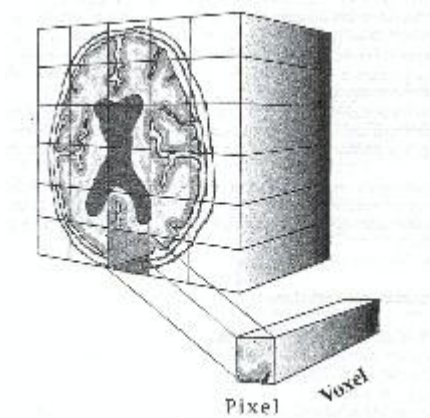


# Universidad de las Ciencias Informáticas

## Facultad 5



### Reconstrucción Tridimensional de Modelos Anatómicos a partir de Imágenes Médicas Digitales.



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

**Autor(es):** Osvaldo Pereira Barzaga.

Rolando Kindelan Nuñez.

**Tutor(es):** Lic. Juan Manuel Mederos Martínez.

Ing. Osmanys Valdes Puga.

**Junio 2008**

*“La virtud, como el arte, se consagra constantemente a lo que es difícil de hacer, y cuanto más dura es la tarea, más brillante es el éxito.”*

*Aristóteles.*

## Declaración de Autoría.

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del Autor: Osvaldo Pereira Barzaga.

---

Firma del Autor: Rolando Kindelan Nuñez.

---

Firma del Tutor: Juan Manuel Medero  
Martínez

---

Firma del Tutor: Osmanys Valdes Puga.

## Datos de Contacto.

**Tutor:** Lic. Juan Manuel Medero Martínez.

**Edad:** 26 años.

**Ciudadanía:** cubano

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

**Título:** Licenciado en Ciencias de la Computación

**Categoría Docente:** Profesor Instructor

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

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

**Tutor:** Ing. Osmanys Valdes Puga.

**Edad:** 25 años.

**Ciudadanía:** cubano.

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

**Título:** Ingeniero en Ciencias Informáticas.

**Categoría Docente:** Profesor Adiestrado

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

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

## Dedicatoria.

*A mi madre:*

*Que con gusto daría un año de su felicidad por evitarme un segundo de dolor, que sería capaz de mendigar por mí y se mataría por salvarme.*

*A mi padre:*

*Que fue siempre mi mejor amigo, que cuando se veía obligado a reprenderme o castigarme sufría más que yo, guiándome siempre por el mejor camino y no deseando otra cosa que mi bien.*

**Oswaldo**

*Quiero dedicar este trabajo a todas las personas que han estado a mi lado aún cuando no los necesitaba.*

*A mi madre, esa fiel compañera que supo ser mi familia entera aún sabiendo que ya la tenía, que ha sido mi más fiel amiga, mi confidente, mi confesora, por inculcarme quizás a propósito quizás sin querer, el deseo de ser alguien en la vida y de intentar hacer correctamente cada cosa que hiciera.*

*A mi padre, por ser también más que un padre, un amigo; por ser aquella persona que siempre tuvo la palabra o la frase perfecta para hacerme entrar en razón, e inculcarme desde pequeño las ansias de aprender. Gracias por decir en algún momento que estudiara mucho.*

*En fin, esto va dedicado a todos aquellos que han confiado en mí y a los que no también, pues su actitud me ayudó a demostrarles que sí se puede.*

**Rolando**

## **Agradecimientos.**

*Quisiera agradecerles a todos, lo que ahora soy, a aquellas personas que de una forma u otra, tanto en las clases como en la calle y en todas partes me han ayudado a formarme como todo un profesional.*

### **En especial a mis padres:**

*A ti, madre, amado ángel de mi guarda, que has estado siempre en todas mis alegrías y sufrimientos, en mis amarguras, que has estudiado, te has cansado y has llorado conmigo, acariciándome con una mano la frente e indicándome con la otra el camino correcto a seguir.*

*A ti, padre, mi primer maestro, amigo y confidente, que me has dado tantos buenos consejos y me has enseñado tantas cosas mientras trabajabas por mí, ocultándome siempre tus tristezas y tratando por todos los medios de hacerme fácil el estudio y bella la vida.*

### **A mi novia:**

*Yadira, que estuvo siempre a mi lado sin importar cuán difícil fuera el momento y me dió las fuerzas y el amor necesario para salir adelante.*

### **A mis maestros:**

*Mis buenos maestros, que tan indulgentes y cariñosos se han mostrado conmigo, para quienes han trabajado por cada nuevo conocimiento que he adquirido y que ahora es para mí motivo de satisfacción y de sano orgullo, en especial a nuestro tutor Juan Manuel.*

### **A la UCI:**

*La escuela que me tomó de sus brazos cuando apenas sabía programar y me devuelve ahora a la revolución, convertido en un profesional.*

### **A mis compañeros y amigos:**

*También les agradezco a ustedes, mis admirables compañeros de clases, las explicaciones con que me han hecho comprender de amable manera tantas veces las cosas.*

*A Rolo, mi compatriota incansable en la lucha por la superación científica, que me ha demostrado que con voluntad todo se alcanza.*

*A ustedes, mis amigos, Angelito, Rafa, Ailincita y Yurima que se ganan la simpatía y la admiración de cuántos los conocen.*

*También a mis compañeros de Laboratorio: Yalina, Yulien, Ismael, Hassan, Sandor, Carlos, Yordanis, Pacheco, Marvin y Andy; que siempre han sido ejemplo de abnegación y de constancia en el trabajo.*

*A todos ustedes y a todos los demás que por motivo de espacio no he nombrado, le agradezco de todo corazón, por ayudarme a ser quién soy... por eso y mucho más, hoy y siempre me arrodillo ante ustedes, y les doy gracias con toda la ternura que han puesto en mi alma durante estos cinco años de sacrificio y de amor.*

## **Oswaldo**

*Un famoso clérigo inglés llamado Charles Caleb Colton, dijo en una ocasión que: "...Ningún metafísico ha experimentado nunca tantas dificultades para expresarse, como el hombre agradecido...". Jamás imaginé que fuera tan complicado para mí: no, agradecer (porque es algo que sale del corazón y da un inmenso placer hacerlo y uno mayor recibirlo), sino por el temor de que en mi calidad de ser humano, cometa el error de olvidar a alguna de las tantas personas que han tenido que ver con que este día haya sido posible.*

*Quisiera dar las gracias a cada una de aquellas personas que han participado de una forma u otra a lo largo de mi vida. Porque siento que si algo no hubiera sucedido exactamente igual, y si no hubiera intercambiado y convivido con las mismas personas, todo sería diferente: esto implica que no hubiera estudiado en la Universidad de las Ciencias Informáticas ni hubiera sido posible hacerme ingeniero.*

*Es por eso que me gustaría agradecer primeramente a Fidel y a Raúl Castro, por creer que un mundo mejor es posible y luchar tantos años por mejorarlo, me siento como un fruto de la semilla que una vez plantaron, que está madurando y pretende ser simiente de futuros frutos. A mis padres, por ser más que eso para mí, por esforzarse tanto conmigo y guiarme por el camino correcto. A Yorley, que a pesar de no hacer nada por la tesis, ha hecho mucho por mí. A mis compañeros de aula, por soportarme durante estos largos 5 años. A*

*los laboratorios de la UCI, por no cansarse de mí y brindarme los medios necesarios para llegar hasta hoy. A Osvaldo, por ser mi compañero de tesis y además de lucha científica. A Juan, no solo por ser tutor, sino por su manera de serlo. A Osmanys, por los consejos y sugerencias.*

*Agradezco a todas aquellas amistades que puedo recordar, y a las que no también, porque han sido parte importante de mi vida.*

*"Lo que hizo más importante tu rosa, es el tiempo que pasaste con ella..." -dijo el zorro al principito en el libro de Antoine De Saint Exupéry-. Así pudiera decir yo: fue el tiempo que pasé con ustedes, lo que los hizo más importantes. Gracias por todo.*

**Rolando**



## Resumen.

La reconstrucción tridimensional de modelos anatómicos es un proceso fundamental en el área de la medicina. El proceso que se sigue para realizar la reconstrucción consta de dos partes fundamentales, la segmentación y la triangulación o reconstrucción propiamente dicha.

La segmentación de imágenes tridimensionales es un proceso que se encarga de reconocer objetos o estructuras de interés dentro de las imágenes médicas y constituye la etapa de mayor peso en este tipo de aplicaciones, por lo tanto hay que orientar la mayor cantidad de esfuerzo en su realización. Aunque se han planteado muchas alternativas para resolver el problema, aún no existe una que pueda cubrir todas las necesidades. En este trabajo se propone un método novedoso de segmentación de imágenes digitales 3D, y basado en un algoritmo de crecimiento de regiones, conocido como Voxel Growing, con la opción de la incorporación de conocimiento adicional.

La triangulación por su parte, toma como punto de entrada los resultados del proceso de segmentación, y la calidad de sus resultados es directamente proporcional a la calidad de la segmentación. Su principal objetivo es la generación de una malla que reproduzca lo mejor posible las estructuras anatómicas reales. El algoritmo desarrollado está basado en vóxeles y es una especie de híbrido de algunos algoritmos y técnicas clásicas de reconstrucción.

Se creó el formato de fichero \*.KPS, para almacenar los modelos obtenidos como resultado del proceso de reconstrucción, y puedan ser utilizados posteriormente por terceras aplicaciones.

### **Palabras clave:**

Reconstrucción, Segmentación, Tridimensional, Voxel Growing, Vóxel, KPS, Estructuras Anatómicas, Malla, Modelo, Generación, Regiones, Imágenes Médicas, DICOM, 3D, Simulador Quirúrgico, Realidad Virtual.

# Índice.

|                                                             |             |
|-------------------------------------------------------------|-------------|
| <b>Declaración de Autoría.</b> .....                        | <b>I</b>    |
| <b>Datos de Contacto</b> .....                              | <b>II</b>   |
| <b>Dedicatoria</b> .....                                    | <b>III</b>  |
| <b>Agradecimientos.</b> .....                               | <b>IV</b>   |
| <b>Resumen.</b> .....                                       | <b>VII</b>  |
| <b>Índice.</b> .....                                        | <b>VIII</b> |
| <b>Índice de Figuras</b> .....                              | <b>XIII</b> |
| <b>Índice de Tablas.</b> .....                              | <b>XV</b>   |
| <b>Introducción.</b> .....                                  | <b>1</b>    |
| <b>Capítulo 1. Fundamentación Teórica</b> .....             | <b>8</b>    |
| 1.1. Segmentación de Imágenes. ....                         | 8           |
| 1.1.1. Tipos de Segmentación de Imágenes. ....              | 9           |
| 1.1.2. Técnicas de Segmentación de Imágenes. ....           | 9           |
| 1.1.2.1. Técnicas de Detección de Discontinuidades. ....    | 9           |
| 1.1.2.2. Técnicas de Umbrales. Umbralización. ....          | 11          |
| 1.1.2.3. Técnicas de Orientadas a Regiones.....             | 12          |
| 1.2. Reconstrucción Tridimensional. ....                    | 13          |
| 1.2.1. Algoritmos basados en Marching.....                  | 13          |
| 1.2.1.1. Marching Cubes. ....                               | 14          |
| 1.2.1.2. Marching Tetrahedra. ....                          | 15          |
| 1.2.2. Algoritmos basados en Triangulación de Delaunay..... | 16          |
| 1.2.2.1. Teselación de Delaunay Extendida. ....             | 17          |
| 1.2.2.2. Triangulación de Delaunay Restringida.....         | 19          |
| 1.3. Visualización. ....                                    | 19          |
| 1.3.1. Unidad de Hounsfield. ....                           | 20          |
| 1.3.2. Surface Rendering.....                               | 21          |
| 1.3.3. Volume Rendering. ....                               | 21          |

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| 1.4. Formatos de Ficheros. ....                                    | 22        |
| 1.4.1. Ficheros de Gráficos 3D. ....                               | 22        |
| 1.4.1.1. Características de los 3DS. ....                          | 22        |
| 1.4.1.2. Características de los ASE. ....                          | 23        |
| 1.4.2. Especificación del Estándar DICOM. ....                     | 23        |
| 1.4.2.1. Características de los DICOM. ....                        | 24        |
| 1.4.2.2. Formato de un Archivo DICOM. ....                         | 24        |
| 1.5. Principales Bibliotecas de Clases. ....                       | 25        |
| 1.5.1. Imebra. ....                                                | 25        |
| 1.5.2. DicomLib. ....                                              | 26        |
| 1.5.3. VTK. ....                                                   | 27        |
| 1.5.4. ITK. ....                                                   | 27        |
| 1.5.5. TetGen. ....                                                | 28        |
| 1.6. Metodologías y Herramientas de Desarrollo. ....               | 28        |
| 1.6.1. Metodología de Ingeniería de Software. ....                 | 28        |
| 1.6.2. Herramientas de Desarrollo. ....                            | 29        |
| 1.6.3. Lenguajes. ....                                             | 29        |
| 1.6.3.1. Lenguaje de Programación. ....                            | 29        |
| 1.6.3.2. Lenguaje de Modelado. ....                                | 29        |
| <b>Capítulo 2. Solución Propuesta. ....</b>                        | <b>31</b> |
| 2.1. ¿Por qué no utilizar las librerías de clases existentes?..... | 31        |
| 2.2. Construcción de las Imágenes Tridimensionales. ....           | 32        |
| 2.3. Segmentación de Imágenes 3D. Voxel Growing.....               | 34        |
| 2.3.1. Especificación de la Semilla. ....                          | 35        |
| 2.3.2. Aceptación de Vóxeles. ....                                 | 36        |
| 2.3.3. Incorporación de Vóxeles. ....                              | 37        |
| 2.4. Reconstrucción de Superficie. ....                            | 39        |
| 2.4.1. Triangulación Inicial. ....                                 | 39        |
| 2.4.1.1. Tabla Hash Cúbica .....                                   | 41        |

|                                                                        |           |
|------------------------------------------------------------------------|-----------|
| 2.4.2. Suavizado de la Malla.....                                      | 41        |
| 2.5. Especificación del formato de fichero a exportar. ....            | 43        |
| 2.5.1. Estructura Interna. ....                                        | 43        |
| 2.5.1.1. Bloques. ....                                                 | 43        |
| 2.5.1.2. Encabezado.....                                               | 45        |
| 2.5.1.3. Cuerpo.....                                                   | 45        |
| 2.5.1.4. La Región. ....                                               | 46        |
| 2.5.1.5. Geometría. ....                                               | 47        |
| 2.5.1.6. Materiales.....                                               | 47        |
| 2.6. Visualización. OpenGL y Raytracing.....                           | 48        |
| 2.6.1. Tablas de Opacidad.....                                         | 48        |
| 2.6.2. Tablas o Mapas de Color.....                                    | 49        |
| <b>Capítulo 3. Descripción de la Solución Propuesta.....</b>           | <b>50</b> |
| 3.1. Reglas del Negocio.....                                           | 50        |
| 3.2. Modelo de Dominio. ....                                           | 50        |
| 3.3. Glosario de Términos.....                                         | 51        |
| 3.4. Captura de Requisitos.....                                        | 53        |
| 3.4.1. Requisitos Funcionales.....                                     | 53        |
| 3.4.2. Requisitos No Funcionales. ....                                 | 54        |
| 3.5. Modelo de Casos de Uso del Sistema. ....                          | 55        |
| 3.5.1. Actores del Sistema.....                                        | 55        |
| 3.5.2. Diagrama de Casos de Uso del Sistema. ....                      | 56        |
| 3.5.3. Descripción de Casos de Uso del Sistema. ....                   | 57        |
| <b>Capítulo 4. Análisis y Diseño del Sistema. ....</b>                 | <b>64</b> |
| 4.1. Diagrama de Clases de Análisis. ....                              | 64        |
| 4.2. Diagramas de Secuencia. ....                                      | 65        |
| 4.2.1. Diagrama de Secuencia del Caso de Uso Cargar Fichero. ....      | 66        |
| 4.2.2. Diagrama de Secuencia del Caso de Uso Segmentar.....            | 67        |
| 4.2.3. Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. .... | 68        |

|                                                                                     |           |
|-------------------------------------------------------------------------------------|-----------|
| 4.3. Diagrama de Clases del Diseño del Sistema. ....                                | 70        |
| 4.3.1. Diagrama de Clases del Paquete DICOM. ....                                   | 71        |
| 4.3.2. Diagrama de Clases del Paquete Segmentación. ....                            | 72        |
| 4.3.3. Diagrama de Clases del Paquete Reconstrucción.....                           | 73        |
| 4.4. Diagrama de Secuencia del Diseño. ....                                         | 74        |
| 4.4.1. Diagrama de Secuencia del Diseño del Caso de Uso Cargar Fichero. ...         | 74        |
| 4.4.2. Diagrama de Secuencia del Diseño del Caso de Uso Segmentar.....              | 75        |
| 4.4.3. Diagrama de Secuencia del Diseño del Caso de Uso Gestionar Modelo<br>3D..... | 76        |
| <b>Capítulo 5. Implementación y Resultados. ....</b>                                | <b>79</b> |
| 5.1. Diagrama de Componentes. Paquetes de Implementación. ....                      | 79        |
| 5.1.1. Componentes del Paquete de Implementación DICOM. ....                        | 80        |
| 5.1.2. Componentes del Paquete de Implementación Segmentación. ....                 | 81        |
| 5.1.3. Componentes del Paquete de Implementación Reconstrucción.....                | 82        |
| 5.2. Valoración de los Resultados.....                                              | 83        |
| 5.2.1. Datos de entrada. ....                                                       | 83        |
| 5.2.2. Parámetros a medir. ....                                                     | 84        |
| <b>Conclusiones.....</b>                                                            | <b>85</b> |
| <b>Recomendaciones.....</b>                                                         | <b>87</b> |
| <b>Bibliografía. ....</b>                                                           | <b>88</b> |
| <b>Glosario.....</b>                                                                | <b>93</b> |
| <b>Anexos. ....</b>                                                                 | <b>99</b> |
| Anexo A: Segmentación de Imágenes.....                                              | 99        |
| A.1 Detección de Puntos. ....                                                       | 99        |
| A.2 Detección de Líneas.....                                                        | 99        |
| A.3 Detección de Bordos. ....                                                       | 100       |
| A.4 Técnicas de Umbrales. Umbralización. ....                                       | 105       |
| A.5 Segmentación Orientada a Regiones.....                                          | 108       |
| Anexo B: Reconstrucción.....                                                        | 111       |
| B.1 Algoritmos Basados en Marching.....                                             | 111       |

|                                                     |     |
|-----------------------------------------------------|-----|
| B.1.1 Marching Cube.....                            | 111 |
| B.1.2 Marching Tetrahedra.....                      | 112 |
| Anexo C: Formato de Ficheros. ....                  | 112 |
| C.1 Especificación y Características del DICOM..... | 112 |
| Anexo D: Imágenes de la Aplicación. ....            | 117 |

# Índice de Figuras.

|                                                                                                                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Figura 1.1</b> <i>Máscara 3x3.</i> .....                                                                                                                                                                                | 10 |
| <b>Figura 1.2</b> <i>Vista 2D de una isosuperficie.</i> .....                                                                                                                                                              | 14 |
| <b>Figura 1.3</b> <i>Casos de Triangulación Marching Cubes.</i> .....                                                                                                                                                      | 15 |
| <b>Figura 1.4</b> <i>Casos de Triangulación Marching Tetrahedra.</i> .....                                                                                                                                                 | 15 |
| <b>Figura 1.5</b> <i>Triangulación de Delaunay.</i> .....                                                                                                                                                                  | 17 |
| <b>Figura 1.6</b> <i>Polígonos circulares en 2D y dos vistas de un poliedro esférico en 3D.</i> .....                                                                                                                      | 18 |
| <b>Figura 1.7</b> <i>Triangulación de Delaunay Restringida, Nube de puntos (a) y reconstrucción del PLC (b).</i> .....                                                                                                     | 19 |
| <b>Figura 1.8</b> <i>Umbrales. Todos los tejidos por encima del umbral se mostrarán, los que estén por debajo serán eliminados.</i> .....                                                                                  | 20 |
| <b>Figura 1.9</b> <i>Coeficientes de atenuación medidos en Unidades de Hounsfield.</i> .....                                                                                                                               | 21 |
| <b>Figura 1.10</b> <i>Estructura de un archivo DICOM (*.dcm).</i> .....                                                                                                                                                    | 25 |
| <b>Figura 2.1</b> <i>Representación de una Imagen 3D.</i> .....                                                                                                                                                            | 33 |
| <b>Figura 2.2</b> <i>Obtención del Volumen final de vóxeles que será luego segmentado.</i> .....                                                                                                                           | 34 |
| <b>Figura 2.3</b> <i>Tipos de conectividades entre un vóxel y sus vecinos.</i> .....                                                                                                                                       | 38 |
| <b>Figura 2.4</b> <i>Triangulación de las caras frontera de los vóxeles.</i> .....                                                                                                                                         | 40 |
| <b>Figura 2.5</b> <i>Ejemplo de matriz cúbica, que almacena solo los vértices v1 (minx+2, miny+2, minz) y v2 (minx+3, miny+3, minz) que ocupan las posiciones 103 y 204 respectivamente en la lista de vértices.</i> ..... | 41 |
| <b>Figura 2.6</b> (a) <i>Descripción poligonal de la superficie. Triangulación Inicial.</i> (b) <i>Descripción poligonal de la superficie. Superficie Suavizada.</i> .....                                                 | 42 |
| <b>Figura 2.7</b> <i>Estructura del Encabezado de un fichero *.KPS.</i> .....                                                                                                                                              | 45 |
| <b>Figura 2.8</b> <i>Estructura del Cuerpo de un fichero *.KPS.</i> .....                                                                                                                                                  | 46 |
| <b>Figura 2.9</b> (a) <i>Tablas de Opacidad.</i> (b) <i>Tablas o Mapas de Color.</i> .....                                                                                                                                 | 48 |
| <b>Figura 2.10</b> <i>Curvas de Opacidad.</i> .....                                                                                                                                                                        | 49 |
| <b>Figura 3.1</b> <i>Modelo de dominio de la aplicación.</i> .....                                                                                                                                                         | 51 |
| <b>Figura 3.2</b> <i>Diagrama de Casos de Uso del Sistema.</i> .....                                                                                                                                                       | 56 |
| <b>Figura 4.1</b> <i>Diagrama General de Clases de Análisis.</i> .....                                                                                                                                                     | 65 |
| <b>Figura 4.2</b> <i>Diagrama de Secuencia Caso de Uso Cargar Fichero.</i> .....                                                                                                                                           | 66 |
| <b>Figura 4.3</b> <i>Diagrama de Secuencia Caso de Uso Segmentar.</i> .....                                                                                                                                                | 67 |

|                                                                                                                                      |    |
|--------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Figura 4.4</b> <i>Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Generación del Modelo.</i> .....             | 68 |
| <b>Figura 4.5</b> <i>Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Exportar Modelo.</i> .....                   | 69 |
| <b>Figura 4.6</b> <i>Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Importar Modelo.</i> .....                   | 69 |
| <b>Figura 4.7</b> <i>Diagrama General de Paquetes de Clases del Diseño.</i> .....                                                    | 70 |
| <b>Figura 4.8</b> <i>Diagrama de Clases de Diseño Paquete de DICOM.</i> .....                                                        | 71 |
| <b>Figura 4.9</b> <i>Diagrama de Clases de Diseño del Paquete de Segmentación.</i> .....                                             | 72 |
| <b>Figura 4.10</b> <i>Modelo de Clases del Diseño del Paquete de Reconstrucción.</i> .....                                           | 73 |
| <b>Figura 4.11</b> <i>Diagrama de Secuencia del Diseño del Caso de Uso Cargar Fichero.</i> .....                                     | 74 |
| <b>Figura 4.12</b> <i>Diagrama de Secuencia del Diseño del Caso de Uso Segmentar.</i> .....                                          | 75 |
| <b>Figura 4.13</b> <i>Diagrama de Secuencia del Diseño del Caso Uso Gestionar Modelo 3D. Sección Generación del Modelo 3D.</i> ..... | 76 |
| <b>Figura 4.14</b> <i>Diagrama de Secuencia del Diseño del CU Gestionar Modelo 3D. Sección Importar Modelo.</i> .....                | 77 |
| <b>Figura 4.15</b> <i>Diagrama de Secuencia del CU Gestionar Modelo 3D. Sección Exportar Modelo.</i> .....                           | 78 |
| <b>Figura 5.1</b> <i>Paquetes del diagrama de componentes.</i> .....                                                                 | 79 |
| <b>Figura 5.2</b> <i>Diagrama de componentes. Paquete DICOM.</i> .....                                                               | 80 |
| <b>Figura 5.3</b> <i>Diagrama de Componentes. Paquete Segmentación.</i> .....                                                        | 81 |
| <b>Figura 5.4</b> <i>Diagrama de Componentes. Paquete Reconstrucción.</i> .....                                                      | 82 |



## Índice de Tablas.

|                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Tabla 2.1</b> <i>Tipos de datos en un archivo *.KPS.</i> .....                                                              | 44 |
| <b>Tabla 3.1</b> <i>Actor del Sistema.</i> .....                                                                               | 56 |
| <b>Tabla 3.2</b> <i>Descripción CU Cargar Ficheros.</i> .....                                                                  | 58 |
| <b>Tabla 3.3</b> <i>Descripción CU Segmentar.</i> .....                                                                        | 59 |
| <b>Tabla 3.4</b> <i>Descripción CU Gestionar Modelo 3D.</i> .....                                                              | 60 |
| <b>Tabla 3.5</b> <i>Descripción de la sección Generar Modelo del CU Gestionar Modelo 3D.</i> .....                             | 61 |
| <b>Tabla 3.6</b> <i>Descripción de la sección Exportar Modelo del CU Gestionar Modelo 3D.</i> .....                            | 62 |
| <b>Tabla 3.7</b> <i>Descripción de la sección Importar Modelo del CU Gestionar Modelo 3D.</i> .....                            | 63 |
| <b>Tabla 5.1</b> <i>Datos de Entrada.</i> .....                                                                                | 83 |
| <b>Tabla 5.2</b> <i>Resultados de los tiempos en segundos y consumo de recursos de hardware en los datos de entrada.</i> ..... | 84 |

## Introducción.

En la actualidad la información en forma de imágenes ha adquirido una gran importancia en el campo de la medicina. En los últimos años han surgido una gran variedad de equipos y aparatos médicos para obtener imágenes de algún órgano en específico del paciente, gracias a que la tecnología ha avanzado gradualmente, el médico especialista dispone de imágenes médicas de diversas modalidades, tales como: Tomografías Computarizadas (CT), Resonancias Magnéticas (MRI), Encefalogramas, Ultrasonidos, entre otros. Todos estos equipos exportan sus imágenes en ficheros DICOM, el estándar internacional para imágenes médicas digitales.

Las herramientas informáticas y los métodos de tratamiento, análisis y visualización de imágenes digitales han resultado de gran utilidad para el desarrollo de aplicaciones médicas de diversos tipos. Estas aplicaciones son de gran utilidad en la medicina para facilitar y permitir un aprovechamiento adecuado de la cuantiosa información disponible sobre las imágenes de los pacientes.

Actualmente las aplicaciones de tratamiento y análisis de imágenes médicas aparecen frecuentemente integrados con aplicaciones de visualización tridimensional (3D), y con un fuerte apoyo de técnicas de reconstrucción 3D. A los usuarios de estas aplicaciones como radiólogos, cirujanos, etc., les resulta de mucha utilidad en planeaciones de cirugía, ortopedia, traumas músculo-esqueléticos, visualización de órganos, o de partes del cerebro para su estudio, entre otros. Los equipos de radiología utilizados para el estudio de estas imágenes traen integrados varios software, algunos de estos son limitados en cuanto al análisis de las imágenes, y su visualización. En el mercado existe una gran variedad de este tipo de sistemas, algunos resultan un tanto ineficientes y otros son excelentes pero muy costosos. De cualquier forma, los usuarios de estas aplicaciones demandan de forma creciente aplicaciones que integren técnicas de tratamiento y análisis de imágenes con funcionalidades de visualización tridimensional, que sean eficientes, confiables y robustas.

La Realidad Virtual (RV), como una de las expresiones más destacadas de la digitalización, permite abordar estos temas desde un punto de vista más científico, y orientado a lograr una pseudorrealidad alternativa, perceptivamente hablando. Puesto que se entiende como RV a la simulación por computadora, dinámica y 3D, con alto contenido gráfico, acústico y

táctil, orientada a la visualización de situaciones y variables complejas, durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a "mundos" (o entornos) que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial.

La Realidad Virtual en nuestros días ha logrado insertarse en las más diversas áreas del conocimiento humano en especial en el área médica, mediante la creación de Simuladores Quirúrgicos (SQ). Estos simuladores se caracterizan por sus grandes exigencias tanto visual como en el sentido del tacto, sentidos que son fundamentales en la cirugía; además de tener como uno de sus principios esenciales la modelación y simulación del comportamiento de los órganos y de las herramientas que intervienen en una determinada operación, donde el clínico observa y es capaz de sentir como si estuviera operando a un paciente de forma real. Sobre todo en los SQ utilizados para el entrenamiento de la cirugía de mínimo acceso como: la cirugía laparoscópica, la artroscopía, la radiología y la endoscopia.

En función de los avances aplicados en el desarrollo de Simuladores Quirúrgicos y de los objetivos que éstos persiguen, se pueden agrupar en tres generaciones tecnológicamente secuenciales:

1. **La primera generación:** formada por los Simuladores Quirúrgicos que únicamente consideran la naturaleza geométrica de la anatomía humana.
2. **La segunda generación:** está constituida por aquellos que, además, permiten la interacción física con las estructuras anatómicas.
3. **La tercera generación:** que además de las características ya mencionadas de la segunda concepción, tienen en cuenta la naturaleza funcional de los órganos.

Actualmente, la gran mayoría de las investigaciones y los desarrollos de simuladores de estas características se encuentran dentro del grupo de la segunda generación.

Los SQ ideales o de generaciones superiores, deben estar formados por varias fases adicionales, como por ejemplo: la utilización como punto de entrada al proceso de simulación, de modelos o estructuras anatómicas obtenidos de la reconstrucción 3D de imágenes médicas digitales (DICOM), en vista de reproducir con una gran exactitud la anatomía de un paciente determinado.

En nuestro país, poco a poco ha ido creciendo el desarrollo de Sistemas de Realidad Virtual junto a las investigaciones y los avances tecnológicos aplicados al campo de la medicina, muchas empresas, universidades, grupos de investigación y desarrollo están colaborando en pos de estos adelantos.

Ejemplo de esto es la Universidad de Ciencias Informáticas, creada al calor de la batalla de ideas para apoyar el desarrollo científico y tecnológico del país, donde la docencia está guiada por la producción mediante la creación de proyectos productivos que respondan a las esferas económicas de la nación. Así surge, en la Facultad # 5 de dicha universidad, un proyecto para el desarrollo de Simuladores Quirúrgicos debido a la necesidad que tiene nuestro país de sistemas de software que contribuyan a mejorar la formación de especialistas en la esfera quirúrgica, favoreciendo la reducción de errores durante el proceso de formación dentro de dicha especialidad, dando la posibilidad de repetir los procedimientos tantas veces sea necesario sin la utilización de cadáveres o pacientes reales en el entrenamiento, reduciendo de esta manera los costes de riesgo asociados al proceso de formación de los especialistas en cirugía; además de posibilitar la planificación y práctica sobre la anatomía del paciente específico previamente a su intervención quirúrgica real.

En dicho proyecto no existe una herramienta que a partir de una serie de imágenes DICOM, realice la reconstrucción y generación de modelos anatómicos 3D que serán utilizados en los diferentes módulos de entrenamiento del simulador quirúrgico, por lo que los modelos deben desarrollarse en una herramienta de diseño como 3D Studio Max. Esto retrasa la entrega del software y provoca la esquematización de la anatomía humana, que es un elemento único y característico de cada persona, disminuyendo el nivel de realismo de la simulación.

Dada la **situación problemática** anteriormente expuesta, se plantea como **problema**: ¿Cómo lograr que los modelos anatómicos 3D visualizados en el Simulador Quirúrgico representen la anatomía real de una persona? A raíz de esto, el trabajo toma como **objeto de investigación**: los modelos anatómicos 3D en los Simuladores Quirúrgicos. Y dentro de esta extensa rama se propone como **campo de acción**: la reconstrucción de modelos anatómicos 3D a partir de imágenes DICOM, en los Simuladores Quirúrgicos.

El **objetivo** de esta investigación es elaborar una herramienta para la reconstrucción y generación de modelos anatómicos 3D, a partir del procesamiento de imágenes médicas digitales.

Este trabajo defiende la siguiente idea: el empleo de técnicas y algoritmos de procesamiento de imágenes digitales y reconstrucción de modelos 3D a partir de imágenes DICOM, aumentará el nivel de realismo de los modelos anatómicos 3D empleados en el software de simulación quirúrgica para el entrenamiento médico.

Para dar cumplimiento a los objetivos planteados en este trabajo se necesita un grupo de **tareas investigativas** a las cuales se hará referencia:

1. Identificar y Caracterizar las principales etapas que conforman el proceso de reconstrucción 3D a partir de imágenes.
2. Analizar la estructura y especificación del formato de las imágenes DICOM.
3. Cargar y Visualizar las imágenes DICOM.
4. Identificar y caracterizar las principales técnicas y algoritmos de programación para la segmentación de imágenes médicas.
5. Identificar y caracterizar las principales técnicas y algoritmos de programación basados en la reconstrucción, a partir de imágenes DICOM, de estructuras anatómicas tridimensionales.
6. Seleccionar las técnicas y algoritmos de segmentación y de reconstrucción a aplicar en el proceso de elaboración de la herramienta.
7. Analizar la estructura y especificación de los principales formatos existentes que almacenan información de modelos 3D. (Ej.: \*.3ds, \*.ase).
8. Definir la estructura y especificación del formato de fichero en que se exportará la información de los modelos anatómicos 3D resultantes del proceso de reconstrucción.
9. Elaborar una herramienta que realice la reconstrucción y generación de las estructuras anatómicas a partir de una serie de imágenes digitales de un estudio médico de Tomografía Axial Computarizada, Resonancia Magnética, Angiografías y otras técnicas.

