

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 3**



**Grupo de Investigación AIRI**

**Algoritmo para la segmentación de la opacidad de la cápsula posterior en imágenes en retroiluminación provenientes de la lámpara de hendidura.**

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

**Autor:**

Jean Carlos Madruga Chaviano

**Tutor:**

Ing. Michel Álvarez Cancio.

Lic. Reyder de la Cruz Osa

**La Habana, junio de 2016**

**“Año 58 de la Revolución”**

## DECLARACIÓN DE AUTORÍA

---

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2016.

---

**Jean Carlos Madruga Chaviano**

Autor

---

**Ing. Michel Álvarez Cancio**

Tutor

---

**Lic. Reyder de la Cruz Osa**

Tutor

## DATOS DE CONTACTO

---

### **Síntesis del Tutor**

El ingeniero Michel Alvarez Cancio graduado de la universidad de las ciencias informáticas en el año 2014 y el Licenciado Reydel de la Cruz Osa. En el año 2015 se crea el grupo de investigación AIRI y ellos forman parte del grupo donde se desarrollan temas de inteligencia artificial y otras materias.

## DEDICATORIA

---

*Con todo el amor de mi alma te dedico esta tesis a ti madre mía que siempre has sido mi fuente de luz y mi inspiración. A ti padre por ser mi guía en la vida y por enseñarme a ser un hombre de bien. Lo que hoy soy es el fruto del amor que siempre me han dado los quiero mucho y gracias por estar siempre a mi lado.*

## AGRADECIMIENTOS

---

*Hay momentos en la vida que nos marcan para siempre, hoy es uno de esos momentos, es mi culminación de una carrera universitaria que ha sido posible gracias a muchas personas y hoy les quiero agradecer*

*A ti madre por darme la vida, por darme una familia maravillosa, por ser guía, ejemplo, fuente de luz he inspiración, por ser mi confidente y por todo el apoyo que siempre me has dado. Quiero reiterarte que te amo mucho, que siempre estaré a tu lado y que vivo para hacerte feliz, gracias por tu amor. Te amo.*

*A ti padre por escoger como esposa a mi madre y juntos traerme al mundo, por luchar cada día por mi felicidad, por complacerme en todos mis caprichos, por tantos consejos que me has dado, por ser el mejor padre del mundo. Te amo.*

*A mis abuelos que, aunque no están vivos siempre están presente en mi recuerdo, ustedes todos, dieron sus vidas por hacerme sonreír cuando aún era un bebe, recuerdo las canciones que me cantaban, los dulces que hacían, y hasta la forma en que se vestían, pero lo que más recuerdo era esa sonrisa que se dibujaba en el rostro de cada uno de ustedes cuando me veían llegar. Donde quiera que estén gracias.*

*A mi tía Maldalena que tiene 92 años y sé que le encantaría estar aquí a mi lado, tú también eres más que especial para mí, solo deseo que tengas fuerzas para llegar a los 120 años y más porque te quiero con mi vida y daría todo por ti.*

*A mis padrinos Omar y Fe María, ustedes han sido mis segundos padres porque siempre que mi mamá y mi papá trabajan son ustedes quienes cuidan de mí, me han brindado su casa, su amor, su cariño, su comida y siempre estaré agradecidos con ustedes, gracias por sus consejos y por ayudarme en todo, gracias por estar presentes cuando más lo he necesitado. Los quiero muchísimo.*

*A mi prima Yaque por ser una persona maravillosa, por llevarme a los museos, por enseñarme la buena música, por cuidar de mí y por mecarme cuando niño para que me quedara dormido. Te quiero.*

## AGRADECIMIENTOS

---

*A mi tío Paito, a mi tío Lachi, a Misleidi, a mis hermanas, a mis primos y a toda esa gran familia que tengo que saben que siempre pueden contar conmigo para lo que sea, gracias por todo su apoyo.*

*A mis amigos, de la infancia, los del barrio, los de las escuelas y los que como Migue vienen conmigo desde que tenía 6 años, gracias por todos estos años de amistad y por toda la ayuda brindada.*

*Y a ti Wendy por ser la persona que ha compartido conmigo todo este tiempo, por conocer cada detalle de mí, por saber cómo tratarme, por darme tu amor incondicional y por cuidar de nuestro amor. Gracias por dejarme ser parte de tu vida y gracias por haber llegado a la mía. Te amo con todas las fuerzas de mi corazón y deseo que siempre seas parte de mi vida.*

### **RESUMEN**

En la actualidad la opacidad de la cápsula posterior es la complicación posoperatoria tardía más frecuente tras una cirugía de catarata exitosa, estando 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. Diagnosticarla depende de la experiencia que tenga el médico para identificar dicha complicación posoperatoria. En la universidad de las ciencias informáticas fue creado un software llamado PANDOC capaz de cuantificar la opacidad de la cápsula posterior en imágenes provenientes del Óculos Pentacam. PANDOC no trabaja con imágenes provenientes de la lámpara de hendidura que es el equipo más comúnmente utilizado para diagnosticar la opacidad de la cápsula posterior. En el presente trabajo de diploma se realizó una búsqueda de técnicas de segmentación de imágenes médicas usando el toolbox de procesamiento digital de imágenes de Matlab, permitiendo elaborar un algoritmo para la detección de la opacidad de la cápsula posterior en las imágenes en retroiluminación provenientes de la lámpara de hendidura. El algoritmo está compuesto por 3 etapas fundamentales: captura, mejora y segmentación de la imagen, y se ha implementado en el software PANDOC dotándolo de una herramienta capaz de identificar las estructuras de opacidad en las imágenes en retroiluminación. La calidad con que el algoritmo detecta la opacidad de la cápsula posterior ha sido calculada según la métrica de precisión, arrojando resultados de hasta un 96% y demostrando que puede ser utilizado en la medicina para la detección de dicha enfermedad posoperatoria.

**Palabras claves:** Opacidad de la capsula posterior, cirugía de catarata, software PANDOC, segmentación de imágenes

## **ABSTRACT**

---

### **ABSTRACT**

Currently the opacity of the posterior capsule is the postoperative complications more frequent late after a successful cataract surgery, being associated with decreased visual acuity, impaired contrast sensitivity and glare problems that have important social, medical and economic. Diagnose depends on the experience you have the physician to identify such postoperative complication. At the University of Information Science, he was created a software called Pandoc able to quantify the opacity of the posterior capsule in images from the Óculos Pentacam. Pandoc not work with images from the slit lamp is the equipment most commonly used to diagnose posterior capsule opacity. In this dissertation a search techniques segmentation of medical images using the toolbox of digital image processing of Matlab was made, allowing develop an algorithm for detecting the opacity of the posterior capsule in the images in backlight from the slit lamp. The algorithm is composed of three basic stages: capture, enhancement and image segmentation, and has been implemented in the Pandoc giving it a software tool capable of identifying the structures opacity backlit images. The quality with which the algorithm detects the opacity of the posterior capsule was calculated according to the measuring accuracy, yielding results of up to 96% and demonstrating that can be used in medicine for detection of said postoperative disease.

**Keywords:** Posterior capsule opacification, cataract surgery, PANDOC software, image segmentation.

<b>INTRODUCCIÓN</b> .....	8
<b>CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.</b> .....	13
1.1. Introducción. ....	13
1.2. Marco teórico. ....	13
<b>1.2.1.</b> Opacidad de la cápsula posterior.....	13
<b>1.2.2.</b> Lámpara de hendidura.....	13
<b>1.2.3.</b> Imagen digital. ....	14
<b>1.2.4.</b> Ruido digital.....	14
<b>1.2.5.</b> Procesado digital de imágenes. ....	15
1.2.6. Filtrado de imágenes. ....	16
1.2.6.1. Filtro de paso bajo.....	16
1.2.6.2. Filtro de paso alto.....	17
1.2.6.3. Filtro de la mediana.....	18
1.2.6.4. Selección de la técnica de filtrado. ....	20
<b>1.2.7.</b> Segmentación de Imágenes. ....	21
1.2.7.1. Segmentación Basada en el umbral.....	22
1.2.7.2. Segmentación basada en el borde. ....	24
1.2.7.2.1. Derivada de primer orden, el operador gradiente.....	24
1.2.7.2.2. Derivada de segundo orden. Operador Laplaciano.....	25
1.2.7.3. Segmentación Basada en la región.....	25
1.2.7.4. Selección de la técnica de segmentación a utilizar.....	26
1.3. Estado del arte. ....	29
<b>1.3.1.</b> Sistemas desarrollados para la identificación de OCP. ....	29
<b>1.3.1.1.</b> POCO. ....	29

# ÍNDICE

---

1.3.1.2.	EPCO.....	30
1.3.1.3.	AQUA.....	31
1.3.1.4.	AA.....	31
1.3.1.5.	Image J.....	32
1.3.1.6.	Programa Analizador de Opacidad Capsular: PANDOC.....	33
1.3.1.6.1.	Interfaz visual.....	33
1.3.1.6.2.	Funcionalidades.....	34
1.3.1.7.	Selección del sistema a utilizar.....	35
1.4.	Conclusiones parciales.....	35
<b>CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA</b>		<b>37</b>
2.1.	Introducción.....	37
2.2.	Metodología, Herramientas y Técnicas.....	37
2.2.1.	Metodología de desarrollo.....	37
2.2.2.	Entorno integrado de software.....	37
2.2.3.	Lenguaje de programación.....	38
2.2.4.	Herramienta para el modelado.....	39
2.3.	Historias de usuarios.....	39
2.4.	Tarjetas de Clase Responsabilidad Colaborador.....	45
2.5.	Requisitos del sistema.....	45
2.5.1.	Requisitos funcionales.....	45
2.5.2.	Requisitos no funcionales.....	46
2.6.	Descripción general de la propuesta.....	46
2.7.	Conclusiones parciales.....	52
<b>CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA</b>		<b>53</b>

# ÍNDICE

---

3.1. Introducción .....	53
3.2. Prueba de Software.....	53
3.2.1. Pruebas de unidad.....	53
3.2.2. Prueba de aceptación.....	56
3.3. Aplicación de la métrica para medir la calidad de la segmentación .....	58
3.4. Conclusiones parciales .....	63
<b>CONCLUSIONES GENERALES .....</b>	<b>64</b>
<b>RECOMENDACIONES .....</b>	<b>65</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>66</b>

## ÍNDICE DE TABLAS

---

### ÍNDICE DE TABLAS

Tabla 1 Historia de usuario: Cargar imagen .....	40
Tabla 2 Historia de usuario: eliminar ruido de la imagen. ....	41
Tabla 3 Historia de usuario: Cargar imagen. ....	42
Tabla 4 Historia de usuario: Guardar imagen. ....	42
Tabla 5 Obtener OCP en la imagen en retroiluminación.....	43
Tabla 6 Tarjeta CRC de la clase Inicial.....	45
Tabla 7 Iteraciones de las pruebas unitarias. ....	56
Tabla 8 Caso de prueba Aceptación HU5.....	56
Tabla 9 Muestra los resultados de la métrica Precisión .....	62

# ÍNDICE DE ILUSTRACIONES

---

## ÍNDICE DE ILUSTRACIÓN.

Ilustración 1 Imagen con ruido digital. ....	15
Ilustración 2 Mascaras que permiten el filtrado de paso bajo de una imagen. ....	17
Ilustración 3 (a) Imagen original, (b) Imagen con ruido y (c) Imagen con filtro paso bajo. ....	17
Ilustración 4 Mascara de convolución para el filtro paso alto. ....	18
Ilustración 5 (a) Imagen original, (b) Imagen luego de aplicada el filtro de paso alto. ....	18
Ilustración 6 Implantación de un filtro de la mediana. ....	19
Ilustración 7 (a) imagen original, (b) imagen contaminada con ruido impulsivo, (c) imagen con filtro de la mediana aplicada. ....	20
Ilustración 8 Imagen en retroiluminación proveniente de la lámpara de hendidura. ....	21
Ilustración 9 (a) imagen mostrando semilla inicial (b) ejemplo de región creciente. ....	26
Ilustración 10 OCP señalada manualmente. ....	27
Ilustración 11 Imagen en retroiluminación con método de segmentación por umbral aplicado. ....	27
Ilustración 12 Imagen en retroiluminación con método de segmentación por regiones aplicado. ....	28
Ilustración 13 Imagen en retroiluminación con método de segmentación en el borde aplicado. ....	28
Ilustración 14 Software POCO. ....	30
Ilustración 15 Software EPCO. ....	30
Ilustración 16 Software AQUA. ....	31
Ilustración 17 Software AA. ....	32
Ilustración 18 Software Image J. ....	32
Ilustración 19 Interfaz visual de PANDOC. ....	33
Ilustración 20 Diagrama de procesos del negocio vinculado con el proceso de detectar la OCP (Por elaboración propia). ....	47

## ÍNDICE DE ILUSTRACIONES

---

Ilustración 21 Imagen en retroiluminación proveniente de la lámpara de hendiduras. ....	48
Ilustración 22 Ejemplo de código Clahe.....	49
Ilustración 23 Imagen con método Clahe aplicado. ....	49
Ilustración 24 Imagen en escala de grises.....	50
Ilustración 25 Ejemplo de código magnitud del gradiente. ....	51
Ilustración 26 Imagen que muestra el resultado de aplicar Gradiente.....	51
Ilustración 27 Imagen del método Segmentar de la clase ImagenClass. ....	54
Ilustración 28 Grafo de flujo asociado al método Segmentar de la clase ImagenClass.....	54
Ilustración 29 Resultados de las pruebas de aceptación .....	58
Ilustración 30 Imagen OCP-1 segmentada por el especialista.....	61
Ilustración 31 Imagen OCP-1 segmentada por el algoritmo.....	61
Ilustración 32 Resultados de la métrica precisión .....	62

## INTRODUCCIÓN

---

### INTRODUCCIÓN

La opacidad de la cápsula posterior (OPC) es actualmente uno de los aspectos más importante en la cirugía de catarata de los tiempos modernos, siendo la complicación posoperatoria 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 (López et al. 2010).

La extracción extracapsular de la catarata (EECC) con implante de lente intraocular es la técnica quirúrgica preferida para el tratamiento de catarata en la actualidad. En ella, a diferencia de la extracción intracapsular de la catarata (EICC), se extrae del ojo únicamente el cristalino opacificado dejando la cápsula posterior intacta. Esta estructura constituye una barrera anatómica entre los segmentos anterior y posterior del ojo, que se reduce el riesgo de complicaciones en el segmento posterior después de la cirugía (como el edema macular cistoideo y el desprendimiento de retina). Sin embargo, dejar la cápsula intacta puede dar lugar a complicaciones a largo plazo, principalmente la opacificación de la cápsula posterior (OCP) (Findl et al. 2010).

Según López et al. (2010) la opacidad de la cápsula posterior tiene una gran incidencia en la actualidad, debido a que en el mundo hasta el 47,6 % de los pacientes operados de catarata tienen que regresar a las consultas médicas para atenderse la OCP, y específicamente en Cuba esta cifra llega hasta un 50 %. La OCP puede aparecer hasta 5 años después de una operación de catarata exitosa,

Según Ramos Pereira et al (2015) el diagnóstico de la OCP se realiza mediante la lámpara de hendidura. No obstante, la evaluación biomicroscópica de la extensión y severidad de la opacidad es subjetiva y está sujeta a amplia variación entre un observador y otro. Para obtener resultados confiables y comparables entre los múltiples estudios que sobre la prevención de la opacidad capsular se realizan, se hace sumamente necesaria la evaluación objetiva y cuantificación estandarizada del grado de OCP. Múltiples sistemas han surgido en años recientes que persiguen este objetivo mediante el análisis de imágenes en retroiluminación, así como de tomogramas tomados con la cámara rotatoria Scheimpflug del Pentacam. Sin embargo, no ha habido aún consenso en cuanto al método estandarizado universal a utilizar.

Por tales motivos en la Universidad de la Ciencias Informáticas (UCI) y con la colaboración del Instituto Cubano de Oftalmología "Ramón Pando Ferrer" fue creado un software llamado PANDOC (Programa Analizador de Opacidad Capsular) que provee al oftalmólogo de una herramienta por medio de la cual

## INTRODUCCIÓN

---

éste es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano). Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía posoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto obtiene las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas, aunque sólo trabaja con imágenes de tomogramas del Pentacam y tanto en el mundo como en Cuba el equipo más común en una consulta de oftalmología es la lámpara de hendidura. Otro problema de dicho software es que para una correcta cuantificación de la opacidad el especialista debe identificar manualmente los puntos que considera opacos, valoración que depende mucho de su nivel de experticia, y se desea evitar este grado de subjetividad. Dentro de las imágenes usadas en los pacientes operados de cataratas se encuentran las imágenes en retroiluminación provenientes de la lámpara de hendidura, las cuales dicho software no es capaz de procesar. Es por ello que surge como necesidad la identificación automática de Opacidad de la Cápsula Posterior (OCP) en estas imágenes en retroiluminación.

Por la situación antes descrita se define como **problema a resolver**: ¿Cómo detectar automáticamente las estructuras de opacidad capsular en imágenes en retroiluminación provenientes de la lámpara de hendidura?

La presente investigación se encuentra acotada al **Objeto de estudio**: Segmentación de imágenes y al **Campo de acción**: Procesamiento de imágenes en retroiluminación provenientes de la lámpara de hendidura.

Como **Objetivo general** se define: Desarrollar un algoritmo para la identificación automática de la OCP mediante imágenes en retroiluminación provenientes de la lámpara de hendidura.

### **Objetivos específicos:**

1. Elaborar el marco teórico de la investigación por medio del análisis de la literatura científica consultada para identificar líneas de trabajo.
2. Diseñar un algoritmo para la detección de opacidad de la cápsula posterior mediante imágenes de la lámpara de hendidura.
3. Implementar un algoritmo para la detección de opacidad de la cápsula posterior mediante imágenes de la lámpara de hendidura.

## INTRODUCCIÓN

---

4. Validar la propuesta de solución mediante pruebas de software y juicio de expertos en una muestra real para medir el funcionamiento del algoritmo.

**Idea a defender:** con el desarrollo de un algoritmo para la segmentación de la opacidad capsular, entonces el software PANDOC identificará automáticamente las estructuras de opacidad capsular en imágenes en retroiluminación provenientes de la lámpara de hendidura.

