga una secuencia de imágenes o simplemente una imagen, se pueden añadir los metadatos, a partir del botón añadir metadatos.	

Tabla 9: Tabla de ingeniería de la funcionalidad Añadir metadatos.

Tarea	
Número de tarea: 5	Número de HU: 5
Nombre: Visualizar metadatos	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Programador responsable: Daimarys Laureiro González	
Descripción: La HU comienza cuando el usuario carga una secuencia de imágenes, se pueden visualizar los metadatos.	

Tabla 10: Tabla de ingeniería de la funcionalidad Visualizar metadatos.

4.1.2 Iteración 2

Esta iteración se encarga de aplicar zoom y filtros de mejora de imágenes, realce de borde y suavizado, así como ajustar el brillo y contraste.

Historias de usuarios (HU)	Tiempo de implementación	
	Estimación	Real
Zoom de la imagen	2	2
Ajuste de brillo y contraste	2	2
Manipulación de valores RGB	1	1
Filtros de realce	3	3
Filtros de suavizado	3	3

Tabla 11: Tiempo de implementación de las HU de la iteración #2.

Tarea	
Número de tarea: 6	Número de HU: 6
Nombre: Zoom de la imagen	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Programador responsable: Alexis De Armas Francisco	
Descripción: La HU comienza cuando se carga la imagen, a partir de este momento se puede ajustar el zoom para determinar un menor o mayor tamaño para cada uno de los pixeles.	

Tabla 12: Tabla de ingeniería de la funcionalidad Zoom de la imagen.

Tarea	
Número de tarea: 7	Número de HU: 7
Nombre: Ajuste de brillo y contraste	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Programador responsable: Daimarys Laureiro González	
Descripción: La HU comienza cuando se carga la imagen, a partir de este momento se puede aumentar o disminuir la intensidad de cada uno de los pixeles de la imagen.	

Tabla 13: Tabla de ingeniería de la funcionalidad Ajuste de brillo y contraste.

Tarea	
Número de tarea: 8	Número de HU: 8
Nombre: Manipulación de valores RGB	
Tipo de tarea: Desarrollo	Puntos de estimación: 1
Programador responsable: Alexis De Armas Francisco	
Descripción: La HU comienza cuando se carga la imagen, a partir de este momento se puede modificar los valores de los pixeles en cada uno de los canales, rojo verde y azul.	

Tabla 14: Tabla de ingeniería de la funcionalidad de Manipulación de valores RGB.

Tarea	
Número de tarea: 9	Número de HU: 9
Nombre: Filtros de realce	
Tipo de tarea: Desarrollo	Puntos de estimación: 3
Programador responsable: Daimarys Laureiro González	
Descripción: La HU comienza cuando se carga la imagen, a partir de este momento se puede modificar la imagen para obtener un mayor nivel de detalle, aumentando los valores de intensidad de los bordes, delimitando mejor las estructuras con respecto al fondo.	

Tabla 15: Tabla de ingeniería de la funcionalidad de Filtros de realce.

Tarea	
Número de tarea: 10	Número de HU: 10
Nombre: Filtros de suavizado	
Tipo de tarea: Desarrollo	Puntos de estimación: 3
Programador responsable: Alexis De Armas Francisco	
Descripción: La HU comienza cuando se carga la imagen, el objetivo es eliminar los componentes de altas y bajas frecuencias que difieren con el comportamiento de los vecinos, eliminando la menor cantidad de bordes posible.	

Tabla 16: Tabla de ingeniería de la funcionalidad de Filtros suavizado.

4.1.3 Iteración 3

Esta iteración captura el tráfico USB, extrae las imágenes, graba imágenes y las exporta luego de realizadas las técnicas anteriormente mencionadas.

Historias de usuarios (HU)	Tiempo de implementación	
	Estimación	Real
Captura de tráfico USB	2	2
Extracción de imágenes del tráfico USB	3	3
Grabar imágenes de la lámpara de hendidura	2	2
Exportar imágenes en formato DICOM	3	3

Tabla 17: Tiempo de implementación de las HU de la iteración #3.

Tarea	
Número de tarea: 11	Número de HU: 11
Nombre: Captura de tráfico USB	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Programador responsable: Daimarys Laureiro González	
Descripción: Analiza todos los paquetes enviados a través de un puerto USB desde el equipo hacia la PC.	

Tabla 18: Tabla de ingeniería de la funcionalidad de Captura de tráfico USB.

Tarea	
Número de tarea: 12	Número de HU: 12
Nombre: Extracción de imágenes del tráfico USB	
Tipo de tarea: Desarrollo	Puntos de estimación: 3
Programador responsable: Alexis De Armas Francisco	
Descripción: Reúne los paquetes de datos relacionados con el envío directo de las imágenes provenientes del equipo médico y conforma las mismas.	

