

**ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE CIENCIAS**

**TRATAMIENTO INFORMÁTICO PARA LA SEGMENTACIÓN DE  
TEJIDO PULMONAR EN IMÁGENES DE TOMOGRAFÍA  
COMPUTARIZADA, MEDIANTE EL DESARROLLO DE UN  
CÓDIGO EN PYTHON**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE FÍSICA**

**PROYECTO DE INVESTIGACIÓN**

**JENNIFER LORENA ORTEGA AGUILAR**

`jennifer.ortega@epn.edu.ec`

**Director: DR. NICOLÁS ALEJANDRO VASQUÉZ PAZMIÑO**

`nicolas.vasquez@epn.edu.ec`

**QUITO, DICIEMBRE 2021**

## DECLARACIÓN

Yo, JENNIFER LORENA ORTEGA AGUILAR, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual, correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

---

Jennifer Lorena Ortega Aguilar

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por JENNIFER LORENA ORTEGA AGUILAR, bajo mi supervisión.

---

Dr. Nicolás Alejandro Vasquez Pazmiño  
Director del Proyecto

## AGRADECIMIENTOS

A mi director de tesis, Dr. Nicolás Vásquez por brindarme su ayuda, guía y las herramientas necesarias para la realización de esta tesis. También agradezco al M. Sc. Erick Rundo por su colaboración y por solventar las preguntas que se presentaron al inicio de este proyecto.

Al Dr. Germán Salazar, Director de Docencia de la Clínica Novaclínica Santa Cecilia por su favorable apoyo y apertura para que la Institución participe en este Proyecto, en especial al área de Radiología y a la Dra. Elizabeth Zamora por su colaboración en la validación de los resultados de segmentación de pulmones. Del mismo modo agradezco a la Dra. Manola Pazmiño por solventar algunas dudas con respecto a las estructuras anatómicas.

A mi madre Lorena Aguilar, por su apoyo y amor incondicional en cada etapa de mi vida, en especial a lo largo de mi carrera, por impulsarme y ayudarme en cada objetivo que me planteo. Y finalmente a Galoget por todos sus consejos, ayuda y por compartirme parte de su conocimiento.

## **DEDICATORIA**

*A mi madre, la única persona que siempre estuvo ahí para apoyarme, en las buenas y malas, por siempre motivarme y jamás permitir que me rinda. La vida no me alcanzará para agradecerte por todo lo que sacrificaste por mí y por este sueño.*

# Índice general

<b>Resumen</b>	<b>XIV</b>
<b>Abstract</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Tomografía computarizada y el estándar DICOM</b>	<b>4</b>
2.1. Tomografía Computarizada . . . . .	4
2.2. Interacción de los rayos X con la materia . . . . .	5
2.2.1. Efecto Fotoeléctrico . . . . .	6
2.2.2. Efecto Compton . . . . .	7
2.2.3. Intensidad Transmitida . . . . .	7
2.3. Formación de la imagen . . . . .	8
2.3.1. Proyecciones . . . . .	9
2.3.2. Unidades Hounsfield . . . . .	9
2.3.3. Contraste . . . . .	9
2.3.4. Reconstrucción de imágenes . . . . .	10
2.4. Estándar DICOM . . . . .	10
2.4.1. Modelo de información DICOM . . . . .	11
2.4.2. Diccionario de datos DICOM . . . . .	11
2.4.3. Jerarquía de información DICOM . . . . .	12
2.4.4. IOD de Imagen de Tomografía Computarizada . . . . .	13
<b>3. Metodología</b>	<b>14</b>

3.1.	Datos . . . . .	14
3.1.1.	Preprocesamiento de Datos . . . . .	14
3.1.2.	Base de Datos Inicial . . . . .	14
3.1.3.	Características de los Datos . . . . .	15
3.2.	Python y librerías . . . . .	16
3.3.	Entorno Jupyter Notebook . . . . .	17
3.4.	Software desarrollado para la obtención de Atributos DICOM para TC	17
3.5.	Procesamiento de Imágenes . . . . .	18
3.5.1.	Procesamiento de imágenes DICOM para la visualización . .	19
3.5.2.	Procesamiento para la segmentación de imágenes de TC . . .	23
3.5.3.	Segmentación de pulmones en imágenes de TC . . . . .	26
<b>4.</b>	<b>Tratamiento de imágenes de TC</b>	<b>32</b>
4.1.	Visualización de una imagen de TC . . . . .	32
4.1.1.	Visualización Plano Axial . . . . .	38
4.1.2.	Visualización Plano Coronal . . . . .	41
4.1.3.	Visualización Plano Sagital . . . . .	41
4.2.	Segmentación de tejido pulmonar . . . . .	43
4.2.1.	Resultados de Segmentación de Pulmones . . . . .	51
<b>5.</b>	<b>Conclusiones</b>	<b>54</b>
<b>A.</b>	<b>Anonimización de Imágenes DICOM</b>	<b>57</b>
A.1.	Script anonimizacion.py . . . . .	57
A.2.	Script manejo_tags.py . . . . .	57
A.3.	Archivo private_tags.dat . . . . .	58
<b>B.</b>	<b>Visualización de Atributos DICOM</b>	<b>59</b>
B.1.	Script imprimir_informacion.py . . . . .	59
B.2.	Archivo CT_tags.dat . . . . .	59
B.3.	Ejemplos de Visualización de datos DICOM . . . . .	60

<b>C. Visualización de imagen de TC</b>	<b>61</b>
C.1. Script imprimir_imagen.py . . . . .	61
C.2. Notebook Visualización de imágenes DICOM . . . . .	61
<b>D. Segmentación de tejido pulmonar</b>	<b>62</b>
D.1. Script segmentacion_pulmonar.py . . . . .	62
D.2. Notebook Segmentación de tejido pulmonar . . . . .	62
D.3. Notebook Resultados (Segmentación de Pulmones) . . . . .	62
D.4. Descripción de los exámenes de tórax para segmentación . . . . .	63
<b>Bibliografía</b>	<b>64</b>

# Índice de figuras

2.1. (a) Tomografía computarizada de sección transversal del tórax, tomada en el plano axial. La imagen es de una paciente anónima de un hospital de Ecuador. (b) Una geometría posible para detectores y fuente en un escáner TC, que se mueven en sincronía durante el examen alrededor del paciente. . . . .	5
2.2. La imagen derecha representa la matriz de reconstrucción de una rebanada anatómica dividida en pequeñas cajas de tejido llamadas vóxeles, cada una con un coeficiente de atenuación $\mu$ , la imagen izquierda representa la imagen reconstruida en 2 dimensiones reemplazando vóxeles por píxeles [13,16]. . . . .	8
3.1. Histograma de la base de datos inicial según el elemento Protocol name en modalidad CT. . . . .	15
3.2. Visualización de metadata correspondiente a parámetros de adquisición de un archivo DICOM de TC, utilizando la herramienta creada.	18
3.3. Flujo de trabajo para el procesamiento de imágenes[35]. . . . .	18
3.4. Flujo de trabajo para la visualización de una imagen y de un examen de TC en diferentes planos anatómicos. . . . .	19
3.5. Esquema de la interpretación de los atributos Pixel Spacing, Image Position y de vóxel en cortes consecutivos. . . . .	22
3.6. Técnicas de segmentación de imágenes [59]. . . . .	23
3.7. Ejemplo de segmentación basada en Regiones. . . . .	24
3.8. Ilustración del funcionamiento del algoritmo Watershed [6]. . . . .	25
3.9. Flujo de trabajo para la segmentación de tejido pulmonar en imágenes de TC de tórax. . . . .	27

