

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 5**



**Módulo de filtrado y segmentación de imágenes  
médicas digitales para el proyecto Vismedic.**

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

**Autor: Adrián Peña Peñate.**

**Tutor: M.Sc. Osvaldo Pereira Barzaga.**

**Co-tutores: Ing. Rubén Alcolea Núñez.**

**Ing. Luis Guillermo Silva Rojas.**

**La Habana, Junio de 2013**

## 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 firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Adrián Peña Peñate.  
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:** 28.

**Ciudadanía:** Cubano.

**Institución:** Instituto de Cibernética, Matemática y Física (ICIMAF).

**Título:** M.Sc. en Informática Aplicada.

**Categoría Docente:** Instructor.

**E-mail:** opereira@icimaf.cu.

M.Sc. Osvaldo Pereira Barzaga. Graduado en el año 2008 como Ing. en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI). Con 8 años de experiencias en los temas de visualización médica y procesamiento digital de imágenes, desde el 2008-2009 se desempeña como líder del Proyecto Simulador Quirúrgico. En el 2009-2011 es el líder del área temática de visualización científica y líder del proyecto Vismedic. En el año 2010 se gradúa de máster en la disciplina de Informática Aplicada. Durante el 2012 cumple misión internacionalista en Venezuela, desempeñándose como arquitecto general del software GALBA-CAD, desarrollado por el proyecto, Centro de Diseño y Simulación de Estructuras Mecánicas (CDSEM), en el marco de la empresa mixta entre Petróleos de Venezuela Sociedad Anónima (PDVSA) y Guardián del Alba. Actualmente posee la categoría docente de instructor y trabaja en el grupo de procesamiento de imágenes digitales del Departamento de Matemática Disciplinaria en el Instituto de Cibernética, Matemática y Física (ICIMAF).

**Co-tutor:** Ing. Luis Guillermo Silva.

**Edad:** 25.

**Ciudadanía:** Cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ing. en Ciencias Informáticas.

**Categoría Docente:** -.

**E-mail:** lgsilva@uci.cu.

Graduado en la UCI como Ingeniero en Ciencias Informáticas en el año 2011. Tiene cuatro años de experiencia en procesamiento de imágenes digitales y visualización de volúmenes. Actualmente es Jefe del

Proyecto Vismedic, perteneciente a la Línea Visualización Científica del centro CEDIN. Es miembro del Grupo de Investigación de Visualización y Realidad Virtual (ViViRG) de dicho centro.

**Co-tutor:** Ing. Rubén Alcolea Núñez.

**Edad:** 25.

**Ciudadanía:** Cubano.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ing. en Ciencias Informáticas.

**Categoría Docente:** -.

**E-mail:** ralcolea@uci.cu.

Graduado como Ing. en Ciencias Informáticas en la UCI en el año 2011. Tiene cuatro años de experiencias en el procesamiento digital de imágenes y la visualización médica. Actualmente es arquitecto del proyecto Vismedic, perteneciente a la Línea Visualización Científica del centro CEDIN. Es miembro del Colectivo de Entrenadores del Movimiento ACM-ICPC en la UCI.

## Dedicatoria

### **A mi madre:**

*Mi espíritu iluminador de caminos, que aunque no esté físicamente, siempre ha estado a mi lado protegiéndome de todo mal, con las mejores intenciones del mundo.*

### **A mi papá:**

*Mi amigo, mi ejemplo, mi guía en los momentos difíciles, a veces mi madre físicamente. Siempre ha estado cuidándome y enseñándome de todo, puedo decir que mucho de lo que se lo debo a él.*

### **A mi hermano:**

*A quien acompañaré siempre, en los momentos difíciles y alegres, siempre juntos. A donde sea que vaya siempre te llevaré, y predicaré con el ejemplo.*

### **A mis hermanas:**

*Mis compañeras de la vida, mi ayuda a salir de las angustias, hermanas por siempre digan lo que digan, piensen lo que piensen.*

### **A mis abuelos:**

*A los que de una manera u otra han dado la oportunidad de que mis padres me dieran lo que más necesitaba, cariño y amor. Todos siempre están en mi corazón.*

## Agradecimientos

*Quisiera agradecer a todas las personas que de una manera u otra han contribuido a mi formación como profesional y como ser humano, aunque las palabras no son suficientes para demostrar un agradecimiento.*

*Agradezco a mi familia por el apoyo que me han brindado para poder llegar a cumplir mis metas, agradezco a mis tíos y tías en especial a mi tía Meri, que ya no está con nosotros en vida, a Mayito, a mi tía Maritza, que gracias a ella vivo. A mi prima Yanneris y a su marido Daniel por brindarme tecnología para mi carrera, como a mi primo Arián. A mi sobrino Fabian que aunque aún es pequeño me ha inspirado alegría y deseos de continuar. A Duniel, el marido de Kirenía por sus criterios de temas relacionados con la carrera y a su primo Javier, un amigo que llegó para quedarse. A mis compañeros que me han rodeado día a día, aprendiendo algo nuevo. Le agradezco a Yoandy Gutiérrez por ser mi amigo de todos los tiempo, los buenos y los malos, para todo, siempre está presente. Le agradezco a Lianet, mi novia actualmente por su comprensión en momentos difíciles.*

*Agradecer a todas las personas que me han ayudado en la confección de este trabajo a pesar de no tener que hacerlo como, Alina por ayudarme a aclarar temas de ingeniería, a Madonna y Amal por sus criterios, Adonis, Livan y Elvis, que aunque no lo crean los compañeros son importantes para despejar la mente. Le agradezco a Alejandro (el Chino, mi padrino) y a su novia Mairelys por darme aliento en momentos súper difíciles, que pasé en la Universidad.*

*Agradezco a alguien que desde primer año me captó y la guerra formó para tenerme en el proyecto, a mi tutor Osvaldo incansable e infatigable amigo y compañero que aunque en la UCI no se encontraba me siguió apoyando y guiando en el trabajo. Al Guille y Alcolea, que su apoyo fue incondicional y meritorio, ellos fueron más que co-tutores, fueron amigos y hermanos en lo personal y en la guía del trabajo, las noches y los horarios eran interminables.*

## Resumen

El avance tecnológico de los dispositivos de exploración radiológica para la obtención de imágenes médicas digitales y el avance de los sistemas de cómputo, permite mejorar la calidad de visualización a través de algoritmos de filtrado y segmentación. Permitiendo seleccionar estructuras anatómicas de interés de las imágenes médicas digitales para un mejor diagnóstico clínico.

En el presente trabajo se propone un módulo que encapsula varias técnicas de filtrado y segmentación de imágenes médicas digitales utilizando la biblioteca *Insight Segmentation and Registration Toolkit* (ITK), con el objetivo de mejorar la calidad de la visualización de estas imágenes. El módulo propuesto se independiza de las etapas de Reconstrucción y Visualización de estructuras tridimensionales del proyecto Vismedic, a través de la transformación de los datos adquiridos para su procesamiento y su devolución en el mismo tipo de dato que se recibió.

Como resultado se obtuvo una aplicación con una interfaz de comunicación común que brinda la posibilidad de agregar nuevos algoritmos de filtrado y segmentación para extender la aplicación sin modificar otras etapas de la visualización de las imágenes médicas digitales. Se realizaron comparaciones entre los algoritmos de filtrado y segmentación incorporados en la solución en cuanto a la calidad de la imagen final que se obtiene.

**Palabras clave:** DICOM, filtro, ITK, pre-procesamiento, segmentación.

## Índice de Contenido

<i>Declaración de Autoría</i> .....	<i>I</i>
<i>Datos del Contacto</i> .....	<i>II</i>
<i>Dedicatoria</i> .....	<i>IV</i>
<i>Agradecimientos</i> .....	<i>V</i>
<i>Resumen</i> .....	<i>VI</i>
<i>Índice de Contenido</i> .....	<i>VII</i>
<i>Índice de Figuras</i> .....	<i>X</i>
<i>Índice de Tablas</i> .....	<i>XII</i>
<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica</b> .....	<b>5</b>
<b>1.1 ¿Qué es Vismedic?</b> .....	<b>5</b>
<b>1.2 Imagen Digital</b> .....	<b>6</b>
1.2.1 Imagen médica digital.....	7
<b>1.3 Pre-procesamiento</b> .....	<b>7</b>
1.3.1 Reducción de ruido.....	8
Filtro Gaussiano.....	9
Filtro Promediador .....	9
Filtro de la Mediana .....	9
<b>1.4 Segmentación de imágenes</b> .....	<b>9</b>
1.4.1 Enfoques de la segmentación de imágenes .....	10
Segmentación manual .....	10
Segmentación automática .....	10
Segmentación semiautomática.....	11
<b>1.5 Algunas técnicas de segmentación desarrolladas en Vismedic</b> .....	<b>11</b>
1.5.1 <i>Clustering</i> .....	11
1.5.2 <i>Watershed</i> .....	12
1.5.3 <i>Snakes</i> .....	12
<b>1.6 Bibliotecas de clases ITK y MDSTk</b> .....	<b>13</b>
1.6.1 MDSTk .....	13
1.6.2 ITK.....	13
1.6.2.1 Algunos filtros de ITK.....	14
Umbral Binario .....	14
Umbral General.....	15
Filtro del Promedio.....	16



Filtro de la Mediana.....	17
Detección de Bordes <i>Canny</i> .....	18
1.6.2.2 Algunas técnicas de segmentación de ITK.....	19
Conectar umbral.....	19
Segmentación <i>Otsu</i> .....	20
Conectar Vecindad.....	21
<b>1.7 Conclusiones parciales.....</b>	<b>22</b>
<b>Capítulo 2. Solución Propuesta .....</b>	<b>23</b>
<b>2.1 Algoritmos seleccionados .....</b>	<b>23</b>
2.1.1 Umbral Binario ( <i>Binary Threshold</i> ) .....	23
2.1.2 Umbral General ( <i>General Threshold</i> ) .....	23
2.1.3 Promedio ( <i>Mean</i> ).....	24
2.1.4 Detección de Bordes <i>Canny</i> . ( <i>Canny Edge Detection</i> ) .....	24
2.1.5 Conectar el Umbral ( <i>Connected Threshold</i> ) .....	24
2.1.6 Segmentación <i>Otsu</i> ( <i>Otsu Segmentation</i> ) .....	24
2.1.7 Conectar Vecindad ( <i>Neighborhood Connected</i> ) .....	25
<b>2.2 Funcionamiento general del módulo.....</b>	<b>25</b>
<b>2.3 Patrón Estrategia.....</b>	<b>26</b>
<b>2.4 Herramientas de desarrollo .....</b>	<b>27</b>
<b>2.5 Lenguaje de modelado.....</b>	<b>28</b>
<b>2.6 Lenguaje de programación.....</b>	<b>28</b>
<b>2.7 Metodología de desarrollo de software.....</b>	<b>29</b>
<b>Capítulo 3. Características del Sistema.....</b>	<b>30</b>
<b>3.1 Reglas del negocio.....</b>	<b>30</b>
<b>3.2 Modelo de Dominio.....</b>	<b>30</b>
<b>3.3 Captura de requisitos .....</b>	<b>31</b>
3.3.1 Requisitos funcionales .....	32
3.3.2 Requisitos no funcionales.....	32
<b>3.4 Modelo de Casos de Uso del Sistema .....</b>	<b>33</b>
3.4.1 Actores del sistema .....	33
3.4.2 Diagrama de Casos de Uso del Sistema .....	34
3.4.3 Descripción de Casos de Uso del Sistema .....	35
<b>3.5 Diagrama de paquetes de clases .....</b>	<b>37</b>
<b>3.6 Diagramas de clases del paquete <i>vismedic_plugins</i>.....</b>	<b>40</b>
<b>3.7 Diagramas de Secuencia del Diseño .....</b>	<b>42</b>

<b>Capítulo 4. Implementación y Presentación de Resultados</b> .....	<b>45</b>
<b>4.1 Implementación</b> .....	<b>45</b>
4.1.1 Diagrama de Componentes. Modelo de Implementación en paquetes .....	45
<b>4.2 Presentación de los resultados</b> .....	<b>48</b>
4.2.1 Entrada de datos.....	48
4.2.2 Resultados obtenidos con la carga de imágenes .....	49
4.2.3 Resultados obtenidos al aplicar filtros a la imagen cargada.....	50
4.2.3.1 Resultados de aplicar el filtro Promedio.....	51
4.2.3.2 Resultados al aplicar el filtro Umbral Binario.....	51
4.2.3.3 Resultados de aplicar el filtro Umbral General.....	52
4.2.3.4 Resultados de aplicar el filtro de Detección de Bordes <i>Canny</i> .....	54
4.2.4 Resultados obtenidos al aplicar las segmentaciones a las imágenes cargadas .....	55
4.2.4.1 Resultados al aplicar la técnica de segmentación Conectar Umbral .....	55
4.2.4.2 Resultados al aplicar la técnica de Segmentación <i>Otsu</i> .....	56
4.2.4.3 Resultados al aplicar la técnica de segmentación Conectar Vecindad.....	57
4.2.1 Resultados obtenidos al aplicar filtros y segmentaciones a imágenes estandarizadas .....	57
<b>Conclusiones</b> .....	<b>61</b>
<b>Recomendaciones</b> .....	<b>62</b>
<b>Bibliografía</b> .....	<b>63</b>
<b>Glosario de Términos</b> .....	<b>65</b>

## Índice de Figuras

Figura 1. Etapas para la visualización tridimensional de estructuras anatómicas.....	5
Figura 2. Representación de una imagen digital.....	6
Figura 3. Izquierda: Recorrido del filtro sobre la imagen. Derecha: Máscara de convolución.....	8
Figura 4. Izquierda: Imagen original para aplicar <i>Clustering</i> . Derecha: Resultados del algoritmo <i>K-means</i> para 2 <i>clusters</i> .....	11
Figura 5. Izquierda: Imagen médica original. Derecha: Imagen médica segmentada mediante técnicas <i>Watershed</i> .....	12
Figura 6. Izquierda: Contorno inicial con pocos vértices sobre el área que se desea segmentar. Derecha: Resultado de la segmentación <i>Snakes</i> .....	13
Figura 7. Función de transferencia del filtro de umbral binario.....	15
Figura 8. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro de Umbral Binario.....	15
Figura 9. Efecto del filtro Umbral General en los modos: a) Modo Debajo, b) Modo Arriba, c) Modo Fuera.....	16
Figura 10. Efecto local del filtro Promedio en un caso bidimensional.....	17
Figura 11. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro Promedio.....	17
Figura 12. Efecto local del filtro la Mediana en un caso bidimensional.....	18
Figura 13. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro Mediana.....	18
Figura 14. Izquierda: Imagen original. Derecha: Efecto del filtro de Detección de Borde <i>Canny</i> .....	19
Figura 15. Izquierda: Imagen original. Derecha: Efecto de la segmentación <i>Otsu</i> .....	21
Figura 16. Flujo de datos del módulo propuesto.....	25
Figura 17. Representación del Patrón Estrategia.....	26
Figura 18. Representación del Patrón Estrategia en el módulo.....	27
Figura 19. Modelo de Dominio.....	31
Figura 20. Diagrama de Casos de Uso del Sistema.....	34
Figura 21. Vista general del diagrama de paquete de clases y sus dependencias.....	38
Figura 22. Vista específica del diagrama de paquete de clases y sus relaciones.....	39
Figura 23. Diagrama de clases del subpaquete <i>plugin_Openltk</i> .....	40
Figura 24. Diagramas de clases del subpaquete <i>plugin_Filters</i> .....	41
Figura 25. Diagramas de clases del subpaquete <i>plugin_Segmentations</i> .....	42
Figura 26. Diagrama de Secuencia del caso de uso Cargar Imagen DICOM con ITK.....	43
Figura 27. Diagrama de Secuencia del caso de uso Filtrar Imagen DICOM.....	43
Figura 28. Diagrama de Secuencia del caso de uso Segmentar Imagen DICOM.....	44
Figura 29. Diagrama de Componentes. Paquetes de Implementación.....	46
Figura 30. Diagrama de Componentes del subpaquete <i>plugin_Openltk</i> .....	46
Figura 31. Diagrama de Componentes del subpaquete <i>plugin_Filters</i> .....	47
Figura 32. Diagrama de Componentes del subpaquete <i>plugin_Segmentations</i> .....	48
Figura 33. Cuadro de diálogo para cargar imagen.....	50
Figura 34. Imagen DICOM del cráneo de un paciente cargada con ITK.....	50
Figura 35. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Promedio.....	51
Figura 36. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral Binario.....	52
Figura 37. Izquierda: Imagen DICOM al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral Binario.....	52

Figura 38. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Debajo ( <i>Mode Below</i> ).....	53
Figura 39. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Arriba ( <i>Mode Above</i> ).....	53
Figura 40. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera ( <i>Mode Out Side</i> ).....	54
Figura 41. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro de Detección de Bordes <i>Canny</i> .....	55
Figura 42. Resultado de aplicar el filtro de Detección de Bordes <i>Canny</i> a una imagen DICOM, después de aplicar otros filtros.....	55
Figura 43. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.....	56
Figura 44. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la Segmentación <i>Otsu</i> . ....	56
Figura 45. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Vecindad. ....	57
Figura 46. Izquierda: Imagen Bárbara original. Derecha: Resultado de aplicar el filtro Umbral Binario. ....	58
Figura 47. Izquierda: Imagen Bárbara al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral Binario. ....	58
Figura 48. Izquierda: Imagen Lena original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera.....	59
Figura 49. Izquierda: Imagen Lena al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera.....	59
Figura 50. Izquierda: Imagen <i>Clown</i> original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.....	60
Figura 51. Izquierda: Imagen <i>Clown</i> al aplicar el filtro Promedio. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.....	60

## Índice de Tablas

Tabla 1. Ejemplos de los resultados aplicando el algoritmo Conectar Umbral. ....	20
Tabla 2. Ejemplos de los resultados aplicando el algoritmo Conectar Vecindad. ....	21
Tabla 3. Actores del Sistema.....	34
Tabla 4. Descripción del Caso de Uso Cargar Imagen DICOM con ITK. ....	35
Tabla 5. Descripción del Caso de Uso Filtrar Imagen DICOM. ....	36
Tabla 6. Descripción del Caso de Uso Segmentar Imagen DICOM. ....	37
Tabla 7. Funcionalidades de los paquetes de clases. ....	38
Tabla 8. Caso de prueba #1. ....	49
Tabla 9. Caso de prueba #2. ....	49
Tabla 10. Caso de prueba #3. ....	49

## Introducción

En los últimos años la tecnología ha tenido un avance circunstancial en todas las esferas de la sociedad. Una de estas esferas es la medicina, donde los médicos han podido diagnosticar a los pacientes mediante dispositivos de exploración radiológica, tales como Tomografía Axial Computarizada (TAC) e imágenes por Resonancia Magnética (RM), con la obtención de imágenes de zonas pertenecientes al interior del cuerpo del paciente de forma menos invasiva a las tradicionales, con importante información clínica.

El avance de estas tecnologías y los sistemas de cómputo, permite mejorar la calidad de la visualización de las imágenes médicas adquiridas. Actualmente se emplean técnicas para reducir el ruido producido durante la adquisición de imágenes, realzar zonas específicas y segmentar las imágenes en diferentes partes. Algunas de estas técnicas han sido desarrolladas por organizaciones académicas y comerciales, encapsulándolas en bibliotecas de clases, un ejemplo lo constituye la biblioteca de código abierto (*open source*) ITK, que se especializa en el procesamiento, segmentación y registro de imágenes [1].

En la facultad 5 de la Universidad de la Ciencias Informáticas (UCI), específicamente en el Centro de Informática Industrial (CEDIN), se desarrolla el proyecto de Visualización Médica (Vismedic) desde el año 2008. En dicho proyecto se desarrollan aplicaciones para la visualización y análisis tridimensional de datos médicos, obtenidos de las modalidades de adquisición de imágenes TAC y RM. Una de las etapas más importantes en la visualización médica es la identificación correcta de las estructuras anatómicas de interés, tema que se conoce en la bibliografía como “segmentación de imágenes”.

