



Universidad de las Ciencias Informáticas
Grupo de Investigación: AIRI
Facultad 3

Módulo para la validación de la segmentación en imágenes de opacidad de la cápsula posterior provenientes del Pentacam.

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

Autor:

Ruben Hayron Rodríguez Espinosa

Tutor:

Ing. Michel Álvarez Cancio

La Habana. Junio, 2017.

Declaratoria de autoría

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

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

Ruben Hayron Rodríguez Espinosa

Autor

Ing. Michel Álvarez Cancio

Tutor

Dedicatoria

A toda mi familia por haberme ayudado en todos estos años.

Agradecimientos

A mi madre por el amor y la dedicación que me ha brindado en estos 5 años.

A mis abuelos y a mi tía, por su entrega constante y apoyo incondicional.

A mi tutor Michel Álvarez Cancio por haberme acogido como diplomante y ofrecerme su ayuda.

Al grupo de Investigación AIRI por sus críticas constructivas.

A todos mis compañeros de año por brindarme su apoyo en el transcurso de la carrera.

A todo el que de manera general contribuyó a mi desarrollo y formación como ingeniero incluyendo a la Revolución Cubana que me dio la posibilidad de desarrollarme de forma intelectual.

Resumen

La opacidad de la cápsula posterior es actualmente uno de los aspectos más importantes en la cirugía de catarata de los tiempos modernos. Sigue siendo la complicación posoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan a importantes repercusiones sociales, médicas y económicas. En nuestra Universidad se han desarrollado varios algoritmos para cuantificar la Opacidad de la Cápsula Posterior en imágenes del PENTACAM; ellos han logrado diversos resultados, aunque no se ha concretado cuál identifica mejor este tipo de complicación postoperatoria. Por tal motivo la presente investigación tiene como objetivo la creación de un módulo para validar la segmentación realizada por un algoritmo atendiendo a una segmentación de referencia brindada por el usuario, con el fin de mostrar el resultado de varias métricas usadas en la literatura y llevar a cabo la validación del mismo. Se implementaron las medidas estadísticas de error (precisión, exactitud, revocación, primer plano, relación señal ruido, tasa de error y tasa negativa), métricas de localización (error medio, error cuadrático medio y la figura de mérito de Pratt) y métricas de Hausdorff (distancia de Hausdorff y distancia de Baddeley), logrando así una evaluación objetiva del grado de exactitud de la segmentación. Con la aplicación de estas métricas sobre los métodos existentes se pudo constatar la eficacia del método propuesto por Paula Vidal en el 2016.

Palabras claves: Validación de la segmentación, medidas estadísticas de error, medidas de localización, métricas de Hausdorff, opacidad de la cápsula posterior.

ABSTRACT

The opacity of the posterior capsule is currently one of the most important in cataract surgery in modern times. It remains the most common late postoperative complication following cataract surgery associated with decreased visual acuity, deterioration of contrast sensitivity, and glare problems leading to significant social, medical and economic repercussions. In our University several algorithms have been developed to quantify the Opacity of the Posterior Capsule in PENTACAM images; They have achieved several results, although it has not been determined which best identifies this type of postoperative complication. For this reason, the present research aims at the creation of a module to validate the segmentation performed by an algorithm according to a reference segmentation provided by the user, in order to show the result of several means used in the literature and lead to Validation of the same. The statistical measures of error (precision, accuracy, revocation, foreground, signal noise ratio, error rate and negative rate), location metrics (mean error, mean square error and merit figure of Pratt) and Hausdorff metrics (Hausdorff distance and Baddeley distance), thus achieving an objective assessment of the degree of accuracy of the segmentation. With the application of these metrics on the existing methods it was possible to verify the effectiveness of the method proposed by Paula Vidal in 2016.

Keywords: Cataract surgery, opacity of the posterior capsule, the Pentacam images.

ÍNDICE DE CONTENIDO

Declaratoria de autoría	2
Dedicatoria	3
Agradecimientos	3
Resumen.....	4
ÍNDICE DE CONTENIDO.....	6
ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS	8
Introducción.....	10
Capítulo 1. Validación de la segmentación en imágenes médicas: Bases Conceptuales.	14
1.1 Procesamiento digital de imágenes	14
1.2 Imágenes médicas.....	15
1.3 Segmentación.....	15
1.4 Validación de la segmentación	22
1.4.1 Métricas para segmentación de imágenes	22
1.4.2 Medidas del error	24
1.5 Programa analizador de Opacidad Capsular	30
1.6 Herramientas usadas durante el desarrollo.....	32
1.6.1 Metodología de desarrollo de Software	32
1.6.2 Lenguaje de Programación	34
1.6.3 Entorno integrado de desarrollo	35
1.6.4 Herramienta para el modelado	36
1.6.5 Sistema Gestor de Base de Datos.....	36
1.6.6 Patrones de diseño	38
1.7 Conclusiones del capítulo	38
Capítulo 2. Validación de la segmentación en imágenes del Pentacam: Propuesta de solución.....	39
2.1 Descripción de la solución	39
2.2 Fase de Planificación.....	42
2.2.1 Historias de Usuario	42
2.2.2 <i>Estimación de esfuerzos por Historias de Usuario</i>	43
2.2.3 Requisitos No Funcionales	44

2.2.4 Plan de iteraciones.....	44
2.2.5 Plan de entrega.....	46
2.3 Diseño de la propuesta de solución.....	46
2.3.1 Tarjetas CRC.....	46
2.3.3 Estilos de codificación.....	47
2.3.2 Patrones de diseño utilizados.....	48
2.4 Conclusiones del capítulo.....	49
Capítulo 3. Validación y resultados del sistema.....	50
3.1 Fase de Implementación.....	50
3.2 Interfaz del Programa analizador de segmentación de imágenes.....	53
3.3 Pruebas.....	56
3.3.1 Pruebas unitarias.....	57
3.3.2 Pruebas de aceptación.....	60
3.4 Resultados.....	64
3.5 Validación de los resultados.....	66
3.6 Conclusiones del capítulo.....	67
Conclusiones generales.....	68
Recomendaciones.....	68
Glosario de términos.....	69
Referencias.....	70
Anexos.....	75
Anexo #1: Historias de usuario.....	75
Anexo #2: Pruebas de aceptación.....	79

ÍNDICE DE FIGURAS

Figura 1: Máscara Binomial de 3X3 (24).....	17
Figura 2: Ejemplo de condiciones de comparación de segmentación (38).....	23
Figura 3: Ejemplo de variables de evaluación (40).....	25
Figura 4: Ejemplo de precisión (40).....	26
Figura 5: Ejemplo de exactitud (40).....	26
Figura 6: Software PANDOC para la cuantificación de opacidad capsular.....	31
Figura 7: Diagrama de flujo del sistema.....	40

Figura 8: Clase Imagen	48
Figura 9: Creación de instancias	49
Figura 10: Interface principal del software PASI.....	54
Figura 11: Panel de edición del software PASI.....	55
Figura 12: Ventana de visualización de los resultados de las métricas.....	56
Figura 13: Algoritmo para binarizar las imágenes de entrada.....	58
Figura 14: Diagrama de flujo del algoritmo “binarizarBufferedImage”.....	58
Figura 15: A la izquierda el “Ground Truth” y al lado la segmentación automática.....	62
Figura 16: A la izquierda el “Ground Truth” y al lado la segmentación automática	62
Figura 17: A la izquierda el “Ground Truth” y al lado la segmentación automática	63

ÍNDICE DE TABLAS

Tabla 1: Historia de usuario # 8.....	43
Tabla 2: Puntos de estimación por historias de usuario.....	43
Tabla 3: Plan de duración de las iteraciones	45
Tabla 4: Plan de entrega	46
Tabla 5: Tarjeta CRC de la clase NewJFrame.....	47
Tabla 6: Tiempo de implementación de las HU de la iteración #1	51
Tabla 7: Tarea de ingeniería de la funcionalidad Cargar imagen de tomograma.	51
Tabla 8:Tarea de ingeniería de la funcionalidad Reconocer imagen de tomograma.....	52
Tabla 9: Tarea de ingeniería de la funcionalidad Segmentar imagen “ground truth”.	52
Tabla 10: Tiempo de implementación de las HU de la iteración #2	52
Tabla 11: Tiempo de implementación de las HU de la iteración #3	53
Tabla 12: Caminos básicos	60
Tabla 13: Caso de prueba del camino básico #1	60
Tabla 14: Caso de prueba de aceptación de la HU #7	61
Tabla 15: Resultados de las pruebas del caso de estudio #1	62
Tabla 16: Resultados de las pruebas del caso de estudio #1	63
Tabla 17: Resultados de las pruebas del caso de estudio #1	63
Tabla 18: Resultado de las pruebas de aceptación	64
Tabla 19: Resultado de las métricas para el algoritmo de Alonso P y Machado S.....	65
Tabla 20: Resultado de las métricas para el algoritmo de Rodríguez M	65

Tabla 21:Resultado de las métricas para el algoritmo de E. Paula V.....	65
Tabla 22: Resultado de los tiempos de complejidad media por algoritmo.....	66
Tabla 23: Evaluación de los expertos por imágenes.....	67
Tabla 24:Historia de usuario #1.....	75
Tabla 25:Historia de usuario #2.....	75
Tabla 26:Historia de usuario #3.....	76
Tabla 27:Historia de usuario #4.....	76
Tabla 28:Historia de usuario #5.....	76
Tabla 29:Historia de usuario #6.....	77
Tabla 30:Historia de usuario #7.....	77
Tabla 31:Historia de usuario #9.....	78
Tabla 32:Historia de usuario #10.....	78

Introducción

En la actualidad miles de personas de todo el mundo padecen de la enfermedad de cataratas, sufriendo como consecuencia la pérdida total o parcial de la visión. La catarata es la opacidad en el cristalino del ojo, trae como consecuencia que la luz se disperse dentro del mismo y no sea posible su enfoque en la retina, de esta forma se crean imágenes difusas. La misma genera problemas con la apreciación de cambios de colores y cambios de contrastes. Es la causa más común de ceguera tratable con cirugía. Tiene diversas causas, pero se le atribuye mayormente a la edad, este proceso se acelera si el paciente padece de enfermedades como la diabetes o la hipertensión. Con frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad (1).

Tras una intervención quirúrgica el paciente puede recuperar su visibilidad total o parcialmente, pero no en todos los casos la cirugía es un éxito a largo plazo, pues en ocasiones los pacientes pueden presentar complicaciones postoperatorias. La complicación postoperatoria tardía más frecuente a largo o mediano plazo es la Opacidad de la Cápsula posterior (OCP). La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata. Para garantizar la recuperación visual del paciente es imprescindible la transparencia de esta estructura. Su incidencia en la actualidad se encuentra entre 0,7% y 47,6 % en los primeros cinco años de realizada la cirugía, según un análisis de 90 estudios publicados, mientras que en Cuba asciende hasta el 50% de los casos (2). En la comunidad médica internacional no existe consenso alguno acerca de la correcta cuantificación de la OCP, se deja a cada especialista la elección de alguna de las técnicas propuestas (2).

El Pentacam es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo. El análisis de estas imágenes resultantes es de gran utilidad para la cuantificación objetiva de la OCP. Una ventaja notable que poseen las imágenes provenientes del Pentacam es la no presencia de destellos de luz que interfieran con el análisis de las mismas (3).

Las imágenes provenientes del PENTACAM pueden poseer regiones de OCP tan pequeñas que se hace difícil su detección a simple vista, lo que provoca un mal análisis por parte del especialista sobre esta enfermedad. En la actualidad existen diversos softwares que identifican de forma automática la OCP, destacándose entre ellos el

PANDOC, el cual fue desarrollado en el Grupo de Investigación AIRI (Artificial Intelligence: Research and Innovation) de la Universidad de las Ciencias Informáticas (UCI) en colaboración con el Instituto Cubano de Oftalmología (ICO) “Ramón Pando Ferrer”.

PANDOC es un software para identificar regiones con OCP, mediante el uso de las imágenes provenientes del PENTACAM y evaluar su correlación con la graduación 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 (4).

Para la identificación automática de la opacidad por el software PANDOC, en el año 2015 se realizó una propuesta fundamentada por un sistema basado en casos (SBC) (5) (6). En esta propuesta cada caso a identificar era un pixel, donde se analizaban sus características de color y las de sus vecinos, así como también su cercanía con respecto al centro de la estructura ocular.

También se realizó una variante en el 2016 que utilizaba un SBC, pero en esta ocasión con el objetivo principal de minimizar tiempo de ejecución con respecto al SBC anterior. En esta versión se usan como casos, regiones que fueron previamente extraídas usando un algoritmo de componentes conexas (7), entre los rasgos de estas regiones seguía estando el de cercanía al centro de la estructura ocular debido a la importancia de este rasgo para la decisión del SBC (8).

En el propio 2016 se propone otra variante utilizando mayormente técnicas de procesamiento digital de imágenes, en esta ocasión buscando los bordes de la Capsulorrexix¹ (9), esto bajo la premisa de que toda variación de la imagen de fondo en la región comprendida por la Capsulorrexix es OCP. Esta solución está basada en una combinación de filtros espaciales, la transformada circular de Hough y un algoritmo de contornos activos (9) (10).

Estos algoritmos desarrollados han tenido como objetivo cuantificar OCP en tomogramas Scheimpflug del PENTACAM logrando diversos resultados, principalmente en cuanto a calidad de la segmentación y tiempo de ejecución. Pero, hasta el momento no se ha concretado cuál identifica mejor este tipo de complicación postoperatoria. Por tal motivo la presente investigación tiene como objetivo la creación de un módulo para validar la

¹ Capsulorrexix: apertura en forma de rotura circular de la cápsula anterior del cristalino con la finalidad de acceder a la *catarata*.

segmentación realizada por un algoritmo atendiendo a la imagen segmentada manualmente por el especialista “*ground truth*” y la devolución del resultado de varias métricas usadas en la literatura para poder realizar la validación del mismo.

Una vez culminada la etapa de Análisis se dio paso a la próxima etapa de desarrollo donde se identifica el siguiente **problema a resolver**: ¿Cómo validar la segmentación de opacidad de la cápsula posterior en imágenes provenientes de tomogramas Scheimpflug del PENTACAM?

Objeto de estudio: Procesamiento Digital de Imágenes.

Campo de Acción: Validación de la Segmentación de Imágenes Digitales.

Objetivo general: Desarrollar una herramienta para la validación de los algoritmos de segmentación en imágenes provenientes de tomogramas Scheimpflug del PENTACAM.

Dado el objetivo general expuesto se describen los siguientes objetivos específicos.

Objetivos específicos:

- Elaborar el marco teórico de la investigación mediante el estudio de las técnicas actuales de segmentación digital de imágenes.
- Identificar los requisitos funcionales y no funcionales para el desarrollo de la propuesta de solución.
- Diseñar un módulo para la validación de la segmentación en imágenes del Pentacam.
- Implementar los elementos definidos en la etapa del diseño.
- Validar la solución propuesta mediante pruebas unitarias y de aceptación.

Teniendo en cuenta el problema a resolver se formuló la siguiente Idea a Defender:

Idea a defender: El desarrollo de un módulo para validar la segmentación de los algoritmos de detección de opacidad capsular en imágenes de tomogramas Scheimpflug del PENTACAM, permitirá la selección del más apropiado para medir la incidencia de la OCP en pacientes sometidos a cirugía de catarata.

Para dar cumplimiento al objetivo de la investigación se trazan las siguientes tareas a cumplir:

Tareas a cumplir:

- Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
- Revisión de las técnicas de binarización en imágenes médicas.
- Revisión de las técnicas de validación de la segmentación en imágenes médicas.
- Estudio de medidas estadísticas para la validación de la segmentación.
- Descripción de los pasos a seguir para el uso de un sistema propuesto.
- Estudio de una metodología para el desarrollo de aplicaciones de este tipo.
- Diseñar la solución propuesta en función de los requisitos especificados.
- Implementación de las técnicas de validación de la segmentación seleccionadas.
- Implementación de las medidas estadísticas para el sistema de validación de la segmentación.
- Implementación del mecanismo de selección y binarización seleccionado.
- Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

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

Experimento: Gracias a este método se pudieron probar las distintas técnicas y algoritmos para la validación de la segmentación en imágenes médicas.

Análisis-Síntesis: Se efectuó la búsqueda y análisis de los algoritmos, teorías y documentos relacionados con el tema y a partir de esto se seleccionaron los elementos claves para el desarrollo de este trabajo.

