



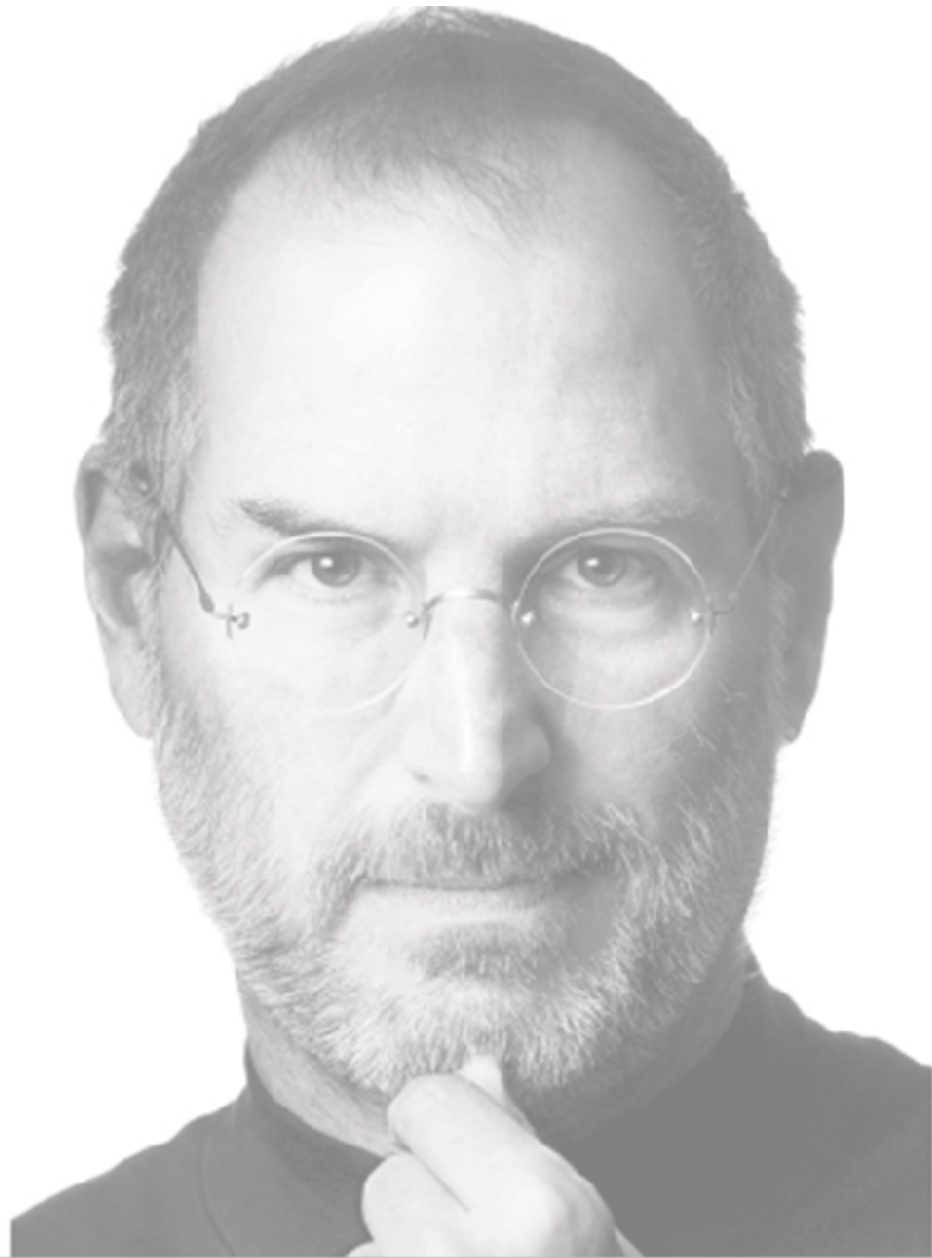
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
AIRI, ARTIFICIAL INTELLIGENCE RESEARCH AND INNOVATION, FACULTAD 3

ALGORITMO PARA LA SEGMENTACIÓN DE LA OPACIDAD DE LA CÁPSULA
POSTERIOR EN IMÁGENES OBLICUAS PROVENIENTES DE LA LÁMPARA DE
HENDIDURA

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Odette Romero Rodríguez
Liliana Rodríguez Martínez
Tutores: Lic. Reyder Cruz de la Osa
Ing. Michel Álvarez Cancio

La Habana, 2016



Piensa diferente y serás diferente
Steve Jobs

Todo mi trabajo y dedicación se lo dedico a mi familia, especialmente a mi mamá Olga Lidia Martínez Barrio, a mi papá Luis Rodríguez Sánchez, a mi compañera de tesis y mejor amiga Odette Romero Rodríguez y a todas las personas que de una forma u otra me han ayudado a cumplir mis metas.

Liliana Rodríguez Martínez

En especial dedicar este trabajo a mi familia por haber depositado tanta confianza en mí y apoyarme en los momentos más importantes de mi vida. A mi mejor amiga, hermana y compañera de tesis por estar siempre conmigo y ayudarme a realizar este trabajo. A todos mis amigos por hacer más llevadera mi vida como estudiante y brindarme su apoyo en todo momento.

Odette Romero Rodríguez

Agradecimientos

Agradezco primeramente a toda mi familia por estar presente en los momentos malos y buenos, por darme las fuerzas para hoy cumplir uno de mis sueños, ser ingeniera. Especialmente le agradezco a mis padres a los cuales les debo todos mis logros y a mi hermano Luis Ángel Rodríguez Martínez por inspirarme a coger la UCI y por ayudarme durante mi trayectoria en la universidad, a mis abuelos por sus buenos consejos que hoy me hicieron tomar el camino correcto.

A mi hermana y compañera de tesis Odette Romero Rodríguez por aguantarme durante estos cinco años, por demostrarme lo que es una verdadera amistad y por estar conmigo cuando más necesité de ella.

A mis tutores Reyder y Michel por ayudarme durante la elaboración de mi trabajo de diploma, por sus buenos consejos y por darme la oportunidad de defender un tema que me gusta tanto.

A mis compañeras de apartamento Yessida, Lidia, Laura, Yaima y Claudia, pues la convivencia con ellas fue excelente, por darme tantos buenos momentos, ya que hoy se han convertido en parte de mi familia.

A todos mis amigos por estar en las malas y en las buenas, por hacer que mi vida en la universidad fuera la mejor etapa que he pasado, especialmente a Nelson, Dayanis, Eliuvis, Juan, Roberto y a Ernesto por ayudarme tanto. A muchos que han sido parte de alguna etapa de mi vida y también significan algo muy importante para mí.

Liliana Rodríguez Martínez

Quiero agradecer primeramente a mi abuela Josefina y mi mamá Odalys que han estado desde el primer momento apoyándome en todas mis decisiones , ya que de algún modo esta tesis es fruto de sus esfuerzos y sacrificios. Gracias a su amor he alcanzado mis metas. Mis logros son suyos también.

A mis tios Daniel y Lidia por su amor, apoyo , confianza y los consejos brindados durante el transcurso de mi carrera . A mi primita Daniela por brindarme alegrías .

A mi abuelo Alberto y mi papá Alejandro por su preocupación y apoyo incondicional.

A mi novio Luisito por su paciencia , amor y ayuda durante mi etapa como estudiante, gracias a él pude alcanzar mis objetivos y sueños.

Agradecimiento especial a mi hermanita y compañera de tesis que tiene conocimiento de las noches y la madrugadas que pasamos estudiando para llegar a este momento, que estuvo conmigo en las buenas y las malas. Gracias por tu comprensión y cariño durante este tiempo.

Agradecer a mi amigo Eliuvis que estuvo ayudándome desde mi comienzos en la universidad, a Juan y Roberto por los gratos momento que pasamos juntos, a los amigos del aula con lo cual compartí momentos inolvidables.

A mis compañeras de cuarto Yessida, Lili Laura , Claudia y Yaima por la ayuda brindada.

A los tutores Michel Álvarez Cancio y Reyder de la Cruz por la ayuda durante todo el desarrollo investigativo de este trabajo.

En general agradezco a todas las personas que de una forma u otra han contribuido a la realización de este trabajo de diploma.

Odette Romero Rodríguez

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

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

Odette Romero Rodríguez
Autor

Liliana Rodríguez Martínez
Autor

Lic. Reyder Cruz de la Osa
Tutor

Ing. Michel Álvarez Cancio
Tutora

En la actualidad la enfermedad de cataratas es una de las principales causas de ceguera en el mundo, estimándose que cuenta con la mitad de los casos de ceguera en la población mundial. La Opacidad de la Cápsula Posterior constituye hoy en día el aspecto más importante en la cirugía de cataratas, actualmente sigue siendo la complicación postoperatoria más frecuente a largo o mediano plazo. Las imágenes que se utilizaron para la identificación de la La Opacidad de la Cápsula Posterior son imágenes oblicuas provenientes de las lámparas de hendiduras, las cuales presentan cierta desventaja como resultado de la reflexión de la luz proveniente del artefacto, por lo que se hizo necesario realizar un preprocesado a las imágenes para obtener resultados favorables en el proceso de segmentación. Debido a lo antes expuesto la presente investigación estuvo relacionada con la identificación de la opacidad en imágenes oblicuas provenientes de pacientes operados de catarata. Para el caso del preprocesado se utilizó el filtro promedio para disminuir el ruido y la ecualización del histograma para mejorar el contraste de las imágenes. Posteriormente se realizó la segmentación por textura y dentro de esta se utilizó el método de Markov para segmentar las regiones de interés, logrando con esto segmentar las imágenes con opacidad.

Palabras clave: Filtro, Markov, preprocesamiento, segmentación .

Introducción	1
1 Fundamentación Teórica	5
1.1 Opacidad de la Cápsula Posterior:Conceptualización	5
1.1.1 Incidencia y prevención de la Opacidad de la Cápsula Posterior.	5
1.2 Procesamiento de Imágenes.	6
1.2.1 Captura de imágenes digitales.	7
1.2.2 Preprocesamiento de imágenes	8
1.2.3 Segmentación de imágenes	12
1.3 Herramientas usadas durante el desarrollo	24
1.3.1 Metodología de Software	24
1.3.2 Entorno integrado de desarrollo	25
1.3.3 Lenguaje de Programación	27
1.3.4 Herramienta utilizada para la realización de las pruebas unitarias	27
1.3.5 Herramienta para el modelado	28
1.3.6 Herramienta para la creación del documento	28
2 Propuesta de solución	30
2.1 Técnicas de procesamiento utilizadas	30
2.1.1 Comparación entre las técnicas de preprocesamiento	31
2.1.2 Comparación entre los métodos de segmentación	32
2.2 Selección del método propuesto	32
2.3 Fase de Planificación	34
2.3.1 Historias de Usuarios	34
2.3.2 Planificación de la Entrega	36
2.3.3 Plan de iteraciones	37
2.3.4 Plan de entrega	38
2.4 Diseño	38
2.4.1 Tarjetas CRC	39

3 Implementación y Pruebas	41
3.1 Fase de implementación	41
3.1.1 Iteración 1	41
3.1.2 Iteración 2	44
3.1.3 Iteración 3	46
3.2 Interfaz del sistema	47
3.3 Resultados de aplicar el algoritmo	48
3.4 Pruebas	49
3.4.1 Pruebas unitarias	49
3.4.2 Pruebas de aceptación	52
3.4.3 Validación del sistema	58
Conclusiones	62
Recomendaciones	63
Referencias bibliográficas	64
Apéndices	68
A Historias de Usuarios	69
B Diseño	74
C Pruebas	76
C.1 Pruebas de aceptación	77
C.2 Técnica del camino básico	82
C.3 Validación	84

Índice de figuras

1.1	Etapas del procesamiento de imágenes	7
1.2	Ejemplo de filtros lineales de paso bajo	10
1.3	Ejemplo de filtro no lineal	11
1.4	Imagen obtenida luego de aplicar el operador de Canny.	14
1.5	Máscara utilizada para aplicar el Operador Laplaciano.	15
1.6	Histograma de intensidades de grises y a la derecha imagen en escala de grises.	17
1.7	Procedimiento para aplicar el Modelo de Gabor.	22
2.1	Imagen original.	30
2.2	Resultados de aplicar técnicas para mejorar el contraste en la imagen.	31
2.3	Resultados de aplicar un conjunto filtros a la imagen.	31
2.4	Diferentes métodos de segmentación aplicados a la imagen.	32
2.5	Diagrama de actividades que modela la solución.	33
3.1	Interfaz del sistema.	47
3.2	Resultados de aplicar el algoritmo.	48
3.3	Resultados de aplicar el framework MATLAB Unit Testing.	49
3.4	Código del método ecualizar_histograma.	50
3.5	Representación del grafo de flujo de camino básico de ecualizar_histograma.	51
3.6	Resultado de las pruebas de aceptación.	56
3.7	Imagen original cargada satisfactoriamente con su correspondiente histograma.	56
3.8	Selección manual de la región circular.	57
3.9	Imagen ecualizada con su correspondiente histograma expandido.	57
3.10	Aplicación del filtro Promedio para la reducción de ruido en la imagen.	58
3.11	Resultado Final de la segmentación por textura usando Markov.	58
3.12	Caso de estudio 1	60
3.13	Caso de estudio 2	60
C.1	Representación del grafo de flujo de camino básico del método cargar_imagen.	82
C.2	Carta de aval del resultado de la investigación.	84
C.3	Carta de aceptación del resultado de la investigación.	85

Índice de tablas

1.1	Comparación entre las metodologías tradicionales y ágiles.	24
2.1	Historia de usuario # 1	35
2.2	Historia de usuario # 2	36
2.3	Estimación de esfuerzo por Historia de Usuario	36
2.4	Plan de duración de las iteraciones	38
2.5	Plan de entrega de versiones	38
2.6	Tarjeta CRC # 1	39
2.7	Tarjeta CRC # 2	39
2.8	Tarjeta CRC # 3	40
3.1	Puntos de estimación por historias de usuario de la iteración 1.	42
3.2	Tarea de desarrollo # 1	42
3.3	Tarea de desarrollo # 2	42
3.4	Tarea de desarrollo # 3	42
3.5	Tarea de desarrollo # 4	43
3.6	Tarea de desarrollo # 5	43
3.7	Puntos de estimación por historias de usuarios de la iteración 2.	44
3.8	Tarea de desarrollo # 6	44
3.9	Tarea de desarrollo # 7	45
3.10	Tarea de desarrollo # 8	45
3.11	Tarea de desarrollo # 9	45
3.12	Puntos de estimación por historias de usuarios de la iteración 3.	46
3.13	Tarea de desarrollo # 10	46
3.14	Tarea de desarrollo # 11	46
3.15	Tarea de desarrollo # 12	47
3.16	Caso de prueba de caja blanca para el camino básico 2	52
3.17	Caso de prueba de aceptación de la HU 1.	53
3.18	Caso de prueba de aceptación de la HU 5.	54
3.19	Caso de prueba de aceptación de la HU 6.	55
3.20	Resultados de la evaluación	60

A.1	Historia de usuario # 3	69
A.2	Historia de usuario # 4	69
A.3	Historia de usuario # 5	70
A.4	Historia de usuario # 6	70
A.5	Historia de usuario # 7	70
A.6	Historia de usuario # 8	71
A.7	Historia de usuario # 9	71
A.8	Historia de usuario # 10	72
A.9	Historia de usuario # 11	72
A.10	Historia de usuario # 12	72
A.11	Historia de usuario # 13	72
A.12	Historia de usuario # 14	73
B.1	Tarjeta CRC # 4	74
B.2	Tarjeta CRC # 5	74
B.3	Tarjeta CRC # 6	74
B.4	Tarjeta CRC # 7	75
C.1	Caso de prueba de aceptación de la HU 2.	77
C.2	Caso de prueba de aceptación de la HU 3.	78
C.3	Caso de prueba de aceptación de la HU 4.	79
C.4	Caso de prueba de aceptación de la HU 7.	80
C.5	Caso de prueba de aceptación de la HU 8.	81
C.6	Caso de prueba de aceptación de la HU 12.	82
C.7	Caso de prueba de caja blanca para el camino básico 1	83

En la actualidad la enfermedad de catarata es una de las principales causas de ceguera en el mundo, estimándose que cuenta con la mitad de los casos de ceguera en la población mundial. La catarata corresponde a una opacificación del cristalino que interfiere en forma progresiva con la agudeza visual, lo cual trae como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, dando lugar a una serie de imágenes difusas. Tiene diversas causas, pero se le atribuye mayormente a la edad, acelerando este proceso si el paciente padece de enfermedades como la diabetes o hipertensión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad [1] [2]. La catarata continúa siendo un problema de salud pública en la mayoría de los países subdesarrollados o en vías de desarrollo, debido a la carencia de especialistas y a un incremento de la expectativa de vida de la población, lo que lleva a un aumento de la prevalencia de esta enfermedad, ligada al envejecimiento [1] [2] .

A diferencia de otras enfermedades las catarata tienen tratamiento quirúrgico altamente efectivo que permite recuperar la visión parcial o totalmente en la mayoría de los pacientes, con una baja tasa de complicaciones. Durante las últimas décadas se han desarrollado distintas técnicas quirúrgicas para el tratamiento de las catarata, que han ido experimentando modificaciones con el objetivo de reducir las complicaciones intra y postoperatorias.

En muchos casos la cirugía no es un éxito a largo plazo, debido a que los pacientes pueden presentar complicaciones postoperatorias. La complicación postoperatoria más frecuente a largo o mediano plazo es la Opacidad de la Cápsula Posterior (OCP). La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata. Para garantizar una buena recuperación visual del paciente es imprescindible la transparencia de esta estructura [2].

Su incidencia en la actualidad se encuentra entre 0,7 % y 47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados. En Cuba la incidencia de la OCP asciende hasta el 50 % de los casos [2].

La OCP o catarata secundaria representa la principal causa de disminución en la visión de los pacientes operados de catarata. Existen varias maneras de tratar la catarata secundaria, desde la cirugía convencional (capsulotomía posterior) hasta la que emplea el láser neodimium [3]. Sin embargo, la aparente simplicidad

de estos tratamientos no implica una total inocuidad [4]. Por ello, los especialistas deben conocer los mecanismos que conducen a esta complicación y encontrar la forma de evitarlas, ya que la OCP incluso en grado muy leve puede ocasionar una desproporcionada disminución de agudeza visual en el caso del lente multifocal e interfiere con el adecuado funcionamiento de las lentes acomodativas. La opacificación es inevitable en la edad pediátrica y un retraso en su diagnóstico puede causar ambliopía [5].

La medición de los resultados tanto de la cirugía de cataratas, como de la capsulotomía, se ha centrado fundamentalmente en la agudeza visual, ya que su finalidad en ambos casos es la de mejorarla, y por tanto ganar en función visual, considerando que esto implica un beneficio en la calidad de vida.

Disponer de un método objetivo y estandarizado para la evaluación de la opacificación de la cápsula posterior es fundamental para investigar y medir la influencia de modificaciones de la técnica quirúrgica o diseño de las lentes intraoculares y para documentar la progresión de la opacificación [5]. Existen diversos métodos para valorar la opacificación capsular con diferentes grados de validación y objetividad, pero actualmente, no existe consenso sobre cuál de ellos es el sistema óptimo. Los métodos objetivos requieren un sistema de captación estandarizado y un sistema automático de evaluación.

La captación de imágenes digitalizadas es particularmente adecuada para la evaluación de la opacificación capsular ya que permite un acceso rápido a las imágenes y su almacenamiento para un análisis posterior computarizado. La ventaja principal de estos sistemas radica en que se reduce la variabilidad del observador y se incrementa la exactitud [5]. La importancia de utilizar métodos automatizados confiables que identifique la opacidad de capsula posterior en los pacientes reside no sólo en la relevancia de evitar las consecuencias de la enfermedad o en el hecho de impedir que el enfermo se someta a un segundo procedimiento, sino también en el gasto económico que este conlleva y en el tiempo que tarda el paciente en reincorporarse a sus actividades diarias y profesionales [6] [7].

