

Universidad de las Ciencias Informáticas

FACULTAD 5



**Título: Segmentación de imágenes médicas
mediante el algoritmo Snakes.**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor(es): Ernesto Fuentes Gómez

Tutor(es): MSc. Osvaldo Pereira Barzaga

Co-tutor: Ing. Luis Guillermo Silva

Co-Tutor: Ing. Rubén Alcolea Núñez

Junio 2012

Declaración de Autoría

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

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

Ernesto Fuentes Gómez

Autor

M.Sc. Osvaldo Pereira Barzaga

Tutor

Ing. Luis Guillermo Silva Rojas

Co-tutor

Ing. Rubén Alcolea Núñez

Co-tutor

Datos del Contacto

Tutor: M.Sc. Osvaldo Pereira Barzaga.

Edad: 27.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: M.Sc. en Informática Aplicada.

Categoría Docente: Instructor.

E-mail: opereira@uci.cu

Graduado de la UCI, con cinco años de experiencia en el tema de la Gráfica Computacional y líder de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

Co-tutor: Ing. Luis Guillermo Silva.

Edad: 24.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ing. en Ciencias Informáticas.

Categoría Docente: Instructor Recién Graduado.

E-mail: lgsilva@uci.cu.

Graduado de la UCI, con un año de experiencia en el tema de la Gráfica Computacional y profesor de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

Co-tutor: Ing. Rubén Alcolea Núñez.

Edad: 24.

Ciudadanía: Cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ing. en Ciencias Informáticas.

Categoría Docente: Instructor Recién Graduado.

E-mail: ralcolea@uci.cu.

Graduado de la UCI, con un año de experiencia en el tema de la Gráfica Computacional y profesor de un proyecto de Realidad Virtual en la Universidad de las Ciencias Informáticas.

Dedicatoria

A mi madre:

Mi persona especial siempre tan atenta y cariñosa, que me ha permitido estar aquí hoy y cuidado hasta a veces pasar al punto de la sobreprotección pero siempre con las mejores intenciones.

A mi papá:

Mi amigo, mi guía en la vida, mi ejemplo en muchos aspectos. Siempre ha estado para mí cuidándome y enseñándome de todo, puedo decir que mucho de lo que soy se lo debo a él.

A mi hermanita:

A quien llevo siempre conmigo, con la que comparto muchas de mis cosas e intento siempre enseñarla con mi ejemplo.

A mis abuelos:

A Mamí por ser la persona más buena que conozco, siempre complaciente conmigo demostrándome su cariño y a Papí por enseñarme el valor de la familia y estar unidos, por tratarme como su otro hijo.

Agradecimientos

Quisiera agradecer a cada una de las personas que me han ayudado en la confección de este trabajo y en mi formación como hombre y profesional.

De manera particular a mi familia: a mi papá por ayudarme con la revisión del documento y entendimiento de algunos modelos matemáticos así como su presencia en mi vida para ayudarme en las buenas y en las malas, por ser mi amigo. Agradecer a mi mamá por ayudarme con la bibliografía, por su apoyo en todo momento de mi vida y el cariño que siempre puedo encontrar en ella. A mi hermanita por darme buenos momentos cada vez que estamos juntos y por las broncas también que siempre me enseñan que la quiero mucho. A Mamí y a Papí por estar siempre para mí ayudándome y apoyándome en todo.

Agradecer a todas las personas que me ayudaron directamente con la tesis a pesar de no tener por qué hacerlo, a Ulises, el Bares, el sarnón por ayudarme con algunas cosas de ingeniería y modelos matemáticos, por ser uno de mis mejores amigos desde hace 6 años y una de las personas que sabe más de mí. A Angel el brother por las ideas, discusiones, momentos y proyectos que hemos compartido, por ser mi otro mejor amigo en la UCI. A Jose Carlos por ser un gran amigo uno de los mejores que he conocido en la UCI, por poder contar con él en todo momento. A Jose Luis Simon, el Simons por ayudarme con el QTimer y demostrar ser un buen amigo. A Enelis por ayudarme con el diagrama de casos de uso, por ser mi mejor amiga de la UCI y su ayuda incondicional. A Lefebre por las ideas que me ha aportado y su ayuda en la confección del documento, al Boris por las perspectivas del documento al Faifor, a Carlitos, a Iván, a Karel Piorno, a Jose Andres el fusso, entre otros que me ayudaron, les voy a estar eternamente agradecidos.

A mi tutor Osvaldo por preocuparse, ayudarme y guiarme aunque no se encontraba en el país, de manera especial a mis co-tutores Guille y Alcolea por ayudarme en cada momento que los necesité, principalmente Alcolea por el tiempo gastado arreglando el documento y la paciencia inmensa que tiene. Al equipo de revisión de los cortes por aportarme ideas y ayudarme a limar mis errores.

Agradecer a mis amigos del edificio con los que he compartido fiestas y preocupaciones Osmay, Rohelis, el David, Ever, Iliá, Yoandy, Yoan, el Denis, Yosmany, la gente del laboratorio, las personas que he conocido en todos mis grupos desde 1er año. A mis amigos de Cienfuegos Jansel, Juanma, al Charli, al Croviet a todos por ser mis mejores amigos de siempre.

A las chicas que he conocido en la UCI y con las que he compartido momentos de todo tipo Leyanis, Yanet, Sonia, Nallia, Lilibet, Lauren y todas las que me han enseñado algo.

Finalmente agradecer a la UCI por formarme como profesional y aportarme las herramientas necesarias para aprender de todo lo que me ha interesado.

Resumen

La segmentación es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos. El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y fácil de analizar. Generalmente en estos procesos se requiere de una etapa de pre-procesamiento encargada de mejorar la calidad de las imágenes digitales para lograr mejores resultados en la posterior segmentación.

En este trabajo se emplean técnicas de segmentado y pre-procesamiento necesarios para el tratamiento y mejora de las imágenes médicas con el objetivo de mejorar la calidad y el nivel de detalle. Surge como una necesidad del proyecto Visualización Médica de la Universidad de las Ciencias Informáticas para mejorar la etapa de segmentación en el proceso de reconstrucción tridimensional (3D) a partir de imágenes médicas digitales en formato DICOM adquiridas a partir de Tomografías Computarizadas y Resonancias Magnéticas. El objetivo principal es segmentar las imágenes mejorando la calidad de las mismas y contribuyendo al diagnóstico de patologías y enfermedades.

Para el cumplimiento de este objetivo se implementó el algoritmo de segmentación semiautomático *Snakes* basado en contornos activos y el filtro de la Mediana para la reducción de ruido presente en las imágenes médicas. Como resultado se obtuvo una aplicación que les brinda a los especialistas la funcionalidad de enmarcar estructuras relevantes presentes en las imágenes médicas e intervenir en el proceso de segmentación. Además se describen los resultados obtenidos con la aplicación y las pruebas que se realizaron para validar la misma.

Palabras clave: DICOM, filtro, pre-procesamiento, segmentación, Snakes.

Índice de Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1. IMAGEN DIGITAL.....	5
1.2. VECINDAD DE UN PÍXEL.....	6
1.3. IMAGEN MÉDICA.....	7
1.4. PRE-PROCESAMIENTO.....	7
1.4.1. <i>Suavizado y reducción de ruido</i>	8
1.4.1.1. Filtro Gaussiano.....	8
1.4.1.2. Filtro Promediador.....	9
1.4.1.3. Filtro de la Mediana.....	9
1.4.2. <i>Agudización de detalles y detección de bordes</i>	9
1.4.2.1. Operador de Roberts.....	10
1.4.2.2. Operador de Sobel.....	10
1.4.2.3. Operador de Prewitt.....	12
1.4.2.4. Operador Isotrópico.....	12
1.5. SEGMENTACIÓN DE IMÁGENES.....	13
1.5.1. <i>Tipos de segmentación de imágenes</i>	14
1.5.1.1. Segmentación manual.....	14
1.5.1.2. Segmentación automática.....	14
1.5.1.3. Segmentación semiautomática.....	14
1.5.2. <i>Técnicas de segmentación de imágenes</i>	15
1.5.2.1. Segmentación basada en Umbrales.....	15
1.5.2.2. Crecimiento de Regiones.....	15
1.5.2.3. Agrupamiento.....	16
1.5.2.4. Watershed.....	17
1.5.2.5. Segmentación basada en modelos.....	18
1.5.2.6. Modelos deformables.....	18
1.5.2.7. Modelos de Contornos Activos o Snakes.....	18
1.5.3. <i>Valoración sobre los algoritmos de segmentación</i>	22
CAPÍTULO 2: SOLUCIÓN PROPUESTA	23
2.1. REDUCCIÓN DE RUIDO EN PRE-PROCESAMIENTO: FILTRO DE LA MEDIANA.....	23
2.2. SNAKES.....	24
2.3. ESPECIFICACIÓN DEL CONTORNO.....	24
2.4. FUNCIÓN DE ENERGÍA.....	25
2.5. TÉRMINOS DE LA FUNCIÓN DE ENERGÍA.....	25
2.6. DESCRIPCIÓN DEL ALGORITMO SNAKES.....	26
2.7. PSEUDOCÓDIGO DEL ALGORITMO.....	27
2.8. HERRAMIENTAS.....	28
2.9. METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	29
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	30
3.1. REGLAS DEL NEGOCIO.....	30

3.2. MODELO DEL DOMINIO	30
3.3. CAPTURA DE REQUISITOS.....	31
3.3.1. <i>Requisitos funcionales</i>	32
3.3.2. <i>Requisitos no funcionales</i>	32
3.4. MODELO DE CASOS DE USO DEL SISTEMA.....	33
3.4.1. <i>Actores del sistema</i>	33
3.4.2. <i>Diagrama de Casos de Uso del Sistema</i>	34
3.4.3. <i>Descripción de Casos de Uso del Sistema</i>	35
3.5. DIAGRAMA DE CLASES	41
3.6. DIAGRAMAS DE SECUENCIA DEL DISEÑO.....	42
3.6.1. <i>Diagrama de Secuencia del caso de uso Cargar Imagen</i>	43
3.6.2. <i>Diagrama de Secuencia del caso de uso Aplicar Filtro</i>	43
3.6.3. <i>Diagrama de Secuencia del caso de uso Gestionar Contorno</i>	44
3.6.4. <i>Diagrama de Secuencia del caso de uso Modificar Contorno</i>	45
3.6.5. <i>Diagrama de Secuencia del caso de uso Segmentar Imagen</i>	46
CAPÍTULO 4: IMPLEMENTACIÓN Y VALIDACIÓN DE LOS RESULTADOS	47
4.1. IMPLEMENTACIÓN	47
4.1.1. <i>Diagrama de Componentes</i>	47
4.2. VALIDACIÓN DE LOS RESULTADOS.....	48
4.2.1. <i>Datos de Entrada</i>	48
CONCLUSIONES	57
RECOMENDACIONES	58
BIBLIOGRAFÍA.....	59
GLOSARIO DE TÉRMINOS.....	62

Índice de Figuras

FIG. 1 REPRESENTACIÓN DE UNA IMAGEN DIGITAL.....	5
FIG. 2 REPRESENTACIÓN DE UN PÍXEL Y SU VICINIDAD.....	6
FIG. 3 IZQUIERDA: RECORRIDO DEL FILTRO SOBRE LA IMAGEN DERECHA: MÁSCARA DE CONVOLUCIÓN.....	8
FIG. 4 MÁSCARAS DEL OPERADOR DE ROBERTS.....	10
FIG. 5 COEFICIENTES DEL OPERADOR DE SOBEL.....	11
FIG. 6 MÁSCARAS DEL OPERADOR DE SOBEL: MÁSCARA PARA Sx (IZQUIERDA), MÁSCARA PARA Sy (DERECHA).....	11
FIG. 7 MÁSCARAS DEL OPERADOR DE PREWITT: MÁSCARA PARA Sx (IZQUIERDA), MÁSCARA PARA Sy (DERECHA).....	12
FIG. 8 MÁSCARAS DEL OPERADOR ISOTRÓPICO: MÁSCARA PARA Sx (IZQUIERDA), MÁSCARA PARA Sy (DERECHA).....	13
FIG. 9 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.....	16
FIG. 10 (A) IMAGEN ORIGINAL (B) IMAGEN SEGMENTADA CON EL ALGORITMO K-MEANS.....	17
FIG. 11 PROCESO ITERATIVO DE AJUSTE Y SEGMENTACIÓN DEL SNAKE SOBRE UN OBJETIVO EN UNA IMAGEN.....	19
FIG. 12 REPRESENTACIÓN PARAMÉTRICA DEL CONTORNO.....	20
FIG. 13 OPERACIÓN DEL FILTRO DE LA MEDIANA.....	23
FIG. 14 IZQUIERDA: CIRCUNFERENCIA REPRESENTADA POR UN POLÍGONO DE 4 VÉRTICES. DERECHA: CIRCUNFERENCIA REPRESENTADA POR UN POLÍGONO DE 6 VÉRTICES.....	24
FIG. 15 IZQUIERDA: POLÍGONO INICIAL GENERADO. DERECHA: POLÍGONO LUEGO DE APLICADO EL ALGORITMO DE RELLENO DE VÉRTICES.....	25
FIG. 16 REPRESENTACIÓN DE LA VICINIDAD DE BÚSQUEDA DE UN <i>SNAXEL</i>	27
FIG. 17 DIAGRAMA DEL MODELO DE DOMINIO.....	31
FIG. 18 DIAGRAMA DE CASOS DE USO.....	35
FIG. 19 DIAGRAMA DE CLASES.....	42
FIG. 20 DIAGRAMA DE SECUENCIA DEL CASO DE USO CARGAR IMAGEN.....	43
FIG. 21 DIAGRAMA DE SECUENCIA DEL CASO DE USO APLICAR FILTRO.....	43
FIG. 22 DIAGRAMA DE SECUENCIA DE LA SECCIÓN ADICIONAR CONTORNO DEL CASO DE USO GESTIONAR CONTORNO.....	44
FIG. 23 DIAGRAMA DE SECUENCIA DE LA SECCIÓN ELIMINAR CONTORNO DEL CASO DE USO GESTIONAR CONTORNO.....	44
FIG. 24 DIAGRAMA DE SECUENCIA DE LA SECCIÓN CAMBIAR POSICIÓN DE VÉRTICE DEL CASO DE USO MODIFICAR CONTORNO.....	45
FIG. 25 DIAGRAMA DE SECUENCIA DE LA SECCIÓN ELIMINAR VÉRTICE DEL CASO DE USO MODIFICAR CONTORNO.....	45
FIG. 26 DIAGRAMA DE SECUENCIA DE LA SECCIÓN RELLENAR ARISTAS DEL CASO DE USO MODIFICAR CONTORNO.....	46
FIG. 27 DIAGRAMA DE SECUENCIA DEL CASO DE USO SEGMENTAR IMAGEN.....	46
FIG. 28 DIAGRAMA DE COMPONENTES.....	48
FIG. 29 SUPERIOR IZQUIERDA: INTERFAZ INICIAL. SUPERIOR DERECHA: CUADRO DE DIÁLOGO DE BÚSQUEDA DEL FICHERO DICOM A CARGAR. INFERIOR: APLICACIÓN CON LA IMAGEN CARGADA.....	50
FIG. 30 IZQUIERDA: IMAGEN INICIAL. DERECHA: IMAGEN DESPUÉS APLICADO EL FILTRO DE LA MEDIANA.....	51
FIG. 31 IZQUIERDA: IMAGEN MOSTRANDO EL CONTORNO CREADO SOBRE LA ESTRUCTURA A SEGMENTAR. DERECHA: IMAGEN MOSTRANDO EL CONTORNO UNA VEZ REALIZADO EL ALGORITMO DE RELLENO DE VÉRTICES.....	52
FIG. 32 IZQUIERDA: CONTORNO INICIAL CON POCOS VÉRTICES SOBRE EL ÁREA QUE SE DESEA SEGMENTAR. DERECHA: RESULTADO DE LA SEGMENTACIÓN.....	53
FIG. 33 IZQUIERDA: CONTORNO INICIAL AL QUE SE LE APLICÓ EL ALGORITMO DE RELLENO DE VÉRTICES SOBRE EL ÁREA QUE SE DESEA SEGMENTAR. DERECHA: RESULTADO DE LA SEGMENTACIÓN.....	53
FIG. 34 IZQUIERDA: CONTORNO INICIAL SOBRE EL ÁREA DEL CEREBRO. DERECHA: RESULTADO DE LA SEGMENTACIÓN.....	54
FIG. 35 IZQUIERDA: CONTORNO INICIAL SOBRE EL ÁREA QUE SE DESEA SEGMENTAR. DERECHA: RESULTADO DE LA SEGMENTACIÓN.....	55
FIG. 36 IZQUIERDA: CONTORNO INICIAL SOBRE EL ÁREA QUE SE DESEA SEGMENTAR. DERECHA: RESULTADO DE LA SEGMENTACIÓN.....	55

Índice de Tablas

TABLA 1 ACTORES DEL SISTEMA	34
TABLA 2 DESCRIPCIÓN DEL CASO DE USO CARGAR IMAGEN.	35
TABLA 3 DESCRIPCIÓN DEL CASO DE USO FILTRAR IMAGEN.....	36
TABLA 4 DESCRIPCIÓN DEL CASO DE USO GESTIONAR CONTORNO.	37
TABLA 5 DESCRIPCIÓN DE LA SECCIÓN INSERTAR CONTORNO, PERTENECIENTE AL CASO DE USO GESTIONAR CONTORNO.	37
TABLA 6 DESCRIPCIÓN DE LA SECCIÓN ELIMINAR CONTORNO, PERTENECIENTE AL CASO DE USO GESTIONAR CONTORNO.....	38
TABLA 7 DESCRIPCIÓN DEL CASO DE USO MODIFICAR CONTORNO.	38
TABLA 8 DESCRIPCIÓN DE LA SECCIÓN CAMBIAR POSICIÓN DE VÉRTICE, PERTENECIENTE AL CASO DE USO GESTIONAR CONTORNO.	39
TABLA 9 DESCRIPCIÓN DE LA SECCIÓN ELIMINAR VÉRTICE, PERTENECIENTE AL CASO DE USO GESTIONAR CONTORNO.....	40
TABLA 10 DESCRIPCIÓN DE LA SECCIÓN RESETEAR, PERTENECIENTE AL CASO DE USO GESTIONAR CONTORNO.	40
TABLA 11 DESCRIPCIÓN DEL CASO DE USO SEGMENTAR.....	40
TABLA 12 CASO DE PRUEBA #1.	49
TABLA 13 CASO DE PRUEBA #2.	49

Introducción

Las imágenes digitales se utilizan en varias esferas de la ciencia, teniendo una relevancia importante en la medicina. Mediante las diversas modalidades de adquisición de imágenes como Tomografía Axial Computarizada (TAC) y Resonancia Magnética (RM), se generan imágenes de zonas del cuerpo del paciente de forma no invasiva con importante información clínica. Estas imágenes se denominan imágenes médicas y proveen un medio efectivo de representar la anatomía de un paciente para su posterior uso por los especialistas.