Tabla 19: Tabla de ingeniería de la funcionalidad de Extracción de imágenes del tráfico USB.

Tarea	
Número de tarea: 13	Número de HU: 13
Nombre: Grabar imágenes de la lámpara de hendidura	
Tipo de tarea: Desarrollo	Puntos de estimación: 2
Programador responsable: Daimarys Laureiro González	
Descripción: Convierte las imágenes obtenidas de la captura <i>en BufferedImage</i> almacenándolas en ficheros en formato de imágenes o video, según selección de usuario.	

Tabla 20: Tabla de ingeniería de la funcionalidad de Grabar imágenes de la lámpara de hendidura.

Tarea	
Número de tarea: 14	Número de HU: 14
Nombre: Exportar imágenes en formato DICOM	
Tipo de tarea: Desarrollo	Puntos de estimación: 3
Programador responsable: Alexis De Armas Francisco	

Descripción: Se convierte la imagen a formato DICOM (almacenando las características del equipo de captura y los datos del paciente) para su almacenamiento en ficheros.

Tabla 21: Tabla de ingeniería de la funcionalidad de Exportar imágenes en formato DICOM.

4.2 Interfaz del sistema

A continuación, se muestra la interfaz principal del sistema:

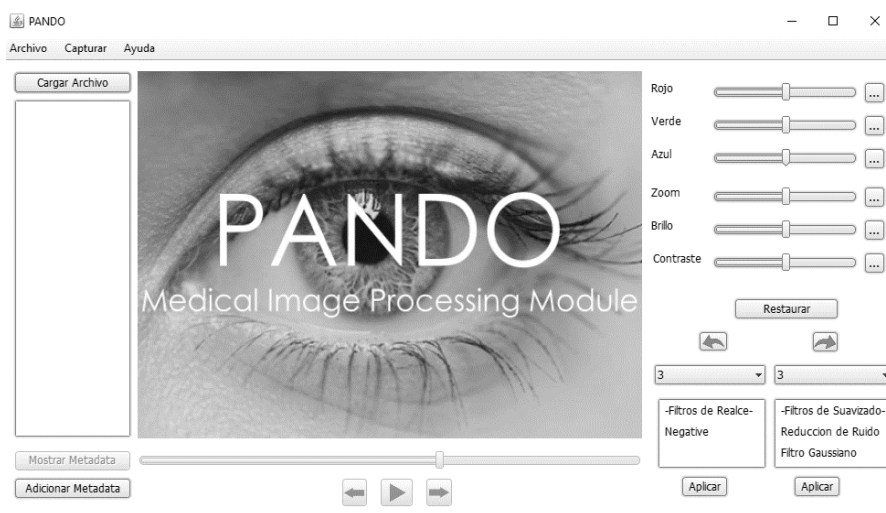


Ilustración 16: Interfaz principal.

A continuación, se muestra la interfaz de captura del sistema:



Ilustración 17: Interfaz de captura.

4.3 Validación del sistema

Los errores sencillos de un sistema que suelen estar ocultos, con el transcurso del desarrollo del mismo son más difíciles de detectar que los grandes errores que están a simple vista. Por esto es de suma importancia tener en cuenta la aplicación de las pruebas de software en las distintas etapas de desarrollo

del producto, pues las correctas aplicaciones de estas garantizan su calidad, provocando satisfacción al cliente y conformidad con lo realizado. Teniendo dominio sobre todos los procesos involucrados en el mismo, se aplican diferentes métodos y técnicas.

4.3.1 Pruebas de caja blanca

Las pruebas de caja blanca se centran en lo que se espera de un módulo, es decir, intentan encontrar casos de prueba en que el módulo no se atiene a su especificación. Esto se refiere a que se llevan a cabo para verificar el ajuste del sistema con los requerimientos determinados. Además, se enfocan especialmente en los módulos que se relacionan con la interfaz de usuario. No requieren el conocimiento de la estructura interna del programa para su puesta en marcha [57].

4.3.1.1 Prueba del camino básico

Para la aplicación de las pruebas de caja blanca se hizo uso de la técnica camino básico. El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Se toma como ejemplo el método `convert2BufferImage` de la clase `MyImage` como base para realizar la técnica del camino básico.

Partiendo del fragmento de código tomado se obtiene el siguiente grafo de flujo:

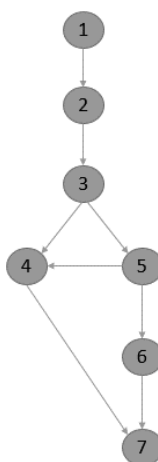


Ilustración 5: Representación del grafo de flujo de camino básico.