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Grupo de Investigación AIRI (Artificial Intelligence Research and Innovation), el cual tiene como principal resultado y línea de investigación el software PANDOC (Programa Analizador de Opacidad Capsular), realizado en colaboración con el Instituto Cubano de Oftalmología Ramón Pando Ferrer. El software PANDOC provee al oftalmólogo de una herramienta capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), para lograr una evaluación objetiva del grado de la misma. Para ello el especialista debe identificar manualmente los puntos que considera opacos, valoración que depende mucho de su nivel de experiencia, y se desea evitar este grado de subjetividad.

Una de las limitaciones del software PANDOC es que trabaja con imágenes resultantes de la cámara de Scheimpflug del Pentacam el cual es un equipo oftalmológico de alta tecnología capaz de reconstruir imá-

genes tridimensionales de alta resolución del polo anterior del ojo, pero actualmente Cuba cuenta con pocos equipos como este, por lo que se hace necesario utilizar lámpara de hendidura.

La lámpara de hendidura es otro de los dispositivos óptico también muy utilizado en oftalmología para la exploración de las estructuras de la porción anterior del ojo. Las imágenes provenientes del artefacto constituyen una buena base para el análisis de la opacificación de la cápsula posterior, ya que una estructura transparente como la córnea o el cristalino, no puede ser analizada con una iluminación directa por lo que se utiliza una fina hendidura luminosa incidiendo oblicuamente. La hendidura proporciona una mejor percepción del relieve y ubica el sitio de la opacidad, a este tipo de imágenes obtenidas se les denomina oblicuas provenientes de la lámpara de hendidura.

Actualmente, en la comunidad médica internacional no existe consenso acerca de la correcta detección de la OCP, ya que muchos métodos dependen del sesgo del observador y requieren tiempo, por lo que cada especialista tiene la elección de seleccionar la herramienta o el sistema que considere adecuado [5].

Dada la situación problemática expuesta anteriormente, se plantea como problema a resolver: ¿Cómo disminuir el sesgo intraobservador en la identificación de la opacidad capsular posterior en imágenes oblicuas provenientes de la lámpara de hendidura?

A raíz de esto, se precisa como Objeto de estudio: Segmentación y mejora de imágenes médicas, enmarcándose en el Campo de acción: Estructuras de OCP en imágenes oblicuas de la lámpara de hendidura en pacientes operados de catarata. En aras de dar solución al problema planteado se define como Objetivo general: Desarrollar un algoritmo que permita disminuir el sesgo intraobservador en la identificación automática de las regiones de OCP mediante imágenes oblicuas provenientes de la lámpara de hendidura.

Teniendo en cuenta el problema a resolver se formuló la siguiente Idea a Defender: Con el diseño e implementación de un algoritmo para la identificación de regiones con opacidad capsular, entonces se dotará al oftalmólogo de una herramienta para la detección automática de la incidencia de la OCP en pacientes sometidos a cirugía de catarata.

Para dar cumplimiento al objetivo propuesto se ejecutaron las siguientes Tareas :

- Identificación de las diferentes técnicas de segmentación de imágenes médicas.
- Caracterización de las técnicas de segmentación en imágenes médicas.
- Estudio del uso de algoritmos de segmentación en imágenes médicas.
- Descripción de los pasos a seguir para el uso del algoritmo propuesto.
- Diseño de la solución propuesta en función de los requisitos especificados.
- Implementación de las técnicas de segmentación de OCP en imágenes oblicuas.

- Implementación del algoritmo de identificación de OCP propuesto.
- Validación de la implementación del sistema a partir de la aplicación de pruebas.

La lógica de la investigación se desarrolla a partir de **indagaciones teóricas y empíricas** en las cuales rige el **enfoque general dialéctico materialista del conocimiento**.

En el **nivel teórico** del conocimiento se utilizaron los métodos siguientes:

Análisis documental: Permitió realizar el estudio bibliográfico como aspecto esencial en la conformación del marco teórico referencial de la investigación, como sustento de las valoraciones realizadas, así como el estudio de los contenidos relacionados con la identificación de la OCP en pacientes operados de catarata y principales técnicas de preprocesamiento y segmentación de imágenes médicas digitales.

Análisis histórico - lógico: Posibilitó la aproximación a la evolución de los referentes teóricos del tema, analizar diferentes criterios relacionados con 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 logrará un mayor entendimiento y comprensión de este tipo de sistemas, basándose en las experiencias existentes.

Enfoque sistémico: Permitió el estudio del objeto de la investigación y su proyección, la determinación de nexos y las relaciones para determinar tendencias y regularidades.

Sistematización de experiencia: Contribuyó a identificar ideas, aportaciones teóricas y prácticas de los estudios realizados.

Dentro de los métodos del **nivel empírico** del conocimiento se destacan los siguientes métodos :

Entrevista: Proporciona datos importantes acerca de las necesidades de los especialistas en la identificación de las regiones con opacidad.

Sistematización de experiencias: Posibilitó la obtención de valoraciones empíricas de la factibilidad en la aplicación de esta herramienta, para la detección automática de la incidencia de la OCP en pacientes sometidos a cirugía de catarata.

Fundamentación Teórica

En el siguiente capítulo se realizará una disquisición desde el punto de vista teórico del problema general en que se enmarca la investigación. Se abordan los principales conceptos asociados al dominio del problema, y se realiza un estudio del arte del procesamiento de imágenes, haciendo énfasis en las principales técnicas de preprocesamiento y los métodos de segmentación existentes en imágenes médicas, para luego identificar un algoritmo que detecte regiones con opacidad capsular en pacientes sometidos a cirugías de cataratas.

1.1. Opacidad de la Cápsula Posterior: Conceptualización

La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata, su transparencia es imprescindible para una buena recuperación visual del paciente. 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 [8]. En el campo de la oftalmología esto es conocido como OCP y constituye la complicación tardía más importante en la cirugía de catarata en los tiempos modernos [9].

1.1.1. Incidencia y prevención de la Opacidad de la Cápsula Posterior.

La incidencia de la OCP en pacientes operados de catarata en la actualidad se encuentra entre 0,7 % y 47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados [2]. Diferentes investigaciones han sido dirigidas a identificar los principales factores que influyen en la aparición de esta complicación posoperatoria para su posible prevención, entre estos se encuentran los relacionados con el diseño de los lentes intraoculares (LIOs), como se describe en el estudio de Findl [10], en el cual se muestran datos referentes a la geometría de los LIOs y como esta puede influir de manera determinante en la formación de la OCP; también se observa cómo influye la biocompatibilidad de la LIO en relación con el saco capsular,

ello se define basándose en la propiedad que tiene el material con el que está confeccionado, para inhibir la estimulación de la proliferación de las LEC.

Las modificaciones de las técnicas quirúrgicas son otro de los factores determinantes, estas incluyen el pulido del saco capsular y el control del tamaño de la capsulorrexia, [9] [2]; otro factor influyente son los implantes adicionales como los anillos de tensión capsular (Capsular Tension Ring: CTR), a pesar de su eficacia demostrada para la reducción significativa de la OCP, no logran eliminarla totalmente; este efecto limitado y el costo adicional de la cirugía han dificultado que se generalice su uso [2].

1.2. Procesamiento de Imágenes.

En los últimos años han surgido varias herramientas de software capaces de expresar el grado de incidencia de OCP. La ventaja principal de estos sistemas radica en que se reduce la variabilidad del observador y aumenta la exactitud, aunque en ocasiones no son los más favorables ya que estos sistemas trabajan con imágenes obtenidas de las lámparas de hendidura, dichas imágenes poseen una desventaja muy peculiar al presentar una mancha pronunciada de luz o reflejo especular, producto del resultado de la reflexión del artefacto utilizado, y son denominadas imágenes de Purkinje [9] [11]. Por todo lo expuesto anteriormente es necesario realizar un proceso de mejora y tratamiento de la imagen digital para disminuir la proyección de la luz en las imágenes y de esta manera adquirir una imagen mejorada para la obtención de resultados investigativos que den cumplimiento al objetivo propuesto.

Una de las áreas más ligadas a la visión computacional es el procesamiento de imágenes, el cual tiene como objetivo realizar de forma sistemática un conjunto de operaciones con el fin de reformar y adaptar una imagen original para su posterior utilización o interpretación [12][13].

El tratamiento digital de imágenes consiste en hacer el mapeo de una imagen a puntos definidos discretamente, a los cuales se les asigna un par de coordenadas y un valor de intensidad. La alteración de los valores de intensidad por medio de una computadora permite efectuar con gran facilidad operaciones de realce y de análisis de la imagen [14].

El procesamiento de imágenes se divide en cinco etapas fundamentales [15]:

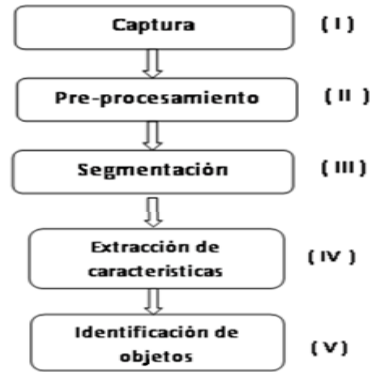


Figura 1.1. Etapas del procesamiento de imágenes .

Durante el presente trabajo la investigación estará englobada hasta la etapa de segmentación de la imagen, el cual es objetivo fundamental de la investigación para cuantificar objetivamente la opacidad de la cápsula posterior.

1.2.1. Captura de imágenes digitales.

La idea general de la captura o adquisición de una imagen, es llevar la imagen dentro de la computadora, donde pueda ser almacenada o visualizada para luego ser manipulada y mejorada. El principal elemento en la adquisición de una imagen es una cámara que captura las imágenes de un objeto. Entre las modalidades de adquisición de imágenes médicas se encuentra el Pentacam el cual es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo. Esto lo realiza a partir de múltiples fotografías tomadas mediante una cámara rotacional del sistema *Scheimpflug* con que cuenta el equipo. El análisis de estos tomogramas en 3D puede potencialmente ser utilizado para la cuantificación objetiva de la OCP [9]. El principal inconveniente para la utilización del artefacto es que en el país solo se cuenta con pocos equipos, por lo que se hace necesario trabajar con otro equipo para la adquisición de las imágenes. Entre las modalidades de adquisición de imágenes médicas se encuentra las lámparas de hendiduras, el cual es uno de los instrumentos de diagnóstico más comúnmente usados por un oftalmólogo. Proporciona iluminación y magnificación para examinar las varias partes del ojo. La luz se proyecta como una franja o hendidura brillante, lo que permite el examen detallado del ojo en pequeños segmentos. Se utiliza en el examen del segmento anterior del ojo, incluyendo el lente cristalino. Con lentes suplementarios la lámpara de hendidura es útil en el examen de la región posterior del ojo, el fondo del ojo y buena parte de la retina. Una serie de accesorios se pueden añadir a una lámpara de hendidura para convertirlo en un instrumento de medida. Uno puede medir la presión intraocular, la curvatura de la córnea, el espesor de la córnea, la distancia entre la córnea y el lente, el volumen de la cámara anterior, la opacidad, etc. utilizando accesorios diferentes [16]. Una de las ventajas que tiene la utilización de este artefacto es que

en el país por cada consulta de oftalmología se encuentra como mínimo una lámpara de hendidura, lo que hace que la mayoría de las imágenes utilizadas para la identificación de la OCP son provenientes del mismo. Las imágenes utilizadas para la identificación de la OCP en la presente investigación son imágenes oblicuas, ya que la luz provenientes de la lámpara de hendidura incide oblicuamente sobre las mismas. Estas imágenes presentan cierta desventaja como resultado de la reflexión de la luz proveniente del artefacto, por lo que se hace necesario realizar un preprocesado para lograr resultados favorables en la etapa segmentación.

1.2.2. Preprocesamiento de imágenes

Al digitalizar una imagen, es común la presencia de ruido o degradación, así como otros efectos indeseados como los reflejos luminosos ya mencionados, razón por la cual es importante restaurarla antes de ser procesada. En la etapa de preprocesamiento de imágenes se intenta aumentar la calidad de la imagen con el fin de reconocer mejor las fallas que puedan existir en ella. Algunas de las técnicas empleadas en el preprocesamiento son eliminación de ruido mediante filtros digitales, mejora del contraste y restauración [17].

Mejora del contraste

A continuación se muestran algunas de las técnicas utilizadas para mejorar el contraste en las imágenes.

La ecualización adaptativa del histograma (CLAHE) es una técnica muy utilizada para mejorar el contraste en las imágenes. Un histograma es una función que muestra una imagen con los distintos niveles de gris que contiene. La ecualización del histograma consiste en una expansión del histograma, dotando al mismo de mayor linealidad y haciendo que éste ocupe el ancho del espectro de tonalidades grises por completo, lo cual implica una mayor utilización de los recursos disponibles y un aumento del contraste.

El Algoritmo White Patch (WP), o Parche Blanco en Español toma en cuenta el valor más grande en cada componente de color como una representación del blanco de la imagen [18]. Computacionalmente, este parche blanco es calculado al encontrar la máxima intensidad en cada canal. El algoritmo WP puede hacerse más robusto si se calcula un histograma para cada componente de color y se considera el iluminante como un alto porcentaje de la acumulación del mismo [18]. Este método es usado frecuentemente para remover sombras en la imagen .

El Algoritmo Gray World (GW), o Mundo Gris en Español es el método más conocido de constancia de color, usado como referencia por otros algoritmos [18], el GW está basado en la suposición de que en promedio, el mundo es gris, y estima el iluminante usando el color promedio de todos los píxeles. Se asume que la información dada por el promedio de cada canal de la imagen representa el nivel gris [18].

Eliminación de ruido mediante filtros digitales

A continuación se muestran algunas de las técnicas utilizadas para disminuir el ruido en las imágenes mediante la utilización de filtros digitales.

Los filtros son operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio [19] [13].

El dominio de frecuencia se basa en modificaciones de la transformada de Fourier. Este método trata de calcular la transformada de Fourier de la imagen a intensificar, multiplicar el resultado por la función de transferencia de un filtro y, finalmente, tomar la transformada de Fourier inversa para llegar a una imagen mejorada. El dominio espacial trabaja con técnicas de manipulación de píxeles de la imagen [17][15]. En este proceso se hace corresponder, para cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objeto con la finalidad de obtener una información útil, dependiendo del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado. De este modo se puede mejorar la imagen y obtener datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella. Existen 2 tipos de filtros para el dominio del espacio: Filtros lineales (filtros basados en Kernels o máscaras de convolución) y filtros no lineales [15].

Los filtros lineales se basan en el concepto que afirma, que la función de transferencia y el impulso o función de distribución puntual de un sistema lineal son transformadas de Fourier [19]. Se encuentran en él los denominados filtros de bajas frecuencias o paso bajo que son aquellos que en el dominio de las frecuencias dejan pasar solamente las frecuencias bajas y elimina altas frecuencias, se caracterizan por suavizar los bordes y dar la impresión de imágenes borrosas. Su objetivo es suavizar la imagen, son de gran utilidad cuando la imagen presenta una cierta cantidad de ruido que se desea eliminar [13].

Algunos de los filtros lineales son:

Filtros gaussianos: Simulan una distribución gaussiana bivariente. El valor máximo aparece en el píxel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro de desviación típica. El resultado será un conjunto de valores entre 0 y 1. Para transformar la matriz a una matriz de números enteros se divide toda la matriz por el menor de los valores obtenidos. Este filtro presenta desventajas ya que puede producir la pérdida de detalles, el aumento de borrosidad, y la disminución de la nitidez.

Filtro de la media: Consiste en sustituir cada píxel por la media aritmética de los puntos que tiene alrededor (incluido el mismo). Se toma una ventana de $N \times N$ puntos, donde el punto a sustituir es el central, se toma la media de la suma de los valores de los píxeles presentes en la ventana y se sustituye el píxel en la nueva imagen por el valor obtenido.

Filtro de media ponderada: Los elementos de la matriz de filtrado no son todos 1 sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa [13].

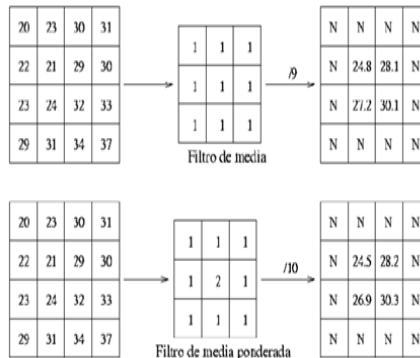


Figura 1.2. Ejemplo de filtros lineales de paso bajo .

Filtros adaptativos: Son considerablemente más complejos ya que los coeficientes de ponderación se recalculan para cada uno de los píxeles en función del histograma. Se han utilizado con gran éxito filtros adaptativos para eliminar el ruido de las imágenes de radar y para detectar con un solo filtro, diferentes elementos.

En contraste los filtros de paso alto tienen como objetivo resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media, precisamente la que detectan los filtros de paso bajo. Por otra parte, la respuesta de cada píxel está contaminada por la de los píxeles vecinos ya que la radiación reflejada por un píxel se reparte hacia los píxeles vecinos. Los filtros de paso alto consiguen también eliminar en parte esta contaminación. Existen diversos métodos:

Sustracción de la media: Si se considera que un filtro de paso bajo sirve para resaltar componentes a gran escala eliminando la variabilidad local. La matriz de filtrado de este filtro de sustracción de la media puede calcularse directamente restando a la matriz de filtrado identidad [13].

Filtro laplaciano: Responde a las transiciones de intensidad, rara vez se utiliza en la práctica para la detección de bordes. Como es una derivada de segundo grado es inaceptablemente sensible al ruido. Además produce bordes dobles y es incapaz de detectar direcciones de borde. Un empleo más general del Laplaciano consiste en encontrar la ubicación de bordes utilizando sus propiedades de paso por cero [19].

Filtros no lineales

Existen otros filtros espaciales que no son función lineal de los valores de brillo de los píxeles de una imagen. Es decir, no se calculan como una suma lineal de elementos (brillo de los píxeles) multiplicados por pesos constantes (coeficientes de la máscara). Estos filtros se conocen como filtros espaciales no lineales. Constituyen también técnicas de procesamiento por grupo de píxeles, operando sobre un núcleo de píxeles de entrada que rodean a un píxel central. La diferencia es que en lugar de utilizar un promedio ponderado, emplean otras técnicas que combinan los valores de brillo del grupo de píxeles de entrada.

Alguno de los filtro no lineales son:

Filtro de la mediana: Se realiza mediante la adopción de la intensidad de todos los píxeles en una vecindad ordenados en un vector de acuerdo con sus valores. El píxel que tenga la magnitud de la mediana del ordenamiento se utiliza entonces para reemplazar el píxel central de la vecindad [15].



Figura 1.3. Ejemplo de filtro no lineal .

Filtro de la moda: La moda de un conjunto de valores se define como el valor que más se repite dentro de ellos. Por lo tanto, el filtro de moda consiste en calcular el valor más repetido dentro de todos los píxeles de una ventana [13].

Las técnicas de tratamiento no lineal de imágenes digitales son a menudo mejor que los filtros lineales en remover el ruido sin distorsiones en las características de la imagen. Sin embargo, el diseño y el análisis de filtros no lineales es mucho más difícil que para filtros lineales. Una vía para diseñar filtros no lineales es la morfología matemática, la cual crea filtros basados en las características de forma y tamaño [17].