Además para todo el proceso de investigación y elaboración de este trabajo se tomará en cuenta la utilización de varios **métodos científicos de investigación** como:

- **Histórico – Lógico:** método teórico mediante el cual se constatará como ha sido la trayectoria histórica real, la evolución y desarrollo de los aspectos principales de la investigación como: el Formato DICOM, la Segmentación de Imágenes, la Reconstrucción 3D de modelos anatómicos.
- **Analítico – Sintético:** este método teórico será utilizado en la investigación para buscar la esencia de lo que existe en el mundo acerca de las principales técnicas y algoritmos de segmentación de Imágenes y Reconstrucción 3D de modelos anatómicos, los rasgos que los caracterizan y los distinguen.
- **Modelación:** método teórico que se utilizará para representar parte del conocimiento que vayamos acumulando, así como los diferentes diagramas que apoyarán el proceso de elaboración de la herramienta, como los diagramas de casos de uso, diagramas de clases etc.
- **Entrevista:** método empírico para obtener información acerca del Formato DICOM, la Segmentación de Imágenes y la Reconstrucción 3D de los modelos, que guíen el resto de la investigación.
- **Experimento:** método empírico mediante el cual, se realizarán experimentos y pruebas de los principales algoritmos y técnicas de Segmentación y Reconstrucción 3D, para examinar los resultados en busca de escoger los más adecuados.
- **Observación:** mediante este método empírico se observarán los resultados obtenidos en la caracterización e identificación de los principales algoritmos utilizados, para poder decidir luego cuál o cuáles serán más adecuados.

Para tener un conocimiento general de lo que se abordará en este trabajo, se muestra a continuación una síntesis de cada capítulo:

### **Capítulo 1:-** Fundamentación Teórica.

En este capítulo, se indicarán las bases teóricas fundamentales de las técnicas y algoritmos fundamentales de la segmentación, reconstrucción, visualización, formatos de ficheros 3D,

---

librerías de clases más utilizadas mundialmente y aspectos importantes sobre las tecnologías a utilizar.

### **Capítulo 2:-** Solución Propuesta.

En esta sección se propone una solución técnica del problema, teniendo en cuenta las técnicas y algoritmos escogidos, las adaptaciones y aportes realizados por los autores.

### **Capítulo 3:-** Descripción de la solución propuesta.

Se comienza a desarrollar la Ingeniería de Software, para darle solución al problema planteado. Se realiza un levantamiento de requisitos, modelo de dominio y modelo de casos de uso del sistema.

### **Capítulo 4:-** Análisis y Diseño.

Se muestran los diagramas de clases del análisis y el diseño agrupados por paquetes, los diagramas de secuencia, las descripciones de las clases de diseño.

### **Capítulo 5:-** Implementación y Resultados.

Se procede a la implementación del sistema basados en los resultados del análisis y el diseño elaborado, en este capítulo se exponen además, algunos resultados de las pruebas que se le realizaron al software.

### **Glosario**

Se elaboró un glosario de términos con el objetivo de facilitar la comprensión del lenguaje del que se ha hecho uso.

### **Anexo A:** Segmentación de Imágenes.

Se profundiza en las técnicas de segmentación clásicas para la detección de puntos, líneas y bordes; y se aborda en detalle otros aspectos de interés en las técnicas basadas en umbrales y orientadas a regiones.

### **Anexo B:** Reconstrucción.

Se expresan otras ideas relativas a los algoritmos clásicos de reconstrucción Marching Cube y Marching Tetrahedra.

**Anexo C:-** Formatos de Ficheros.

Se explican más detalladamente los elementos fundamentales para la comprensión del Formato de Fichero DICOM.

**Anexo D:-** Imágenes de la Aplicación.

Se muestran imágenes de la interfaz gráfica de la aplicación referentes a algunas etapas de los procesos de segmentación y reconstrucción.



## Capítulo 1. Fundamentación Teórica.

En este capítulo se abordará de una manera lo más transparente posible, los principales elementos teóricos que conforman las principales técnicas y algoritmos que se han utilizado a nivel mundial en lo referente a la segmentación de imágenes, reconstrucción 3d, visualización y formatos de ficheros. Además se comentarán algunas de las principales librerías especializadas en alguna de esas ramas del Procesamiento de Imágenes y la Realidad Virtual. Y por último se tratará el tema de las tecnologías a utilizar para el desarrollo de la aplicación y el por qué de su elección.

### 1.1.Segmentación de Imágenes.

La segmentación de imágenes es un procesado digital que consiste en reconocer de forma manual, automática o semiautomática los objetos de una imagen digital. Existen distintos niveles de automatización dependiendo del método que se utilice para segmentar. Este automatismo pone a prueba los algoritmos utilizados por los sistemas para tal fin, ya que los objetos obtenidos deben corresponderse lo más acertadamente posible con el contenido semántico de la imagen.

El primer paso en cualquier proceso de análisis de una imagen cualquiera es la segmentación, mediante la cual se divide la imagen en las partes u objetos que la forman. El nivel al que se realiza esta subdivisión depende de la aplicación en particular, es decir, la segmentación terminará cuando se hayan detectado todos los objetos de interés para la aplicación. En general, la segmentación automática es una de las tareas más complicadas dentro del procesado de imagen. La segmentación va a dar lugar en última instancia al éxito o fallo del proceso de análisis. 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 la etapa de segmentación.

Los algoritmos de segmentación de imagen generalmente se basan en dos propiedades básicas de los niveles de gris de la imagen: discontinuidad y similitud. Dentro de la primera categoría se intenta dividir la imagen basados en los cambios bruscos en el nivel de gris. Las áreas de interés en estas categorías son la detección de puntos, de líneas y de bordes

en la imagen. Las áreas dentro de la segunda categoría están basadas en las técnicas de umbrales, crecimiento de regiones, y técnicas de división y fusión [34].

### **1.1.1. Tipos de Segmentación de Imágenes.**

#### **Manual.**

El usuario realiza la segmentación él mismo con la ayuda de una herramienta informática. Consiste en seleccionar manualmente las fronteras de las regiones que se desea segmentar, bien mediante el marcado de puntos de las mismas o usando algunas herramientas de apoyo más complejas. Es un método muy preciso, pero es muy lento y se hace impracticable cuando necesitamos segmentar un número alto de imágenes.

#### **Automática.**

El ordenador realiza todo el proceso de forma automática.

#### **Semiautomática.**

El ordenador realiza el proceso, pero el usuario interviene en determinados momentos sobre el mismo para definir parámetros o corregir resultados.

### **1.1.2. Técnicas de Segmentación de Imágenes.**

#### **1.1.2.1. Técnicas de Detección de Discontinuidades.**

Existen diversas técnicas para detectar varios tipos de discontinuidades. El método más común de buscar discontinuidades es la correlación de la imagen con una máscara. En la **Figura 1.1** se puede ver un caso general de máscara de  $3 \times 3$ .

|       |       |       |
|-------|-------|-------|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

**Figura 1.1** *Máscara 3x3.*

En este procedimiento se realiza el producto de los elementos de la máscara por el valor de gris correspondiente a los píxeles de la imagen encerrados por la máscara.

La respuesta a la máscara de cualquier píxel de la imagen viene dado por:

$$R = \sum_{i=1}^9 w_i z_i$$

Donde  $z_i$  es el nivel de gris asociado al píxel de la imagen con coeficiente de la máscara  $w_i$ . Como suele ser habitual, la respuesta de la máscara  $R$ , viene referida a su posición central. Cuando la máscara esté centrada en un píxel de borde de la imagen, la respuesta se determina empleando el vecindario parcial apropiado.

### **Detección de Puntos.**

Esta técnica se basa en medir la diferencia entre el píxel central y sus vecinos, puesto que un píxel será un punto aislado siempre que sea suficientemente distinto de sus vecinos. Solamente se considerarán puntos aislados aquellos cuya diferencia con respecto a sus vecinos sea significativa [34 y 36]. Ver [Anexo A.1](#)

### **Detección de Líneas.**

La detección de líneas es básicamente el proceso en el cual se enlazan mediante líneas los puntos que cumplen con un patrón de orientación (horizontal, vertical y diagonal) dentro de la imagen [34 y 36]. Ver [Anexo A.2](#)

## **Detección de Bordes**

La detección de bordes es el procedimiento empleado más habitualmente para la detección de discontinuidades. Un borde se define como la frontera entre dos regiones con nivel de gris relativamente diferente [34 y 36]. Ver [Anexo A.3](#)

### **Limitaciones de las técnicas de segmentación basadas en discontinuidades.**

Las principales limitaciones de los algoritmos de detección de discontinuidades [3] son:

- a) Los bordes obtenidos mediante los métodos clásicos a menudo no necesariamente corresponden a objetos de la frontera. En muchas imágenes de baja calidad, capturadas utilizando dispositivos baja calidad, algunos de los métodos convencionales producen bordes erróneos sus aplicabilidades son por lo tanto limitado.
- b) El límite de detección de las técnicas dependen de la información que figura en la vecindad local de la imagen. La mayoría de las técnicas de detección de bordes no consideran el modelo basado en la información incrustada en una imagen.
- c) En la mayoría de los casos el límite de detección de las estrategias hace caso omiso de la información de orden superior que puede ser más significativa en la imagen.
- d) Después que los puntos de borde son extraídos de la imagen, estos puntos son relacionados con el fin de determinar las fronteras. Esto se suele hacer primeramente asociando elementos en segmentos de borde y, a continuación, asociando estos segmentos en las fronteras. El proceso de vinculación de borde en ocasiones puede dar lugar a falsas discontinuidades y lagunas en la imagen.
- e) A menudo es difícil de identificar y clasificar los bordes ilegítimos.

### **1.1.2.2. Técnicas de Umbrales. Umbralización.**

La imagen  $I_{(x, y)}$  está compuesta de objetos claros sobre fondo oscuro de tal forma que los niveles de gris están agrupados en dos modos predominantes. Una forma de separar los

objetos del fondo consiste en seleccionar un umbral  $T$  que separe esos modos. Entonces, cualquier punto  $(x, y)$  para el que se cumpla que

$$I(x, y) > T,$$

Se le etiqueta como objeto; en otro caso, como fondo [34 y 36]. Ver [Anexo A.4](#)

### 1.1.2.3. Técnicas de Orientadas a Regiones.

El objeto de los métodos de segmentación descritos en las secciones anteriores fue encontrar bordes entre regiones. Los métodos que examinaremos ahora construyen las regiones directamente.

Sea  $R$  la región correspondiente a la imagen a segmentar. Vamos a ver el proceso de segmentación como un proceso en el cual dividimos la región  $R$  en  $n$  subregiones  $R_1, R_2, \dots, R_n$ , tal que:

$$\bigcup_{i=1}^n R_i = R$$

$R_i$  es una región conectada,  $i = 1, 2, \dots, n$

$R_i \cap R_j = \emptyset$  para todo  $i$  y  $j$  con  $j \neq i$

$P(R_i) = \text{CIERTO}$  para  $i = 1, 2, \dots, n$

$P(R_i \cup R_j) = \text{FALSO}$  para todo  $i$  y  $j$  adyacentes con  $j \neq i$ ,

Donde  $P(\cdot)$  es un predicado lógico definido sobre la partición de  $R$  y  $\emptyset$  es el conjunto vacío.

### Crecimiento de Regiones.

Como el nombre indica, el crecimiento de regiones es un procedimiento mediante el cual se agrupan píxeles o subregiones en regiones mayores. El procedimiento más sencillo se denomina agregación de píxeles, que comienza a partir de un conjunto de píxeles semilla, de forma que a partir de cada semilla se crecen regiones añadiendo píxeles a dicha semilla

de entre aquellos píxeles vecinos que tienen propiedades similares [34 y 36]. Ver [Anexo A.5](#)

## 1.2.Reconstrucción Tridimensional.

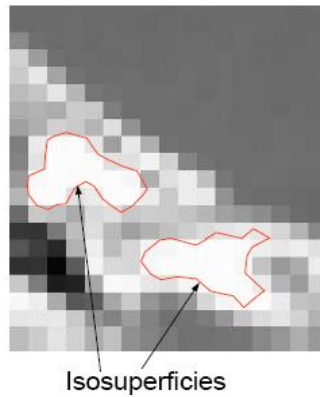
La aplicación de técnicas de reconstrucción 3D, provee una gran cantidad de ventajas, como puede ser la utilización en el diagnóstico, en el tratamiento de diferentes enfermedades y el entrenamiento de profesionales. Aplicando técnicas de segmentación y reconstrucción sobre los datos de los estudios de los pacientes, pueden generarse los volúmenes que representan los diferentes tejidos y regiones dentro del cuerpo del mismo. Normalmente el médico utiliza un conjunto de imágenes 2D para llevar a cabo el diagnóstico. El médico debe reconstruir mentalmente el modelo tridimensional representado por una serie de imágenes en dos dimensiones. El objetivo de la segmentación es dividir (segmentar) y separar la información original para generar nuevas imágenes que representen las estructuras internas del paciente. Esta técnica, combinada con la reconstrucción de superficies, facilita la labor del médico al presentar esta información de una manera que le sea fácil de interpretar y visualizar, permitiéndole realizar un diagnóstico más preciso del paciente.

Actualmente estas técnicas de reconstrucción están muy difundidas en los simuladores quirúrgicos de última generación; que con vistas a ganar un mayor nivel de realismo en los modelos que sirven de base de entrenamiento, utilizan imágenes DICOM para obtener los datos. Aquí es donde la reconstrucción juega su papel, pues de ella depende que los modelos anatómicos reproduzcan con la mayor certeza posible la anatomía humana.

### 1.2.1. Algoritmos basados en Marching.

Los algoritmos de Marching son métodos simples y bastante populares para extraer isosuperficies de funciones implícitas o información tridimensional discreta, como muestra la **Figura 1.2**. Estos se basan en subdividir una superficie o volumen en formas más elementales y detectar uno o más contornos, ubicando el color, valor, o intensidad en los vértices de la figura elegida. De esta manera, por medio de interpolación es posible detectar

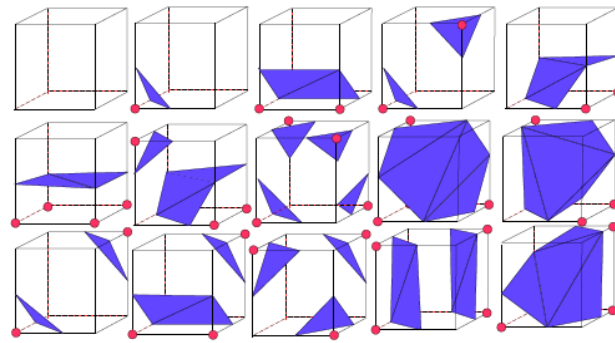
cuáles aristas de la figura son intersecadas por la superficie y reconstruir (a partir de los puntos de intersección), la superficie con figuras más elementales, como por ejemplo los triángulos.



**Figura 1.2** Vista 2D de una isosuperficie.

### 1.2.1.1. Marching Cubes.

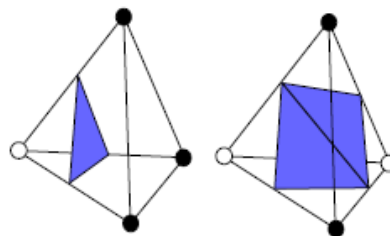
Marching Cubes (MC) es un algoritmo para extraer Isosuperficies a partir de datos volumétricos. La idea fundamental es que se puede definir un cubo (Vóxel) que contenga en sus 8 vértices la información de la intensidad que corresponde a ese punto. Si uno o más vértices tienen valores mayores que el de la superficie buscada, y uno o más tienen valores menores, entonces se sabe que el cubo es cortado por la isosuperficie que se está buscando. Al determinar los lugares del cubo por donde pasa la superficie se pueden generar triángulos que unan los puntos de intersección. Al unir todos los triángulos generados se obtiene la superficie buscada [14 y 31]. Ver [Anexo B.1.1](#)



**Figura 1.3** *Casos de Triangulación Marching Cubes.*

### 1.2.1.2. Marching Tetrahedra.

El concepto de Marching Tetrahedra (MT) es el mismo que se aplica en Marching Cubes. La principal diferencia es que el MT utiliza como figura elemental de corte un tetraedro y no un cubo. Esa consiste en dividir los cubos en una serie de tetraedros. Un tetraedro solo tiene 16 triangulaciones posibles, las que se reducen a 3 por simetría [14]. Ver **Figura 1.4**. Ver [Anexo B.1.2](#)



**Figura 1.4** *Casos de Triangulación Marching Tetrahedra.*

Estos algoritmos basados en Marching tienen por lo general el mismo problema, que en todo momento dividen la superficie en paralelepípedos (figuras elementales de corte: cubos, tetraedros, etc.) de igual tamaño, sea cual sea la región que se esté estudiando. Esto provoca que los algoritmos no obtengan un conjunto de triángulos que determinen de manera óptima la forma y la malla obtenida no sea de calidad. Afortunadamente estas deficiencias fueron detectadas desde las versiones iniciales de los algoritmos, y muchos investigadores y estudiosos del tema han trabajado en mejorarlos, pero desgraciadamente



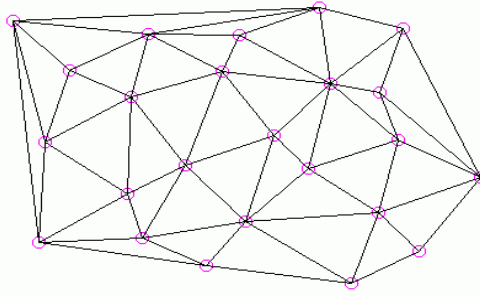
los algoritmos obtenidos no están accesibles para todo el que los necesite, sino que muchos son privativos y otros simplemente no han sido liberados para el uso de la comunidad internacional.

### **1.2.2. Algoritmos basados en Triangulación de Delaunay.**

La Triangulación de Delaunay es una de las triangulaciones más interesantes por ser aplicable para la resolución de multitud de problemas aparentemente sin relación entre sí, debido a sus propiedades geométricas, y por contar con algoritmos bastante eficientes para su cálculo.

En una triangulación de Delaunay:

- Todos los puntos están conectados entre sí y forman el mayor número de triángulos posibles sin que se crucen sus aristas (imprescindible para que se trate de una triangulación).
- Los triángulos se definen de forma que los puntos más próximos están conectados entre sí por una arista.
- Esto implica que los triángulos formados sean lo más regulares posible, es decir, que se maximicen sus ángulos menores y se minimice la longitud de sus lados.
- La triangulación forma la envolvente convexa del conjunto de puntos.
- La triangulación es unívoca si en ningún borde de circunferencia circunscrita hay más que tres vértices en el caso 2D y sino existen más de cinco vértices en el borde de la esfera circunscrita en 3D.



**Figura 1.5** *Triangulación de Delaunay.*

Debido a las propiedades de la Triangulación de Delaunay se derivan algunas de sus desventajas al utilizarse en la Reconstrucción 3D, por ejemplo:

- Cuando la nube de puntos tiene naturaleza no convexa, o sea que posea huecos, la triangulación de Delaunay los ignora, porque una de sus propiedades es proporcionar la envoltura convexa del conjunto de puntos.
- Es posible obtener más de una triangulación, debido a que una esfera vacía (sin vértices en el interior) esté circunscrita en más de 4 puntos (en el caso 3D). Por tanto es altamente sensible a perturbaciones geométricas de las posiciones de los nodos.

En vista a las desventajas anteriores se han obtenido numerosos algoritmos de Triangulación basados en Delaunay que las eliminan, por ejemplo la Teselación Delaunay Extendida (EDT) y la Teselación Delaunay Restringida (CDT), entre otros.

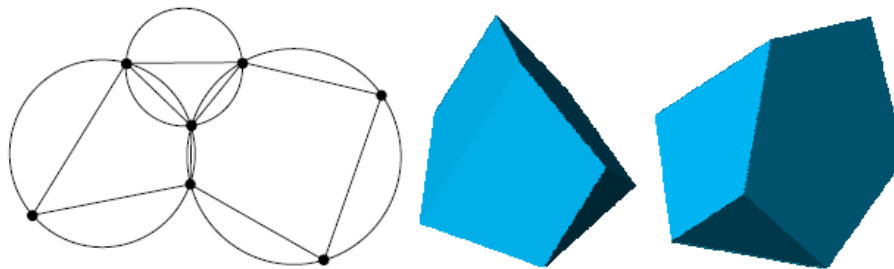
### **1.2.2.1. Teselación de Delaunay Extendida.**

La Teselación Delaunay Extendida o EDT es una partición de la envoltura convexa (*convex-hull*) de un conjunto de nodos en poliedros de vértices aproximadamente coesféricos. Los poliedros se pueden llamar poliedros casi-esféricos o más sencillamente poliedros esféricos.

La EDT se genera mediante el siguiente procedimiento:

- 1) Triangulación Delaunay.
- 2) Unión de poliedros con esferas cercanas.
- 3) Captura de poliedros contenidos.

La Teselación Extendida de Delaunay, admite que todos los nodos equidistantes del centro formen un polígono esférico, reteniendo la continuidad del diagrama de Voronoi y proveyendo además de interpolantes más naturales, en el sentido de que todos los nodos del poliedro serán tenidos en cuenta para la interpolación en un punto interior del poliedro.



**Figura 1.6** Polígonos circulares en 2D y dos vistas de un poliedro esférico en 3D.

Las ventajas de “engordar” los poliedros (rellenando más la esfera) han resultado tan apreciables que hemos ido un paso más adelante admitiendo la formación de poliedros con nodos que no son exactamente equidistantes de un vértice del diagrama sino sólo en forma aproximada.

Con referencia en la **Figura 1.6**, dado que las esferas son huecas, debe cumplirse la siguiente relación entre sus radios y sus centros:

$$|r_2 - r_1| \leq \|c_1 - c_2\|$$

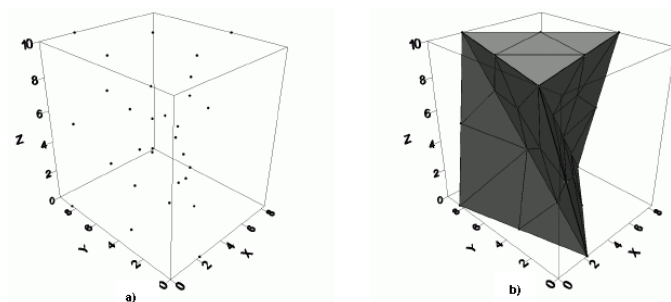
Para saber si dos esferas son parecidas alcanza con medir la distancia entre sus centros, decimos que dos esferas son similares si se cumple que:

$$\|c_1 - c_2\| < \delta r$$

Donde  $\delta$  es un pequeño valor y  $r$  es el radio medio cuadrático. Si dos (o más) esferas son similares, entonces formamos un poliedro con sus vértices. El valor de  $\delta$  es arbitrario. Dado que pretendemos unir esferas similares, debe ser un valor pequeño. Si es nulo sólo se formarán poliedros con nodos exactamente equidistantes de un vértice del diagrama de Voronoi. A medida que  $\delta$  aumenta se formarán más poliedros supernumerarios [6 y 7].

### 1.2.2.2. Triangulación de Delaunay Restringida.

La triangulación restringida de Delaunay o Constrained Delaunay Tessellation (CDT), proporciona un método para forzar la aparición de las aristas de un PSLG (Planar Straight-Line Graph),  $G$ , en la triangulación de Delaunay, como muestra la **Figura 1.7**. Un triángulo  $abc$  aparece en la triangulación restringida de Delaunay si su circunferencia circunscrita no contiene ni pasa por ningún vértice de  $G$  visible desde cualquiera de los puntos de  $abc$ . Esta definición generaliza la triangulación de Delaunay en el caso de que  $G$  no contenga aristas. Si  $G$  es un polígono, entonces la triangulación restringida de Delaunay contiene sólo triángulos interiores a  $G$  [52 y 53].



**Figura 1.7** Triangulación de Delaunay Restringida, Nube de puntos (a) y reconstrucción del PLC (b).

## 1.3. Visualización.

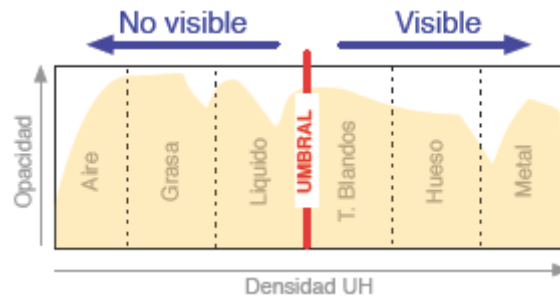
Dentro del área de Visualización Científica (Scientific Visualization) existen dos grandes ramas de representación de objetos tridimensionales:

- Surface Rendering (Basados en Superficies).
- Volume Rendering (Basados en Volumen).

Las primeras técnicas desarrolladas para la visualización 3D se basaban en el concepto de "umbrales" que es en esencia la manera más sencilla de clasificar tejidos con un solo criterio. Se define un umbral de **Unidades Hounsfield** y se eliminan de la visualización todos los tejidos por debajo de ese umbral.

### 1.3.1.Unidad de Hounsfield.

Es una condición "todo o nada" (se ve o no se ve), y lo que se ve se clasifica como el mismo tipo de tejido, es relativamente sencillo generar estas superficies donde se encontraban vóxeles por encima de ese umbral. Esto permitió que las computadoras más antiguas tuvieran la capacidad de procesamiento para realizar estas visualizaciones, y por ende fue la primera técnica de representación tridimensional utilizada.



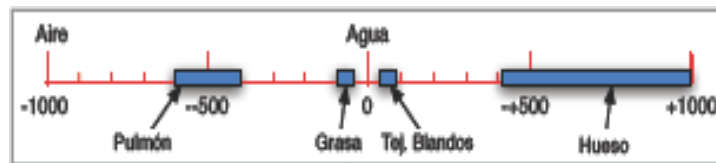
**Figura 1.8** Umbrales. Todos los tejidos por encima del umbral se mostrarán, los que estén por debajo serán eliminados.

Los equipos de Tomografía Axial Computarizada (TAC) solo aportan un criterio para determinar de qué está formado un determinado tejido; el coeficiente de atenuación, medido en Unidades de Hounsfield (UH). Escala que toma como referencia el agua, y está dada por la aplicación de la siguiente fórmula:

$$UH = (\mu_{\text{objeto}} - \mu_{\text{agua}}) / \mu_{\text{agua}} \times 1000$$

Esta representa un conjunto de valores en escala de grises que oscilan en el intervalo de -2000 y +2000. Hasta ahora la UH constituye el único criterio que permite identificar los distintos tipos de tejidos y otras sustancias.

Generalmente se pueden clasificar los diferentes tejidos según su coeficiente de atenuación en 6 grandes grupos: aire (< -1000), pulmón (-400 a -600), grasa (-60 a -100), agua (0), tejidos blandos (+40 a +80), hueso (> +400). Ver **Figura 1.9**



**Figura 1.9** Coeficientes de atenuación medidos en Unidades de Hounsfield.

Estas son las denominadas ventanas de reconstrucción, que no son más que los rangos de visualización en unidades de Hounsfield. Todas las visualizaciones tridimensionales se basan en estas diferencias de atenuación, usando diversas técnicas.

### 1.3.2.Surface Rendering.

Los métodos de Surface Rendering describen el objeto mediante aproximaciones matemáticas a las superficies que lo componen. En estos métodos basados en superficies no se distinguen los cambios de intensidad (o densidad) dentro del mismo objeto (ya que se basan en la escena binaria, y esta no preserva el rango de intensidades provista por el equipo TAC).

Existen distintas maneras de representación de superficies en el espacio, pero todas se basan en la "figura" (o silueta) del objeto. Las ventajas de estos métodos radican principalmente en la disminución del volumen de datos necesario para describir el objeto, la manejabilidad y maleabilidad de estas estructuras matemáticas y la posibilidad de implementación en equipos de menor porte que los necesarios para los métodos basados en volume rendering.

### 1.3.3.Volume Rendering.

La visualización por Volumen Rendering se basa en la escena binaria (o inclusive, en los datos originales sin hacer la segmentación, o haciéndola en el momento del renderizado) y es una idea conceptualmente muy atractiva. No se intenta describir al objeto de ninguna manera, simplemente se divide en elementos cúbicos muy pequeños llamados vóxeles,

pudiéndose direccionar, clasificar, esconder y modificar cada uno de estos. Para lograr la graficación tridimensional de algún objeto mediante este método simplemente se grafican, en el orden adecuado, todos los vóxeles que lo componen.

El Volume Rendering es una disciplina con mucho futuro, ya que las capacidades computacionales aumentan a un ritmo impresionante. Como herramienta para la medición, se la considera más exacta que los métodos más "inteligentes", ya que describe todos y cada uno de los elementos que componen a un objeto.

En representación en volumen, se persigue la representación de múltiples estructuras con diferentes tipos de tejido. A cada vóxel se le asocia un nivel de opacidad y se asume que el valor de cada uno está correlacionado con el tipo de material que lo integra. Para cada versión de despliegue se examina cada vóxel, lo que hace a esta representación muy costosa computacionalmente. [\[29\]](#)

## **1.4.Formatos de Ficheros.**

### **1.4.1.Ficheros de Gráficos 3D.**

Los ficheros contenedores de la información relativa a las escenas virtuales son aquellos que por lo general almacenan los datos correspondientes a puntos, vértices, polígonos, vectores normales, color, textura, materiales, luces y otros que garantizan el realismo de los objetos y la escena virtual. Dentro de los formatos de ficheros más utilizados están los \*.3DS y los \*.ASE.

#### **1.4.1.1.Características de los 3DS.**

Es un fichero de composición binaria cuya especificación está patentada bajo una licencia propietaria; por lo que se conoce poco sobre la información almacenada dentro del mismo. Su estructura está caracterizada por bloques de información indexados, conocidos como CHUNCK; lo que facilita el acceso a los bloques de interés de forma rápida sin necesidad de leer e interpretar los restantes bloques, característica que lo ha hecho muy popular en la comunidad de desarrolladores dentro del campo de la Realidad Virtual, un ejemplo de esto

es el hecho de que es el formato principal de exportación de grandes software de desarrollo gráficos 3D como 3D Studio Max y Maya.

#### **1.4.1.2. Características de los ASE.**

El formato de fichero ASE (ASCII Scene Exporter), es originario de 3D Studio Max, pero otros grandes software de diseño 3D, tales como ActorX y Maya también pueden exportarlo. Es un fichero de texto que contiene información correspondiente a geometría, texturas, materiales, cámaras y luces. Se puede editar con cualquiera de los editores de textos conocidos., por lo que es fácil de leer e interpretar, pero esto hace que no se utilice en software con fines comerciales puesto que la información que contiene se puede modificar sin garantizar la validez de la misma. Otra desventaja del mismo es que la información está almacenada de forma secuencial por lo que no es posible acceder a un dato x sin antes haber leído el dato x-1, provocando que el proceso de carga del mismo sea lento y costoso en memoria. El uso de este fichero está generalmente favorecido por la estructura en que almacena la información de los materiales de los objetos y de la escena virtual misma.

#### **1.4.2. Especificación del Estándar DICOM.**

Las necesidades de almacenamiento y manipulación de imágenes médicas surge a partir de los años 70's como consecuencia del nacimiento de la tomografía computarizada como medio de diagnóstico basado en imágenes digitales. Desde entonces, se han desarrollado diferentes técnicas y equipos médicos para la obtención y manipulación de imágenes digitales como: la medicina nuclear, la resonancia magnética, la radiografía computarizada y la angiografía por sustracción digital, entre otras.

Junto con el desarrollo de las imágenes médicas digitales nacen en el mercado grandes cantidades de compañías productoras de equipos médicos especializados en la obtención y visualización de imágenes médicas digitales a partir de estudios computarizados realizados a pacientes mediante las técnicas de Tomografía Axial Computarizada (TAC), Resonancia Magnética, etc., debido a éste fenómeno la comunidad productora de equipos médicos se



vió obligada a establecer un lenguaje de comunicación común entre los productos de hardware y software desarrollados por diferentes compañías.

En 1992 quedó establecido el Estándar para el Tratamiento Digital de Imágenes y Comunicación en Medicina (DICOM). DICOM está basado en el estándar ACR-NEMA y fue lanzado por el Colegio Americano de Radiografía (ACR) y por la Asociación Nacional de Fabricantes Eléctricos (NEMA) para satisfacer las necesidades de conectividad entre equipos de tratamientos de imágenes [1 y 45].

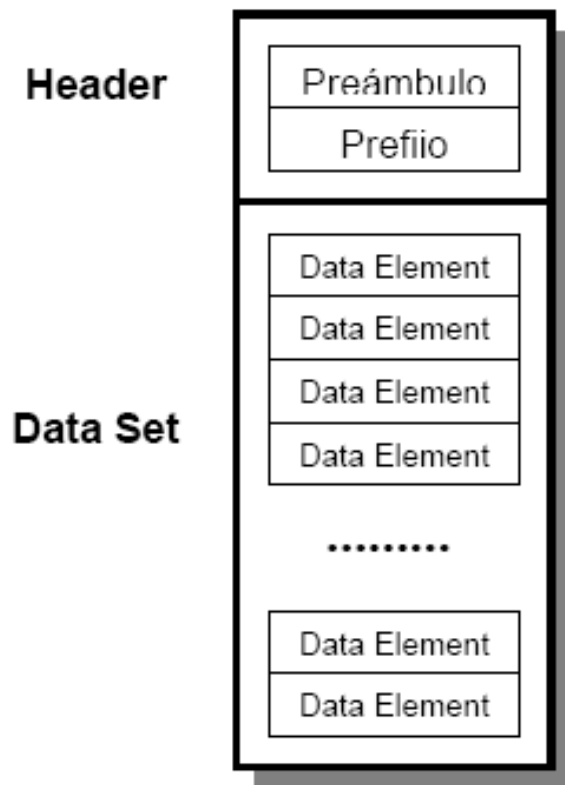
#### **1.4.2.1. Características de los DICOM.**

El estándar describe el formato de archivos y la especificación de los datos primordiales de un paciente en la imagen así como el encabezado requeridos, describiendo un lenguaje común a distintos sistemas médicos. De esta forma las imágenes vienen acompañadas de mediciones, cálculos e información descriptiva relevante para diagnósticos. Utiliza archivos con extensión \*.dcm [1 y 45].

#### **1.4.2.2. Formato de un Archivo DICOM.**

Cada archivo DICOM contiene una cabecera que almacena la información sobre el nombre del paciente, el tipo de exploración, la dimensión de la imagen, etc.; así como todos los datos de la imagen que pueden contener la información en tres dimensiones.

El formato genérico del archivo de DICOM consiste en dos partes: **Header** seguido inmediatamente por un **DataSet** de DICOM. El **DataSet** de DICOM contiene la imagen o las imágenes especificadas. El **Header** contiene sintaxis de transferencia **UID (identificador único)** que especifica la codificación y la compresión del Data Set [1 y 45]. Ver [Anexo C.1](#)



**Figura 1.10** Estructura de un archivo DICOM (\*.dcm).

## 1.5.Principales Bibliotecas de Clases.

A continuación se relacionan una serie de bibliotecas de clases mundialmente conocidas, y que representan gran parte de todo el desarrollo alcanzado por la comunidad científica internacional a lo largo de estos años. Entre las librerías más utilizadas en el tratamiento de ficheros DICOM se encuentran: Imebra, DicomLib. Entre las más importantes en la visualización y la segmentación se encuentran VTK e ITK, y entre las más utilizadas para la triangulación está TETGEN.