Planteada la idea a defender son identificables dos elementos (1) **variable independiente:** desarrollo de un algoritmo para la segmentación de la opacidad capsular, y (2) **variable dependiente:** el software PANDOC identificará automáticamente las estructuras de opacidad capsular en imágenes en retroiluminación provenientes de la lámpara de hendidura.

### Tareas a cumplir:

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de filtrado y mejora en imágenes médicas.
3. Estudio del uso de algoritmos de segmentación en imágenes médicas.
4. Descripción de los pasos a seguir para el uso del algoritmo propuesto.
5. Diseñar la solución propuesta en función de los requisitos especificados.
6. Implementación de las técnicas de filtrado y mejora de la imagen en retroiluminación.
7. Implementación del algoritmo de segmentación propuesto.
8. Integración de la solución al software PANDOC.
9. Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

Durante el desarrollo de la investigación es necesario utilizar varios **métodos científicos**, los cuales se mencionan a continuación.

**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 la identificación de la OCP en pacientes operados de catarata.

**Histórico-Lógico:** Permite estudiar y analizar la trayectoria y evolución de software de análisis de OCP en pacientes operados de catarata, para poder comprender el nivel de desarrollo alcanzado por este tema en el mundo, además con estas experiencias se podrá tener un mayor entendimiento y comprensión de este tipo de sistemas, basándose en las experiencias existentes.

## INTRODUCCIÓN

---

**Inductivo-Deductivo:** Se utiliza con el objetivo de inducir una serie de conocimientos referentes a la segmentación de imágenes, para luego arribar a razonamientos que conlleven a la deducción de conocimientos que pueden ser aplicables al problema en cuestión.

**Experimento:** Método empírico mediante el cual se realizarán experimentos y pruebas de los principales algoritmos de identificación, aprendizaje y cálculo de la opacidad de la cápsula posterior (OCP), para examinar los resultados en busca de escoger los más adecuados.

**Observación:** Mediante este método empírico se observarán los resultados obtenidos en la caracterización e identificación de los principales algoritmos utilizados, para poder decidir luego cuál o cuáles serán más adecuados.

El documento de tesis está estructurado por 3 capítulos, y a continuación se dará una breve descripción de los mismos.

En el **capítulo 1** se realiza un análisis de la literatura consultada y se hace una representación de los conceptos asociados con la investigación. Se enfatiza en los principales métodos para la segmentación de las imágenes y se expone lo que es el tratamiento de imágenes digitales. Se explican las técnicas existentes en el campo del tratamiento de imágenes, y se realiza un estudio de sistemas homólogos que detecten OCP.

En el **capítulo 2** se presenta un algoritmo para la segmentación de las imágenes provenientes de la lámpara de hendidura. También son tratadas las principales técnicas y herramientas que se han hecho necesaria utilizar para la implementación de dicho algoritmo.

En el **capítulo 3** se presenta una muestra con la cual se hacen la validación del algoritmo propuesto. Se muestran resultados reales obtenidos por el algoritmo y se presenta una comparación entre un experto en el campo de la oftalmología y el algoritmo presentado.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

### CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

#### 1.1. Introducción.

En este capítulo se abordará el procesamiento digital de imágenes (PDI) con cada una de sus etapas. Así como los principales conceptos asociados con la investigación. Se realizará un estudio de los principales sistemas que existen en el mundo para la detección de la OCP. También se explicarán las técnicas de filtrado y de segmentación de la imagen.

#### 1.2. Marco teórico.

##### 1.2.1. Opacidad de la cápsula posterior.

La opacidad de la cápsula posterior (OCP) es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de cataratas, para garantizar una buena recuperación visual del mismo es imprescindible la transparencia de esta estructura. Desafortunadamente no son pocos los casos en los que se desarrolla esta anomalía, la opacificación de la cápsula posterior repercute negativamente en el resultado visual del paciente (Alvarez Cancio 2014). En el campo de la oftalmología esto es conocido como OCP y constituye la complicación tardía más importante en la cirugía de cataratas en los tiempos modernos (López et al. 2010).

##### 1.2.2. Lámpara de hendidura.

La lámpara de hendidura es un equipo oftalmológico que sirve para tomar imágenes médicas del ojo humano. Cuando un oftalmólogo va a examinar a un paciente con la lámpara de hendidura el equipo proyecta un haz de luz que llega hasta el fondo del ojo y permite al oftalmólogo observar todas las partes del ojo humano. La lámpara provoca en las imágenes un efecto llamado: *"Imágenes de Purkinje"* que no es otra cosa que los reflejos provocados por el mismo equipo y que le permiten al oftalmólogo definir las diferentes regiones del ojo humano, la ausencia de algunos de los reflejos de Purkinje indica la región del ojo que está dañada.

Para detectar opacidad de la cápsula posterior se toman imágenes en retroiluminación provenientes de la lámpara de hendidura, que es una imagen digital del fondo del ojo capturada cuando el haz de luz es reflejado por el cristalino del ojo y cuentan con las siguientes propiedades:

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

1. Se encuentran en un espacio de color RGB lo que da niveles de intensidades desde 0 a 255 colores.
2. Tienen como formato JPG, lo que significa que la imagen obtenida no es la misma que la deseada, esto provocado por el algoritmo de compresión que utiliza dicho formato, que es un algoritmo de reducción con pérdida para reducir el tamaño del archivo (V. Gómez et al. 2012).
3. Tiene una profundidad de 24 bits.
4. Una resolución de 1024 x 768 píxeles.
5. Presenta una luz provocada por el mismo equipo que puede dificultar la obtención de la Opacidad.

### 1.2.3. Imagen digital.

El término imagen se refiere a una función bidimensional de intensidad de luz  $f(x, y)$  donde  $x$  e  $y$  denotan las coordenadas espaciales, y el valor de  $f$  en cualquier punto  $(x, y)$  es proporcional a la intensidad de la imagen en ese punto. Una imagen digital puede escribirse como una matriz cuyos índices de fila y columna indican un punto en la imagen y cuyo valor coincide con el nivel de intensidad de luz en ese punto. Cada elemento del array se corresponde con un elemento de la imagen y se le denomina píxel (Laorden Fiter 2012). Las imágenes digitales en ocasiones presentan ruido que puede ser provocado por el lente del equipo con que ha sido capturada.

### 1.2.4. Ruido digital.

El ruido digital son pequeñas anomalías en la imagen que no se corresponden con la realidad. En ocasiones el lente de la cámara con la que es tomada la imagen digital puede tener dificultades y presentar en la imagen diferentes alteraciones, a esto es lo que se le conoce como ruido. Cuando se adquiere una imagen digital y está contaminada por ruido, se debe en la mayoría de los casos al equipo de captación electrónico utilizado, en la bibliografía este ruido es llamado ruido de cuantificación de imagen o efecto niebla en la imagen (Giménez, Manguillot, y Escribá 2013). El daño ocasionado en la imagen por el ruido puede ser revertido con técnicas de procesamiento digital de imágenes que se encarguen de hacer una reconstrucción de la imagen original. La siguiente ilustración muestra un ejemplo de una imagen con ruido.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

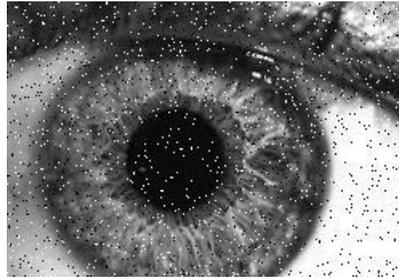


Ilustración 1 Imagen con ruido digital.

### 1.2.5. Procesado digital de imágenes.

El procesamiento digital de imágenes (PDI) es el procesado, entendiendo éste como el almacenamiento, transmisión y representación de información, de imágenes digitales por medio de una computadora digital. Según González et al (1996) el PDI consta con 4 etapas:

1. Obtención de la imagen
2. Pre-procesamiento de la imagen
3. Segmentación de la imagen.
4. Tratamiento automático de la imagen.

Con el objetivo de lograr un mejor entendimiento de las etapas antes presentadas resulta conveniente clasificar dichos procesos en categorías. De acuerdo con Laorden Fiter (2012) cada una de ellas con diferentes algoritmos involucrados. A continuación, se enuncian las cada una de estas categorías:

1. **Mejora o realce de la imagen:** Procesado de la imagen para que el resultado sea más apropiado para una aplicación en particular. Tareas habituales: mejora de la nitidez o aclarado de las imágenes desenfocadas, eliminación de ruido, mejora del contraste, mejora del ruido y detención de bordes.
2. **Restauración de la imagen:** Se puede considerar como revertir el daño ocasionado a la imagen por una causa conocida. Tareas habituales: eliminar el desenfoco por movimiento, eliminar distorsiones ópticas, eliminar interferencias periódicas, etc.
3. **Segmentación de la imagen:** Subdivide la imagen en parte o aísla ciertos objetos de una imagen. Tareas habituales: búsqueda y selección de formas determinadas en la imagen, máscaras de la imagen, etc.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Es importante destacar que no existe un método común que logre realizar correctamente el procesamiento de una imagen digital, por lo que se tiene que analizar según las características propias de la imagen en cuestión, cuál de los algoritmos existentes en cada una de la etapa planteada responde mejor nuestros intereses. De todas las etapas planteadas la segmentación de una imagen digital es la más importante debido a que de ella se obtendrán los objetivos deseados para su posterior análisis, por lo que una correcta selección de los mismos se hace imprescindible. La presente investigación se centra más en la etapa de la segmentación que en el resto por la importancia que conlleva la misma, aunque todas son respetadas como parte del proceso del tratamiento de una imagen digital.

### 1.2.6. Filtrado de imágenes.

Las técnicas de filtrado de imágenes no son más que métodos para resaltar o suprimir de forma selectiva información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o bien para ocultar valores anómalos. Según Cabezuelo el proceso de filtrado consiste en la aplicación a cada uno de los píxeles de la imagen de una matriz de filtrado de tamaño  $N \times N$  (generalmente de  $3 \times 3$  aunque puede ser mayor) compuesta por números enteros y que genera un nuevo valor mediante una función del valor original y los de los píxeles circundantes. El resultado final se divide entre un escalar, generalmente la suma de los coeficientes de ponderación.

Por las características que presentan las imágenes en retroiluminación provenientes de la lámpara de hendiduras se hace necesario aplicar técnicas de filtrados para la eliminación del ruido o el desenfoque que puede presentar dichas imágenes. Entre las técnicas de filtrado que se estudiarán están los filtros de paso bajo, los filtros de paso alto y los filtros de la mediana. Cada uno de ellos presenta diferentes técnicas, las cuales serán posteriormente analizadas.

#### 1.2.6.1. Filtro de paso bajo.

El filtro espacial paso bajo, deja el contenido de baja frecuencia inalterado mientras que atenúan los contenidos de alta frecuencia, este tipo de filtros resulta adecuado para atenuar ruido aditivo aleatorio presente en la imagen. En la ilustración 2 se muestran tres máscaras de convolución frecuentemente utilizadas para realizar el filtraje pasa bajo, una de las propiedades de tales máscaras es que la suma de todos sus valores debe ser igual a la unidad. Uno de los efectos que es necesario tener presente cuando se aplica este tipo de filtros, es que los mismos pueden introducir apreciable borrosidad en la imagen. En la ilustración 3 se muestra este tipo de procesamiento, en la ilustración 3a se muestra la

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

imagen original la cual ha sido contaminada con ruido aleatorio tal como se muestra en la ilustración 3b, por su parte en la ilustración 3c se muestra el resultado luego de procesar con un filtro pasa-bajo en donde se observa que si bien el contenido de ruido disminuye, sin embargo la borrosidad de la imagen aumenta (Botella et al. 2013).

$$\frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{10} \cdot \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 2 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Ilustración 2 Mascaras que permiten el filtrado de paso bajo de una imagen.

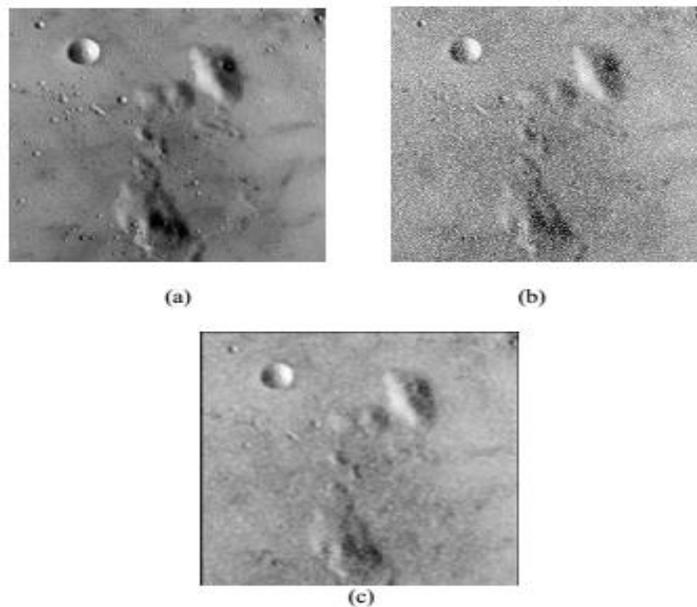


Ilustración 3 (a) Imagen original, (b) Imagen con ruido y (c) Imagen con filtro paso bajo.

### 1.2.6.2. Filtro de paso alto

Los filtros pasa alto, tienen la propiedad de acentuar los detalles de alta frecuencia de una imagen, normalmente los filtros pasa alto se utilizan cuando se quiere examinar objetos con alto contenido de

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

frecuencia espacial, como consecuencia de tal procesamiento, las porciones de una imagen que presentan componentes de alta frecuencia, serán resaltadas mediante la utilización de niveles de gris más claros, mientras que aquellas con componentes de baja frecuencia serán más oscuras, en este sentido, este tipo de filtro puede ser utilizado para reforzar los bordes presentes en la imagen. Uno de los efectos indeseados de estos filtros es que pueden acentuar el ruido de la imagen.

-1	-1	-1
-1	9	-1
-1	-1	-1

(a)

0	-1	0
-1	5	-1
0	-1	0

(b)

1	-2	1
-2	5	-2
1	-2	1

(c)

Ilustración 4 Mascara de convolución para el filtro paso alto.

En la Ilustración 4 se muestran tres máscaras de convolución para implantar el filtro pasa alto, por su parte en la Ilustración 5 se muestra el resultado de procesar una imagen con un filtro pasa alto, así en la Ilustración 5a se muestra la Ilustración original que al ser procesada utilizando la máscara de convolución mostrada en la Ilustración 4a, produce como resultado la imagen de la Ilustración 5b en donde se observa que efectivamente los bordes aparecen reforzados (Lovay et al. 2013).

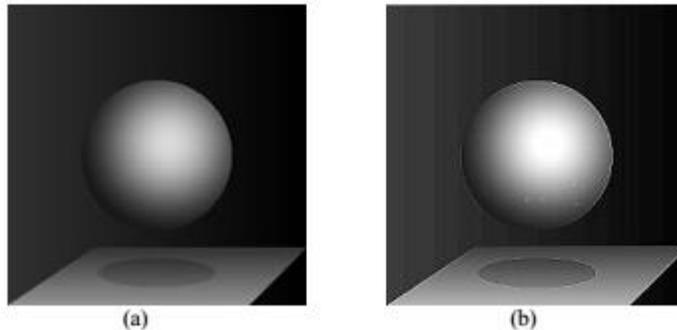


Ilustración 5 (a) Imagen original, (b) Imagen luego de aplicada el filtro de paso alto.

### 1.2.6.3. Filtro de la mediana.

El filtraje de mediana, es un procedimiento no-lineal, útil para reducir el ruido impulsivo y del tipo "sal y pimienta", muchas veces presente en las imágenes. El filtro de mediana utiliza los valores de los píxeles contenidos en una vecindad de tamaño impar, para determinar el nuevo valor del píxel de interés. El

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

procedimiento para ello, consiste en clasificar todos los píxeles incluidos en la ventana en orden creciente y sustituir el ubicado en el centro de la vecindad por el píxel mediano luego de la clasificación, es decir, si se tiene una secuencia discreta de tamaño  $N$  impar, entonces la mediana de tal secuencia, es aquel miembro de la secuencia, para el cual,  $(N-1)/2$  elementos son más pequeños o a lo sumo iguales y  $(N-1)/2$  elementos son más grandes (Molina et al. 2014).

En la Ilustración 6 se muestra un ejemplo de la implantación del filtro de mediana. El costo computacional más importante corresponde a la clasificación del arreglo en orden creciente, por lo cual es muy importante utilizar algoritmos optimizados para realizar tal tarea. En (Kak 1984) se describen varios algoritmos para realizar eficientemente tal clasificación. Una de las propiedades del filtro de mediana, es que el mismo tiende a preservar en lo posible la información referente a los contornos de la imagen, atenuando, por el contrario, variaciones impulsivas aleatorias, por lo cual se utiliza frecuentemente. En la Ilustración 7 se ilustra el filtraje de mediana así en la Ilustración 7a se muestra la imagen original, en la Ilustración 7b la imagen contaminada con ruido impulsivo y en la Ilustración 7c el resultado luego de procesar con un filtro de mediana de  $5 \times 5$ .