Los operadores morfológicos permiten determinar cuáles son las estructuras relevantes de una imagen para transformarlas de acuerdo a algún propósito predefinido. Generalmente, para esto se utiliza un conjunto denominado elemento estructurante o *strel* , el cual permite explorar la imagen a investigar. La forma y el tamaño del elemento estructurante es seleccionada considerando la geometría de las estructuras relevantes e irrelevantes de la imagen [20]. Las operaciones morfológicas utilizadas principalmente para el procesamien-

to de la imagen son: Erosión, Dilatación, Apertura y Cierre. Para aplicar las operaciones morfológicas, el elemento estructurante debe tener un origen o punto de referencia, el cual permite trasladarlo a diferentes puntos o píxeles de la imagen [21].

La operación de erosión, junto a la dilatación son la base de cualquier transformación morfológica. Cualquier operador, transformación o algoritmo, incluirá una erosión, una dilatación o ambas primitivas en su implementación. Sin necesidad de formar nuevos operadores es posible encontrar aplicaciones interesantes en las transformaciones básicas.

La erosión es una operación que permite, en general, hacer disminuir el tamaño de un conjunto, en particular, una imagen binaria. Así, para un conjunto de prueba, el conjunto erosionado está formado por todos sus puntos que coinciden con el punto de referencia del elemento estructurante, cuando este está completamente incluido en el conjunto de prueba. El resultado es una señal de menor valor, es decir, una imagen más oscura, puesto que la erosión pretende minimizar el valor de la señal que en el caso de los grises tiene una definición de [0,255] [21][19].

La dilatación es la operación dual de la erosión. Si se considera un conjunto en una imagen binaria, la dilatación de este conjunto estará formada por todos los puntos de la imagen, que coinciden con las diferentes traslaciones del punto de referencia del elemento estructurante cuando este intersecta al conjunto. El resultado de la dilatación en señales bidimensionales de escala de grises (imágenes) es generalmente una señal de mayor valor, es decir, una imagen más clara, puesto que la dilatación maximiza el valor de la señal, aumentando la definición de los objetos claros frente a los oscuros.

Las operaciones de dilatación y erosión no admiten inversa, por lo que no hay manera de obtener la imagen original desde la imagen dilatada ni erosionada. Sin embargo, es posible mediante la adjunción de operadores básicos aproximarse a la forma original. Se puede intentar recuperar el elemento original X erosionado si una vez efectuada la erosión, se realiza una dilatación, adjunción de operaciones conocida como apertura. Lo mismo ocurre si se efectúa una dilatación seguida de una erosión, proceso conocido como cierre.

1.2.3. Segmentación de imágenes

La segmentación, es un proceso que en visión artificial de imágenes proporciona información útil de regiones y bordes [22]. La misma consiste en subdividir la imagen en segmentos que corresponden a los diferentes objetos existentes [23]. El nivel al que se hace la segmentación depende del problema a resolver, por esto deberá detenerse cuando se hayan aislado los objetos de interés. La segmentación automática de una imagen es una de las etapas más complejas del procesamiento de imágenes, ya que no se tiene una información adecuada de los objetos a extraer y también por la eventual presencia de ruido en la imagen a segmentar. De la segmentación depende que el análisis de la imagen sea acertado. De ahí la importancia de

aplicar técnicas que aumenten la posibilidad de tener una segmentación robusta.

Los métodos para llevar a cabo las segmentaciones varían ampliamente dependiendo de la aplicación específica, tipo de imagen y otros factores [24]. En la actualidad no existe un método de segmentación que alcance resultados aceptables para todo tipo de imagen, por lo tanto, la selección de un método apropiado para un problema de segmentación puede ser muy difícil.

La segmentación de imagen tiene diferentes clasificación, cada una para resolver diferentes problemas. Los algoritmos de segmentación generalmente están basados en dos criterios: uno es la homogeneidad de la región y otro la discontinuidad entre regiones disjuntas adyacentes [20], los principales métodos que aplican estas propiedades son [23]:

- Segmentación por detección de bordes, donde los segmentos de la imagen se obtienen tomando en cuenta la transición brusca en el nivel de gris o color, esto es, la discontinuidad.
- Segmentación por detección de regiones, donde se analiza la similitud entre los píxeles que formarán un segmento.

Segmentación por detección de borde

En el caso de la segmentación basada en la **detección de borde**, esta contiene un gran número de técnicas que se basan en analizar los límites de los objetos de una imagen. Los bordes pueden determinarse con operadores de detección, pero es necesario aplicar procesos posteriores para enlazar los bordes obtenidos de una manera más precisa y así obtener segmentos perfectamente delimitados.

El objetivo de esta técnica es resaltar las fronteras del objeto a segmentar [25] [26], tiene como ventaja la simplicidad, una vez encontradas las fronteras, pero a su vez presenta dificultades en encontrar estas fronteras (borde), requiriendo operadores paso alto de detección de bordes (Laplaciana, Sobel, Canny) mostrando alto nivel de ruido [25] [26]. Esta técnica utiliza operadores sofisticados y la construcción de fronteras se hace a partir de bordes

Una de las técnicas más utilizadas para la detección de bordes es mediante el uso de operadores diferenciales, en los cuales se realiza un procesamiento espacial de la imagen digital $f(x, y)$ y se obtiene una imagen diferencial $G(x, y)$ que indica donde se encuentran las discontinuidades de la imagen original[23]. Existen dos tipos de operadores diferenciales detectores de bordes: de primero y de segundo orden. Los de primer orden corresponden al gradiente (primera derivada), y los de segundo al laplaciano (segunda derivada)[23]. A continuación se realiza una breve explicación de los principales operadores para la detección de borde:

Operador de Sobel

Los operadores de Sobel trabajan con el peso del píxel central de las filas y columnas externas de las máscaras, lo que hace que los bordes detectados sean más definidos [23]. Con los operadores de Sobel se obtienen los bordes y además proporcionan un efecto de suavizado, ya que detectan bordes principales[23]. Este efecto es una ventaja considerando que las derivadas realzan el ruido. El operador de Sobel se encuentra dentro

de los operadores de primer orden.

Operador de Canny

El Algoritmo de Canny fue desarrollado por John Canny en 1986 [27]. Este método está considerado como uno de los más eficientes para la detección de bordes mediante el empleo de máscaras de convolución, porque es menos sensible al ruido y es capaz de detectar bordes débiles verdaderos [23]. Permite detectar los bordes presentes en una imagen. Este algoritmo utiliza la primera derivada, la cual toma el valor de cero en las regiones donde no varía la intensidad de los píxeles y toma un valor constante en toda la transición de intensidad [27]. Un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada, la detección de bordes reduce significativamente la cantidad de datos a procesar [27].

El algoritmo consiste básicamente en la aplicación de un filtro Gaussiano que suaviza la imagen y reduce el ruido, para luego calcular el gradiente de esta respuesta [23]. En la figura 1.4 se muestra el resultado de aplicar dicho método.

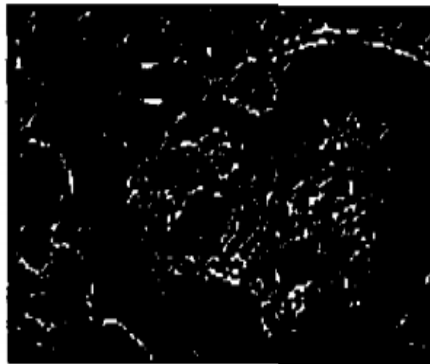


Figura 1.4. Imagen obtenida luego de aplicar el operador de Canny.

El Algoritmo de Canny consta de tres etapas principales: cálculo del gradiente, supresión no máxima e histéresis de umbral [23] [28] .

Operador Laplaciano

Los métodos precedentes se desempeñan de mejor forma cuando el cambio del nivel de gris o color es bastante brusco. Cuando la región de transición se hace más amplia, es más conveniente aplicar derivadas de segundo orden [23]. Un operador de segundo orden común es el Laplaciano, definido como [23]:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1)$$

Esta ecuación también puede aproximarse de varias maneras para su implementación en forma digital. Una de las más utilizadas corresponde a la máscara de la figura 1.5.

0	-1	0
-1	4	-1
0	-1	0

Figura 1.5. Máscara utilizada para aplicar el Operador Laplaciano.

Segmentación por crecimiento de regiones

Los enfoques de segmentación basados en el crecimiento de regiones buscan delimitar zonas homogéneas de la imagen, mediante la inclusión progresiva de los elementos que cumplen determinado criterio de similitud [29] [30] [31]. Específicamente lo que se hace es extraer regiones de la imagen que están conectadas bajo cierto criterio predefinido [32] [25]. Este criterio puede estar basando en informaciones de intensidades y/o bordes de la imagen. En su forma más simple, este método requiere un punto semilla que es seleccionado manualmente por el usuario, y que extrae todos los píxeles conectados a la semilla, que tengan el mismo valor de intensidad [32].

El objetivo de esta técnica es resaltar regiones de características similares, esta técnica enfoca la segmentación como un problema de clasificación de píxeles o grupos de píxeles. Es una técnica muy simple de utilizar pero presenta inconvenientes y dificultades para encontrar características discriminantes en el proceso de clasificación [26]. Se basa en el nivel de gris y usa parámetros estadísticos de textura, varianza, movimiento, entre otras [26].

Entre los métodos de segmentación que trabajan bajo el principio de crecimiento de regiones se puede mencionar:

- El método de segmentación propuesto por la Universidad del Sur de la Florida (USF) que consiste en realizar un crecimiento de regiones a partir de un píxel semilla [33].
- El método propuesto por la Universidad Estatal de Washington (WSU) que realiza un agrupamiento utilizando un algoritmo para cluster [33].
- El método de segmentación propuesto por la Universidad de Edimburgo (UE) basado en el cálculo de curvaturas medias y gaussianas [33].
- El método de segmentación propuesto por la Universidad de Bern (UB) que hace un crecimiento de regiones agrupando segmentos de líneas en vez de puntos [33].
- El método de Besl y Jain que sigue el principio de crecimiento de regiones para detección de superficies curvas y planas y ajusta superficies de orden variable [34].

Los métodos de Crecimiento de Regiones Basados en Semillas han sido ampliamente usados en segmentación de imágenes médicas [35] [36]. A partir de píxeles iniciales adecuadamente escogidos se comienza un proceso iterativo que asigna, paso a paso, píxeles vecinos al píxel inicial con propiedades similares a éste (como un rango específico de nivel de gris o color) [37], de acuerdo con un criterio de uniformidad seleccionado [38].

En general, la determinación del punto de partida para la segmentación suele ser un aspecto crítico del algoritmo. Por este motivo en casi todos los sistemas es el usuario experto quien selecciona el punto semilla para iniciar el algoritmo.

Considerando todo lo visto de estos métodos de segmentación, se puede concluir que son muy útiles en el proceso de segmentación, pero cada uno de ellos tiene ciertas desventajas, las cuales son de gran importancia dominar para lograr un buen desarrollo de los métodos. En el caso de los métodos basados en el crecimiento de regiones tienen como desventaja que su eficiencia depende mucho de las regiones iniciales que se tomen como semilla, además, el crecimiento de regiones sin una previa guía de los límites, conduce a bordes mal definidos. Por otra parte, los métodos basados en la detección de bordes o de formas, se enfrentan con el problema de que encontrar bordes o formas en una imagen de rango no es una tarea fácil. Por ejemplo, en el caso de la detección de bordes muchas técnicas entregan como resultados bordes gruesos, incompletos, falsos, entre otros.

Existen otros métodos que son de gran ayuda en el proceso de segmentación, todo depende del problema que se quiera resolver, a continuación se mostrarán otros métodos utilizados en la segmentación de imágenes digitales.

Segmentación por umbral

La técnica basada en umbral (*thresholding*) segmenta imágenes creando una partición binaria de las intensidades de las imágenes [24]. Estas técnicas tratan de determinar un valor de intensidad llamado umbral (*threshold*), que separa las clases deseadas. La segmentación se logra agrupando todos los píxeles cuyo valor de intensidad es menor que el umbral por un lado y los píxeles con valores de intensidad mayores que el umbral por el otro (Figura 3.1) [24]. La selección del nivel de umbral T se obtiene generalmente a partir del histograma de la imagen [26]. El histograma muestra la frecuencia relativa de aparición de cada uno de los niveles de intensidad de gris de la imagen [26].

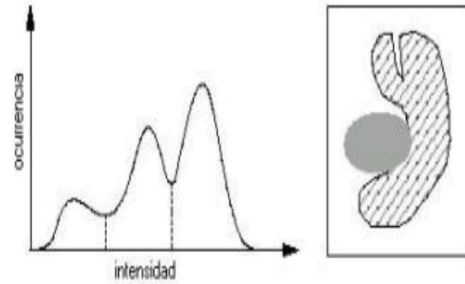


Figura 1.6. Histograma de intensidades de grises y a la derecha imagen en escala de grises.

La técnica del umbral es una técnica efectiva para obtener la segmentación de imágenes donde estructuras diferentes tienen intensidades constantes u otras características diferenciables [24]. Su principal limitación se encuentra en que su forma más simple sólo se genera dos clases y por lo tanto no se puede aplicar a imágenes multicanal. Además, la técnica de aplicación de umbrales usualmente no toma en cuenta las características espaciales de la imagen, lo que hace que sea sensible al ruido y a las no homogeneidades de intensidad [24]. Normalmente los métodos de aplicación de umbral convierten la imagen de partida en una imagen binaria. Es decir, se construyen dos segmentos, el fondo de la imagen y los objetos buscados. La regla de cálculo es la siguiente [24]:

$$T_{g\text{lobal}} g = \begin{cases} 0 & \text{si } g < t \\ 1 & \text{si } g \geq t \end{cases} \quad (2)$$

Donde $T_{g\text{lobal}}$ es el píxel segmentado, g es el valor de cada píxel y t es el valor del umbral.

El punto clave en esta técnica es la elección del valor umbral adecuado. Esto se puede realizar de forma manual, pero normalmente se necesita un método que permita calcular este valor de forma automática. El histograma de la imagen será el elemento más importante para definir el valor umbral, tanto si se elige hacerlo de una forma manual como automática.

Para el cálculo del umbral existen diferentes métodos, como por ejemplo el algoritmo Otsu, el cual calcula el valor umbral de forma que la dispersión (distancia entre los valores respecto a un valor medio) dentro de cada segmento sea la más pequeña posible, pero al mismo tiempo la dispersión sea lo más alta posible entre segmentos diferentes [24]. Para ello se calcula el cociente entre ambas varianzas y se busca un valor umbral para el cual este cociente sea máximo [24].

Segmentación por textura

En muchas imágenes los objetos no se encuentran diferenciados claramente en relación con el fondo, más bien tienen como característica espacial cierta textura visual [39], lo que dificulta la clasificación independiente de cada píxel, ya que la textura está definida por regiones de píxeles y no por píxeles individuales.

En las imágenes médicas ocurre esta situación, por ejemplo se pueden hallar valores similares de gris entre tejidos saludables y dañados, e incluso bordes no definidos claramente [40]. Normalmente se obtiene un vector de rasgos para cada píxel, con esto se pueden obtener características de acuerdo con el comportamiento del entorno del mismo [40]. Como las imágenes médicas con las que se quiere trabajar tienen este tipo de característica, se selecciona la segmentación por textura como la más adecuada para resolver el problema en cuestión.

La textura no es más que una característica que representa una cuantificación de la variación espacial de valores de tono de gris [41] y constituye una herramienta útil en la identificación de regiones de interés en una imagen. Está definida por la uniformidad, densidad, grosor, rugosidad, regularidad, intensidad y direccionabilidad de medidas discretas del tono y de sus relaciones espaciales [15].

El proceso de segmentación de textura consiste en realizar una partición de la imagen de entrada, en un conjunto de regiones disjuntas, haciendo que cada región sea homogénea con respecto a la característica de la textura [42]. Muchos trabajos se han realizado sobre distintos métodos para solucionar el problema, los cuales podrían resumirse principalmente en dos categorías [42]:

- Los métodos que consideran que los valores de cada uno de los píxeles de la imagen están regidos por una función de probabilidad, en cuyo caso el problema consiste en estimar los parámetros que la definen, para con ellos obtener la segmentación.
- Los métodos basados en aplicar un banco de filtros a la imagen, para obtener de cada uno de los filtros un valor por píxel para el vector de características que permiten luego realizar la segmentación.

Para obtener la textura existen tres enfoques: estructural, estadístico y espectral [15].

El enfoque estructural se aplica a texturas regulares que se pueden describir como formas básicas con propiedades invariantes, que ocurren en distintas posiciones y orientaciones en un área [15] y que se repiten uniformemente.

El enfoque estadístico caracteriza a las texturas no-regulares [43], utiliza rasgos estadísticos de la imagen, modelados como una función de los tonos de gris. Puede definirse como un conjunto de datos estadísticos extraídos de un gran conjunto de propiedades locales de la imagen [42]. Entre estos métodos se encuentran las matrices de co-ocurrencia propuesto por Haralick en 1973 [43] y los campos aleatorios de Markov (CAM) [44] [45] [46] [47].

Las técnicas del enfoque espectral tienen relación con la frecuencia de aparición de los niveles de gris en la imagen, lo cual se considera una de las características de las texturas [15] y puede tener un patrón periódico, que constituye la base de estos métodos. Emplean la transformada en frecuencia de la imagen y a partir de esta, ciertas características que sirven de base para la segmentación de la misma [40]. Entre estos métodos se encuentra el filtro de Gabor.

En este trabajo no es posible aplicar el enfoque estructural, pues las imágenes médicas presentan texturas

irregulares en gran parte de la zona de la imagen; por lo tanto, se determinó aplicar métodos del enfoque estadístico y espectral .

A continuación se explican algunos de los métodos principales para lograr la segmentación por textura:

Segmentación por textura usando matriz de co-ocurrencia

La idea fundamental de la matriz de co-ocurrencia, es contar el número de veces de aparición de un patrón en una textura [44]. Por ejemplo, la ocurrencia de pares de niveles de gris de píxeles que son separados por una cierta distancia y están en una cierta dirección. Diferentes matrices de co-ocurrencia pueden ser obtenidas usando distintas propiedades y a partir de cada una de ellas varias estadísticas pueden ser calculadas como medidas de textura: energía, entropía, correlación e inercia [44]. Una desventaja de la matriz de co-ocurrencia es que es posible derivar muchas características de textura y no existe una manera simple para extraer sólo una característica efectiva [48].

La matriz de co-ocurrencia C de niveles de grises de una imagen I para una distancia d y una orientación α está definida formalmente como [39]:

$$C_{d, \alpha}(i, j) = |\{(r, s)(t, v) : I(r, s) = i, I(t, v) = j\}| \quad (3)$$

Donde $((r,s),(t,v)) \in \mathbb{N} \times \mathbb{N}$ y $|\cdot|$ es el cardinal del conjunto. Cada elemento $C(i,j)$ de la matriz de co-ocurrencia es el número de veces que un píxel de valor i tiene una relación (d, α) con otros píxel de valor j . Normalmente se utilizan los siguientes 4 ángulos: 0° , 45° , 90° y 135° y las matrices obtenidas se combinan en una única matriz, que se normaliza con el empleo de la ecuación que a continuación se muestra.

$$P_{d, \alpha}(i, j) = \frac{C_{d, \alpha}(i, j)}{\sum_{l,k=0}^{G-1} (C_{d, \alpha}(l, k))} \quad (4)$$

Características calculadas a partir de la matriz de co-ocurrencia:

Haralick propone varias medidas que son posibles emplear a partir de la matriz de co-ocurrencia, de ellas las tres más utilizadas son el segundo momento angular, la desviación estándar y la correlación, junto a estas la media y la varianza, que son parámetros de dos de las anteriores [42].

