Universidad de las Ciencias Informáticas
Facultad 3

Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM.

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

Autor:

Erlis Paula Vidal

Tutor:

Ing. Michel Álvarez Cancio

Cotutor:

Lic. Reyder Cruz de la Osa

Junio, 2016

"Año 58 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____

Erlis Paula Vidal

_____ Ing. Michel Álvarez Cancio

_____ Lic. Reyder Cruz de la Osa

AGRADECIMIENTOS

A toda mi familia, es la mejor que existe. Sin todo el apoyo brindado no estuviera donde estoy.

A la revoluci ó n por darme la opci ó n de estudiar, a mi novia, a mis amigos, al gran colectivo de profesores y especialistas que tiene la facultad por brindarme su ayuda en todo momento.

DEDICATORIA

A mi mamá y mi familia por haberme apoyado siempre, por haber sido tan exigentes conmigo en mis estudios, por su educación, sus consejos y por haber confiado siempre en mí.

RESUMEN

En la actualidad la opacidad de la cápsula posterior es la complicación postoperatoria más frecuente después que un paciente es operado de catarata. Está asociada con la disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento, provocando problemas sociales, médicos y económicos. El software PANDOC provee al oftalmólogo de una herramienta para detectar la opacidad (a veces imperceptibles para el ojo humano), minimizando el error de observación entre un médico y otro. En la actualidad este software no detecta correctamente la OCP, y el tiempo de respuesta es elevado. El presente trabajo se ha realizado con el propósito de desarrollar un algoritmo de procesamiento digital de imágenes, que permita mejorar la efectividad y el tiempo de respuesta del PANDOC. La solución 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: algoritmo, opacidad de la cápsula posterior, PANDOC, procesamiento, realce y mejora, segmentación.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Muchas personas en la actualidad padecen de la enfermedad de cataratas, sufriendo como consecuencia la pérdida total o parcial de la visión. Esta enfermedad es opacidad que se crea en el cristalino del ojo, provocando que la luz se disperse y no se enfoque en la retina, creando una serie de imágenes difusas. La catarata es uno de los ejemplos más comunes de ceguera tratables con cirugía y aunque tiene varias causas de origen, se les atribuye mayormente el padecimiento de esta enfermedad a personas que sufren diabetes, hipertensión o mayores de 50 años. El paciente después de una intervención quirúrgica puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito a largo plazo, pues en muchos de ellos el paciente puede presentar, a mediano o largo plazo, complicaciones posoperatorias como la Opacidad de la Cápsula Posterior (OCP) (22).

La opacidad de la cápsula posterior (OCP) es uno de los aspectos más importantes en la cirugía de catarata en la actualidad. 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. (19). Un oftalmologo para diagnosticar esta enfermedad, realiza un análisis de una imagen de un tomograma del ojo del paciente, estas imágenes son tomadas con el PENTACAM.

El PENTACAM es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo, lo cual se realiza a partir de varias fotografías tomadas mediante una cámara rotacional del sistema Scheimpflug perteneciente a dicho equipo. La identificación de la OCP es el resultado del análisis de las imágenes provenientes del mismo. En la actualidad este equipo cuenta con tecnología capaz de capturar hasta 50 imágenes en 2 segundos, en un único escaneo automatizado. Consecutivamente, calcula un modelo tridimensional del segmento anterior del ojo, a partir de los 25 000 puntos de elevación real.

También permite aislar específicamente la estructura deseada de la imagen tomográfica, en este caso el saco capsular, para su posterior análisis. La no presencia de destellos de luz es una ventaja de las imágenes provenientes del PENTACAM, ya que interfiere con el análisis (18).

Las imágenes provenientes del PENTACAM poseen regiones de OCP muy pequeñas, las cuales el especialista no las detecta y por ello en muchas ocasiones emiten un criterio erróneo acerca de esta enfermedad. Existen diversos softwares que identifican la OCP de forma automática, entre ellos se destaca el PANDOC.

PANDOC es un sistema basado en casos para identificar las regiones de la OCP, mediante el uso de las imágenes provenientes del PENTACAM y evaluar su correlación con la gradación subjetiva a través de la lámpara de hendidura. Este software provee al oftalmólogo de una herramienta capaz de detectar diferencias de opacidad, a veces imperceptibles para el ojo humano (16).

En una entrevista realizada al oftalmólogo Iván Hernández López del hospital Pando Ferrer sobre el uso del software PANDOC, este planteó tres elementos fundamentales:

1. Demora en identificar la OCP.
2. Identifica como OCP áreas que no lo son.
3. La necesidad de identificar en caso de ser posible, el borde que divide la capsulorrexis de la OCP.

Se descubrió que la demora del software se debe a que, en el momento de identificar la opacidad, debe recorrer toda la inmensa base de casos y comparar con cada uno de ellos. Muchos de los nuevos casos no cuantifican con el caso más semejante, sino, con el primer caso donde el valor de la función de semejanza sea admisible, por lo que identifica una región errónea.

La problemática antes descrita ha generado el siguiente **problema de la investigación**:

¿Cómo disminuir el tiempo de respuesta y aumentar la efectividad del software PANDOC al identificar las estructuras de opacidad capsular en imágenes provenientes del PENTACAM?

Del problema antes expuesto se define como **objeto de estudio**: procesamiento digital de imágenes médicas.

Identificándose como **campo de acción:** segmentación de imágenes médicas.

Determinándose como **objetivo general:** diseñar e implementar un algoritmo para la identificación automática de la OCP en imágenes provenientes del PENTACAM.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Elaborar el marco teórico referencial relacionado con la identificación de regiones en imágenes médicas.
2. Desarrollar un algoritmo que permita identificar la opacidad en imágenes provenientes del PENTACAM.
3. Validar la solución propuesta aplicando diferentes métodos de validación de software.

Para dar cumplimiento al objetivo de la investigación se definieron las siguientes **tareas de la investigación:**

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

Teniendo en cuenta el problema a resolver se formuló la siguiente **idea a defender**:

El desarrollo de un algoritmo permitirá disminuir el tiempo de respuesta y aumentar la efectividad del software PANDOC al identificar las estructuras de opacidad capsular en imágenes provenientes del PENTACAM.

Para lograr la realización de las tareas antes mencionadas se emplearon los

siguientes métodos de investigación:

Métodos teóricos:

Análisis histórico-lógico: Permitió analizar y estudiar la trayectoria de los algoritmos de segmentación en imágenes médicas, para así contar con una noción del desarrollado del tema a nivel global y tomar la decisión de cuál o cuáles desarrollar.

Método Analítico-Sintético: Con el objetivo de analizar las teorías, documentos e información, permitiendo la extracción de los elementos más importantes que se relacionan con la segmentación en imágenes médicas.

Inductivo – deductivo: Para luego de inducir una serie de conocimientos referentes a la segmentación de la opacidad de la cápsula posterior en imágenes médicas, poder arribar a razonamientos que conlleven a la deducción de conocimientos que puedan ser aplicables al problema a tratar en particular.

Métodos empíricos:

Observación: Se observaron los resultados obtenidos en la caracterización e identificación de los principales algoritmos utilizados, para decidir cuáles serán los adecuados.

Experimento: Método empírico mediante el cual, se realizarán experimentos y pruebas de los principales algoritmos de segmentación de imágenes médicas, para examinar los resultados en busca de escoger los más adecuados.

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

Capítulo 1: Fundamentación Teórica.

Se realizó un estudio de los algoritmos existentes para la identificación de regiones en las imágenes. Se presenta un conjunto de conceptos relacionados con la problemática antes descrita. Además, se describe la metodología, las herramientas y el lenguaje de programación utilizado en el desarrollo de la investigación.

Capítulo 2: Análisis y diseño del algoritmo.

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

Capítulo 3: Implementación y pruebas al algoritmo.

Se da continuidad a las fases de la metodología XP, con la realización de pruebas unitarias, integración y aceptación. Además, se describen los resultados de la aplicación de las métricas para validar el diseño y las técnicas para la validación de los requisitos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se realiza una descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación. Se abordan los principales conceptos asociados al dominio del problema, como el de incidencia de la OCP. Se realiza un estudio para ver las técnicas más utilizadas en la segmentación de imágenes médicas, las que ayudarán en la identificación de la opacidad capsular en pacientes sometidos a cirugía de cataratas.

1.2 Marco teórico

Existen algunos conceptos que se consideran necesarios conocer para tener una mejor comprensión de la investigación. A continuación, se detallan cada uno de ellos.

Procesamiento digital de imágenes: son los cambios que se le realizan a las imágenes para posteriormente extraer información de la misma (7).

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 (7).

Realce y mejora: conjunto de técnicas destinadas a realzar colores y eliminar el ruido en una imagen (7).

Segmentación: conjunto de técnicas destinadas a identificar regiones dentro de una imagen (7).

1.3 Procesamiento digital de imágenes.

El procesamiento digital de imágenes cuenta con 4 etapas: captura, realce y mejora, segmentación y extracción de características. Dentro de la segunda y tercera etapa existen un conjunto de técnicas aplicables a imágenes médicas, a continuación, se relacionan cada etapa con sus técnicas.

1.3.1 Realce y mejora

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.

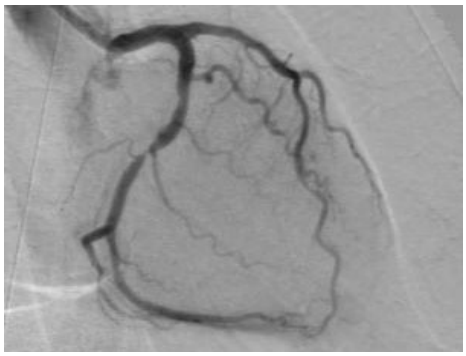


Figura 1 Imagen original (7).

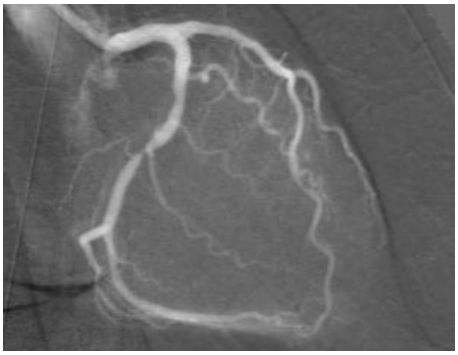


Figura 2 Resultado de Negativo (7).

Técnica Control de brillo

En ciertas ocasiones, la apariencia de una imagen puede realzarse 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 (7).

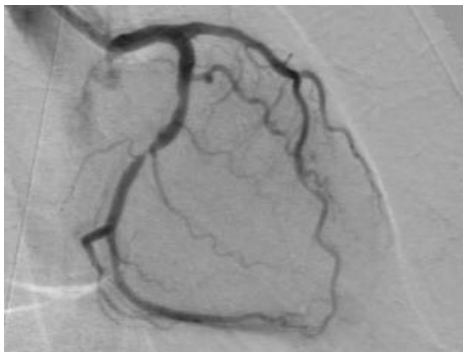


Figura 3 Imagen original (7).

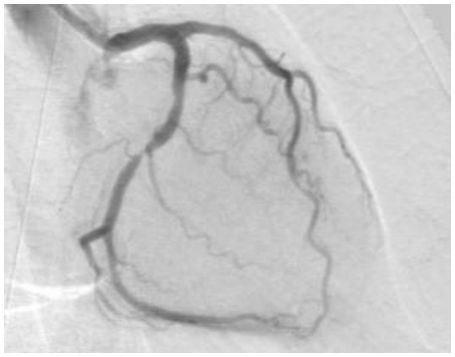


Figura 4 Resultado de control de brillo (7).

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 píxeles 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 píxeles de la imagen de entrada cuyos valores están fuera de la banda especificada, se les asigne el color negro (7).

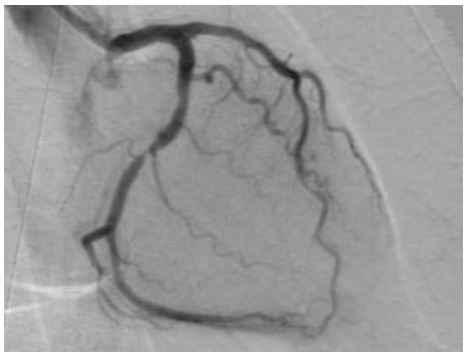


Figura 5 Imagen original (7).

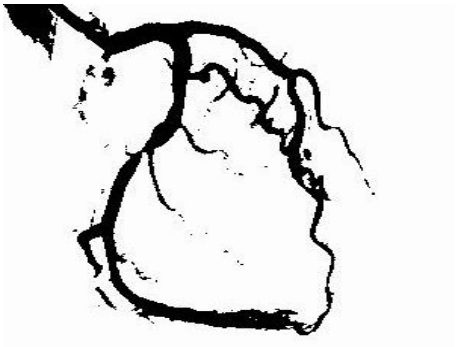


Figura 6 Resultado de Binarización (7).

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 o LUT's, 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 (7).

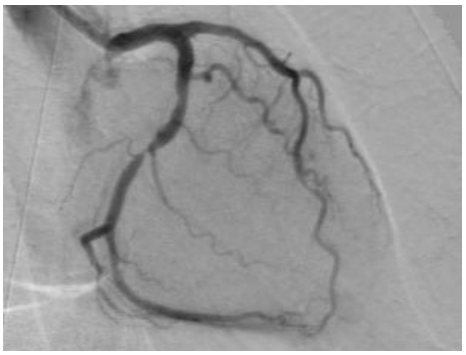


Figura 7 Imagen original (7).