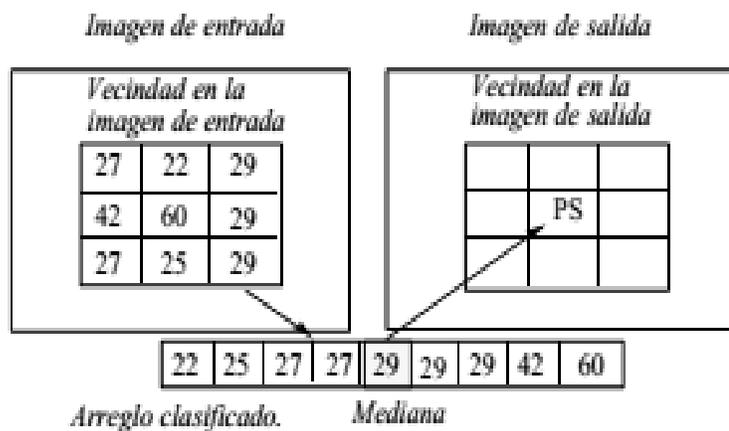


Ilustración 6 Implantación de un filtro de la mediana.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

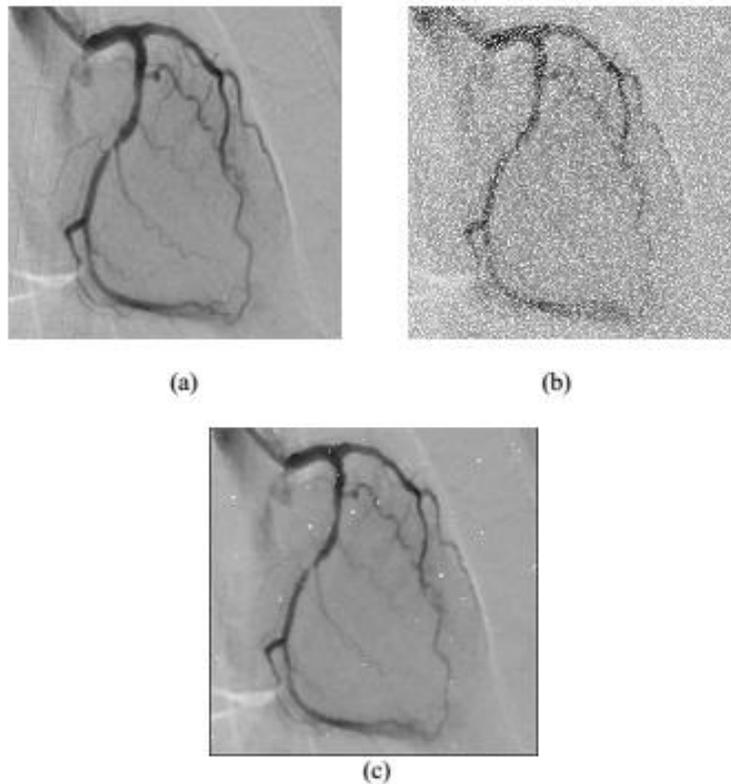


Ilustración 7 (a) imagen original, (b) imagen contaminada con ruido impulsivo, (c) imagen con filtro de la mediana aplicada.

### 1.2.6.4. Selección de la técnica de filtrado.

Las imágenes en retroiluminación tienen una apariencia rojiza donde los bordes son difíciles de identificar. También las áreas dañadas por la OCP puede que en la mayoría de los casos no sea totalmente visible al ojo humano. Por lo que se hace necesario resaltar los bordes y la luminosidad de las imágenes en retroiluminación. Atendiendo a estas necesidades se empleará en el algoritmo un filtro de paso alto para resaltar los bordes y la luminosidad de las imágenes. Teniendo en cuenta que son los filtros de paso alto los que mejores resultados obtienen cuando se desea resaltar detalles en las imágenes. La ilustración siguiente muestra un ejemplo de imagen en retroiluminación tomada con la lámpara de hendidura.



Ilustración 8 Imagen en retroiluminación proveniente de la lámpara de hendidura.

### 1.2.7. Segmentación de Imágenes.

La segmentación de imágenes es el área dentro del tratamiento de imágenes que se encarga de obtener objetos de importancia para su posterior análisis. Según Quintero Torres (2013), clásicamente, la segmentación de imágenes se define como la partición de una imagen en regiones constituyentes no solapadas, las cuales son homogéneas con respecto a alguna característica como una intensidad o una textura. Idealmente, un método de segmentación encuentra aquellos conjuntos que corresponden a distintas estructuras o regiones anatómicas de interés de la imagen. La segmentación de imágenes es el proceso de aislar o separar de la imagen los objetos considerados de interés para un problema en cuestión. En el caso de las imágenes en retroiluminación lo importante es señalar las regiones que se consideren opacidad de la cápsula (OCP) posterior. Pero según Rahim et al (2012) los métodos para llevar a cabo la segmentación varían ampliamente dependiendo de la aplicación específica, tipo de la imagen y otros factores. Los mismos pueden dividirse en 8 categorías:

1. Métodos de umbralización.
2. Métodos de región creciente.
3. Clasificadores.
4. Métodos de agrupamiento.
5. Modelos de campos aleatorios de Markov.
6. Redes neurales artificiales.
7. Modelos deformables.
8. Métodos guiados por plantillas.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Otras bibliografías hacen otro tipo de división de estos métodos tal es el caso de (Barba Guamán 2015) que los divide en 3 tipos:

1. Segmentación basada en el umbral.
2. Segmentación basada en el borde.
3. Segmentación basada en la región.

En dicha investigación se escoge la última división de los métodos de segmentación debido a que en la misma se ven más generales englobando todos de los métodos expuestos por (Rahim et al. 2012) y siendo esta la tendencia más usada en la actualidad.

### 1.2.7.1. Segmentación Basada en el umbral.

Según Barba Guamán (2015) este tipo de segmentación el más simple de los métodos. Una función o proceso se realiza en cada píxel de la imagen y el valor de salida se compara con un valor umbral establecido. Al píxel se le asigna una clase basada en qué lado del umbral cae su valor. Esto se convierte en una imagen de salida que contiene dos segmentos, uno que pasa el umbral y otro que no lo hace. Este método se esfuerza por identificar más de dos clases de regiones en una imagen.

Los umbrales actúan como separadores que permitirán decidir qué conjunto de tonos de gris pertenece a una determinada región. Estas técnicas son aplicadas sobre una imagen completa, y también pueden combinarse con otras durante el pre-procesamiento o post-procesamiento de la imagen, de manera que se obtengan mejores resultados.

De acuerdo a González et al (1996) la umbralización es cuando una imagen en gris es binarizada consiguiendo un umbral óptimo  $T$  y con ese valor se separan los píxeles en dos regiones, una de zonas claras y otra de zonas oscuras. En la umbralización hay dos posibles situaciones:

1. **Umbral único.** Se da cuando solamente hay dos regiones de píxeles, para separarlos se establece un único umbral  $T$ . Este tipo de umbral se obtiene fácilmente a partir de histogramas bimodales.
2. **Umbral multinivel.** Dada una imagen con varios objetos, para separarlos hace falta más de un umbral, de forma que los píxeles que se encuentren entre cada par de umbrales  $T_i$  y  $T_j$  representarán a un objeto. Los umbrales elegidos pueden ser de varios tipos, dependiendo de las características tenidas en cuenta para su elección.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

En Nobuyuki Otsu (1976) se presenta un método no paramétrico y no supervisado para la selección automática del umbral para segmentación de imágenes, propuesto para imágenes en escalas de grises. El umbral óptimo se selecciona mediante el método de discriminantes que maximiza la separabilidad de las clases resultantes en niveles de gris. Usa los momentos acumulativos de orden cero y uno del histograma de niveles de grises.

Años después extendieron el método de Otsu, proponiendo un método recursivo para la obtención de umbrales. Este enfoque segmenta en cada ciclo el objeto homogéneo más brillante de una imagen dada, dejando sólo los objetos homogéneos más oscuros luego de la última recursión. Este método fue posteriormente usado como base para el análisis de imágenes en color. En Nimbarte y Mushrif (2010) lo usaron como base, comparando el método de Otsu con el MCVT (Minimum Class Variance Thresholding), obteniendo como conclusión que Otsu sigue siendo más potente para una amplia gama de imágenes.

La selección del umbral basada en la entropía, fue primeramente propuesta por (Pun 1981) y luego desarrollada en varios papers (Luthon, Liévin, y Faux 2004). Este método maximiza a priori una función de evaluación, la cual es una entropía, determinada a posteriori. El criterio de entropía y entropía relativa (también conocida como distancia de información de (Kullback-Leibler) fue ampliamente desarrollado por varios autores. La entropía mide la incertidumbre de una fuente de información con un umbral óptimo, que se obtiene maximizando la entropía de Shannon; mientras que la última, mide la discrepancia de información entre dos fuentes con un umbral obtenido minimizando la entropía relativa.

Ambos métodos (Otsu y entropía de Shannon) son considerados métodos clásicos de testeo de imágenes y también son métodos base para desarrollo de otros (Besbes 2004), (Chang et al. 2006), (Nimbarte y Mushrif 2010).

Por último, se han propuesto optimizaciones multiobjetivo (u optimización Pareto), basándose en el hecho de que el umbralizado mediante la optimización de un solo criterio no suele conducir a valores que se comporten de manera óptima con una amplia gama de imágenes. Para ello, se buscan umbrales con varios métodos, que obviamente discreparán entre ellos, y se busca el óptimo mediante la optimización Pareto (Nakib, Oulhadj, y Siarry 2007).

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

De los métodos de umbral antes propuestos el método de Otsu es el que mejores resultados ha obtenido, por tanto, será el usado en la presente investigación. Aunque esto no quiere decir que será el método de segmentación que se expondrá en la propuesta de solución, debido a que es necesario probar que también funciona con las imágenes en retroiluminación proveniente de la lámpara de hendidura.

### 1.2.7.2. Segmentación basada en el borde.

El borde de una imagen se puede definir como una discontinuidad en el número de píxeles en una dirección colineal (los puntos se encuentran en la misma recta), al encontrar los bordes, éstos determinan los límites de cada segmento en la imagen para así poder identificar un objeto. Para encontrar estos bordes se tiene algunos métodos, incluyendo el método de Canny, Sobel y el algoritmo Laplaciano de Gauss. (Laorden Fiter 2012).

Los métodos de extracción de bordes de una imagen, se basan en la diferencia que experimenta una característica en dos regiones adyacentes y que indican la existencia de un borde. A la vez los bordes pueden clasificarse por su anchura, ángulo de su pendiente de variación, y las coordenadas de su punto medio. En general, se identifican diferentes modelos de bordes o contornos: línea, tipo escalón, tipo rampa y tipo tejado. Las discontinuidades son detectadas usando derivadas de primer y segundo orden, en el caso de derivadas de primer orden se utiliza el operador gradiente, mientras que en derivadas de segundo orden se utiliza el operador Laplaciano.

#### 1.2.7.2.1. Derivada de primer orden, el operador gradiente.

Según Palomino y Concha (2009) una de las técnicas más comunes para detectar un borde de dos dimensiones (2D), es el uso de derivadas de primer orden mediante el operador **gradiente**. El gradiente de una imagen es una función  $f(x, y)$  que se define como el vector bidimensional presentado en la ecuación 1:

$$G(F(x, y)) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} dF/dx \\ dF/dy \end{bmatrix} \quad (1)$$

El operador gradiente  $G$  alcanza su máximo valor en la dirección en que la variación es máxima, por lo tanto, puede definirse como la ecuación 2:

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

$$G(F(x, y)) = \left( [G_x(x, y)]^2 + [G_y(x, y)]^2 \right)^{\frac{1}{2}} \quad (2)$$

Es práctica común aproximar el gradiente mediante los valores absolutos, con el fin de conseguir un procesamiento más rápido a costa de un error despreciable, y la expresión se aproxima a la ecuación 3.

$$G(F(x, y)) = ([G_x(x, y)] + [G_y(x, y)]) \quad (3)$$

A partir de las ecuaciones 1 y 2, se deduce que el cálculo del gradiente de una imagen se basa en la obtención de las derivadas parciales  $dF/dx$  y  $dF/dy$  en cada posición de píxel. Las derivadas de primer orden permiten encontrar lugares en una imagen en donde la intensidad que cambia rápidamente es mayor en magnitud que un umbral especificado. Existen varios estimadores derivativos como los de Sobel, Prewit y Canny.

### 1.2.7.2.2. Derivada de segundo orden. Operador Laplaciano.

Otra posibilidad para la acentuación de bordes, consiste en usar la segunda derivada, en tal caso se usan los puntos de cruce por cero para realizar la detección de borde. La extensión de la segunda derivada unidimensional, la Laplaciano tiene la siguiente ecuación (Miró Pérez et al. 2016):

$$\Delta^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (4)$$

La Laplaciano puede ser utilizado para realizar el reforzamiento de bordes o bien para detectar contornos en base a los cruces por cero. En lo referente al reforzamiento de bordes, este operador es no direccional y en consecuencia puede reforzar los bordes independientemente de su dirección y de su pendiente.

### 1.2.7.3. Segmentación Basada en la región.

Este tipo de segmentación intenta dividir una imagen en regiones similares a través de un criterio basado en su semejanza u homogeneidad. Algunos ejemplos de estas medidas usan la textura, el color y la intensidad. Los valores de los criterios son diferentes para cada región y las regiones se caracterizan típicamente por una distribución gaussiana de los píxeles que se encuentran ahí. Algunos ejemplos de los algoritmos de segmentación basada en la región son: el umbral del histograma, los

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

métodos basados en gráficos, y la segmentación basada en la superficie (Daigavane y Bajaj 2010). Este método puede experimentar problemas con los segmentos, si los criterios son demasiado imprecisos o muy estrictos, lo que puede dar que los segmentos abarquen a más de un objeto o varios segmentos dentro de un objeto (Kee et al. 2014).

Otro método de segmentación de regiones es el método Región Creciente (region growing). Es una técnica para extraer regiones de la imagen que están conectadas según cierto criterio predefinido. Este criterio puede estar basado en información de intensidades y/o bordes de la imagen. En su forma más simple, este método requiere un punto semilla (seed point) que es seleccionado manualmente por el usuario, y extrae todos los píxeles conectados a la semilla, que tengan el mismo valor de intensidad. La ilustración 9 muestra un ejemplo donde se utiliza el método de región creciente para aislar una estructura de la Ilustración 9a.

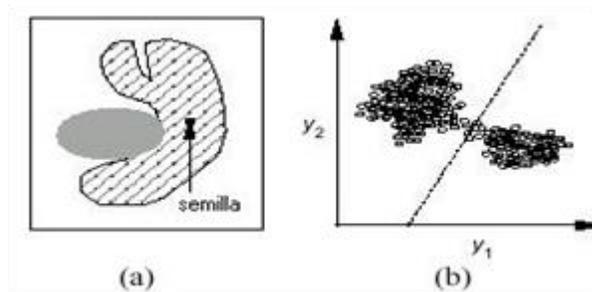


Ilustración 9 (a) imagen mostrando semilla inicial (b) ejemplo de región creciente.

Al igual que la umbralización, por lo general no se utiliza la región creciente solamente en una imagen, sino que se utiliza como parte de un conjunto de operaciones de procesamiento de imágenes, particularmente en la delineación de pequeñas y simples estructuras como tumores y lesiones. Su desventaja principal es que requiere interacción manual para obtener el punto semilla. Los algoritmos de división y mezcla (split and merge) están relacionados con la región creciente pero no requieren una semilla. La región creciente también puede ser sensible al ruido, causando que las regiones extraídas tengan agujeros e inclusive que se desconecten.

### 1.2.7.4. Selección de la técnica de segmentación a utilizar.

Atendiendo a la necesidad de obtener la OCP en las imágenes en retroiluminación se hace necesario buscar un método de segmentación que logre identificar dicha región. La OCP puede estar constituida

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

tanto por regiones circulares como lineales. La ilustración 10 muestra una imagen en retroiluminación donde se ha señalado por un especialista manualmente el área considerada OCP.

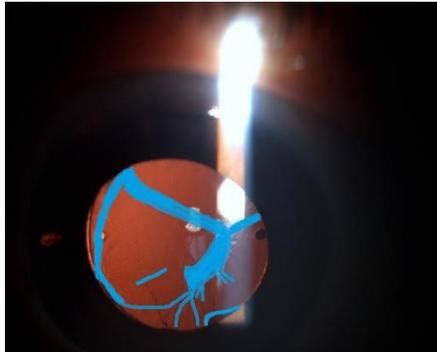


Ilustración 10 OCP señalada manualmente.

Las técnicas de segmentación basada en el umbral suelen dañar el área conformada por la OCP en la imagen en retroiluminación como se muestra en la siguiente ilustración.

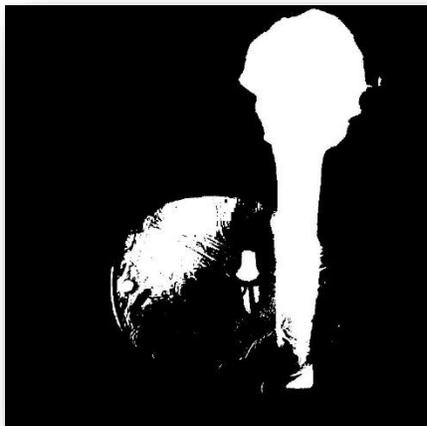


Ilustración 11 Imagen en retroiluminación con método de segmentación por umbral aplicado.

Mientras que las técnicas de segmentación por regiones y segmentación en el borde obtienen mejores resultados, tal y como se muestra en la ilustración 12 y 13 respectivamente. Siendo la segmentación por el borde la que mejor identifica las regiones consideradas OCP, por lo que se utilizará para formar parte del algoritmo.

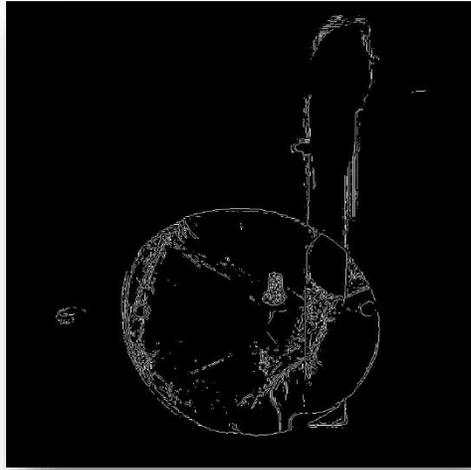


Ilustración 12 Imagen en retroiluminación con método de segmentación por regiones aplicado.

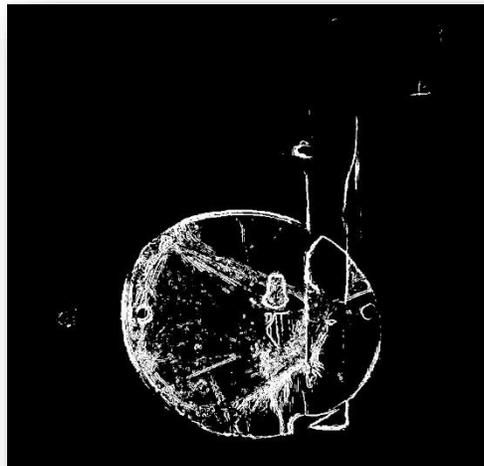


Ilustración 13 Imagen en retroiluminación con método de segmentación en el borde aplicado.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

### 1.3. Estado del arte.

#### 1.3.1. Sistemas desarrollados para la identificación de OCP.

Los sistemas más usados en la actualidad por el oftalmólogo que le permite visualizar la OCP son los sistemas basados en la lámpara de hendidura, también conocida como biomicroscopio. Es un instrumento de visualización directa de la estructura ocular, pero su limitación radica en que la evaluación del grado de opacidad de la cápsula posterior (OCP) la hace el médico de manera subjetiva según su apreciación visual, lo que puede variar según la experiencia del cirujano (Hernández López et al. 2011). Es por esta razón que algunos autores han desarrollado soluciones informáticas para lograr una evaluación objetiva y reproducible de esta complicación. A continuación, se describen algunos de estos sistemas, así como la influencia de la calidad de las imágenes en la obtención de resultados investigativos de elevada calidad.

