

Universidad de las Ciencias Informáticas

Facultad 3



Transformada de Radon para la detección de estructuras circulares en imágenes de opacidad capsular.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Dayana Díaz Torres

Tutor: Ing. Michel Álvarez Cancio

La Habana, junio 2018

"Muchas veces las personas no saben lo que quieren, hasta que se lo enseñan..."

Steve Jobs

DECLARACIÓN DE AUTORÍA

Declaro ser la autora de la presente tesis que tiene por título: “Transformada de Radon para la detección de estructuras circulares en imágenes de opacidad capsular” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Dayana Díaz Torres

Firma del Autor

Michel Álvarez Cancio

Firma del Tutor

DATOS DE CONTACTO

Ing. Michel Alvarez Cancio: mcancio@uci.cu

AGRADECIMIENTOS

Le agradezco a todos los que de una forma u otra ayudaron al desarrollo de mi tesis, a mis familiares y amigos.

Gracias.

DEDICATORIA

Esta tesis va dedicada a mi familia, los amo.

RESUMEN

La opacidad de la cápsula posterior es uno de los aspectos más importantes en la cirugía de catarata en la actualidad. La opacidad capsular es la complicación posoperatoria tardía más frecuente tras la cirugía de catarata asociada con el deterioro de la sensibilidad al contraste, problemas de deslumbramiento y disminución de la agudeza visual que conllevan a importantes repercusiones sociales, médicas y económicas. Para la cuantificación de la opacidad capsular se han creado actualmente varios sistemas basados en las imágenes en retroiluminación obtenidas por la lámpara de hendidura y han sido usados para el apoyo a la toma de decisiones. Pero la mayoría de estos sistemas tienen como deficiencia el mismo problema y es que no identifican de forma automática la opacidad capsular. La presente investigación se realizó con el propósito de desarrollar un algoritmo que permita identificar la estructura circular que representa la capsulorrexis en imágenes en retroiluminación proveniente de la lámpara de hendidura. El algoritmo propuesto se basa en el uso de técnicas de realce y mejora y segmentación, las cuales fueron implementadas a partir del uso de tecnologías libres, cumpliendo con las políticas de soberanía tecnológicas establecidas en el país.

PALABRAS CLAVES: opacidad de la capsula posterior, algoritmo, procesamiento, realce y mejora, segmentación

ÍNDICE

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN, METODOLOGÍA Y TECNOLOGÍAS EMPLEADAS.	15
1.1 Marco conceptual	15
1.2 Procesamiento digital de imágenes	16
1.2.1 Realce y mejora	16
1.2.2 Segmentación	23
1.3 Metodologías de desarrollo de software	30
1.3.1 Descripción de la metodología XP	31
1.4 Herramientas de Ingeniería del Software Asistida por Computadora (CASE)	32
1.5 Lenguaje de programación	32
1.6 Entorno de Desarrollo Integrado	33
1.7 Herramienta para la realización de pruebas al algoritmo	34
1.8 Patrones de diseño	35
1.9 Pruebas de software y validación de la segmentación	35
1.10 Conclusiones parciales	36
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	37
2.1 Fases del proceso de desarrollo	37
2.1.1 Fase de Planificación	37
2.1.1.1 Requisitos del algoritmo	37
2.1.1.2 Historias de usuario	38
2.1.1.3 Estimación de esfuerzo por historias de usuario	42
2.1.1.4 Plan de entregas	43
2.1.1.5 Plan de iteraciones	43
2.1.1.5.1 Plan de duración de las iteraciones	44
2.1.2 Diseño	44
2.1.2.1 Tarjetas CRC (Clase, Responsabilidad y Colaboración)	44
2.1.2.2 Estándares de codificación	45
2.1.2.3 Patrones de diseño	47
2.2 Propuesta de solución	48
2.3 Conclusiones parciales	57
CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA	58
3.1 Implementación	58
3.2 Técnica de validación de los requisitos	59
3.3 Pruebas	59
3.3.1 Pruebas unitarias	60
3.4 Análisis estadístico de los resultados	63

3.4.1 Pruebas estadísticas	63
3.4.2 Análisis estadístico	68
3.4.2.1 Resultados	69
3.5 Conclusiones parciales.....	70
CONCLUSIONES	72
RECOMENDACIONES	73
ANEXOS.....	74
ANEXO 1: ACTA DE ACEPTACIÓN DE LA REVISIÓN DE LOS REQUISITOS.....	74
ANEXO 2: ACTA DE ACEPTACIÓN	75
BIBLIOGRAFÍA REFERENCIADA.....	76

ÍNDICE DE FIGURAS

Figura 1 Imagen a la que se le desea aplicar la técnica Negativo de la imagen (Elizondo , 2002).	16
Figura 2 Imagen resultante al aplicarle la técnica Negativo de la imagen (Elizondo , 2002)	17
Figura 3 Imagen a la que se le desea aplicar la técnica Control de brillo (Elizondo , 2002).	17
Figura 4 Imagen resultante al aplicarle la técnica Control de brillo (Elizondo , 2002).	18
Figura 5 Imagen a la que se le desea aplicar la técnica Binarización (Elizondo , 2002)	18
Figura 6 Imagen resultante al aplicarle la técnica Binarización (Elizondo , 2002)	19
Figura 7 Imagen a la que se le desea aplicar la técnica Ampliación del contraste (Elizondo , 2002).	20
Figura 8 Imagen resultante al aplicarle la técnica Ampliación del contraste (Elizondo , 2002).	20
Figura 9 Imagen en Escala de grises	20
Figura 10 Imagen a la que se le desea aplicar la técnica la Mediana	21
Figura 11 Imagen resultante al aplicarle la técnica la Mediana	21
Figura 12 Imagen a la que se le desea aplicar la técnica de la Moda (Elizondo , 2002)	22
Figura 13 Imagen resultante al aplicarle la técnica de la Moda (Elizondo , 2002)	22
Figura 14 Imagen a la que se le desea aplicar la técnica Detector de bordes iterativo (Elizondo , 2002)	23
Figura 15 Imagen con los bordes detectados (Elizondo , 2002)	23
Figura 16 Ecuación de la transformada de Radon	26
Figura 17 Ecuación de la transformada directa	27
Figura 18 Ecuación de la retroproyección	28
Figura 20 Declaración de la clase <i>ListaEstatica</i>	46
Figura 21 Declaración del método <i>incrementarVoto</i>	46
Figura 22 Declaración de la variable <i>imageAlgoritmo</i>	46
Figura 23 Declaración del método <i>cantBlancos</i>	46
Figura 24 Métodos <i>identificarCapsulorrexis</i> y <i>detencionBordes</i> con la explicación de su uso en el algoritmo propuesto	47
Figura 25 Variable estática <i>hm</i> de la clase <i>ListaEstatica</i>	47
Figura 26 Variable subiendo del método <i>filtroAdaptado</i> de la clase <i>Segmentacion</i>	47
Figura 27 Fragmento de la clase <i>Segmentacion</i>	48
Figura 28 Imagen oblicua proveniente de la lámpara de hendidura	49
Figura 29 Imagen con las nuevas dimensiones(1024*1024)	49
Figura 30 Imagen después de aplicarle escala de grises	50
Figura 31 Imagen después de aplicarle la binarizacion	51
Figura 32 Imagen después de aplicarle el filtro de la moda	52
Figura 33 Imagen después de aplicarle el algoritmo de detección de bordes iterativo	53
Figura 34 Sinograma de la imagen	54
Figura 35 Sinograma después de aplicarle un filtro adaptado	55
Figura 36 Imagen reconstruida	56
Figura 37 Imagen resultante	56
Figura 38 Método <i>escalaGris</i> de la clase <i>Segmentación</i>	61

Figura 39 Grafo del camino básico del método <i>escalaGrises</i> .	62
Figura 46 Resultados de la métrica Tasa de correctos equilibrada	65
Figura 47 Resultados de la métrica Índice de Sokal-Sneath	66
Figura 48 Resultados de la métrica Modificación de Distancia de Hausdorff	67
Figura 49 Media	68
Figura 50 Desviación estándar	68

ÍNDICE DE TABLAS

Tabla 1 HU Cargar imagen.....	38
Tabla 2 HU Aplicar técnica de realce en la imagen.....	39
Tabla 6. Estimación de esfuerzo por HU.....	42
Tabla 7. Plan de duración de las iteraciones.....	44
Tabla 8 Tarjeta CRC de la clase <i>Circulo</i>	45
Tabla 9 Tarea de ingeniería 4.....	58
Tabla 10 Tarea de ingeniería 5.....	59
Tabla 11 Caso de prueba de caja blanca para el camino básico #3.	63
Tabla 14 Valores medios y desviaciones por cada métrica.....	67
Tabla 15 Test de Friedman sobre los resultados de las variables Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath.....	69
Tabla 16 Resultados de las pruebas.....	70

INTRODUCCIÓN

El padecimiento de la enfermedad de cataratas, al pasar del tiempo ha ido en aumento. La aparición de la enfermedad generalmente va asociada a la edad, aunque también puede ser debido a enfermedades como la diabetes. Entre los síntomas más comunes de la enfermedad están la visión borrosa u opaca y visión doble o imágenes múltiples en un ojo. En una etapa inicial estos síntomas pueden mejorar con nuevos espejuelos, mejor luz o gafas para el sol, pero cuando estas medidas son insuficientes es necesario recurrir a cirugía. (Cabrera Martínez, y otros, 2007)

La opacidad de la cápsula posterior (OCP) es uno de los aspectos más importantes en la cirugía de catarata en la actualidad (Tetz, y otros, 1997). La OCP es la complicación posoperatoria tardía más frecuente tras la cirugía de catarata asociada con el deterioro de la sensibilidad al contraste, problemas de deslumbramiento y disminución de la agudeza visual que conllevan a importantes repercusiones sociales, médicas y económicas. En la actualidad su incidencia se encuentra entre 0,7% y 47,6% (hasta 50% en Cuba) en los primeros cinco años de la cirugía. Resulta de gran importancia un análisis lo más exacto posible de la OCP para poder evaluar objetivamente el efecto de las diferentes intervenciones que actualmente se realizan para reducir su desarrollo. Entre ellas el uso de múltiples sustancias farmacológicas, modificaciones en las técnicas quirúrgicas y diferentes lentes intraoculares en cuanto a composición y diseño. (Hernández López, y otros, 2011)

Para la cuantificación de la OCP se han creado actualmente varios sistemas como EPCO del inglés (*Evaluation of posterior opacity capsule*), POCO del inglés (*Posterior Capsule Opacification*), AQUA del inglés (*Automatic quantification of secondary cataract*), y el sistema AA del inglés (*Aslam Analysis*), dichos sistemas están basados en las imágenes en retroiluminación obtenidas por las lámparas de hendiduras¹ y han sido usados para dar apoyo a los médicos a la hora de tomar una decisión. Pero la mayoría de estos sistemas tienen como deficiencia el mismo problema y es que no identifican de manera automáticamente la OCP. Además de que no emplean en imágenes provenientes del Pentacam² ni con iluminación oblicua.

¹ Lámpara de hendidura: También conocida como biomicroscopio, es un dispositivo óptico que se utiliza en oftalmología y optometría para la exploración de las estructuras de la porción anterior del ojo

² Pentacam: Tomógrafo que toma imágenes tridimensionales de la parte anterior del ojo, incluyendo la córnea, el iris, el ángulo de la cámara anterior y una porción del cristalino; con estas imágenes el Pentacam pueden evaluar las medidas de cada una de estas estructuras, se pueden diferenciar los tejidos, ubicar cuerpos extraños, determinar la extensión de ulceraciones y heridas de esta porción del globo ocular.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Centro de Investigación AIRI (*Artificial Intelligence: Research and Innovation*), el cual como principal resultado y una de sus líneas de investigación tiene al software PANDOC (Programa Analizador de Opacidad Capsular), dicho software fue realizado en colaboración con el Instituto Cubano de Oftalmología “Ramón Pando Ferrer”. PANDOC provee al oftalmólogo de una herramienta mediante la cual este es capaz de cuantificar numéricamente y detectar diferencias de opacidad (imperceptible en ocasiones para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, minimizando el sesgo de observación entre un médico y otro. Una de las imágenes con las que trabaja el software PANDOC, son las imágenes en retroiluminación provenientes de la lámpara de hendidura. En estas se ha segmentado la OCP utilizando campos aleatorios de Markov (atendiendo a texturas), pero no se ha tenido en cuenta la alternativa de localizar la capsulorrexis. Esta no es más que la región posterior que compete la cirugía de catarata, la cual se aprecia en las imágenes con formas aparentemente circulares. El grupo AIRI realizó una investigación donde se propone un algoritmo que emplea la transformada de Hough para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del Pentacam, pero el mismo tiene como limitación que solo analiza este tipo de imágenes. A raíz de este problema el cliente resalta la necesidad de realizar este procedimiento para identificar la capsulorrexis en imágenes en retroiluminación provenientes de la lámpara de hendidura. Además, en algunos estudios se ha demostrado que el método de la transformada Radon es equivalente al de la transformada de Hough, la diferencia reside simplemente en su interpretación computacional y la transformada de Hough es más costosa computacionalmente (M.v, y otros, 2004).

La situación problemática antes descrita ha generado el siguiente **problema a resolver**:

¿Cómo identificar la estructura circular que representa la Capsulorrexis en imágenes con opacidad de la cápsula posterior de pacientes operados de catarata?

Como **objeto de estudio** la segmentación de imágenes médicas enmarcado en el **campo de acción** la detección de estructuras circulares.

Teniendo en cuenta el problema antes expuesto se define como **objetivo general** desarrollar un algoritmo para la identificación de las estructuras circulares que representan la Capsulorrexis en imágenes en retroiluminación de opacidad capsular.

Del objeto de estudio se desglosan los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación relacionado con la identificación de estructuras circulares en imágenes médicas.
- Diseñar la propuesta de solución que permita identificar las estructuras circulares que representan la Capsulorrexis en imágenes de opacidad capsular.
- Implementar un algoritmo para la detección de estructuras circulares atendiendo a los elementos del diseño obtenidos.
- Evaluar el algoritmo propuesto utilizando imágenes de referencia marcada por los especialistas mediante métricas de evaluación bi-variada, para cuantificar el desempeño del software desarrollado.