Media: El valor del píxel se pondera por su frecuencia y por la frecuencia de su co-ocurrencia en combinación con el valor del píxel vecino [49].

$$\mu_i = \sum_{i,j=0}^{G-1} iP_{i,j} \quad \mu_j = \sum_{i,j=0}^{G-1} jP_{i,j} \quad (5)$$

Varianza: La fórmula para el cálculo de la varianza es la siguiente [49]:

$$\sigma_i^2 = \sum_{i,j=0}^{G-1} P_{i,j}(i - \mu_i)^2, \quad \sigma_j^2 = \sum_{i,j=0}^{G-1} P_{i,j}(j - \mu_j)^2, \quad \sigma_i^2 = \sigma_j^2 \quad (6)$$

Segundo Momento Angular (Energía): Mide la homogeneidad de una imagen. Cuanto más suave es la textura mayor valor toma ENE [49]:

$$ENE = \sum_{i,j=0}^{G-1} P_{i,j}^2 \quad (7)$$

Desviación estándar: Mide la desviación estándar de los niveles de gris [49].

$$\sigma_i = \sqrt{\sigma_i^2}, \quad \sigma_j = \sqrt{\sigma_j^2}, \quad \sigma_i^2 = \sigma_j^2 \quad (8)$$

Correlación: Mide las dependencias lineales de los niveles de grises y la similitud entre píxeles vecinos [50].

$$C = \sum_{i,j=0}^{G-1} \frac{P_{i,j}(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \quad (9)$$

Segmentación por textura usando filtro de Gabor

Al ser aplicado el filtro de Gabor a cierta frecuencia tendrá valores altos donde exista esta y atenuará las zonas donde no esté presente, lo que permite separar la imagen por regiones de frecuencias diferentes [40]. Este método tiene el inconveniente de que con un solo filtro de Gabor se obtiene información acerca de una única orientación y frecuencia, por eso se emplea otro método, denominado **Modelo de Gabor**, que utiliza un banco de filtros de Gabor para la detección de texturas en cualquier orientación y frecuencia [40].

La formula general del modelo de Gabor es la siguiente [40]:

$$g_{\lambda,\theta,\Psi}(X, Y) = 10^{\frac{(X^2 + Y^2)}{2\sigma^2}} * \cos\left(2\pi \frac{X}{\lambda} + \Psi\right), \quad X = x * \cos\theta + Y * \sin\theta, \quad Y = -x * \sin\theta + y * \cos\theta \quad (10)$$

Donde:

Y : Relación de aspecto horizontal / vertical o excentricidad del núcleo.

θ : Ángulo de inclinación de ondas paralelas del filtro.

σ, λ, Ψ : Parámetros relacionados con amplitud, varianza de la onda y fase del filtro. Características calculadas a partir del filtro Gabor:

Kruizinga propone varias medidas que pueden ser empleadas a partir de los filtros de Gabor, de ellas las más utilizadas son la varianza, la desviación estándar y la energía de activación [23].

Varianza: Es la medida del contraste del nivel de gris [40].

$$\sigma^2 = \sum_{i,j=0}^{G-1} G(i, j)(i - \mu)^2 \quad \mu = \sum_{i,j=0}^{G-1} G(i, j) \quad (11)$$

A partir de la varianza se halla un descriptor de suavidad, dado por la ecuación que a continuación se muestra y que es 0 para áreas de intensidad constante y se aproxima a 1 para valores grandes de σ^2 .

$$R = 1 - \frac{1}{1 + \theta^2} \quad (12)$$

Desviación Estándar: Se calcula mediante la expresión [40]:

$$\theta = \sqrt{\theta^2} \quad (13)$$

Energía de Activación: Es la energía emitida por el filtro [40]:

$$E = \sum_{i,j=0}^k G(i, j) \quad (14)$$

En la figura se muestra el procedimiento para aplicar el Modelo de Gabor.

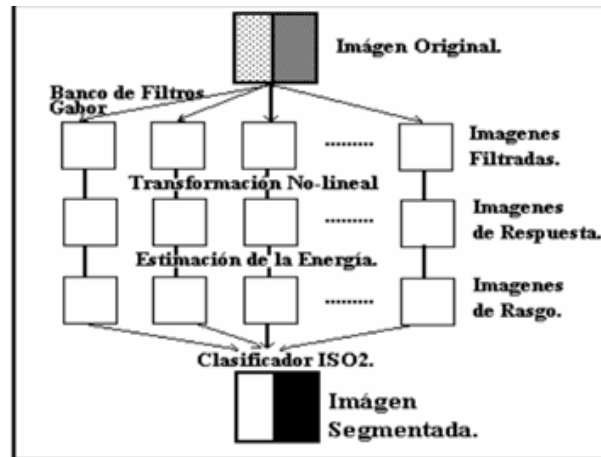


Figura 1.7. Procedimiento para aplicar el Modelo de Gabor.

Segmentación por textura usando campos aleatorios de Markov

Los CAM son un modelo estadístico que puede ser usado dentro de los métodos de segmentación [51]. Tienen la capacidad para integrar la información visual, en problemas de restauración y segmentación de imágenes [52]. Modelan las interacciones espaciales entre vecinos o píxeles cercanos. Estas correlaciones locales proveen un mecanismo para modelar una variedad de propiedades de la imagen [32]. Los CAM son incorporados frecuentemente en los algoritmos de segmentación por agrupamiento, como las K-medias bajo un modelo Bayesiano previo.

En el caso del modelo Bayesiano la solución óptima del problema se define a través de la estimación de probabilidad máxima a posteriori (MAP) y es calculada por minimizar la energía posterior [53]. La energía posterior dependerá del modelo a priori, en el cual se codifican las restricciones contextuales y el modelo de probabilidad que depende de los datos observados [46]. La definición de un modelo CAM válido, involucra especificar la estructura del modelo y los parámetros. En la estructura del modelo se selecciona el sistema de vecinos, la función de energía válida o modelo de probabilidad condicional. Mientras que, la estimación de los parámetros involucra un problema de optimización, el cual se resuelve comúnmente con técnicas como [45] [46]: ML (Maximum Likelihood), codificación (Coding), PML (Pseudo-likelihood), EM (Expectation-Maximization), etc.

Los parámetros de la energía posterior se asumen conocidos cuando se trata de una segmentación supervisada. Entretanto, en el caso no supervisado los parámetros son desconocidos (problema de datos incompletos [46]) y tienen que ser inferidos a partir de los datos de la imagen.

Definición de la Energía a Priori

$$U(x) = \sum_{s \in S} (\alpha_s x_s + \sum_{r \in X_{\eta(s)}} \beta_{sr} x_s x_r) \quad (15)$$

donde α_s y β_{sr} son los parámetros del modelo de variación espacial, los cuales representan el campo externo e interno respectivamente.

Definición de la Energía de Probabilidad

$$p(y/x) = \prod_{s \in S} p(y_s/x_s) \quad (16)$$

Energía Posterior

Lo que se busca es maximizar la probabilidad posterior $p(x/y)$ para encontrar el etiquetado x . Entonces la energía posterior a maximizar considerando la energía a priori $p(x)$ y la energía de probabilidad $p(y/x)$ definidas anteriormente es:

$$x_{MAP} = \operatorname{argmax}_x [\ln p(y/x) + \ln p(x)] \quad (17)$$

Dentro del método de Markov se utiliza el algoritmo ICM (Iterated Conditional Modes). ICM proporciona el mínimo local del campo más cercano a la configuración de partida (puede oscilar entre varios mínimos locales cercanos a la configuración de partida cuando la energía de estos es similar). No es considerado un algoritmo de optimización energética como pueden ser otros algoritmos, puesto que depende mucho de la configuración inicial de partida y del esquema de visita, sino que es considerado como otro tipo de estimador diferente de los clásicos. El algoritmo terminará cuando no encuentre más vecinos con menor energía u oscilará entre varios vecinos con la misma energía.

En el tratamiento de imágenes médicas se utilizan frecuentemente para tomar en cuenta el hecho que la mayoría de los píxeles pertenecen a la misma clase a la que pertenecen sus píxeles vecinos. En términos físicos, esto implica que bajo la asunción del CAM, cualquier estructura anatómica que consista de un solo píxel tiene una probabilidad muy baja de ocurrir [32].

Una dificultad asociada con los modelos CAM es la selección apropiada de los parámetros que controlan la fuerza de las interacciones espaciales [32]. Una selección muy alta puede resultar en segmentación excesivamente suave y una pérdida de los detalles estructurales. Adicionalmente, los métodos CAM usualmente requieren algoritmos computacionales intensivos. A pesar de estas desventajas, los CAM son ampliamente utilizados no sólo para modelar clases de segmentación, sino también para modelar propiedades de texturas y homogeneidades de las intensidades [32].

Las técnicas de preprocesamiento y segmentación que mejores resultados brindan se presentan en el capítulo 2 de la investigación.

1.3. Herramientas usadas durante el desarrollo

A continuación se describen las herramientas y tecnologías a utilizar durante el desarrollo del sistema, así como una explicación de las principales características de estas herramientas, para arribar a conclusiones apropiadas sobre cuáles son las idóneas para resolver el problema en cuestión. Se explican los principales elementos de las metodologías de software, para seleccionar la más adecuada, donde se ofrecen las ventajas que tendría utilizar la escogida. Se argumenta acerca del lenguaje de programación a utilizar y el IDE de desarrollo, donde se hace una breve explicación de cada una.

1.3.1. Metodología de Software

Todo desarrollo de software es riesgoso y difícil de controlar, prueba de ello es que existen numerosas propuestas metodológicas que inciden dentro del desarrollo del software, pero si no se lleva una metodología de por medio, lo que se obtiene es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. A continuación se muestra en la tabla 1.1 una comparación entre las metodologías ágiles y las tradicionales, para así llegar a la conclusión de cuál sería las más recomendable a utilizar en el sistema.

Tabla 1.1. Comparación entre las metodologías tradicionales y ágiles.

Metodologías Tradicionales	Metodologías Ágiles
Cambios: Cierta resistencia a los cambios.	Cambios: Especialmente preparado para cambios durante el proyecto.
Procesos: Proceso mucho más controlado, con numerosas políticas/normas.	Procesos: Proceso menos controlado, con pocos principios.
Artefactos: El cliente interactúa con el equipo de desarrollo mediante reuniones. Más artefactos.	Artefactos: El cliente es parte del equipo de desarrollo. Pocos artefactos.
Roles: Grupos grandes y posiblemente distribuidos.	Roles: Grupos pequeños (menos de 10) y trabajando en el mismo sitio.

Después de realizar un análisis de las metodologías de desarrollo y el estudio de sus principales características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software, el equipo de desarrollo es de solo dos personas y la dificultad para un equipo pequeño de adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo.

Programación Extrema (XP)

XP es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y

la realimentación o reutilización del código desarrollado [54]. Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, equipos pequeños y cuyo plazo de entrega es inminente. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo de desarrollo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto [55].

La metodología XP se basa en [54]:

- Pruebas Unitarias: pruebas realizadas a los principales procesos, para detectar errores futuros.
- Refabricación: reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué nos brinda XP?:

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

Fundamentación de la elección

Luego de un análisis de algunas de las metodologías ágiles se decide utilizar XP ya que propone una estructura de roles adaptada al proyecto, la realimentación continua entre el cliente y el equipo de desarrollo constituye un factor muy importante ya que conduce el desarrollo del trabajo y planifica el tiempo de entrega de cada iteración. El ambiente de desarrollo basado en un único local y un ordenador por parte del equipo de trabajo se ajustan perfectamente a las características de XP. El software será desarrollado por dos programadores lo cual coincide con la característica de XP de programación en pareja. Otras características son la comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. La metodología XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software. Promueve el trabajo en equipo, propiciando un buen clima de trabajo [56].

1.3.2. Entorno integrado de desarrollo

Matlab (abreviatura de MATrix LABoratory, laboratorio de matrices), es un software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está

disponible para las plataformas Unix, Windows y Apple Mac OS X. Es una poderosa herramienta para la resolución numérica de problemas. MATLAB ofrece un entorno interactivo sencillo mediante una ventana en la que podemos introducir órdenes en modo texto y en la que aparecen los resultados [57]. Los gráficos se muestran en ventanas independientes. Cada ventana dispone de una barra de menús que controla su funcionalidad. Lo que distingue a MATLAB de otros sistemas de cálculo es su facilidad para trabajar con vectores y matrices [57]. Las operaciones ordinarias, suma, producto, potencia, operan por defecto sobre matrices, sin más restricción que la compatibilidad de tamaños en cada caso.

El software cuenta con un amplio abanico de herramientas especializadas denominadas toolboxes que extienden significativamente la funcionalidad del programa principal. Estas herramientas cubren en la actualidad prácticamente todas las aéreas principales del mundo de la ingeniería y la simulación. La toolbox para el procesado digital de imágenes es una de sus herramientas más famosas y utilizadas, y está formada por un conjunto de funciones que aplican las capacidades de MATLAB para el desarrollo de aplicaciones y algoritmos en el campo del procesado y análisis digital de imágenes.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, el trabajo con imágenes, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes. Matlab dispone además de las herramientas: GUIDE (editor de interfaces de usuario GUI por sus siglas en inglés) y Simulink (plataforma de simulación multi-dominio). También se pueden ampliar las capacidades de Matlab con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets). Es un software muy usado en universidades y centros de investigación y desarrollo.

Características principales:

- Lenguaje de alto nivel para cálculo técnico.
- Entorno de desarrollo para la gestión de código, archivos y datos.
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica.
- Herramientas para crear interfaces gráficas.
- Funciones para integrar los algoritmos basados en Matlab con aplicaciones y lenguajes externos, tales como: C/C++, FORTRAN, Java, COM y Microsoft Excel.

Principales ventajas de Matlab

- Posee una excelente ayuda incorporada.
- Es multiplataforma.
- Alta precisión en el cálculo y procesamiento de imágenes.
- Colección útil de funciones predefinidas .
- Comunidad de usuarios muy extendida

- Potente herramienta matemática

La principal desventaja de Matlab es que es un software propietario pero a pesar de este inconveniente se decidió el uso de esta herramienta principalmente por las ventajas que ofrece, ya que es una herramienta multiplataforma, con alta precisión en el cálculo y en el procesamiento de imágenes, y tiene incorporado una excelente ayuda.

1.3.3. Lenguaje de Programación

Como lenguaje de programación en el sistema se utilizó el lenguaje de programación de Matlab o también conocido como lenguaje M. Matlab permite a la hora de programar una serie de elementos típicos para la modificación del flujo de una secuencia de instrucciones [58]. La sintaxis es muy parecida a la de cualquier lenguaje de programación. Todos estos operadores se pueden usar en la ventana de comandos, en línea, o en un fichero .m [58].

La programación se lleva a cabo mediante un lenguaje que es muy parecido a lenguajes de alto nivel como BASIC o C [58]. Esto permite que el usuario pueda agrupar sentencias que utiliza frecuentemente dentro de un programa que puede ser invocado posteriormente [58]. De este modo se ahorra tiempo y esfuerzo en sucesivas sesiones pues no es necesario escribir todas las sentencias de nuevo.

1.3.4. Herramienta utilizada para la realización de las pruebas unitarias

Las pruebas de algoritmos escritos en MATLAB se realizan mediante *frameworks externos*, sobre todo se emplean marcos del tipo *xUnit*. Con la versión R2008a se desarrolló un primer intento de marco de pruebas propio conocido como *MATLAB xUnit Test Framework*, pero resultó muy limitado y su uso fue escaso ya que no formaba parte del paquete de Matlab. Con la versión R2013a, se desarrolló y distribuyó como parte integrante de MATLAB un nuevo framework, *MATLAB UnitTesting*, su código es abierto y puede encontrarse en el paquete *matlab.unittest*.

MATLAB UnitTesting permite a los usuarios escribir pruebas unitarias de forma automatizadas, administrarlas y verificar los resultados esperados de las pruebas contra los resultados reales. El framework permite escribir pruebas unitarias en tres formas distintas *scripts*, *funciones* y *clases*, donde cada tipo de prueba permite niveles de complejidad distintos.

Las pruebas en forma de *scripts* se emplean para comprobar que los resultados producidos por *scripts*, *funciones* y *clases* son los esperados. Sólo dos funciones están permitidas en estas pruebas *assert* para comprobar el valor, tipo, tamaño de los resultados, etc y *runttests*, para ejecutar las pruebas. Las pruebas en forma de *funciones* también se emplean para comprobar que los resultados producidos por *scripts*, *funciones* y *clases* son los esperados. Las pruebas en forma de *funciones* se basan en una función principal acompañada de tantas funciones locales como sea necesario. Por claridad, cada función suele asociarse con un caso de prueba. Las pruebas en forma de *clases* permiten test paramétricos, etiquetar las pruebas o emplear bloques

de código compartido por varias pruebas. Su uso, estructura, comprensión y manejo es mucho más complejo.

1.3.5. 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. Ofrece un conjunto completo de herramientas, brindando 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.

1.3.6. Herramienta para la creación del documento

LaTeX es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Está formado por un gran conjunto de marcos de TeX, con la intención de facilitar el uso del lenguaje de composición. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad de los documentos realizados con LaTeX es comparable a la de una editorial científica de primera línea. Entre las ventajas, además de la calidad ya mencionada, está el hecho de poder trabajar con estructuras matemáticas, gestionar referencias de manera adecuada, y todo en cuanto a formato: encabezados, pies de páginas, estructuras de listas, índices de figuras, tablas y contenido generados automáticamente, numeración de ecuaciones, teoremas, etc.

Características

- Es una herramienta de composición de textos.
- LaTeX, no es un procesador de texto.
- No es necesario preocuparse por el aspecto del documento, sino concentrarse en conseguir el contenido correcto.

¿Quiénes lo utilizan?

- La comunidad científica (matemática, física, computación científica).
- La industria editorial (principalmente para libros científicos o técnicos).
- Los proyectos de documentación de software libre.

Ventajas

- Facilita que el autor se concentre en el mensaje del documento en lugar de preocuparse de su apariencia.

- Permite crear estructuras complejas como notas a pie de página, bibliografía, índices, tablas, sin producir gran esfuerzo.
- Es libre y multiplataforma.

Conclusiones parciales

En el presente capítulo se realizó un estudio sobre la identificación de opacidad en pacientes operados de catarata, valorándose la necesidad de automatizar este proceso para aumentar la exactitud y reducir la variabilidad del observador. Se describen las principales etapas y conceptos del procesamiento digital de imágenes, destacando la importancia del preprocesamiento de imágenes provenientes de la lámpara de hendidura, haciendo énfasis en técnicas como la ecualización del histograma para la mejora del contraste, así como los filtros en el dominio del espacio y frecuencia. Además, se pudo profundizar en los principales métodos de segmentación de imágenes llegando a la conclusión que no existe un método de segmentación que alcance resultados aceptables para todo tipo de imagen médica, por lo que la selección del método apropiado puede ser muy difícil y tediosa. Durante el desarrollo del capítulo se seleccionaron y describieron las herramientas y tecnologías a ser usadas en la propuesta de solución. Se llegó a la conclusión de que XP sería la metodología apropiada para guiar el desarrollo del sistema, debido a sus características.

En el presente capítulo se realiza una comparación entre distintos métodos de preprocesamiento y segmentación para arribar a conclusiones de cual sería la propuesta de solución adecuada para identificar la OCP en imágenes oblicuas provenientes de la lámpara de hendidura, se describe la fase inicial de la metodología XP: Planificación, obteniendo los artefactos importantes de esta fases, tales como, las Historias de Usuarios, Plan de Iteraciones, Plan de Duración de Iteraciones y Plan de Entregas. También durante el desarrollo del capítulo se describe la fase de Diseño basándose fundamentalmente en el desarrollo de las tarjetas Clase-Responsabilidad-Colaboracion (CRC).