Luego se calculó la complejidad ciclomática $V(G)$, obteniendo el resultado siguiente:

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

$$V(G) = 8 - 7 + 2$$

$$V(G) = 3$$

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 3 caminos:

Camino	Entrada	Prueba	Salida
123	delta_zoom !=1 (false)	delta_zoom = 1	BufferedImage Original
1245	delta_zoom != 1 (true) delta_zoom > 1 (true)	delta_zoom = 2	BufferedImage Segmento
1243	delta_zoom != 1 (true) delta_zoom > 1 (false)	delta_zoom = 0.5	BufferedImage Original

Tabla 22: Tabla de caso de prueba de caja blanca.

4.3.2 Pruebas de caja negra

Las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. Estas permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa [58]. Hacen referencia a pruebas que se llevan a cabo sobre la interfaz del software sin tener en cuenta el código de la aplicación. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. Según [58] estas pruebas tienen como propósito detectar:

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y de terminación.

Para la aplicación se realizaron 11 diseños de casos de prueba (DCP) uno por cada requisito funcional definido, los cuales se basan en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica [58].

A continuación, se muestra el comportamiento de las no conformidades según las iteraciones de pruebas. En la primera iteración se detectaron un total de 6 no conformidades de las cuales 3 son significativas y 3 no significativas. En la segunda iteración se resolvieron las anteriores, encontrándose 3 nuevas no conformidades, de las cuales 1 es significativa y 2 no significativas, quedando resultas todas en la tercera iteración.

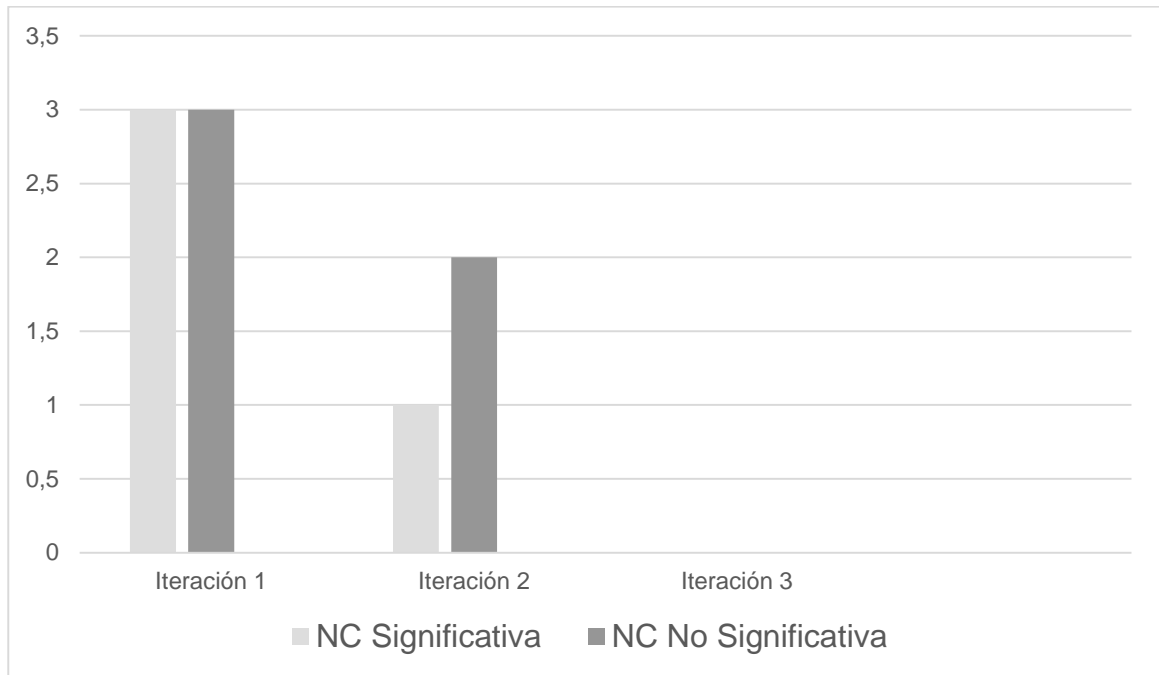


Tabla 23: Gráfico de barras que muestra las no conformidades por iteraciones.

4.3.3 Pruebas utilizando métricas de tamaño operacional de clase

Permiten medir de forma cuantitativa la calidad de los atributos internos del software. Esto permite al ingeniero evaluar la calidad durante el desarrollo del sistema.

Son varios los puntos de vista relacionados con la calidad del software. Las métricas de diseño a nivel de componentes se concentran en las características internas de los componentes del software con medidas que pueden ayudar al desarrollador a juzgar la calidad de un diseño a nivel de componente.