Histórico-Lógico: Gracias a este método se realizó un estudio con el fin de conocer la evolución y desarrollo de las técnicas existentes en la actualidad enfocadas a la validación de la segmentación en imágenes médicas.

Capítulo 1. Validación de la segmentación en imágenes médicas: Bases Conceptuales.

En este capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación. Como puntos fundamentales se abordan los principales conceptos asociados al dominio del problema, como la incidencia de la OCP. También se realiza un estado del arte de las técnicas más usadas en la validación de la segmentación de imágenes médicas, las que ayudarán en la identificación de diferencias en la segmentación automática de las mismas.

1.1 Procesamiento digital de imágenes

El procesamiento digital de imágenes (PDI) es el proceso, de almacenamiento, transmisión y representación de información, de imágenes digitales por medio de una computadora. El PDI consta de 5 etapas las cuales se definen a continuación (11):

1. **Adquisición de la imagen:** Es el proceso de captura y almacenamiento de una imagen.
2. **Pre-procesamiento:** Incluye técnicas tales como la reducción de ruido y realce de detalles se realiza para que el resultado sea más apropiado para una aplicación en particular, dichas técnicas consisten en la mejora de la nitidez o aclarado de las imágenes desenfocadas, eliminación de ruido, mejora del contraste, mejora del ruido y detención de bordes.
3. **Segmentación de la imagen:** Subdivide la imagen en partes o aísla ciertos objetos de la misma. Tareas habituales: búsqueda y selección de formas determinadas en la imagen.
4. **Representación y descripción:** Mediante los procesos de descripción se obtienen características (tamaño, perímetro, etc.) convenientes para diferenciar un objeto de otro.
5. **Reconocimiento e interpretación:** El reconocimiento es el proceso que identifica los objetos, y la interpretación le asocia un significado a un conjunto de objetos reconocidos. Generalmente es conveniente agrupar estas áreas de acuerdo con la complicación y el grado de detalles que lleva aparejada su implementación.

De acuerdo a las etapas mencionadas, existe un conjunto de técnicas aplicables a imágenes médicas que se relacionan de forma directa cada una con su etapa.

1.2 Imágenes médicas

El desarrollo y avance de las tecnologías de obtención de imágenes médicas está revolucionando la medicina. Estas imágenes juegan un papel importante en el diagnóstico y tratamiento de muchas enfermedades, debido a que permiten la obtención de información vital del interior del cuerpo humano de forma no invasiva. En consecuencia, la comunidad de analistas de imágenes se ha sumergido en la búsqueda de la solución del problema de extracción de información clínica útil sobre estructuras anatómicas a partir de imágenes de Tomografía Computarizada (CT – *Computer Tomography*), Resonancia Magnética (MR – *Magnetic Resonance*) y otras modalidades con ayuda de la computadora (12) (13). Los algoritmos dedicados a la extracción de información a partir de imágenes, son conocidos como “algoritmos de segmentación de imágenes”, estos juegan un papel significativo en numerosas aplicaciones biomédicas con tratamiento de imágenes médicas anatómicas ya que ayudan a la toma de decisiones y mejoran la calidad del diagnóstico médico (12).

1.3 Segmentación

La segmentación en el campo del procesamiento digital de imágenes es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos cuyo objetivo es simplificar y cambiar la representación de una imagen en otra más significativa y más fácil de analizar (14). En el caso de las imágenes médicas 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 (15).

De manera general, los métodos de segmentación, se clasifican de la siguiente forma (16):

- **Métodos basados en la comparación de umbrales en el histograma**, en los cuales se obtiene un umbral de comparación para el agrupamiento de los píxeles.
- **Métodos basados en la detección de discontinuidades**, en donde se calculan las derivadas de la función de la imagen, éstos dividen la imagen a partir de cambios bruscos de los niveles de grises.
- **Métodos basados en la propiedad de similitud de los valores de los niveles de grises**, que permiten la agrupación de puntos a partir de ciertos criterios de homogeneidad.

- **Métodos heurísticos de segmentación**, los cuales se basan en el conocimiento previo de la imagen a segmentar y en la experiencia del observador, ellos incluyen en muchas ocasiones los métodos supervisados de segmentación.

De estas clasificaciones existen algoritmos que por ellos mismos o combinados han sido utilizados para la identificación de OCP en imágenes de tomogramas Scheimpflug del PENTACAM.

Transformada de Hough

La transformada de Hough es un algoritmo usado en reconocimiento de patrones que permite aislar características de forma particular dentro de una imagen. La idea principal es encontrar curvas que puedan ser parametrizadas como líneas rectas, polinomios y círculos, es decir, que si se tiene un segmento de línea, esta se representaría con los parámetros p y θ , donde p es la distancia entre la línea y el origen, y θ es el ángulo del vector desde el origen al punto más cercano de la línea, la ecuación de la recta se podría escribir de la siguiente forma (17):

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

Una de las variantes de la Transformada de Hough es la circular, la cual se utiliza para la detección de formas circulares, las cuales pueden ser descritas mediante su centro y radio (18), donde, sea $c1$ y $c2$ coordenadas del centro de la circunferencia y $c3$ radio de la circunferencia para cada punto del plano formado por los ejes x e y , al igual que para $c1$ y $c2$ se calcula el valor del radio de la circunferencia $c3$ actualizándose posteriormente el acumulador, que es una sub-divisiones del espacio paramétrico de la forma $A(i,j,k)$, en este caso $a(c1,c2,c3)$. En otras palabras el algoritmo para la variante de la transformada de Hough para detección de círculos al pasarle el valor radio a la transformada de Hough buscará círculos del radio pasado por parámetro, obteniendo el mayor acumulador (19).

Detección de bordes

La detección de borde es una parte esencial del procesamiento digital de imágenes, su función es identificar los límites de las regiones homogéneas en una imagen basándose en propiedades como la intensidad y la textura. Muchos algoritmos de detección de bordes han sido desarrollados sobre la base del cálculo de la intensidad del vector gradiente el cual en general tiene en cuenta al ruido de la imagen (20).

Esta técnica de procesamiento de imágenes se utiliza para la búsqueda de los límites de los objetos dentro de la misma. Su funcionamiento se basa en la detección de discontinuidades en los niveles de intensidad o brillo. La detección de bordes se utiliza para la segmentación de imágenes y extracción de datos en áreas como procesamiento de imágenes, visión por computadora, y la visión de máquina (21).

Entre los problemas que presenta la detección de bordes al igual que otras técnicas en el campo del procesamiento digital de imágenes se encuentra el ruido, el cual forma parte de la información no deseada en la imagen en sí, pudiendo tener su existencia diversos orígenes.

Un borde, como parte de una imagen, es considerado como un tipo de discontinuidad, por lo que puede ser detectado usando derivadas de primer y segundo orden. La derivada de primer orden para una imagen es el gradiente, mientras que la segunda derivada tiene una respuesta más fuerte ante detalles como puntos aislados o líneas (22) .

Operador Sobel

Este operador es capaz de detectar los bordes horizontales y verticales de forma separada sobre una imagen en escala de grises. El Operador Sobel calcula el gradiente de la intensidad de cada punto en una imagen, dando para cada punto la magnitud del mayor cambio posible, la dirección de éste y el sentido desde los tonos oscuro a claro (12).

El resultado de este operador muestra cómo cambia una imagen de forma abrupta o suave en el punto analizado, y en consecuencia, cuán probable es que este represente un borde en la imagen, así como la orientación a la que tiende ese borde (23) (24).

Se trata de una técnica basada en el gradiente para realizar la detección de bordes, de manera que las variaciones de intensidad en los píxeles de la imagen prevalezcan frente a las zonas de intensidad constante. En función de cual sea la dirección en la que se aplica el gradiente, las máscaras que realizan la detección de bordes, son las siguientes:

$$G_X = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_Y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figura 1: Máscara Binomial de 3X3 (24)

Operador Canny

El Algoritmo de Canny es considerado como uno de los métodos más eficientes para la detección de bordes mediante el empleo de máscaras de convolución, ya que es menos sensible al ruido y es capaz de detectar bordes débiles (25).

En este método se utiliza la primera derivada para la detección de contornos, debido a que toma el valor cero en las regiones donde la intensidad se mantiene invariante y tiene un valor constante en toda la transición de intensidad (26) (24).

El Operador Canny utiliza el cálculo de variaciones, el cual, es una técnica que se encarga de encontrar la función que optimiza a su vez a una función dada. La función óptima definida por Canny está descrita por la suma de cuatro funciones exponenciales.

El algoritmo de Canny se divide en diferentes etapas las cuales son (27):

1. Suavizado de la Imagen
2. Encontrar gradientes
3. Suprimir los puntos que no sean máximos
4. Umbral doble
5. Seguimiento de bordes por histéresis

1-Suavizado de la imagen: El suavizado de la imagen en el algoritmo de Canny forma parte del preprocesamiento de la imagen, este operador utiliza un filtro basado en el concepto de la primera derivada gaussiana convolucionando la función con filtros gaussianos unidimensionales suavizando de esta forma el ruido presente en la imagen.

2-Encontrar el gradiente: Determinando el gradiente de la imagen se puede localizar dentro de la imagen los puntos de mayor intensidad. El borde de una imagen apunta a varias direcciones por lo que este algoritmo trabaja con cuatro filtros para detectar bordes horizontales, verticales y diagonales. Los operadores de detección de bordes como *Roberts*, *Prewitt* y *Sobel* devuelven un valor para la primera derivada en la dirección horizontal y vertical (28) (21) (24).

Las magnitudes del gradiente, las cuales representan los puntos fuertes de los bordes, se pueden determinar mediante la distancia Euclidiana y aplicando el teorema de Pitágoras

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2)$$

donde G_x y G_y son los gradientes en las direcciones x e y respectivamente.

Para la localización de los bordes ya que en ocasiones estos son muy gruesos se utiliza la ecuación siguiente:

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \quad (3)$$

3-Suprimir los puntos que no sean máximos: Para suprimir los puntos no máximos se realiza, un “thresholding”, eliminando aquellos píxeles que no tienen una magnitud (módulo del gradiente) alta, marcando los píxeles que resultan ser máximos con un recuadro de borde blanco obviando el resto.

4-Umbral doble: Al terminar la etapa de supresión los bordes que superen el umbral alto, son clasificados como fuertes, los que estén por debajo del umbral bajo son eliminados, mientras que los bordes intermedios se clasifican como débiles. La umbralización con histéresis requiere de dos umbrales, alto y bajo. La suposición de que los bordes deben ser curvas continuas en la imagen, nos permite poder descartar algunos píxeles que no forman parte de una línea. Comenzar la umbralización mediante la aplicación de un umbral alto nos permitirá marcar los bordes reales en la imagen.

5-Seguimiento de bordes por histéresis: Los bordes fuertes son incluidos también en la imagen final, ya que son interpretados como bordes reales y los bordes débiles solo se adicionan en la imagen final si son conexos con un borde fuerte. Los bordes débiles debido a los bordes verdaderos son mucho más propensos a estar directamente conectados a los bordes fuertes. Sin embargo, los bordes débiles que se deben a variaciones de color estarán distribuidos de forma independiente con respecto a los bordes fuertes (29) (27).

Basada en el uso de un umbral

La segmentación basada en umbral, permite separar un objeto de la imagen del fondo que lo circunda, esta técnica se basa en la comparación de alguna propiedad de una imagen con un umbral fijo o variable, realizando dicha comparación para cada uno de los píxeles que conforman la imagen, si el valor de la propiedad (intensidad, color, etc.) a medir de un pixel supera el valor de un umbral determinado, entonces el pixel pertenece al objeto, en caso contrario, el pixel pertenece al fondo. Cuando la segmentación se

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 el pixel evaluado pertenece al objeto o al fondo. La imagen resultante, es una imagen binaria donde los pixeles de valor uno, pertenecen al objeto y los pixeles con valor cero, pertenecen al fondo de la imagen (15).

Generalmente la selección del valor del umbral, se realiza a partir del histograma de la imagen. Ya que si una imagen está compuesta de un objeto que aparece sobre un fondo, entonces el histograma será bimodal, es decir, si el objeto es mucho más claro que el fondo, en el histograma aparecerán dos picos, el pico correspondiente al objeto ubicado en los valores de gris más elevados y otro para niveles de gris más bajos, correspondientes al fondo de la imagen (15).

Contornos Activos

El modelo Snake o modelo contorno activo deformable 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. Esta curva está representada de forma paramétrica, teniendo como único parámetro a s . Este parámetro, que está relacionado con ambas variables en el espacio (x, y) , representa el espacio que ocupa la curva ubicada en el espacio que en este caso es la imagen de operación. Existe otro parámetro que indica la cantidad de iteraciones representada como la secuencia de tiempo, el parámetro t . Se tiene entonces un modelo definido como una curva $v[s, t] = [x(s, t), y(s, t)]$

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 buscando minimizar el funcional de energía siguiente (30).

$$E_{snake}^* = \int_0^1 E_{snake}(v(s)) ds \quad (4)$$

El modelo Contornos Activos permite ajustar curvas en una imagen a los bordes de la misma. El funcionamiento de este modelo imita a una serpiente que pasa por una superficie no plana, ajustando su cuerpo a las irregularidades del camino mientras que los Contornos Activos ajustan las curvas a las irregularidades presentes en la imagen. Aunque los algoritmos que ponen en práctica esta técnica tienen un gran costo

computacional, existen variantes que eliminan esta desventaja. Este algoritmo es también muy usado en la segmentación de imágenes médicas con el objetivo de identificar regiones con gran variedad de estructuras (31).

Sistemas basados en casos

Un sistema con razonamiento basado en casos es un modelo de razonamiento que permite resolver problemas, entender situaciones y aprender utilizando mecanismos de memorización, problemas superpuestos y criterios de optimalidad (32). Dentro de las técnicas de inteligencia artificial, el Razonamiento Basado en Casos (RBC) es un paradigma de solución de problemas que difiere de otros enfoques y técnicas por su capacidad de utilizar el conocimiento específico adquirido en situaciones previas y utilizarlo en la situación presente. El nuevo problema se resuelve buscando en su memoria, un caso similar a este que haya sido resuelto en el pasado. Además incrementa su conocimiento, almacenando el nuevo caso para ser usado en situaciones futuras (33).

El enfoque que utilizan los Sistemas Basados en Casos (SBC) para la adquisición de conocimiento es una de las ventajas que se le acreditan a este tipo de sistemas; pues razonan desde episodios específicos, lo cual evita el problema de descomponer el conocimiento del dominio y generalizarlo en reglas o patrones (34). Las ventajas de los SBC están fundamentadas en la flexibilidad para representar el conocimiento a través de los casos, la organización de la Base de Conocimiento (BC) y de las estrategias de recuperación y adaptación de los casos, además de que el usuario puede ser capaz de agregar nuevos casos a la BC sin la intervención de un experto. Otra importante ventaja que ofrecen estos sistemas es la reutilización de soluciones previas a la hora de resolver un problema, además los SBC pueden almacenar casos que en situaciones pasadas resultaron ser un fracaso, lo que permite advertir sobre problemas potenciales a evitar en situaciones futuras. Otra de las ventajas que ofrecen los SBC consiste en que dichos sistemas modelan mejor el razonamiento propio del ser humano, ya que se basa en experiencias previas para resolver nuevos problemas (35) (36). Las principales limitaciones de un SBC radican en la definición de la función de semejanza y en lo difícil que resulta encontrar una estructura apropiada para describir el contenido de un caso y decidir cómo la memoria de casos debe ser organizada e indexada para un almacenamiento, recuperación y reutilización efectivo. Otra de las limitaciones de este

tipo de sistemas es que la veracidad de su resultado depende de la aplicación de una buena ingeniería del conocimiento (36).

1.4 Validación de la segmentación

En el mundo del procesamiento digital de imágenes médicas el desarrollo de nuevas técnicas de segmentación es un campo de gran interés, debido a esto, se requieren junto con los algoritmos de segmentación, métodos consistentes con los cuales se pueda evaluar y validar el desempeño de los mismos.

La validación es la medición (evaluación cuantitativa) del desempeño de un algoritmo de procesamiento de imágenes. La evaluación del desempeño tiene dos aspectos principales (37):

- eficiencia computacional
- exactitud del algoritmo.

En esta sección se expondrán algunos de los métodos más usados para la evolución de la exactitud de un algoritmo de segmentación de imágenes médicas, teniendo en cuenta que sólo tiene sentido hablar de la eficiencia computacional cuando la exactitud de dos algoritmos que realizan la misma tarea es comparable. Este tipo de evaluación nos permitirá hacer afirmaciones del tipo: “El algoritmo A es mejor que el algoritmo B para la detección de la opacidad de la capsula posterior en pacientes operados de catarata” (37).