### 1.5.1.Imebra.

Imebra es una librería Open Source para el trabajo con ficheros DICOM, NEMA y JPEG, además de proporcionar un código de alto nivel y un rápido acceso a los datos cargados.

Las imágenes pueden ser descomprimidas o convertidas a otros formatos de color. Esta biblioteca posee licencia GPL y una licencia comercial. Actualmente se encuentra en la versión 0.0.36.

Desafortunadamente en su versión libre, la librería, no posee una adecuada documentación que explique como explotar de la mejor manera, todas las funcionalidades con que cuenta la misma. [\[56\]](#)

### **1.5.2.DicomLib.**

Es una librería para el trabajo con ficheros DICOM desarrollada bajo la licencia GNU GPL. Está inspirada en la librería udcmc99, pero fue casi completamente reprogramada, con el objetivo de proveer una API clara, muy vinculada al estándar DICOM y fuertemente tipeada. Un conjunto de ideas recopiladas para mapear de la mejor manera los conceptos del DICOM, tales como las "Representaciones de Valor" (VR), a conceptos de C++ tales como los "tipos". El resultado es una librería que hace posible integrar funcionalidad DICOM de una manera sencilla en una aplicación. Actualmente la librería está en su versión 0.9.0.

Algunas de sus características claves:

1. Mecanismos de plantillas (templates) para reforzar las relaciones entre los VRs del DICOM y los tipos de C++.
2. Mecanismos de multihilos utilizando la librería Boost de C++.
3. Un API ampliamente optimizada.

Esta librería ha sido desarrollada independientemente de las plataformas existentes, ha sido probada en Windows 2000, XP, Linux (Intel) y Solaris (Sparc). Esta librería tiene un nivel de dependencia muy alto con otras librerías de C++, como por ejemplo Boost. [\[57\]](#)

### 1.5.3.VTK.

VTK (Visualization Toolkit), es una librería de clases para el trabajo con sistemas de gráficos 3D, procesamiento de imágenes y visualización, libremente disponible y Open Source.

Esta librería es usada por cientos de investigadores y desarrolladores de todo el mundo, programada en C++, y posee capas de interfaces para la comunicación con lenguajes como Tcl/Tk, Java, Python y por supuesto C++. VTK soporta una amplia variedad de algoritmos de visualización incluyendo métodos para el trabajo con escalares, vectores, tensores, texturas y volúmenes; y técnicas avanzadas de modelación tales como modelación implícita, reducción de polígonos, suavizado de mallas, algoritmos de triangulación como la Triangulación de Delaunay, entre otros. Además tiene una arquitectura basada en el diseño Orientado a Objetos y su uso ha sido probado en los diferentes sistemas de Hardware y Sistemas Operativos. [\[58\]](#)

### 1.5.4.ITK.

ITK (Insight Segmentation and Registration Toolkit), es una librería de clases Open Source que soporta el Proyecto Humano Visible (Visible Human Project). Fue desarrollada por seis organizaciones fundamentalmente, tres comerciales (Kitware, GE Corporate R&D, y Insightful) y tres académicas (UNC Chapel Hill, Universidad de Utah, y Universidad de Pennsylvania). Equipos adicionales de trabajo incluyen el Hospital Harvard Brigham & Women's, Universidad de Pittsburg, y la Universidad de Columbia.

ITK está implementada en C++. Es multiplataforma, y puede ser utilizada desde múltiples lenguajes de programación como Tcl/Tk, Java, Python y por supuesto C++. Es altamente eficiente y está en constante mantenimiento por la comunidad internacional.

Esta librería es una de las más utilizadas a nivel mundial para investigación analítica de imágenes en cuanto a segmentación, clasificación y técnicas de registro para procesar grandes volúmenes de datos médicos. [\[59\]](#)

### 1.5.5.TetGen.

TetGen una librería Open Source, multiplataforma, implementada en C++ y actualmente soportada por el Instituto Weierstrass de Estocástica y Análisis Aplicado (WIAS) en el Grupo de Investigación de Matemática Numérica y Ciencias de la Computación.

Esta librería tiene como objetivo la generación de mallas y Triangulaciones de Delaunay en un dominio 3D. Para un dominio tridimensional, definido por su frontera (tal como una malla de superficie), TetGen genera la Tetraedralización Restringida de Delaunay (CDT), refinamiento de calidad de la malla. Para un conjunto de puntos 3D, se genera la Triangulación de Delaunay y la envoltura convexa (convex hull). [\[60\]](#)

## 1.6.Metodologías y Herramientas de Desarrollo.

### 1.6.1.Metodología de Ingeniería de Software.

RUP (Proceso Unificado de Desarrollo) se usó en esta tesis para realizar el análisis y diseño de la aplicación, porque es una metodología que acumula muchos años de experiencia en el desarrollo de software, además de estar rigurosamente probada a nivel mundial con innumerables sistemas de software.

A continuación se muestran las características que más influyeron en la selección de esta metodología:

- **Guiado por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo del sistema.
- **Centrado en arquitectura:** La arquitectura muestra la visión común del sistema completo.
- **Iterativo e Incremental:** RUP divide el proyecto en fases de desarrollo, propone además que cada una de ellas se desarrolle en iteraciones, las cuales aportan un incremento en el proceso de desarrollo y terminan con el cumplimiento del punto de control trazado en la fase.

## **1.6.2.Herramientas de Desarrollo.**

Rational Rose Enterprise Edition se usó como herramienta de modelado porque además de brindar una interfaz cómoda para trabajar, es la mejor herramienta de apoyo a la Ingeniería de Software que puede usarse como soporte al Proceso Unificado de Racional, además de ofrecer un soporte total y sencillo al proceso de desarrollo.

Microsoft Visual Studio.NET 2003, como entorno integrado de desarrollo (**IDE**), ya que el mismo cuenta con pequeñas herramientas auxiliares de completamiento de códigos como el Visual Assist lo que ayuda mucho a la codificación de la aplicación, y otras como el VTune que le permite al desarrollador ir comprobando el rendimiento en cuanto a memoria y uso del CPU de la aplicación que se está codificando.

Borland C++ Builder 6 para el diseño y programación de la GUI de la aplicación, para lograr una portabilidad y facilidad de diseño, ya que las aplicaciones con GUI desarrolladas en .NET necesitan obligatoriamente de la instalación del Framework de .NET para poder ser ejecutadas en una PC.

## **1.6.3.Lenguajes.**

### **1.6.3.1.Lenguaje de Programación.**

El lenguaje de programación escogido fue el C++, porque además de ser un lenguaje multiplataforma, es el lenguaje por excelencia para las aplicaciones de Realidad Virtual puesto que en este tipo de aplicaciones se manejan grandes volúmenes de datos y el mismo permite un uso óptimo de la memoria y del CPU.

### **1.6.3.2.Lenguaje de Modelado.**

Para modelar el análisis y el diseño del software se escogió el lenguaje UML (Unified Modeling Language), porque es un lenguaje independiente de la plataforma y además

constituye un estándar mundial en el modelado. Modelar usando este lenguaje es establecer una comunicación de alto nivel entre todo el que interactúe con los modelos, pero además:

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, sin ambigüedades y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).

UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

## Capítulo 2. Solución Propuesta.

Después del estudio teórico realizado en el capítulo anterior, en este capítulo se describen las técnicas y algoritmos básicos seleccionados que conformarán los procesos de segmentación, reconstrucción tridimensional, post-procesamiento de superficies, visualización y exportación de los modelos; así como las modificaciones que fueron realizadas a los mismos, con el objetivo de adaptarlos a las particularidades de nuestra solución y se fundamentan las razones por las cuales no se hace uso de ninguna librería de clases de las existentes a nivel mundial.

### 2.1.¿Por qué no utilizar las librerías de clases existentes?

Es importante resaltar que a pesar de existir numerosas librerías de clases, como las abordadas anteriormente en el [\[Epígrafe 1.5\]](#) y que han sido muy utilizadas en aplicaciones de reconstrucción 3D avanzadas, en este trabajo no se hace uso de ninguna de ellas, por varias cuestiones como las siguientes:

- Las librerías Imebra y DicomLib, no cuentan con una documentación que haga más comprensible el trabajo con ellas, además de que fueron creadas para un trabajo a fondo con los ficheros DICOM, y para la realización de este trabajo solo se requiere la carga de las imágenes.
- La biblioteca de clases TetGen es una de las más potentes en lo que a Triangulación de Delaunay y a la Triangulación de Delaunay Restringida (CDT) se refiere. Sin embargo, para obtener buenos resultados, esta librería toma como dato de entrada un archivo \*.poly que contiene el PLC<sup>1</sup> de lo que se desea triangular, el principal problema de esto es que no existe ningún algoritmo que sea capaz de calcular el PLC de una nube de puntos arbitraria o en caso de existir sería demasiado costoso.

---

<sup>1</sup> PLC ( Piecewise Linear Complex): *Es el conjunto de vértices, segmentos y caras, donde cada cara es una región poligonal, que puede tener cualquier cantidad de lados y puede ser no convexa, posiblemente con huecos, segmentos y vértices dentro de ella. Es lo que se almacena en el fichero \*.poly de TetGen.*



- Las librerías VTK e ITK son Open Source, pero la documentación existente de ambas hay que comprarla. Lo que se puede adquirir libremente no es suficiente para emprender una investigación como la que aquí se desarrolla, además algunos algoritmos también están patentados y hay que pagar una licencia para poder utilizarlos.
- La necesidad de contar con las bases algorítmicas que permitan el desarrollo de investigaciones de este tipo en la universidad, que luego puedan ser generalizadas y liberadas a una comunidad de desarrollo.

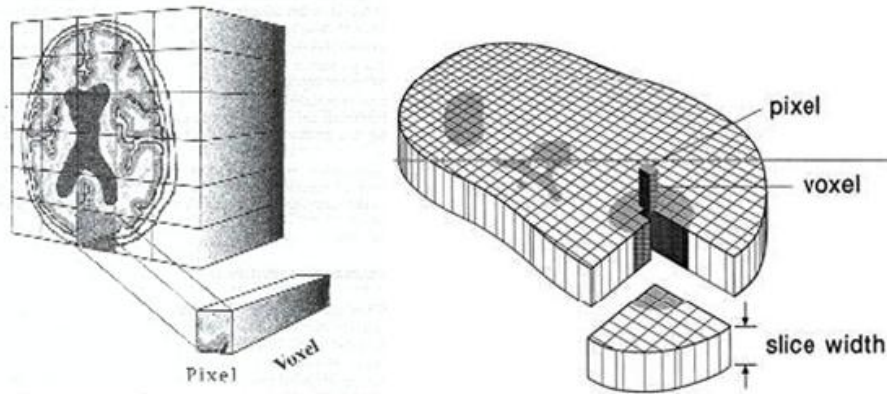
Luego de aportar los elementos que justifican el por qué de la decisión de no utilizar ninguna de las bibliotecas de clases anteriores, se procede a explicar la solución propuesta.

## 2.2.Construcción de las Imágenes Tridimensionales.

Cada imagen DICOM obtenida a partir de un estudio realizado a algún paciente, al ser mostrada en la pantalla del ordenador es bidimensional, sin embargo corresponde en la realidad a un volumen. El soporte donde se crea la imagen es una **MATRIZ**, es un concepto abstracto y matemático. La matriz es una rejilla cuadrada compuesta de un número variable de cuadraditos, cada cuadradito recibe el nombre de **PÍXEL**.

Como la imagen obtenida es una representación bidimensional de un cierto volumen de tejido, esta matriz no es plana sino que tiene un grosor, que se le denomina grosor de corte.

Si se toma un solo píxel, ver **Figura 2.1**, se observa que tiene un grosor (grosor de corte) pues al píxel más el grosor de corte se le denomina **VÓXEL**.



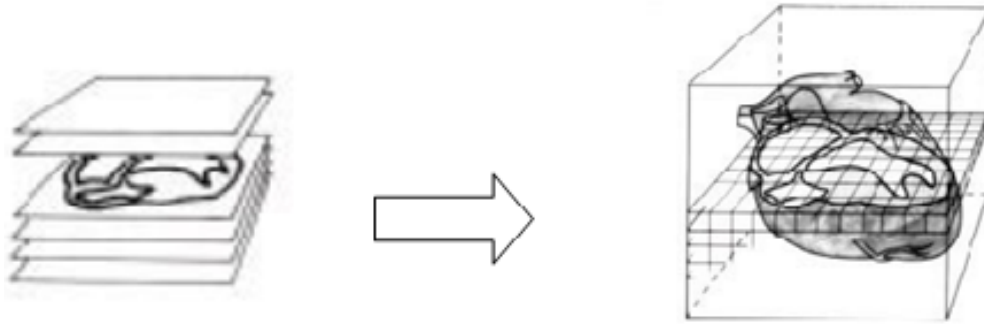
**Figura 2.1** Representación de una Imagen 3D.

Una vez que el ordenador ha obtenido la imagen, a cada píxel se le otorga un valor, gracias a que el ordenador ha digitalizado los datos. Este valor corresponde a la media de atenuación que sufrieron los distintos fotones de Rayos X que después de atravesar al paciente, llegaron a los detectores y que se representan en dicho vóxel. Es decir el coeficiente de atenuación representado en un píxel es la media de todos los coeficientes de atenuación que existan en el volumen del vóxel. No se puede representar algo más pequeño que el vóxel.

Dependiendo del tamaño del objeto a representar y el tamaño de la matriz que se vaya a utilizar, cambiará la resolución espacial de la imagen, la imagen obtenida de una estructura geométrica regular con un borde nítido puede ser borrosa. El grado de borrosidad de dicha imagen es una medida de la resolución espacial del sistema.

El ordenador después de computar toda la información, otorga un valor numérico a cada píxel (que se corresponde con el coeficiente de atenuación), este número del píxel se corresponde con un color en una escala de grises, si se hace esto con todos los píxel se obtiene una amplia gama de grises capaz de representar cualquier imagen.

Una vez obtenido el conjunto de imágenes tridimensionales correspondientes a los cortes paralelos de cada serie de un estudio determinado, se obtiene el volumen final de vóxeles como se muestra en la **Figura 2.2**, que será procesado posteriormente por el algoritmo de segmentación.



**Figura 2.2** Obtención del Volumen final de vóxeles que será luego segmentado.

### 2.3. Segmentación de Imágenes 3D. Voxel Growing.

Los métodos orientados a regiones constituyen un enfoque poderoso de segmentación de imágenes. El objetivo del algoritmo Voxel Growing de crecimiento de regiones es determinar zonas uniformes dentro de la imagen, a partir de sus propiedades locales, basándose en la idea de conectividad y similitud de los vóxeles. En el esquema propuesto es posible efectuar la detección de una o más regiones de interés, de acuerdo a cada aplicación particular, considerando como entrada una matriz tridimensional de vóxeles de dimensiones arbitrarias. El crecimiento se inicia a partir de la especificación de puntos iniciales para cada región y luego, mediante un proceso iterativo, se van incorporando a las mismas los vóxeles de la imagen que satisfacen un criterio de conectividad y similitud determinado, hasta que no se encuentren más elementos que cumplan la condición, obteniendo como salida varias matrices tridimensionales de menores dimensiones que la matriz original, en las cuales aparecen rotuladas las posiciones correspondientes a los vóxeles incluidos en cada una de las regiones.

Numerosos algoritmos de segmentación se aplican a los cortes individuales de la imagen y luego integran los resultados parciales para conformar el volumen total segmentado. En este trabajo, la segmentación se aplica directamente sobre el modelo 3D, evitando de este modo los problemas de puesta en correspondencia entre cortes que poseen los esquemas mencionados. Este esquema resulta conveniente sobre todo en casos de componentes complejas, como por ejemplo en el caso de segmentación de estructuras como cráneo o cerebro, que al poseer numerosas circunvoluciones puede provocar que puntos que no han

sido integrados a la región en alguno de los cortes, sí pueden resultar incluidos en los cortes contiguos. [\[14\]](#)

### 2.3.1. Especificación de la Semilla.

La determinación del punto de partida para la segmentación de las regiones suele ser un aspecto crítico del algoritmo. Algunos trabajos tratan de especificar semillas solo de forma automática, por ejemplo a partir de un análisis del histograma de la imagen o como resultado de una segmentación previa con otras técnicas. Sin embargo, la complejidad y variabilidad de las estructuras anatómicas comúnmente presentes en las imágenes médicas tiende a limitar el alcance de la aplicabilidad de los métodos completamente automáticos. Además, la ubicación de los puntos iniciales sobre zonas ruidosas o correspondientes a bordes puede provocar una segmentación errónea de la imagen, la cual finalmente podría requerir una etapa posterior de corrección. Contrariamente, otros enfoques de segmentación se basan en una especificación de semillas por parte del usuario, a partir de su conocimiento experto.

En este trabajo se han abordado tanto la segmentación automática como la semiautomática con un conocimiento experto previo, considerado este último enfoque y ofreciendo un entorno de visualización con una interface interactiva para la determinación de las semillas, con el fin de facilitar esta tarea a través de la exploración de la imagen, sabiendo que tales puntos ciertamente pertenecen a cada una de las estructuras buscadas. Con el fin de asistir al usuario en este proceso, la herramienta brinda información sobre la posición, tonalidad y gradiente en cada punto. Ver [Anexo D.1](#).

Debido a que la segmentación resultante del crecimiento desde una única semilla suele ser insuficiente en ciertos casos, se ha considerado la posibilidad de poder establecer más de una semilla como una opción del algoritmo. Esto es especialmente conveniente para la segmentación de regiones que no corresponden necesariamente a una única sustancia. Una vez especificada, cada semilla contiene la siguiente información:

- **Posición:** Dada por las coordenadas  $x$ ,  $y$ ,  $z$  dentro del punto dentro del volumen de datos.

- **Intensidad:** El valor de tonalidad de cada punto semilla se utiliza en la evaluación los vóxeles visitados por el proceso de crecimiento.

Cabe destacar que mediante las semillas se especifican los puntos de la imagen que brindan información sobre lo que el usuario considera relevante y, de esta manera, se incorpora al proceso de segmentación, conocimiento relevante acerca de las características de interés. A partir de ella, el algoritmo de crecimiento es capaz de determinar el conjunto de componentes conectadas con propiedades similares que conforman cada región.

### 2.3.2. Aceptación de Vóxeles.

Debido a la variabilidad de intensidades dentro de una misma componente y a la posible presencia de ruido en la imagen, el criterio de similitud utilizado para el crecimiento debe contemplar un cierto valor de tolerancia para las diferencias de tonalidades entre los vóxeles de cada región.

Hay diferentes formas para evaluar la pertenencia o no de un vóxel a una región. El criterio más simple considera la variación de intensidad del vóxel con respecto al valor de la semilla. También se puede tener en cuenta la diferencia de intensidad con el vóxel recientemente incorporado a la región o incluso la variación con respecto al entorno del vóxel. El algoritmo de crecimiento desarrollado provee las tres posibilidades:

1. **Crecimiento según semilla.** En este caso, si la diferencia entre el valor de la semilla  $s$  y el del vóxel analizado se mantiene dentro de un límite de tolerancia  $t$  dado, se acepta su incorporación. Las regiones resultantes se componen de vóxeles que satisfacen el siguiente criterio:  $\text{valor}(v) - \text{valor}(s) \leq t$
2. **Crecimiento según el vecino.** Esta opción considera que un vóxel  $v$  pertenece a la región si la diferencia entre su intensidad y la del vecino  $v'$  que lo precede se mantiene por debajo de  $t$ :  $\text{valor}(v) - \text{valor}(v') \leq t$
3. **Crecimiento por gradiente.** La magnitud del gradiente en un punto es una medida de la variación de intensidades alrededor de ese punto. Esta alternativa de crecimiento utiliza este valor para establecer los límites de la región, ya que el proceso de llenado se detiene si el gradiente en un vóxel  $v$  supera el umbral dado.

Entonces, la condición de aceptación de un Vóxel en este caso es:  
 **$valor(gradiente(v)) \leq t$**

Finalmente, la opción que ha brindado los mejores resultados consiste en la integración del criterio de Crecimiento según semilla unido a la información sobre el valor del gradiente en el punto analizado.

### **2.3.3. Incorporación de Vóxeles.**

Una región inicialmente queda constituida por la semilla que ha sido especificada. Como se mencionó anteriormente, el algoritmo de crecimiento se basa en extender la región por medio de la incorporación progresiva de los vóxeles vecinos a los que ya están integrados a la región, mientras satisfagan el criterio de aceptación establecido. Para ello, los vóxeles visitados que cumplen la condición son insertados en una cola hasta el momento de ser procesados e integrados a la componente de interés.

En cada ciclo del algoritmo, se debe extraer uno de los vóxeles de la cola para incluirlo a la región y luego analizar la posible extensión sobre sus vecinos. Con el fin de adaptar las capacidades del algoritmo de crecimiento al resultado final deseado se ha analizado una estrategia novedosa de inserción de los vóxeles a la región, de forma que el vóxel extraído de la cola solo formará parte de la región en cuestión si y solo si, además de cumplir con el criterio de vecindad, el vóxel es frontera de la región; optimizando de esta forma el consumo de memoria y uso del CPU del algoritmo.

El siguiente pseudocódigo **Algoritmo 1** muestra el esquema general de crecimiento. Inicialmente, la semilla es insertada en la cola. Luego, en cada ciclo, se extrae un elemento de la cola, se le rotula como visitado, evitando su re-evaluación, y se visitan sus vecinos usando conectividad 26 ver **Figura 2.3**, evaluando si éstos satisfacen el criterio de aceptación. Los vecinos que no cumplen con el criterio deseado se marcan como visitados y no se adicionan a la cola. Esta información se tiene en cuenta además durante la etapa de la generación de la geometría de las superficies detectadas.

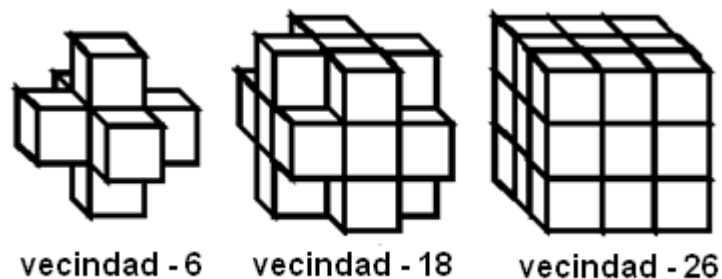
```

VoxelGrowingAlgorithm( Región Ri)
{
  Incertamos de Ri en la Semilla en la cola C.
  Mientras( La cola C no este Vacía)
  {
    Voxel v = extraer el primer elemnto de C.
    Para cada vecino v' de v.
    {
      Si (v' no ha sido visitado y
          satisface el criterio de aceptación)
      {
        Marcar a v' como vistado.
        Adicionar v' a la cola C.
      }
    }
    Sino
      Marcar a v como frontera.
  }

  Si (v es frontera)
  Adicionar v a la region Ri.
}
}

```

**Algoritmo 1** Seudocódigo algoritmo Voxel Growing.



**Figura 2.3** Tipos de conectividades entre un voxel y sus vecinos.

Uno de los aspectos interesantes de esta técnica es que las regiones crecen a través de la incorporación de los vecinos de los vóxeles ya incluidos, por lo que se asegura la conectividad entre los elementos de la región segmentada.

## 2.4.Reconstrucción de Superficie.

El proceso de reconstrucción de modelos anatómicos parte de una segmentación previa de la zona de interés, y a partir de estos datos es posible determinar una malla de polígonos para su posterior utilización, donde estos polígonos generalmente están compuestos por triángulos (en la reconstrucción de superficie) o bien por tetraedros (en reconstrucción de volumen). Anteriormente se discutieron las características fundamentales de algunos métodos que tradicionalmente son utilizados en estos propósitos, como por ejemplo los algoritmos basados en marching (MC y MT) y los basados en Triangulación de Delaunay (EDT y CDT). Ver [\[Epígrafe 1.2\]](#)

En este trabajo de diploma los esfuerzos han sido concentrados en la reconstrucción de superficies, por ser de ambos tipos de reconstrucciones la que mayores bondades brinda en cuestiones de rendimiento y menos gasto de recursos computacionales. Aunque esto no quita que sea fácilmente adaptable a reconstrucción de volumen, si se ajusta el algoritmo para que al realizar la segmentación se tengan en cuenta todos los vóxeles que componen una región, en vez de solamente los que son frontera; y adicionándose al proceso de triangulación que se describe, la formación de los tetraedros.

La técnica que se propone es un enfoque eficiente al problema de la generación de modelos de superficie, aprovechando las ventajas que reporta el método de segmentación escogido en el [\[Epígrafe 2.2\]](#). El algoritmo consta de dos partes fundamentales: La Triangulación inicial y el PostProcesado o Suavizado de la malla.

### 2.4.1.Triangulación Inicial.

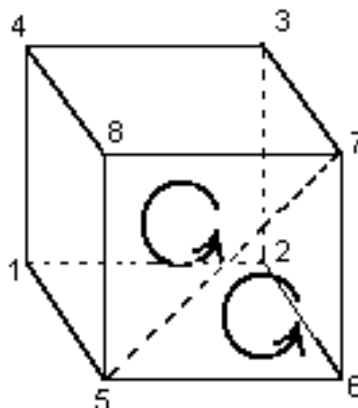
Cada región segmentada provee un marco de referencia para la generación de la descripción poligonal de su superficie. Afortunadamente el proceso de segmentación garantiza que en las regiones solo se almacenen aquellos vóxeles que son frontera (pertenecientes al contorno), donde cada uno posee una referencia a sus vóxeles vecinos. Gracias a esta característica se evita hacer búsquedas innecesarias de los vecinos de cada vóxel, así como la detección de los que conforman el contorno de la región de interés.



En un primer paso, se recorre el volumen segmentado, y por cada vóxel se consideran las caras que son frontera. Las caras fronteras son aquellas que corresponden a los vecinos del vóxel que tengan conectividad 6 y no existan, ver **Figura 2.3**.

A partir de cada una de estas caras frontera se puede generar una triangulación inicial, dividiendo cada una de ellas en dos triángulos, a partir del trazado de una diagonal. Cada vóxel contiene una posición a partir de la cual pueden generarse los vértices que formarán los triángulos. Para obtener superficies cerradas y coherentes, se debe tener cuidado al especificar la dirección del vector normal a cada triángulo y por lo tanto el sentido de los vértices como se observa en la **Figura 2.4**.

Como se expresaba anteriormente, de cada cara frontera se obtienen dos triángulos, que contienen la información de la terna de vértices que lo forman. Pero es necesario generar los vértices que darán lugar a los triángulos, esto se logra a partir del vector de posición del vóxel, al cual se le adiciona la longitud del vóxel según convenga (en los ejes x, y, z). Esto trae como inconveniente que al procesar varios vóxeles se generen vértices repetidos, hecho que influye negativamente en el desempeño del algoritmo de suavizado que se abordará más adelante. [\[17\]](#)



**Figura 2.4** Triangulación de las caras frontera de los vóxeles.

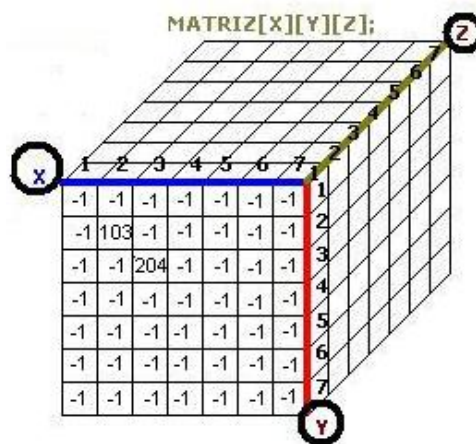
Para garantizar que cada vértice sea adicionado una sola vez a la lista, y todos los datos estén actualizados y sean consistentes, se creó una variante de Tabla Hash Cúbica.

### 2.4.1.1. Tabla Hash Cúbica

Para la creación de la tabla es preciso hacer un pre-procesamiento de los vóxeles para obtener las componentes x, y, z máxima y mínima, para luego crear la Tabla Hash Cúbica (THC), que es una matriz cúbica de enteros (M), cuyas dimensiones responden a (**Xmax-Xmin, Ymax-Ymin, Zmax-Zmin**). Esto garantiza un ahorro sustancial de los recursos de memoria, y una accesibilidad de **O (1)**.

En la matriz se almacenan los índices que ocupan los vértices en una lista. Inicialmente dicha matriz es inicializada con un valor = -1, que significa que no hay ningún vértice adicionado. A medida que los vértices se vayan adicionando a la lista de vértices, la THC será actualizada.

En la THC cada vértice es indexado según su posición, la primera dimensión corresponde a las **X**, la segunda a las **Y**, y la última a las **Z**. De modo que a un vértice con una posición (**X'**, **Y'**, **Z'**), le corresponde la casilla: **M [X' - Xmin][Y' - Ymin][Z' - Zmin]**. Ver **Figura 2.5**



**Figura 2.5** Ejemplo de matriz cúbica, que almacena solo los vértices  $v_1$  ( $minx+2, miny+2, minz$ ) y  $v_2$  ( $minx+3, miny+3, minz$ ) que ocupan las posiciones 103 y 204 respectivamente en la lista de vértices.

### 2.4.2. Suavizado de la Malla.

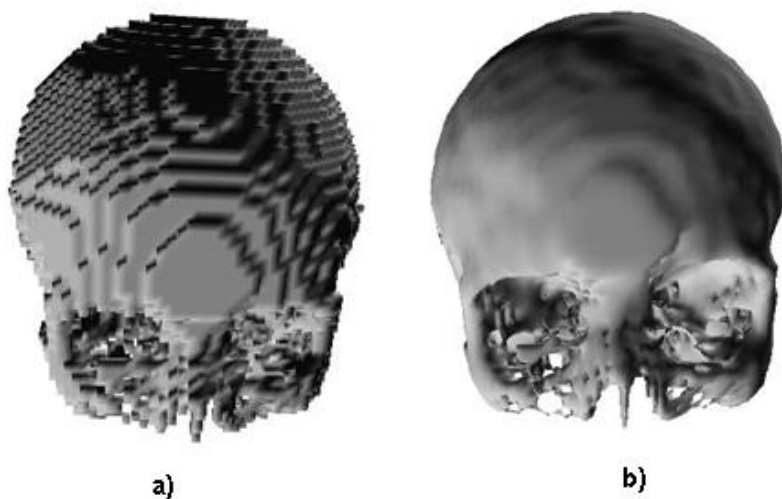
Si bien el dominio generado a partir del conjunto de triángulos que describe la superficie de cada región segmentada corresponde al objeto en estudio, la imagen resultante puede ofrecer un aspecto escalonado que entorpece la visualización. Esto se debe a que los

triángulos se generan a partir de las caras de los vóxeles y por lo tanto muchos de ellos resultan ortogonales entre sí, ver **Figura 2.6a**.

Una manera de mejorar la calidad de la superficie es aplicar un post-procesamiento de suavizado, para disminuir las irregularidades y lograr una apariencia más semejante a las formas originales de los materiales, ver **Figura 2.6b**. El suavizado se realiza mediante la aplicación de un filtro promedio. Para ello, se considera por cada vértice de la malla generada que se desea suavizar, todos los triángulos que poseen un vértice coincidente con él y se calcula su nueva posición como el promedio de las posiciones de los demás vértices de los triángulos involucrados. [17]

El proceso se puede aplicar en forma iterativa, hasta obtener los resultados deseados. Al tratar con superficies de forma convexa, este procesamiento puede provocar una reducción del volumen del cuerpo, ya que los puntos tenderán a moverse hacia el interior de la región, siendo mayor el efecto según la cantidad de iteraciones efectuadas. Por lo tanto, debe ser aplicado moderadamente y sólo a efectos de visualización.

Otra posibilidad es realizar un análisis de frecuencias sobre los vértices que componen la malla de polígonos. A partir de este análisis, es posible clasificar cada uno de los vértices como de baja o alta frecuencia y determinar si es necesario procesarlos. El propósito de este algoritmo es suavizar la superficie, distribuyendo los vértices de forma más homogénea.



**Figura 2.6 (a)** Descripción poligonal de la superficie. Triangulación Inicial. **(b)** Descripción poligonal de la superficie. Superficie Suavizada.

## 2.5. Especificación del formato de fichero a exportar.

Finalizado el proceso de generación de superficies es necesario contar con un mecanismo de persistencia que permita utilizar posteriormente la malla obtenida. Para almacenar los datos se creó un fichero binario, ya que es la mejor opción en estos tipos de aplicaciones por las potencialidades de accesibilidad a la información, tener mayor confidencialidad en los datos y contar con mayor velocidad de carga que su contraparte en texto plano. Este fichero está basado en algunas de las características que poseen algunos formatos contenedores de información 3D ya estandarizados como los 3DS, ASE y DICOM, además tendrá extensión \*.KPS, debido a que será utilizado por un proyecto para desarrollo de Simuladores Quirúrgicos, cuyo producto en desarrollo se denomina KHEIPROS.

### 2.5.1. Estructura Interna.

#### 2.5.1.1. Bloques.

Internamente el archivo está conformado por bloques de datos cuya estructura básica está definida por:

**[Nombre del dato (ND)] [Tipo de dato (TD)] [Tamaño del dato (TBD)] [Valor del dato (VD)]**

Los nombres de cada uno de los campos de esta estructura fueron escogidos exactamente de acuerdo al objetivo que persiguen.