Las métricas se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software. Inclina sus objetivos a mejorar la comprensión de la calidad del producto, a estimar la efectividad del proceso y mejorar la calidad del trabajo.

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

- ✓ Responsabilidad: Un aumento del tamaño operacional de clase(TOC) implica un aumento de la responsabilidad asignada a la clase.
- ✓ Complejidad de implementación: Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
- ✓ Reutilización: Un aumento del TOC implica una disminución del grado de reutilización de la clase.

Luego de realizar la prueba, se obtuvieron los siguientes resultados:

- ✓ En la métrica responsabilidad los resultados revelaron un porcentaje de baja responsabilidad de 75%, de media responsabilidad de 17% y de alta responsabilidad de 8%. A continuación, se muestra el gráfico de pastel.

Responsabilidad	Cantidad de clases	Promedio
Baja	9	40,90909091
Media	2	9,090909091
Alta	1	4,545454545

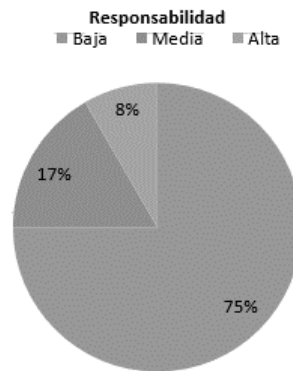


Tabla 24: Gráfico de pastel que muestra la métrica Responsabilidad.

- ✓ En la métrica complejidad los resultados revelaron un porcentaje de baja complejidad de 75%, de media complejidad de 17% y de alta complejidad de 8%. A continuación, se muestra el gráfico de pastel.

Complejidad	Cantidad de clases	Promedio
Baja	9	40,90909091
Media	2	9,090909091
Alta	1	4,545454545

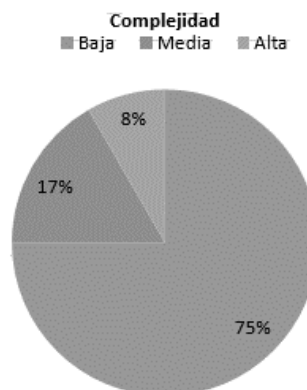


Tabla 25: Gráfico de pastel que muestra la métrica Complejidad.

- ✓ En la métrica reutilización los resultados revelaron un porcentaje de baja reutilización de 75%, de media reutilización de 17% y de alta reutilización de 8%. A continuación, se muestra el gráfico de pastel.

Reutilización	Cantidad de clases	Promedio
Alta	9	40,90909091
Media	2	9,090909091
Baja	1	4,545454545

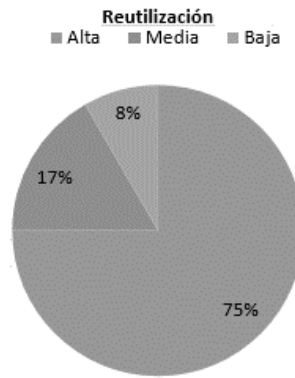


Tabla 26: Gráfico de pastel que muestra la métrica Reutilización.

4.4 Conclusiones del capítulo

Luego de realizadas las pruebas y las validaciones, se pueden arribar a las siguientes conclusiones:

- ✓ Con la validación realizada a la solución propuesta a partir de pruebas de aceptación y de pruebas unitarias empleando los métodos de caja negra y caja blanca, se pudo constatar después de 3 iteraciones, que el sistema quedó libre de no conformidades, por lo que el cliente puede utilizarlo en un entorno real.
- ✓ La métrica aplicada de tamaño operacional de clase obtuvo como resultado una responsabilidad baja, complejidad baja y reutilización alta; lo cual evidenció el correcto diseño de la aplicación.
- ✓ Después de realizada la prueba de aceptación el cliente emitió un aval, lo que evidenció su satisfacción con el producto desarrollado.

Conclusiones generales

- ✓ El software utilizado actualmente para la comunicación con la lámpara de hendidura utiliza el formato de compresión jpg, formato de compresión con pérdida, lo que provoca que se pierdan partes de los datos de la imagen, aunque esto en ocasiones sea invisible a simple vista. Por otra parte, las imágenes DICOM posibilitan almacenar los datos del paciente y del estudio realizado, además de almacenar secuencia de imágenes; lo cual lo hacen ideal para el almacenamiento de las imágenes provenientes de la lámpara de hendidura.
- ✓ La métrica aplicada de tamaño operacional de clase obtuvo como resultado una responsabilidad baja, complejidad baja y reutilización alta; lo cual evidenció el correcto diseño de la aplicación.
- ✓ Con la validación realizada a la solución propuesta a partir de pruebas de aceptación y de pruebas unitarias empleando los métodos de caja negra y caja blanca, se pudo constatar después de 3 iteraciones, que el sistema quedó libre de no conformidades y el cliente emitió un aval, lo que evidenció su satisfacción con el producto desarrollado.