1.4.1 Métricas para segmentación de imágenes

Los problemas de segmentación de imágenes involucran la división de una imagen en un conjunto de regiones diferentes, de acuerdo a una medida de similitud determinada. Las métricas de desempeño para este tipo de problemas son generalmente más complejas. Existen dos versiones del problema de segmentación: segmentación genérica de bajo nivel y detección y localización de anomalías. En la segmentación genérica de bajo nivel el objetivo es producir descripciones del contenido de una imagen, mientras que en el problema de detección y localización de anomalías el objetivo es encontrar todas las regiones de una imagen que contengan formas anómalas (37).

Al comparar el resultado de un algoritmo de segmentación con una imagen segmentada de referencia (p.e. Segmentación manual de un experto) se tienen en cuenta las siguientes condiciones de comparación (37):

Segmentación correcta de una región: Esto sucede cuando el solapamiento entre la región segmentada por el algoritmo propuesto y la región correspondiente en la imagen de referencia cumplen con cierto umbral mínimo de similitud.

Sub-segmentación de una región: La sub-segmentación sucede cuando varias regiones de la imagen de referencia corresponden sólo a una región de la imagen segmentada por el algoritmo propuesto.

Sobre-segmentación de una región: Ocurre cuando varias regiones de la imagen segmentada por el algoritmo propuesto, corresponden a una sola región de la imagen de referencia.

Región faltante: La región faltante, existe cuando una región en la imagen de referencia no corresponde con ninguna región existente en la imagen segmentada por el algoritmo propuesto.

Región producida por ruido: Este tipo de regiones aparecen en la imagen segmentada y no corresponden a ninguna región de la imagen de referencia.

Un ejemplo de las condiciones de comparación anteriores se puede ver en la siguiente imagen.

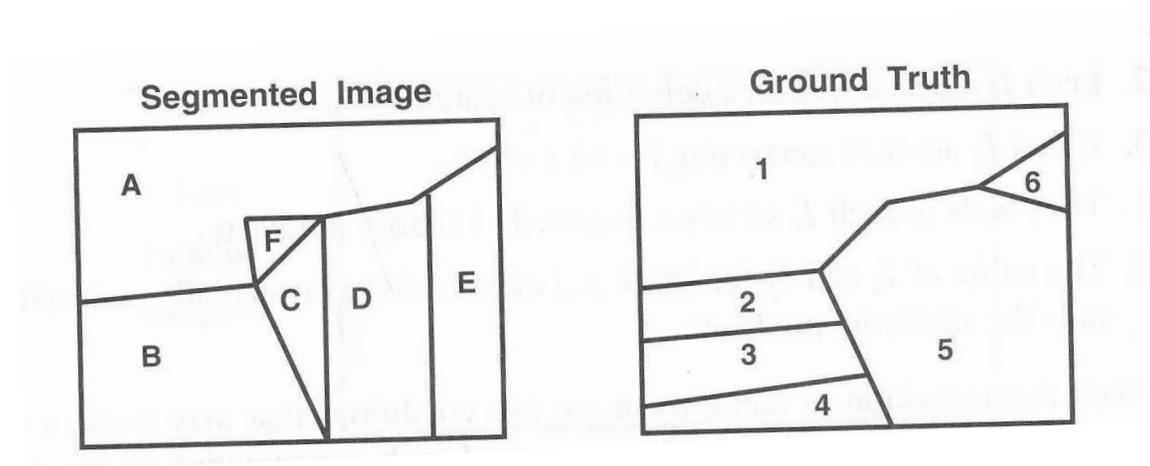


Figura 2: Ejemplo de condiciones de comparación de segmentación (38)

Donde:

A corresponde a 1 como una instancia de una segmentación correcta.

5 corresponde a C,D y E como una sub_ segmentación.

B corresponde a 2,3 and 4 como una sobre_ segmentación.

6 es una región perdida.

F es una región producida por ruido.

1.4.2 Medidas del error

Las medidas de error son usadas frecuentemente a la hora de diseñar algoritmos de segmentación, clasificación y en la detección de contornos en sistemas de inteligencia artificial. Estas medidas se clasifican en tres grupos (39):

- Medidas estadísticas de error.
- Medidas de localización.
- Métrica Hausdorff

1.4.2.1 Medidas estadísticas de error.

Las medidas estadísticas de error nos darán la medida de la frecuencia con la que se clasifican incorrectamente los “pixels”, esta medida tiene cuatro variables de evaluación definidas a continuación (39) (40):

Verdaderos Positivos (VP): Son los números de pixeles que pertenecen a la imagen clasificados correctamente.

Verdaderos Negativos (VN): Son los números de pixeles pertenecientes al fondo clasificados como tales

Falsos Positivos (FP): Son los números de pixeles del fondo clasificados como parte de la imagen.

Falsos Negativos (FN): Son los números de pixeles pertenecientes a la imagen clasificados como parte del fondo.

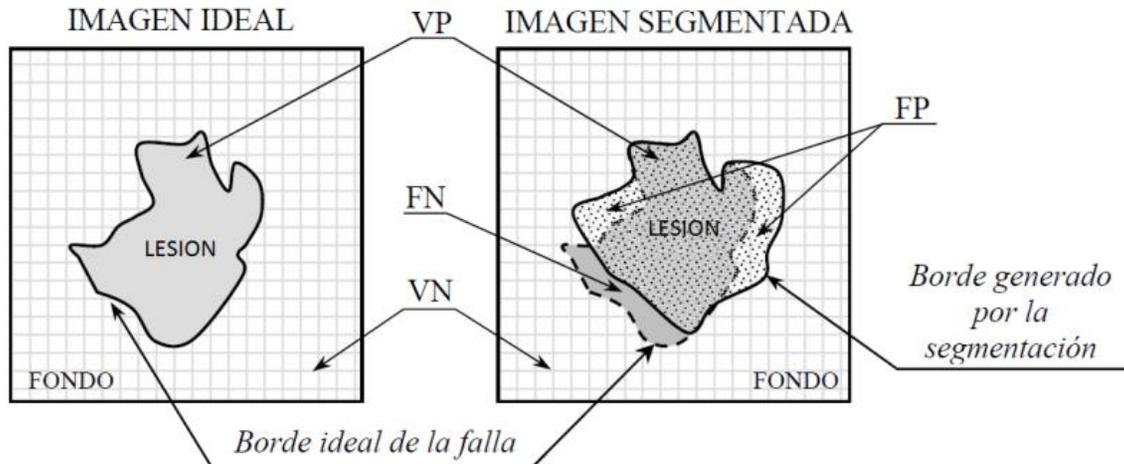


Figura 3: Ejemplo de variables de evaluación (40)

La Precisión (P), definida por la ecuación (5) nos dará una idea de la sobre-segmentación en la segmentación obtenida con el algoritmo, respecto a la segmentación verdadera, tomando valor 1, cuando no existen zonas de falsos positivos (sobre-segmentadas), y valores cercanos a 0, en caso contrario (40).

$$P = \frac{VP}{VP+FP} \quad (5)$$

La Exactitud (E), a diferencia de la precisión se diferencia en que la precisión toma su valor máximo 1 en caso de que la segmentación automática de la imagen esté completamente contenida dentro de la segmentación ideal, y la exactitud toma su valor máximo 1 en caso que la segmentación realizada por el algoritmo coincida plenamente con la realizada por el experto, la ecuación es la siguiente (40):

$$E = \frac{VP+VN}{VP+VN+FP+FN} \quad (6)$$

Un ejemplo de lo antes dicho se puede ver en esta imagen donde para el caso de la Precisión la segmentación automática queda contenida en la lesión verdadera por lo que la Precisión nos dará su máximo valor como una aproximación a la segmentación correcta.

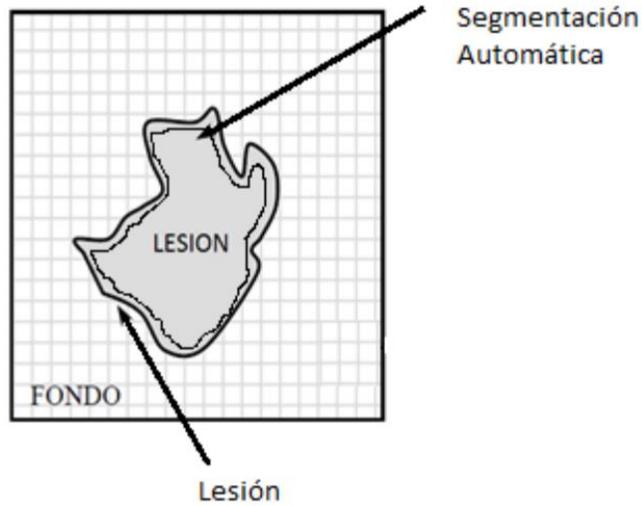


Figura 4: Ejemplo de precisión (40)

Mientras que, la Exactitud tomará su mayor valor cuando realmente coincida con el contorno de la lesión.

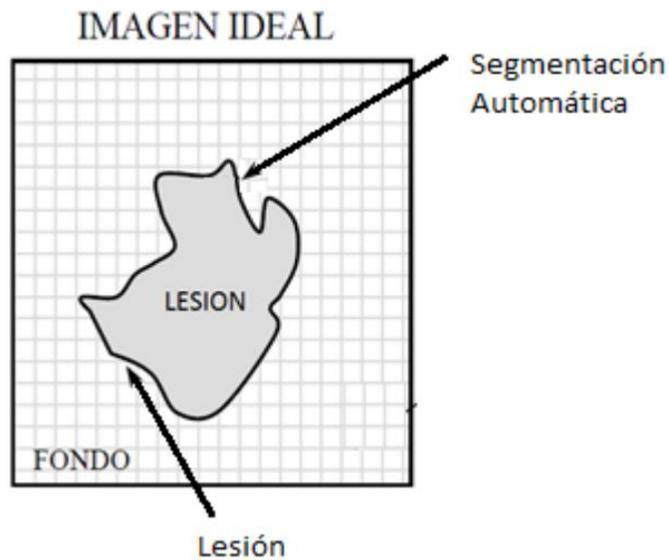


Figura 5: Ejemplo de exactitud (40)

Otra de las medidas estadísticas de error es la Revocación (R), la cual nos dará cuenta de la sub-segmentación que existe en la imagen segmentada con respecto a la segmentación verdadera, tomando valor máximo 1, cuando no existen zonas de falsos

negativos (sub-segmentadas) y valores cercanos a 0, en caso contrario. Esta medida devuelve el máximo valor en caso de que la segmentación sea perfecta, es decir, que coincida plenamente con la realizada por el experto. Puede ser descrita por la ecuación siguiente (40).

$$R = \frac{VP}{VP+FN} \quad (7)$$

El Promedio Señal a Ruido (NSR) se expresa mediante la ecuación (8). Esta expresión es matemáticamente igual a la división de los Falsos Positivos con los Verdaderos Positivos (FP/VP) (39).

$$NSR = \frac{n(B \setminus A)}{n(B \cap A)} \quad (8)$$

Donde, A representa a la segmentación ideal realizada por el especialista, B la segmentación automática, y $n(X)$ la cantidad de píxeles de X .

La Medida FM o Medida de Primer Plano es otra de las medidas estadísticas de error y depende plenamente de la Precisión y la Revocación tomando su valor máximo 1 en caso de que la segmentación sea perfecta. Puede ser descrita por la ecuación siguiente (40).

$$FM = \frac{2 * P * R}{P + R} \quad (9)$$

A diferencia de las medidas anteriores la Tasa de Error (ER) devuelve 0 en caso de que la segmentación sea perfecta, lo que indica que no existe error en la segmentación, ya que la segmentación manual y la automática coinciden plenamente. La ecuación que la describe se muestra a continuación (40).

$$ER = \frac{FP}{VP+FN} \quad (10)$$

La Tasa negativa (NRM), la cual esta expresada por la ecuación (11), al igual que la Tasa de Error devolverá 0 en caso de que la segmentación sea perfecta, esta expresión se describe por la ecuación siguiente (40),

$$NRM = \frac{NR_{FN} + NR_{FP}}{2} \quad (11)$$

donde

$$NR_{FN} = \frac{FN}{VP + FN} \quad (12)$$

y

$$NR_{FP} = \frac{FP}{VP + FN} \quad (13)$$

1.4.2.2 Medidas de localización

Las medidas de localización son aquellas que se basan en el cálculo de la distancia mínima entre un punto y un conjunto de puntos, es decir, cumplen con la condición siguiente:

Si $\rho(x,y)$ es la distancia entre dos píxeles x e y , entonces la distancia más corta entre un píxel $x \in X$ y $A \subseteq X$ se define como (39):

$$d(x, A) = \inf\{p(x, a) : a \in A\} \quad (14)$$

donde X es el conjunto de píxeles de la imagen y A es el conjunto de píxeles de valor 1 (binario) en la imagen.

Entre las medidas de localización se encuentran las medidas de Peli y Malah, estas utilizan dos criterios estadísticos característicos de la detección de errores, los cuales son el error medio (Media) y el error cuadrático medio (Varianza) (41) (39). Estas dos medidas están definidas por las ecuaciones siguientes:

$$Media = \frac{1}{n(B)} * \sum_{x \in B} d(x, A) \quad (15)$$

$$Varianza = \frac{1}{n(B)} * \sum_{x \in B} d(x, A)^2 \quad (16)$$

donde $d(x,A)$ es la distancia entre el i -ésimo píxel de la segmentación automática B y el píxel más cercano en el contorno de prueba A y $n(B)$ es la cantidad de píxeles de la imagen binaria B .

La Media como medida de localización nos dará una idea de la distancia media que existe entre los píxeles del contorno de prueba y los de la segmentación, aunque puede darse el caso en el que una segmentación tenga píxeles muy cercanos y otro grupo de píxeles muy lejanos, o sea, que exista dispersión en los píxeles y aun así obtener una buena medida. Sin embargo, esto no pasa con la Varianza, ya que penaliza los píxeles más alejados haciendo que la medida de similitud crezca mucho si hay píxeles muy dispersos. Como consecuencia de esto se puede afirmar que si una segmentación aporta un buen valor para la media y uno malo para la varianza entonces los píxeles de la misma están dispersos, por lo que no es una buena segmentación (41).

Otra de las medidas de localización es la Figura de Merito de Pratt (FOM), definida por la ecuación siguiente (39):

$$FOM(A, B) = \frac{1}{\max\{n(A), n(B)\}} * \sum_{x \in B} \frac{1}{1 + \alpha * d(x, A)^2} \quad (17)$$

donde alfa es una constante de escala con valor 1/9, A (segmentación ideal), B (prueba), $n(A)$ y $n(B)$ cantidad de píxeles de A y B respectivamente y $d(x, A)$ distancia de un píxel en la imagen B al píxel más cercano en A .

FOM es una medida para errores de Falsos Positivos y no es sensible a errores de Falsos Negativos. Lo mismo ocurre con el error medio y cuadrático medio, que además son muy sensibles al ruido de fondo ya que si el error de la imagen B contiene incluso un sólo píxel muy distante de A , su valor de la distancia elevará drásticamente la distancia media (39).

1.4.2.3 Métrica Hausdorff

Dentro de la métrica Hausdorff, la distancia de Hausdorff (h) de un contorno segmentado (CS) a un contorno de referencia (CR) se define por la ecuación siguiente (42) (39) (37):

$$h(CS, CR) = \max[\sup d(cs, CR), \sup d(cr, CS)] \quad (18)$$

donde cs y cr son puntos de los contornos CS y CR respectivamente, y $d(cs, cr)$ es cualquier métrica entre éstos puntos, por su simplicidad se utiliza frecuentemente la distancia euclidiana.

La distancia de Hausdorff nos da una buena medida de similitud entre dos imágenes, pero en la práctica no suele usarse como medida de error, pues es muy sensible al ruido, ya

que un sólo pixel erróneo puede causar que el valor de $h(CS, CR)$ se eleve hasta el máximo posible.

La distancia de Baddeley es una medida alternativa a la distancia de Hausdorff para imágenes binarias, donde, Baddeley propone reemplazar el operador Max(máximo) por un valor medio para reducir la sensibilidad al ruido. Esta distancia se describe por la ecuación siguiente (43) (39):

$$\Delta^p(A, B) = \left[\frac{1}{N} \sum_{x \in X} |d(x, A) - d(x, B)|^p \right]^{1/p} \quad (19)$$

donde X es un dominio común de dos resultados de segmentación A y B , N = número total de píxeles en el conjunto X y $p > 1$, teniendo en cuenta que cuando p se acerca al infinito la distancia de Baddeley se acerca a la distancia de Hausdorff.