Con el incremento en tamaño y número de imágenes médicas que se pueden obtener de un paciente a partir de un estudio realizado por alguna modalidad de adquisición, se ha hecho necesario el uso del computador para facilitar el procesamiento y análisis de las mismas [1]. Mediante el computador el médico analiza las imágenes y procesa la información más importante, esto le posibilita incorporar datos útiles para su diagnóstico sobre las patologías presentes en las imágenes.

Las imágenes médicas digitales están en escala de grises, debido a que la fuente de radiación que se utiliza para obtenerlas no cae dentro del espectro visible al ojo humano. Durante la adquisición de las imágenes se introduce ruido, el cual es provocado entre otros factores por defectos técnicos de los equipos o por implantes metálicos que tienen los pacientes, como por ejemplo, los empastes dentales. El ruido introducido en la imagen dificulta su análisis por parte de los especialistas. Por esta razón se hace necesario procesar estas imágenes para mejorar su calidad y realzar detalles importantes para el especialista. Una técnica muy utilizada con este propósito es la segmentación de las imágenes médicas.

Los algoritmos de segmentación permiten identificar y delinear con precisión los bordes de las estructuras anatómicas relevantes en las imágenes médicas. 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. 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.

Numerosas aplicaciones informáticas para el tratamiento de imágenes digitales han surgido en nuestro país y algunas de estas especializadas en la reconstrucción de modelos anatómicos tridimensionales (3D). En el Centro de Informática Industrial (CEDIN) de la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI), el proyecto de Visualización Médica (Vismedic) desarrolla aplicaciones que emplean técnicas de visualización médica 3D de modelos anatómicos, reconstruidos a partir de imágenes médicas digitales. Para la realización de estos procesos es necesaria la segmentación de estas imágenes, para su posterior uso en el análisis detallado de las estructuras anatómicas presentes. En el proyecto se han implementado algunas técnicas de segmentación como el Crecimiento de Regiones y el *Watershed*, que aunque obtienen resultados aceptables presentan la deficiencia de que no le permiten al especialista seleccionar la región de interés en la imagen que desea analizar para su posterior delineación.

Dada la **situación problemática** planteada anteriormente se define como **problema científico**: ¿Cómo detectar los contornos de las estructuras anatómicas presentes en imágenes médicas digitales en formato DICOM?

Se define como **objeto de estudio**: la segmentación de imágenes digitales. Para darle solución al problema formulado se plantea como **objetivo general**: elaborar un módulo de segmentación de imágenes médicas digitales mediante el algoritmo *Snakes*, que permita identificar los contornos de las estructuras anatómicas. El **campo de acción** queda definido como: la segmentación de imágenes médicas digitales en formato DICOM. Se toma como **idea a defender**: con la implementación del módulo de segmentación propuesto se logrará que el usuario sea capaz de seleccionar las estructuras anatómicas de interés durante el proceso de segmentación de imágenes médicas digitales para su posterior estudio, análisis y visualización.

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

- Elaboración del marco teórico a partir de la bibliografía actual referente al tema.
- Selección del método de pre-procesamiento más eficaz para mejorar la calidad de las imágenes médicas.
- Definición de las clases que componen la arquitectura del módulo.
- Selección de las herramientas y metodologías de desarrollo para la elaboración del módulo.

- Implementación del algoritmo de segmentación basado en contornos activos o *Snakes*.
- Validación de los resultados y el funcionamiento del módulo.

En la realización de este trabajo se utilizan algunos métodos científicos de investigación:

- Métodos Teóricos:
 - Análisis–Síntesis: Para el despliegue de las principales ideas sobre las técnicas de segmentación de una imagen digital y su concreción en las principales ideas generales.
 - Modelación: En la adquisición de los datos necesarios para crear abstracciones del módulo propuesto con el objetivo de representar la realidad.
- Métodos Empíricos:
 - Consulta de fuentes de información: Consulta de todo tipo de fuente de investigación para la actualización del conocimiento sobre el tema.
 - Observación: Para apreciar el impacto del proceso de segmentación propuesto en la calidad de la visualización y la interpretación de las estructuras segmentadas.
 - Medición: Para la obtención de información numérica acertada de magnitudes medibles y usarlas en la comparación de resultados.
 - Pruebas: Para validar los resultados obtenidos a partir de la investigación realizada.

El presente trabajo está conformado por cuatro capítulos, los cuales se sintetizan a continuación con el objetivo de mostrar una estructura general sobre los temas abordados.

- **Capítulo 1. Fundamentación Teórica:** En este capítulo se mencionan los diferentes métodos de pre-procesamiento y segmentación utilizados en imágenes digitales, se caracterizan estos métodos y se mencionan sus ventajas y desventajas, conformando de esta forma el marco teórico del trabajo.
- **Capítulo 2. Solución Propuesta:** En este capítulo se presenta la descripción de la solución propuesta para la segmentación mediante *Snakes*, así como el pre-procesamiento y algunas mejoras y detalles incorporados a la solución. También se explicarán las herramientas y metodologías de desarrollo utilizadas para la elaboración de la solución propuesta.
- **Capítulo 3. Análisis y Diseño del Sistema:** En este capítulo se definen las reglas del negocio, el modelo de dominio y se expone la captura de requisitos. Se elabora el Diagrama de Casos de Uso y se describen textualmente cada caso de uso. Finalmente se elabora el diagrama de clases y los diagramas de secuencia del diseño relacionados con la aplicación.

- **Capítulo 4. Implementación y Validación de los Resultados:** En este capítulo se abordan los temas de la implementación del módulo basados en todo el trabajo realizado durante la investigación. Se hace un análisis de los resultados obtenidos en cuanto a la calidad de la segmentación y según los diferentes parámetros y situaciones a las que se puede enfrentar el algoritmo *Snakes*, lo que permite cumplir con los objetivos propuestos.
- **Glosario de términos:** Se presenta un Glosario de Términos con el objetivo de facilitar la comprensión del lenguaje utilizado en la investigación.

Capítulo 1: Fundamentación Teórica

Este capítulo abarca los principales elementos teóricos necesarios para la comprensión de algunas técnicas y algoritmos que se utilizan a nivel mundial para la segmentación, mejora de calidad de las imágenes así como su visualización. Además se explica el funcionamiento de estas técnicas haciendo énfasis en sus puntos fuertes y débiles en cada caso.

1.1. Imagen digital

Una imagen se puede definir matemáticamente como una función bidimensional $f(x,y)$ que representa la intensidad o el nivel de gris del píxel con coordenadas espaciales (x,y) dentro de la imagen [2], como se puede apreciar en la Fig. 1.

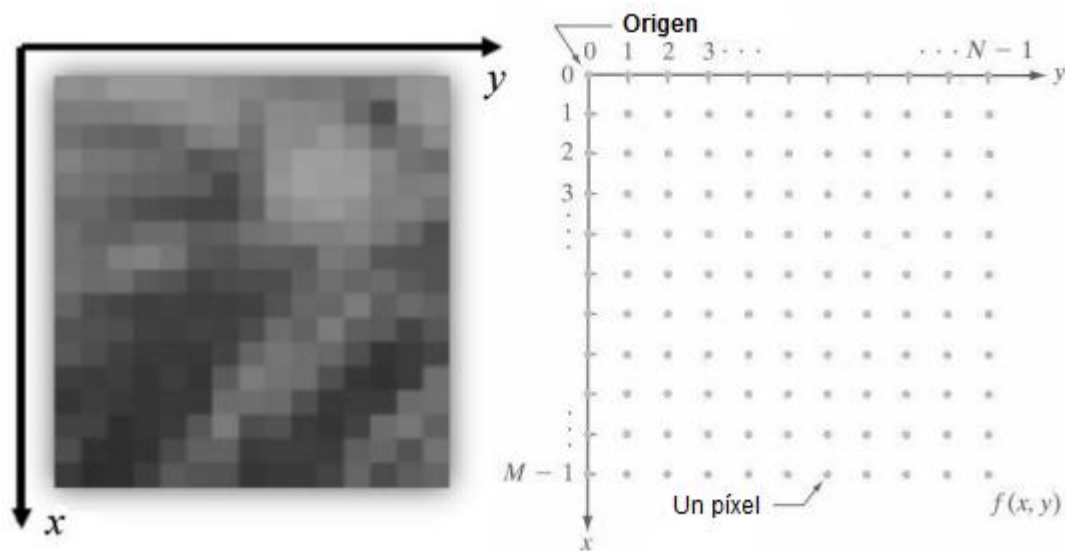


Fig. 1 Representación de una imagen digital.

Cuando f y los valores de x,y son cantidades finitas y discretas, la imagen se denomina imagen digital [2]. Una imagen digital se compone de un número finito de elementos con un lugar y valor específicos. Estos elementos se conocen como píxeles [2]. Una imagen digital puede ser convenientemente representada por una matriz $f(x,y)$ de tamaño $M \times N$ de la forma:

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

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 [3]. 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 $0 \leq i \leq N - 1, 0 \leq j \leq M - 1$ [2].

1.2. Vecindad de un píxel

Un píxel p en la coordenada (x, y) tiene cuatro vecinos: dos horizontales con coordenadas $(x, y + 1)$ y $(x, y - 1)$ y dos verticales con coordenadas $(x + 1, y)$ y $(x - 1, y)$.

Este grupo de píxeles, llamados los 4-vecinos de p , es denotado por $N_4(p)$. Los cuatro vecinos diagonales de p tienen como coordenadas:

$$(x + 1, y + 1), (x - 1, y + 1), (x + 1, y - 1), (x - 1, y - 1)$$

y están denotados por $N_D(p)$. Estos puntos, juntos con los 4-vecinos, son llamados los 8-vecinos de p , denotado por $N_8(p)$. En la Fig. 2 se puede ver una representación gráfica de un píxel y sus vecinos. Cada píxel está a una unidad de distancia desde (x, y) , por lo que algunos de los vecinos de p pueden encontrarse fuera de la imagen digital si (x, y) está en el borde de la imagen [2].

$x-1, y-1$	$x-1, y$	$x-1, y+1$
$x, y-1$	x, y	$x, y+1$
$x+1, y-1$	$x+1, y$	$x+1, y+1$

Fig. 2 Representación de un píxel y su vecindad.

1.3. 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 [4].

Entre las modalidades de adquisición de imágenes médicas se destacan las de Rayos X y la 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 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 [4].

1.4. Pre-procesamiento

Durante el proceso de digitalización de una imagen, en ocasiones se introduce ruido o se degrada la imagen, lo que provoca pérdida de calidad. Las imágenes producidas por los equipos médicos se pueden ver afectadas por la respiración del paciente o por implantes metálicos presentes en sus dientes, por citar algunos ejemplos. Por ello es crucial mejorar la calidad de las imágenes antes de hacer cualquier análisis sobre ellas. La reducción de ruido en las imágenes conduce a obtener áreas más homogéneas que podrían ser delineadas con menor esfuerzo, eliminando información que no se considera relevante, aunque por otra parte puede comprometer la detección de pequeños elementos importantes [5]. La mejora de la calidad de la imagen generalmente se realiza en la etapa de pre-procesamiento, etapa que antecede al proceso de segmentación. Del tipo de análisis que se vaya a realizar sobre la imagen dependerá la técnica o algoritmo de pre-procesamiento que se aplique. Entre las técnicas más utilizadas se encuentran los algoritmos de suavizado [2], los algoritmos de reducción de ruido [2], los algoritmos de detección de bordes [2] y las técnicas híbridas [2], que combinan algunas de las técnicas antes mencionadas.

Las técnicas para mejorar la imagen se clasifican principalmente en dos grandes categorías: métodos del dominio espacial y métodos del dominio de frecuencia. El término dominio espacial se refiere al plano de la imagen en sí mismo y están basados en la manipulación directa de los píxeles. Las técnicas de procesamiento de dominio de frecuencia están basadas en modificar la transformada de Fourier en una imagen [2].

El tratamiento y mejora de la imagen en el dominio espacial se basa fundamentalmente en el uso de filtros (máscaras / plantillas de ventana como también se le conoce). Básicamente un filtro o máscara de convolución, es un pequeño arreglo 2D, como el de la Fig. 3, en el cual los valores de los coeficientes de la máscara determinan la naturaleza del proceso, ya sea suavizado, agudización de detalles u otra mejora en la imagen. El procedimiento consiste principalmente en mover la máscara punto a punto sobre la imagen. Estas técnicas de mejoramiento son llamadas filtros o procesamiento de máscara[6].

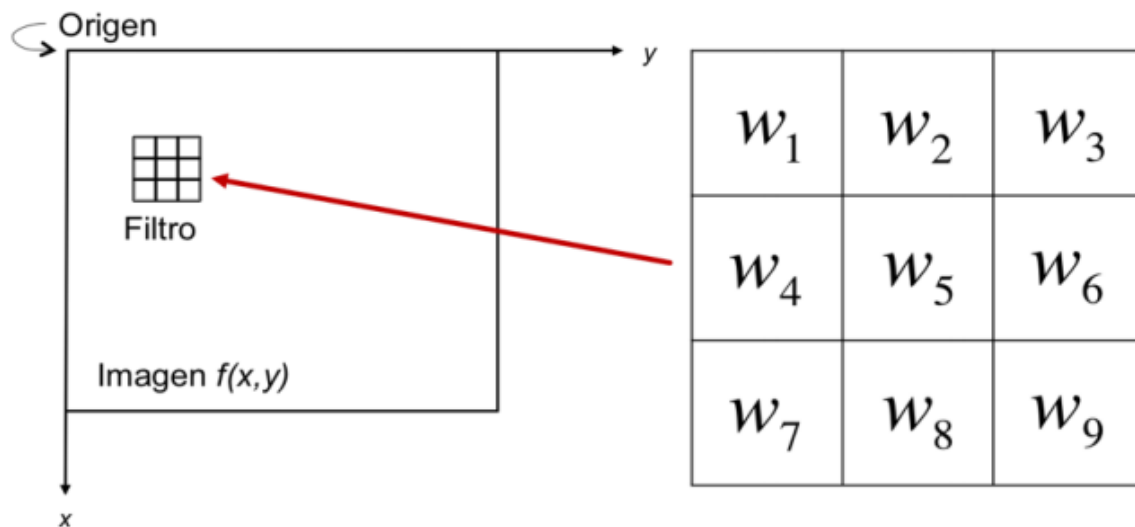


Fig. 3 Izquierda: Recorrido del filtro sobre la imagen Derecha: Máscara de convolución.

1.4.1. Suavizado y reducción de ruido

El ruido digital, producido por el dispositivo de entrada, es la variación aleatoria (que no se corresponde con la realidad) del brillo o el color en las imágenes digitales [7]. Dentro de los filtros que se aplican a las imágenes con el fin de reducir la cantidad de ruido o suavizarlas en pos de lo que se quiera realizar con ellas, se encuentran el filtro Gaussiano, el filtro Promediador, el filtro de la Mediana, entre otros.

1.4.1.1. Filtro Gaussiano

Este filtro implementa máscaras que intentan imitar una función gaussiana la forma $G(x,y) = e^{-(x+y)^2/2\sigma^2}$ donde (x, y) son las coordenadas de la imagen y sigma (σ) una desviación estándar de la probabilidad de distribución asociada. Este es el único parámetro del filtro Gaussiano y es proporcional al área de los vecinos más cercanos en que opera este filtro. Los píxeles más lejanos al centro del operador tienen

menor influencia y aquellos con una diferencia mayor de tres valores de sigma a partir del centro tienen una influencia insignificante. Este filtro tiene el inconveniente de que además de remover el ruido, empaña la imagen ocasionando pérdida de los detalles más finos [8].

1.4.1.2. Filtro Promediador

Este tipo de filtro se basa en reemplazar cada píxel que se analiza por el promedio del nivel de gris de la vecindad. El resultado es una imagen con reducción en las transiciones agudas de los niveles de gris donde uno de los usos más significativos es la reducción de detalles irrelevantes o regiones que sean más pequeñas que la máscara. Sin embargo, este tipo de filtro tiene el efecto indeseado de suavizar también los bordes, que son características generalmente deseadas [8].

1.4.1.3. Filtro de la Mediana

El objetivo de este filtro es la reducción del empañamiento de los bordes por medio de la sustitución del valor de gris del píxel analizado por la media de los valores de sus vecinos más cercanos, evaluados por el tamaño de la máscara aplicada. Tiene las ventajas de no ser afectado por píxeles individuales ruidosos y una fácil implementación. La eliminación de ruido obtenida con este filtro es bastante buena, con la ventaja de no empañar demasiado los bordes. Tiene como desventaja que su desempeño disminuye cuando el número de píxeles con ruido es mayor a la mitad del número total de píxeles en la vecindad [8].

1.4.2. Agudización de detalles y detección de bordes

Además de la corrección de faltas de homogeneidad y de supresión del ruido, la etapa de pre-procesamiento también puede involucrar la mejora de las estructuras relevantes. La detección de esquinas y líneas se basa en los operadores de detección de bordes, que mediante el cálculo de primeras (gradiente) y segundas (laplaciana) derivadas permiten determinar y resaltar los puntos donde ocurren cambios importantes en la intensidad de los píxeles.