Recomendaciones

- ✓ Se recomienda la implementación y prueba de filtros para el mejoramiento y restauración de las imágenes.
- ✓ Implementar algoritmos de superresolución para mejorar la calidad de las imágenes.
- ✓ Implementar e integrar los distintos algoritmos de segmentación.

Referencias Bibliográficas

- [1] I. e. a. HERNÁNDEZ LÓPEZ, «Estrategias de prevención de la opacidad de la cápsula posterior.,» *Revista Cubana de Oftalmología*, vol. 23, pp. 608-623, 2010.
- [2] R. Acosta, L. Hoffmeister, R. Roman, M. Comas, M. Castilla y X. Castells, «Revisión sistemática de estudios poblacionales de prevalencia de cataratas.,» *Archivos de la Sociedad Española de oftalmología*, vol. 81, nº 9, pp. 509-516, 2006.
- [3] M. ALVAREZ CANCIO, R. CRUZ DE LA OSA y I. HERNÁNDEZ LÓPEZ, «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*, vol. 8, pp. 545-558, 2016.
- [4] Y. González Iglesias, «Comportamiento de la calidad de vida relativa a salud antes y después de la cirugía de catarata.,» *Revista Cubana de Oftalmología*, vol. 20, nº 1, pp. 0-0, 2007.
- [5] E. P. V. R. R. P. S. H. G. Michel Alvarez Cancio, «Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM,» *Revista Cubana de Ciencias Informáticas*, vol. 11, nº 1, 2017.
- [6] H. S. J. R. C. G. Y. G. F. Hernández López Iván, «Estrategias de prevención de la opacidad de la cápsula posterior.,» *Revista Cubana de Oftalmología*, vol. 23, 2010.
- [7] P. D. Nuñez, *Realce y Restauración de imagen*, 2013.
- [8] B. Escalante, «Procesamiento Digital de Imágenes».
- [9] «Pagina oficial de DICOM,» [En línea]. Available: <http://medical.nema.org/>. [Último acceso: 6 febrero 2018].
- [10] W. González, *Digital Image Processing*, Prentice Hall, 2002.
- [11] V. Plataniotis, «Color image processing and applicatios,» *Springer Science & Business Media*, 2013.
- [12] H. Chen, «Optimal transform in perceptually uniform color space,» *Springer*, 2004.
- [13] P. S. C. Liu, «Comparative assessment of content-based face image retrieval in different color spaces,» *International Journal of Pattern Recognition and Artificial Intelligence*, 2005.
- [14] H. Cui, «Color-to-gray conversion using ISOMAP,» *The Visual Cumputer*, 2010.
- [15] N. P. S. More, «Adaptive local thresholding for detection of nuclei in diversity stained cytology images,» *Communications and Signal Processing*, 2011.
- [16] J. W. X. Shen, «Interactive two-scale color-to-gray.,» *The Visual Computer*, 2012.
- [17] K. M. B. R. L. de Queiroz, «olor to gray and back: Color embedding nto textured gray images,» *Color and Imaging Conference, Society for Imaging Science and Technology*, 2005.
- [18] N. A. D. Mohamad, «Microscopic RGB color images enhancement for blood cells segmentation in YCBCR color space for k-means clustering,» *Journal of Theoretical and Applied Information Technology*, 2013.

