

Detección de la capsulorrexis en imágenes de opacidad capsular usando un algoritmo de contornos activos.

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

Autor: Daniel Pérez Hernández

Tutor: Ing. Michel Álvarez Cancio

I

FACULTAD 3

junio del 2018

Año 60 de la Revolución

Declaración de autoría

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

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2018.

Autor:

Daniel Pérez Hernández

Tutor:

Michel Alvarez Cancio

Dedicatoria

Dedico mi trabajo de diploma a la persona más importante de mi vida a mi madre por ser la luz de mi vida, por darme su amor sin condición, así como su apoyo incondicional en todo momento. También a Surama Borges Zayas, por ser mi segunda madre por haber estado a mi lado en estos cinco años de la carrera.

AGRADECIMIENTOS

Quiero agradecer a mi tutor Michel por su apoyo durante el desarrollo de mi trabajo de diploma.

A los profesores que contribuyeron a mi formación como profesional en el transcurso de mi carrera, especialmente a Yanet, Maigrect, Rosalina, Dariela, Yadira, Olga Yarisbel, Mailen Edith, Lieen, Graciela y Reina Victoria.

A mi familia:

A mi madre por ser mi ejemplo a seguir, porque es la razón de mi existencia, por darme su amor sin condición, así como su apoyo incondicional, te quiero muchísimo....

A mi hermano preferido Jose Manuel que lo quiero muchísimo , espero servirte de ejemplo y que la vida nos permita estar siempre juntos.

A mis abuelas: Nancy y Maria Dolores por apoyarme en todo momento, por siempre querer lo mejor para mi, por darme animo a continuar y terminar la carrera, porque en distintas etapas de vida cada una de ellas me cuidaron siempre dandome lo mejor y apoyandome en cada una de mis decisiones .

A mi abuelo Gregorio por ser para mi el padre que nunca tuve, a mi abuelo Juan Francisco que no se encuentra físicamente pero estoy seguro que desde el cielo, me esta acompañando y se siente muy feliz porque yo haya llegado hasta aquí.

A mi segunda madre Surama por apoyarme siempre, por estar a mi lado en los momentos buenos y malos, te quiero mucho....

A mis amigos verdaderos:

Para agradecerle a: **Wilmer, Dalilis, Frank Alain, Leonel Eduardo, Roberto, Thalia, Hector, Alexander y Hector Daniel** necesitaria otro documento de tesis, siempre los recordaré como una de las mejores experiencias vividas en la universidad, millones de bendiciones, ustedes son los mejores del mundo mundial... mil y millones de gracias....

Amistades inolvidables:

Agradecerle a team favorito **Anayanci, Chandy , Samper y Adrian** gracias por ser mis amigos, mis mejores compañeros de grupo, los extraño muchooooo.....

Agradezco de una manera muy especial a **Erick Prade, Maite y Rubén** los quiero mucho, gracias por las risas, las tristezas compartidas les deseo lo mejor y espero que nuestra amistad siga creciendo...

Agradecer a **Dailin, Ively, Claudia María, Osvaldo, Kilmer, Manuel Alejandro, Miguel, Yoandry, Elizabet, Celida y Danelis**, por todos los buenos momentos que compartimos juntos, por los momentos de tensión en cada examen y sobre todo gracias por estar ahí siempre dispuestos a estrecharme la mano, con el objetivo que saliera adelante...

Millones de bendiciones, para todos.....

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 postoperatoria tardía más frecuente tras la cirugía de catarata asociada con disminución de la agudeza visual, deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas.

El software PANDOC tiene implementado un algoritmo para identificar y cuantificar objetivamente la Opacidad de la Cápsula Posterior mediante el uso de las imágenes resultantes del Pentacam utilizando una combinación de contornos activos y la transformada circular de Hough. En la presente investigación se propone implementar Snake como propuesta de mejora del anterior algoritmo de contornos activos implementado. Se validan los resultados de los algoritmos propuestos usando imágenes de *ground truth* brindadas por los especialistas del Instituto Cubano de Oftalmología. Como resultado sobresale el algoritmo Snake en una población de 28 imágenes de 20 pacientes donde el valor medio de la métrica "Desajuste de Bordes" fue de 0.06 y la desviación estándar de 0.06 mientras que en la métrica "Índice de Yule" tuvo una media de 1.98 y la desviación estándar de 0.02.

Palabras claves: cápsula posterior, catarata, contornos activos, snake.

Índice

| | |
|---|----|
| <i>Introducción</i> | 1 |
| CAPÍTULO 1. IDENTIFICACIÓN DE OPACIDAD CAPSULAR EN PACIENTES OPERADOS DE CATARATA: BASES CONCEPTUALES, METODOLOGÍAS, HERRAMIENTAS Y TECNOLOGÍAS USADAS DURANTE EL DESARROLLO..... | 6 |
| 1.1 <i>Opacidad de la Cápsula Posterior: conceptualización</i> | 6 |
| 1.1.1 <i>Incidencia y prevención</i> | 6 |
| 1.2 <i>Sistemas desarrollados para la identificación de OCP</i> | 7 |
| 1.2.1 <i>POCO</i> | 7 |
| 1.2.2 <i>EPCO</i> | 8 |
| 1.2.3 <i>AQUA</i> | 8 |
| 1.2.4 <i>AA</i> | 9 |
| 1.2.5 <i>Image J</i> | 10 |
| 1.2.6 <i>Análisis de los sistemas para la identificación de OCP</i> | 10 |
| 1.3 <i>Programa Analizador de Opacidad Capsular: PANDOC</i> | 11 |
| 1.4 <i>Imagen digital</i> | 11 |
| 1.4.1 <i>Imagen médica</i> | 12 |
| 1.5 <i>Segmentación de imágenes</i> | 12 |
| 1.5.1 <i>Tipos de segmentación de imágenes</i> | 13 |
| 1.5.2 <i>Técnicas de segmentación de imágenes</i> | 13 |
| 1.5.3 <i>Valoración sobre los algoritmos de segmentación</i> | 19 |
| 1.6 <i>Herramientas usadas durante el desarrollo</i> | 19 |
| 1.7 <i>Metodología de software</i> | 19 |
| 1.7.1 <i>Metodologías tradicionales</i> | 20 |
| 1.7.2 <i>Metodologías ágiles</i> | 20 |
| 1.8 <i>Lenguaje de Programación</i> | 23 |
| 1.9 <i>Entorno integrado de desarrollo</i> | 24 |
| 1.10 <i>Herramienta para el análisis estadístico utilizando SPSS</i> | 24 |
| 1.11 <i>Herramienta para el modelado</i> | 25 |
| 1.12 <i>Conclusiones del capítulo</i> | 25 |
| CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN. | 27 |
| 2.1 <i>Reducción de ruido en pre-procesamiento: filtro de la Mediana</i> | 27 |
| 2.2 <i>Snakes</i> | 27 |
| 2.3 <i>Especificación del contorno</i> | 28 |
| 2.4 <i>Función de energía</i> | 28 |

| | | |
|---|--|--------------------------------------|
| 2.5 | <i>Términos de la función de energía</i> | 29 |
| 2.6 | <i>Descripción del algoritmo Snakes</i> | 29 |
| 2.7 | <i>Pseudocódigo del algoritmo</i> | 30 |
| 2.8 | <i>Requisitos</i> | 33 |
| 2.9 | <i>Historia de Usuario</i> | 34 |
| 2.10 | <i>Plan de entrega</i> | 38 |
| 2.11 | <i>Tarjetas CRC</i> | 38 |
| 2.12 | <i>Estándares de codificación</i> | 39 |
| 2.13 | <i>Patrones de diseño utilizados</i> | 39 |
| 2.14 | <i>Conclusiones del capítulo</i> | 42 |
| CAPÍTULO 3: RESULTADOS Y VALIDACIÓN DEL SISTEMA | | 43 |
| 3.1 | ANÁLISIS ESTADÍSTICO DE LOS RESULTADOS EN IMÁGENES DEL PANDOC | 43 |
| 3.1.1 | <i>Pruebas Paramétricas</i> | 43 |
| 3.1.2 | <i>Pruebas no Paramétricas</i> | 43 |
| 3.1.2.1 | <i>Test de suma de rangos de Wilcoxon</i> | 44 |
| 3.1.2.2 | <i>Test de Friedman</i> | 45 |
| 3.2 | PRUEBAS | 45 |
| 3.2.1 | <i>Pruebas unitarias</i> | 45 |
| 3.2.2 | <i>Pruebas de Caja Negra</i> | 47 |
| 3.2.3 | <i>Análisis de los resultados en las imágenes del PENTACAM</i> | 48 |
| 3.2.4 | <i>Análisis Estadístico</i> | 52 |
| 3.3 | <i>Conclusiones parciales</i> | 53 |
| CONCLUSIONES GENERALES | | 54 |
| RECOMENDACIONES | | 55 |
| REFERENCIAS BIBLIOGRÁFICAS | | ¡ERROR! MARCADOR NO DEFINIDO. |
| ANEXOS | | 62 |
| ANEXO 1: TARJETAS CRC | | 62 |
| ANEXO 2: CASOS DE PRUEBA | | ¡ERROR! MARCADOR NO DEFINIDO. |
| GLOSARIO DE TÉRMINOS | | 63 |

Índice de ilustraciones

| | |
|---|----|
| <i>Ilustración 1. Software POCO</i> | 7 |
| <i>Ilustración 2. Software EPCO</i> | 8 |
| <i>Ilustración 3. Software AQUA</i> | 9 |
| <i>Ilustración 4. Software AA</i> | 10 |
| <i>Ilustración 5. Software Image J</i> | 10 |
| <i>Ilustración 6. Representación de una imagen digital</i> | 11 |
| <i>Ilustración 7. Matriz que representa una imagen digital</i> | 11 |
| <i>Ilustración 8. Diferentes estados de un algoritmo de crecimiento de regiones. En la imagen más a la izquierda se marca la semilla inicial y en las restantes de izquierda a derecha el sucesivo crecimiento de la selección de los píxeles involucrados en la segmentación</i> | 14 |
| <i>Ilustración 9. (a) Imagen original (b) Imagen segmentada con el algoritmo K-Means</i> | 15 |
| <i>Ilustración 10. Proceso iterativo de ajuste y segmentación del Snake sobre un objetivo en una imagen</i> | 16 |
| <i>Ilustración 11. Representación paramétrica del contorno</i> | 17 |
| <i>Ilustración 12. Operación del filtro de la Mediana</i> | 27 |
| <i>Ilustración 13. Izquierda: circunferencia representada por un polígono de 4 vértices. Derecha: circunferencia representada por un polígono de 6 vértices</i> | 28 |
| <i>Ilustración 14. Izquierda: polígono inicial generado. Derecha: polígono luego de aplicado el algoritmo de relleno de vértices</i> | 28 |
| <i>Ilustración 15. Representación de la vecindad de búsqueda de un snaxel</i> | 30 |
| <i>Ilustración 16. Pasos del Algoritmo Snake</i> | 31 |
| <i>Ilustración 17. Empleo del patrón Experto</i> | 40 |
| <i>Ilustración 18. Empleo del patrón Bajo Acoplamiento en a y b</i> | 40 |
| <i>Ilustración 19. Empleo del patrón Alta cohesión</i> | 41 |
| <i>Ilustración 20. Empleo del patrón Controlador</i> | 41 |
| <i>Ilustración 21. Empleo del patrón Creador</i> | 42 |
| <i>Ilustración 22. Método para el cálculo de distancia media</i> | 46 |
| <i>Ilustración 23. Diagrama del flujo del método "distancia_Media"</i> | 46 |
| <i>Ilustración 24. Resultados de las pruebas unitarias</i> | 47 |
| <i>Ilustración 25. Resultados de las pruebas de caja negra</i> | 48 |
| <i>Ilustración 26. Imágenes del Pentacam segmentadas por los algoritmos propuestos(b,c) y la imagen de referencia marcada por los especialistas(a)</i> | 49 |
| <i>Ilustración 27. Resultados de la métrica Desajuste de Bordes en los dos algoritmos</i> | 49 |
| <i>Ilustración 28. Resultados de la métrica Tasa de Correctos Equilibrada en los dos algoritmos</i> | 50 |
| <i>Ilustración 29. Resultados de la métrica del Índice de Sokal-Sneath en los dos algoritmos</i> | 50 |
| <i>Ilustración 30. Resultados de la métrica del Índice de Rogers-Tanimoto en los dos algoritmos</i> | 51 |

Ilustración 31. Resultados de la métrica del Índice de Yule en los dos algoritmos..... 51

Índice de tablas

| | |
|---|-----------|
| <i>Tabla 1. HU1 Cargar imagen.....</i> | <i>34</i> |
| <i>Tabla 2. HU2 Filtrar imagen</i> | <i>34</i> |
| <i>Tabla 3. HU3 Determinar la distancia media</i> | <i>35</i> |
| <i>Tabla 4. HU4 Determinar la energía del contorno</i> | <i>35</i> |
| <i>Tabla 5. HU5 Determinar la energía de la curvatura</i> | <i>35</i> |
| <i>Tabla 6. HU6 Determinar la energía de la imagen.....</i> | <i>36</i> |
| <i>Tabla 7. HU7 Determinar el color del punto.....</i> | <i>36</i> |
| <i>Tabla 8. HU8 Segmentar la imagen</i> | <i>36</i> |
| <i>Tabla 9. Plan de duración de las iteraciones.....</i> | <i>37</i> |
| <i>Tabla 10. Plan de entrega</i> | <i>38</i> |
| <i>Tabla 11. Tarjeta CRC para la clase Snakes.....</i> | <i>38</i> |
| <i>Tabla 13. Valores medios y desviaciones por cada métrica mejor resultado marcado en gris y peor gris oscuro.....</i> | <i>52</i> |
| <i>Tabla 14. Test de Friedman sobre los resultados de las variables Índice de Yule y Desajuste de Bordes</i> | <i>52</i> |
| <i>Tabla 15. Prueba por pares de Wilcoxon sobre los resultados de la variable en el Índice de Yule</i> | <i>53</i> |
| <i>Tabla 16. Prueba por pares de Wilcoxon sobre los resultados de la variable Desajuste de Bordes ..</i> | <i>53</i> |
| <i>Tabla 17. Tarjeta CRC de la clase Imagen.....</i> | <i>62</i> |
| <i>Tabla 18. Tarjeta CRC de la clase Snakes_ventana</i> | <i>62</i> |
| <i>Tabla 19. Tarjeta CRC de la clase Snaxel.....</i> | <i>62</i> |

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 no es más que la opacidad en el cristalino del ojo, trayendo como consecuencia que la luz se disperse dentro del ojo y no sea posible enfocar en la retina, de esta forma crea una serie de imágenes difusas. Las cataratas generan 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, acelerando este proceso si el paciente padece de enfermedades como la diabetes o hipertensión. Con mayor frecuencia esta enfermedad tiende a aparecer en pacientes mayores de cincuenta años de edad [1].

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 muchos de ellos el paciente puede presentar complicaciones postoperatorias. La complicación postoperatoria más frecuente a largo o mediano plazo es la Opacidad de la Cápsula Posterior (OCP). La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de catarata. Para garantizar una buena recuperación visual del paciente es imprescindible la transparencia de esta estructura. La OCP trae consigo deterioro de la sensibilidad al contraste y problemas de deslumbramiento que conllevan importantes repercusiones sociales, médicas y económicas; esta complicación postoperatoria puede aparecer meses e incluso años después de realizada la cirugía. Su incidencia en la actualidad se encuentra entre 0,7% y 47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados. Siendo en Cuba una cifra considerable que asciende hasta el 50% de los casos [2].

Disímiles investigaciones se han realizado con el objetivo de identificar los principales factores que influyen en la aparición de esta anomalía postoperatoria, un importante factor es la biocompatibilidad de la lente intraocular en relación con el saco capsular [3]. Otro factor influyente son los implantes adicionales como los anillos de tensión capsular (Capsular Tensión Ring: CTR), los cuales a pesar de su eficacia para la reducción significativa de la OCP no logran eliminarla totalmente [4]. Las aplicaciones de las nuevas tecnologías también han permitido obtener cierto grado de avance en este campo, el uso de técnicas como Cleanbagjet, la fotólisis y la pieza de mano láser las cuales se encuentran actualmente en estudio para definir su utilidad en la reducción de la OCP. También se menciona la terapia genética. 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].

En los últimos años en todo el mundo se han desarrollado disímiles sistemas como Opacificación de la capsula posterior (POCO, por sus siglas en inglés), Evaluación de la opacidad de la cápsula posterior (EPCO, por sus siglas en inglés) [5], Cuantificación automática de catarata secundaria (AQUA, por sus siglas en inglés) [6], y el sistema AA del inglés Aslam Analysis [7], dichos sistemas están basados en las imágenes en retroiluminación obtenidas por las lámparas de hendiduras [8]. Los mismos han sido desarrollados con el objetivo de servir de apoyo a los médicos a la hora de dar un correcto diagnóstico. La mayoría de estos sistemas tienen en común una desventaja muy peculiar producto de las imágenes resultantes de la lámpara de hendidura, el proceso no es completamente automático y los softwares son privativos. Han sido desarrollados varios softwares propietarios para eliminar el efecto indeseable de dichas imágenes, con la función de múltiples imágenes de un mismo ojo [9], fotografiadas en direcciones ligeramente diferentes de la mirada [10], pero ello involucra una curva de aprendizaje muy larga, ya que es un procesamiento tedioso y engorroso para el paciente.

Por otra parte, el Pentacam es un equipo oftalmológico de alta tecnología capaz de reconstruir imágenes tridimensionales de alta resolución del polo anterior del ojo. Estas imágenes son reconstruidas a partir de múltiples fotografías tomadas mediante una cámara rotacional del sistema Scheimpflug con que cuenta el equipo. El análisis de estos tomogramas en 3D es de gran utilidad para la cuantificación objetiva de la OCP [11]. Para la identificación automática de la OCP en las imágenes del Pentacam es necesario la implementación de algoritmos de segmentación.