**ND:** Es el símbolo que permite reconocer el dato que está próximo a leerse.

**TD:** Estos tipos de datos son propios del formato y solo es un mecanismo para tener una idea del tipo de dato a leer ya sea simple o compuesto. En la siguiente tabla se tratan los tipos con más detalle.

| Tipo de Dato    | Descripción                                                                                          | Representación en Fichero |
|-----------------|------------------------------------------------------------------------------------------------------|---------------------------|
| <b>BLOCK</b>    | Bloque compuesto de datos, no contiene un <b>VD</b> porque tiene toda la información en su interior. |                           |
| <b>INT</b>      | Entero tradicional                                                                                   | 4                         |
| <b>STRING</b>   | Cadena de Caracteres                                                                                 | “Una cadena”              |
| <b>VECTOR3I</b> | Vector 3D de valores enteros                                                                         | 1 2 3                     |
| <b>VECTOR3F</b> | Vector 3D de valores reales                                                                          | 120.0 3.4 9.6             |
| <b>VECTOR4I</b> | Vector 4D de valores enteros                                                                         | 1 2 3 4                   |
| <b>VECTOR4F</b> | Vector 4D de valores reales                                                                          | 120.0 3.4 9.6 5.1         |
| <b>NORMAL</b>   | Tipo de dato conformado por la unión de un<br>INT + VECTOR3F                                         | 1 120.0 3.4 9.6           |

**Tabla 2.1** Tipos de datos en un archivo \*.KPS.

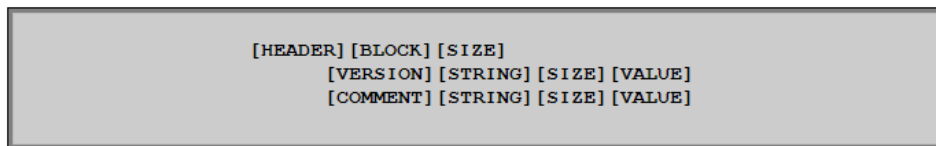
**TBD:** Es el tamaño en bytes del valor que representa el dato, una de las bondades de este atributo es que permite ignorar aquellas informaciones que no sean de interés. Normalmente este campo aparece con el nombre de **SIZE** en la estructura interna del fichero.

**VD:** Es como su nombre indica la información real del dato, es un campo común en todos los tipos de datos excepto en aquellos datos de tipo **BLOCK** como se explica en la **Tabla 2.1**.

Desde el punto de vista informativo los datos más importantes son el Encabezado y el Cuerpo, una de sus características principales es que son datos de tipo **BLOCK** (ver **Tabla 2.1**) lo que significa que en su interior se almacenará la información correspondiente de los modelos que corresponden a cada una de las regiones obtenidas durante el proceso de reconstrucción.

### 2.5.1.2. Encabezado.

El encabezado es un contenedor de informaciones generales que describen aspectos básicos del fichero, es donde se almacena la información referente a la versión del fichero y da la posibilidad al usuario de almacenar otros datos de tipo explicativos dentro del espacio reservado para los comentarios.



**Figura 2.7** Estructura del Encabezado de un fichero \*.KPS.

### 2.5.1.3. Cuerpo.

El cuerpo constituye el bloque principal de información del fichero, sin él no existiría nada que visualizar una vez exportados los modelos reconstruidos, por lo cual es el más importante dentro del archivo. Como bloque principal, el cuerpo contiene en su interior toda la información referente a las regiones, cada una de las cuales está constituida por dos sub-bloques, almacenando primero toda la información de la geometría y después todos los datos referentes a visualización, brindando mayor comodidad y organización.

```

[BODY] [BLOCK] [SIZE]
  [CANT_REGIONS] [INT] [SIZE] [VALUE]

    [REGION] [BLOCK] [SIZE]
      [REGION_ID] [STRING] [SIZE] [VALUE]

        [GEOMETRY] [BLOCK] [SIZE]
          [CANT_VERTEZES] [INT] [SIZE] [VALUE]
          [VERTEX] [VECTOR3F] [SIZE] [VALUE]

            [CANT_FACES] [INT] [SIZE] [VALUE]
            [FACE] [VECTOR3I] [SIZE] [VALUE]

              [CANT_TET] [INT] [SIZE] [VALUE]
              [TET] [VECTOR4I] [SIZE] [VALUE]

                [MATERIAL] [BLOCK] [SIZE]
                  [COLOR] [VECTOR4F] [SIZE] [VALUE]
                  [NORMALS] [BLOCK] [SIZE]
                    [FACES_NORMALS] [BLOCK] [SIZE]
                      [CANT_FACES_NORMALS] [INT] [SIZE] [VALUE]
                      [FACE_NORMAL] [NORMAL] [SIZE] [VALUE]

                        [VERTEZES_NORMALS] [BLOCK] [SIZE]
                          [CANT_VERTEZES_NORMALS] [INT] [SIZE] [VALUE]
                          [VERTEX_NORMAL] [NORMAL] [SIZE] [VALUE]

```

**Figura 2.8** Estructura del Cuerpo de un fichero \*.KPS.

Como se aprecia en la **Figura 2.8** el **BODY** (o cuerpo) es de tipo **BLOCK**, lo que significa que no tiene un campo **VALUE** o (**VD**) sino que contiene bloques de datos internos, el primero de los cuales es la cantidad de regiones que posee el fichero, seguido de todas las regiones.

#### 2.5.1.4.La Región.

Como se expresaba anteriormente cada región esta representada como un **BLOCK**, y en su interior alberga toda la información referente a ella, como su tamaño en bytes (campo **SIZE**) información que puede ser utilizada para saltarse el bloque de datos completo.

Cada región contiene un identificador, que se identifica de tipo **STRING** debido a que puede ser una cadena alfanumérica ("1", "44", "Pulmón", "Reg023S" etc.). Además cuenta con la información de geometría y de los materiales.

### 2.5.1.5. Geometría.

La geometría es un bloque de datos que encierra toda la información de los vértices, los triángulos y tetraedros.

De los vértices se almacena la cantidad existente, seguido de todos los vértices. Cada uno es referenciado por un símbolo **VERTEX** de tipo **VECTOR3F** (ver **Tabla 2.1**) y cuyo valor de dato (**VD**) representa las coordenadas del vértice.

De los triángulos también se almacena la cantidad almacenada, luego cada uno es identificado por el símbolo **FACE** de tipo **VECTOR3I** (ver **Tabla 2.1**) y su (**VD**) está dado por los índices de los vértices del triángulo.

Luego viene el bloque de los tetraedros que es bastante semejante a los anteriores, pues además de la cantidad almacenada y el tamaño en bytes del bloque completo, para cada tetraedro identificado por **TET**, y de tipo **VECTOR4I** se almacena la información (**VD**) de cada uno de los índices de las caras (o triángulos) que lo forman.

### 2.5.1.6. Materiales.

El bloque de los materiales contiene la información necesaria para la correcta visualización de los datos de la región: color y las normales. El color identificado por **COLOR** es de tipo **VECTOR4I** debido a que su **VD** son las 4 componentes de color (**RGBA**).

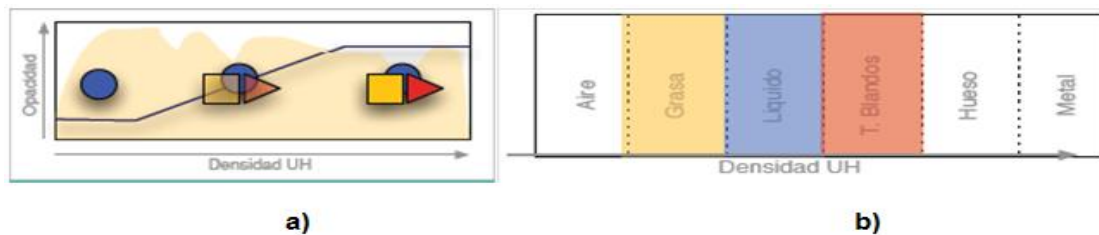
Las normales se definen por el bloque **NORMALS**, que además de los datos tradicionales (tipo **BLOCK**, **SIZE**), contiene además las normales de cada triángulo y de cada vértice. El bloque de las normales de los triángulos se identifica como **FACES\_NORMALS** y además de la cantidad de normales, contiene cada normal de cada triángulo representado por **FACE\_NORMAL** de tipo **NORMAL** (ver **Tabla 2.1**), donde el **VD** representa el índice del triángulo seguido de las coordenadas de su vector normal. El bloque **VERTEXES\_NORMALS** es muy semejante al anterior solo que en este caso se trabaja con los vértices, la cantidad hace referencia a las normales de los vértices, cada normal está representada con **VERTEX\_NORMAL**, y el **VD** se refiere al índice del vértice precediendo las coordenadas de la normal.



## 2.6. Visualización. OpenGL y Raytracing.

Para la visualización 3D utilizamos la librería gráfica OpenGL la cual implementa una simulación básica de un raytracing cuando sobre las primitivas visualizadas se le activan los colores, transparencia, normales, materiales y luces.

La librería se encarga de modelar la manera de cómo millones de rayos luz virtuales atravesarían un volumen de tejido y es la base de la visualización de volúmenes. En su forma más sencilla se asume que los tejidos con mayores coeficientes de atenuación (**UH**) son más "sólidos" y por lo tanto un rayo de luz virtual que trate de atravesar un tejido logrará pasar en mayor o menos grado según la solidez de este tejido. Este haz de luz simulado está sujeto a las mismas leyes físicas de reflexión y refracción, calculadas para cada haz de luz que interactúa con nuestro volumen de datos. Este es el principio detrás del concepto más importante de las representaciones 3D en TAC, las tablas de opacidad. Ver **Figura 2.9a** junto a las tablas o mapas de color, ver **Figura 2.9b**.

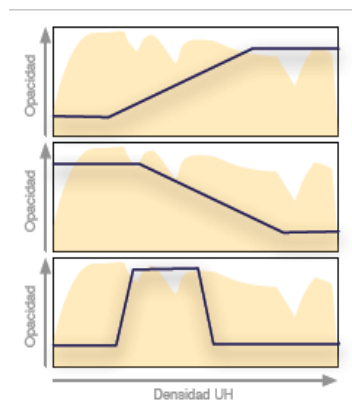


**Figura 2.9 (a) Tablas de Opacidad. (b) Tablas o Mapas de Color.**

### 2.6.1. Tablas de Opacidad.

En esencia, las tablas de opacidad permiten manipular la transparencia de los diferentes tejidos, asignando un valor de opacidad según el valor de **UH**. El tipo de curva que se utilice nos indica qué tejidos se verán. De esta forma, todos los tejidos por debajo de cierta UH tendrán una opacidad baja, y aumentan gradualmente hasta un 100%. La manipulación de transparencias en las tablas de opacidad nos permite visualizar selectivamente los diferentes tejidos.

Estas manipulaciones de transparencias se evidencian mediante varios tipos de curvas de opacidad como se muestra en la **Figura 2.10**. Las más utilizadas son las curvas ascendentes, porque son estas las más intuitivas (objetos más densos son más opacos). Las curvas inversas son útiles para resaltar estructuras de baja densidad y las curvas trapezoidales nos permiten visualizar selectivamente un tipo de estructura con valores UH muy marcados.



**Figura 2.10** *Curvas de Opacidad.*

### 2.6.2. Tablas o Mapas de Color.

Una vez se tengan visibles los tejidos deseados, es útil asignarles colores. Nuevamente se evidencia la limitación del DICOM, que solamente posee la Unidad de Hounsfield como criterio para diferenciar los tejidos. Por eso es preciso asignarle artificialmente colores a ciertos rangos de **UH** para crear las estructuras. Estas técnicas se llaman tablas o mapas de color, y funcionan de manera muy similar a las tablas de opacidad. Ver **Figura 2.9b**.

Se pueden asignar tantos colores como se desee y hacer que las transiciones entre colores sean graduales, para obtener un efecto más natural.

## **Capítulo 3. Descripción de la Solución Propuesta.**

En este capítulo se hará una descripción a nivel conceptual de la solución propuesta en el capítulo anterior. Se definen las condiciones a tener en cuenta para el correcto funcionamiento del sistema, relacionadas como reglas del negocio. Se tratarán los conceptos más importantes del área de interés mediante un modelo de dominio. Se describirán las capacidades que deberá cumplir el sistema a partir de los requisitos funcionales, así como las características del mismo expresadas por los requisitos no funcionales. Además se definirán los casos de uso del sistema y actores que interactúan con el mismo.

### **3.1.Reglas del Negocio.**

Los ficheros de imágenes médicas digitales (DICOM) deben estar en su versión nativa, porque el intérprete que se implementó solo responde a las especificaciones del estándar, y en la actualidad existen muchas variantes del fichero creadas por un considerable número de fabricantes de equipos de TAC y MRI.

### **3.2.Modelo de Dominio.**

A continuación se representa un acercamiento a la solución propuesta, donde se modelan los principales conceptos con los que se trabajarán en el desarrollo de la solución, así como las relaciones existentes entre ellos.

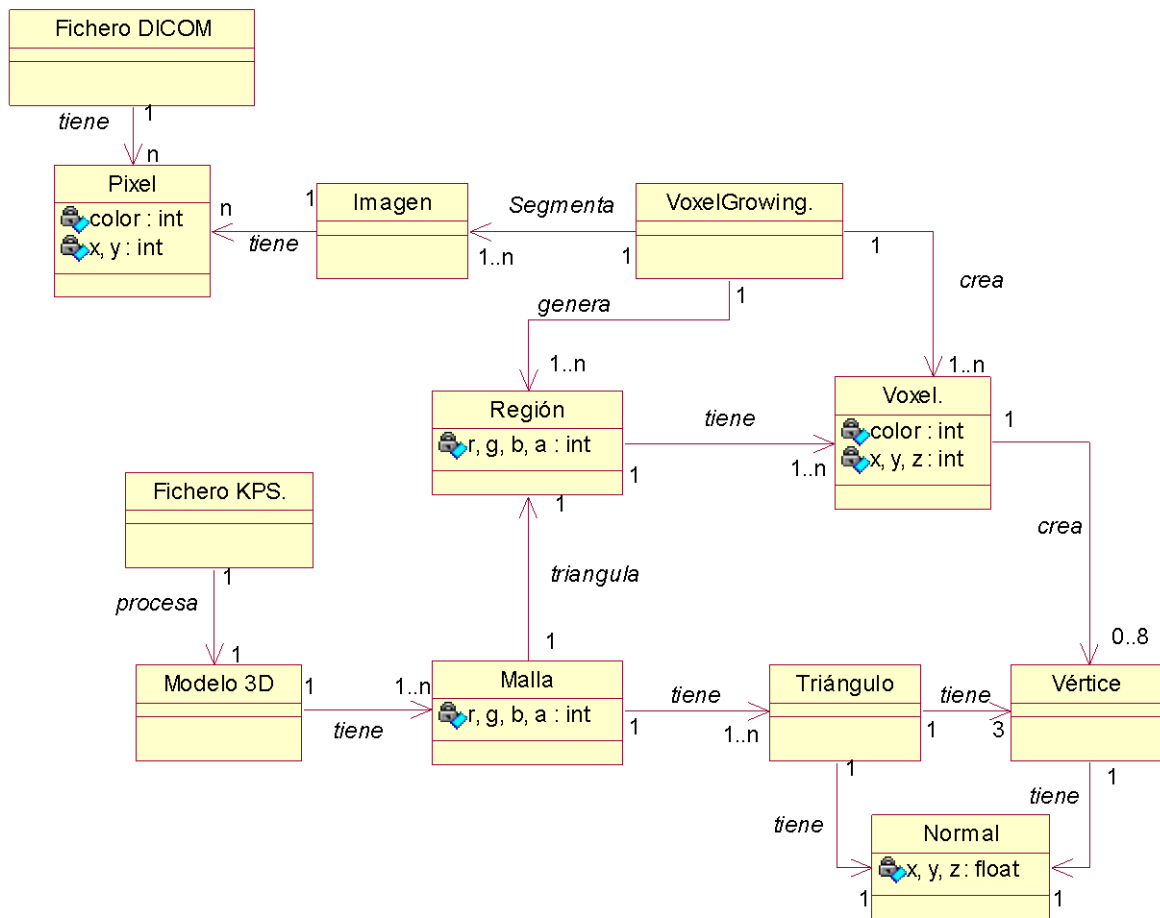


Figura 3.1 Modelo de dominio de la aplicación.

### 3.3.Glosario de Términos.

A continuación se relacionan los conceptos presentes en el modelo anterior, con sus respectivas especificaciones para facilitar su comprensión.

**Fichero DICOM:** Representa una abstracción de los ficheros DICOM, que son obtenidos directamente de los equipos de TAC, MRI, entre otros. Contienen toda la información a partir de la cual se desencadena el proceso de reconstrucción, entre la que se encuentra la información referente a los píxeles.

**Píxel:** Es la mínima unidad de información dentro de una imagen, se obtienen del fichero DICOM. Entre sus atributos están el color y la posición que ocupa en la imagen.

**Imagen:** Es como su nombre indica, una imagen definida a partir del conjunto de píxeles obtenidos del fichero DICOM. Sobre estas imágenes se desarrollará el proceso de segmentación.

**Vóxel:** Son obtenidos a partir de cada uno de los píxeles que forman las imágenes, por lo que constituyen sus homólogos 3D, además constituyen la mínima unidad de información de una Región, y el elemento básico del algoritmo de segmentación. A partir de los datos que posee es posible generar hasta 8 vértices, y de la unión de estos se generarán los triángulos que darán lugar a la malla triangulada. De ahí la importancia que tienen.

**Región:** Es una estructura anatómica del cuerpo que ha sido reconocida. Es la agrupación de varios vóxeles con similares propiedades de color y que además cumplan con los criterios de vecindad (distancia entre ellos  $\leq 3$ ), la combinación de una o varias regiones es lo que resulta del proceso de segmentación.

**Voxel Growing:** Realiza el procesamiento de las imágenes Incluyendo la segmentación. Transforma las imágenes en un conjunto de vóxeles, sobre los cuales realiza el proceso de segmentación, obteniendo como resultado las regiones.

**Vértice:** Es la mínima unidad de un triángulo, es un punto en el espacio que tiene un vector normal. Estos vértices son obtenidos a partir de los datos contenidos en los vóxeles, como se expresó anteriormente.

**Triángulo:** Es la agrupación de 3 vértices, además es el elemento básico de una malla, y posee también un vector normal.

**Normal:** En el modelo mallado los vectores normales se utilizan para representar la orientación del triángulo o del vértice. Es muy útil para obtener mejores resultados en la iluminación de la escena y dar mejor calidad visual a los modelos 3d.

**Malla:** Es el resultado de la triangulación de una Región, por lo que constituye una estructura anatómica 3d en sí misma. Está integrada por un conjunto de triángulos, sus respectivos vértices y las normales de ambos.

**Modelo 3D:** Es la unión de un conjunto de mallas, es como tal el resultado de todo el proceso de reconstrucción en la memoria. Para convertir un modelo en algo observable por terceros, es preciso convertirlo en un Fichero KPS.

**Fichero KPS:** Es el fichero que puede ser utilizado por terceros, en el que se encuentra almacenada toda la información del modelo 3D generado durante el proceso de reconstrucción.

## **3.4.Captura de Requisitos.**

Los requisitos constituyen capacidades o condiciones que el sistema debe cumplir. Estos facilitan el entendimiento entre usuarios y desarrolladores del sistema a elaborar. A continuación se exponen los requisitos funcionales por los que se regirá el sistema y los no funcionales que exponen las características de la aplicación.

### **3.4.1.Requisitos Funcionales.**

Los siguientes requisitos establecen las funcionalidades e instrucciones que el sistema debe cumplir en su implementación.

#### ***R1. Cargar Archivos.***

El sistema permitirá:

R1.1. Seleccionar el directorio donde se encuentran guardados los ficheros.

R1.2. Cargar Archivos en Formato DICOM.

R1.3. Cargar Archivos en Formato KPS.

#### ***R2. Segmentar Imágenes.***

El sistema posibilitará:

R2.1 Definir regiones de forma automática en las imágenes.

R2.2 Definir regiones de forma semiautomática en las imágenes.

R2.3 Realizar el proceso de segmentación semiautomática hasta que el usuario experto obtenga los resultados deseados.

### **R3. Reconstrucción de modelos 3D.**

El sistema debe realizar:

R3.1. La generación de los modelos 3D a partir de las regiones definidas en el proceso de segmentación.

R3.2 El mallado del modelo utilizando como primitiva básica los triángulos.

R3.3 El refinamiento de las mallas obtenidas en la generación del modelo 3D.

R3.4 La optimización de la geometría relativa al modelo obtenido.

R4. El sistema exportará toda la información generada durante el proceso de reconstrucción en un fichero binario, de extensión \*.KPS.

### **3.4.2.Requisitos No Funcionales.**

Los siguientes requisitos son propiedades o cualidades que el sistema debe tener, estableciendo de esta forma aspectos que regulan el comportamiento del mismo.

#### **1. Requerimientos de Software**

Se requiere el SO Windows XP o superior.

#### **2. Requerimientos de Hardware**

Microprocesador Intel Pentium 4 a 3.00GHz.

Memoria RAM de 512MB DDR2.

Tarjeta de Video: Intel 82945G de 128MB, onboard.

#### **3. Restricciones en el Diseño y la implementación**

Se utilizará el lenguaje de programación C++ bajo el paradigma de programación Orientado a Objeto.

#### **4. *Requerimientos de Seguridad***

Confiabilidad: Es estrictamente necesario que los modelos generados por la aplicación, reproduzcan acertadamente la anatomía humana.

Integridad: No debe haber pérdidas de información ni de calidad en las imágenes obtenidas.

#### **5. *Requisitos de Usabilidad***

Los usuarios deben tener un dominio básico de los conceptos fundamentales que rigen los procesos de segmentación de imágenes y reconstrucción 3D aplicados a la medicina.

#### **6. *Requerimientos de Soporte***

Compatibilidad con el Sistema Operativo Windows y cualquier distribución de Linux.

#### **7. *Legales***

Se registrará por las normas ISO 9000.

### **3.5. Modelo de Casos de Uso del Sistema.**

En esta sección se reconocen los actores del sistema a desarrollar, y se definen los casos de uso del sistema. Además, se seleccionan los casos de uso correspondientes al primer ciclo de desarrollo para hacerles sus especificaciones textuales en formato expandido.

#### **3.5.1. Actores del Sistema.**

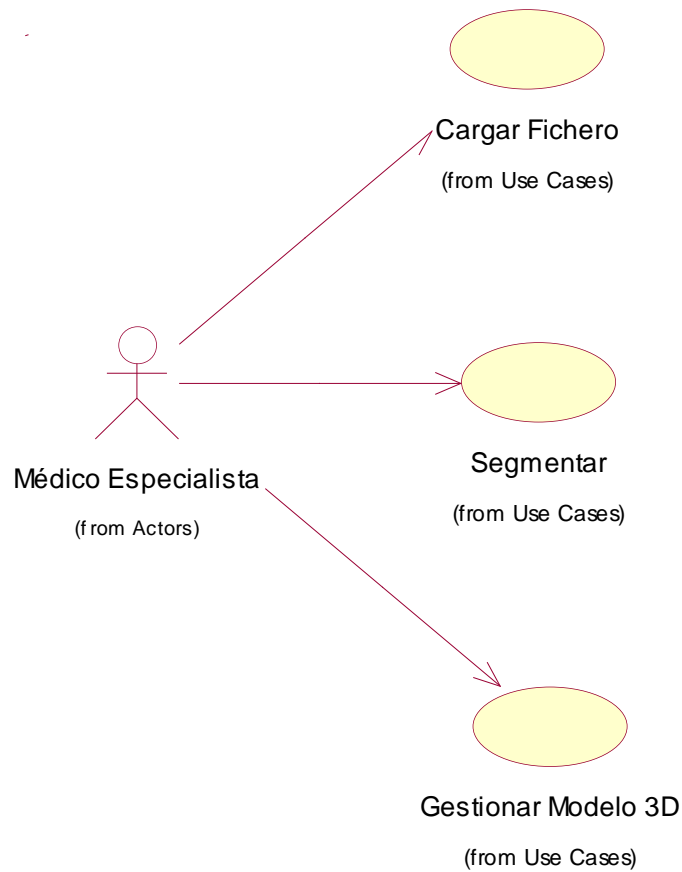
Los actores de un sistema son agentes externos, roles que las personas (usuarios) o dispositivos juegan cuando interactúan con el software. En este caso particular quien hará uso del sistema será un especialista médico, que como actor del sistema será llamado médico especialista.



| Actores             | Justificación                                                                                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Médico Especialista | Es quien interactúa con el sistema para ejecutar las funcionalidades de: Cargar Imágenes DICOM pertenecientes a uno o varios estudios realizados a un paciente, segmentarlas y generar modelos 3D a partir de estas. |

**Tabla 3.1** Actor del Sistema.

### 3.5.2. Diagrama de Casos de Uso del Sistema.



**Figura 3.2** Diagrama de Casos de Uso del Sistema.

### 3.5.3.Descripción de Casos de Uso del Sistema.

Cada caso de uso tiene una descripción de las funcionalidades que ejecutará el sistema propuesto como respuesta a las acciones del usuario. Las tablas presentadas a continuación argumentan los flujos operacionales de cada caso de uso.

|                                                                  |                                                                                                                                                                                            |  |
|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Caso de Uso:</b>                                              | Cargar Fichero                                                                                                                                                                             |  |
| <b>Actores:</b>                                                  | Médico Especialista                                                                                                                                                                        |  |
| <b>Propósito:</b>                                                | Su propósito es interpretar los ficheros *.DICOM                                                                                                                                           |  |
| <b>Resumen:</b>                                                  | El caso de uso de inicia cuando el usuario selecciona la opción de cargar fichero. Selecciona los ficheros DICOM que desea cargar. El sistema procede a cargar los ficheros especificados. |  |
| <b>Referencia:</b>                                               | R1.1, R1.2                                                                                                                                                                                 |  |
| <b>Flujo Normal de Eventos</b>                                   |                                                                                                                                                                                            |  |
| <b>Acción del Actor</b>                                          | <b>Respuesta del Sistema</b>                                                                                                                                                               |  |
| 1. El usuario selecciona la opción de cargar ficheros DICOM.     | 1.1 Muestra un cuadro de diálogo que le permitirá al usuario seleccionar el directorio y los ficheros que desea cargar.                                                                    |  |
| 2. Selecciona el directorio y los ficheros que se desean cargar. | 2.1. El cuadro de diálogo de abrir fichero solo mostrará solo los ficheros de la extensión especificada por el usuario en directorio seleccionado.                                         |  |

|                             |                                                                                                                                                                                                           |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. Oprime el botón aceptar. | <p>3.1 Cierra el cuadro de diálogo y se procede a cargar los ficheros seleccionados por el usuario.</p> <p>3.2 Muestra en pantalla las imágenes definidas en cada uno de los ficheros DICOM cargados.</p> |
| <b>Postcondiciones:</b>     | Se visualiza en pantalla las imágenes definidas en cada uno de los ficheros DICOM cargados.                                                                                                               |
| <b>Prioridad:</b>           | Crítica.                                                                                                                                                                                                  |

**Tabla 3.2** Descripción CU Cargar Ficheros.

|                                                |                                                                                                                                                                                                                                     |  |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Caso de Uso:</b>                            | Segmentar Imágenes                                                                                                                                                                                                                  |  |
| <b>Actores:</b>                                | Médico Especialista                                                                                                                                                                                                                 |  |
| <b>Propósito:</b>                              | Su propósito es identificar las diferentes regiones presentes en una imagen DICOM.                                                                                                                                                  |  |
| <b>Resumen:</b>                                | El caso de uso de inicia cuando el usuario selecciona la opción de segmentar. El sistema realiza la segmentación de forma automática o semiautomática. Se visualizan las regiones identificadas durante el proceso de segmentación. |  |
| <b>Referencia:</b>                             | R2.1, R2.2, R2.3                                                                                                                                                                                                                    |  |
| <b>Precondición:</b>                           | Debe haberse ejecutado el CU Cargar Fichero.                                                                                                                                                                                        |  |
| <b>Flujo Normal de Eventos</b>                 |                                                                                                                                                                                                                                     |  |
| <b>Acción del Actor</b>                        | <b>Respuesta del Sistema</b>                                                                                                                                                                                                        |  |
| 1. Selecciona la opción de segmentar imágenes. | 1.1. Muestra una ventana con la relación de los parámetros que intervienen en el                                                                                                                                                    |  |

|                                                                                                         |                                                                                                           |
|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
|                                                                                                         | proceso de segmentación con sus valores por defecto.                                                      |
| 2. Inicia el proceso de segmentación pudiendo o no modificar los parámetros que se ofrecen por defecto. | 2.1. Ejecuta el proceso de segmentación.<br>2.2 Muestra el resultado de la segmentación                   |
| <b>Flujos Alternos</b>                                                                                  |                                                                                                           |
| <b>Acción del Actor</b>                                                                                 | <b>Respuesta del Sistema</b>                                                                              |
| 1. Selecciona la opción de segmentar imágenes.                                                          | 1.1 Si no hay imágenes cargadas o no se ha especificado una serie el sistema muestra un mensaje de error. |
| <b>Postcondiciones:</b>                                                                                 | Se visualiza en la pantalla el resultado obtenido del proceso de segmentación.                            |
| <b>Prioridad:</b>                                                                                       | Crítica.                                                                                                  |

**Tabla 3.3** Descripción CU Segmentar.

|                     |                                                                                                                                                                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Caso de Uso:</b> | Gestionar Modelo 3D                                                                                                                                                                                                                                                                                                     |
| <b>Actores:</b>     | Médico Especialista                                                                                                                                                                                                                                                                                                     |
| <b>Propósito:</b>   | Su propósito es obtener el modelo 3D a partir de una serie de imágenes DICOM y exportar dicho modelo en formato *.KPS, así como importar un modelo *.KPS previamente exportado.                                                                                                                                         |
| <b>Resumen:</b>     | El caso de uso inicia cuando el usuario selecciona la opción de generar el modelo 3D a partir de las regiones definidas en el proceso de segmentación. Puede ser inicializado también al seleccionar la opción de exportar un modelo generado, o al escoger la opción de importar un modelo exportado con anterioridad. |

|                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Referencia:</b>                                                                         | R1.3, R3.1, R3.2, R3.3, R3.4, R4                                                                                                                                                                                                                                                                                                                                                     |
| <b>Precondición:</b>                                                                       | Debe haberse realizado el Caso de Uso Segmentar.                                                                                                                                                                                                                                                                                                                                     |
| <b>Flujo Normal de Eventos</b>                                                             |                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Acción del Actor</b>                                                                    | <b>Respuesta del Sistema</b>                                                                                                                                                                                                                                                                                                                                                         |
| 1. El actor puede ejecutar la opción Generar Modelo 3D, Exportar Modelo o Importar Modelo. | <p>1.1 El sistema ejecuta algunas de las siguientes acciones:</p> <ul style="list-style-type: none"> <li>a) Si el actor desea generar el modelo 3d consultar sección Generación del Modelo.</li> <li>b) Si el actor desea exportar el modelo 3d consultar sección Exportar Modelo.</li> <li>c) Si el actor desea importar el modelo 3d consultar sección Importar Modelo.</li> </ul> |
| <b>Postcondiciones:</b>                                                                    | Se genera un modelo 3d o se exporta el modelo generándose un fichero .KPS o se importa un modelo visualizando el contenido.                                                                                                                                                                                                                                                          |
| <b>Prioridad:</b>                                                                          | Crítica.                                                                                                                                                                                                                                                                                                                                                                             |

**Tabla 3.4** Descripción CU Gestionar Modelo 3D.

|                                                              |                                                                                                                                                                                                                          |  |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <b>Sección:</b>                                              | Generación del modelo.                                                                                                                                                                                                   |  |
| <b>Flujo Normal de Eventos</b>                               |                                                                                                                                                                                                                          |  |
| <b>Acción del Actor</b>                                      | <b>Respuesta del Sistema</b>                                                                                                                                                                                             |  |
| 1. Selecciona la opción de Generar el modelo tridimensional. | <p>1.1. Realiza una triangulación inicial de las regiones obtenidas en el proceso de segmentación.</p> <p>1.2. Agrega al modelo todas las estructuras trianguladas refinadas.</p> <p>1.3 Muestra el modelo generado.</p> |  |
| <b>Flujos Alternos</b>                                       |                                                                                                                                                                                                                          |  |
| <b>Acción del Actor</b>                                      | <b>Respuesta del Sistema</b>                                                                                                                                                                                             |  |
| 1. Selecciona la opción de Generar el modelo tridimensional. | 1.1 Si no hay imágenes cargadas o no se ha especificado una serie el sistema muestra un mensaje de error.                                                                                                                |  |
| <b>Postcondiciones:</b>                                      | Se visualizará en la pantalla el modelo generado.                                                                                                                                                                        |  |
| <b>Prioridad:</b>                                            | Crítica.                                                                                                                                                                                                                 |  |

**Tabla 3.5** Descripción de la sección Generar Modelo del CU Gestionar Modelo 3D.

|                                                                             |                                                                                                                      |  |
|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|--|
| <b>Sección:</b>                                                             | Exportar modelo.                                                                                                     |  |
| <b>Flujo Normal de Eventos</b>                                              |                                                                                                                      |  |
| <b>Acción del Actor</b>                                                     | <b>Respuesta del Sistema</b>                                                                                         |  |
| 1. Selecciona la opción de exportar el modelo tridimensional.               | 1.1 Muestra un cuadro de diálogo solicitando el nombre con que será exportado el modelo.                             |  |
| 2. Introduce el nombre que tendrá el fichero en que se exportará el modelo. | 2.1. Procede a exportar el modelo.                                                                                   |  |
| <b>Flujos Alternos</b>                                                      |                                                                                                                      |  |
| <b>Acción del Actor</b>                                                     | <b>Respuesta del Sistema</b>                                                                                         |  |
| 1. Selecciona la opción de Exportar el modelo tridimensional.               | 1.1 Si no existe ningún modelo generado o previamente importado el sistema muestra un mensaje de error.              |  |
| <b>Postcondiciones:</b>                                                     | Se genera el fichero con extensión *.KPS correspondiente al modelo 3D obtenido durante el proceso de reconstrucción. |  |
| <b>Prioridad:</b>                                                           | Secundario.                                                                                                          |  |

**Tabla 3.6** Descripción de la sección Exportar Modelo del CU Gestionar Modelo 3D.

|                                                               |                                                                                                          |  |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|--|
| <b>Sección:</b>                                               | Importar Modelo.                                                                                         |  |
| <b>Flujo Normal de Eventos</b>                                |                                                                                                          |  |
| <b>Acción del Actor</b>                                       | <b>Respuesta del Sistema</b>                                                                             |  |
| 1. Selecciona la opción de Importar un modelo tridimensional. | 1.1. Muestra un cuadro de diálogo solicitando el nombre del fichero *.KPS a importar.                    |  |
| 2. Selecciona el fichero *.KPS que desea que se importe.      | 2.1 Carga el fichero *.KPS escogido por el usuario.<br><br>2.2 Visualiza el modelo importado al usuario. |  |
| <b>Postcondiciones:</b>                                       | Se visualizará en la pantalla el modelo generado.                                                        |  |
| <b>Prioridad:</b>                                             | Opcional.                                                                                                |  |

**Tabla 3.7** Descripción de la sección Importar Modelo del CU Gestionar Modelo 3D.

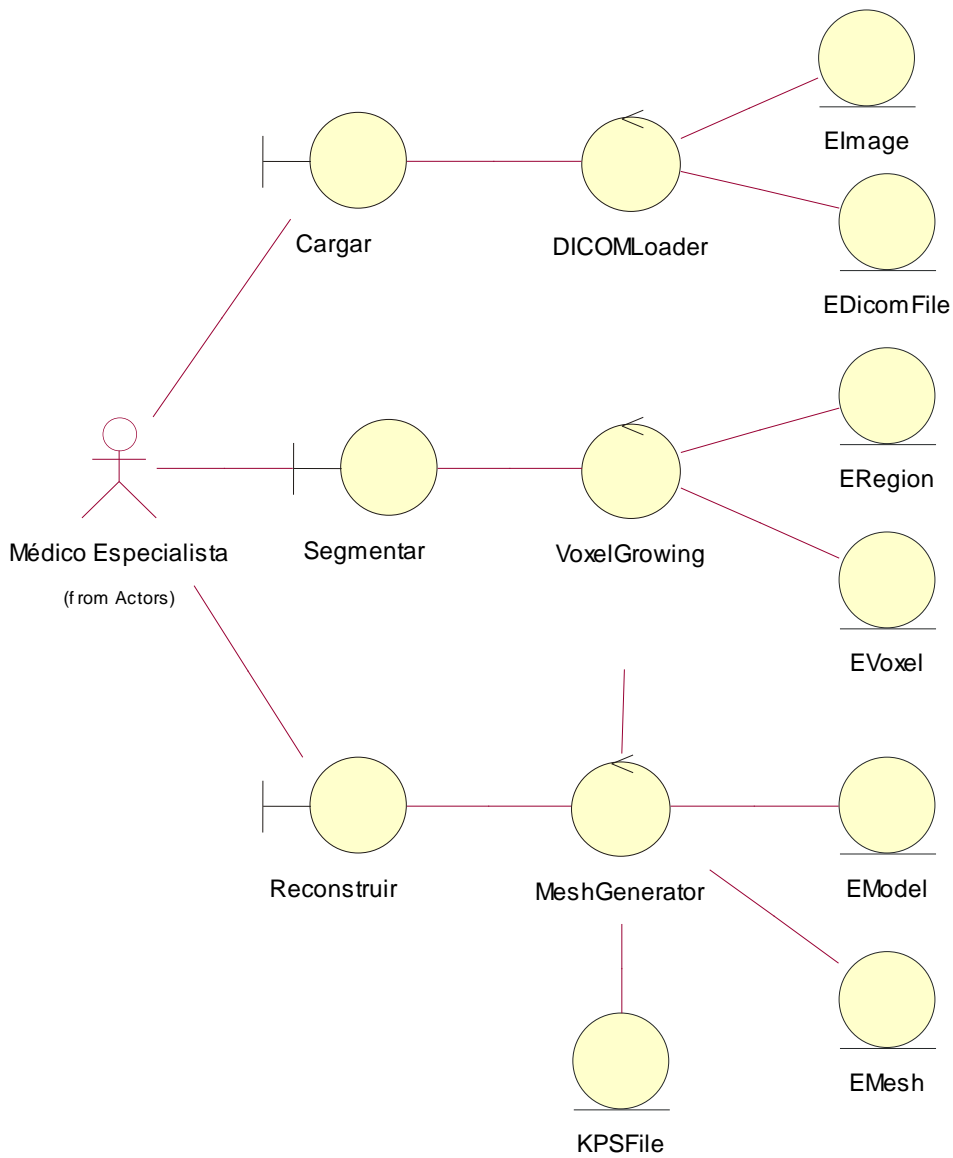


## **Capítulo 4. Análisis y Diseño del Sistema.**

En este capítulo se profundiza aún más en el desarrollo del sistema, dando continuidad a lo tratado en el capítulo anterior. En la primera parte se abordan los temas del análisis, identificados con los diagramas de clases del análisis y los diagramas de secuencia correspondientes. A partir del Análisis se realizará el diseño de la aplicación, soportada sobre los diagramas de clases del diseño y sus respectivos diagramas de interacción.