- [19] R. Orozco-Morales, Image Color Dimension Reduction. A comparative study of state-of-the-art methods, 2016.
- [20] A. Archarya, Image Processing Principles and Applications Tempe, Arizona JOHN WILEY & SONS, MC., 2005.
- [21] G. TOPCON, «<http://www.topcon-medical.es/>,» [En línea]. Available: <http://www.topcon-medical.es/es/products/302-dc-4.html#description>. [Último acceso: 06 12 2017].
- [22] V. Srinivasan y R. T. , Instrumentos y Equipos Oftalmológicos,Manual de Cuidado y Mantenimiento, India, 2003.
- [23] «Digital Eye Center,» [En línea]. Available: <https://www.digitaleyecenter.com/es/product/adaptador-camara-lampara-hendidura/>. [Último acceso: 25 05 2018].
- [24] «Oftálmica Andaluza,» [En línea]. Available: http://oftalmicaandaluza.com/dt_catalog/adaptador-de-movil-a-lampara-de-hendidura/. [Último acceso: 25 05 2018].
- [25] Topcon, «<http://www.topcon-medical.es/>,» 15 01 2018. [En línea]. Available: <http://www.topcon-medical.es/es/categories/27-diagnostico/#products/es/categories/27-diagnostico/30-lamparas-de-hendidura/>. [Último acceso: 15 01 2018].
- [26] F. a. notes, «Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers,» *AN991/D*, 2004.
- [27] F. N. GUERRERO, «Dispositivo modular para adquisición de biopotenciales con conexión USB.,» de *En XVIII Congreso Argentino de Bioingeniería SABI 2011*, Buenos Aires, 2011.
- [28] M. Suarez, «Competencias en TIC: Colección de Fascículos Digitales,» 2015. [En línea]. Available: http://competenciastic.educ.ar/pdf/lenguajes_de_programacion_1.pdf. [Último acceso: 04 12 2017].
- [29] O. B. Fernández, «Introducción al lenguaje JAVA,» 2004.
- [30] «Oracle. Java,» 2015. [En línea]. Available: Obtenido de <http://www.java.com/es/about/>. [En línea]: <http://www.java.com/es/about/>. [Último acceso: 04 12 2017].
- [31] I. H. López, Matlab y sus aplicaciones en las Ciencias y la Ingeniería, Madrid, 2010.
- [32] E. L. Fiter, Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesamiento digital de imágenes, Madrid, 2012.
- [33] Pressman, Ingeniería de Software, un enfoque práctico, 2002.
- [34] U. Monitor, «USB Monitor,» [En línea]. Available: <https://www.fabulatech.com/usb-monitor-pro.html>. [Último acceso: 20 03 2018].
- [35] B. & F. Tabachnick, Screening data prior to analysis, 2007.
- [36] A. Software, «<https://www.aggsoft.com/>,» [En línea]. Available: <https://www.aggsoft.com/usb-port-monitor.htm>. [Último acceso: 07 02 18].
- [37] U. Analyzer, «Free USB Analyzer,» [En línea]. Available: <https://freeusbalyzer.com/>. [Último acceso: 13 02 2018].

- [38] Wireshark, «Wireshark,» [En línea]. Available: <https://www.wireshark.org/>. [Último acceso: 20 03 2018].
- [39] R. C. Z. Martínez, «Repositorio Digital de la Universidad de Israel,» [En línea]. Available: <http://157.100.241.244/handle/47000/168>. [Último acceso: 15 02 2018].
- [40] S. D. Luz, «Redes Zone,» [En línea]. Available: <https://www.redeszone.net/2014/12/15/usbpcap-un-sniffer-para-windows-de-los-puertos-usb-de-nuestro-ordenador/>. [Último acceso: 19 02 2018].
- [41] F. S. P. Monitor, «Free Serial Port Monitor,» [En línea]. Available: <https://www.download3k.es/Desarrollo-Software/Depuracion/Download-free-Serial-Port-Monitor.html>. [Último acceso: 13 02 2018].
- [42] s. p. monitor, «Free Serial Port Monitor,» [En línea]. Available: <http://www.serial-port-monitor.com/>. [Último acceso: 07 02 2018].
- [43] M. A. S. Mendoza, Metodologías de desarrollo de software, 2004.
- [44] S. V. N. Juristo, Técnicas de evaluación de software, 2006.
- [45] J. Joskowics, Reglas y prácticas en eXtreme Programming, 2008.
- [46] J. R. D. Bustamante, Metodología Actual: Metodología XP, 2014.
- [47] K. L. R. Grau, Requirements engineering and agile development, 2010.
- [48] G. Méndez, Ingeniería de requisitos, Madrid, 2008.
- [49] Sommerville, Ingeniería de software, España, 2005.
- [50] K. Beck, Extreme Programming Explained, 1999.
- [51] H. R. S. Casas, Aspectos tempranos: un enfoque basado en Tarjetas CRC, 2009.
- [52] E. d. codificación, «Estándares de codificación,» [En línea]. Available: https://docs.google.com/document/d/1rbxDFM0zsbFDNRZeM2FoXfRDbYSiSt6tCdbYPA0qdzs/edit?hl=en_US#bookmark=id.6e6a203b40fe.. [Último acceso: 21 02 2018].
- [53] D. Mühlrad, Patrones de diseño, 2008.
- [54] R. H. J. V. E. Gamma, Elements of Design Patterns. Addison-Wesley, 1995.
- [55] C. Larman, UML y Patrones, Madrid, 2003.
- [56] J. Gracia, «Patrones de Diseño. Ingeniero Software, Análisis y Diseño,» 2005. [En línea]. Available: <http://www.ingenierosoftware.com/analisisydiseño/patrones-diseno.php>. [Último acceso: 27 03 2018].
- [57] M. E. M. M. J. T. J Gutierrez, «Preuabs del sistema en programación extrema,» Universidad se Sevilla, 2010.
- [58] R. S. Pressman, Ingeniería del Software. Un Enfoque Práctico., 2010.
- [59] M. Alvarez Cancio, R. Cruz de la Osa y I. Hernández López, «Aplicación de un sistema basado en casos para la,» *Revista Cubana de Informática Médica*, p. 546, 2016.