1.5 Programa analizador de Opacidad Capsular

El Programa analizador de Opacidad Capsular (PANDOC) es una herramienta para la cuantificación y análisis de la OCP. Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía postoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto toma las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas orientadas sólo al uso de imágenes de tomogramas del Pentacam, lo cual hace más intuitivo la interfaz actual y la focaliza en su propósito principal (44).

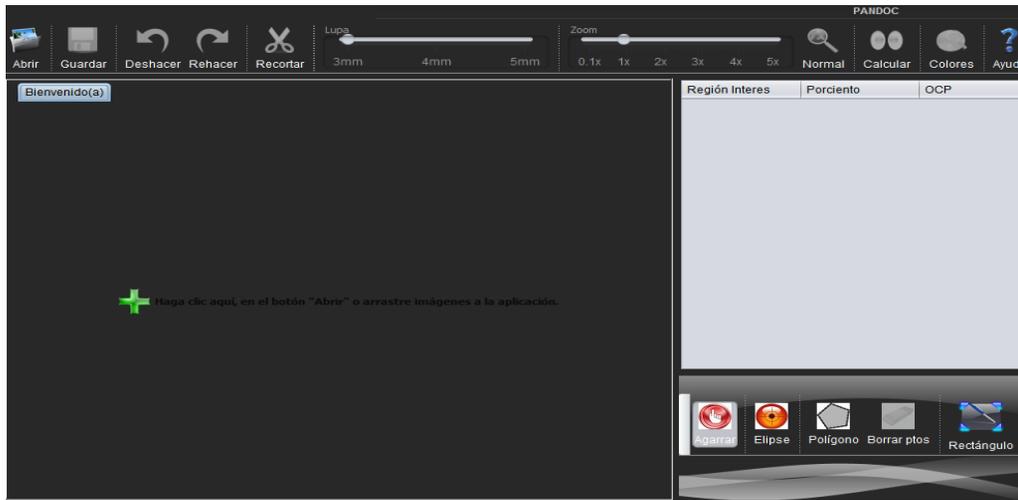


Figura 6: Software PANDOC para la cuantificación de opacidad capsular

Para realizar la segmentación de la OCP en imágenes provenientes del Pentacam dentro del grupo de investigación AIRI se han propuesto 3 algoritmos, en función de obtener mejores resultados en tiempo de ejecución y en calidad de la segmentación:

- 1 **Algoritmo para identificar OCP (SBC_PP):** Segmentación de la OCP mediante Sistema Basado en Casos y algoritmo de trilateración celular, donde cada pixel diferente del fondo entre los 3 y 6 mm del centro de la estructura ocular es un nuevo caso. Los rasgos están basados en el comportamiento de la intensidad del píxel, su cercanía al centro y el comportamiento de sus vecinos. Este algoritmo está formulado fundamentalmente bajo el concepto de que la Capsulorrexis sólo varía entre esas distancias.
- 2 **Algoritmo para identificar OCP (SBC_CC):** Segmentación de la OCP mediante Sistema Basado en Casos y algoritmo de componentes conexas, donde cada región extraída que tenga elementos entre los 3 y los 6 mm del centro de la estructura ocular es un nuevo caso. Los rasgos están basados en las características (como valores de intensidad y tamaño) y variaciones de las regiones (como forma), también se tuvo en cuenta la cercanía al centro de la región. Este algoritmo atiende a la formación de la opacidad en el saco posterior, en el cual se van acumulando cada una de las células (regiones opacas).
- 3 **Algoritmo para identificar OCP (TCH_CA):** Segmentación de la OCP mediante el uso de la transformada circular de Hough y algoritmo de contornos activos, donde primeramente se realiza un conjunto de técnicas de filtrado, para con la transformada circular de Hough detectar la forma circular de la

Capsulorrexis, para posteriormente usando contornos activos ajustar este círculo a los bordes (contornos) reales de la misma. Este algoritmo está formulado bajo la premisa de que todo píxel diferente del fondo dentro de la región comprendida por la Capsulorrexis es OCP.

1.6 Herramientas usadas durante el desarrollo

En este epígrafe se describen las tecnologías y herramientas a utilizar durante el desarrollo del software, se exponen las principales características de la metodología de software escogida, así como las ventajas que esta ofrece para este proyecto. Se describe la herramienta para el modelado, el entorno integrado de desarrollo y se argumenta acerca de las características y beneficios que ofrece el lenguaje de programación seleccionado, así como las ventajas que trae consigo el sistema gestor de base de datos propuesto.

1.6.1 Metodología de desarrollo de Software

El proceso de desarrollo de software, se define como el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, teniendo como objetivo la obtención de un producto que cumpla con las expectativas del cliente (45). Este proceso es una de las tareas más difíciles de la ingeniería; prueba de ello es la existencia de numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, aquellas propuestas más tradicionales o robustas que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán, o sea se basa en el uso exhaustivo de documentación durante todo el ciclo del proyecto. Por otra parte se encuentran las metodologías ágiles o ligeras las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas (46). La selección de una metodología apropiada, resulta en ocasiones un factor esencial para obtener un software con la calidad esperada.

Luego de la evaluación de las metodologías de desarrollo y el estudio de sus características, etapas de desarrollo y ventajas que posibilitan, se determina utilizar una metodología ágil, dado que la prioridad es satisfacer al cliente mediante tempranas y

continuas entregas de software; el cliente es parte del equipo de desarrollo, el equipo de desarrollo es de 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, lo cual permitió identificar la Programación Extrema (XP) como una alternativa acertada, además de ser la metodología de desarrollo del grupo de investigación (47) (45).

Metodología de software XP

La metodología XP además de ser una de las más conocidas y más usadas de las metodologías ágiles fue seleccionada 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 activa del mismo, permitiendo la comunicación y retroalimentación frecuente entre ambas partes. Una de las prácticas más significativas que posee esta metodología, es que con ella es posible simplificar el diseño para agilizar el desarrollo, así como facilitar el mantenimiento del sistema. La ejecución de las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código. Es también recomendable utilizar esta metodología debido a que los requisitos tienen altas probabilidades de cambiar con el tiempo (45). 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* por sus siglas en inglés (AIRI) creador del mismo, lo cual constituye un buen criterio a tener en cuenta siendo el resultado de este proyecto un módulo para el software antes mencionado.

Algunas de las características de XP son (45):

- Pruebas unitarias: se basan en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexibles al cambio.
- 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.

- Entrega de software en espacio de tiempo pequeño: La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.

Ventajas que brinda XP (45):

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

XP consta de cuatro fases para el desarrollo (45) (48):

1. **Planificación:** Es la fase donde se realiza una recopilación de todos los requerimientos del proyecto, se crean las HU, el plan de iteraciones y el plan de entregas.
2. **Diseño:** La metodología XP hace especial énfasis en los 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:** Esta fase 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:** Se basan en la comprobación del funcionamiento de la codificación que se haya implementado, garantizando la evaluación de las distintas tareas en las que han sido divididas las HU de manera que si se encuentran errores darles solución para prever que no vuelvan a ocurrir.

1.6.2 Lenguaje de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar instrucciones que pueden ser llevadas a cabo por un ordenador. Puede usarse para crear programas

que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana. Permiten especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones deben tomar bajo una gran cantidad de opciones posibles. Todo esto, a través de un lenguaje que intenta ser relativamente próximo al lenguaje humano o natural (49).

Debido a que el software PANDOC está desarrollado en el lenguaje de programación Java y el resultado de este proyecto constituye un módulo de esa aplicación informática se escogió el mismo para desarrollar la solución propuesta. Además, se cuenta con abundante documentación sobre el lenguaje ya que es uno de los más utilizados. A continuación, se mencionan algunas de sus características.

Características fundamentales de Java (50):

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

1.6.3 Entorno integrado de desarrollo

Para el desarrollo de la solución, como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) se seleccionó el NetBeans 8.2. Es libre, de código abierto y 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, permite la utilización y edición de los componentes visuales de forma sencilla. Incluye también el control de versiones, lo cual representa una ventaja debido a que permite administrar las diferentes versiones del código fuente. Además es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris (50):

1.6.4 Herramienta para el modelado

Como herramienta para el modelado de la solución se seleccionó Visual Paradigm for UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés) esta herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. El VP-UML en su versión v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, que brinda a los equipos de desarrollo de software todo lo necesario para la planificación de software, captura de requisitos, planificación de controles y modelado de clases y de datos. Esta herramienta ofrece gran interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo, además de hacer posible la generación de código Java desde los diagramas (51).

1.6.5 Sistema Gestor de Base de Datos.

Como sistema gestor de base de datos (SGBD) se utilizó SQLite, el cual es un SGBD relacional que está contenido en una pequeña (~500kb) biblioteca construida en el lenguaje C. Es un proyecto de dominio público creado por Dr. Richard Hipp. A diferencia de los SGBD cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones (52).

Esto reduce el tiempo de acceso a la base de datos (BD), debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la BD (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero

estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de BD al inicio de cada transacción. Este SGBD dispone de una completa interfaz orientada a objetos, con distintas funciones que facilitan la manipulación de datos y posee funciones muy similares a las que se pueden manejar con MySQL.

Combina el motor y la interfaz de la BD en una única biblioteca, y almacena los datos en un único archivo de texto plano. Esto hace que cada usuario pueda crear tantas BD como desee sin la necesidad de la intervención de un administrador de BD que gestione los espacios de trabajo, usuarios y permisos de acceso.

La potencia de SQLite se basa fundamentalmente en la simplicidad, lo que hace que no sea una buena solución cuando se trata de entornos de tráfico muy elevado y/o alto acceso concurrente a datos, además encapsula toda la BD en un único fichero. El hecho de almacenar toda la BD en un único archivo, facilita la portabilidad de los datos, y solamente tiene la restricción del espacio de disco asignado.

Se puede utilizar SQLite de dos formas:

- Como gestor de base de datos local en un computador. De esta forma, se pueden gestionar BD con SQLite igual que si se estuviese trabajando con un sistema gestor de base de datos como MySQL, sin necesidad de instalar nada, ya que SQLite se compone de un único archivo ejecutable.
- Como una extensión más de PHP, utilizando las funcionalidades de SQLite configuradas, o bien como módulo de PHP, o como biblioteca; sin necesidad de tener instalado o estar conectado con un servidor de base de datos. Ofrece una rápida interfaz de la BD almacenada en el archivo de texto plano.

Varios procesos o hilos pueden acceder a la misma BD sin problemas. Varios accesos de lectura pueden ser servidos en paralelo. Un acceso de escritura sólo puede ser servido si no se está utilizando otro de manera concurrente. En caso contrario, el acceso de escritura falla devolviendo un código de error (o puede automáticamente reintentarse hasta que expira un “timeout” configurable). Esta situación de acceso concurrente podría cambiar cuando se está trabajando con tablas temporales. Sin embargo, podría producirse un “deadlock” debido al proceso multihilo.

SQLite Admin es un administrador de bases de datos SQLite que básicamente permite (53):

- Abrir, crear, compactar bases de datos.
- Crear, renombrar y eliminar tablas vacías.
- Crear, renombrar y eliminar columnas.
- Crear, modificar y eliminar registros.
- Encriptar o reenciptar bases de datos
- Correr sentencias SQL personalizadas.
- Crear y eliminar índices, Triggers y Cursores.

1.6.6 Patrones de diseño

Un patrón es una agrupación de información que proporciona respuesta a un conjunto de problemas similares. Estos ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño “orientado a objetos” más flexible, elegante y reusable en algunos casos (54).

Entre los patrones de diseño de manera general se pueden mencionar: los GRASP (General Responsibility Assignment Software Patterns), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos; dentro de este grupo de patrones se encuentran: experto, creador, bajo acoplamiento, alta cohesión, controlador, fabricación pura, indirección, variaciones protegidas, no hables con extraños y polimorfismo, y los patrones GOF (Gang of Four), que hacen referencia a los cuatro autores del libro: Design Patterns, encargados de la inicialización, agrupación y comunicación de los objetos. (54).

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, ya que sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares (55).

1.7 Conclusiones del capítulo

En este capítulo se realizó un análisis de las características que presentan las imágenes médicas en el campo del procesamiento digital de imágenes. Se expusieron las técnicas de segmentación usadas en el desarrollo del software PANDOC. Posteriormente, partiendo del presupuesto de que no existe un método de segmentación que alcance

resultados aceptables para todo tipo de imágenes, se realizó un estudio detallado de las técnicas y métodos de validación de la segmentación de las mismas, identificando tres tipos de medidas de error, medidas estadísticas de error, medidas de localización y métrica Hausdorff. Además, se seleccionaron y describieron las herramientas y tecnologías a ser usadas en el desarrollo de la propuesta de solución, permitiendo seleccionar a Java como lenguaje de programación, NetBeans como IDE, SQLite como SGBD, Visual Paradigm como herramienta para el modelado y a XP como la metodología más apropiada para guiar el desarrollo del sistema.

Capítulo 2. Validación de la segmentación en imágenes del Pentacam: Propuesta de solución.

En este capítulo se presenta la propuesta de solución para validar la segmentación en imágenes médicas procedentes del equipo oftalmológico Pentacam. Se describe la fase inicial de la metodología de desarrollo utilizada para implementar la solución propuesta, y se obtienen los artefactos de esta fase, tales como, las Historias de Usuarios, Plan de Iteraciones, Plan de Duración de Iteraciones, Plan de Entregas y tarjetas CRC.

2.1 Descripción de la solución

Para elaborar la propuesta de solución se tuvieron en cuenta todos los métodos de validación de la segmentación expuestos en este trabajo, con el objetivo de brindarle al usuario las métricas necesarias para validar la calidad de la segmentación de la OCP en imágenes provenientes del Pentacam. A continuación, se muestra el diagrama de actividades que modela la secuencia de pasos del sistema.

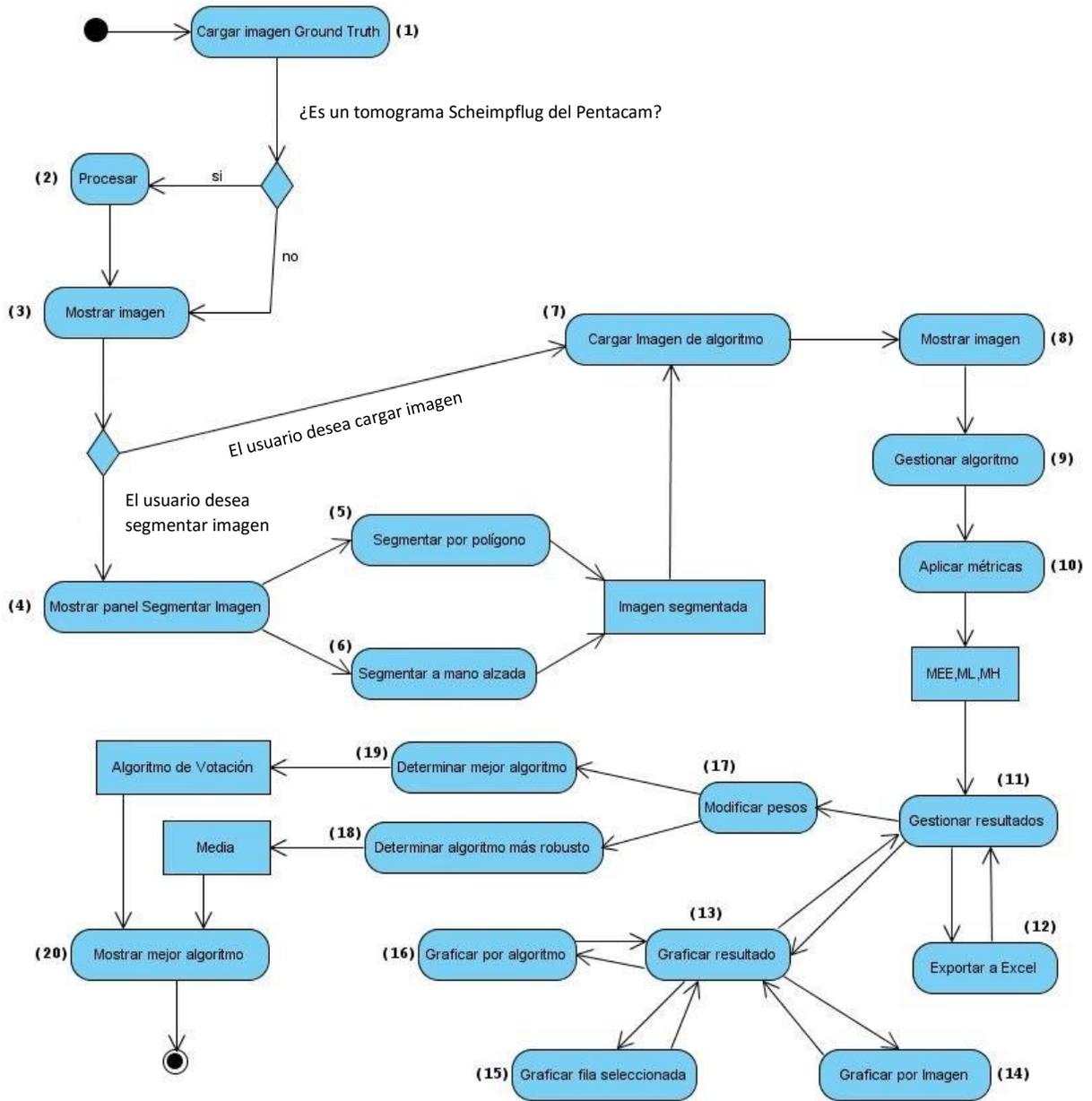


Figura 7: Diagrama de flujo del sistema

1. En esta etapa se carga la imagen "Ground Truth" en el formato jpg y se verifica si es un tomograma Scheimpflug proveniente del Pentacam.
2. En esta etapa se recorta la imagen proveniente del Pentacam de forma automática.
3. En esta etapa se muestra la imagen cargada y el usuario verifica visualmente si esta segmentada o no.