4.1. Visualización de una imagen de TC de tórax en escala de grises, a partir de la matriz de píxeles en unidades Hounsfield, con 512 filas y 512 columnas. . . . .	33
4.2. Histograma con los valores de píxel en unidades Hounsfield de una imagen de TC. Se demuestra que existe una gran cantidad de píxeles en el valor de $-3024HU$ , que esta fuera del rango de medida de un tomógrafo. . . . .	34
4.3. Visualización de una imagen de TC de tórax con corte axial de una paciente femenina de 48 años de edad, mediante diferentes métodos de visualización. (a) Visualización de la imagen utilizando el rango completo de unidades Hounsfield y adaptándolos automáticamente a la escala de grises. (b) Visualización de la imagen utilizando la ventana con valores de $[-100HU, 300HU]$ . (c) Visualización a través del software incorporado en el Instrumento de TC (iQ-LITE 3.0.0). . . . .	35
4.4. Visualización de una imagen de TC de tórax con corte axial, con diferentes ventanas. (a) Visualización de la imagen utilizando la ventana de $[-1000, 1000]HU$ . (b) Visualización de la imagen utilizando la ventana de $[200, 500]$ . . . . .	36
4.5. Comparación entre la imagen original y la imagen final producto del proceso de remoción de la camilla para una imagen de TC. La visualización sigue el estilo del software iQ-LITE 3.0.0. . . . .	37
4.6. Plano axial de un examen de TC a partir de archivos DICOM con un total de 63 cortes en el plano axial. La visualización de la imagen sigue el estilo del software iQ-LITE 3.0.0. . . . .	40
4.7. Plano coronal de un examen de TC generado a partir de archivos DICOM con un total de 63 cortes en el plano axial y 512 cortes en el plano coronal, utilizando la ventana con valores entre $-100HU$ a $300HU$ . . . . .	41
4.8. Plano sagital de un examen de TC generado a partir de archivos DICOM con un total de 63 cortes en el plano axial y 512 cortes en el plano sagital, utilizando la ventana con valores entre $-100HU$ a $300HU$ . . . . .	42
4.9. Imágenes de gradiente de diferentes cortes axiales de un examen de TC de tórax. . . . .	43

4.10. Proceso para la generación del marcador interno en el corte 40 del examen de TC de tórax de prueba. (a) Imagen binaria resultado del filtraje por unidades Hounsfield con la condición ( $\text{píxel} < -400HU$ ). (b) Pulmones y parte de la camilla que se etiquetaron como región pulmonar (c) Marcador interno, después de eliminar todos los objetos que se etiquetaron erróneamente como región pulmonar al momento de realizar el filtraje por unidades Hounsfield. . . . .	44
4.11. Cortes antes y después del proceso de eliminación de la tráquea. (a)(b)(c)(d) Primeros cortes del examen de TC de torác, donde se aprecia la tráquea, que corresponde al círculo en el centro del corte. (e)(f)(g)(h) Marcador interno después del proceso de remoción de la tráquea y de todos los objetos que no pertenecen a los pulmones. . . . .	45
4.12. Proceso para la generación del marcador externo en el corte 40 del examen de TC de tórax de prueba. (a). Primera dilatación con 10 interacciones. (b) Segunda dilatación con 55 interacciones. (c) Marcador externo, la franja generada es de 45 píxeles. . . . .	46
4.13. Aplicación de algoritmo de Watershed y Black-Tophat para la segmentación en el corte 40. (a) Imagen con los marcadores etiquetados para la aplicación del algoritmo de Watershed. La zona gris representa el marcador externo, la región blanca el marcador interno y la franja negra entre ambos marcadores es el marcador de Watershed. (b) Imagen de gradiente morfológico, aplicado al resultado de la segmentación de Watershed. (c) Reincorporación de estructuras densas mediante la transformación Black-Tophat, que suaviza y rellena cavidades. (d) Pulmones segmentados. . . . .	47
4.14. Muestra de imágenes de un examen de TC de radioterapia de Tórax, el examen cuenta con 75 cortes, la visualización utiliza la ventana con valores entre $-1000HU$ a $1000HU$ . . . . .	48
4.15. Pulmones segmentados de la muestra de imágenes de un examen de TC de radioterapia de Tórax, presentados en la figura 4.14 que contiene un total de 75 cortes. . . . .	49
4.16. (a) Muestra de los últimos cortes de un examen de TC de radioterapia de Tórax. (b) Pulmones segmentados de la muestra de TC presentada en (a). . . . .	50

4.17. Cortes con errores de segmentación de pulmones (a) Tráquea identificada como parte de los pulmones. (b) Los bordes del pulmón no son delimitados correctamente. . . . .	52
B.1. Visualización de la metadata correspondiente al perfil de confidencialidad dentro de los archivos DICOM. . . . .	60

# Índice de cuadros

2.1. Extracto del Diccionario de datos DICOM [27]. . . . .	12
2.2. Extracto de los Módulos IOD de una imagen CT [28]. . . . .	13
4.1. Valores para definir la matriz de píxeles de un archivo DICOM de la base de datos. . . . .	32
4.2. Valores para definir el cambio de unidades de la matriz de píxeles para un archivo DICOM de la base de datos. . . . .	33
4.3. Elementos que definen los valores de escalamiento en imágenes consecutivas para dos archivos DICOM de la base de datos, el atributo InstanceNumber determina el orden de las imágenes respecto de su adquisición. . . . .	38
4.4. Resultados de la segmentación de pulmones en diferentes exámenes. N.º Examen (Identificador del examen), N.º Cortes (Número de cortes por examen), N.º Cortes segmentados (con errores) (Número de cortes con errores en la segmentación de pulmones). . . . .	51
4.5. Resultados cuantitativos de la segmentación de pulmones del número total de cortes, correspondientes a todos los exámenes. . . . .	52
D.1. Características de los exámenes usados para las pruebas de segmentación de pulmones. 10.7RT (10.7 RADIOTERAPIA TORAX), PT (PLANEAMIENTO DE TERAPIA), GEMS (GE MEDICAL SYSTEMS), 6.1 AP (6.1 ABDOMEN Y PELVIS), TCC (TC CORPORAL), HFS (HEAD FIRST-SUPINE), FFS (FEET FIRST-SUPINE). . . . .	63

# Resumen

En el presente proyecto, se desarrollaron dos herramientas que servirán como paso previo para la generación de futuros sistemas de Diagnóstico Asistido por Computadora (CAD) para la detección automática de numerosas enfermedades pulmonares.

La primera herramienta fue realizada con el objetivo de extraer información de archivos con formato DICOM de una manera rápida, sencilla y sin la necesidad de utilizar otro tipo de programas externos, sino solamente mediante el uso de Python y el entorno de programación Jupyter Notebook, que son de acceso libre. El proceso seguido hace posible la generación de imágenes en diferentes planos anatómicos, como es el plano coronal y el plano sagital, por medio de la matriz de píxeles que representa un corte axial del proceso de tomografía computarizada. Las técnicas utilizadas son independientes de la imagen generada en escala de grises, además de que no depende del tipo de ventana utilizada, puesto que para obtener la imagen se emplea la matriz de píxeles en unidades Hounsfield.

Mientras que la segunda herramienta presenta una técnica de segmentación de pulmones en imágenes de tomografía computarizada de tórax. La segmentación hace uso de la matriz de píxeles en unidades Hounsfield para identificar en un principio el tejido pulmonar mediante su grado de atenuación lineal de rayos X, luego se utiliza el algoritmo de Watershed basado en marcadores, siendo esta la base principal de la técnica, además se emplean diferentes operaciones morfológicas y transformaciones sobre los cortes axiales, obteniendo así la segmentación de pulmones.

En este proyecto, se utilizaron 35 exámenes en modalidad de tomografía computarizada (TC) con diferentes protocolos y con una media de 97 imágenes por examen para evaluar la primera herramienta. De modo similar la segmentación de pulmones fue realizada para 17 exámenes de TC de tórax, lo que demostró que la herramienta es precisa y rápida.

# Abstract

In this project, we developed two tools that will serve as a preliminary step for generating future Computer Aided Detection (CAD) systems for the automated detection of numerous lung diseases.

The first tool was made with the aim of extracting information from files with DICOM format in an easy and fast way without needing to use other types of external programs, but only by using Python and the Jupyter Notebook programming environment, which are open source and freely available. The implemented process makes it possible to generate images in different anatomical planes, such as: the coronal plane and the sagittal plane, by means of the pixel matrix that represents an axial section of the computed tomography process. We used techniques that are independent of the image generated in grayscale. In addition, they do not depend on the type of window used, since the pixel matrix in Hounsfield units is used to obtain the image.

On the other hand, the second tool consists of a lung segmentation technique for chest computed tomography images. The segmentation process uses the pixel matrix in Hounsfield units to initially identify lung tissue through its degree of linear X-ray attenuation. Then, the marker-based Watershed algorithm is used, which is the main basis of the technique. In addition, different morphological operations and transformations are used on the axial sections for obtaining the lung segmentation.