Los algoritmos de segmentación permiten identificar y delinear con precisión los bordes de las estructuras anatómicas relevantes en las imágenes médicas [12]. Estos algoritmos son útiles para asistir y automatizar el tratamiento de las imágenes y les facilitan a los especialistas una visión diferente, con datos específicos sobre las imágenes de los pacientes. Los métodos para segmentar varían ampliamente y actualmente no existe un algoritmo que alcance resultados aceptables para todo tipo de imágenes médicas [13]. De cualquier forma, los métodos que se especializan en aplicaciones concretas pueden dar buenos resultados en el campo en que se desarrollan, por lo que se hace necesaria la correcta selección del método en función de lo que se pretende obtener con este. Uno de los algoritmos que se utiliza para la segmentación y que permite al especialista la selección en la imagen médica de las estructuras de interés, es aquel que se basa en modelos de contornos activos y que se conoce como Snakes, de ahí la importancia de su utilización.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el grupo de Investigación AIRI (Artificial Intelligence: Research and Innovation), el cual tiene como principal resultado el software PANDOC (Programa Analizador de Opacidad Capsular), el mismo fue realizado en colaboración con el Instituto Cubano de Oftalmología "Ramón Pando Ferrer". El software PANDOC provee al oftalmólogo de una herramienta por medio de la cual este es capaz de cuantificar numéricamente y detectar diferencias de opacidad (a veces imperceptibles para el ojo humano), logrando así una evaluación objetiva del grado de opacidad, minimizando el sesgo de observación entre un médico y otro. En el

grupo de investigación AIRI se han desarrollado diversos algoritmos para la identificación de OCP en imágenes resultantes del Pentacam por PANDOC. En el 2015 se realizó una propuesta fundamentada por un sistema basado en casos (SBC) [14] [15]. En esta propuesta cada caso a identificar era un píxel, donde se analizan sus características de color y las de sus vecinos, así como su cercanía con respecto al centro de la estructura ocular [14] [15]. En el 2016 se realizó una variante que también utilizaba un SBC, pero en esta ocasión con el objetivo principal de minimizar tiempo de ejecución con respecto al anterior. En esta versión se usan como casos, regiones que fueron previamente extraídas usando un algoritmo de componentes conexas [16], 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 [17].

En el propio 2016 se propone otra variante utilizando mayormente técnicas de procesamiento digital de imágenes, en esta ocasión buscando los bordes de la Capsulorrexis [18], bajo la premisa de que toda variación de la imagen de fondo en la región comprendida por la Capsulorrexis es OCP. Esta solución estaba basada en una combinación de filtros espaciales; la transformada circular de Hough y un algoritmo de contornos activos [18] [19]. Este algoritmo de contornos activos propuesto, solo ubicaba el valor de la capsulorrexis cuando el valor de esta estaba en un rango de 0 a 20 píxel con respecto al círculo de la transformada circular de Hough. Esto en algunas imágenes puede provocar errores en la adecuada identificación de la capsulorrexis. En la presente investigación se propone utilizar un algoritmo de contornos activos, diferente al ya utilizado para la detección de la capsulorrexis.

Después de analizar la situación anteriormente planteada se define como **problema a resolver**: ¿cómo mejorar los resultados en la identificación la capsulorrexis en imágenes provenientes del Pentacam? Se define como **objeto de estudio**: segmentación de imágenes digitales. Para darle solución al problema formulado se plantea como **objetivo general**: desarrollar un algoritmo de contornos activos para la identificación de la capsulorrexis en imágenes del Pentacam. Enmarcado en el **campo de acción**: algoritmos de contornos activos. Se toma como **idea a defender**: con el desarrollo de un algoritmo de contornos activos para la identificación automática de la capsulorrexis en imágenes del Pentacam se mejoran los resultados en la detección de la Opacidad de la Cápsula Posterior en el software PANDOC. Para darle cumplimiento al objetivo general se desglosan los siguientes **objetivos específicos**:

- Elaborar el marco teórico referencial relacionado con la identificación de regiones en imágenes médicas.
- Desarrollar un algoritmo que permita identificar la capsulorrexis en imágenes provenientes del Pentacam.
- Evaluar el algoritmo propuesto utilizando imágenes de referencia marcadas por el especialista, mediante métricas de evaluación bi-variadas, y probar el desempeño del software desarrollado.

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

1. Identificación de las diferentes técnicas de procesamiento de imágenes médicas.
2. Revisión de las técnicas de realce, filtrado y mejora en imágenes médicas
3. Estudio del uso de los métodos de identificación de contornos en la segmentación de imágenes médicas.
4. Estudio del uso del algoritmo Snake para la segmentación de contornos.
5. Descripción de los pasos a seguir para el uso del algoritmo de segmentación propuesto.
6. Diseñar la solución propuesta en función de los requisitos especificados.
7. Implementación del algoritmo diseñado para la identificación de la capsulorrexis en imágenes del Pentacam.
8. Validación de los resultados mediante pruebas estadísticas no paramétricas y variables de calidad de la segmentación.
9. Validación de la implementación del sistema a partir de la aplicación de pruebas unitarias y funcionales.

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

Analítico-Síntesis: se utiliza con el objetivo de analizar las teorías, documentos e información referente al tema en cuestión, para luego extraer los elementos más importantes que se relacionan con la identificación de la OCP en pacientes operados de catarata.

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

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

La presente investigación está estructurada en tres capítulos; a continuación, se describe el objetivo principal de cada uno de ellos:

Capítulo 1: Identificación de opacidad capsular en pacientes operados de catarata: bases conceptuales, metodologías, herramientas y tecnologías, usadas durante el desarrollo.

En este capítulo se realiza la descripción y análisis desde el punto de vista teórico del problema general en que se enmarca la investigación, también se realiza un estudio del arte de los sistemas basados en casos y se describen las herramientas y tecnologías a utilizar durante el desarrollo de la solución. Son expuestos los argumentos que conllevan a la selección de dichas herramientas y tecnologías. Se explican las principales características de las metodologías de software y las ventajas que ofrece la escogida en la presente investigación.

Capítulo 2: Descripción de la propuesta de solución.

En este capítulo se describe la propuesta de solución, se generan los artefactos de la metodología definida anteriormente, se establecen los requerimientos funcionales y no funcionales y los estándares de codificación, ayudando todo esto a darle solución al problema planteado, a través de la implementación del algoritmo Snake para la detección de la capsulorrexis.

Capítulo 3: Resultados y validación del sistema.

En este capítulo se exponen los aspectos relacionados con el desarrollo de la solución y se presentan las pruebas realizadas al algoritmo y al software.

Capítulo 1. Identificación de opacidad capsular en pacientes operados de catarata: bases conceptuales, metodologías, herramientas y tecnologías usadas durante el desarrollo.

En el presente 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 el de incidencia y prevención de la OCP. También se realiza un estudio del estado del arte de los principales sistemas para la identificación de OCP que existen en el mundo, se estudian los sistemas basados en contornos activos, los cuales ayudarán en la identificación de la opacidad capsular en pacientes sometidos a cirugía de cataratas. Se describen las metodologías, herramientas y tecnologías usadas durante el desarrollo.

1.1 Opacidad de la Cápsula Posterior: conceptualización.

La cápsula posterior es una estructura que anatómicamente queda por detrás de la lente intraocular implantada en el paciente operado de cataratas, para garantizar una buena recuperación visual del mismo es imprescindible la transparencia de esta estructura. Desafortunadamente no son pocos los casos en los que se desarrolla esta anomalía, la opacificación de la cápsula posterior repercute negativamente en el resultado visual del paciente [20]. En el campo de la oftalmología esto es conocido como OCP y constituye la complicación tardía más importante en la cirugía de cataratas en los tiempos modernos [12].

1.1.1 Incidencia y prevención.

La incidencia de la OCP en la actualidad se encuentra entre 0,7 y 47,6 % en los primeros cinco años de la cirugía, según un análisis de 90 estudios publicados. Que el rango de variación reportado sea tan amplio probablemente responda a diferentes criterios de selección de la muestra, edad, técnica quirúrgica, tipo de lente intraocular (LIO) y sistemas de evaluación de la opacidad [2].

Debido a esto, las investigaciones recientes han dirigido su atención a estrategias de prevención de la OCP. Estas incluyen el diseño de los lentes intraoculares, técnicas quirúrgicas modificadas y/o mejoradas, implantes adicionales como los anillos de tensión capsular, el empleo de fármacos y agentes citotóxicos [2].

El conocimiento de los mecanismos etiopatogénicos involucrados en el desarrollo de la opacidad de la cápsula posterior tras la cirugía de catarata ha permitido concluir que para prevenir la aparición de esta complicación se necesita: bloquear la migración de las células epiteliales cristalinas residuales, impedir la proliferación y/o transformación fibroblástica o eliminarlas completamente del saco capsular [13].

1.2 Sistemas desarrollados para la identificación de OCP.

Los sistemas más usados en la actualidad por el oftalmólogo que le permite visualizar la OCP son los sistemas basados en la lámpara de hendidura, también conocida como biomicroscopio. Es un instrumento de visualización directa de la estructura ocular, pero su limitación radica en que la evaluación del grado de opacidad de la cápsula posterior la hace el médico de manera subjetiva según su apreciación visual, lo que puede variar según la experiencia del cirujano [21]. Es por esta razón que algunos autores han desarrollado soluciones informáticas para lograr una evaluación objetiva y reproducible de esta complicación. A continuación, se describen algunos de estos sistemas, así como la influencia de la calidad de las imágenes en la obtención de resultados investigativos de elevada calidad.

1.2.1 POCO

Opacificación de la Capsula Posterior (POCO, por sus siglas en inglés): en este sistema las imágenes obtenidas son por retroiluminación, las mismas son evaluadas usando un análisis de los píxeles basado en la diferencia de texturas. Esta evaluación es semiobjetiva, ya que el examinador debe elegir el área afectada y clasificarla según su severidad: el programa divide la zona dentro de la capsulorrexia en 56 pequeños segmentos de igual área, el examinador debe marcar los segmentos que estén opacificados en más del 50% de su área y luego asignar a éstos un nivel de severidad (leve, moderado o severo). El resultado viene dado en forma de porcentaje de opacificación de 0 a 100% [5]. Se calcula la severidad de la opacificación mediante la fórmula:

$$ocp = \frac{[A1 + (A2*2) + (A3*3)]}{At} \quad (1)$$

Donde $A1$ es el área marcada como grado 1, $A2$ la marcada como grado 2, $A3$ la marcada como grado 3 y At el área total definida.

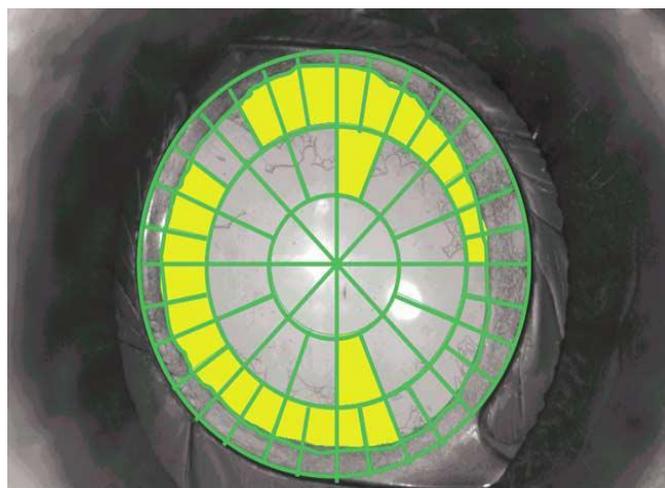


Ilustración 1. Software POCO

1.2.2 EPCO

Evaluación de la opacidad de la cápsula posterior (EPCO, por sus siglas en inglés) desarrollado por Tetz y Col, en 1997. Se obtienen imágenes de la lente por retroiluminación, se transfieren al programa, se marca el área a estudiar (algunos autores analizan el área tras la óptica, otros el área dentro de la capsulorrexis anterior y también es posible examinar la zona central comprendida por 3-4 mm), luego se remarcan las zonas opacificadas y se clasifican subjetivamente en 4 grupos: mínimo, leve, moderado y severo. El índice de OCP viene dado al multiplicar el grado de opacificación por el área seleccionada. El resultado obtenido es un índice de 0 a 4 [5]. Calcula la severidad de la opacificación mediante la fórmula

$$ocp = \frac{[Amin + (Alev*2) + (Amod*3) + (Asev*4)]}{At} \quad (2)$$

Donde *Amin* es el área marcada con afectación mínima, *Alev* la marcada como leve, *Amod* la marcada como moderada, *Asev* la marcada como severa y *At* el área total definida.

En su versión más reciente (EPCO 2000) fue añadida una nueva funcionalidad, permitiendo mezclar 2 imágenes para intentar eliminar la influencia de la reflexión de la luz como se muestra en la Ilustración 2. A pesar de sus funcionalidades este software no es totalmente automático pues el especialista debe aún marcar los distintos niveles de opacidad en las áreas señaladas, siendo esta su principal deficiencia [5].

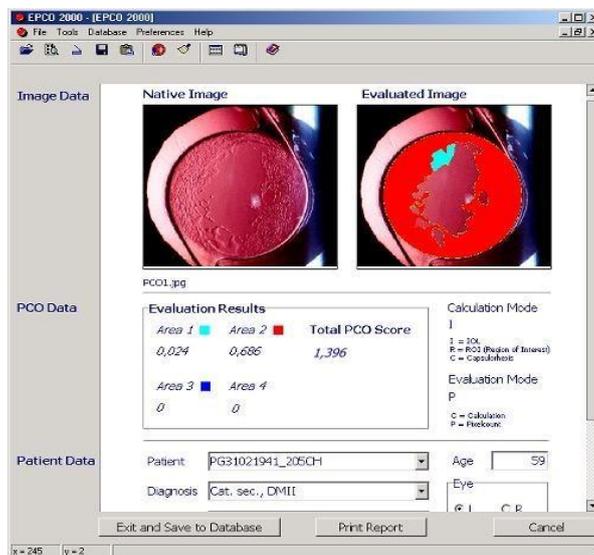


Ilustración 2. Software EPCO

1.2.3 AQUA

Cuantificación automática de catarata secundaria (AQUA, por sus siglas en inglés): desarrollado por la Universidad de Viena en cooperación con la Universidad Autónoma de Graz, Austria [6]. Al obtener imágenes en retroiluminación, estas se importan al programa, se selecciona y analiza el área dentro de la capsulorrexis de manera automática (el resultado visual se puede ver en la Ilustración 3).

Este programa está basado en texturas, usando matriz de co-ocurrencia para el análisis por píxel obteniendo primero el borde de la capsulorrexis [6]. Es totalmente automático y el resultado que se obtiene es un índice de 0 a 10 [6]. El Dr. Findl [6] enfatiza en que los resultados de AQUA dependen de obtener imágenes de alta calidad en parámetros como brillo, foco y afectación de la luz entre otros. Este software no presenta ninguna técnica para la eliminación de la luz de fondo, aunque segmenta esa región para no tenerla en cuenta en su resultado (si en esa región afectada por la luz existía opacidad no se tiene en cuenta).

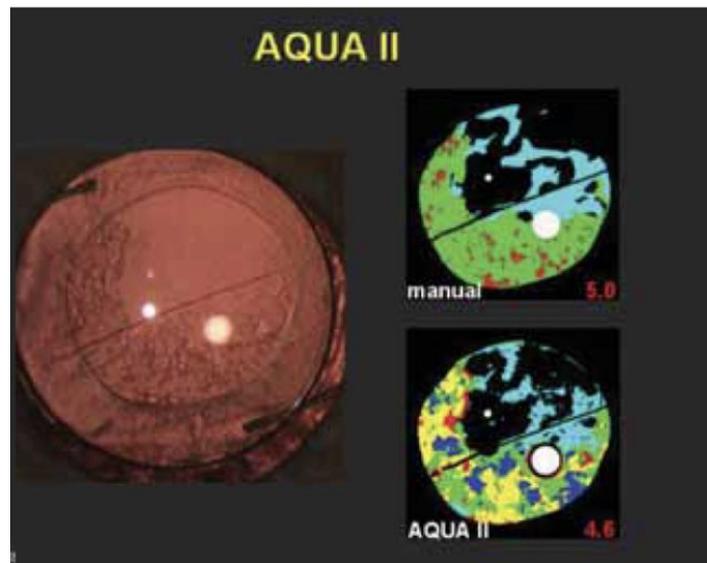


Ilustración 3. Software AQUA

1.2.4 AA

Este sistema fue diseñado y programado utilizando la plataforma de programación MatLab. Todo el diseño de la programación y la escritura de código fueron por TM Aslam y para mayor claridad el sistema se conoce por lo tanto como el análisis Aslam (AA). El primer problema encontrado fue uno de iluminación desigual en las imágenes. Incluso con las grandes áreas de reflejos de luz aberrantes retirados, utilizando la fusión con imágenes similares, pero vírgenes, una iluminación de fondo generalmente variables de toda la imagen puede causar errores en el análisis de imágenes. Aunque un sistema para la eliminación de registro basado de imágenes de luz está siendo probado y desarrollado por TM Aslam, todavía se necesita una curva de aprendizaje larga y sigue influyendo el efecto negativo de la iluminación de fondo [22]. La ilustración (4) muestra una imagen del software:

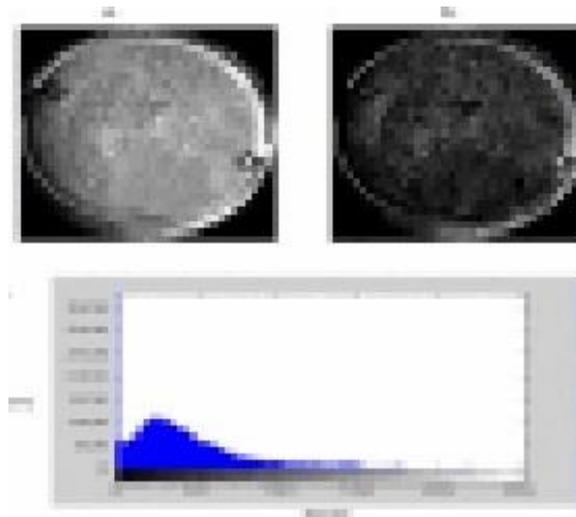


Ilustración 4. Software AA

1.2.5 Image J

Desarrollado por el National Institute of Health de Estados Unidos. Está basado en los cálculos realizados por Grewal y Jain, quienes determinaron que para los tomogramas del Pentacam un área de 4 mm corresponde a 16 752 píxeles en la imagen. Se utilizan las herramientas Find Edges (hallar bordes) y Measure (medición) para calcular el valor promedio de la intensidad de píxeles del área seleccionada en los tomogramas Scheimpflug, como indicador de la opacidad capsular posterior, expresado en unidades (U). Esta herramienta es creada con otros fines, lo que hace que la curva de aprendizaje sea larga, además de que no diferencia los píxeles de fondo a la hora de dar el resultado [23]. La ilustración (5) muestra una imagen del software:

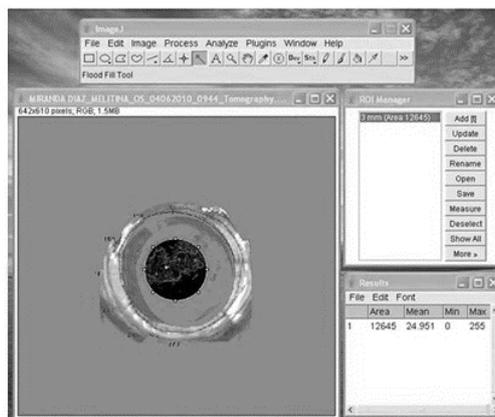


Ilustración 5. Software Image J

1.2.6 Análisis de los sistemas para la identificación de OCP

Los sistemas anteriormente analizados, a pesar de los buenos resultados que obtienen en la detección de la opacidad, presentan una desventaja en común y es que el proceso no es completamente automático y los softwares son privativos, además ninguno de ellos trabaja con las imágenes

provenientes del Pentacam. Por lo anterior planteado es que no se puede utilizar ninguno de estos sistemas para darle solución a la problemática planteada.