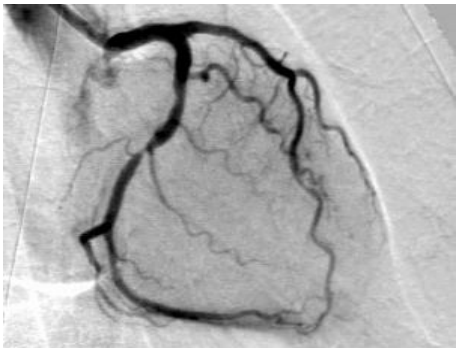


Figura 8 Resultado de Ampliación del contraste (7).

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 que se muestra se encuentra en Escala de grises.

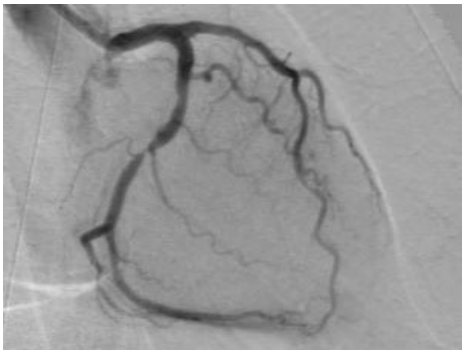


Figura 9 Imagen original

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 píxeles con niveles de gris diferentes a los demás se asemejen más a sus vecinos (7). Esta técnica de filtrado 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 (9).

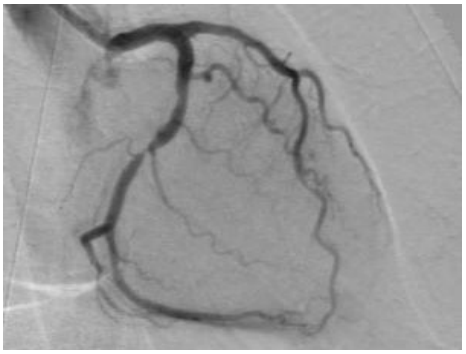


Figura 10 Imagen original

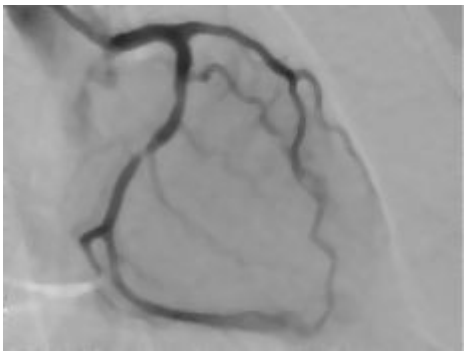


Figura 11 Filtrada por la Mediana

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 píxeles con niveles de gris diferentes a los demás se asemejen más a sus vecinos (7). Esta técnica de filtrado 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 (9).

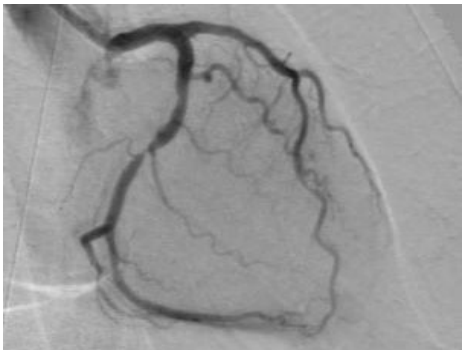


Figura 12 Imagen original

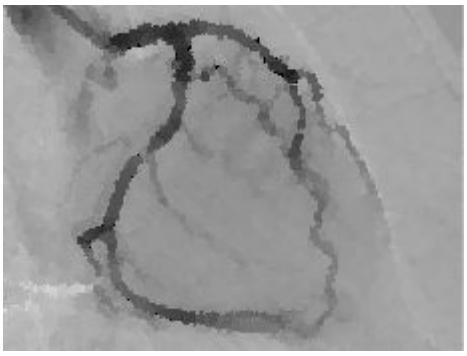


Figura 13 Filtrado por la Moda

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 4 vecinos por lo menos un píxel blanco el cual activa una bandera que indica que hay un borde. Si todos los 4 vecinos son negros no es borde (10).

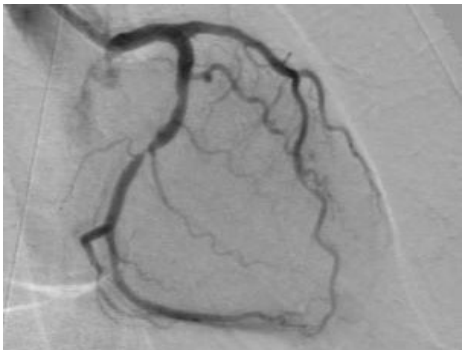


Figura 14 Imagen original

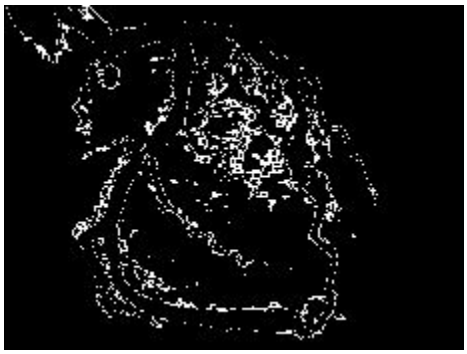


Figura 15 Bordes detectados

1.3.2 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 (7). En general 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. Las técnicas más utilizadas en segmentación son: basada en fronteras, basada en el uso de un umbral y por crecimiento de regiones. A continuación, se describen cada una de ellas.

Técnicas basadas en fronteras

El objetivo de esta técnica es resaltar las fronteras del objeto a segmentar, tiene como ventaja la simplicidad para encontrar las fronteras (24). La efectividad de los algoritmos que implementan estas técnicas dependen del detector de bordes utilizado en la imagen, aunque la presencia de huecos en los bordes no afecta el funcionamiento del mismo. A continuación, se presentan algoritmos basados en esta técnica.

1. **Transformada de Hough**: está diseñada especialmente para encontrar líneas[1], toma una línea encontrada mediante un detector de bordes y busca sobre cuales se encuentra. Es un algoritmo de votación resultante del conjunto de puntos que forman parte de una línea. El arreglo de contadores de votos en el espacio de parámetros puede ser estimado a través de un histograma. Los votos finales totales, serán un contador de coordenadas, que indicará la probabilidad relativa de la hipótesis nula, de que una recta con un conjunto de parámetros exista en la imagen. Como todos los puntos son procesados independientemente, combatirá bien la oclusión y es relativamente robusto al ruido, ya que los puntos erróneos no contribuirán consistentemente y sólo generarán ruido de fondo (30). Para aplicar este algoritmo en una imagen, es necesario obtener primero una imagen binaria de los píxeles que forman parte de la frontera del objeto (24).

   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) (10).

   (1)

[1] Líneas: Colecciones de puntos de borde que son adyacentes y tienen la misma dirección.

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

La estrategia para extraer círculos mediante La Transformada de Hough (30) es la siguiente:

- Detección de los píxeles de borde de los círculos mediante un filtro de bordes.
- Establecimiento de un dominio de parámetros cuyas dimensiones sean el propio espacio de búsqueda y una cuantización suficientemente precisa: la de los píxeles de la imagen original.
- Se barre la imagen de manera que cada pixel etiquetado como borde da lugar a un círculo de radio K centrado sobre el mismo. Las celdas que pertenecen al círculo reciben un voto.
- En teoría, todos los píxeles que pertenecen a un mismo círculo son, en el espacio de parámetros, círculos que se cortan en la misma celda de manera que el centro de cada círculo es determinado como la celda más votada.

  Si el radio es conocido, el dominio de parámetros de cada círculo es bidimensional: coordenadas del centro de cada círculo. En este dominio, cada círculo del espacio se representa con un punto y simétricamente, un punto del dominio espacial se representa en el dominio de parámetros mediante un círculo formado por todos los puntos (dominio de parámetros) que representan a todos los círculos (dominio espacial) que pueden pasar por el punto (dominio espacial).

2. **Contornos Activos:** el modelo contorno activo deformable o Snake como también se le conoce, es representado matemáticamente como una curva v[s]= [x(s), y(s)] que se mueve en un espacio constantemente dentro de un número de iteraciones que se puede interpretar como una secuencia de tiempo. La curva está representada paramétricamente, teniendo como único parámetro a s. Este parámetro, que está relacionado con ambas variables en el espacio (x, y), representa la curva ubicada en el espacio que en este caso es la

imagen de operación. Existe otro parámetro relacionado con la cantidad de iteraciones representadas para desenvolverse. Debido a su representación se considera a este parámetro como t. Se tiene entonces un modelo activo definido como una curva v [s, t]= [x(s, t), y(s, t)] en donde:

s: es el espacio que ocupa la curva.

t: es la cantidad de iteraciones representado como la secuencia de tiempo.

El modelo original está representado como una curva (ecuación 2) parametrizada v[s]= [x(s), y(s)], s $\in$ [0, 1] que se mueve a través de un dominio espacial y busca minimizar el siguiente funcional de energía (29).

(2)

La técnica Contornos Activos permite ajustar curvas dentro de una imagen a los bordes de la misma. Su funcionamiento intenta imitar a una serpiente que pasa por una superficie no plana, ella ajusta su cuerpo a las irregularidades del camino mientras que los Contornos Activos ajustan las curvas a las irregularidades de las imágenes. Los algoritmos que implementan esta técnica tienen un costo computacional elevado, pero existen variantes y aproximaciones que eliminan esta desventaja. Es muy usado en la segmentación de imágenes médicas para identificar regiones con una gran variedad de sus estructuras (28).

Técnicas basadas en el uso de un umbral

Este tipo de segmentación, 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 pixel supera el valor del umbral, entonces el pixel pertenece al objeto, en caso contrario, el pixel 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 pixel

debe ser comparado con el umbral, para decidir si tal pixel 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 (7).

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 que buscamos y qué píxeles son sólo el entorno de estos objetos. De acuerdo a (9) 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 (7). 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

Técnicas 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 pixel, 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 pixel vecino que cumpla con tal condición de similitud, es agrupado junto con los anteriores para conformar así una región (7).

1. **División y fusión de regiones**

Consiste en dividir inicialmente una imagen en un conjunto de regiones arbitrarias disjuntas, por ejemplo 64 divisiones, después, dependiendo del criterio de segmentación, regiones adyacentes son fusionadas si tienen propiedades similares como nivel de gris similares, o son divididos si no comparten las mismas propiedades, como variaciones considerables de niveles de gris. Finalmente, la imagen queda segmentada en un conjunto de regiones homogéneas (27).

1.4 Metodologías de desarrollo de software

Entre las tareas más difíciles de la ingeniería se encuentra el desarrollo de software, debido a que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Las metodologías de software se dividen en dos grupos, ágiles y pesadas. Las metodologías tradicionales o pesadas se caracterizan por llevar una documentación exhaustiva de todo el proceso, centrándose especialmente en el control, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Las metodologías ágiles o ligeras dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (12).

Posteriormente al estudio de las metodologías de desarrollo ágiles y al análisis de sus características, etapas de desarrollo y ventajas, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software; el cliente es parte del equipo de desarrollo, siendo este último de solo una persona, además de la dificultad para un equipo de desarrollo pequeño el adoptar una metodología robusta a causa de la cantidad de documentación generada y la alta resistencia a los cambios durante el desarrollo. Dentro de las metodologías ágiles se destacan: AUP (Proceso Unificado Ágil), Scrum, XP (Programación Extrema) y SXP (SCRUM-XP).

AUP: se enfoca especialmente en la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas, identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos (13).

SCRUM: se centra en la gestión de proyectos y es óptima en aquellas situaciones en las que resulta complicado planificar el futuro con mecanismos de control de procesos empíricos. La retroalimentación entre iteraciones constituye el elemento más potente de la metodología. El software es desarrollado por equipos que se auto organizan en iterativos e incrementales ciclos de corta duración (no más de 30 días) denominados Sprints, comenzando cada uno de ellos con una planificación y finalizando con una revisión retrospectiva. Como puede observarse, Scrum resulta válido en los entornos que trabajan con requisitos cambiantes, y necesitan rapidez de respuesta y flexibilidad (13).

XP: pone énfasis en la colaboración estrecha pero informal (verbal) entre los clientes y los desarrolladores, en el establecimiento de metáforas[2] para comunicar conceptos importantes, en la retroalimentación continua y en evitar la documentación voluminosa como medio de comunicación. Restringe a los desarrolladores para que diseñen sólo para las necesidades inmediatas, en lugar de considerar las del futuro. El objetivo es crear un diseño sencillo que se implemente con facilidad en forma de código. La retroalimentación se obtiene de tres fuentes: el software implementado, el cliente y otros miembros del equipo de software (13).

SXP: es un híbrido cubano de las metodologías ágiles XP y Scrum, que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles

[2] En el contexto de XP, una metáfora es “una historia que cada quien —clientes, programadores y gerentes— narra, acerca de cómo funciona el sistema”

que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto (13).

Se define la metodología XP para guiar el proceso de desarrollo de la presente investigación, 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. En el período de elaboración del algoritmo, esta metodología se utiliza para establecer el control y utilizar un marco de trabajo definido y de probada eficiencia. Una de las prácticas más significativas que posee, es que con XP es posible simplificar el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que no se necesiten. Al realizar pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código. Tiene como objetivo fundamental la satisfacción del cliente. Se recomienda el uso de esta metodología debido a que los requisitos tienen altas probabilidades de cambiar con el tiempo y el proyecto cuenta con un grupo pequeño de programadores (13). Esta metodología ha sido la utilizada durante el desarrollo del software PANDOC, además, es la metodología definida por el grupo de investigación científica *Inteligencia Artificial, Programación e Innovación* (AIRI), al cual pertenece el autor del trabajo.