El término de detección de bordes sugiere que la aplicación de un algoritmo con este propósito dará como resultado un contorno. Sin embargo, el objetivo de un algoritmo de detección de bordes es obtener imágenes cuya salida muestre píxeles de mayor intensidad en los valores que detecten transiciones cercanas [8]. Los bordes se encuentran en zonas de la imagen donde el nivel de intensidad fluctúa bruscamente, cuanto más rápido el cambio de intensidad, más fuerte el borde. Un buen proceso de

detección de bordes facilita la elaboración de fronteras de objetos, simplificando así el proceso de reconocimiento de objetos.

Existe una gran variedad de métodos para la detección de bordes y el cálculo de la magnitud del gradiente, algunos de los más conocidos son el operador de Roberts, el de Sobel, el de Prewitt y el Isotrópico [8]. Todos estos operadores realizan el cálculo de la magnitud del gradiente mediante la siguiente expresión:

$$M = \sqrt{S_x^2 + S_y^2} \quad (1)$$

Donde:

S_x : representa la derivada parcial calculada para el eje x.

S_y : representa la derivada parcial calculada para el eje y.

1.4.2.1. Operador de Roberts

El operador de Roberts es el operador de cálculo de gradiente más simple. Este utiliza las diagonales para calcular el vector gradiente [8] mediante las máscaras que se muestran en la Fig. 4.

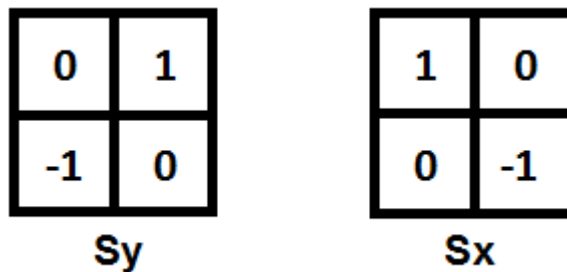


Fig. 4 Máscaras del operador de Roberts.

Estos términos representan una aproximación finita de la derivada del brillo. Este operador es fácil de implementar y requiere un bajo costo computacional debido a que utiliza algunas operaciones aritméticas sobre una vecindad reducida de cuatro píxeles. La principal desventaja de este operador es su alta sensibilidad al ruido, lo que provoca una respuesta débil a los bordes cuando estos no son muy pronunciados.

1.4.2.2. Operador de Sobel

En el operador de Sobel el cálculo de las derivadas parciales se realiza con las siguientes expresiones:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (2)$$

$$S_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2) \tag{3}$$

Donde:

a_k : representa el valor de la máscara y se representa en la Fig. 5.

c : representa una constante con valor 2.

a_0	a_1	a_2
a_7	a_8	a_3
a_6	a_5	a_4

Fig. 5 Coeficientes del operador de Sobel.

Al igual que en el operador de Roberts, S_x y S_y se pueden implementar con el uso de máscaras de convolución. Estas se muestran en la Fig. 6. El operador de Sobel pone un énfasis especial en los píxeles cercanos al centro de la máscara. Por su poca sensibilidad al ruido y su bajo costo computacional, es uno de los operadores más utilizados en la detección de bordes [8].

-1	0	1
-2	0	2
-1	0	1
Sx		

-1	-2	-1
0	0	0
1	2	1
Sy		

Fig. 6 Máscaras del operador de Sobel: Máscara para Sx (Izquierda), Máscara para Sy (Derecha).

1.4.2.3. Operador de Prewitt

El operador de Prewitt utiliza para cada derivada parcial S_x y S_y una máscara de 3x3 como se representan en la Fig. 7. Este operador tiene una alta inmunidad al ruido y utiliza la misma ecuación que el operador de Sobel, pero en este caso, la constante c tiene valor 1. Esto provoca que este operador no otorgue una importancia especial a los píxeles cercanos al centro de la máscara. Sin embargo, aporta resultados muy similares al operador de Sobel, pero mejores cuando los bordes presentes son mayoritariamente horizontales o verticales [8].

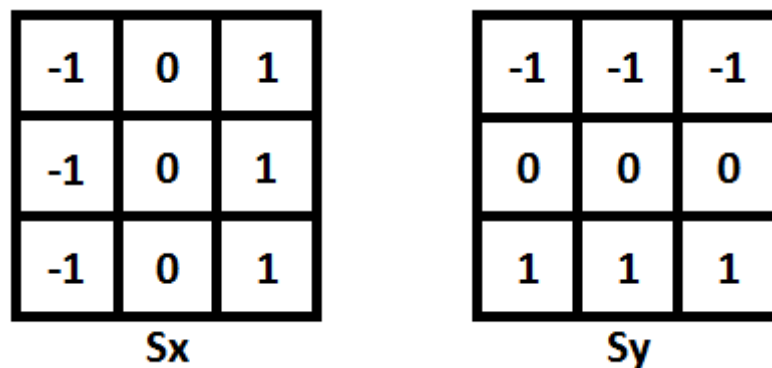


Fig. 7 Máscaras del operador de Prewitt: Máscara para S_x (Izquierda), Máscara para S_y (Derecha).

1.4.2.4. Operador Isotrópico

El operador de Prewitt proporciona una buena detección para bordes verticales y horizontales, mientras que el operador de Sobel es mejor en la detección de bordes diagonales. El operador Isotrópico intenta llegar a un equilibrio entre ellos aplicando las máscaras que se muestran en la Fig. 8. Utiliza la misma ecuación que el operador de Sobel, pero en este caso, la constante c tiene un valor de $\sqrt{2}$. El costo computacional del operador Isotrópico es mayor debido a que introduce la operación de radicalización. Este operador es también conocido como operador de Frei-Chen [8].

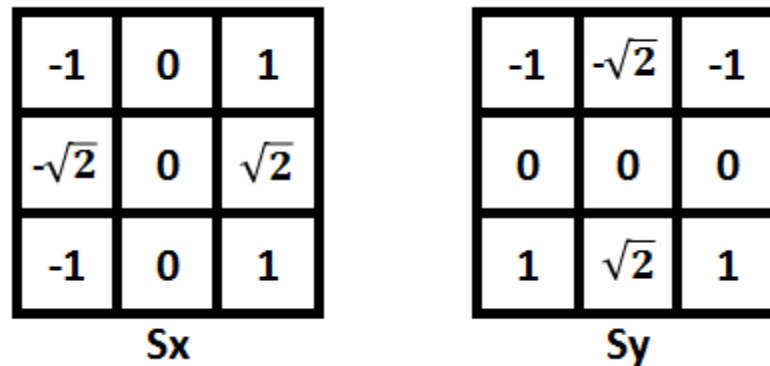


Fig. 8 Máscaras del operador Isotrópico: Máscara para S_x (Izquierda), Máscara para S_y (Derecha).

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 [5]. 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 [9].

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.

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 [5].

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 [5].

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 [5].

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 [1].

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 [9].

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. Segmentación basada en Umbrales

Los métodos de umbralización se basan en el valor del píxel y no consideran el contexto espacial, o sea, toman sus decisiones a partir de las características de color de los píxeles. La idea fundamental se basa en suponer que el rango de valores de un objeto respecto a una determinada propiedad es relativamente pequeño. Aquellos píxeles que estén en ese rango serán parte del objeto y aquellos que queden fuera, formarán parte del fondo de la imagen. La ecuación $e = |I(p) - a|$ especifica la situación típica, en la cual un intervalo de umbral es definido por un valor aproximado a y una tolerancia e . Todos los valores de los píxeles $I(p)$ que caen dentro del intervalo resultante $([a-e, a+e])$ serán seleccionados, generándose de esta forma una imagen binaria. La parte más difícil de este proceso es encontrar el valor correcto del umbral, por lo que a veces se usan varios de estos valores [5].

Este método se emplea cuando hay una clara diferencia entre los objetos a extraer respecto del fondo de la escena. Los elementos más importantes a tener en cuenta son: la similitud entre los píxeles pertenecientes a un objeto y sus diferencias respecto al resto. Por tanto, la escena debe caracterizarse por un fondo uniforme y por objetos parecidos para que el proceso de segmentación brinde buenos resultados [6].

1.5.2.2. 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.

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 [5].

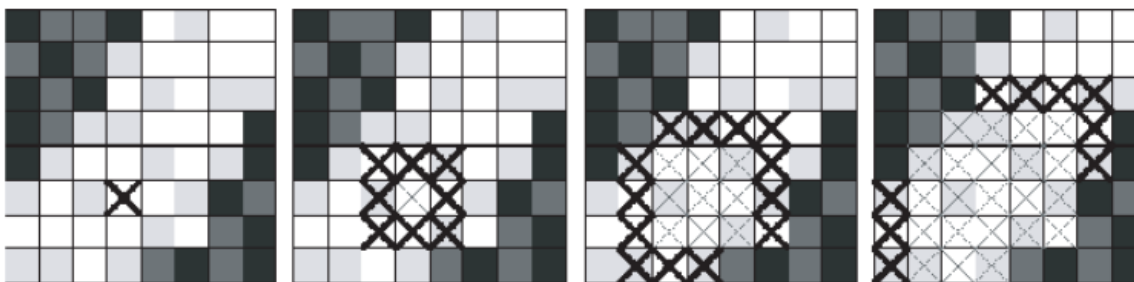


Fig. 9 Diferentes estados de un algoritmo de crecimiento de regiones. En la imagen más a la izquierda se marca la semilla inicial y en las restantes de izquierda a derecha el sucesivo crecimiento de la selección de los píxeles involucrados en la segmentación.

1.5.2.3. Agrupamiento

Los algoritmos de Agrupamiento (*Clustering*) llevan a cabo esencialmente la misma función que los métodos clasificadores tratados con mayor profundidad en [1], 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 [1].

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 Fig. 10 muestra el resultado de aplicar el algoritmo *K-Means* a un corte de la 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 [1].

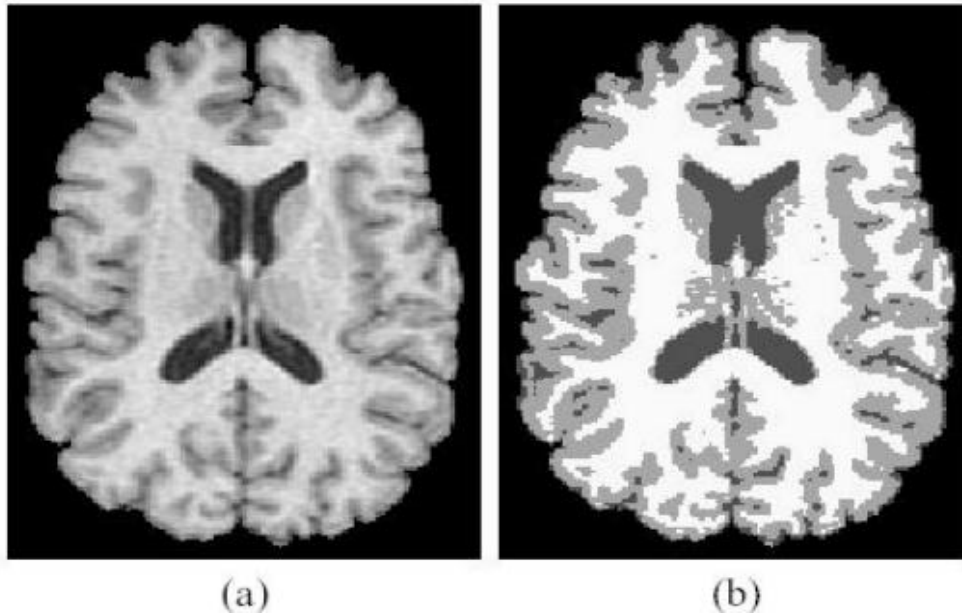


Fig. 10 (a) Imagen original (b) Imagen segmentada con el algoritmo K-Means.

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 [1].

1.5.2.4. 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.

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 [5].

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 [6].

1.5.2.5. Segmentación basada en modelos

Una de las estrategias que mejor soportan las tareas de segmentación es la basada en modelos, estas emplean conocimientos del tamaño, la forma de los objetos o distribuciones de niveles de gris para la segmentación. En las aplicaciones de planeación de terapias generalmente existe una variedad de objetivos de segmentación y algunos presentan localizaciones ajustadas en relación con otros o formas específicas a partir de las cuales con una secuencia apropiada de pasos de segmentación se pueden lograr resultados muy buenos y rápidos [5].

1.5.2.6. Modelos deformables

Los modelos deformables se basan en una flexible representación geométrica como los *Splines*, que proveen el grado de libertad necesaria para adaptar el modelo a una gran variedad de formas. El proceso de ajustar el modelo a la estructura objetivo es guiado por principios físicos y restricciones, principalmente principios de la teoría de la elasticidad que describen cómo un cuerpo deformable responde a las fuerzas ejercidas sobre él. Las variaciones resultantes de la forma dependen de algunas propiedades de rigidez de dicho organismo. El uso de modelos deformables para la segmentación de imágenes fue sugerido en [10] y ha sido usado intensivamente desde entonces [5].

1.5.2.7. 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 [5].

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 [5]. 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 [5].

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 [11].

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 [12].

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

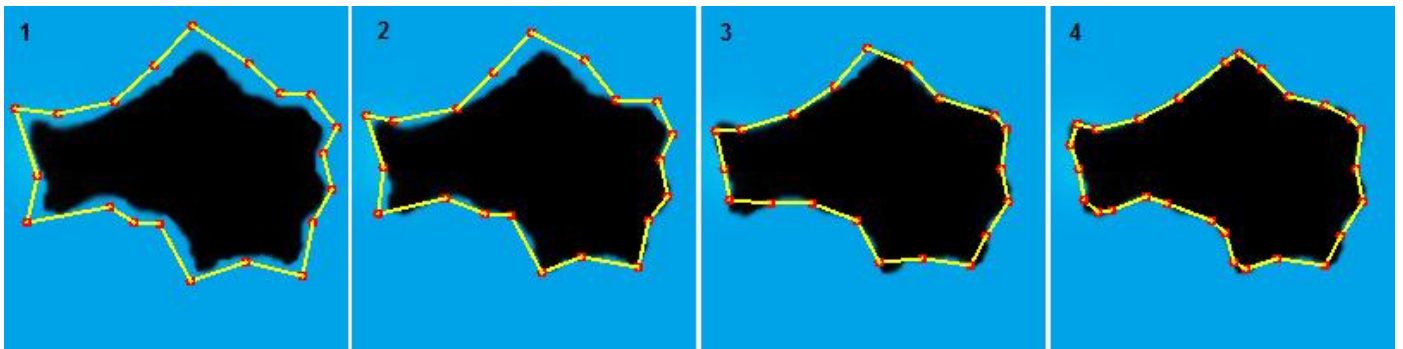


Fig. 11 Proceso iterativo de ajuste y segmentación del Snake sobre un objetivo en una imagen.

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.

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 [11]. En la Fig. 12 se observa una representación paramétrica del contorno de segmentación.

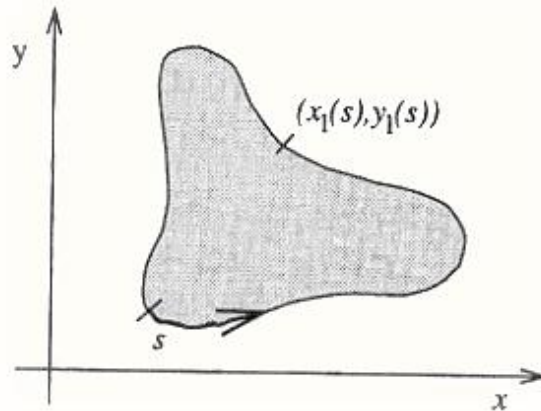


Fig. 12 Representación paramétrica del contorno.

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:

$$E(s) = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{imag}) \quad (4)$$

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 $c(s)$ [11] 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.

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:

$$E_{cont} = \left\| \frac{dc}{ds} \right\|^2 \quad (5)$$

Para el contorno discreto, el contorno es aproximado por N vértices p_1, p_2, \dots, p_n y la primera derivada se puede aproximar por la diferencia finita:

$$E_{cont} = \|p_i - p_{i-1}\|^2 \quad (6)$$

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:

$$E_{cont} = (d - \|p_i - p_{i-1}\|)^2 \quad (7)$$

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 [11].

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

p_{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 figura [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 [11]:

$$E_{curv} = \left\| \frac{d^2c}{ds^2} \right\|^2 \quad (8)$$

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

$$E_{curv} = \|p_{i-1} - 2p_i + p_{i+1}\|^2 \quad (9)$$

Donde:

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

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

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

Energía de la Imagen

La energía de la imagen (E_{img}) 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 [12]. 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 [11]. La ecuación de la energía de la imagen es la siguiente:

$$E_{img} = -||\nabla I|| \quad (10)$$

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 [13].

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 implementados en el proyecto *Vismedic* 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 brindarle al usuario la posibilidad de intervenir en el proceso de segmentación.

Capítulo 2: Solución Propuesta

Este capítulo abarca la solución propuesta para resolver el problema planteado. 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. Finalmente se mencionan las herramientas que se utilizaron para la elaboración de la solución propuesta así como la metodología de desarrollo que se empleó durante el proceso.

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. La investigación realizada demostró que es la técnica que mejor 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 Fig. 13.

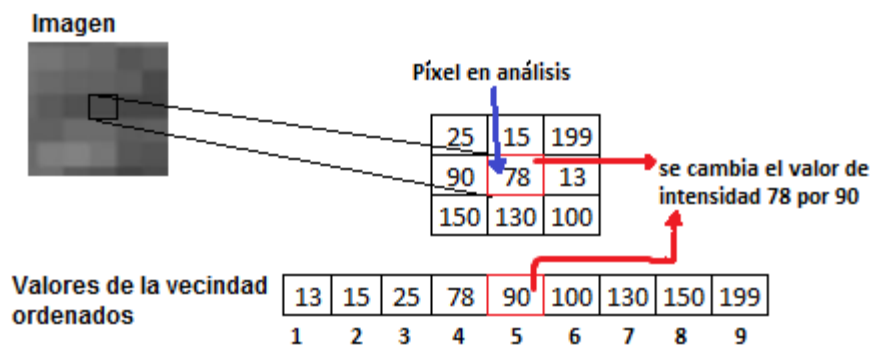


Fig. 13 Operación del filtro de la Mediana.