En el proyecto Vismedic, desde su fundación, se han desarrollado aplicaciones como el Visualizador 2D y el Visualizador 3D, en los cuales se han implementado algoritmos de filtrado y segmentación de imágenes, que se ejecutan antes de realizar el proceso de visualización. Actualmente, resulta engorroso el proceso de adición de nuevos algoritmos de filtrado y segmentación, debido que al introducir un nuevo algoritmo, es necesario modificar los algoritmos de visualización existentes en el proyecto Vismedic. Las deficiencias antes mencionadas, ocasionan reiterados atrasos en el cronograma del proyecto y disminuyen la capacidad de respuesta a la hora de adicionar nuevos requisitos funcionales.

Dada la situación problemática presentada anteriormente, se plantea como problema a resolver: ¿Cómo adicionar nuevos algoritmos de filtrado y segmentación de imágenes al proyecto Vismedic, sin modificar los algoritmos de visualización existentes?

Por tanto, se toma como **objeto de estudio**: el procesamiento de imágenes médicas digitales. Para darle solución al problema formulado, se plantea como **objetivo**: implementar un módulo de filtrado y segmentación de imágenes médicas digitales, que brinde una interfaz de comunicación común, para los algoritmos empleados en el proyecto Vismedic.

El **campo de acción** queda definido como: los algoritmos de filtrado y segmentación de imágenes médicas digitales. El presente trabajo defiende la siguiente idea: la implementación del módulo propuesto, permitirá adicionar nuevos algoritmos de filtrado y segmentación de imágenes sin necesidad de modificar el código de los algoritmos de visualización, lo que permitirá aumentar la extensibilidad del proyecto Vismedic.

Para dar cumplimiento a los objetivos planteados se establecen las siguientes tareas de investigación:

1. Elaboración del marco teórico a partir del estado del arte referente a los algoritmos de filtrado y segmentación de imágenes digitales.
2. Caracterización de los principales algoritmos de filtrado y segmentación empleados en el proyecto Vismedic.
3. Identificación de los algoritmos de filtrado y segmentación para el procesamiento de imágenes médicas digitales.
4. Diseño e implementación de una interfaz de comunicación común para los algoritmos de filtrado y segmentación.
5. Selección de los algoritmos de filtrado y segmentación para incorporarlos al módulo propuesto.
6. Adición de los algoritmos de filtrado y segmentación seleccionados.
7. Integración del módulo con las aplicaciones del proyecto Vismedic.

Para la realización de la investigación y elaboración del presente trabajo se emplearon varios **métodos científicos de investigación**, entre los cuales se pueden mencionar los siguientes:

## Métodos teóricos:

- **Histórico – Lógico:** Mediante este método se analizará la evolución y desarrollo de los algoritmos de filtrado y segmentación de imágenes médicas digitales y sus elementos más importantes.
- **Analítico – Sintético:** Se usará para analizar la información de las técnicas existentes para el filtrado y segmentación de imágenes.
- **Modelación:** Se empleará para realizar una representación simplificada de la realidad a través de diagramas de clases, de flujo y de componentes.

## Métodos empíricos:

- **Consulta de fuentes de información:** Método empírico utilizado para la consulta de las fuentes bibliográficas durante la investigación.
- **Observación:** Se empleará para constatar los resultados visuales alcanzados y determinar la influencia de estos sobre el rendimiento de la computadora.

**Capítulo 1. Fundamentación Teórica:** En este capítulo se mencionan los principales conceptos relacionados con el filtrado y segmentación de imágenes digitales y se caracterizan algunos algoritmos empleados frecuentemente.

**Capítulo 2. Solución Propuesta:** En este capítulo se presenta la descripción de la solución propuesta para la integración de los algoritmos de filtrado y segmentación. Se muestra, la definición del patrón de diseño de clases **Estrategia** para el módulo y el flujo de los algoritmos seleccionados. Se mencionan las herramientas y metodología de desarrollo, así como el lenguaje de modelado y de programación empleados en el presente trabajo.

**Capítulo 3. Características del Sistema:** En este capítulo se describe la solución propuesta desde el punto de vista de la ingeniería de software. Se muestran los principales artefactos generados, tales como: Modelo de Dominio, Diagrama de Casos de Uso y la descripción textual de cada caso de uso. Finalmente se presentan los diagramas de clases del diseño y los diagramas de secuencia del diseño.

**Capítulo 4. Implementación y Presentación de Resultados:** En este capítulo se presenta el Modelo de Implementación de la solución propuesta y se realiza un análisis de los resultados obtenidos empleando el módulo propuesto.



**Glosario de Términos:** Se elaboró un Glosario de Términos con el objetivo de facilitar la comprensión del lenguaje utilizado.

## Capítulo 1. Fundamentación Teórica

### 1.1 ¿Qué es Vismedic?

Vismedic es un proyecto desarrollado en la línea de Visualización Científica, perteneciente al Departamento de Visualización y Realidad Virtual del centro de CEDIN de la facultad 5 de la UCI. Este proyecto permite la visualización de estructuras anatómicas humanas adquiridas a partir de RM y TAC. Para una correcta visualización de estas estructuras, se siguen cuatro etapas fundamentales (ver Figura 1): Adquisición de datos, etapa donde se adquiere la información en forma de imágenes; Procesamiento de imágenes, este corresponde al proceso de filtrado y segmentación de las imágenes adquiridas; Reconstrucción tridimensional, se refiere al proceso de reconstruir, a partir de las imágenes adquiridas, un modelo tridimensional de las estructuras anatómicas de interés y Visualización tridimensional, esta etapa se encarga de la representación visual del modelo tridimensional obtenido. La presente investigación se centra en la etapa de Procesamiento de imágenes.

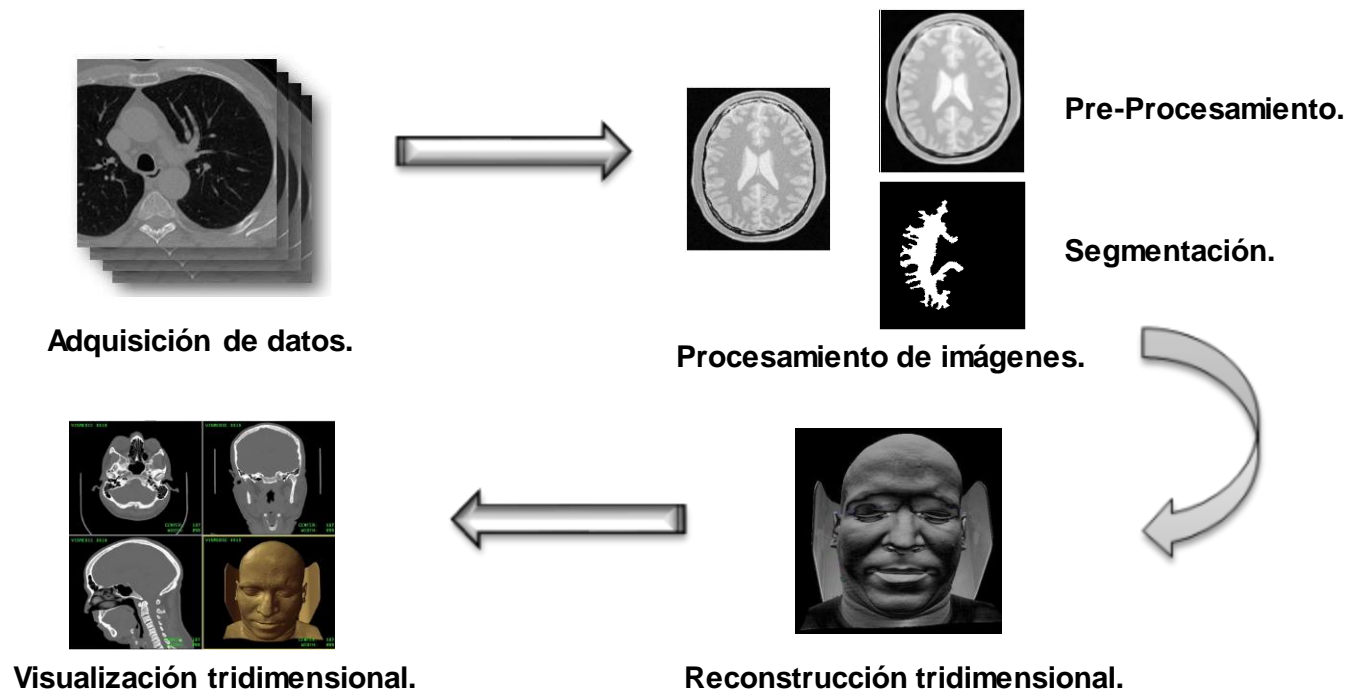


Figura 1. Etapas para la visualización tridimensional de estructuras anatómicas.

## 1.2 Imagen Digital

La imagen digital es la representación bidimensional de una imagen empleando bits, unidad mínima de información compuesta por dígitos binarios (0 y 1), que se emplea a instancias de la informática o cualquier dispositivo digital [26]. Se puede definir matemáticamente como una función bidimensional  $f(x,y)$ , donde  $f$  y los valores de  $(x, y)$  son cantidades finitas y discretas [1], que representan, el valor específico de un píxel dentro de la imagen (ver Figura 2) [2].

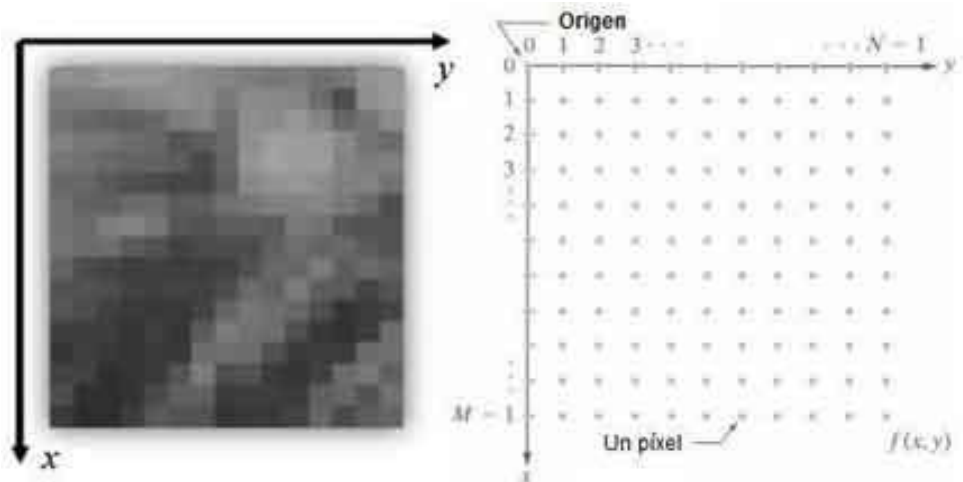


Figura 2. Representación de una imagen digital.

Una imagen digital puede ser representada por una matriz  $f(x,y)$  de tamaño  $M \times N$  de la siguiente forma:

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

Los elementos de la matriz (píxeles), en una imagen monocromática (niveles de gris), típicamente son del orden  $2^8$  o 256 niveles de gris, de tal manera que,  $f(x,y)$  puede estar en el rango de 0 a 255, donde el 0 representa el negro y 255 el blanco, por tanto, pueden ser representados como caracteres en la mayoría de los lenguajes de programación.

## 1.2.1 Imagen médica digital

Las imágenes médicas digitales se obtienen a partir de realizar procesos, y aplicar un conjunto de técnicas con el fin de crear imágenes del cuerpo humano o parte de este; aprovechando sus características químicas y físicas con propósitos clínicos (procedimiento médico que busca diagnosticar, relevar o examinar enfermedades) o para la ciencia médica [3].

Para la adquisición de imágenes médicas se destacan las modalidades TAC y RM. La TAC, es una exploración radiológica con rayos X, para ayudar al médico especialista a realizar un diagnóstico. Se denomina “axial”, porque con la TAC, se obtienen cortes transversales de una región u órgano específico del cuerpo, de forma perpendicular a un eje longitudinal [27]. A diferencia de las radiografías convencionales, que solo ofrecen una imagen plana, con la TAC, se obtiene un conjunto de imágenes en forma de finos cortes [27].

La RM es una técnica no invasiva que se utiliza para obtener información sobre la estructura y composición del cuerpo a analizar [3]. Es utilizada principalmente en medicina, para observar alteraciones en los tejidos, detectar cáncer u otras patologías. A diferencia de la TAC, no usa radiación ionizante, sino campos magnéticos, para alinear la magnetización nuclear de núcleos de hidrógeno del agua en el cuerpo.

## 1.3 Pre-procesamiento

Al digitalizar una imagen en ocasiones se introduce ruido o se degrada la imagen, lo que provoca pérdida de calidad. En el caso de las imágenes médicas se pueden ver afectadas por la respiración del paciente o por implantes metálicos en huesos o dientes, por citar algunos ejemplos. Por ello que antes de hacer cualquier análisis es fundamental mejorar la calidad de las imágenes [4]. Reducir el ruido presente en las imágenes conduce a la obtención de áreas más homogéneas que pueden ser delineadas con menor esfuerzo, eliminando información que no se considera relevante, aunque se puede comprometer la detección de pequeños elementos importantes [4]. Generalmente la mejora de la calidad de la imagen se hace en la etapa de pre-procesamiento, etapa que antecede al proceso de segmentación. El algoritmo o técnica de pre-procesamiento que se empleará dependerá del tipo de análisis que se vaya a realizar. Dentro de las técnicas más utilizadas se encuentran los algoritmos de suavizado, reducción de ruido, detección de bordes y técnicas híbridas [2].

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 está basado 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].

La mejora y tratamiento de la imagen en el dominio espacial se basa fundamentalmente en el uso de filtros. Básicamente un filtro o máscara de convolución, es un pequeño arreglo bidimensional (2D), como el que se muestra en la 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 [5]. El procedimiento consiste principalmente en mover la máscara punto a punto sobre la imagen. Estas técnicas son llamadas filtros o procedimientos de máscaras [5].

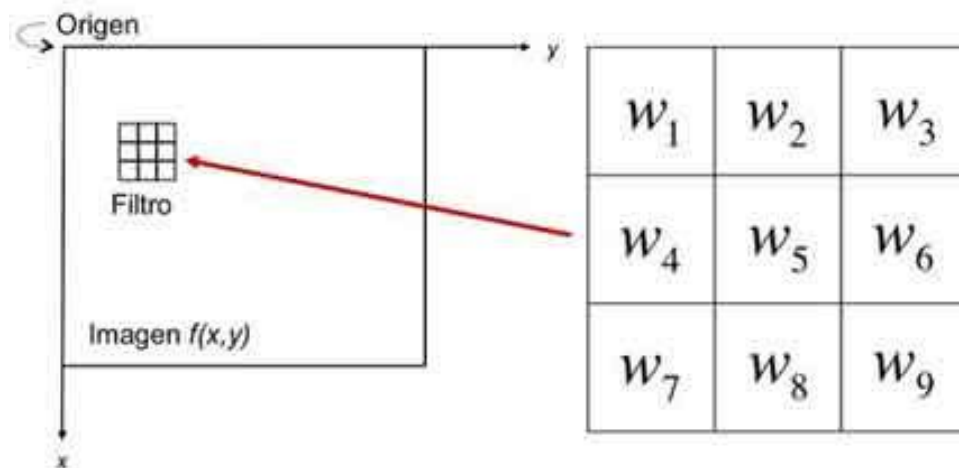


Figura 3. Izquierda: Recorrido del filtro sobre la imagen. Derecha: Máscara de convolución.

## 1.3.1 Reducción de ruido

Las imágenes médicas, obtenidas a través de diferentes instrumentos, requieren de métodos de filtrado, con la intención de ganar nitidez y conducir así a un mejor diagnóstico clínico [28]. Dentro de los filtros que se aplican a las imágenes con el fin de reducir el ruido o suavizarlas, en pos de lo que se quiera realizar con ellas, se encuentran el filtro Gaussiano, el filtro Promediador y el filtro de la Mediana [6].

## 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 ( $\sigma$ ) 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 [7]. 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 [7].

## 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 [7].

## Filtro de la Mediana

El objetivo de este filtro es la reducción del desenfoque de los bordes por medio de la sustitución del valor de gris del píxel analizado por la mediana de los valores de sus vecinos más cercanos, evaluados por el tamaño de la máscara aplicada. Tiene la ventaja de no ser afectado por píxeles individuales ruidosos y presenta una fácil implementación. La reducción de ruido obtenida con este filtro es superior al Promediador, con la ventaja de no desenfocar demasiado los bordes. Tiene como desventaja que su calidad disminuye cuando el número de píxeles con ruido es mayor a la mitad del número total de píxeles en la vecindad [7].

## 1.4 Segmentación de imágenes

La segmentación es la tarea de descomponer los datos de la imagen en estructuras significativas que son relevantes para tareas específicas [4]. Se encarga de asignarle índices a cada elemento de la imagen para representar su membresía a una estructura en particular. Existen diferentes enfoques de la segmentación dependiendo del nivel de automatización de los algoritmos utilizados [8].

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 [4]. Sin embargo, la delineación de los mismos es una tarea que requiere una precisión esencial, mejor obtenida con la utilización de medios computacionales. El nivel al que se realiza esta subdivisión depende de la aplicación en particular [4]. 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 [4].

## 1.4.1 Enfoques de la segmentación de imágenes

Para realizar el proceso de segmentación existen algoritmos que aportan diversos resultados, en dependencia de las imágenes con las que trabajen y su funcionamiento en particular. Los métodos de segmentación se pueden clasificar en dependencia de su nivel de interacción en: métodos manuales, semiautomáticos y automáticos. Cada uno presenta ventajas y desventajas que influyen a la hora seleccionarlos.

### Segmentación manual

Para lograr la segmentación consiste en dibujar manualmente las caras y elementos importantes de las estructuras presentes en las imágenes médicas. El usuario delinea con alguna herramienta las estructuras relevantes, lo que le da la ventaja de redibujar cualquier porción y corregir algún error introducido [4]. Este método es robusto pero consume mucho tiempo, cuando se trabaja con cúmulos de imágenes muy grandes se vuelve impracticable. La precisión de este método se ve afectada debido a que el usuario generalmente se desvía del contorno deseado y los objetos pueden ser difíciles de delinear y por tanto de segmentar [9].

### Segmentación automática

La segmentación automática, como su nombre lo indica, es el tipo de segmentación donde un 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 [9].

## 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, modificar los parámetros del algoritmo o corregir los resultados [8].

## 1.5 Algunas técnicas de segmentación desarrolladas en Vismedic

### 1.5.1 *Clustering*

Un *cluster* es una colección de objetos que son similares entre sí según un determinado criterio de similitud y distintos a los objetos que pertenecen a otros *clusters* [10]. El *Clustering* es una técnica común que se utiliza en diversos campos, donde la distribución de la información puede ser de cualquier tamaño y forma. Esta técnica ayuda a discernir la estructura y simplifica la complejidad de cantidades masivas de datos. La eficiencia de los algoritmos de *Clustering* es extremadamente necesaria cuando se trabaja con enormes bases de datos y tipos de datos de grandes dimensiones [10]. Estos métodos descomponen el conjunto de objetos en un conjunto de *clusters* disjuntos, minimizando una función criterio que enfatiza la estructura local de los objetos, asignando *clusters* a máximos locales en la estructura global [2]. Un resultado se muestra en la Figura 4.