1.4.1 Descripción de la metodología XP

Entre las características más significativas de XP se encuentran (13):

- **Programación en pares:** consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

- **Pruebas unitarias:** se basa en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- **Entrega de software en espacio de tiempo pequeño:** tiende a entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

XP plantea cuatro fases para el desarrollo:

1. **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.
2. **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.
3. **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 HU.
4. **Pruebas:** consiste en comprobar el funcionamiento de la codificación que se halla implementado, garantizando la evaluación de las distintas tareas en las que ha sido divididas las HU.

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

Visual Paradigm 8.0 para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad,

mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (17).

1.6 Lenguaje de programación

El software PANDOC está desarrollado en el lenguaje Java por tanto se escogió el mismo para desarrollar la solución propuesta. Además, se cuenta con abundante documentación sobre el lenguaje y en el futuro se tiene planeado utilizarse la solución propuesta en una plataforma androide, la cual utiliza java como lenguaje de programación. A continuación, se mencionan algunas de sus características.

- **Interpretado:** se ejecuta en una máquina virtual (15).
- **Robusto:** administra la memoria de la computadora para que el programador no se tenga que preocupar por ello, además de realizar verificaciones en busca de errores lo mismo en tiempo de compilación que en tiempo de ejecución (15).
- **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 (15).
- **Simple:** se enfoca en el contexto de los lenguajes orientados a objetos y elimina la complejidad de otros lenguajes como C (15).
- **Multiproceso:** puede ejecutar diferentes líneas de código al mismo tiempo (15).
- **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 (15).

1.7 Entorno de Desarrollo Integrado

Como entorno de desarrollo integrado para la confección de la herramienta, se seleccionó el NetBeans 8.0. Es libre y de código abierto; contiene todas las herramientas necesarias para crear aplicaciones profesionales de escritorio, empresariales, web y aplicaciones móviles con la plataforma Java. Es conocido por la integración con el lenguaje Java, facilita el desarrollo utilizando funcionalidades como completamiento de código, coloreo de sintaxis, también permite utilizar y editar los componentes visuales de forma sencilla. Además, incluye el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente. Es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris (15).

1.8 Herramienta para la realización de pruebas al algoritmo

Se definió por el grupo de investigación AIRI, la herramienta MatLab 2015 para realizarle pruebas a las técnicas del algoritmo. Es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux. Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Además, se pueden ampliar las capacidades de MatLab con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets). Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL (26).

1.9 Patrones de diseño

Un patrón de diseño nombra, motiva y explica de forma sistemática un diseño general que afronta un problema de diseño recurrente en los sistemas orientados a objetos. Describe el problema, la solución, cuándo aplicar la

solución y sus consecuencias. También ofrece pistas para la implementación y ejemplos (31).

Se pueden agrupar en dos grandes grupos; los GRASP (General Responsibility Assignment Software Patterns, en inglés), que son patrones generales de software para asignación de responsabilidades y los GOF (Gang of Four, en inglés), encargados de la inicialización, agrupación y comunicación de los objetos.

1.10 Pruebas

Las pruebas de software constituyen esencial para la garantía de la calidad del software, debido a que en el proceso de desarrollo los errores pueden empezar a darse desde el primer momento, es por eso que debe ir acompañado de una actividad que garantice la calidad. Estas pruebas se ejecutan dirigidas a componentes o al sistema en su totalidad, con el objetivo de medir el grado en el que este cumple con los requerimientos pactados. Existen cuatro niveles de pruebas:

- **Pruebas unitarias:** se concentra en el esfuerzo de verificación de la unidad más pequeña del diseño, ya sea un componente o un módulo del software (32).

- **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 aplicó una prueba de unidad y construir una estructura de programa que determine el diseño (32).

- **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 (32).

- **Pruebas de aceptación:** una vez culminado el proceso de pruebas por parte del equipo de desarrollo, es indispensable, que el cliente verifique que el producto ha sido desarrollado con las normas y criterios establecidos, y cumple con todos los requisitos especificados por el cliente (32).

1.11 Conclusiones parciales

- El análisis de los conceptos relacionados con la OCP y el procesamiento digital de imágenes permitió una mejor comprensión del tema de la investigación.

- Con el estudio de las técnicas existentes para la identificación de regiones en imágenes se determinaron las útiles para la solución propuesta.

- Con el análisis de las metodologías y las tecnologías de desarrollo de software se seleccionaron las empleadas en el proceso de desarrollo.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL ALGORITMO

2.1 Introducción

En el presente capítulo se describen los principales elementos tenidos en cuenta en el desarrollo de la solución propuesta, las técnicas de procesamiento de imágenes empleadas, los patrones de diseño y los estándares de codificación. El trabajo se organizó en función de las fases que define la metodología XP y además se hace referencia a los artefactos generados en cada una de ellas.

2.2 Fases del proceso de desarrollo

2.1.1 Planificación

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

2.1.1.1 Historias de usuario

La HU es la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (25). A continuación, se describen las HU del negocio.

Tabla 1 HU “Recortar imagen’’.

Historia de Usuario	
Número: 1	**Nombre de la Historia de Usuario:** Recortar imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados:** 2semanas
Riesgo en desarrollo: Alto	**Puntos reales:** 2 semanas
Programador responsable: Erlis Paula Vidal	
Descripción: **Recortar imagen:** permite extraer solo el área de interés de la imagen proveniente del PENTACAM .	
Observaciones: **Recortar imagen:** la imagen debe ser proveniente del PENTACAM.	

Tabla 2 HU “Filtrar imagen’’.

Historia de Usuario	
Número: 2	**Nombre de la Historia de Usuario:** Filtrar imagen
Modificación a la Historia de Usuario: Ninguna	

Usuario: Michel Álvarez Cancio	**Iteración asignada:** 1
Prioridad en negocio: Alta	**Puntos estimados: 3** semanas
Riesgo en desarrollo: Alto	**Puntos reales:** 3 semanas
Programador responsable: Erlis Paula Vidal	
Descripción: **Filtrar imagen:** Elimina el ruido presente en la imagen y realza los bordes de la misma.	
Observaciones: **Filtrar imagen:** La imagen proveniente del PENTACAM debe estar recortada.	

Tabla 3 HU ‘’Segmentar imagen’’.

Historia de Usuario	
Número: 3	**Nombre de la Historia de Usuario:** Segmentar imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 3 semanas
Riesgo en desarrollo: Alto	**Puntos reales:** 3 semanas
Programador responsable: Erlis Paula Vidal	

Descripción: **Segmentar imagen:** permite identificar la opacidad y la capsulorrexis en la imagen.
Observaciones: **Segmentar imagen:** la imagen debe estar filtrada.

3. **Tabla 4** HU ‘Visualizar imagen’’.

Historia de Usuario	
Número: 4	**Nombre de la Historia de Usuario:** Visualizar imagen
Modificación a la Historia de Usuario: Ninguna	
Usuario: Michel Álvarez Cancio	**Iteración asignada:** 2
Prioridad en negocio: Alta	**Puntos estimados:** 2 semanas
Riesgo en desarrollo: Alto	**Puntos reales: 2** semanas
Programador responsable: Erlis Paula Vidal	
Descripción: **Visualizar imagen:** permite visualizar la imagen con la opacidad realzada en el software PANDOC.	
Observaciones: **Visualizar imagen:** recibe las coordenadas donde está ubicada la opacidad.	

2.1.1.2 Requisitos del algoritmo

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

Requisitos funcionales

- RF1: Cargar imagen
- RF2: Filtrar imagen
- RF3: Segmentar imagen
- RF4: Visualizar imagen.

Requisitos no funcionales

- **Requisito de software**

RNF 1: Se debe tener instalada en la computadora la máquina virtual de Java en su versión 8 o superior.

- **Requisitos de hardware**

RNF 2: Para ejecutar el algoritmo 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.

2.1.1.3 Estimación de esfuerzo por historias de usuario

La estimación de esfuerzo asociado a la implementación de las historias de usuario la establece los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Si la estimación supera las 3 semanas, la HU deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. En caso de que el esfuerzo sea menor que una semana, la HU será combinada por otra (25).

Tabla 5 Estimación de esfuerzo por HU

No	Historias de usuario	Puntos de estimación
1	Capturar imagen	2
2	Filtrar imagen	3
3	Segmentar imagen	3
4	Visualizar imagen	2

2.1.1.4 Plan de iteraciones

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

Tabla 6 Plan de iteraciones

Iteraciones	Historias de usuario a implementar	Duración
1	HU 1	5
	HU2	

2	HU 3	5
	HU 4	

2.1.1.5 Plan de entrega

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

Tabla 7 Pan de entrega

	Iteración 1	**Iteración 2**
Cantidad de HU	2	2
Fecha de entrega	1/4/2016	6/5/2016

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. De esta forma se garantiza una constante retroalimentación entre los desarrolladores y clientes. 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

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) ayudan al equipo de desarrollo a definir actividades durante el diseño del sistema. Permiten trabajar con una metodología basada en objetos, haciendo posible que el equipo de

desarrollo completo contribuya en la tarea del diseño. El título de la tarjeta es el nombre de la clase en cuestión y posteriormente se sitúa a la izquierda las responsabilidades de la clase, y a la derecha las clases que se utilizan para cumplir con cada responsabilidad. A continuación, se muestra la tarjeta CRC de la clase *Circulo.*

CRCCírculo	
Description: En esta clase se crean los círculos instanciando a la misma.	
Attributes:	
Name	Description
x	Posición en el eje x del centro del círculo.
y	Posición en el eje y del centro del círculo.
radio	Radio del círculo.
cantVotos	En esta variable se guarda la cantidad de veces que es votado dicha instancia.

Figura 16 Tarjeta CRC de la clase *Circulo*

2.1.2.2 Estándares de codificación

Los estándares de codificación permiten entender de manera rápida y sencilla el código empleado en el desarrollo de un software. El estándar de codificación empleado en el desarrollo del algoritmo fue definido por el equipo de desarrollo. A continuación, se muestran algunas pautas de dicho estándar.

- 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.
- Los nombres de los métodos serán con minúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula.
- 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.

- Los nombres de variables o funciones deben ser lo suficientemente descriptivos, sin exceder de 30 caracteres.
- Todas las funciones deben tener comentarios explicando que realiza cada una de ellas.

2.1.2.3 Patrones de diseño

Los patrones de diseño representan una descripción de las clases y objetos, comunicándose entre sí para resolver un problema de diseño general en un contexto particular. El uso de patrones posibilita estandarizar el modo en que se realiza el diseño y proporciona reusabilidad, extensibilidad y mantenimiento del código. En la presente investigación se emplean los siguientes:

Patrones de comportamiento:

- **Estrategia (Strategy):** Utilizado para manejar la selección de un algoritmo. (Se utilizó en el algoritmo de Contornos Activos a la hora de buscar un borde hacia adelante o hacia detrás).
- **Observador (Observer):** Notificaciones de cambios de estado de un objeto. (Notificaciones de cuando acaban de ejecutarse los hilos de ThreadCentro).
- **Iterador (Iterator):** Define una interfaz que declara los métodos necesarios para acceder secuencialmente a una colección de objetos sin exponer su estructura interna. (La clase ListaEstatica que contiene variables estáticas)

Patrones Creacionales

- **Singleton:** Restringe la instanciación de una clase o valor de un tipo, a un solo objeto. (Se utiliza en la clase ListaEstatica)

2.3 Propuesta de solución

Un algoritmo de segmentación consta de dos etapas previas a la segmentación, la captura y el realce y mejora. A partir de los resultados de estas etapas, se

enuncian un conjunto de pasos para el desarrollo del algoritmo. A continuación, se presentan las etapas del algoritmo propuesto, implementadas en MatLab para observar y elegir las técnicas que ofrezcan mejores resultados en cada etapa del procesamiento digital de imágenes, para identificar de forma automática la OCP en imágenes provenientes del PENTACAM:

1. Captura.
2. Recorte de la imagen.
3. Aplicación de filtros de realce y mejora de la imagen.
4. Aplicación de filtros de eliminación de ruido de la imagen.
5. Detección de bordes de la imagen.
6. Aplicación del algoritmo La Transformada de Hough para formas circulares.
7. Aplicación del algoritmo de Contornos Activos.

Captura

La imagen es tomada por el oftalmólogo, haciendo uso del PENTACAM. Posteriormente se da la ubicación de esta imagen al software PANDOC, que es el encargado de realizar el resto de los pasos.

Recorte de la imagen.

El PANDOC luego de haber cargado la imagen con éxito, realiza un recorte de la misma con el objetivo de solo analizar la zona en la cual está contenida la estructura ocular del ojo humano, esta tarea se realiza automáticamente una vez seleccionada la imagen en el software.