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. 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 será más exacta mientras más vértices contenga este. Esta representación se puede apreciar en la Fig. 14.

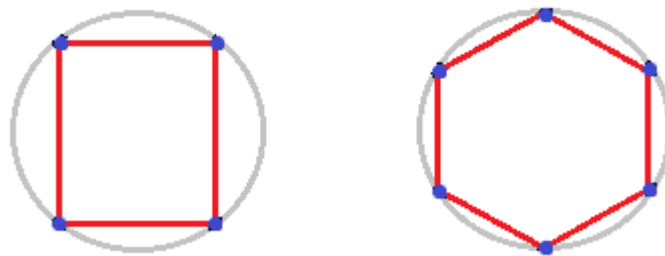


Fig. 14 Izquierda: Circunferencia representada por un polígono de 4 vértices. Derecha: Circunferencia representada por un polígono de 6 vértices.

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 Fig. 15.

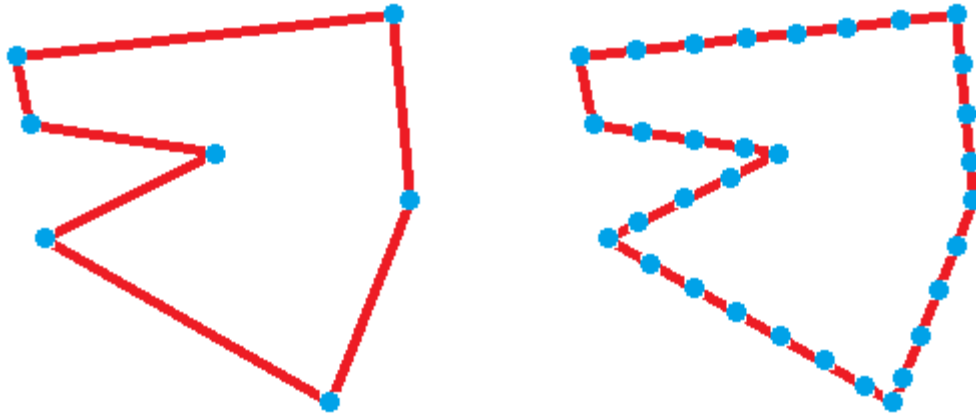


Fig. 15 Izquierda: Polígono inicial generado. Derecha: Polígono luego de aplicado el algoritmo de relleno de vértices.

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 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} . Para el cálculo de la energía del contorno se utiliza la siguiente expresión:

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

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 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:

$$E_{curv} = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2 \quad (12)$$

Donde:

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

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:

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 Fig. 16.
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.

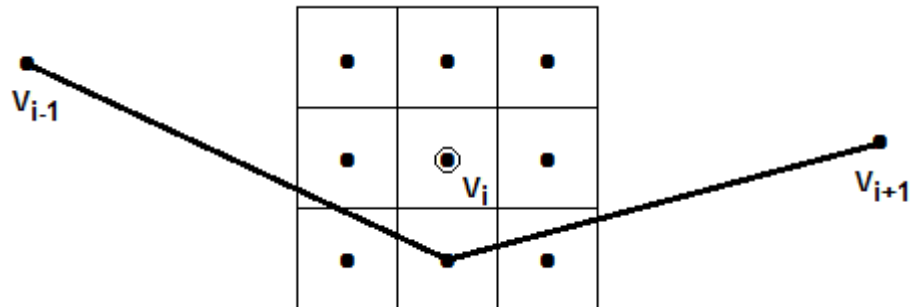


Fig. 16 Representación de la vecindad de búsqueda de un *snaixel*.

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 [11].