Para dar cumplimiento al objetivo de la investigación plantean las siguientes **tareas investigativas**:

- Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
- Revisión de las técnicas de realce, filtrado y mejora en imágenes médicas.
- Revisión de las técnicas de identificación y binarización de bordes en imágenes médicas.
- Estudio de la transformada circular de Hough para la identificación de estructuras circulares en imágenes.
- Estudio de la transformada de Radon para la identificación de estructuras circulares en imágenes.
- Descripción de los pasos a seguir para el uso del algoritmo de segmentación propuesto.
- Diseño la solución propuesta en función de los requisitos especificados.
- Implementación del algoritmo diseñado para la identificación de la Capsulorrexis en imágenes en retroiluminación.
- Validación de los resultados mediante pruebas estadísticas no paramétricas y variables de calidad de la segmentación.
- Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

Para dar solución al objetivo planteado, teniendo en cuenta lo antes presentado se plantea la siguiente **hipótesis**: Si se desarrolla un algoritmo para la identificación automática de la Capsulorrexis mediante la

transformada de Radon, se podrá identificar de forma más eficiente la opacidad capsular que usando la transformada circular de Hough en imágenes en retroiluminación.

Para lograr la realización de dichas tareas se emplearon los siguientes métodos de investigación:

Métodos teóricos:

Histórico-lógico se empleó para estudiar los algoritmos de segmentación de imágenes médicas brindando una noción de la evolución del tema a nivel global y determinar su influencia en el problema de investigación actual.

Analítico – sintético se empleó para analizar las teorías, documentos e información, permitiendo la extracción de los elementos más importantes que se relacionan con la segmentación en imágenes médicas.

Métodos empíricos:

Observación se utilizó para realizar una evaluación de la situación problemática en cuestión.

Experimento permitió realizar experimentos y pruebas a los principales algoritmos de segmentación de imágenes médicas, para examinar los resultados y escoger los más adecuados.

Estructura de la investigación

La presente investigación consta de tres capítulos, definidos de la siguiente forma:

Capítulo 1: Fundamentos de la investigación, metodología y tecnologías empleadas.

Se presentan un conjunto de conceptos relacionados con la problemática presente en la investigación. Se realiza un estudio del estado del arte de las herramientas que permiten detectar la capsulorrexia en imágenes oblicuas provenientes de la lámpara de hendidura. Además, se describen la metodología, lenguaje de programación y herramientas utilizadas en el desarrollo de la investigación.

Capítulo 2: Descripción de la solución propuesta

Se describe el proceso de desarrollo del algoritmo propuesto, guiando el mismo por las fases definidas por la metodología XP (*Extreme Programming* o Programación Extrema). Se analizan los elementos necesarios para garantizar el éxito en el proceso de desarrollo, tales como: las historias de usuario, patrones de diseño y tarjetas CRC (Clase, Responsabilidad y Colaborador). Se detallan los pasos del algoritmo para la identificación de la capsulorrexia en imágenes oblicuas provenientes de la lámpara de hendidura.

Capítulo 3: Evaluación de la solución propuesta

Se da continuidad a las fases de la metodología XP, con la realización de revisiones formales a los requisitos y pruebas de aceptación con el cliente al algoritmo desarrollado, describiéndose los principales artefactos generados, tales como las actas de aceptación de los requisitos y del algoritmo por parte del cliente.

CAPÍTULO 1: FUNDAMENTOS DE LA INVESTIGACIÓN, METODOLOGÍA Y TECNOLOGÍAS EMPLEADAS.

En el presente capítulo se analizan los principales conceptos asociados al dominio del problema de la investigación. Se realiza un estudio desde el punto de vista teórico del problema general y se definen la metodología y herramientas que guiarán el proceso de desarrollo. Se analizan las técnicas que ayudarán a la identificación de la OCP en imágenes provenientes de la lámpara de hendidura.

1.1 Marco conceptual

Para garantizar una mejor comprensión del problema de la presente investigación se consideran necesarios conocer los conceptos que se detallan a continuación:

Procesamiento digital de imágenes: Son los cambios que se le realizan a las imágenes para posteriormente extraer información de la misma (Elizondo , 2002).

Ruido en imágenes: Son píxeles que su intensidad de colores difiere con la de sus vecinos, provocando una distorsión de la imagen. El ruido más frecuente es del tipo sal y pimienta, que son píxeles de colores blanco y negro (Elizondo , 2002).

Realce y mejora: Conjunto de técnicas destinadas a realzar colores y eliminar el ruido en una imagen (Elizondo , 2002).

Segmentación: Conjunto de técnicas destinadas a identificar regiones dentro de una imagen (Elizondo , 2002).

Después de mostrados los conceptos anteriores, se hace necesario abundar en cada uno de sus componentes y técnicas. A continuación, se mencionan cada uno de ellos:

1.2 Procesamiento digital de imágenes

El procesamiento digital de imágenes cuenta con cuatro etapas: **Captura, Realce y mejora, Segmentación y Extracción de características**. Existen un conjunto de técnicas en las etapas de Realce y mejora y Segmentación que pueden ser aplicadas a imágenes médicas, a continuación, se detallan cada una de ellas con sus técnicas.

1.2.1 Realce y mejora

Las técnicas de realce y mejora analizadas durante el desarrollo de la investigación son las siguientes:

Técnica Negativo de la imagen

Las imágenes en negativo, son parecidas a los negativos fotográficos y son muy fáciles de producir mediante el uso de tablas de búsqueda. La idea es convertir aquellas porciones de la imagen que son claras en oscuras y las que son oscuras en claras. La negación de la imagen, puede resultar de utilidad cuando se quiere apreciar los detalles en las porciones brillantes de una imagen, pues el ojo humano, es más capaz de discernir los detalles en áreas oscuras de una imagen que en las áreas más brillantes (Elizondo , 2002). La figura 1 muestra una imagen a la que se le desea aplicar la técnica Negativo de la imagen y en la figura 2 la imagen obtenida al aplicar la técnica.

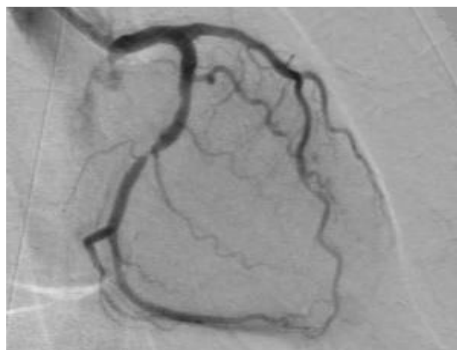


Figura 1 Imagen a la que se le desea aplicar la técnica Negativo de la imagen (Elizondo , 2002).

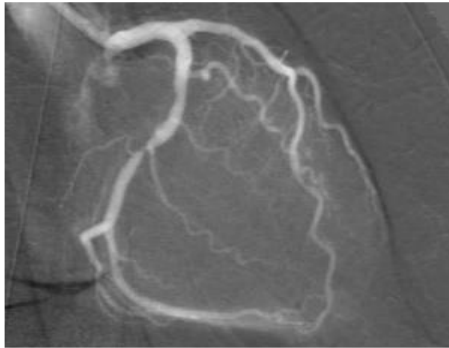


Figura 2 Imagen resultante al aplicarle la técnica Negativo de la imagen (Elizondo , 2002)

Técnica Control de brillo

En ciertas ocasiones, la apariencia de una imagen puede realizarse visualmente ajustando el brillo de la misma. Esto se logra sumando o restando un valor constante a cada píxel de la imagen de entrada. El efecto de tal transformación sobre el histograma de la imagen es desplazarlo hacia la derecha (zona más brillante), en caso de que se sume un valor constante o por el contrario lo desplaza hacia la izquierda (zona más oscura) cuando se resta un valor constante (Elizondo , 2002). La figura 3 muestra una imagen a la que se le desea aplicar la técnica Control de brillo y en la figura 4 la imagen obtenida al aplicar la técnica.



Figura 3 Imagen a la que se le desea aplicar la técnica Control de brillo (Elizondo , 2002).

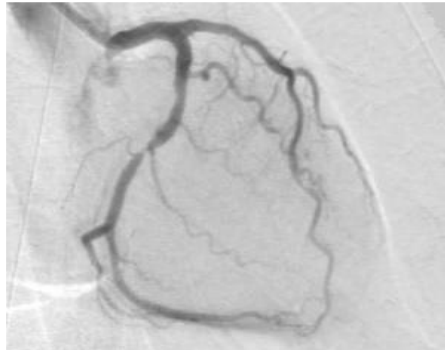


Figura 4 Imagen resultante al aplicarle la técnica Control de brillo (Elizondo , 2002).

Técnica Binarización de imágenes

La Binarización es una técnica que permite convertir imágenes con niveles de gris, en una imagen binaria (blanco y negro). De acuerdo a esta técnica, los valores de pixel en la imagen de entrada que son menores a un cierto umbral pre-especificado, son convertidos a negro, mientras que los pixeles con valores mayores al umbral, son convertidos a blanco. En algunas ocasiones se desea realizar una Binarización tal que, a una banda especificada por dos umbrales, se les asigne el color blanco, mientras que los pixeles de la imagen de entrada cuyos valores están fuera de la banda especificada, se les asigne el color negro (Elizondo , 2002). La figura 5 muestra una imagen a la que se le desea aplicar la técnica Binarización y en la figura 6 la imagen obtenida al aplicar la técnica.

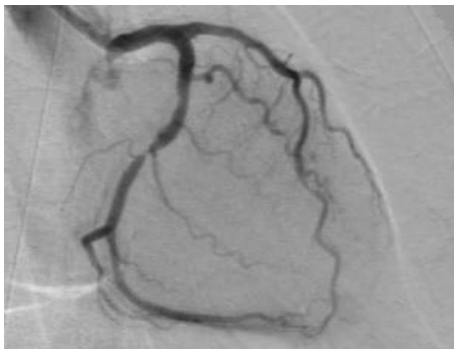


Figura 5 Imagen a la que se le desea aplicar la técnica Binarización (Elizondo , 2002)



Figura 6 Imagen resultante al aplicarle la técnica Binarización (Elizondo , 2002)

Técnica Ampliación del contraste

A esta técnica también se le conoce como dilatación del histograma (*histogram stretching*). La misma combina el uso del histograma con la utilización de las tablas de búsqueda, la razón para ello es que el histograma constituye una herramienta ideal para examinar el contraste de una imagen. Para ampliar el contraste, se realiza en el histograma una búsqueda desde los valores más pequeños de niveles de gris, hacia el máximo valor. Cuando se consiga que el número de píxeles correspondiente a un nivel de gris dado, supera un cierto umbral pre-establecido, se habrá determinado el umbral inferior (umbral 1), que estará especificado por el nivel de gris para el cual ocurre el evento mencionado. A continuación, se realiza una búsqueda en el histograma desde el valor más elevado de nivel de gris, hacia los valores más pequeños. Cuando el número de píxeles para un nivel de gris dado, supere el umbral pre-establecido, se habrá determinado el umbral superior (umbral 2) en la escala de niveles de gris. Una vez determinados los umbrales 1 y 2, se procesa la imagen mediante una transformación tal que a los píxeles de la imagen cuyo valor es inferior al umbral 1, se les asigna el valor de cero, por otra parte, si los píxeles de la imagen de entrada son superiores al valor del umbral 2, entonces se les asigna el máximo valor de gris ($L-1$). Por su parte, los píxeles comprendidos entre los dos umbrales son escalados de manera lineal.

El resultado de la ampliación del contraste será una imagen que utiliza más apropiadamente todo el rango disponible de niveles de gris y como consecuencia de ello, tendrá una apariencia más balanceada. El histograma se expande para ocupar todo el rango disponible, también la imagen de salida presenta mayor contraste y en consecuencia resulta fácil percibir todas las estructuras que la componen (Elizondo , 2002). La figura 7 muestra una imagen a la que se le desea aplicar la técnica Ampliación del contraste y en la figura 8 la imagen obtenida al aplicar la técnica.

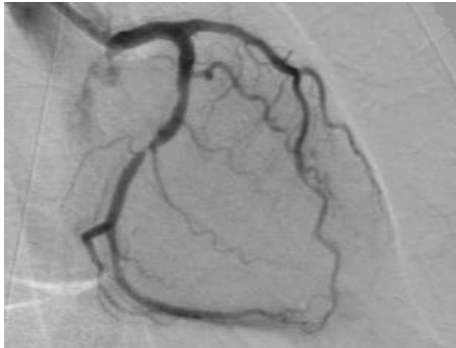


Figura 7 Imagen a la que se le desea aplicar la técnica Ampliación del contraste (Elizondo , 2002).



Figura 8 Imagen resultante al aplicarle la técnica Ampliación del contraste (Elizondo , 2002).

Técnica Escala de grises

Esta técnica consiste en llevar una imagen con una amplia intensidad de colores a tonalidad gris, preservando los cambios de contraste. La imagen escogida para aplicarle las técnicas explicadas anteriormente se encuentra en escala de grises.

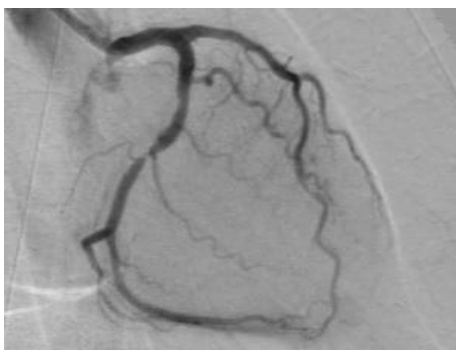


Figura 9 Imagen en Escala de grises

Técnica la Mediana

Se reemplaza el píxel original por la Mediana formada por él y los del vecindario. Esto produce mejor resultado que un paso bajo. Este filtro fuerza a que pixeles con niveles de gris diferentes a los demás se asemejen más a sus vecinos (Elizondo , 2002). Esta técnica está destinada a eliminar el ruido de tipo sal y pimienta, el cual aumenta considerablemente el tiempo de respuesta de la mayoría de los algoritmos de segmentación. La figura 10 muestra una imagen a la que se le desea aplicar la técnica la Mediana y en la figura 11 la imagen obtenida al aplicar la técnica.



Figura 10 Imagen a la que se le desea aplicar la técnica la Mediana

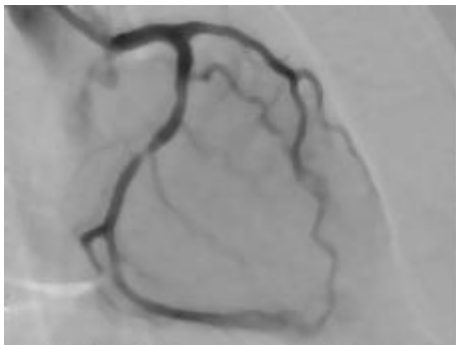


Figura 11 Imagen resultante al aplicarle la técnica la Mediana

Técnica de la Moda

Se reemplaza el píxel original por la moda formada por él y los del vecindario. Esto produce mejor resultado que un paso bajo. Este filtro fuerza a que pixeles con niveles de gris diferentes a los demás se asemejen más a sus vecinos (Elizondo , 2002). Esta técnica está destinada a eliminar el ruido de tipo sal y pimienta, el cual aumenta considerablemente el tiempo de respuesta de la mayoría de los algoritmos de

segmentación. La figura 12 muestra una imagen a la que se le desea aplicar la técnica de la moda y la figura 13 la imagen obtenida al aplicar la técnica.