In this project, 35 tests in computed tomography (CT) modality with different protocols and with average of 97 images per test were used to evaluate the first tool. Similarly, the lung segmentation tool was tested against 17 chest CT tests, demonstrating that the tool is accurate and fast.

# Capítulo 1

## Introducción

En la medicina actual resulta indispensable el uso de tecnologías que se basan en análisis de imágenes por computadora para tareas de diagnóstico, lo que facilita a médicos o investigadores la visualización y valoración de diferentes anomalías en los pacientes. Existen varios métodos para obtener imágenes de las estructuras internas del paciente como son: Radiografías Computarizadas, Tomografías Computarizadas, Resonancias Magnéticas, etc. [1]. Debido a la gran cantidad de proveedores y tipos de sistemas de adquisición de imágenes en 1993 NEMA (National Electrical Manufacturers Association) diseñó el estándar DICOM (Digital Imaging and Communications in Medicine), que es un formato para codificar y transmitir imágenes médicas. Los archivos DICOM además de guardar la imagen tienen la facultad de almacenar detalles del paciente y parámetros de adquisición. Asimismo, este formato almacena información que da soporte a la imagen visual y permite producir cortes en diferentes planos anatómicos, así como también generar estructuras en 3D, mediante técnicas de procesamiento de imágenes [2].

La tomografía computarizada por rayos X (TC) es un método no invasivo, que permite adquirir imágenes de secciones transversales de la anatomía interna del paciente, en función de la atenuación de los rayos X que atraviesan los tejidos [3]. La tomografía computarizada de tórax es ampliamente utilizada para diagnosticar enfermedades pulmonares, como es el caso del cáncer de pulmón; que según "World Cancer Research Fund International", es el cáncer más común (excluyendo el cáncer de piel) y representa el 12% del total de cáncer. Por esta razón, en los últimos años se han desarrollado una gran cantidad de sistemas CAD (Diagnóstico Asistido por Computadora), con el fin de lograr una detección rápida mediante la utilización de imágenes de TC de tórax, puesto que esto ayuda a disminuir la tasa de muertes [4].

La segmentación de imágenes médicas tiene un gran potencial en el dominio clínico, dado que estas técnicas son necesarias para el desarrollo de sistemas CAD para detección automática de un gran número de enfermedades. En particular, la segmentación de pulmones constituye un paso importante previo al desarrollo de la mayoría de sistemas CAD de identificación de enfermedades pulmonares, ya que, dependiendo del tipo de segmentación utilizada el resultado podría verse afectado [5]. La segmentación pulmonar en imágenes de TC de tórax permiten que los algoritmos de detección de enfermedades pulmonares se ejecuten de una manera eficaz, sin la necesidad de que la imagen completa cause distracciones en zonas que no pertenecen a pulmones. En una imagen de tomografía computarizada de tórax, se observa que los pulmones están rodeados por otras partes del cuerpo humano. La segmentación hace posible que estas otras partes se filtren y desaparezcan de la imagen, lo que significa que un posible clasificador de imágenes o sistema CAD solo se enfoque netamente en los pulmones y no se vea afectado por ruido adicional y exceso de información provocado por zonas irrelevantes, que lo único que generarían son problemas al momento de aplicar los algoritmos de diagnóstico [5,6].

En virtud de lo expuesto anteriormente, en este proyecto se desarrollan dos herramientas necesarias para la generación de futuros sistemas de diagnóstico asistido por computadora para detección automática de un importante número de enfermedades pulmonares. Así, la primera herramienta permite la extracción de información de archivos DICOM para generar imágenes en diferentes planos anatómicos, a partir de las matrices de píxeles que representan cortes axiales obtenidos del proceso de tomografía computarizada por rayos X. Mientras que la segunda herramienta consiste en un método para segmentación de pulmones en imágenes de TC de tórax. Ambas herramientas hacen uso de diferentes tecnologías para procesamiento de imágenes, donde se utilizan librerías del lenguaje de programación de alto nivel Python, que son especializadas en manipulación y tratamiento de imágenes.

La segmentación pulmonar empleada en este proyecto utiliza la transformada de Watershed basada en marcadores, que se describió por primera vez en el artículo "Automatic lung segmentation in ct images using watershed transform" [5], con ciertas modificaciones que mejoran y delimitan de una manera más acertada los pulmones. Esta técnica hace uso de un marcador interno y de un marcador externo, que son utilizados para seleccionar las zonas donde se desea aplicar el algoritmo en la imagen de gradiente del corte axial de TC de tórax, evitando así una sobre segmentación que constituye un problema típico del algoritmo [7].

La segmentación se realiza con el fin de generar bordes que separen los pulmones

de las demás estructuras. La técnica hace uso del concepto de unidades Hounsfield y de los valores tabulados de la atenuación de rayos X de las regiones pulmonares, para generar en un principio el marcador interno. Luego, se realizan operaciones morfológicas sobre el marcador interno para obtener el marcador externo. Posteriormente se aplica el algoritmo de Watershed sobre la imagen de gradiente utilizando los marcadores y generando así una matriz que contiene tejido pulmonar, mientras que todas las demás estructuras en la imagen son eliminadas. La imagen de gradiente encuentra mínimos regionales con el propósito de mejorar la segmentación y eliminar ruido.

Como paso final se rellenan las cavidades que son excluidas por el algoritmo de Watershed, ya que algunas estructuras densas que se encuentran en los bordes de los pulmones son descartadas debido a que tienen coeficiente de atenuación diferentes a las que usualmente se encuentran para este tejido, como es el caso de los nódulos yuxtapleurales y vasos hiliares. Lo que diferencia esta técnica de otras es el hecho de que no se eliminan las estructuras densas localizadas en los bordes de los pulmones como se menciona en “Lung nodule detection by using deep learning” [6], en consecuencia produce una mejor segmentación con respecto a otros métodos y permite implementar algoritmos de clasificación generando mejores resultados.

Todo lo explicado en este proyecto, el código utilizado para su demostración, cuadernos Jupyter, carpetas generadas, así como también las herramientas creadas y los datos utilizados se encuentran disponibles para el lector en el siguiente repositorio: <https://github.com/jennifer95/visualizacion-segmentacion-pulmonar-DICOM-python>. El objetivo de la publicación de las herramientas, scripts y resultados en el repositorio es para contribuir al conocimiento libre y la colaboración científica. Si está interesado en aportar al repositorio puede enviar su Pull Request o contactarse con la autora vía correo electrónico. Asimismo, en la sección Apéndices encontrará una pequeña descripción de los principales archivos generados.

# Capítulo 2

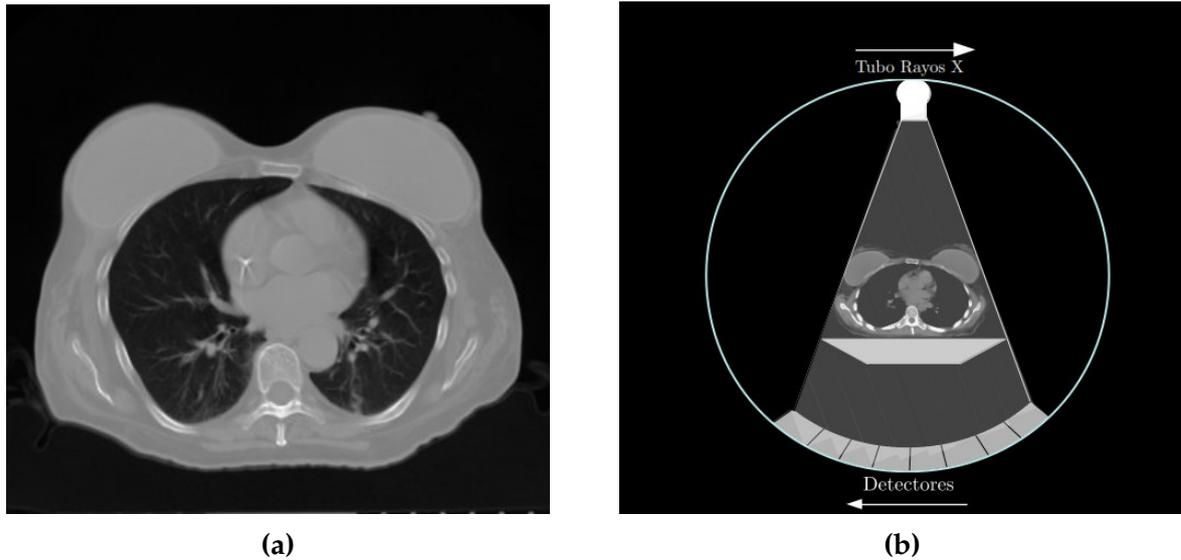
## Tomografía computarizada y el estándar DICOM

### 2.1. Tomografía Computarizada

La tomografía computarizada (TC) es una técnica no invasiva por la cual se obtienen imágenes de secciones transversales del cuerpo usando rayos X. La imagen es un mapa transversal de la atenuación de rayos X al atravesar los diferentes tejidos dentro del paciente. La diferencia principal entre la tomografía computarizada y las técnicas convencionales de rayos X es la ausencia de superposición de estructuras adyacentes [1].