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 $\text{num_snaixels}-1$

 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).
```

2.8. Herramientas

Para implementar la solución propuesta se utilizan algunas herramientas que se describen en este epígrafe. Como herramienta de modelado se utiliza Visual Paradigm. Este utiliza UML como lenguaje de modelado y se considera una herramienta muy completa, con soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Ofrece además la capacidad de ingeniería directa (versión profesional) e inversa.

Para el desarrollo se utiliza el kit de desarrollo de software o SDK (siglas en inglés de *software development kit*) de Qt [14] que utiliza C++ de manera nativa, pero ofrece soporte para otros lenguajes como Python mediante PyQt, Java o C#. Qt es un framework muy poderoso, comparable con Swing de Java o .NET de Microsoft. Está disponible bajo licencia GPL y además ofrece una suite de aplicaciones para facilitar y agilizar las tareas de desarrollo. Las aplicaciones que componen esta suite son:

- Qt Assistant: Herramienta para visualizar la documentación oficial de Qt.
- Qt Designer: Herramienta para crear interfaces de usuario.
- Qt Linguist: Herramienta para la traducción de aplicaciones.
- Qt Creator: IDE para el lenguaje C++, pero especialmente diseñado para Qt, integra las primeras dos herramientas mencionadas.

Para el trabajo con imágenes DICOM se utiliza la biblioteca libre y multiplataforma DCMTK en su versión 3.6.0 [15]. Esta biblioteca es una colección de aplicaciones que implementan el estándar DICOM e incluye software para examinar, construir y convertir imágenes DICOM. DCMTK está escrito en una mezcla de los lenguajes ANSI C y C++ [15].

2.9. Metodología de desarrollo de software

Como metodología de desarrollo de software se selecciona el Proceso Unificado para Desarrollo de Software (RUP) [16]. Utiliza como único lenguaje de modelado UML y puede ser utilizado para la construcción de todos los artefactos necesarios durante el desarrollo de la aplicación [18]. Otras de las ventajas que tiene RUP son que está guiada por casos de uso, centrada en la arquitectura y se desarrolla como un proceso iterativo e incremental [16].

Capítulo 3: Análisis y Diseño del Sistema

Este capítulo abarca como tema fundamental el análisis y diseño de la aplicación. Primeramente se definen algunas reglas del negocio y el modelo de dominio a utilizar. Luego se hace la captura de requisitos, que se utiliza posteriormente en la elaboración del Modelo de Casos de Uso. Se definen los actores del sistema y se especifica cada caso de uso obtenido a partir de la captura de requisitos. Se muestra un diagrama de clases del sistema y por último se modelan diferentes diagramas de secuencia del diseño con el objetivo de proporcionar una comprensión más profunda del sistema que se desea desarrollar.

3.1. Reglas del negocio

- Las imágenes médicas que se deseen segmentar deben estar en formato DICOM (*.dcm).
- Las imágenes médicas deben tener la estructura original, las que no cumplan este requisito pueden no ser cargadas o visualizadas de forma correcta.

3.2. Modelo del dominio

El modelo de dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema y conceptos del mundo real en vez de componentes de software [16]. El entorno donde está presente el problema se describe de la siguiente forma: un médico le orienta a un paciente una serie de estudios médicos que generan un conjunto de imágenes, las cuales son generadas por las TAC o RM; donde a partir del análisis de las mismas el médico emite un diagnóstico. En la Fig. 17 se muestra la descripción del negocio mediante un modelo de dominio. A continuación se describen los conceptos que forman parte del modelo de negocio:

El **Médico** es la persona capacitada en el manejo y conocimiento de los métodos de Imagenología para emitir un diagnóstico.

Un **Estudio** es el examen orientado por el médico al paciente para obtener una patología y emitir un diagnóstico.

Las **Imágenes** constituyen el resultado del estudio orientado por el médico al paciente a través de las TAC y RM.

El **Paciente** es aquel que recibe los servicios de un médico u otro profesional de la salud, sometiéndose a un examen.

Una **RM** es una prueba que realiza una fotografía de los órganos y tejidos internos sin exponer al paciente a radiaciones, como resultado se obtienen imágenes computarizadas.

Una **TAC** es un procedimiento de diagnóstico médico que utiliza rayos X con un sistema informático que procesa las imágenes y permite obtener imágenes radiográficas en secciones progresivas de la zona del organismo estudiada, y si es necesario, imágenes tridimensionales de los órganos o estructuras orgánicas.

El **Diagnóstico** es el resultado que da el médico luego de realizar un estudio a un paciente.

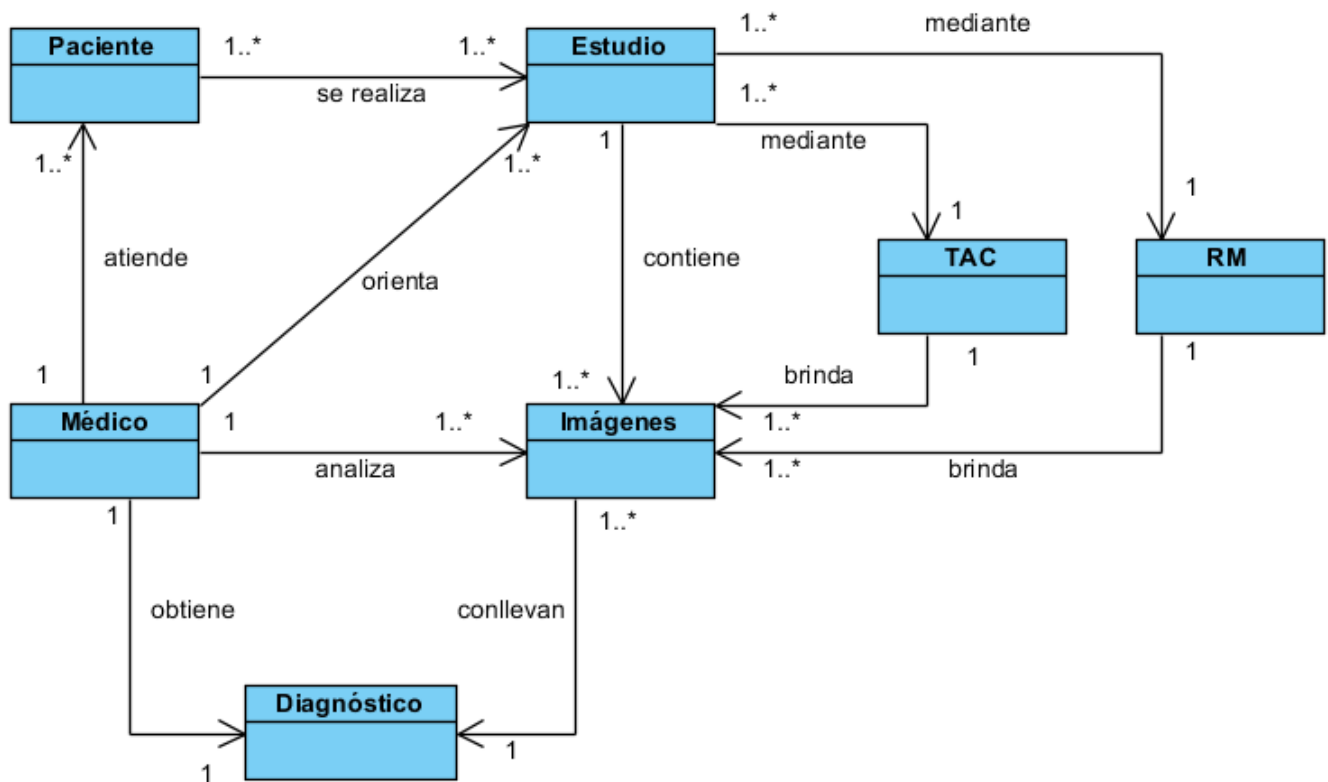


Fig. 17 Diagrama del Modelo de Dominio.

3.3. Captura de requisitos

Un requerimiento es una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, norma, especificación u otro documento formal, facilitando el

entendimiento entre clientes y desarrolladores. Seguidamente se exponen los requisitos funcionales y no funcionales, por los cuales se rige el desarrollo del módulo.

3.3.1. Requisitos funcionales

Los requisitos funcionales representan las funcionalidades del sistema [16]. Estos son transformados en casos de uso y modelados posteriormente. Los siguientes requisitos responden a las funcionalidades que el sistema debe tener una vez concluida su implementación.

RF1. Cargar imagen.

RF2. Filtrar imagen.

RF2.1 Configurar parámetros del filtro.

RF2.2 Eliminar ruido.

RF3. Gestionar contorno.

RF3.1 Insertar contorno inicial.

RF3.2 Modificar contorno.

RF3.3 Eliminar contorno.

RF4. Segmentar imagen.

RF4.1 Configurar parámetros de segmentación.

3.3.2. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener y representan las características que hacen al producto atractivo, usable, rápido o confiable [16]. Seguidamente se enumeran los requisitos no funcionales del módulo a desarrollar.

1. Software

El sistema operativo sobre el cual debe ejecutarse la aplicación será Ubuntu.

2. Hardware

El modelo de microprocesador será Intel Pentium IV a 1.6 GHz o superior.

La memoria RAM será DDR2 o superior de 1GB.

Espacio libre en disco duro de 512 Mb.

3. Seguridad

El sistema debe ser seguro frente a posibles ataques que se produzcan en la introducción de los datos de las imágenes y debe realizar un correcto tratamiento de errores para evitar el colapso de la aplicación al ocurrir algún error.

4. Interfaz externa

La interfaz de usuario debe ser sencilla y amigable, lo que le permitirá al usuario interactuar de una forma rápida y cómoda con las funcionalidades del módulo.

5. Restricciones en el diseño e implementación

Se empleará como lenguaje de programación C++ bajo el paradigma de Programación Orientada a Objetos y el Framework Qt para el diseño de las interfaces gráficas.

6. Eficiencia

Rendimiento: El sistema debe brindar los resultados en tiempo real.

7. Soporte

Se brindará soporte para el sistema operativo Ubuntu.

3.4. Modelo de Casos de Uso del Sistema

En este epígrafe se identifican los actores del sistema que se desea desarrollar así como los casos de uso. Además se realiza la descripción textual de los casos de uso del sistema, lo que permite una comprensión más precisa y una mejor idea de la lógica del funcionamiento del módulo.

3.4.1. Actores del sistema

Los actores del sistema son entidades externas al sistema que guardan una relación con este y que le demandan una o más funcionalidades. Esto incluye a los operadores humanos, pero también incluye a

todos los sistemas externos. En este caso particular quien interactúa con el módulo es un especialista médico, por lo que como actor del sistema se le llamara Médico Especialista.

Tabla 1 Actores del sistema.

Actores	Justificación
Médico Especialista	El Médico Especialista es quien interactúa con el sistema para ejecutar las funcionalidades de: cargar imágenes DICOM, mejorar la calidad visual de la imagen y segmentar las estructuras anatómicas de interés.

3.4.2. Diagrama de Casos de Uso del Sistema

El Diagrama de Casos de Uso del Sistema representa una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debe interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. En la Fig. 18 se muestra el Diagrama de Casos de Uso del Sistema para el módulo en cuestión.

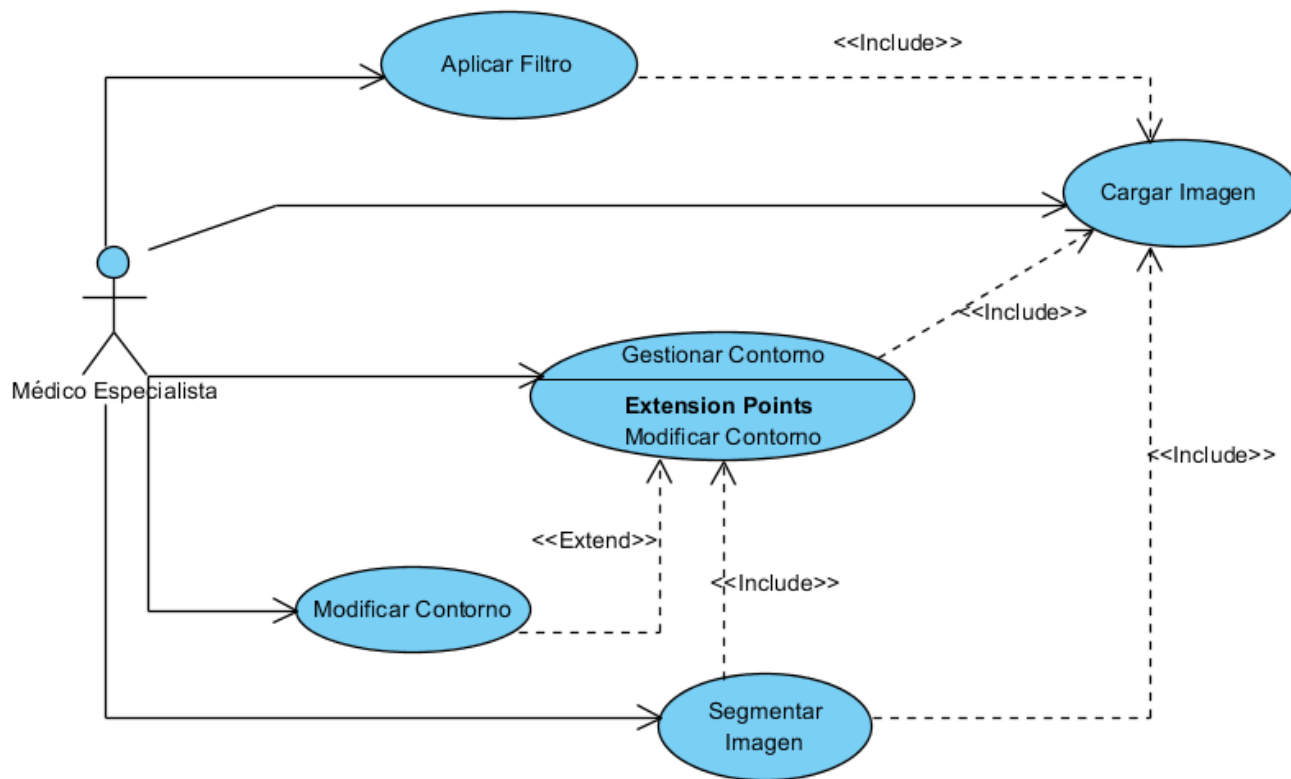


Fig. 18 Diagrama de Casos de Uso.

3.4.3. Descripción de Casos de Uso del Sistema

Cada caso de uso posee una descripción de las acciones que realiza el sistema como respuesta a las peticiones del usuario. A continuación se relacionan las tablas correspondientes a las descripciones de los casos de uso detectados y se argumentan los flujos operacionales de cada uno. Los casos de uso más complejos se han dividido en secciones para lograr una mejor comprensión de los mismos.

Tabla 2 Descripción del caso de uso Cargar Imagen.