### **4.1. Diagrama de Clases de Análisis.**

Con vistas a esclarecer la comprensión global del sistema se decidió realizar un diagrama general de clases del análisis, en vez de un diagrama por Caso de Uso como tradicionalmente se hace.



**Figura 4.1** *Diagrama General de Clases de Análisis.*

## 4.2. Diagramas de Secuencia.

Con motivo de aclarar la interacción entre objetos de las clases del análisis se realizan los diagramas de secuencia de cada uno de los CU, capturados en el capítulo anterior.

### 4.2.1. Diagrama de Secuencia del Caso de Uso Cargar Fichero.

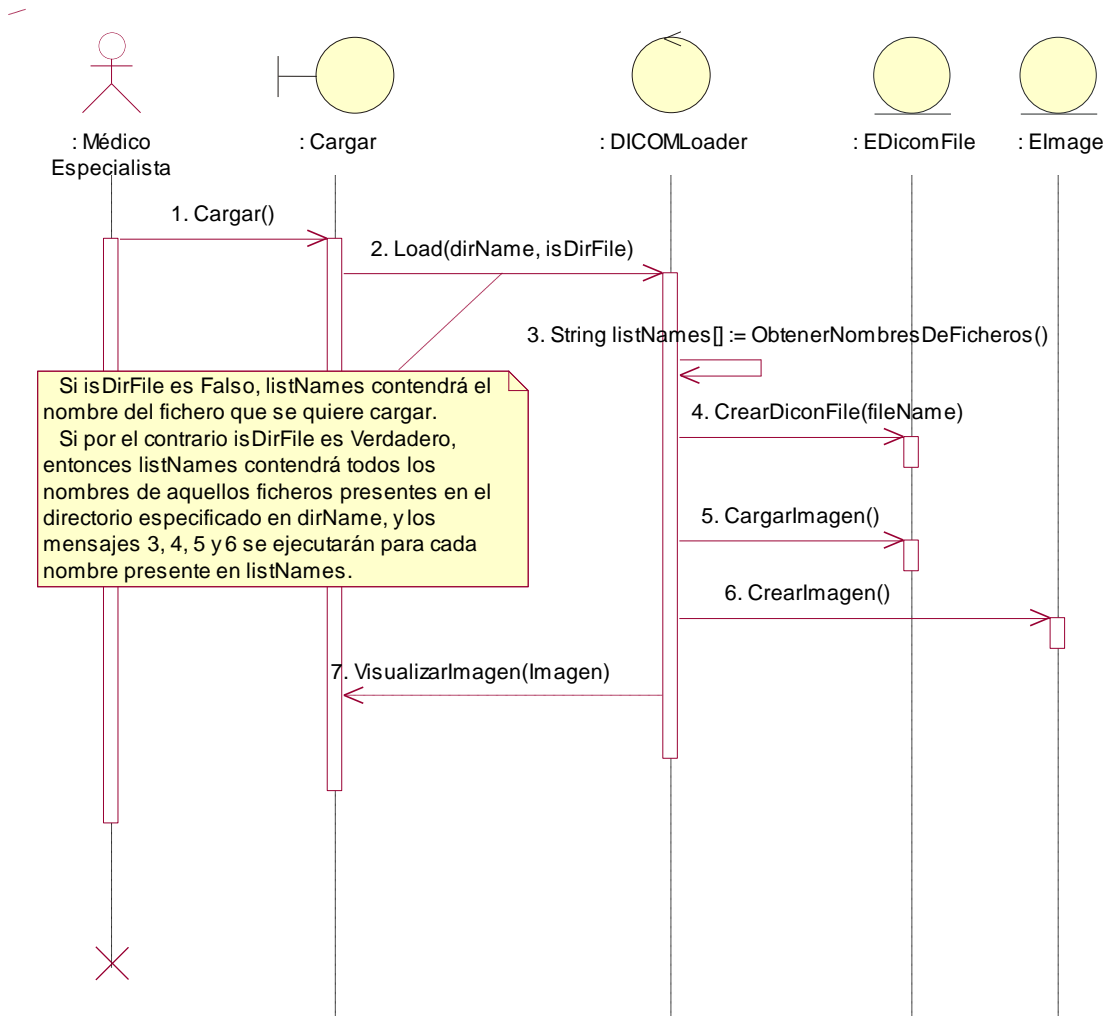


Figura 4.2 Diagrama de Secuencia Caso de Uso Cargar Fichero.

### 4.2.2. Diagrama de Secuencia del Caso de Uso Segmentar.

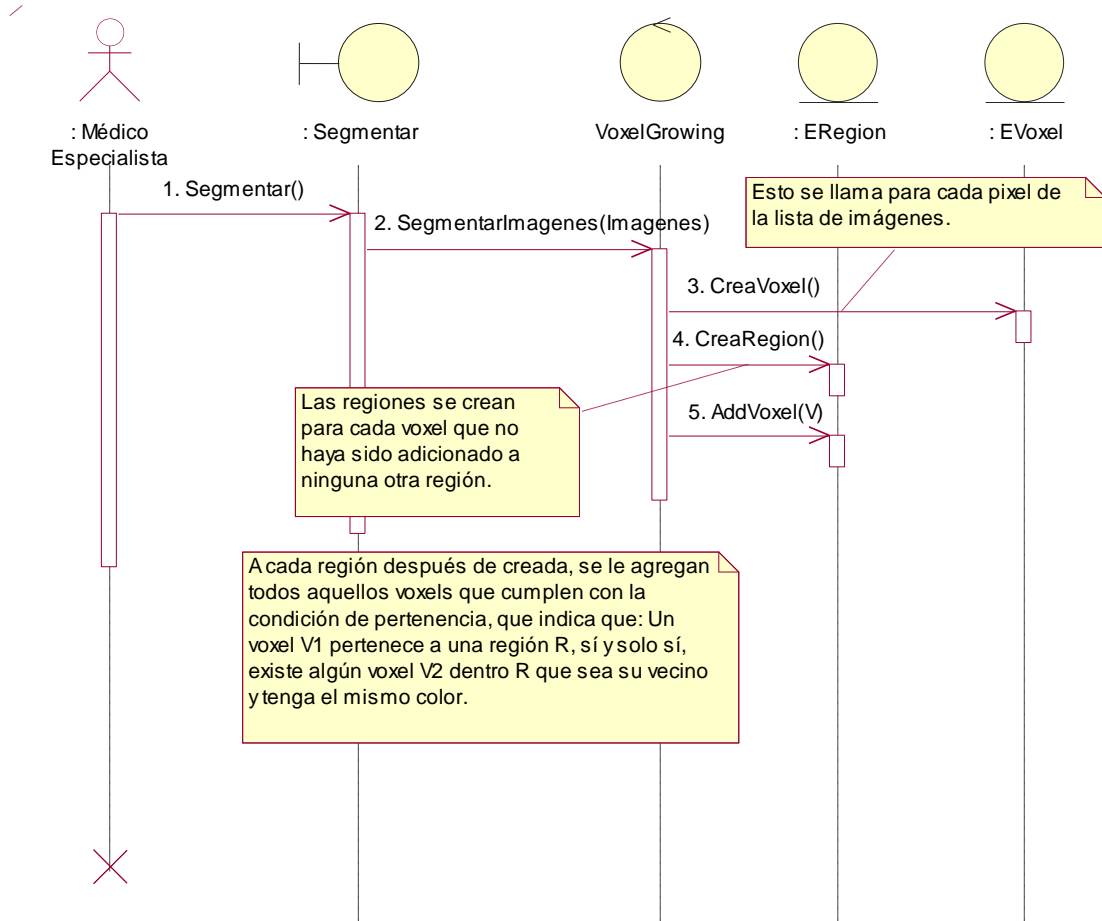
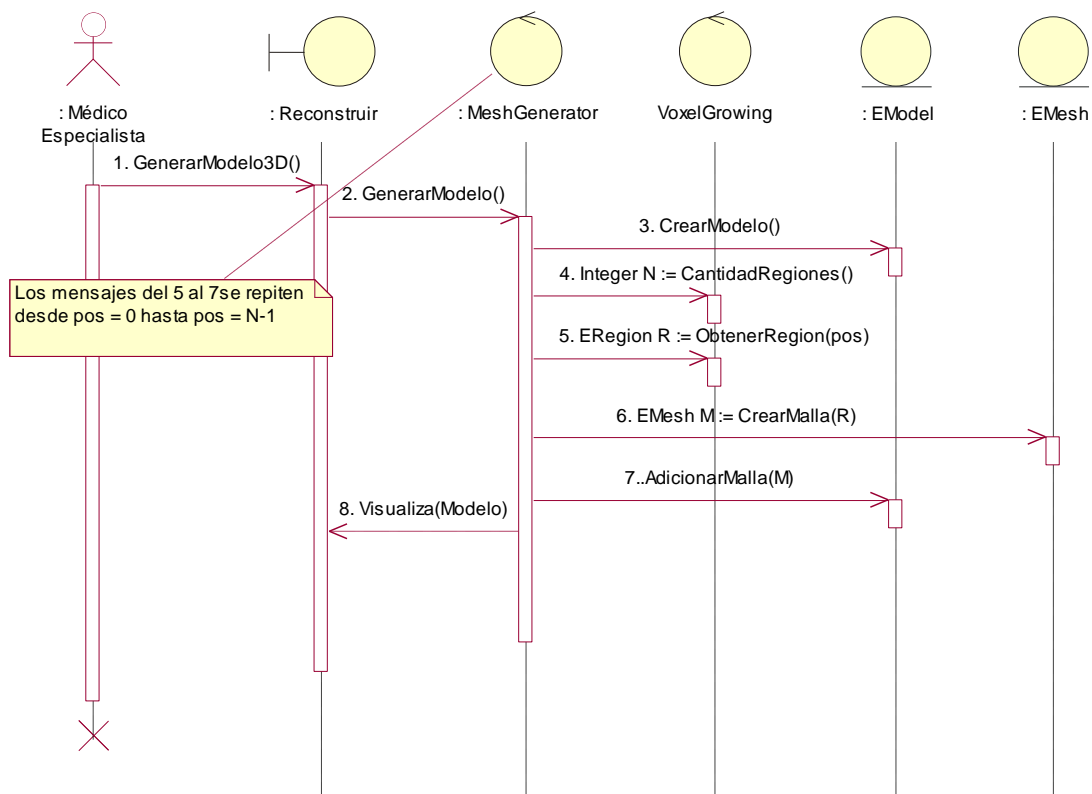


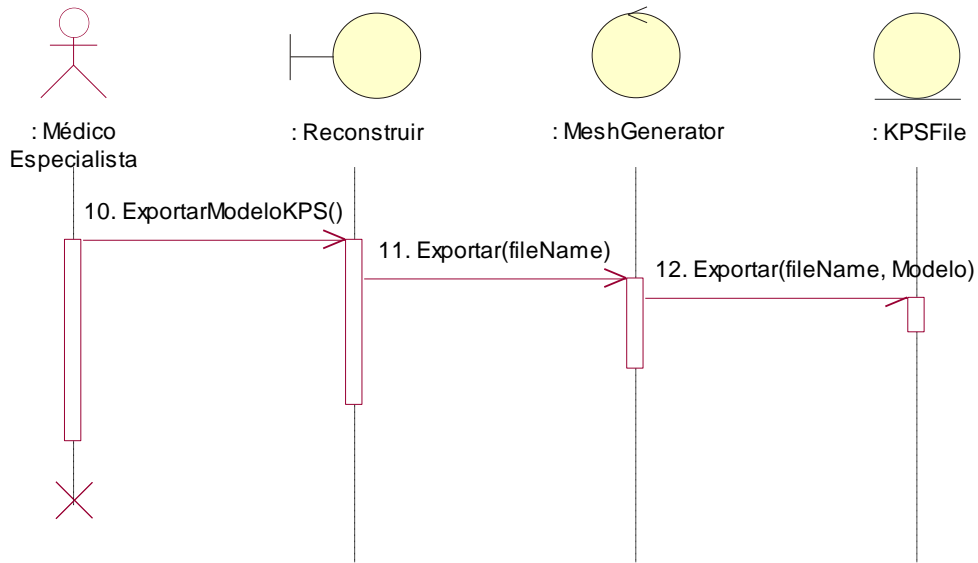
Figura 4.3 Diagrama de Secuencia Caso de Uso Segmentar.

### 4.2.3. Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D.

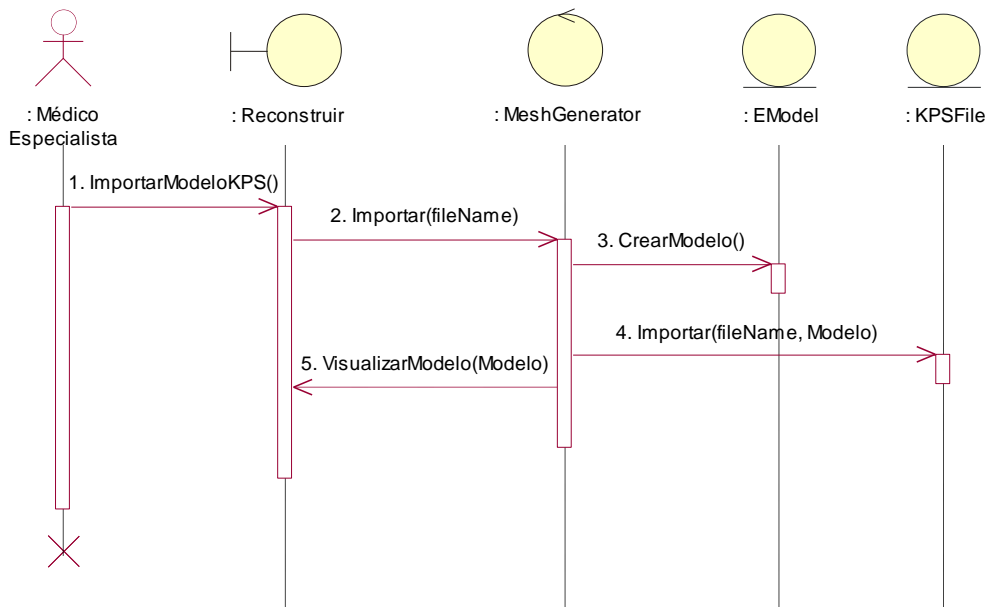
Como en los casos anteriores, al CU Gestionar Modelo 3D también se le realizó el modelo de secuencia, pero por razones de claridad el caso de uso fue separado en las secciones: Generación del Modelo, Importar Fichero, Exportar Fichero, por este motivo a continuación se presentan los diagramas de secuencia de cada una de estas secciones que componen el CU.



**Figura 4.4** Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Generación del Modelo.



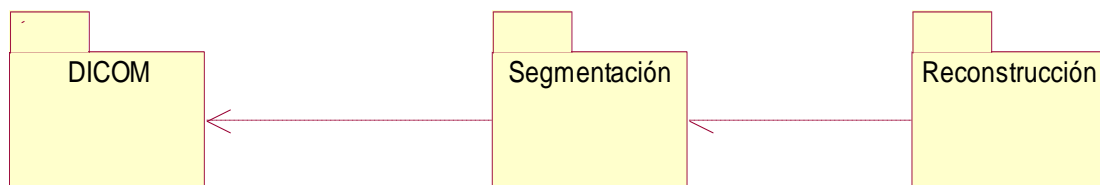
**Figura 4.5** Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Exportar Modelo.



**Figura 4.6** Diagrama de Secuencia del Caso de Uso Gestionar Modelo 3D. Sección Importar Modelo.

### 4.3. Diagrama de Clases del Diseño del Sistema.

El diagrama de clases del diseño fue desglosado por paquetes, para mantener la claridad y la fácil comprensión del mismo. Fueron desarrollados 3 paquetes: el DICOM, Segmentación y el de Reconstrucción, donde el DICOM, encierra toda lo referente a la carga de los ficheros y obtención de las Imágenes, que luego serán procesadas por el paquete de segmentación para obtener los vóxeles y regiones que servirán como punto de entrada al paquete de Reconstrucción para ejecutar sobre ellos el proceso de generación de los modelos y la manipulación de los mismos mediante los ficheros de extensión \*.KPS.



**Figura 4.7** *Diagrama General de Paquetes de Clases del Diseño.*

### 4.3.1. Diagrama de Clases del Paquete DICOM.

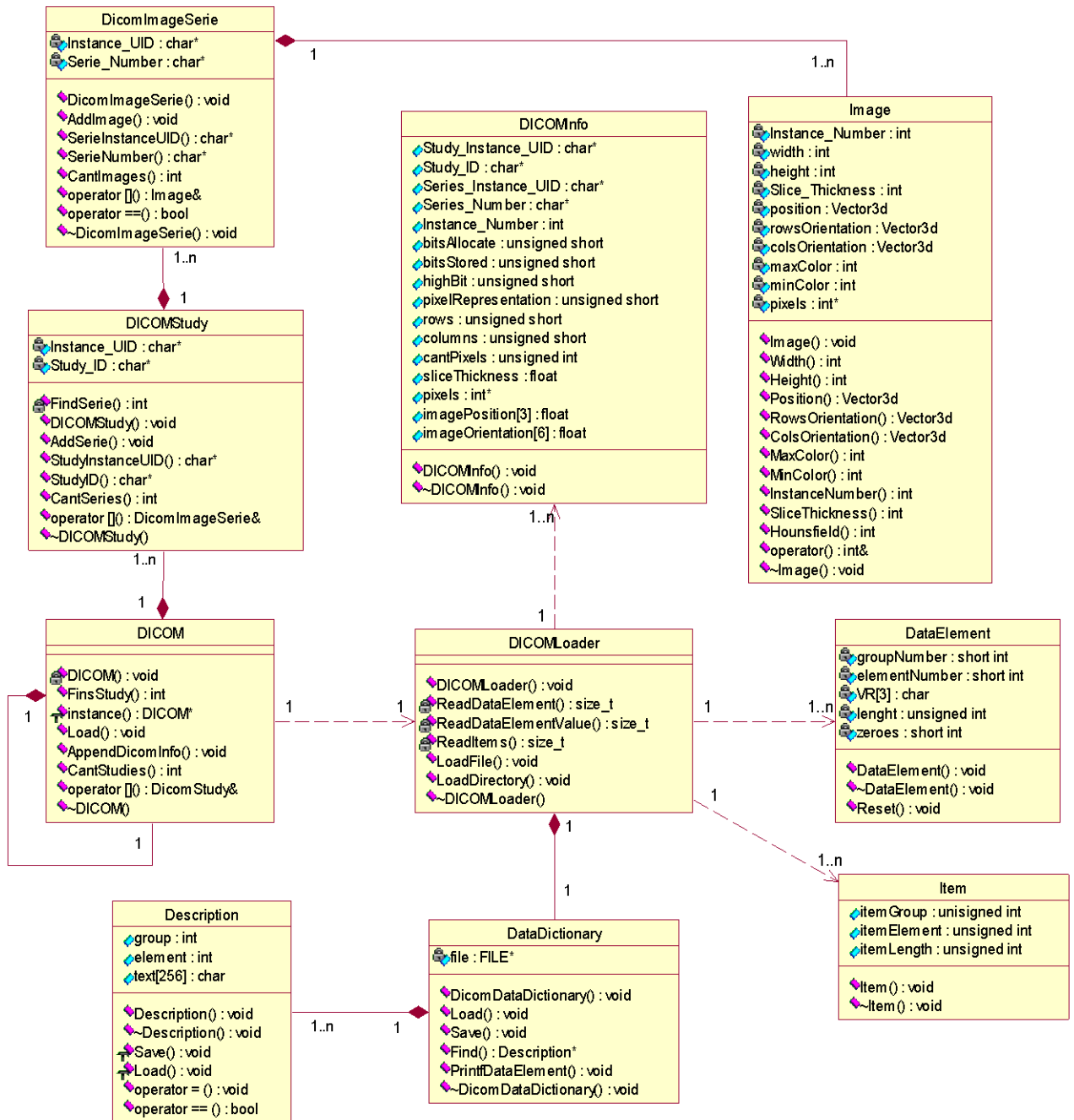


Figura 4.8 Diagrama de Clases de Diseño Paquete de DICOM.



### 4.3.2. Diagrama de Clases del Paquete Segmentación.

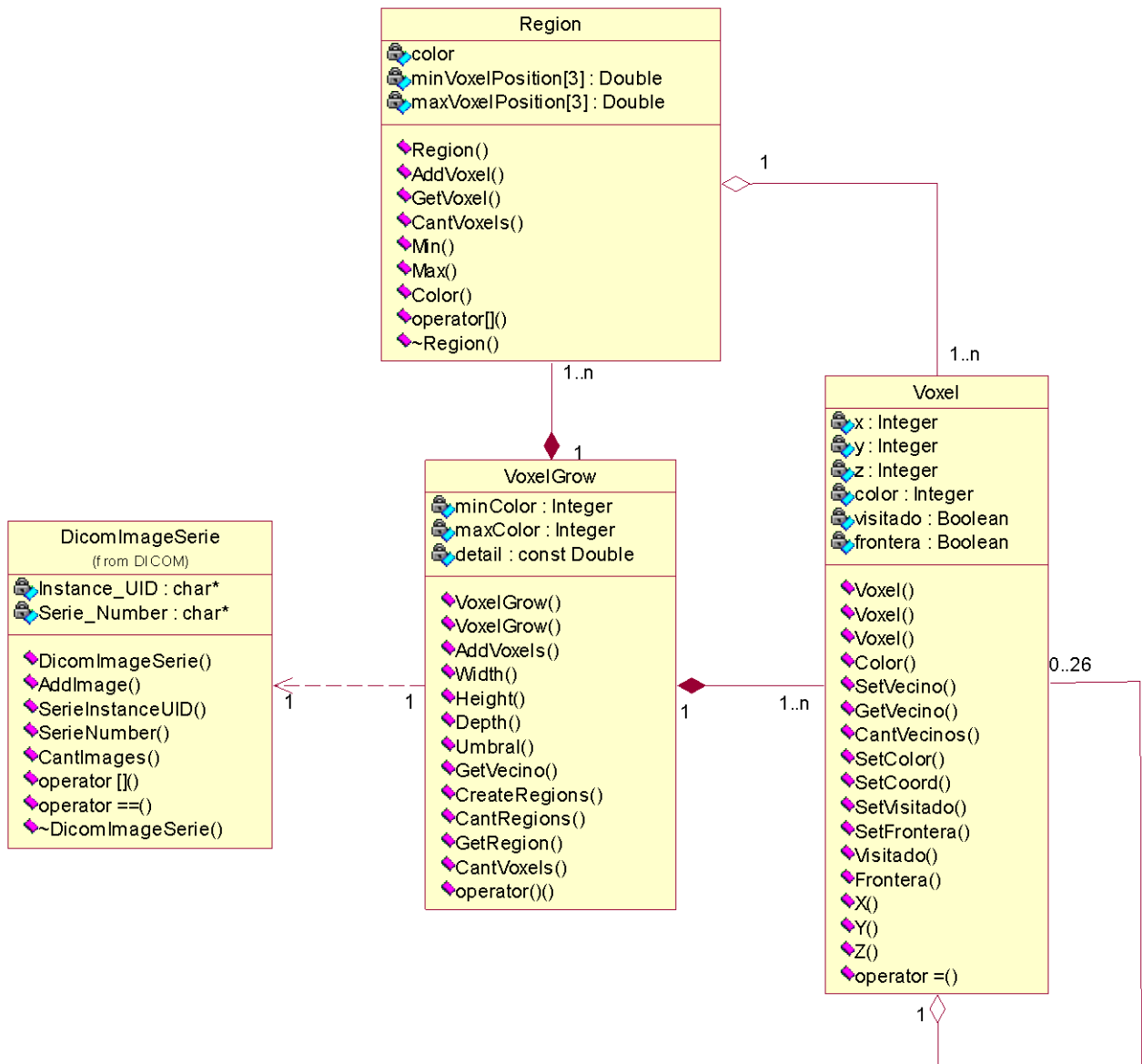


Figura 4.9 Diagrama de Clases de Diseño del Paquete de Segmentación.

### 4.3.3. Diagrama de Clases del Paquete Reconstrucción.

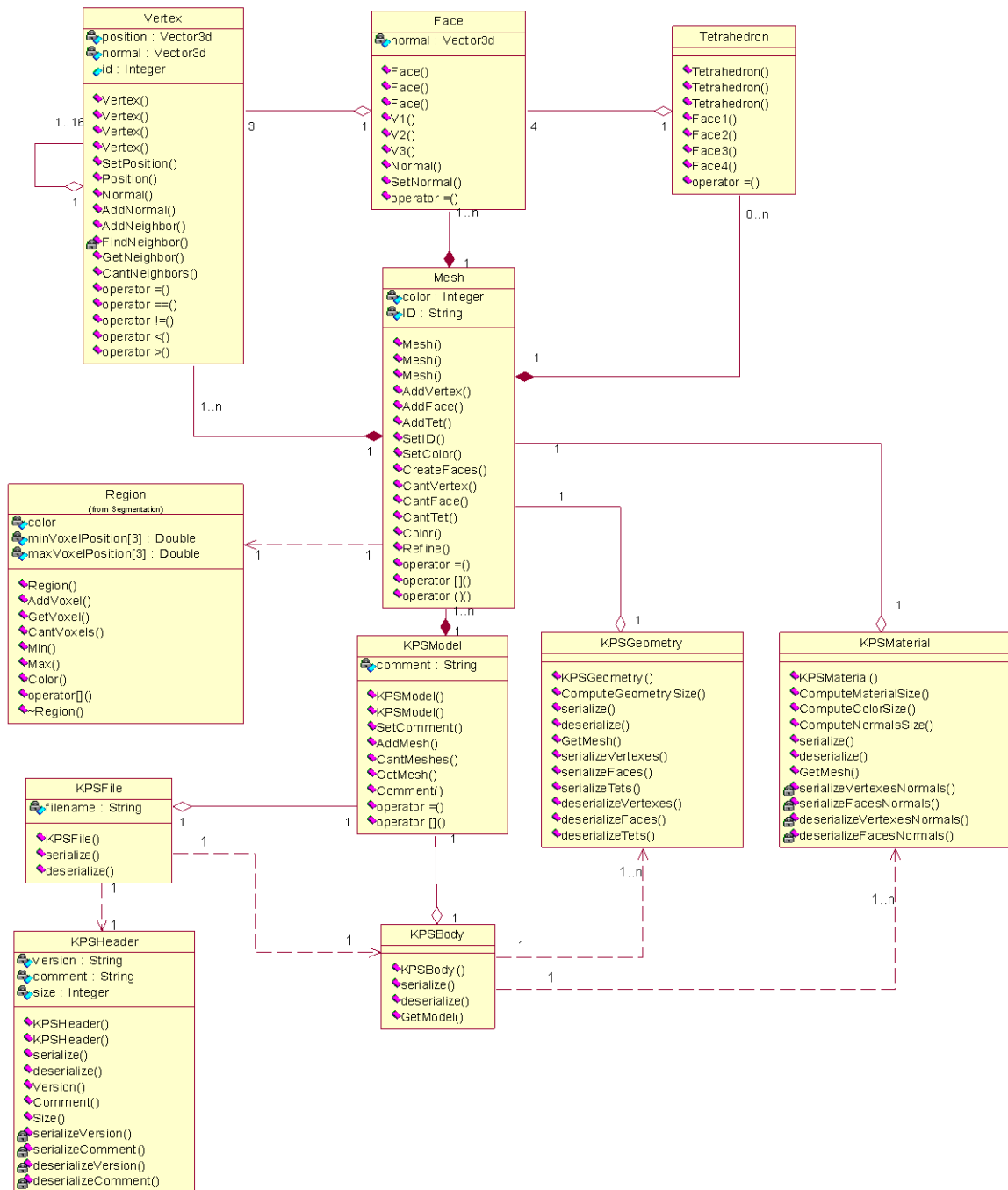


Figura 4.10 Modelo de Clases del Diseño del Paquete de Reconstrucción.

## 4.4. Diagrama de Secuencia del Diseño.

### 4.4.1. Diagrama de Secuencia del Diseño del Caso de Uso Cargar Fichero.

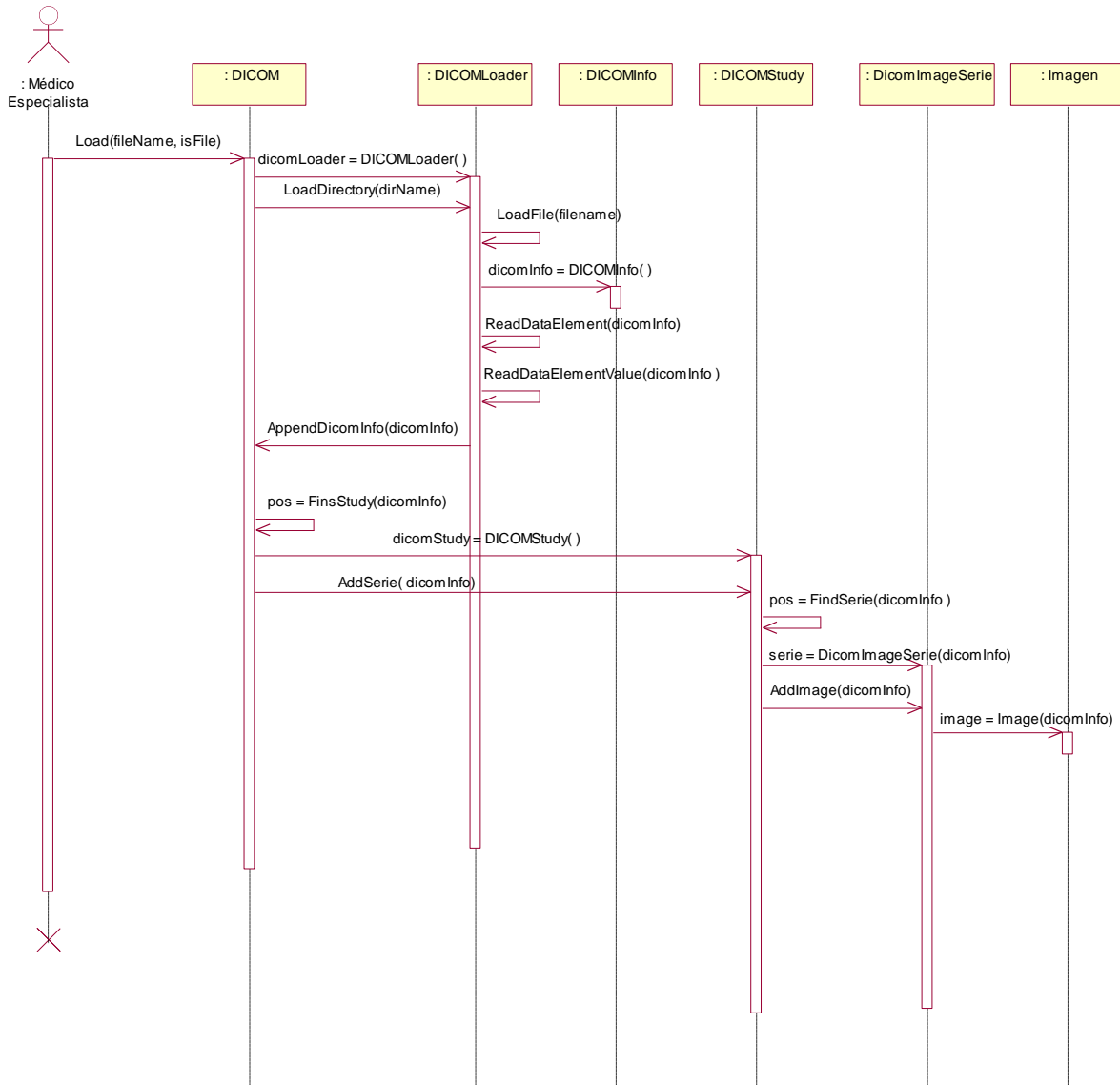


Figura 4.11 Diagrama de Secuencia del Diseño del Caso de Uso Cargar Fichero.