Figura 12 Imagen a la que se le desea aplicar la técnica de la Moda (Elizondo , 2002)

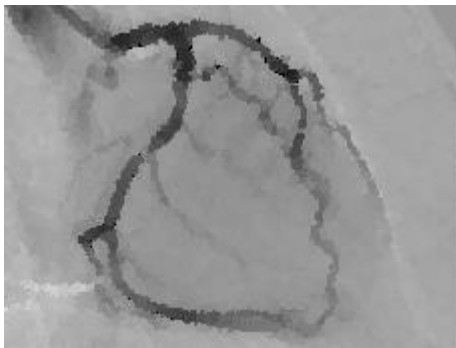


Figura 13 Imagen resultante al aplicarle la técnica de la Moda (Elizondo , 2002)

Técnica Detector de bordes iterativo.

Un borde es el límite entre dos regiones con niveles de gris distintos, de manera que la transición entre dichas regiones se pueda mirar como un cambio abrupto. La idea básica para la detección de bordes es el cálculo de una derivada, teniendo en cuenta que para una constante el valor es cero y para un cambio será diferente de cero. En el sistema de visión, para cada píxel de la imagen binaria (que se encuentra almacenada en la memoria externa del microcontrolador) se implementa el algoritmo de la siguiente manera: Se recorre la imagen (la memoria), buscando un píxel negro, es decir, si es blanco se omite el cálculo. Cuando lo encuentra busca en los cuatro vecinos por lo menos un píxel blanco el cual activa una bandera que indica que hay un borde. Si todos los cuatro vecinos son negros no es borde (URREA, y

otros, 2004). La figura 14 muestra una imagen a la que se le desea aplicar la técnica Detector de bordes iterativos y la figura 15 la imagen obtenida al aplicar la técnica.



Figura 14 Imagen a la que se le desea aplicar la técnica Detector de bordes iterativo (Elizondo , 2002)

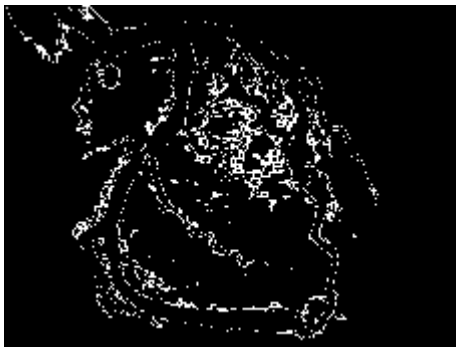


Figura 15 Imagen con los bordes detectados (Elizondo , 2002)

1.2.2 Segmentación

Las técnicas de segmentación se utilizan para realizar la identificación de estructuras anatómicas presentes en una imagen, las cuales permiten particionar la imagen en un conjunto no solapado de regiones, cuya unión es la imagen completa. En muchas ocasiones, dependiendo de la aplicación específica, el proceso de segmentación es uno de los pasos difíciles y críticos para determinar la geometría de las diversas estructuras que componen la imagen (Elizondo , 2002). Las técnicas más utilizadas en segmentación son: basada en fronteras, basada en el uso de un umbral y por crecimiento de regiones. Las técnicas de segmentación tienden a ajustarse a las siguientes reglas:

1. Las regiones resultantes del proceso de segmentación debieran ser uniformes y homogéneas respecto a alguna característica, tal como el nivel de gris o la textura.

2. Las regiones interiores debieran ser simples y no incluir abundantes huecos o estructuras ruidosas.
3. Las regiones adyacentes en una segmentación debieran tener valores diferentes con respecto a la característica según la cual son uniformes.
4. Los límites de cada segmento debieran ser lo más simple posibles.

El cumplimiento de estas propiedades resulta difícil y generalmente se obtienen regiones en las que se observan la presencia de huecos (Paula Vidal, 2016).

Técnicas basadas en fronteras

Esta técnica tiene como objetivo resaltar fronteras del objeto a segmentar. Tiene como ventaja la simplicidad una vez encontradas las fronteras, pero presenta dificultades para encontrar los bordes (Martínez de Dios, 2016).

1. Transformada de Hough

“La transformada de Hough es un método de extracción de características patentado por Paul Hough en 1962, para la detección de bordes en imágenes parametrizadas, es decir, los objetos que se desean detectar se representan por medio de una ecuación conocida. La idea es transformar la imagen del espacio cartesiano xy a un espacio de parámetros y a través de un proceso de votos identificar los puntos que describen una línea recta u otra formulación de bordes” (Hough, 1962).

Es una herramienta que permite detectar curvas en una imagen. Es una técnica muy robusta frente al ruido y a la existencia de brechas en la frontera del objeto, por ello es muy utilizada para aislar características de forma particular dentro de una imagen o escaneo láser (Berrío, y otros, 2012). Inicialmente esta técnica solo se aplicaba a la detección de rectas en una imagen, más tarde se extendió para identificar cualquier figura que se pudiera describir con unos cuantos parámetros. A la hora de aplicar la transformada de Hough a una imagen es necesario obtener primero una imagen binaria de los píxeles que forman parte de la frontera del objeto (Departamento de Ingeniería electrónica, 2015).

Existe una variante para la detección de formas circulares donde la transformada de Hough es utilizada para aislar características de forma particular dentro de una imagen. La idea básica es encontrar curvas que puedan ser parametrizadas como círculos. Se puede analíticamente describir un segmento de línea de varias formas, la ecuación conveniente para describir un conjunto de líneas es la notación paramétrica o normal (ecuación 1) (URREA, y otros, 2004).

$$p = x \cos \phi + y \sin \phi \quad (1)$$

Donde p es la longitud de una normal desde el origen hasta la línea y ϕ es el ángulo de p con respecto al eje x .

2. Contornos Activos

Estos contornos modelan las fronteras entre un objeto, el fondo y el resto de objetos de la imagen. Permiten extraer los contornos de los objetos de interés basándonos en modelos que utilizan información a priori de la forma de los objetos. Estas técnicas son mucho más robustas frente a la presencia de ruido y otros elementos espúreos y permiten segmentar imágenes mucho más complejas que las imágenes para las que eran aplicables los métodos de segmentación de bajo nivel, como son las imágenes médicas, por ejemplo. Las soluciones proporcionadas por estas técnicas en general no requieren procesamiento posterior y son directamente interpretables, puesto que se basan en un modelo establecido a priori. Si este modelo es el adecuado, la presencia de falsos positivos o negativos será muy pequeña. Los contornos activos se pueden clasificar en snakes (serpientes), patrones deformables y contornos dinámicos. Los snakes son mecanismos para dar cierto grado de conocimiento a priori a la interpretación de la imagen a bajo nivel. En lugar de esperar que propiedades deseables de los contornos como son continuidad y suavidad provengan de los datos de imagen, estas propiedades son impuestas desde el principio. Se impone un modelo elástico de curva continua y flexible, que posteriormente se ajustará a los datos de la imagen. Variando los parámetros de elasticidad de la curva se puede controlar la cantidad de información a priori que se asume. El modelado a priori se puede hacer más específico construyendo un conjunto de curvas flexibles que formarán el contorno global, con un conjunto de parámetros que controlen las variables cinéticas de la curva, como, por ejemplo, los tamaños de las diferentes partes y los ángulos con las que se unen. En caso de que sea necesario localizar objetos en movimiento, las cosas se complican aún más dando lugar al problema del seguimiento de objetos. Esto se denomina modelado dinámico, para el que es necesario añadir inercia, fuerzas de restauración y factor de amortiguamiento al snake estático.

Cuando las curvas sean de seguimiento y utilicen información dinámica a priori reciben el nombre de contornos dinámicos. (Martin, 2008)

3. Transformada de Radon

La base matemática para la reconstrucción de una imagen a partir de una serie de proyecciones es la **Transformada de Radon**, llamada así por Johann Radon. Es una transformación integral que consiste en la integral de una función sobre un conjunto de rectas. Para una función arbitraria $f(x, y)$, su transformada de Radon $\mathfrak{R}\{f(x, y)\}$ se define como una integral de la función $f(x, y)$ a lo largo de la línea L , en la siguiente imagen se muestra la ecuación:

$$\mathfrak{R}\{f(x, y)\} = \int_L f(x, y) dl$$

Figura 16 Ecuación de la transformada de Radon

Esta transformada en su versión bidimensional y tridimensional, fueron introducidas por primera vez en un artículo en el año 1917. La transformada inversa de Radon se calcula mediante el método de las retroproyecciones filtradas. Existe un algoritmo inverso de la transformada de Radon, que se llama algoritmo de las retroproyecciones filtradas³. Este algoritmo es computacionalmente eficiente y es capaz de dar valor a un conjunto finito de píxeles, a partir de un número finito de proyecciones. Sus propiedades básicas son linealidad, homogeneidad, traslación (García Ruiz, 2015).

La transformada de Radon es una herramienta fundamental en muchas áreas, pues reconstruye una imagen a partir de una colección de sus proyecciones en varias direcciones, siendo el campo de aplicación original de esta técnica la radioastronomía. No obstante, la aplicación más importante y conocida de la transformada Radon se da en el campo de la imagenología médica proporcionando las bases teóricas para los dispositivos de tomografía computarizada. Como ejemplos de aplicaciones se encuentra la detección de barcos en imágenes de Radar de Apertura Sintética (SAR por sus siglas en inglés), la detección de blancos pequeños en clutter de mar que no son más que ruido provocado por los ecos o reflexiones en elementos ajenos al sistema, así como una herramienta para la detección de curvas

³ Retroproyección filtrada: se planteó inicialmente como un algoritmo de inversión de la transformada de Radon; este hace uso de herramientas matemáticas propias del análisis de Fourier como lo son el teorema de proyecciones y el producto convolución además de una elaborada construcción geométrica que permitirá reconstruir la imagen a partir de un conjunto de rectas (Juzga, y otros, 2011).

en imágenes digitales ruidosas. También se usa para la segmentación de imágenes médicas, usando la combinación del histograma multinivel basado en técnicas de segmentación con la transformada Radon reversa para incrementar la eficiencia de la segmentación de imágenes. Para la detección de vasos sanguíneos en la retina en imágenes a color basada en la transformada Radon y reconstrucción morfológica (González Betancourt, 2014). Aunque la transformada de Radon surge del estudio de un problema de carácter aplicativo, constituye hoy en día uno de los aportes más importantes a la matemática y ha arrojado aplicaciones en campos tan diversos como la astronomía, la óptica, y la geofísica entre otros (Juzga, y otros, 2011).

La inversión de la transformada de Radon presenta algunos inconvenientes a nivel práctico debido a que es imposible atravesar una sección transversal de un objeto con infinitos rayos o proyecciones. La reconstrucción obtenida con este método es considerablemente buena, a pesar del ruido presente, si se analizan aspectos de forma y contraste en la imagen. Por tanto, es natural pensar que la búsqueda de nuevos algoritmos de reconstrucción o de herramientas que mejoren el método de retroproyección filtrada está aún lejos de su fin.

Transformada Radon directa

La transformada Radon bidimensional continua es la integral de una función sobre un conjunto de rectas. Una recta $p = x \cos(\theta) + y \sin(\theta)$ donde p es la mínima distancia desde la recta al origen y θ es el ángulo que forma el eje x con el vector posición del punto de la recta más cercano al origen. Las coordenadas de una imagen son usualmente positivas con el origen en su esquina superior izquierda; no obstante, para el cálculo de la RT-2D el origen debe desplazarse al centro de la imagen, lo que involucra un desplazamiento espacial. La ecuación de la transformada se presenta en la siguiente imagen

$$R_f(\theta, p) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos(\theta) + y \sin(\theta) - p) dx dy$$

Figura 17 Ecuación de la transformada directa

Calcula las integrales de líneas de múltiples fuentes a lo largo de rayos paralelos que atraviesan el objeto, en este caso la imagen, en una cierta dirección. Para representar una imagen, toma múltiples proyecciones de rayos paralelos, de diferentes ángulos, rotando la fuente alrededor del centro de la imagen (González Betancourt, 2014).

Transformada Radon reversa

Johann Radon en 1917 demostró matemáticamente que es posible recuperar una función de dos variables si se conocen las integrales de la misma en todas las líneas del plano. Este resultado equivale a que se puede reconstruir la imagen de un objeto bidimensional si se tienen suficientes proyecciones del objeto en 180 grados. Los métodos para reconstruir una imagen a partir de su sinograma con la transformada Radon reversa haciendo que el resultado de la misma $f'(x, y)$ tienda a $f(x, y)$ con mayor exactitud se dividen en (González Betancourt, 2014):

- Métodos algebraicos: Técnicas de Reconstrucción Algebraica (ART por sus siglas en inglés)
- Métodos analíticos: Retroproyección (BP) y Retroproyección Filtrada (FBP por sus siglas en inglés)
- Métodos iterativos: Reconstrucción Iterativa (IR por sus siglas en inglés)

El Teorema de la Proyección (Jain, 1989) es un resultado fundamental, pues demuestra que a partir de suficientes proyecciones se puede conseguir una aproximación suficientemente buena de $f'(x, y)$, lo cual trae consigo poder reconstruir $f(x, y)$. Si tal reconstrucción es posible, debe existir una forma de expresar la transformada inversa de Radon, pues dicha operación consiste en recuperar la señal $f(x, y)$ a partir de sus proyecciones.

La recuperación mediante retroproyección es el aporte de cada proyección sobre los píxeles de la imagen desde su ángulo de adquisición entre el número de proyecciones. Para la reconstrucción de la imagen $x[n_1, n_2]$, donde $R_x(n_1', \theta)$ es su sinograma⁴ en las direcciones $0 \leq \theta \leq 180$, puede calcularse su retroproyección mediante la ecuación que se muestra en la siguiente imagen

$$x'[n_1, n_2] = \sum_{n_1' \rightarrow \infty}^{\infty} \sum_{\theta \rightarrow 0}^{\pi} R_x(n_1', \theta)(n_1 \cos(\theta) + n_2 \sin(\theta) - n_1')$$

Figura 18 Ecuación de la retroproyección

La principal dificultad de la retroproyección es que tras su única iteración no alcanza la solución esperada, por tanto, es necesario mejorar el resultado (González Betancourt, 2014).