1.3 Programa Analizador de Opacidad Capsular: PANDOC

El software PANDOC es una herramienta para la cuantificación y análisis de la OCP. Es un software intuitivo con una curva de aprendizaje pequeña, el cual enfoca sus resultados en pos del análisis estadístico que aporten información de alto valor para el estudio y disminución de esta anomalía postoperatoria. Es un software de nuevo tipo, ya que a diferencia del resto toma las imágenes de los tomogramas de Scheimpflug del Pentacam y brinda datos antes no vistos por la comunidad oftalmológica mundial como la correlación opacidad/transparencia en el centro y la periferia, el mismo comprende funcionalidades automáticas y semiautomáticas orientadas solo al uso de imágenes de tomogramas del Pentacam, lo cual hace más intuitivo la interfaz actual y la focaliza en su propósito principal [20].

1.4 Imagen digital

Una imagen se puede definir matemáticamente como una función bidimensional que representa la intensidad o el nivel de gris del píxel con coordenadas espaciales dentro de la imagen [24], como se puede apreciar en la Ilustración 7.

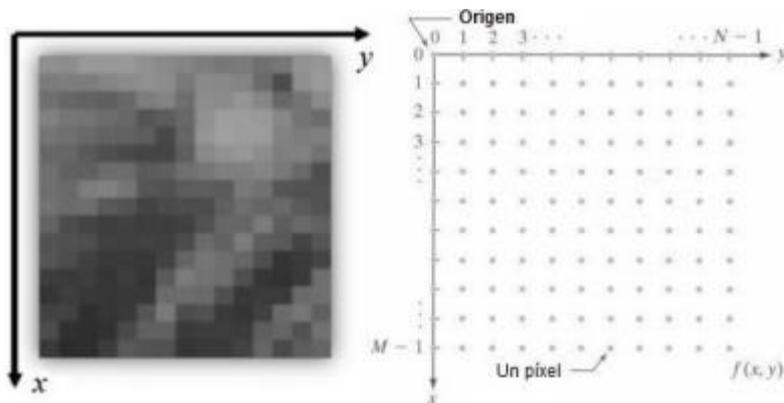


Ilustración 6. Representación de una imagen digital

Cuando los valores son cantidades finitas y discretas, la imagen se denomina imagen digital [24]. Una imagen digital se compone de un número finito de elementos con un lugar y valor específicos. Estos elementos se conocen como píxeles [24]. Una imagen digital puede ser convenientemente representada por una matriz de tamaño de la forma:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Ilustración 7. Matriz que representa una imagen digital

Los elementos de la matriz (píxeles) en una imagen monocromática (niveles de gris) típica son del orden de 2^8 o 256 niveles de gris de tal manera que $0 \leq f(x,y) \leq 255$, donde 0 representa el negro y 255

el blanco [25]. Por esta razón, los píxeles pueden representarse como caracteres en la mayoría de los lenguajes de programación. Las columnas y las filas de las matrices de imágenes digitales tienen un rango de $0 \leq i \leq N-1$, $0 \leq j \leq M-1$ [24].

1.4.1 Imagen médica

Se le llama imagen médica al resultado de aplicar un conjunto de técnicas y procesos con el fin de crear imágenes del cuerpo humano, o parte de él, aprovechando las características físicas y químicas de este con propósitos clínicos (procedimiento médico que busca revelar, diagnosticar o examinar enfermedades) o para la ciencia médica [26]. Entre las modalidades de adquisición de imágenes médicas se destacan las de Rayos X y la Tomografía Axial Computarizada (TAC). Esta última, por su facilidad de realización, la precisión diagnóstica y la ausencia de riesgo, se ha convertido en la primera alternativa de diagnóstico que se realiza a pacientes con patología neurológica. Otra modalidad que se destaca es la Resonancia Magnética (RM) y se basa en la capacidad de algunos núcleos para absorber ondas de radiofrecuencia cuando son sometidos al efecto de un campo magnético, el ultrasonido y otros. Todas estas modalidades aportan imágenes médicas digitales [26].

1.5 Segmentación de imágenes

La segmentación es la tarea de descomponer los datos de la imagen en estructuras significativas que son relevantes para tareas específicas [27]. Técnicamente la segmentación se encarga de darle índices a cada elemento de la imagen para representar su membresía a una estructura en particular. Existen distintos niveles de automatización dependiendo del método que se utilice para segmentar. Este automatismo pone a prueba los algoritmos utilizados por los sistemas para tal fin, ya que las estructuras obtenidas deben corresponderse lo más acertadamente posible con el contenido semántico de la imagen [28].

El reconocimiento de los elementos de las imágenes es una tarea de alta complejidad, que usualmente muestra los mejores resultados al ser realizada por los humanos. Sin embargo, la delineación de los mismos es una tarea que requiere una precisión esencial, mejor obtenida con la utilización de medios computacionales. Algoritmos cuidadosamente diseñados pueden guiar a excelentes soluciones para el problema de la delineación. El nivel al que se realiza esta subdivisión depende de la aplicación en particular. En la mayor parte de los casos, una buena segmentación dará lugar a buenos resultados de la aplicación que se esté desarrollando, por lo que se debe poner todo el esfuerzo posible en esta etapa [27].

En las aplicaciones médicas, por lo general es suficiente representar únicamente las estructuras relevantes o estructuras objetivo y no todas las presentes, lo cual influye en la elección de los métodos a aplicar para una aplicación en particular [27].

1.5.1. Tipos de segmentación de imágenes

Para realizar las tareas de segmentación existen muchos algoritmos que aportan diversos resultados, en dependencia de las imágenes con las que trabajen y su funcionamiento particular. Estos métodos se pueden clasificar en dependencia de su nivel de automatización en: métodos manuales, semiautomáticos y automáticos. Cada uno aporta una serie de ventajas y desventajas que actúan a la hora de tomar decisiones sobre ellos.

1.5.1.1. Segmentación manual

La manera más general y fácil de lograr la segmentación es manualmente dibujando las caras y elementos importantes de las estructuras presentes en las imágenes médicas. En este caso, el usuario delinea con alguna herramienta las estructuras relevantes, lo que le da la ventaja de redibujar cualquier porción y corregir algún error introducido [27].

Este método es robusto (siempre aplicable) pero consume mucho tiempo, cuando se trabaja con cúmulos de imágenes muy grandes se vuelve impracticable y no es preciso debido a que el usuario generalmente se desvía del contorno deseado y si los objetos son difíciles de delinear, pueden ser difíciles de segmentar. Un ejemplo donde se aprecia la segmentación manual es la segmentación de tumores [27].

1.5.1.2. Segmentación automática

La segmentación automática, como su nombre lo indica es el tipo de segmentación donde el ordenador realiza todo el proceso de manera automática sobre la imagen objetivo. Es improbable que los métodos de segmentación automática reemplacen alguna vez a los métodos manuales pero si es probable que se conviertan en elementos cruciales para el análisis de imágenes médicas [29].

1.5.1.3. Segmentación semiautomática

En este tipo de modalidad el ordenador realiza el proceso, pero el usuario interviene en determinados momentos sobre el mismo, como por ejemplo para definir las regiones de interés mediante dispositivos como el ratón y otros parámetros o para corregir resultados. Luego, los algoritmos se aplicarán de forma que se elige el camino que mejor se ajusta al borde de la imagen [28].

1.5.2 Técnicas de segmentación de imágenes

Las técnicas de segmentación de imágenes son muy variadas y muchas se basan en distintos principios e ideas, por lo que obtienen diversos tipos de resultados. Por esta razón, no existe una técnica que sea buena para todo tipo de imagen y es necesaria la selección del método que mejor se ajuste a las necesidades.

1.5.2.1 Crecimiento de regiones

Las técnicas de segmentación basadas en regiones tienen su base en las reglas de similitud y en la conectividad de los píxeles, por lo que son similares a las técnicas de umbralización. Las regiones se forman mediante píxeles que tengan conectividad y presenten alguna propiedad de similitud respecto al resto de los píxeles que no pertenecen a la región. Este método es muy usado en la práctica debido a su simplicidad de implementación [27].

El proceso de crecimiento se inicia a partir de la selección por el usuario de puntos “semillas” a partir de los cuales se comienza la adición de puntos vecinos cuyas intensidades se encuentran dentro del umbral determinado. La elección de los puntos semilla y el umbral necesario para una buena segmentación no son fáciles de determinar y es probablemente, el punto más débil del método, pues la solución final depende directamente de estos valores. En la siguiente figura se puede apreciar un seguimiento del algoritmo [27].

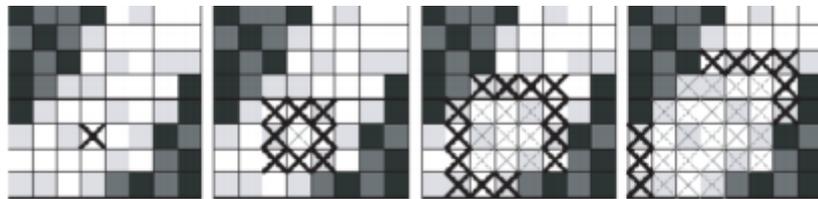


Ilustración 8. Diferentes estados de un algoritmo de crecimiento de regiones. En la imagen más a la izquierda se marca la semilla inicial y en las restantes de izquierda a derecha el sucesivo crecimiento de la selección de los píxeles involucrados en la segmentación

1.5.2.2 Agrupamiento

Los algoritmos de agrupamiento (*Clustering*) llevan a cabo esencialmente la misma función que los métodos clasificadores tratados con mayor profundidad en [29], pero sin utilizar datos de entrenamiento.

Por lo tanto, son métodos no supervisados. Para compensar la falta de los datos de entrenamiento, los métodos de agrupamiento iteran entre segmentar la imagen y caracterizar las propiedades de cada clase. En este sentido, los métodos de agrupamiento se entrenan a sí mismos usando los datos disponibles. Un algoritmo de agrupamiento común es el algoritmo *K-Means* o algoritmo ISODATA [29].

Este algoritmo agrupa datos calculando iterativamente la media de la intensidad para cada clase y segmentando la imagen mediante la clasificación de cada píxel en la clase con la media más cercana. La Ilustración 10 muestra el resultado de aplicar el algoritmo *K-Means* a un corte de la Resonancia Magnética (RM) del cerebro. Se asumió que el número de clases era tres, representando (de gris oscuro a blanco) el fluido cerebroespinal, materia gris y materia blanca [29].

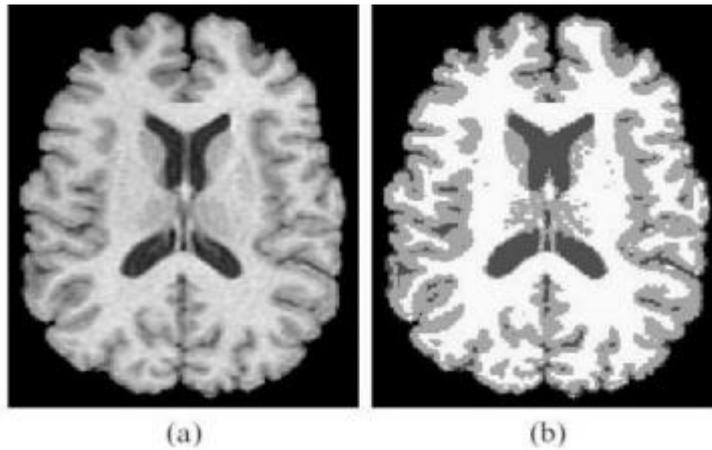


Ilustración 9. (a) Imagen original (b) Imagen segmentada con el algoritmo K-Means

Aunque los algoritmos de agrupamiento no requieren que los datos se entrenen, si requieren una segmentación inicial (o de manera equivalente, requiere parámetros iniciales). Como los métodos de clasificación, los algoritmos de agrupamiento no incorporan directamente un modelo espacial. De cualquier forma, esta falta de modelado espacial puede proveer ventajas significativas para realizar los cálculos velozmente. Es posible incorporar robustez al ruido usando campos aleatorios de Markov [29].

1.5.2.3 Watershed

La transformación *Watershed* es un método de segmentación basado en regiones que divide todo el dominio de la imagen en conjuntos conexos. Se aplica a imágenes 2D y 3D. Esta técnica encierra varios de los conceptos de las otras tres técnicas antes mencionadas y produce a menudo un resultado más estable de la segmentación, incluyendo bordes de segmentación continuos [30].

La segmentación *Watershed* está basada en la idea de tomar la imagen como un mapa topográfico con valles y cuencas. Los valores de las elevaciones del terreno están definidos por los niveles de gris de sus respectivos píxeles o la magnitud de sus gradientes [27].

Debido al gran número de mínimos locales que presenta una imagen digital ruidosa, cada uno de ellos asociado a una cuenca *Watershed*, este operador morfológico produce una gran sobre segmentación en pequeñas regiones cuando se aplica a una imagen sin pre-procesar. En una RM típica los mínimos locales pueden ser del orden de decenas de miles [30].

1.5.2.4. Modelos de contornos activos o Snakes

Dentro del grupo de estrategias basadas en modelos deformables se encuentran los modelos de contornos activos o *Snakes*. Estos se basan en la definición de un contorno inicial que algorítmicamente se deforma aproximándose hacia los bordes de la imagen, enmarcando los objetos asumiendo que estos son suaves [27].

El proceso de ajuste del contorno inicial es guiado por la minimización de una función de energía, en la que se calculan diferentes fuerzas, conocidas como: fuerzas internas y fuerzas externas [27]. Las

fuerzas internas se calculan en el interior de la curva o superficie para mantenerla suave a lo largo de la deformación. Las fuerzas externas son frecuentemente derivadas de la imagen para llevar la curva o superficie hacia la característica de interés deseada [27].

El *Snake* está compuesto por un conjunto de puntos denominados *snaxels* que mediante interpolación forman un contorno. El contorno debe ser inicializado cerca del objeto a identificar de manera que lo envuelva, esto le permitirá ajustarse a cualquier contorno cerrado que represente el objeto que se desee reconocer [31].

Para lograr este objetivo, se asocia al *Snake* una función de energía que representa cada característica trascendente del contorno y el objeto. El funcionamiento del *Snake* se basa en tres ideas fundamentales:

- La curva que envuelve al objeto debe ser continua.
- La curva que envuelve al objeto no puede retorcerse ni doblarse, debe ser suave.
- La expresión “atracción al contorno” matemáticamente se traduce por la dirección y sentido en que se acerca cada *snaxel*, al contorno del objeto a identificar [32].

En la siguiente imagen se puede apreciar un recorrido y resultado de la aplicación del algoritmo Snakes sobre una imagen.

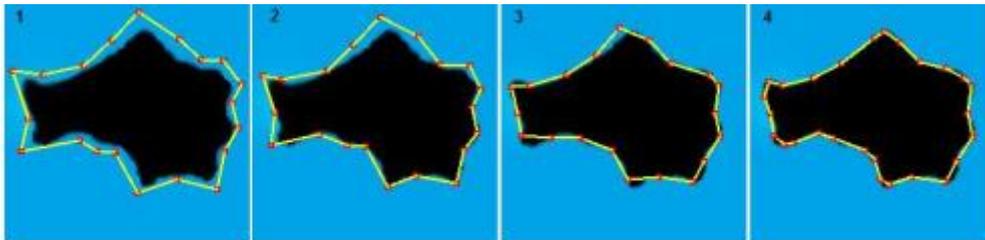


Ilustración 10. Proceso iterativo de ajuste y segmentación del Snake sobre un objetivo en una imagen

Parametrización del contorno

El contorno de segmentación del *Snake* se introduce manualmente por el usuario que desea realizar la segmentación, preferiblemente lo más cerca posible de la estructura objetivo en la imagen. Esto representa una importante ventaja de este algoritmo, puesto que permite la decisión sobre la zona que se desea segmentar [31].

El contorno se puede representar paramétricamente como $c(s)=(x(s), y(s))$ donde $x(s)$ y $y(s)$ son las coordenadas a lo largo del contorno [31]. En la Ilustración 12 se observa una representación paramétrica del contorno de segmentación.

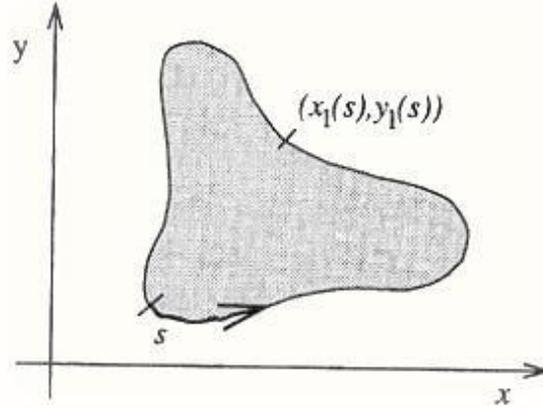


Ilustración 11. Representación paramétrica del contorno

Función de energía

El movimiento del *Snake* está guiado por la función de energía, que selecciona las nuevas posiciones de los *snaxels* como resultado de una minimización de esta. La función de energía es una suma de términos representados en la siguiente ecuación:

$$E(s) = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{imag}) \quad (3)$$

Donde los parámetros α , β y γ son los encargados de controlar la influencia relativa de los términos correspondientes a los que multiplican. Estos términos pueden variar a través de $c(s)$ [31] tomando valores generalmente entre 0 y 1, lo cual puede restringir o liberar los movimientos de cada *snaxel*.

Los términos E_{cont} , E_{curv} y E_{imag} son los que definen los movimientos del *Snake* y cada uno cumple una función específica. E_{cont} y E_{curv} se conocen como términos de energía interna y E_{imag} como término de energía externa [31].