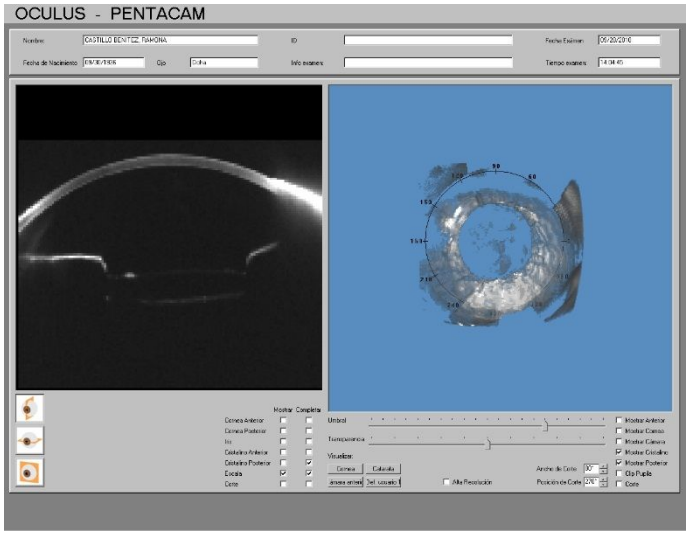


Figura 17 Imagen sin recortar

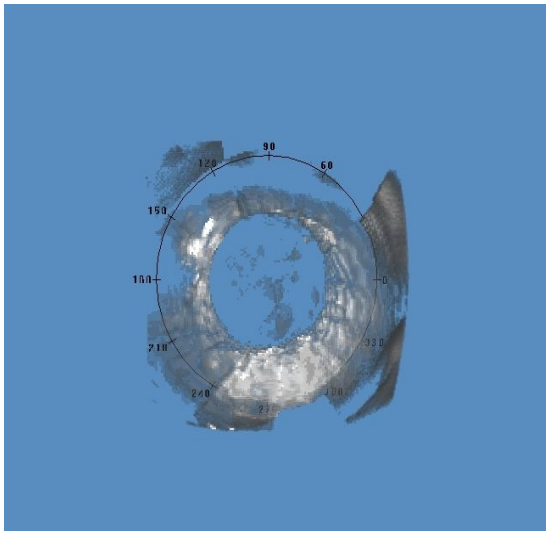


Figura 18 Imagen después de recortar

Aplicación de filtros de realce y mejora de la imagen

En esta fase el algoritmo propuesto utiliza la técnica de Binarización para utilizar solo dos colores (blanco y negro) pasando a blanco todos los píxeles que no me sean necesarios para el análisis, así como resaltando a negro el resto de la imagen. Esta tarea es realizada utilizando umbrales fijos, si el valor en escala de gris del pixel está contenido entre 138 y 150 o es menor que 40, este pixel es considerado del fondo de la imagen y se le da color 255 (blanco). Al resto de los píxeles que no cumplan con lo antes descrito se le asigna

automáticamente el color 0 (negro). A continuación, se muestra el seudocódigo y la salida del mismo.

```
//Binarización
imagenBinaria→imagenRecortada
Desde 0 hasta el final del largo de la imagenBinaria
Desde 0 hasta el final del ancho de la imagenBinaria
Si el píxel pertenece al fondo de la imagen se cambia a color blanco y si no a color negro.
```

Figura 19 Seudocódigo de la técnica de Binarización

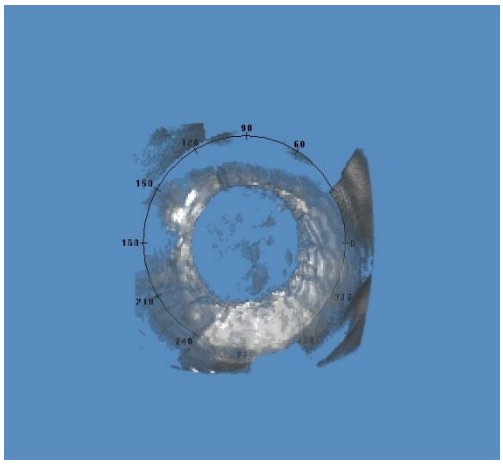


Figura 20 Imagen antes de binarizar

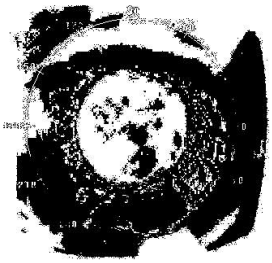


Figura 21 Imagen después de binarizar

Aplicación de filtros de eliminación de ruido de la imagen

Después de encontrarnos con la imagen ya binarizada nos percatamos que la elección del umbral influye a la hora de llevar algunos píxeles a negro. Existen casos de píxeles aislados, que no los necesitamos por lo que se decidió la aplicación de filtros de eliminación de ruido. Estos píxeles independientes, también provocaban la detección de bordes sobrados, los que provocaban un mayor tiempo de respuestas de los algoritmos existentes de segmentación. A continuación, se muestra el seudocódigo y la salida del mismo.

```
//Mediana
imagenMediana→imagenBinaria
Desde 0 hasta el final del largo de la imagenMediana
Desde 0 hasta el final del ancho de la imagenMediana
Se va cambiado el valor de cada píxel por la media aritmética de sus vecinos.
```

Figura 22 Seudocódigo de la técnica de la Mediana

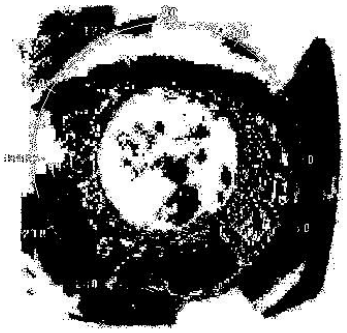


Figura 23 Imagen binarizada

Figura 24 Resultado de la aplicación del filtro de la Mediana

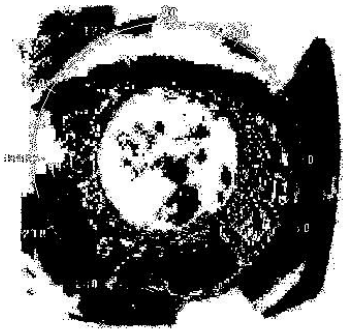


Figura 25 Imagen binarizada

Figura 26 Resultado de la aplicación del filtro de la Moda

Selección de las técnicas de preprocesamiento de imágenes

Para la selección de la mejor técnica de preprocesamiento se realizó una tabla comparativa tomando como puntos a evaluar la eliminación del ruido sal y pimienta, su efectividad al hacerlo y la complejidad del mismo. A continuación, se muestra esta comparación:

Tabla 8 Comparación de las técnicas de preprocesamiento.

	Negativo	Mediana	Ampliación del contraste	Control de brillo	Moda
Eliminación de ruido (Sal y pimienta).	No	Si	No	No	Si
Efectividad.	-------	Alta	--------	-------	Media
Complejidad	Baja	Media	Media	Media	Media

Dado el resultado de esta comparación se toma como mejor técnica de eliminación de ruido el filtro de la Mediana, debido a elimina el ruido de tipo sal y pimienta con una efectividad alta y posee una complejidad media.

Detección de bordes de la imagen.

Dado que la imagen a la cual se le quiere detectar bordes es binaria, se pueden identificar los bordes utilizando el algoritmo iterativo. Este algoritmo es muy efectivo cuando se tratan de imágenes binarias, además posee un tiempo de respuesta superior comparado con otros algoritmos. Funciona recorriendo todos los píxeles de la imagen con valor 0, y de estos píxeles los que tengan un vecino 255 será considerado un borde asignándole el valor 255, Todos los demás píxeles serán llevados a color 0. A continuación se muestra el seudocódigo y la salida del mismo.

```
//Bordes

imagenBordes→imagenMediana

Desde 0 hasta el final del largo de la imagenBordes

Desde 0 hasta el final del ancho de la imagenBordes

Si se encuentre una vecindad blanco-negro entre píxeles se cambia a color blanco.

Sino se cambia a negro.
```

Figura 27 Seudocódigo de la técnica de realce de bordes

Figura 28 Imagen filtrada por la Mediana

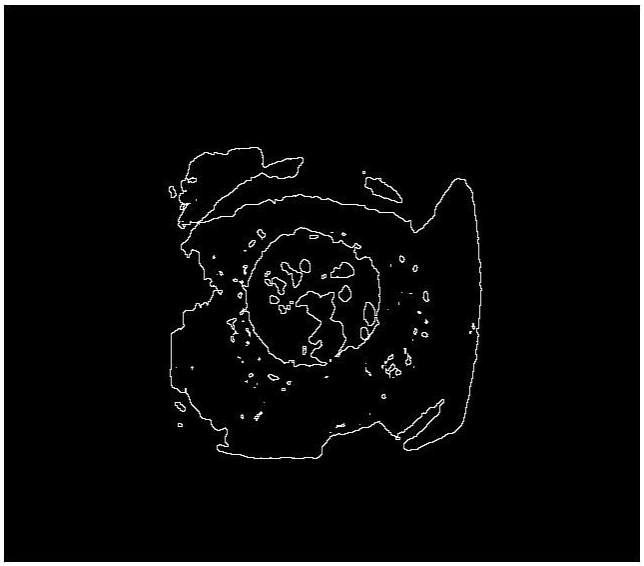


Figura 29 Imagen con los bordes detectados

Selección de la técnica de segmentación

Las imágenes provenientes del PENTACAM presentan una estructura muy variada, además la región que es de interés segmentar presenta en la mayoría de los casos el mismo valor RGB que sus zonas aledañas, dificultando la selección de un umbral. La efectividad de una técnica basada en el uso de umbrales depende de la selección de los mismos, por lo que esta técnica no resulta de utilidad aplicarla debido a lo antes descrito. Las técnicas basadas en fronteras no necesitan umbrales, segmentan las imágenes con algoritmos que utilizan los bordes (cambios de intensidades) como único requisito de entrada. Resulta útil aplicar La Transformada de Hough debido a la estructura redonda que en algunos tramos presenta la capsulorrexis, quedando de esta forma queda identificada la OCP. También se necesita identificar la capsulorrexis, y el algoritmo Transformada de Hough no es capaz de hacerlo del todo, partiendo de la idea del algoritmo de Contornos Activos se le realiza una modificación al primero deformando la circunferencia encontrada hacia los bordes de la capsulorrexis. De esta forma realiza una fusión de ambos algoritmos basados en fronteras.

Aplicación del algoritmo La Transformada de Hough para formas circulares.

Este algoritmo depende de un buen detector de bordes ya que depende de ellos a la hora de realizar sus cálculos y comparaciones. Es el algoritmo de segmentación más utilizado en la detección de formas circulares y el resultado de haberlo aplicado a nuestras imágenes se muestra a continuación. Funciona recorriendo todos los píxeles color 255 de la imagen, para cada uno de ellos traza círculos tangentes en los 360 grados, los círculos son realizados con un radio variable de entre 55 y 100 píxeles. De estos círculos los que cumplan que el centro del eje numérico está contenido dentro de los mismos, son adicionados a una lista en la cual, si el círculo ya se encuentra, se incrementan sus contadores en 1. Posteriormente se recorre esta lista sacando el círculo con mayor cantidad de repeticiones (votos), siendo este en la mayor cantidad de casos de prueba el que limita la opacidad. A continuación, se muestra el seudocódigo y la salida del mismo.

```
//Hough
listaCirculos→imagenMediana
Desde 0 hasta el final del largo de la imagenBordes
Desde 0 hasta el final del ancho de la imagenBordes
Se guardan en listaCirculos todos los círculos detectados en cada píxeles, aumentando el contador si ya existe.
Fin desde
Fin desde
Se ordena listaCirculos de mayor a menor y se escoge el primer círculo que es el más votado.
```

Figura 30 Seudocódigo de La Transformada de Hough

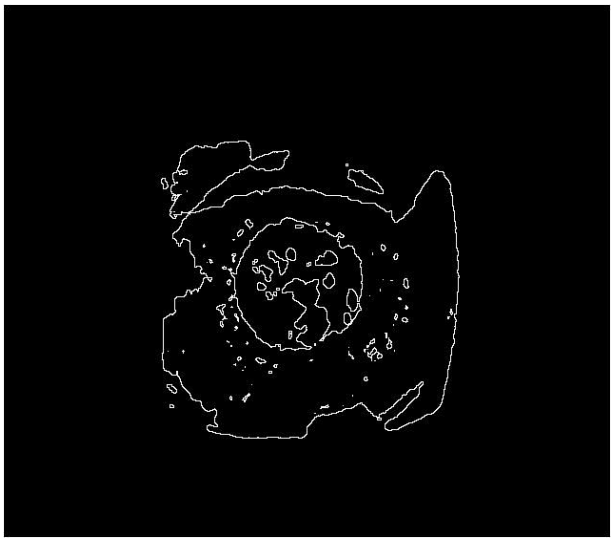


Figura 31 Imagen con los bordes detectados

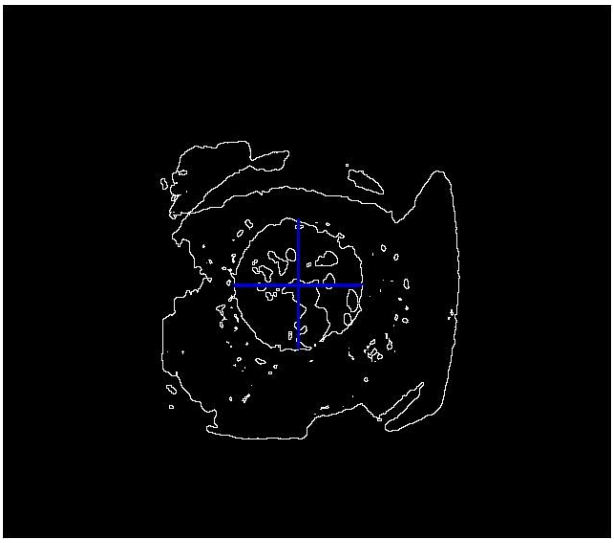


Figura 32 Resultado de aplicar La Transformada circular de Hough

Aplicación del algoritmo de Contornos Activos

La efectividad de este algoritmo depende de una imagen con los bordes bien detectados y de una correcta detección del círculo que limita la opacidad, toma como partida las coordenadas de cada punto del borde de la circunferencia detectada, deslizando cada punto hacia el borde más cercano que este contenido en la recta que corta el centro del círculo y el punto que se está analizando de la circunferencia. En la presente investigación este algoritmo se fusionó con La Transformada de Hough, tomando solo de él la característica de

ajustarse a los bordes más próximos al punto que se esté analizando. A continuación, se muestra el seudocódigo y la salida del mismo.

```
//Snakes

Desde 0 hasta el final del largo del radioCirculo.

Se desplaza el píxel del borde de la circunferencia más votada a los bordes más próximos dentro de la
imagen.
```