⁴ Representación gráfica de una matriz, en la que el índice de las filas depende del ángulo α que forma la línea de medida con el eje x y el índice de las columnas depende de la distancia d de la línea de medida al centro de coordenadas del sistema de detección. Según esta representación todas las líneas de medida, con la misma inclinación (mismo ángulo), estarán situadas en una fila una matriz, y todas las líneas de medida posibles de un detector con el resto quedarán situadas sobre una diagonal cuya pendiente depende de la posición del detector común, y el tamaño de la matriz (número de celdas) es tal que a cada una de ellas le corresponde una línea de medida definida por los detectores (Fernández Marrón, y otros, 2007)

La retroproyección filtrada es la variante mejorada, la misma realiza las altas frecuencias mediante un filtrado paso-alto de las proyecciones del sinograma previo a la reconstrucción. Para ello es necesario utilizar el teorema de línea central de Fourier de la transformada Radon. El problema de utilizar el teorema de la línea central de Fourier como método para recuperar la imagen a partir de sus proyecciones radica en el muestreo no uniforme. Una vez filtrado el sinograma en el dominio de la frecuencia se retorna al dominio del espacio mediante la transformada de Fourier inversa 1D de sus columnas (González Betancourt, 2014).

El método de retroproyección filtrada es el más utilizado en tomografía para reconstrucción tanto en imágenes de transmisión como de emisión y es actualmente el algoritmo recomendado por la Asociación Nacional de la Industria Eléctrica (NEMA por sus siglas en inglés) debido a su bajo costo computacional, velocidad y sencillez (González Betancourt, 2014).

Basada en el uso de un umbral

El objetivo de este tipo de segmentación es diferenciar objetos de distintos niveles de gris (Martínez de Dios, 2016). Permite separar un objeto dentro de la imagen del fondo que lo circunda, la técnica se basa en comparar alguna propiedad de una imagen con un umbral fijo o variable, realizando tal comparación para cada uno de los píxeles que conforman la imagen, si el valor de la propiedad de un píxel supera el valor del umbral, entonces el píxel pertenece al objeto, en caso contrario, el píxel pertenece al fondo. Cuando la segmentación se realiza basada en el nivel de gris de la imagen, el valor del nivel de gris de cada píxel debe ser comparado con el umbral, para decidir si tal píxel pertenece al objeto o al fondo. La imagen de salida, es una imagen binaria en la cual aquellos píxeles cuyo valor es uno, pertenecen al objeto y los píxeles cuyo valor es cero, pertenecen al fondo (Elizondo , 2002).

La selección del valor del umbral, se realiza generalmente a partir del histograma de la imagen. Así si una imagen está compuesta de un objeto que aparece en la escena sobre un fondo, entonces es de esperar que el histograma sea bimodal, es decir, si el objeto es más claro que el fondo, pues en el histograma aparecerán dos picos, el ubicado en los valores de gris más elevados correspondiente al objeto y otro pico para niveles de gris más bajos, correspondientes al fondo (Elizondo , 2002).

Los métodos umbralización o del valor umbral son algoritmos cuya finalidad es segmentar gráficos rasterizados, separando los objetos de una imagen que nos interesen del resto. Con la ayuda de los métodos de valor umbral en las situaciones más sencillas se puede decidir qué píxeles conforman los objetos buscados y qué píxeles son sólo el entorno de estos objetos. Este método es especialmente útil

para separar el texto de un documento del fondo de la imagen (papel amarillento, con manchas y arruguita) y así poder llevar a cabo el reconocimiento óptico de texto (OCR) con más garantías de obtener el texto correcto. Esto es especialmente útil si se quiere digitalizar libros antiguos, en los que el contraste entre el texto (que ya ha perdido parte de sus pigmentos) y el papel (oscurecido y manoseado) no es demasiado elevado. La umbralización es un método que busca segmentar imágenes escalares creando una partición binaria de las intensidades de las imágenes. La umbralización es cuando una imagen en gris es binarizada consiguiendo un umbral óptimo T y con ese valor se separan los píxeles en dos regiones, una de zonas claras y otra de zonas oscuras (Elizondo , 2002). En la umbralización hay dos posibles situaciones:

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

Por crecimiento de regiones

De acuerdo a esta técnica, se buscan píxeles que tengan características similares (por ejemplo, niveles de gris similares) y que adicionalmente sean vecinos. El método comienza con un píxel, el cual es seleccionado automáticamente o proporcionado por el usuario y a continuación examina los píxeles vecinos para decidir si tienen características similares. De ser así, el píxel vecino que cumpla con tal condición de similitud, es agrupado junto con los anteriores para conformar así una región (Elizondo , 2002). El algoritmo finaliza cuando dejan de haber regiones adyacentes que cumplan criterios (Martínez de Dios, 2016).

1.3 Metodologías de desarrollo de software

Las metodologías de desarrollo de software se emplean para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, se han ido desarrollando diferenciándose por

su fortaleza y debilidad. Las metodologías se clasifican en dos grupos, las metodologías tradicionales o pesadas y las metodologías ágiles o ligeras. En el proceso de desarrollo de proyectos grandes y que generan mucha documentación se recomienda emplear las metodologías tradicionales, mientras que para proyectos pequeños se recomiendan las metodologías ligeras. Teniendo en cuenta las características de la presente investigación se escoge una metodología ágil. Dentro de las metodologías ágiles se destacan: AUP (Proceso Unificado Ágil), Scrum, XP (Programación Extrema) y SXP (SCRUM-XP). Se define la metodología XP para guiar el proceso de desarrollo, debido a que se está en presencia de un proyecto pequeño, de corta duración, con un reducido equipo de desarrollo, donde el cliente forma parte del mismo, permitiendo la comunicación y retroalimentación frecuente ambas partes. Esta metodología ha sido la utilizada durante el desarrollo del software PANDOC y es la metodología definida por el grupo de investigación científica *Inteligencia Artificial, Programación e Innovación (AIRI)*, al cual pertenece la autora del trabajo.

1.3.1 Descripción de la metodología XP

El ciclo de vida de un proyecto XP es muy dinámico, se puede separar en las siguientes fases (Sánchez Mendoza, 2004):

- Planificación
- Diseño
- Desarrollo
- Pruebas

A continuación, se describen cada una de las fases.

Planificación: Consiste en realizar una recopilación de todos los requerimientos del proyecto, se crean las HU, el plan de iteraciones y el plan de entregas (Sánchez Mendoza, 2004).

Diseño: Se basa en conseguir diseños simples y sencillos, con el objetivo de procurar hacerlo todo lo menos complicado posible para el cliente o usuario, se crean las tarjetas CRC, las cuales definen una clase expresando sus funcionalidades y las otras clases con las que colabora (Sánchez Mendoza, 2004).

Desarrollo: Consiste en establecer una buena comunicación entre el equipo y el cliente, para que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere. Además, se definen las tareas de ingeniería para que los desarrolladores tengan una guía para implementar todas las historias de usuario (Sánchez Mendoza, 2004).

Pruebas: Esta fase se comprueba el funcionamiento de la codificación que se halla implementado, garantizando la evaluación de las distintas tareas en las que ha sido divididas las historias de usuario (Sánchez Mendoza, 2004).

1.4 Herramientas de Ingeniería del Software Asistida por Computadora (CASE)

Se selecciona Visual Paradigm para UML en su versión 8.0 para la modelación del proceso de desarrollo de la herramienta propuesta porque es multiplataforma, fácil de usar, proporciona facilidad de trabajo con modelos UML y permite la generación de diversos diagramas como los de clases, de objetos, de casos de uso del negocio y de paquetes, además de generar código a partir de los mismos y guardar todo el modelo en un solo fichero. También para la selección se tuvo en cuenta que la UCI posee una licencia para su uso.

1.5 Lenguaje de programación

Teniendo en cuenta que el software PANDOC está desarrollado en el lenguaje Java y que es un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, securizado, independiente de las arquitecturas, portable, eficaz, multihilo y dinámico se decidió utilizar el mismo en el desarrollo de la solución propuesta. A continuación, se detallan algunas de sus características:

- **Interpretado:** se ejecuta en una máquina virtual. No es necesario editar los enlaces (obligatorio en C++) antes de ejecutar un programa. En Java, por lo tanto, solo hay dos etapas, la compilación y la ejecución. La máquina virtual se encarga de la operación de edición de enlaces en tiempo de ejecución del programa (Groussard, 2014).
- **Simple:** se enfoca en el contexto de los lenguajes orientados a objetos y elimina la complejidad de otros lenguajes como C (Groussard, 2014).
- **Robusto:** es un lenguaje fuertemente tipado y estricto. se verifica el código en el momento de la compilación y también de la ejecución lo que permite reducir los errores y los problemas de incompatibilidad de versiones (Groussard, 2014).
- **Multitarea:** Java permite desarrollar aplicaciones q ponen en marcha la ejecución simultanea de varios hilos (o procesos ligeros). Esto permite efectuar simultáneamente varias tareas, con el fin de aumentar la velocidad de las aplicaciones, ya se compartiendo el tiempo del CPU o repartiendo las tareas entre varios procesadores (Groussard, 2014).

- **Dinámico:** no es necesario que compile todas las clases de un programa para que este funcione. Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases (Groussard, 2014).
- **Portable:** un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java, ya que su código compilado es interpretado (Groussard, 2014).
- **Independiente de las arquitecturas:** El compilador java no produce un código específico para un tipo de arquitectura. El compilador genera un bytecode (lenguaje binario intermedio) que es independiente de cualquier arquitectura, de todo sistema operativo y de todo dispositivo de gestión de la interfaz gráfica de usuario. La ventaja de este bytecode reside en su fácil interpretación o transformación dinámica en código nativo para aumentar el rendimiento. Basta con disponer con una máquina virtual, específica de su plataforma para hacer funcionar un programa java. Esta última se encarga de traducir el bytecode a código nativo (Groussard, 2014).

1.6 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado, traducido del inglés Integrated Development Environment (IDE) es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Provee de un marco de trabajo amigable para la mayoría de los lenguajes de programación (Sanchez, 2014).

Teniendo en cuenta que el Netbeans es libre, se integra perfectamente con Java que es el lenguaje de programación definido para el desarrollo de la solución propuesta, el equipo posee dominio de esta herramienta, además de un conjunto de elementos que se describen a continuación, se escogió para el desarrollo de la solución informática se seleccionó la herramienta Netbeans en su versión 8.0.

Netbeans es un entorno de desarrollo integrado, modular y de base estándar. Consiste en un IDE de código abierto y una plataforma de aplicación, las cuales pueden ser usadas como una estructura de soporte general para compilar cualquier tipo de solución. Presenta una interfaz amigable e intuitiva y tiene todas las herramientas para crear soluciones profesionales ya sean de escritorio, empresariales, web, móviles y aplicaciones SOA (Arquitectura Orientada a Servicios), no solo en Java sino también en C/C++ y

Ruby. Netbeans ya viene con plugins⁵ y módulos integrados, evitando tener que configurar el ambiente, brindando todo el entorno listo para trabajar (Sanchez, 2014).

1.7 Herramienta para la realización de pruebas al algoritmo

Para realizar las pruebas a la solución propuesta se empleará la herramienta Matlab 2015 y SPSS v22 definida por el grupo de investigación AIRI.

Matlab es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir herramientas reutilizables. Permite crear fácilmente funciones y programas especiales, a parte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos, etc. (Fernández, 2015).

Contiene el toolbox de procesamiento de imágenes que soporta imágenes y vídeos generados por una amplia gama de dispositivos, tales como dispositivos de generación de imágenes médicas, microscopios, telescopios y otros instrumentos científicos. Las técnicas de mejora de imágenes le permiten aumentar la proporción de señal a ruido y acentuar las características de las imágenes modificando los colores o las intensidades de una imagen. Además incluye rutinas de filtrado especializadas y una función de filtrado multidimensional generalizado que gestiona tipos de imágenes de enteros, ofrece diversas opciones de relleno de contornos y realiza tareas de convolución y correlación (MathWorks, 2018).

SPSS: Es un sistema amplio y flexible de análisis estadístico y gestión de información que capaz de trabajar con datos procedentes de distintos formatos generando, desde sencillos gráficos de distribuciones y estadísticos descriptivos hasta análisis estadísticos complejos que nos permitirán descubrir relaciones de dependencia e interdependencia, establecer clasificaciones de sujetos y variables, predecir comportamientos. Su aplicación fundamental está orientada al análisis multivariante de datos experimentales (López-Roldán, y otros., 2018).

⁵ plugins: es una aplicación informática que añade funcionalidades específicas a un programa principal.

1.8 Patrones de diseño

Los patrones de diseño son descripciones de la comunicación de objetos y clases que pueden personalizarse para resolver un problema de diseño en general en un contexto particular (Gamma, y otros, 1994). Es una forma reutilizable de resolver un problema común, ayudan a estandarizar el código y a hacer que el diseño sea más comprensible para otros programadores. Pueden ser agrupados en dos grandes grupos; los GRASP (*General Responsibility Assignment Software Patterns*) por sus siglas en inglés, que son patrones generales de software para asignación de responsabilidades y los GOF también conocidos como banda de los cuatro (*Gang of Four*) por sus siglas en inglés, encargados de la inicialización, agrupación y comunicación de los objetos.

1.9 Pruebas de software y validación de la segmentación

En el desarrollo de software los errores pueden empezar a darse desde el inicio del mismo. Es por eso que el desarrollo debe ir acompañado de una actividad que garantice la calidad, para ello se ejecutan un conjunto de pruebas dirigidas a un componente o al sistema en general para identificar posibles fallos de funcionamiento, configuración o usabilidad y así garantizar que se cumplan los requerimientos pactados.

Existen cuatro niveles de pruebas:

- **Pruebas de unidad:** se concentra en el esfuerzo de verificación de la unidad más pequeña del diseño, ya sea un componente o un módulo del software (Pressman, 2003).
- **Pruebas de integración:** es una técnica sistemática para confeccionar la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. El objetivo es tomar componentes a los que se les aplicó una prueba de unidad y construir una estructura de programa que determine el diseño (Pressman, 2003).
- **Pruebas de sistema:** abarca una serie de pruebas diferentes cuyo propósito principal es ejercitar profundamente el sistema de cómputo. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realizan las funciones adecuadas (Pressman, 2003).