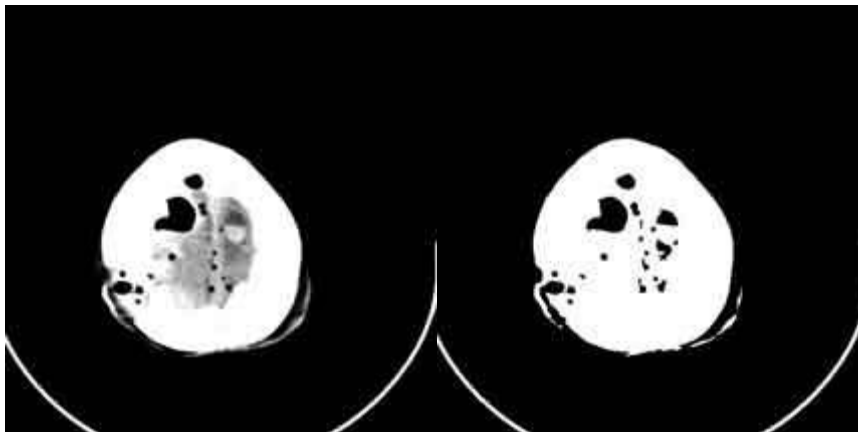


Figura 4. Izquierda: Imagen original para aplicar *Clustering*. Derecha: Resultados del algoritmo *K-means* para 2 *clusters*.



## 1.5.2 Watershed

*Watershed* está basado en la visualización de una imagen en tres dimensiones: dos coordenadas espaciales contra niveles de grises [11]. En esa interpretación "topográfica", se consideran tres tipos de puntos: (a) puntos pertenecientes a un mínimo local; (b) puntos en los cuales una gota de agua, si es colocada en alguno de esos puntos, caería con certeza a un mínimo; y (c) puntos en los cuales el agua estaría en las mismas posibilidades de caer en más de un mínimo [10]. Para un mínimo local en particular, el conjunto de puntos que satisfacen la condición (b) es llamado la cuenca o *watershed* de ese mínimo. Los puntos que satisfacen la condición (c) forman las crestas en la superficie topográfica y son llamadas líneas divisorias o líneas de *watershed* [4]. En la Figura 5 se muestra un ejemplo.



**Figura 5. Izquierda: Imagen médica original. Derecha: Imagen médica segmentada mediante técnicas *Watershed*.**

## 1.5.3 Snakes

Esta técnica le permite al usuario la selección del área que se desea segmentar. 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 [10]. Luego de tener estos datos, se introducen los valores de las variables  $\alpha$ ,  $\beta$  y  $\gamma$  que controlan la influencia de los términos de la función de energía y se procede al ejecutar el algoritmo. 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 [10]. 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 [3]. Esta representación se puede apreciar en la Figura 6.

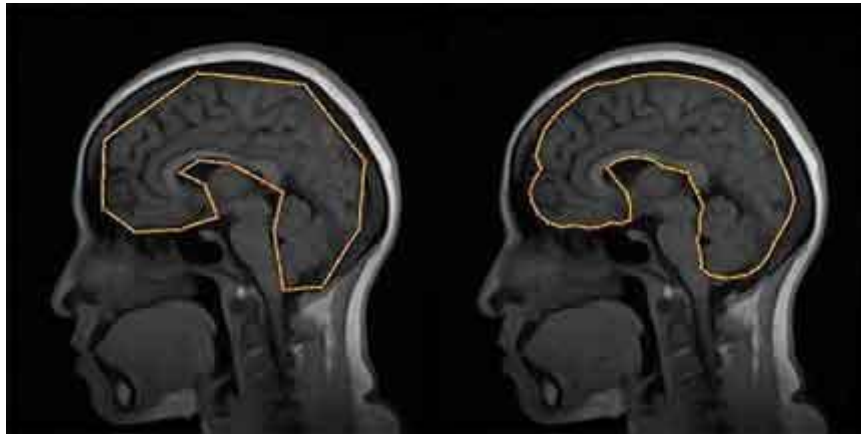


Figura 6. Izquierda: Contorno inicial con pocos vértices sobre el área que se desea segmentar. Derecha: Resultado de la segmentación *Snakes*.

## 1.6 Bibliotecas de clases ITK y MDSTk

### 1.6.1 MDSTk

*Medical Data Segmentation Toolkit* (MDSTk) es un conjunto de herramientas de procesamiento de imágenes bidimensionales y tridimensionales, originalmente destinados a la segmentación de imágenes médicas [19]. Contiene un número de rutinas de procesamiento de datos volumétricos (filtrado, detección de bordes, segmentación), así como un bajo nivel del vector de gráficos de superficie [19]. Parte de la base de la MDSTk (procesamiento de imágenes, bibliotecas del sistema y módulos) son públicos, distribuidos en su sitio web [19], por lo que no todas las herramientas son libres. La versión actual de esta biblioteca es la 1.1.1 [19].

### 1.6.2 ITK

ITK, es una biblioteca de clases que soporta el Proyecto Humano Visible (*Visible Human Project*) [1] [12]. Fue desarrollada por seis organizaciones, tres comerciales (*Kitware*, *GE Corporate R&D*, e *Insight*) y tres académicas (*UNC Chapel Hill*, Universidad de *Utah*, y Universidad de *Pennsylvania*) [1] [8].

Equipos adicionales de trabajo incluyen el *Hospital Harvard Brigham & Women's*, la Universidad de *Pittsburg*, y la Universidad de Columbia [1]. ITK es una biblioteca multiplataforma que está implementada en C++ y puede ser utilizada desde múltiples lenguajes de programación como *Tcl/Tk*, *Java*, *Python* y *C++* [1]. Es altamente eficiente, y está en constante mantenimiento por la comunidad internacional, teniendo al menos una versión por año, en la cual se realizan mejoras en el rendimiento de las técnicas de segmentación que tiene [14]. Esta biblioteca tiene una gran potencia de cálculo y versatilidad para la investigación analítica de imágenes, en cuanto a segmentación, clasificación y técnicas de registro para procesar grandes volúmenes de datos médicos [14]. ITK desde su versión 3.6 se distribuye bajo la licencia de código abierto *Berkeley Software Distribution* (BSD) [1] [13] [15]. BSD es una licencia de software libre permisiva que al contrario que la *General Public License* (GPL) permite el uso del código fuente en software no libre [16] [17]. A partir de ITK 4.0, el kit de herramientas de ITK se distribuye bajo la licencia *Apache 2.0* aprobada por la *Open Source Initiative* (OSI), en la que su uso es ilimitado, incluyendo el uso en productos comerciales [13] [15].

## 1.6.2.1 Algunos filtros de ITK

### Umbral Binario

La operación de umbralización se utiliza para cambiar o identificar los valores de los píxeles, basados en la especificación de un rango de valores que definen un umbral que permite calcular el valor final de intensidad para cada píxel [12].

El filtro Umbral Binario se utiliza para transformar una imagen, en una imagen binaria, mediante el cambio de los valores del píxel, de acuerdo con la regla que se ilustra en la Figura 7. El usuario define la intensidad de los umbrales superior e inferior, y los valores de intensidad para cuando el píxel se encuentre dentro y fuera del umbral. Para cada píxel en la imagen de entrada, el valor del píxel se compara con los umbrales inferior y superior [12].

Si el valor del píxel, está dentro del rango definido por [inferior, superior], al píxel de salida se le asigna el valor de intensidad definido para el umbral. De lo contrario a los píxeles de salida se les asigna el valor de intensidad definido para los píxeles que se encuentran fuera del umbral. La umbralización se aplica comúnmente como la última operación de una tubería de segmentación [12].

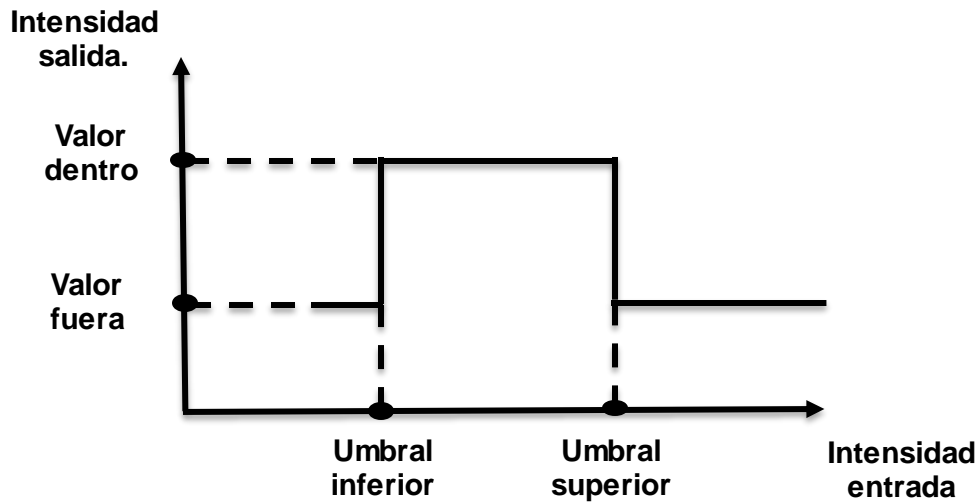


Figura 7. Función de transferencia del filtro de umbral binario.

En la Figura 8 se ilustra el efecto de este filtro, en una imagen capturada por una RM del cerebro. Esta figura muestra las limitaciones de este filtro, para realizar segmentación sin aplicar otras técnicas de filtrado. Estas limitaciones son particularmente notables en imágenes ruidosas y en imágenes que carecen de uniformidad espacial, como es el caso de la RM debido a la intensidad del campo [12].

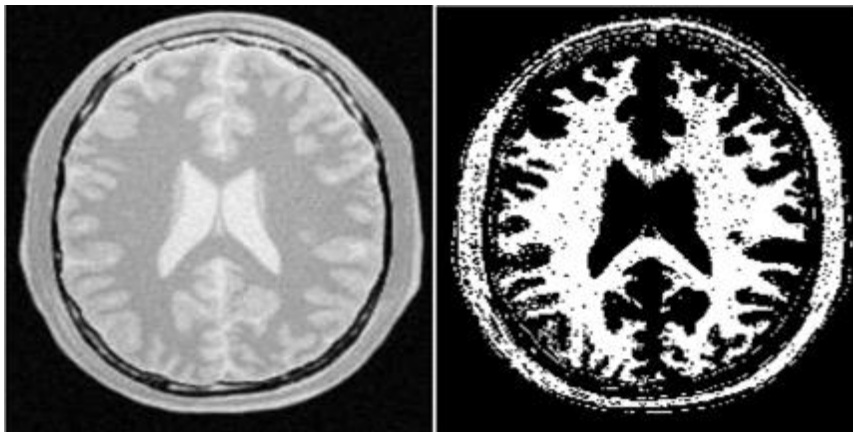


Figura 8. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro de Umbral Binario.

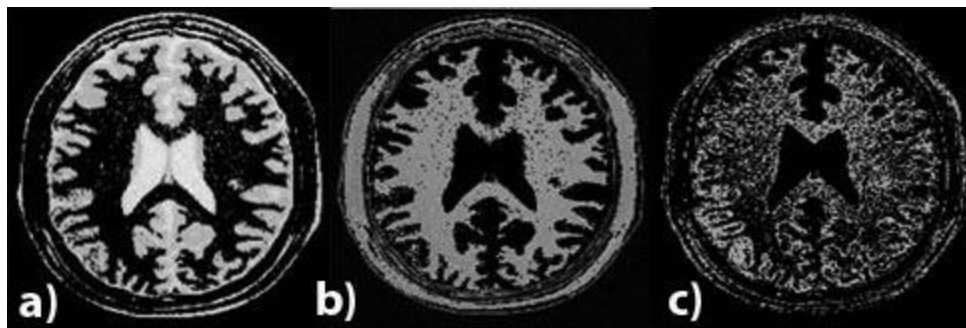
## Umbral General

Este filtro puede ser utilizado para transformar los niveles de intensidad de una imagen en tres modos diferentes.

**Modo Debajo:** el usuario puede definir un único umbral. Los píxeles con valores por debajo de este umbral se sustituyen por un valor definido por el usuario, llamado *OutsideValue*. Los píxeles con valores por encima del umbral no cambian [12].

**Modo Arriba:** el usuario puede definir un umbral en particular, de tal manera que todos los píxeles con valores por encima del umbral se sustituyen por el *OutsideValue*. Los píxeles con valores por debajo del umbral permanecen sin cambios [12].

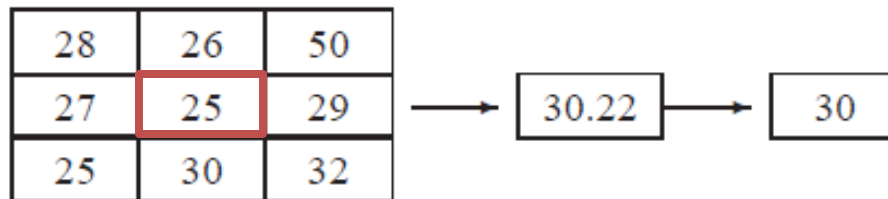
**Modo Fuera:** el usuario puede proporcionar dos umbrales. Todos los píxeles con valores de intensidad dentro del rango definido por los dos umbrales se mantendrán sin cambios. Los píxeles con valores fuera de este rango se le asignan, el valor de *OutsideValue* [12]. En la Figura 9 se muestra el efecto de este filtro en los diferentes modos. Observe también las limitaciones del filtro Umbral General para realizar la segmentación en imágenes ruidosas e imágenes que carecen de uniformidad espacial.



**Figura 9. Efecto del filtro Umbral General en los modos: a) Modo Debajo, b) Modo Arriba, c) Modo Fuera.**

## Filtro del Promedio

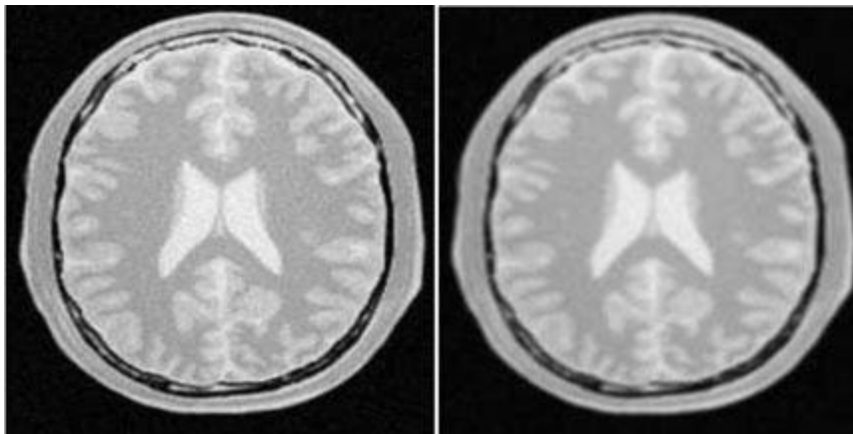
Se utiliza comúnmente para la reducción de ruido. El filtro calcula el valor de cada píxel de salida mediante la búsqueda de la media estadística de la vecindad del píxel de entrada correspondiente. La Figura 10 ilustra el efecto local de la media en un caso bidimensional. La media estadística de la zona de la izquierda se pasa como el valor de salida asociado con el píxel en el centro de la zona [12].



**Figura 10. Efecto local del filtro Promedio en un caso bidimensional.**

Los resultados de aplicar el filtro del Promedio, son sensibles a la presencia de valores extraños en la zona que se aplica. Este filtro funciona en imágenes de cualquier dimensión, sin embargo, ITK permite al usuario ajustar el tamaño de la zona sobre la que se calcula el promedio [12].

En la Figura 11, se ilustra el efecto de este filtro sobre una imagen de RM, usando radios de vecindad de 1,1 que corresponde a la vecindad clásica de  $3 \times 3$  píxeles. Se puede observar a partir de esta imagen, que los bordes se degradan rápidamente por la difusión de los valores de intensidad entre los píxeles vecinos.

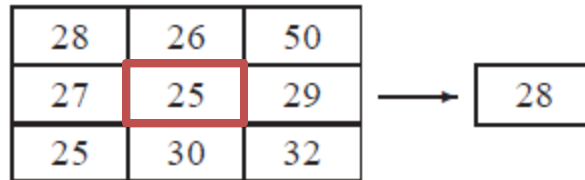


**Figura 11. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro Promedio.**

## Filtro de la Mediana

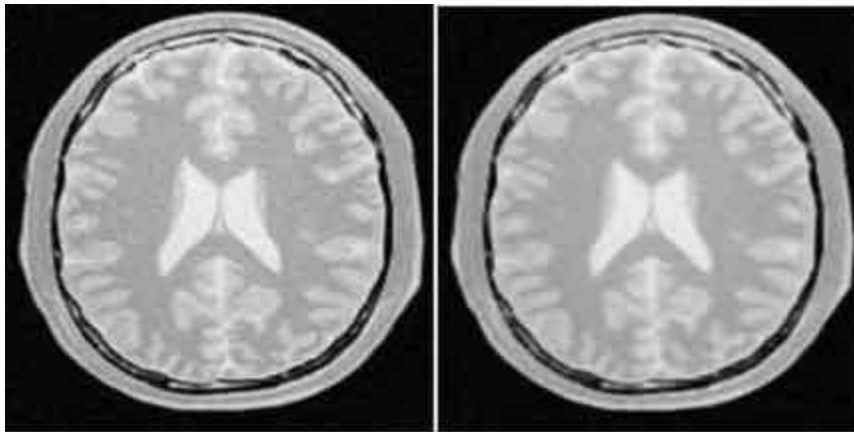
Se usa comúnmente como un sólido método para la reducción de ruido [18]. Este filtro es particularmente eficaz contra el ruido de sal y pimienta. En otras palabras, es robusto a la presencia de niveles de gris en los valores atípicos. Se calcula el valor de cada píxel de salida, como la mediana estadística de la zona de

valores en la vecindad del píxel de entrada correspondiente [12]. En la Figura 12 se ilustra el efecto local de este filtro, en un caso bidimensional.



**Figura 12. Efecto local del filtro la Mediana en un caso bidimensional.**

La Figura 13 ilustra el efecto del filtro Mediana, sobre una imagen de RM del cerebro, utilizando un radio de vecindad de 1,1, lo que corresponde a la vecindad clásica de  $3 \times 3$  píxeles. La imagen filtrada demuestra la tendencia moderada del filtro de mediana para preservar los bordes de la imagen.



**Figura 13. Izquierda: Imagen original del cerebro. Derecha: Efecto del filtro Mediana.**

## Detección de Bordes *Canny*

Este filtro se utiliza para la detección de bordes, ya que es una de las soluciones para satisfacer las limitaciones de sensibilidad, localización y robustez frente al ruido [12]. El propósito de *Canny* era descubrir el algoritmo óptimo de detección de bordes [29]. Para que un detector de bordes pueda ser considerado óptimo debe cumplir los siguientes puntos:

**Buena detección:** el algoritmo debe marcar el mayor número real en los bordes de la imagen como sea posible [29].

**Buena localización:** los bordes de marca deben estar lo más cerca posible del borde de la imagen real [29].

**Respuesta mínima:** El borde de una imagen sólo debe ser marcado una vez, y siempre que sea posible, el ruido de la imagen no debe crear falsos bordes [29].

Para satisfacer estos requisitos *Canny* utiliza el cálculo de variaciones, una técnica que encuentra la función que optimiza un funcional indicado. La función óptima en el algoritmo de *Canny* es descrita por la suma de cuatro términos exponenciales, pero se puede aproximar por la primera derivada de una gaussiana [29]. En la Figura 14 se ilustra el resultado al aplicar este filtro.

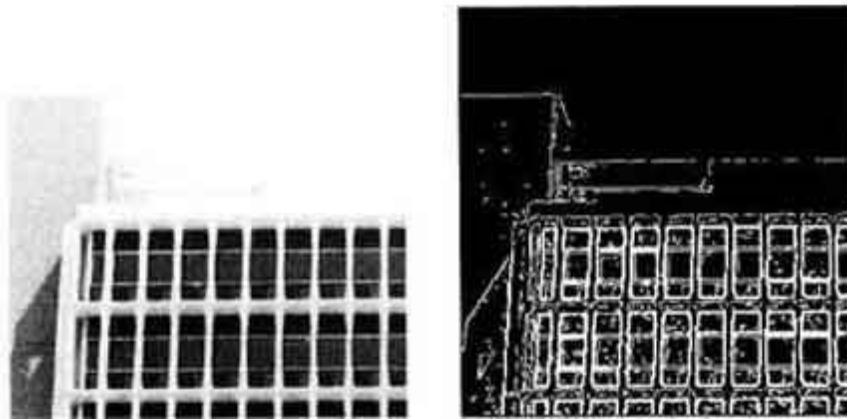


Figura 14. Izquierda: Imagen original. Derecha: Efecto del filtro de Detección de Borde *Canny*.