Energía del contorno

La energía del contorno (E_{cont}) fuerza a los *snaxels* a contraerse en busca del contorno objetivo y a moverse de manera uniforme. Otra función de la energía del contorno es garantizar la estabilidad de la forma del *Snake*, esto es posible porque mantiene una equidistancia entre los *snaxels* durante el proceso de contracción. Esta energía se calcula y se trata a nivel local en cada vértice del *Snake*. El valor de esta energía viene dado por el cuadrado de la norma de la primera derivada del contorno, cuya ecuación es la siguiente [31]:

$$E_{cont} = \left\| \frac{dc}{ds} \right\|^2 \quad (4)$$

Para el contorno discreto, el contorno es aproximado por N vértices $\rho_1, \rho_2, \dots, \rho_n$ y la primera derivada se puede aproximar por la diferencia finita:

$$E_{cont} = \|\rho_i - \rho_{i-1}\|^2 \quad (5)$$

Este término trata de minimizar la distancia entre los vértices y causa el efecto de contracción del contorno. Una mejor formulación para E_{cont} es:

$$E_{cont} = (d - \|\rho_i - \rho_{i-1}\|)^2 \quad (6)$$

Donde:

d : es la distancia media entre los vértices del *Snake*, logrando que los vértices mantengan una distancia aproximadamente igual entre ellos para todo el contorno [31].

ρ_i : Representa la posición del *snaxel* en análisis.

ρ_{i-1} : Representa la posición del *snaxel* anterior al que se encuentra en análisis.

Energía de la curvatura

La energía de la curvatura (E_{curv}) le brinda flexibilidad al *Snake*, evita que se curve en exceso haciendo que se mantenga su forma mientras encuentra las zonas curvas de la figura a detectar. Dependiendo del contorno al que se quiera adaptar habrá vértices del *Snake* que relajen su curvatura permitiendo que el contorno se adapte a zonas convexas o salientes pronunciados de la Ilustración 12. Esta energía, al igual que la anterior, se calcula y se trata a nivel local en cada vértice del *Snake*, minimizando la segunda derivada del contorno y expresada como se muestra a continuación [31]:

$$E_{curv} = \left\| \frac{d^2 c}{ds^2} \right\|^2 \quad (7)$$

En el contorno discreto, la curvatura se puede aproximar por la siguiente diferencia finita:

$$E_{curv} = \|\rho_{i-1} - 2\rho_i + \rho_{i+1}\|^2 \quad (8)$$

Donde:

ρ_i : Representa la posición del *snaxel* en análisis.

ρ_{i-1} : Representa la posición del *snaxel* anterior al que se encuentra en análisis.

ρ_{i+1} : Representa la posición del *snaxel* posterior al que se encuentra en análisis.

Energía de la imagen

La energía de la imagen (E_{imag}) también conocida como término de energía externa, es la encargada de atraer el *Snake* hacia los puntos de mayor contraste de la imagen con la que se está trabajando [32]. Esta energía se representa por el gradiente de los píxeles de la imagen con signo negativo, logrando que en los puntos de mayor contraste la energía de la imagen sea menor, para poder buscar estos puntos mientras se minimiza la función de energía [31]. La ecuación de la energía de la imagen es la siguiente:

$$E_{imag} = -\|\nabla I\| \quad (9)$$

Donde ∇I es el gradiente de intensidad calculado en cada punto del *Snake*.

Matemáticamente, el gradiente de una función de dos variables (para este caso la función de intensidad de la imagen) para cada punto es un vector bidimensional cuyos componentes están dados por las primeras derivadas de las direcciones verticales y horizontales. Para cada punto de la imagen, el gradiente del vector apunta en la dirección del incremento máximo posible de intensidad y la magnitud gradiente del vector corresponde al cambio de intensidad en esa dirección. Lo anterior implica que el resultado de aplicar algún operador sobre la región de una imagen con intensidad de brillo constante es un vector cero y el resultado de aplicarlo en un punto sobre un borde es un vector que apunta cruzando el borde (perpendicular) en sentido de los puntos más oscuros hacia los más claros [33].

1.5.3 Valoración sobre los algoritmos de segmentación

El marco teórico sobre los algoritmos de segmentación analizados demuestra que los algoritmos de Agrupamiento, *Watershed* y Crecimiento de Regiones brindan buenos resultados para segmentar estructuras anatómicas. Sin embargo, estos métodos no le permiten al usuario seleccionar la región de interés o la estructura objetivo que desea segmentar. Por estas razones se establece el método de *Snakes* como solución en la presente investigación debido a las ventajas que ofrece al brindar al usuario la posibilidad de intervenir en el proceso de segmentación.

1.6 Herramientas, metodología y tecnologías usadas durante el desarrollo

En este epígrafe se describen las herramientas y tecnologías a utilizar durante el desarrollo de la solución. Son expuestos los argumentos que conllevan a la selección de dichas herramientas y tecnologías. Se explican las principales características de las metodologías de software y las ventajas que ofrece la escogida en la presente investigación; se argumenta acerca de las características y beneficios que ofrece el lenguaje de programación seleccionado, el entorno integrado de desarrollo y las herramientas que el mismo brinda.

1.6.1 Metodología de software

Algunos autores [34, 35] definen una metodología como una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo. En las dos últimas décadas, respecto a estas metodologías se ha establecido un intenso debate entre dos grandes corrientes. Las denominadas metodologías tradicionales, centradas en el control del proceso, con un riguroso seguimiento de las actividades involucradas en ellas y las metodologías ágiles, centradas en el factor humano, en la colaboración y participación del cliente en el proceso de desarrollo [34, 35].

1.6.2 Metodologías tradicionales

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo de software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total del trabajo a realizar y una vez que esté todo detallado, comienza el ciclo de desarrollo del producto. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada [36].

Rational Unified Process (RUP)

RUP constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. No es un sistema cerrado, más bien es un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Su ciclo de vida es una implementación del Desarrollo en espiral¹. Es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Constituye una forma disciplinada de asignar tareas y responsabilidades, que pretende implementar las mejores prácticas en ingeniería de software, se basa en la administración de requisitos, control de cambios, modelado visual de software y verificación de la calidad de este [37].

Divide el proceso de desarrollo en cuatro fases [38]:

- Fase de inicio: abarca la comunicación con el cliente, las actividades de planeación y destaca el desarrollo y refinamiento de casos de uso como un modelo primario.
- Fase de elaboración: incluye la comunicación con el cliente y las actividades de modelado con un enfoque en la creación de modelos de análisis y diseño, con énfasis en las definiciones de clases y representaciones arquitectónicas.
- Fase de construcción: se refina y después se traduce el modelo de diseño en componentes de software ya implementados.
- Fase de transición: se transfiere el software del desarrollador al usuario final para realizar las pruebas betas y obtener la aceptación.

Estas fases se realizan durante varias iteraciones en número variable según el proyecto, haciendo un mayor o menor énfasis en distintas actividades durante las que se pueden distinguir nueve flujos o disciplinas a realizar en mayor o menor medida en dependencia de cada etapa [38].

1.6.3 Metodologías ágiles

Los procesos de desarrollo rápido de software están diseñados para producir software útil de forma rápida. Generalmente, son procesos interactivos en los que se entrelazan la especificación, el diseño, el desarrollo y las pruebas [36].

¹**Desarrollo en Espiral (Boehm 1988)**: serie de ciclos que se repiten en forma de espiral comenzando desde el centro.

Metodología de software XP

Durante la elaboración de la aplicación se utiliza para establecer el control y utilizar un marco de trabajo definido y de probada eficiencia, la metodología de software XP programación extrema, dado que una de las prácticas más significativas que posee, es que con XP es posible simplificar el diseño para agilizar el desarrollo, facilitar el mantenimiento y descartar las ideas que no se necesiten. El ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código. XP tiene como objetivo fundamental la satisfacción del cliente. Es recomendable utilizar esta metodología debido a que los requisitos tienen altas probabilidades de cambiar con el tiempo y el proyecto cuenta con un grupo pequeño de programadores [39]. Además, esta metodología ha sido la utilizada durante el desarrollo del software PANDOC.

Algunas de las características de XP son [36]:

- **Pruebas unitarias:** se basa 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 flexible 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. La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.

¿Qué nos brinda XP? [39]:

- 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 4 fases [39]:

- **Planificación:** es la fase donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada iteración.
- **Diseño:** la metodología XP hace especial énfasis en los diseños simples y claros. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.
- **Codificación:** la fase de codificación se desarrolla en función de cada historia de usuario, además se definen las tareas de la ingeniería y los tiempos reales de cada una de las

funcionalidades especificadas, en la cual la implementación, debe realizarse de acuerdo a los estándares de codificación.

- **Pruebas:** estas pruebas se realizan al final del ciclo en el que se desarrollan, para verificar que las iteraciones no han afectado a las anteriores. Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

SCRUM

SCRUM aplica las mismas premisas conceptuales que XP pero para resolver un problema ligeramente distinto como es el de desarrollo evolutivo de aplicaciones. Es una metodología ágil y flexible que sirve para gestionar el desarrollo de software. Se basa principalmente en construir la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, autogestión e innovación [40].

Con SCRUM el cliente es pieza fundamental en el desarrollo de software, se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo, le permite en cualquier momento realinear el software con los objetivos de negocio de su empresa, pues puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración. Esta forma de trabajo promueve la innovación, motivación y el compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades. SCRUM genera algunas ventajas a diferencia de otras metodologías ágiles [40]:

- Cumplimiento de expectativas: el cliente establece sus expectativas indicando el valor que aporta a cada requisito/historia del proyecto, el equipo los estima y con esta información el propietario del producto establece su prioridad.
- Flexibilidad a cambios: genera una alta capacidad de reacción ante los cambios de requerimientos generados por necesidades del cliente o evoluciones del mercado. La metodología está diseñada para adaptarse a los cambios de requerimientos que conllevan los proyectos complejos.
- Reducción del tiempo: el cliente puede empezar a utilizar las funcionalidades más importantes del proyecto antes de que esté finalizado por completo.
- Mayor calidad del software: La forma de trabajo y la necesidad de obtener una versión funcional después de cada iteración ayuda a la obtención de un software de calidad superior.
- Mayor productividad: se consigue entre otras razones, gracias a la eliminación de la burocracia y a la motivación del equipo que proporciona el hecho de que sean autónomos para organizarse.

- Maximiza el retorno de la inversión (ROI): producción de software únicamente con las prestaciones que aportan mayor valor de negocio gracias a la priorización por retorno de inversión.
- Predicciones de tiempos: mediante esta metodología se conoce la velocidad media del equipo por sprint (los llamados puntos historia), con lo que consecuentemente, es posible estimar fácilmente el tiempo en el que se dispondrá de una determinada funcionalidad que todavía está retrasada.
- Reducción de riesgos: el hecho de llevar a cabo las funcionalidades de más valor en primer lugar y de conocer la velocidad con que el equipo avanza en el proyecto, permite despejar riesgos eficazmente de manera anticipada.

1.6.4 Selección de la metodología de desarrollo de software a utilizar

El análisis de diferentes metodologías de desarrollo de software, teniendo en cuenta las particularidades del algoritmo a desarrollar, permitieron determinar la utilización de la metodología XP debido a:

- A pesar de que RUP busca constantemente calidad en sus procesos, llevando a cabo un robusto análisis, diseño, implementación, soporte y documentación de sistemas informáticos, contando para esto con un equipo de desarrollo grande y un completo dominio de todos los requisitos que cumplirá, no se ajusta a las particularidades de la solución a implementar. El grupo de trabajo es pequeño, lo cual dificulta la adopción de roles en cada etapa, así como la generación de múltiples artefactos, indicando la necesidad de usar una metodología ágil.
- Aunque SCRUM es una metodología ágil se enfoca en las prácticas de organización y gestión, en vez de en la programación como XP. Es la indicada para proyectos con un rápido cambio de requisitos y su poca documentación puede provocar efectos negativos en el modo de confeccionar el producto final.
- Esta metodología ágil está caracterizada por un código simple y abierto a los cambios. Es apropiada para entornos volátiles y se define para equipos de desarrollo pequeños. El equipo de desarrollo definido tiene acceso todo el tiempo al cliente. Este debe permanecer disponible para responder preguntas y fijar prioridades, de forma que no atrase la toma de decisiones, facilitado así la retroalimentación y una planificación más transparente. Todo esto propicia que la tensión no esté solo en la entrega final del producto sino a lo largo de todo el proceso de desarrollo. Además, es la metodología definida por el grupo de investigación.

1.6.5 Lenguaje de Programación

Para el desarrollo de esta aplicación se escogió como lenguaje de programación a Java, este lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel. La selección de dicho lenguaje se debe a que es puro orientado a

objetos, lo que propina una gran reusabilidad, independencia de la plataforma, esto significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware [41]. Existe además variada documentación y una amplia comunidad sobre este lenguaje y el mismo se caracteriza por ser muy utilizado en la implementación de aplicaciones que incorporan técnicas de inteligencia artificial.

Características fundamentales de Java [41]:

- **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 este funcione. Al efectuar al menos un cambio en alguna de las clases, Java se encarga de realizar un enlace dinámico o una carga dinámica para encontrar las clases.
- **Interpretado:** corre en máquina virtual.

1.6.6 Entorno integrado de desarrollo

Para el desarrollo de la solución se seleccionó como Entorno de Desarrollo Integrado (IDE, según sus siglas en inglés) el NetBeans 8.2. Es libre y de código abierto; posee 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. Java es Además es un entorno de desarrollo disponible para varios sistemas operativos como Windows, Mac, Linux y Solaris [41].

1.6.7 Herramienta para el análisis estadístico utilizando SPSS

El programa estadístico SPSS (del inglés *Statistical Package for the Social Sciences*) es uno de los programas de mayor uso en los Estados Unidos de Norteamérica, así como en América Latina. Los procedimientos estadísticos que incluye son de mucha utilidad para desarrollar y subsecuentemente analizar bases de datos para aplicaciones prácticas o para diversas necesidades de investigación.

SPSS permite manejar bancos de datos de gran magnitud y también efectuar análisis estadísticos muy complejos. [42]

Familiarizarse con las diversas opciones y procedimientos estadísticos de un programa como SPSS permite la realización de varios procedimientos estadísticos, brindando gráficos, ejemplos de análisis de hipótesis en una interfaz sencilla y con una curva de aprendizaje pequeña.

Ventajas y desventajas del programa SPSS

SPSS 14.0 se desarrolló en una forma que, aparentemente, es fácil de navegar (user-friendly). El programa utiliza una serie de cuadros de diálogo (dialog boxes) que permiten, en forma secuencial, determinar las acciones a tomar y seleccionar aquellos análisis útiles. Sin embargo, si el usuario no tiene experiencia previa utilizando SPSS o si sus conocimientos de estadística no están actualizados, es difícil discernir que opciones seleccionar. Esta característica del programa puede convertirse en una desventaja e inhibir al usuario. Otro aspecto del programa que puede causar inconvenientes es el hecho de que la mayoría de los reportes de resultados contiene un nivel excesivo de información que, más que aclarar, confunde al usuario. El objetivo es minimizar el uso de términos técnicos y al mismo tiempo entrenar al usuario para que reconozca los términos estadísticos necesarios y descarte aquellos que no se necesitan para su problema específico. Desgraciadamente, el programa SPSS incluye una gran cantidad de información en “forma automática” (by default) que distrae al usuario [42].

1.6.8 Herramienta para el modelado

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés) [43].

VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Ofrece un conjunto completo de herramientas, brindando a los equipos de desarrollo de software todo lo necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos [43].

Ofrece interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo. Hace posible la generación de código Java desde los diagramas [44].

1.7 Conclusiones del capítulo

En el presente capítulo se realizó un estudio sobre los diferentes sistemas desarrollados para la identificación de la capsulorrexis identificando al software PANDOC como sistema a utilizar para el desarrollo del algoritmo por las ventajas que ofrece con respecto a los restantes softwares analizados,

ya que el PANDOC es un software libre que realiza la identificación de la capsulorrexis en pacientes operados de catarata. Además, se seleccionaron y describieron las herramientas y tecnologías a ser usadas en el desarrollo de la propuesta de solución, permitiendo seleccionar Java como lenguaje de programación, NetBeans como IDE, Visual Paradigm como herramienta para el modelado y SPSS como herramienta para el análisis estadístico. El estudio realizado sobre metodologías de desarrollo permitió arribar a la conclusión de que XP, debido a sus características, es la metodología más apropiada para guiar el desarrollo del sistema.

Capítulo 2: Descripción de la propuesta de solución.

En el presente capítulo se realiza una descripción detallada de la solución propuesta de acuerdo a la metodología de desarrollo de software seleccionada. Además, se explica el algoritmo de filtrado que se utilizó en la etapa de pre-procesamiento para la reducción del ruido en las imágenes médicas y las consideraciones del método de segmentación de *Snakes* utilizado con algunas funcionalidades que se le adicionaron para lograr un mejor funcionamiento. Se exponen los artefactos generados por XP para la implementación de la solución que se propone tales como: historias de usuario y tarjetas CRC.

2.1 Reducción de ruido en pre-procesamiento: filtro de la Mediana.

Para lograr una mejor calidad de la imagen a segmentar y evitar errores de segmentación que pueden ser provocados por la aparición de ruido, se aplica como técnica para reducir el ruido el filtro de la Mediana. Esta técnica se adapta a la solución porque crea zonas homogéneas sin empañar ni suavizar los bordes de la imagen. Para aplicar este filtro, se selecciona como tamaño de la máscara un valor impar para garantizar que el vértice central siempre sea único. En la solución propuesta se seleccionan máscaras de 3x3 y 5x5, considerando que máscaras más grandes producen mayor pérdida de información en la imagen y mayor tiempo de ejecución del algoritmo. Una vez seleccionada la máscara, se realiza un proceso iterativo sobre cada píxel de la imagen y se sustituye el valor de intensidad por la media de los valores de intensidad de sus píxeles vecinos. Este proceso se muestra en la Ilustración 13.

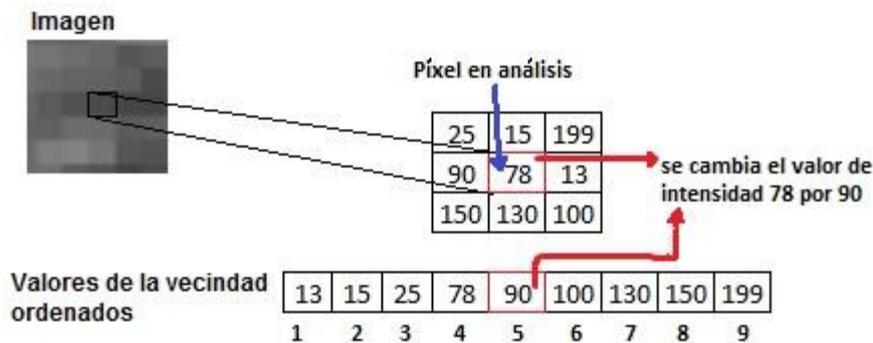


Ilustración 12. Operación del filtro de la Mediana

2.2 Snakes

Una vez aplicado el filtro de la Mediana, la imagen está lista para el proceso de segmentación. Como método de segmentación se emplea el método de *Snakes*. Este método se define por el proyecto debido a que le permite al usuario la selección del área que se desea segmentar y la velocidad que presenta en tiempo de ejecución. En el proceso de aplicación del algoritmo se precisan de dos pasos fundamentales: la selección de la imagen a segmentar y la especificación por parte del usuario del contorno inicial de segmentación. Luego de tener estos datos, se introducen los valores de las variables α , β y γ que controlan la influencia de los términos de la función de energía y se procede a ejecutar el algoritmo.