### 4.4.2. Diagrama de Secuencia del Diseño del Caso de Uso Segmentar.

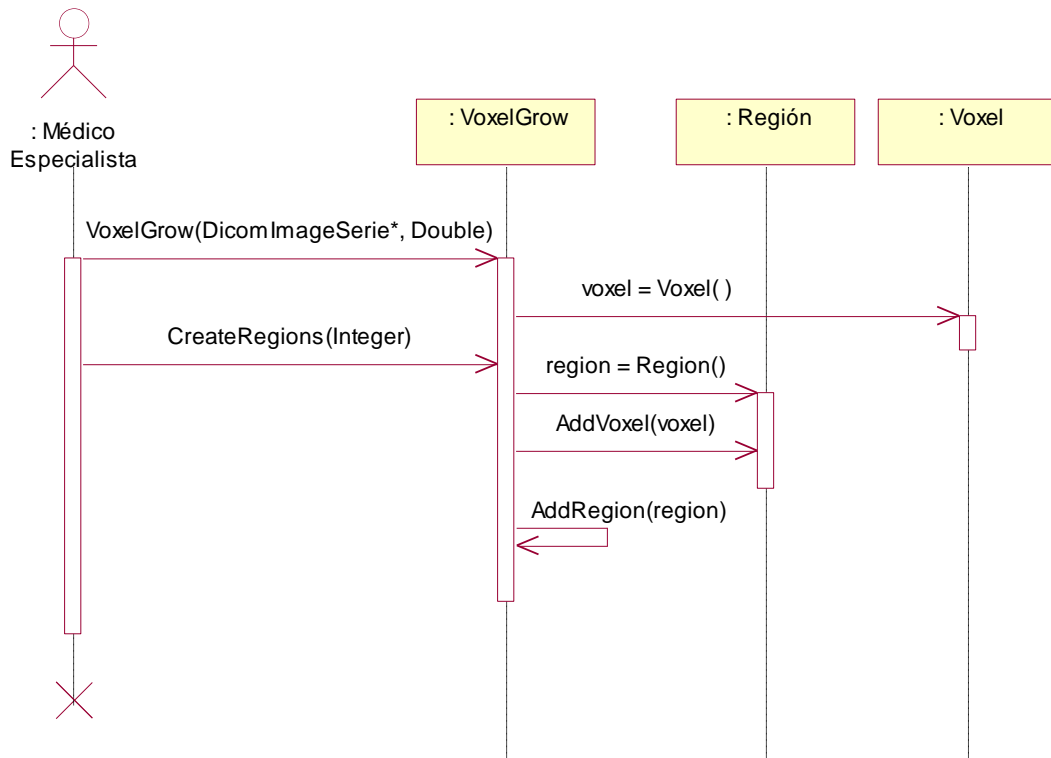
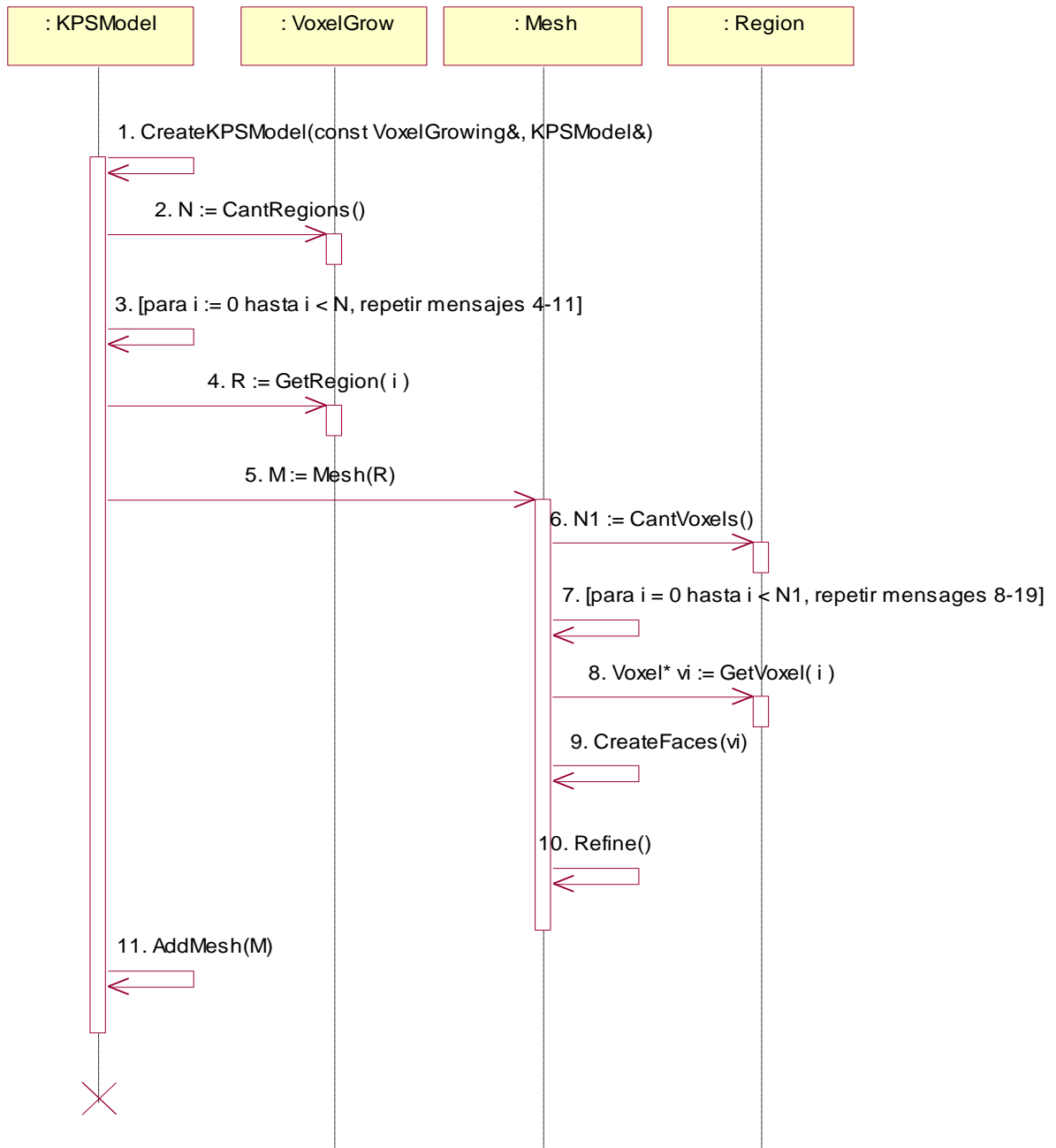
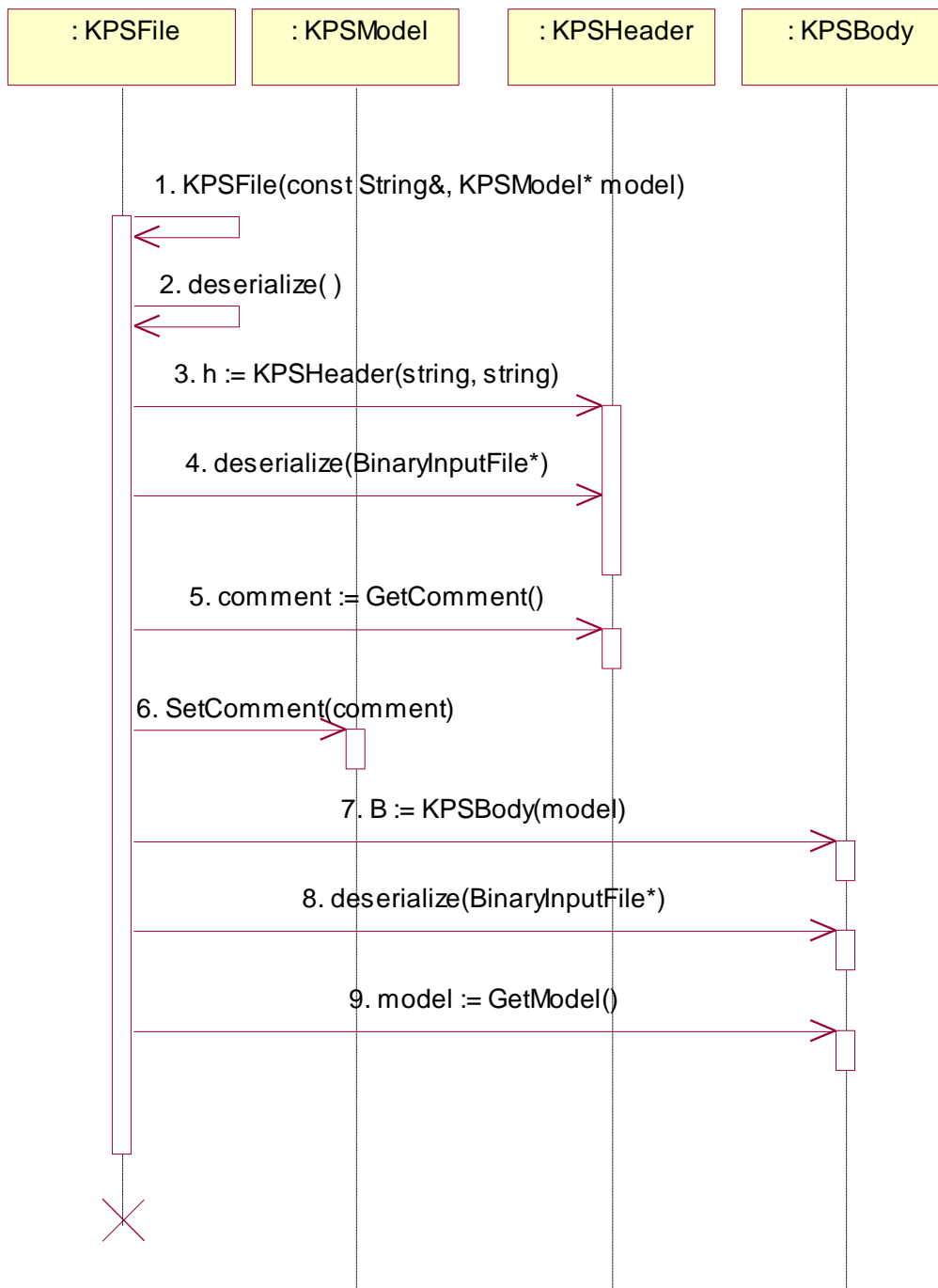


Figura 4.12 Diagrama de Secuencia del Diseño del Caso de Uso Segmentar.

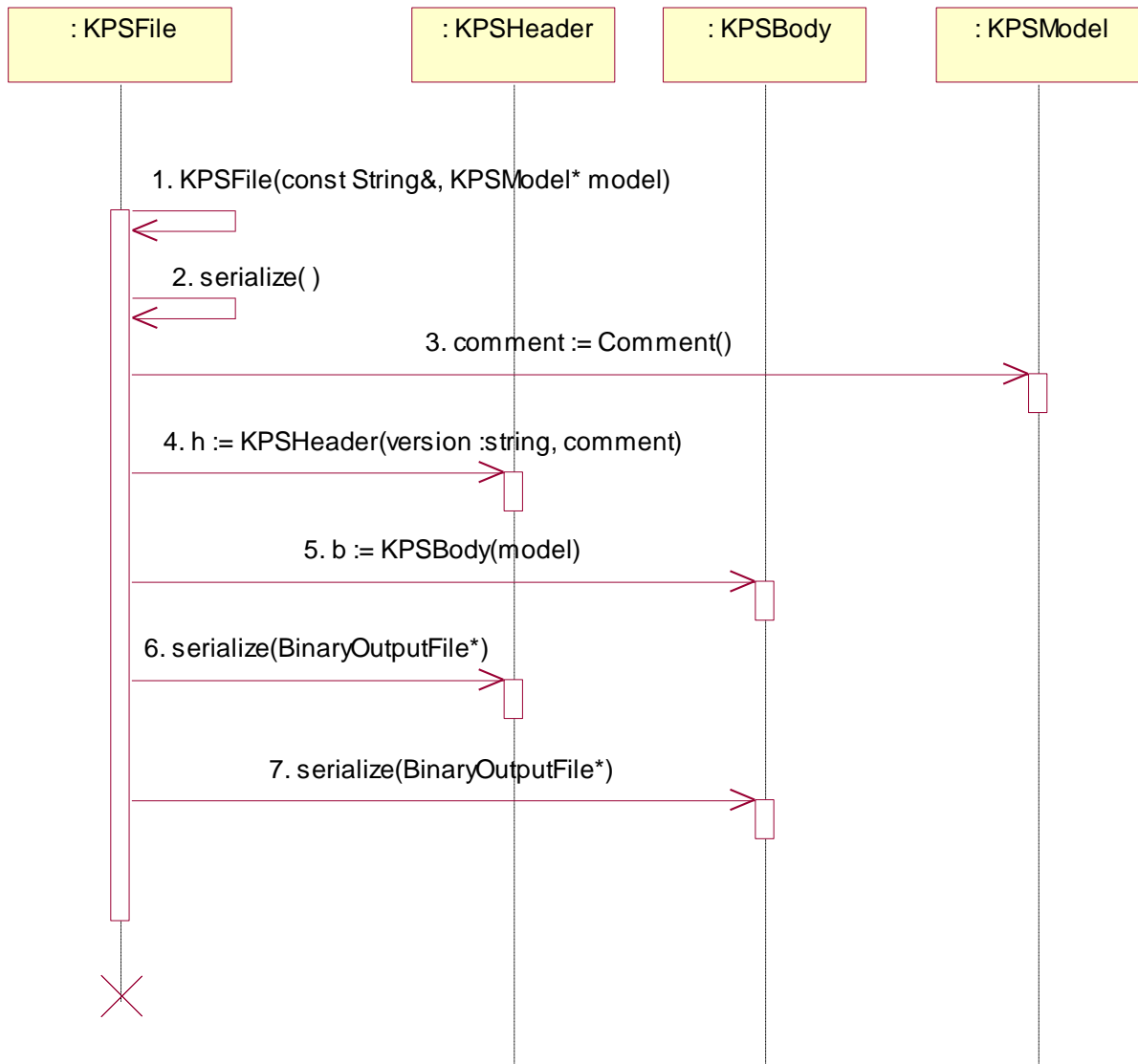
### 4.4.3. Diagrama de Secuencia del Diseño del Caso de Uso Gestionar Modelo 3D.



**Figura 4.13** Diagrama de Secuencia del Diseño del Caso Uso Gestionar Modelo 3D. Sección Generación del Modelo 3D.



**Figura 4.14** Diagrama de Secuencia del Diseño del CU Gestionar Modelo 3D. Sección Importar Modelo.



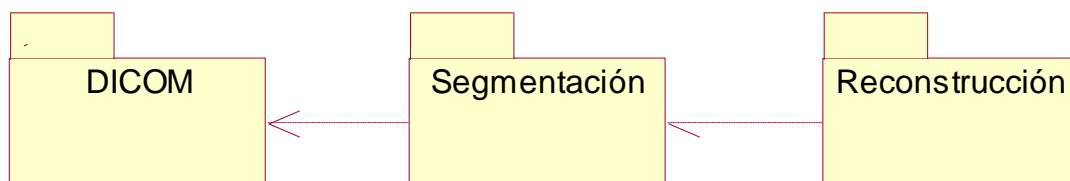
**Figura 4.15** Diagrama de Secuencia del CU Gestionar Modelo 3D. Sección Exportar Modelo.

## Capítulo 5. Implementación y Resultados.

En este capítulo se abordan los temas de la implementación de la aplicación basados en todo el trabajo acumulado a lo largo de los capítulos anteriores, y además se hará un análisis de los resultados del sistema, en cuanto a rendimiento, cumplimiento de los objetivos propuestos, comparaciones entre software similares, entre otras cosas.

### 5.1. Diagrama de Componentes. Paquetes de Implementación.

Las clases resultantes del análisis y el diseño se hacen físicas mediante componentes. A continuación se muestra como se agruparon en paquetes todos los componentes Según las clases que estos contienen.



**Figura 5.1** Paquetes del diagrama de componentes.



### 5.1.1. Componentes del Paquete de Implementación DICOM.

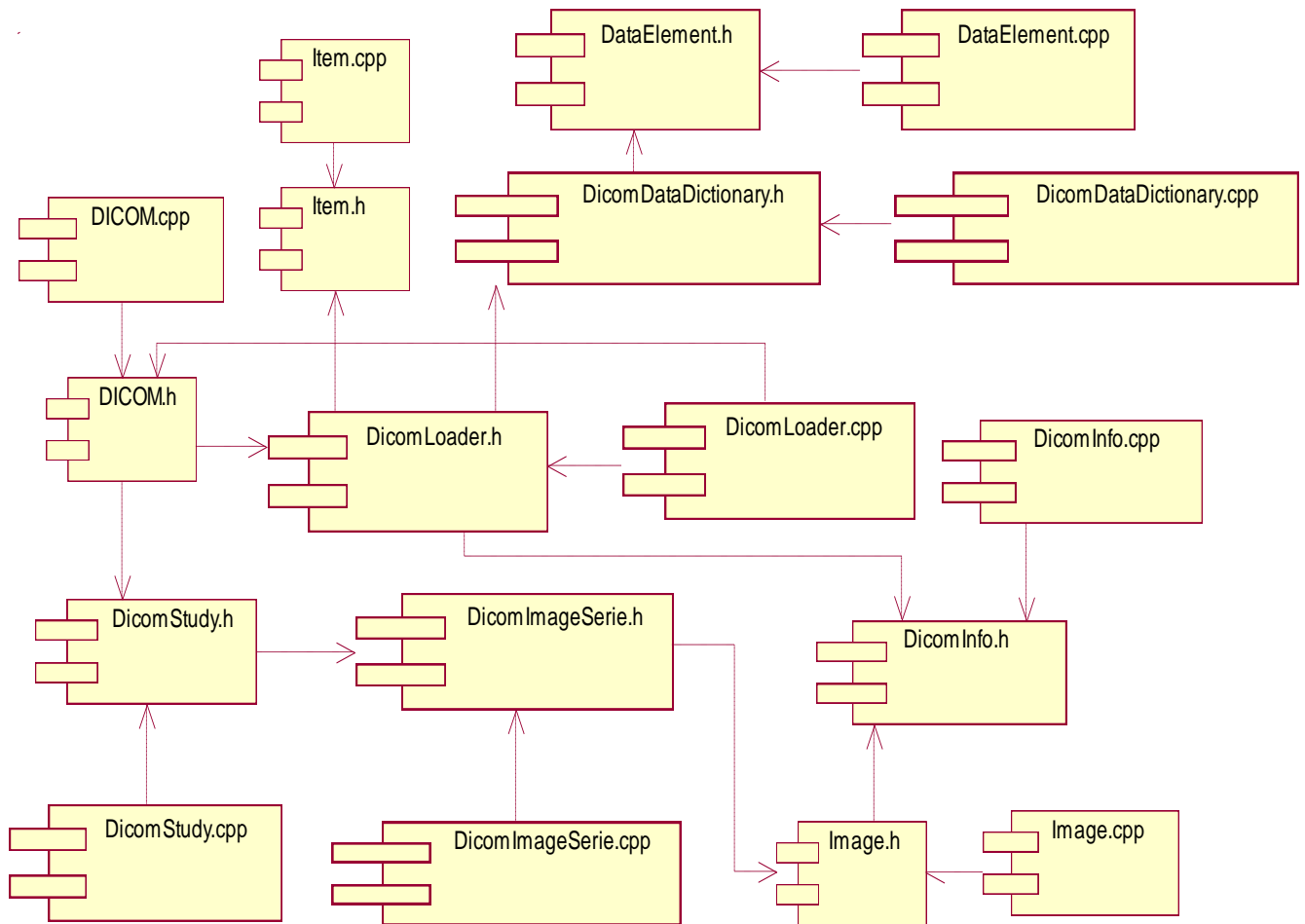


Figura 5.2 Diagrama de componentes. Paquete DICOM.

### 5.1.2. Componentes del Paquete de Implementación Segmentación.

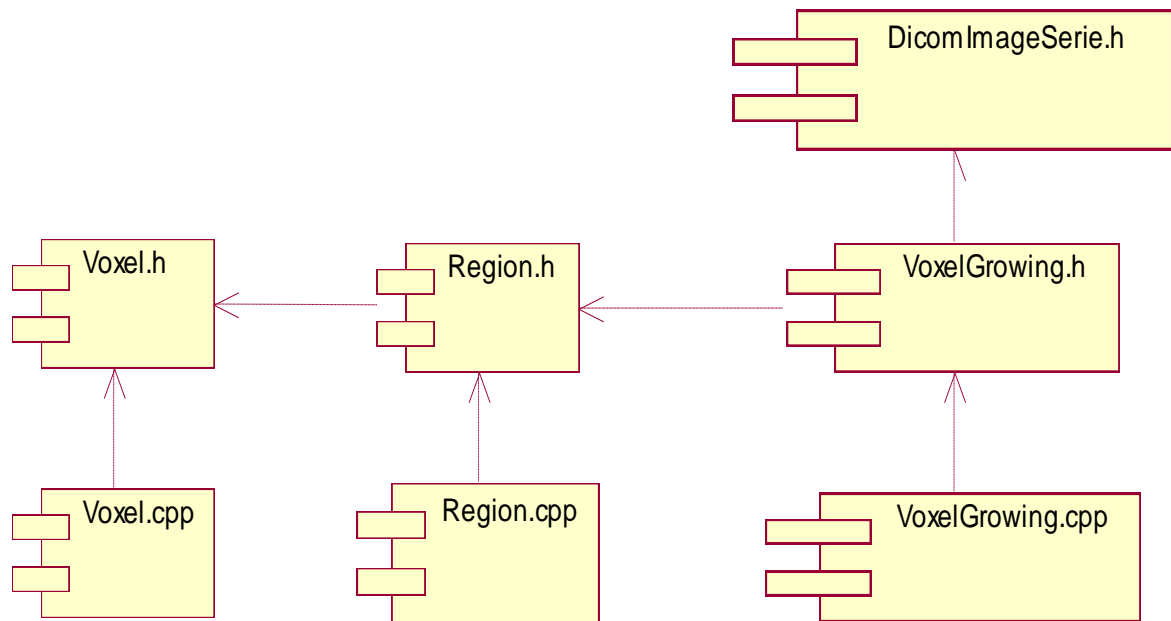


Figura 5.3 Diagrama de Componentes. Paquete Segmentación.

### 5.1.3. Componentes del Paquete de Implementación Reconstrucción.

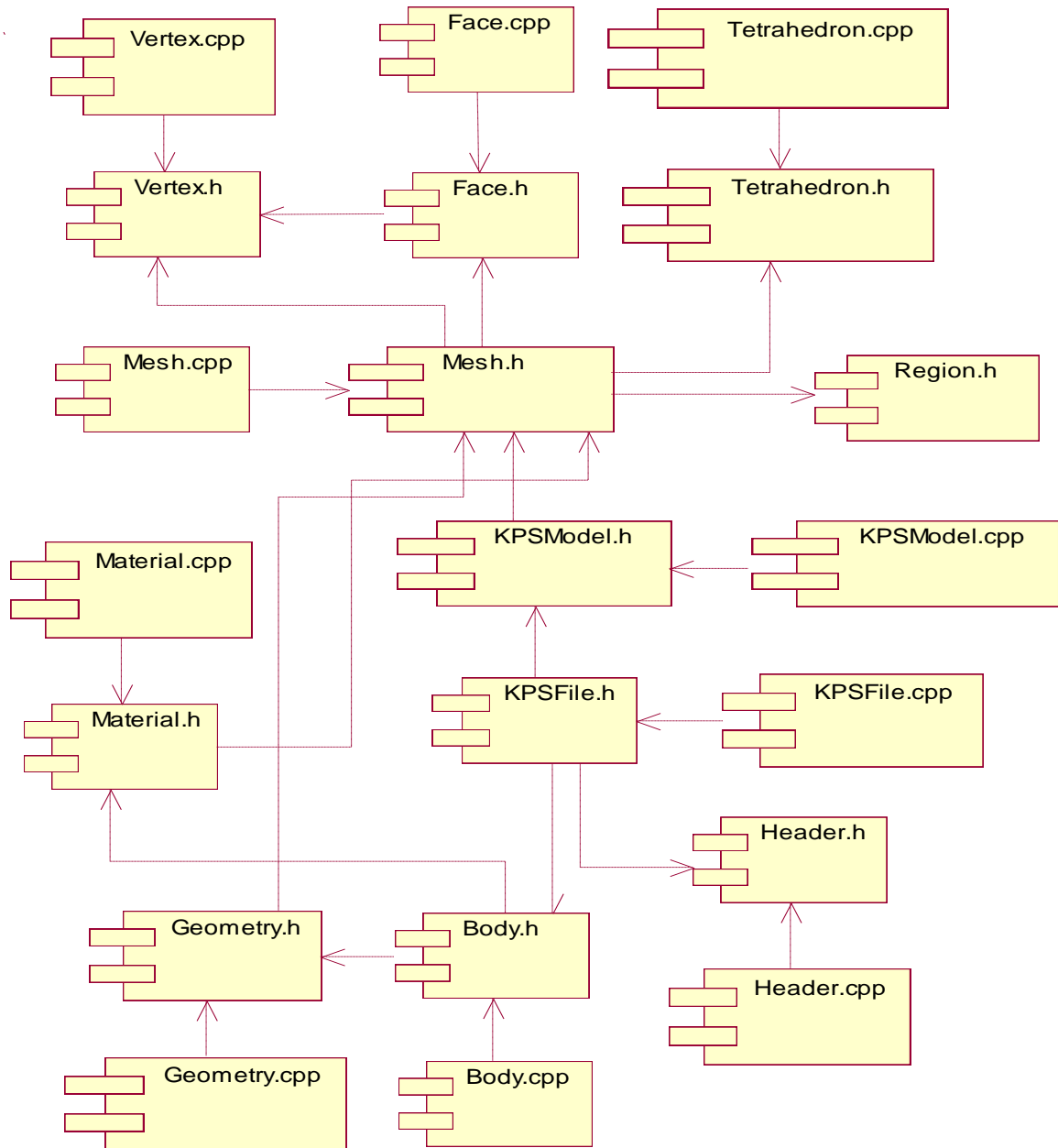


Figura 5.4 Diagrama de Componentes. Paquete Reconstrucción.

## 5.2. Valoración de los Resultados.

En este epígrafe se tiene por objetivo tomar casos de prueba para evaluar el desempeño de los métodos de segmentación, reconstrucción y de suavizado de la malla desarrollados en capítulos anteriores. Los datos de pruebas están clasificados de acuerdo a la modalidad en la que se obtuvo el estudio: MRI o TAC. Las imágenes obtenidas a partir de la modalidad MRI destacan mejor los tejidos blandos (cerebro, tejido conectivo, etc.) mientras que en la modalidad de TAC se pueden observar mejor los tejidos como el hueso, haciéndola más apropiada para extraer tanto el hueso como piel.

### 5.2.1. Datos de entrada.

Para realizar las pruebas, se tomaron como datos de origen las imágenes de los estudios referidos en la **Tabla 5.1**. La tabla está compuesta por cuatro campos de los cuales el primero representa el número o ID del Caso de Estudio, la segunda columna corresponde a la modalidad en que fue realizado el estudio, la próxima es la resolución que esta dada por la calidad del color la cual puede ser de 8 o 16 bits. Por último están las dimensiones del estudio, en un formato de X \* Y \* Z, donde X significa el ancho de la imagen, Y el alto y Z la cantidad de imágenes que componen el estudio.

| Caso | Modalidad | Resolución | Dimensiones(X * Y * Z) |
|------|-----------|------------|------------------------|
| 1    | TAC       | 8 bit      | 64 * 64 * 46           |
| 2    | TAC       | 8 bit      | 64 * 64 * 107          |
| 3    | TAC       | 8 bit      | 64 * 64 * 1024         |
| 4    | MRI       | 8 bit      | 240 * 240 * 23         |

**Tabla 5.1** Datos de Entrada.

### 5.2.2. Parámetros a medir.

Para evaluar los resultados de los algoritmos desarrollados se tendrán en cuenta dos criterios fundamentales para cada uno de los casos de prueba mencionados en la **Tabla 5.1**: uso de memoria RAM y tiempo de procesamiento en segundos.

Estos criterios fueron escogidos teniendo en cuenta que la mayoría de los sistemas de reconstrucción 3D, justifican la demora en tiempo de procesamiento y el alto consumo de recursos computacionales, con los elevados volúmenes de datos que se manipulan en el proceso.

| Caso | T(s). <sup>2</sup><br>Cargar<br>Imágenes | T(S).<br>Segmentar | T(s).<br>Reconstruir | T(s).<br>Generar<br>*.KPS | T(s).<br>Total | Uso<br>memoria<br>(MB) <sup>3</sup> |
|------|------------------------------------------|--------------------|----------------------|---------------------------|----------------|-------------------------------------|
| 1    | 7                                        | 1                  | 1                    | 7                         | 16             | 21.6                                |
| 2    | 18                                       | 3                  | 3                    | 38                        | 62             | 50.2                                |
| 3    | 148                                      | 479                | 418                  | 858                       | 1903           | 480.0                               |
| 4    | 9                                        | 8                  | 3                    | 93                        | 113            | 151.6                               |

**Tabla 5.2** Resultados de los tiempos en segundos y consumo de recursos de hardware en los datos de entrada.

La **Tabla 5.2** evidencia que a pesar de que el proceso de reconstrucción es un gran consumidor de recursos computacionales, la solución propuesta en este trabajo garantiza resultados de alta calidad en un tiempo relativamente corto en hardware con bajas prestaciones.

<sup>2</sup> T(s): Tiempo en Segundos.

<sup>3</sup> MB: Mega Bytes.

## Conclusiones.

Con la realización de este trabajo se logró definir una herramienta que permite la reconstrucción y generación de manera efectiva de un modelo 3D a partir del procesamiento de Imágenes Médicas Digitales como resultado de los procesos de carga del fichero, segmentación de las imágenes, reconstrucción, visualización y exportación del modelo, utilizando para ellos métodos y técnicas eficientes, identificadas y seleccionadas a partir del amplio estudio bibliográfico realizado. Lográndose de esta forma el cumplimiento de los objetivos planteados ya que:

- Se analizó la estructura del fichero DICOM identificándose en el mismo la información relevante referente a la imagen médica, definida a partir de las intensidades de colores, lográndose el diseño e implementación de un módulo eficiente para cargar varios ficheros pertenecientes al estudio realizado a un paciente. El módulo está cimentado en la utilización de matriz cúbica para almacenar y manipular la alta cantidad de información generada.
- Se brinda la posibilidad de seleccionar las sustancias de interés a partir de los valores definidos por la escala de Hounsfield o siguiendo el criterio del experto ya que el módulo implementado brinda la opción de segmentación automática y semiautomática.
- Se utilizó el VoxelGrowing como método de segmentación para determinar las regiones de interés presentes en el volumen de datos. El método arroja buenos resultados teniendo en cuenta el carácter global del proceso, y además es transparente al tipo de segmentación que se realiza. En el análisis de la vecindad se definieron vóxeles con vecindad 26 lo que ha permitido que los algoritmos de segmentación y reconstrucción utilizados obtengan mayor rendimiento de forma más natural, al tiempo que mejoran los resultados y el nivel de realismo.
- En el proceso de reconstrucción se realizó una triangulación inicial de cada una de las regiones detectadas y un post-procesamiento de la malla obtenida. El algoritmo utilizado para la triangulación, es un híbrido de los algoritmos clásicos estudiados el cual ofrece buenos resultados y un rendimiento bastante prometedor, superando la inevitable desventaja de ser dependiente del proceso de segmentación.

- Se definió el formato \*.KPS en función del estudio de las principales estructuras como el 3DS, ASE, para un correcto almacenamiento de los modelos 3D y todas las regiones o estructuras anatómicas que conforman el modelo reconstruido. Este formato tiene una estructura perfectamente extensible a otras necesidades del simulador como por ejemplo: información de propiedades físicas, entre otras.
- Se desarrolló una herramienta con una interfaz de usuario amigable que integra lo anteriormente expuesto. La cual brinda, a partir de cada una de las acciones realizadas, la información visual de los resultados obtenidos; lo que le facilita al médico especialista la interacción con el sistema.

Esta investigación evidenció que el proceso de la reconstrucción 3D es, además de un gran consumidor de recursos de hardware, un proceso complejo que involucra un amplio conocimiento en varias ramas científicas; elementos que fundamentan las futuras investigaciones que darán continuidad a este trabajo de diploma.

## Recomendaciones.

Son muchas las recomendaciones que se pueden hacer a los métodos, técnicas y algoritmos desarrollados durante la investigación realizada en este trabajo de diplomas, alguna de ellas son las siguientes:

1. Extender el intérprete de los ficheros con Sintaxis de Transferencia Nativa (\*.dcm) a que sea capaz de interpretar cualquiera de las sintaxis de transferencias establecidas por el estándar DICOM. Ver [Anexo C.1](#)
2. Investigar sobre los algoritmos de segmentación especializados en definir una estructura anatómica específica dentro de una imagen.
3. Investigar sobre la generación automática de materiales y textura de los modelos reconstruidos a partir de la intensidad expresada en Unidades de Hounsfield de los vóxeles que conforman las regiones.
4. Investigar sobre la obtención de las propiedades físicas de los modelos o estructuras anatómicas a partir de la interpretación de las intensidades expresadas en Unidades de Hounsfield de los vóxeles que conforman las regiones.
5. Investigar cómo eliminar triángulos de la malla sin perder realismo en la misma, con el objetivo de optimizar la visualización de la superficie.
6. Investigar cómo aumentar la velocidad del render de los modelos a partir de la aplicación de directivas básicas de OpenGL como: TRIANGLE\_STRIP o TRIANGLE\_FAN.
7. Investigar cómo reducir el consumo de recursos de hardware utilizando técnicas de Programación Paralela.
8. Investigar cómo aplicando técnicas de multi-resolución de imágenes, se puede consumir menos recursos de hardware en el proceso de obtención de los modelos 3d sin afectar el realismo de los mismos.