2.1. Técnicas de procesamiento utilizadas

A continuación se muestran las técnicas utilizadas durante la investigación y una de la 15 imágenes usadas para realizar el análisis, realizando una comparación de las mismas en cada una de las etapas a las que pertenecen, ya sea preprocesamiento o segmentación, para arribar a conclusiones de cual sería la propuesta de solución adecuada para dar cumplimiento al objetivo de la investigación.

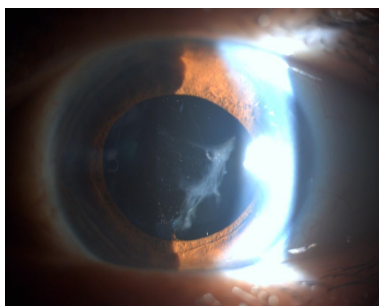


Figura 2.1. Imagen original.

2.1.1. Comparación entre las técnicas de preprocesamiento

A continuación se muestran las técnicas utilizadas durante la etapa de preprocesamiento para arribar a conclusiones de cuales brindarían mejores resultados para posteriormente realizar la segmentación.

En el caso de la mejora de contraste se obtuvieron los siguientes resultados:

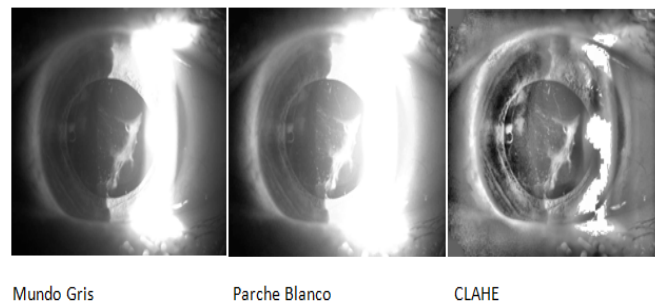


Figura 2.2. Resultados de aplicar técnicas para mejorar el contraste en la imagen.

Luego de probar estas distintas técnicas para mejorar el contraste de las imágenes se selecciona la ecualización adaptativa del histograma (CLAHE) como la más adecuada, debido a que logra una distribución uniforme de los colores de las imágenes, a diferencia de la técnicas de Mundo gris y Parche blanco que no aclaran toda la imagen, y en ocasiones aclaran demasiado algunas de las zonas de interés y la luz proveniente de la lámpara de hendidura.

Resultados obtenidos al aplicar filtros para disminuir el ruido en las imágenes.

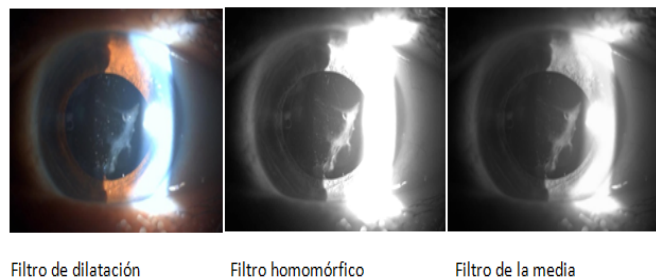


Figura 2.3. Resultados de aplicar un conjunto filtros a la imagen.

Luego de probar un conjunto de filtros para disminuir el ruido introducido durante la etapa de adquisición de las imágenes, se llega a la conclusión de que el filtro de la media es el que mejores resultados brinda, ya

que permite atenuar el ruido, a diferencia del homomórfico que lo que hace es trabajar con los elementos de luminancia y reflectancia, obteniendo como resultado una imagen muy clara y aumentando la intensidad de la luz proveniente de la lámpara de hendidura, en el caso del filtro de dilatación lo que permite es agrandar las zonas con opacidad pero no disminuye el ruido.

2.1.2. Comparación entre los métodos de segmentación

A continuación se muestran los métodos de segmentación aplicados a la imagen.

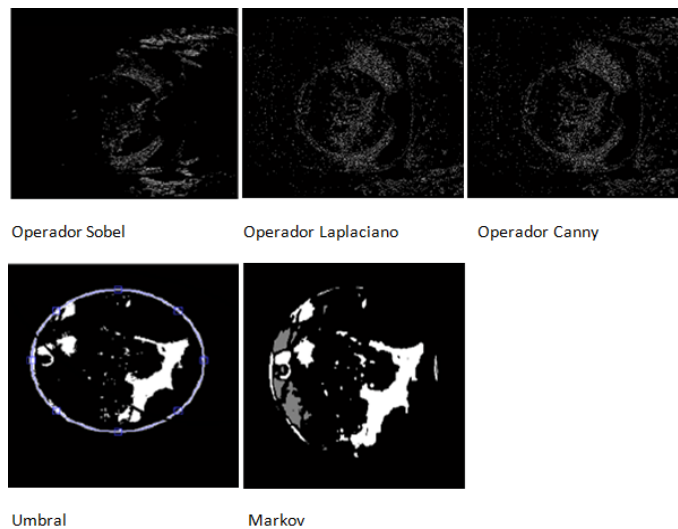


Figura 2.4. Diferentes métodos de segmentación aplicados a la imagen.

Luego de probar estas distintas técnicas de segmentación se llega a la conclusión de que los métodos que mejores resultados brindan para obtener las regiones con opacidad son el de Markov y el de Umbralización. Debido a esto se realiza un análisis a los resultados obtenidos por estos dos métodos, donde el especialista decidió que el método que mejor segmenta la opacidad es el de Markov, ya que las regiones se muestran con mayor homogeneidad, a diferencia del método de Umbralización que las muestra más fragmentadas. Después de realizar este análisis se lanza la propuesta de solución.

2.2. Selección del método propuesto

Para el diseño de la propuesta de solución se seleccionaron los métodos de procesamiento de imágenes que mejores resultados brindaron durante las comparaciones realizadas anteriormente a los métodos de pre-procesamiento y segmentación, con el objetivo de obtener una solución adecuada que permita identificar la

OCP en pacientes operados de catarata. Teniendo en cuenta las características esenciales de las imágenes y haciendo uso del entorno de computación Matlab 8.1, se seleccionaron una secuencia de pasos modelados en un diagrama de actividades, en aras de alcanzar una alta precisión en la segmentación de las imágenes.

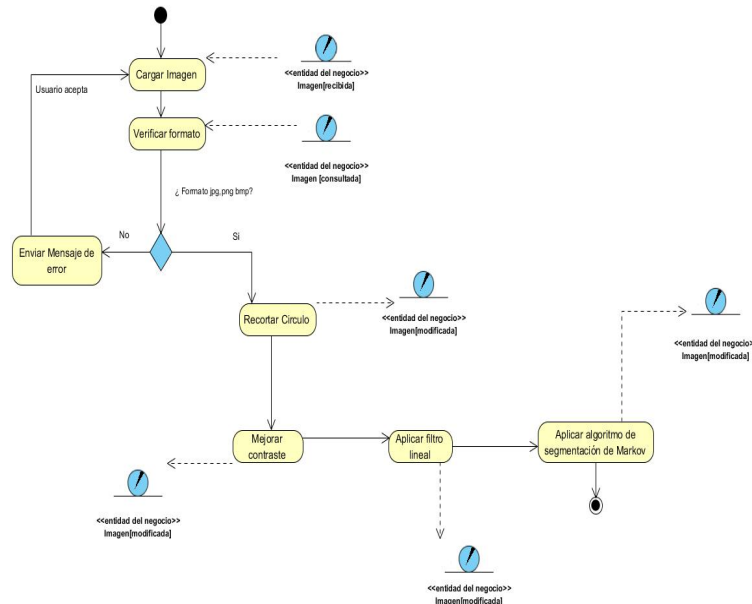


Figura 2.5. Diagrama de actividades que modela la solución.

1-Explorar los archivos y cargar la imagen médica

En el primer paso se procede a explorar los archivos y cargar la imagen oblicua proveniente de la lámpara de hendidura en los formatos jpg, png o bmp para su posterior análisis y procesamiento.

2-Recortar área circular

Para seleccionar el área circular en la imagen se definió una región de interés (ROI), la cual es una porción de una imagen que se quiere filtrar o se desea realizar alguna otra operación con ella. Se puede definir una región ROI creando una máscara binaria, la cual tenga el mismo tamaño que la imagen que se quiere procesar con los píxeles, definidos por la región ROI, a 1 lógico y los demás píxeles hasta completar el tamaño de la imagen definidos a 0. Para realizar este proceso se escoge la forma del objeto de la región ROI. En este caso es circular por lo que se utiliza la instrucción **imellipse** y para crear la máscara se usa la función **createMask** que ofrece Matlab.

3-Mejorar contraste de la imagen

Para mejorar el contraste de la imagen cargada, se propone expandir el histograma o realizar una ecualización del mismo buscando una distribución uniforme con el objetivo de potenciar detalles y características de

la imagen. Para ello se utiliza un algoritmo predefinido en la caja de herramienta Procesamiento de Imágenes en Matlab 8.1 mediante la función **adaphisteq** que mejora el contraste de la imagen utilizando una ecualización adaptativa con contraste limitado. La función trabaja con imágenes en escala de grises y puede ser del tipo `uint8`, `uint16`, `int16`, `single` o `double` que son los tipos de datos que contendrá una imagen en Matlab.

4-Aplicar filtro lineal para la reducción de ruido

Para eliminar el ruido introducido en el proceso de captura de las imágenes se selecciona una técnica de filtrado lineal empleando un algoritmo de análisis y mejora. Para ello se aplica un filtro de media con el objetivo de atenuar el ruido existente y desenfocar la imagen. Se utiliza la función **fspecial** la cual crea un filtro bidimensional del tipo especificado en una de sus variables, en este caso se propone el filtro **average** que devuelve un filtro promedio de un tamaño especificado. El valor por defecto es [3 3] con el cual se va a estar trabajando en la propuesta de solución.

5-Aplicar algoritmo de segmentación

La segmentación es una importante tarea para un sistema de análisis de la imagen y de su calidad depende en gran medida el resultado final de la interpretación automática de la secuencia de imágenes médicas. Para el caso de las imágenes oblicuas se propone el método de segmentación basada en textura, proponiendo utilizar un método de segmentación supervisado utilizando campos aleatorios de Markov el cual utiliza *Iterative-ConditionMode* (ICM), un algoritmo determinístico para estimar los parámetros específicos de cada textura y mejorar la clasificación obtenida.

2.3. Fase de Planificación

La metodología XP define como fase inicial del desarrollo de software la planificación. Durante el transcurso de esta etapa se realiza el proceso de identificación y elaboración de las historias de usuario (HU), además el equipo de trabajo se familiariza con las tecnologías y herramientas seleccionadas para el desarrollo. El resultado de la presente fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración, de forma tal que sea una guía durante el desarrollo [59].

2.3.1. Historias de Usuarios

Las Historias de Usuarios (HU) sustituyen a los documentos de especificación funcional y a los casos de uso. Estas historias son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar [54]. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido [54]. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa

del tiempo que llevará su desarrollo. El tiempo de desarrollo ideal para una HU varía entre 1 y 3 semanas [54].

Según Kent Beck cada HU recoge al menos los siguientes aspectos:[60]

- **Número:** Posee el número asignado a la HU.
- **Nombre de HU:** Atributo que contiene el nombre de la HU.
- **Usuario:** El usuario del sistema que utiliza o protagoniza la HU.
- **Prioridad en el negocio:** Evidencia el nivel de prioridad de la HU en el negocio.
- **Riesgo de desarrollo:** Evidencia el nivel de riesgo en caso de no realizarse la HU.
- **Puntos estimados:** Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal como 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias, por lo que cuando el valor de dicho atributo es 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas.
- **Puntos reales:** Igual que el parámetro anterior, pero en este caso será el tiempo real en el que se realizó la HU.
- **Descripción:** Posee una breve descripción de lo que realizará la HU

A las HU que describen las características del sistema se le decidió agregar los campos Observación, que brindará información extra para que la HU sea más comprensible y el campo Programador Responsable. Como resultado del trabajo realizado durante la fase de planificación se identificaron un total de 14 HU. A continuación se muestran algunas de ellas.

Tabla 2.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Mejorar contraste de la imagen.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Las imágenes resultantes de la lámpara de hendidura presentan diferentes sombras debido a que hay partes de la misma que no siempre se pueden iluminar con la luz proveniente del artefacto, por lo que se hace necesario obtener una distribución uniforme de los diferentes contrastes de la imagen para poder identificar con mayor claridad las regiones con opacidad.	
Observaciones:	

Tabla 2.2. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Aplicar método de segmentación por textura.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2,5	Iteración asignada: 2,5
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: A partir de la imágenes capturadas por el artefacto es necesario obtener la textura que representa la OCP con diferente color en relación a las demás regiones presentes en la imagen.	
Observaciones:	

2.3.2. Planificación de la Entrega

El cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días [56].

Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación del esfuerzo para cada una de las HU, arrojando los siguientes resultados:

Tabla 2.3. Estimación de esfuerzo por Historia de Usuario

Historias de Usuario		Puntos estimados(semanas)
1	Cargar Imagen	0.5
2	Aplicar Filtros Lineales	2.5
3	Aplicar Filtros No Lineales	2.5
4	Aplicar Filtros Morfológicos	2.0
5	Mejorar contraste de la imagen	2.0
6	Seleccionar región de interés	2.5
7	Aplicar método de segmentación por textura	2.5
8	Obtener los bordes de la imagen	2.0
9	Binarizar la imagen	2.0
10	Manipular la imagen	0.5
11	Obtener imagen original	1.0
12	Minimizar y maximizar la imagen	0.5
13	Mostrar características de la imagen	1.0
14	Guardar cambios realizados en la imagen	1.0

2.3.3. Plan de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Así mismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

En esta fase se definieron tres iteraciones para la realización del sistema:

Iteración 1

En esta iteración se realizan las HU que obtuvieron los primeros resultados en el sistema, relacionadas con los filtros para disminuir el ruido y mejora del contraste en las imágenes. Se realizan las HU Cargar Imagen, Aplicar Filtros Lineales, Aplicar Filtros No Lineales, Aplicar Filtros Morfológicos y Mejorar contraste de la imagen.

Iteración 2

En esta iteración se realiza la HU relacionada con el proceso para obtener la región de interés seleccionada por el usuario en la imagen y se realizan las HU relacionadas con los procesos para identificar las zonas con opacidad, tales como: Seleccionar región de interés, Aplicar método de segmentación por textura, Obtener los bordes de la imagen y Binarizar la imagen.

Iteración 3

En esta iteración se realizan las HU que facilitan el trabajo de los usuarios en el sistema , tales como: Manipular la imagen, Obtener imagen original, Minimizar y Maximizar la imagen, Mostrar características de la imagen y Guardar cambios realizados en la imagen.

Tabla 2.4. Plan de duración de las iteraciones

Iteración	Historias de Usuario		Duración (semanas)
1	1	Cargar Imagen	10.1
	2	Aplicar Filtros Lineales	
	3	Aplicar Filtros No Lineales	
	4	Aplicar Filtros Morfológicos	
	5	Mejorar contraste de la imagen	
2	1	Seleccionar región de interés	9.4
	2	Aplicar método de segmentación por textura	
	3	Obtener los bordes de la imagen	
	4	Binarizar la imagen	
3	1	Manipular la imagen	4.0
	2	Obtener imagen original	
	3	Minimizar y Maximizar la imagen	
	4	Mostrar características de la imagen	
	5	Guardar cambios realizados en la imagen	
Total			23.5

2.3.4. Plan de entrega

El plan de entregas tiene como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una. Luego de realizar el plan, se obtuvo como resultado tres entregables.

Tabla 2.5. Plan de entrega de versiones

Entregable	Iteración 1	Iteración 2	Iteración 3
Versiones	Versión 0.1	Versión 0.2	Versión 0.3
Fecha	10/2/2016	3/4/2016	9/5/2016

2.4. Diseño

En el presente acápite se hace referencia a los elementos que conforman el diseño del sistema a construir, que acorde a lo planteado por la metodología XP debe ser un diseño simple, sencillo y de fácil interacción, basado principalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC).

2.4.1. Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) son utilizadas para representar las responsabilidades de las clases y sus interacciones. Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño [60]. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente. Las tarjetas determinan el comportamiento de cada actividad.

Como resultado del trabajo realizado durante esta fase se generan como artefactos 7 tarjetas CRC. A continuación se muestran algunas de ellas.

Tabla 2.6. Tarjeta CRC # 1

Tarjeta CRC	
Clase: ProcesamientoDeImágenes	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Controla los procesos que se realizan en el sistema, mostrando los resultados visuales de aplicar cada uno de ellos. • Permite cargar la imagen con la cual se va a trabajar. • Permite cortar círculos de forma manual. • Permite realizar diferentes procesos que contribuyen a facilitar el trabajo del usuario en el sistema, tales como, Manipular imagen, Obtener imagen original y Minimizar y maximizar la imagen. 	<ul style="list-style-type: none"> • PreprocesamientoDeImágenes • SegmentacionDeImágenes

Tabla 2.7. Tarjeta CRC # 2

Tarjeta CRC	
Clase: PreprocesamientoDeImágenes	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla 2.7. Continuación de la página anterior

<ul style="list-style-type: none"> • Permite disminuir el ruido de la imagen, aplicando diferentes filtros para obtener distintos resultados. • Permite mejorar el contraste de la imagen. 	<ul style="list-style-type: none"> • Filtro • Contraste
--	---

Tabla 2.8. Tarjeta CRC # 3

Tarjeta CRC	
Clase: SegmentacionDeImagenes	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Permite obtener los bordes de la imagen. • Permite segmentar la imagen aplicando métodos de segmentación por regiones. 	<ul style="list-style-type: none"> • PreprocesamientoDeImagenes • Borde • Regiones

Conclusiones Parciales

Durante el desarrollo del presente capítulo se pudo llegar a la conclusión de que las técnicas que más resultados brindan son, en el caso del preprocesamiento, la ecualización adaptativa del histograma y el filtro de la media y para el caso de la segmentación, el algoritmo de segmentación por textura de Markov. Se utiliza la metodología XP para realizar el análisis y diseño de la aplicación, generando como artefactos un total de 14 HU que describen los aspectos principales a tener en cuenta para el desarrollo de la solución. El cliente estableció la prioridad de cada una de las HU que describen las funcionales del sistema, puntualizando el orden de su implementación y las iteraciones en que serán implementadas. También se logra traducir los requerimientos a funcionalidades a implementar, usando las tarjetas CRC que propone la fase de Diseño de la metodología.

Implementación y Pruebas

Durante el desarrollo del presente capítulo se estará transitando por las fases de implementación y prueba que propone la metodología XP, generando como artefacto en el caso de la fase de implementación las tareas de desarrollo que dan solución a cada una de las historias de usuarios identificadas en la fase de planificación. También se realizan un conjunto de pruebas al sistema, divididas en pruebas unitaria y de aceptación como propone la metodología y finalmente se realiza la validación a los resultados obtenidos utilizando la técnica de juicio de experto.

3.1. Fase de implementación

Se especifica en esta fase la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Lo primero es hacer un chequeo de cada HU, en conjunto con el plan de iteraciones y se modifica en caso de ser necesario, para esto se crean tareas de desarrollo para de esta forma poder organizar la implementación. Estas tareas, al contrario de las HU escritas en el lenguaje del cliente, son escritas en un lenguaje técnico. Como parte de la planificación realizada en el capítulo anterior se detallan a continuación las iteraciones de desarrollo sobre el sistema.

3.1.1. Iteración 1

En esta iteración se implementan las funcionalidades que obtuvieron los primeros resultados en el sistema, relacionadas con los filtros para disminuir el ruido y mejorar el contraste en las imágenes:

3.1. FASE DE IMPLEMENTACIÓN

Tabla 3.1. Puntos de estimación por historias de usuario de la iteración 1.