##### 1.3.1.1. POCO.

POCO (Posterior Capsule Opacity): En este sistema las imágenes obtenidas son por retroiluminación, las mismas son evaluadas usando un análisis de los píxeles basado en la diferencia de texturas. Esta evaluación es semiobjetiva, ya que el examinador debe elegir el área afectada y clasificarla según su severidad: el programa divide la zona dentro de la capsulorrexis en 56 pequeños segmentos de igual área, el examinador debe marcar los segmentos que estén opacificados en más del 50% de su área y luego asignar a éstos un nivel de severidad (leve, moderado o severo). El resultado viene dado en forma de porcentaje de opacificación de 0 a 100%. Se calcula la severidad de la opacificación mediante la fórmula (Barman et al. 2000):

$$[(\text{área de grado 1} \times 1) + (\text{área de grado 2} \times 2) + (\text{área de grado 3} \times 3)] / \text{área total.}$$

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

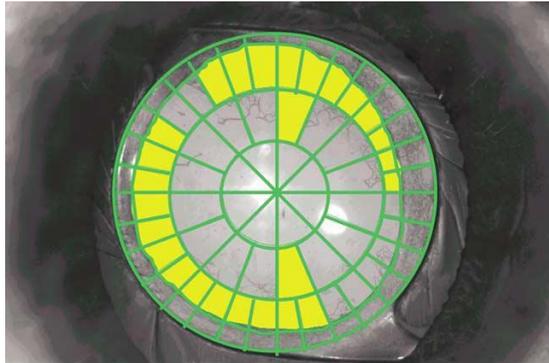


Ilustración 14 Software POCO.

## 1.3.1.2. EPCO

EPCO (Evaluación de la opacidad de la cápsula posterior) desarrollado por Tetz y Col, en 1997. Se obtienen imágenes de la lente por retroiluminación, se transfieren al programa, se marca el área a estudiar (algunos autores analizan el área tras la óptica, otros el área dentro de la capsulorrexis anterior y también es posible examinar la zona central comprendida por 3-4 mm), luego se remarcan las zonas opacificadas con el ratón y se clasifican subjetivamente en 4 grupos: mínimo, leve, moderado y severo. El índice de OCP viene dado al multiplicar el grado de opacificación por el área seleccionada. El resultado obtenido es un índice de 0 a 4. Al igual que el software POCO calcula la severidad de la opacificación mediante la fórmula (Barman et al. 2000):

$$[(\text{área de grado 1} \times 1) + (\text{área de grado 2} \times 2) + (\text{área de grado 3} \times 3)] / \text{área total}.$$



Ilustración 15 Software EPCO.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

### 1.3.1.3. AQUA.

AQUA (Automated Quantification of After-Cataract): Se obtienen imágenes con retroiluminación y se importan al programa. Se selecciona y analiza el área dentro de la capsulorrexis. Este programa está basado en texturas. Para calcular la no homogeneidad de la imagen se calcula la entropía (grado de desorden) de un mapa de bits. Es totalmente automático y no tiene pasos subjetivos. El resultado que se obtiene es un índice de 0 a 10 (Findl et al. 2003).

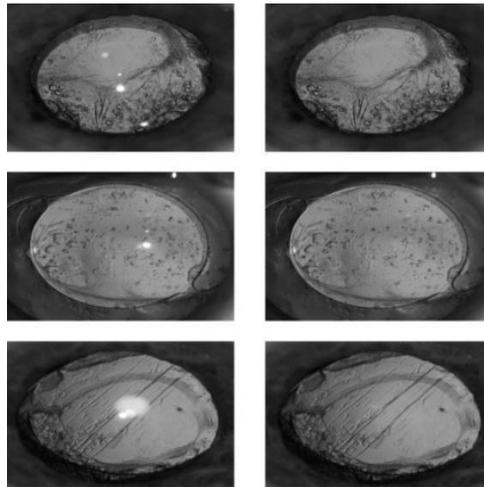


Ilustración 16 Software AQUA.

### 1.3.1.4. AA.

Este sistema fue diseñado y programado utilizando la plataforma de programación MatLab. Todo el diseño de la programación y la escritura de código fueron por TM Aslam y para mayor claridad el sistema se conoce por lo tanto como el análisis Aslam (AA) (Aslam, Patton, y Graham 2005). El primer problema encontrado fue uno de iluminación desigual en las imágenes. Incluso con las grandes áreas de reflejos de luz aberrantes retirados, utilizando la fusión con imágenes similares, pero vírgenes, una iluminación de fondo generalmente variables de toda la imagen puede causar errores en el análisis de imágenes. Aunque un sistema para la eliminación de registro basado de imágenes de luz está siendo probado y desarrollado por TM Aslam, todavía se necesita una curva de aprendizaje larga y sigue influyendo el efecto negativo de la iluminación de fondo.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

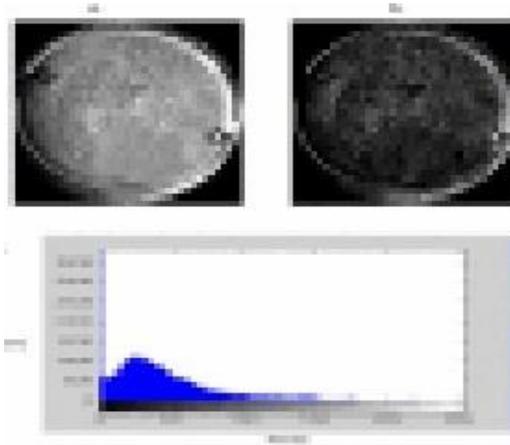


Ilustración 17 Software AA.

## 1.3.1.5. Image J.

Desarrollado por el National Institute of Health de Estados Unidos. Está basado en los cálculos realizados por Grewal y Jain, quienes determinaron que para los tomogramas del Pentacam un área de 4 mm corresponde a 16 752 píxeles en la imagen (Grewal et al. 2008). Se utilizan las herramientas Find Edges (hallar bordes) y Measure (medición) para calcular el valor promedio de la intensidad de píxeles del área seleccionada en los tomogramas Scheimpflug, como indicador de la opacidad capsular posterior, expresado en unidades (U). Esta herramienta es creada con otros fines, lo que hace que la curva de aprendizaje sea larga, además de que no diferencia los píxeles de fondo a la hora de dar el resultado.



Ilustración 18 Software Image J.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

## 1.3.1.6. Programa Analizador de Opacidad Capsular: PANDOC.

El software PANDOC es una herramienta para la cuantificación y análisis de la OCP. Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía postoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto toma las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas orientadas solo al uso de imágenes de tomogramas del Pentacam, lo cual hace más intuitivo la interfaz actual y la focaliza en su propósito principal (Alvarez Cancio 2014).

### 1.3.1.6.1. Interfaz visual.

A continuación, se describen los principales elementos que componen la interfaz visual del software PANDOC y las funcionalidades de los mismos, mostrados en la Ilustración 19:

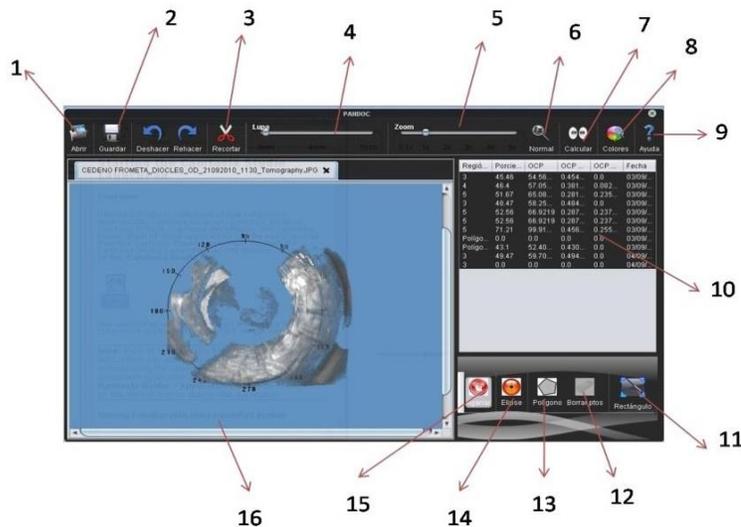


Ilustración 19 Interfaz visual de PANDOC.

1. (Botón abrir): Se utiliza para explorar los archivos y cargar la imagen médica resultante del tomograma de Scheimpflug del PENTACAM.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

2. (Botón guardar): Se usa para guardar esta imagen médica después de haber sido tratada por el programa, obteniendo solo el área de interés.
3. (Botón recortar): Se usa para obtener un segmento de la imagen después de haber sido seleccionado con la herramienta rectángulo.
4. (Herramienta lupa): Se utiliza para seleccionar la dimensión del área circular que será utilizada para el cálculo de la OCP.
5. (Herramienta zoom): Se utiliza para magnificar la imagen en la proporción seleccionada por el usuario.
6. (Botón normal): Se utiliza para llevar a la imagen a su tamaño original.
7. (Botón calcular): Se utiliza para realizar el cálculo de la OCP en el área de la imagen seleccionada por el usuario previamente.
8. (Botón color): Se utiliza para cambiar el color de las líneas de selección de área.
9. (Botón ayuda): Se utiliza para mostrar la ayuda del software.
10. (Panel resultados): Se utiliza para mostrar al usuario los cálculos realizados sobre la imagen cargada.
11. (Botón rectángulo): Se utiliza para realizar una selección rectangular de la imagen cargada.
12. (Botón borrar puntos): Se utiliza para borrar el último punto marcado por el usuario en una selección de área poligonal tras marcar el botón polígono.
13. (Botón polígono): Se utiliza para realizar una selección de área poligonal, uniendo los puntos de las referencias marcadas por el usuario mediante clics en la imagen.
14. (Botón elipse): Se utiliza para realizar una selección de área circular, mediante un área predefinida por el tamaño seleccionado en la herramienta lupa.
15. (Botón mano): Se utiliza para mover la imagen a través del panel imagen.
16. (Panel imagen): Se utiliza para mostrar la imagen y la selección de área realizada sobre esta.

A continuación, se describen las principales herramientas y funcionalidades con que cuenta el software PANDOC.

### 1.3.1.6.2. Funcionalidades.

1. Extraer de manera automática la porción relevante de las imágenes brindadas por los tomogramas del Pentacam. Esta funcionalidad permite abstraer al especialista de realizar dicha operación manualmente según su experiencia.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

2. Calcular el grado de la OCP sin incluir los píxeles del fondo. Esta funcionalidad permite que los píxeles de fondo no influyan de forma negativa en la obtención de la intensidad media de los píxeles.
3. Extracción de la intensidad media de píxeles de áreas irregulares de la imagen. Esta funcionalidad amplía la flexibilidad de la aplicación y le da una mayor maniobrabilidad al usuario a la hora de extraer la información que sea relevante al especialista.
4. Mostrar al usuario el porcentaje del área que fue seleccionada considerándose opacidad. Con esta funcionalidad se muestra una variación del color de fondo en las zonas que se consideran opacidad.
5. Determinar el centro de la estructura ocular, la cual mediante la identificación de variación de colores en los píxeles y trilateración celular, define automáticamente el centro de la estructura para funcionalidades posteriores.
6. Calcular el valor de la OCP. Con esta funcionalidad se calcula el valor de la OCP teniendo en cuenta la intensidad de píxeles de esa área multiplicado por el porcentaje de opacidad de la misma, esta expresa dicho valor también con respecto al centro y a la periferia de la estructura ocular.

### 1.3.1.7. Selección del sistema a utilizar.

Diversas son las plataformas existentes en el mundo para la detección de la OCP; pero debido a que la presente investigación forma parte de un grupo de investigación de la universidad de las ciencias informáticas en la cual fue desarrollado el software PANDOC, se escoge el mismo como herramienta a utilizar para desarrollar el algoritmo de segmentación. Con el objetivo de dotar a PANDOC de nuevas funcionalidades y poder extender su uso a las imágenes provenientes de la lámpara de hendiduras con la cual en la actualidad no puede trabajar.

### 1.4. Conclusiones parciales.

En este capítulo fue realizado un estudio de los principales conceptos asociados con la investigación, así como un análisis de los sistemas existentes en el mundo para la detección de la opacidad de la capsula posterior, y de las técnicas de filtrado y segmentación de imágenes. Concluyendo que el software PANDOC será utilizado como herramienta para detectar la OCP por las funcionalidades expuesta. También se concluye que las técnicas de segmentación por el borde obtienen mejores

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

resultados que el resto de las estudiadas y que el filtrado de paso alto ayudará a resaltar los bordes en las imágenes en retroiluminación.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

### CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

#### 2.1. Introducción

En este capítulo se presenta un algoritmo para la segmentación de imágenes en retroiluminación provenientes de la lámpara de hendidura. Dicho algoritmo será implementado en Matlab para luego ser vinculado con el software PANDOC utilizando tecnologías actuales de desarrollo de software.

#### 2.2. Metodología, Herramientas y Técnicas.

##### 2.2.1. Metodología de desarrollo.

Para (Sommerville y Galipienso 2005) las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de software. En el presente contexto se identifican dos tipos de metodologías de desarrollo: (1) ágiles y (2) tradicionales. La correcta selección de esta puede propiciar en la mayoría de los casos la creación de un software con la calidad requerida (J. M. M. Gómez, Marín, y Díaz 2014).

Atendiendo a la necesidad de elaborar un algoritmo en un período de tiempo relativamente breve, con un número reducido de implicados en su realización, obteniéndose la menor cantidad de artefactos y documentos y tener en consideración la variación de los requisitos propuestos se propone el empleo de una metodología ágil para la presente investigación. De acuerdo con la revisión de la literatura científica consultada han sido consideradas las metodologías de desarrollo ágiles siguientes: Extreme Programming (XP por sus siglas en inglés), Scrum (Kniberg 2015) y **AUP variación UCI**; siendo esta última la adoptada puesto que sólo define en su escenario cuatro la consecución de historias de usuario, en lugar de modelar un negocio minuciosamente definido, tal y como ocurre en este caso. También se ha tenido en cuenta que es **AUP variación UCI** la metodología de desarrollo por la que opta la universidad de las ciencias informáticas (Área donde se desarrolla la investigación en curso) para el desarrollo de sus proyectos productivos.

##### 2.2.2. Entorno integrado de software.

**Matlab** es un paquete de software para el desarrollo de algoritmos, el análisis de datos, la visualización y el cálculo numérico de datos y matrices. El mismo cuenta con un toolbox de procesamiento digital de imágenes que tiene implementado un conjunto de funciones que amplían las capacidades de Matlab

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

de desarrollar algoritmos y aplicaciones en el campo del análisis digital de imágenes (Laorden Fiter 2012). Matlab es un software matemático que ofrece un entorno de desarrollo (IDE), con un lenguaje de programación propio (lenguaje M). El mismo está disponible en diferentes plataformas como Unix, Windows y Mac OS X.

Matlab ha sido utilizado junto a su **toolbox** de procesamiento de imágenes para implementar el algoritmo propuesto en el **epígrafe 2.6** y realizar las pruebas para determinar que método de segmentación de imágenes obtiene mejores resultados. Con el toolbox se hace más fácil implementar y probar la amplia gama de métodos de segmentación que existen en la actualidad. Gracias a las implementaciones presentes en el mismo el desarrollo del algoritmo propuesto se hace más factible. Matlab también sirve para hacer las validaciones del algoritmo propuesto debido a que en él se puede implementar la métrica **precisión** que se utiliza para medir la calidad de la segmentación

Por otra parte, se precisa el empleo de **Netbeans** puesto que PANDOC es un software que ha sido desarrollado en este entorno de desarrollo sobre el lenguaje de programación Java y sobre el cual será incluido el algoritmo propuesto. Netbeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Según Jiménez, J (2014) Netbeans es una herramienta que se utiliza para desarrollar aplicaciones Web, móviles y de escritorio para diferentes lenguajes de programación como son Java, C++, Ruby, PHP y otros. Es de código abierto, es multiplataforma, multilenguaje, contiene servidores web y es fácil de instalar y utilizar.

### 2.2.3. Lenguaje de programación.

**Java** es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de Lenguaje de Programación C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Dentro de las características fundamentales de Java se encuentran:

**Simple:** elimina la complejidad de otros lenguajes como C y se enfoca en el contexto de los lenguajes orientados a objetos.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

**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 implementado 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.

### 2.2.4. Herramienta para el modelado.

Como herramienta para el modelado de la solución se ha seleccionado **Visual Paradigm** for 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. Entre las bondades de VP destacan: ofrecer un conjunto completo de herramientas; ofreciendo a los equipos de desarrollo de software todo lo necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos, así como brindar 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.

### 2.3. Historias de usuarios.

Las historias de usuario son la técnica que define la metodología ágil adoptada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. A continuación, se describen las historias de usuarios existentes en la presente investigación para así tener una mejor comprensión de lo que se debe hacer.