2.3 Especificación del contorno

Una de las ventajas del algoritmo de *Snakes* es la posibilidad de que el usuario puede introducir el contorno que se utilizará en el proceso de segmentación [31]. Para la especificación del contorno se utiliza un algoritmo que posibilita la inserción de vértices sin intersectar segmentos ni construir lazos. Para esto se inserta el nuevo vértice entre los vértices que componen la arista más cercana al vértice que se desea insertar. También se incorpora la funcionalidad de rellenar las aristas del contorno inicial definido aumentando el número de vértices, lo que posibilita lograr una segmentación de mayor calidad. La definición de una estructura por medio de un polígono es más exacta mientras más vértices contenga este. Esta representación se puede apreciar en la Ilustración 13.

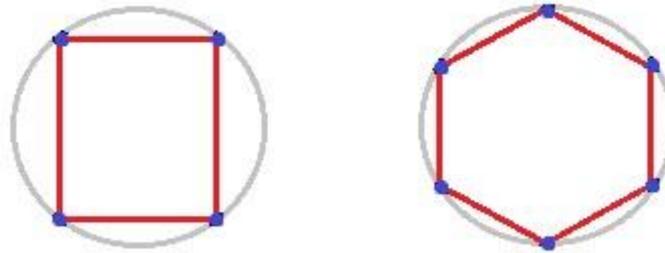


Ilustración 13. *Izquierda: circunferencia representada por un polígono de 4 vértices. Derecha: circunferencia representada por un polígono de 6 vértices*

El algoritmo para rellenar las aristas inserta D/T puntos equidistantes entre sí en cada arista, siendo T la distancia introducida por el usuario y D la distancia euclidiana entre los puntos extremos de cada arista. El resultado que se obtiene se muestra en la Ilustración. 14.

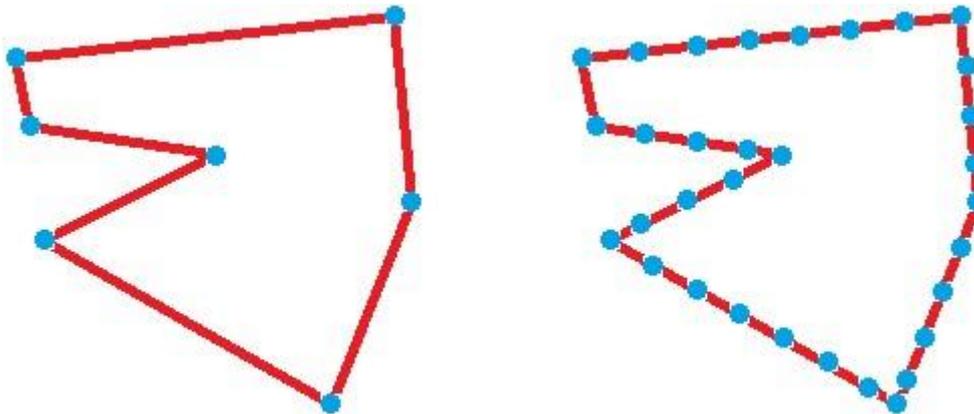


Ilustración 14. *Izquierda: polígono inicial generado. Derecha: polígono luego de aplicado el algoritmo de relleno de vértices*

2.4 Función de energía

Una vez cargada la imagen y dispuesto el contorno de segmentación sobre el área objetivo se ejecuta el algoritmo *Snakes*. Este realiza un proceso iterativo donde se ajusta el contorno inicial a los bordes de la imagen, enmarcando las estructuras con mayor exactitud. Primeramente, es necesario definir la función de energía, así como cada uno de sus términos para lograr un correcto comportamiento del *Snake*. La función de energía se utiliza con los siguientes valores: $\alpha = 0.6$, para que los puntos tengan

cierta libertad a la hora de separarse entre ellos; $\beta = 0.2$, para una mejor segmentación de las zonas pronunciadas o bordes salientes de las estructuras presentes en la imagen y $\gamma = 1.0$, para una mayor sensibilidad a los cambios de intensidad en la imagen. Estos valores pueden cambiar por decisión del usuario en dependencia de las características de la imagen.

2.5 Términos de la función de energía

En el epígrafe 1.5.2.4 se describe la función de manera muy general. En la función de energía están presentes los términos de energía interna y energía externa. La energía interna se define por los términos E_{cont} y E_{curv} y la energía externa se define por el término E_{imag} , llevándola al uso del algoritmo, se centra en los snaxels escogidos y su representación. Para el cálculo de la energía del contorno se utiliza la siguiente expresión:

$$E_{cont} = \left(d - \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \right)^2 \quad (10)$$

Donde:

d : representa la distancia media entre cada *snaxel*.

(x_i, y_i) : representa las coordenadas del *snaxel* sobre el que se está evaluando la función de energía.

(x_{i-1}, y_{i-1}) : representa las coordenadas del *snaxel* anterior al que se le está evaluando la función de energía.

Para el cálculo del término de la energía de la curvatura se utiliza la siguiente expresión:

$$E_{curv} = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \quad (11)$$

Donde:

(x_i, y_i) : representa las coordenadas del *snaxel* sobre el que se está evaluando la función de energía.

(x_{i+1}, y_{i+1}) : representa las coordenadas del *snaxel* siguiente al que se le está evaluando la función de energía.

(x_{i-1}, y_{i-1}) : representa las coordenadas del *snaxel* anterior al que se le está evaluando la función de energía.

Para el cálculo de la energía de la imagen como método de cálculo del gradiente se utiliza el operador de Sobel, debido a que posee una aceptable inmunidad al ruido y ofrece resultados de gran calidad utilizando pocas operaciones matemáticas complejas, haciéndolo más veloz que otros operadores [31].

2.6 Descripción del algoritmo Snakes

El algoritmo *Snakes* se define como un algoritmo *greedy* que toma óptimos locales en cada iteración que contribuyen a la obtención de un óptimo global. La especificación del mismo es la siguiente [31]:

La entrada es una imagen conteniendo el contorno objetivo y vértices p_1, p_2, \dots, p_n definiendo la posición inicial y forma del *Snake*.

1. Para cada p_i , $i = 1, \dots, N$, se busca en su vecindad $M \times M$ para encontrar el lugar que minimice la función de energía, ver Ilustración 15.
2. Se mueve p_i a ese lugar.
3. Se actualiza el valor de d (promedio de la distancia entre los vértices).
4. Se repiten los pasos 1-3 hasta que todos los vértices alcancen su valor mínimo de función de energía.

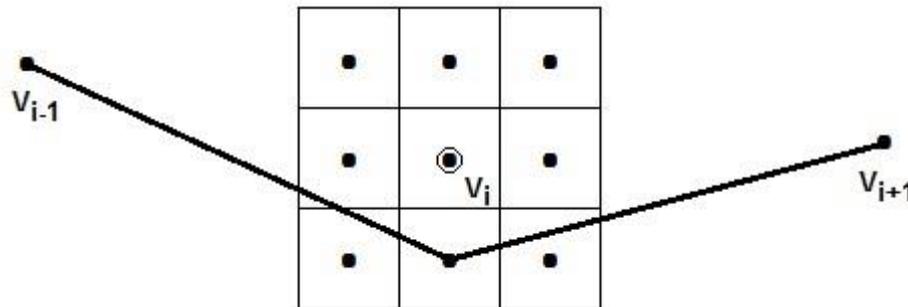


Ilustración 15. Representación de la vecindad de búsqueda de un snaxel

Siendo M el tamaño de la matriz de búsqueda de la vecindad de cada punto. Cuanto mayor sea M , mayor será el número de posiciones a las que se puede mover el punto del *Snake* y también será mayor el tiempo de procesamiento de cada iteración del algoritmo [31].

Desde el punto de vista de la programación un primer acercamiento al funcionamiento de este algoritmo podría mostrarse a través de su pseudocódigo:

2.7 Pseudocódigo del algoritmo

A continuación el pseudocódigo del algoritmo:

//Inicializar los pesos de las energías

HACER

Inicializar α β γ

REPETIR $x = 0$ HASTA $\text{num_snaxels}-1$

ENERGIAmin = valor_inicial;

Enc = 0;

REPETIR $y = 0$ HASTA $\text{num_vecinos}-1$

ENERGIAcontorno = CalcularEnergiaContorno(x, y);

ENERGIAcurvatura = CalcularEnergiaCurvatura(x, y);

ENERGIAimagen = CalcularEnergiaImagen(x, y);

ENERGIAglobal = α *ENERGIAcontorno + β *ENERGIAcurvatura + γ *ENERGIAimagen;

SI (ENERGIAglobal < ENERGIAmin) ENTONCES

ENERGIAmin = ENERGIAglobal;

Pos = y ;

```

Enc = 1;
FINSI
FINREPETIR
IF (Enc==1) ENTONCES
    //mover el snaxel a la nueva posición
    Pos Veces_movido = Veces_movido + 1;
FINSI
FINREPETIR
HASTA QUE (Todos los nodos alcancen su energía mínima).
    
```

Para dar solución a la problemática planteada se decidió la implementación del algoritmo Snake como método de segmentación. En la ilustración a continuación se muestra el funcionamiento de este método.

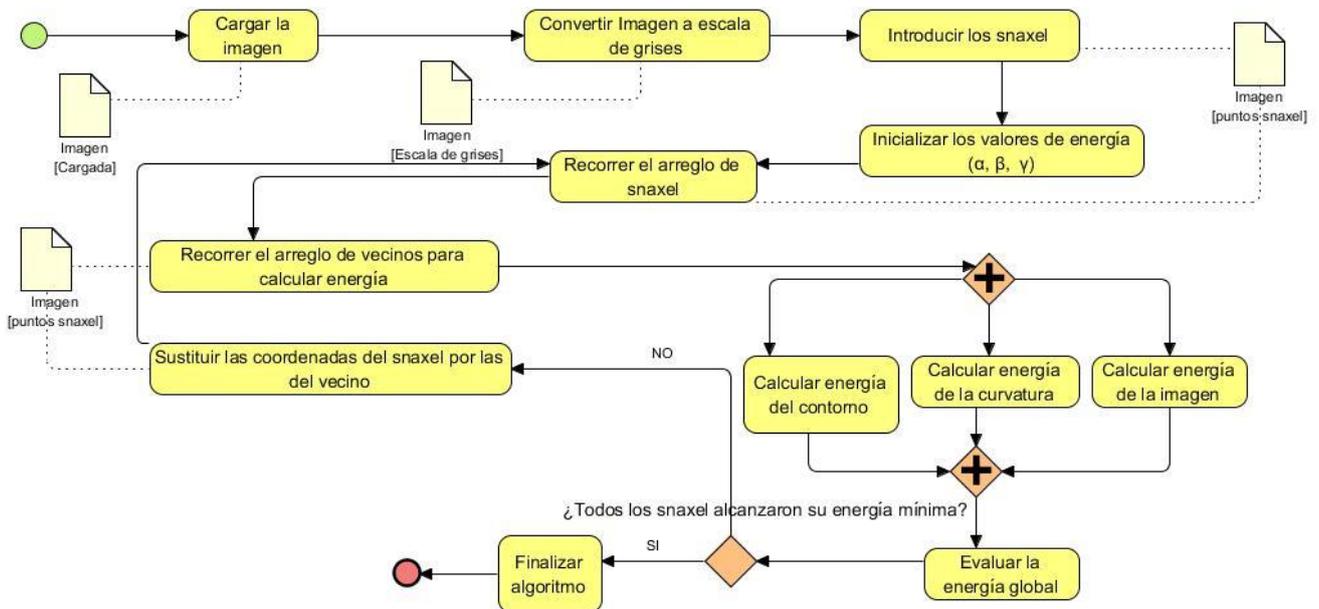


Ilustración 16. Pasos del Algoritmo Snake

A continuación, se describe paso a paso el funcionamiento de Snake.

Paso 1: Cargar la imagen

El algoritmo inicia al cargar una imagen. Para lograr la calidad de la imagen a segmentar y evitar errores de segmentación que pueden ser provocados por la aparición de ruido, se aplica como técnica para reducir el ruido el filtro de la Mediana.

Paso 2: Convertir la imagen en escala de grises

Se convierte la imagen a escala de grises para un mejor trabajo con la misma, este proceso es imperceptible para el usuario, se recorre la imagen píxel a píxel, se almacena el color del píxel y se calcula la media de los tres canales.

Paso 3: Introducir los snaxel

Se introducen los puntos que representan las coordenadas (x,y) , colocando un punto en el centro de la imagen dándole un radio desde el punto hasta el borde de la imagen para que los snaxel se posicionen alrededor de la imagen de forma automática para lograr que el algoritmo converja en el menor tiempo posible.

Paso 4: Inicializar los valores de la energía

Primeramente, es necesario definir la función de energía, así como cada uno de sus términos para lograr un correcto comportamiento del *Snake*. La función de energía se utiliza con los siguientes valores: $\alpha = 0.6$, para que los puntos tengan cierta libertad a la hora de separarse entre ellos; $\beta = 0.2$, para una mejor segmentación de las zonas pronunciadas o bordes salientes de las estructuras presentes en la imagen y $\gamma = 1.0$, para una mayor sensibilidad a los cambios de intensidad en la imagen y la energía mínima para cálculos posteriores dentro del algoritmo.

Paso 5: Recorrer el arreglo de snaxels

Los puntos se adicionan en un arreglo de snaxel después se recorre el arreglo de snaxel generando para cada snaxel un arreglo de vecinos.

Paso 6: Recorrer el arreglo de vecinos

Se construye el arreglo de vecinos de 8 posiciones, se itera por cada uno de los vecinos del arreglo. Se calcula la energía del contorno que fuerza a los *snaxels* a contraerse en busca del contorno objetivo y a moverse de manera uniforme, se calcula a través de la fórmula (10).

$$E_{cont} = \left(d - \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \right)^2$$

Paso 6.1:

Se calcula la energía de la curvatura evita que se curve en exceso haciendo que se mantenga su forma mientras encuentra las zonas curvas de la figura a detectar se calcula a través de fórmula (11).

$$E_{curv} = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2$$

Paso 6.2:

Se calcula la energía de la imagen que es la encargada de atraer al *Snake* hacia los puntos de mayor contraste de la imagen, se representa como el gradiente de los píxeles de la imagen con signo negativo logrando que en los puntos de mayor contraste la energía de la imagen sea menor, para poder buscar estos puntos mientras se minimiza la función de energía, se calcula a través de la fórmula (9)

$$E_{imag} = -\|\nabla I\|$$

Paso 6.3:

Se calcula la energía global a través de la siguiente fórmula:

(Energía_{global}= α *Energía_{contorno}+ β *Energía_{curvatura}+ γ *Energía_{imagen}) y verificar que la misma sea menor que la energía mínima.

Paso 7: Sustituir las coordenadas del snaxel por las del vecino

Si la energía global es menor que la energía mínima actualizo la energía mínima y guardo la coordenada (x, y) del vecino en la posición actual del arreglo de snaxel, ya que por cada snaxel se genera un arreglo de vecinos.

Paso 8: Repetir hasta que todos los snaxel alcancen su energía mínima

Se repite el ciclo hasta que la energía mínima sea la misma, es decir, hasta que ningún vecino tenga una energía mínima menor a la del snaxel.

2.8 Requisitos.

Funcionales

Los requisitos funcionales representan las funcionalidades del sistema [45]. Los siguientes requisitos responden a las funcionalidades que el sistema debe tener una vez concluida su implementación.

RF1. Cargar imagen.

RF2. Filtrar la imagen.

RF3. Determinar la distancia media.

RF4. Determinar la energía del contorno.

RF5. Determinar la energía de la curvatura.

RF6. Determinar la energía de la imagen.

RF7. Determinar el color del punto.

RF8. Segmentar la imagen.

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 [46]. 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 debe tener como mínimo las siguientes características:

- 1 GB de memoria RAM.
- Más de un núcleo físico con frecuencia mayor de 1.6 GHz
- el algoritmo fue probado en una PC con estas prestaciones por lo que se recomienda que tenga estas características o superiores.

2.9 Historia de Usuario

La metodología XP define como fase inicial del desarrollo de software la planificación [47]. Durante esta etapa se realizan las historias de usuario (HU), el equipo de trabajo se familiariza con las tecnologías y herramientas de desarrollo. Como resultado de la fase se tiene un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su elaboración.

Las Historias de Usuario se definen a continuación.