- **Pruebas de aceptación:** consisten en formalizar la aceptación de los entregables del proyecto que se han completado. Incluye revisar los entregables con el cliente para asegurarse de que se han completado satisfactoriamente y para obtener de ellos su aceptación formal. (Figuerola, 2011)

Además, se pueden realizar pruebas estadísticas, que se utilizan para confirmar una hipótesis de trabajo o bien para explorar conjuntos de datos sin hipótesis previas. Ambas finalidades, la confirmación y la exploración, están vinculadas a la naturaleza de los objetivos del estudio, a la actitud con que el investigador se enfrenta a los datos y a los términos en que deberán interpretarse los resultados. Una hipótesis se confirma cuando se diseña un estudio con el propósito de hacerlo. Se explora cuando se rastrean datos en busca de información, sin objetivos concretos y formales que hayan gobernado el diseño del estudio. La exploración puede servir para sugerir nuevas hipótesis, pero de ningún modo para contrastarlas, sino que la confirmación deberá obtenerse en un nuevo estudio diseñado específicamente para ello. Las pruebas estadísticas que utilizan datos cuantitativos suelen realizar determinadas asunciones sobre la distribución de las variables en las poblaciones que están siendo comparadas. Estas pruebas son conocidas como pruebas paramétricas. La mayoría son robustas, es decir, que toleran relativamente violaciones de estas asunciones, especialmente si el número de sujetos estudiado es elevado. Cuando las pruebas estadísticas aplicables a las variables cuantitativas no cumplen las asunciones necesarias para su uso, se recurre a las pruebas correspondientes como si la variable de respuesta fuera ordinal (pruebas no paramétricas) (Jiménez, 2015).

1.10 Conclusiones parciales

A continuación, se detallan las conclusiones obtenidas hasta este momento en la investigación:

- En el capítulo se realizó un estudio de los conceptos relacionados con la OCP y la propuesta anterior para la detección de la Capsulorrexis basada en la transformada de Hough la cual solo está diseñada para ser aplicada en tomogramas del Pentacam hizo necesaria la implementación de un algoritmo para la detección de la Capsulorrexis en imágenes en retroiluminación.
- Se realizó el estudio de las técnicas existentes para la identificación de regiones en imágenes como son: las basadas en fronteras, por crecimiento de regiones y contornos activos lo permitió seleccionar la más adecuada para aplicar en el desarrollo de la solución propuesta.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describen los principales elementos que se tuvieron en cuenta en el proceso de desarrollo del algoritmo propuesto. El uso de la metodología XP permitió organizar el trabajo. Además, se hace referencia a los artefactos generados en cada una de sus fases, entre los que se destacan las Historias de Usuario (HU), Estimación por esfuerzo y Plan de duración de las iteraciones.

2.1 Fases del proceso de desarrollo

En el capítulo se describe como fueron aplicadas las fases de planificación, diseño y desarrollo de la metodología XP teniendo en cuenta que en el capítulo 1 se definió para guiar el proceso de desarrollo.

2.1.1 Fase de Planificación

A continuación, se describen los principales artefactos generados en esta fase.

2.1.1.1 Requisitos del algoritmo

Los requisitos son especificaciones que describen con claridad el comportamiento de un sistema, en este caso del algoritmo, para minimizar problemas en el desarrollo. Se clasifican en requisitos funcionales y no funcionales. Los requisitos funcionales permiten que el desarrollador conozca las necesidades que el cliente necesita satisfacer mediante el algoritmo. Los requisitos no funcionales, por su parte, se refieren a propiedades que debe cumplir el software, tales como capacidad de almacenamiento, fiabilidad o mantenibilidad. A continuación, se describen los requisitos del algoritmo propuesto.

Requisitos funcionales

RF1: Cargar imagen.

RF2: Aplicar técnica de realce en la imagen.

RF3: Aplicar técnica de filtrado y mejora en la imagen.

RF4: Identificar los bordes de la imagen.

RF5: Identificar la capsulorrexis mediante el algoritmo Radon.

RF6: Identificar la capsulorrexis mediante el algoritmo Hough.

Requisitos no funcionales (RNF)

Los RNF se definieron a partir de la definición que da la IEEE 830.

- **Requisitos de hardware**

RNF 1: Para ejecutar la herramienta la computadora debe contar con las siguientes características:

- 2 GB de memoria RAM.
- Más de un núcleo físico con frecuencia mayor de 1.6 GHz.

- **Requisito de software**

RNF 2: Se debe tener instalada en la computadora la máquina virtual de Java en su versión 8 o superior.

2.1.1.2 Historias de usuario

La historia de usuario (HU) es la técnica que utiliza la metodología XP para especificar los requisitos del software.

A continuación, se muestran las HU definidas en la presente investigación.

Tabla 1 HU Cargar imagen

Historia de Usuario	
Número: 1	Nombre de la Historia de Usuario: Cargar imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semana
Riesgo en desarrollo: Bajo	Puntos reales: 1 semana
Programador responsable: Dayana Díaz Torres	
Descripción:	

<p>Cargar imagen: permite cargar la imagen oblicua proveniente de la lámpara de hendidura que será analizada por el algoritmo.</p>
<p>Observaciones:</p> <ul style="list-style-type: none"> • Cargar imagen: la imagen debe ser en retroiluminación.

Tabla 2 HU Aplicar técnica de realce en la imagen

Historia de Usuario	
Número: 2	Nombre de la Historia de Usuario: Aplicar técnica de realce en la imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 1 semanas
Riesgo en desarrollo: Alto	Puntos reales: 1 semanas
Programador responsable: Dayana Díaz Torres	
Descripción:	
Aplicar técnica de realce en la imagen: a la copia de la imagen se le aplican las técnicas de realce.	
Observaciones:	
Aplicar técnica de realce en la imagen: la imagen debe ser en retroiluminación.	

Tabla 3 HU Aplicar técnicas de filtrado y mejora en la imagen

Historia de Usuario

Número: 3	Nombre de la Historia de Usuario: Aplicar técnica de filtrado y mejora en la imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Dayana Díaz Torres	
Descripción: Aplicar técnica de filtrado y mejora en la imagen: a la copia de la imagen se le aplican las técnicas de filtrado y mejora.	
Observaciones: Aplicar técnica de filtrado y mejora en la imagen: la imagen debe ser en retroiluminación.	

Tabla 4 HU Identificar los bordes de la imagen

Historia de Usuario	
Número: 4	Nombre de la Historia de Usuario: Identificar los bordes de la imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas

Programador responsable: Dayana Díaz Torres
Descripción: Identificar los bordes de la imagen: luego de aplicadas las técnicas de realce filtrado y mejora se identifican los bordes.
Observaciones: Identificar los bordes de la imagen: La imagen debe estar binarizada

Tabla 5 HU Identificar la capsulorrexis mediante el algoritmo Radon

Historia de Usuario	
Número: 5	Nombre de la Historia de Usuario: Identificar la capsulorrexis mediante el algoritmo Radon
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Dayana Díaz Torres	
Descripción: Identificar la capsulorrexis mediante el algoritmo Radon: a la imagen con los bordes identificados se le identifica la capsulorrexis aplicando el algoritmo Radon.	
Observaciones:	

Identificar la capsulorrexis mediante el algoritmo Radon: La imagen debe tener identificado los bordes

Tabla 6 HU Identificar la capsulorrexis mediante el algoritmo Hough

Historia de Usuario	
Número: 6	Nombre de la Historia de Usuario: Identificar la capsulorrexis mediante el algoritmo Hough
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	Iteración asignada: 1
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Programador responsable: Dayana Díaz Torres	
Descripción:	
Identificar la capsulorrexis mediante el algoritmo Hough: a la imagen con los bordes identificados se le identifica la capsulorrexis aplicando el algoritmo de Hough.	
Observaciones:	
Identificar la capsulorrexis mediante el algoritmo Hough: La imagen debe tener identificado los bordes	

2.1.1.3 Estimación de esfuerzo por historias de usuario

En la siguiente tabla se muestra la estimación de esfuerzos por HU definidas para el desarrollo del algoritmo propuesto:

Tabla 3. Estimación de esfuerzo por HU

Historias de usuario	Puntos de estimación(semanas)
----------------------	-------------------------------

Cargar imagen.	1
Aplicar técnica de realce en la imagen	1
Aplicar técnicas de filtrado y mejora en la imagen.	2
Identificar los bordes de la imagen	2
Identificar la capsulorrexis mediante el algoritmo Radon	2
Identificar la capsulorrexis mediante el algoritmo Hough	2

2.1.1.4 Plan de entregas

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	1	3	2
Fecha de entrega	24/03/2018	28/04/2018	26/05/2018

2.1.1.5 Plan de iteraciones

A continuación, se describen las iteraciones propuestas:

Iteración #1: en la iteración se implementó la HU1, la que se refiere a cargar la imagen.

Iteración #2: en esta iteración se implementaron las HU2, HU3 y 4, las cuales se refieren a aplicar técnicas de realce, filtrado y mejora de la imagen y la identificación de los bordes.

Iteración #3: en esta iteración se implementaron las HU5 y 6, las cuales se refieren a Identificar la capsulorrexis mediante el algoritmo de Radon y Hough.

2.1.1.5.1 Plan de duración de las iteraciones

Para mostrar la duración y el orden en que serán implementadas las HU es necesario crear un plan de duración para cada una de las iteraciones. Para el algoritmo propuesto se definieron 3 iteraciones que fueron desarrolladas en 8 semanas como se muestra en la tabla siguiente:

Tabla 4. Plan de duración de las iteraciones

Iteraciones	Historias de usuario	Duración
1	Cargar imagen	1 semana
2	Aplicar técnica de realce en la imagen	3 semanas
	Aplicar técnicas de filtrado y mejora en la imagen	
	Identificar los bordes de la imagen	
3	Identificar la capsulorrexix mediante el algoritmo Radon	4 semanas
	Identificar la capsulorrexix mediante el algoritmo Hough	

2.1.2 Diseño

La metodología de desarrollo de software XP propone que al concluir de cada iteración un producto funcional que debe ser examinado y mostrado al cliente. A la hora de darle cumplimiento a la actividad de diseñar, XP no especifica ninguna técnica de modelado, o sea, pueden utilizarse sencillos esquemas en pizarras, diagramas de clases usando UML o tarjetas CRC.

2.1.2.1 Tarjetas CRC (Clase, Responsabilidad y Colaboración)

Las tarjetas CRC son una técnica para el diseño de software orientado a objetos. Se utilizan para representar objetos. La clase del objeto puede ser escrita en la parte superior de la tarjeta, las responsabilidades se colocan en la parte izquierda y las clases que colaboran son listadas a la derecha de cada responsabilidad (Zambrano, 2014). A continuación, se detalla la tarjeta CRC de la clase principal del algoritmo llamada *Círculo*.

Tabla 5 Tarjeta CRC de la clase *Círculo*

Tarjeta CRC	
Clase: <i>Círculo</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • En esta clase se crean los círculos 	<ul style="list-style-type: none"> • LinkedList

2.1.2.2 Estándares de codificación

El estándar de codificación es un conjunto no formal de reglas establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código que brinda la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes. A continuación, se muestra el estándar de codificación definido por el equipo de desarrollo para el algoritmo propuesto.

- Todas las nomenclaturas a utilizar se definirán en idioma español.
- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma (*ListaEstatica.java*). Ver figura 30.


```
public class ListaEstatica {
```

Figura 19 Declaración de la clase *ListaEstatica*

- Los nombres de los métodos serán con minúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula (*incrementarVoto*). Ver figura 31.

```
public static void incrementarVoto
```

Figura 20 Declaración del método *incrementarVoto*

- Los identificadores para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula (*imageAlgoritmo*). Ver figura 32.

```
public static BufferedImage imageAlgoritmo;
```

Figura 21 Declaración de la variable *imageAlgoritmo*

- Los nombres de variables o funciones deben ser lo suficientemente descriptivos, sin exceder de 30 caracteres (*cantBlancos*). Ver figura 33.

```
int cantBlancos = 1000000;
```

Figura 22 Declaración del método *cantBlancos*

- Todas las funciones deben tener comentarios explicando que realiza cada una de ellas. Ver figura 34.

```

//metodo para pintar el circulo de la capsulorrexis identificado por radon
public static BufferedImage identificarCapsulorrexis(BufferedImage imageActual, BufferedImage retornar) {...18 lines }

//Deteccion de bordes
public BufferedImage detencionBordes(BufferedImage imageActual) {...26 lines }

```

Figura 23 Métodos *identificarCapsulorrexis* y *detencionBordes* con la explicación de su uso en el algoritmo propuesto

2.1.2.3 Patrones de diseño

Los patrones de diseño que se emplearon en la presente investigación son los siguientes:

Patrones Creacionales

- **Singleton:** Restringe la instanciación de una clase o valor de un tipo, a un solo objeto. Este patrón se utiliza en la clase *ListaEstatica*, adicionándole el comando *static* a la variable *hm*, evitando que sea instanciado otras veces. La siguiente imagen muestra la declaración de la variable.

```

static ConcurrentHashMap<String, Circulo> hm =new ConcurrentHashMap<String, Circulo>();

```

Figura 24 Variable estática *hm* de la clase *ListaEstatica*

Patrones de comportamiento:

- **Observador (Observer):** Notificaciones de cambios de estado de un objeto. Se evidencia en la clase *Segmentacion*, en el método *filtroAdaptado*, en el cual la variable *subiendo* se encarga de notificar cuando está descendiendo o ascendiendo la herradura de caballo en el histograma de la imagen. En la figura 31 se evidencia el uso de este patrón.

```

if (array[i].isSubiendo() == subiendo)

```

Figura 25 Variable *subiendo* del método *filtroAdaptado* de la clase *Segmentacion*

- **Creador:** se refleja en las clases que tienen la responsabilidad de instanciar objetos de otras clases. El uso de este patrón se evidencia en la clase *Segmentacion*, que es la clase principal encargada de crear objetos de otras clases, la siguiente imagen muestra un fragmento de la clase antes mencionada.

```
public class Segmentacion {  
    private BufferedImage imageBordes;  
    public static BufferedImage imageAlgoritmo;  
  
    public BufferedImage algoritmoRadon(BufferedImage imageActual) {...47 lines }  
  
    public BufferedImage algoritmoHough(BufferedImage imageActual) {...39 lines }
```

Figura 26 Fragmento de la clase *Segmentacion*

2.2 Propuesta de solución

La presente investigación propone un algoritmo que permita identificar la capsulorrexis en imágenes oblicuas provenientes de la lámpara de hendidura mediante el algoritmo de Radon. Antes de aplicar dicho algoritmo se le deben aplicar varios filtros a la imagen para eliminar el ruido que puedas tener e identificar bordes y así garantizar una correcta identificación de la capsulorrexis. El algoritmo propuesto consta de los siguientes pasos:

1. Cargar la imagen oblicua.
2. Cambiar tamaño de la imagen.
3. Aplicar técnicas de filtrado.
4. Aplicar algoritmo iterativo de detección de bordes.
5. Aplicar transformada de Radon.