## 1.6.2.2 Algunas técnicas de segmentación de ITK



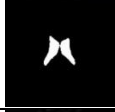

### Conectar umbral

Se basa en un intervalo de valores de intensidad definidos por el usuario. Este intervalo está acotado por los valores inferior y superior que definen el umbral que se utiliza en la segmentación. El algoritmo de crecimiento de la región incluye los píxeles cuyas intensidades están dentro del intervalo  $I(\mathbf{X}) \in [\text{inferior}, \text{superior}]$  [12]. Dentro de los valores que necesita esta técnica se encuentra "paso de tiempo", que debe ser suficientemente pequeño para asegurar la estabilidad numérica. Esta condición asegura que cada contorno



no se mueva de una posición en la rejilla de cada paso de tiempo. El paso de tiempo es típicamente especificado por el usuario [1].

En la Tabla 1 se muestran varios ejemplos de segmentación. Se puede observar que la sustancia gris no está completamente segmentada. Esto demuestra la vulnerabilidad de los métodos de crecimiento de la región, cuando las estructuras anatómicas a segmentar, no tienen una distribución estadística homogénea sobre el espacio de imagen [12].

Estructura	Indice de la semilla	Inferior	Superior	Imagen de salida
Original	-	-	-	
Sustancia blanca	(60,116)	150	180	
Ventrículo	(81,112)	210	250	
Sustancia gris	(107,69)	180	210	

**Tabla 1. Ejemplos de los resultados aplicando el algoritmo Conectar Umbral.**

## Segmentación *Otsu*

Otro criterio para clasificar píxeles, es minimizar el error de clasificación errónea. La meta es encontrar un umbral que clasifica la imagen en dos grupos, algo semejante, sería minimizar el área bajo el histograma para un grupo, que difiere, en el lado del umbral de otro grupo [12].

La Figura 15 ilustra el efecto de segmentar una imagen de RM del cerebro con la técnica *Otsu*. En esta figura se observan las limitaciones de este algoritmo para realizar segmentaciones, particularmente en imágenes ruidosas e imágenes que carecen de uniformidad espacial como es el caso de la RM debido a la intensidad del campo [12].

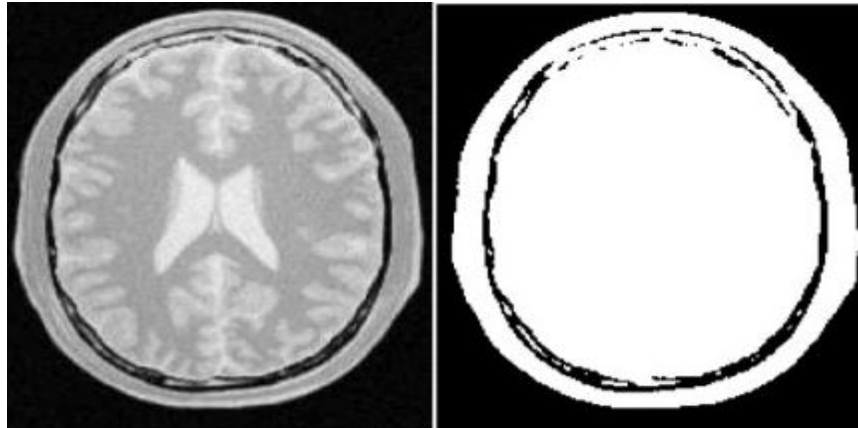


Figura 15. Izquierda: Imagen original. Derecha: Efecto de la segmentación *Otsu*.

## Conectar Vecindad

Este filtro es una variante cercana del Conectar Umbral. Por un lado, el Conectar Umbral acepta un píxel en la región si su intensidad, está en el intervalo definido por valores de intensidad del umbral proporcionado por el usuario. El Conectar Vecindad, por otro lado, sólo aceptará un píxel si todos sus vecinos tienen intensidades que se ajustan en el intervalo. En la Tabla 2, se muestran varios resultados al aplicar esta técnica de segmentación.

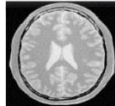



Estructura	Indice de la semilla	Inferior	Superior	Imagen de salida
Original	-	-	-	
Sustancia blanca	(60,116)	150	180	
Ventrículo	(81,112)	210	250	
Sustancia gris	(107,69)	180	210	

Tabla 2. Ejemplos de los resultados aplicando el algoritmo Conectar Vecindad.

El tamaño de la vecindad, a considerar alrededor de cada píxel, se define según el radio de número entero que proporciona el usuario. La razón para considerar las intensidades locales, en lugar, de considerar sólo la intensidad del píxel actual, es que las estructuras pequeñas tienen menos probabilidades de ser aceptada en la región.

Es posible que desee jugar con el valor del radio y ver cómo afecta a la suavidad de los bordes de objetos que se segmentaron, el tamaño de la región segmentada y la cantidad de los costos de tiempo de cálculo.

## 1.7 Conclusiones parciales

El filtrado y segmentación de las imágenes médicas digitales, es un proceso de vital importancia para mejorar la calidad de visualización de estructuras anatómicas de interés. Esta etapa en el proyecto Vismedic, se encuentra de forma explícita en las etapas de Reconstrucción y Visualización tridimensional. Una vez estudiado algunos algoritmos de filtrado y segmentación, implementados en el proyecto y los que utiliza ITK, podemos concluir, que estos algoritmos se pueden agrupar bajo un patrón de diseño de clases, que se analizará en el próximo capítulo. Hay varios algoritmos implementados en el proyecto que se encuentran desarrollados en la biblioteca de ITK, que está especializada en la segmentación de imágenes médicas. Por lo que se propone la integración de algoritmos implementados en ITK al proyecto Vismedic, sin descartar la posibilidad de que se pueda adicionar al módulo los algoritmos existentes en el proyecto o que puedan ser desarrollados posteriormente.

## Capítulo 2. Solución Propuesta

En este capítulo se aborda la propuesta de solución al problema planteado, haciendo énfasis en los algoritmos que se incorporan en el módulo. También se aborda el patrón de diseño Estrategia para la integración de los algoritmos de filtrado y segmentación. Finalmente, se describe la metodología de desarrollo usada, así como las herramientas y lenguajes utilizados.

### 2.1 Algoritmos seleccionados

Después del estudio realizado, se seleccionaron varias técnicas de filtrado y segmentación para adicionar al módulo, y observar los resultados de calidad de visualización de los algoritmos de la biblioteca ITK. Las técnicas seleccionadas fueron las siguientes: Umbral Binario, Umbral General, Promedio, Detección de Bordes *Canny*, Conectar Umbral, Segmentación *Otsu* y Conectar Vecindad. En los siguientes epígrafes se aborda lo referente a estos algoritmos para su adición al módulo.

#### 2.1.1 Umbral Binario (*Binary Threshold*)

El filtro Umbral Binario (ver sección 1.6.2.1) se utiliza para transformar una imagen en binaria, mediante el cambio de los valores de los píxeles. Para adicionar este algoritmo se definió la clase concreta *BinaryThreshold* la cual se construye pasándole los parámetros con los cuales trabajara el algoritmo, que son: los valores de intensidad de los píxeles de entrada y salida, así como los umbrales inferior y superior. Se definió el método *execute*, que recibe y devuelve un volumen abstracto, el cual se transforma a una imagen de ITK. Después se definen los tipos de píxeles de entrada y salida, con los que trabajará el algoritmo, así como las imágenes de ITK de entrada y salida que se definen con sus tipos de píxeles y dimensiones. Seguidamente se instancia el filtro con el que se trabajará, que en este caso es con el *itk::BinaryThresholdImageFilter* el cual necesita las imágenes de entrada y salida.

#### 2.1.2 Umbral General (*General Threshold*)

Para adicionar el filtro Umbral General (ver sección 1.6.2.1) se definió la clase *GeneralThresholding*, la cual necesita varios parámetros en dependencia del modo con el que se vaya a trabajar. Este filtro trabaja con la clase *itk::ThresholdImageFilter*, haciendo uso de los siguientes métodos: *ThresholdBelow* (para el utilizar el modo Debajo), *ThresholdAbove* (para el utilizar el modo Arriba) y *ThresholdOutside* (para el utilizar el modo Fuera). Los tres modos utilizan el valor de intensidad del píxel de salida, que define el usuario y en

dependencia del modo activado se trabaja con los siguientes parámetros: umbral debajo, umbral arriba y umbral fuera.

### 2.1.3 Promedio (*Mean*)

Para la adición al módulo del filtro Promedio (ver sección 1.6.2.1) se definió la clase *Mean*. El filtro Promedio en ITK trabaja con la clase *itk::MeanImageFilter*. Esta técnica no utiliza parámetros ya que trabaja sobre todos los píxeles de la imagen que se está filtrando. Para que el algoritmo fuese interactivo, se decidió establecer de manera dinámica el parámetro radio sobre el que trabaja el algoritmo. Al aumentar el radio que se definió, incrementa el desenfoque de la imagen que se es filtrada.

### 2.1.4 Detección de Bordes *Canny*. (*Canny Edge Detection*)

Para la adición del filtro Detección de Bordes *Canny* (ver sección 1.6.2.1) se definió la clase concreta *CannyEdgeDetection*, definiendo el parámetro de varianza que permitirá decidir al especialista el nivel de detalle de la detección de los bordes. El filtro *Canny* trabaja con la clase *itk::CannyEdgeDetectionImageFilter*. Este filtro opera sobre una imagen que almacena los píxeles utilizando números flotantes. Por lo que es necesario convertir el tipo de la imagen de entrada, que es por lo general de tipo entero o de caracteres, a número flotante. En ITK se resuelve esta limitante con la clase *itk::CastImageFilter*.

### 2.1.5 Conectar el Umbral (*Connected Threshold*)

Para adicionar el algoritmo Conectar umbral (ver sección 1.6.2.2) se definió la clase *ConnectedThreshold*. Este algoritmo utiliza la clase *itk::ConnectedThresholdImageFilter*, y emplea valores de umbral inferior y superior, así como el índice del píxel semilla. Este algoritmo utiliza internamente la clase *itk::CurvatureFlowImageFilter*, que es un filtro que necesita como parámetros el número de iteraciones y el paso de tiempo.

### 2.1.6 Segmentación *Otsu* (*Otsu Segmentation*)

El algoritmo Segmentación *Otsu* (ver sección 1.6.2.2) se adicionó al módulo con la definición de la clase *OtsuSegmentation*. Esta técnica emplea la clase *itk::OtsuThresholdImageFilter* para su definición en concreto y solo emplea tres parámetros esenciales los valores de intensidades de los píxeles de entrada y salida y el número de iteraciones del histograma.

### 2.1.7 Conectar Vecindad (*Neighborhood Connected*)

Esta técnica es una variante similar a la técnica de Conectar el Umbral (ver sección 1.6.2.2). El Conectar Vecindad (ver sección 1.6.2.2) sólo acogerá un píxel si todos sus vecinos tienen intensidades que encajan en el intervalo. El tamaño de la vecindad a ser considerado alrededor de cada píxel, está definido por un radio provisto por el usuario. Para darle solución a su estructura, dentro del módulo se definió la clase *NeighborhoodConnected*. El algoritmo emplea las clases *itk::NeighborhoodConnectedImageFilter* y *itk::CurvatureFlowImageFilter*, que necesita los siguientes parámetros: número de iteraciones y tiempo de paso.

### 2.2 Funcionamiento general del módulo

En el módulo se utiliza un volumen abstracto que es el que recibe cada una de las interfaces de los algoritmos, según sea necesario, se debe de convertir al tipo de dato que utilice cada técnica específica. En la solución propuesta se emplea la biblioteca ITK, por lo que antes de utilizar cada algoritmo en concreto, se convierte este volumen al tipo de datos que emplea ITK. Una vez que se convierten los valores de los píxeles de la imagen, al tipo de dato con el que trabaja el algoritmo que se seleccionó, se introducen los parámetros que necesita la técnica de filtrado o de segmentación en cuestión. Después de aplicar el algoritmo se devuelve una imagen con los tipos de datos que emplea el algoritmo, por lo que se transforma la imagen al tipo de dato del volumen abstracto, que fue el que recibió la interfaz del algoritmo. En la Figura 16 se representa el flujo del módulo para una mejor comprensión.

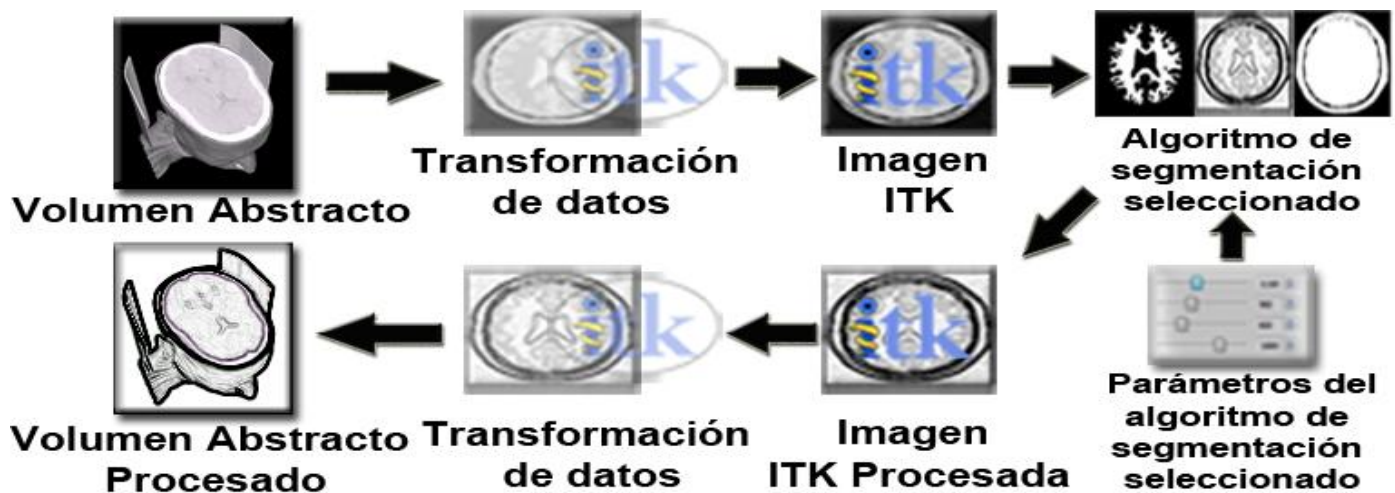


Figura 16. Flujo de datos del módulo propuesto.

## 2.3 Patrón Estrategia

Para encapsular los algoritmos en el módulo y tener la posibilidad de agregar nuevas técnicas de filtrado y de segmentación, se propone utilizar el patrón Estrategia, que se utiliza para organizar una familia de algoritmos, de manera que compartan una interfaz y se puedan utilizar indistintamente [20] [21]. En la Figura 17 se muestra una representación de las clases involucradas en el patrón Estrategia. En este patrón, los objetos de la clase de contexto que acceden a la familia de algoritmos los hacen de forma abstracta, sin importarles el algoritmo específico que se está utilizando.

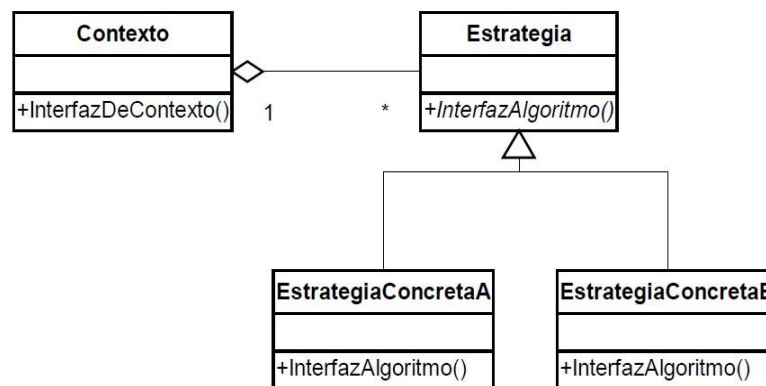


Figura 17. Representación del Patrón Estrategia.

En la Figura 18 se muestra cómo queda representado el patrón Estrategia en el módulo, donde la clase *ProcessImage*, es la interfaz global que agrupa a las interfaces de *Filters* y *Segmentation*, a su vez estas son las interfaces generales de los algoritmos en concreto. Las interfaces y las clases concretas tienen un constructor, un destructor y un método *execute* que es donde se implementará el algoritmo con sus detalles específicos, teniendo como parámetro un volumen abstracto (*AbstractVolume*) y como salida el volumen procesado. En el constructor de las interfaces se generaliza la entrada de parámetros ya que algunos de los algoritmos lo necesitan.

Otro aspecto que se tuvo en cuenta en el desarrollo del módulo propuesto, fue brindar al programador la posibilidad de agregar nuevos algoritmos de filtrado y segmentación sin la necesidad de modificar el resto de la aplicación.

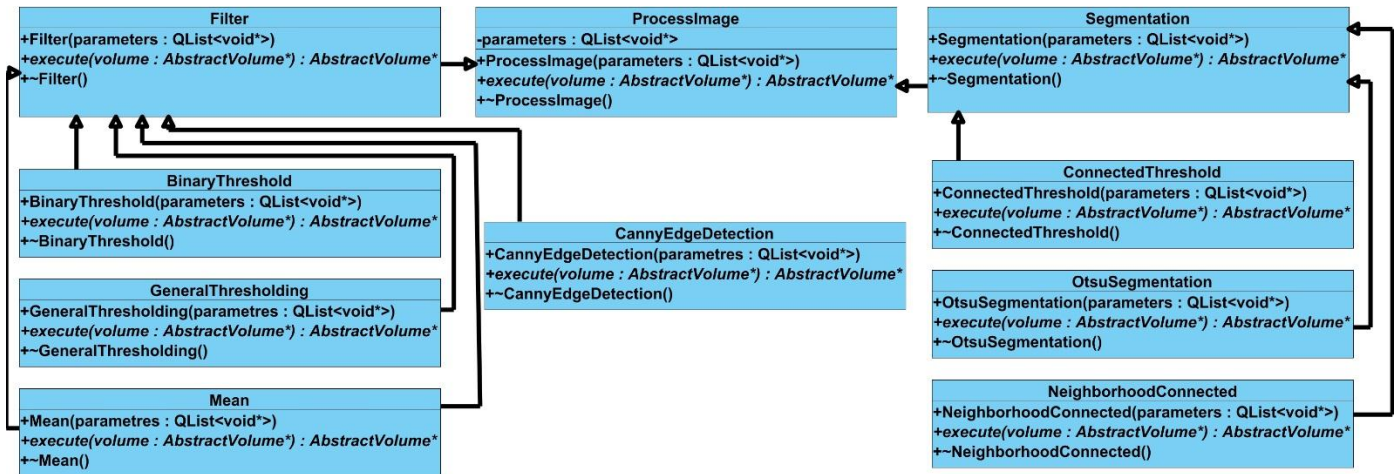


Figura 18. Representación del Patrón Estrategia en el módulo.

Se concluye que para agregar nuevos algoritmos de filtrado y de segmentación, solo se tienen que agregar nuevas clases concretas, e interfaces que necesite el algoritmo en cuestión al *plugin* correspondiente, con el fin de gestionar los parámetros de las técnicas que lo requieran. De este modo se agiliza el proceso de agregar nuevas funcionalidades de procesamiento de imágenes digitales ya que no se tiene que tener conocimiento del funcionamiento de las restantes etapas del proyecto.

## 2.4 Herramientas de desarrollo

Como herramienta de modelado se empleó *Visual Paradigm*, creado para asistir el proceso de Ingeniería de Software, este se encuentra basado en UML y soporta el ciclo de vida completo del desarrollo de software, además cuenta con funcionalidades más avanzadas que las presentes en el *Rational Rose*, lo que permite agilizar considerablemente el trabajo [22]. A continuación se describen sus principales características.