HU1. Cargar imagen

HU2. Filtrar imagen

HU3. Determinar la distancia media

HU4. Determinar la energía del contorno

HU5. Determinar la energía de la curvatura

HU6. Determinar la energía de la imagen

HU7. Determinar el color del punto

HU8. Segmentar la imagen

Tabla 1. HU1 Cargar imagen

| Historia de usuario | |
|--|-----------------------------------|
| Número: 1 | Nombre: Cargar imagen |
| Iteración asignada: 1 | |
| Prioridad en negocio: Alta | Puntos estimados: 1 semana |
| Riesgo en desarrollo: Alto | Puntos reales: 1 semana |
| Descripción: selecciona el directorio donde están las imágenes y las visualiza. | |
| Observaciones: las imágenes deben ser las provenientes del Pentacam. | |

Tabla 2. HU2 Filtrar imagen

| Historia de usuario | |
|------------------------------------|------------------------------------|
| Número: 2 | Nombre: Filtrar imagen |
| Iteración asignada: 1 | |
| Prioridad en negocio: Medio | Puntos estimados: 3 semanas |

| | |
|---|---------------------------------|
| Riesgo en desarrollo: Medio | Puntos reales: 3 semanas |
| Descripción: elimina el ruido presente en la imagen y realza los bordes de la misma. | |
| Observaciones: la imagen debe ser proveniente del PENTACAM . | |

Tabla 3. HU3 Determinar la distancia media

| | |
|--|--|
| Historia de usuario | |
| Número: 3 | Nombre: Determinar la distancia media |
| Iteración asignada: 1 | |
| Prioridad en negocio: Alta | Puntos estimados: 1 semana |
| Riesgo en desarrollo: Medio | Puntos reales: 1 semana |
| Descripción: calcula el promedio de la distancia entre los snaxel y se devuelve su valor medio. | |
| Observaciones: los snaxel ya fueron colocados. | |

Tabla 4. HU4 Determinar la energía del contorno

| | |
|--|---|
| Historia de usuario | |
| Número: 4 | Nombre: Determinar la energía del contorno |
| Iteración asignada: 2 | |
| Prioridad en negocio: Alta | Puntos estimados: 3 semanas |
| Riesgo en desarrollo: Alto | Puntos reales: 3 semanas |
| Descripción: se utiliza para que los puntos tengan cierta libertad a la hora de separarse entre ellos, para una mejor segmentación de las zonas pronunciadas o bordes salientes de las estructuras presentes en la imagen y para una mayor sensibilidad a los cambios de intensidad en la imagen. | |
| Observaciones: la imagen debe estar cargada y dispuesto el contorno sobre el área objetivo que se desea segmentar. | |

Tabla 5. HU5 Determinar la energía de la curvatura

| | |
|---------------------|--|
| Historia de usuario | |
| Número: 5 | Nombre: Determinar la energía de la curvatura |

| | |
|---|------------------------------------|
| Iteración asignada: 2 | |
| Prioridad en negocio: Alta | Puntos estimados: 3 semanas |
| Riesgo en desarrollo: Alto | Puntos reales: 3 semanas |
| Descripción: se representan las coordenadas del snaxel sobre el que se está evaluando la función de energía, así como las coordenadas del snaxel siguiente y el snaxel anterior, también se les evalúan la función de energía. | |
| Observaciones: la imagen debe estar cargada y dispuesto el contorno sobre el área objetivo que se desea segmentar. | |

Tabla 6. HU6 Determinar la energía de la imagen

| | |
|---|---|
| Historia de usuario | |
| Número: 6 | Nombre: Determinar la energía de la imagen |
| Iteración asignada: 2 | |
| Prioridad en negocio: Alta | Puntos estimados: 2 semanas |
| Riesgo en desarrollo: Alto | Puntos reales: 2 semanas |
| Descripción: calcula la energía de la imagen especificando el punto visitado. | |
| Observaciones: la imagen debe estar cargada y dispuesto el contorno sobre el área objetivo que se desea segmentar. | |

Tabla 7. HU7 Determinar el color del punto

| | |
|---|--|
| Historia de usuario | |
| Número: 7 | Nombre: Determinar el color del punto |
| Iteración asignada: 3 | |
| Prioridad en negocio: Alta | Puntos estimados: 2 semanas |
| Riesgo en desarrollo: Alto | Puntos reales: 2 semanas |
| Descripción: calcula el color del punto del snaxel visitado. | |
| Observaciones: la imagen debe estar cargada. | |

Tabla 8. HU8 Segmentar la imagen

| |
|---------------------|
| Historia de usuario |
|---------------------|

| | |
|--|------------------------------------|
| Número: 8 | Nombre: Segmentar la imagen |
| Iteración asignada: 3 | |
| Prioridad en negocio: Alta | Puntos estimados: 2 semanas |
| Riesgo en desarrollo: Alto | Puntos reales: 2 semanas |
| Descripción: permite identificar la opacidad y la capsulorrexis en la imagen. | |
| Observaciones: la imagen debe estar filtrada. | |

2.10 Plan de iteraciones

Luego de identificar y definir las HU y estimar el esfuerzo propuesto para la realización de cada una de ellas, se propone un plan de iteraciones donde las historias de usuario están contenidas, regulando la cantidad de HU a implementar dentro del rango establecido por la estimación efectuada. Tomando como referencia los aspectos antes tratados la aplicación que se pretende construir se desarrollará en 3 iteraciones, explicadas más detalladamente a continuación:

Iteración 1

La iteración tiene como finalidad implementar las HU que se consideraron más necesarias atendiendo a su relevancia e impacto. En esta iteración se carga la imagen, se filtra y se determina la distancia media entre los puntos de cada snaxel.

Iteración 2

En esta iteración se realizan todas las HU relacionadas con determinar las energías del contorno, la curvatura y la imagen.

Iteración 3

En esta iteración se realizan todas las HU relacionadas con determinar el color del punto y segmentar imagen.

Tabla 9. Plan de duración de las iteraciones

| Iteraciones | Historias de usuario | Duración |
|-------------|---------------------------------------|----------|
| Iteración 1 | Cargar Imagen | 5 días |
| | Filtrar Imagen | |
| | Determinar la distancia media | |
| Iteración 2 | Determinar la energía del contorno | 5 días |
| | Determinar la energía de la curvatura | |

| | | |
|-------------|------------------------------------|--------|
| | Determinar la energía de la imagen | |
| Iteración 3 | Determinar el color del punto | 5 días |
| | Segmentar Imagen | |

2.11 Plan de entrega

El plan de entregas es el cronograma que define cuáles HU se agrupan para conformar una entrega, y el orden en que se realizan estas [48]. Este plan se encarga de definir el número de liberaciones que se realizan en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una de ellas.

Tabla 10. Plan de entrega

| | Iteración 1 | Iteración 2 | Iteración 3 |
|-------------------------|-------------|-------------|-------------|
| Cantidad de HU | 3 | 3 | 2 |
| Fecha de entrega | 24/01/2018 | 20/03/2018 | 28/05/2018 |

2.12 Tarjetas CRC

Siguiendo los planteamientos de la metodología XP debe ser un diseño simple y de fácil interacción, basándose fundamentalmente en el desarrollo de las tarjetas Clases, Responsabilidad y Colaboración (CRC). Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño. A continuación, se evidencia en la tabla 11 la tarjeta CRC correspondiente a la clase Snakes. El resto de las tarjetas se encuentran en los anexos.

Tabla 11. Tarjeta CRC para la clase Snakes

| Clase: Snakes | |
|---|------------------------|
| Responsabilidades | Colaboradores |
| 1. Implementa los métodos necesarios para ejecutar el algoritmo Snake. 2. Crear objeto de tipo Snaxel. | 1. Snakes 2. Snaxel |

2.13 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada [49]. Para la realización del presente trabajo se establecen los siguientes estándares de codificación:

- Todas las nomenclaturas a utilizar se definirán en idioma español.
- Los nombres de las clases serán con mayúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán de igual forma.
- Los nombres de los métodos serán con minúscula, en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula.
- Los identificadores para las variables y los parámetros serán con letras en minúsculas y en caso de ser un nombre compuesto las siguientes palabras se escribirán con mayúscula.
- Los nombres de variables o funciones deben ser lo suficientemente descriptivos, sin exceder de 30 caracteres.
- Todas las funciones deben tener comentarios explicando que realiza cada una de ellas.

2.14 Patrones de diseño utilizados

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares [50].

Un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. Para ello se aíslan los aspectos comunes y su solución y se añaden cuantos comentarios y ejemplos sean oportunos. Los patrones ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexibles, elegante y en algunos casos reusable [51].

Los Patrones Generales de Software para Asignación de Responsabilidades (*General Responsibility Assignment Software Patterns*, GRASP por sus siglas en inglés), son los que ofrecen orientación de cómo asignar estas a los objetos ante determinada categoría de problemas, describiendo los principios fundamentales de la asignación [52]. Los que se evidencian en el diseño del sistema son: Experto, Bajo acoplamiento, Alta cohesión, Controlador y Creador.

- **Experto:** asigna la responsabilidad al experto en la información, es decir, a la clase que cuenta con la información necesaria para cumplir con la responsabilidad [52]. En la ilustración 17 se evidencia la aplicación del patrón en la clase Snaxel que tiene la información necesaria para ser experta en el trabajo con los puntos de las coordenadas (x,y).

```
14 public class Snaxel {
15     private int x;
16     private int y;
17
18     + public Snaxel(int x, int y) {...4 lines }
22
23     + public int getX() {...3 lines }
26
27     + public void setX(int x) {...3 lines }
30
31     + public int getY() {...3 lines }
34
35     + public void setY(int y) {...3 lines }
38
39 }
```

Ilustración 17. Empleo del patrón Experto

- **Bajo acoplamiento:** el propósito del patrón es tener las clases lo menos ligadas posible y de producirse una modificación en alguna de ellas, se tenga la mínima repercusión en el resto. Al no asociar las clases de la capa de presentación con la de negocio o la de datos, la dependencia entre las clases se mantiene baja [52]. En la ilustración 18 se evidencia el uso del patrón mostrando que las relaciones que se establecen entre las clases es la necesaria para cumplir su función.

```
23 a) public class Imagen extends javax.swing.JPanel{
19 b) public class Snakes {
```

Ilustración 18. Empleo del patrón Bajo Acoplamiento en a y b

- **Alta cohesión:** la información que almacena una clase debe ser coherente y en la medida de lo posible relacionada con ella [52]. En la ilustración 19 se ejemplifica el patrón a través de la clase imagen con los métodos (paint(), getImageActual(), setImageActual()).

```
23 public class Imagen extends javax.swing.JPanel{
24     private BufferedImage imageActual;
25
26     public Imagen(){
27         this.setSize(256,256);
28     }
29
30     @Override
31     public void paint(Graphics grafico){
32         Dimension heigth=getSize();
33         grafico.drawImage(imageActual, 0, 0, heigth.height,heigth.width,null);
34         setOpaque(false);
35         super.paint(grafico);
36     }
37
38     public BufferedImage getImageActual() {
39         return imageActual;
40     }
41
42     public void setImageActual(BufferedImage imageActual) {
43         this.imageActual = imageActual;
44     }
}
```

Ilustración 19. Empleo del patrón Alta cohesión

- **Controlador:** mantiene el control actuando como intermediario entre una determinada interfaz y el algoritmo que la implementa. Además, recibe los datos del usuario y los envía a las distintas clases según el método llamado y permite dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento [53]. En la ilustración 20 se ejemplifica el patrón a través de la clase Snakes que es la manejadora de los eventos.

```
public Snakes() { ...5 lines }
public Snakes(LinkedList<Snaxel> snaxeles, BufferedImage imgAct) { ...5 lines }
public double distancia_Media() { ...13 lines }
public double calcularEnergiaContorno(int pos) { ...10 lines }
public double calcularEnergiaCurvatura(int pos) { ...11 lines }
public double calcularEnergiaImagen(int pos) { ...14 lines }
public void contruirVecinos(int x, int y) { ...10 lines }
```

Ilustración 20. Empleo del patrón Controlador

Creador: se encarga de guiar la asignación de responsabilidades relacionadas con la creación de objetos. Su intención es encontrar un creador que necesite conectarse al objeto creado en alguna situación [54]. Dentro de la aplicación este patrón se observa en las acciones del controlador. En la ilustración 21 se ejemplifica el patrón porque la clase Snakes crea instancias de la clase Snaxel.

```
17 public class Snakes {
18
19     private Snaxel[] snaxels;
20     private Snaxel[] svecinos;
21     private int max_num_snaxels;
22     private int CR_snaxels;
23     private BufferedImage imgAct;
24
25     public Snakes() {
26         this.snaxels = new Snaxel[max_num_snaxels];
27         this.svecinos = new Snaxel[8];
28         this.CR_snaxels = 0;
29         this.imgAct = imgAct;
30     }
31 }
```

Ilustración 21. Empleo del patrón Creador

2.15 Conclusiones del capítulo

En este capítulo se ha presentado el ciclo de vida de la solución propuesta según la metodología XP, generando las HU que caracterizan al sistema, así como la estimación del esfuerzo necesario para la implementación de las mismas. Además, se construyó el plan de iteraciones, se realizó la estimación del tiempo que requiere la implementación de cada una y se llevó a cabo el plan de entregas en el cual se define qué historias de usuario forman parte de cada entrega. Fueron realizadas las tarjetas CRC correspondientes a cada clase y quedan definidos cada uno de los pasos realizados durante el desarrollo del sistema.

Capítulo 3: Resultados y validación del sistema

En el presente capítulo se presenta un análisis estadístico de los resultados del conjunto de métricas implementadas, identificando el algoritmo que estadísticamente presenta los mejores resultados en las imágenes provenientes del software PANDOC. Se realiza una validación de las métricas implementadas y se ejecutan pruebas al sistema, describiendo también la fase de implementación.

3.1 Análisis estadístico de los resultados en imágenes del PANDOC

Las pruebas estadísticas permiten contrastar la veracidad o falsedad de las hipótesis enunciadas desde el punto de vista estadístico. Las cuales se clasifican en dos grandes grupos de pruebas de significación estadística, el referido a las paramétricas y el relacionado con las no paramétricas, con rasgos distintivos que las caracterizan [55].

3.1.1 Pruebas Paramétricas

Las paramétricas exigen a los datos a los que se aplica, que se cumplan los siguientes requisitos [55]:

- Variable numérica: que las variables de estudio estén medidas en una escala que sea por lo menos de intervalo.
- Normalidad: que los valores de las variables sigan una distribución normal, por lo menos, en la población al que pertenece la muestra.
- Homocedasticidad: que las varianzas de la variable dependiente en los grupos que se comparan sean aproximadamente iguales (homogeneidad de las varianzas).

Cuando los datos cumplen con los requisitos indicados, las pruebas estadísticas paramétricas exhiben su máximo poder. Cuando estas pruebas estadísticas se aplican a datos que no cumplen al menos uno de los requisitos señalados, pierde parte de su poder. Si no se cumple una de las tres condiciones se consideran las pruebas no paramétricas, las cuales no hacen ningún tipo de suposición acerca de la forma exacta de la población en la que fueron extraídas las muestras [55].

3.1.2 Pruebas no Paramétricas

Los métodos no paramétricos son la manera más directa de solucionar el problema de falta de normalidad. Estos métodos son muy simples de usar y están disponibles en SPSS [56]. Pero tienen dos desventajas. Primero que tienen menos poder que las equivalentes soluciones paramétricas y que las pruebas de hipótesis no paramétricas no contestan a la misma pregunta que las pruebas paramétricas. Por ejemplo, si queremos hacer un test para docimar sobre el centro de la distribución, el test no paramétrico establece la hipótesis en términos de la mediana y el test paramétrico usa la media. Además existen dos grandes tipos de test no paramétricos, los que usan cuentas o números y los que usan rangos [57].

El tamaño de la muestra es de solo 28 elementos, lo cual es la principal razón que dificulta probar la normalidad de los datos, lo que indica el uso de pruebas no paramétricas. En este trabajo se usa el test de suma de rangos de Wilcoxon [58] y el Test de Friedman [59] como las pruebas a utilizar.

3.1.2.1 Test de suma de rangos de Wilcoxon

Se tiene una Muestra Aleatoria Simple (MAS) de tamaño n_1 de una población, y una segunda MAS de tamaño n_2 de otra población. Hay n observaciones en total, donde $n = n_1 + n_2$. Se calcula el rango de las n observaciones. El test estadístico será la suma W de los rangos del grupo con menor suma de rangos, este será el estadístico de suma de rangos de Wilcoxon [58]. Si las dos poblaciones tienen la misma distribución continua, entonces W tiene media:

$$\mu_W = \frac{n_1(n+1)}{2} \quad (12), \text{ siendo:}$$

μ_W : media de los rangos

n : cantidad de observaciones en total

n_1 : tamaño de una población

y desviación estándar:

$$\sigma_W = \sqrt{\frac{n_1 n_2 (n+1)}{12}} \quad (13), \text{ siendo:}$$

σ_W : desviación estándar de los rangos

n : cantidad de observaciones en total

n_1 : tamaño de una población

n_2 : tamaño de una segunda población

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

El test de suma de rangos de Wilcoxon rechaza la hipótesis nula de que las dos poblaciones tienen la misma distribución cuando la suma de rangos W está lejos de su media [58].

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

Las hipótesis que se prueban con la prueba de signo y la prueba de rango con signo de Wilcoxon son las siguientes [60] :

H_0 : La probabilidad de una diferencia positiva es igual a la probabilidad de una diferencia negativa.

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

3.1.2.2 Test de Friedman

La prueba de Friedman es una alternativa no paramétrica a un análisis de varianza de medidas repetidas utilizado para comparar observaciones repetidas sobre los mismos sujetos, a diferencia de las medidas paramétricas ANOVA [60]. Además, si la investigación tiene k variables en columnas y n elementos en filas se trata de ordenar cada fila de menor a mayor según las diferentes columnas de 1 hasta k (esto es el rango que ocupa cada variable para ese caso). Si no hay diferencias estadísticamente significativas entre las variables se espera que los rangos estén repartidos en cada columna de manera uniforme y sólo se encontrarán entre las variables pequeñas diferencias debidas al azar.

Hipótesis para la prueba de Friedman

Las hipótesis que se prueban en la prueba de Friedman son las siguientes [60]:

H_0 : Las distribuciones son las mismas a través de medidas repetidas.

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

3.2 Pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas [61]. 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 [62]. XP exhorta constantemente a realizar pruebas tanto como sea posible, permitiendo aumentar la calidad de los sistemas, reduciendo el número de errores no detectados [63]. 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].

3.2.1 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. A uno de los métodos que se le aplicaron estas pruebas fue al método "distanciaMedia". A este se le aplica la métrica de complejidad ciclomática debido a su relevancia para el sistema.

```
public double distanciaMedia() {  
    double distancia = 0;  
    for (int i = 0; i < arr_smaxel.size(); i++) {  
        if (i == arr_smaxel.size() - 1) {  
            distancia += Math.sqrt((Math.pow(arr_smaxel.get(0).getX() - arr_smaxel.get(i).getX(), 2)) +  
                (Math.pow(arr_smaxel.get(0).getY() - arr_smaxel.get(i).getY(), 2)));  
        } else {  
            distancia += Math.sqrt((Math.pow(arr_smaxel.get(i + 1).getX() - arr_smaxel.get(i).getX(), 2)) +  
                (Math.pow(arr_smaxel.get(i + 1).getY() - arr_smaxel.get(i).getY(), 2)));  
        }  
    }  
    return distancia / arr_smaxel.size();  
}
```

Ilustración 22. Método para el cálculo de distancia media

Partiendo del fragmento de código mostrado en la Ilustración 22 se obtuvo el grafo de flujo que muestra la Ilustración 23.

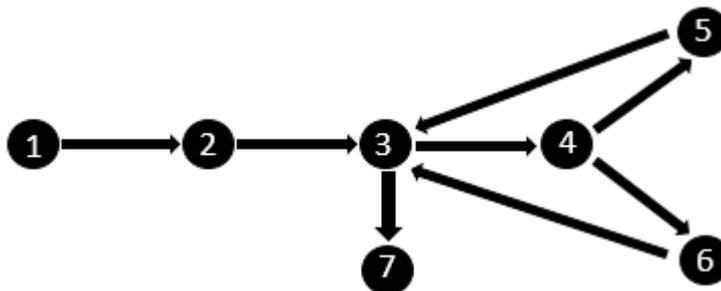


Ilustración 23. Diagrama del flujo del método "distanciaMedia"

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:

$$V(G) = (A - N) + 2 = (8 - 7) + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

$$V(G) = R = 3$$

Tras aplicar las tres formas para afirmar un resultado estable, se obtiene una complejidad ciclomática $V(G) = 3$ la cual representa la cantidad de caminos independientes para el grafo de flujo construido para el método "distanciaMedia", por lo que se definen los 3 caminos: 1-2-3-7, 1-2-3-4-5-3-7 y 1-2-3-4-6-3-7.