A continuación, se detalla lo que se realiza en cada una de las etapas

1. Capturar la imagen oblicua

Para comenzar la ejecución del algoritmo se debe contar con una imagen oblicua proveniente de oblicua proveniente de la lámpara de hendidura, sus dimensiones deben ser de 1024*768 pixeles. El algoritmo permite capturar esa imagen a la que le realiza una copia, con la que seguirá trabajando.

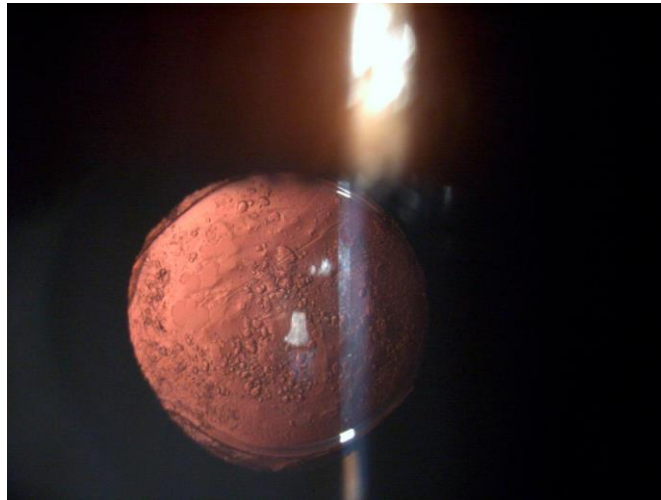


Figura 27 Imagen oblicua proveniente de la lámpara de hendidura

2. Cambiar tamaño de la imagen

Una vez realizada la copia de la imagen, automáticamente cambiará las dimensiones de la misma a 1024*1024 píxeles.

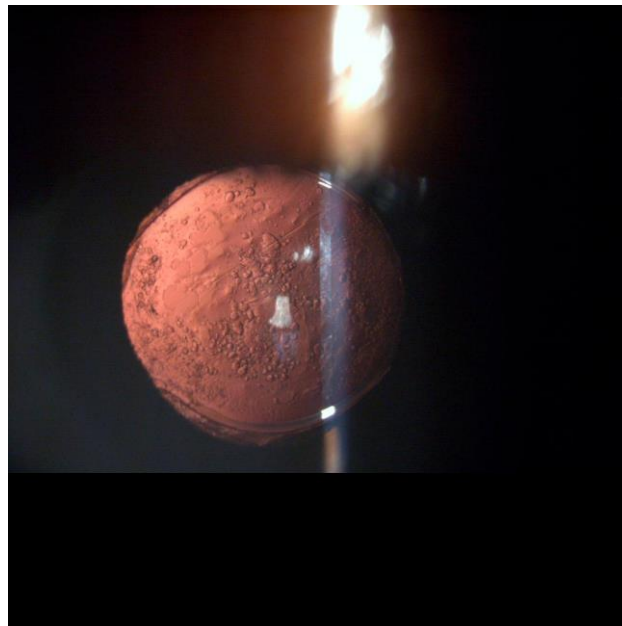


Figura 28 Imagen con las nuevas dimensiones(1024*1024)

3. Aplicar técnicas de filtrado

El siguiente paso del algoritmo es aplicarle a la nueva imagen los siguientes filtros: escala de grises, binarización y el filtro de la moda. Para aplicar escala de grises la imagen es llevada a escala de grises sin perder los cambios en el contraste. A continuación, se muestra la imagen resultante al aplicarle el filtro.

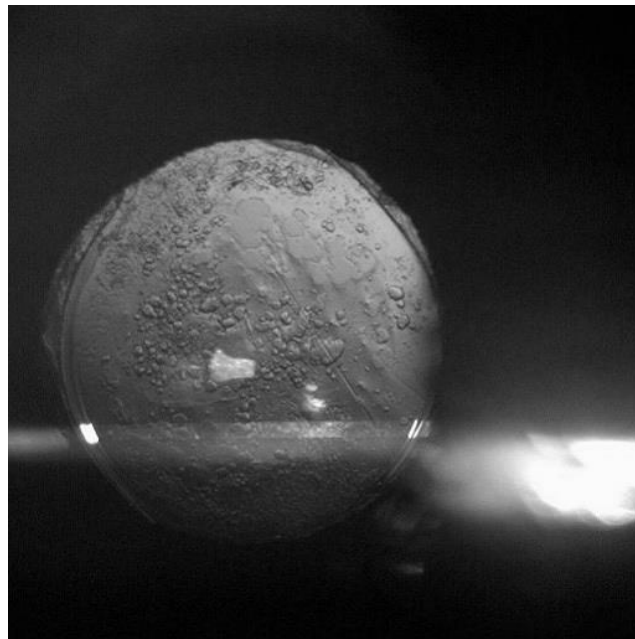


Figura 29 Imagen después de aplicarle escala de grises

Seguidamente se aplica binarización, este filtro recorre la imagen llevada a escala de grises y selecciona un umbral automático por cada región de la misma, tomando los valores que sean mayores que el umbral en blanco y los que sean menores en negro. A continuación, se muestra la imagen resultante.

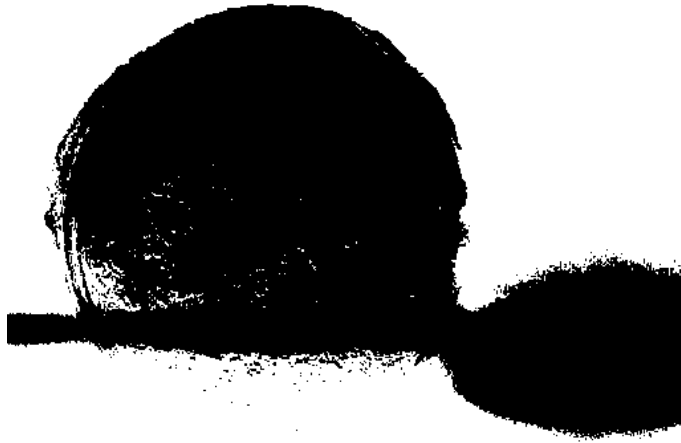


Figura 30 Imagen después de aplicarle la binarización

Luego se aplica el filtro de la moda, en el cual se recorre la imagen binarizada y se elimina el ruido presente en la misma, asignándole el color a un pixel dependiendo del color más frecuente en sus vecinos. A continuación, se muestra la imagen obtenida.



Figura 31 Imagen después de aplicarle el filtro de la moda

4. Aplicar algoritmo de detección de bordes iterativo

El siguiente paso del algoritmo propuesto es aplicar algoritmo de borde iterativo, el mismo recorre la imagen libre de ruido y en cada cambio de color blanco al negro o de negro al blanco se toma como pixel de borde. A continuación, se muestra la imagen que se obtuvo al aplicar el algoritmo.



Figura 32 Imagen después de aplicarle el algoritmo de detección de bordes iterativo

5. Aplicar transformada de Radon

Una vez aplicadas a la imagen las técnicas de filtrado y el algoritmo de detección de bordes ya está lista para aplicarle la transformada de Radon. El cual toma como entrada una imagen con las dimensiones 1024×1024 y con los bordes bien detectados, y devuelve una imagen de 1024×180 donde las columnas van a ser los ángulos (0-180) y en las casillas van a estar los valores de las integrales de línea para cada ángulo dado. Esta imagen es llamada sinograma.



Figura 33 Sinograma de la imagen

Posteriormente se aplica un filtro adaptado para eliminar del sinograma todas las regiones que no conformen la herradura de caballo, ya que la misma delimita un círculo en la imagen original.



Figura 34 Sinograma después de aplicarle un filtro adaptado

Por último, se reconstruye la imagen con el sinograma resultante, quedando una propuesta de círculo en esta imagen reconstruida.

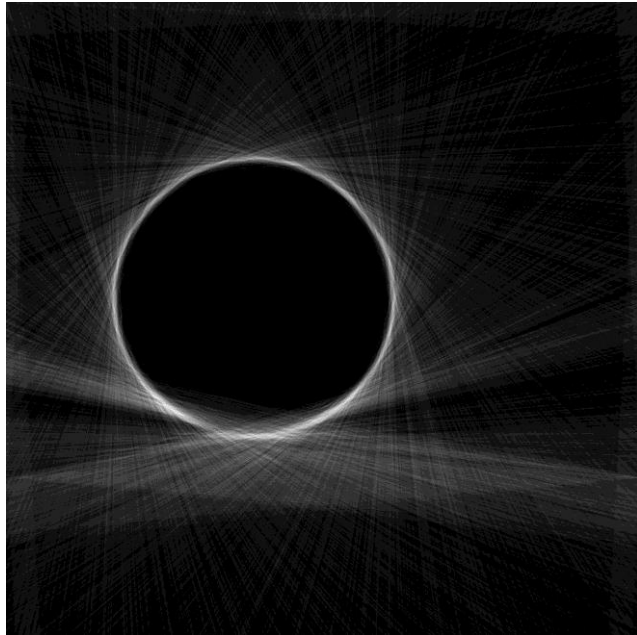


Figura 35 Imagen reconstruida

Este círculo es marcado en la imagen de entrada identificando así la capsulorrexis de ojo humano.

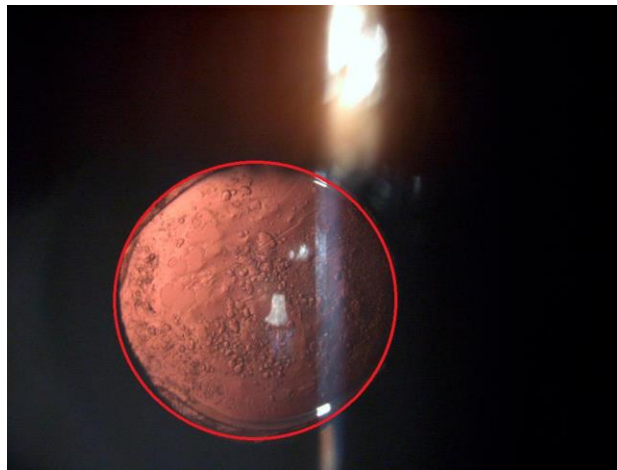


Figura 36 Imagen resultante

2.3 Conclusiones parciales

A continuación, se detallan las conclusiones obtenidas hasta este momento en la investigación:

- Se utilizó la metodología XP durante el proceso de desarrollo lo que garantizó un trabajo organizado y estructurado, generando los artefactos de cada una de sus fases.
- Se identificaron las tarjetas CRC necesarias para la implementación de funcionalidades, las cuales fueron desarrolladas haciendo uso de los patrones de diseño.

CAPÍTULO 3: EVALUACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describen la fase de implementación y prueba de la metodología de desarrollo empleada. Además, se exponen los artefactos que se obtuvieron en cada una de ellas, tales como: las tareas ingenieriles por cada historia de usuario identificada y las pruebas empleadas para validar la solución obtenida.

3.1 Implementación

Dentro del proceso de desarrollo de la metodología del desarrollo del software la parte más importante es la implementación. En esta fase se descomponen las HU en tareas de programación o ingeniería, que a su vez son convertidas en código.

Una vez identificadas las HU, los programadores proceden a descomponer cada una en tareas específicas, las denominadas tareas de la programación que están escritas técnicamente y que darán solución a la HU correspondiente. Las tareas de ingeniería permiten a los desarrolladores obtener un nivel de detalle más avanzado por las HU. A continuación, se describe las Tareas de ingeniería 4 y 5, que tienen como objetivo implementar las HU Segmentar imagen e integrar la solución al software PANDOC respectivamente:

Tabla 6 Tarea de ingeniería 4

Tarea de ingeniería	
Número de tarea: 4	Historia de Usuario (No.4): Identificar la capsulorrexis mediante el algoritmo Radon
Nombre de tarea: Implementar HU_ Identificar la capsulorrexis mediante el algoritmo Radon	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semanas
Fecha de inicio: 30/04/2018	Fecha de fin: 11/05/2018
Programador responsable: Dayana Torres	
Descripción: permite identificar la capsulorrexis mediante el algoritmo de Radon en imágenes oblicuas provenientes de la lámpara de hendidura	

Tabla 7 Tarea de ingeniería 5

Tarea de ingeniería	
Número de tarea: 5	Historia de Usuario (No.4): Identificar la capsulorrexis mediante el algoritmo Hough
Nombre de tarea: Implementar HU_ Identificar la capsulorrexis mediante el algoritmo Hough	
Tipo de tarea: Desarrollo	Puntos estimados: 2 semanas
Fecha de inicio: 14/05/2018	Fecha de fin: 25/05/2018
Programador responsable: Dayana Torres	
Descripción: permite identificar la capsulorrexis mediante el algoritmo de Hough en imágenes oblicuas provenientes de la lámpara de hendidura	

3.2 Técnica de validación de los requisitos

Se aplicaron revisiones formales de los requisitos para demostrar que los requerimientos previamente definidos cumplen con las expectativas del cliente. Se realizaron revisiones formales a cada requisito por parte del cliente y del equipo de desarrollo, obteniéndose un total de cuatro no conformidades de tipo redacción, formato, y técnica; las que fueron corregidas en tiempo, generándose por parte del cliente un Acta de Aceptación.

3.3 Pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas. 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: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas

funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (Gutiérrez, y otros, 2006).

3.3.1 Pruebas unitarias

La prueba de unidad es la primera fase de las pruebas dinámicas y se realizan sobre cada módulo del software de manera independiente. El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y lo hará, generalmente, la persona que lo creó. En general, un módulo se entiende como un componente software que cumple las siguientes características:

- Debe ser un bloque básico de construcción de programas.
- Debe implementar una función independiente simple.
- Podrá ser probado al cien por cien por separado.
- No deberá tener más de 500 líneas de código.

Tanto las pruebas de caja blanca como las de caja negra han de aplicarse para probar de la manera más completa posible un módulo (LSI, 2017).