Tabla 1 Historia de usuario: Cargar imagen

Historia de usuario	
<b>Número:</b> 1	<b>Nombre del requisito:</b> Cargar Imagen
<b>Programador:</b> Jean Carlos Madruga Chaviano	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> <i>Alta</i>	<b>Tiempo Estimado:</b> 5 días
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 1 semana
<b>Descripción:</b> Cuando el usuario accede al programa para hacer uso de la segmentación necesita una imagen. Por lo que se hace necesario poder cargar imágenes en retroiluminación provenientes de la lámpara de hendidura.	
Observaciones:	
Prototipo de interfaz:	

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

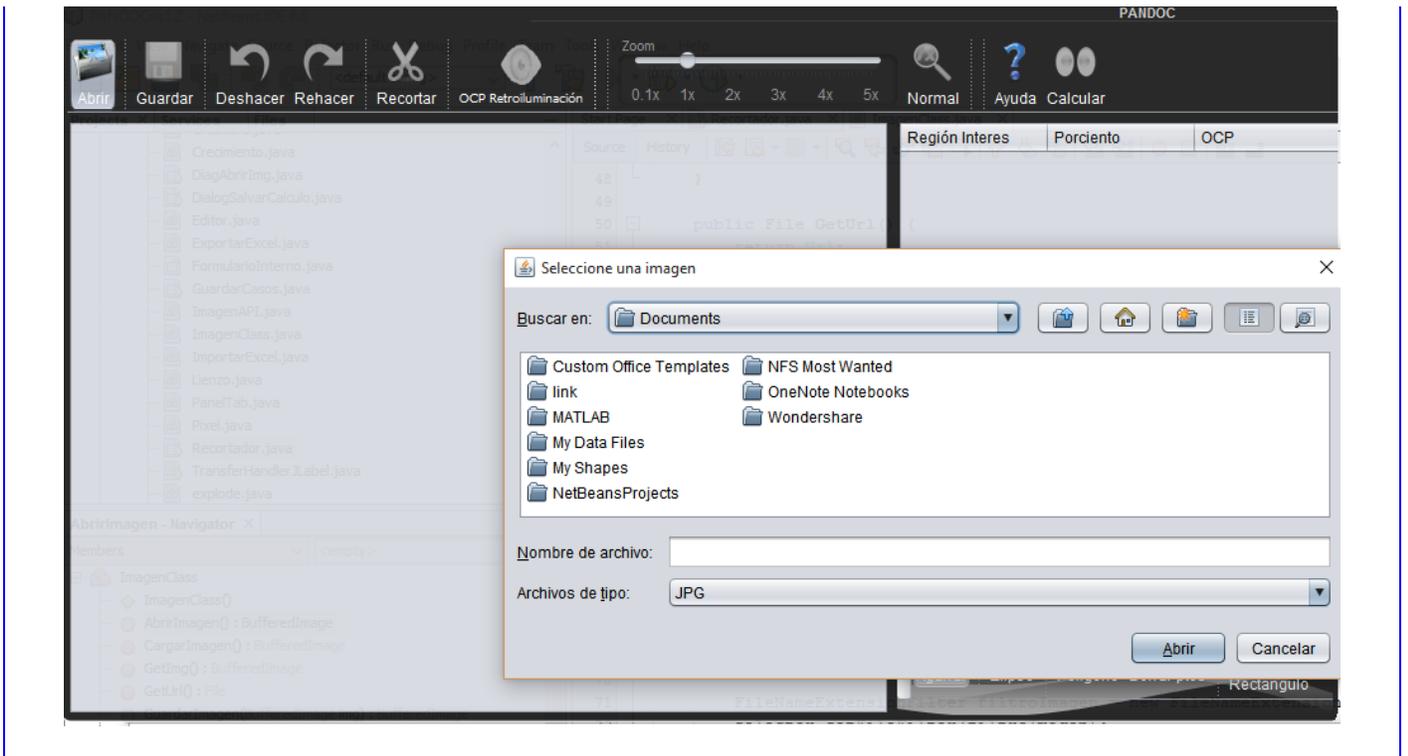


Tabla 2 Historia de usuario: eliminar ruido de la imagen.

Historia de usuario	
<b>Número:</b> 2	<b>Nombre del requisito:</b> Eliminar Ruido de la Imagen
<b>Programador:</b> Jean Carlos Madruga Chaviano.	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Tiempo Estimado:</b> 30 días
<b>Riesgo en Desarrollo:</b> Medio	<b>Tiempo Real:</b> 3 semanas
<b>Descripción:</b> Esta tarea tiene por objetivo identificar el ruido presente en las imágenes y poder eliminarlo para poder hacer una mejor segmentación que logre detectar con mayor eficiencia la OCP.	

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

Observaciones:
Prototipo de interfaz:

Tabla 3 Historia de usuario: Cargar imagen.

Historia de usuario	
<b>Número: 3</b>	<b>Nombre del requisito:</b> Mejorar de la imagen.
<b>Programador:</b> Jean Carlos Madruga Chaviano.	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> <i>Alta</i>	<b>Tiempo Estimado:</b> <i>30 días</i>
<b>Riesgo en Desarrollo:</b> Alto	<b>Tiempo Real:</b> <i>3 semanas</i>
<b>Descripción:</b> Esta actividad tiene como objetivo mejorar la calidad visual de la imagen. Mejorar el brillo presente en la misma y lograr una imagen que contengan mayor nitidez.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tabla 4 Historia de usuario: Guardar imagen.

Historia de usuario	
<b>Número: 4</b>	<b>Nombre del requisito:</b> Guardar imagen.
<b>Programador:</b> Jean Carlos Madruga Chaviano.	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> <i>Alta</i>	<b>Tiempo Estimado:</b> <i>20 días</i>

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

**Riesgo en Desarrollo:** Alto

**Tiempo Real:** 3 semanas

**Descripción:** Esta actividad tiene como objetivo darle al usuario la opción de poder guardar una imagen que se encuentre en el sistema en cualquier ubicación de la PC. La imagen se guarda en formato JPG.

**Observaciones:**

**Prototipo de interfaz:**

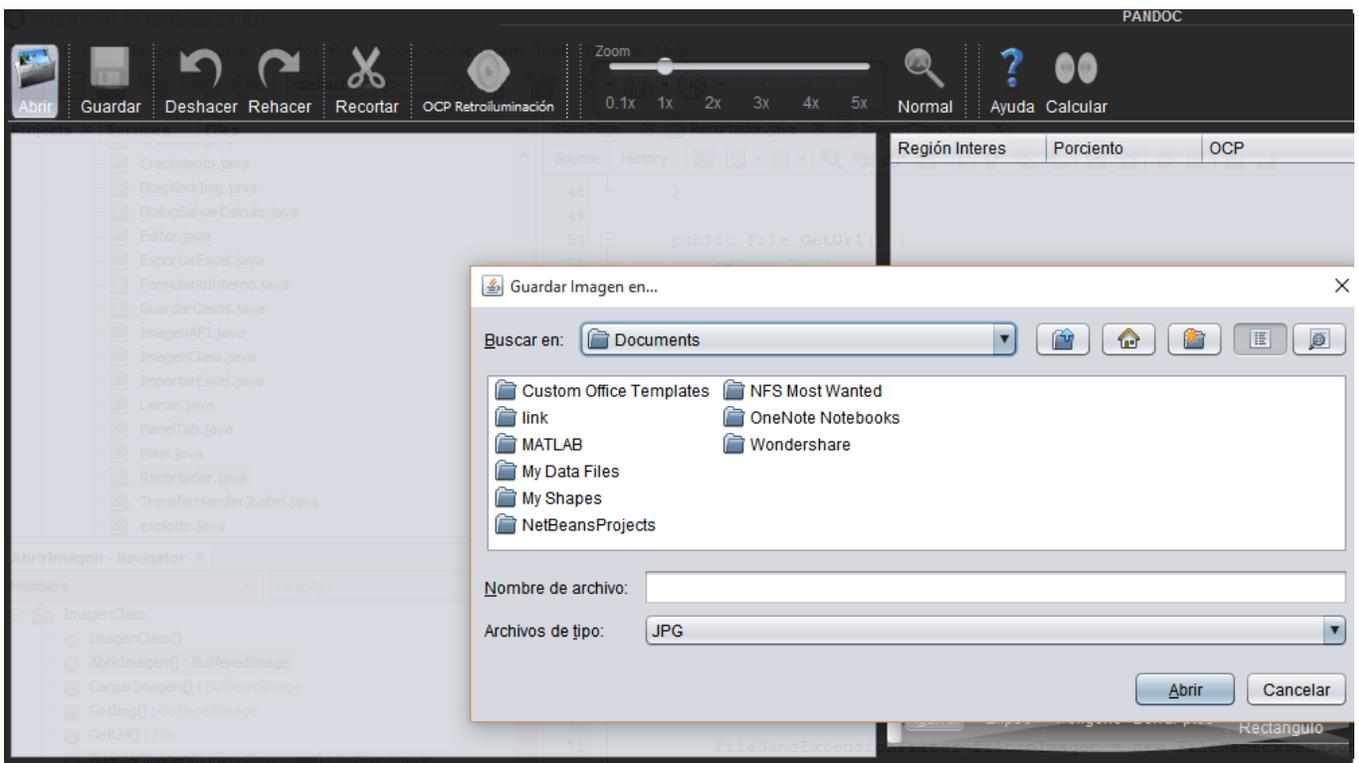


Tabla 5 Obtener OCP en la imagen en retroiluminación

Historia de usuario	
<b>Número:</b> 5	<b>Nombre del requisito:</b> Obtener OCP en la imagen en retroiluminación.
<b>Programador:</b> Jean Carlos Madruga Chaviano.	<b>Iteración Asignada:</b> 1

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

**Prioridad:** *Alta*

**Tiempo Estimado:** *30 días*

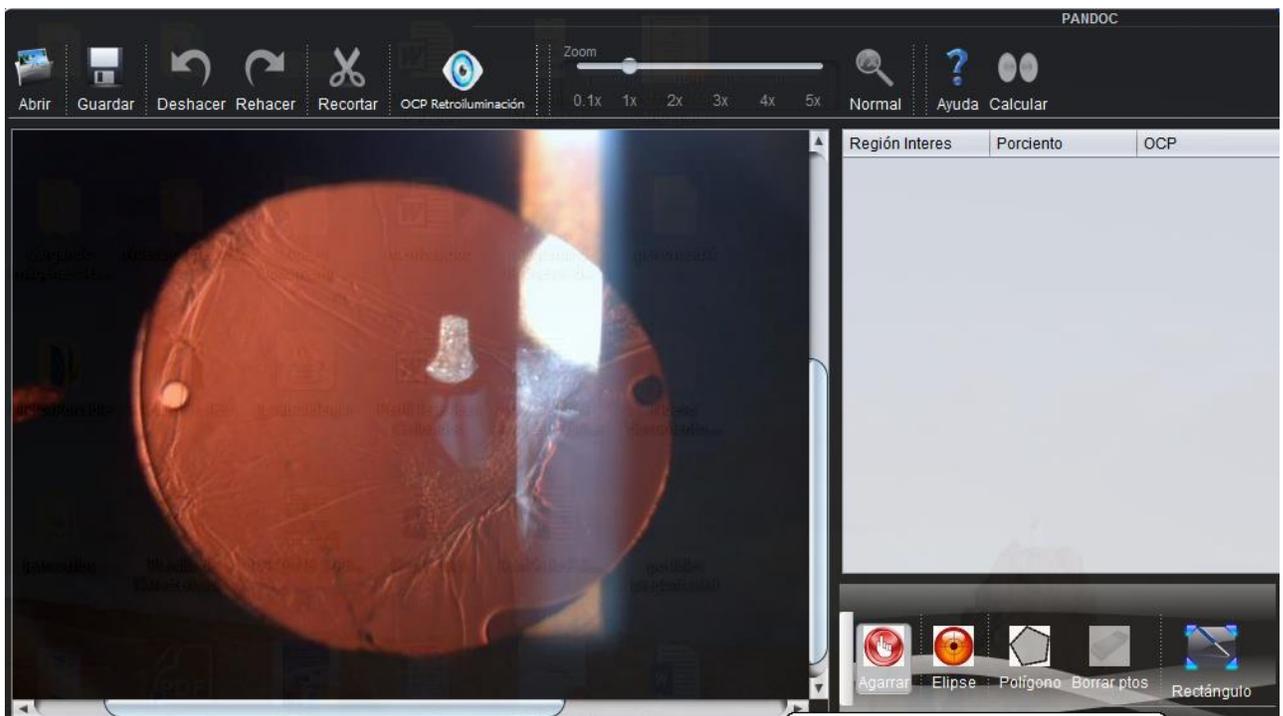
**Riesgo en Desarrollo:** *Alto*

**Tiempo Real:** *3 semanas*

**Descripción:** Esta actividad es el pilar de todo el proyecto, puesto que de su éxito depende el éxito de toda la investigación. Una vez la imagen ha sido pre-procesada estará lista para que sea segmentada. Es actividad tiene que lograr obtener en la imagen los objetos que sean opacidad de la cápsula posterior (OCP).

**Observaciones:**

**Prototipo de interfaz:**



A continuación, se explica cada uno de los datos que deben ser llenados en una HU:

1. **Número:** identificador de la HU.
2. **Nombre:** nombre que identifica a la HU.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

3. **Prioridad en el negocio:** prioridad de la HU con respecto al resto de las HU (alta, media o baja).
4. **Riesgo en desarrollo:** riesgo en la implementación de la HU (alto, medio o bajo)
5. **Descripción:** descripción sintetizada de la HU.

### 2.4. Tarjetas de Clase Responsabilidad Colaborador.

Las tarjetas de Clase-Responsabilidad-Colaboración (CRC) permiten ver las clases no como un depósito de datos, sino que permiten conocer el comportamiento de cada una en un alto nivel. La metodología XP estipula su uso como un artefacto obligatorio durante el desarrollo de un proyecto, debido a los beneficios que aportan a los desarrolladores (Pressman, Roger S 2014). A continuación, se muestra la tarjeta CRC de la clase Inicial y el **Anexo 1** muestra el resto de las tarjetas.

Tabla 6 Tarjeta CRC de la clase Inicial.

Nombre de la clase: Inicial	
Responsabilidades	Clases Relacionadas
Esta clase es la encargada de brindar los enlaces a las diferentes módulos dependiendo del equipo oftalmológico que se esté utilizando Óculos Pentacam o Lámpara de hendiduras.	Recortador. Principal.

### 2.5. Requisitos del sistema.

La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional (Sommerville y Galipienso 2005).

#### 2.5.1. Requisitos funcionales.

Los requisitos funcionales con los que cuenta el sistema son los siguientes:

1. Cargar imagen.
2. Guardar imagen.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

3. Obtener OCP en la imagen en retroiluminación.
4. Mejorar imagen.
5. Eliminar ruido en la imagen.

### 2.5.2. Requisitos no funcionales.

**Requisito de software:** el usuario debe tener instalada la máquina virtual de java en la versión 1.6 o una más moderna y el Compiler Runtime de Matlab versión 8.1.

**Requisitos de interfaz:** el sistema ofrece una interfaz amigable y fácil de operar. Representando de modo simple la solución que se quiere alcanzar con el sistema.

**Requisitos de hardware:** las capacidades del cómputo deben ser:

1. **Microprocesador:** Intel Core 2 Duo de Tercera Generación.
2. **Memoria RAM:** 4 GB DDR2.
3. **Tarjeta de Red:** Fast-Ethernet 100 MB/s.

**Requisitos de diseño:** la apariencia de las vistas debe ser de color negro, azul y blanco. El tipo de letra a utilizar en las interfaces y los mensajes de la aplicación es Open Sans, SansSerif.

### 2.6. Descripción general de la propuesta.

Para realizar la segmentación de las imágenes en retroiluminación provenientes de la lámpara de hendidura se propone la implementación de un algoritmo de segmentación que está compuesto por el siguiente ciclo:

4. Captura de la imagen.
5. Mejorar la calidad de la imagen.
6. Transformar la imagen a escalas de grises.
7. Reducir el ruido en la imagen.
8. Segmentar la imagen.

Dicho algoritmo se implementará en Matlab utilizando el toolbox de procesamiento digital de imágenes por las bondades que brinda el sistema a la hora de trabajar con dicho recurso. Luego, se hace necesario vincular el resultado de la segmentación con el software PANDOC que ha sido desarrollado en java. Para ello se propone integrar dichas herramientas (Matlab con Java) mediante la ejecución en java del archivo ejecutable encargado de realizar la segmentación previamente construido por

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

Matlab. Para lograr entender cómo funciona el proceso de segmentar la imagen para obtener la OCP se hace uso de la Notación de Modelado de Procesos del Negocio o en inglés *Business Process Modeling Notation* (BPMN). Utilizando el diagrama de procesos de negocio para representar gráficamente la secuencia de todas las actividades que ocurren para lograr la segmentación de la imagen. La siguiente ilustración muestra el diagrama de proceso asociado a el proceso de segmentar la OCP en las imágenes en retroiluminación provenientes de la lámpara de hendidura.

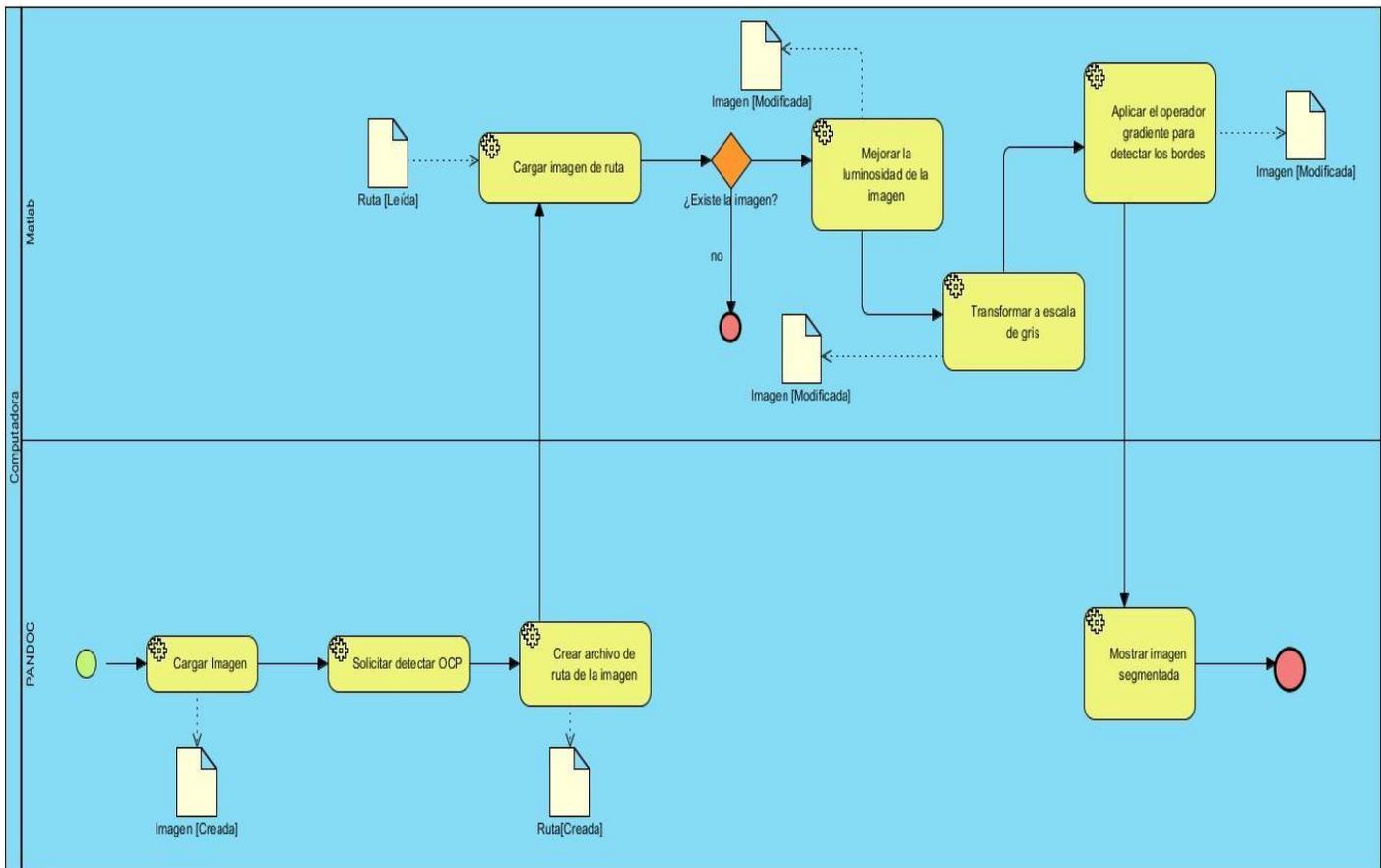


Ilustración 20 Diagrama de procesos del negocio vinculado con el proceso de detectar la OCP (Por elaboración propia).