NO	Historias de Usuario	Puntos estimados(semanas)
1	Cargar Imagen	0,5
2	Aplicar Filtros Lineales	2,5
3	Aplicar Filtros No Lineales	2,5
4	Aplicar Filtros Morfológicos	2,0
5	Mejorar contraste de la imagen	2,0

A continuación se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 3.2. Tarea de desarrollo # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Desarrollo de la funcionalidad para cargar imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 0,5
Fecha de inicio: 15 de diciembre de 2015	Fecha de fin: 17 de diciembre de 2015
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para cargar la imagen se utiliza la función uigetfile la cual permite cargar la imagen en formato jpg,png y bmp.	

Tabla 3.3. Tarea de desarrollo # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 2
Nombre de la tarea: Desarrollo de la funcionalidad para aplicar filtros lineales	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,5
Fecha de inicio: 18 de diciembre de 2015	Fecha de fin: 1 de enero de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: La funcionalidad de los filtros lineales se puede efectuar de 2 maneras. El usuario selecciona el filtro que estime conveniente probar. Para implementar el filtro lineal gaussiano se utiliza la función fspecial con el parámetro gaussian, mientras que para implementar el filtro de la media se utiliza también la función fspecial pero con el parámetro average .	

Tabla 3.4. Tarea de desarrollo # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 3
Nombre de la tarea: Desarrollo de la funcionalidad para aplicar filtros no lineales	

Continúa en la próxima página

Tabla 3.4. Continuación de la página anterior

Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,5
Fecha de inicio: 2 de enero de 2016	Fecha de fin: 17 de enero de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: La aplicación de los filtros no lineales se puede efectuar de 2 maneras. El usuario selecciona el filtro que estime conveniente probar. Para aplicar el método de la mediana es necesario utilizar la función medfilt2 , la cual permite disminuir el ruido en las imágenes y no elimina los bordes, mientras que para el filtro no lineal homomórfico se utiliza la función homomorph , la cual permite separar en las imágenes los componentes de iluminación y reflectancia.	

Tabla 3.5. Tarea de desarrollo # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 4
Nombre de la tarea: Desarrollo de la funcionalidad para aplicar filtros morfológicos	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,0
Fecha de inicio: 18 de enero de 2016	Fecha de fin: 29 de enero de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para aplicar los métodos morfológicos es necesario utilizar un conjunto de funciones para cada uno de los métodos con los que trabaja, en el caso de la dilatación se utiliza la función imdilate , la cual permite dada una imagen RGB obtener una señal de mayor valor, es decir una imagen más clara, en el caso de la erosión se utiliza imerode , la cual trabaja también con imágenes RGB, pero obtiene una señal de menor valor, es decir una imagen más oscura, en el caso de la apertura se trabaja con la función imopen , donde la misma permite eliminar los objetos blancos y para el cierre imclose , la cual permite eliminar los objetos oscuros.	

Tabla 3.6. Tarea de desarrollo # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 5
Nombre de la tarea: Desarrollo de la funcionalidad para mejorar contraste de la imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,0
Fecha de inicio: 30 de enero de 2016	Fecha de fin: 10 de febrero de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para mejorar el contraste de las imágenes se utiliza el método de equalización adaptativa del histogramas, donde es necesario utilizar la función adaphisteq que mejora el contraste de la imagen utilizando una equalización adaptativa con contraste limitado. La función trabaja con imágenes en escala de grises y puede ser del tipo uint8, uint16, int16, single o double que son los tipos de datos que contendrá una imagen en Matlab.	

3.1.2. Iteración 2

En esta iteración se implementan las funcionalidades relacionadas con los procesos utilizados para obtener las regiones de interés en el sistema:

Tabla 3.7. Puntos de estimación por historias de usuarios de la iteración 2.

NO	Historias de Usuario	Puntos estimados(semanas)
1	Selecciona región de interés	2,5
2	Aplicar método de segmentación por textura	2,5
3	Obtener los bordes de la imagen	2,0
4	Binarizar la imagen	2,0

A continuación se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 3.8. Tarea de desarrollo # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 6
Nombre de la tarea: Desarrollo de la funcionalidad para seleccionar la región de interés	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,5
Fecha de inicio: 10 de febrero de 2016	Fecha de fin: 27 de febrero de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Se define una región de interés (ROI), creando una máscara binaria, la cual tenga el mismo tamaño que la imagen que se quiere procesar con los píxeles definidos por la región ROI. Para realizar este proceso se escoge la forma del objeto de la región ROI que se desea, en este caso es un círculo por lo que se utiliza la instrucción imellipse .	

Tabla 3.9. Tarea de desarrollo # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 7
Nombre de la tarea: Desarrollo de la funcionalidad para segmentar por textura	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,5
Fecha de inicio: 28 de febrero de 2016	Fecha de fin: 15 de marzo de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para realizar la segmentación por textura se usa el método de Markov y dentro del mismo se utiliza la función <i>IterativeConditionMode</i> (ICM), un algoritmo determinístico el cual permite estimar los parámetros específico de cada textura y mejorar la clasificación obtenida.	

Tabla 3.10. Tarea de desarrollo # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 8
Nombre de la tarea: Desarrollo de la funcionalidad para obtener los bordes en la imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 1,0
Fecha de inicio: 16 de marzo de 2016	Fecha de fin: 29 de marzo de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: La funcionalidad de segmentación por borde se puede efectuar de 3 maneras. El usuario selecciona el método que estime conveniente probar. Para implementar la segmentación se utiliza la función edge , donde por parámetro se le pasa el tipo de algoritmo que se considere utilizar ya sea Canny, Sobel o LoG.	

Tabla 3.11. Tarea de desarrollo # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 9
Nombre de la tarea: Desarrollo de la funcionalidad para binarizar la imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 2,0
Fecha de inicio: 30 de marzo de 2016	Fecha de fin: 12 de abril de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para binarizar la imagen es necesaria aplicar la tecnica del umbral, la cual permite realizar una partición binaria de las intensidades de la imagen, determinando un valor de intensidad denominado umbral que separa las clases deseadas. Para implementar la técnica es necesario utilizar el algoritmo Otsu para calcular automáticamente el valor del umbral, y posteriormente se utiliza la función im2bw para obtener por un lado los valores menores que dicho umbral y por el otro los mayores.	

3.1.3. Iteración 3

En esta iteración se implementan las funcionalidades que permiten mejorar el trabajo de los usuarios en el sistema.

Tabla 3.12. Puntos de estimación por historias de usuarios de la iteración 3.

NO	Historias de Usuario	Puntos estimados(semanas)
1	Manipular imagen	0,5
2	Obtener imagen original	1,0
3	Minimizar y maximizar la imagen	0,5
4	Mostrar características de la imagen	1,0
5	Guardar cambios realizados en la imagen	1,0

A continuación se muestran las tareas de desarrollo realizadas para las funcionalidades de esta iteración:

Tabla 3.13. Tarea de desarrollo # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 10
Nombre de la tarea: Desarrollo de la funcionalidad para obtener imagen original	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 1,0
Fecha de inicio: 16 de abril de 2016	Fecha de fin: 14 de abril de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para implementar el algoritmo es necesario utilizar la función imshow la cual muestra la imagen original cargada al inicio.	

Tabla 3.14. Tarea de desarrollo # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 13
Nombre de la tarea: Desarrollo de la funcionalidad para mostrar las características de la imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 1,0
Fecha de inicio: 26 de abril de 2016	Fecha de fin: 2 de mayo de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: El sistema muestra un conjunto de características de la imagen, utilizando las funciones imshow y whos para obtener las distintas características de la misma.	

Tabla 3.15. Tarea de desarrollo # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 14
Nombre de la tarea: Desarrollo de la funcionalidad para guardar cambios en la imagen	
Tipo de tarea: configuración - desarrollo	Puntos estimados: 1,0
Fecha de inicio: 3 de mayo de 2016	Fecha de fin: 9 de mayo de 2016
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez	
Descripción: Para aplicar el método es necesario utilizar la función imwrite , la cual permite guardar los cambios realizados en el sistema.	

3.2. Interfaz del sistema

A continuación se muestra la interfaz que permite visualizar los resultados de aplicar el algoritmo. La misma cuenta con otro conjunto de funcionalidades para que los clientes puedan aplicar otros métodos de acuerdo a las características de las imágenes con las cuales se haga el estudio.



Figura 3.1. Interfaz del sistema.

- 1: Permite cargar la imagen en formato jpg, png o bmp.
- 2: Permite aplicar los filtros lineales gaussiano y la media.
- 3: Permite aplicar los filtros no lineales homomórfico y la mediana.
- 4: Permite aplicar los filtros morfológicos de dilatación, erosión, apertura y cierre.
- 5: Permite mejorar el contraste aplicando ecualización adaptativa del histograma, mundo gris o parche blanco.

- 6: Permite al usuario seleccionar una porción circular de la imagen.
- 7: Permite segmentar por textura usando Markov.
- 8: Permite segmentar por borde, donde pueden ser seleccionados por los usuarios los operadores Sobel, Canny o LoG.
- 9: Permite segmentar por umbralización usando el algoritmo de Otsu para la selección del umbral.
- 10: Permite manipular la imagen.
- 11: Permite llevar la imagen a su fase original.
- 12: Permite minimizar y maximizar la imagen.
- 13: Muestra las características esenciales de la imagen.
- 14: Guarda los cambios realizados en la imagen.
- 15: Muestra el histograma de la imagen.
- 16: Muestra el análisis espectral de la imagen.

3.3. Resultados de aplicar el algoritmo

El algoritmo fue aplicado a 15 imágenes oblicuas provenientes de la lámpara de hendidura. A continuación se muestran los resultados obtenidos de aplicar este algoritmo a 3 de estas imágenes.

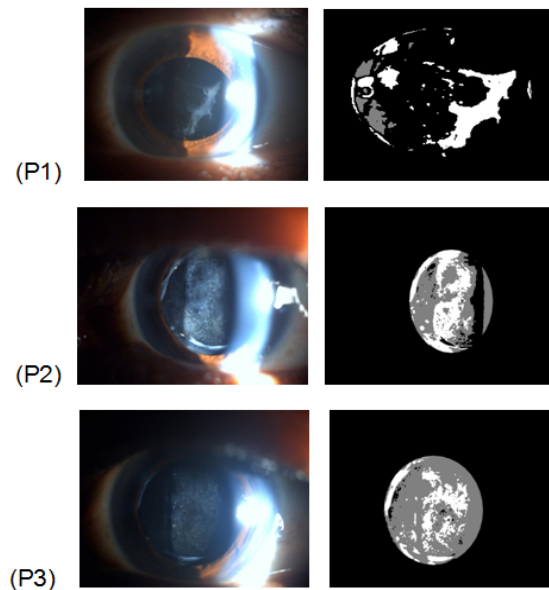


Figura 3.2. Resultados de aplicar el algoritmo.

3.4. Pruebas

Uno de los pilares de la Programación Extrema (XP) es el proceso de pruebas [60]. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos [60]: pruebas unitarias las cuales se concentran en los componentes individuales asegurándose que funcionen de manera apropiada como unidad, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final.

3.4.1. Pruebas unitarias

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. Los programadores realizan estas pruebas cuando: la interfaz de un método no es clara, la implementación es complicada, para testear entradas y condiciones inusuales luego de modificar algo. Para poder integrar el código realizado al ya existente, el mismo debe aprobar satisfactoriamente todos los casos de prueba definidos.

En XP los programadores deben escribir las pruebas unitarias para cada módulo antes de escribir el código. No es necesario escribir casos de prueba para todos los módulos, sólo para aquellos en que exista la posibilidad de que puedan fallar. Las pruebas se realizaron automatizadas utilizando el framework MATLAB Unit Testing del cual se obtuvieron los siguientes resultados.

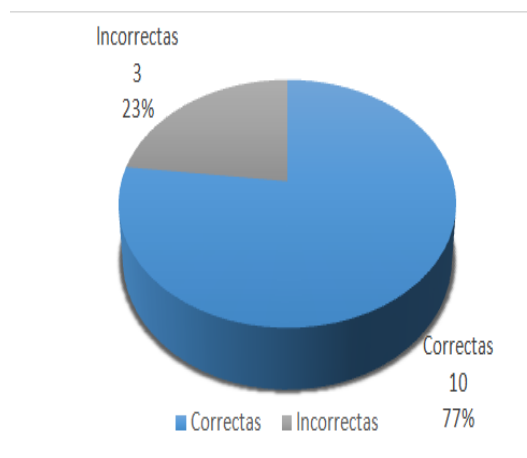


Figura 3.3. Resultados de aplicar el framework MATLAB Unit Testing.

Pruebas de caja blanca

El método prueba de caja blanca se basa en un examen cercano al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de pruebas que ejerciten conjuntos específicos de condiciones, bucles o ambos [61]. A continuación, se describirá la realización de la técnica de camino básico, 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.

A continuación se muestra la prueba realizada al código perteneciente al método `ecualizar_histograma`, el cual permite mejorar el contraste de la imagen. Es uno de los métodos a los cuales se le aplica la métrica de complejidad ciclomática debido a su relevancia para el sistema.

```
estado = get(handles.checkbox16, 'Value');
est_mundo=get(handles.checkbox22, 'Value');
est_patch=get(handles.checkbox23, 'Value');

S =getimage(handles.axes1);

if isrgb(S)
    S=rgb2gray(S);

if estado ==1

A = adapthisteq(S, 'clipLimit',0.8, 'Distribution', 'rayleigh');
axes(handles.axes1)
imshow(A);
handles.EH=A;
guidata(hObject,handles)
axes(handles.axes3)
imhist(handles.EH);

elseif estado ~=1 && est_patch==1
axes(handles.axes1)
imshow(handles.patch);

else
axes(handles.axes1)
imshow(handles.direccion);
axes(handles.axes3)
f=rgb2gray(imread(handles.direccion));
imhist(f);

end
```

Figura 3.4. Código del método `ecualizar_histograma`.

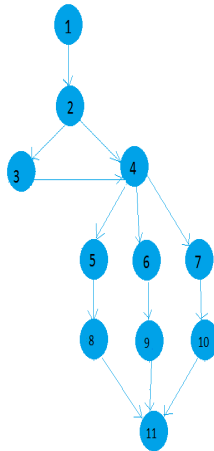


Figura 3.5. Representación del grafo de flujo de camino básico de `ecualizar_histograma`.

A partir del código fuente se diseñó el grafo como se muestra en la figura 3.5, luego se procede a calcular la complejidad ciclomática $V(G)$, la cual define el número de caminos independientes del conjunto básico de un programa, dando un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

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

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

$$V(G) = 1 + \text{nodospredicados}$$

$$V(G) = 1 + 3$$

$$V(G) = 4$$

$$V(G) = r$$

$$V(G) = 4$$

El valor $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del

programa, por lo que se definen los siguientes 4 caminos:

Camino básico 1: 1-2- 3- 4- 5- 6-8-11

Camino básico 2: 1-2-4-5-8-11

Camino básico 3: 1-2-4-6-9-11

Camino básico 4: 1-2-4-7-10-11

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

Tabla 3.16. Caso de prueba de caja blanca para el camino básico 2

Entrada	Se exploran los archivos y se carga la imagen.
Resultados Esperados	Se muestra la imagen logrando mejorar el contraste de la misma.
Condiciones	<ul style="list-style-type: none">• Se carga una imagen en el formato jpg, png o bmp.• La imagen debe estar en escala de grises.

3.4.2. Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las Historias de Usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son de gran importancia, dado que miden el grado de satisfacción del cliente con el producto desarrollado [61]. Por lo tanto, son los clientes los responsables de verificar que los datos de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, son ellos los encargados de indicar el orden de resolución de los fallos [61]. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. A continuación se presenta el diseño de caso de prueba de 3 de las HU a las que se le realizaron las pruebas.

Tabla 3.17. Caso de prueba de aceptación de la HU 1.

Caso de prueba de aceptación
Código: HU_P1 Historia de Usuario: 1
Nombre: Cargar Imagen.
Descripción: Comprueba la funcionalidad que permite explorar los archivos y verificar el formato de las mismas.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución:
<ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario carga la imagen en el formato seleccionado. • El sistema muestra la imagen en pantalla con su correspondiente histograma y análisis espectral.
Resultado Esperado: Se muestra la imagen cargada en el sistema con su correspondiente histograma.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 3.18. Caso de prueba de aceptación de la HU 5.

Caso de prueba de aceptación
Código: HU_P1 Historia de Usuario: 5
Nombre: Mejorar Contraste de la imagen.
Descripción: Comprueba la funcionalidad que mejora el contraste de la imagen.
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de la lámpara de hendidura.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none">• El usuario selecciona la imagen en formato jpg, bmp o png.• El usuario selecciona la opción de ecualización del histograma en Contraste.• El sistema muestra la imagen en escala de grises con un ajuste de contraste y su correspondiente histograma expandido.
Resultado Esperado: El sistema devuelve una imagen en RGB o en escala de grises, logrando mejorar el contraste y aclarando las partes oscuras de la misma.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 3.19. Caso de prueba de aceptación de la HU 6.

Caso de prueba de aceptación
Código: HU_P2 Historia de Usuario: 6
Nombre: Aplicar método de segmentación por textura.
Descripción: Comprueba la funcionalidad que permite segmentar la opacidad de la cápsula posterior en las imágenes oblicuas.
Condiciones de Ejecución: Se debe haber cargado una imagen resultante de la lámpara de hendidura.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de segmentación por Markov. • El sistema muestra la imagen segmentada en diferentes texturas, mostrando la opacidad con un color diferente en relación a las demás texturas de la imagen.
Resultado Esperado: El sistema muestra la imagen segmentada, obteniendo las regiones con opacidad con otra textura que la diferencia de las demás regiones existentes en las imágenes.
Evaluación de la Prueba: Prueba satisfactoria.

Resultados de las pruebas de aceptación

Por cada HU se realizó una prueba de aceptación. Fueron detectadas un total de 11 no conformidades en las 3 iteraciones realizadas, las cuales ya han sido solucionadas. A continuación se muestra en la figura 3.4 los resultados obtenidos.

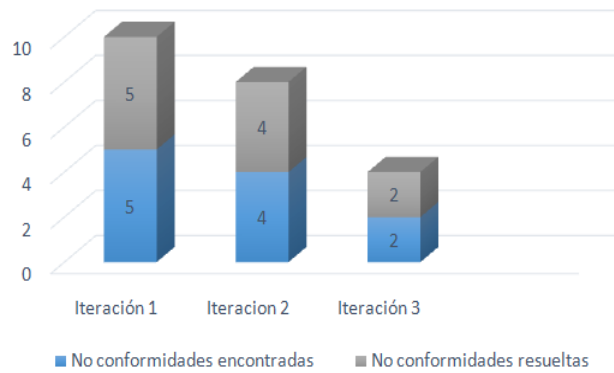


Figura 3.6. Resultado de las pruebas de aceptación.

Como parte de estas pruebas se procedió a la creación de 15 casos de estudio para verificar los resultados del software, en los mismos se tomaron imágenes de pacientes a los que llamaremos (Paciente 1, Paciente 2..., Paciente 15). Estas pruebas arrojaron los siguientes resultados:

1. Caso de estudio perteneciente a la imagen oblicua resultante de la lámpara de hendidura del Paciente 1

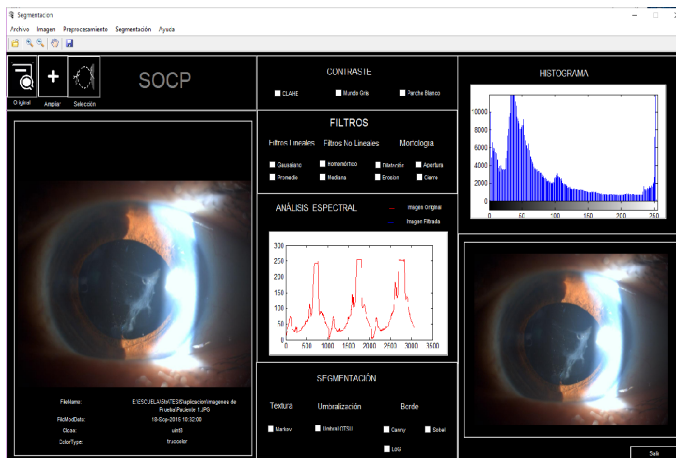


Figura 3.7. Imagen original cargada satisfactoriamente con su correspondiente histograma.