4. En esta etapa se muestra el panel de herramientas para realizar la segmentación manual de la imagen.
5. En esta etapa se le brinda la posibilidad al usuario de seleccionar la región a segmentar de forma poligonal (punto a punto).
6. En esta etapa se le brinda la posibilidad al usuario de seleccionar la región a segmentar de forma libre, a mano alzada.
7. En esta etapa se carga la imagen segmentada automáticamente para su posterior análisis comparada.
8. En esta etapa se muestra la imagen cargada segmentada automáticamente.
9. En esta etapa se introducen los datos del algoritmo a ser analizado (datos como nombre del algoritmo, nombre del creador, descripción etc.).
10. En esta etapa se aplican sobre la imagen segmentada automáticamente, las medidas de error propuestas las cuales se clasifican en, medidas estadísticas de error, medidas de localización y métrica Hausdorff.
11. En esta etapa se muestran de forma tabular los resultados de la aplicación de las métricas.
12. Tras mostrar los resultados de las métricas se brinda la posibilidad al cliente de exportarlos como un documento Excel.
13. Tras mostrar los resultados de las métricas se brinda la posibilidad al cliente de mostrarlos de forma gráfica.
14. En esta etapa al seleccionar el usuario el modo de graficación por imagen el sistema recorrerá todas las imágenes del mismo nombre calculará el promedio de las métricas entre ellas y mostrará el resultado de forma gráfica.
15. En esta etapa al seleccionar el usuario el modo de graficación por fila, el sistema graficará los valores de cada métrica en la fila seleccionada.
16. En esta etapa al seleccionar el usuario el modo de graficación por algoritmo el sistema recorrerá todas las imágenes que utilizan el mismo algoritmo, calculará el promedio de las métricas entre ellas y mostrará el resultado de forma gráfica.
17. En esta opción el usuario puede ponderar cada métrica según el nivel de importancia que el estime para cada métrica con el objetivo de obtener el mejor algoritmo de segmentación.
18. En esta etapa el sistema determinará el algoritmo más robusto hallando la media de todas las métricas entre todas las imágenes y mostrando una tabla con los valores medios de todas las métricas por algoritmo.

19. En esta etapa el sistema determinará el mejor algoritmo hallando la media entre los valores medios de cada algoritmo de segmentación probado apoyándose en un algoritmo de Votación para su selección.
20. En esta etapa el sistema mostrará el nombre del mejor algoritmo al igual que el nombre del algoritmo más robusto (el que mostró mejores resultados en las métricas).

2.2 Fase de Planificación

La metodología XP define como fase inicial del desarrollo de software la planificación. Durante el transcurso de esta etapa se realiza el proceso de identificación y elaboración de las historias de usuario (HU), además el equipo de trabajo se familiariza con las tecnologías y herramientas seleccionadas para el desarrollo. El cliente define el nivel de prioridad con que se deben implementar las HU, así como la estimación del esfuerzo que costará implementar las mismas. El resultado de la presente fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración, de forma tal que sea una guía durante el desarrollo (56).

2.2.1 Historias de Usuario

Uno de los artefactos definidos por la metodología XP son las HU las cuales son utilizadas para especificar las funcionalidades que brindará el sistema. Cada HU es una representación de un requerimiento de software escrito en una o dos frases, utilizando el lenguaje común del usuario. Estas representan una forma rápida de administrar los requerimientos de los usuarios, sin necesidad de elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos (56). En correspondencia con la prioridad asignada por el cliente a cada HU y teniendo en cuenta la complejidad y riesgo determinado por el programador, se realiza la estimación de cada una de las identificadas.

A continuación, se muestran algunas de las Historias de Usuario identificadas en esta fase.

Tabla 1: Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Determinar algoritmo más robusto
Iteración asignada: 3	
Prioridad en negocio: Alta (Alta / Media / Baja)	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto (Alto / Medio / Bajo)	Puntos reales: 2 semanas
Descripción: Dado los resultados obtenidos en las comparaciones entre el valor medio de las métricas por cada algoritmo el sistema devuelve el nombre del algoritmo más robusto.	
Observaciones:	

2.2.2 Estimación de esfuerzos por Historias de Usuario

Tras la definición de prioridades de cada HU se realiza una estimación del esfuerzo necesario de cada una de ellas. Donde la unidad de estimación es el punto, un punto equivale a una semana ideal de programación. Esta estimación arrojó los resultados siguientes:

Tabla 2: Puntos de estimación por historias de usuario

No	Historias de usuario	Puntos de estimación
1	Cargar imagen de tomograma	2
2	Reconocer imagen de tomograma	2
3	Segmentar imagen "ground truth"	2
4	Calcular medidas estadísticas de error	2
5	Calcular medidas de localización	2

6	Calcular métrica Hausdorff	2
7	Determinar mejor segmentación	2
8	Determinar algoritmo más robusto	2
9	Gestionar resultados	2
10	Graficar resultados	2

2.2.3 Requisitos No Funcionales

Los requisitos no funcionales son aquellos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento que debe cumplir o cumple el producto de software (57). Estos requisitos tienen lugar por la demanda del sistema de ciertas funciones o cualidades, o porque el cliente desea que ese requerimiento forme parte del producto de software final.

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

Requisitos de hardware: La computadora deben tener como mínimo las siguientes características:

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

2.2.4 Plan de iteraciones

Después de definir las HU y estimar el esfuerzo propuesto para la realización de cada una de ellas, se requiere establecer el contenido de trabajo de estas, y establecer el plan de iteraciones, regulando la cantidad de HU a implementar dentro del rango establecido por la estimación efectuada. Es decir, una iteración tiene como finalidad implementar las HU que se consideraron más necesarias atendiendo a su relevancia e impacto. Partiendo de los aspectos tratados anteriormente, la aplicación que se pretende construir se desarrollará en 3 iteraciones, expuestas de forma detallada a continuación:

Iteración 1

En esta iteración se implementan las funcionalidades que permiten reconocer las imágenes provenientes del Pentacam y realizar el proceso de segmentación manual por un especialista.

Iteración 2

En esta iteración se realizan todas las funcionalidades relacionadas con el cálculo y estimación de las métricas de error entre una imagen segmentada de manera automática y el “*ground truth*”.

Iteración 3

En esta iteración se realizan las HU que se encargan de mostrar un resultado estadístico de forma gráfica, ya que en ellas se implementan las funcionalidades, de gestión de resultados, calcular mejor segmentación y calcular algoritmo más robusto los cuales se generan de forma estadística.

Tabla 3: Plan de duración de las iteraciones

Iteraciones	Historias de usuario	Duración
Iteración 1	Cargar imagen de tomograma	6
	Reconocer imagen de tomograma	
	Segmentar imagen “ <i>ground truth</i> ”	
Iteración 2	Calcular medidas estadísticas de error	6
	Calcular medidas de localización	
	Calcular métrica Hausdorff	
Iteración 3	Determinar mejor segmentación	8
	Determinar algoritmo más robusto	
	Gestionar resultados	
	Graficar resultados	

2.2.5 Plan de entrega

El plan de entregas es el cronograma que define cuáles HU serán agrupadas para conformar una entrega, y el orden en que se realizarán estas (47). Este plan se encarga de definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una de ellas.

Tabla 4: Plan de entrega

	Iteración 1	Iteración 2	Iteración 3
Cantidad de HU	3	3	4
Fecha de entrega	13/02/2017	27/03/2017	22/05/2017

2.3 Diseño de la propuesta de solución

En este epígrafe se hace alusión a los principales elementos que componen el diseño del sistema propuesto, donde siguiendo los planteamientos de la metodología XP este debe ser un diseño simple y de fácil interacción con el usuario, basándose fundamentalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC). También en este epígrafe se detallan las tareas generadas por cada HU, así como las tres iteraciones llevadas a cabo en la fase anterior.

2.3.1 Tarjetas CRC

La metodología XP a diferencia de otras metodologías en lugar de utilizar diagramas para desarrollar modelos, representa las clases mediante tarjetas *Clases, Responsabilidad y Colaboración*. Las tarjetas CRC ayudan a definir actividades durante el diseño del sistema y evitan el enfoque procedimental destacando la orientación a objetos (58). Al poder trabajar con una metodología basada en objetos estas tarjetas permiten que el equipo de desarrollo completo contribuya también en la tarea del diseño. La estructura física de las tarjetas CRC facilita la interacción entre los integrantes del proyecto, en sesiones en las que se aplican técnicas de grupos como, juego de roles o tormenta de ideas y se ejecutan escenarios a partir de la especificación de los requisitos, historias de usuarios o casos de uso. De esta forma, van surgiendo las entidades del sistema junto con sus responsabilidades y colaboraciones (59). El nombre de la clase se coloca a modo de título

en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Tabla 5: Tarjeta CRC de la clase NewJFrame

Clase: NewJFrame	
Responsabilidades	Colaboradores
Permite mostrar y editar la imagen proveniente del Pentacam.	LabelImage
Permite realizar el cálculo de las métricas de error.	Métodos
Permite cargar la imagen de un fichero externo y tomar todos sus atributos.	Imagen
Permite a la aplicación conectarse a la base de datos.	ConexionSqlite
Permite exportar los valores resultantes de cada imagen por métrica a un documento Excel	ExportarExcel
Permite importar los valores resultantes de cada imagen de un documento Excel a la aplicación, para su posterior análisis.	ImportarExcel

2.3.3 Estilos de codificación

Los estilos de codificación comprenden todos los aspectos de la generación de código, debiendo reflejar un estilo armónico en su confección (como si un único programador hubiera escrito todo el código). Estos estilos permiten entender de manera rápida y

sencilla el código empleado en el desarrollo del software, es por eso que al comenzar un proyecto, se debe establecer un estilo de codificación para asegurarse de que se trabaje de forma coordinada, permitiendo realizar el mantenimiento a un sistema de software, al establecer como operar con la base de código existente (60).

- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Los comentarios irán encima del método correspondiente.
- La primera letra de los nombres de los métodos se escribirán en minúscula y si el nombre es compuesto las siguientes palabras empezarán con mayúscula.
- Se escribirá una declaración por línea.
- Los identificadores de las variables se escribirán en minúscula y en caso de ser un nombre compuesto las siguientes palabras empezarán con mayúscula.

2.3.2 Patrones de diseño utilizados

Los patrones de diseño constituyen una descripción de la interacción entre las clases, con el objetivo de resolver un problema de diseño general en un determinado contexto. El uso de patrones permite la estandarización de la forma en que se realiza el diseño de manera que el código pueda ser reutilizable, y se le pueda dar mantenimiento cuando se requiera. En este proyecto se utilizaron los siguientes patrones de diseño:

Experto: El uso de este patrón se evidencia en la clase Imagen. Esta es la encargada de cargar la imagen desde un fichero externo y de procesarla en caso de provenir del Pentacam. Ver Figura 8.

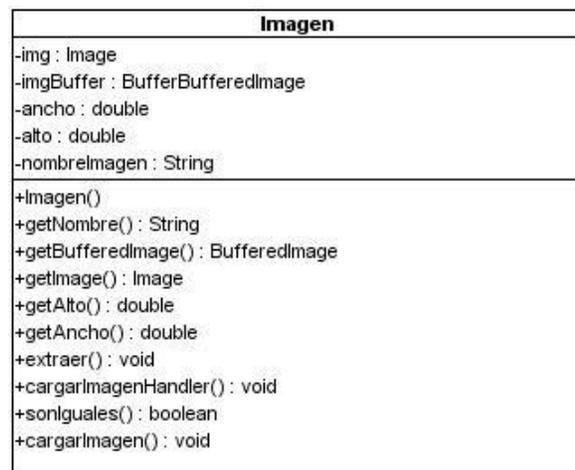


Figura 8: Clase Imagen

Creador:

Dentro de la aplicación este patrón se observa en las acciones del controlador NewJFrame, en la cual se crean objetos para el procesamiento de la imagen. Estas clases tienen la responsabilidad de instanciar objetos para cumplir sus funcionalidades. Ver figura 9

```
JFrame frame = new JFrame("Graficación");  
  
frame.setLayout(new GridLayout(2, 1));  
JPanel jp = new ChartPanel(Grafica);  
JPanel pa = new ChartPanel(Grafical);  
frame.add(jp);  
frame.add(pa);
```

Figura 9: Creación de instancias

2.4 Conclusiones del capítulo

Durante el desarrollo de este capítulo se describió la propuesta de solución, enmarcada en el procesamiento digital de imágenes médicas según la metodología XP, generando como artefactos un total de 10 HU que caracterizan al sistema y describen los aspectos a tener en cuenta para el desarrollo de la solución propuesta. Se realizó también la estimación del esfuerzo necesario para la implementación de las HU, llevándose a cabo el plan de entregas en el cual se define cuál de ellas formará parte de cada entrega, determinándose como fecha de entrega final 22/05/2017. Se lograron definir las actividades a implementar durante la fase de diseño mediante las tarjetas CRC. Se describió el funcionamiento del sistema, y cada una de sus funcionalidades, así como el empleo de estándares de codificación y patrones de diseño, los cuales, permitieron realizar un diseño organizado y de fácil comprensión para otros desarrolladores.

Capítulo 3. Validación y resultados del sistema.

En este capítulo se desarrollarán las fases de implementación y las pruebas al sistema propuestas por la metodología de desarrollo escogida. Las pruebas según la metodología XP se dividirán en dos grupos: pruebas unitarias, encargadas de verificar el código, la cual es diseñada por los programadores, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al terminar una iteración se consiguió la funcionalidad requerida, estas últimas son diseñadas por el cliente final (61). Además, en el caso de la fase de implementación se expondrán los artefactos que se obtuvieron en esta, tales como: las tareas ingenieriles identificadas por cada historia de usuario en la fase de planificación.

3.1 Fase de Implementación.

En esta fase se especifica la implementación de las HU en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. El primer paso es hacer un reconocimiento de cada HU a la par con el plan de iteraciones, pudiendo ser modificados si fuera necesario; una vez identificadas las HU, los programadores procederán a descomponer cada una en tareas específicas, para esto se crearán las tareas de ingeniería, para así poder organizar la implementación de cada una de las HU. Estas tareas, a diferencia de las HU las cuales son escritas en el lenguaje del cliente, son realizadas por el programador en un lenguaje técnico. Como parte de la planificación elaborada en el capítulo anterior se definen a continuación las iteraciones del sistema.

Iteración 1

En esta iteración se implementaron las funcionalidades que permiten reconocer las imágenes provenientes del Pentacam y realizar el proceso de segmentación manual por un especialista.

Tabla 6: Tiempo de implementación de las HU de la iteración #1

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Cargar imagen de tomograma	2	2
Reconocer imagen de tomograma	2	2
Segmentar imagen "ground truth"	2	2

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

Tabla 7: Tarea de ingeniería de la funcionalidad Cargar imagen de tomograma.