Las radiografías, al ser sombras creadas por la proyección de todos los objetos radiopacos (que absorben rayos X), pierden inherentemente información de profundidad y solo proporcionan una imagen plana del cuerpo. Debido a que la TC ha mejorado enormemente la sensibilidad al contraste, la absorción de rayos X del esqueleto no abruma los detalles del resto del cuerpo proporcionando una excelente visualización de los tejidos blandos. Por lo tanto, se puede distinguir entre las muchas estructuras que se superponen en una radiografía de diagnóstico, presentando una vista transversal de la anatomía del cuerpo (Figura 2.1a). Además, a partir de un conjunto de imágenes de TC y mediante una reconstrucción tridimensional generada por computadora, se pueden obtener imágenes finales en planos sagitales o coronales. Estas imágenes se ven desde la misma perspectiva que una radiografía digital, pero muestran cortes finos de tejido en lugar de tejidos y estructuras superpuestas [1,8,9].

Para la recopilación de datos en un examen de TC se tiene el siguiente proceso: un haz estrecho de rayos X emitidos por el tubo de rayos X irradia al paciente. Una



**Figura 2.1:** (a) Tomografía computarizada de sección transversal del tórax, tomada en el plano axial. La imagen es de una paciente anónima de un hospital de Ecuador. (b) Una geometría posible para detectores y fuente en un escáner TC, que se mueven en sincronía durante el examen alrededor del paciente.

vez que la radiación atraviesa al paciente, los rayos X son atenuados por los diferentes tejidos, para finalmente ser medidos por una fila o matriz de detectores ubicados debajo del paciente (Figura B.1b). Cada detector registra la atenuación de los rayos X a lo largo de una línea a través del cuerpo, similar a una radiografía convencional pero en una dimensión. Al terminar este proceso, tanto el tubo de rayos X como los detectores giran en sincronía alrededor de la misma sección del paciente, obteniendo mediciones de la atenuación lineal de rayos X, vistos desde diferentes ángulos en la misma sección o rebanada. Una vez que se hayan registrado suficientes orientaciones, el cuerpo del paciente es desplazado y todo el proceso se repite para un nuevo plano axial [1, 10].

## 2.2. Interacción de los rayos X con la materia

Cuando la radiación atraviesa los diferentes tejidos en un paciente se producen procesos de absorción y dispersión de la radiación incidente, que dependen de la energía de los fotones incidentes y de las propiedades del tejido corporal a través de los cuales viajan. Todas las interacciones entre los rayos X y la materia son probabilísticas, es decir; para cada caso, solo hay una probabilidad de transmisión o de absorción del fotón. Estos procesos son de gran importancia en la física médica ya

que juegan un papel fundamental en la imagenología, en la radioterapia así como en la dosimetría de las radiaciones.

A continuación se describirán los tipos de interacción de rayos X con la materia que están presentes en el proceso de tomografía computarizada: el efecto fotoeléctrico y el efecto Compton. La producción de pares no será analizada, ya que se producen a energías de rayos X muy altas, las cuales están fuera de las energías utilizadas para el proceso de TC [11].

### **2.2.1. Efecto Fotoeléctrico**

En este proceso, se produce la absorción total de rayos X con energías por debajo de aproximadamente  $25KeV$ . El fotón incidente interactúa con un electrón fuertemente ligado al átomo que se encuentra en un orbital interno, depositando su energía. Esta interacción da como resultado la absorción completa del fotón de rayos X. Una parte de la energía del fotón es utilizada para liberar al electrón del átomo y la energía sobrante se transforma en energía cinética del electrón liberado, también conocido como fotoelectrón. Los fotoelectrones, en general, viajan distancias muy cortas antes de que su energía sea reabsorbida. Por otro lado, el orbital con un electrón vacante se rellena rápidamente mediante la captura de un electrón. Cuando esto ocurre, se libera energía en forma de un fotón conocido como rayos X característicos, que contribuye a la dispersión si se escapa del cuerpo. La probabilidad de que un fotón de rayos X sea absorbido mediante efecto fotoeléctrico es proporcional al cubo del número atómico  $Z$  del material absorbente. Los tejidos blandos del cuerpo consisten principalmente con  $Z$  efectivos bajos, por lo que la probabilidad de absorción por efecto fotoeléctrico es baja en estos tejidos. Por el contrario, el hueso tiene un  $Z$  efectivo mayor, lo que hace que tenga más probabilidad de absorber fotones por efecto fotoeléctrico. La densidad del material también influye en la cantidad de átomos que encuentra el rayo X cuando pasa a través de un determinado espesor de material, por lo que los materiales más densos generalmente absorben mejor los rayos X que los menos densos. La probabilidad de tener efecto fotoeléctrico es inversamente proporcional a la energía de los rayos X incidentes. Sin embargo, si la energía de los rayos X coincide con la cantidad de energía requerida para promover al electrón a un orbital de mayor energía, la probabilidad de absorción aumenta considerablemente. Llamamos a esa energía borde de absorción. La energía a la que se producen los bordes de absorción depende del elemento químico [3, 12].

### 2.2.2. Efecto Compton

A medida que aumenta la energía de los fotones de rayos X por encima de  $25\text{KeV}$ , la interacción que prevalece es el efecto Compton. Para estudiar este efecto, se puede considerar que el fotón incidente choca con un electrón de una órbita externa del átomo, transfiriéndole parte de su energía y dispersándose. La energía transferida al electrón es suficiente como para liberarlo del átomo al cual se encuentra ligado. Como las energías de los rayos X incidentes son tan grandes en comparación con la energía que une al electrón al átomo, se considera que la radiación incidente interactúa con un electrón libre. Dado que en el efecto Compton, los fotones de rayos X interactúan con los electrones como si no estuvieran unidos a un átomo, solo importa el número total de electrones dentro del material. Por lo tanto, solo el grosor de un absorbente y su densidad son importantes para la absorción de rayos X mediante este mecanismo, por lo que la probabilidad de dispersión por efecto Compton depende linealmente de  $Z$  y disminuye con el aumento de energía. El efecto Compton es la interacción que predomina en tejidos blandos, por ejemplo el pulmón [13].

### 2.2.3. Intensidad Transmitida

El parámetro que caracteriza la penetración de rayos X en el medio es el coeficiente de atenuación  $\mu$ . Este coeficiente contiene las contribuciones de los procesos de interacción de fotones con la materia descritos anteriormente. Además,  $\mu$  depende de la energía del fotón incidente, del número atómico y densidad del absorbente. Cuanto mayor sea su coeficiente de atenuación, más fácilmente atenuará un material los rayos X con una energía en específico. Cada elemento químico tiene su propio coeficiente de atenuación específico. Un material, como el tejido corporal, tiene un coeficiente de atenuación que depende del promedio de todas las sustancias químicas que lo constituyen. Igualmente, se define  $\mu_m = \mu/\rho$ , siendo  $\rho[\text{cm}^2/\text{g}]$  la densidad del material absorbente. El coeficiente  $\mu_m$  depende únicamente de la composición química del material y la energía de los fotones.