El método de caja blanca garantiza que se ejerciten por lo menos una vez todos los caminos independientes del código, así como la ejecución de todos los bucles en sus límites operacionales y todas las decisiones lógicas en las vertientes verdaderas y falsas (Pressman, 2003). Para aplicar las mismas se empleó la técnica del camino básico. Para explicar cómo se aplicó el método se tomó como ejemplo el método *escalaGrises* de la clase *Segmentación*. Esto permitió obtener una medida de la complejidad lógica para el diseño de los casos de prueba (CP) y usar dicha medida como guía para la definición de un conjunto básico de caminos de ejecución, en la figura siguiente se muestra el código.

```

//escala de grises
public BufferedImage escalaGrisés(BufferedImage imageActual) {
    //Variables que almacenarán los píxeles
    System.out.println("Escala de grises");
    int mediaPixel, colorSRGB;
    Color colorAux;

    BufferedImage copiaImg = new BufferedImage(imageActual.getWidth(), imageActual.getHeight(), imageActual.getType());
    copiaImg.setData(imageActual.getData());

    //Recorremos la imagen pixel a pixel
    for (int i = 0; i < copiaImg.getWidth(); i++) {
        for (int j = 0; j < copiaImg.getHeight(); j++) {
            colorAux = new Color(copiaImg.getRGB(i, j));

            mediaPixel = (int) ((colorAux.getRed() + colorAux.getGreen() + colorAux.getBlue()) / 3);

            colorSRGB = (mediaPixel << 16) | (mediaPixel << 8) | mediaPixel;

            copiaImg.setRGB(i, j, colorSRGB);
        }
    }

    return copiaImg;
}

```

Figura 37 Método escalaGrisés de la clase Segmentación

A continuación, se describen los pasos que se realizaron para desarrollar la técnica del camino básico.

1. **Confeccionar el grafo de flujo:** usando el código mostrado anteriormente se realizó la representación del grafo de flujo, el cual está compuesto por los siguientes elementos:
 - Nodos: son círculos que representan una o más sentencias procedimentales.
 - Aristas: son flechas que representan el flujo de control y son análogas a las flechas del diagrama de flujo.
 - Regiones: son las áreas delimitadas por aristas y nodos

En la figura 41 se muestra el grafo de flujo obtenido.

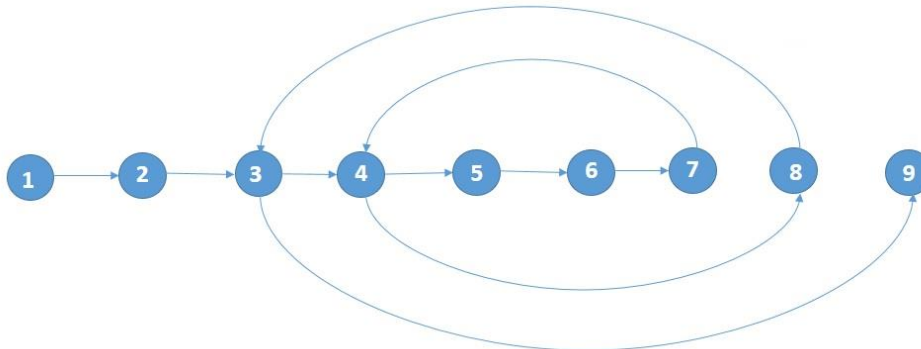


Figura 38 Grafo del camino básico del método *escalaGrises*.

2. **Calcular la complejidad ciclomática:** proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa, la complejidad ciclomática se calculó con la siguiente fórmula:

- $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.

$$V(G) = A - N + 2 = 10 - 9 + 2 = 3$$

3. **Determinar un conjunto básico de caminos linealmente independientes:** el valor de $V(G)$ da el número de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 3 caminos:

Camino básico #1: 1-2-3-4-5-6-7-4-8-3-9

Camino básico #2: 1-2-3-4-8-3-9

Camino básico #3: 1-2-3-9

Obtención de casos de pruebas (CP): cada camino independiente es un caso de prueba a realizar, de forma que los datos introducidos provoquen que se visiten las sentencias vinculadas a cada nodo del camino. En este caso se obtuvieron 3 caminos básicos, por tanto, se hace necesario la confección de igual número de CP, para aplicar las pruebas a este método. A continuación, se muestra uno de ellos:

Tabla 8 Caso de prueba de caja blanca para el camino básico #3.

Caso de prueba: Camino básico #3	
Entrada	Una imagen que será llevada a escala de grises.
Resultados esperados	Una imagen en escala de grises.
Condiciones	Debe haber una imagen en retroiluminación proveniente de la lámpara de hendidura

Los resultados obtenidos una vez ejecutados todos los casos de pruebas fueron satisfactorios demostrando que el código generado no presenta ciclos infinitos y no existe código innecesario en el algoritmo desarrollado.

3.4 Análisis estadístico de los resultados

Luego de realizar las pruebas para comprobar que el algoritmo cumple con las necesidades del cliente, se procedió a realizar la comparación de las métricas de error y los tiempos de ejecución entre el algoritmo propuesto y la transformada de Hough, con el objetivo de seleccionar el algoritmo más eficiente para la detección de la opacidad. Para realizar las comparaciones se escogieron de 10 imágenes de pacientes con esta complicación postoperatoria que fueron ejecutadas por cada uno de estos algoritmos.

Se confecciono un equipo de trabajo compuesto por doctores del hospital Pando Ferrer, los que se reunieron y en consenso marcaron en las imágenes las regiones que ellos consideraron opacidad.

3.4.1 Pruebas estadísticas

Una prueba estadística es una forma de evaluar la evidencia que los datos proporcionan para probar una hipótesis. Esta hipótesis se denomina hipótesis nula, y suele denominarse H_0 . Bajo H_0 , los datos se generan mediante procesos aleatorios. En otras palabras, los procesos controlados (las manipulaciones experimentales, por ejemplo) no afectan a los datos. Normalmente, H_0 establece la igualdad (entre las

medias, o entre las varianzas, o entre un coeficiente de correlación y cero, por ejemplo). H_0 normalmente se opone a una hipótesis denominada hipótesis alternativa, denominada H_1 o H_a . La mayoría de las veces, la hipótesis alternativa es aquella que el usuario querría demostrar. Implica establecer una diferencia (por ejemplo, diferencia entre medias). Si los datos no proporcionan suficiente evidencia contra H_0 , H_0 no se rechaza. Si, por el contrario, muestran una fuerte evidencia contra H_0 , H_0 se rechaza, y H_a se considera verdadera con un riesgo cuantificado (bajo) de ser errónea. Una prueba estadística permite rechazar / no rechazar H_0 (XLSTAT, 2017).

Para comparar el desempeño de los algoritmos se procedió a realizar un análisis estadístico. Para ellos se seleccionó una prueba no paramétrica teniendo en cuenta el tamaño de la muestra (10 imágenes).

Se escoge usar el análisis de dos vías de Friedman y como prueba post-hoc la prueba de rango de Wilcoxon por pares, ajustándose el nivel de significancia con una corrección de Bonferroni. Esto se aplicó sobre la variable Sokal-Sneath como medida estadística del error cometido en la segmentación, sobre la variable Tasa de Correctos Equilibrada como medida de localización y sobre Modificación de la Distancia de Hausdorff Normalizada como una variante de las métricas de Hausdorff. Para realizar todos los cálculos se empleó el software IBM SPSS Statistics 21 (Elliott, y otros, 2007).

Tasa de Correctos Equilibrada esta tomará valor 1 cuando lo segmentado por el algoritmo coincida completamente con la OCP marcada por los especialistas. En la Figura 46 se pueden apreciar los resultados de esta medida sobre los algoritmos analizados. En el caso del algoritmo desarrollado, partir de este momento Radon, tiene una marcada superioridad en esta medida marcando una media de 0.6 y desviación 0.08 a diferencia de Hough que solo alcanzan un valor medio de 0.245.

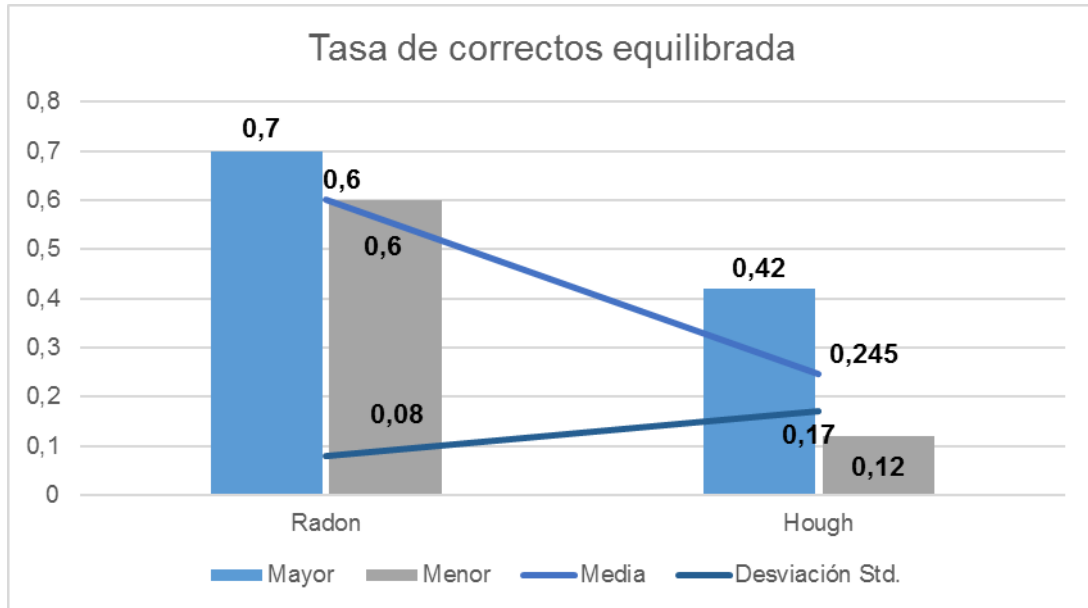


Figura 39 Resultados de la métrica Tasa de correctos equilibrada

Índice de Sokal-Sneath es una de las medidas estadística del error tomada en cuenta. Esta penaliza las regiones que fueron mal identificadas. Por lo que cuando toma valor 1, significa que todas las regiones con opacidad fueron correctamente segmentadas. En la Figura 47 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos. En el caso de Radon tiene una marcada superioridad en esta medida marcando una media de 0.51 a diferencia de Markov que solo alcanzan un valor medio de 0.265.

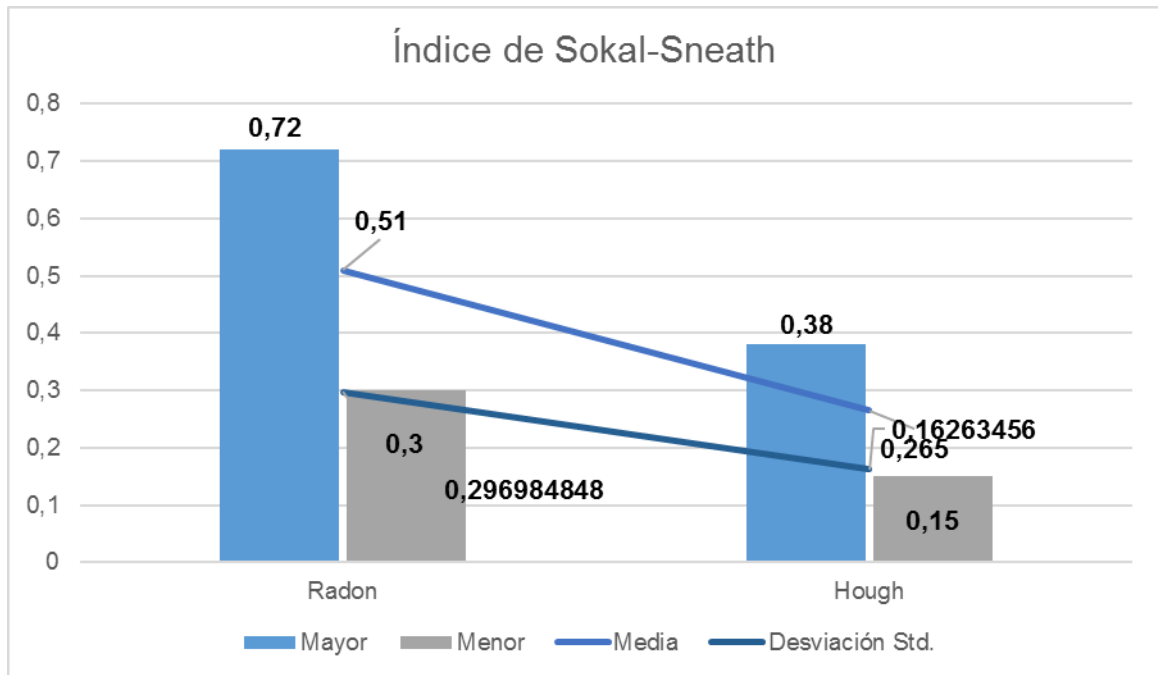


Figura 40 Resultados de la métrica Índice de Sokal-Sneath

Modificación de la Distancia de Hausdorff Normalizada esta métrica devuelve valores de distancia por las diferencias existentes entre la segmentación de los algoritmos y el *Ground Truth* significando esto que el valor sea 0 en caso de una segmentación perfecta, y mientras mayor sea el valor de la misma, mayor la cantidad de diferencias encontradas.

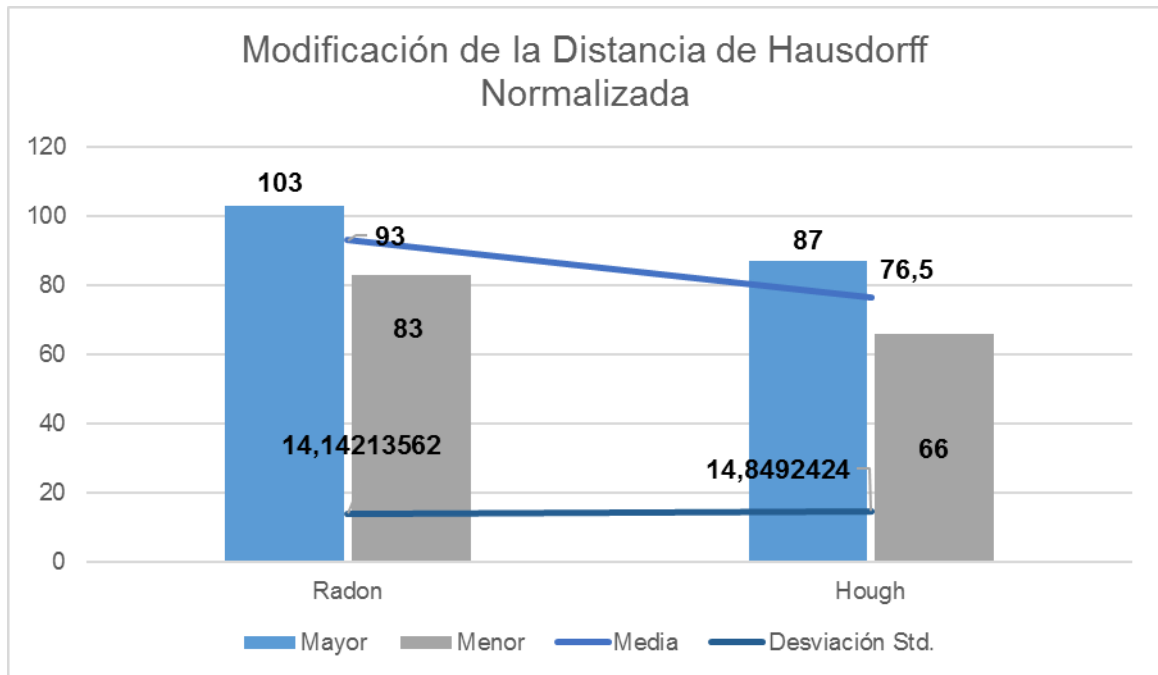


Figura 41 Resultados de la métrica Modificación de Distancia de Hausdorff

Se analizaron las distintas estadísticas del error (Tasa de Correctos Equilibrada, Índice de Sokal-Sneath y Modificación de la Distancia de Hausdorff Normalizada) donde Radon arrojó mejores resultados como se aprecia en la tabla 13, donde se muestran los resultados medios por cada una de las métricas calculadas y la desviación típica.