Figura 33 Seudocódigo de Snake

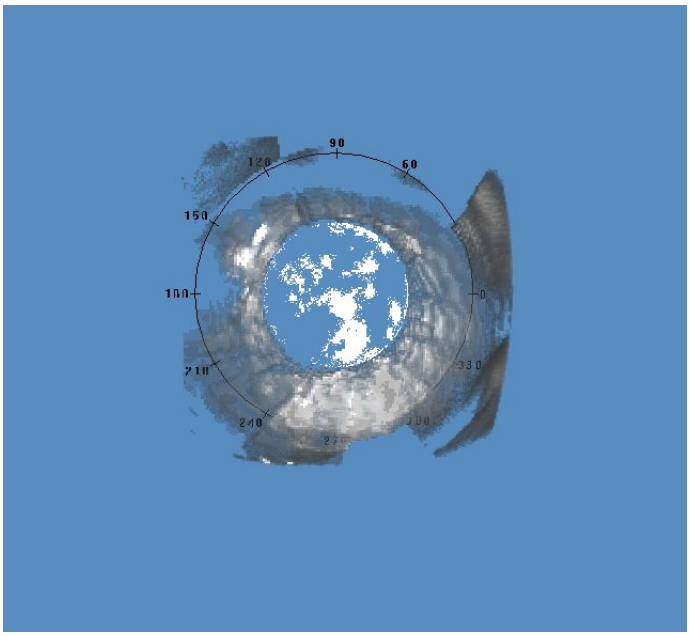


Figura 34 Imagen con La Transformada de Hough aplicada

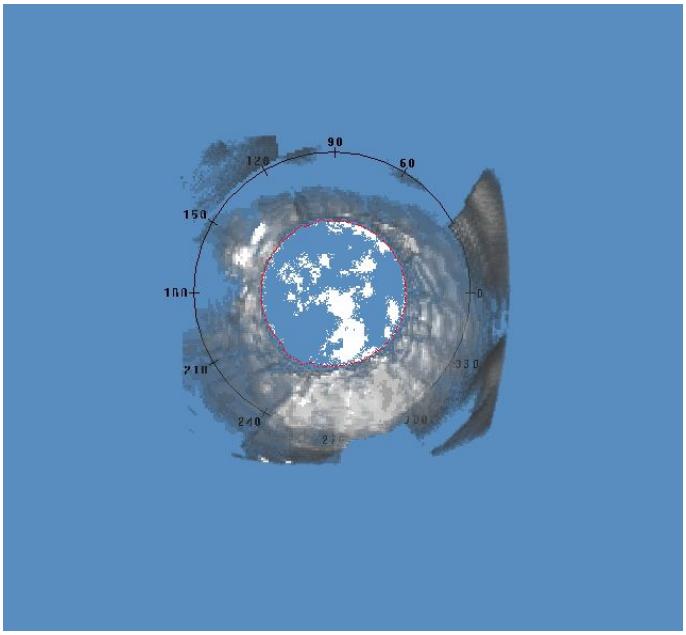


Figura 35 Resultado de la fusión de ambos algoritmos

2.4 Conclusiones parciales

- La utilización de la metodología XP para describir el proceso de desarrollo del algoritmo permitió realizar un trabajo organizado y estructurado, generando los artefactos de cada una de sus fases.
- El uso de estándares de codificación y patrones de diseño permitió obtener un diseño estándar y organizado que será de fácil comprensión para otros desarrolladores.
- Las pruebas realizadas en el software MatLab permitieron la elección de las mejores técnicas de las etapas de realce y mejora y segmentación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS AL ALGORITMO

3.1 Introducción

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

3.2 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.

3.2.1 Tareas de ingeniería

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 3 y 5, que tienen como objetivo implementar las HU Segmentar imagen e integrar la solución al software PANDOC respectivamente:

Tabla 9 Tarea de ingeniería 3

Tarea de ingeniería	
Número de tarea: 3	**Historia de Usuario (No.3):** Segmentar imagen
Nombre de tarea: Implementar HU_ Segmentar imagen	
Tipo de tarea: Desarrollo	**Puntos estimados:** 3 semanas
Fecha de inicio:04/04/2016	**Fecha de fin:** 22/04/2016

Programador responsable: Erlis Paula Vidal
Descripción: permite identificar la opacidad y la capsulorrexis en imágenes médicas.

Tabla 10 Tarea de ingeniería 5

Tarea de ingeniería	
Número de tarea: 5	**Historia de Usuario (No.4):** Visualizar imagen
Nombre de tarea: Integrar al software PANDOC	
Tipo de tarea: Desarrollo	**Puntos estimados:** 1 semana
Fecha de inicio:02/05/2016	**Fecha de fin:** 06/05/2016
Programador responsable: Erlis Paula Vidal	
Descripción: permite integrar el algoritmo al software PANDOC.	

3.2.2 Resultado de la implementación

Ya concluida la implementación de las tareas de ingeniería, el software PANDOC cuenta con un algoritmo para detectar de forma automática la OCP. El algoritmo para identificar la OCP en imágenes provenientes del PENTACAM es un conjunto de técnicas ordenadas del procesamiento digital de imágenes, las cuales son: Escala de grises, Binarización, Mediana, detección de bordes iterativo, La Transformada de Hough y Contornos Activos. Para garantizar la calidad de lo implementado, se avanzó hacia la fase de prueba según propone la metodología en uso.

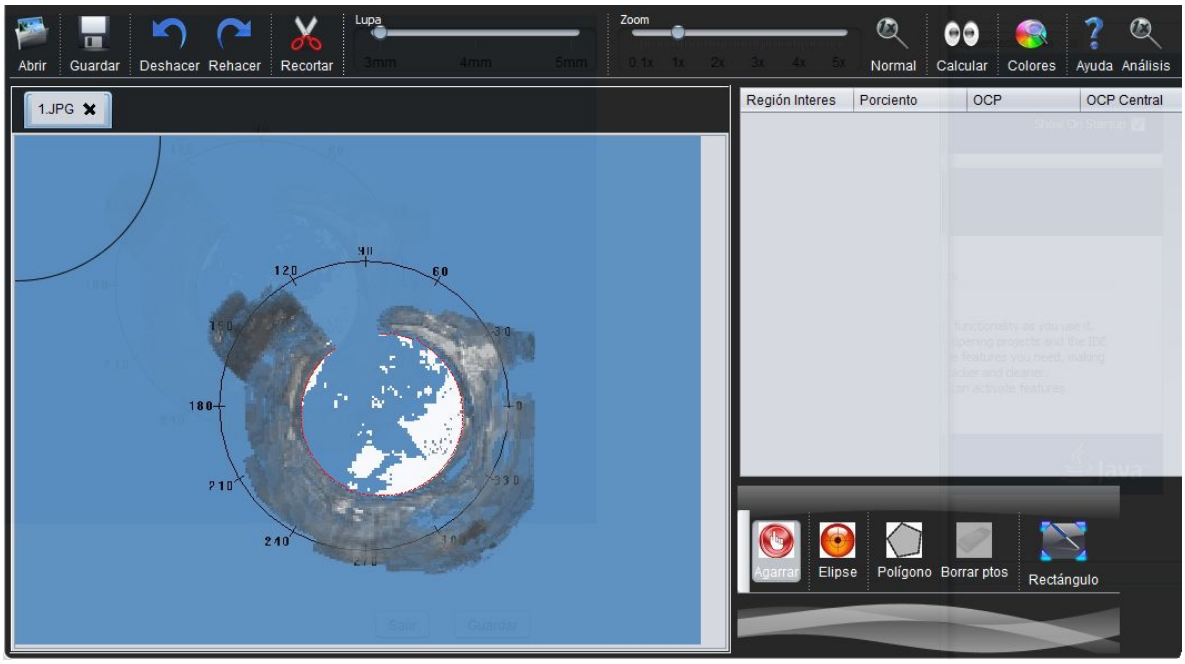


Figura 36 Algoritmo propuesto integrado al software PANDOC

3.3 Técnica de validación de los requisitos

Para demostrar que los requerimientos previamente definidos cumplen con las expectativas del cliente se realizó revisiones formales de los requisitos para la validación de los mismos. Se obtuvieron un total de 5 no conformidades de tipo formato, redacción y técnica; estas fueron corregidas a tiempo, obteniéndose por parte del cliente un Acta de Aceptación (ver Anexo 1).

3.4 Validación del diseño

Para comprobar la calidad del diseño se emplearon las métricas de diseño relaciones entre clases (RC) y Tamaño operacional de clase (TOC).

3.4.1 Relaciones entre clases (RC)

La métrica RC está dada por el número de relaciones de uso de una clase con otra. El primer paso es evaluar un conjunto de atributos de calidad entre los que se encuentran el acoplamiento, complejidad de mantenimiento y reutilización de cada clase (33).

Para determinar el grado de afectación para los atributos de calidad que mide la métrica RC es necesario conocer la cantidad de relaciones de uso (CRU) que

poseen las clases a medir. Una vez obtenida la CRU, se procede a calcular el promedio de las mismas. Teniendo en cuenta los valores antes obtenidos se determina la incidencia de los atributos de calidad en cada una de las clases, según los criterios expuestos en la tabla 11.

Tabla 11 Rango de valores para medir la afectación de los atributos de calidad (RC)

Atributos de calidad	**Clasificación**	**Criterio**
Acoplamiento	Ninguna	CRU = 0
	Baja	CRU = 1
	Media	CRU = 2
	Alta	CRU > 2
Complejidad de mantenimiento	Baja	CRU <= Promedio
	Media	Promedio < CRU < = 2* promedio
	Alta	CRU > 2* promedio
Reutilización	Baja	CRU > 2* promedio
	Media	Promedio < CRU < = 2* promedio
	Alta	CRU <= Promedio
Cantidad de pruebas	Baja	CRU <= Promedio
	Media	Promedio < CRU < = 2* promedio
	Alta	CRU > 2* promedio

En la siguiente imagen se muestra el resultado obtenido al aplicar la métrica RC

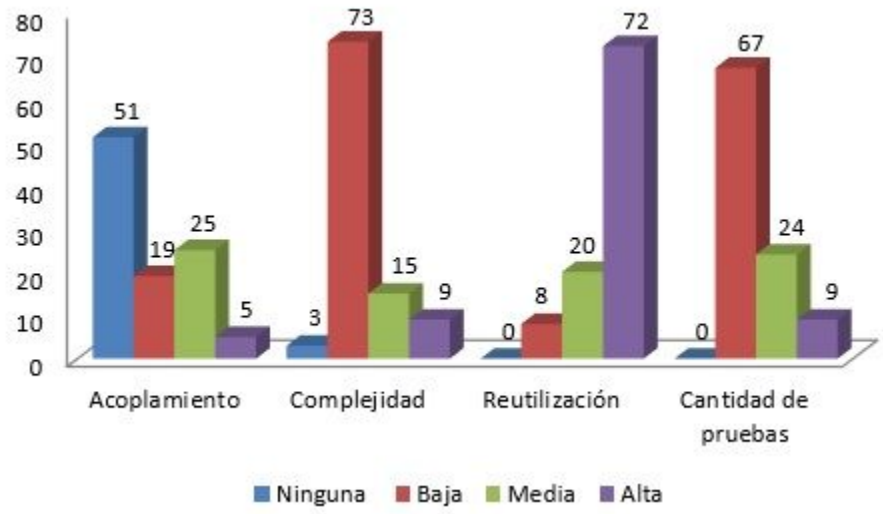


Figura 37 Representación en (%) de los resultados de la aplicación de la métrica RC

Acoplamiento: según los resultados que se muestran, el 51% de las clases no posee
relaciones de uso por lo que la mayoría de las clases no tienen valores de acoplamiento,
validando una realización correcta del diseño.

Complejidad de mantenimiento: según los resultados que se muestran en la figura anterior, el 73% de las clases se comportan de forma satisfactoria pues, son de fácil soporte.

Reutilización: según los resultados que se mostraron en la figura, el 72% de las clases tiene un alto grado de reutilización.

Cantidad de pruebas: luego de aplicar la métrica se obtuvo que el 67% de las clases
poseen un bajo grado de esfuerzo a la hora de realizar cambios, rectificaciones y pruebas de software.

Según lo analizado anteriormente, los valores de RC se comportan de forma satisfactoria. Estos resultados expresan que las clases del diseño del algoritmo presentan bajo acoplamiento, la complejidad de mantenimiento y la cantidad de pruebas son bajas y en consecuencia el grado de reutilización es alto.