A continuación, se dará una explicación de cada una de las etapas del algoritmo de segmentación propuesto, así como de la vinculación del Matlab con java:

1. La captura de la imagen se realiza mediante el equipo lámpara de hendidura por el operario de la misma, para luego ser cargada por el software PANDOC. En esta primera etapa se hace importante definir estándares de toma de imagen para que la calidad sea la adecuada. El equipo

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

debe tener un haz de luz de 2 a 3 milímetros y un ángulo de 30 a 45 grados. La ilustración 21 muestra un ejemplo de una imagen perfectamente tomada.

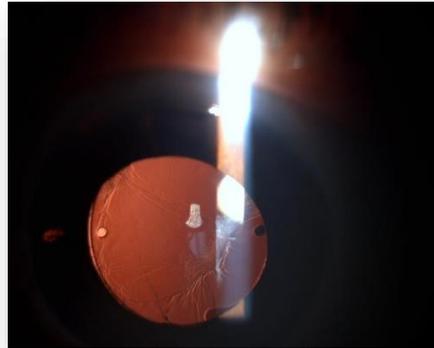


Ilustración 21 Imagen en retroiluminación proveniente de la lámpara de hendiduras.

2. Para mejorar la imagen se aplicará el método Clahe (ecualización adaptativa del histograma con contraste limitado) por su traducción del inglés. Según (Laorden Fiter 2012) el histograma es un gráfico que relaciona los niveles de intensidad de grises de una imagen y el número de píxeles que hay para cada nivel de intensidad. Por lo que al expandir el histograma de la imagen se puede realizar una mejora del contraste de la misma. Con la ayuda Matlab mejorar el contraste en las imágenes se hace más sencillo. Lo primero a tener en cuenta es que las imágenes en retroiluminación tienen un formato RGB (Rojo, Verde y Azul) por su traducción del inglés, por lo que hay que pasar dicha imagen a un espacio de color que tenga el componente luminosidad dentro de él, como es el caso del espacio  $L^*a^*b$ , donde L es luminosidad, a es la tonalidad de rojo a verde y b es la tonalidad amarillo azul (Ripoll, Panea, y Albertí 2012), con el objetivo de afectar solo la intensidad de los píxeles y preservar los colores originales. Luego se modifican los valores de luminosidad pasando del rango de [0 100] a [0 1] que es el apropiado para imágenes del espacio  $L^*b^*a$ . Finalmente se aplica Clahe. La ilustración 22 muestra los comandos de Matlab utilizados para realizar lo antes descrito y la ilustración 23 muestra el resultado de aplicar Clahe a la ilustración 21.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

```
cform2lab=makecform('srgb2lab');  
Lab=applycform(Imagen,cform2lab);  
Lab=lab2double(Lab);  
L=Lab(:,:,1)/100;  
Lab(:,:,1)=adapthisteq(L)*100;  
cform2srgb=makecform('lab2srgb');  
PRGB= applycform (Lab ,cform2srgb);  
imshow(PRGB);  
handles.Imagen=PRGB;
```

Ilustración 22 Ejemplo de código Clahe.

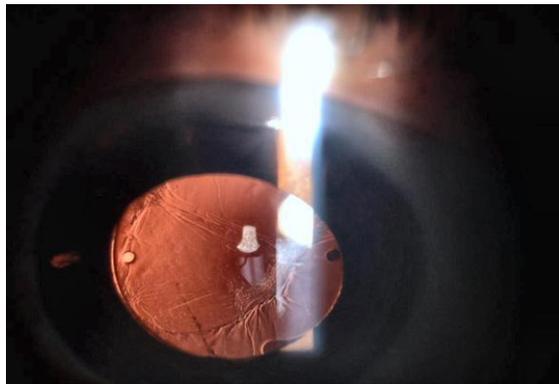


Ilustración 23 Imagen con método Clahe aplicado.

3. Luego que la imagen ha sido mejorada es necesario llevar la misma a escala de grises debido a que todos los algoritmos para realizar los siguientes pasos tienen como entrada una imagen en escala de grises. Para ello se utiliza el comando `rgb2gray` de Matlab que según (Swaminathan 2015) convierte una imagen de formato RGB a escalas de grises, siendo la escala de grises la representación de una imagen en la que cada píxel se dibuja usando un valor numérico individual que representa su luminancia, en una escala que se extiende entre blanco y negro (Herrera Navarro 2014). La ilustración 24 muestra una imagen en escala de grises.

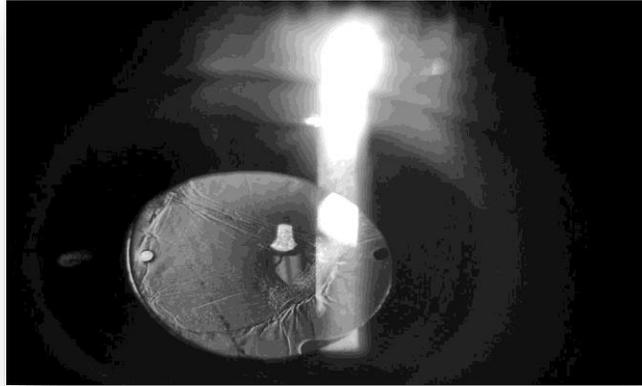


Ilustración 24 Imagen en escala de grises.

4. En ocasiones la imagen puede presentar ruido, que no es más que alteraciones en la imagen que no se corresponde con la realidad, el ruido presente en la imagen debe ser eliminado para lograr una mejora en el momento que los bordes tengan que ser detectados se haga de una manera más eficiente. Para eliminar el ruido se hace uso del comando `fspecial('sobel')` que aplica un filtro de  $3 \times 3$  que enfatiza los bordes horizontales logrando un efecto de suavizado.
5. Es de vital importancia para el correcto funcionamiento del algoritmo que el mismo detecte todos los bordes existentes en la imagen como son el contorno de la pupila, los círculos de la misma y los contornos de opacidad presentes en la imagen. Para ello se puede utilizar el gradiente que permite ir siguiendo el contorno de los objetos utilizando los bordes debido a que él mismo es alto en las fronteras y bajo en el resto de la imagen. Cabe destacar que las imágenes en retroiluminación provenientes de la lámpara de hendidura poseen un haz de luz que es provocado por el mismo equipo y que al ser intensificado con el método Clahe al aplicar el gradiente solo detecta los bordes del mismo, por lo que es eliminado. La ilustración 25 muestra el tomado en Matlab para realizar lo antes descrito y la ilustración 26 muestra el resultado de aplicar dicho código a la ilustración 24.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

```
hy = fspecial('sobel');  
hx = hy';  
Iy = imfilter(double(Imagen), hy,  
'replicate');  
Ix = imfilter(double(Imagen), hx,  
'replicate');  
gradmag = sqrt(Ix.^2 + Iy.^2);
```

Ilustración 25 Ejemplo de código magnitud del gradiente.

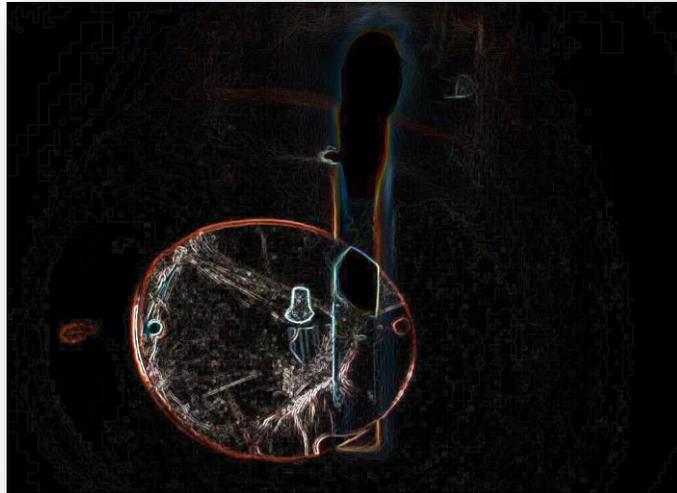


Ilustración 26 Imagen que muestra el resultado de aplicar Gradiente.

El software PANDOC ha sido desarrollado en el lenguaje de programación java, por lo que se hace necesario vincular el proceso realizado con Matlab con dicho software. Para ello se hace uso de los siguientes pasos:

1. Construir un ejecutable del código de Matlab, dicha operación se puede realizar mediante el comando `dplytool` que es el compilador de Matlab y permite crear ejecutables.

## CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA

---

2. Copiar el ejecutable en la raíz del software PANDOC para que más tarde pueda ser llamado desde java.
3. En la aplicación java hay que cargar la imagen que se desea segmentar y guardar la dirección en un archivo que pueda ser leído desde Matlab ejemplo un fichero de notas.
4. Ejecutar el código de Matlab desde java y cargar la imagen resultante.

Para poder darle implementación al algoritmo propuesto se hace necesario el uso de herramientas, tecnología y metodologías de desarrollo que antes descritas.

### **2.7. Conclusiones parciales.**

En este capítulo se propone la implementación de un algoritmo de segmentación de imágenes realizado en el toolbox de procesamiento digital de imágenes de Matlab por las facilidades que este brinda y que luego es vinculado para el software PANDOC, concluyendo que con la implementación de dicho algoritmo el software PANDOC es capaz de detectar la opacidad de la cápsula posterior (OCP) en las imágenes en retroiluminación provenientes de la lámpara de hendidura. También se expone como metodología a utilizar AUP-UCI en su fase 4 y se definen las historias de usuarios y las tarjetas CRC necesarias para la implementación de las funcionalidades.

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

---

### CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

#### 3.1. Introducción

En este capítulo se presenta las diferentes pruebas realizadas al software PANDOC con la incorporación de los métodos de segmentación para las imágenes en retroiluminación provenientes de la lámpara de hendidura. Así como se realizan pruebas para medir la calidad de la segmentación y determinar qué tan eficiente llega a ser la misma.

#### 3.2. Prueba de Software.

Las pruebas de software son un conjunto de herramientas y técnicas que hacen posible el aseguramiento técnico de la calidad del software presentado y constituyen una revisión final de las especificaciones, del diseño y de la codificación. Según Escalona, Vos, y Gutiérrez (2012) la prueba de software es la técnica más usada en la industria para garantizar la calidad , y a menudo, constituye el 50% del esfuerzo total del ciclo de vida de desarrollo de un proyecto.

Uno de los pilares de AUP-UCI es el proceso de pruebas. AUP exhorta a probar constantemente, tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados. AUP-UCI divide las pruebas del software en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

##### 3.2.1. Pruebas de unidad.

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

Estas pruebas se aplican al código perteneciente al método "**Segmentar**" de la clase "**ImagenClass**", el cual a partir de la ruta de la imagen cargada por el usuario segmenta la imagen. A éste se le aplica la métrica de complejidad ciclomática debido a su relevancia para el sistema.

La ilustración 27 muestra el método Segmentar y la ilustración 28 muestra el grafo de flujo asociado a

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

la prueba antes mencionada.

```
public void Segmentar() throws IOException {
    if (Url == null) {
        System.out.println("No hay ninguna imagen cargada");
    } else {
        File archivo = new File("ruta.txt");
        if(archivo.getName()!="ruta.txt")
        {
            FileWriter escribir = new FileWriter(archivo);
            escribir.write(Url.toString());
            escribir.close();
        } else {
            System.out.println("Error al escribir en el archivo ruta.txt");
        }
        Runtime aplicacion = Runtime.getRuntime();
        if(aplicacion.exec("SegmentarImg.exe").isAlive()){
            aplicacion.exec("SegmentarImg.exe");
        } else {
            System.out.println("El archivo SegmentarImg.exe no ha sido encontrado por el sistema ");
        }
    }
}
```

Ilustración 27 Imagen del método Segmentar de la clase ImagenClass.

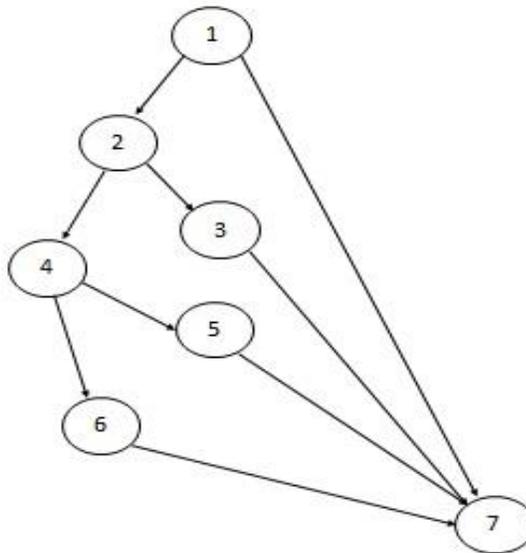


Ilustración 28 Grafo de flujo asociado al método Segmentar de la clase ImagenClass.

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

---

Después de diseñado el grafo de flujo se procede a calcular la complejidad ciclomática, la cual es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa (Harriette-Cabrera y Sanamé-Álvarez 2014). El valor calculado define el número de caminos independientes del conjunto básico de un programa. Esto indica el límite superior para el número de pruebas que se deben realizar, para asegurar que se ejecuta cada sentencia al menos una vez. Se utilizó la siguiente forma:  $V(G)$ , de un grafo de flujo  $G$  se define como:  $V(G)=A-N+2$ , donde  $A$  es el número de aristas del grafo de flujo y  $N$  es el número de nodos. Otra forma de definir la complejidad ciclomática es  $V(G)=$  Regiones o  $V(G)=$  Nodos Predicados. En todos los casos el valor de  $V(G)$  debe de ser el mismo.

### Complejidad ciclomática:

$$V(G)= \text{Regiones}= 4$$

$$V(G)= A-N+2 = 9 -7 +2= 4$$

$$V(G)= \text{Nodos Predicados} +1 = 3 + 1= 4$$

### Caminos independientes determinados:

El valor de  $V(G)$  determina el número de caminos linealmente independientes y con el objetivo de probarlos todos se diseñó un caso de prueba para cada uno de ellos. Obteniendo los siguientes resultados.

**Camino 1:** 1-2-4-6-7 (El usuario ha cargado la imagen en el sistema, el sistema guarda la ruta de la imagen para luego proceder a ejecutar la segmentación y el sistema crea la imagen segmentada de manera satisfactoria)

**Camino 2:** 1-2-4-5-7 (El usuario ha cargado la imagen en el sistema, el sistema guarda la ruta de la imagen, cuando va a ejecutar la segmentación no encuentra el archivo en el sistema por lo que notifica dicho suceso para luego volver a la pantalla inicial de manera correcta)

**Camino 3:** 1-2-3-7 (El usuario ha cargado la imagen en el sistema, el sistema no puede escribir la ruta satisfactoriamente por lo que procede a notificar dicha acción y vuelve a la pantalla inicial de manera correcta)

**Camino 4:**1-7 (El usuario no ha cargado la imagen, el sistema notifica dicha acción y vuelve a la pantalla inicial satisfactoriamente)

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

---

Las pruebas unitarias realizadas al software sumaron un total de 20, 19 de las cuales (para un 95%) resultaron satisfactoria. De esta manera fue evidenciada la estabilidad de la lógica del código empleado durante la implementación de la funcionalidad. En una segunda iteración el 100 % de las pruebas resultaron satisfactorias. La tabla 6 evidencia dicho suceso.

Tabla 7 Iteraciones de las pruebas unitarias.

Iteración	Satisfactoria	Insatisfactoria	Total
1	19	1	20
2	20	0	20

### 3.2.2. Prueba de aceptación.

Las pruebas de aceptación cumplen con el objetivo de indicar cuando las funcionalidades de una iteración han sido completamente exitosas. La validación del software se consigue mediante una serie de pruebas de caja negra que muestran la satisfacción del cliente con los requisitos. El objetivo de las pruebas de aceptación es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Para la descripción de las pruebas se presenta el diseño del caso de prueba correspondiente a la historia de usuario "Obtener OCP en la imagen" que incluye funcionalidades críticas para el sistema. La siguiente tabla 7 muestra los resultados obtenidos en la prueba de aceptación.

Tabla 8 Caso de prueba Aceptación HU5.

Caso de Prueba de Aceptación	
<b>Código:</b> HU5	<b>HU:</b> 5
<b>Descripción:</b> Prueba para la funcionalidad que detecta la OPC en las imagen en retroiluminación provenientes de la lámpara de hendidura.	