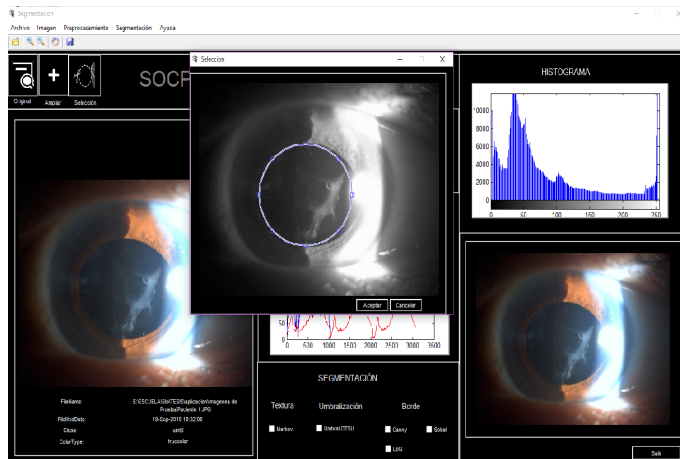


Figura 3.8. Selección manual de la región circular.

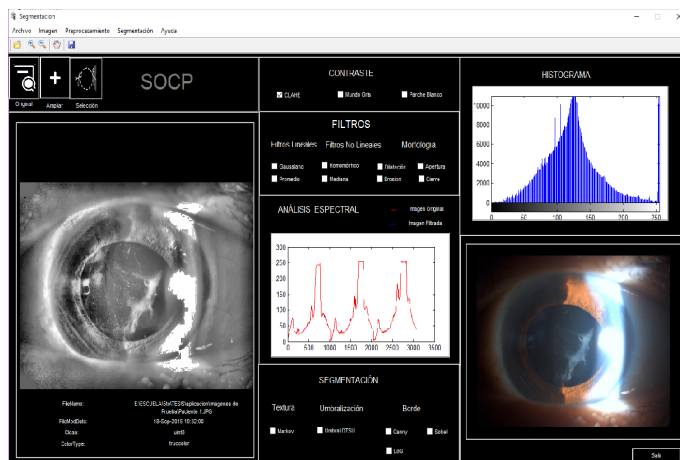


Figura 3.9. Imagen ecualizada con su correspondiente histograma expandido.

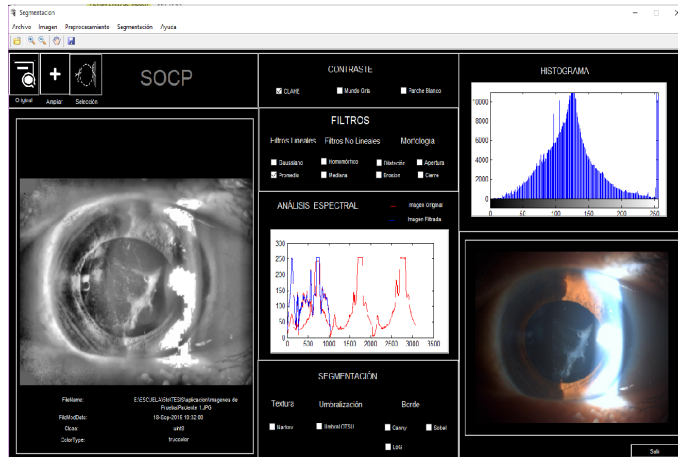


Figura 3.10. Aplicación del filtro Promedio para la reducción de ruido en la imagen.

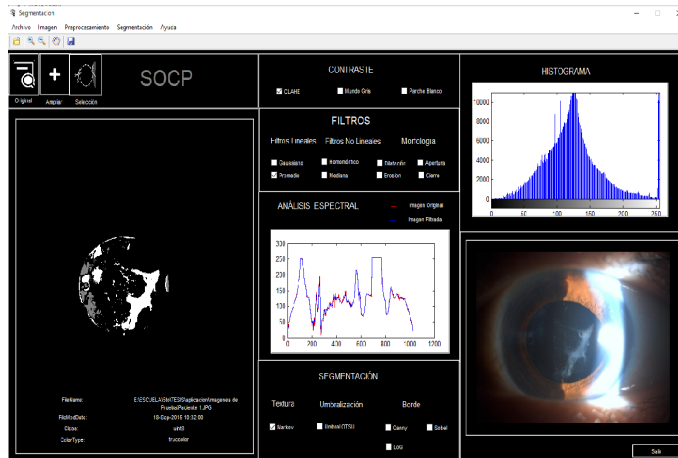


Figura 3.11. Resultado Final de la segmentación por textura usando Markov.

3.4.3. Validación del sistema

La validación de un sistema médico es algo complejo, de ahí la importancia de contar con buenos criterios a la hora de arribar a conclusiones sobre los resultados obtenidos. En este trabajo los especialistas afirman que no existen formas de experimentación bien establecidas, predominando los casos de estudios, que representan a los pacientes operados de cataratas que presentan la patología.

Como parte de las pruebas realizadas al sistema se hizo necesaria aplicar la técnica de juicio de experto, ya que es necesario la valoración de los especialistas para seleccionar las regiones con opacidad en las imágenes originales y verificar las que selecciona el sistema, dando una valoración de cada uno de los resultados obtenidos.

El juicio de expertos es un método de validación útil para verificar la fiabilidad de una investigación que se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos calificados, y que pueden dar información, evidencia, juicios y valoraciones [62].

Para la validación del algoritmo se construyó un interfaz utilizando la herramienta Matlab 8.1. Mediante esta se muestran los resultados que posteriormente se someten al criterio del panel de expertos, los cuales emiten la evaluación de los resultados obtenidos luego de aplicar el algoritmo.

Panel de expertos

Al frente del panel de experto se encuentra el especialista de II grado en Oftalmología Iván Hernández López, el cual es el cliente principal y parte del equipo de desarrollo. El mismo ha publicado más de seis artículos en relación a la OCP, de los cuales más de tres en revistas de primer nivel. Actualmente ejerce como jefe del laboratorio experimental del Instituto Cubano de Oftalmología Ramón Pando Ferrer y está realizando su tesis doctoral en relación a la cuantificación de la OCP. Este especialista fue el encargado de seleccionar los demás integrantes, donde el panel de expertos quedo conformado finalmente por la Dra Eneida de la Caridad Pérez Candelaria, especialista de II grado en Oftalmología, profesora auxiliar y miembro del comité editorial de la Revista Cubana de Oftalmología y la Dr.C Taimí Cárdenas Días, también especialista de II grado, todos pertenecen al Instituto Cubano de Oftalmología.

Criterios de los expertos

Para la evaluación del panel de experto se utilizaron 15 imágenes ofrecidas por los especialistas, las cuales tenían señaladas las partes con opacidad en diferentes tonalidades incluyendo el tipo de opacidad: leve, moderada o severa, donde todo lo que resalta en color verde es opacidad. El panel de experto realiza la evaluación de los resultados de la siguiente forma:

- 5 : La imagen resultante detecta toda la opacidad señalada.
- 4 : La imagen resultante detecta toda la opacidad señalada excepto algunas pequeñas zonas.
- 3 : La imagen resultante deja de detectar varias zonas con opacidad.
- 0 : La imagen resultante detecta zonas que no son opacidad.

A continuación se muestran los resultados de 2 de los casos de estudio analizados.

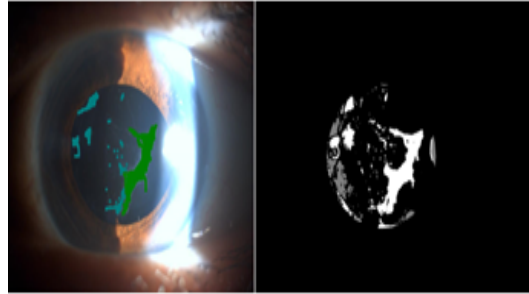


Figura 3.12. Caso de estudio 1

Evaluación: 5

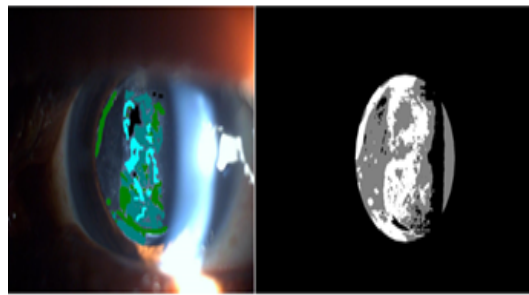


Figura 3.13. Caso de estudio 2

Evaluación: 4

Evaluación final de los expertos

Tabla 3.20. Resultados de la evaluación

Cantidad de imágenes	Evaluación
7	5
6	4
2	3
Promedio	4.3

Luego de un consenso realizado entre todos los expertos arribaron a la conclusión de que el sistema cumple con el objetivo propuesto, ya que de las imágenes analizadas solo 2 fueron evaluadas de 3 y ninguna de ellas fue evaluada de 0, logrando que la mayoría de los resultados oscilaran entre 4 y 5 con un promedio

de evaluación de 4.3, lo que permite que el algoritmo cumpla con los requerimientos especificados, disminuyendo el sesgo intraobservador en relación a la identificación visual de la opacidad.

Conclusiones Parciales

En el desarrollo del capítulo se abordaron los temas referentes a la implementación de la solución y la estrategia de pruebas a seguir. Se describieron las tareas de desarrollo para darle solución a las HU y se llevaron a cabo las pruebas realizadas a la solución, mostrando resultados satisfactorios, lo que demuestra que el desarrollo dirigido por pruebas desempeña un papel fundamental en el proceso de desarrollo de la aplicación con una metodología ágil, en este caso XP. También se realizó la validación del sistema la cual permitió demostrar que el algoritmo satisface las necesidades de los clientes y cumple con el objetivo de la investigación.

Con el desarrollo del presente trabajo de diploma se cumplieron todas las tareas propuestas. De esta forma se arribó a las siguientes conclusiones:

- Luego de consultar diferentes literaturas se aprecia que la OCP o catarata secundarias representa la principal causa de disminución en la visión de los pacientes operados de catarata, por lo que se hace necesario prevenir a tiempo dicha patología.
- El preprocesamiento permite mejorar la calidad de las imágenes oblicuas provenientes de la lámpara de hendidura, donde las técnicas que mejores resultados brindaron fueron la ecualización adaptativa del histograma y el filtro lineal de la media, aplicados en ese mismo orden, los cuales permitieron mejorar las imágenes.
- El proceso de segmentación constituye una herramienta efectiva para la identificación de la OCP en imágenes oblicuas provenientes de la lámpara de hendidura, donde durante el desarrollo de esta etapa las técnicas que mejores resultados brindaron fueron la de segmentación por umbral y la segmentación por textura. Destacándose la técnica de segmentación por textura de Markov como un método efectivo para identificar las zonas con OCP debido a que muestras las regiones con mayor homogeneidad.
- Se comprobó que el algoritmo realizado cumple con los requerimientos especificados, ya que permite la identificación de las distintas regiones con opacidad dentro de las imágenes oblicuas, ayudando en el diagnóstico de los pacientes operados de catarata.

Recomendaciones

- Se recomienda para próximas investigaciones que a la hora de la captura de las imágenes a utilizar en el sistema se haga un bosquejo general de todo el ojo, con tres secciones, iluminando la parte izquierda, el centro y la derecha y entonces componer la segmentación de las tres y mostrarlo como resultado final, para que desaparezcan las zonas de sombra y no se dejen de mostrar zonas con opacidad presentes en la imagen.
- Dividir las regiones de interés en tres texturas de acuerdo al tipo de opacidad: leve, moderada o densa.

Referencias bibliográficas

- 1 ACOSTA, R; HOFFMEISTER, L; ROMAN, R et al., Revisión sistemática de estudios poblacionales de prevalencia de catarata. *Archivos de la Sociedad Española de oftalmología*. 2006, vol. 81, n.º 9, págs. 509-516.
- 2 HERNÁNDEZ LÓPEZ, Iván; HERNÁNDEZ SILVA, Juan Raúl; CASTRO GONZÁLEZ, Yadira et al., Estrategias de prevención de la opacidad de la cápsula posterior. *Revista Cubana de Oftalmología*. 2010, vol. 23, págs. 608-623.
- 3 B., Dr. Alberto Arntz. CATARATA. *Departamento de Oftalmología*. 2011.
- 4 PONTIGO AGUILAR, Manuel Esteban; BARRAQUER COMPTE, Rafael Ignacio y QUESADA MARÍN, Pedro. *Prevención de la opacificación de la cápsula posterior mediante aspiración de las células epiteliales del cristalino*. 2001.
- 5 LORENTE, Ramón y ROJAS, Victoria de. Opacificación de la cápsula posterior. *Lorente R, Mendicute J. Cirugía de Cristalino. LXXXIV Ponencia Oficial de la Sociedad Española de Oftalmología*. 2008, vol. 2, págs. 1711-1741.
- 6 STEINBERG, Earl P; JAVITT, Jonathan C; SHARKEY, Phoebe D et al., The content and cost of cataract surgery. *Archives of ophthalmology*. 1993, vol. 111, n.º 8, págs. 1041-1049.
- 7 BERTELMANN, Eckart y KOJETINSKY, Corina. Posterior capsule opacification and anterior capsule opacification. *Current opinion in ophthalmology*. 2001, vol. 12, n.º 1, págs. 35-40.
- 8 CANCIO, Michel Alvarez. PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata. 2014.
- 9 ALVAREZ CANCIO, Michel; HERNÁNDEZ BARRIOS, Adrián; RODRÍGUEZ PUENTES, Rafael y HERNÁNDEZ LÓPEZ, Iván. PANDOC: software para la cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas scheimpflug del pentacam. 2013.
- 10 FINDL, Oliver; BUEHL, Wolf; BAUER, Peter y SYCHA, Thomas. Intervenciones para la prevención de la opacificación de la cápsula posterior. 2010.
- 11 BARMAN, Sarah A; HOLLICK, Emma J; BOYCE, James F et al., Quantification of posterior capsular opacification in digital images after cataract surgery. *Investigative ophthalmology & visual science*. 2000, vol. 41, n.º 12, págs. 3882-3892.

- 12 CARDENAS HIDALGO, PAUL; FLORES VARGAS, JOSE ALFREDO; LOPEZ ZAVALA, JAIME y MARTINEZ MORENO, PABLO. DISEÑO DE SISTEMA DE RECONOCIMIENTO DE PLACAS UTILIZANDO MATLAB. 2010.
- 13 YAIMARA MÁRQUEZ ÁLVAREZ, Licel Salazar Reyes. *Propuesta de un filtro para la eliminación de ruido causado por la iluminación en el sistema de visión de un robot móvil*. 2009.
- 14 GONZÁLEZ RODRÍGUEZ, Lázara Liset; DRC. HERNÁNDEZ MONTERO, Fidel Ernesto; GONZÁLEZ CORDÓN, Carlos Javier y MOHAMMED AMER, Mohammed Ahmad. *Procesamiento de imágenes de fondo de ojo para el diagnóstico médico*.
- 15 GONZALEZ, Rafael C y WOODS, Richard E. *Digital image processing*. 2002.
- 16 V. SRINIVASAN, R.D. Thulasiraj. *Instrumentos y Equipos Oftalmológicos*. 2003.
- 17 OLIVA, Karla Rodriguez. *Suavizado mediante técnicas de filtrado morfológico en imágenes de microscopía celular*. 2014.
- 18 CEPEDA NEGRETE, Jonathan y SANCHEZ YANEZ, Raul E. Algoritmos de Constancia de Color para el Mejoramiento de Imágenes Oscuras. *V Congreso Internacional de la Ciencia de Sistemas*. 2010.
- 19 OYARZO, Weslly Ogier Jara. *Visión Artificial: Análisis Teórico del Tratamiento Digital de Imágenes Para su aplicación en la identificación de objetos*. 2006.
- 20 KANUNGO, Tapas; HARALICK, Robert M y PHILLIPS, Ihsin. Global and local document degradation models. En. *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. 1993, págs. 730-734.
- 21 VALENZUELA, Javier Vidal e INFORMÁTICO, Ingeniero Civil. *Interpolación de Formas en Imágenes Usando Morfología Matemática*.
- 22 RAMOS, Frank N GIRALDO y BENAVIDES, Jaime F PANTOJA. RECONOCIMIENTO DE GESTOS MANUALES ALFABÉTICOS POR SEGMENTACION DE IMAGENES UTILIZANDO CAMPOS ALEATORIOS DE MARKOV.
- 23 SALCEDO ECHES, Edith Andrea y CHIRIBOGA MALDONADO, María del Pilar. *Implementación de algoritmos para preprocesamiento, detección de bordes y segmentación de imágenes digitales a color utilizando MATLAB*. 2003.
- 24 GONZÁLEZ MARTÍNEZ, Vanesa et al., Técnicas de umbralización para la detección de anomalías en la pared aórtica mediante OCT. 2013.
- 25 COTO, Ernesto. Métodos de segmentación de imágenes médicas. *Universidad Central de Venezuela: Lecturas en Ciencias de la Computación*. 2003, vol. 1, págs. 9-15.
- 26 VARGAS, Ernesto Gómez; NEIRA, Nelson Obregón y ARANGO, Diego Fernando Rocha. Métodos de segmentación de nubes en imágenes satelitales. *Tecnura*. 2013, vol. 17, n.º 36, págs. 96-110.

- 27 JAIME-ORTEGA, V; SIMANCAS-ACEVEDO, E y ROBLES-CAMARILLO, D. Aplicación para la Segmentación de Imágenes en el Reconocimiento Biométrico del Iris.
- 28 FREIRE ESPINOZA, Marco Fabricio. *Detección de bordes de imágenes digitalizadas*. 1998.
- 29 FRESNO, Mariana del; VÉNERE, Marcelo y CLAUSSE, Alejandro. Un enfoque integrado de segmentación para extracción de superficies en imágenes digitales 3D. En. *XIV Congreso Argentino de Ciencias de la Computación*. 2008.
- 30 CHEN, Ting y METAXAS, Dimitris. A hybrid framework for 3D medical image segmentation. *Medical Image Analysis*. 2005, vol. 9, n.º 6, págs. 547-565.
- 31 KASS, M; WITKIN, A y TETZOPOULOS, D. Active contour models» *International Journal of Computer Vision*. 1988.
- 32 GONZÁLEZ MARTÍNEZ, Vanesa et al., Técnicas de umbralización para la detección de anomalías en la pared aórtica mediante OCT. 2013.
- 33 HOOVER, Adam; JEAN-BAPTISTE, Gillian; GOLDFOF, Dmitry B y BOWYER, Kevin W. A methodology for evaluating range image segmentation techniques. En. *WACV*. 1994, págs. 264-271.
- 34 BESL, Paul J y JAIN, Ramesh C. Segmentation through variable-order surface fitting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 1988, vol. 10, n.º 2, págs. 167-192.
- 35 ASLAM, Tariq M; PATTON, Niall y DHILLON, Baljean. Assessment of systems of analyzing PCO. *Journal of Cataract & Refractive Surgery*. 2005, vol. 31, n.º 7, págs. 1270-1271.
- 36 MARTÍN, M Pilar; DÍAZ DELGADO, R; CHUVIECO, Emilio y VENTURA, Gemma. Burned land mapping using NOAA-AVHRR and TERRA-MODIS. En. *IV International conference on forest fire research*. 2002, págs. 45.
- 37 BASTARRIKA, Aitor y CHUVIECO, Emilio. Cartografía del área quemada mediante crecimiento de regiones: aplicación en entornos mediterráneos con imágenes TM y ETM+. *GeoFocus. Revista Internacional de Ciencia y Tecnología de la Información Geográfica*. 2006, n.º 6, págs. 182-204.
- 38 LIRA CHÁVEZ, Jorge. Introducción al tratamiento digital de imágenes. *Ciencia de la Computación, Instituto Politécnico Nacional, Universidad Nacional Autónoma de México*. 2002.
- 39 JAHANE, B. *Digital Image Processing*. 2000.
- 40 NICOT, Miriela Milagros Escobedo; GARCÍA, Silena Herold y ÁLVAREZ, Frank Calzadilla. Determinación De Zonas De Interés Con El Empleo De Texturas. *Ciencia en su PC*. 2010, n.º 1, págs. 13-23.
- 41 GRIGORESCU, Simona E; PETKOV, Nicolai y KRUIZINGA, Peter. Comparison of texture features based on Gabor filters. *Image Processing, IEEE Transactions on*. 2002, vol. 11, n.º 10, págs. 1160-1167.
- 42 ANDRÉS TRIBUJ, Mariano; DAVID WAISBAUM, Ariel; DRA. MEJAIL, Marta y DR. JACOBO BELLÉS, Julio. *Segmentación de imágenes texturadas*. 2006.