3.4.2 Tamaño Operacional de clases (TOC)

La métrica TOC fue aplicada a cada una de las clases del diseño con el objetivo de medir la calidad de las mismas con respecto a su grado de responsabilidad, complejidad de implementación y reutilización (33).

El tamaño general de una clase se puede determinar sumado todas las operaciones que posee, el resultado es tomado como umbral. El próximo paso es calcular el promedio correspondiente a los umbrales, una vez obtenidos se procede a calcular el comportamiento de cada clase respecto a los atributos de calidad establecidos, de acuerdo a lo expuesto en la tabla 8:

Tabla 12 Rango de valores para medir la afectación de los atributos de calidad (TOC).

Atributos de calidad	**Clasificación**	**Criterio**
Responsabilidad	Baja	Umbral<= Promedio
	Media	Promedio < Umbral < = 2* promedio
	Alta	Umbral > 2* promedio
Complejidad de implementación	Baja	Umbral <= Promedio
	Media	Promedio < Umbral < = 2* promedio
	Alta	Umbral > 2* promedio
Reutilización	Baja	Umbral > 2* promedio
	Media	Promedio < Umbral < = 2* promedio
	Alta	Umbral <= Promedio

En la siguiente figura se muestra el resultado de la aplicación de la métrica TOC.

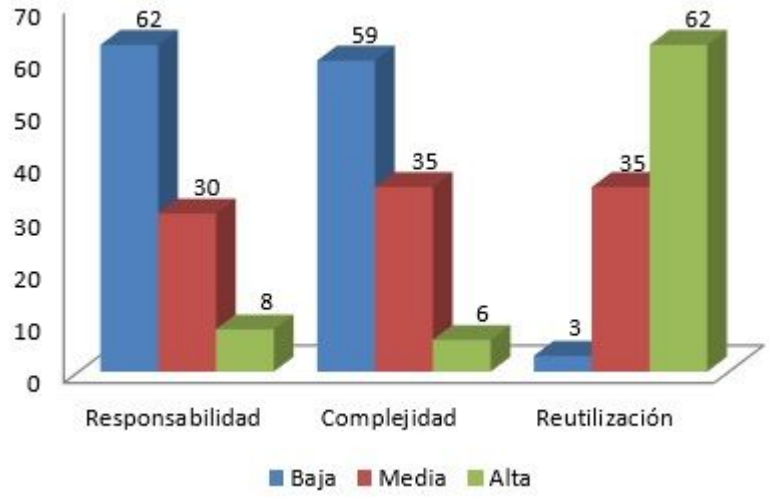


Figura 38 Representación en (%) de los resultados de la aplicación de la métrica TOC.

Responsabilidad: luego de aplicar la métrica se obtuvieron resultados satisfactorios que reflejan una responsabilidad baja con un valor del 62%.

Complejidad de implementación: después de haberse realizado la medición de la métrica, arrojó también resultados positivos ya que la complejidad de las clases es baja en un 59 %.

Reutilización: se obtuvieron valores que según muestra la gráfica de la figura anterior se comporta en un nivel alto con un 62%

Los resultados arrojados por la métrica TOC expresan que las clases del diseño del algoritmo presentan una elevada reutilización y baja complejidad y responsabilidad. Por lo que se concluye que los resultados obtenidos en esta métrica son positivos.

3.5 Pruebas

XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las primeras son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las otras están destinadas a verificar que al final de cada iteración las historias de usuario cumplan con la funcionalidad asignada y satisfagan las necesidades del cliente (23). Debido a la necesidad de integrar el algoritmo al software PANDOC, se realizaron pruebas de integración a la clase Editor y Segmentacion, para la correcta realización de la misma.

3.5.1 Pruebas unitarias

Como parte de las pruebas unitarias se aplicaron al algoritmo las técnicas de caja blanca. Estas pruebas precisan el acceso al código del programa de modo que se pueda comprobar su lógica interna (23). Para realizar esta técnica, se eligió la técnica del camino básico.

A continuación, se muestra el resultado obtenido al aplicarle la técnica del **camino básico** al método binarizacion de la clase Segmentacion, que devuelve una imagen donde todos los píxeles serán blancos o negros.

```java
//Binarizacion con umbral fijo
public BufferedImage binarizacion(BufferedImage imageActual) {
    int mediaPixel;//1
    Color colorAux;//2
    BufferedImage copiaImg = new BufferedImage(imageActual.getWidth(), imageActual.getHeight(), imageActual.getType());//3
    copiaImg.setData(imageActual.getData());//4
    //Recorremos la imagen pixel a pixel
    for (int i = 0; i < imageActual.getWidth(); i++) {//5
        for (int j = 0; j < imageActual.getHeight(); j++) {//6
            colorAux = new Color(imageActual.getRGB(i, j));//7
            mediaPixel = (int) ((colorAux.getRed() + colorAux.getGreen() + colorAux.getBlue()) / 3);//8
            copiaImg.setRGB(i, j, -1);//12
            if ((mediaPixel <= 142 && mediaPixel >= 138) || mediaPixel <= 20) {//9
                copiaImg.setRGB(i, j, (255 << 16) | (255 << 8) | 255);//10

            } else {//11
                copiaImg.setRGB(i, j, (0 << 16) | (0 << 8) | 0);//12
            }
        }//13
    }//14
    return copiaImg;//15
}//16
```

Figura 39 Método binarizacion

Se confecciona el grafo de flujo del camino básico.

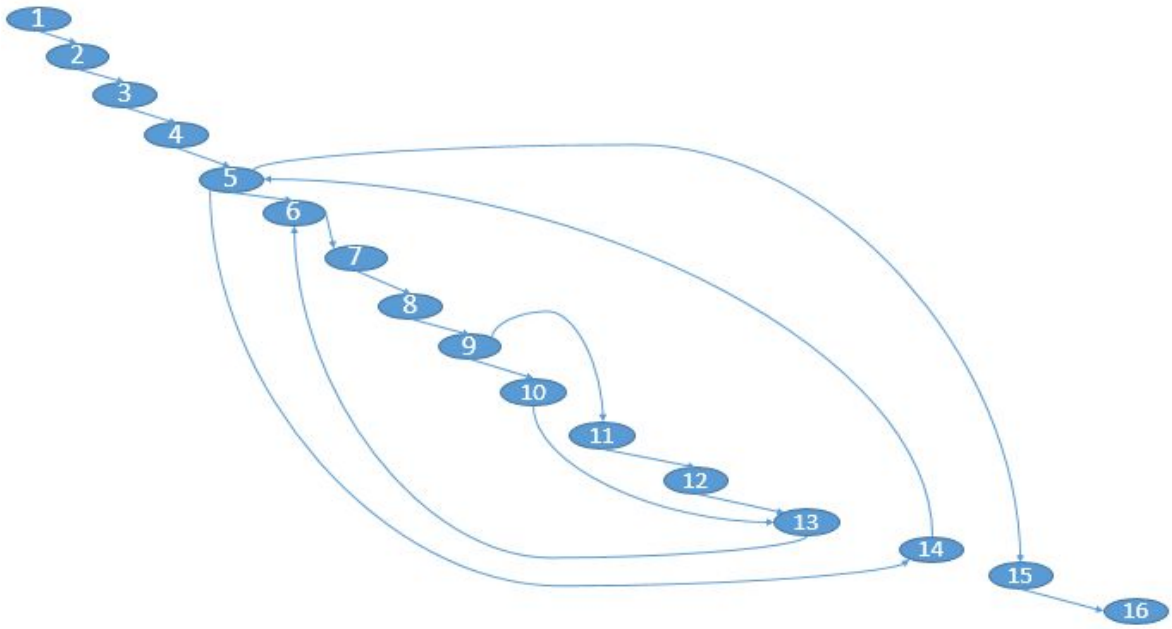


Figura 40 Grafo del flujo de camino básico

Se calcula la complejidad ciclomática V (G), obteniendo el resultado siguiente:

V (G) = Aristas (A) – Nodos (N) + 2
V (G)= 18-16+2

V (G)= 4

El valor V (G) expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 4 caminos:

Camino básico 1: 1-> 2 -> 3 ->4 -> 5 -> 15-> 16

Camino básico 2: 1-> 2 -> 3 ->4 -> 5 -> 6-> 14-> 5-> 15-> 16

Camino básico 3: 1-> 2 -> 3 ->4 -> 5 -> 6-> 7-> 8-> 9-> 11-> 12-> 13-> 6-> 14-> 5-> 15-> 16

Camino básico 4: 1-> 2 -> 3 ->4 -> 5 -> 6-> 7-> 8-> 9-> 10->13-> 6-> 14-> 5-> 15-> 16

Los casos de pruebas se definen a partir de los caminos básicos detectados, de forma que los datos señalados causen que se visiten las sentencias vinculadas a cada nodo del camino. Para este método se calcularon cuatro caminos básicos, por lo que se debe hacer igual número de casos de prueba, para aplicar las pruebas.

Tabla 13 Caso de prueba para el camino básico 3

Caso de prueba: Camino básico #3	
Entrada	Imagen con una amplia gama de colores.
Resultados Esperados	Se devuelve una imagen donde solo existen 2 colores, el blanco y el negro.
Condiciones	Se selecciona una imagen proveniente del PENTACAM.

3.5.2 Prueba de integración

Se decidió realizarle pruebas de integración al algoritmo debido a la necesidad de que el mismo se integre correctamente al PANDOC, para lograrlo se empleó la técnica **matriz de interdependencia**.

Para realizar esta prueba se debe tener en cuenta el nivel de complejidad en cada integración, en lo que influye mucho el grado de interdependencia (32). A

continuación, se muestra la matriz de interdependencia obtenida al integrar las clases Segmentacion y Editor.

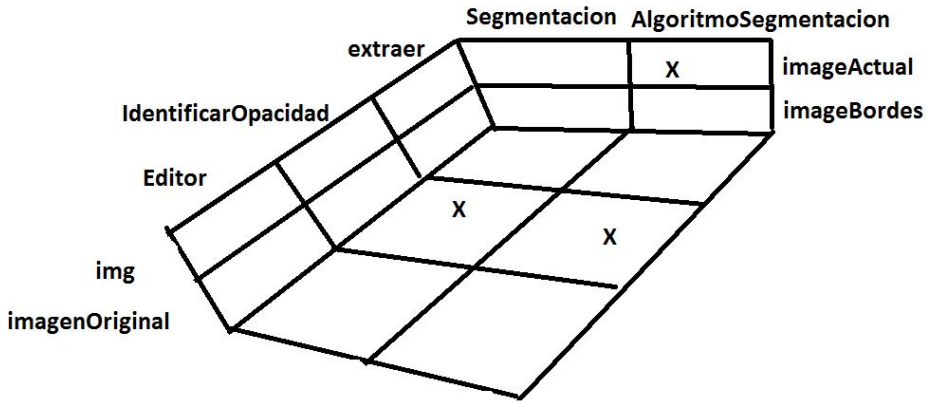


Figura 41 Tarjeta Matriz de interdependencia Segmentacion-Editor

En la matriz anterior se observan malas prácticas de programación que impiden seguir ejecutando las pruebas de integración: el método AlgoritmoSegmentacion(BufferedImage) accede directamente a los atributos de la clase Editor, lo cual está prohibido. Se debe, por tanto, tener un método get() para el atributo img de Editor. Se añadió el método y se realizó nuevamente la matriz de interdependencia entre las clases. El método añadido fue el siguiente:

```
public BufferedImage getBimg(){

return img;

}
```

A continuación, se muestra como queda luego de la adición del método la nueva matriz de interdependencia.

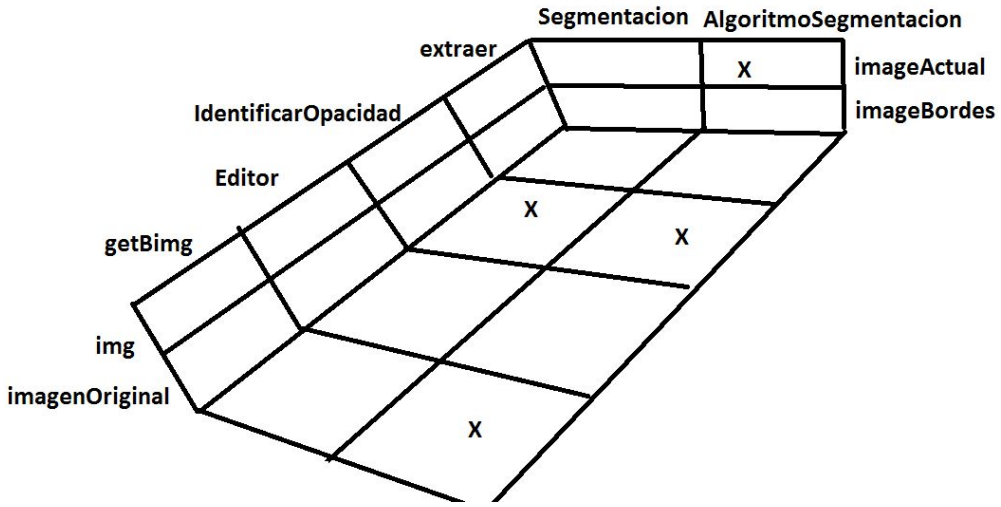


Figura 42 Tarjeta Matriz de interdependencia Segmentacion-Editor obtenida luego de añadir el método getBimg.

En esta nueva matriz de interdependencia se puede observar una mejor organización del código, donde los métodos de la clase Segmentacion para acceder a los atributos de la clase Editor utilizan los métodos get.