- Presenta licencia gratuita y comercial.
- Soporta aplicaciones web.
- Disponible en varios idiomas.
- Fácil de instalar y actualizar.
- Compatible entre versiones.
- Entorno gráfico amigable para el usuario.



- Disponible en múltiples plataformas (Windows/Linux/Mac OS X).

Para la creación de la interfaz gráfica de usuario (GUI) se utilizó el *framework* QT, este permite la portabilidad de la aplicación hacia diferentes sistemas operativos y facilita en gran medida el desarrollo de nuevos componentes gráficos [22]. Entre sus principales características se encuentran:

- Con el mismo código base, permite desplegar el sistema en múltiples plataformas.
- Producir aplicaciones de alto rendimiento con apariencia nativa.
- La concentración de los desarrolladores en la producción de código y no en las particularidades del sistema operativo.
- Acceso total al código fuente para su revisión y modificación.

### 2.5 Lenguaje de modelado

UML es un lenguaje de modelado visual que se usa para especificar, construir, documentar y visualizar artefactos de un sistema de software. El mismo está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Su objetivo es visualizar, especificar, construir y documentar los artefactos que se crean durante el proceso de desarrollo [22].

Este lenguaje de modelado está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas [22]. La especificación de UML no define un proceso estándar pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

### 2.6 Lenguaje de programación

Como lenguaje de programación se utilizó C++, lenguaje que se utiliza en la mayoría de las aplicaciones de realidad virtual que hace uso eficiente del paradigma de Programación Orientada a Objetos [22]. Permite un excelente control de la memoria y una buena administración de los recursos de la computadora. Dentro de las principales ventajas que presenta el lenguaje C++ se encuentran:

- **Difusión:** al ser uno de los lenguajes más empleados en la actualidad, posee gran número de usuarios y tiene una excelente bibliografía.
- **Versatilidad:** C++ es un lenguaje de propósito general, se puede emplear para resolver cualquier tipo de problema.
- **Portabilidad:** se encuentra estandarizado, por tanto, el mismo código fuente puede ser compilado en diferentes plataformas.
- **Eficiencia:** C++ es uno de los lenguajes más rápidos en tiempo de ejecución.
- **Herramientas:** existe gran cantidad de compiladores, depuradores y bibliotecas de clases basadas en este lenguaje.

### 2.7 Metodología de desarrollo de software

Se escogió como metodología de desarrollo de software el Proceso Unificado de Desarrollo (RUP). Esta robusta metodología unifica los mejores elementos de las metodologías anteriores y está preparada para guiar el desarrollo de prácticamente todo tipo de proyectos [23]. Su diseño orientado a objetos facilita la comprensión a alto nivel para su posterior implementación usando este paradigma de programación [24].

**Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, estos se obtienen durante el modelado del negocio. El proceso de desarrollo de software avanza a través de una serie de flujos que parten de los casos de uso, se puede afirmar que estos proporcionan un hilo conductor y una guía para todo el proceso [23].

**Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo y describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. La arquitectura debe diseñarse para que el software evolucione, no solo en su desarrollo inicial, sino también a lo largo de las futuras generaciones [23].

**Iterativo e incremental:** RUP propone que cada proyecto se desarrolle en fases y que cada fase se desarrolle en iteraciones, donde cada iteración resulta un incremento del proceso de desarrollo, lo cual se realiza de forma planificada y culmina con el cumplimiento del punto de control trazado en la fase [23].

### Capítulo 3. Características del Sistema

En este capítulo se definen las reglas del negocio así como el Modelo de Dominio. Se expone la captura de requisitos funcionales y no funcionales. Se elabora el Diagrama de Casos de Uso y la descripción textual de cada caso de uso. Finalmente se elaboran los diagramas de clases del diseño y los diagramas de secuencia del diseño.

#### 3.1 Reglas del negocio

- Las imágenes médicas que se deseen abrir desde ITK deben estar en formato DICOM (\*.dcm).
- El color de los píxeles de las imágenes brindadas al módulo para el filtrado y la segmentación estarán en escala de grises.

#### 3.2 Modelo de 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, conceptos del mundo real en lugar de componentes de software [24]. El entorno del problema a resolver se puede describir de la siguiente manera: el médico le orienta al paciente una serie de estudios, los cuales se realizan a través de TAC o RM. Estos estudios están formados por un conjunto de imágenes DICOM, las cuales son analizadas a través de un software de visualización que le ayuda al médico para emitir un diagnóstico. Antes de realizar la visualización, se realiza el procesamiento de estas imágenes, con el objetivo de reducir ruido y realizar una correcta selección de las estructuras anatómicas de interés mediante la segmentación. Una vez que estas imágenes han sido procesadas, se reconstruye el modelo tridimensional, para su visualización con una mejor calidad.

El **Médico** es la persona capacitada con conocimientos sobre Imagenología para emitir un diagnóstico.

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

Un **Estudio** es el examen que le orienta el médico al paciente para poder diagnosticar de manera correcta la patología que tiene el paciente.

La **Adquisición** es el proceso donde se adquieren las imágenes médicas digitales.

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

El **Pre-Procesamiento** es la etapa donde se mejora la calidad de las imágenes mediante la aplicación de filtros para reducir el ruido producido por las modalidades de adquisición de imágenes.

La **Segmentación** es donde se realiza una correcta identificación de las estructuras anatómicas de interés.

La **Reconstrucción** es la etapa donde se construye el modelo tridimensional a partir de la información presente en las imágenes médicas.

La **Visualización** es donde se representa de forma realista y en tiempo real la anatomía tridimensional.

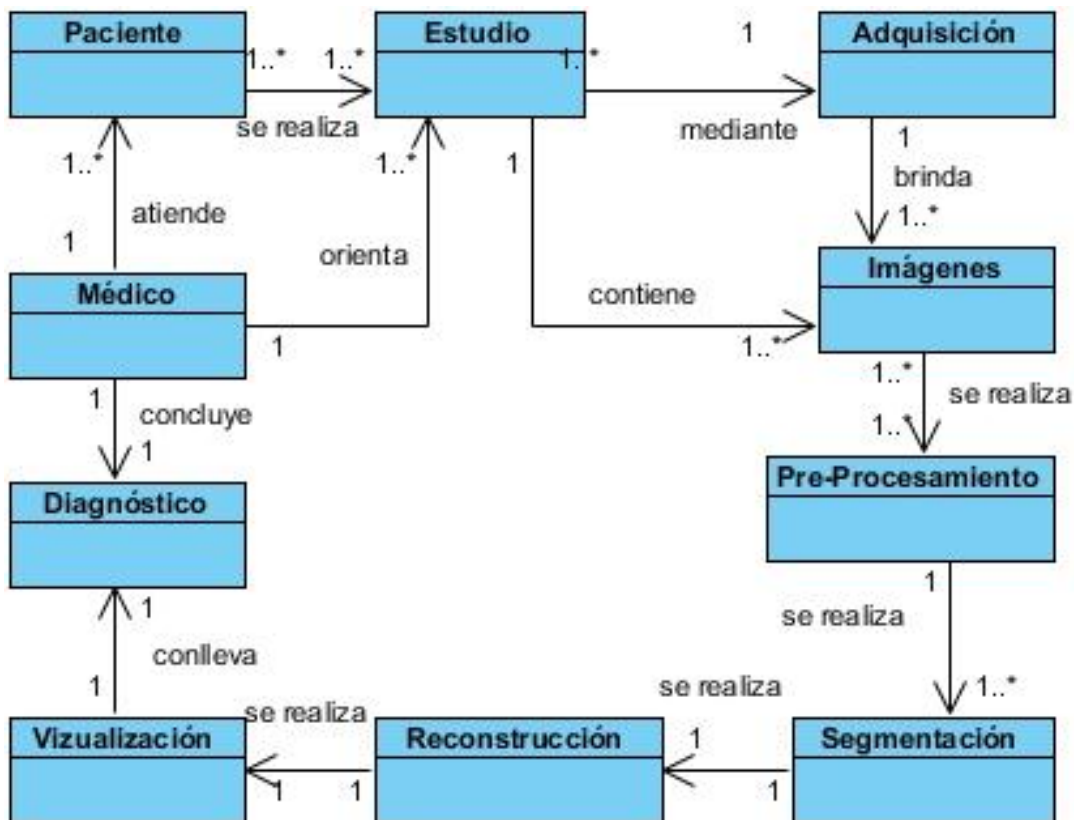


Figura 19. Modelo de Dominio.

## 3.3 Captura de requisitos

Un requerimiento es una capacidad o condición que debe tener un sistema o un componente de un sistema para satisfacer un contrato, norma, especificación y otro documento formal, para facilitar el entendimiento entre clientes y desarrolladores [24]. A continuación 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 [24]. Estos se transforman en casos de uso y se modelan posteriormente. Los siguientes requisitos responden a las funcionalidades que el sistema debe tener una vez concluida su implementación.

#### RF1. Cargar imagen DICOM con ITK.

#### RF2. Filtrar imagen DICOM.

RF2.1 Seleccionar filtro.

RF2.2 Configurar parámetros del filtro.

RF2.3 Aplicar filtro.

#### RF3. Segmentar imagen DICOM.

RF3.1 Seleccionar algoritmo de segmentación.

RF3.2 Configurar parámetros de segmentación.

RF3.3 Aplicar 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 [24]. 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 y Windows XP o superior.

#### 2. Hardware

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

La memoria RAM será 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 y Windows XP.

## 3.4 Modelo de Casos de Uso del Sistema

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

### 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 [23]. En este caso particular quien interactúa con el módulo es un médico, por lo que como actor del sistema se le llamara Médico.

Actores	Justificación
Médico	Interactúa con el sistema para ejecutar las funcionalidades de: cargar imágenes DICOM con ITK, mejorar la calidad visual de la imagen a través de los filtros y segmentaciones de las estructuras anatómicas de interés.

Tabla 3. Actores del Sistema

## 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 [25]. 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.

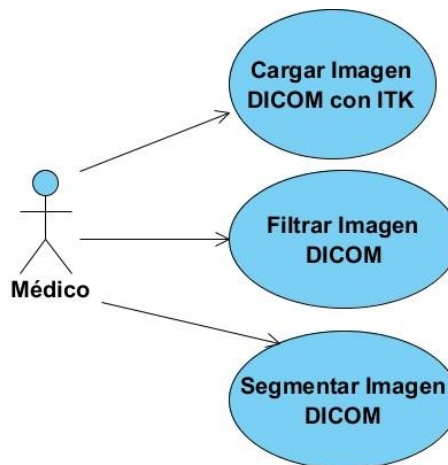


Figura 20. Diagrama de Casos de Uso del Sistema.

## 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.

<b>Caso de Uso:</b>	Cargar Imagen DICOM con ITK.
<b>Actores:</b>	Médico.
<b>Propósito:</b>	Importar la imagen DICOM que el médico desea procesar.
<b>Resumen:</b>	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.
<b>Referencia:</b>	RF1.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción de cargar imagen con ITK.	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 habilita las funcionalidades de filtrado y segmentación.
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.
<b>Postcondiciones:</b>	Se carga y se muestra la imagen seleccionada así como la interfaz de las demás funcionalidades.
<b>Prioridad:</b>	Media.

Tabla 4. Descripción del Caso de Uso Cargar Imagen DICOM con ITK.

<b>Caso de Uso:</b>	Filtrar Imagen DICOM.
---------------------	-----------------------



<b>Actores:</b>	Médico.
<b>Propósito:</b>	Detectar bordes, cambiar intensidades de los píxeles y reducir ruidos de las imágenes.
<b>Resumen:</b>	Se inicia cuando el actor selecciona la opción de filtrar imagen. Se selecciona la técnica de pre-procesamiento a emplear. Finaliza después de aplicado el filtro a la imagen.
<b>Referencia:</b>	RF2, RF2.1, RF2.2, RF2.3.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción filtrar imagen.	1.1 Muestra un menú con las distintas técnicas de pre-procesamiento.
2. Selecciona la técnica de pre-procesamiento a utilizar.	2.1 Se muestra un cuadro de dialogo con los campos de los parámetros necesarios para utilizar la técnica de pre-procesamiento seleccionada.
3. Selecciona la opción de Aplicar.	3.1 Se aplica el filtro y se muestra el resultado en la visualización.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El usuario presiona el botón Cancelar.	3.1 Se cierra el cuadro de diálogo y se muestra la interfaz principal de la aplicación.
<b>Postcondiciones:</b>	Se muestra el resultado de aplicar el filtro seleccionado sobre la imagen cargada.
<b>Prioridad:</b>	Crítica.

**Tabla 5. Descripción del Caso de Uso Filtrar Imagen DICOM.**

<b>Caso de Uso:</b>	Segmentar Imagen DICOM.
<b>Actores:</b>	Médico.
<b>Propósito:</b>	Segmenta estructuras de interés en la imagen cargada.

<b>Resumen:</b>	Se inicia cuando el actor selecciona la opción de segmentar imagen. Se selecciona la técnica de segmentación a emplear. Finaliza después de aplicado la segmentación a la imagen.	
<b>Referencia:</b>	RF3, RF3.1, RF3.2.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. Selecciona la opción segmentar imagen.	1.1 Muestra un menú con las distintas técnicas de segmentación.	
2. Selecciona la técnica de segmentación a utilizar.	2.1 Se muestra un cuadro de diálogo con los campos de los parámetros necesarios para utilizar la técnica de segmentación seleccionada.	
3. Selecciona la opción de Aplicar.	3.1 Se aplica la segmentación y se muestra el resultado en la visualización.	
<b>Flujos Alternos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
3. El usuario presiona el botón Cancelar.	3.1 Se cierra el cuadro de diálogo y se muestra la interfaz principal de la aplicación.	
<b>Postcondiciones:</b>	Se muestra la imagen segmentada con la técnica seleccionada.	
<b>Prioridad:</b>	Crítica.	

**Tabla 6. Descripción del Caso de Uso Segmentar Imagen DICOM.**

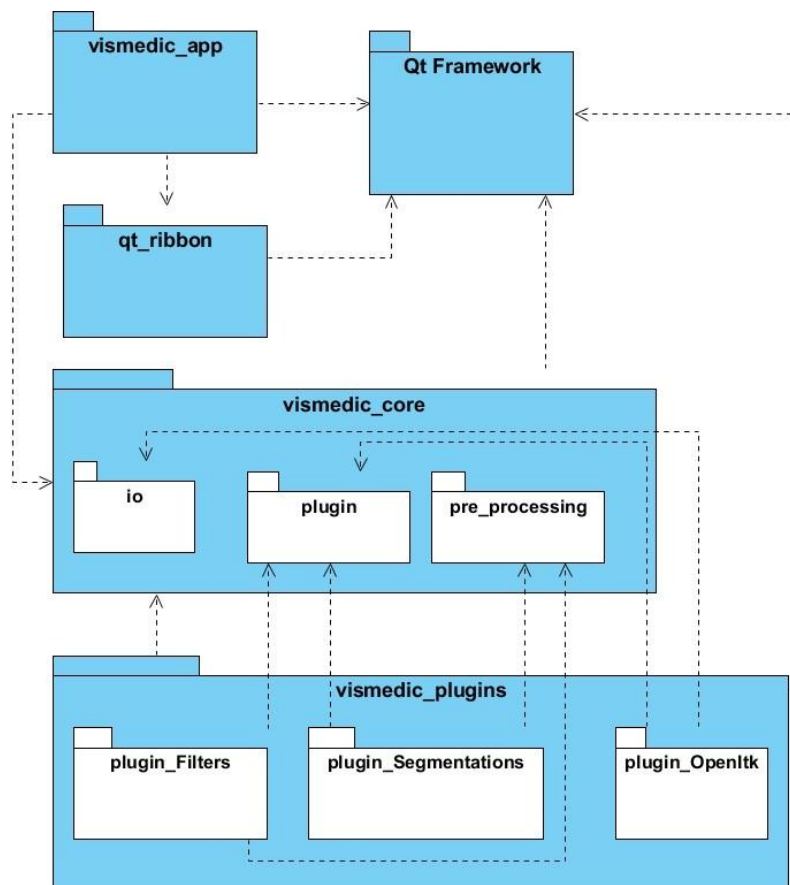
### 3.5 Diagrama de paquetes de clases

El diagrama de clases del diseño se organizó por paquetes, para mantener la claridad y la relación entre cada una de las clases del módulo. Se definieron cuatro paquetes que se muestran en la Figura 21 de manera general y en la Figura 22 de manera más específica. En la Tabla 7 se muestran las funcionalidades de los paquetes de clases.

Paquetes	Subpaquetes	Funcionalidad
<i>qt_ribbon</i>		Sistema de menú de la aplicación.
<i>vismedic_app</i>		Visualización de los datos procesados.

<i>Vismedic_core</i>	<i>io</i>	Entrada y salida de los datos.
	<i>plugin</i>	Define las interfaces de los <i>plugins</i> .
	<i>pre_processing</i>	Define las interfaces de comunicación del filtrado y la segmentación.
<i>vismedic_plugins</i>	<i>plugin_Openltk</i>	Apertura de ficheros DICOM utilizando la biblioteca ITK.
	<i>plugin_Filters</i>	Define técnicas concretas de filtrado.
	<i>plugin_Segmentations</i>	Define algoritmos concretos de segmentación.

**Tabla 7. Funcionalidades de los paquetes de clases.**



**Figura 21. Vista general del diagrama de paquete de clases y sus dependencias.**

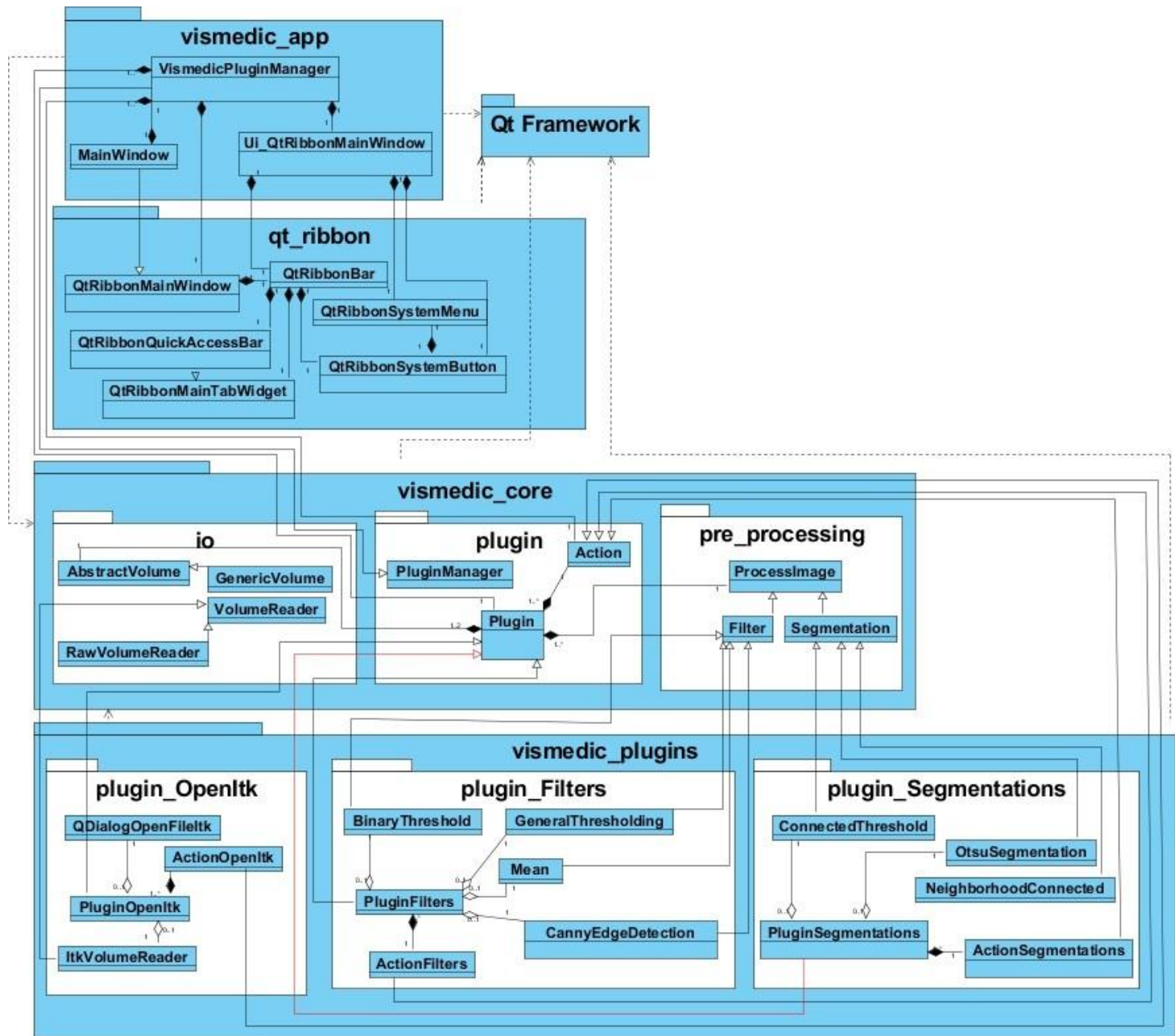


Figura 22. Vista específica del diagrama de paquete de clases y sus relaciones.