Tarea	
Número de tarea: 1	Número de HU: 1
Nombre: Cargar imagen de tomograma	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Ruben Hayron Rodríguez Espinosa	
Descripción: Carga de un fichero seleccionado por el usuario la imagen del tomograma.	

Tabla 8: Tarea de ingeniería de la funcionalidad Reconocer imagen de tomograma.

Tarea	
Número de tarea: 2	Número de HU: 2
Nombre: Reconocer imagen de tomograma	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Ruben Hayron Rodríguez Espinosa	
Descripción: Si la imagen procede del Pentacam la recorta.	

Tabla 9: Tarea de ingeniería de la funcionalidad Segmentar imagen "ground truth".

Tarea	
Número de tarea: 3	Número de HU: 3
Nombre: Segmentar imagen "ground truth"	
Tipo de tarea: desarrollo	Puntos de estimación: 2
Programador responsable: Ruben Hayron Rodríguez Espinosa	
Descripción: Brinda la posibilidad al usuario de segmentar la imagen manualmente.	

Iteración 2

Tabla 10: Tiempo de implementación de las HU de la iteración #2

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Calcular medidas estadísticas de error	2	2
Calcular medidas de localización	2	2
Calcular métrica Hausdorff	2	2

Las tareas de ingeniería generadas por la iteración 2 se encuentran en el apartado “Anexo # 2”.

Iteración 3

Tabla 11: Tiempo de implementación de las HU de la iteración #3

Historias de usuario	Tiempo de implementación	
	Estimación	Real
Determinar mejor segmentación	2	2
Determinar algoritmo más robusto	2	2
Gestionar resultados	2	2
Graficar resultados	2	2

Las tareas de ingeniería generadas por la iteración 3 se encuentran en el apartado “Anexo # 2”.

3.2 Interfaz del Programa analizador de segmentación de imágenes

A continuación, se muestran las interfaces principales del software PASI (Programa analizador de segmentación de imágenes), así como la descripción de sus elementos y funcionalidades.

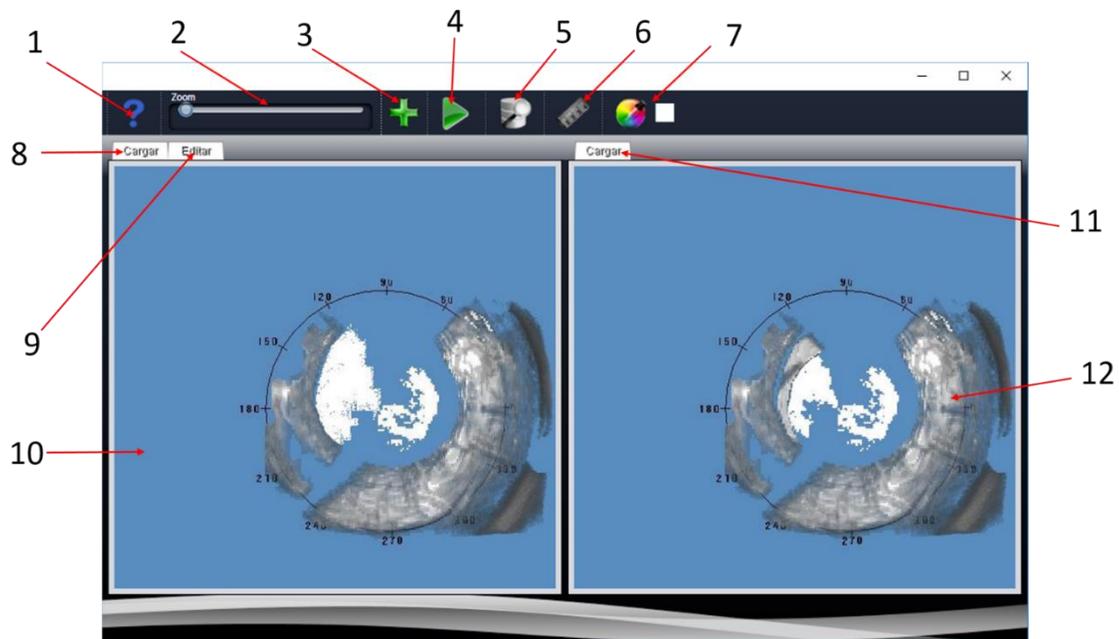


Figura 10: Interface principal del software PASI.

1. Muestra la ayuda del software.
2. Permite ampliar la imagen en la proporción seleccionada por el usuario.
3. Permite insertar un nuevo algoritmo al sistema y consultar la descripción de los mismos.
4. Permite iniciar el cálculo de las métricas.
5. Permite consultar la tabla de resultados de las métricas.
6. Permite ponderar por nivel de importancia para el usuario cada métrica.
7. Permite obtener el color de segmentación de la imagen segmentada automáticamente con el objetivo de que al segmentar manualmente la imagen "ground truth" coincidan los colores de segmentación y se obtengan resultados válidos.
8. Permite cargar la imagen "ground truth" pre-segmentada o segmentarla manualmente.
9. Permite desplegar la ventana de edición de la imagen "ground truth".
10. Permite visualizar la imagen "ground truth".
11. Permite cargar la imagen segmentada manualmente.
12. Permite visualizar la imagen segmentada automáticamente.

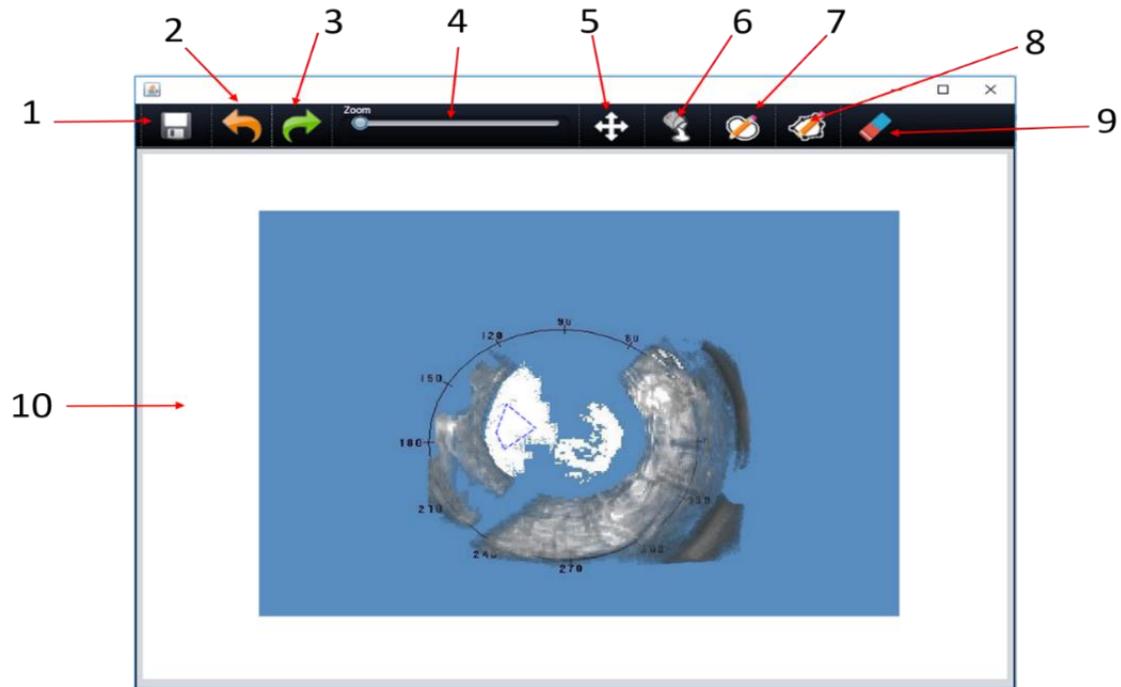


Figura 11: Panel de edición del software PASI.

1. Permite salvar la imagen segmentada manualmente "ground truth" en un archivo seleccionado por el usuario.
2. Permite obtener la imagen modificada anteriormente en el historial de modificación de la imagen.
3. Permite obtener la imagen posterior en el historial de modificación de la imagen.
4. Permite ampliar la imagen en la proporción seleccionada por el usuario.
5. Permite mover la imagen dentro del panel de edición.
6. Permite segmentar la región previamente seleccionada.
7. Permite seleccionar la región a segmentar dentro de la imagen de forma libre.
8. Permite seleccionar la región a segmentar dentro de la imagen punto a punto.
9. Permite borrar los puntos de selección en la imagen.
10. Permite visualizar la imagen en proceso de edición.

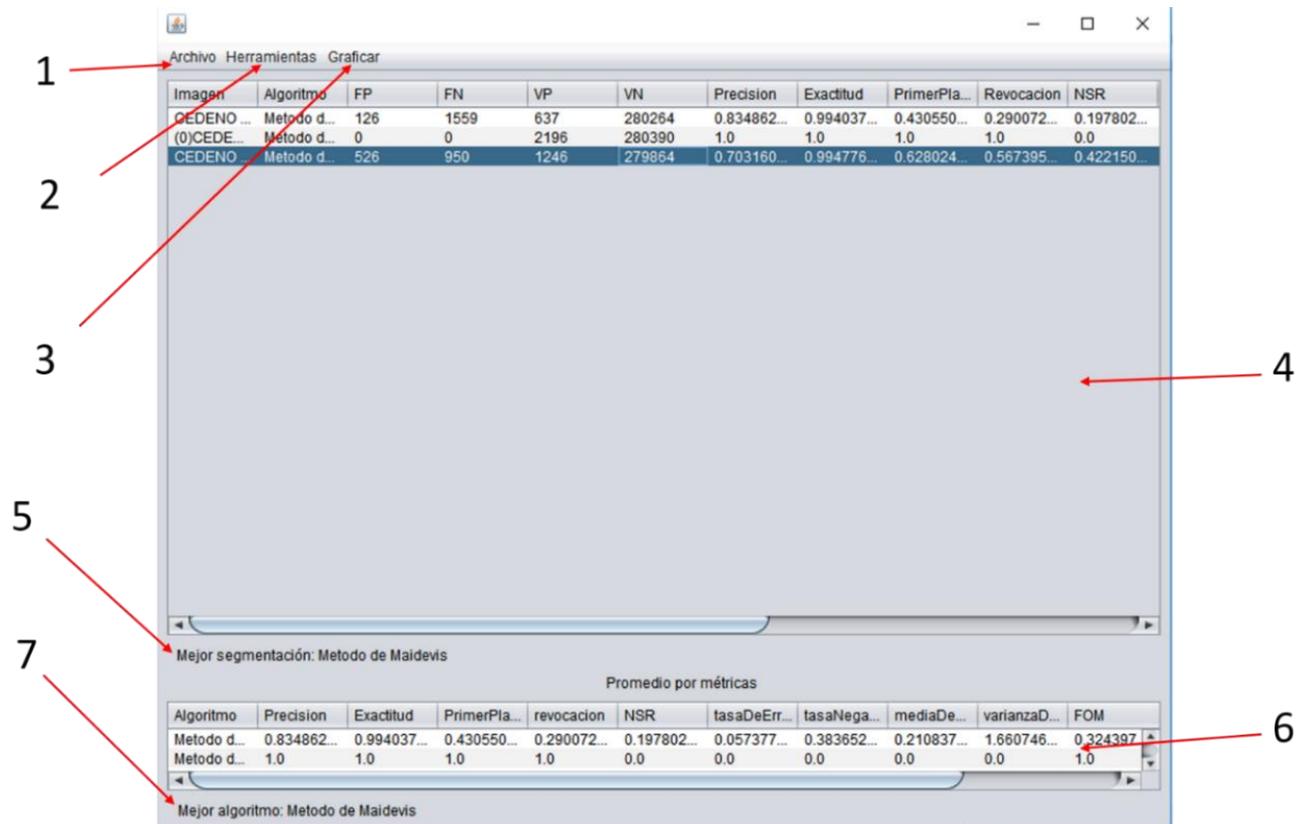


Figura 12: Ventana de visualización de los resultados de las métricas.

1. Permite exportar e importar los resultados de las métricas en un archivo Excel.
2. Permite eliminar la tabla de resultados o una fila seleccionada por el usuario.
3. Permite graficar los resultados de las métricas tras seleccionar una fila, por imagen, por algoritmo, o los de la fila seleccionada.
4. Permite visualizar los resultados de las métricas.
5. Permite visualizar el nombre del algoritmo mejor segmentado.
6. Permite visualizar los promedios por métricas entre los algoritmos probados.
7. Permite visualizar el nombre del mejor algoritmo entre los probados.

3.3 Pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas (62). Las pruebas de software son un elemento crítico para garantizar la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación (63). XP exhorta constantemente a realizar pruebas tanto como sea posible, permitiendo aumentar la calidad de los sistemas, reduciendo el número de errores no detectados (61). La

metodología XP divide las pruebas del software en dos grupos: pruebas unitarias, encargadas de verificar el código de manera automática y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida en las historias de usuario diseñadas por el cliente final (64).

3.3.1 Pruebas unitarias

Una prueba unitaria es la verificación de una unidad de código(módulo) determinado dentro de un sistema. Estas pruebas nos aseguran que un módulo determinado cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema, es decir, debe aprobar satisfactoriamente todos los casos de prueba definidos. El uso de estas pruebas evita tener que invertir una gran cantidad de horas en sesiones de “*debugging*” al momento de integrar el código con el sistema existente. En la metodología XP los programadores deben definir las pruebas unitarias para cada módulo antes de escribir el código, y sólo para aquellas unidades de código donde exista la posibilidad de que puedan fallar, es necesario escribir casos de prueba (65).

A continuación, se describirá la confección de las pruebas unitarias realizadas, desglosándola en sus elementos fundamentales: confección del grafo de flujo, cálculo de la complejidad ciclomática, extracción de los caminos independientes, realización de los casos de pruebas y análisis de los resultados obtenidos. Estas pruebas se aplican al método “mostrarMejorAlgoritmo” perteneciente a la clase “NewJFrame”, el cual nos retorna el nombre del mejor algoritmo probado. A este se le aplica la métrica de complejidad ciclomática debido a su relevancia para el sistema.

```

public int[][] binarizarBufferedImage(BufferedImage bi,Color color){

    WritableRaster raster = bi.getRaster();//1
    //Lleno la imagenMatriz
    double ancho = bi.getWidth();//1
    double alto = bi.getHeight();//1
    int[][]imagenMatriz=new int[(int) alto][(int) ancho];//1

    for (int i = 0; i < (int) alto; i++) {//2
        for (int j = 0; j < (int) ancho; j++) {//3
            //formato de colores del 0 - 255.
            if (raster.getSample(j, i, 0)==color.getRed() &&
                raster.getSample(j, i, 1)==color.getGreen() &&
                raster.getSample(j, i, 2)==color.getBlue()) {//4
                imagenMatriz[i][j]= 1; //5
            }
            else{
                imagenMatriz[i][j]= 0;//6
            }
        }
    }
    return imagenMatriz;//9
}

```

Figura 13: Algoritmo para binarizar las imágenes de entrada.

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

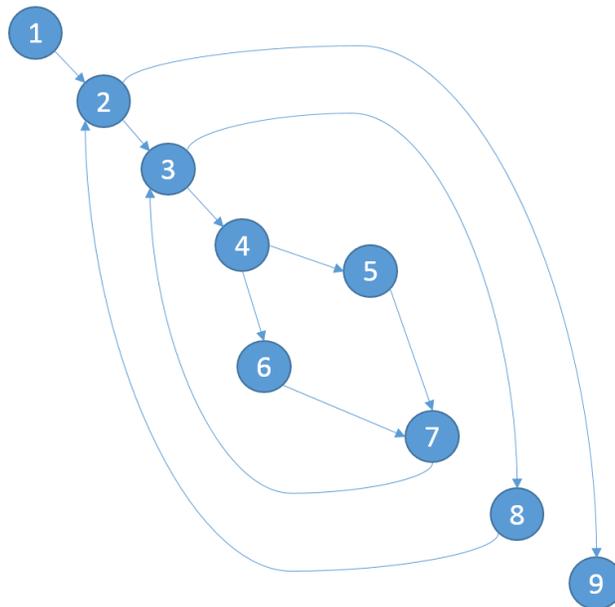


Figura 14: Diagrama de flujo del algoritmo “binarizarBufferedImage”.