Caso de Uso:	Cargar imagen
Actores:	Médico Especialista.
Propósito:	Importar la imagen que el especialista desea segmentar.
Resumen:	Se inicia cuando el actor selecciona la opción de cargar imagen. Se selecciona el directorio donde están las imágenes DICOM. Finaliza con la

	carga del fichero objetivo y su posterior visualización.
Referencia:	RF1
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de Cargar Imagen.	1.1 Muestra el cuadro de diálogo con las imágenes DICOM disponibles para que el usuario seleccione la imagen que desea cargar.
2. Selecciona la imagen que desea cargar y presiona el botón Aceptar.	2.1 Se cierra el cuadro de diálogo y se carga la imagen seleccionada. El sistema muestra las opciones para habilitar las demás funcionalidades.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
2. El usuario presiona el botón Cancelar.	2.1 Se cierra el cuadro de diálogo y se muestra la interfaz principal de la aplicación.
Postcondiciones:	Se carga y se muestra la imagen seleccionada así como la interfaz de las demás funcionalidades.
Prioridad:	Crítica.

Tabla 3 Descripción del caso de uso Filtrar Imagen.

Caso de Uso:	Aplicar Filtro
Actores:	Médico Especialista.
Propósito:	Aplicar filtros a la imagen para mejorar la calidad de su visualización.
Resumen:	Se inicia una vez cargada la imagen a segmentar y establecido el parámetro de orden de filtrado, se selecciona la opción de filtrar que mejora la calidad de la imagen de manera automática reduciendo el ruido presente en esta. Finaliza después de aplicado el filtro a la imagen cargada.
Referencia:	RF2, RF2.1, RF 2.2
Precondiciones:	Debe haberse ejecutado el caso de uso Cargar imagen.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

1. Introduce en una caja de texto el parámetro de orden de filtrado en el espacio correspondiente y se selecciona la opción de aplicar filtro.	1.1 Se valida que el valor introducido sea un número entero que esté dentro del rango entre 1 y 5. 1.2 Se aplica el filtro de reducción de ruido a la imagen y se muestra la imagen resultante de en el visor.
Postcondiciones:	Se muestra la imagen obtenida después de aplicado el filtro.
Prioridad:	Media.

Tabla 4 Descripción del caso de uso Gestionar Contorno.

Caso de Uso:	Gestionar Contorno.
Actores:	Médico Especialista.
Propósito:	Gestionar la introducción y eliminación del contorno de segmentación.
Resumen:	Se inicia cuando el actor define la descripción del contorno de segmentación de forma manual. Este caso de uso se encarga de crear el contorno para la segmentación y de la eliminación del mismo.
Referencia:	RF3, RF3.1, RF3.3
Precondiciones:	Debe haberse ejecutado del caso de uso Cargar imagen.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor decide entre insertar un contorno o eliminar el contorno que se utilizará para la segmentación.	1.1 El sistema muestra el resultado según se realizan las siguientes acciones: a) Si el actor desea insertar un contorno: consultar sección Insertar Contorno. b) Si el actor desea eliminar el contorno entero: consultar sección Eliminar Contorno.
Postcondiciones:	Se gestiona el contorno que se utilizará para la segmentación.
Prioridad:	Crítica.

Tabla 5 Descripción de la sección Insertar Contorno, perteneciente al caso de uso Gestionar Contorno.

Sección:	Insertar Contorno
Flujo Normal de Eventos	

Acción del Actor		Respuesta del Sistema
1. Comienza la descripción del contorno presionando el botón izquierdo del ratón sucesivamente sobre la imagen para introducir los puntos que lo conforman iterativamente hasta alcanzar la forma deseada.		1.1 El sistema introduce y visualiza cada vértice, uniendo cada uno de ellos mediante un segmento y mostrando como resultado la forma del contorno sobre la imagen.
Postcondiciones:	Se muestra el contorno definido sobre la imagen.	
Prioridad:	Crítica.	

Tabla 6 Descripción de la sección Eliminar Contorno, perteneciente al caso de uso Gestionar Contorno.

Sección:		Eliminar Contorno
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. Selecciona la opción de eliminar el contorno.		1.1 Se elimina el contorno completo. 1.2 Muestra el resultado en pantalla.
Postcondiciones:	Se elimina el contorno de segmentación y se muestra la imagen.	
Prioridad:	Media.	

Tabla 7 Descripción del caso de uso Modificar Contorno.

Caso de Uso:	Modificar Contorno.
Actores:	Médico Especialista.
Propósito:	Modificar los vértices que componen el contorno de segmentación.
Resumen:	Se inicia cuando el actor realiza las siguientes operaciones: modificar la posición de un vértice del contorno, adicionar vértices entre las aristas o eliminar un vértice del contorno.
Referencia:	RF3, RF3.2
Precondiciones:	Debe haberse ejecutado la sección Insertar Contorno del caso de uso Gestionar Contorno.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El actor decide entre cambiar la posición de un vértice, eliminar un vértice del contorno o incrementar la cantidad de vértices del contorno.	1.1 El sistema muestra el resultado según se ejecutan las siguientes acciones: a) Si el actor desea cambiar la posición de un vértice del contorno: consultar sección de Cambiar Posición de Vértice. b) Si el actor desea eliminar un vértice: consultar sección de Eliminar Vértice. c) Si el actor desea incrementar la cantidad de vértices del contorno: consultar la sección de Rellenar Aristas.
Postcondiciones:	Se modifica el contorno de segmentación de acuerdo a las operaciones realizadas por el usuario.
Prioridad:	Crítica.

Tabla 8 Descripción de la sección Cambiar Posición de Vértice, perteneciente al caso de uso Gestionar Contorno.

Sección:	Cambiar Posición de Vértice
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Presiona el botón izquierdo del ratón sobre un vértice del contorno y arrastra el vértice hacia la nueva posición presionando el botón izquierdo del ratón, donde se libera. 2. Presiona clic derecho.	1.1 Se dibuja el vértice seleccionado de otro color ilustrando que se encuentra seleccionado y cambia la forma del cursor del ratón indicando visualmente la operación, se va desplazando el vértice mientras se mueve el ratón. 1.2 Una vez liberado el ratón, el cursor obtiene su forma original. 2.1 Se dibuja el vértice anteriormente seleccionado de su color original.
Postcondiciones:	Se dibuja el contorno modificado.
Prioridad:	Media.

Tabla 9 Descripción de la sección Eliminar Vértice, perteneciente al caso de uso Gestionar Contorno.

Sección:	Eliminar Vértice
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Presiona el botón izquierdo del ratón sobre un vértice del contorno. 2. Presiona la tecla Delete.	1.1 Se dibuja el vértice seleccionado de otro color para identificarlo. 2.1 Se elimina el vértice seleccionado y se actualiza la forma del contorno.
Postcondiciones:	Se muestra en pantalla el contorno modificado.
Prioridad:	Media.

Tabla 10 Descripción de la sección Resetear, perteneciente al caso de uso Gestionar Contorno.

Sección:	Rellenar Aristas
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce el valor de orden de relleno en el campo correspondiente y selecciona la opción de Rellenar Aristas.	1.1 Se valida que el valor de orden de relleno sea un número entero y se encuentre dentro del rango entre 5 y 200. 1.2 Se generan vértices adicionales en cada una de las aristas del contorno en dependencia del valor de orden introducido y se muestra el nuevo resultado en la pantalla.
Postcondiciones:	Se obtiene un contorno con mayor cantidad de vértices.
Prioridad:	Media.

Tabla 11 Descripción del caso de uso Segmentar.

Caso de Uso:	Segmentar Imagen
Actores:	Médico Especialista.
Propósito:	Segmentar estructuras de interés presentes en la imagen.
Resumen:	Se inicia cuando el actor ha definido el contorno de la estructura que desea segmentar. Posteriormente este introduce los parámetros necesarios y finalmente realiza el proceso de segmentación de la imagen.

Referencia:	RF4, RF4.1
Precondiciones:	Debe haberse ejecutado el caso de uso Gestionar Contorno.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Introduce los parámetros <i>alpha</i> , <i>betha</i> , <i>ganma</i> y el orden de la matriz de búsqueda. Luego selecciona la opción de iniciar la segmentación.	1.1 Se valida que valor del orden de la matriz de búsqueda introducido sea entero y se encuentre en el rango de 2 a 50. 1.2 El sistema muestra el proceso iterativo de segmentación y muestra el contorno ajustado al objetivo de segmentación.
Postcondiciones:	Se visualiza la imagen con el contorno objetivo segmentado.
Prioridad:	Crítica.

3.5. Diagrama de Clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases se utilizan durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. A continuación se muestra una imagen que representa el diagrama de clases del sistema.

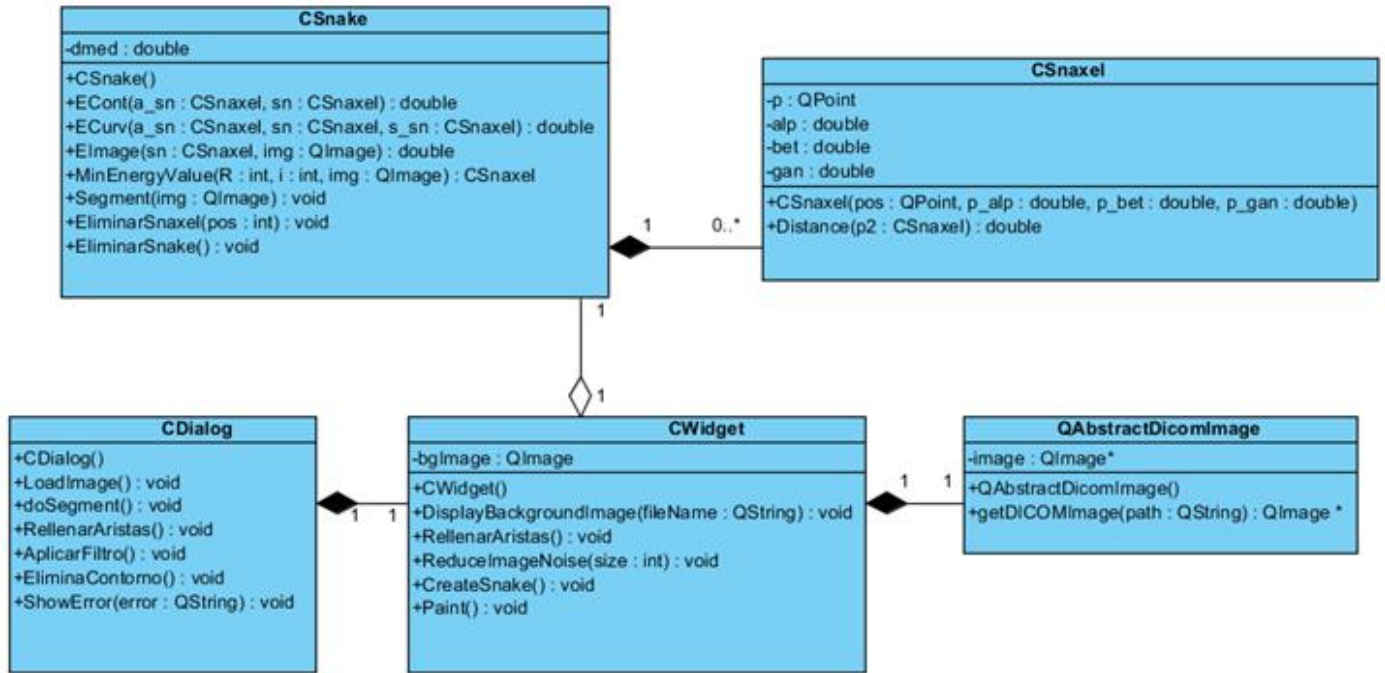


Fig. 19 Diagrama de Clases.

3.6. Diagramas de Secuencia del Diseño

A continuación se representan los diagramas de secuencia del diseño para tener una idea más general sobre el flujo que se realiza entre las clases del diseño y que posibilita comprender mejor el módulo elaborado en términos de implementación.