- [60] Y. González Iglesias, I. Zamora Galindo, Y. Fojaco Colina, B. Suárez Rodríguez y H. García Álvarez, «Comportamiento de la calidad de vida relativa a salud antes y después de la cirugía de catarata.,» *Revista Cubana de Oftalmología.*, 2007.
- [61] J. R. HERNÁNDEZ SILVA, «Resultados del Programa Nacional de Prevención de Ceguera por Catarata: Cuba 2000-2003.,» *Revista Cubana de Oftalmología*, vol. 17, nº 2, pp. 0-0, 2004.
- [62] K. V. Suaza, Mejora de historias de usuario y casos de prueba de metodologías ágiles con base TDD, 2015.
- [63] C. Larman, UML y patrones. Introducción al análisis y diseño orientado a objetos, México: Prentice Hall Hispanoamericana.
- [64] E. Gamma, R. Helm, R. Johnson y J. Vlissides, Design Patterns. Elements of Reusable Object-Oriented Software, 2003.
- [65] J. S. L. G. Carlos Guerrero, «Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web.,» 2013. [En línea]. Available: https://scielo.conicyt.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext. [Último acceso: 23 3 2018].
- [66] J. G. Agustin Yagüe, «Revista Española de Innovación, Calidad e Ingeniería del Software,» [En línea]. Available: <http://www.redalyc.org/html/922/92217159004/>. [Último acceso: 23 05 2018].
- [67] H. M. Schenone, «Diseño de una Metodología Ágil de Desarrollo de Software,» 2004.
- [68] L. P. P. M. C. Canós José H, «Metodologías Ágiles en el Desarrollo de Software,» 2003.
- [69] P. L. M. P. José Canós, «Metodologías Ágiles en el Desarrollo de Software,» Universidad Politécnica de Valencia, 2003.

Historias de usuarios

Historia de Usuario	
Número: 2	Nombre de la Historia de Usuario: Listar secuencia de imágenes
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando el usuario carga una imagen DICOM multiframe, quedan listadas los diferentes frames o la secuencia puede ser de una sola imagen.	
Observaciones: Las imágenes se hayan cargado satisfactoriamente y los formatos de las imágenes sean png, tiff, DICOM.	

Tabla 27: Historia de usuario 2 (HU2)

Historia de Usuario	
Número: 3	Nombre de la Historia de Usuario: Reproducir secuencia de imágenes
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro González	
Descripción: la HU comienza cuando quedan listadas los diferentes frames de la imagen, se da en el botón reproducir y se muestra la secuencia de imágenes, con la opción de pausar; ir al frame anterior o al siguiente.	

Observaciones: Debe haberse cargado una serie de imágenes

Tabla 28: Historia de usuario 3 (HU3)

Historia de Usuario	
Número: 4	Nombre de la Historia de Usuario: Añadir metadatos
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando el usuario carga una secuencia de imágenes o simplemente una imagen, se pueden añadir los metadatos, a partir del botón añadir metadatos.	
Observaciones:	

Tabla 29: Historia de usuario 4 (HU4)

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Visualizar metadatos.
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	
Descripción: la HU comienza cuando el usuario carga una secuencia de imágenes, se pueden visualizar los metadatos.	

Observaciones:

Tabla 30: Historia de usuario 5 (HU5)

Historia de Usuario	
Número: 6	Nombre de la Historia de Usuario: Zoom de la imagen.
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando se carga la imagen, a partir de este momento se puede ajustar el zoom para determinar un menor o mayor tamaño para cada uno de los pixeles.	
Observaciones:	

Tabla 31: Historia de usuario 6 (HU6)

Historia de Usuario	
Número: 7	Nombre de la Historia de Usuario: Ajuste de brillo y contraste
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	
Descripción: la HU comienza cuando se carga la imagen, a partir de este momento se puede aumentar o disminuir la intensidad de cada uno de los pixeles de la imagen.	

Observaciones: La secuencia debe estar en los formatos permitidos, que son (jpg, DICOM, png, tiff).