Tras definir el grafo de flujo se procede al cálculo de la complejidad ciclomática siendo esta una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa (66). Para esta operación existen tres vías de solución, las cuales se enuncian a continuación:

- $V(G)=(A-N) +2$
- $V(G)=P +1$
- $V(G)=R$

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , se define como:

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

La complejidad ciclomática, $V(G)$, de un grafo de flujo G , también se define como:

$V(G)= P + 1$, donde P es el número de nodos predicados contenidos en el grafo de flujo G .

$V(G)= R$, donde R es la cantidad total de regiones.

Del grafo de flujo G se obtiene que:

$$V(G)= (A-N) +2 = (11-9) +2 =4$$

$$V(G)=P +1=3+1=4$$

$$V(G)= R=4$$

Tras aplicar las tres formas para afirmar un resultado seguro y confiable, se obtiene una complejidad ciclomática $V(G)=4$ la cual representa la cantidad de caminos independientes para el grafo de flujo construido para el método "binarizarBufferedImage", por lo que se definen los 4 caminos siguientes:

Tabla 12: Caminos básicos

Número	Caminos básicos
1	1-2-9
2	1-2-3-8-2-9
3	1-2-3-4-5-7-3-8-2-9
4	1-2-3-4-6-7-3-8-2-9

Luego de tener elaborado el grafo de flujo e identificados los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos. Con el fin de comprobar cada camino, se escogen los datos de manera que las condiciones de los nodos predicados estén adecuadamente establecidas. A continuación, se especifican estos casos de prueba.

Tabla 13: Caso de prueba del camino básico #1

Caso de Prueba: Camino Básico # 1
Descripción: Este método dada una imagen, devuelve la imagen binarizada.
Condiciones de ejecución: Se debe haber cargado la imagen antes.
Entradas\Pasos de ejecución: La imagen inicial y el objeto color.
Resultado esperado: La matriz de la imagen binarizada con valores 0 y 1.
Evaluación de la prueba: Satisfactoria.

3.3.2 Pruebas de aceptación

Cada prueba de aceptación o prueba funcional como también se le conoce está asociada a una historia de usuario, estas pruebas se realizan en esta etapa del proyecto describiéndose en ellas las posibles formas de utilización del software. Las pruebas funcionales no sólo validan la transformación de una entrada en una salida, sino que validan una característica completa (64). En estos documentos de prueba se indican las posibles respuestas que da el software tras la utilización de cada funcionalidad, así como

los posibles mensajes de error, información o de aceptación que emite el software cuando se utiliza dicha funcionalidad (67).

Para la descripción de estas pruebas se presenta el diseño del caso de prueba correspondiente a la historia de usuario “Determinar mejor algoritmo” que incluye funcionalidades críticas para el sistema. La siguiente tabla muestra los resultados satisfactorios obtenidos en la prueba de aceptación “Determinar mejor algoritmo”.

Tabla 14: Caso de prueba de aceptación de la HU #7

Caso de prueba de aceptación	
Código: HU7_P7	Historia de Usuario: 7
Nombre: “Determinar mejor algoritmo”	
Descripción: Prueba para la funcionalidad “Determinar mejor algoritmo”.	
Condiciones de Ejecución: Se debe tener al menos el resultado de las métricas de dos algoritmos de segmentación.	
Resultado Esperado: El sistema devuelve el nombre del mejor algoritmo.	
Evaluación de la Prueba: Prueba satisfactoria.	

Como parte de estas pruebas se procedió a la creación de 15 casos de estudio para verificar los resultados del software, en los mismos se realizaron 5 comparaciones por cada uno de los tres algoritmos desarrollados por el grupo AIRI. A continuación, se presenta una muestra de 1 caso de estudio, donde se muestran los resultados de las métricas entre dos imágenes completamente iguales, dos completamente diferentes y el resultado del algoritmo propuesto por Paula Vidal con el fin de comparar los valores de

las métricas. Las pruebas realizadas a este conjunto de imágenes provenientes del Pentacam arrojaron los siguientes resultados:

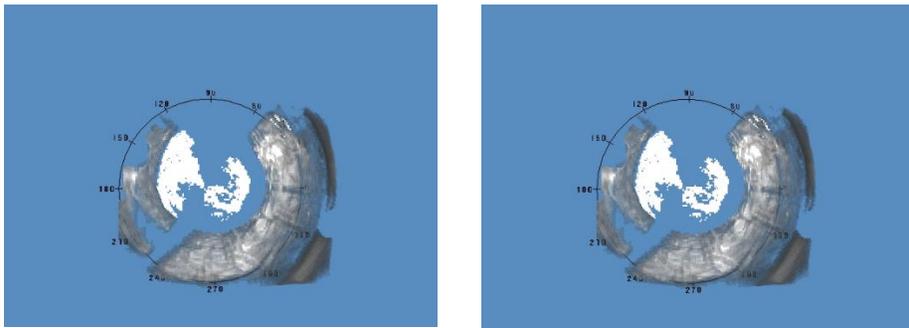


Figura 15: A la izquierda el "Ground Truth" y al lado la segmentación automática.

Tabla 15: Resultados de las pruebas del caso de estudio #1

Imagen	Algoritmo	FP	FN	VP	VN	Precisión	Exactitud	Primer Plano
Paciente 1	Carlos & Pedro	0	0	7093	275493	1.0	1.0	1.0
Revocación	NSR	Tasa de Error	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

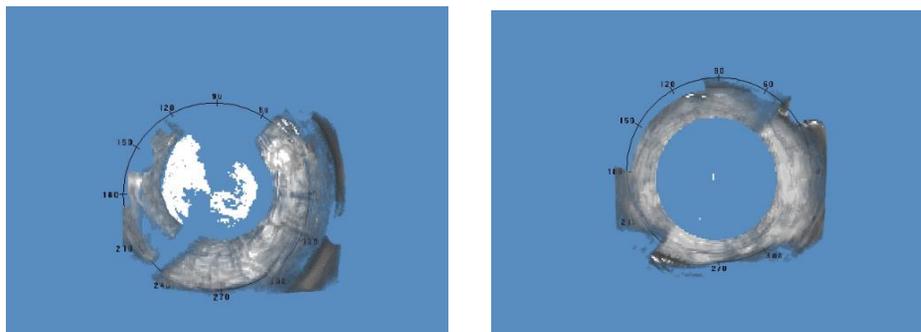


Figura 16: A la izquierda el "Ground Truth" y al lado la segmentación automática

Tabla 16: Resultados de las pruebas del caso de estudio #1

Imagen	Algoritmo	FP	FN	VP	VN	Precisión	Exactitud	Primer Plano
Paciente 2	Carlos & Pedro	47	7088	5	275446	0.096	0.974	0.001
Revocación	NSR	Tasa de Error	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
7.049	9.4	0.006	0.502	0.156	6.996	0.002	85	30.392

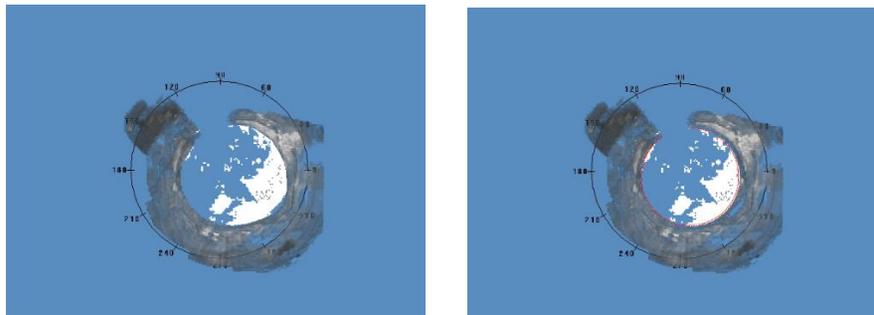


Figura 17: A la izquierda el "Ground Truth" y al lado la segmentación automática

Tabla 17: Resultados de las pruebas del caso de estudio #1

Imagen	Algoritmo	FP	FN	VP	VN	Precisión	Exactitud	Primer Plano
Paciente 2	E. Paula V	75	1429	4847	276235	0.984	0.994	0.865
Revocación	NSR	Tasa de Error	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
0.772	0.015	0.011	0.119	0.032	0.253	0.780	24	3.577

De manera general los resultados de las pruebas se comportaron de la siguiente forma:

Tabla 18: Resultado de las pruebas de aceptación

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

Las pruebas se realizaron de forma iterativa e incremental, y se comprobó en cada iteración la corrección de los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del sistema, por lo que el sistema quedó aprobado por parte del cliente, quien evaluó el 100 % de los casos de prueba de forma satisfactoria.

3.4 Resultados

Luego de efectuar las pruebas correspondientes para validar la fiabilidad del sistema, se procedió a realizar la comparación de las métricas de error y los tiempos de ejecución entre los tres algoritmos de segmentación propuestos por el grupo AIRI, con el fin de seleccionar el algoritmo más eficiente para la detección de la opacidad de la capsula posterior en imágenes provenientes de tomogramas Scheimpflug del PENTACAM. Estas comparaciones fueron ejecutadas sobre un conjunto de 28 imágenes de pacientes con esta complicación postoperatoria por cada uno de los algoritmos propuestos. A continuación, se muestran los resultados de los promedios de las métricas por algoritmos, así como los tiempos de ejecución de cada uno de ellos:

Tabla 19: Resultado de las métricas para el algoritmo de Alonso P y Machado S

Algoritmo	Precisión	Exactitud	Primer Plano	Revocación	NSR	Tasa de Error
Alonso P y Machado S (69)	0.840	0.984	0.685	0.695	0.845	0.818
	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
	0.561	20.503	713.343	0.593	39.714	13.490

Tabla 20: Resultado de las métricas para el algoritmo de Rodríguez M

Algoritmo	Precisión	Exactitud	Primer Plano	Revocación	NSR	Tasa de Error
Rodríguez M (70)	0.651	0.978	0.604	0.681	1.619	1.451
	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
	0.884	31.977	1126.219	0.655	39.071	12.967

Tabla 21: Resultado de las métricas para el algoritmo de E. Paula V

Algoritmo	Precisión	Exactitud	Primer Plano	Revocación	NSR	Tasa de Error
E. Paula V (71)	0.927	0.993	0.890	0.892	0.111	0.103
	Tasa Negativa	Media	Varianza	FOM	Hausdorff	Baddeley
	0.105	1.295	35.925	0.864	29.785	8.060

Tabla 22: Resultado de los tiempos de complejidad media por algoritmo

Algoritmo	Promedio (ms)
Método propuesto por Paula Vidal (2016)	4326,96
Método propuesto por Rodríguez Moreno (2015)	98994,5
Método propuesto por Alonso & Machado (2016)	5946,03

Tras constatar los resultados de los promedios de las métricas por cada algoritmo y sus respectivos tiempos de ejecución se pudo apreciar que el más eficiente para la detección de la opacidad de la capsula posterior en imágenes provenientes de tomogramas Scheimpflug del PENTACAM entre los tres algoritmos es el propuesto por E. Paula Vidal en el 2016 proporcionando los mejores valores en cuanto a métricas y en cuanto a tiempos de ejecución.

3.5 Validación de los resultados

La validación de un sistema de validación es algo complejo, de ahí la importancia de contar con buenos criterios a la hora de arribar a conclusiones sobre los resultados obtenidos. En este trabajo los especialistas afirman que no existen formas de experimentación bien establecidas, por tal motivo, como parte de las pruebas realizadas al sistema, se hizo necesaria aplicar la técnica de juicio de experto, ya que se requiere de la valoración de los especialistas, para verificar los resultados de las métricas proporcionando una valoración de cada uno de los casos de estudio.

El juicio de expertos es un método de validación útil para verificar la fiabilidad de una investigación que se define como una opinión informada de personas con trayectoria en el tema, que son reconocidas por otros como expertos calificados, y que pueden dar información, evidencia, juicios y valoraciones (68)

Panel de expertos

Integran el panel de expertos el Ingeniero en Ciencias Informáticas Michel Álvarez Cancio profesor de la Universidad de las Ciencias Informáticas, el Doctor en Ciencias Rubén Orozco Morales profesor titular de la Universidad Central de las Villas(UCLV) y la Doctora en Ciencias Silena Herold Garcia profesora titular de la Universidad de Oriente(UO), todos

cuentan con gran experiencia y un amplio número de publicaciones en el campo del procesamiento digital de imágenes.

Criterios de los expertos

Para la evaluación del panel de expertos sobre los casos de pruebas realizadas se analiza la evaluación de los resultados de las métricas de la siguiente forma:

- 5: Totalmente de acuerdo con el resultado de las métricas.
- 4: Concuero con la mayoría de los resultados de las métricas.
- 3: Concuero medianamente con el resultado de las métricas.
- 0: No estoy de acuerdo con el resultado de las métricas.

Evaluación final de los expertos

Tabla 23: Evaluación de los expertos por imágenes

Cantidad de Imágenes	Evaluación
18	5
9	4
1	3
Promedio	4,6

Luego de un análisis realizado por el grupo de expertos, arribaron a la conclusión de que el sistema cumple con el objetivo propuesto, ya que de las imágenes analizadas solo 1 fue evaluada de 3 y ninguna de ellas fue evaluada de 0, logrando que la mayoría de los resultados fueran de nivel 5, con un promedio de evaluación de 4.6, lo que permite afirmar que las métricas de evaluación cumplen con los requerimientos especificados.

3.6 Conclusiones del capítulo

Durante el desarrollo del presente capítulo se abordaron los temas referentes a las fases de implementación y pruebas al sistema. En esta última se realizaron las pruebas de aceptación y las pruebas unitarias, presentándose los casos de prueba correspondientes para validar las funcionalidades del software propuesto, las cuales arrojaron resultados satisfactorios, lo que significó, que las historias de usuarios implementadas cumplen con los requisitos y necesidades del cliente. Con la validación realizada a la solución desarrollada se pudo efectuar la comparación entre los tres algoritmos de segmentación propuestos por el grupo AIRI, mostrando como mejor algoritmo de segmentación de

imágenes provenientes de tomogramas Scheimpflug del PENTACAM el método propuesto por E. Paula Vidal (71). Con el desarrollo de estas pruebas se logró una cobertura elevada, garantizando así la calidad del sistema construido probándose con esto que el software PASI puede ser utilizado en un entorno real.

Conclusiones generales

Con el desarrollo de este trabajo se cumplieron todas las actividades propuestas, respondiendo a la necesidad de buscar una solución al problema general planteado, arribando a las siguientes conclusiones:

- El cálculo de métricas de error de la segmentación de una imagen, puede ser utilizado para diferenciar la calidad de la segmentación de un algoritmo.
- Se implementaron las métricas estadísticas de error, medidas de localización y métricas Hausdorff para la validación de la segmentación.
- Con el uso de la solución, se dotó al Grupo de investigación AIRI de una herramienta de apoyo al análisis de la segmentación de imágenes de tomogramas Scheimpflug del PENTACAM.
- Mediante las pruebas realizadas a un conjunto de 28 imágenes de tomogramas Scheimpflug del PENTACAM se pudo obtener que el algoritmo más estable para la detección de la OCP es el propuesto por Paula Vidal 2016 atendiendo a que brindo los mejores resultados en cuanto a métricas estadísticas de error, medidas de localización y métricas Hausdorff.

Recomendaciones

Se recomienda para versiones futuras:

- Implementar otras métricas para la validación de la segmentación de imágenes de manera general.
- Adaptar al software PASI para validar la segmentación en todo tipo de imágenes.

Glosario de términos

Capsulorrexia: Maniobra realizada en la cirugía de catarata donde se crea una abertura circular continua en la cápsula anterior del cristalino.

Sistema Scheimpflug: Sistema de cámaras de alta definición acopladas a diferentes equipos médicos como el Pentacam.

Lente intraocular: Lente artificial que se coloca dentro del ojo para sustituir el cristalino extraído.

Pentacam: Equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo.

Saco capsular: Estructura que rodea el cristalino dándole soporte.

Convolución: Es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen f y una versión trasladada e invertida de g .

Retina: Membrana interior del ojo en la cual se reciben las impresiones luminosas que son transmitidas al cerebro; cubre la coroides hasta el iris y está formada esencialmente por expansiones del nervio óptico.

Lámpara de hendidura: Es un microscopio de bajo poder combinado con una fuente de luz de alta intensidad que puede enfocarse como un rayo delgado.

Pixel: Unidad básica de una imagen digitalizada en pantalla a base de puntos de color o en escala de grises.

Histéresis: Es la tendencia de un material a conservar una de sus propiedades, en ausencia del estímulo que la ha generado. Podemos encontrar diferentes manifestaciones de este fenómeno.