## 3.6 Diagramas de clases del paquete *vismedic\_plugins*

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 [24]. 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 muestran los diagramas de clases de los subpaquetes *plugin\_OpenItk*, *plugin\_Filters* y *plugin\_Segmentations*.

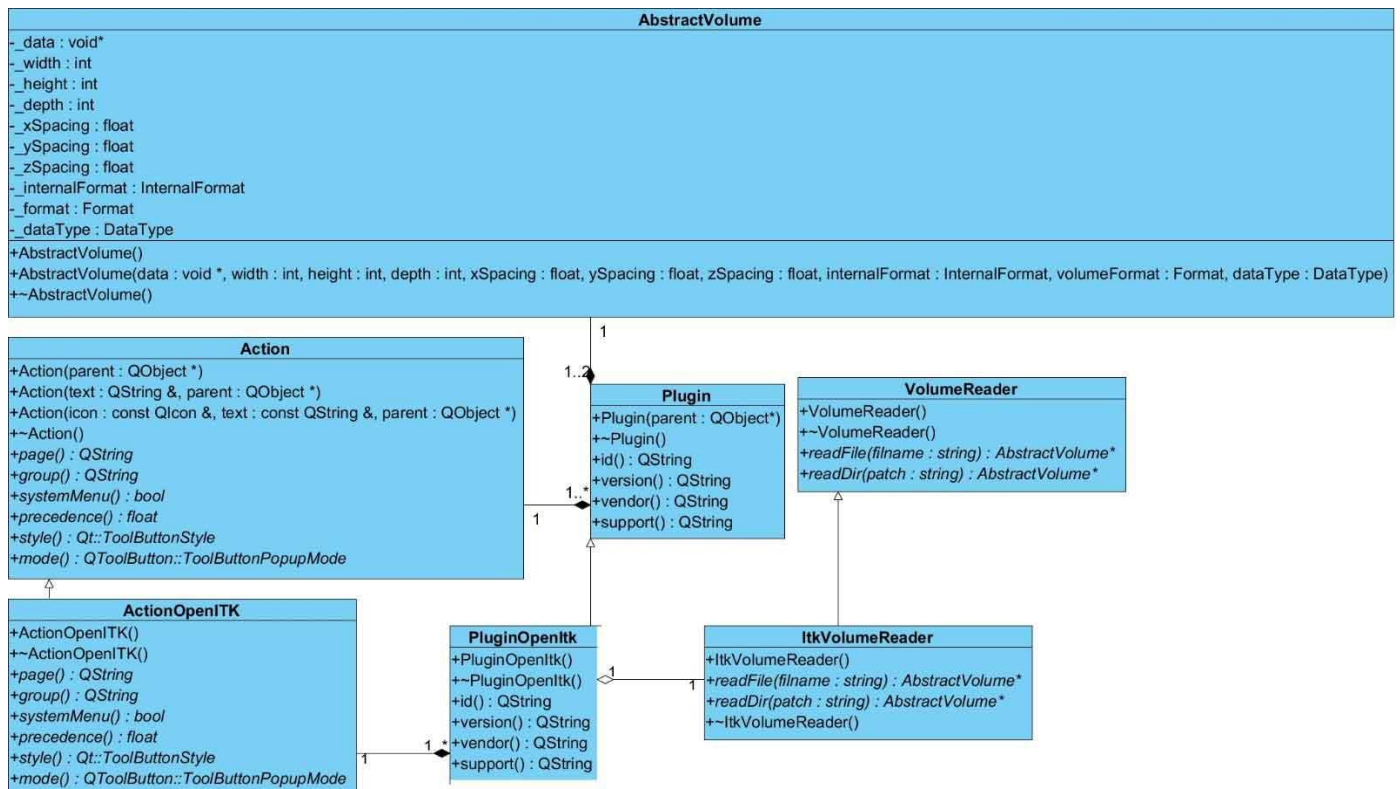


Figura 23. Diagrama de clases del subpaquete *plugin\_OpenItk*.

# Capítulo 3. Características del Sistema

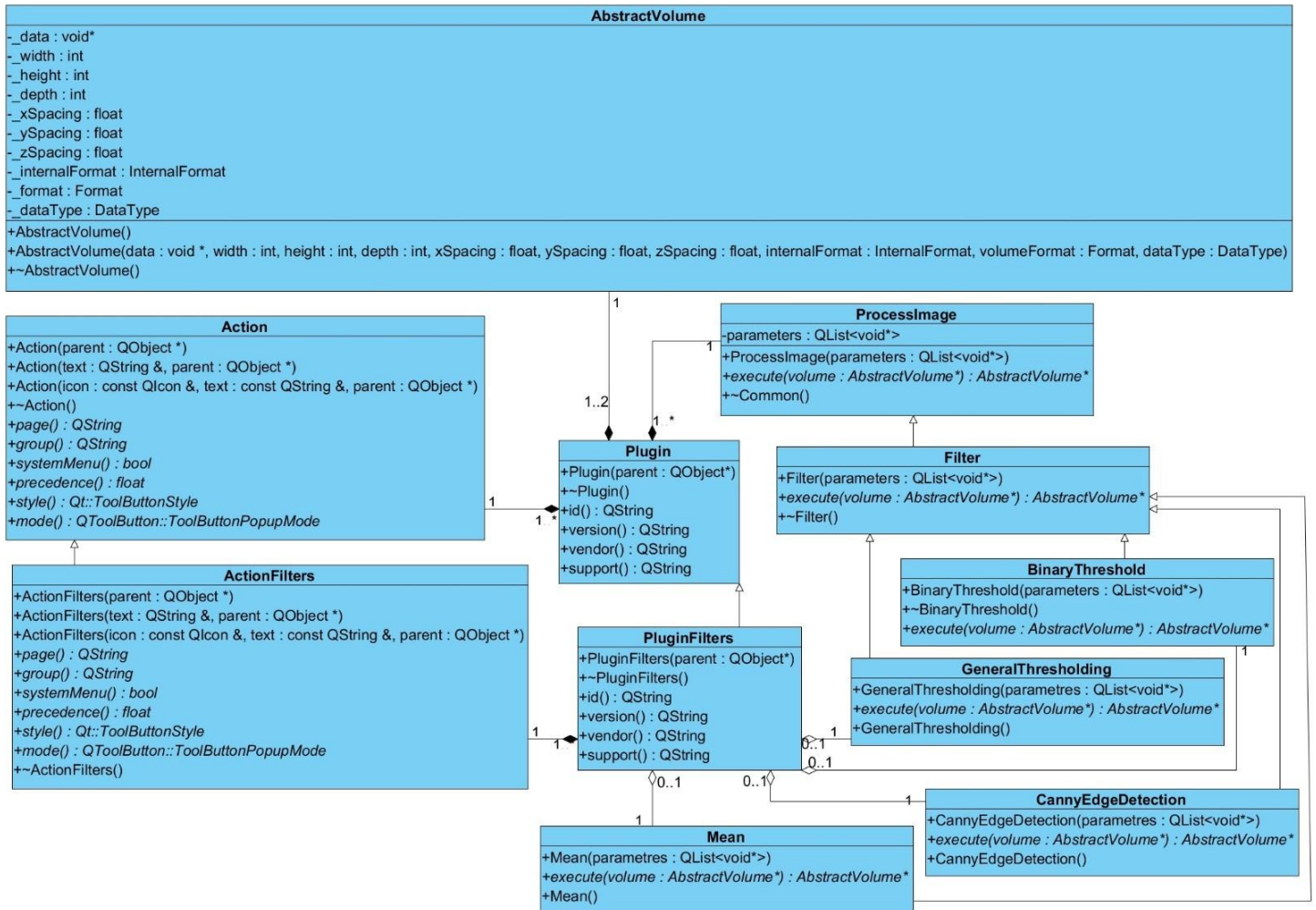


Figura 24. Diagramas de clases del subpaquete *plugin\_Filters*.

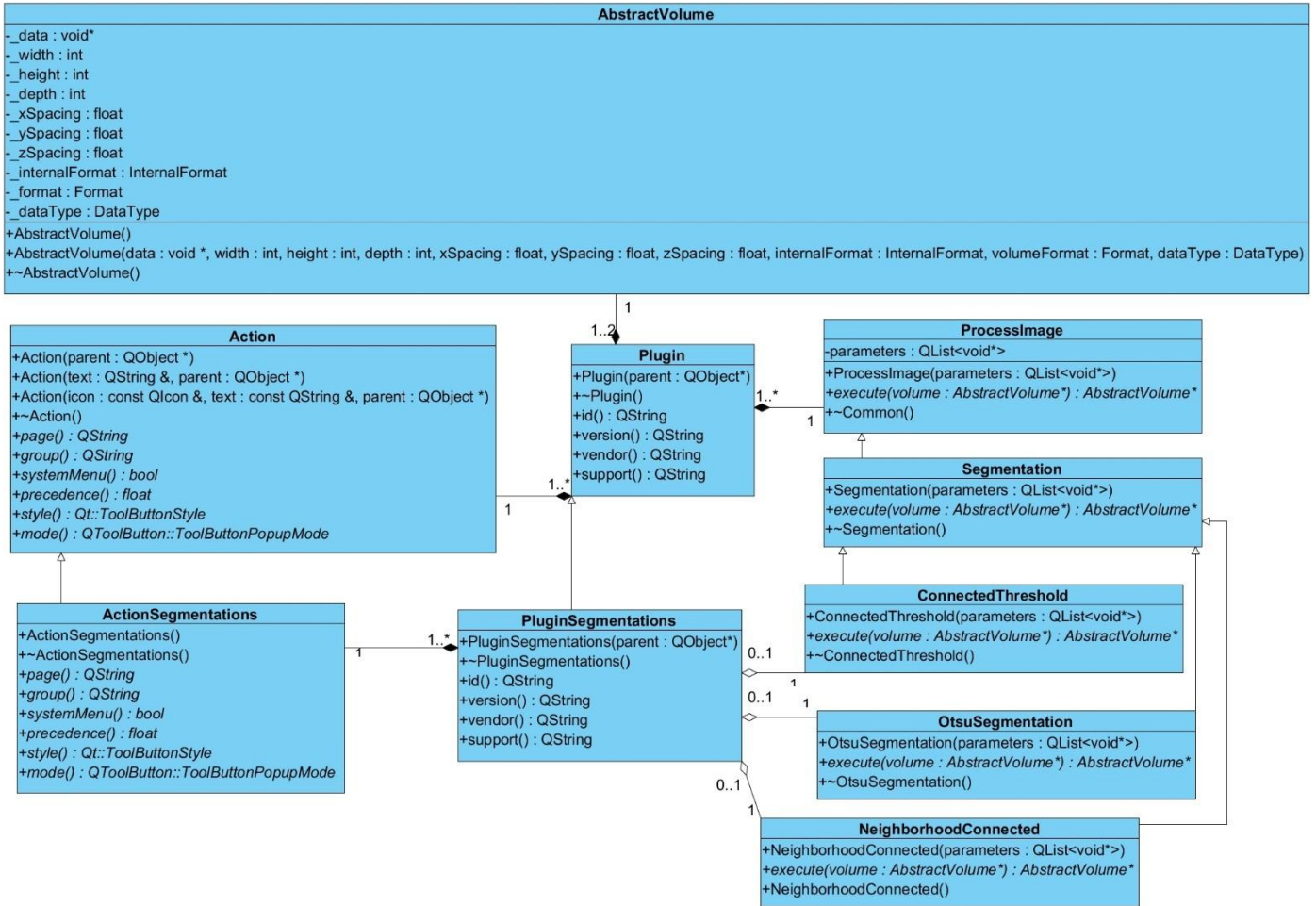


Figura 25. Diagramas de clases del subpaquete *plugin\_Segmentations*.

## 3.7 Diagramas de Secuencia del Diseño

A continuación se representan los diagramas de secuencia del diseño de los paquetes *plugin\_Openltk*, *plugin\_Filters* y *plugin\_Segmentations*, 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.

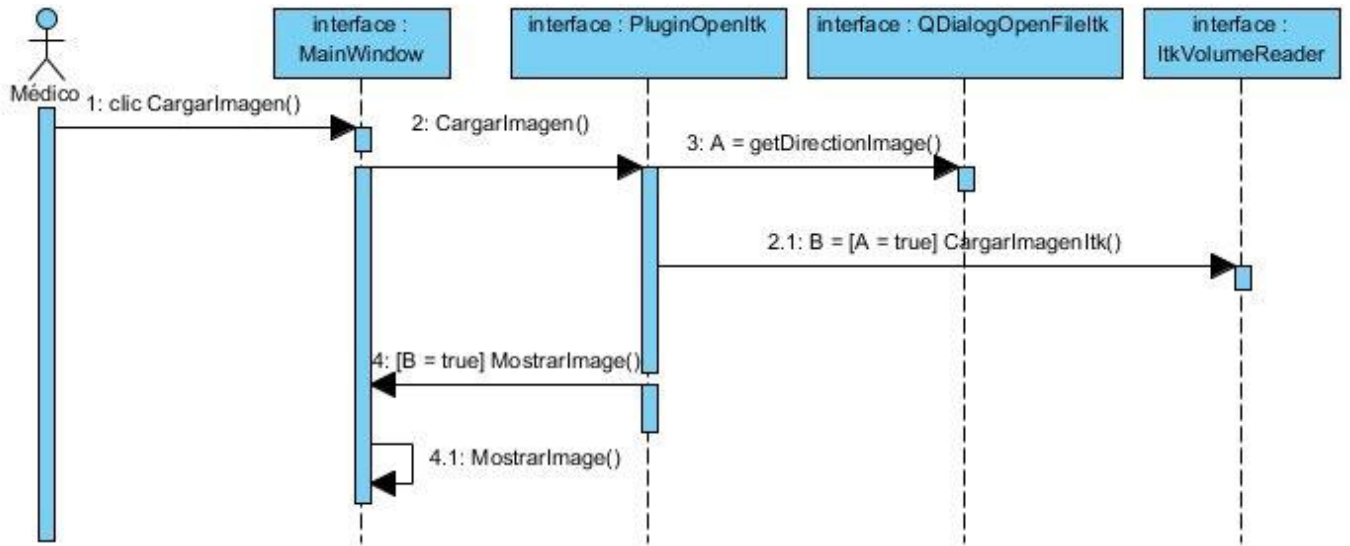


Figura 26. Diagrama de Secuencia del caso de uso Cargar Imagen DICOM con ITK.

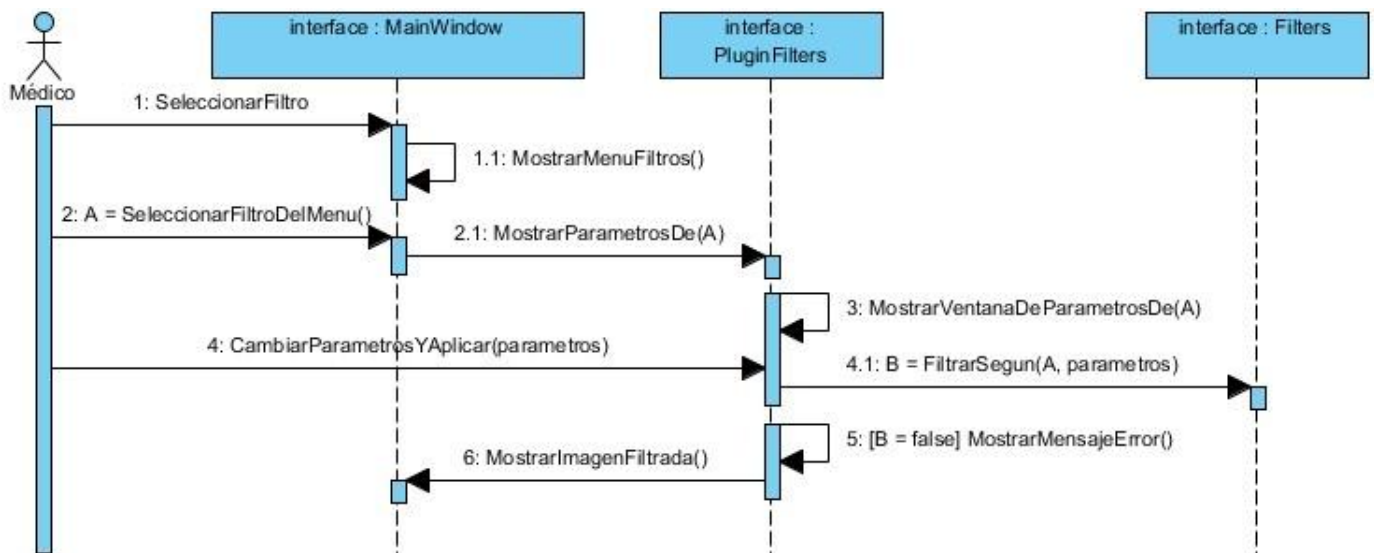


Figura 27. Diagrama de Secuencia del caso de uso Filtrar Imagen DICOM.



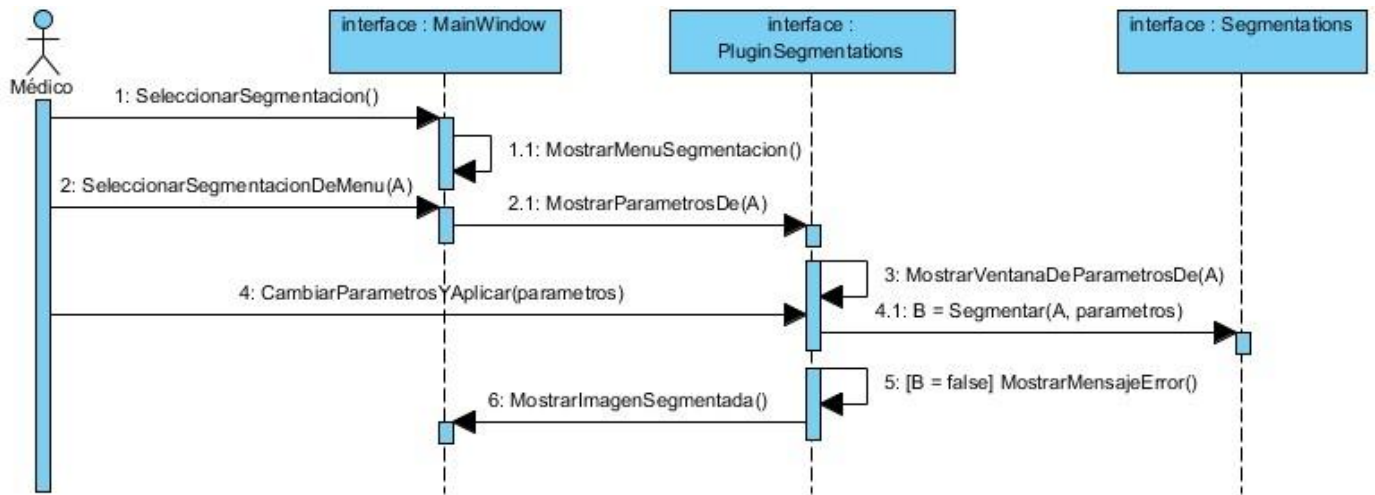


Figura 28. Diagrama de Secuencia del caso de uso Segmentar Imagen DICOM.