La intensidad de rayos X monoenergéticos transmitida a través de dos o más regiones consecutivas, con espesores  $x_1, x_2, x_3, x_4$ , etc. y coeficientes de atenuación  $\mu_1, \mu_2, \mu_3, \mu_4$ , etc., viene dada por:

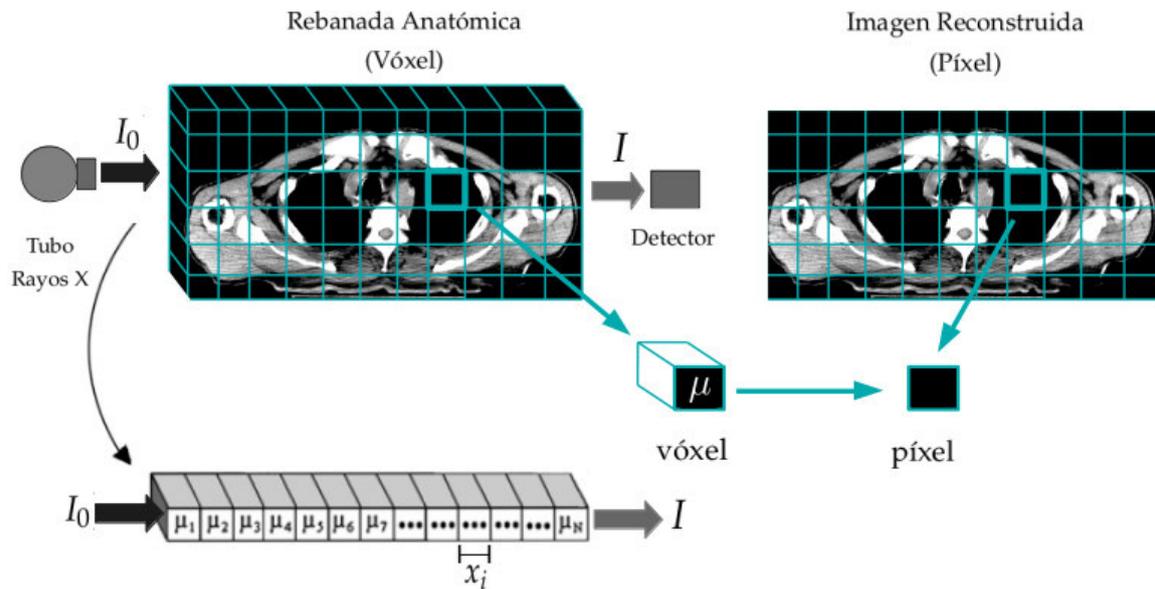
$$I_{trans} = I_0 e^{-(\mu_1 x_1 + \mu_2 x_2 + \mu_3 x_3 + \dots)}. \quad (2.1)$$

Donde  $I_0$  representa la intensidad de rayos X monoenergéticos incidentes. Adicionalmente, podemos definir la intensidad atenuada [14,15] como:

$$I_{aten} = I_0 - I_{trans}. \quad (2.2)$$

### 2.3. Formación de la imagen

Una imagen o slice de TC, es una matriz donde la posición de un elemento dentro de la matriz simboliza la posición de un píxel en la imagen. Es decir, la posición del elemento en la matriz representa la posición del tejido penetrado por el haz de rayos X dividido en pequeños elementos de volumen  $\Delta x$ ,  $\Delta y$  y  $\Delta z$  conocido como vóxel (o región discretizada del objeto). A cada vóxel, se le asigna un coeficiente de atenuación  $\mu$ . Las mediciones de transmisiones de rayos X ( $I$ ), se pueden expresar como la suma de los valores de atenuación que ocurre en cada vóxel a lo largo de la trayectoria del rayo (Figura 2.2).



**Figura 2.2:** La imagen derecha representa la matriz de reconstrucción de una rebanada anatómica dividida en pequeñas cajas de tejido llamadas vóxeles, cada una con un coeficiente de atenuación  $\mu$ , la imagen izquierda representa la imagen reconstruida en 2 dimensiones reemplazando vóxeles por píxeles [13,16].

### 2.3.1. Proyecciones

Las proyecciones  $X_i$  son usadas en la reconstrucción de las imágenes. Para definir las proyecciones vamos a suponer que la radiación incidente sobre el paciente es  $I_0$  y la intensidad de rayos X captada por los detectores es  $I$ . Utilizando 2.1 se define a la proyección como [13,16]:

$$X_i = -\ln(I_i/I_0) = \sum_i x_i \mu_i \quad (2.3)$$

Donde  $x_i \mu_i$  es la atenuación del vóxel  $i$ , como se muestra en la imagen 2.2.

### 2.3.2. Unidades Hounsfield

Por convención, y además porque los valores de atenuación lineal no difieren mucho entre los tejidos, se utilizan las unidades Hounsfield, que se definen como:

$$HU = 1000 \times \frac{\mu_t - \mu_{H_2O}}{\mu_{H_2O}}. \quad (2.4)$$

Donde  $\mu_t$  y  $\mu_{H_2O}$  son los valores de atenuación lineal (medida en un vóxel) correspondientes a cierto tejido y del agua, respectivamente. Esta magnitud nos permite presentar la imagen en escala de grises. Generalmente, una máquina de TC mostrará unidades Hounsfield en el rango de  $-2000HU$  a  $+4000HU$  [17].

Los valores positivos altos corresponden a un hueso compacto y altamente atenuado, mientras que los números negativos altos se acercan a la baja atenuación del aire. La grasa tiene un factor de atenuación ligeramente más pequeño que el agua, alrededor de  $-100HU$ . Los pulmones se encuentran entre los rangos de  $-600HU$  a  $-400HU$  [16,17].

### 2.3.3. Contraste

El contraste en una imagen se manifiesta como diferencias en los valores de escala de grises de la imagen. En una imagen de TC se define el contraste como:

$$C = \frac{I_1 - I_2}{I_1}. \quad (2.5)$$

Donde  $I_1$  y  $I_2$  son las intensidades de rayos X transmitidas a través de diferentes trayectorias de la radiación incidente. Por consiguiente, si dos tejidos tienen propiedades de absorción de rayos X muy diferentes para un valor particular de energía

de rayos X, muestran alto contraste [18].

### **2.3.4. Reconstrucción de imágenes**

La reconstrucción de imágenes de TC es el proceso por el cual se asigna la atenuación adecuada  $\mu$  para cada vóxel que se utilizó al momento de discretizar el objeto, utilizando las medidas de intensidad transmitida e intensidad irradiada para diferentes ángulos (proyecciones  $X_i$ ). Existen 2 tipos de métodos con los cuales se puede reconstruir imágenes: analíticos e iterativos [12].

#### **Reconstrucción por métodos analíticos**

Este método consiste en encontrar la solución directa del sistema de ecuaciones lineales (retroproyección). Este método utiliza la transformada de Radon, la transformada de Fourier en una dimensión y la transformada inversa de Fourier bidimensional para reconstruir el objeto.

#### **Reconstrucción por métodos iterativos**

Dentro de los métodos interactivos podemos encontrar el método iterativo algebraico ART (del inglés Algebraic Reconstruction Technique) y el método iterativo estadístico. Debido a los avances recientes en poder computacional, los algoritmos de reconstrucción iterativa (IR) se han convertido en una opción clínicamente viable en la obtención de imágenes por tomografía computarizada (TC) [19].

## **2.4. Estándar DICOM**

Digital Imaging and Communications in Medicine (DICOM) es un formato estándar abierto para codificar y transmitir imágenes médicas digitales como: Radiografías Computarizadas, Resonancias Magnéticas y Tomografías Computarizadas, fue introducida por primera vez en 1993. El formato DICOM integra y facilita la comunicación entre la adquisición de imágenes, propiedades de la radiación incidente, el archivo y los componentes del sistema de información. Cada procedimiento de adquisición de imágenes (tomografía computarizada (TC), resonancia magnética (RM), medicina nuclear (NM), etc.) contiene atributos propios correspondientes a cada modalidad [2,20,21].

### 2.4.1. Modelo de información DICOM

El Modelo de Información DICOM define la estructura y organización de la información relacionada con la comunicación de imágenes médicas. El modelo de información en imágenes DICOM está orientado a objetos. Se considera que los datos del mundo real (series, equipos, marcos de referencia, estudios, pacientes, etc.) son objetos de información con una colección de atributos o propiedades (nombre del paciente, ID, sexo, edad, peso, estado de tabaquismo, etc.).

Los objetos se definen de acuerdo con las definiciones de objetos de información (IOD) de DICOM y sirven para especificar información sobre objetos del mundo real. Un IOD no representa una instancia específica de un objeto del mundo real, sino más bien una clase de objetos del mundo real que comparten las mismas propiedades. Un IOD se divide en varios módulos que se utilizan para agrupar los atributos de un objeto de información. Por lo tanto, los módulos son utilizados por diferentes tipos de objetos, cada objeto tiene que definir si el módulo es obligatorio (M), condicional requerido (C) u opcional (U) [22–25].