Se realizaron 3 iteraciones de estas pruebas, encontrándose en la primera iteración 8 no conformidades, en la segunda 3 y en la tercera ninguna. Entre las no conformidades identificadas estuvieron errores ortográficos y funcionalidades incorrectas.

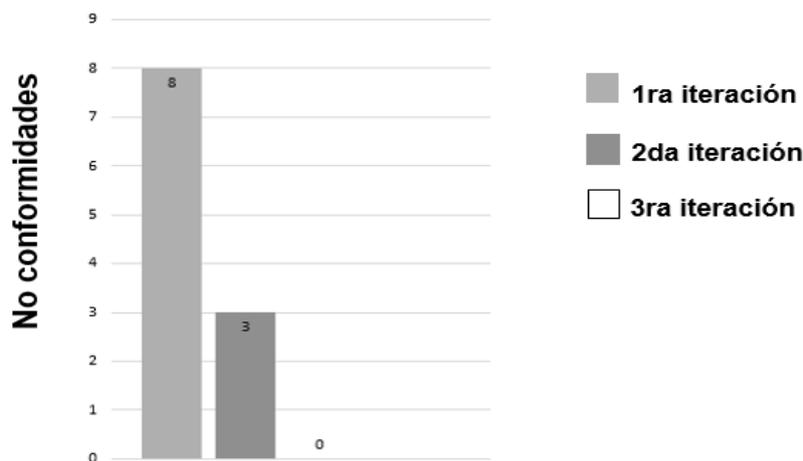


Ilustración 24. Resultados de las pruebas unitarias

3.2.2 Pruebas de Aceptación

Cada prueba de Aceptación está asociada a una funcionalidad del sistema, estas pruebas se realizan en esta etapa del proyecto describiéndose en ellas las posibles formas de utilización del software. Estas pruebas funcionales no sólo validan la transformación de una entrada en una salida, sino que validan una característica completa [67]. En estas pruebas 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 emita cuando se utiliza dicha funcionalidad [68].

Se realizaron 3 iteraciones de estas pruebas, encontrándose en la primera iteración 7 no conformidades, en la segunda 2 y en la tercera ninguna. Entre las no conformidades identificadas estuvieron el insuficiente tratamiento de excepciones y la falta de claridad de respuesta del algoritmo.

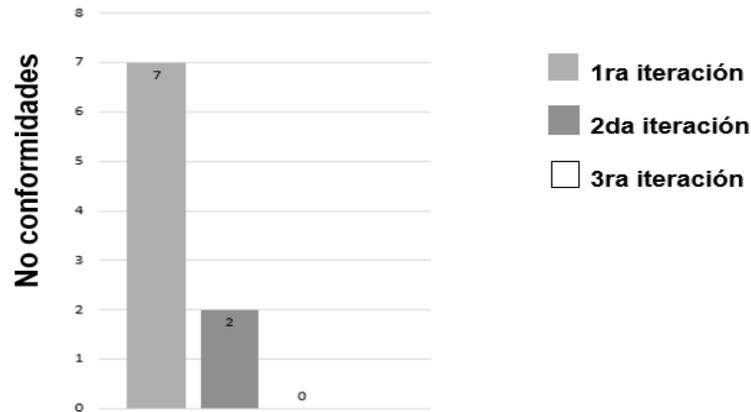


Ilustración 25. Resultados de las pruebas de caja negra

3.2.3 Análisis de los resultados en las imágenes del PENTACAM

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 el algoritmo propuesto (Snake) y el algoritmo TCH_CA con el fin de seleccionar el algoritmo más eficiente para la detección de la capsulorrexis en imágenes provenientes del PENTACAM. Estas comparaciones fueron ejecutadas sobre un conjunto de 28 imágenes de pacientes con esta complicación postoperatoria por cada uno de los algoritmos antes mencionados.

Para esta labor se confeccionó un equipo de 3 especialistas del servicio de cataratas del ICO con amplia experiencia en el trabajo con la OCP:

- Dr. Iván Hernández López, Especialista de I grado en Oftalmología, Investigador Auxiliar.
- Dr. Eneida Pérez Candelaria, Especialista de II grado en Oftalmología, Profesor Auxiliar, Investigador Auxiliar.
- Dr. Zucell Veitía Rovirosa, Especialista de II grado en Oftalmología, Profesor Auxiliar, Investigador Auxiliar.

Estos se reunieron y marcaron en consenso en las 28 imágenes las regiones que consideraron opacidad. Estas imágenes *ground truth* facilitadas por el especialista fueron usadas en la validación de la segmentación por los algoritmos como se muestra en la Ilustración 26.



a-) Ground Truth

b-) Snake

c-) TCH_CA

Ilustración 26. Imágenes del Pentacam segmentadas por los algoritmos propuestos(b,c) y la imagen de referencia marcada por los especialistas(a)

A continuación, se analizan los resultados (Mínimo Valor, Máximo Valor, Media y Desviación Estándar) de las métricas (Desajuste de Bordes, Tasa de Correctos Equilibrada, Índice de Sokal-Sneath, Índice de Rogers-Tanimoto, Índice de Yule) aplicadas a las segmentaciones resultantes de los algoritmos propuestos.

Desajuste de Bordes: penaliza las discrepancias entre el mapa de borde de la imagen de nivel gris y el mapa de borde obtenido de la imagen de umbral. Mientras más cercano esté su valor a 1, menor cantidad de elementos del fondo fueron identificados como parte de la enfermedad [69] [70]. Los resultados de la misma sobre los algoritmos son mostrados en la Ilustración 27, donde se puede apreciar la superioridad en los valores del algoritmo Snake en comparación con los valores del algoritmo TCH_CA ya que los valores de la desviación estándar del algoritmo Snake están más cercanos a 1, al igual que los de la media.

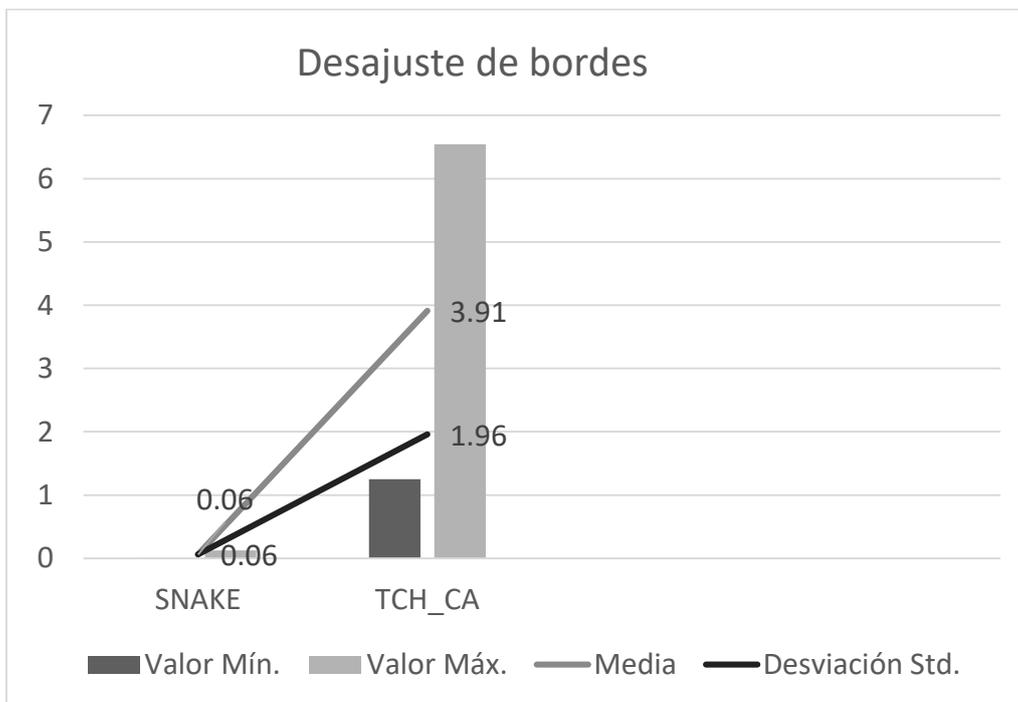


Ilustración 27. Resultados de la métrica Desajuste de Bordes en los dos algoritmos

Tasa de Correctos Equilibrada: esta toma valor 1 cuando lo segmentado por el algoritmo coincida completamente con la OCP marcada por los especialistas [69] [70]. En la Ilustración 28 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos. En el caso de Snake se evidencia la superioridad de esta medida marcando una media de 0.38 y desviación 0.06 a diferencia del algoritmo TCH_CA que alcanza un valor medio de 0.02.

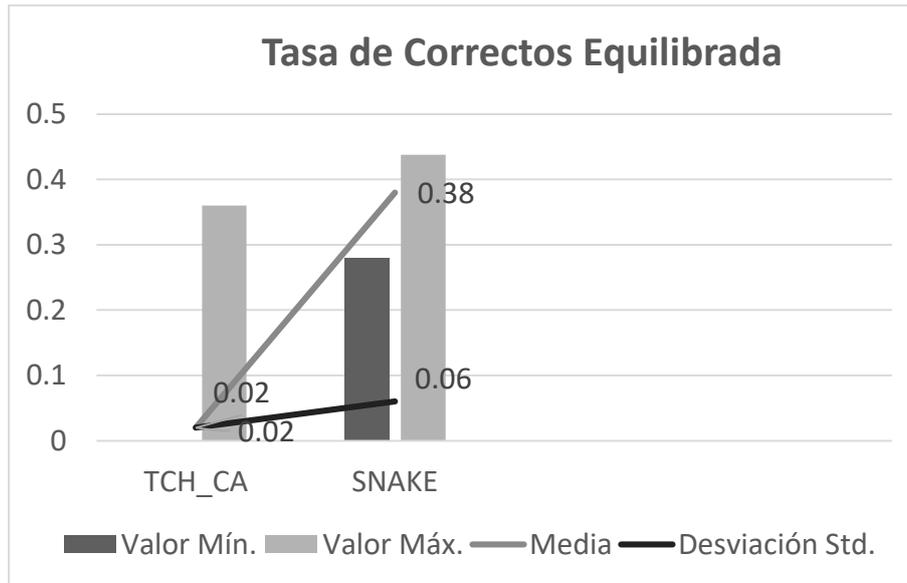


Ilustración 28. Resultados de la métrica Tasa de Correctos Equilibrada en los dos algoritmos

Índice de Sokal-Sneath: es una de las medidas estadísticas del error tomada en cuenta. Esta penaliza las regiones que fueron mal identificadas (FN y FP). Cuando toma valor 1, significa que todas las regiones con opacidad fueron correctamente segmentadas [71] [72]. En la Ilustración 29 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos. En el caso del algoritmo Snake tiene una marcada superioridad en esta medida marcando una media de 0.92 a diferencia del TCH_CA que alcanza un valor medio de 0.14.

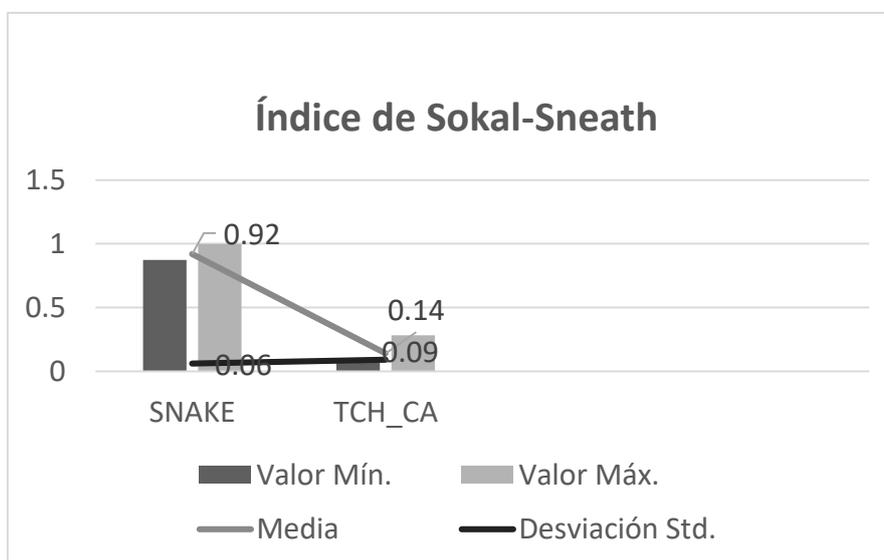


Ilustración 29. Resultados de la métrica del Índice de Sokal-Sneath en los dos algoritmos

Índice de Rogers-Tanimoto: es otra de las métricas calculadas. Esta tomará valor 1 cuando lo segmentado por el algoritmo coincida completamente con la OCP marcada por los especialistas [73] [74]. Los resultados de esta métrica (ver en la Ilustración 30) son bastante parecidos, aunque sobresale que la media del algoritmo Snake es aproximadamente 1. Significando esto un alto nivel de

coincidencia entre las imágenes de plantilla y las resultantes del algoritmo, aunque los resultados de esta métrica en el algoritmo TCH_CA también es elevado obteniendo un valor medio de 0.91.

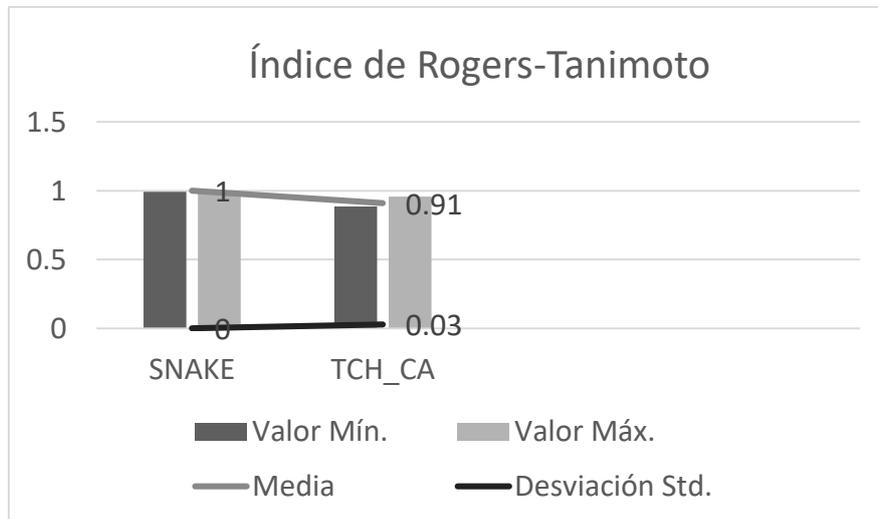


Ilustración 30. Resultados de la métrica del Índice de Rogers-Tanimoto en los dos algoritmos

Índice de Yule: cuando toma valor 2, significa que todas las regiones con opacidad fueron correctamente segmentadas [75] [76] En la Ilustración 31 se pueden apreciar los resultados de esta medida sobre los algoritmos propuestos (TCH_CA y Snake). En el caso de Snake tiene superioridad en esta medida marcando una media de 1.98 a diferencia del TCH_CA que alcanza un valor medio de 1.95.

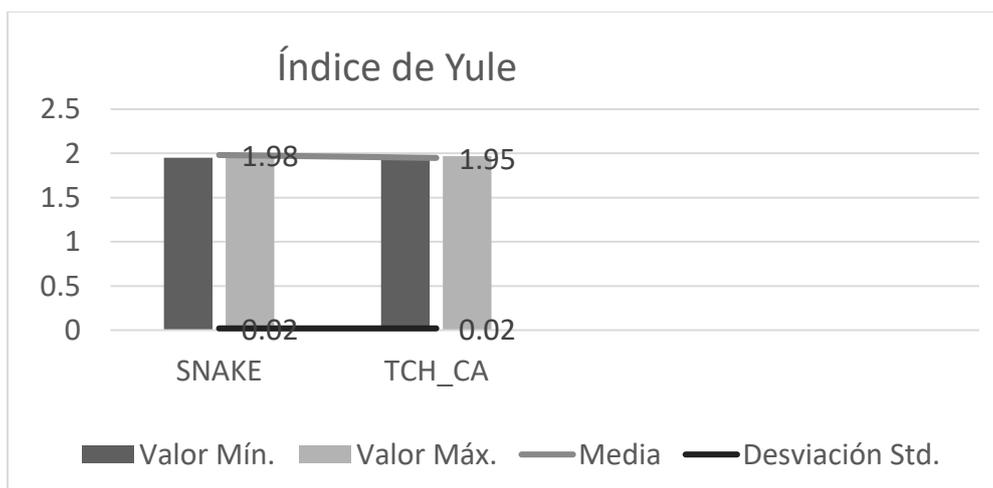


Ilustración 31. Resultados de la métrica del Índice de Yule en los dos algoritmos

Se analizaron distintas métricas estadísticas del error (Desajuste de Bordes, Tasa de Correctos Equilibrada, Índice de Sokal-Sneath, Índice de Rogers-Tanimoto, Índice de Yule). En las 5 métricas de calidad de la segmentación calculadas, el algoritmo Snake fue el de mejores resultados medios y

menor desviación en comparación con el algoritmo TCH_CA, esto se puede apreciar mejor en la Tabla 13, donde se muestran los resultados medios por cada una de las métricas calculadas y la desviación típica.

Tabla 12. Valores medios y desviaciones por cada métrica mejor resultado marcado en gris y peor gris oscuro

| Métrica | Mejor Resultado | SNAKE | | TCH_CA | |
|-------------------------------|-----------------|-------|-------|--------|-------|
| | | Media | Desv. | Media | Desv. |
| Desajuste de Bordes | 1 | 0.06 | 0.06 | 3.91 | 1.96 |
| Tasa de Correctos Equilibrada | 1 | 0.38 | 0.06 | 0.02 | 0.02 |
| Índice de Sokal-Sneath | 1 | 0.92 | 0.06 | 0.14 | 0.09 |
| Índice de Rogers-Tanimoto | 1 | 1 | 0 | 0.91 | 0.03 |
| Índice de Yule | 2 | 1.98 | 0.02 | 1.95 | 0.02 |

Estos resultados deben estar relacionados con el enfoque de los algoritmos, debido a que ambos buscan los bordes de la capsulorrexis para encontrar la OCP.

3.2.4 Análisis Estadístico

Con el objetivo de comparar el desempeño de los algoritmos y de corroborar lo expresado anteriormente se procede a realizar un análisis estadístico. Para esto se selecciona el uso de pruebas no paramétricas debido al tamaño de la muestra (solo 28 imágenes). Se escoge usar el análisis de dos vías de Friedman y como prueba *post-hoc* la prueba de rango de Wilcoxon por pares, ajustándose el nivel de significancia con una corrección de Bonferroni. Esto se aplicó sobre la variable Índice de Yule como medida estadística del error cometido en la segmentación, sobre la variable Desajuste de Bordes como medida de localización. Para realizar todos los cálculos se empleó el software IBM SPSS Statistics 21.