### Capítulo 4. Implementación y Presentación de Resultados

Este capítulo aborda la presentación de los resultados y la implementación del sistema. 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. Se definen los casos de prueba para presentar los resultados obtenidos y se hace una comparación de la calidad visual del filtrado y la segmentación que se obtienen cuando se aplican diferentes algoritmos implementados en la aplicación.

#### 4.1 Implementación

En esta etapa se implementan 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. Modelo de Implementación en paquetes

Un componente representa una parte física del sistema, por ejemplo, una biblioteca de clases, un ejecutable, una tabla, que encapsula la implementación de un grupo de clases del diseño. Cada componente define una interfaz que describe su funcionalidad y forma de empleo. El diagrama de componentes, permite conocer a los desarrolladores y clientes la estructura física que tiene el sistema y cómo se relacionan sus partes [24]. A continuación se muestra el diagrama de componentes en paquetes de implementación y posteriormente los diagramas de componentes de los paquetes *plugin\_Openltk*, *plugin\_Filters* y *plugin\_Segmentations*.

# Capítulo 4. Implementación y Presentación de los Resultados

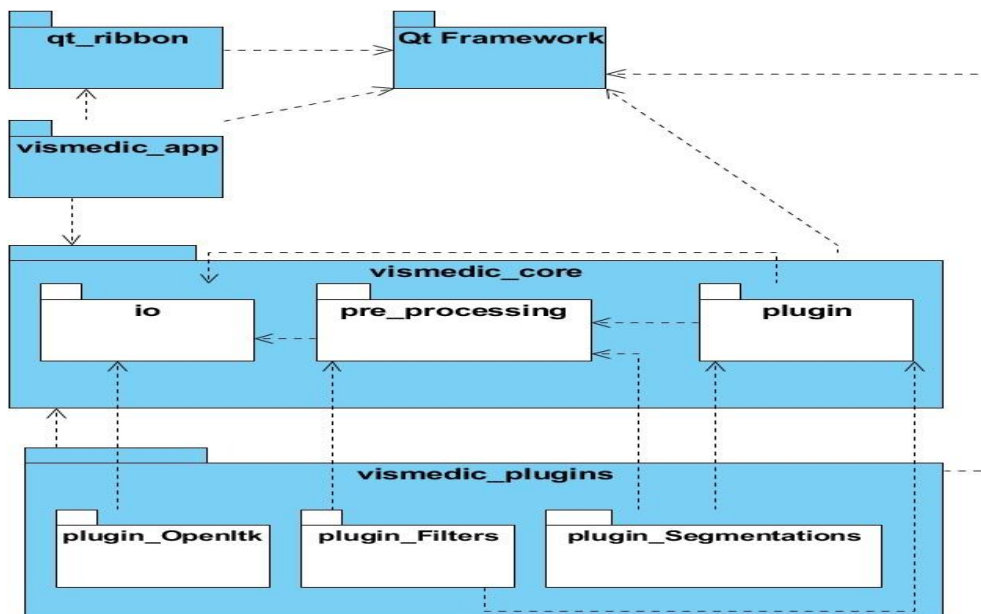


Figura 29. Diagrama de Componentes. Paquetes de Implementación.

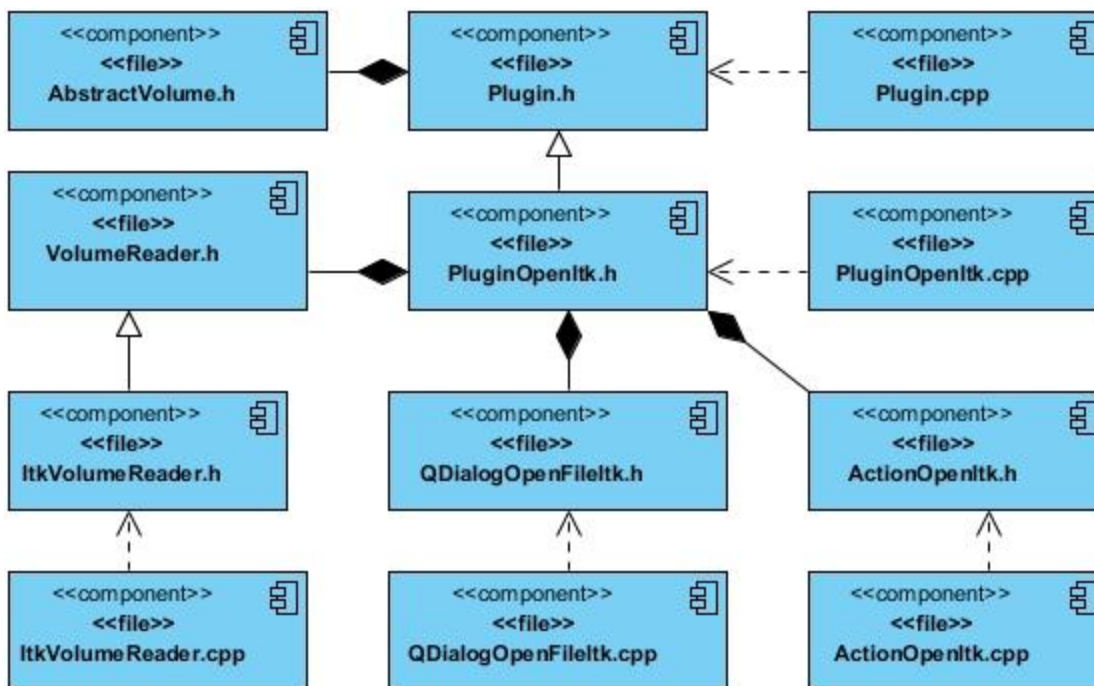


Figura 30. Diagrama de Componentes del subpaquete `plugin_Openitk`.

## Capítulo 4. Implementación y Presentación de los Resultados

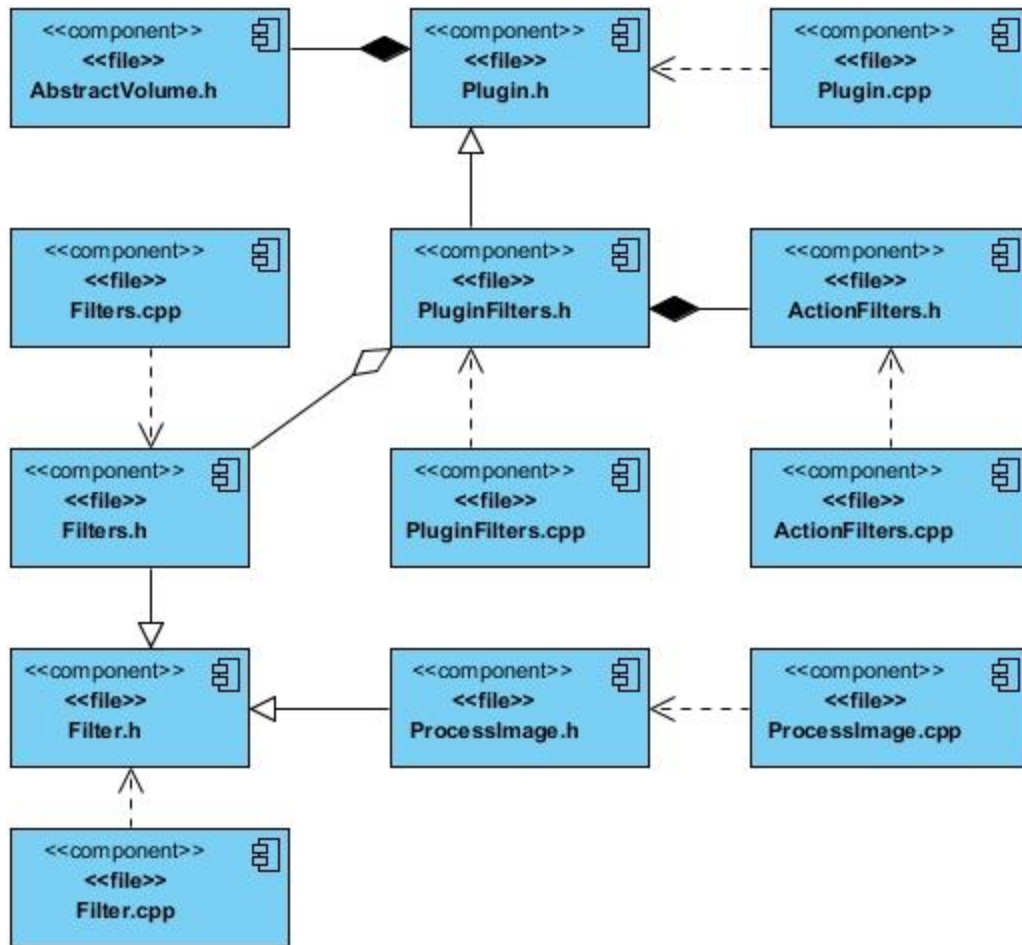


Figura 31. Diagrama de Componentes del subpaquete `plugin_Filters`.

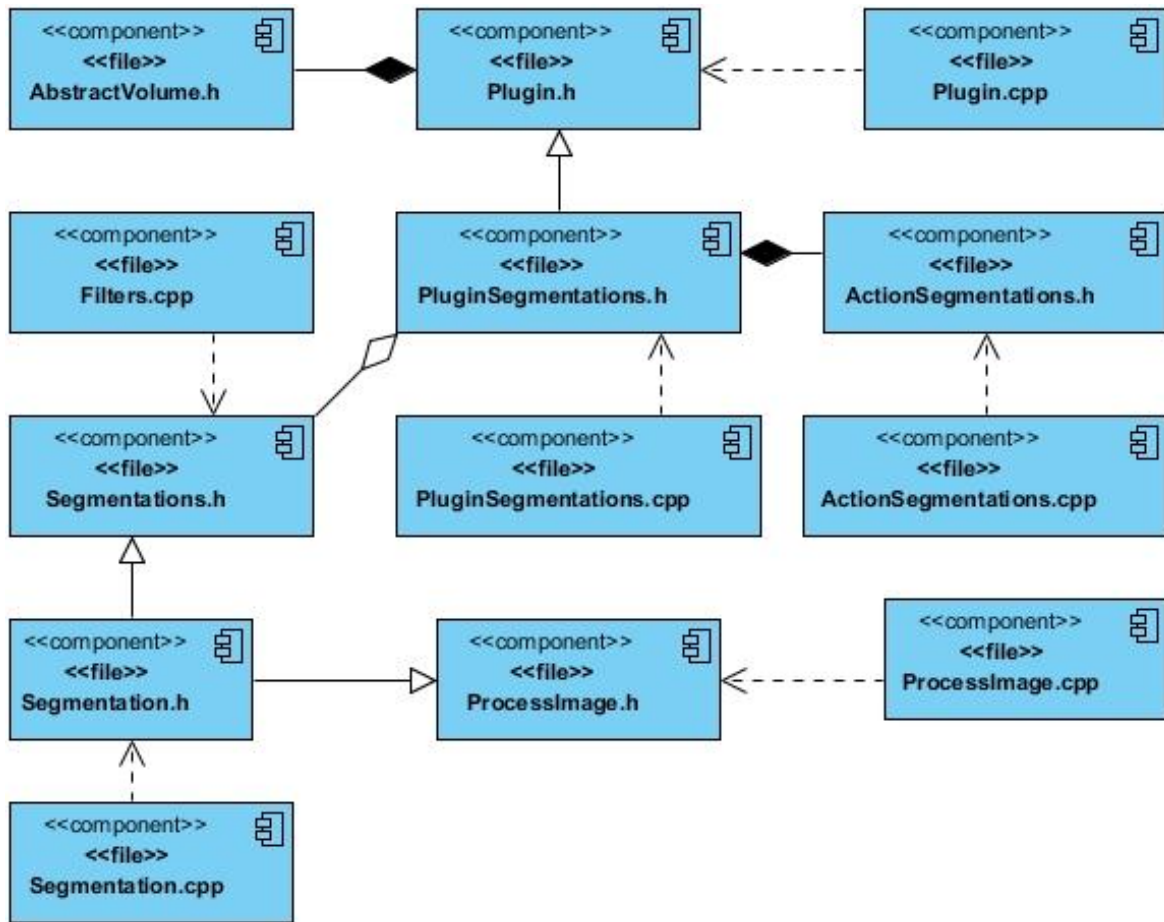


Figura 32. Diagrama de Componentes del subpaquete *plugin\_Segmentations*.

## 4.2 Presentación de los resultados

En este epígrafe se muestran los resultados que se obtienen con el módulo propuesto. A continuación se hace un análisis y se discuten los algoritmos de filtro y segmentación implementados. Todas las pruebas se realizan sobre una computadora personal con un procesador AMD *Dual-Core* E450 a una frecuencia de 2.70GHz con una memoria RAM DDR2 de 2 GB.

### 4.2.1 Entrada de datos

Para la realización de las pruebas se utilizaron 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.

# Capítulo 4. Implementación y Presentación de los Resultados

Tabla 8. Caso de prueba #1.

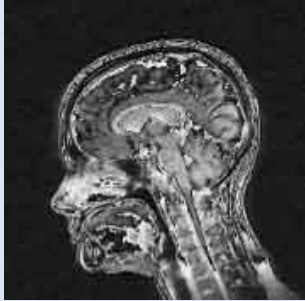
Identificador	Modalidad	Dimensiones	Imagen
Vista sagital del cráneo de un paciente.	RM	512x512	

Tabla 9. Caso de prueba #2.

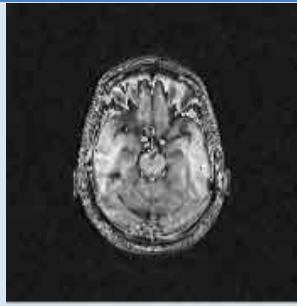

Identificador	Modalidad	Dimensiones	Imagen
Vista axial del cráneo de un paciente.	RM	512x512	

Tabla 10. Caso de prueba #3.

Identificador	Modalidad	Dimensiones	Imagen
Vista sagital del cráneo de un paciente.	RM	512x512	

## 4.2.2 Resultados obtenidos con la carga de imágenes

En el módulo propuesto se realiza la carga de las imágenes mediante la biblioteca ITK, convirtiéndose posteriormente a una estructura de datos genérica (*AbstractVolume*) para la utilización de los datos de los píxeles en otras etapas de la aplicación. A continuación se muestra como ocurre este proceso.

## Capítulo 4. Implementación y Presentación de los Resultados

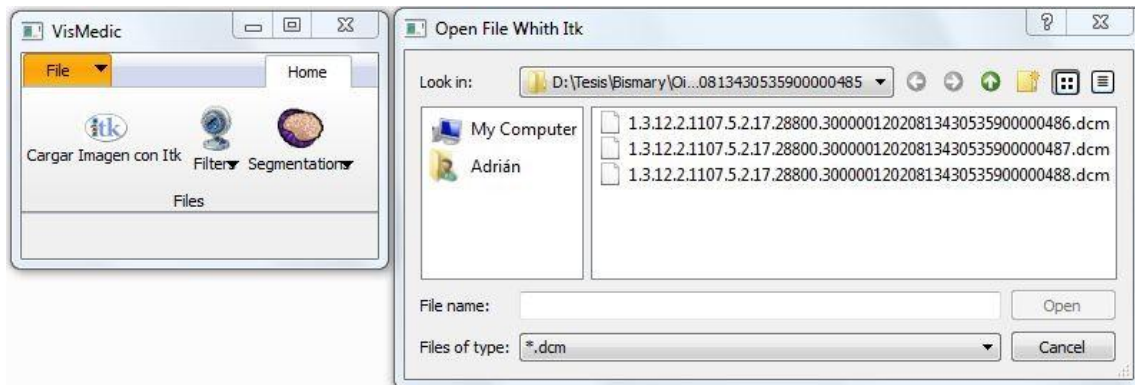


Figura 33. Cuadro de diálogo para cargar imagen.



Figura 34. Imagen DICOM del cráneo de un paciente cargada con ITK.

### 4.2.3 Resultados obtenidos al aplicar filtros a la imagen cargada

Durante las pruebas realizadas se aplicaron diferentes filtros, algoritmos de segmentación y combinaciones de estos, con el objetivo de observar el comportamiento de los resultados al cargar diferentes imágenes. A continuación se muestran los resultados obtenidos con las imágenes seleccionadas.

## Capítulo 4. Implementación y Presentación de los Resultados

### 4.2.3.1 Resultados de aplicar el filtro Promedio

El filtro Promedio suaviza la imagen en dependencia del radio que el usuario le modifique como parámetro. Esta técnica mejora la calidad de la imagen reduciendo algunos ruidos ocasionados por los equipos de adquisición de imágenes en las terminaciones de los contornos y suavizando los valores de intensidad de los píxeles. En la Figura 35 se puede observar el resultado al aplicar este filtro en una vista axial de un cráneo.

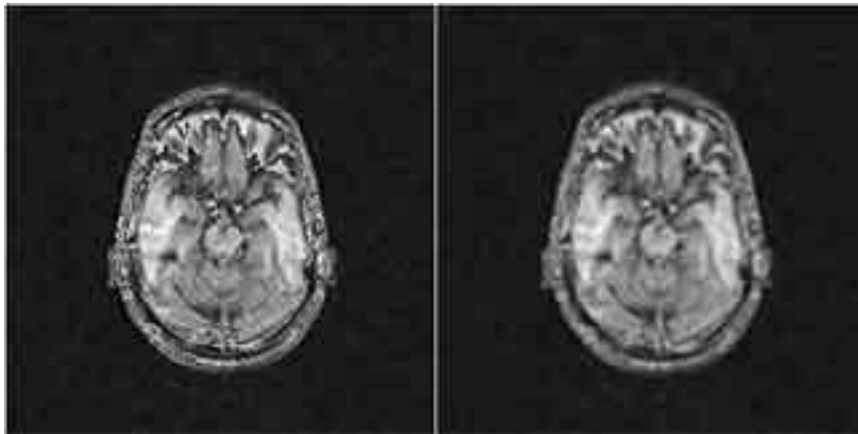


Figura 35. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Promedio.

### 4.2.3.2 Resultados al aplicar el filtro Umbral Binario

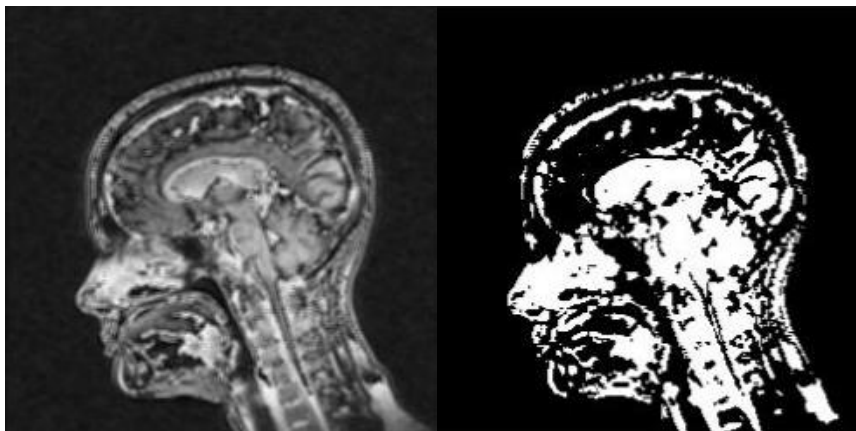
En la Figura 36 se evidencia el resultado de aplicar el filtro de Umbral Binario con los siguientes valores: valor lateral de salida = 0, valor lateral de entrada = 255, umbral superior = 115 y umbral inferior = 255. En este caso se observa como la intensidad de los píxeles de la imagen se reduce y solo se resaltan algunos elementos de la imagen original. Este resultado se obtiene aplicando solamente el filtro de Umbral Binario, es decir, sin aplicar ninguna combinación de otros filtros.