- 43 HARALICK, Robert M; SHANMUGAM, Karthikeyan y DINSTEIN, Its' Hak. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*. 1973, n.º 6, págs. 610-621.
- 44 LÓPEZ-ESPINOZA, Erika Danaé y ROBLES, Leopoldo Altamirano. Segmentación Markoviana Usando Modelos de Textura.
- 45 WINKLER, Gerhard. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*. 2012.
- 46 LI, Stan Z. *Markov random field modeling in computer vision*. 2012.
- 47 PUJOL, M; RIZO, R y ARQUES, P. Aplicación de los modelos de campos aleatorios de Markov en visión artificial. *Revista Electrónica de Visión por Computador*. N.º 4.
- 48 SOTO, A. Textures classification using markov random fields.
- 49 ORTIZ, A. Texturas y descripciones de regiones. 2008.
- 50 MOVELLAN, Javier R. Tutorial on Gabor filters. *Open Source Document*. 2002.
- 51 LIAPIS, Spyros; SIFAKIS, Eftychios y TZIRITAS, Georgios. Colour and texture segmentation using wavelet frame analysis, deterministic relaxation, and fast marching algorithms. *Journal of Visual Communication and Image Representation*. 2004, vol. 15, n.º 1, págs. 1-26.
- 52 GEMAN, Stuart y GEMAN, Donald. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*. *Journal of Applied Statistics*. 1993, vol. 20, n.º 5-6, págs. 25-62.
- 53 SRIRAM, R; FRANCOS, Joseph M y PEARLMAN, William A. Texture coding using a Wold decomposition model. *Image Processing, IEEE Transactions on*. 1996, vol. 5, n.º 9, págs. 1382-1386.
- 54 JOSKOWICZ, José. Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*. 2008, págs. 22.
- 55 SANCHEZ, M.A.M. Metodologías De Desarrollo De Software. 2004.
- 56 LETELIER, Patricio y Mª CARMEN PENADÉS, José H. Canó. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
- 57 JULIO BENÍTEZ LÓPEZ, José Luis Hueso Pagoaga. Introducción a MATLAB.
- 58 VILLANUEVA, M. Vargas; SORIA, Manuel Berenguel y CANTARERO, Teodoro Álamo. Tutorial de Introducción a MATLAB. 2012.
- 59 WESLEY, Addison y . BECK, Kent. Una explicación de la programación extrema: aceptar el cambio. 2002.
- 60 BECK, Kent. *Extreme Programming Explained*. 1999.
- 61 PRESSMAN, Roger S. *Software engineering: a practitioner's approach*. 2005.
- 62 GARROTE, Pilar Robles y CARMEN ROJAS, Manuela del. La validación por juicio de expertos: dos investigaciones cualitativas en Lingüística aplicada.

Apéndices

Historias de Usuarios

Tabla A.1. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Cargar Imagen.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0,5	Iteración asignada: 1
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Carga la imagen en formato jpg, png o bmp, donde el usuario especifica la dirección donde se encuentra la imagen.	
Observaciones:	

Tabla A.2. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Aplicar Filtros Lineales.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2,5	Iteración asignada: 1
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Pueden ser seleccionados los filtros gaussiano y la media, los cuales permiten eliminar el ruido existente en la imagen, en el caso de ser seleccionado la media se obtiene una difuminación de la imagen, produciendo un suavizado en los bordes, mientras que al aplicar el gaussiano se obtiene un efecto de desenfoque en la imagen.	
Observaciones:	

Tabla A.3. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Aplicar Filtros No Lineales.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2,5	Iteración asignada: 1
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Pueden ser seleccionados los filtros Homomórfico y Mediana, en caso de ser seleccionado el Homomórfico la imagen obtenida permite separa los componentes de iluminación y reflectancia, mientras que si se selecciona la Mediana se disminuye el ruido en la imagen obtenida.	
Observaciones:	

Tabla A.4. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Aplicar Filtros Morfológicos .
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2,0	Iteración asignada: 1
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Pueden ser seleccionados los filtros de Dilatación, Erosión, Apertura y Cierre, brindando cada uno de ellos resultados diferentes al ser aplicado en las imagen. A continuación se explicaran los resultados que brindan cada uno de estos métodos al ser aplicados a la imagen. <ul style="list-style-type: none"> • Dilatación: Se obtiene una imagen en escala de grises, la cual tiene como resultado una señal de mayor valor, es decir, una imagen más clara, puesto que la dilatación maximiza el valor de la señal. • Erosión: Se obtiene una imagen en escala de grises, la cual va a tener una señal de menor valor, es decir, una imagen más oscura, ya que la erosión pretende minimizar el valor de la señal. • Cierre: Se obtiene una imagen en escala de grises, eliminando los objetos oscuros que se encuentran en la imagen. • Apertura: Se obtiene una imagen en escala de grises, eliminando los objetos claros que se encuentran en imagen. 	
Observaciones:	

Tabla A.5. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Seleccionar región de interés.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto

Continúa en la próxima página

Tabla A.5. Continuación de la página anterior

Puntos estimados: 2,5	Iteración asignada: 2
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Permite realizar un corte circular a la imagen, de forma manual, por lo que es el usuario el que define el tamaño de la porción de la imagen que estime conveniente utilizar.	
Observaciones:	

Tabla A.6. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Obtener los bordes de la imagen.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2,0	Iteración asignada: 2
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
<p>Descripción: Pueden ser seleccionados varios métodos para obtener los bordes de la imagen tales como: Canny, Sobel y LoG. A continuación se explicarán los resultados que se obtienen de aplicar cada uno de ellos.</p> <ul style="list-style-type: none"> • Sobel: Se obtiene una imagen en escala de grises, donde solamente el algoritmo detecta los bordes en las parte claras de la imagen. • LoG: Se obtiene una imagen en escala de grises, donde a diferencia del Sobel se detectan mayor cantidad de bordes, pero hay algunos bordes significativos que no son detectados en la imagen. • Canny: Se obtiene una imagen en escala de grises, pero a diferencia de los anteriores algoritmos, se obtienen resultados favorables, logrando detectar todos los bordes significativos de la imagen. 	
Observaciones:	

Tabla A.7. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Binarizar la imagen.
Usuario: Especialista.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2,0	Iteración asignada: 2
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Permite separar la imagen en dos clases mostrando el fondo en negro y la opacidad de color blanco.	
Observaciones:	

Tabla A.8. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Manipular la imagen.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Se utiliza para mover la imagen a través del panel imagen.	
Observaciones:	

Tabla A.9. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Obtener imagen original.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1,0	Iteración asignada: 3
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Esta funcionalidad permite que después de realizar algún tipo de operación en la imagen esta pueda ser llevada a su etapa original.	
Observaciones:	

Tabla A.10. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Minimizar y maximizar la imagen.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0,5	Iteración asignada: 3
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Se utiliza para disminuir o aumentar el tamaño de la imagen.	
Observaciones:	

Tabla A.11. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Mostrar características de la imagen.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1,0	Iteración asignada: 3

Continúa en la próxima página

Tabla A.11. Continuación de la página anterior

Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.
Descripción: Esta funcionalidad permite mostrar debajo de la imagen sus características principales.
Observaciones:

Tabla A.12. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Guardar cambios realizados en la imagen.
Usuario: Especialista.	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1,0	Iteración asignada: 3
Programador responsable: Odette Romero Rodríguez y Liliana Rodríguez Martínez.	
Descripción: Esta funcionalidad permite guardar los cambios realizados en el sistema.	
Observaciones:	

Tabla B.1. Tarjeta CRC # 4

Tarjeta CRC	
Clase: Contraste	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Permite modificar el contraste de las imágenes teniendo en cuenta su correspondiente histograma. 	<ul style="list-style-type: none"> • Filtros

Tabla B.2. Tarjeta CRC # 5

Tarjeta CRC	
Clase: Filtros	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Permite eliminar y desenfocar la imagen utilizando filtros lineales. • Permite utilizar filtros morfológicos. • Permite atenuar el ruido sin distorsiones en la imagen utilizando filtros no lineales. 	<ul style="list-style-type: none"> • Contraste

Tabla B.3. Tarjeta CRC # 6

Tarjeta CRC	
Clase: Borde	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla B.3. Continuación de la página anterior

<ul style="list-style-type: none"> • Permite detectar bordes teniendo en cuenta transiciones brusca de los niveles de gris o color. • Permite aplicar operadores diferenciales para la detección de borde. 	
--	--

Tabla B.4. Tarjeta CRC # 7

Tarjeta CRC	
Clase: Regiones	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Permite analizar la similitud de los píxeles que forman los segmentos en la imagen. • Permite realizar una segmentación basada en textura. • Permite una segmentación basada en umbral. 	

APÉNDICE C

Pruebas

C.1. Pruebas de aceptación

Tabla C.1. Caso de prueba de aceptación de la HU 2.

Caso de prueba de aceptación
Código: HU_P2 Historia de Usuario: 2
Nombre: Aplicar filtros lineales.
Descripción: Comprueba la funcionalidad que permite disminuir el ruido de la imagen.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución: Para aplicar filtro de la media <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de media en los filtros lineales. • El sistema muestra la imagen en escala de grises donde se atenúa el ruido y se desenfoca la imagen digital. Para aplicar filtro gaussiano <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción del filtro gaussiano en los filtros lineales. • El sistema muestra la imagen en escala de grises donde se atenúa el ruido y se desenfoca la imagen digital.
Resultado Esperado: Se muestra la imagen, con una disminución del ruido introducido en el proceso de captura.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla C.2. Caso de prueba de aceptación de la HU 3.

Caso de prueba de aceptación
Código: HU_P3 Historia de Usuario: 3
Nombre: Aplicar filtros no lineales.
Descripción: Comprueba la funcionalidad que permite disminuir el ruido o iluminar la imagen.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución: Para aplicar filtro homomórfico <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de homomórfico en los filtros no lineales. • El sistema muestra la imagen en escala de grises separando los elementos de iluminación y reflectancia. Para aplicar filtro de la mediana <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de filtro mediana en los filtros no lineales. • El sistema muestra la imagen en escala de grises donde se atenúa el ruido.
Resultado Esperado: Se muestra la imagen con una disminución del ruido introducido en el proceso de captura y sin eliminar los bordes como lo haría un filtro lineal en el caso de utilizar la mediana, mientras que si se utiliza el homomórfico se ilumina la imagen.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla C.3. Caso de prueba de aceptación de la HU 4.

Caso de prueba de aceptación
Código: HU_P4 Historia de Usuario: 4
Nombre: Aplicar filtros morfológicos.
Descripción: Comprueba la funcionalidad que permite maximizar o minimizar el valor de la señal y eliminar los objetos oscuros o claros en la imagen.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
<p>Entrada/Pasos de Ejecución:</p> <p>Para aplicar filtro de Dilatación</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Dilatación en los filtros Morfológicos. • El sistema muestra la imagen en escala de grises resaltando los colores claros. <p>Para aplicar filtro de Erosión</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Erosión en los filtros Morfológicos. • El sistema muestra la imagen en escala de grises resaltando los colores oscuros. <p>Para aplicar filtro de Cierre</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Cierre en los filtros Morfológicos. • El sistema muestra la imagen en escala de grises eliminando los objetos claros. <p>Para aplicar filtro de Apertura</p> <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Apertura en los filtros Morfológicos. • El sistema muestra la imagen en escala de grises eliminando los objetos oscuros.
Resultado Esperado: Se muestra una imagen más clara en el caso de aplicar Dilatación o Apertura y una imagen más oscura en el caso de aplicar Erosión o Cierre.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla C.4. Caso de prueba de aceptación de la HU 7.

Caso de prueba de aceptación
Código: HU_P7 Historia de Usuario: 7
Nombre: Obtener bordes de la imagen.
Descripción: Comprueba la funcionalidad que permite segmentar las regiones de interés en la imagen utilizando segmentación por borde.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución: Para aplicar el algoritmo de Sobel <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Sobel dentro de la segmentación por borde. • El sistema muestra la imagen en escala de grises, obteniendo los bordes en las partes clara de la misma. Para aplicar el algoritmo LoG <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de LoG dentro de la segmentación por borde. • El sistema muestra la imagen en escala de grises obteniendo a diferencia del algoritmo Sobel mayor cantidad de bordes, pero aún no resalta algunos bordes significativos de la imagen. Para aplicar el algoritmo de Canny <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Canny dentro de la segmentación por borde. • El sistema muestra la imagen en escala de grises obteniendo los bordes significativos de la imagen.
Resultado Esperado: Se muestra una imagen detectando los bordes de la misma, dependiendo del algoritmo que se utilice en el sistema.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla C.5. Caso de prueba de aceptación de la HU 8.

Caso de prueba de aceptación
Código: HU_P9 Historia de Usuario:8
Nombre: Binarizar la imagen.
Descripción: Comprueba la funcionalidad que permite segmentar las regiones de interés en la imagen utilizando un umbral.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de Umbralización dentro de los métodos de segmentación. • El sistema muestra la imagen en escala de grises obteniendo las regiones de interés.
Resultado Esperado: Se muestra una imagen segmentada, obteniendo las regiones de interés de color blanco.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla C.6. Caso de prueba de aceptación de la HU 12.

Caso de prueba de aceptación
Código: HU_P12 Historia de Usuario: 12
Nombre: Seleccionar región de interés.
Descripción: Comprueba la funcionalidad que permite al usuario obtener una región circular en la imagen.
Condiciones de Ejecución: Que exista una imagen con las características y el formato definido.
Entrada/Pasos de Ejecución: <ul style="list-style-type: none"> • El usuario selecciona la imagen en formato jpg, bmp o png. • El usuario selecciona la opción de selección. • El usuario define la región circular que estime conveniente. • El sistema muestra la región seleccionada por el usuario.
Resultado Esperado: Se muestra la región circular definida por el usuario.
Evaluación de la Prueba: Prueba satisfactoria.

C.2. Técnica del camino básico

A continuación se muestra la técnica del camino básico realizado al método cargar_imagen:

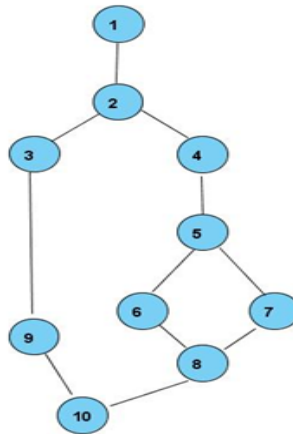


Figura C.1. Representación del grafo de flujo de camino básico del método cargar_imagen.

A partir del código fuente se diseñó el grafo como se muestra en la figura C.1, luego se procede a calcular

la complejidad ciclomática $V(G)$, la cual define el número de caminos independientes del conjunto básico de un programa, dando un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez.

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

$$V(G) = 11 - 10 + 2$$

$$V(G) = 3$$

El valor $V(G)$ expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 3 caminos:

Camino básico 1: 1->2->3->9->10

Camino básico 2: 1->2->4->5->6->8->9->10

Camino básico 3: 1->2->4->5->7->8->9->10

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

Tabla C.7. Caso de prueba de caja blanca para el camino básico 1

Entrada	Se exploran los archivos y se carga la imagen.
Resultados Esperados	Se muestra la imagen cargada en la interfaz de usuario.
Condiciones	Se carga una imagen en el formato jpg, png o bmp.

C.3. Validación

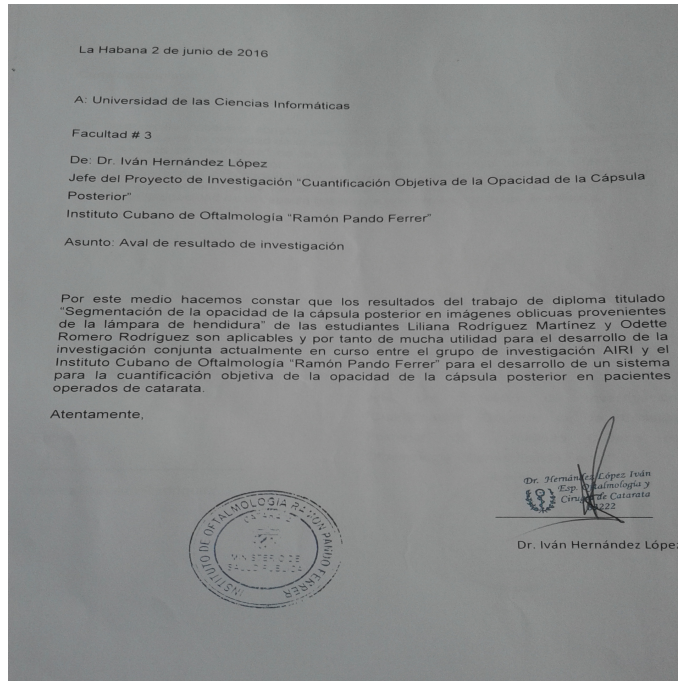


Figura C.2. Carta de aval del resultado de la investigación.

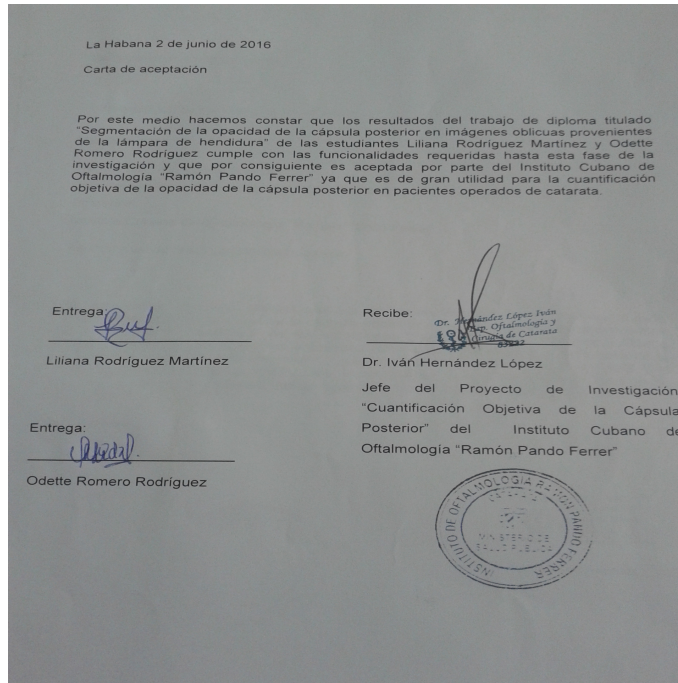


Figura C.3. Carta de aceptación del resultado de la investigación.