### 2.4.2. Diccionario de datos DICOM

El Diccionario de datos DICOM es el registro de todos los atributos utilizados en medicina digital. Los atributos se identifican por su etiqueta de elemento de 8 dígitos hexadecimales llamado **Tag**, primera columna del cuadro 2.1. Los 4 primeros dígitos establecen el grupo y los 4 siguientes el elemento (Grupo, Elemento).

El diccionario contiene unos 2000 elementos diferentes, divididos en grupos (según la similitud del contenido del ítem). Por ejemplo, el grupo 0010 en el Diccionario de datos reúne todos los elementos relacionados con un paciente y el grupo 0028 está dedicado a la información sobre la imagen [26,27]. Los grupos, dentro del estándar DICOM, solo pueden estar relacionados con números pares, ya que los números de grupo impares están reservados para que los fabricantes almacenen sus datos patentados. La convención (Grupo, Elemento) con los cuales se identifica un atributo son muy útiles porque son etiquetas cortas. Estos tienen un tamaño fijo y siguen un formato hexadecimal muy estricto, pero no nos dan una descripción de los datos del mundo real, por lo que en la segunda columna del diccionario de datos DICOM, encontramos la columna **Name**, también conocida como nombre del atributo, la cual explica qué datos del mundo real corresponden al elemento con una etiqueta definida.

La columna **Keyword** del cuadro 2.1 es una adición reciente al estándar, con una estructura similar al Name del atributo, es una alternativa textual a las etiquetas hexadecimales.

La columna **VR** en el diccionario especifica el tipo de datos del elemento. Cada VR tiene: un nombre propio abreviado de 2 letras, una definición de lo que representa, una descripción de los caracteres permitidos en sus datos y una longitud de datos. Un ejemplo de VR es el correspondiente al elemento “Fecha de nacimiento del paciente” (0010,0030) con un nombre propio abreviado DA, es decir, como una cadena de datos AAAAMMDD de 8 dígitos (“1993082” que representa el 22 de agosto de 1993).

Por último tenemos el valor **VM** que describe el número de elementos permitidos de un atributo. Por ejemplo, el elemento “Otros nombres de pacientes” (0010,1001) puede incluir claramente más de un nombre, por lo que su valor VM es 1-n, donde n es cualquier número.

<b>Tag</b>	<b>Name</b>	<b>Keyword</b>	<b>VR</b>	<b>VM</b>
(0010,0010)	Patient’s Name	PatientName	PN	1
(0010,0020)	Patient ID	PatientID	LO	1
(0010,0021)	Issuer of Patient ID	IssuerOfPatientID	LO	1
(0010,0022)	Type of Patient ID	TypeOfPatientID	CS	1
...				
(0028,0002)	Samples per Pixel	SamplesPerPixel	US	1
(0028,0003)	Samples per Pixel Used	SamplesPerPixelUsed	US	1
(0028,0004)	Photometric Interpretation	PhotometricInterpretation	CS	1

**Cuadro 2.1:** Extracto del Diccionario de datos DICOM [27].

### 2.4.3. Jerarquía de información DICOM

Además de la clasificación por grupos de datos, que ya se tiene dentro de los diccionarios DICOM, tenemos también la estructuración de datos por jerarquía Paciente-estudio-serie-imagen. Ya que a cada paciente se le pueden programar varios estudios (por ejemplo, exámenes de resonancia magnética, tomografía computarizada y ecografías). Posteriormente podrían ser necesarios varios estudios de seguimiento. Cada estudio puede tener múltiples series de imágenes (TC, ecografía, reconstrucciones 3D, con distintos protocolos de imagen, etc.), y cada serie tendrá una o más imágenes. Para cada nivel de jerarquía dentro de este tipo de estructura existen identificadores únicos (**Tag**) [22].

#### 2.4.4. IOD de Imagen de Tomografía Computarizada

El IOD de una imagen de tomografía computarizada (TC) especifica los módulos de una imagen que ha sido creada por un dispositivo de tomografía computarizada.

	Module	Reference	Usage
Patient	Patient	C.7.1.1	M
	Clinical Trial Subject	C.7.1.3	U
Study	General Study	C.7.2.1	M
	Patient Study	C.7.2.2	U
	Clinical Trial Study	C.7.2.3	U
Series	General Series	C.8.8.1	M
	Clinical Trial Series	C.7.3.2	U
...			
Image	Common Instance Reference	C.12.2	U

**Cuadro 2.2:** Extracto de los Módulos IOD de una imagen CT [28].

# Capítulo 3

## Metodología

### 3.1. Datos

#### 3.1.1. Preprocesamiento de Datos

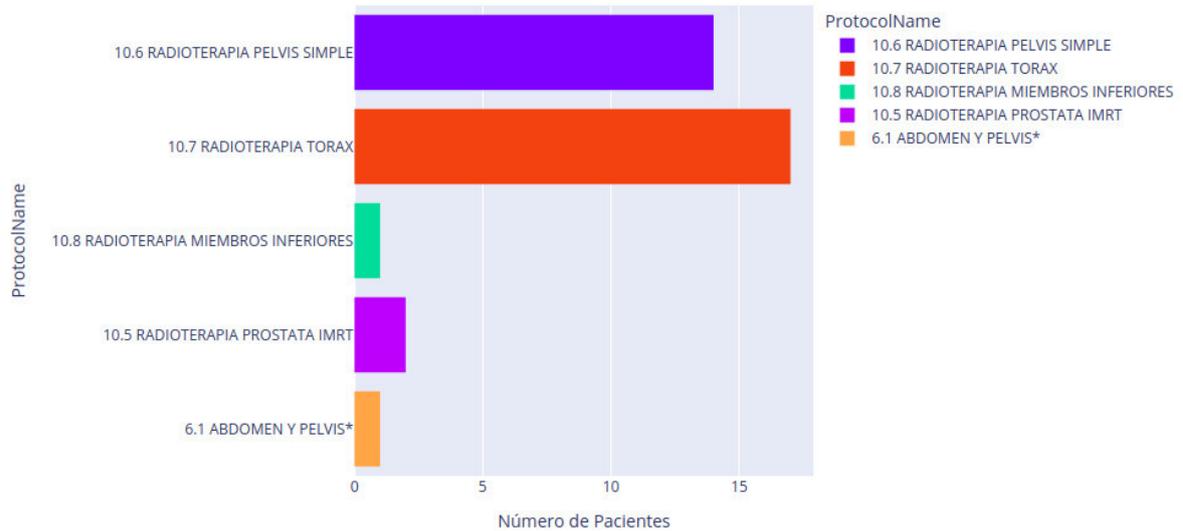
Para el preprocesamiento de las imágenes DICOM es necesario proteger la privacidad de los pacientes por razones éticas y legales. Así pues, se procede a la eliminación y reemplazo de atributos dentro de un conjunto de datos DICOM que potencialmente pueden resultar en la filtración de la información de identificación individual sobre el paciente y personas u organizaciones involucradas en el proceso de adquisición de las imágenes [29,30].

Para anonimizar las imágenes desarrollamos un código en Python utilizando los atributos del perfil de confidencialidad [31]. Algunos atributos privados fueron modificados como es el caso del Patient's Name con el tag (0010,0010). Otros datos como el Patient's Age (0010, 1010) se mantuvieron, ya que es información que podría ser útil para futuros trabajos y pueden coexistir en las imágenes sin vulnerar la identidad del paciente. La información más detallada del código diseñado se encuentra en el Apéndice A.

#### 3.1.2. Base de Datos Inicial

La base de datos inicial consta de 35 exámenes correspondientes a distintos pacientes en diferentes zonas del cuerpo, en la modalidad TC. Estas imágenes fueron adquiridas en un hospital de la ciudad de Quito-Ecuador, mediante el software iQ-VIEW / PRO versión 3.0.0. [32].

En la Figura 3.1 se observa la distribución y número de exámenes de TC en la base de datos total, según el atributo Protocol name con el tag (0018,1030), el cual es llenado por el operador con las condiciones bajo las cuales se realizó la Serie [33].



**Figura 3.1:** Histograma de la base de datos inicial según el elemento Protocol name en modalidad CT.

### 3.1.3. Características de los Datos

La base de datos utilizada para la visualización consta de 35 exámenes de TC de diferentes pacientes, mientras que, para la segmentación se utilizaron 18 exámenes. Las siguientes son las principales características del sistema de datos.