---

## Bibliografía.

- [1] *Digital Imaging and Communications in Medicine (DICOM). Standar Dicom Specification*, National Electrical Manufacturers Association, 1999.
- [2] ABREU, M. M. F. D.; M. V. BARRIOS, *et al. Selección automática del nivel óptimo de umbral para segmentación por histograma de imágenes RGB de cortes histológicos de cáncer de mama.*
- [3] ACHARYA, T. and A. K. RAY. *Imagen Processing - Principles and Applications*. New Jersey, Wirley InterScience, 2005. 451 p.
- [4] AMARGÓS, J. D. P. and R. A. MANCHO. *Nuevo método para la reconstrucción de superficies a partir de imágenes tomográficas*, 2004. 25.
- [5] ATTENE, M. and M. SPAGNUOLO. *Automatic surface reconstruction from point sets in space.*
- [6] CALVO, N. A. *GENERACIÓN DE MALLAS TRIDIMENSIONALES POR MÉTODOS DUALES*. FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS, UNIVERSIDAD NACIONAL DEL LITORAL: 118. p.
- [7] CALVO, N. A. and S. R. IDELSOHN. *GENERACIÓN DE MALLAS TRIDIMENSIONALES EN TIEMPO LINEAL*, 2002. 21.
- [8] CISILINO, A. P.; D. D'AMICO, *et al. Construcción de modelos computacionales para el análisis de esfuerzos mecánicos de piezas óseas utilizando imágenes de TC: aplicación a la articulación gleno-humeral.*
- [9] CISILINO, A. P.; G. V. MAZZETTAB, *et al. RECONSTRUCCIÓN GEOMÉTRICA Y DISCRETIZACIÓN CON ELEMENTOS FINITOS DEL COMPLEJO CRÁNEO-MANDIBULAR DEL DINOSAURIO CARNOTAURUS SASTREI*, 2002. 21.
- [10] FELZENSZWALB, P. F. and D. P. HUTTENLOCHER. *Efficient Graph-Based Image Segmentation.*
- [11] ---. *Efficient Graph-Based Image Segmentation.*, 2001. 26.
- [12] FERNÁNDEZ, G.; M. KUNT., *et al. A New Plant Cell Image Segmentation Algorithm\**.

- [13] FERRERO, M. G. *Reconstrucción Tridimensional de la cabeza y el cerebro a partir de IRM.*
- [14] FIORENTINI, S. and I. LARRABIDE. *Segmentación de Imágenes Digitales con Modelos de Superficie.* Facultad de Ciencias Exactas. Buenos Aires, Universidad Nacional del Centro, 2002. 85. p.
- [15] FRAGA, L. G. D. L. *Reconstrucción Tridimensional a Partir de Varias Imágenes,* 2005.
- [16] FRESNO, M. D.; M. MORENO, *et al.* *Segmentación de regiones de interés en imágenes médicas.* Buenos Aires, Universidad Nacional del Centro, 2005. 59. p.
- [17] FRESNO, M. D. and M. J. VÉNERE. *Segmentación de imágenes médicas por crecimiento de regiones con conocimiento adicional.*
- [18] GAO, S. and H.-Q. LU. *A Fast Algorithm for Delaunay based Surface Reconstruction.*
- [19] GARCÍA, O.; A. SUSÍN, *et al.* *Modelo Dinámico para la Segmentación Automática de Imágenes 3D.*
- [20] GARZÓN, J. P. G. and G. MARAÑÓN. *Técnicas Avanzadas de procesamiento de imágenes médicas,* 2007.
- [21] GONZALEZ, R. C. and R. E. WOODS. *Digital Imagen Processing.* New Jersey, Prentice Hall, 2002. 813 p.
- [22] HAMARNEH, G.; H. DELINGETTE, *et al.* *3D segmentation of mouse organs from MR images using deformable simplex mesh models,* 2003.
- [23] HEGE, H.-C.; M. SEEBASS, *et al.* *A Generalized Marching Cubes Algorithm Based On Non-Binary Classifications,* 1997.
- [24] HUANG, J. *Discrete Space, Voxelization and Distance Fields,* 2002.
- [25] J., L. P.; D. M. E., *et al.* *Operadores Morfológicos Multiescala y Distancia Geodésica Aplicados a la Segmentación de Imágenes de Tomografía Axial Computada. IEEE LATIN AMERICA TRANSACTIONS,* 2007. 5.
- [26] JIRÍ, J. *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods.* New York, Taylor & Francis Group, 2006. 725 p.

- [27] KAYYALI, M. S. E. *New methodology to enhance 3D segmentation for CT biomedical volume data images*. p.
- [28] LI, J.; H.-Y. ZHAO, *et al.* *A novel technique of three-dimensional reconstruction segmentation and analysis for sliced images of biological tissues*. *Journal of Zhejiang University SCIENCE B*, 2005.
- [29] LIZANDRA, M. C. J.; C. M. ARANDA, *et al.* *Síntesis de Imágenes en imagen médica*.
- [30] LÓPEZ, X. M. P. *Reconstrucción y Visualización 3D de Estructuras en Oftalmología a partir de Imágenes de Resonancia Magnética en un Entorno de Telemedicina*, 2006.
- [31] LORENSEN, W. E. and H. E. CLINE. *Marching Cubes: A High Resolution 3D Surface Construction Algorithm*.
- [32] MAGEEA, D.; A. BULPITT, *et al.* *Level Set Methods for the 3D Segmentation of CT Images of Abdominal Aortic Aneurysms*.
- [33] MARTÍ, J. M. G. and P. P. C. CALLEJA. *Reconstrucción de Modelos Solidos 3D a partir de Vistas Ortográficas 2D utilizando técnicas de combinacion de cuerpos elementales*.
- [34] MARTÍN, M. *Técnicas Clásicas de Segmentación de Imágen*, 2002.
- [35] MARTÍNEZ, H. S. V. *ALGORITMO DE SEGMENTACIÓN TOPOLÓGICO PARA IMÁGENES ADQUIRIDAS DE LA TOMOGRAFÍA COMPUTADA Y LA RESONANCIA MAGNÉTICA.: REVISTA CUBANA DE INFORMÁTICA MÉDICA*, 2000. 7.
- [36] MOLINA, R. *Introducción al Procesamiento y Análisis de Imágenes Digitales*, 1998.
- [37] MORENO, I. B. *Desarrollo de algoritmos de procesamiento de imágenes con VTK*. *Electronica, Automática e Informática Industrial*, Universidad Politécnica de Madrid, 2004. 164. p.
- [38] MORSE, B. S. and B. Y. UNIVERSITY. *Segmentation (Region Based)*, 2000.
- [39] NAIN, D.; S. HAKER, *et al.* *Shape-Driven 3D Segmentation Using Spherical Wavelets*, 2006.
- [40] NAKAMURA, K. *Manual Registration of 3D Rendered Magnetic Resonance Images by Marching Cube Surface Construction Algorithm*, 2003.

- [41] OIZA, I. N. *Segmentación de Imágenes Bicolor*, 2005.
- [42] ORTUÑO, J. E.; N. MALPICA, *et al.* *Algoritmo Morfológico de Segmentación de Imágenes de Resonancia Magnética Normales Ponderadas en T1*. Madrid, 2001.
- [43] PANTOFARU, C. and M. HEBERT. *A Comparison of Image Segmentation Algorithms*, 2005. 31.
- [44] ---. *A Comparison of Image Segmentation Algorithms*, 2005.
- [45] PAS, G. *Estándar y Protocolo de Imágenes Médicas DICOM*, 1999. 17.
- [46] PASTORE, J. I.; V. L. BALLARIN, *et al.* *Aplicación de Operadores Morfológicos Multiescala y Distancia Geodésica a la Segmentación de Imágenes de Tomografía Axial Computada*.
- [47] RASCOVSKY, M. S. *3d en Tomografía*, 2006.
- [48] ROJASA, J. J. B.; M. L. GUERREROB, *et al.* *Segmentación de imágenes de color*. *REVISTA MEXICANA DE FISICA*, 2004.
- [49] SABALISCK, N. S. P. *Reconstrucción y clasificación de estructuras anatómicas a partir de secuencias de imágenes ecográficas*.
- [50] SÁNCHEZ, D. A. S. and O. G. PANYELLA. *Segmentación Automática del Ventrículo Izquierdo del Corazón Mediante un Modelo Dinámico*.
- [51] SANTAMARÍA, J.; O. CORDÓN, *et al.* *Reconstrucción de modelos 3D en antropología forense mediante Scatter Search*.
- [52] SHEWCHUK, J. R. *General-Dimensional Constrained Delaunay and Constrained Regular Triangulations*, 2007.
- [53] SI, H. *A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator. TetGen User Manual*, 2006. 1.4.
- [54] YU, W.; J. FRITTS, *et al.* *A HIERARCHICAL IMAGE SEGMENTATION ALGORITHM*, 2005. 4.
- [55] ZHANG, H. *Mesh Generation for Voxel-Based Objects*, 2005.

- [56] Brandoli, Paolo. *Puntoexe Software a C++ Open Source DICOM Library*. [Online] *Puntoexe software*. <http://puntoexe.com/content/view/11/2/>.
- [57] DicomLib. [Online] *Sunnybrook & women's College Health Sciences Centre*. <http://dicomlib.swri.ca/dicomlib.html>.
- [58] Visualization ToolKit. [Online] *Kitware, Inc.* <http://public.kitware.com/VTK/>.
- [59] NLM Insight Segmentation & Registration Toolkit. [Online] <http://www.itk.org/>.
- [60] TetGen. A Quality Tetrahedral mesh Generator and Three-Dimensional Delaunay Triangulator. [Online] *Weierstrass Institute for Applied Analysis and Stochastics, April 16, 2007*. <http://puntoexe.com/content/view/11/2/>.

## Glosario.

### B

**Biblioteca:** Ver librería

### C

**Cara:** Cada una de las superficies que forman o limitan un poliedro.

**Cara Frontera:** Cara de un Vóxel que no es compartida con otro Vóxel, cara que esta en el contorno de la región.

**Cirugía de Mínimo Acceso:** La cirugía laparoscópica, sin ingreso o mínimamente invasiva es una técnica quirúrgica practicada a través de pequeñas incisiones, asistida de una cámara de video que permite cirujano accionar sobre el campo quirúrgico, evitando los grandes cortes de bisturí requeridos por la cirugía abierta o convencional y posibilita un periodo post-operatorio mucho más rápido y confortable.

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

### D

**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.

### E

**Estudio:** Conjunto de ficheros DICOM que almacenan el resultado de un análisis realizado a algún paciente.

### F

**Face:** Abstracción de un triángulo.

**Fichero DICOM:** Es el fichero exportado por los equipos de TAC, MRI, entre otros. Contiene datos relacionados con el medico, el paciente, las imágenes, y además información para la transmisión por la red de los datos.

**Fichero KPS:** Es el fichero diseñado por los autores del presente trabajo para almacenar los modelos 3D generados en el proceso de reconstrucción, y puedan ser utilizados por terceros.

**Frontera:** Sinónimo de borde, límite.

## H

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

## I

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

**Imagen DICOM:** Imagen obtenida a partir de un fichero DICOM.

**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.

**Interpolación:** algoritmo matemático que a partir de varios puntos en el espacio, describe una función que contiene a los puntos intermedios.

## L

**Librerías:** En Inglés *library*, cuando se habla de ordenadores, refiere al conjunto de rutinas que realizan las operaciones usualmente requeridas por los programas. Las librerías pueden ser compartidas, lo que quiere decir que las rutinas de la librería residen en un fichero distinto de los programas que las utilizan. Los programas enlazados con bibliotecas compartidas no funcionarán a menos que se instalen las bibliotecas o librerías necesarias.

**M**

**Malla:** Forma de representar un modelo a partir de polígonos. Colección de vértices, aristas y polígonos conectados de forma que cada arista es compartida como máximo por dos polígonos.

**Material:** Combinación de luces y colores usados para definir una apariencia.

**Modelo 3D:** Es la unión de un conjunto de mallas, es como tal el resultado de todo el proceso de reconstrucción en la memoria. Para convertir este resultado en algo observable por terceros, es preciso convertirlo en un Fichero KPS.

**Multiplataforma:** término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

**N**

**Normal:** Ver Vector Normal.

**P**

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

**Piecewise Linear Complex (PLC):** Es el conjunto de vértices, segmentos y caras, donde cada cara es una región poligonal (PSLG), que puede tener cualquier cantidad de lados y puede ser no convexa, posiblemente con huecos, segmentos y vértices dentro de ella. El PLC es lo que se almacena en el fichero \*.poly de tetgen su dato de entrada más importante.

**Planar Straight Line Graph (PSLG):** Es un término usado en Geometría computacional para un grafo planar de líneas rectas. En geometría computacional a los PSLG se les denomina comúnmente: subdivisiones planares.

**Poly:** Extensión de uno de los archivos de entrada y salida de datos de la librería TetGen.

**Polígono:** Porción de plano limitada por líneas rectas.



**R**

**Rayos X:** Radiación ionizante electromagnética emitida por un tubo al vacío, resultante de la excitación de los electrones orbitales internos por el bombardeo del ánodo de destino con un haz de electrones proveniente de un cátodo calentado.

Radiación ionizante electromagnética producida por la excitación de los electrones orbitales internos de un átomo por otros procesos, como el retardo nuclear y sus secuelas.

**Radiografía:** Examen de cualquier parte del cuerpo para propósitos de diagnóstico mediante rayos X; el resultado normalmente se registra en una película fotográfica.

**Realidad Virtual:** Simulación generada por computadora de imágenes o ambientes tridimensionales interactivos con cierto grado de realismo físico o visual.

**Reconstrucción:** Es el proceso de generar el modelo 3D a partir de imágenes medicas digitales (Carga + Segmentación + Reconstrucción). También se le denomina reconstrucción a la etapa de generar la malla resultante de una segmentación previa, dentro del proceso descrito anteriormente.

**Región:** Es una estructura anatómica del cuerpo que ha sido reconocida. Es la agrupación de varios vóxeles con similares propiedades de color y que además cumplan con los criterios de vecindad (distancia entre ellos  $\leq 3$ ), la combinación de una o varias regiones es lo que resulta del proceso de segmentación.

**Render:** Proceso de obtención de imágenes por computadora.

**Resonancia Magnética:** Modalidad de diagnóstico radiológico que utiliza tecnología de resonancia magnética nuclear en la que los núcleos magnéticos (especialmente los protones) del paciente se alinean en un campo magnético potente y uniforme, absorben energía de impulsos afinados de radiofrecuencia y emiten señales de radiofrecuencia a medida que decae su excitación. Estas señales, que varían en intensidad según la abundancia nuclear y el entorno químico molecular, se convierten en imágenes tomográficas (cortes seleccionados) mediante el uso de gradientes de campo en el campo magnético, lo que permite la localización tridimensional de las fuentes de las señales.

**S**

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

**Serie de Imágenes:** Conjunto de imágenes dentro de un estudio que representan un corte determinado: Axial, Coronal o Longitudinal.

**Simulador Quirúrgico:** Sistema de Realidad Virtual especializado en imitar o simular procesos quirúrgicos reales, con vistas al entrenamiento de especialistas en la medicina.

**Sistemas/Sistemas de Realidad Virtual:** Sistema informático interactivo que ofrece una percepción sensorial al usuario de un mundo tridimensional sintético que suplanta al real.

## T

**Tetraedro:** Sólido terminado por cuatro planos o caras. En términos de computación es una estructura volumétrica, formada por 4 triángulos.

**Tomografía Axial Computarizada (TAC):** Es un examen médico no invasivo ni doloroso que ayuda al médico a diagnosticar y tratar enfermedades. Las imágenes por TAC utilizan un equipo de rayos X especial para producir múltiples imágenes o visualizaciones del interior del cuerpo, a la vez que utiliza conjuntamente una computadora que permite obtener imágenes transversales del área en estudio. Luego, las imágenes pueden imprimirse o examinarse en un monitor de computadora.

**Triángulo:** Polígono de tres lados. Se representa en este trabajo por 3 vértices.

## V

**Vector Normal:** Vector cuyos puntos están en dirección perpendicular a una superficie.

**Vértice:** Punto donde concurren tres o más planos.

**Vértice Vecino:** Un vértice A es vecino de un vértice B si ambos pertenecen a un mismo triángulo.

**Vóxel o Voxel:** (la palabra proviene de la contracción del término en inglés "volumetric píxel") es la unidad cúbica que compone un objeto tridimensional. Constituye la unidad mínima procesable de una matriz tridimensional y es, por tanto, el equivalente del píxel (o pixel) en un objeto 2D.

**Vóxel Frontera:** Vóxel que se encuentra en la frontera de la región a que pertenece, esto sucede si el Vóxel tiene al menos una cara frontera.

**Voxel Growing:** Algoritmo de segmentación basado en técnicas de segmentación orientadas a regiones que trabaja directamente sobre los vóxeles de un volumen de datos.

**Vóxel Vecino:** Un Vóxel A es vecino de otro Vóxel B, si ambos poseen propiedades similares (pertenecen a la misma sustancia) y además tienen conectividad 6, 18, o 26, esto es: que comparten al menos una cara, una arista o un vértice respectivamente.

## Anexos.

### Anexo A: Segmentación de Imágenes.

#### A.1 Detección de Puntos.

La detección de puntos aislados es inmediata. Empleando la máscara de la **Figura A.1**

|    |    |    |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8  | -1 |
| -1 | -1 | -1 |

**Figura A.1** *Máscara para la detección de puntos aislados.*

Vamos a decir que se ha detectado un punto en la posición en la cual está centrada la máscara si

$$|R| > T$$

Donde **T** es un umbral. Básicamente se mide la diferencia entre el píxel central y sus vecinos, puesto que un píxel será un punto aislado siempre que sea suficientemente distinto de sus vecinos. Solamente se considerarán puntos aislados aquellos cuya diferencia con respecto a sus vecinos sea significativa.

#### A.2 Detección de Líneas.

En este caso se consideran las máscaras de la **Figura A.2**.

|    |    |    |
|----|----|----|
| -1 | -1 | -1 |
| 2  | 2  | 2  |
| -1 | -1 | -1 |

Horizontal

|    |    |    |
|----|----|----|
| -1 | -1 | 2  |
| -1 | 2  | -1 |
| 2  | -1 | -1 |

45°

|    |   |    |
|----|---|----|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

Vertical

|    |    |    |
|----|----|----|
| 2  | -1 | -1 |
| -1 | 2  | -1 |
| -1 | -1 | 2  |

-45°

**Figura A.2** Máscaras para la detección de líneas.

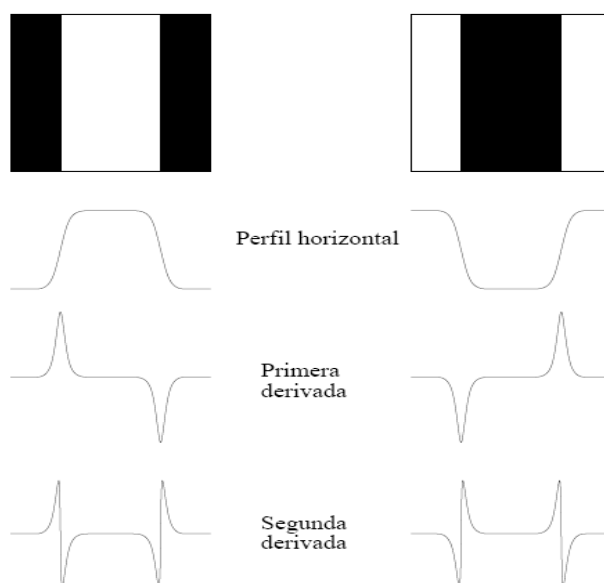
Si pasamos la primera de las máscaras a lo largo de la imagen, tendrá mayor respuesta para líneas de ancho un pixel orientadas horizontalmente. Siempre que el fondo sea uniforme, la respuesta será máxima cuando la línea pase a lo largo de la segunda fila de la máscara. La segunda máscara de la **Figura A.2** responderá mejor a líneas orientadas a 45°; la tercera máscara a líneas verticales; y la última a líneas orientadas a -45°. Estas direcciones se pueden establecerse observando que para la dirección de interés las máscaras presentan valores mayores que para otras posibles direcciones. Si denotamos con  $R_1$ ,  $R_2$ ,  $R_3$ , y  $R_4$  las respuestas de las cuatro máscaras de la **Figura A.2** para un píxel en particular, entonces si se cumple que  $|R_i| > |R_j|$   $i \neq j$ , será más probable que dicho píxel esté asociado a la dirección correspondiente a la máscara  $i$ .

### A.3 Detección de Bordes.

Los bordes corresponden a cambios locales significativos en una imagen y son, probablemente, los rasgos más significativos para el análisis de la misma.

La detección de bordes es el procedimiento empleado más habitualmente para la detección de discontinuidades. Un borde se define como la frontera entre dos regiones con nivel de gris relativamente diferente.

La idea básica detrás de cualquier detector de bordes es el cálculo de un operador local de derivación. En la **Figura A.2** se puede ver este concepto. En la parte derecha se puede ver una imagen de una banda clara sobre un fondo oscuro, el perfil a lo largo de una línea horizontal y la primera y segunda derivada de dicho perfil. Se puede observar que el perfil del borde se ha modelado como una discontinuidad suave. Esto tiene en cuenta el hecho de que en las imágenes los bordes están ligeramente desenfocados, debido en parte al proceso de digitalización.



**Figura A.3** *Detección de bordes empleando operadores de derivación, la segunda derivada tiene un cruce por cero en la posición de cada borde.*

Como se puede observar en la **Figura A.3** la primera derivada es positiva para cambio a nivel de gris más claro, negativa en caso contrario y cero en aquellas zonas con nivel de gris uniforme. La segunda derivada presenta valor positivo en la zona oscura de cada borde, valor negativo en la zona clara de cada borde y valor cero en las zonas de valor de gris constante y justo en la posición de los bordes. El valor de la magnitud de la primera derivada nos sirve para detectar la presencia de bordes, mientras que el signo de la segunda derivada nos indica si el pixel pertenece a la zona clara o a la zona oscura. Además la segunda derivada presenta siempre un cruce por cero en el punto medio de la transición.

El conjunto de bordes que produce un detector de fronteras puede dividirse en dos clases: los correctos y los falsos. Un tercer conjunto corresponde a los bordes que deberían haber sido detectados. Este tipo de bordes reciben el nombre de **falsos negativos** mientras que los falsos corresponden a los **falsos positivos**.

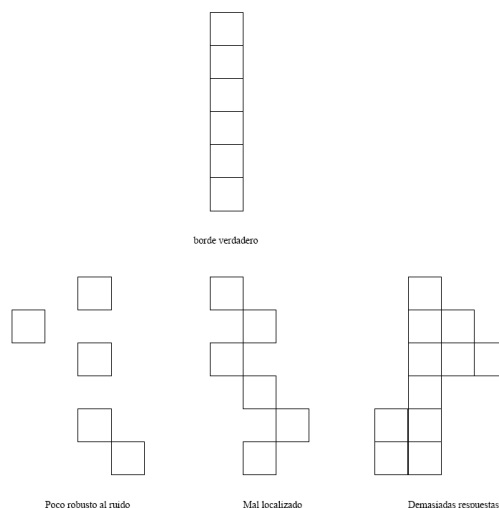
### **Pasos Para la detección de bordes.**

Los algoritmos para la detección de bordes tienen cuatro pasos:

- 1. Filtrado.** Puesto que el cálculo del gradiente basado en los valores de intensidad de un número finito de puntos es muy susceptible al ruido, normalmente se filtra la imagen inicialmente para mejorar el rendimiento del detector respecto al ruido. Es obvio que este proceso debe aplicarse con cuidado, puesto que el emborronamiento oculta los bordes.
- 2. Realce.** Este proceso es esencial para determinar cambios de intensidad de los vecinos de punto. El proceso es básicamente el cálculo del gradiente.
- 3. Detección.** Muchos puntos en una imagen tienen un gradiente no nulo. Obviamente no todos ellos son bordes para todas las aplicaciones. Es necesario un método para determinar qué puntos son bordes. En criterio ampliamente usado es la umbralización.
- 4. Estimación.** Esta debería llevarse a cabo en la resolución apropiada y también en algunos casos será necesario también estimar la dirección del gradiente.

### **Medida de Calidad de un detector de Bordes.**

La idea de medir la calidad de un detector de bordes es bastante antigua y Abdou y Pratt lo han usado para definir el mérito de cada detector. La idea no es usar el mérito para definir el detector sino usar el mérito para usar el detector. La **Figura A.4** muestra unos cuantos defectos que puede tener un detector.



**Figura A.4** Defectos de los detectores de bordes.

El mérito definido por Abdou y Pratt es:

$$F = \frac{1}{\max(I_i, I_A)} \sum_{I=1}^{I_A} \frac{1}{1 + \alpha d^2(i)}$$

Donde  $I_i$  es el número ideal de puntos de bordes,  $I_a$  es el número de puntos de bordes detectados,  $d(i)$  es la menor distancia del borde  $i$ -ésimo al borde ideal y  $\alpha$  es una constante positiva.  $F$  es siempre menor o igual que uno y alcanza la igualdad cuando  $I_i = I_a$  y  $d(i) = 0$  para todo  $i$ .

### Enlazado de Bordes. Procesado Local y Global.

Las técnicas de detección de discontinuidades idealmente, proporcionarán los píxeles correspondientes a los contornos o fronteras entre las regiones de la imagen. En la práctica este conjunto de píxeles no suele caracterizar completamente a esos contornos debido a la presencia de ruido, ruptura en los propios contornos debido a la iluminación no uniforme y otros efectos que introducen ruidos en las discontinuidades de la intensidad. En general, después de los procedimientos de detección de bordes se suele emplear técnicas de enlazado u otras técnicas de detección de contornos designadas para unir los píxeles de bordes en contornos significativos.



**Procesado Local.**

Uno de los procedimientos más simples para enlazar puntos de bordes es analizar las características de los píxeles en un vecindario pequeño alrededor de cada punto  $(x, y)$  donde se ha detectado la presencia de un borde. Todos los puntos que son similares se enlazan, dando lugar a un contorno o frontera de píxeles que comparten ciertas propiedades en común. Las dos propiedades que se suelen utilizar son la magnitud del operador de gradiente que ha dado lugar al píxel de borde y la dirección del gradiente en ese punto.

Un píxel de borde con coordenadas  $(x', y')$  en el vecindario predefinido para el píxel  $(x, y)$  va a ser similar en magnitud al píxel  $(x, y)$  si:

$$|\nabla f(x, y) - \nabla f(x', y')| \leq T,$$

Donde  $T$  es cierto umbral de amplitud.

Un píxel de borde con coordenadas  $(x', y')$  en el vecindario predefinido para el píxel  $(x, y)$  va a ser similar en ángulo al píxel  $(x, y)$  si:

$$|\alpha(x, y) - \alpha(x', y')| \leq A,$$

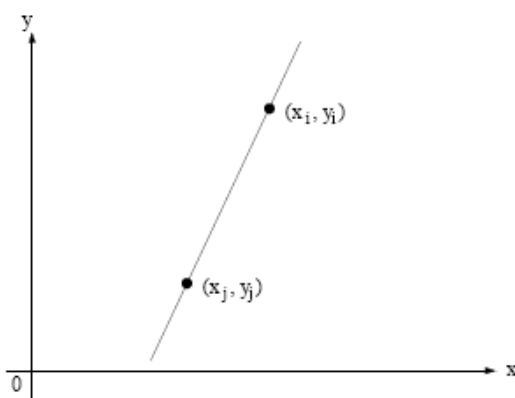
Donde  $A$  es un cierto umbral angular. Hay que tener en cuenta que la dirección del borde en el punto  $(x, y)$  es perpendicular a la dirección del gradiente en ese punto. Sin embargo, de cara a comparar las direcciones, la ecuación es equivalente.

Se enlazará un punto, en el vecindario predefinido para  $(x, y)$ , con el píxel  $(x, y)$  si se cumplen simultáneamente las condiciones de magnitud y de ángulo. Este procedimiento se repetirá para todos los píxeles de borde de la imagen. Tras encontrar un píxel de borde que enlaza con el  $(x, y)$  se procede a probar con este nuevo píxel de borde, hasta que no se puedan enlazar más píxeles de borde. Todos estos píxeles enlazados se etiquetan con cierto valor de gris y se procederá a enlazar los restantes píxeles de borde con otra etiqueta. Al final habrá tantas etiquetas como conjuntos de píxeles enlazados, y que corresponderán a los contornos o fronteras correspondientes a todos los objetos detectados de la imagen.

## Procesado Global.

En esta sección vamos a enlazar aquellos puntos de borde que caigan sobre una determinada curva con determinada forma. A diferencia del método de análisis local explicado, ahora se van a considerar relaciones globales entre píxeles de borde.

Supongamos que para  $n$  puntos de la imagen, se desean encontrar aquellos subconjuntos de puntos que caen en línea rectas. Una posible solución podría ser en primer lugar encontrar todas las líneas determinadas por cada par de puntos y entonces encontrar todos los subconjuntos de puntos que están cerca de cada recta en particular. Este problema así planteado requiere encontrar  $n(n - 1)/2 \sim n^2$  rectas y realizar  $n(n(n - 1))/2 \sim n^3$  comparaciones de cada punto a cada línea. Este método será inviable salvo en casos triviales. Una alternativa al método de búsqueda exhaustiva es la Transformada de Hough..



**Figura A.5** Ejemplo de enlazado global borde mediante una recta.

## A.4 Técnicas de Umbrales. Umbralización.

La imagen  $I(x, y)$  está compuesta de objetos claros sobre fondo oscuro de tal forma que los niveles de gris están agrupados en dos modos predominantes. Una forma de separar los objetos del fondo consiste en seleccionar un umbral  $T$  que separe esos modos. Entonces, cualquier punto  $(x, y)$  para el que se cumpla que

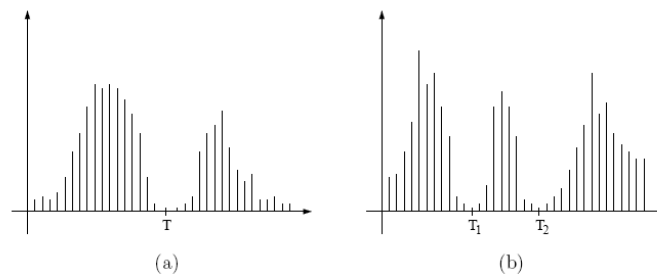
$$I(x, y) > T,$$

Se le etiqueta como objeto; en otro caso, como fondo.

Supongamos que el histograma de los niveles de gris de una imagen  $I(x, y)$  es el que se muestra en la . La imagen  $I(x, y)$  está compuesta de objetos claros sobre fondo oscuro de tal forma que los niveles de gris están agrupados en dos modos predominantes. Una forma de separar los objetos del fondo consiste en seleccionar un umbral  $T$  que separe esos modos. Entonces, cualquier punto  $(x, y)$  para el que se cumpla que

$$I(x, y) > T,$$

Se le etiqueta como objeto; en otro caso, como fondo.



**Figura A.6** Histograma de nivel de gris de una imagen que se puede segmentar (a) con un único valor de umbral y (b) con múltiples umbrales.

En general, un método de umbral se puede ver como una operación en la que se hace un test de cada píxel con respecto a una función  $T$  de la forma

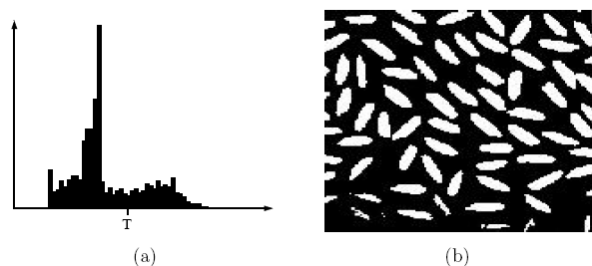
$$T = T(x, y, p(x, y), I(x, y)),$$

Donde  $I(x, y)$  es el nivel de gris del punto  $(x, y)$  y  $p(x, y)$  denota cualquier propiedad local de ese punto. El método de umbral dará lugar a otra imagen  $B(x, y)$  definida por:

$$B(x, y) = \begin{cases} 1 & \text{si } I(x, y) > T \\ 0 & \text{si } I(x, y) \leq T. \end{cases}$$

En este caso un píxel con etiqueta 1 de la imagen  $B$  corresponderá a objetos, mientras que un píxel con etiqueta 0 corresponderá al fondo.

Cuando  $T$  dependa sólo del nivel de gris  $I(x, y)$  se denomina **umbral global** (en la **Figura A.7a** se puede ver un ejemplo en este caso). Si  $T$  depende tanto del nivel de gris  $I(x, y)$  como de la propiedad local  $p(x, y)$ , el umbral se denomina **umbral local**. Si, además,  $T$  depende de las coordenadas espaciales  $x$  e  $y$ , el umbral se denomina **umbral dinámico**.



**Figura A.7** Segmentación con umbral global de una imagen con granos de arroz. (a) Histograma correspondiente a la imagen original y en (b) la imagen resultado de la segmentación.

### Umbral Óptimo.

- **Método de P-Cuantil.**

Este método usa conocimiento sobre el área o el tamaño del objeto que se desea extraer para segmentar la imagen. En una aplicación determinada un los objetos pueden ocupar alrededor de  $p$  porciento del área de la imagen. Utilizando este conocimiento es posible particionar la imagen de modo el  $p$  porciento de los píxeles al objeto.

Esta claro que el uso de este método es muy limitado. Sin embargo en aplicaciones como pre procesamiento para reconocimiento de caracteres es muy útil.

- **Método de la Moda**

El histograma de para imágenes en niveles de grises se puede considerar una estimación de la función densidad de probabilidad de nivel de gris,  $p(z)$ . Esta función será la mezcla de dos funciones unimodales, una para la región clara y otra para la región oscura de la imagen. Además, los pesos de estas densidades van a ser proporcionales al área de cada modo. Si se conoce o se supone la forma de las funciones de densidad de cada modo, se puede determinar el umbral óptimo (en términos de mínimo error) para la imagen a segmentar.