Tabla 9 Valores medios y desviaciones por cada métrica

Métrica	Mejor Resultado	Radon		Hough	
		Media	Desv.	Media	Desv.
Tasa de Correctos Equilibrada	1	0.6	0.08	0.245	0.17
Índice de Sokal-Sneath	1	0.51	0.296	0.265	0.1626
Modificación de la Distancia de Hausdorff Normalizada	0	93	14.142	76.5	14.849

3.4.2 Análisis estadístico

Se realiza un análisis estadístico con el objetivo de comparar el desempeño de los algoritmos y de corroborar los resultados anteriormente expuestos. Para esto se seleccionan pruebas no paramétricas debido al tamaño de la muestra (solo 10 imágenes). Se escoge usar el análisis de dos vías de Friedman y como prueba *post-hoc* la prueba de rango de Wilcoxon por pares, ajustándose el nivel de significancia con una corrección de Bonferroni. Esto se aplicó sobre la variable Tasa de correctos equilibrado como medida estadística del error cometido en la segmentación, sobre la variable índice de Sokal-Sneath como medida de localización y sobre Modificación de la Distancia de Hausdorff Normalizada como una variante de las métricas de Hausdorff. Para realizar todos los cálculos se empleó el software IBM SPSS Statistics 21.

Test de suma de rangos de Wilcoxon

Se tiene una m.a.s de tamaño n_1 de una población, y una segunda m.a.s de tamaño n_2 de otra población. Hay n observaciones en total, donde $n = n_1 + n_2$. Se calcula el rango de las n observaciones. El test estadístico será la suma W de los rangos del grupo con menor suma de rangos, este será el estadístico de suma de rangos de Wilcoxon. Si las dos poblaciones tienen la misma distribución continua, entonces W tiene media:

$$\mu_W = \frac{(n_1(n+1))}{2}$$

Figura 42 Media

Y desviación estándar:

$$\sigma_W = \sqrt{\frac{n_1 n_2 (n+1)}{12}}$$

Figura 43 Desviación estándar

Donde n_1 será el tamaño muestral del grupo con menor suma de rangos.

El test de suma de rangos de Wilcoxon rechaza la hipótesis nula de que las dos poblaciones tienen la misma distribución cuando la suma de rangos W está lejos de su media (Martinez, 2011).

Hipótesis para una prueba de signo o una prueba de rango con signo de Wilcoxon

Las hipótesis que se prueban con la prueba de signo y la prueba de rango con signo de Wilcoxon son las siguientes:

H₀: La probabilidad de una diferencia positiva es igual a la probabilidad de una diferencia negativa.

H_a: La probabilidad de una diferencia positiva no es igual a la probabilidad de una diferencia negativa.

Test de Friedman

La prueba de Friedman es una alternativa no paramétrica a un análisis de varianza de medidas repetidas utilizado para comparar observaciones repetidas sobre los mismos sujetos. A diferencia de las medidas paramétricas ANOVA (Elliott, y otros, 2007). Además, si la investigación tiene k variables en columnas y n elementos en filas se trata de ordenar cada fila de menor a mayor según las diferentes columnas de 1 hasta k (esto es el rango que ocupa cada variable para ese caso). Si no hay diferencias estadísticamente significativas entre las variables se espera que los rangos estén repartidos en cada columna de manera uniforme y sólo se encontrarán entre las variables pequeñas diferencias debidas al azar.

Hipótesis para la prueba de Friedman

Las hipótesis que se prueban en la prueba de Friedman son las siguientes:

H₀: Las distribuciones son las mismas a través de medidas repetidas.

H_a: Las distribuciones a través de medidas repetidas son diferentes.

3.4.2.1 Resultados

En la Tabla X se muestra el valor de precisión (p) es menor que 0.05, por lo que se puede confirmar que existen diferencias significativas (en la Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal - Sneath) entre los grupos de algoritmos analizados. Es decir, los resultados de los algoritmos en las variables analizadas cambiaron significativamente.

Tabla 10 Test de Friedman sobre los resultados de las variables Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal – Sneath.

No	Hipótesis nula	Test	Sig.	Decisión	Variables
1	Las distribuciones Radon y Hough son la misma.	Análisis de dos vías de Friedman de varianza por rangos de	.005	Rechazar la hipótesis nula.	Tasa de correctos equilibrado
2	Las distribuciones Radon y Hough son la misma.		.004	Rechazar la hipótesis nula.	Modificación de la Distancia de

		muestras relacionadas.			Hausdorff Normalizada
3	Las distribuciones Radon y Hough son la misma.		.003	Rechazar la hipótesis nula.	Sokal-Sneath

Se desarrolló una prueba post-hoc de rangos con signos de Wilcoxon con el objetivo de comparar dos a dos las muestras relacionadas (algoritmos de segmentación) y determinar entre qué resultados se presentaron las diferencias significativas. Se usó la corrección de Bonferroni para evitar cometer un error de Tipo I (rechazar la hipótesis nula cuando debería aceptarse). El nuevo nivel de significación que se tomó por tanto fue de 0,016 ($0.05/3=0.0166$).

Tabla 11 Resultados de las pruebas

No	Hipótesis nula	Test	Sig.	Decisión	Variables
1	La mediana de las diferencias entre Radon y Hough es igual a 0.	Prueba de Wilcoxon de los rangos con signo de muestras relacionales.	.006	Rechazar la hipótesis nula.	Tasa de correctos equilibrado
2	La mediana de las diferencias entre Radon y Hough es igual a 0.		.010	Rechazar la hipótesis nula.	Modificación de la Distancia de Hausdorff Normalizada
3	La mediana de las diferencias entre Radon y Hough es igual a 0..		.007	Rechazar la hipótesis nula.	Sokal-Sneath

En la tabla 15 se observa que existen diferencias significativas entre los algoritmos en las variables analizadas (en la Tasa de correctos equilibrados, Modificación de la Distancia de Hausdorff Normalizada y Sokal - Sneath), rechazándose en todos los casos la hipótesis nula. Por lo que se asume que el algoritmo que mejores resultados brinda es Radon.

3.5 Conclusiones parciales

En este capítulo se exponen los resultados de la validación del software mediante pruebas unitarias y

- La aplicación de revisiones formales de los requisitos permitió asegurar la validez de los mismos.
- La realización de pruebas al algoritmo permitió verificar que el mismo es funcional y cumple las necesidades del cliente.

- La comparación de los resultados obtenidos mediante la Transformada de Hough y el algoritmo propuesto certifica el cumplimiento del objetivo general de la investigación.
- La utilización de la validación estadística permitió comprobar que el algoritmo desarrollado ofrece mejores resultados que la transformada de Hough.

CONCLUSIONES

1. El estudio de las técnicas utilizadas para la identificación de estructuras circulares en imágenes médicas demostró que no proporcionan una respuesta correcta al objetivo general de la presente investigación, fomentando la necesidad de desarrollar el algoritmo propuesto.
2. La utilización de la metodología XP para guiar el proceso de desarrollo, los estándares de codificación y los patrones de diseño permitió obtener un algoritmo que detecta la capsulorrexis en imágenes en retroiluminación provenientes de la lámpara de hendidura.
3. La aplicación de técnicas para validar los requisitos y la realización de pruebas unitarias y validación estadísticas, permitió verificar que el algoritmo es funcional y cumple las necesidades del cliente.

RECOMENDACIONES

Se propone expandir la investigación a la identificación de la Capsulorrexis en imágenes oblicuas.

Realizar un estudio con otros filtros morfológicos con el objetivo de ajustar más los resultados de la transformada de Radon.

Implementar la transformada de radón utilizando un paradigma de programación concurrente.

ANEXOS

Anexo 1: Acta de Aceptación de la revisión de los requisitos.

Anexo 2: Acta de aceptación

BIBLIOGRAFÍA REFERENCIADA

- Hernández López,, Dr. Iván, y otros. 2011.** *Cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas Scheimpflug del Pentacam.* 2011. Rev Cubana Oftalmol vol.24 no.2 Ciudad de la Habana jul.-dic..
- URREA, JUAN PABLO y OSPINA, EMMANUEL. 2004.** *IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN.* 2004.
- Alan C. Elliott, Wayne A. Woodward. Sage, 2007.** *Statistical analysis quick reference guidebook: With SPSS examples.* Sage, 2007. págs. 201-203.
- Berrío, J. Stephany , Ordoñez, Samith O. y Caicedo, Eduardo F. 2012.** *Extracción de Líneas a partir de Escaneos Láser integrando Transformada de Hough, Mínimos Cuadrados Totales y Seguimiento de Bordes Sucesivos.* 2012. Vol. 16 No. 2, pág. 48 - 59.
- Borja López, Yolanda. 2015.** *Metodología Ágil de Desarrollo de Software -XP.* 2015.
- Cabrera Martínez, Abel, y otros. 2007.** *Prevalencia de ceguera y limitación visual severa, en personas mayores de 50 años.* s.l. : Instituto Cubano de Oftalmología “Ramón Pando Ferrer”, 2007. Rev Cubana Oftalmol v.20 n.2 Ciudad de la Habana jul.-dic. 2007.
- Departamento de Ingeniería electrónica, Telecomunicación y Automática. 2015.** *Segmentación. Transformada de Hough.* Universidad de Jaén : s.n., 2015.
- Elizondo , M. C. José Jaime Esqueda. 2002.** *Fundamentos de Procesamiento de Imágenes. California : CONATEC 2002.* 2002.
- Fernández Marrón, J.L, Alberdi Primicia, J y Barcala Riveira, J.M. 2007.** *Desarrollo de Algoritmos de Reconstrucción de Imágenes en Tomografía de Capacitancia Eléctrica.* Madrid : CIEMAT, 2007. ISSN: 1135 - 9420.
- Fernández, Maria Cristina Casado. 2015.** *MANUAL BÁSICO DE MATLAB.* s.l. : Servicios Informáticos U.C.M Apoyo a Investigación y Docencia, 2015.
- Figuerola, Norberto. 2011.** *Pruebas de Aceptación del Usuario (UAT).* 2011.
- Gamma , Erich, et al. 1994.** *Design Patterns.Elements of Reusable Object-Oriented Software.* 1994.
- Gamma, Erich, y otros. 1994.** *Design Patterns.Elements of Reusable Object-Oriented Software.* 1994.
- García Ruiz, Jesús Manuel. 2015.** *LA TRANSFORMADA DE RADON Y SU APLICACIÓN EN LA TOMOGRAFÍA AXIAL.* 2015.

González Betancourt, Aniel. 2014. *Búsqueda de marcadores en imágenes de microscopía de eritrocitos utilizando transformada Radon.* Santa Clara : s.n., 2014.

Groussard, Thierry. 2014. *Java8 Los fundamentos del lenguaje Java(con ejercicios prácticos corregidos).* 2014.

Gutiérrez, J.J, y otros. 2006. *Pruebas del sistema en programación extrema.* Sevilla : Department de Lenguajes y Sistemas Informáticos, 2006.

Hough, P. 1962. s.l. : US Patente n° 3069654, 1962.

Jain, A.K. 1989. *Fundamentals of Digital Image Processing.* Prentice Hall : s.n., 1989. NJ, ISBN: 0-13-336165-9.

Jiménez, José. 2015. *10 Métodos Estadísticos.* 2015.

Joskowicz, José. 2008. *Reglas y prácticas en eXtreme Programming.* s.l. : Universidad de Vigo, 2008.

Juzga, Adriana y Rodriguez, Brahian. 2011. *MÉTODO DE RETROPROYECCIÓN FILTRADA.* s.l. : Matemáticos egresados Universidad Distrital Francisco José de Caldas, 2011.

La Serna Palomino, Dra Nora y Román Concha, Lic Ulises. 2009. *Técnicas de Segmentación en Procesamiento Digital de Imágenes: aceptar el cambio.* 2009.

Leiva, Antonio. 2016. *Patrones de diseño de software.* 2016.

López-Roldán, Pedro y Fachelli, Sandra. 2018. *Software para el análisis de datos : SPSS, R y SPAD.* Barcelona : Bellaterra Universitat Autònoma de Barcelona, 2018.

LSI, Departamento de Lenguajes y Sistemas Informáticos. 2017. *TÉCNICAS DE EVALUACIÓN DINÁMICA.* Sevilla : s.n., 2017.

M.v, Ginkel, C.L.L, Hendriks y L.J.v, Vilet. 2004. *A short introduction to the Radon and Hough transforms and how they relate to each other. .* s.l. : Quantitative Imaging Group Technical Report Series,, 2004.

Martin, Marcos. 2008. *Contornos Activos.* 2008.

Martínez de Dios, Ramiro. 2016. *Técnicas de segmentación.* 2016.

Martinez, Elena J. 2011. *Métodos No Paramétricos I .* 2011.

MathWorks. 2018. *Imagen Processing Toolbox.* 2018.

Paula Vidal, Erlis. 2016. *Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM.* 2016.

Pressman, Roger S. 2002. *Ingeniería del Software. Un Enfoque Práctico. Quinta Edición.* España : McGraw-Hill, 2002.

Pressman, Roger S. 2003. *Ingeniería del Software. Un enfoque práctico.*Sexta Edición. s.l. : Mc Graw Hill, 2003. 970-10-5473-3.

Sánchez Mendoza, Maria A. 2004. *Metodologías del desarrollo de software.* 2004.

Sanchez, A. 2014. *Entorno Desarrollo Integrado (IDE).* 2014.

Tetz, Manfred R, y otros. 1997. *Photographic image analysis system of posterior capsule opacification.* 1997. Journal of Cataract & Refractive Surgery, Vol. 23..

XLSTAT. 2017. *¿Qué es una prueba estadística?* s.l. : Centro de soporte XLSTAT, 2017.

Zambrano. 2014. *¿Qué son las tarjetas CRC?*