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

### Condiciones de ejecución:

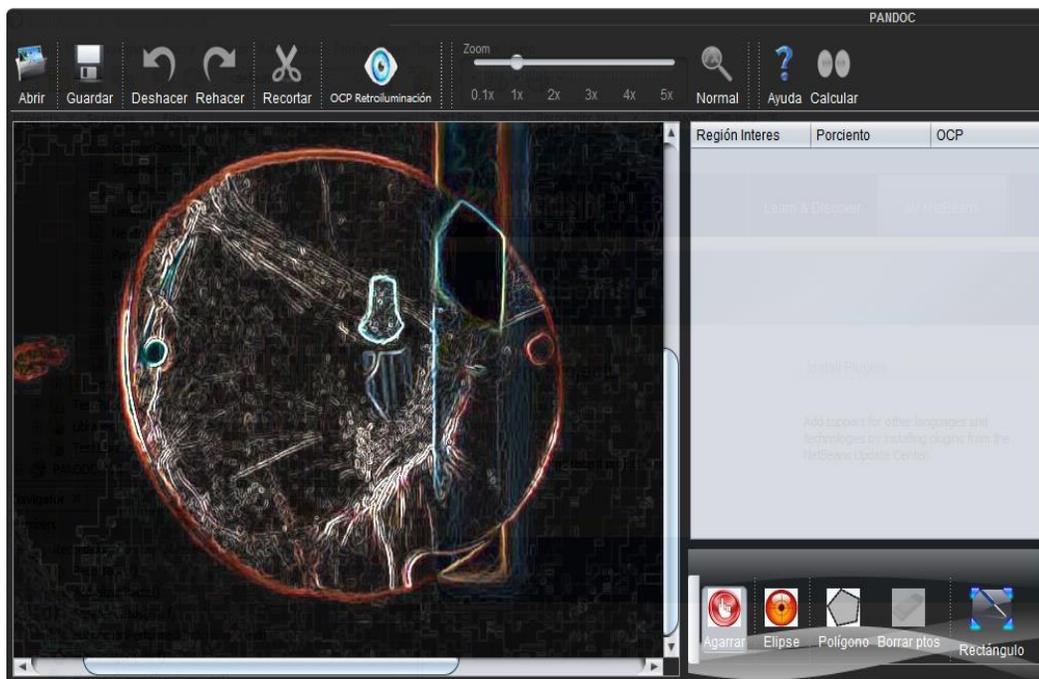
Se debe tener instalado el Compiler Runtime de Matlab versión 8.1

El usuario tiene que haber cargado la imagen en el sistema.

El usuario presiona el botón segmentar imagen.

**Resultado esperado:** El sistema tiene que mostrar la imagen segmentada.

### Resultado obtenido:



**Evaluación de la prueba:** Satisfactoria.

En total las pruebas de aceptación realizadas al software suman una total de 5, de las cuales todas obtuvieron resultados satisfactorios como muestra la Ilustración 29. Para ver el resto de las pruebas realizadas se puede consultar el **Anexo 2**.

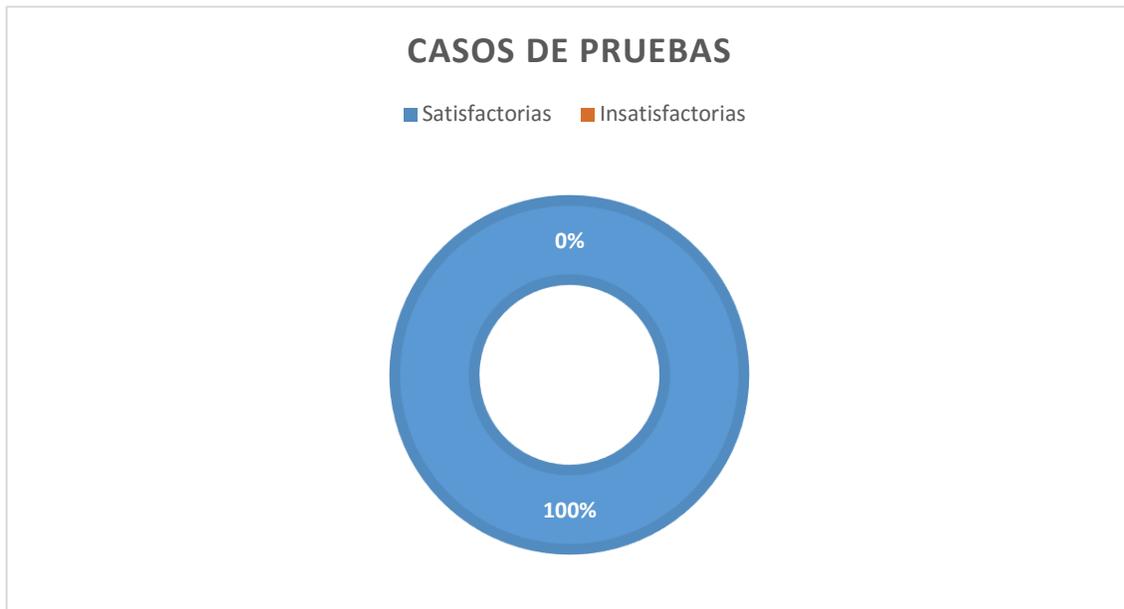


Ilustración 29 Resultados de las pruebas de aceptación

### 3.3. Aplicación de la métrica para medir la calidad de la segmentación

De acuerdo con (Pressman, Roger S 2014) las métricas de software se refieren a un amplio elenco de mediciones para el software de computadora. La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua. Se puede utilizar en el proyecto de software para ayudar en la estimación, el control de la calidad, la evolución de la productividad y el control de proyectos. Según (Greiner et al. 2013) el proceso de software y las métricas del producto son una medida cuantitativa que le permite a los desarrolladores del software tener una visión profunda de la eficiencia del proyecto de software y de los proyectos que dirigen utilizando el proceso como un marco de trabajo. Se reúnen los datos básicos de calidad y productividad. Estos datos son entonces utilizados, comparados con promedios anteriores, y evaluados para determinar las mejoras en la calidad y productividad. Las métricas son también utilizadas para señalar áreas con problemas de manera que se puedan desarrollar los remedios y mejorar el proceso de software.

Como bien se ha expuesto en la presente investigación las técnicas de segmentación de imágenes varían según el tipo de imagen, por lo que se hace necesario medir la calidad de la segmentación realizada y evaluar cuál es el método que mejores resultados obtiene a la hora de segmentar las

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

---

imágenes en retroiluminación provenientes de la lámpara de hendidura. A continuación, se detallan las métricas con las cuales se evaluará la segmentación realizada por los diferentes algoritmos.

La **precisión** de una métrica de segmentación se refiere al grado en el que los resultados de la segmentación coinciden con la segmentación verdadera. Hay situaciones en las que la segmentación verdadera es conocida, y otras, como por ejemplo las que tratan con imágenes de pacientes o modelos para la investigación con animales, en las que la segmentación verdadera no se conoce. En dichas situaciones la segmentación verdadera se sustituye por una segmentación manual realizada por un experto, o realizada con otra técnica de imagen, o usando un algoritmo de segmentación del cual se conoce que produce resultados precisos. Idealmente la medida de la precisión debe reflejar el grado de discordancia entre la segmentación verdadera y aquella que se desea evaluar, y no debería depender de las dimensiones de la imagen (Joaquin Panduro Cívico 2010).

Según (Francisco Ortiz, Bernal del Nozal, y others 2015), si  $S_s$  y  $S_t$  denotan las superficies o regiones interiores al borde en la imagen segmentada y en la segmentación verdadera, respectivamente, se define el área de verdaderos positivos ( $VP$ ) como el área intersección de  $S_s$  y  $S_t$ , es decir  $S_{vp} = S_s \cap S_t$ . Igualmente, el área de falsos positivos ( $FP$ ) comprendería al área segmentada en la segmentación del algoritmo que no lo está en la segmentación verdadera, es decir, la resta de ambas áreas  $S_{fp} = S_s - S_t$ . Asimismo, el área de falsos negativos ( $FN$ ) se correspondería con la resta opuesta,  $S_{fn} = S_t - S_s$ , y el área de verdaderos negativos ( $VN$ ) se correspondería con  $S_{vn} = ESCENA - S_s - S_t$ . Con estas cantidades, las métricas para evaluar la precisión se definirían de la siguiente manera:

$$\text{Fracción de verdaderos positivos: } FVP = \frac{S_{vp}}{S_t} \quad (5)$$

$$\text{Fracción de verdaderos negativos: } FVN = \frac{S_{fn}}{S_t} \quad (6)$$

$$\text{Fracción de falsos positivos: } FVP = \frac{S_{fp}}{Escena - S_t} \quad (7)$$

$$\text{Fracción de falsos negativo: } FFN = \frac{S_{vn}}{Escena - S_t} \quad (8)$$

Con  $ESCENA$  se denota a la región que comprende todas las posibles regiones segmentadas,  $FVP$  sería la fracción del área de la segmentación verdadera que está también comprendida en la segmentación del algoritmo; la  $FVN$  sería la fracción del área de la segmentación verdadera que no se acertó a segmentar en la segmentación del algoritmo, la  $FFP$  sería la fracción del área de la

## CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

---

segmentación del algoritmo que no aparece en la segmentación verdadera, y la FFN es la fracción del área del fondo de la escena que no pertenece al área segmentada ni en la segmentación verdadera ni en la segmentación del algoritmo.

Luego se puede calcular el valor de la precisión mediante la siguiente ecuación:

$$precisión = \frac{VP}{VP + FP} \quad (9)$$

Donde VP representan los píxeles clasificados como verdaderos positivos y FP son los píxeles clasificados como falsos positivos.

En virtud de cuantificar qué tan precisa es la calidad de la segmentación realizada por el algoritmo presentado se ha diseñado el siguiente caso experimental, donde se puede comparar la segmentación realizada por el algoritmo con la segmentación realizada por el especialista en oftalmología.

### **Caso de prueba experimental:**

1. Se procede a tomar 5 casos de estudios de imágenes de pacientes con OCP nombrados (OCP-1, OCP-2, OCP-3, OCP-4 y OCP-5).
2. Luego se realiza la segmentación manual por parte del especialista y la segmentación automática por parte del algoritmo propuesto. Las ilustraciones 30 y 31 muestran el resultado de dicho proceso para el caso OCP-1. Cabe destacar que la segmentación manual por parte del especialista está realizada con Paint y el área señalada en azul es la que el mismo considera OPC.
3. El valor de la precisión está dado por la fórmula  $precisión = VP/(VP+FP)$ , por lo que se procede a calcular dichos valores y se tomará como satisfactoria la segmentación donde la precisión supere el 85% e insatisfactorio si se encuentra por debajo de ese umbral.
4. El tiempo de respuesta del sistema para segmentar las imágenes también será analizado y queda registrado en la variable tiempo de respuesta del sistema (TRS), clasificando como aceptable los que se encuentren por debajo de 10 segundo y no aceptables por encima de 10 segundos.

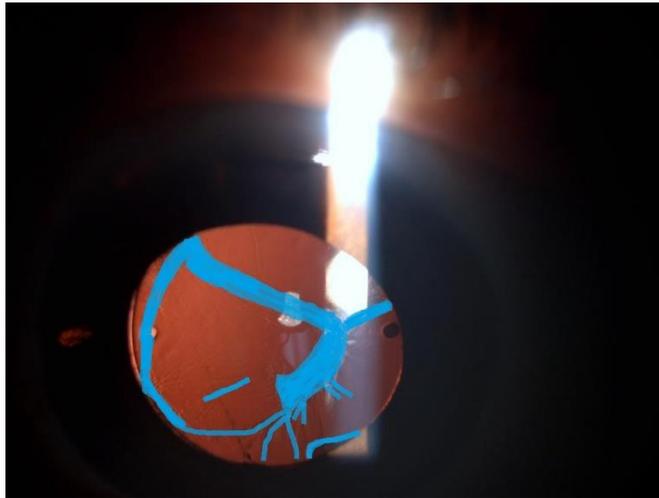


Ilustración 30 Imagen OCP-1 segmentada por el especialista.

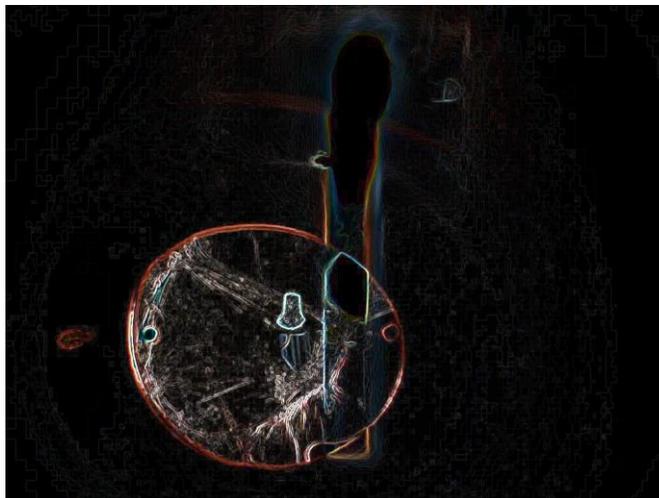


Ilustración 31 Imagen OCP-1 segmentada por el algoritmo.

La tabla 8 muestra los resultados de las comparaciones realizadas. Dando una precisión promedio de un 93.6% y el tiempo de funcionamiento promedio fue de 6.4 segundos. Lo antes expuesto demuestra que la calidad de la segmentación realizada es satisfactoria debido que en todos los casos se ha superado el umbral de 85% y que los tiempos de respuesta del sistema no han pasado de 8 segundos, lo que expone que son aceptables. La métrica de calidad realizada también está acorde con la satisfacción de los especialistas que han evaluado la efectividad del software PANDOC al detectar OCP en imágenes en retroiluminación provenientes de la lámpara de hendidura.

### CAPÍTULO 3. VALIDACIÓN DE LA PROPUESTA

Tabla 9 Muestra los resultados de la métrica Precisión

Caso de estudio	VP	FP	Precisión	TRS (en segundos)
OCP-1	747110	39322	95%	7
OPC-2	723517	62915	92%	8
OCP-3	754974	31458	96%	6
OCP-4	747110	39322	95%	6
OCP-5	707788	78644	90%	5

Para una mejor comprensión de los datos expuesto por la tabla anterior se presenta la ilustración 31, en la cual queda evidenciado que la cantidad de píxeles clasificados como VP siempre es mucho mayor que los clasificados como FP.

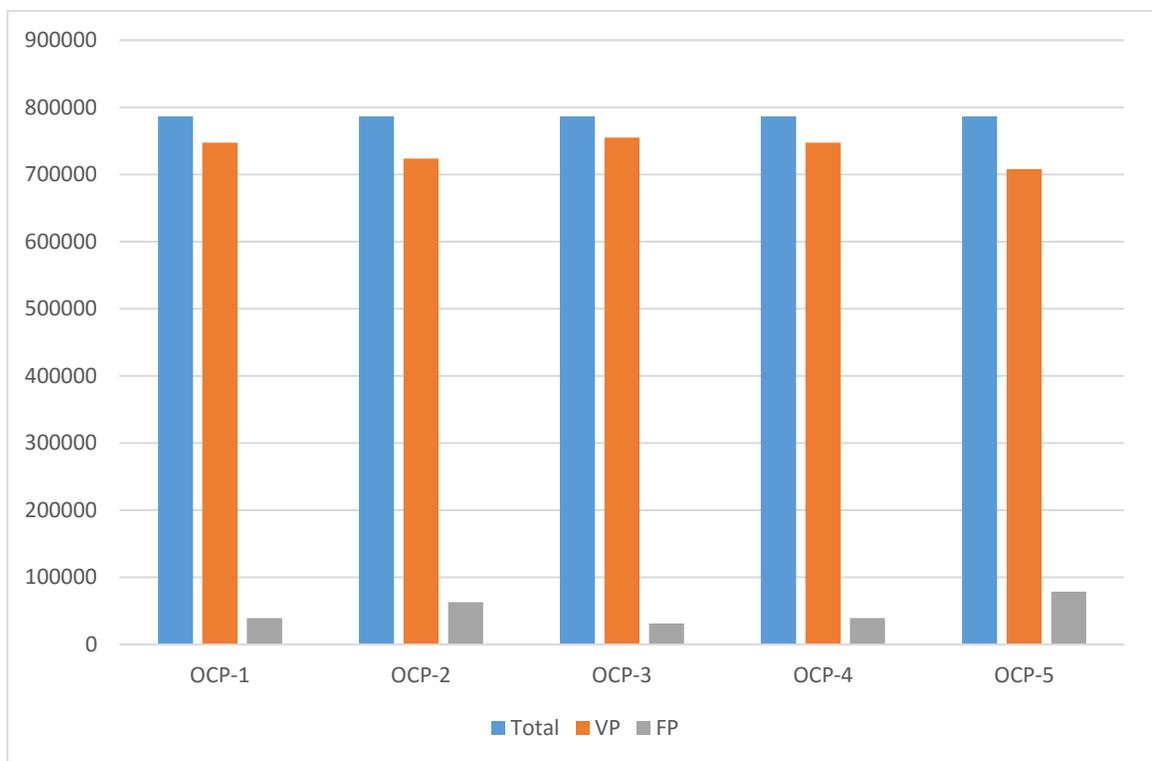


Ilustración 32 Resultados de la métrica precisión

### 3.4. Conclusiones parciales

En este capítulo se puede concluir que la calidad de la segmentación de las imágenes en retroiluminación provenientes de la lámpara de hendidura realizada por el algoritmo propuesto es satisfactoria. También se realizaron las pruebas de software de aceptación y unitarias, presentándose los casos de pruebas correspondientes y se analizaron los resultados concluyendo que el sistema PANDOC está libre de no conformidades. Por otra parte, queda demostrado, que con la implementación del algoritmo de segmentación propuesto el software PANDOC puede identificar la OCP en las imágenes en retroiluminación provenientes de la lámpara de hendidura.

**CONCLUSIONES GENERALES**

Teniendo en cuenta los objetivos propuestos en la investigación se puede concluir lo siguiente:

1. Con el análisis de los principales conceptos asociados con la investigación se logró comprender cómo funciona a nivel mundial los procesos vinculados con la segmentación de imágenes médicas y se pudo trazar una línea de trabajo para elaborar la presente investigación.
2. El estudio de las diferentes técnicas de segmentación mediante la consulta de la bibliografía científica, permitió la elaboración de un algoritmo de segmentación de imágenes con tres etapas fundamentales, que consiste en la captura de la imagen, mejora de la imagen y segmentación de la imagen.
3. El uso del toolbox de procesamiento digital de imágenes de Matlab permitió que la implementación del algoritmo propuesto se hiciera más sencilla, por la amplia gama de funcionalidades críticas para la segmentación que en él se encuentran implementadas.
4. Las pruebas realizadas al algoritmo de segmentación de imágenes en retroiluminación provenientes de la lámpara de hendidura que fue implementado en el software PANDOC, junto con la métrica de calidad de segmentación demostró que el mismo puede ser utilizado en la medicina para la detección de la OCP en pacientes operados de cataratas.