- El número de imágenes por paciente es variable, con una media de 97, un mínimo de 55 y un máximo de 771.
- El tamaño es de 512 x 512 píxeles por cada imagen para todos los casos.
- La dimensión de la resolución del escaneo (Slice Thickness) es de 5mm para 34 pacientes y de 2,5mm para un paciente.
- El tipo de estudio (StudyDescription) es de Planeamiento de Terapia para 34 pacientes y de TC Corporal para un paciente.
- La posición del paciente correspondiente a: HFS (Head First-Supine), con la cabeza primero y con la cara y el torso hacia arriba, para 34 exámenes. FFS (Feet First-Supine), con los pies primero y con la cara y el torso hacia arriba para 1 examen [34].

- La distancia entre píxeles (PixelSpacing) para todos los estudios es de  $0,976562mm$  para el eje  $x$  y  $y$ .

## 3.2. Python y librerías

Python es un popular lenguaje de programación de alto nivel, de código abierto (open source), que puede manejar varias tareas de programación como: cálculo numérico, desarrollo web, procesamiento de imágenes, programación de bases de datos, procesamiento paralelo, etc. Python se ha convertido en uno de los lenguajes interpretados más populares. Además, está disponible en sistemas operativos como Windows, Mac OS o Linux. Este lenguaje viene con varias librerías que son estándar o se pueden instalar, de acuerdo a las necesidades del proyecto. A continuación, se detallan los principales módulos que se utilizarán en el procesamiento de los archivos DICOM usados en este proyecto con una breve descripción.

- **Numpy:** Es un módulo que permite la manipulación matemática de vectores y matrices multidimensionales, que pueden ser guardados como objetos llamados `numpy.ndarray`. Dado que este proyecto se enfoca en el procesamiento de imágenes de TC y que las imágenes son arreglos, usaremos las capacidades de manipulación matemática de matrices de NumPy para su procesamiento [35,36].
- **Pydicom:** Es un módulo de Python para trabajar con archivos DICOM que permite leer, modificar y escribir datos DICOM de una manera sencilla [37].
- **Matplotlib:** Es una biblioteca para generar gráficos en 2D / 3D. Está diseñado para usar el tipo de datos NumPy y para generar imágenes a través de interfaces de usuario y sistemas operativos [38].
- **Scikit-image:** Es una biblioteca para Python con una colección de algoritmos para el procesamiento de imágenes [39].
- **SciPy:** Es una biblioteca con herramientas informáticas científicas para Python que incluyen procesamiento de señales, optimización, estadística, entre otras [40].

### 3.3. Entorno Jupyter Notebook

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) web de código abierto, que permite la creación de scripts en múltiples lenguajes de programación. La herramienta fue creada por científicos de la Universidad de Berkeley en Estados Unidos con el fin de compartir apuntes técnicos, incluir ecuaciones y notas de sus investigaciones de una manera fácil y sencilla.

El entorno permite un acceso a través del navegador donde el autor estará en capacidad de escribir en bloques de código y a su vez ir explicando estos con un contexto en Markdown, o métodos de texto enriquecido similares. Al día de hoy, uno de sus principales usos está orientado a la ciencia de datos, la inteligencia artificial, el machine learning y el procesamiento de datos masivos. Debido a su facilidad de uso se ha popularizado en todo el mundo a lo largo de los últimos años.

El entorno como tal soporta múltiples lenguajes de programación, sin embargo, para este proyecto se lo utilizará con Python. Otra ventaja que tiene el ambiente es la posibilidad de exportar el notebook, sea como archivo de extensión `.py` o en su extensión nativa `.ipynb`, aunque también tiene soporte para exportar la información contenida en un notebook en formato HTML para su lectura y compartición [41–43].

### 3.4. Software desarrollado para la obtención de Atributos DICOM para TC

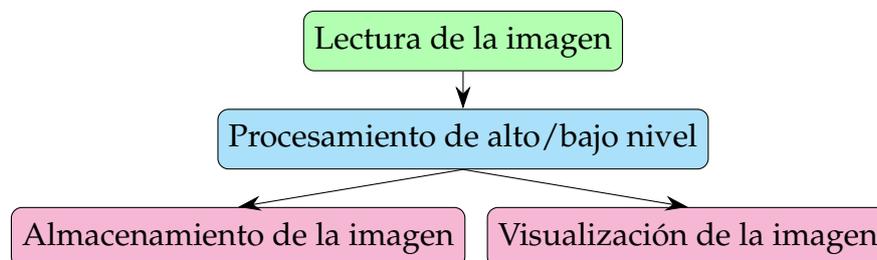
Para la visualización de los datos dentro de los archivos DICOM se diseñó el script `imprimir_informacion.py`. Este código permite la entrada de un archivo con los tags de los atributos que se desea imprimir en pantalla. El archivo puede ser elegido por el usuario, y para este proyecto se utilizaron dos archivos de entrada: el primero es el `private_tags.dat`, el cual contiene los tags correspondientes al perfil de confidencialidad que se utilizó para el proceso de anonimización, y el segundo es el `tags_CT.dat`, el cual contiene los atributos correspondientes a parámetros de adquisición de una imagen de TC. Si el usuario requiere visualizar otros atributos, para diferentes tipos de técnicas de adquisición de imágenes médicas, necesita ingresar su propio archivo con el formato de los ficheros anteriores. El script se encuentra en el Apéndice B. A continuación, se muestra la visualización de datos de una imagen DICOM con tags correspondientes a parámetros de adquisición de la imagen de TC, mediante la herramienta realizada.

Tag	Name	Keyword	Value
(0008, 0008)	Image Type	ImageType	['ORIGINAL', 'PRIMARY', 'AXIAL']
(0008, 0060)	Modality	Modality	CT
(0010, 0010)	Patient's Name	PatientName	paciente3-A
(0010, 0030)	Patient's Birth Date	PatientBirthDate	11/01/1972
(0010, 0040)	Patient's Sex	PatientSex	F
(0018, 1150)	Exposure Time	ExposureTime	1000[usec]
(0018, 1170)	Generator Power	GeneratorPower	35000[kw]
(0028, 0002)	Samples per Pixel	SamplesPerPixel	1
(0028, 0004)	Photometric Interpretation	PhotometricInterpretation	MONOCHROME2
(0028, 0100)	Bits Allocated	BitsAllocated	16
(0028, 0101)	Bits Stored	BitsStored	16
(0028, 0102)	High Bit	HighBit	15
(0028, 0103)	Pixel Representation	PixelRepresentation	1
(0028, 0010)	Rows	Rows	512
(0028, 0011)	Columns	Columns	512
(0018, 0060)	KVP	KVP	140[kV]
(0018, 1151)	X-Ray Tube Current	XRayTubeCurrent	250[mA]
(0018, 1152)	Exposure	Exposure	45[mAs]
(0018, 1190)	Focal Spot(s)	FocalSpots	0.700000[mm]
(0018, 1111)	Distance Source to Patient	DistanceSourceToPatient	541.000000[mm]
(0018, 1110)	Distance Source to Detector	DistanceSourceToDetector	949.147000[mm]
(0018, 1120)	Gantry/Detector Tilt	GantryDetectorTilt	0.000000
(0018, 1210)	Convolution Kernel	ConvolutionKernel	DETAIL
(0018, 9305)	Revolution Time	RevolutionTime	1.0[s]
(0018, 9306)	Single Collimation Width	SingleCollimationWidth	0.625[mm]
(0018, 9307)	Total Collimation Width	TotalCollimationWidth	20.0[mm]
(0018, 9309)	Table Speed	TableSpeed	27.5[mm]
(0018, 9310)	Table Feed per Rotation	TableFeedPerRotation	27.5[mm]
(0018, 9311)	Spiral Pitch Factor	SpiralPitchFactor	1.375
(0028, 1052)	Rescale Intercept	RescaleIntercept	-1024
(0028, 1053)	Rescale Slope	RescaleSlope	1
(0028, 1054)	Rescale Type	RescaleType	HU
(0018, 1030)	Protocol Name	ProtocolName	10.7_RADIOTERAPIA TORAX
(0018, 5100)	Patient Position	PatientPosition	HFS
(0020, 0011)	Series Number	SeriesNumber	3
(0020, 0012)	Acquisition Number	AcquisitionNumber	1

**Figura 3.2:** Visualización de metadata correspondiente a parámetros de adquisición de un archivo DICOM de TC, utilizando la herramienta creada.