3.6.1. Diagrama de Secuencia del caso de uso Cargar Imagen

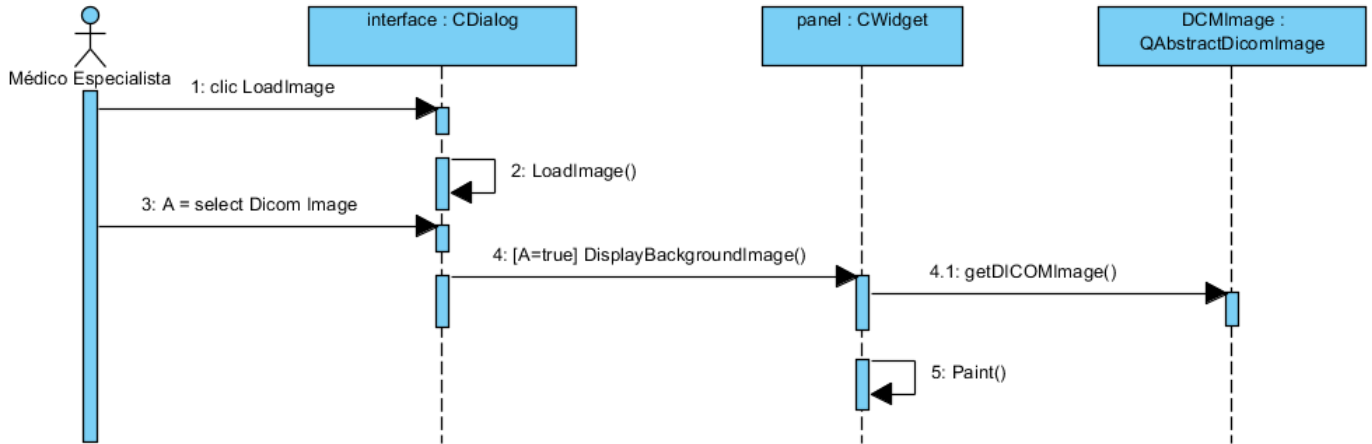


Fig. 20 Diagrama de Secuencia del caso de uso Cargar Imagen.

3.6.2. Diagrama de Secuencia del caso de uso Aplicar Filtro

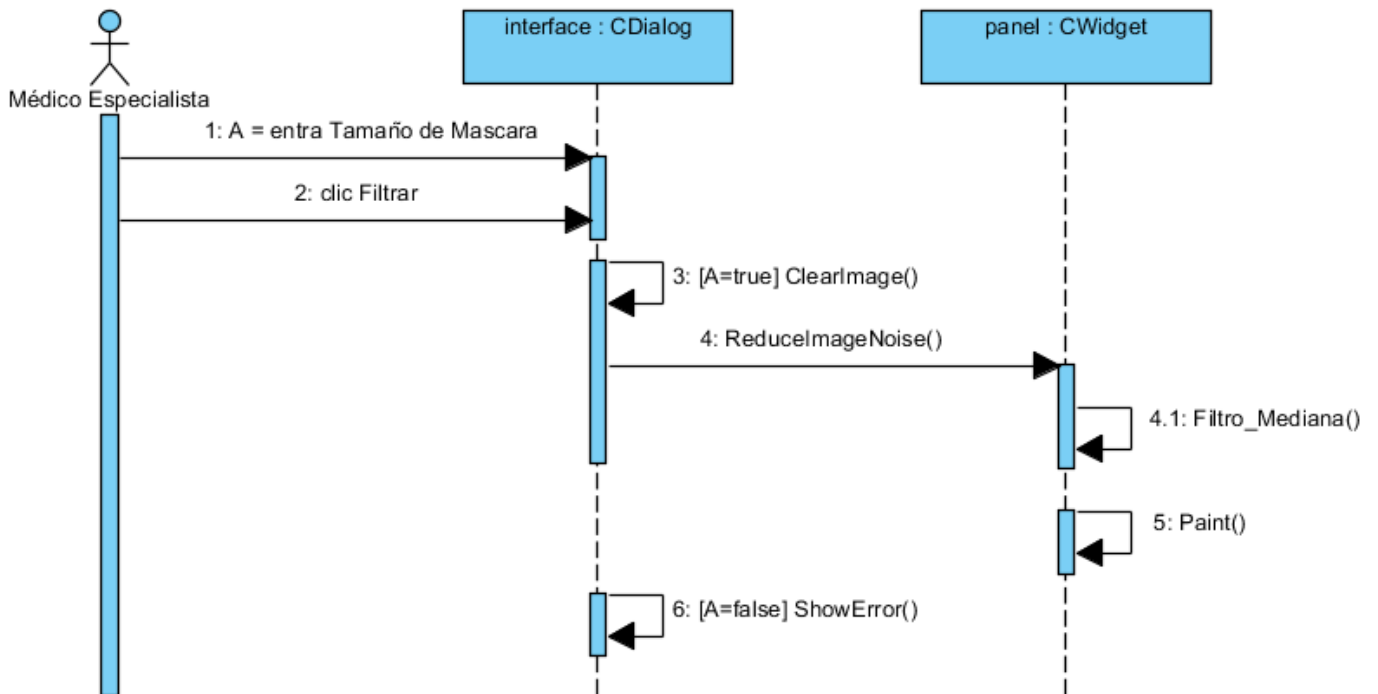


Fig. 21 Diagrama de Secuencia del caso de uso Aplicar Filtro.

3.6.3. Diagrama de Secuencia del caso de uso Gestionar Contorno

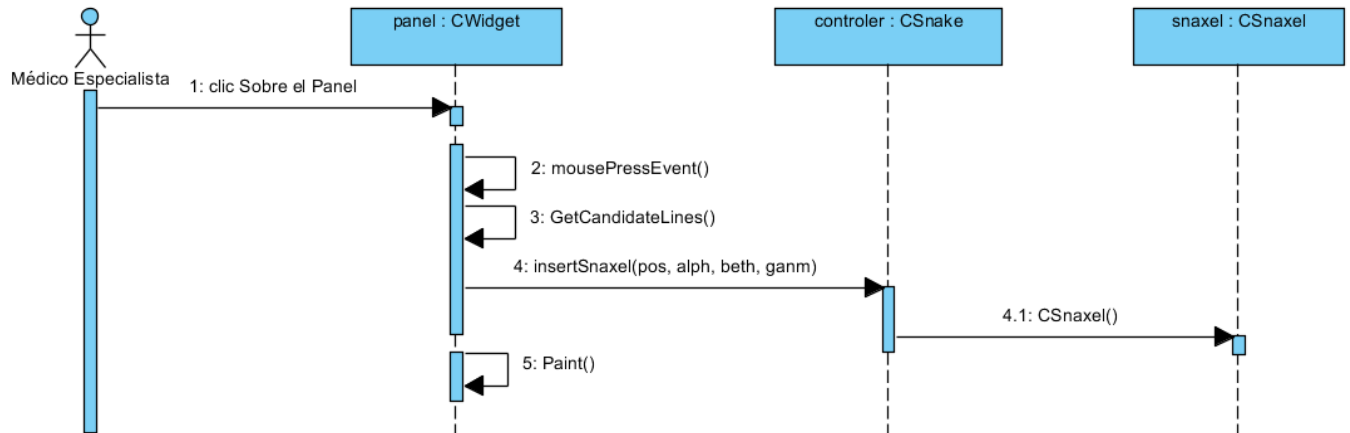


Fig. 22 Diagrama de Secuencia de la sección Adicionar Contorno del caso de uso Gestionar Contorno.

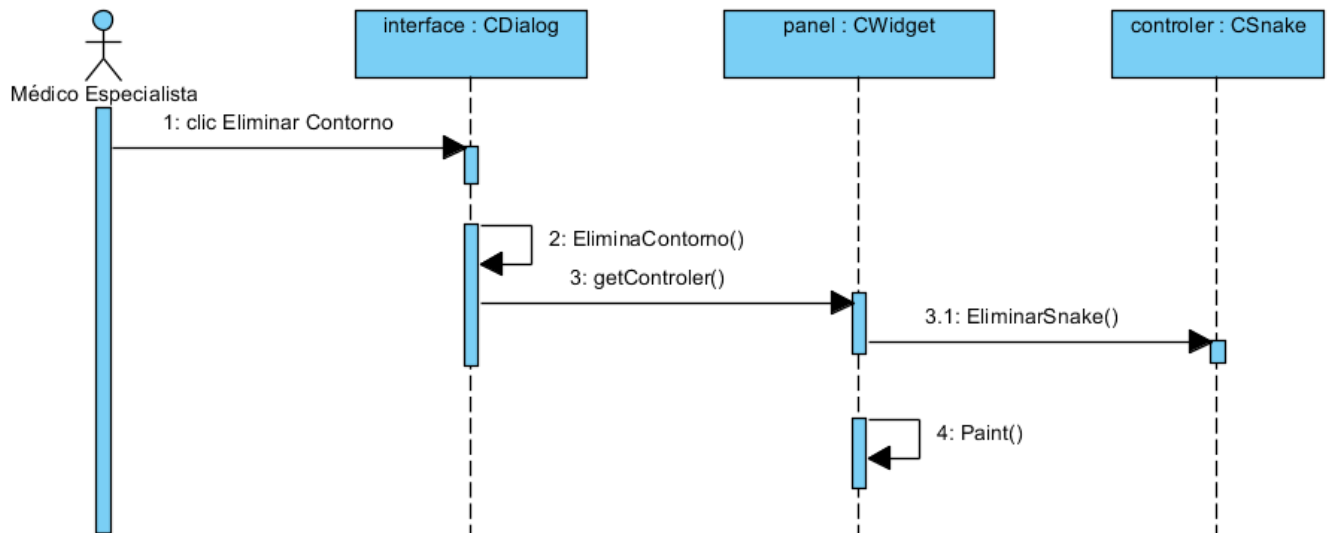


Fig. 23 Diagrama de Secuencia de la sección Eliminar Contorno del caso de uso Gestionar Contorno.

3.6.4. Diagrama de Secuencia del caso de uso Modificar Contorno

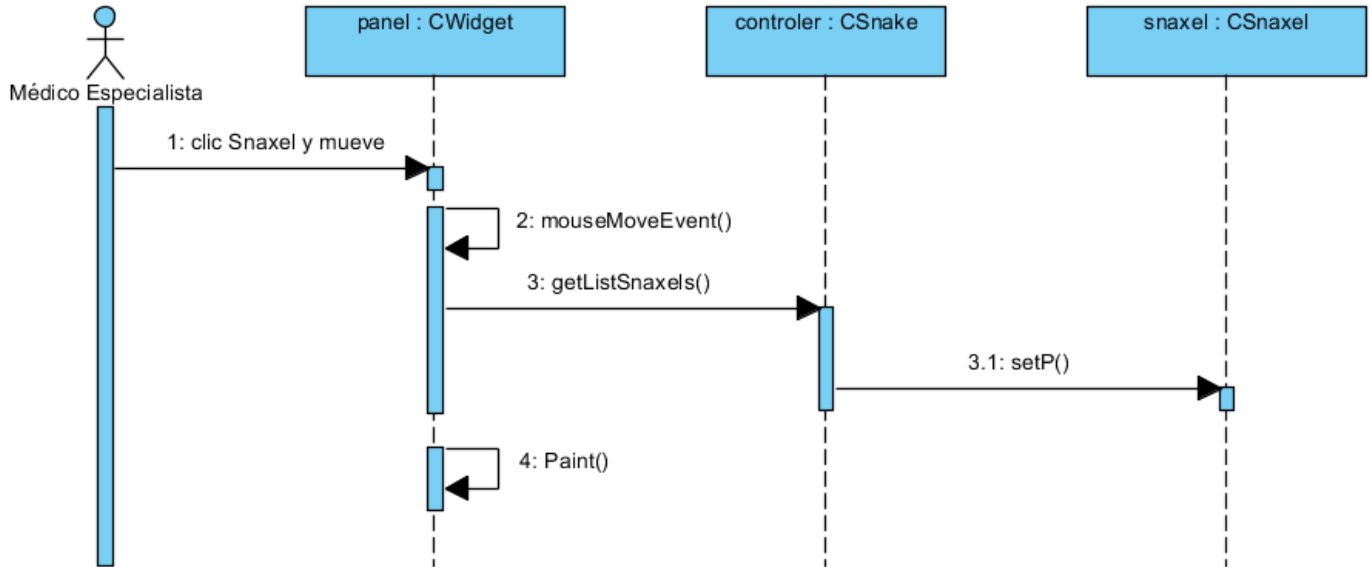


Fig. 24 Diagrama de Secuencia de la sección Cambiar Posición de Vértice del caso de uso Modificar Contorno.

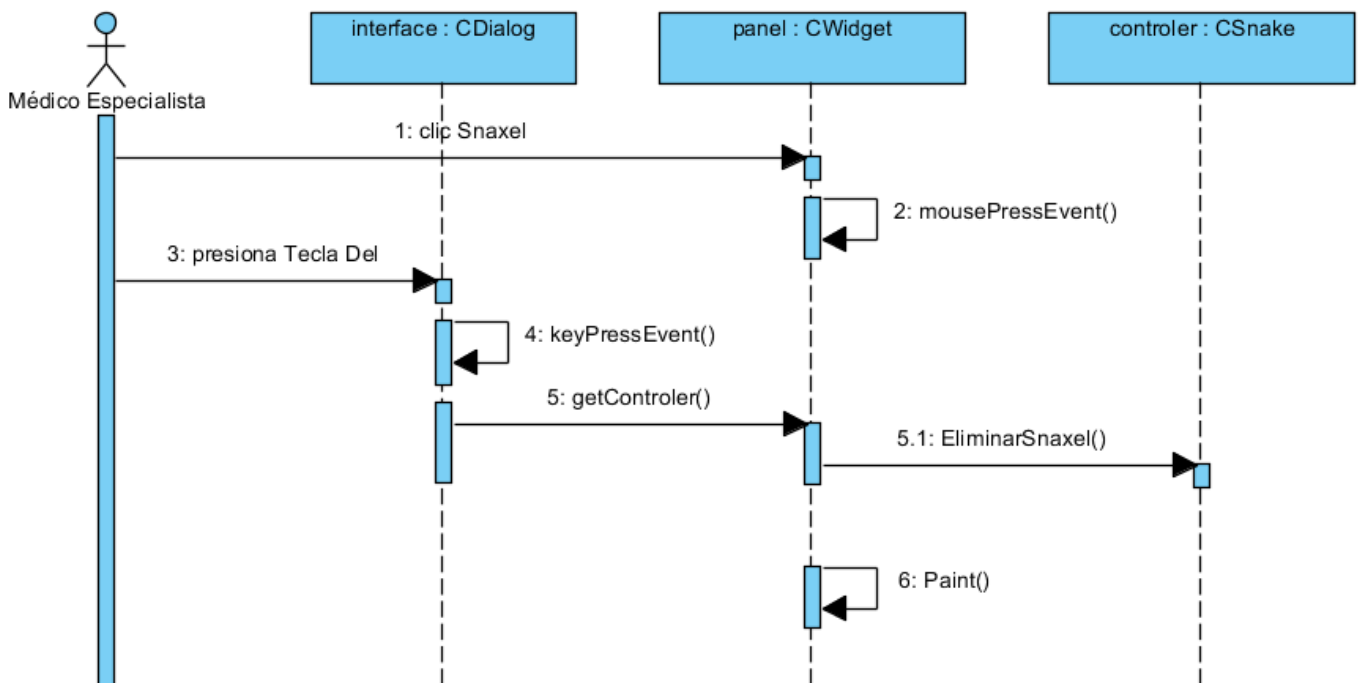


Fig. 25 Diagrama de Secuencia de la sección Eliminar Vértice del caso de uso Modificar Contorno.

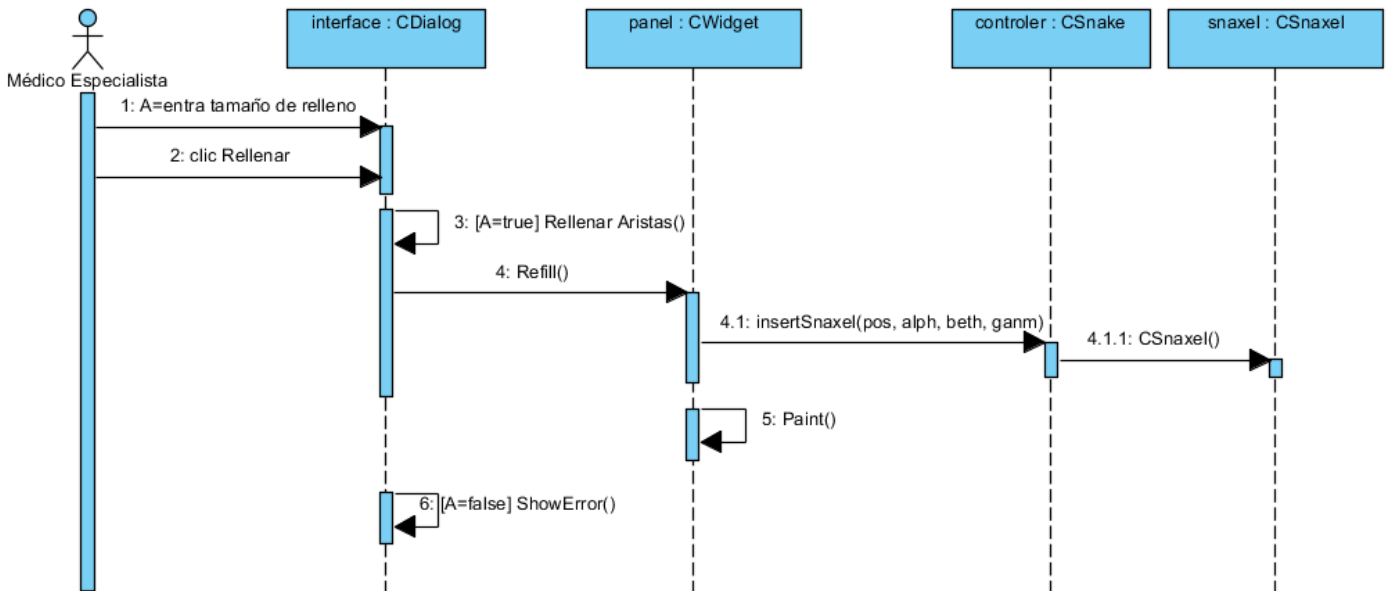


Fig. 26 Diagrama de Secuencia de la sección Rellenar Aristas del caso de uso Modificar Contorno.

3.6.5. Diagrama de Secuencia del caso de uso Segmentar Imagen.

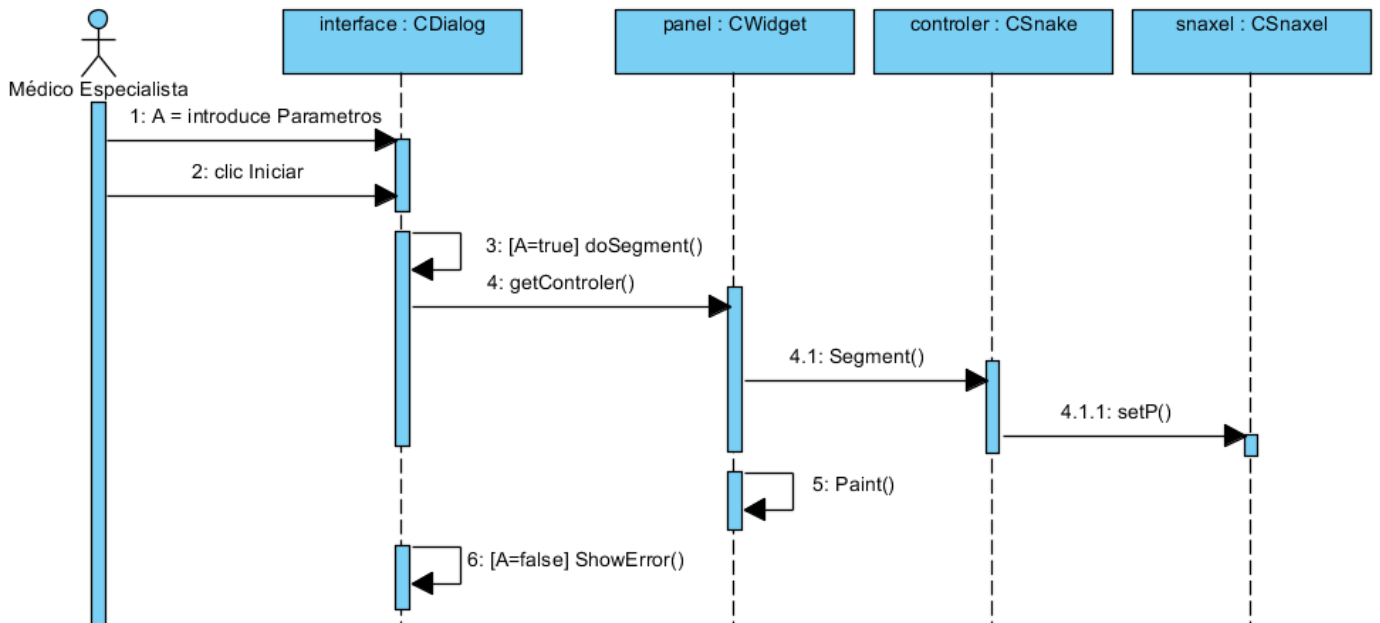


Fig. 27 Diagrama de Secuencia del caso de uso Segmentar Imagen.

Capítulo 4: Implementación y Validación de los Resultados

Este capítulo abarca los temas relacionados con la implementación del sistema y la validación de los resultados obtenidos con la aplicación. Primeramente, se define el diagrama de componentes del sistema desarrollado, para tener una idea más clara sobre la estructura del código fuente de la aplicación elaborada. Además se definen los casos de prueba para validar los resultados obtenidos, estos se expresan mediante imágenes que permiten hacer comparaciones luego de ejecutar los procesos en la aplicación. También se hace un análisis de la influencia de algunos parámetros como *alpha* y *betha* en la calidad de la segmentación. Otros aspectos que se tienen en cuenta para el proceso de validación son la calidad del filtro de reducción de ruido seleccionado y cómo afecta la cantidad de vértices del contorno a los resultados de la segmentación de las estructuras de interés.

4.1. Implementación

En esta etapa se realiza la implementación de las clases y objetos en ficheros fuente, binarios y ejecutables. Como resultado se obtiene un sistema ejecutable que incluye todas las funcionalidades propuestas en la captura de requisitos funcionales. La estructura de estos componentes forma el modelo de implementación.

4.1.1. Diagrama de Componentes

Las clases que se obtienen en el diseño se hacen físicas mediante componentes. Los componentes representan módulos de software (código fuente, código binario, ejecutables, DLL). Para organizar mejor las dependencias de la aplicación propuesta se elabora el diagrama de componentes, este se muestra en la Fig. 28. Este diagrama incluye fundamentalmente componentes de código fuente y ejecutables.

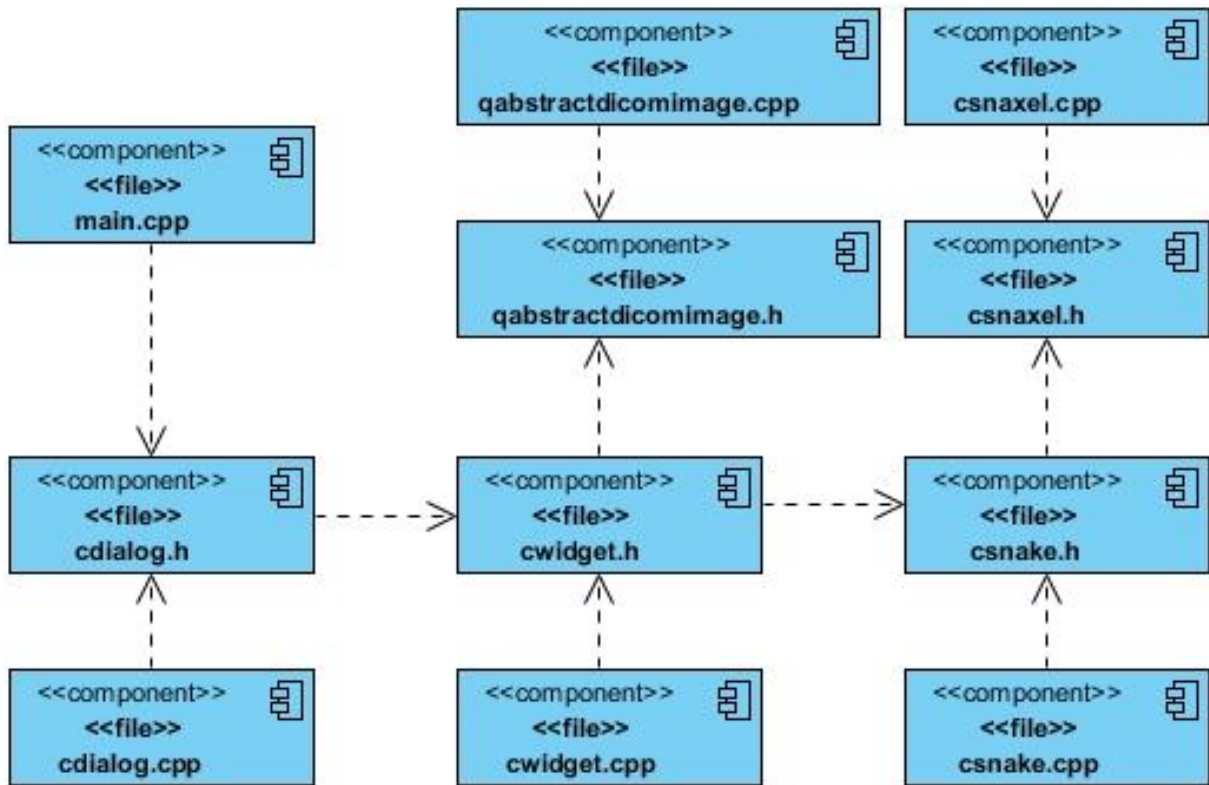


Fig. 28 Diagrama de Componentes.

4.2. Validación de los resultados

En este epígrafe se muestran los resultados que se obtienen con el desarrollo del módulo. Los mismos permiten mostrar los análisis y la discusión del método de segmentación propuesto. Todas las pruebas se realizan sobre una computadora personal con un procesador AMD Dual-Core E450 a una frecuencia de 1.6GHz, 4 GB de memoria RAM DDR3 y tarjeta gráfica AMD Radeon HD 6320 con 256 MB de RAM para video donde se alcanzan resultados positivos.

4.2.1. Datos de Entrada

Para la realización de las pruebas se utilizan imágenes médicas en formato DICOM obtenidas a través de las modalidades de adquisición de imágenes de TAC y RM. A continuación se presenta la descripción de las imágenes de prueba utilizadas.

Tabla 12 Caso de prueba #1.

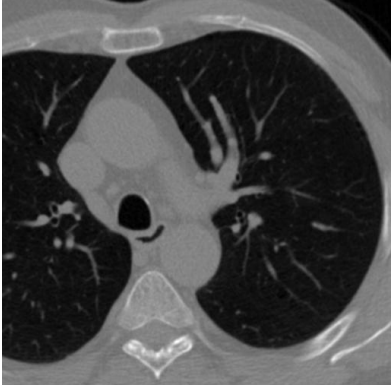

Identificador	Modalidad	Dimensiones	Imagen
Corazón de un paciente.	TAC	512x512	

Tabla 13 Caso de prueba #2.

Identificador	Modalidad	Dimensiones	Imagen
Cráneo de un paciente.	RM	512x512	

Resultados obtenidos de la carga de imágenes

A continuación se muestran las imágenes que comprueban la posibilidad de cargar imágenes DICOM del módulo desarrollado.

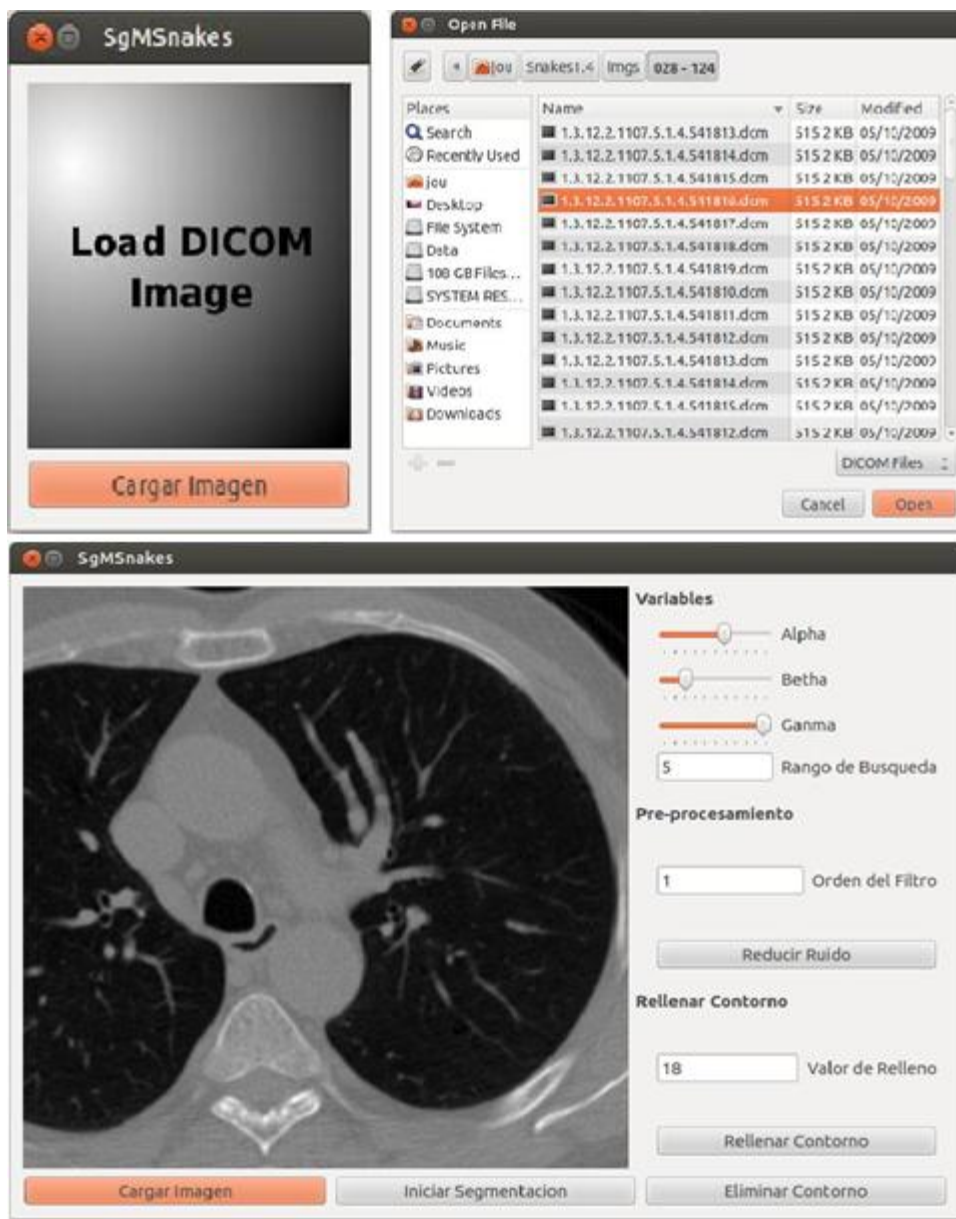


Fig. 29 Superior Izquierda: Interfaz inicial. Superior derecha: Cuadro de diálogo de búsqueda del fichero DICOM a cargar. Inferior: Aplicación con la imagen cargada.

Resultados obtenidos en la etapa de pre-procesamiento.

Otra de las pruebas realizadas fue para medir el nivel de reducción de ruido del filtro de la Mediana. Con este objetivo se aplicó el filtro de la Mediana sobre varias imágenes médicas, en todos los casos con una máscara de 5x5 y varias iteraciones. Los resultados obtenidos se muestran en la Fig. 30.

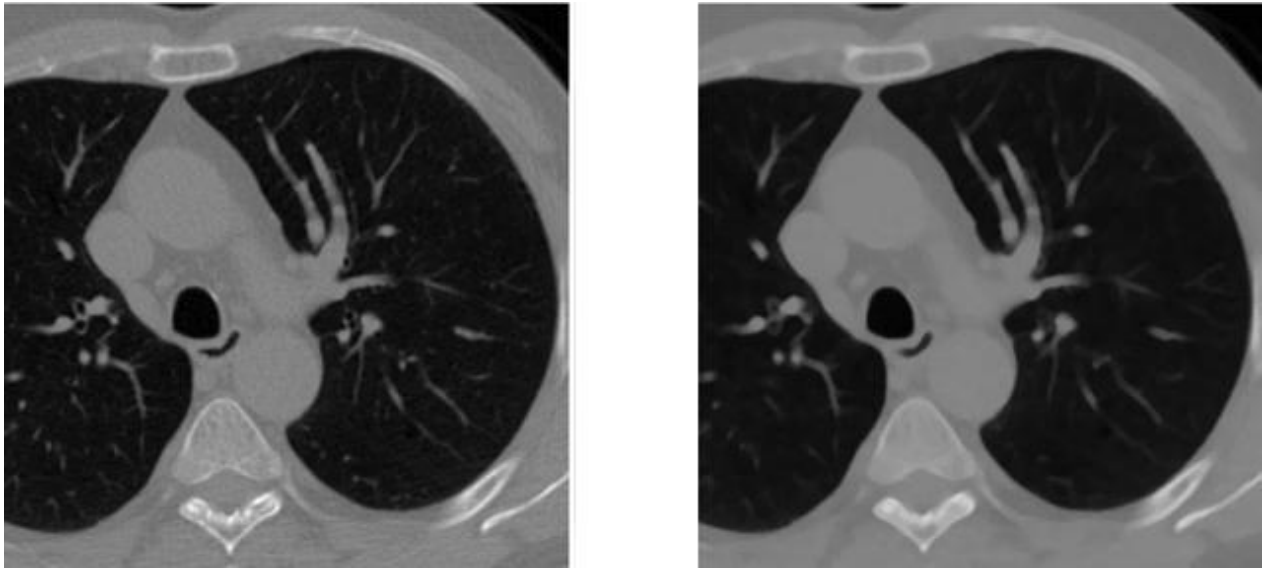


Fig. 30 Izquierda: Imagen inicial. Derecha: Imagen después aplicado el filtro de la Mediana.

En las imágenes se aprecia como el filtro implementado aporta un alto nivel de reducción de ruido y no empaña los bordes de la imagen, siendo esta la característica principal por la que fue elegido.

Resultados obtenidos de la generación y modificación del contorno

En la siguiente imagen se muestra el contorno de segmentación creado inicialmente sobre la estructura objetivo y los resultados de aplicar el algoritmo de relleno de vértices. Se puede apreciar como este algoritmo facilita la generación de contornos con más vértices sin la necesidad de hacerlo manualmente. En la Fig. 31 se muestra como el incremento de la cantidad de vértices facilita la gestión del contorno definido.

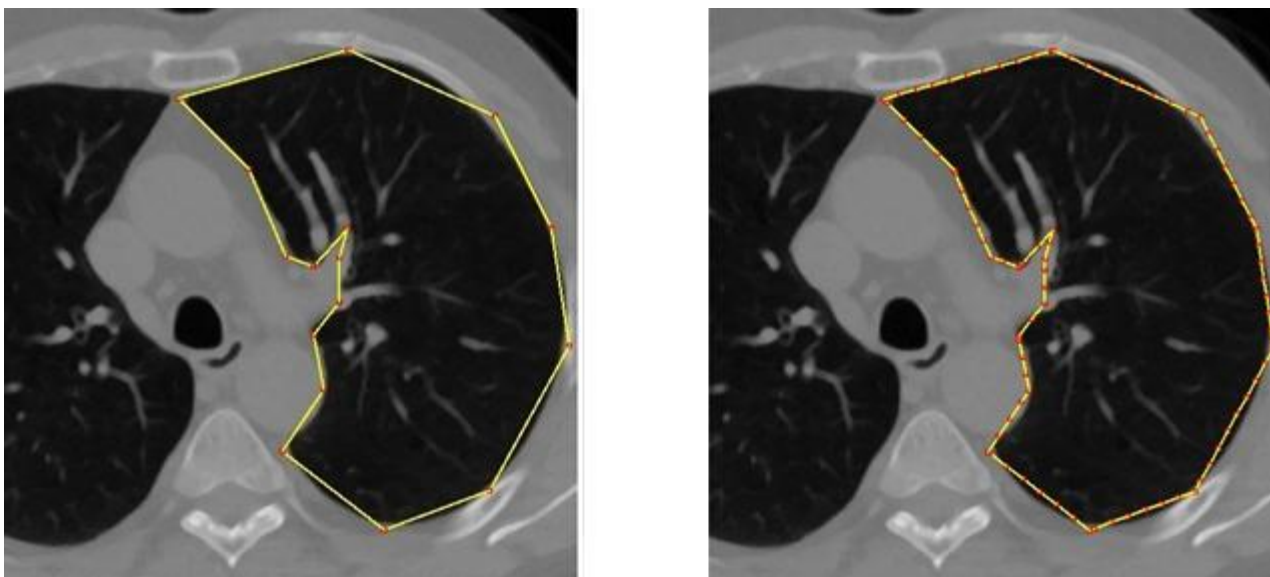


Fig. 31 Izquierda: Imagen mostrando el contorno creado sobre la estructura a segmentar. Derecha: Imagen mostrando el contorno una vez realizado el algoritmo de relleno de vértices.

Resultados obtenidos de la segmentación

En las siguientes imágenes se muestran los resultados alcanzados con el algoritmo de segmentación sobre diferentes imágenes para evaluar cómo influyen algunos elementos en el comportamiento del *Snake*. Los elementos que se toman en consideración para evaluar la respuesta del algoritmo son: segmentar imágenes pre-procesadas con el filtro de la Mediana, variar los valores *alpha*, *betha* y *gamma* e incrementar la cantidad de vértices del contorno. En las siguientes imágenes se muestran resultados de segmentación alcanzados usando contornos con diferentes cantidades de vértices.

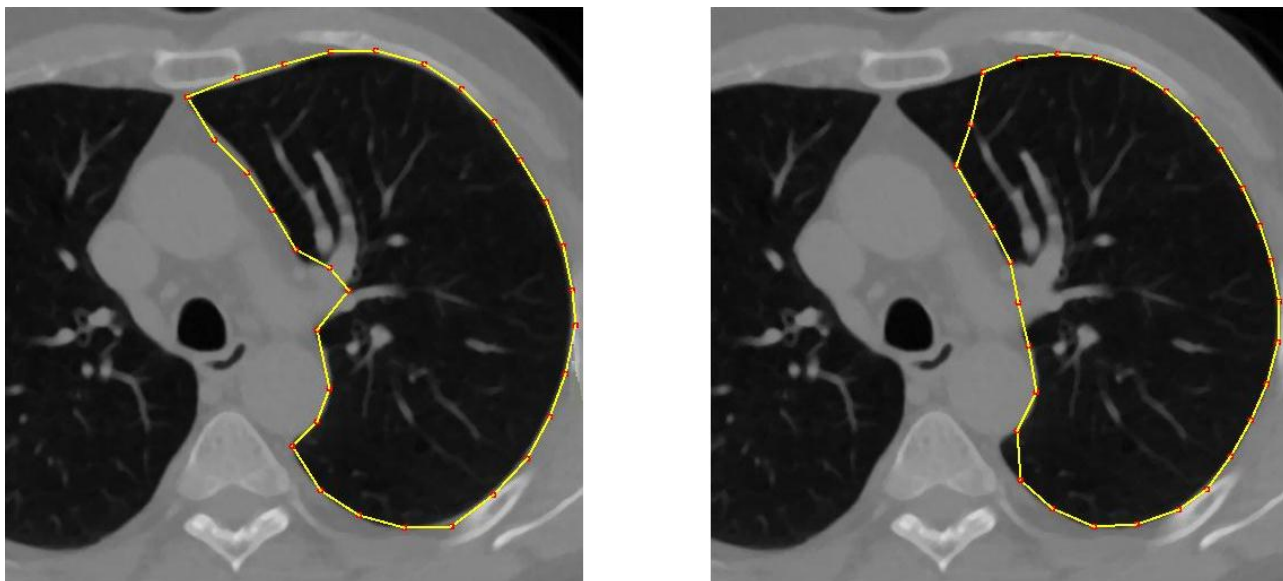


Fig. 32 Izquierda: Contorno inicial con pocos vértices sobre el área que se desea segmentar. Derecha: Resultado de la segmentación.

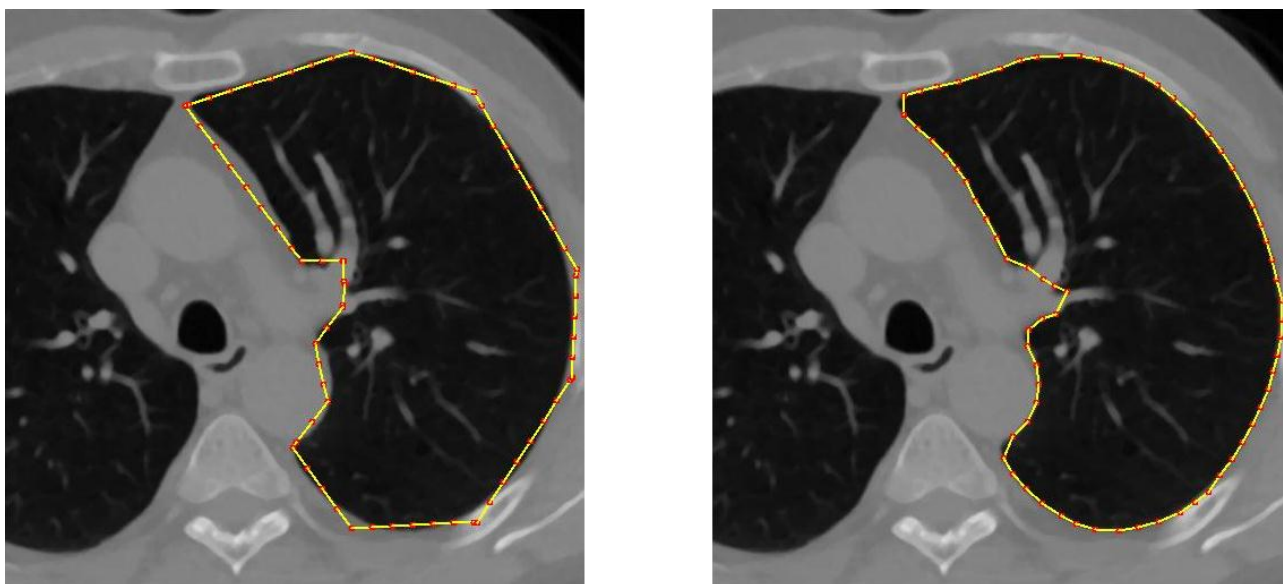


Fig. 33 Izquierda: Contorno inicial al que se le aplico el algoritmo de relleno de vértices sobre el área que se desea segmentar. Derecha: Resultado de la segmentación.

Se puede observar que cuando se usa el algoritmo de relleno de aristas para obtener contornos con más vértices los resultados de la segmentación son más exactos.

Para validar el algoritmo de segmentación propuesto se aplica el método de *Snakes* sobre imágenes previamente filtradas por el método de la Mediana, utilizando una máscara de 5x5 con una sola iteración. En la Fig. 34 se aprecian los resultados de la segmentación.

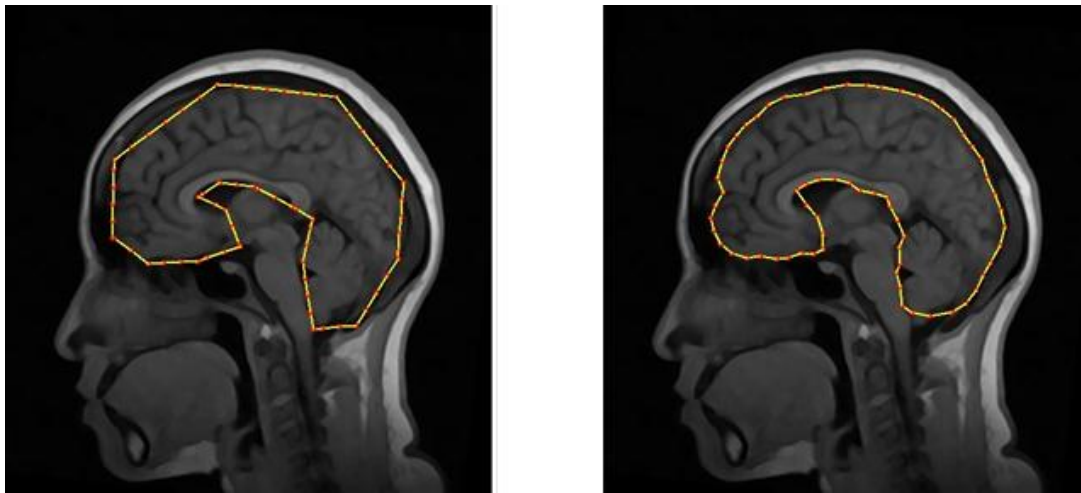


Fig. 34 Izquierda: Contorno inicial sobre el área del cerebro. Derecha: Resultado de la segmentación.

El filtro de la Mediana demuestra que es ideal para el funcionamiento del algoritmo de segmentación propuesto porque al no provocar pérdidas sobre los bordes de la imagen hace posible que el *Snakes* se ajuste correctamente sobre la estructura de interés que se desea segmentar.

La variación de los parámetros *alpha*, *betha* y *ganma* puede influir en gran medida en el resultado de la segmentación en dependencia de los valores de estos términos. Esto se evidencia en otra de las pruebas realizadas. Se aplica el algoritmo *Snakes* sobre la misma imagen pero con diferentes valores de *alpha*, *betha* y *ganma*. En la Fig. 35 se muestra el resultado de la segmentación obtenida con los siguientes valores: *alpha* = 0.0, *betha* = 0.0 y *ganma* = 1.0.

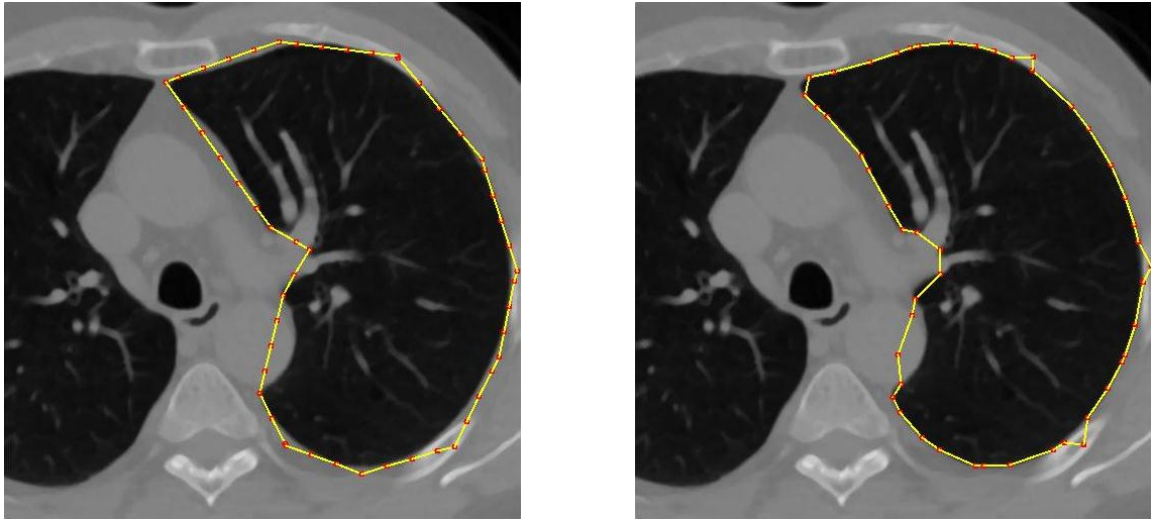


Fig. 35 Izquierda: Contorno inicial sobre el área que se desea segmentar. Derecha: Resultado de la segmentación.

Para estos valores definidos anteriormente la segmentación aporta resultados irregulares ya que no se definen parámetros de fuerzas internas en el algoritmo, generando la aparición de picos y provocando una escasa relación entre los vértices que definen el contorno.

En la Fig. 36 se muestra el resultado de la segmentación obtenida con los siguientes valores: $\alpha = 1.0$, $\beta = 1.0$ y $\gamma = 1.0$.

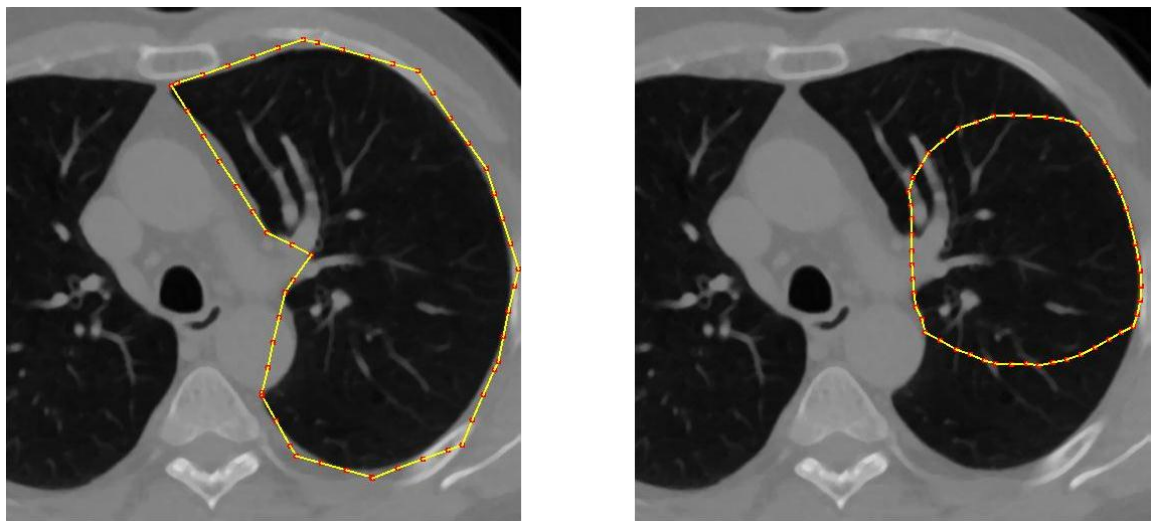