**Figura 36. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral Binario.**

En la Figura 37 se muestra el resultado de esta técnica de filtrado pero después de aplicar el filtro Promedio. Se ilustra la pérdida de detalles del borde superior de la imagen sagital, por la desventaja que posee el filtro Promedio, al suavizar la intensidad de los píxeles que están en los bordes de los contornos de la imagen sagital original.



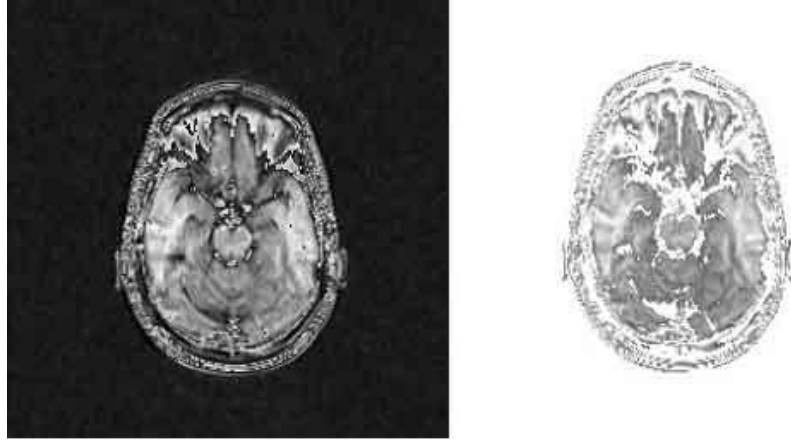
**Figura 37. Izquierda: Imagen DICOM al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral Binario.**

### 4.2.3.3 Resultados de aplicar el filtro Umbral General

Otro de los filtros que se aplicó es el Umbral General, el cual como se ha tratado en él Solución Propuesta tiene varios modos. Los modos dependen de sus valores en particular como se puede apreciar en la Figura

## Capítulo 4. Implementación y Presentación de los Resultados

38 se emplea el Modo Debajo con los parámetros: valor lateral de salida = 255 y umbral debajo = 94. Para estos valores, solo se muestran los píxeles que están por debajo del umbral definido.



**Figura 38. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Debajo (*Mode Below*).**

En la Figura 39 se muestra el resultado de aplicar el Modo Arriba del filtro Umbral General en el cual se muestran los píxeles cuyos valores de intensidad están por encima del umbral determinado por el especialista para resaltar las zonas específicas con las cuales se quiere trabajar. Para este caso se utilizaron los siguientes valores: valor lateral de salida = 255 y umbral arriba = 90.



**Figura 39. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Arriba (*Mode Above*).**

## Capítulo 4. Implementación y Presentación de los Resultados

En la Figura 40 se muestra el resultado de aplicar el filtro Umbral General en el Modo Fuera en el cual se muestran los píxeles que están fuera del rango definido. A la imagen original se le aplicó este filtro sin haberla filtrado antes. Para este caso se utilizaron los siguientes valores: valor lateral de salida = 0, fuera del mínimo lateral = 99 y fuera del máximo lateral = 160.



**Figura 40. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera (*Mode Out Side*).**

### 4.2.3.4 Resultados de aplicar el filtro de Detección de Bordes *Canny*

Este filtro se basa en la detección de los bordes de un grupo de píxeles que se encuentran asociados por su intensidad. En la Figura 41 se observa que el resultado del filtro de Detección de Bordes *Canny* sin aplicar ningún otro filtro de reducción de ruido no es bueno pues es abarcador incluyendo grupos de píxeles que no se centran en las estructuras anatómicas de la imagen DICOM en concreto, sino que están en las partes negras de esta imagen.

Sin embargo, este resultado se puede mejorar si se aplican varios filtros de reducción de ruido antes de aplicar el filtro de *Canny*. En la Figura 42 se muestran los resultados aplicando primero el filtro de Umbral General en Modo Debajo con los siguientes parámetros: valor de salida = 0 y umbral debajo = 63 y aplicando después el filtro Promedio se obtiene un resultado más preciso de la detección de bordes de los contornos de mayor interés en la imagen DICOM.

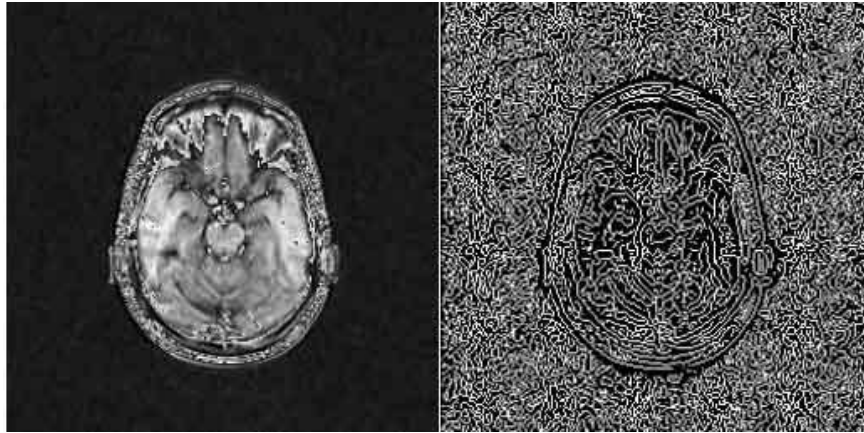


Figura 41. Izquierda: Imagen DICOM original. Derecha: Resultado de aplicar el filtro de Detección de Bordes *Canny*.

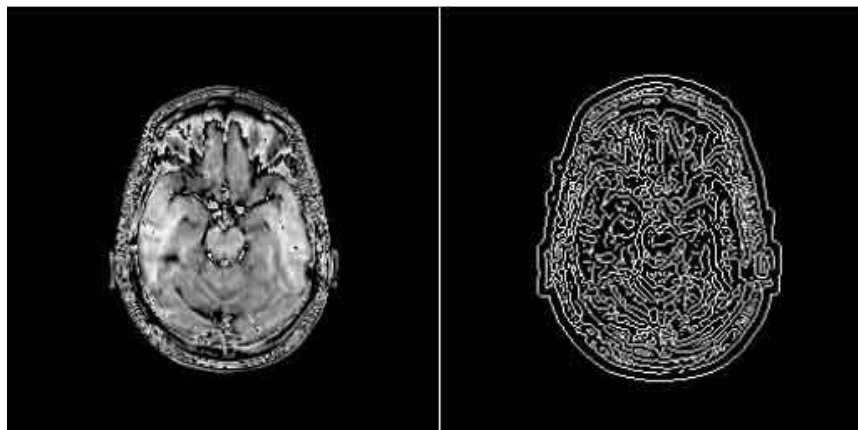


Figura 42. Resultado de aplicar el filtro de Detección de Bordes *Canny* a una imagen DICOM, después de aplicar otros filtros.

### 4.2.4 Resultados obtenidos al aplicar las segmentaciones a las imágenes cargadas

#### 4.2.4.1 Resultados al aplicar la técnica de segmentación Conectar Umbral

La técnica de segmentación Conectar Umbral a la que se hace referencia en el Capítulo 1 y Capítulo 2, tiene como resultado la selección de un grupo de píxeles a partir de un píxel semilla. En la Figura 43 se observa como se ha seleccionado solo una parte de la imagen teniendo en cuenta los siguientes valores:

## Capítulo 4. Implementación y Presentación de los Resultados

número de iteraciones = 5, umbral inferior = 100, umbral superior = 180 e índice del píxel semilla (171; 141; 0).

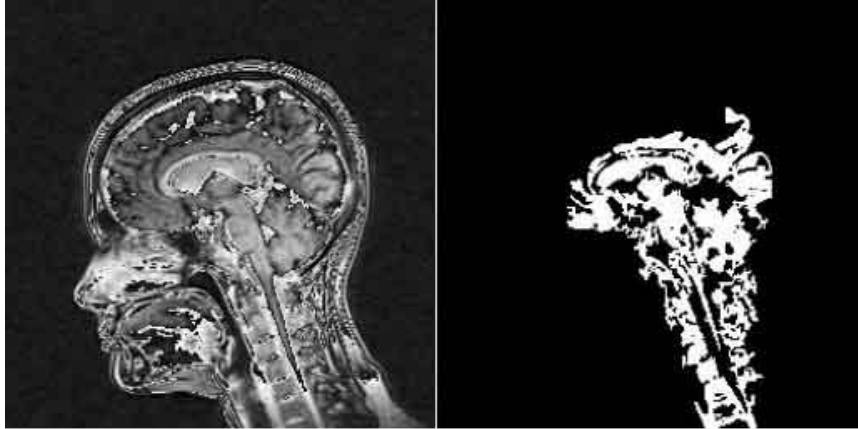


Figura 43. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.

### 4.2.4.2 Resultados al aplicar la técnica de Segmentación *Otsu*

El resultado de esta técnica de segmentación es similar a los resultados de los filtros de umbrales. En el Capítulo 1 y Capítulo 2 se puede consultar más información sobre dicha técnica. En la Figura 44 se puede apreciar el resultado de la técnica Segmentación *Otsu*.



Figura 44. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la Segmentación *Otsu*.

## Capítulo 4. Implementación y Presentación de los Resultados

### 4.2.4.3 Resultados al aplicar la técnica de segmentación Conectar Vecindad

La técnica de Conectar Vecindad es muy similar al conectar umbral, solo que este a partir de un píxel semilla agrupa a aquellos píxeles que se encuentran en su vecindad, dentro de un radio definido por el usuario. En la Figura 45 se muestra el resultado de aplicar el filtro Conectar Vecindad con los siguientes valores: número de iteraciones = 5, umbral inferior = 100, umbral superior = 180, radio = 1 e índice del píxel semilla (116; 126; 0).

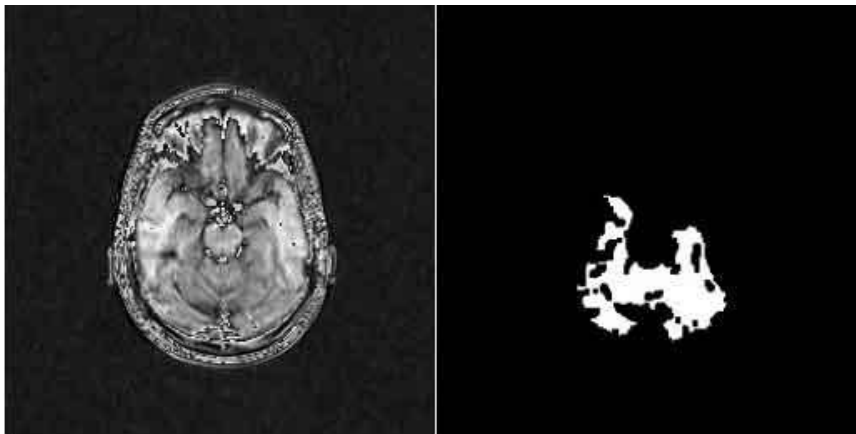


Figura 45. Izquierda: Imagen DICOM original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Vecindad.

### 4.2.1 Resultados obtenidos al aplicar filtros y segmentaciones a imágenes estandarizadas

Existen imágenes estandarizadas para realizar pruebas a los algoritmos de filtrado y de segmentación. A continuación se muestra el resultado de las técnicas implementadas sobre estas imágenes.

En la Figura 46 se muestra el resultado de aplicar el filtro Umbral Binario a una imagen estandarizada para estas pruebas con los siguientes valores: valor lateral de salida = 110, valor lateral de entrada = 255, umbral superior = 115 y umbral inferior = 255. En este caso, no se aplicó anteriormente ningún otro filtro, es por ello que en algunos contornos hay pérdida de datos.

## Capítulo 4. Implementación y Presentación de los Resultados



**Figura 46. Izquierda: Imagen Bárbara original. Derecha: Resultado de aplicar el filtro Umbral Binario.**

La Figura 47 muestra los resultados de aplicar el filtro Promedio a la imagen Bárbara antes de aplicar la técnica de filtrado de Umbral Binario. Observe que se obtiene como resultado una imagen con mayor pérdida de detalles debido a la desventaja del filtro Promedio. Se pueden obtener otros resultados superiores al aplicar otros filtros diferentes del Promedio.



**Figura 47. Izquierda: Imagen Bárbara al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral Binario.**

En la Figura 48 se realiza el mismo proceso a la imagen Lena y en la Figura 49 se muestra el resultado de aplicar la misma técnica de filtrado pero después de aplicar el filtro Promedio, en el resultado obtenido se puede observar la diferencia en la calidad del filtrado. Para este caso se utilizaron los siguientes valores: valor lateral de salida = 255, fuera del mínimo lateral = 48 y fuera del máximo lateral = 112.

## Capítulo 4. Implementación y Presentación de los Resultados



**Figura 48. Izquierda: Imagen Lena original. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera.**



**Figura 49. Izquierda: Imagen Lena al aplicar el filtro Promedio. Derecha: Resultado de aplicar el filtro Umbral General en Modo Fuera.**

En la Figura 50 se muestra el resultado al emplear la técnica Conectar Umbral a la imagen *Clown* sin aplicar ningún filtro con los siguientes valores: número de iteraciones = 5, umbral inferior = 100, umbral superior = 180 e índice del píxel semilla (166; 105; 0). Para estos valores el algoritmo solo selecciona algunos píxeles de la imagen. En la Figura 51 se muestra el resultado sobre la misma imagen y con los mismos valores, después de aplicar el filtro Promedio. En este caso, la selección de píxeles resultante es superior pero no muestra una gran mejora de la calidad, siendo necesario aplicar otros filtros sobre la misma imagen para que el algoritmo obtenga mejores resultados.



## Capítulo 4. Implementación y Presentación de los Resultados



Figura 50. Izquierda: Imagen *Clown* original. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.



Figura 51. Izquierda: Imagen *Clown* al aplicar el filtro Promedio. Derecha: Resultado al aplicar la técnica de segmentación Conectar Umbral.

### Conclusiones

- Se desarrolló un módulo que implementa una interfaz de comunicación común que permite filtrar y segmentar imágenes médicas en formato DICOM, así como agregar nuevos algoritmos de filtrado y segmentación sin necesidad de modificar el resto de la aplicación.
- Se comprobó que la combinación de algoritmos de filtros y segmentación produce un resultado de mejor calidad en la etapa de visualización.

### Recomendaciones

- Agregar nuevos algoritmos de filtrado y segmentación al módulo que permita extender su usabilidad como: el filtro Magnitud del Gradiente, filtro de la Mediana, filtro *Laplacian*, segmentación basada en *Watershed*, que emplea ITK, además de las técnicas de detección de bordes *Sobel*, *Prewitt* y *Robert*.

## Bibliografía

1. NLM Insight Segmentation & Registration Toolkit. [En línea] <http://www.itk.org/>.
2. C. González, Rafael y E. Woods, Richard. *Digital Image Processing*. Second Edition. New Jersey: Prentice Hall : s.n., 2002.
3. González Zaldivar, Alcides. *Segmentación de Imágenes Médicas, mediante Técnicas de Contornos y Técnicas de Regiones*. 2010.
4. Preim, Bernhard y Bartz, Dirk. *Visualization in Medicine Theory, Algorithms, and Applications*. Elsevier: s.n., 2007. 978-0-12-370596-9.
5. Chaple, Andrés Miralles. *Módulo de Segmentación de Imágenes Médicas Digitales Mediante Técnicas de Watershed*. Habana: s.n., 2011.
6. Villamizar, Andrea. CREA IMAGEN. [En línea] <http://creaimagenandrapublicidad.blogspot.com/>.
7. Tovar Carrillo, Luis Carlos. Sistema de reconocimiento de objetos mediante imágenes aplicado a un brazo robótico. México, D.F: s.n., 2009.
8. Pereira Barzaga, Osvaldo y Kindelan Núñez, Rolando. Reconstrucción Tridimensional de Modelos Anatómicos a partir de Imágenes Médicas Digitales. Habana: s.n., 2008.
9. Coto, Ernesto. Métodos de Segmentación de Imágenes Médicas. 2005.
10. Jiménez, J. M. *Introducción al Tratamiento Digital y Clustering de Imágenes*. 2008.
11. Gómez, Ernesto Fuentes. *Segmentación de imágenes médicas*. 2012.
12. Ibáñez, Luis, y otros. *The ITK Software Guide Second Edition Updated for ITK version 2.4*.
13. ToolKit, Visualization. Kitware. Inc. [En línea] <http://public.kitware.com/VTK/>.
14. F. del Moral, A. Ramos, M. Salgado, B. Andrade, V Muñoz. SEFM. [En línea] [http://www.sefm.es/fisica-medica/es/download/?n=2010\\_3\\_11\\_cuantificacion-influencia&idf=452\\_400\\_pdf\\_docrev](http://www.sefm.es/fisica-medica/es/download/?n=2010_3_11_cuantificacion-influencia&idf=452_400_pdf_docrev).
15. ITK. [En línea] <http://www.itk.org/HTML/Copyright.htm>.
16. Open Source Initiative. [En línea] <http://www.opensource.org/licenses/bsd-license.php>.
17. The FreeBSD Project. [En línea] <http://www.freebsd.org/copyright/freebsd-license.html>.

18. Lorca, Gustavo Toranzo. *Módulo de Segmentación de Imágenes Médicas*.
19. MDSTk Home Page. [En línea] <http://mdstk.sourceforge.net/home.html>.
20. Vélez Serrano, José F. y otros. *Técnicas avanzadas de diseño de software: Orientación a objetos, UML, patrones de diseño y Java*.
21. Dofactory. [En línea] <http://www.dofactory.com/Default.aspx>.
22. Rojas, Luis Guillermo Silva. *Visualización directa de volumen para endoscopias virtuales*. 2011.
23. *Rational Unified Process*. Corporation, IBM. 2006.
24. Booch, Grady, Jacobson, Ivar y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. s.l. Addison Wesley. 2000.
25. *Ayuda de Rational Unified Process*.
26. Definición ABC. [En línea] <http://www.definicionabc.com/tecnologia/imagen-digital.php>.
27. Enciclopedia de salud dietética y psicología. [En línea]. <http://www.encyclopediasalud.com/categorias/cuerpo-humano/articulos/imagen-por-tomografia-axial-computerizada-tac>.
28. COMIEX. *Centro de Investigación y Extensión*. [En línea] <http://comiex.epn.edu.ec/index.php/semilla/81-conveccion-natural-fluidos-jcr>.
29. Ray, Tinku Acharya & Ajoy K. *Image Processing Principles and Applications*.

### 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

**Binario:** Está compuesto por dos elementos o unidades: el sistema de numeración que utilizan los ordenadores es binario, solamente utiliza los dígitos 0 y 1.

**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.

**Convolución:** En matemáticas y, en particular, análisis funcional, una convolución es un operador matemático que transforma dos funciones  $f$  y  $g$  en una tercera función que en cierto sentido representa la magnitud en la que se superponen  $f$  y una versión trasladada e invertida de  $g$ .

#### D

**Diagnóstico:** Etimológicamente el concepto diagnóstico proviene del griego, tiene dos raíces, día- que es a través de, 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.

### H

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

**Híbridas:** Que es originado a partir de elementos de distinta naturaleza: así nació el *cómic*, un híbrido entre literatura e ilustración.

**Homogéneas:** Que está formado por elementos con una serie de características comunes referidas a su clase o naturaleza que permiten establecer entre ellos una relación de semejanza: la formación académica de alumnos de un mismo curso es homogénea.

### I

**Imagen:** Figura, representación, semejanza y apariencia de algo. En computación es formada por la unión de  $M \times N$  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

**Matriz:** Conjunto de números colocados en líneas horizontales y verticales, dispuestos en forma de rectángulo; la posición de cada número en la matriz determina las operaciones matemáticas que hay que hacer para hallar un resultado.

**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.

**Plugin:** Se conoce también como un complemento que se relaciona con otra para aportarle de una función nueva y generalmente muy específica.

### 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.

### U

**Umbral:** Nivel de sensibilidad mínima de un sentido.

### 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.