Tabla 13. Test de Friedman sobre los resultados de las variables Índice de Yule y Desajuste de Bordes

| # | Hipótesis Nula | Test | Sig. (p) | Decisión | Variable |
|---|--|----------|----------|-----------------------------|---------------------|
| 1 | Las distribuciones de los algoritmos son las mismas. | Friedman | < 0.04 | Rechazar la hipótesis nula. | Índice de Yule |
| 2 | Las distribuciones de los algoritmos son las mismas. | Friedman | < 0.02 | Rechazar la hipótesis nula. | Desajuste de Bordes |

Como se muestra en la Tabla 14 el valor de precisión (p) es menor que 0.05, por lo que se puede confirmar que existen diferencias significativas (en el Índice de Yule y Desajuste de Bordes) entre los dos algoritmos analizados. Es decir, los resultados de los algoritmos en las variables analizadas cambiaron significativamente.

Se desarrolló entonces una prueba post-hoc de rangos con signos de Wilcoxon con el objetivo de comparar dos a dos las muestras relacionadas (algoritmos de segmentación) y determinar entre qué resultados se presentaron las diferencias significativas. Se usó la corrección de Bonferroni para evitar cometer un error de Tipo I (rechazar la hipótesis nula cuando debería aceptarse). El nuevo nivel de significación que se tomó por tanto fue de 0,025 ($0.05/2=0.025$).

Cuando analizamos los resultados de los algoritmos probados en la investigación, podemos observar en las Tablas 15 y 16 que existen diferencias significativas entre los algoritmos en las variables estudiadas (en el Índice de Yule y Desajuste de Bordes), rechazando en todos los casos la hipótesis nula. Dado este resultado se toma la media como medida para evaluar el rendimiento del algoritmo en las métricas aplicadas. Por tanto, se asume que el algoritmo de segmentación de OCP que mejores resultados brinda, dado los valores de las medias, es Snake.

Tabla 14. Prueba por pares de Wilcoxon sobre los resultados de la variable en el Índice de Yule

| # | Hipótesis Nula | Test | Sig. (p) | Decisión | Variable |
|---|---|----------|----------|-----------------------------|----------------------|
| 1 | La mediana de las diferencias entre los algoritmos Snake y TCH_CA es igual a 0. | Wilcoxon | .004 | Rechazar la hipótesis nula. | en el Índice de Yule |

Tabla 15. Prueba por pares de Wilcoxon sobre los resultados de la variable Desajuste de Bordes

| # | Hipótesis Nula | Test | Sig. (p) | Decisión | Variable |
|---|---|----------|----------|-----------------------------|---------------------|
| 1 | La mediana de las diferencias entre los algoritmos SNAKE y TCH_CA es igual a 0. | Wilcoxon | .006 | Rechazar la hipótesis nula. | Desajuste de Bordes |

3.3 Conclusiones parciales

- 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, 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 aplicaron las métricas Desajuste de Borde, Tasa de Correctos Equilibrada, Índice de Sokal-Sneath, Índice de Rogers-Tanimoto e Índice de Yule, demostrando que existen diferencias significativas entre los resultados de los algoritmos de segmentación, sobresaliendo la superioridad del algoritmo Snake en cuanto al valor medio y a la desviación estándar respecto al algoritmo TCH_CA.

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:

- Para la identificación de la OCP en imágenes del Pentacam se propone en esta investigación la implementación del algoritmo Snake ya que tiene como ventaja minimizar la función de energía seleccionando las nuevas posiciones de los snaxel tomando valores entre 0 y 1 lo cual puede restringir o liberar los movimientos de cada snaxel alrededor de la estructura objetivo a segmentar, ajustándose mejor a su contorno.
- En el algoritmo propuesto, los snaxel se adicionan automáticamente pintando un círculo para determinar la capsulorrexís siendo OCP todo lo que se encuentre dentro de ella, con lo cual se reduce en gran medida la cantidad de cálculos a realizar, obteniendo mejores resultados que el algoritmo TCH_CA.
- Se aplicaron las métricas Desajuste de Borde, Tasa de Correctos Equilibrada, Índice de Sokal-Sneath, Índice de Rogers-Tanimoto y Índice de Yule demostrando que existen diferencias significativas entre los resultados de los algoritmos.

Recomendaciones

Se recomienda para versiones futuras:

- Implementar otro algoritmo de contornos activos para hacer una comparación más detallada en cuanto a la detección de la capsulorrexis e incluir otras métricas estadísticas para la validación de la segmentación.
- Usar como entrada del algoritmo Snake la transformada circular de Hough.

Referencias bibliográficas

- [1] Acosta R, Hoffmeister L, Roman R, Comas M, Castilla M, Castells X, «Revisión sistemática de estudios poblacionales de prevalencia de cataratas.,» vol. 81, 2006.
- [2] J. R. H. S. Y. C. G. A. G. F. Z. V. R. E. P. C. Iván Hernández López, «Estrategias de prevención de la opacidad de la cápsula posterior,» *Revista Cubana de Oftalmología*, vol. 23, 2010.
- [3] W. B. P. B. T. S. Oliver Findl, «Intervenciones para la prevención de la opacificación de la cápsula posterior,» nº 2, 2010.
- [4] D. P. S. V. L. L. J. R. M. Valentín Tinguaro Díaz Alemán, «Anillos de tensión capsular. Nuestra experiencia,» *Sociedad Canaria de Oftalmología*, nº 16, 2005.
- [5] B. SA, «Quantification of posterior capsular opacification in digital images after cataract surgery,» vol. 3979, 2000.
- [6] W. B. Oliver Findl, «Comparison of 4 methods for quantifying posterior capsule opacification,» *Journal of Cataract & Refractive Surgery*, vol. 29, 2003.
- [7] N. P. B. D. Tariq M Aslan, «Assessment of systems of analyzing PCO,» *Journal of Cataract and Refractive Surgery*, vol. 31, 2005.
- [8] D. D. D. D. S Friedman, «Digital image capture and automated analysis of posterior capsular opacification,» *Investigative Ophthalmology & Visual Science*, vol. 40, nº 8, 1999.
- [9] O. F. Wolf Buehl, «Reproducibility of standardized retroillumination photography for quantification of posterior capsule opacification,» *Journal of Cataract & Refractive Surgery*, vol. 28, nº 2, 2002.
- [1 G. U. G. U. M. S. M. B. H. E. V. Manfred R Tetz, «Photographic image analysis system of posterior 0] capsule opacification,» *Journal of Cataract & Refractive Surgery*, vol. 23, nº 10, 1997.
- [1 R. P. R. L. I. Álvarez Cancio Michel, «DISEÑO DE UN SISTEMA BASADO EN CASOS PARA LA 1] IDENTIFICACIÓN DE OPACIDAD MEDIANTE EL PENTACAM,» 2014.
- [1 A. H. B. R. P. I. H. L. Michel Alvarez Cancio, «PANDOC: SOFTWARE PARA LA CUANTIFICACIÓN 2] OBJETIVA DE LA OPACIDAD DE LA CÁPSULA POSTERIOR MEDIANTE TOMOGRAMAS SCHEIMPFLUG DEL PENTACAM,» 2013.
- [1 I. H. López, «Estrategias de prevención de la opacidad de la cápsula posterior,» vol. 23, nº 1, 2010. 3]

- [1 M. R. Moreno, «Desarrollo de un sistema basado en casos para la identificación de opacidad de
4] la cápsula posterior en el software PANDOC,» Junio del 2015.
- [1 R. C. d. I. O. y. I. H. L. Michel Alvarez Cancio, «INFORMÁTICA 2016,» 2016.
5]
- [1 P. L. M. S. y. M. A. C. Carlos Andrés Alonso Peña, «Sistema de cuantificación de opacidad capsular
6] mediante Componentes Conexas,» 2016.
- [1 C. A. A. Peña, «Sistema de cuantificación de opacidad capsular mediante componentes conexas,»
7] Facultad 3, UCI. La Habana, 2016.
- [1 M. Á. C. y. R. C. d. I. O. Erlis Paula Vida, « Algoritmo para la identificación de la opacidad de la
8] cápsula posterior en imágenes provenientes del PENTACAM,» Junio ,2016.
- [1 M. A. Cancio, «Algoritmo para la identificación de la opacidad de la cápsula posterior en imágenes
9] provenientes del PENTACAM,» *Revista Cubana de Ciencias Informáticas*, vol. 11, nº págs. 153-
167. ISSN:2227., Enero-Marzo de 2017.
- [2 M. A. Cancio, «PANDOC: Sistema basado en casos para la cuantificación objetiva de la opacidad
0] capsular de un paciente operado de catarata,» 2014.
- [2 I. H. López, «Cuantificación objetiva de la opacidad de la cápsula posteriormediante tomogramas
1] Scheimpflug del Pentacam,» *Revista Cubana de Oftalmología*, vol. 24, nº 2, 2011.
- [2 P. N. Aslam TM, «A freely accessible, evidence based, objective system of analysis of posterior
2] capasular opacification; evidence for its validity and reliability,» vol. 5, nº 9, 2005.
- [2 R. J. D Grewall, «Pentacam tomograms: A Novel Method for Quantification of Poste-,» vol. 49, nº
3] 5, 2008.
- [2 R. y. E. W. R. C. Gonzalez, «Digital Image Processing,» *New Jersey: Prentice Hall*, 2002.
4]
- [2 A. y. C. C. V. Frangi Caregnato, La segmentación de imágenes el método de los contornos activos
5] geométricos, Fundación General de la Universidad Autónoma de Madrid, 2006.
- [2 A. González Zaldivar, «Segmentación de Imágenes Médicas, mediante Técnicas de Contornos y
6] Técnicas de Regiones,» 2010.
- [2 B. y. B. D. Preim, «Visualization in Medicine Theory, Algorithms, and Applications,» *Elseiver*, 2007.
7]

- [2 O. P. y. K. N. R. Barzaga, Reconstrucción Tridimensional de Modelos anatómicos a partir de
8] imágenes médicas digitales, Habana , 2008.
- [2 E. Coto, «Métodos de Segmentación de Imágenes Médicas II,» 2017.
9]
- [3 A. M. Chaple, Módulo de Segmentación de Imágenes Médicas Digitales mediante Técnicas de
0] Watershed, Habana , 2013.
- [3 G. Bebis. [En línea]. Available:
1] <http://www.cse.unr.edu/bebis/CS791E/Notes/DeformableContours.pdf>.
- [3 C. L. y. Abierto), «Epistemowikia Revista Hiperenciclopedia de Divulgación del Saber,» [En línea].
2] Available:
http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Contorno_activo_borroso.
- [3 G. P. L. P. M. A. y. N. M. J. J. Cisneros Sanchez, «Procesamiento de imágenes,» 2014.
3]
- [3 O. Tinoco Gómez, P. P. Rosales López y . J. Salas Bacalla, «Criterios de selección de
4] metodologías de desarrollo de software,» Revista de la Facultad de Ingeniería Industrial, 2010.
- [3 R. G. Figueroa, C. J. Solís y A. A. Cabrera, «Metodologías tradicionales vs metodologías ágiles.,»
5] Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación..
- [3 R. S. Pressman, Software Engineering: A Practitioner´s Approach, 2010.
6]
- [3 H. Romero, «SlideShare,» LinkedIn Corporation, 2009. [En línea]. Available:
7] <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>. [Último acceso: 06 febrero
2015].
- [3 R. S. Pressman, Software Engineering: A Practitioner's Approach., 2010.
8]
- [3 M. A. Sánchez Mendoza, «Metodologías del desarrollo de software,» 2004.
9]
- [4 J. L. C. Valdés, «eumed.net,» [En línea]. Available: [http://www.eumed.net/tesis-
0\] doctorales/2014/jlcv/software.htm](http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm). [Último acceso: 06 febrero 2015].
- [4 Oracle, «Java,» [En línea]. Available: [http://www.java.com/es/about/..](http://www.java.com/es/about/) [Último acceso: 14 1 2015].
1]

- [4 D. A. F. C. Y. N. W. d. V. Dra. María Belén Castañeda, Procesamiento de datos y análisis estadísticos utilizando SPSS, Porto Alegre, Brasil, 2010, pp. 13-164.
- [4 «Visual Paradigm,» [En línea]. Available: <http://www.visual-paradigm.com/>. [Último acceso: 17 enero 2015].
- [4 P. N. Baeza, «Visual Paradigm DB Visual ARCHITECT SQL,» [En línea]. Available: <http://www.docstoc.com/docs/96492173/Visual-Paradigm-Studio>. [Último acceso: 7 febrero 2015].
- [4 «Ayuda de Rational Unified Process».
- 5]
- [4 A. Stellman y J. Greene, «Applied Software Project Management,» nº 113, 2005.
- 6]
- [4 K. Beck, «Una explicación de la programación extrema: aceptar el cambio,» 2002.
- 7]
- [4 J. Joskowicz, Reglas y prácticas en eXtreme Programming., 2008.
- 8]
- [4 E. d. codificación, «Estándares de codificación,» [En línea]. Available: https://docs.google.com/document/d/1rbxDfM0zsbFDNRZeM2FoXfRDbySiSt6tCdbYPA0qdzs/edit?hl=en_US#bookmark=id.6e6a203b40fe.. [Último acceso: 21 02 2018].
- 9]
- [5 D. Muhlrاد, «Patrones de diseño,» 2008.
- 0]
- [5 R. H. J. V. E. Gamma, «Elements of Design Patterns,» Addison-Wesley, 2014.
- 1]
- [5 C. Larman, UML y patrones, Introducción al análisis y diseño orientado a objetos, 2da edición, 2] 2003.
- [5 C. Larman, UML y Patrones. Introducción al análisis y diseño orientado a objetos., México, 2005.
- 3]
- [5 E. Integra., «ERP Integra,» 2010. [En línea]. Available: http://erp-integra.com/homepage/index.php?option=com_content&view=article&id=29&Itemid=51. [Último acceso: 1 marzo 2018].
- 4]
- [5 D. G. Valle y D. G. Rafael PhD, Guía de SPSS 22 para desarrollo de trabajos de investigación., 5] Malaga, 2016.

- [5 R. R. Ochoa Torrez, Analisis Estadistico en el SPSS, La Paz, 2014.
6]
- [5 C. L. J. Galindo, Técnicas de investigación en sociedad, cultura y comunicación., Pearson
7] Educación, 1998.
- [5 F. Wilcoxon, «Individual comparisons by ranking methods,» *Biometrics bulletin*, vol. 1, pp. 80-83.,
8] 1945.
- [5 M. Friedman, «The use of ranks to avoid the assumption of normality implicit in the analysis of
9] variance,» *Journal of the american statistical association*, vol. 32, pp. 675-701, 1937.
- [6 A. C. Elliott y W. A. Woodward, Statistical analysis quick reference guidebook: With SPSS
0] examples, Sage, 2007, pp. 201-203.
- [6 K. BECK, «Extreme Programming Explained.,» 1999.
1]
- [6 A. Yagüe y J. Garbojosa, «Las pruebas en metodologías ágiles y convencionales: papeles
2] diferentes.,» *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*,
vol. 3, nº 4, 2009.
- [6 G. J y a. et, «Pruebas del Sistema en Programación Externa,» 2010.
3]
- [6 I. Sommerville, Ingeniería del software. Séptima Edición., Madrid. España: Pearson Educación. S.
4] A., 2005.
- [6 D. Malfará, D. Cukerman, F. Cócaro, J. P. Cassinelli y R. Séttimo, Testing en eXtreme
5] Programming, 2006.
- [6 R. Pressman, «Ingeniería del Software un enfoque práctico.,» 2005.
6]
- [6 I. Sommerville, Ingeniería del software. Séptima Edición., Madrid. España: Pearson Educación. S.
7] A., 2005.
- [6 L. P. P. M. C. Canós José H, «Metodologías Ágiles en el Desarrollo de Software,» 2003.
8]
- [6 M. L. Goulden y a. et, Sensitivity of boreal forest carbon balance to soil thaw. *Science*, 1998, pp.
9] 214-217.

- [7 K. Asanovic y e. al, The landscape of parallel computing research: A view from berkeley. Technical
0] Report UCB/EECS-2006-183, EECS Department,, University of California, Berkeley, 2006.
- [7 M. Floris y e. al, «A generalizable definition of chemical similarity for read-across.,» *Journal of*
1] *cheminformatics*, vol. 6, nº 1, p. 39, 2014.
- [7 L. Nieddu y A. Rizzi, Metrics in symbolic data analysis. En *New Developments in Classification and*
2] *Data Analysis*, Springer, Berlin, Heidelberg, 2005, pp. 71-78.
- [7 O. Arenas y e. al, «Comparison of the Haplotypes of the major Histocompatibility complex in
3] Rat:V.A Serological,Genetic and numerical study of the RT1. A Region Antigens,» *International*
Journal of Immunogenetics, vol. 8, nº 4, pp. 307-314, 1981.
- [7 B. Desgraupes, Clustering indices. University of Paris Ouest-Lab ModalX, vol. 1, 2013, p. 34.
4]
- [7 H. Michalopoulou y C. Jacovides, Instability indices for the Cyprus area. *Meteorology and*
5] *Atmospheric Physics*, 1987, pp. 153-158..
- [7 H. Hubert y e. al, «Comparison of the diagnostic quality of body mass index, waist circumference
6] and waist-to-height ratio in screening skinfold-determined obesity among children,» *Journal of*
science and medicine in sport, vol. 12, nº 4, pp. 449-451, 2009.

Anexos

Anexo 1: Tarjetas CRC

Tabla 16. Tarjeta CRC de la clase Imagen

| Clase: Imagen | |
|--|-------------------|
| Responsabilidad | Colaboradora |
| 1. Pinta la imagen para que el usuario la visualice. | 1. Snakes_ventana |

Tabla 17. Tarjeta CRC de la clase Snakes_ventana

| Clase: Snakes_ventana | |
|---|--------------|
| Responsabilidad | Colaboradora |
| 1. Dibuja la interfaz de usuario y le permite interactuar con ella. | 1. Snakes |

Tabla 18. Tarjeta CRC de la clase Snaxel

| Clase: Snaxel | |
|--|--------------|
| Responsabilidad | Colaboradora |
| 1. Representa las coordenadas de los puntos. | 1. Imagen |

Glosario de términos

Cristalino: Medio refringente del ojo con alto poder dióptrico encargado de la acomodación (Enfoque para visión cercana o lejana).

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

Sistema Sheimpflug: Sistema de cámaras de alta definición acopladas a diferentes equipos médicos como el PENTACAM.

PANDOC: Programa analizador de opacidad capsular.

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

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.

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