Fig. 36 Izquierda: Contorno inicial sobre el área que se desea segmentar. Derecha: Resultado de la segmentación.

En esta imagen se observa que valores muy elevados de las fuerzas internas del *Snake* genera contornos demasiado homogéneos que no se corresponden con la estructura real que se desea segmentar y que constituyen una segmentación incorrecta de la estructura.

Es por eso que la elección correcta de los parámetros de acuerdo a la imagen que se quiere segmentar es un tema muy importante que determina en gran medida la precisión y la calidad de la segmentación que se obtiene con el algoritmo propuesto. Para la mayoría de las imágenes que se probaron, los valores *alpha*, *betha*, *ganma* y el rango de búsqueda de cada *snaxe/* que mostraron los mejores resultados fueron 0.6, 0.2, 1.0 y 11 respectivamente.

Conclusiones

- Se desarrolló un módulo que permite segmentar imágenes médicas en formato DICOM a partir de un contorno inicial definido por el usuario.
- Se comprobó que la eficacia del algoritmo *Snakes* depende de la correcta elección de los parámetros *alpha*, *betha* y *ganma* para poder obtener una correcta segmentación de la estructura de interés objetivo.
- El filtro de la Mediana en la etapa de pre-procesamiento mostró buenos resultados porque permitió reducir el ruido sin provocar pérdidas de información en los bordes de la imagen.
- El algoritmo para rellenar las aristas facilitó el proceso de creación del contorno inicial y permitió segmentar estructuras con mayor precisión.

Recomendaciones

- Desarrollar un algoritmo de segmentación de modelos anatómicos tridimensionales utilizando los modelos deformables basados en los principios de los *Snakes*, con el objetivo de visualizar estructuras completas.
- Adicionar una funcionalidad para permitir segmentar varias estructuras en una misma imagen médica.

Bibliografía

1. **Coto, Ernesto.** *Métodos de Segmentación de Imágenes Médicas.* 2005.
2. **C. Gonzalez, Rafael y E. Woods, Richard.** *Digital Image Processing.* New Jersey : Prentice Hall, 2002.
3. *La segmentación de imágenes el método de los contornos activos geométricos.* **Frangi Caregnato, Alejandro y Caselles Costa, Vicent.** s.l. : Fundación General de la Universidad Autónoma de Madrid, 2006. 1139-9325.
4. **González Zaldivar, Alcides.** *Segmentación de Imágenes Médicas, mediante Técnicas de Contornos y Técnicas de Regiones.* 2010.
5. **Preim, Bernhard y Bartz, Dirk.** *Visualization in Medicine Theory, Algorithms, and Applications.* s.l. : Elsevier, 2007. 978-0-12-370596-9.
6. **Chaple, Andres Miralles.** *Módulo de Segmentación de Imágenes Médicas Digitales Mediante Técnicas de Watershed.* Habana : s.n., 2011.
7. **Villamizar, Andrea .** CREA IMAGEN. [En línea] <http://creaimagenandrepUBLICIDAD.blogspot.com/>.
8. **Tovar, Luis Carlos Carrillo.** *Sistema de reconocimiento de objetos mediante imágenes aplicado a un brazo robotico.* México, D.F. : s.n., 2009.
9. **Barzaga, Osvaldo Pereira y Kindelan Nuñez, Rolando.** *Reconstrucción Tridimensional de Modelos Anatómicos a partir de Imágenes Médicas Digitales.* . Habana : s.n., 2008.
10. **A. Singh, D. Goldgof, D. Terzopoulos.** *Deformable Models in Medical Image Analysis.* Los Alamitos, CA : s.n., 1998. 0-8186-8521-2.

11. **Bebis, George.** [En línea] <http://www.cse.unr.edu/~bebis/CS791E/Notes/DeformableContours.pdf>.
12. **CALA(Campus Libre y Abierto).** Epistemowikia Revista «Hiperenciclopédica» de Divulgación del Saber.[En línea] http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Contorno_activo_borroso.
13. **Cisneros Sanchez, Giovana Patricia, Luyo Pineda, Miguel Angel y Nishiguchi Mendoza, Juan José.** *Procesamiento de Imagenes.* 2010.
14. **Qt, Zona.** Introducción a Qt. [En línea] <http://www.zonaqt.com/tutoriales/tutorial-de-qt-4-por-zona-qt-0>.
15. **Software, Black Duck.** ohloh Committed Code. [En línea] 2012. <http://www.ohloh.net/p/dcmk>.
16. **Ayuda de Rational Unified Process.**
17. **Díaz Flores, Miriam Milagro.** RUP vs. XP. [En línea] [http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP vs. XP.pdf](http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP_vs_XP.pdf).
18. **Electronica, Laboratorio de.** Proceso Unificado para Desarrollo de Software. [En línea] <http://www.geocities.ws/gustsucc/Archivos/AnotacionesRUP.pdf>.
19. **Acharya, Tinku y K. Ray, Ajoy.** *Image Processing: Principles And Applications.* s.l. : illustrated, 2005. 0471719986, 9780471719984.
20. **Kass, Michael, Witkin, Andrew y Trezopolous, Demetri.** *Snakes: Active Contour Models.* Palo Alto : s.n., 1987.
21. **Banda, Adam.** *Implementating a Deformable Model Image Segmentation Algorithm for a Multi-core Microprocessor Architecture.* 2007.
22. **Criado Fernández, Luis.** *Identificación de la papila usando el algoritmo de contornos activos.* 2003.

23. **Wang, Yonggang, Zhu, Yun y Guo, Qiang.** *Medical Image Segmentation Based on Deformable Models and its Applications.* 2007.
24. **Juliana Gambini, María.** *Modelos de Segmentación basados en Regiones y Contornos Activos aplicados a Imágenes de Radar de Apertura Sintética.* Buenos Aires: s.n., 2006.
25. **Carzola, Miguel Angel, y otros.** *Detección Automática con Snakes y Representación 3D sobre Imágenes Cerebrales.* 1995.
26. *Reducción de ruido en imagenes de tomografía computarizada usando un filtro bilateral anisotrópico.* **Ramirez Giraldo, Juan Carlos, J. Fletcher, Joel y H. McCollough, Cynthia.** 7, Colombia : s.n., 2010, Vol. 4. 1909-9762.
27. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison Wesley, 2000. 0-201-57169-2.

Glosario de Términos

A

Algoritmo: Es una lista que, dado un estado inicial y una entrada, propone pasos sucesivos para arribar a un estado final obteniendo una solución.

B

Borde: En una imagen, es el lugar en el que coinciden dos formas. Por ejemplo, la unión del cielo con la tierra. La línea que separa dos formas, o una forma y un espacio.

C

Contorno: Conjunto de las líneas que limitan una figura o composición. Ver frontera.

D

Diagnóstico: Etimológicamente el concepto diagnóstico proviene del griego, tiene dos raíces, día- que es a través de, por. y gignoskein que es conocer, así etimológicamente diagnóstico significa conocer a través de. El concepto de este significado (imagen que representamos en la mente) es la identificación de la naturaleza o esencia de una situación o problema y de la causa posible o probable del mismo, es el análisis de la naturaleza de algo.

DICOM: En Inglés (Digital Imaging and Communication in Medicine) es el estándar reconocido mundialmente para el intercambio de imágenes médicas, pensado para el manejo, almacenamiento, impresión y transmisión de imágenes médicas. Incluye la definición de un formato de fichero y de un protocolo de comunicación de red. El protocolo de comunicación es un protocolo de aplicación que usa TCP/IP para la comunicación entre sistemas. Los ficheros DICOM pueden intercambiarse entre dos entidades que tengan capacidad de recibir imágenes y datos de pacientes en formato DICOM.

G

Gradiente: Denota una dirección en el espacio según la cual se aprecia una variación de una determinada propiedad o magnitud física.

H

Hardware: Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

I

Imagen: Figura, representación, semejanza y apariencia de algo. En computación es formada por la unión de MxN píxeles (imagen 2D) o vóxeles (imagen 3D).

Imagenología: Comprende la realización de todo tipo de exámenes diagnósticos y terapéuticos en los cuales se utilizan equipos que reproducen imágenes del organismo. Los siete servicios de Imagenología son Ecotomografía, Imagenología Mamaria, Medicina Nuclear, Radiología, Rayos Infantil, Resonancia Magnética y Tomografía Computada o Scanner.

M

Módulo: Pieza o conjunto unitario de piezas que se repiten en una construcción de cualquier tipo, para hacerla más fácil, regular y económica.

P

Patología: Conjunto de síntomas de una enfermedad.

Píxel: Abreviatura de “picture element”. Es la mínima unidad de información dentro de una imagen bidimensional.

R

RAM: Es la memoria de acceso aleatorio (en inglés: random access memory), desde donde el procesador recibe las instrucciones y guarda los resultados. Es el área de trabajo para la mayor parte del software de un computador.

Reconstrucción 3D: Es el proceso mediante el cual objetos reales son reproducidos en la memoria de un computador, manteniendo sus características físicas (dimensiones, volumen y forma).

S

Segmentación: Se utiliza en el Procesamiento de Imágenes para el reconocimiento de objetos o estructuras de interés en la imagen.

Snakes: Proviene de la palabra del inglés snake que significa serpiente, nombre de un algoritmo de segmentación de imágenes basado en contornos activos.

Snaxel: Elemento del contorno de segmentación que utiliza para su funcionamiento el algoritmo Snakes.

V

Vértices: Son puntos en el espacio 3D que definen primitivas gráficas tales como triángulos, polígonos y rectángulos, usados para construir la geometría de la escena que será dibujada.