Referencias

1. *Revisión sistemática de estudios poblacionales de prevalencia de cataratas*. R Acosta, L Hoffmeister, R Román. 509-516, Barcelona : Ingeniería Mecánica, 2006, Vol. 81.
2. *Estrategias de prevención de la opacidad de la cápsula posterior*. .Hernández López, Iván, y otros. s.l. : Revista Cubana de Oftalmología, 2010, Vol. 23.
3. *PENTACAM tomograms: A Novel Method for Quantification of Poste*. D Grewall, R Jain. 2008, Vol. 49.
4. Alvarez Cancio, Michel, y otros. *PANDOC: SOFTWARE PARA LA CUANTIFICACIÓN OBJETIVA DE LA OPACIDAD DE LA CÁPSULA POSTERIOR MEDIANTE TOMOGRAMAS SCHEIMPFLUG DEL PENTACAM*. La Habana : s.n., 2013.
5. Moreno, Maidevis Rodríguez. *Desarrollo de un sistema basado en casos para la identificación de opacidad de la cápsula posterior en el software PANDOC*. La Habana : s.n., junio del 2015.
6. *APLICACIÓN DE UN SISTEMA BASADO EN CASOS PARA LA IDENTIFICACIÓN DE OPACIDAD EN LA CÁPSULA POSTERIOR MEDIANTE IMÁGENES DEL PENTACAM*. Alvarez Cancio, Michel, Cruz de la Osa, Reyder y Hernández López, Iván. 2016, INFORMÁTICA 2016.
7. Alonso Peña, Carlos Andrés, Machado Sanchez, Pedro Luis y Alvarez Cancio, Michel. *Sistema de cuantificación de opacidad capsular mediante Componentes Conexas*. Facultad 3, Universidad de las Ciencias Informáticas. 2016. Trabajo para optar por el título de Ingeniero en Ciencias Informáticas.
8. Alonso Peña, Carlos A. *Sistema de cuantificación de opacidad capsular mediante componentes conexas*. Facultad 3, UCI. La Habana : s.n., 2016. Tesis para opetar por el título de Ingenieroen Ciencias Informáticas.
9. Paula Vidal, Erlis, Álvarez Cancio, Michel y Cruz de la Osa, Reyder. *Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM*. La Habana : s.n., Junio, 2016.
10. *Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM*. CANCIO, Michel Alvarez, et al. 1, La Habana : Ediciones Futuro, Enero-Marzo de 2017, Revista Cubana de Ciencias Informáticas, Vol. 11, págs. 153-167. ISSN: 2227.
11. Woods, R., & González, R. *Digital Image Processing Second Edition*. . 2002.
12. *Métodos de Segmentación de Imágenes Médicas*. Coto, Ernesto. Caracas,Venezuela : s.n., Marzo, 2003. 1316-6239.

13. *Avances en el diagnóstico por imágenes. A step forward in imaging diagnosis.* . LLANES, Mabel Reyes. 2, s.l. : Revista de Ciencias Médicas de Pinar del Río, 2000, Vol. 4.
14. *Revista de Ingeniería de Sistemas e Informática.* 2, Julio - Diciembre 2009, Vol. 6.
15. Elizondo, M. C. José Jaime Esqueda. *Fundamentos de Procesamiento de Imágenes Digitales.* California : CONATEC, 2002.
16. Aldana, Jeidy Rosell. Algoritmo para la detección y realce de bordes en imágenes oblicuas provenientes de las lámparas de hendidura. La Habana : s.n., Junio, 2016.
17. JUAN PABLO URREA, EMMANUEL OSPINA. *IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN DE LÍNEAS PARA UN SISTEMA DE VISIÓN DE BAJO NIVEL.* 2004.
18. José Abel de la Fuente-Arriaga, Edgardo Felipe-Riverón, Eduardo Garduño-Calderón. *Segmentación del disco óptico en imágenes de retina mediante la transformada de Hough y los contornos activos.* Estado de México, México : s.n., 2012.
19. Gómez, Jesús Tejedor. *Análisis Comparativo de Algoritmos en Segmentación de Iris.* Madrid : s.n., 2009.
20. T. Hermosilla, E. Bermejo, A. Balaguer, L.A. Ruiz. *Detección de bordes con precisión subpíxel en imágenes digitales: Interpolación lineal frente a esquemas de tipo no lineal.* Valencia, España : s.n., 2006.
21. *Image segmentation by using edge detection.* Saleh Alamri, M. S., Kalyankar, D. N., & Khamitkar, S. 3, M.S., India : s.n., 2010, Vol. 2.
22. *Implementacion de un algoritmo para la deteccion y conteo de celulas en imagenes microscopicas.* Valencia Delgado, V., Moreira, J., & Chavez, P. Guayaquil, Ecuador : s.n., 2005.
23. *A Review of Classic Edge Detectors.* Spontón, Haldo y Cardelino, Juan. Uruguay : s.n., 2015. 2105–1232.
24. *A Comparison of various Edge Detection Techniques used in Image Processing .* Shrivakshan, G.T. y Chandrasekar, Dr.C. 1, Tamilnadu, India : s.n., September 2012, Vol. 9. 1694-0814.
25. *IMPLEMENTACION DE ALGORITMOS PARA PREPROCESAMIENTO, DETECCIÓN DE BORDES Y SEGMENTACIÓN DE IMÁGENES DIGITALES A COLOR UTILIZANDO MATLAB.* PILAR, CHIRIBOGA MALDONADO MARÍA DEL y ANDREA, SALCEDO ECHES EDITH. Quito : s.n., Junio 2003.
26. *COMPARATIVE STUDY OF IMAGE EDGE DETECTION ALGORITHMS.* Saini, Shubham, Kasliwal, Bhavesh y Bhatia, Shraey. India : s.n., 2010.
27. *Canny edge detection tutorial.* GREEN, Bill. March, 2002, Vol. vol. 6.

28. MEJORA DE UN ALGORITMO DE SEGMENTACIÓN DE IRIS MEDIANTE CONTORNOS ACTIVOS. Corral, David Alcón. Madrid, España : s.n., Julio 2012.

29. IMPLEMENTACIÓN Y EVALUACIÓN DE ALGORITMOS DE IDENTIFICACIÓN POR RECONOCIMIENTO DEL IRIS OCULAR. MARMOLEJO, LUIS FELIPE CAICEDO y CARVAJAL, CARLOS EDUAR CHAMORRO. Cali, Colombia : s.n., 2012.

30. Redondo, Lic Rocío Sanchiz. **SEGMENTACIÓN DE IRIS MEDIANTE CONTORNOS ACTIVOS.** . 2011.

31. Vargas, Dc Valentín Masero. *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.* . 2015.

32. Sistemas de razonamiento basado en casos aplicado a sistemas de líneas de productos de software. Augusto Cortez Vásquez, Carlos Navarro Depaz, Jaime Pariona Quispe. 2, 2010, Vol. 7.

33. Modelo para diseñar sistemas de enseñanza-aprendizaje inteligentes utilizando el razonamiento basado en casos. Natalia Martínez Sánchez, María Matilde García Lorenzo, Zoila Zenaida García Valdivia. 3, Colombia : s.n., 2009, Vol. 6.

34. Razonamiento Basado en Casos: Una Visión General. Laura Lozano, Javier Fernández. España : s.n., 2008.

35. Un sistema de razonamiento basado en casos para la clasificación de fallos en sistemas dinámicos. Anibal Bregón, Arancha Simón, Carlos Alonso, Belarmino Pulido, Isaac Moro ,Juan José Rodríguez. 2005.

36. UN SISTEMA BASADO EN CASOS PARA LA TOMA DE DECISIONES EN CONDICIONES DE INCERTIDUMBRE. Iliana Gutiérrez Martínez, Rafael E. Bello Pérez y Andrés Tellería Rodríguez. 2, Santa Clara : s.n., 2002, Vol. 23.

37. Arámbula, Fernando. *Análisis de imágenes médicas.* 2008.

38. Handbook of Medical Imaging. M.Sonka y J.M.Fitzpatrick. s.l. : SPIE Press, 2000, Vol. 2. ISBN 0819477605.

39. López, María del Mar Pujol. *Incorporación de características en la función de energía para segmentación de imágenes usando campos aleatorios de Markov.* 2000.

40. Machado, Nery Javier. **SEGMENTACIÓN DE MELANOMA BASADO EN METODOS DE BINARIZACION.** SAN LORENZO, PARAGUAY : s.n., Diciembre, 2014.

41. EVALUACIÓN DE MÉTODOS DE SEGMENTACIÓN APLICADOS A IMÁGENES DE MUESTRAS DE SANGRE. Deniz, Arniel Labrada y García, Silena Herold. Santiago de Cuba, Cuba : s.n., 2016.

42. *Valmet: A new validation tool for assessing and improving 3D object segmentation.* Gerig, Guido, Jomier, Matthieu y Chakos, Miranda. USA : s.n.
43. *Errors in binary images and an Lp version of the Hausdorff metric.* Baddeley, A.J. Netherlands : Nieuw Archief Voor Wiskunde, 1992.
44. Cancio, Michel Alvarez. *PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad capsular de un paciente operado de catarata.* La Habana : s.n., 2014.
45. *Metodologías de desarrollo de software.* Sánchez Mendoza, M. A. 2004.
46. Canós José H, Letelier Patricio, Panadés M Carmen. *Métodologías Ágiles en el Desarrollo de Software.* . Valencia : s.n., 2003.
47. Joskowicz, J. *Reglas y prácticas en eXtreme Programming.* . 2008.
48. *Metodología Actual: Metodología XP.* . Bustamante, D., & Rodriguez, J. 2014.
49. Suarez, M. L. Cuaderno 1: Introducción a la programación y sus lenguajes. *Competencias en TIC: Colección de Fascículos Digitales.* 2015.
50. Oracle. Java. [En línea] [Citado el: 14 de 1 de 2015.] [http://www.java.com/es/about/..](http://www.java.com/es/about/)
51. Visual, Paradigm. Visual Paradigm. [En línea] 2016. [Citado el: 30 de 3 de 2016.] <https://www.visual-paradigm.com..>
52. SQLite. *SQLiteManagenet.org.* [En línea] [Citado el: 5 de 2 de 2015.] <http://www.sqlitemanager.org>.
53. Admin, SQLite. *SQLite Administrador.* [En línea] [Citado el: 10 de 2 de 2015.] sqliteadmin.orbmu2k.de.
54. Gamma, E, Helm, R y J. R., & Vlissides, J. *Elements of Design Patterns.* s.l. : Addison-Wesley, 1995.
55. Mühlrad, D. *Patrones de diseño.* . 2008.
56. *Una explicación de la programación extrema: aceptar el cambio.* Beck, Kent. s.l. : Addison Wesley, 2002.
57. *Applied Software Project Management.* Stellman, Andrew y Greene, Jennifer. 113, s.l. : O'Reilly Media, 2005. ISBN 978-0-596-00948-9.
58. *Embracing change with extreme programming.* BECK, K. 1999.
59. *Aspectos tempranos : un enfoque basado en Tarjetas CRC.* Casas, S y Reinaga, H. Argentina : s.n., 2009.

60. Garcia, O. Nomenclatura estándar del código Java. . [En línea] 22 de 8 de 2011.
<http://www.elclubdelprogramador.com/2011/08/22/java-nomenclatura-estandar-del-codigo-java/>.
61. **PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA.** J .J . Gutiérrez, M. J. Escalona, M. Mejías, J. Torres. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. : s.n., 2010.
62. *Extreme Programming Explained.* . BECK, Kent. 1999.
63. *Las pruebas en metodologías ágiles y convencionales: papeles diferentes.* " . Yagüe, A. G., J. 2009.
64. Sommerville, Ian. *Ingeniería del software. Séptima Edición.* . Madrid. España : Pearson Educación. S. A., 2005. 84-7829-074-5..
65. Malfará, Dayvis, y otros. *Testing en eXtreme Programming.* 2006.
66. Pressman, R. *Ingeniería del Software un enfoque práctico.* . 2005.
67. *Métodologías Ágiles en el Desarrollo de Software.* Canós José H, Letelier Patricio, Panadés M Carmen. Universidad Politécnica de Valencia : s.n., 2003.
68. Garrote, Pilar Robles y del Carmen Rojas, Manuela. *La validación por juicio de expertos: dos investigaciones cualitativas en Lingüística aplicada.*
69. Carlos A, Alonso Peña y Pedro L, Machado Sánchez. *Sistema de cuantificación de opacidad capsular mediante Componentes Conexas.* Ciudad de la Habana : s.n., junio 2016.
70. Maidevis, Rodríguez Moreno. *Desarrollo de un sistema basado en casos para la identificación de opacidad de la cápsula posterior en el software PANDOC.* Ciudad de la Habana : s.n., junio 2015.
71. Erlis, Paula Vidal. *Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes provenientes del PENTACAM.* Ciudad de la Habana : s.n., Junio, 2016.
72. **ALGORITMOS DE DETECCIÓN DE IMÁGENES RUIDOSAS Y DUPLICADAS.** Hernanz, Irene Blasco. Madrid,España : s.n., Diciembre 2008.

Anexos

Anexo #1: Historias de usuario

Tabla 24: Historia de usuario #1

Historia de usuario	
Número: 1	Nombre: Cargar imagen de tomograma
Iteración asignada: 1	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema debe permitirle al usuario cargar una imagen desde cualquier directorio ubicado en su computadora.	
Observaciones:	

Tabla 25: Historia de usuario #2

Historia de usuario	
Número: 2	Nombre: Reconocer imagen de tomograma
Iteración asignada: 1	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema debe reconocer si la imagen a cargar es un tomograma Scheimpflug del Pentacam, y en caso de serlo la recorta sólo en la parte a analizar.	
Observaciones:	

Tabla 26: Historia de usuario #3

Historia de usuario	
Número: 3	Nombre: Segmentar imagen "ground truth"
Iteración asignada: 1	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema debe darle la posibilidad al usuario de segmentar manualmente la imagen "ground truth".	
Observaciones:	

Tabla 27: Historia de usuario #4

Historia de usuario	
Número: 4	Nombre: Calcular medidas estadísticas de error
Iteración asignada: 2	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema tras entrar el "ground truth" y la imagen segmentada automáticamente al presionar el botón "play" debe calcular las medidas estadísticas de error y mostrarlas en una tabla por sus nombres.	
Observaciones:	

Tabla 28: Historia de usuario #5

Historia de usuario	
Número: 5	Nombre: Calcular medidas de localización
Iteración asignada: 2	

Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema tras entrar el “ground truth” y la imagen segmentada automáticamente al presionar el botón “play” debe calcular las medidas de localización y mostrarlas en una tabla por sus nombres.	
Observaciones:	

Tabla 29:Historia de usuario #6

Historia de usuario	
Número: 6	Nombre: Calcular métricas de Hausdorff
Iteración asignada: 2	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema tras entrar el “ground truth” y la imagen segmentada automáticamente al presionar el botón “play” debe calcular las métricas Hausdorff y mostrarlas en una tabla por sus nombres.	
Observaciones:	

Tabla 30:Historia de usuario #7

Historia de usuario	
Número: 7	Nombre: Determinar mejor segmentación
Iteración asignada: 3	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas

Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema tras tener los resultados obtenidos en las comparaciones entre el valor medio de las métricas por cada algoritmo devuelve el nombre del mejor algoritmo entre ellos.	
Observaciones:	

Tabla 31:Historia de usuario #9

Historia de usuario	
Número: 9	Nombre: Gestionar resultados
Iteración asignada: 3	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema debe permitir al usuario exportar e importar los resultados de las métricas en un Excel, así como también debe permitir eliminar una tabla o una fila seleccionada.	
Observaciones:	

Tabla 32:Historia de usuario #10

Historia de usuario	
Número: 10	Nombre: Graficar resultados
Iteración asignada: 3	
Prioridad en negocio: Alta	Puntos estimados: 2 semanas
Riesgo en desarrollo: Alto	Puntos reales: 2 semanas
Descripción: El sistema debe permitir al usuario graficar los resultados de las métricas por fila, por imágenes y por algoritmos, así como también debe permitir exportar el grafico como imagen.	

Anexo #2: Pruebas de aceptación

Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: 01/06/2017

Emitida a favor de: Tesis "Módulo para la validación de la segmentación en imágenes de opacidad de la cápsula posterior provenientes del Pentacam".

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App: Módulo para la validación de la segmentación en imágenes de opacidad de la cápsula posterior provenientes del Pentacam.	1.0	0	2	Evaluación dinámica Pruebas de Funcionalidad	01/06/2017



Ruben Hayron Rodríguez Espinosa

Autor