## RECOMENDACIONES

---

### RECOMENDACIONES

1. Continuar con la cuarta etapa del PDI que consiste en la cuantificación objetiva de la opacidad de la cápsula posterior en imágenes en retro iluminación provenientes de la lámpara de hendidura, con el objetivo de dotar al oftalmólogo de un valor cuantitativo.
2. Extender la investigación de algoritmos de segmentación de imágenes médicas con el fin de que el software PANDOC sea capaz de abarcar otras áreas dentro de la medicina.

## REFERENCIAS BIBLIOGRÁFICAS

---

### REFERENCIAS BIBLIOGRÁFICAS

- Alvarez Cancio. 2014. «PANDOC:Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata.» UCI.
- Aslam, Tariq M., Niall Patton, y Jim Graham. 2005. «A freely accessible, evidence based, objective system of analysis of posterior capsular opacification; evidence for its validity and reliability». *BMC ophthalmology* 5 (1): 1.
- Barba Guamán, Luis Rodrigo. 2015. «Utilización de métodos de visión artificial para PC como apoyo en la automoción». ETSI\_Informatica. <http://oa.upm.es/39016/>.
- Barman, Sarah A., Emma J. Hollick, James F. Boyce, David J. Spalton, Bunyarit Uyyanonvara, Giorgia Sanguinetti, y William Meacock. 2000. «Quantification of posterior capsular opacification in digital images after cataract surgery». *Investigative ophthalmology & visual science* 41 (12): 3882–3892.
- Besbes, Moez. 2004. «Cells images color segmentation based on thresholding and watershed segmentation». En *Industrial Technology, 2004. IEEE ICIT'04. 2004 IEEE International Conference on*, 1:32–37. IEEE. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1490252](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1490252).
- Botella, Guillermo, José Antonio Martín, Joaquín Recas, y Concepción Sanz. 2013. «Bioinspired temporal filter modeling for motion estimation». *REVISTA TECNICA DE LA FACULTAD DE INGENIERIA UNIVERSIDAD DEL ZULIA* 36 (1): 13–22.
- Chang, C.-I., Yingzi Du, J. Wang, S.-M. Guo, y P. D. Thouin. 2006. «Survey and comparative analysis of entropy and relative entropy thresholding techniques». En *Vision, Image and Signal Processing, IEE Proceedings-*, 153:837–850. IET. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4027996](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4027996).
- Daigavane, Prema M., y Preeti R. Bajaj. 2010. «Road lane detection with improved canny edges using ant colony optimization». En *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*, 76–80. IEEE. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5698295](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5698295).
- Escalona, M. J., Tanja E. J. Vos, y J. J. Gutiérrez. 2012. *Pruebas de software en la enseñanza universitaria de la informática: un título propio*. Universidad de Castilla-La Mancha. <http://upcommons.upc.edu/handle/2099/15084>.
- Findl, Oliver, Wolf Buehl, Peter Bauer, y Thomas Sycha. 2010. «Intervenciones para la prevención de la opacificación de la cápsula posterior». <http://summaries.cochrane.org/es/CD003738/intervenciones-para-la-prevencion-de-la-opacificacion-de-la-capsula-posterior>.
- Findl, Oliver, Wolf Buehl, Rupert Menapace, Michael Georgopoulos, Georg Rainer, Hannes Siegl, Alexandra Kaider, y Axel Pinz. 2003. «Comparison of 4 methods for quantifying posterior capsule opacification». *Journal of Cataract & Refractive Surgery* 29 (1): 106–111.
- Francisco Ortiz, Jorge, Jorge Bernal del Nozal, y others. 2015. «Herramienta de segmentación semiautomática de imágenes mediante visión por computador». <http://ddd.uab.cat/record/143606>.

## REFERENCIAS BIBLIOGRÁFICAS

---

- Giménez, Félix Martínez, Alfredo Peris Manguillot, y Francisco Rodenas Escribá. 2013. *Tratamiento de señales digitales mediante wavelets y su uso con Matlab*. Editorial Club Universitario. <https://www.google.com/books?hl=es&lr=&id=WZX9CAAQBAJ&oi=fnd&pg=PR3&dq=comparaci%C3%B3n+y+ejemplos+de+uso+de+la+funciones+del+toolbox+de+procesado+digital+de+im%C3%A1genes+de+Matlab&ots=u1N9xvUNtf&sig=mj2t4btwKz3oQydlxTNH6qIWTFU>.
- Gómez, Javier M. Martínez, Miguel E. Higuera Marín, y Esperanza Aguilar Díaz. 2014. «Enfoque metodológico para el diseño de interfaces durante el ciclo de vida de desarrollo de software». *REVISTA GTI* 12 (34). <http://revistas.uis.edu.co/index.php/revistagti/article/view/3846>.
- Gómez, Vasco, Wilson Daniel, DT-Pomaquero Moreno, y Luis Alfredo. 2012. «Modelo de compresión de imágenes de extensión JPEG mediante Matlab para el mejoramiento de la transmisión de datos». <http://repositorio.uta.edu.ec/handle/123456789/2385>.
- González, Rafael C., Richard E. Woods, Fernando Davue Rodríguez, y Luis Rosso. 1996. *Tratamiento digital de imágenes*. Vol. 3. Addison-Wesley New York. <http://www.sidalc.net/cgi-bin/wxis.exe/?IsisScript=LIBRO.xis&method=post&formato=2&cantidad=1&expresion=mfn=027792>.
- Greiner, Cristina L., Gladys N. Dapozo, J. Acosta, Marcelo G. Estayno, M. Domínguez, y Jorge Chiapello. 2013. «Persistencia de mediciones como apoyo a la gestión de proyectos de software». En *XV Workshop de Investigadores en Ciencias de la Computación*. <http://sedici.unlp.edu.ar/handle/10915/27259>.
- Grewal, Dilraj, Rajeev Jain, Gagandeep Singh Brar, y Satinder Pal Singh Grewal. 2008. «Pentacam tomograms: a novel method for quantification of posterior capsule opacification». *Investigative ophthalmology & visual science* 49 (5): 2004–2008.
- Harriette-Cabrera, Ana D., y Dainerys Sanamé-Álvarez. 2014. «Aplicación informática para obtener la Complejidad Ciclomática de algoritmos que utilicen estructura if y while». *Ciencia & Futuro* 4 (2): 64–75.
- Herrera Navarro, Ana Marcela. 2014. «Construcción de criterios para la segmentación de imágenes en escalas grises». <http://ri.uaq.mx/handle/123456789/1683>.
- Jimenez, J. 2014. «NetBeans IDE entorno de desarrollo para lenguajes como Java PHP C/C++ Groovy». <http://www.genbetadev.com/herramientas/netbeans-1>.
- Joaquin Panduro Cívico. 2010. «Estudio comparativo de algoritmos disponibles en ITK para la segmentación de imágenes médicas». Sevilla.
- Kak, Avinash C. 1984. «Image reconstruction from projections». *Digital image processing techniques*, 111–171.
- Kee, Youngwook, Mohamed Souiai, Daniel Cremers, y Junmo Kim. 2014. «Sequential convex relaxation for mutual information-based unsupervised figure-ground segmentation». En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4082–4089. [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2014/html/Kee\\_Sequential\\_Convex\\_Relaxation\\_2014\\_CVPR\\_paper.html](http://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Kee_Sequential_Convex_Relaxation_2014_CVPR_paper.html).
- Kniberg, Henrik. 2015. *Scrum and XP from the Trenches*. Lulu. com. [https://www.google.com/books?hl=es&lr=&id=R4oXCgAAQBAJ&oi=fnd&pg=PA11&dq=scrum&ots=ecGo3ziRsg&sig=4r\\_nFTvidlw82LtD34uD3dmq7zU](https://www.google.com/books?hl=es&lr=&id=R4oXCgAAQBAJ&oi=fnd&pg=PA11&dq=scrum&ots=ecGo3ziRsg&sig=4r_nFTvidlw82LtD34uD3dmq7zU).

## REFERENCIAS BIBLIOGRÁFICAS

---

- Laorden Fiter, Eduardo. 2012. «Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesamiento digital de imágenes de MATLAB®». <http://oa.upm.es/14016/>.
- López, I., J. Silva, Y. González, A. Fernández, Z. Roviroso, y E. Candelaria. 2010. «Estrategias de prevención de la opacidad de la cápsula posterior». *Revista Cubana De Oftalmología [revista en internet]*, 608–623.
- Lovay, Mónica, Gabriela Peretti, Eduardo Romero, y Carlos Marqués. 2013. «Tolerancia a fallas en un filtro de alto orden mediante una estrategia de hardware evolutivo». En *Libro de Memorias IV Congreso de Microelectrónica Aplicada*, 2013:140–145. [http://uea2013.frbb.utn.edu.ar/wp-content/uploads/S7\\_3.pdf](http://uea2013.frbb.utn.edu.ar/wp-content/uploads/S7_3.pdf).
- Luthon, Franck, Marc Liévin, y Francis Faux. 2004. «On the use of entropy power for threshold selection». *Signal Processing* 84 (10): 1789–1804.
- Miró Pérez, Juan Javier, Estrela Navarro, María José, Jorge Olcina Cantos, y others. 2016. «Reconstrucción de la señal térmica local en la Comunidad Valenciana entre 1948 y 2011 a partir de un downscaling estadístico mediante una red neuronal artificial: detección de patrones locales de cambio». <http://rua.ua.es/dspace/handle/10045/54471>.
- Molina, Valentin, Miguel Vera, Horderlin V. Robles, Edwar Bejarano, y Hermann Dávila. 2014. «Lung segmentation using Support Vector Machine in computed tomography images». En *Health Care Exchanges (PAHCE), 2014 Pan American*, 1–5. IEEE. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6849624](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6849624).
- Nakib, Amir, Hamouche Oulhadj, y Patrick Siarry. 2007. «Image histogram thresholding based on multiobjective optimization». *Signal Processing* 87 (11): 2516–2534.
- Nimbarte, Nita M., y Milind M. Mushrif. 2010. «Multi-level thresholding algorithm for color image segmentation». En *Computer Engineering and Applications (ICCEA), 2010 Second International Conference on*, 2:231–233. IEEE. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5445591](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5445591).
- Nobuyuki Otsu, A. 1976. «Threshold Selection Method from Gray-Level Histograms». *IEEE Trans. on Sys. Man and Cybernetics* 9: 62–65.
- Palomino, Nora La Serna, y Ulises Norberto Román Concha. 2009. «Técnicas de segmentación en procesamiento digital de imágenes». *Revista de investigación de Sistemas e Informática* 6 (2): 9–16.
- Pressman, Roger S. 2014. *Ingeniería del software: Un enfoque práctico*. 7ma ed.
- Pun, Thierry. 1981. «Entropic thresholding, a new approach». *Computer Graphics and Image Processing* 16 (3): 210–239.
- Quintero Torres, Shirley Viviana. 2013. «Estudio comparativo de tres métodos para la selección de zonas candidatas a nódulos pulmonares en imágenes de tomografía computarizada de tórax». <http://repository.javeriana.edu.co/handle/10554/12702>.
- Rahim, M. S. M., A. Norouzi, I. B. M. Amin, y A. E. Rad. 2012. «Current methods in medical image segmentation and its application on knee bone». *Archives Des Sciences Journal* 65 (9).
- Ramos Pereira, Yanay, Juan Carlos Medina Perdomo, Juan Raúl Hernández Silva, Belkys Rodríguez Suárez, Eneida de la Caridad Pérez Candelaria, Marieta Gutiérrez Castillo, y Sorania Rodríguez

## REFERENCIAS BIBLIOGRÁFICAS

---

- Sánchez. 2015. «Diagnóstico y control del astigmatismo en la cirugía del cristalino». *Revista Cubana de Oftalmología* 28 (2): 205–219.
- Ripoll, G., B. Panea, y P. Albertí. 2012. «Apreciación visual de la carne bovina y su relación con el espacio de color CIELab». *ITEA-Informaciones Técnico-Económicas* 108 (2): 218–228.
- Sommerville, Ian, y María Isabel Alfonso Galipienso. 2005. *Ingeniería del software*. Pearson Educación. [https://www.google.com/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=ingenieria+de+requisito&ots=s652quvvzh&sig=5MJyNp5n\\_r6aWNXhYUr8HSVrk5I](https://www.google.com/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=ingenieria+de+requisito&ots=s652quvvzh&sig=5MJyNp5n_r6aWNXhYUr8HSVrk5I).
- Swaminathan, R. 2015. «Identification of Application-Suitability of Point Operations on Digital Images using MATLAB». *International Journal of Computer Applications* 109 (6): 14–17.

## REFERENCIAS BIBLIOGRÁFICAS

---

### ANEXOS.

#### Anexo 1: Tarejetas Clase Responsabilidad Colaborador.

Tabla 10. Tarjeta CRC de la clase Principal.

Nombre de la clase: Principal	
Responsabilidades	Clases Relacionadas
Esta clase es la encargada de brindar la interfaz visual con la cual los usuarios podrán realizar las operaciones:  Cargar Imagen.  Segmentar Imagen.  Guardar Imagen.	ImagenClass.  ImagenAPI.

Tabla 11. Tarjeta CRC de la clase ImagenClass

Nombre de la clase: ImagenClass	
Responsabilidades	Clases Relacionadas
Esta clase es la encargada de realizar el control de los procesos vinculados con la imagen, como es el caso de los procesos:  Cargar Imagen.  Guardar ruta de la imagen.  Segmentar la Imagen.  Guardar la imagen.	ImagenAPI.

Tabla 12. Tarjeta CRC de la clase ImagenAPI

Nombre de la clase: ImagenAPI	
Responsabilidades	Clases Relacionadas

## REFERENCIAS BIBLIOGRÁFICAS

---

Esta clase es la encargada de realizar el manejo de las imágenes permitiendo que el trabajo con ellas sea más fácil, debido a que cuenta con los siguientes procesos:  Obtener pixel.  Obtener tamaño.  Actualizar imagen.  Sistema de coordenadas.	ImagenClass
---	-------------

### Anexo 2: Pruebas de Aceptación.

Tabla 13. Caso Prueba de Aceptación HU1

Caso de Prueba de Aceptación	
<b>Código:</b> HU1	<b>HU:</b> 1
<b>Descripción:</b> Prueba para la funcionalidad de cargar una imagen al sistema.	
<b>Condiciones de ejecución:</b>  Se debe tener instalado el Compiler Runtime de Matlab versión 8.1  El usuario tiene que haber cargado la imagen en el sistema.  El usuario presiona el botón cargar imagen.  El usuario selecciona la imagen.	
<b>Resultado esperado:</b> El sistema tiene que mostrar la imagen seleccionada por el usuario.	

## REFERENCIAS BIBLIOGRÁFICAS

---

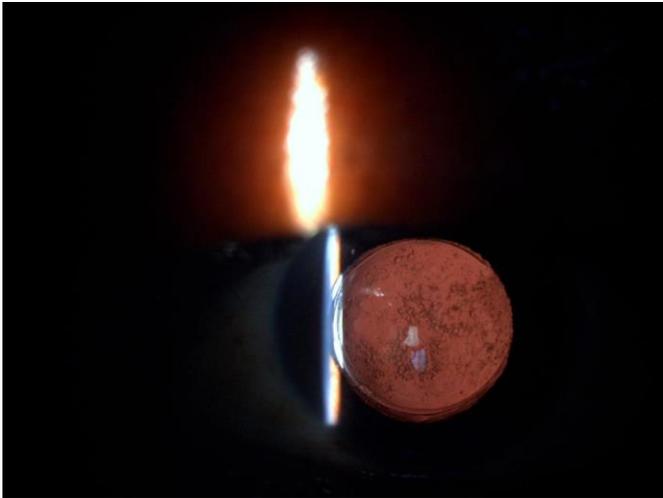
<b>Resultado obtenido:</b>

<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 14. Caso Prueba de Aceptación HU1.1

Caso de Prueba de Aceptación	
<b>Código:</b> HU1.1	<b>HU:</b> 1
<b>Descripción:</b> Prueba para la funcionalidad de cargar una imagen al sistema.	
<b>Condiciones de ejecución:</b>  Se debe tener instalado el Compiler Runtime de Matlab versión 8.1  El usuario tiene que haber cargado la imagen en el sistema.  El usuario presiona el botón cargar imagen.  El usuario presiona el botón cancelar.	
<b>Resultado esperado:</b> El sistema vuelve a la pantalla inicial.	

## REFERENCIAS BIBLIOGRÁFICAS

---

<b>Resultado obtenido:</b>
El sistema retorno a la pantalla inicial.
<b>Evaluación de la prueba:</b> Satisfactoria.

Tabla 15. Caso Prueba de Aceptación HU5

Caso de Prueba de Aceptación	
<b>Código:</b> HU5	<b>HU:</b> 5
<b>Descripción:</b> Prueba para la funcionalidad de guardar una imagen al sistema.	
<b>Condiciones de ejecución:</b> El usuario tiene que haber cargado la imagen en el sistema. El usuario presiona el botón guardar imagen. El usuario selecciona la ruta donde desea guardar la imagen.	
<b>Resultado esperado:</b> El sistema tiene guardar la imagen en la ruta especificada por el usuario.	
<b>Resultado obtenido:</b>	
El sistema ha guardado la imagen en la ruta especificada por el usuario.	
<b>Evaluación de la prueba:</b> Satisfactoria.	

Tabla 16. Caso Prueba de Aceptación HU5.1

Caso de Prueba de Aceptación	
<b>Código:</b> HU5.1	<b>HU:</b> 5

## REFERENCIAS BIBLIOGRÁFICAS

---

<b>Descripción:</b> Prueba para la funcionalidad de guardar una imagen al sistema.
<b>Condiciones de ejecución:</b>  El usuario tiene que haber cargado la imagen en el sistema.  El usuario presiona el botón guardar imagen.  El usuario presiona el botón cancelar.
<b>Resultado esperado:</b> El sistema tiene que volver a la pantalla inicial.
<b>Resultado obtenido:</b>  El sistema ha regresado a la pantalla inicial.
<b>Evaluación de la prueba:</b> Satisfactoria.