- **Selección Iterativa del Umbral.**

Al algoritmo para realizar una iteración iterativa bastante exacta de umbral es el siguiente:

1. Seleccionar un estimado inicial del umbral  $T$ . Un buen valor puede ser la media de la imagen.

2. Realizar la partición de la imagen utilizando este umbral  $T$ , obteniendo así las regiones  $R_1$  y  $R_2$ .
3. Calcular las medias de estas particiones,  $\mu_1$  y  $\mu_2$ , respectivamente.
4. Seleccionar un nuevo umbral  $T = \frac{\mu_1 + \mu_2}{2}$ .
5. Repetir los pasos del 2 al 4 hasta que las medias  $\mu_1$  y  $\mu_2$  no cambien.

### **Umbrales Basados en varias variables.**

Hasta ahora hemos considerado la determinación de umbrales para una única variable: el nivel de gris de la imagen. En algunos casos, se dispone de varias variables que caracterizan cada pixel de la imagen. Un ejemplo muy claro son las imágenes en color, donde se dispone de las componentes RGB para formar la imagen compuesta en color. En este caso, cada pixel se caracteriza por tres variables, y se puede construir un histograma en tres dimensiones.

El concepto de segmentación mediante umbral en este caso da lugar al empleo de técnicas de clasificación o *clustering*.

## **A.5 Segmentación Orientada a Regiones.**

El objeto de los métodos de segmentación descritos en las secciones anteriores fue encontrar bordes entre regiones. Los métodos que examinaremos ahora construyen las regiones directamente.

Las regiones resultantes de los procesos de segmentación que ahora examinaremos deberán ser homogéneas y maximales, donde maximales significa que los criterios de homogeneidad no se cumplirán tras la unión de una región con alguna de sus adyacentes.

Sea  $R$  la región correspondiente a la imagen a segmentar. Vamos a ver el proceso de segmentación como un proceso en el cual dividimos la región  $R$  en  $n$  subregiones  $R_1, R_2, \dots, R_n$ , tal que:

$$\bigcup_{i=1}^n R_i = R$$

$R_i$  es una región conectada,  $i = 1, 2, \dots, n$

$R_i \cap R_j = \emptyset$  para todo  $i$  y  $j$  con  $j \neq i$

$P(R_i) = \text{CIERTO}$  para  $i = 1, 2, \dots, n$

$P(R_i \cup R_j) = \text{FALSO}$  para todo  $i$  y  $j$  adyacentes con  $j \neq i$ ,

Donde  $P(\cdot)$  es un predicado lógico definido sobre la partición de  $R$  y  $\emptyset$  es el conjunto vacío.

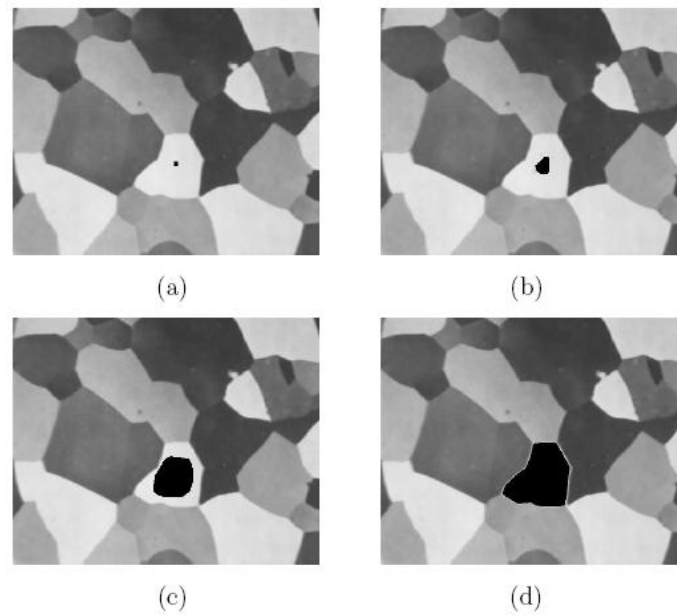
### **Crecimiento de Regiones.**

Como el nombre indica, el crecimiento de regiones es un procedimiento mediante el cual se agrupan píxeles o subregiones en regiones mayores. El procedimiento más sencillo se denomina agregación de píxeles, que comienza a partir de un conjunto de píxeles semilla, de forma que a partir de cada semilla se crecen regiones añadiendo píxeles a dicha semilla de entre aquellos píxeles vecinos que tienen propiedades similares. El resultado de la segmentación dará lugar como mucho a tantas regiones como semillas haya. Sin embargo, puede darse el caso de que dos de esas semillas correspondan a píxeles de la misma región. En este caso el crecimiento desde una de las semillas absorberá a la otra, que en este caso deberá ser descartada.

Dos problemas fundamentales en el crecimiento de regiones son: por un lado, la selección de las semillas o puntos de partida que representen adecuadamente a las regiones de interés; y por otro, la elección de las propiedades adecuadas que permitan ir añadiendo píxeles durante el proceso de crecimiento.

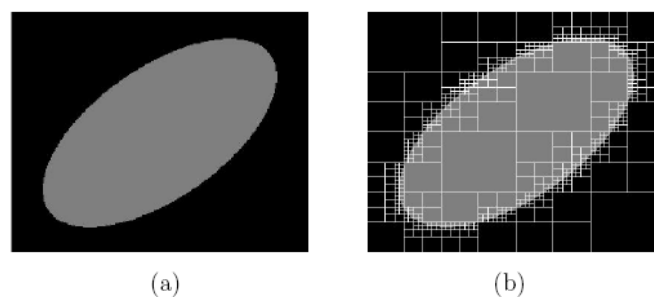
La selección de los puntos de partida en muchos casos depende de la naturaleza de la imagen a segmentar. La selección del criterio de similitud depende no sólo del problema considerado, sino también del tipo de imagen disponible.

Otro problema añadido en el crecimiento de regiones es la regla de parada. Básicamente, el crecimiento termina cuando no existen más píxeles en el vecindario de la región ya crecida que cumplan el criterio de similitud.



**Figura A.8** (a) Imagen que se le ha marcado un punto interior a la región a segmentar. (b) – (d) Proceso de crecimiento de la marcada.

Un proceso alternativo al crecimiento de regiones es subdividir inicialmente la imagen en un conjunto arbitrario de regiones disjuntas y posteriormente fusionar y/o dividir estas regiones con el objetivo de que finalmente se cumplan las condiciones dadas por las ecuaciones del tópico anterior.



**Figura A.9** División de Imagen correspondiente a la representación quadtree.

Sea  $R$  la región correspondiente a la imagen completa y sea  $P$  el predicado de similitud elegido. Un procedimiento para segmentar  $R$  consiste en subdividir recursivamente cada región en cuadrantes más y más pequeños hasta que se cumpla para cada región  $R_i$  que  $P(R_i) = \text{CIERTO}$ . Es decir si  $P(R) = \text{FALSO}$ , se divide la imagen en cuatro regiones (cuadrantes). Si para alguno de los cuadrantes  $P$  es **FALSO**, se vuelve a dividir dicho cuadrante en cuatro subcuadrantes y así sucesivamente.

Tras la etapa de división, la partición final contiene regiones adyacentes con propiedades idénticas, esto se puede solucionar permitiendo la fusión de regiones adyacentes cuando la

unión de ambas cumpla el predicado **P**. Podríamos resumir el procedimiento de división y fusión de regiones como sigue:

1. Dividir una región  $R_i$  en sus cuatro cuadrantes disjuntos siempre que  $P(R_i) = \text{FALSO}$ .
2. Fusionar dos regiones adyacentes  $R_i$  y  $R_j$  siempre que  $P(R_i \cup R_j) = \text{CIERTO}$ .
3. Parar el algoritmo cuando no sea posible realizar más divisiones y fusiones.

## **Anexo B: Reconstrucción.**

### **B.1 Algoritmos Basados en Marching.**

#### **B.1.1 Marching Cube.**

Este algoritmo tiene dos componentes principales. La primera es decidir como definir la sección o secciones de la superficie que cortan un cubo. Si se clasifican los vértices como dentro o fuera de la superficie, existen 256 posibles combinaciones. Dos de estos casos son triviales: todos los puntos dentro y todos los puntos fuera. Para el resto de las configuraciones hace falta determinar, para cada arista del cubo, si corta la superficie. Luego, en una segunda pasada, por medio de interpolación se calcula el punto exacto por donde la superficie corta las aristas. Una vez que se tienen los puntos exactos por donde la superficie corta al cubo, solo hace falta unir estos puntos para formar los triángulos.

Descartando casos simétricos solo quedan 14 de los 254 casos restantes. Cuando un solo vértice es menor que el valor buscado, se forma un solo triángulo que corta a este cubo. Existen 8 casos simétricos a este. Lo mismo sucede cambiando la normal del triángulo, haciendo el caso opuesto; en este caso un vértice es mayor y los otros 7 menores que el color buscado. En la **Figura 1.3** se muestran los principales casos de Marching Cubes.

Todos los demás casos no triviales pueden ser resueltos agregando entre 1 y 4 triángulos a la Isosuperficie. Los vértices de cada triángulo pueden ser calculados por medio de interpolación lineal entre los vértices de los vóxeles, o simplemente tomado el punto medio de las aristas como vértice. Si se decide utilizar el primer caso la superficie obtenida será más suave y regular.



### **B.1.2 Marching Tetrahedra.**

Originalmente se construyó la malla de tetraedros dividiendo el cubo (Vóxel) en 5 tetraedros **Figura 1.4**. Desafortunadamente esta división introduce ambigüedades debido a que la simetría las subdivisiones de los cubos debe alternarse entre cubos para alinear las caras de los tetraedros. Entonces existen dos posibles divisiones de la malla de cubos. Esta ambigüedad solo puede ser resuelta usando interpolación cúbica en lugar de lineal. Sin embargo este enfoque es mucho más complejo y resulta en una LookupTable de 59 casos, muchos más que en el método original de Marching Cubes.

Esta se basa en dividir los cubos en tetraedros centrados. Los tetraedros de esta nueva división se encuentran centrados respecto del cubo. Al contrario que el método descrito con anterioridad en el que se subdivide el cubo, en este caso los tetraedros siempre tienen la misma forma, lo cual elimina las ambigüedades de la malla de triángulos resultante.

La principal desventaja de las técnicas basadas en tetraedros es que crean una cantidad de triángulos aun mayor que Marching Cubes para unos datos de entrada determinados. Esto agrava el segundo problema de Marching Cubes, es decir la regularidad de la malla de la superficie resultante.

## **Anexo C: Formato de Ficheros.**

### **C.1 Especificación y Características del DICOM.**

#### **Header.**

El Header consiste en un preámbulo del archivo de 128 octetos, seguido por 4 octetos DICOM. El Header puede o no incluirse en el archivo.

El Encabezamiento consiste en 128 bytes (Preámbulo) que no tiene por qué contener información DICOM es usado para compatibilidad con otros formatos de ficheros, seguido de un prefijo DICOM de 4 byte.

|             |                             |
|-------------|-----------------------------|
| Preámbulo   | Prefijo                     |
| 128 octetos | 4 octetos = 'D','I','C','M' |

Figura E.1. Representación del Header DICOM.

### DataSet.

Un Data Set representa una instancia de la información del estudio realizado a un paciente; esta constituido por varios **DataElement** que es una etiqueta que va a contener información relevante en conjunto del **DataSet** y esta formado por:

1. **TAG.**- Etiqueta de identificación, compuesto por el número del grupo y número del elemento dentro del grupo.
2. **VR.**- Valor de representación que indica el tipo de dato almacenado (entero, real, cadena de caracteres, etc.)
3. **VALUE LENGTH.**- Indica la longitud del dato.
4. **VALUE FIELD.**- Datos, proporciona información del paciente, análisis realizado, o de la imagen obtenida, dependiendo del valor de la etiqueta.

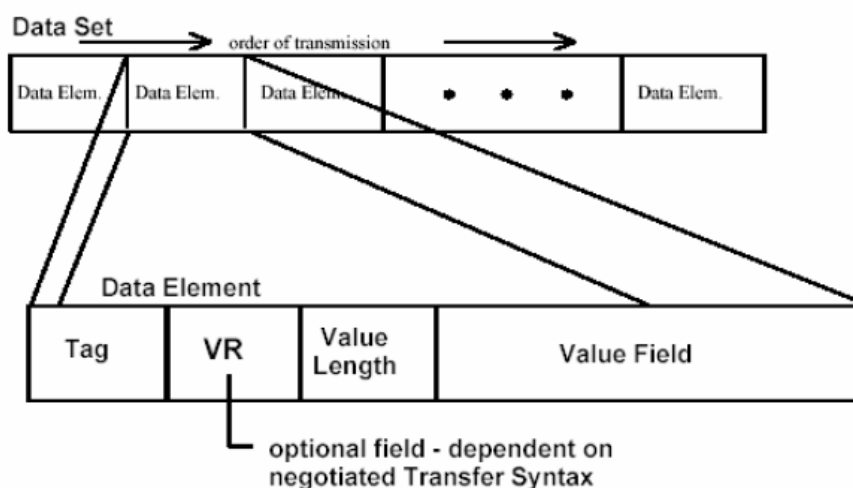


Figura E.2 Representación grafica de un Data Set DICOM.

### DataElement.

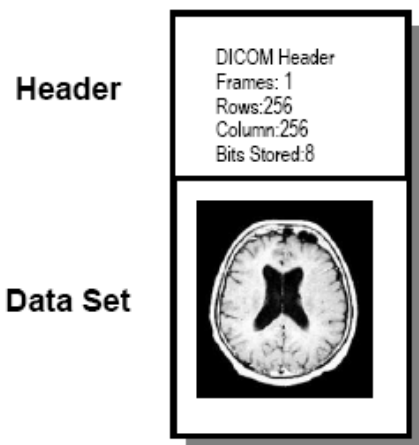
Un **DataElement** tendrá una de tres estructuras. Dos de estas estructuras contienen el **VR** del elemento de datos (**VR explícito**) pero se diferencian en la manera en que se expresan

sus longitudes, mientras que la otra estructura no contiene el **VR (VR implícito)**. Las tres estructuras contienen la Etiqueta del **DataElement** (TAG), la Longitud del Valor (VL) y el Valor para el **DataElement** (VF).

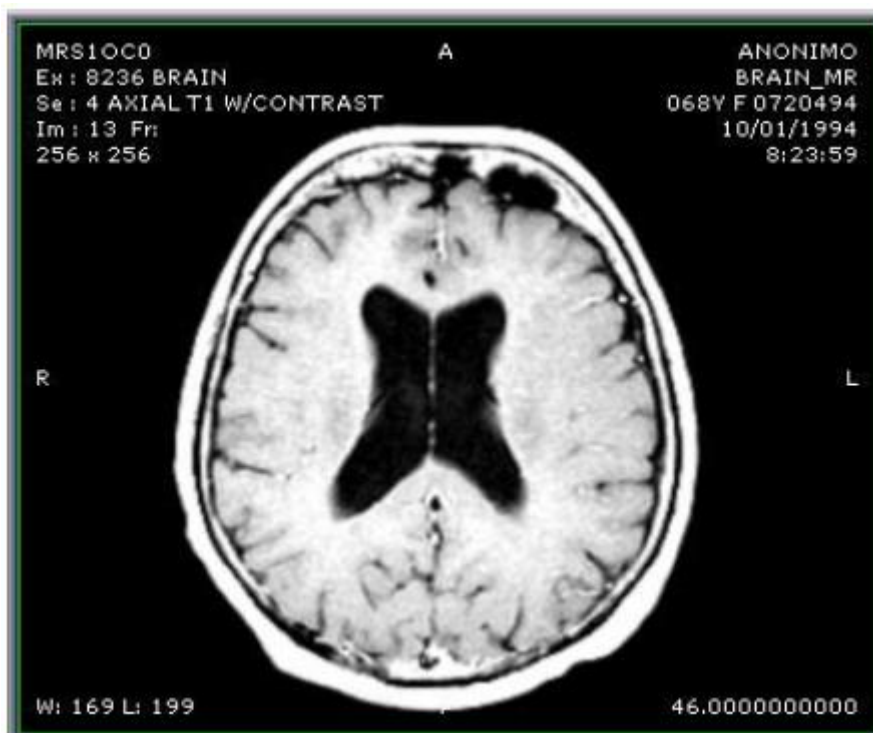
### Sintaxis de Transferencia.

| Formato                | Sintaxis de Transferencia UID           | Descripción                           |
|------------------------|-----------------------------------------|---------------------------------------|
| Nativo                 | 1.2.840.10008.1.2                       | ✓ Implicit VR, Little Endian          |
|                        | 1.2.840.10008.1.2.1                     | ✓ Explicit VR, Little Endian          |
|                        | 1.2.840.10008.1.2.2                     | ✓ Explicit VR, Big Endian             |
| JPEG                   | 1.2.840.10008.1.2.4.50                  | Baseline (1)                          |
|                        | 1.2.840.10008.1.2.4.51                  | Extended (2,4)                        |
|                        | 1.2.840.10008.1.2.4.52                  | Extended (3,5)                        |
|                        | 1.2.840.10008.1.2.4.53                  | Spectral selection, non-hierar. (6,8) |
|                        | 1.2.840.10008.1.2.4.54                  | Spectral selection, non-hierar. (7,9) |
|                        | 1.2.840.10008.1.2.4.55                  | Full progression, non-hierar. (10,12) |
|                        | 1.2.840.10008.1.2.4.56                  | Full progression, non-hierar. (11,13) |
|                        | 1.2.840.10008.1.2.4.57                  | ✓ Lossless, non-hierar. (14)          |
|                        | 1.2.840.10008.1.2.4.58                  | Lossless, non-hierar. (15)            |
|                        | 1.2.840.10008.1.2.4.59                  | Extended, hierar. (16,18)             |
|                        | 1.2.840.10008.1.2.4.60                  | Extended, hierar. (17,19)             |
|                        | 1.2.840.10008.1.2.4.61                  | Spectral selection, hierar. (20,22)   |
|                        | 1.2.840.10008.1.2.4.62                  | Spectral selection, hierar. (21,23)   |
|                        | 1.2.840.10008.1.2.4.63                  | Full progression, hierar. (24,26)     |
| 1.2.840.10008.1.2.4.64 | Full progression, hierar. (24,26)       |                                       |
| 1.2.840.10008.1.2.4.65 | Lossless, hierar (28)                   |                                       |
| 1.2.840.10008.1.2.4.66 | Lossless, hierar (28)                   |                                       |
| 1.2.840.10008.1.2.4.70 | ✓ Lossless, hierar, first-order pred.14 |                                       |
| RLE                    | 1.2.840.10008.1.2.5                     | ✓ Run Length Encoding, Lossless       |

**Tabla C.1** Sintaxis de Transferencia del Estándar DICOM.

**Ejemplos de Archivos DICOM.**

**Figura C.1** Ejemplo de un Archivo \*.dcm.



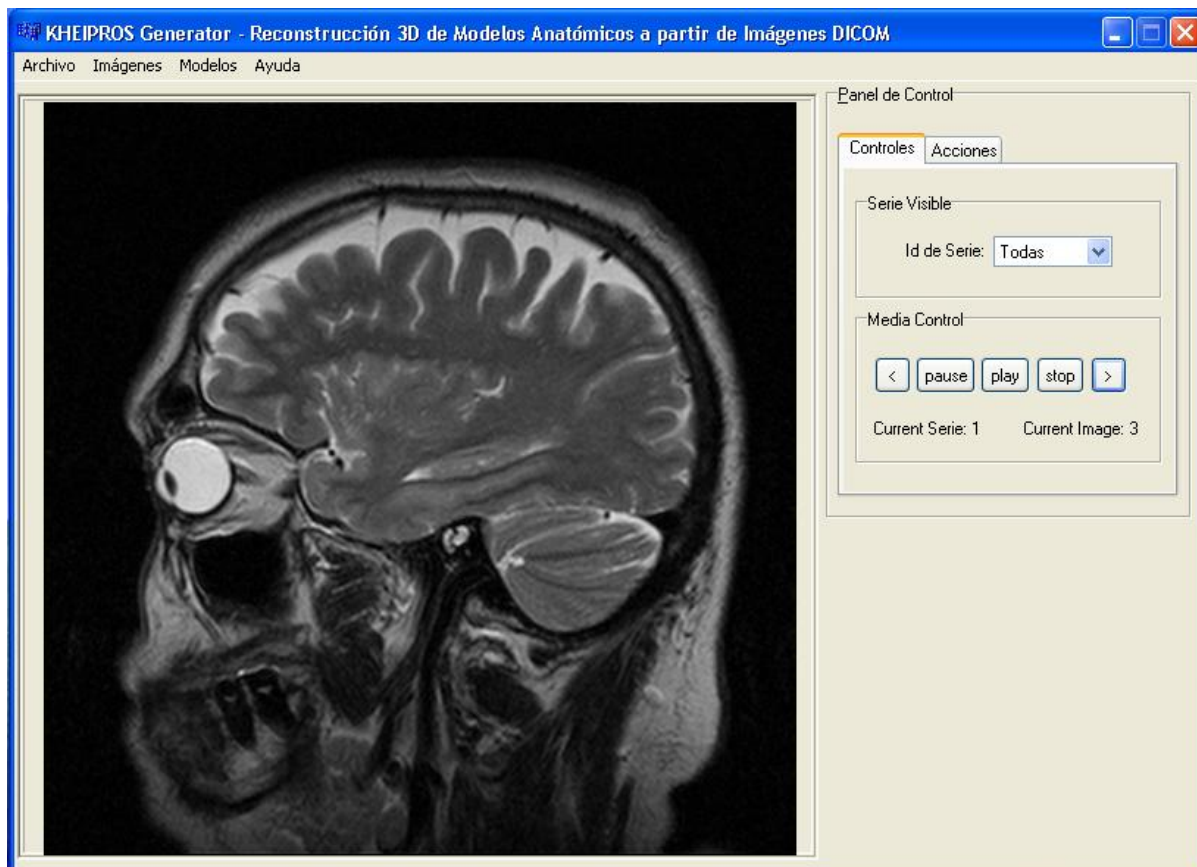
**Figura C.2** Imagen DICOM.

| Cabecera                                            |                                                                                                         |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| First 128 bytes: unused by DICOM format             |                                                                                                         |
| Followed by the characters 'D', 'I', 'C', 'M'       |                                                                                                         |
| This preamble is followed by extra information e.g. |                                                                                                         |
| (0002,0000)                                         | [Group Length] VR: UL, VL: 4, Value: 196                                                                |
| (0002,0001)                                         | [File Meta Information Version] VR: OB, VL: 2, Value: [string of bytes]                                 |
| (0002,0002)                                         | [Media Storage SOP Class UID] VR: UI, VL: 26, Value: 1.2.840.10008.5.1.4.1.1.4                          |
| (0002,0003)                                         | [Media Storage SOP Instance UID VR: UI, VL: 48, Value: 1.2.840.113619.2.1.1.322987149.467.756575561.110 |
| (0002,0010)                                         | [Transfer Syntax UID] VR: UI, VL: 22, Value: 1.2.840.10008.1.2.4.70                                     |
| (0002,0012)                                         | [Implementation Class UID] VR: UI, VL: 32, Value: 1.2.826.0.1.3680043.1.1.4.1.26.2                      |
| (0002,0013)                                         | [Implementation Version Name] --> VR: SH, VL: 14, Value: DCMOBJ4.1.26.2                                 |
| Data Element                                        |                                                                                                         |
| (0008,0008)                                         | [Image Type] --> VR: CS, VL: 16, Value: ORIGINAL\PRIMARY                                                |
| (0008,0016)                                         | [SOP Class UID] --> VR: UI, VL: 26, Value: 1.2.840.10008.5.1.4.1.1.4                                    |
| (0008,0018)                                         | [SOP Instance UID] --> VR: UI, VL: 48, Value: 1.2.840.113619.2.1.1.322987149.467.756575561.110          |
| (0008,0020)                                         | [Study Date] --> VR: DA, VL: 8, Value: 19940110                                                         |
| (0008,0021)                                         | [Series Date] --> VR: DA, VL: 8, Value: 19940110                                                        |
| (0008,0023)                                         | [Content Date] --> VR: DA, VL: 8, Value: 19940110                                                       |
| (0008,0030)                                         | [Study Time] --> VR: TM, VL: 6, Value: 080844                                                           |
| (0008,0031)                                         | [Series Time] --> VR: TM, VL: 6, Value: 082359                                                          |
| (0008,0033)                                         | [Content Time] --> VR: TM, VL: 6, Value: 082405                                                         |
| (0008,0050)                                         | [Accession Number] --> VR: SH, VL: 8, Value: 7340278                                                    |
| (0008,0060)                                         | [Modality] --> VR: CS, VL: 2, Value: MR                                                                 |
| (0008,0070)                                         | [Manufacturer] --> VR: LO, VL: 18, Value: GE MEDICAL SYSTEMS                                            |
| (0008,0080)                                         | [Institution Name] --> VR: LO, VL: 8, Value: ANONIMO                                                    |
| (0008,0090)                                         | [Referring Physician's Name] --> VR: PN, VL: 8, Value: ANONIMO                                          |
| (0008,1010)                                         | [Station Name] --> VR: SH, VL: 8, Value: MRS10C0                                                        |
| (0008,1030)                                         | [Study Description] --> VR: LO, VL: 6, Value: BRAIN                                                     |
| (0008,103E)                                         | [Series Descripción] --> VR: LO, VL: 20, Value: AXIAL T1 W/CONTRAST                                     |
| (0008,1060)                                         | [Name of Physician(s) Reading Study] --> VR: PN, VL: 12, Value: DR.BOURGOUIN                            |
| (0008,1070)                                         | [Operators' Name] --> VR: PN, VL: 2, Value: LR                                                          |
| (0008,1090)                                         | [Manufacturer's Model Name] --> VR: LO, VL: 14, Value: GENESIS_SIGNA                                    |
| (0010,0010)                                         | [Patient's Name] --> VR: PN, VL: 8, Value: BRAIN_MR                                                     |
| (0010,0020)                                         | [Patient Id] --> VR: LO, VL: 8, Value: 0720494                                                          |
| (0010,0030)                                         | [Patient's Birth Date] --> VR: DA, VL: 0, Value:                                                        |
| (0010,0040)                                         | [Patient's Sex] --> VR: CS, VL: 2, Value: F                                                             |
| (0010,1010)                                         | [Patient's Age] --> VR: AS, VL: 4, Value: 068Y                                                          |
| (0010,1030)                                         | [Patient's Weight] --> VR: DS, VL: 10, Value: 64.000000                                                 |
| (0010,21B0)                                         | [Additional Patient History] --> VR: LT, VL: 40, Value: LEFT OPTIC NERVE, POST.FOSSA MENINGIOMA         |
| (0018,0010)                                         | [Contrast/Bolus Agent] --> VR: LO, VL: 16, Value: 12.8 MAGNEVIST                                        |
| (0018,0020)                                         | [Scanning Sequence] --> VR: CS, VL: 2, Value: SE                                                        |
| (0018,0021)                                         | [Sequence Variant] --> VR: CS, VL: 4, Value: NONE                                                       |
| (0018,0022)                                         | [Scan Options] --> VR: CS, VL: 12, Value: CS_GEMS \SP                                                   |
| (0018,0023)                                         | [MR Acquisition Type] --> VR: CS, VL: 2, Value: 2D                                                      |
| (0018,0024)                                         | [Sequence Name] --> VR: SH, VL: 4, Value: memp                                                          |
| (0018,0025)                                         | [Angio Flag] --> VR: CS, VL: 2, Value: Y                                                                |
| (0018,0050)                                         | [Slice Thickness] --> VR: DS, VL: 8, Value: 5.000000                                                    |
| (0018,0080)                                         | [Repetition Time] --> VR: DS, VL: 10, Value: 500.000000                                                 |
| (0018,0081)                                         | [Echo Time] --> VR: DS, VL: 10, Value: 20.000000                                                        |

Figura C.3 Ejemplo de estructura interna de un fichero \*.dcm.

## Anexo D: Imágenes de la Aplicación.

A continuación se muestran varias imágenes obtenidas con la aplicación desarrollada, así como algunas imágenes de la propia aplicación.



**Figura D.1** *Interfaz gráfica de la herramienta. Visualización de Imagen DICOM.*

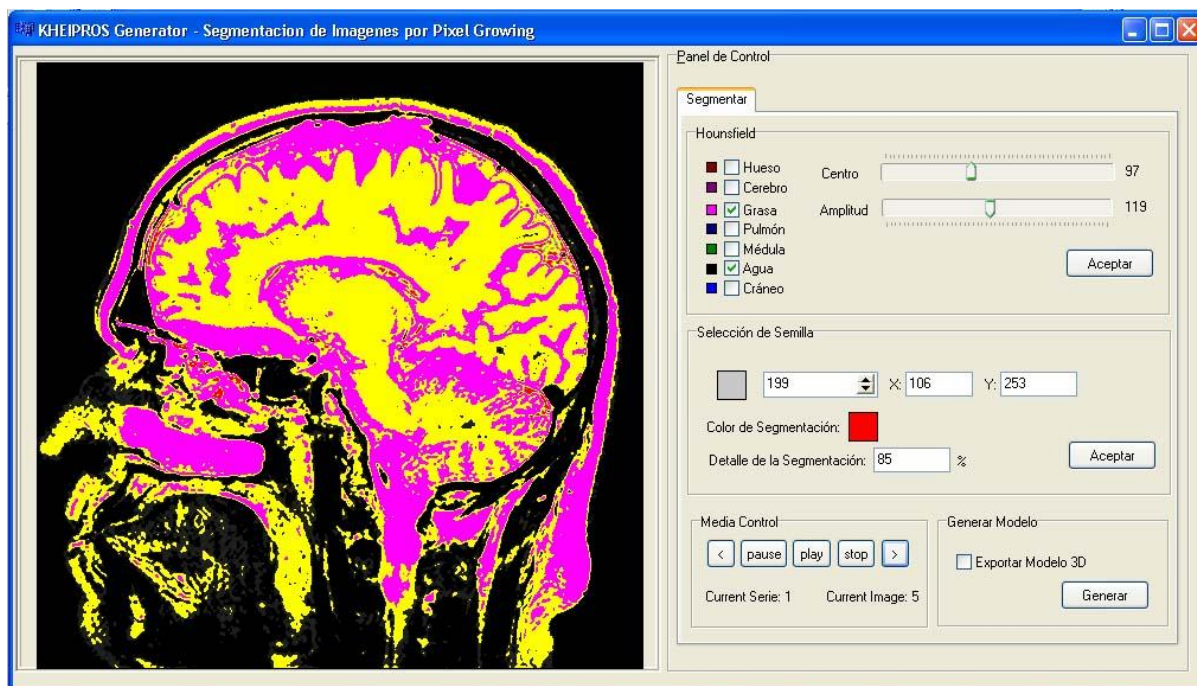


Figura D.2 Interfaz gráfica de la Herramienta. Opción de Segmentar Imagen.

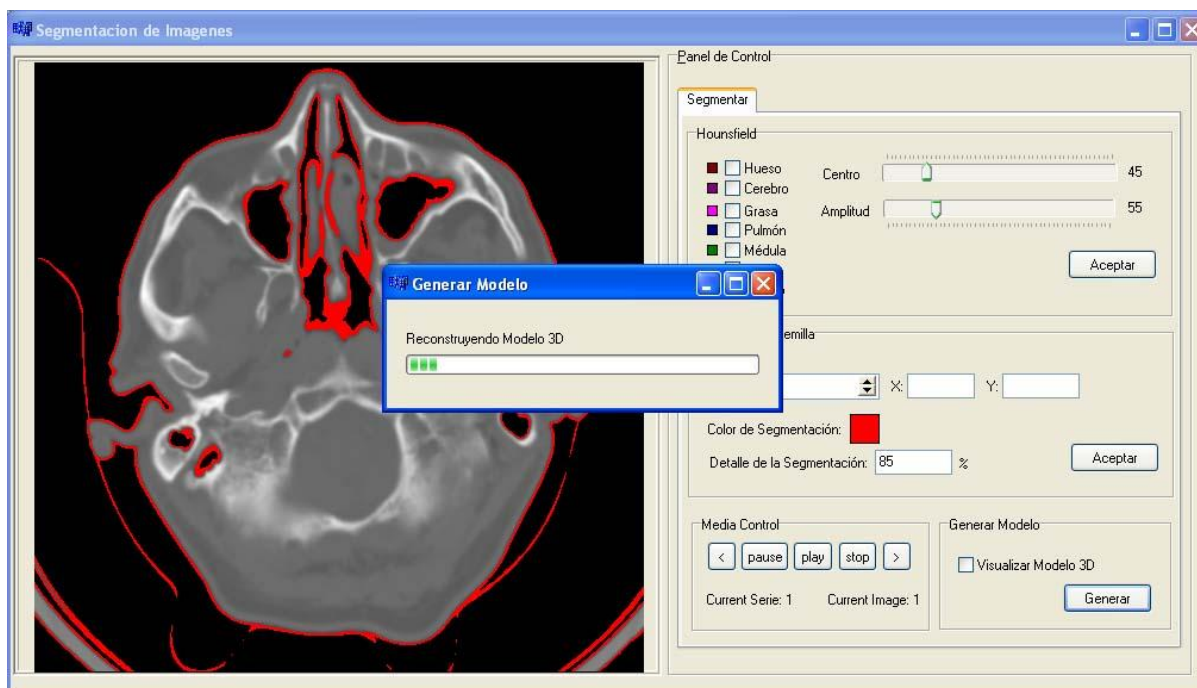
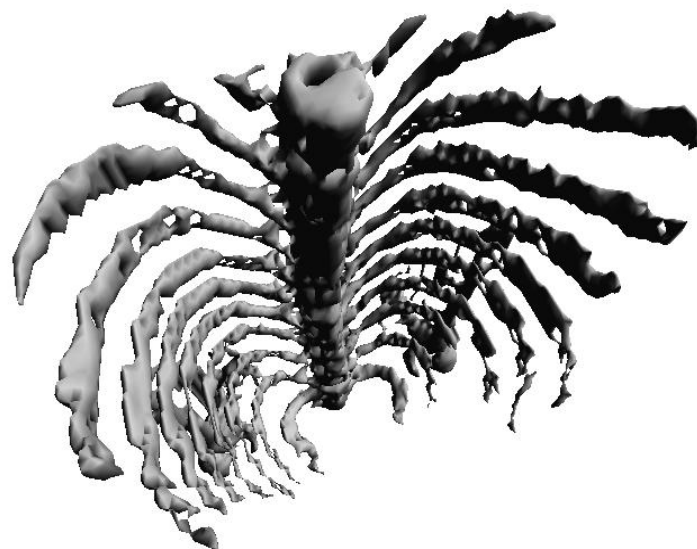


Figura D.3 Interfaz gráfica de la Herramienta. Reconstruyendo Modelo 3D.



**Figura D.4** *Reconstrucción de Cráneo, con un suavizado de dos iteraciones.*



**Figura D.5** *Reconstrucción de Columna Vertebral y Costillas, con un suavizado de una iteración.*