3.5.3 Prueba de aceptación

Las historias de usuario están asociadas a las pruebas de aceptación o pruebas funcionales, en ellas se describen las posibles formas de utilización del software. Las pruebas de aceptación no solo validan la transformación de una entrada en una salida, sino que validan una característica completa (23). Estas intentan encontrar casos de prueba en que el módulo no se ajusta a su especificación. Se enfocan especialmente en los módulos que se relacionan con la interfaz de usuario ya que no requieren el conocimiento de la estructura interna del programa para su puesta en marcha. Para realizar las mismas se empleó la técnica de caja negra, diseñándose 10 casos de pruebas que luego fueron probados en el sistema para comprobar la correcta implementación de los requisitos funcionales.

Como parte de estas pruebas se procedió a la creación de 10 casos de estudio para verificar los resultados del algoritmo y el tiempo de respuesta del mismo, en cada caso se tomaron imágenes que corresponden a diferentes pacientes (1, 2, 3…., 10). En esta tabla se presenta el tiempo de respuesta del software

PANDOC y del algoritmo desarrollado:

Tabla 14 Comparación de tiempo de respuesta entre el algoritmo del PANDOC y la propuesta de solución.

Herramientas	1	2	3	4	5	6	7	8	9	10	Media
Algoritmo del PANDOC	29	30	21	27	27	24	28	24	26	15	25.1
Propuesta de solución	13	15	21	19	14	12	16	18	14	19	16.1

A continuación, se presentan 3 casos de estudio de los 10 realizados donde se mostrará la opacidad según el especialista, después según el software PANDOC y posteriormente el de la solución propuesta. Las pruebas arrojaron los siguientes resultados:

1. Caso de estudio correspondiente a la imagen del ojo proveniente del PENTACAM del paciente “Paciente 1” tomado el 28/05/2016.

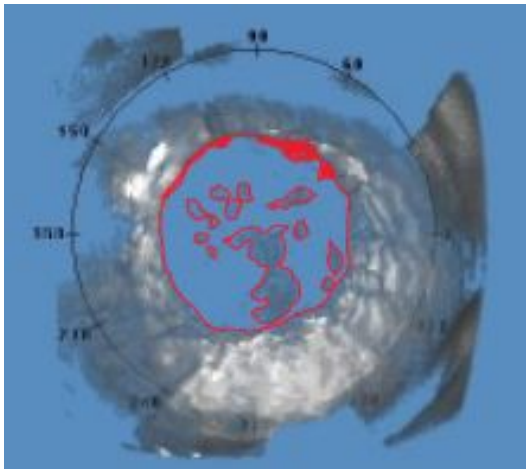


Figura 43 Caso 1 OCP según el especialista

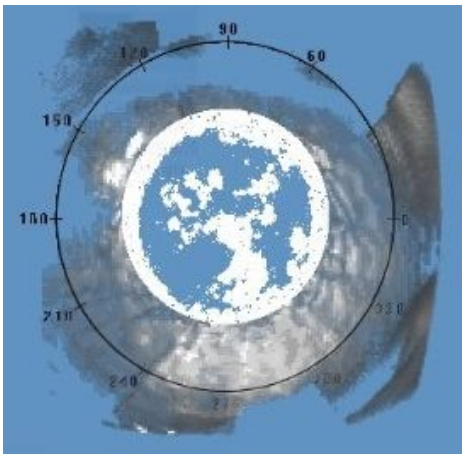


Figura 44 OCP según el Sistema Basado en Casos del PANDOC

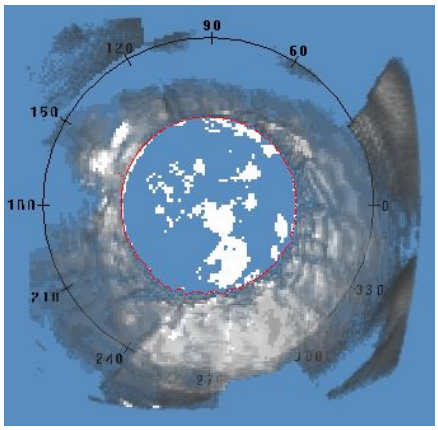


Figura 45 OCP según el algoritmo de segmentación del PANDOC

2. Caso de estudio correspondiente a la imagen del ojo proveniente del PENTACAM del paciente “Paciente 2” tomado el 28/05/2016

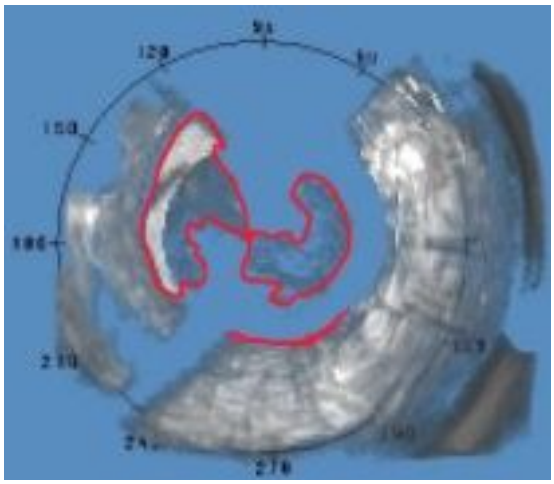


Figura 46 Caso 2 OCP según el especialista

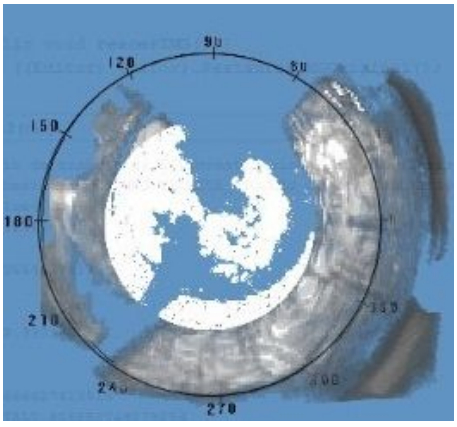


Figura 47 OCP según el Sistema Basado en Casos del PANDOC

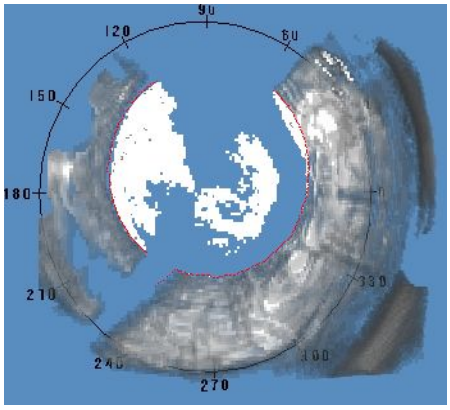


Figura 48 OCP según el algoritmo de segmentación del PANDOC

3. Caso de estudio correspondiente a la imagen del ojo proveniente del PENTACAM del paciente “Paciente 3” tomado el 28/05/2016

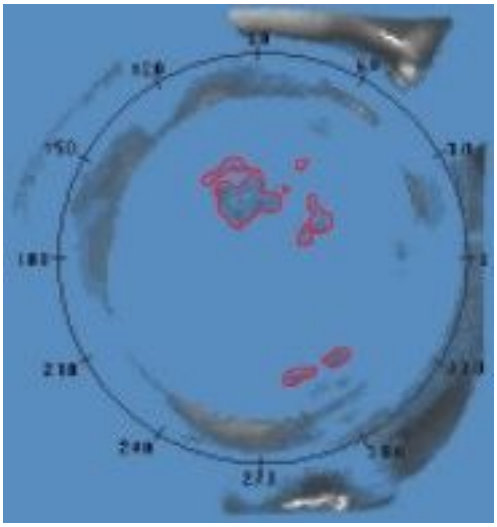


Figura 49 Caso 3 OCP según el especialista

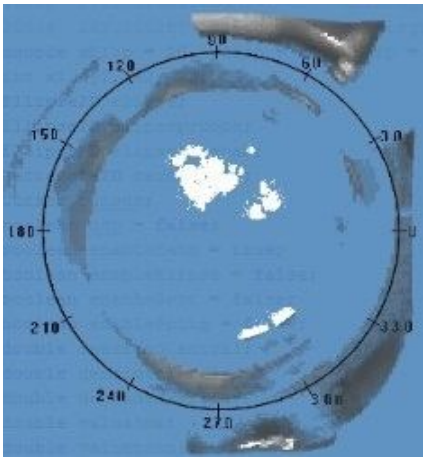


Figura 50 OCP según el Sistema Basado en Casos del PANDOC

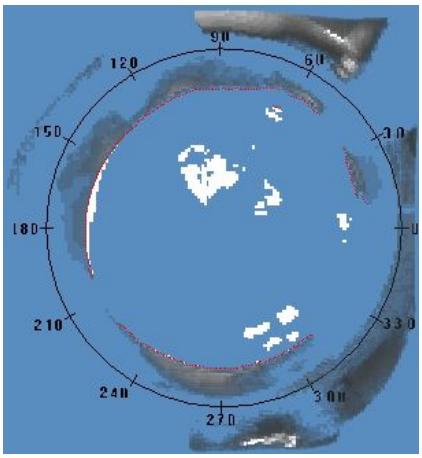


Figura 51 OCP según el algoritmo de segmentación del PANDOC

Los resultados de las comparaciones se muestran en la tabla 11, donde se presentan los resultados de los dos sistemas de identificación de la OCP en los 10 pacientes tomados como casos de estudio. En esta tabla cada caso está ejecutado por ambos algoritmos, dándose una evaluación entre 1 y 10 donde 10 es la identificación ideal de la OCP, esta evaluación es otorgada por especialistas del hospital general Pando Ferrer.

Tabla 15 Comparación de Resultados

	Algoritmo del software PANDOC	Propuesta de solución.
Paciente 1	6	10
Paciente 2	8	9
Paciente 3	6	10
Paciente 4	6	9
Paciente 5	9	10
Paciente 6	6	9
Paciente 7	10	10
Paciente 8	7	8
Paciente 9	7	10

Paciente 10	9	10
Media	7.4	9.5

Los resultados obtenidos con esta prueba fueron satisfactorios, obteniéndose por parte del cliente el acta de aceptación (Ver Anexo 2).

Las pruebas de aceptación se dividieron en 3 iteraciones atendiendo a las necesidades del cliente y se detectaron no conformidades no significativas que se centraron fundamentalmente en errores ortográficos, no conformidades significativas referentes a errores de validación, y varias recomendaciones. En la tabla 12 se muestran las iteraciones:

Tabla 16 Resultado de las pruebas de aceptación

Número de iteración	**No conformidades**		**Recomendaciones**	**No conformidades resueltas**
	Significativas	**No significativas**		
1	1	2	1	4
2	0	2	1	3
3	0	0	1	1

Las pruebas fueron realizadas de forma iterativa e incremental, y se demostró en cada iteración la corrección de los errores detectados en la iteración anterior, lo que ayudó a mejorar la calidad y funcionalidad del sistema, aprobando por parte del cliente la solución informática, quien evaluó todos los casos de prueba de forma satisfactoria.

3.6 Conclusiones parciales

- La aplicación de revisiones formales de los requisitos permitió asegurar la validez de los mismos.

- Las métricas de diseño, demostraron que las clases del diseño poseen bajo acoplamiento, que existe además una baja responsabilidad y complejidad de implementación y una alta reutilización en el diseño propuesto.
- Se aplicaron pruebas unitarias, de integración y de aceptación, las que permitieron verificar que el software es funcional y cumple las necesidades del cliente.

CONCLUSIONES GENERALES

1. El estudio de la técnica utilizada para la identificación de la OCP en el software PANDOC demostró que la misma no proporciona una respuesta correcta al objetivo general de la presente investigación, fomentando la necesidad de desarrollar el algoritmo propuesta.
2. La utilización de la metodología XP, los estándares de codificación y los patrones de diseño permitieron obtener un algoritmo que detecta la opacidad en imágenes provenientes del PENTACAM.
3. La aplicación de técnicas para validar los requisitos, métricas para validar el diseño y la realización de pruebas unitarias, de integración y de aceptación, permitió verificar que el algoritmo es funcional y cumple las necesidades del cliente.

RECOMENDACIONES

- Continuar con la investigación y alcanzar una segunda fase que es la cuantificación objetiva de la OCP.