Tabla 32: Historia de usuario 7 (HU7)

Historia de Usuario	
Número: 8	Nombre de la Historia de Usuario: Manipulación de valores RGB
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando se carga la imagen, a partir de este momento se puede modificar los valores de los pixeles en cada uno de los canales, rojo verde y azul.	
Observaciones: La secuencia puede ser de una sola imagen y los formatos permitidos son (jpg, DICOM, png, tiff).	

Tabla 33: Historia de usuario 8 (HU8)

Historia de Usuario	
Número: 9	Nombre de la Historia de Usuario: Filtros de realce
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	
Descripción: la HU comienza cuando se carga la imagen, a partir de este momento se puede modificar la imagen para obtener un mayor nivel de detalle,	

aumentando los valores de intensidad de los bordes, delimitando mejor las estructuras con respecto al fondo.
Observaciones: Debe haberse cargado la imagen y el usuario haber seleccionado el tamaño del kernel (usa un tamaño de kernel de 3X3 por defecto).

Tabla 34: Historia de usuario 9 (HU9)

Historia de Usuario	
Número: 10	Nombre de la Historia de Usuario: Filtros de suavizado
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: la HU comienza cuando se carga la imagen, el objetivo es eliminar los componentes de altas y bajas frecuencias que difieren con el comportamiento de los vecinos, eliminando la menor cantidad de bordes posible.	
Observaciones: Debe haberse cargado la imagen y el usuario haber seleccionado el tamaño del kernel (usa un tamaño de kernel de 3X3 por defecto).	

Tabla 35: Historia de usuario 10 (HU10)

Historia de Usuario	
Número: 11	Nombre de la Historia de Usuario: Captura de tráfico USB
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	

Descripción: Analiza todos los paquetes enviados a través de un puerto USB desde el equipo hacia la PC.
Observaciones: Debe ser exitoso el proceso de handshaking.

Tabla 36: Historia de usuario 11 (HU11)

Historia de Usuario	
Número: 12	Nombre de la Historia de Usuario: Extracción de imágenes del tráfico USB
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
Descripción: Reúne los paquetes de datos relacionados con el envío directo de las imágenes provenientes del equipo médico y conforma las mismas.	
Observaciones: Debe ser exitoso el proceso de handshaking y de captura.	

Tabla 37: Historia de usuario 12 (HU12)

Historia de Usuario	
Número: 13	Nombre de la Historia de Usuario: Grabar imágenes de la lámpara de hendidura
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Daimarys Laureiro	

<p>Descripción: Convierte las imágenes obtenidas de la captura en <i>BufferedImage</i> almacenándolas en ficheros en formato de imágenes o video, según selección de usuario.</p>
<p>Observaciones: Deben haberse capturado las imágenes y debe haberse seleccionado por el usuario si desea hacer una captura de imagen o video (por defecto se marca la opción de captura de imagen).</p>

Tabla 38: Historia de usuario 13 (HU13)

Historia de Usuario	
Número: 14	Nombre de la Historia de Usuario: Exportar imágenes en formato DICOM
Modificación a la Historia de Usuario: 0	
Usuario: Michel Alvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 3 semanas
Riesgo en desarrollo: Medio	Puntos reales:
Programador responsable: Alexis De Armas Francisco	
<p>Descripción: Se convierte la imagen a formato DICOM (almacenando las características del equipo de captura y los datos del paciente) para su almacenamiento en ficheros.</p>	
<p>Observaciones: Debe haberse capturado la imagen y seleccionado DICOM como formato para el almacenamiento de la imagen.</p>	

Tabla 39: Historia de usuario 14 (HU14)

Tarjetas CRC

Clase: FormatConverter	
Responsabilidades	Colaboradores
Convert	ImageIO CreateADicomFile File MyImage

Tabla 40: Tarjeta CRC de la clase FormatConverter.

Clase: CreateADicomFile	
Responsabilidades	Colaboradores
generate	pixelmed MyImage

Tabla 41: Tarjeta CRC de la clase CreateADicomFile.

Clase: VentanaPrincipal	
Responsabilidades	Colaboradores
Inicializar Actualizar_Imagen	MyImage ImageIO ImageJ FormatConverter Thread

Tabla 42: Tarjeta CRC de la clase VentanaPrincipal.

Clase: MostrarMetadata	
Responsabilidades	Colaboradores
Mostrar metadata	ImageJ ImageICon

Tabla 43: Tarjeta CRC de la clase MostrarMetadata.

Clase: VentanaCaptura	
Responsabilidades	Colaboradores
VentanaCaptura	Thread MyImage ImageJ

Tabla 44: Tarjeta CRC de la clase VentanaCaptura.

Clase: AdicionarMetadata	
Responsabilidades	Colaboradores
Adicionar metadata	MyImage

Tabla 45: Tarjeta CRC de la clase AdicionarMetadata.