BIBLIOGRAFÍA REFERENCIADA

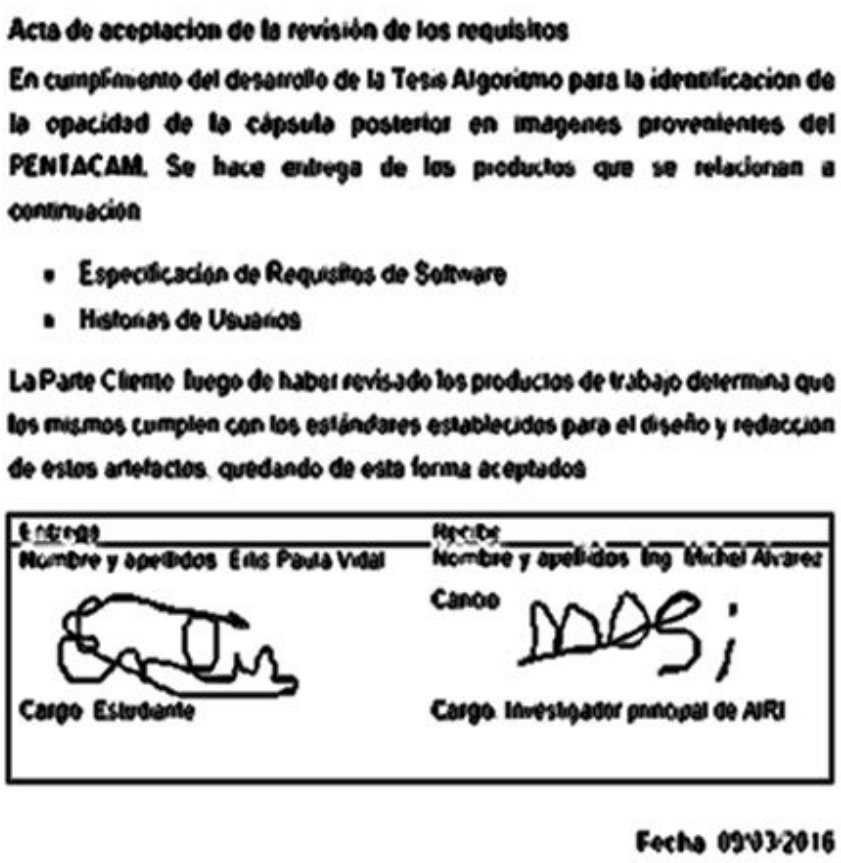
1. *Anillos de tensión capsular. Nuestra experiencia.* **Valentín Tinguaro Díaz Alemán, D. Perera Sanz, V. Lozano López, Javier Rodríguez Martín. 2005.** 16, 2005, Sociedad Canaria de Oftalmología.
2. *Assessment of systems of analyzing PCO.* **Tariq M Aslan, Niall Patton, Baljean Dhillon. 2005.** 2005, Journal of Cataract and Refractive Surgery, Vol. 31.
3. *Comparison of 4 methods for quantifying posterior capsule opacification.* **Oliver Findl, Wolf Buehl. 2003.** 2003, Journal of Cataract & Refractive Surgery, Vol. 29.
4. *Cuantificación objetiva de la opacidad de la cápsula posterior mediante tomogramas Scheimpflug del PENTACAM.* **López, Iván Hernández. 2011.** 2, Ciudad de la Habana : s.n., 2011, Revista Cubana de Oftalmología, Vol. 24.
5. *Digital image capture and automated analysis of posterior capsular opacification.* **D. S Friedman, D D Duncan,. 1999.** 8, 1999, Investigative Ophthalmology & Visual Science, Vol. 40.
6. *DISEÑO DE UN SISTEMA BASADO EN CASOS PARA LA IDENTIFICACIÓN DE OPACIDAD MEDIANTE EL PENTACAM.* **Álvarez Cancio Michel, Rodríguez Puente Rafael,Hernández Lopez Iván. 2014.** La Habana : UCIENCIA, 2014.
7. **Elizondo, M. C. José Jaime Esqueda. 2002.** *Fundamentos de Procesamiento de Imágenes.* California : CONATEC 2002 , 2002.
8. *Estrategias de prevención de la opacidad de la cápsula posterior.* **Iván Hernández López, Juan Raúl Hernández Silva, Yadira Castro González, Ailén Garcés Fernández, Zucell Veitía Rovirosa, Eneida Pérez Candelaria. 2010.** Ciudad de la Habana : s.n., 2010, Revista Cubana de Oftalmología, Vol. 23.
9. *Digital Image Processing,* **Gonzales, RC. 2002.**. 2002.
10. *IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN.* **JUAN PABLO URREA, EMMANUEL OSPINA. 2004.** 24, 2004.
11. *Intervenciones para la prevención de la opacificación de la cápsula posterior.* **Oliver Findl, Wolf Buehl, Peter Bauer, Thomas Sycha. 2010.** 2, s.l. : Biblioteca Cochrane Plus, 2010.
12. *Métodologías Ágiles en el Desarrollo de Software.* **Canós José H, Letelier Patricio, Panadés M Carmen. 2003.** Universidad Politécnica de Valencia : s.n., 2003.
13. *Metodologías del desarrollo de software.* **Sánchez Mendoza, Maria A. 2004.** 2004.
14. *Métodos de segmentación de nubes en imágenes médicas.* **ERNESTO GÓMEZ VARGAS, NELSON OBREGÓN NEIRA, DIEGO FERNANDO ROCHA ARANGO. 2013.** Bogotá : s.n., 2013.
15. **Oracle.** Java. *http://www.java.com/es/about/.* [En línea] [Citado el: 14 de 1 de 2015.] http://www.java.com/es/about/..
16. *PANDOC: SOFTWARE PARA LA CUANTIFICACIÓN OBJETIVA DE LA OPACIDAD DE LA CÁPSULA POSTERIOR MEDIANTE TOMOGRAMAS*

SCHEIMPFLUG DEL PENTACAM. **Michel Alvarez Cancio, Adrián Hernández Barrios,Rafael Rodríguez Puentes, Iván Hernández López. 2013.** La Habana : s.n., 2013.

17. **Paradigm, Visual. 2016.** Visual Paradigm. *Visual Paradigm.* [En línea] 2016. [Citado el: 30 de marzo de 2016.] https://www.visual-paradigm.com.
18. *PENTACAM tomograms: A Novel Method for Quantification of Poste-.* **D Grewall, R Jain. 2008.** 5, 2008, Vol. 49.
19. *Photographic image analysis system of posterior capsule opacification.* **Manfred R Tetz, Gerd U Gerd U, Martina Sperker, Marcus Blum, Hans E Völcker. 1997.** 10, 1997, Journal of Cataract & Refractive Surgery, Vol. 23.
20. *Quantification of posterior capsular opacification in digital images after cataract surgery.* **SA, Barman. 2000.** San Diego, California : s.n., 2000, Vol. 3979.
21. *Reproducibility of standardized retroillumination photography for quantification of posterior capsule opacification.* **Wolf Buehl, Oliver Findl. 2002.** 2, 2002, Journal of Cataract & Refractive Surgery, Vol. 28.
22. *Revisión sistemática de estudios poblacionales de prevalencia de cataratas.* **Acosta R, Hoffmeister L, Roman R, Comas M, Castilla M, Castells X. 2006.** Barcelona, España : s.n., 2006, Vol. 81. 509-516.
23. **Sommerville, Ian. 2005.** *Ingeniería del software. Séptima Edición.* Madrid. España : Pearson Educación. S. A., 2005. 84-7829-074-5.
24. *Thresholding Using an Illumination Model.* **Parker, J. R. 1991.** Canada : s.n., 1991, Vol. 13.
25. *Una explicación de la programación extrema: aceptar el cambio .* **Beck, Kent. 2002.** s.l. : **Addison Wesley**, 2002.
26. **Moler, Cleve. 1982.** MathWorks. *MathWorks.* [En línea] 1982. [Citado el: 15 de 3 de 2016.] *http://www.mathworks.com.*
27. *Técnicas de Segmentación en Procesamiento Digital de Imágenes : aceptar el cambio .* ***Dra. Nora La Serna Palomino, Lic. Ulises Román Concha****, 2009*
28. *Una nueva metodología de segmentación de imágenes basadas en Contornos Activos. Aplicación de la segmentación de imágenes médicas para la reconstrucción tridimensional de estructuras anatómicas.* ***Dc Valentín Masero Vargas****, 2015.*
29. *SEGMENTACIÓN DE IRIS MEDIANTE´CONTORNOS ACTIVOS.* ***Lic Rocío Sanchiz Redondo****, 2011.*
30. *PROCESAMIENTO DE IMÁGENES.* ***Lic Diego González Aguilera****, 2011.*
31. *Design Patterns (Elements of Reusable Object-Oriented Software),* ***Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides****. 2003.*
32. *EJEMPLO DE PRUEBAS UNITARIAS Y DE INTEGRACIÓN,* ***Lic Macario Polo Usaola****, 2009.*
33. *Ingeniería del Software. Un Enfoque Práctico, Quinta Edición. España : McGraw-Hill, 2002 .****Pressman, Roger S****, 2002.*

ANEXOS

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



Acta de aceptacion de la revisión de los requisitos

En cumplimiento del desarrollo de la Tesis Algoritmo para la identificacion de la opacidad de la cápsula posterior en imagenes provenientes del PENTACAM. Se hace entrega de los productos que se relacionan a continuación

- Especificacion de Requisitos de Software
- Historias de Usuarios

La Parte Cliente luego de haber revisado los productos de trabajo determina que los mismos cumplen con los estándares establecidos para el diseño y redaccion de estos artefactos, quedando de esta forma aceptados

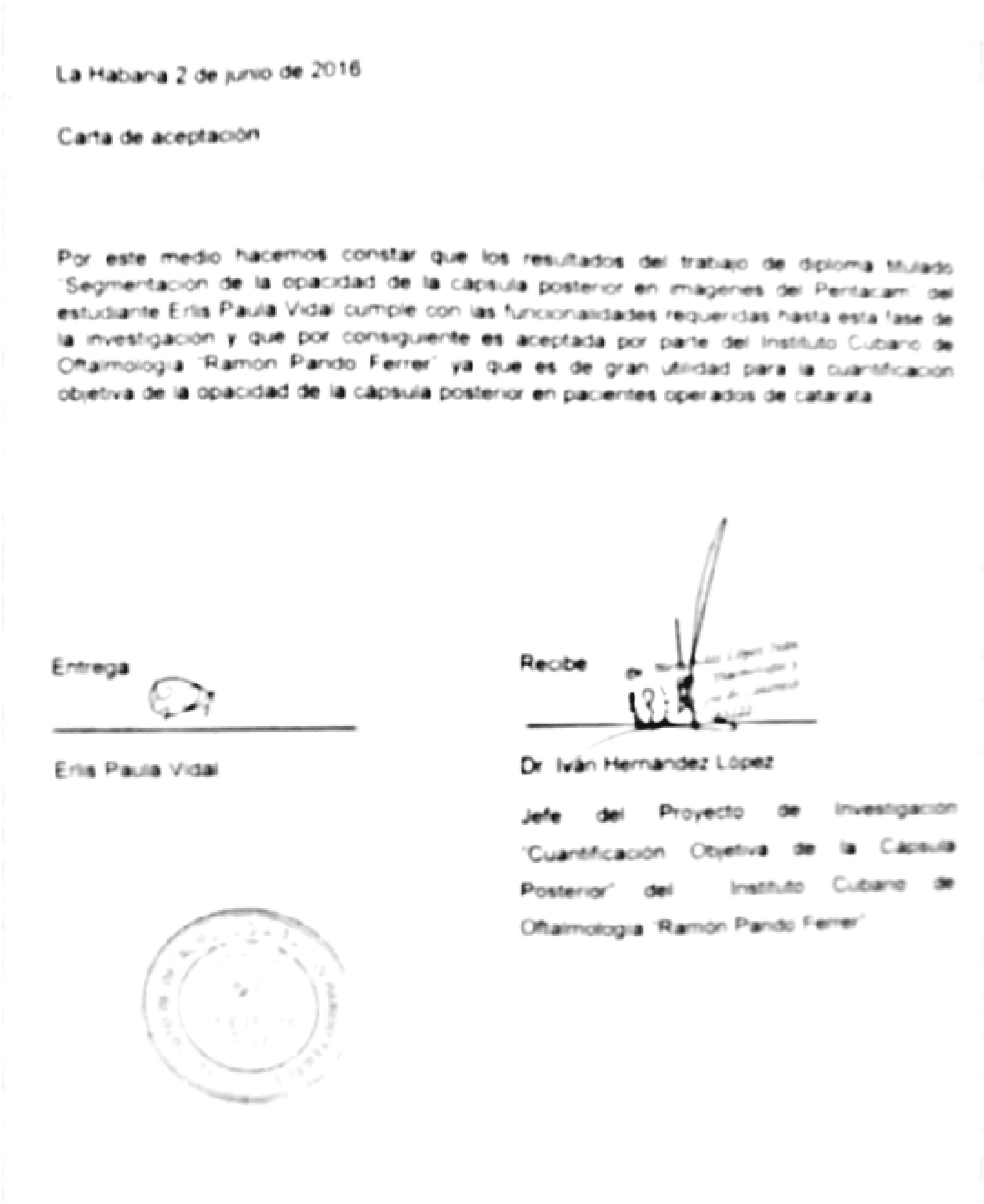
Entrega
Nombre y apellidos Elis Paula Vidal
Cargo Estudiante

Recibe
Nombre y apellidos Ing Michel Alvarez
Cargo
Cargo Investigador principal de AIRI

Fecha 09/03/2016

Figura 52 Aceptación de requisitos

Anexo 2: Carta de aceptación.



La Habana 2 de junio de 2016

Carta de aceptación

Por este medio hacemos constar que los resultados del trabajo de diploma titulado "Segmentación de la opacidad de la cápsula posterior en imágenes del Pentacam" del estudiante Erlis Paula Vidal cumple con las funcionalidades requeridas hasta esta fase de la investigación y que por consiguiente es aceptada por parte del Instituto Cubano de Oftalmología "Ramón Pando Ferrer" ya que es de gran utilidad para la cuantificación objetiva de la opacidad de la cápsula posterior en pacientes operados de catarata

Entrega

Erlis Paula Vidal

Recibe

Dr. Iván Hernández López

Jefe del Proyecto de Investigación "Cuantificación Objetiva de la Cápsula Posterior" del Instituto Cubano de Oftalmología "Ramón Pando Ferrer"

Figura 53 Carta de aceptación