### 3.5. Procesamiento de Imágenes

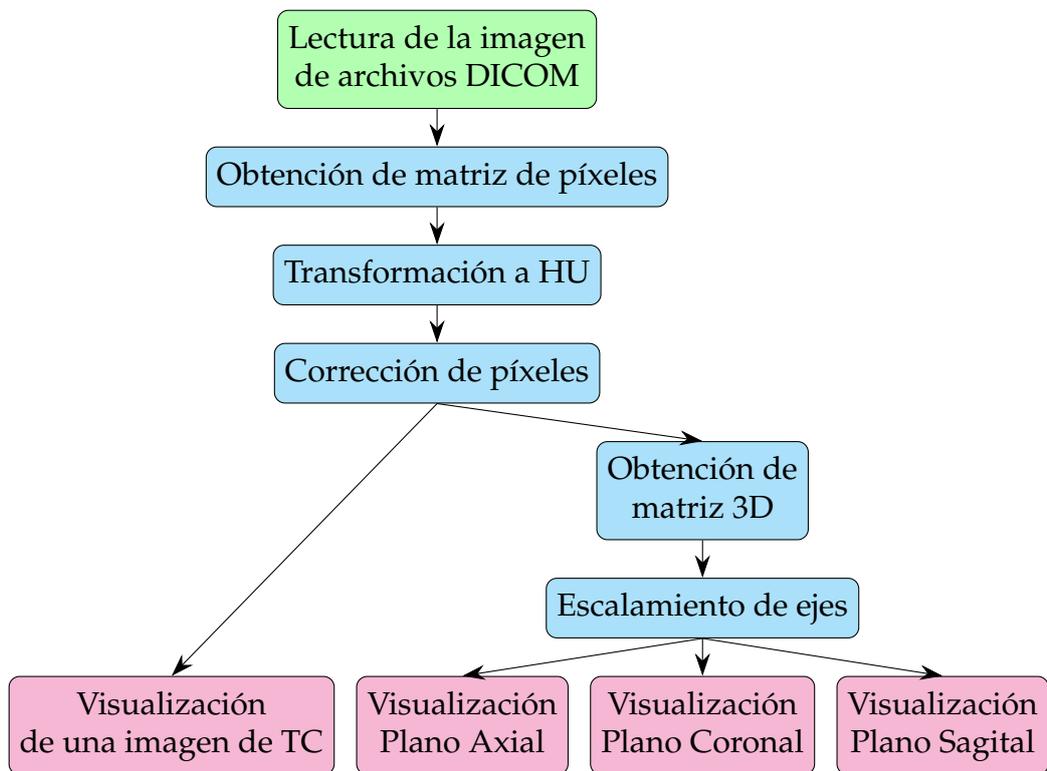
Las operaciones de procesamiento de imágenes se pueden imaginar como un flujo de trabajo similar al de la Figura 3.3. El flujo de trabajo comienza con la lectura de una imagen. Luego, la salida se procesa utilizando operaciones de bajo o alto nivel. Las operaciones de bajo nivel operan en píxeles individuales. Tales operaciones incluyen filtrado, morfología, establecimiento de umbrales, etc. Las operaciones de alto nivel incluyen detección de objetos, reconocimiento de patrones, etc. Una vez procesadas, las imágenes se guardan o se visualizan. La visualización también se puede realizar durante el transcurso del procesamiento.



**Figura 3.3:** Flujo de trabajo para el procesamiento de imágenes[35].

### 3.5.1. Procesamiento de imágenes DICOM para la visualización

Como se describió en la sección anterior, el procesamiento de imágenes consta de tres principales fases: la lectura de los datos, el procesamiento (en este caso de bajo nivel) y finalmente la visualización. Para este proyecto se considera dos tipos de imágenes de salida. La primera es de una sola imagen de TC y la segunda se basa en un examen completo, donde a su vez, se obtendrá la representación de los cortes en diferentes planos anatómicos. El flujo de trabajo para la visualización, en ambos casos, es el que se encuentra en la Figura 3.4.



**Figura 3.4:** Flujo de trabajo para la visualización de una imagen y de un examen de TC en diferentes planos anatómicos.

#### Lectura de la Imagen

La lectura de una imagen de TC a partir de archivos DICOM consiste en obtener información necesaria para la reconstrucción de la imagen, como es el caso de: los elementos que permiten el cambio de unidades, la relación entre píxeles, y el plano de coordenadas. Principalmente, la lectura radica en la extracción de una matriz de 2 dimensiones  $N \times M$ , generalmente de  $512 \times 512$  o de  $256 \times 256$ , que representa una sección transversal (axial) de la anatomía de un paciente [44].

En este caso se utilizará la biblioteca Pydicom, que abre un archivo DICOM como un objeto de la clase `pydicom.dataset.FileDataset`. Los atributos pueden ser extraídos mediante el **Tag** del atributo en cuestión o por su **Keyword**.

### Obtención de matriz de píxeles

Para esta sección se utilizará el atributo `PixelData` con el tag (07FE0, 0010), el cual contiene una matriz de  $N$  filas por  $M$  columnas, elementos especificados en los tags (0028,0010), (0028,0011), para filas y columnas, respectivamente. Cada entrada de la matriz corresponde al valor del píxel que puede ser transformado a unidades Hounsfield. La posición del elemento en la matriz simboliza la posición del componente, representado por el píxel, en coordenadas rectangulares en la imagen [45,46]. Además, se utiliza el elemento (0028,0002) llamado `SamplesPerPixel` para especificar el tipo de imagen que almacenó el instrumento (el valor 3 para imágenes en RGB o 1 para escala de grises) [47]. El atributo `PixelData` es del tipo OW (Other Word String) un tipo de dato de cadenas de 16 bits con codificación Little Endian y Big Endian [48], que puede ser leído con la función `pixel_array` de Pydicom, y que retorna la matriz con los valores de píxeles decodificados como un objeto de la clase `numpy.ndarray` de la librería NumPy. Los objetos `numpy.ndarray` representan una matriz multidimensional, que para este caso es de  $512 \times 512$  dimensiones.

### Obtención de Unidades Hounsfield

Transformar la matriz de píxeles a unidades Hounsfield es necesario, debido a que hace posible realizar un análisis de los datos con base en esta escala. Mediante una transformación lineal (ecuación 2.4) se puede obtener la medida original del coeficiente de atenuación de cada material atravesado por los rayos X, de manera que permite identificar estructuras anatómicas a través de estos valores.

El cambio de la matriz de píxeles a unidades Hounsfield se lo puede realizar mediante el atributo (0028,1054). Este atributo especifica que al realizar la transformación dada por la ecuación  $m \times SV + b$ , se obtiene la matriz de píxeles en unidades Hounsfield. El parámetro  $SV$  es la matriz con los valores de los píxeles dados por el elemento `PixelData` y las constantes  $b$  y  $m$  corresponden a los atributos con los tags (0028,1052), (0028,1053), respectivamente [49].

## Corrección de píxeles

La corrección de píxeles consiste en eliminar y corregir todo valor dentro de la matriz de píxeles que pueda generar problemas al momento de la interpretación de la imagen. Por ejemplo, reemplazar los píxeles que se encuentran fuera del rango de medición del tomógrafo por valores que correspondan al aire, o la eliminación de elementos que no pertenezcan a la anatomía del paciente, como la camilla. En la sección siguiente se discutirá con más detalle estos procesos.

Hasta esta parte del desarrollo, el flujo de trabajo es el mismo para cualquiera de las visualizaciones. Para obtener la imagen de un solo archivo DICOM, una vez obtenida la matriz de píxeles, transformada a unidades Hounsfield y con su respectiva corrección de píxeles, se procede a generar la imagen para un corte axial mediante la biblioteca Matplotlib de Python. Sin embargo, para la obtención de imágenes en diferentes planos es necesario trabajar con un examen completo de TC que cuenta con un gran número de archivos DICOM. Por lo tanto, las técnicas anteriores deben ser implementadas a cada una de las imágenes, dentro de un mismo examen y paciente. Además, se deben seguir algunos pasos extras para lograr su correcta visualización. Dichos pasos serán explicados en las siguientes secciones.

## Obtención de matriz 3D

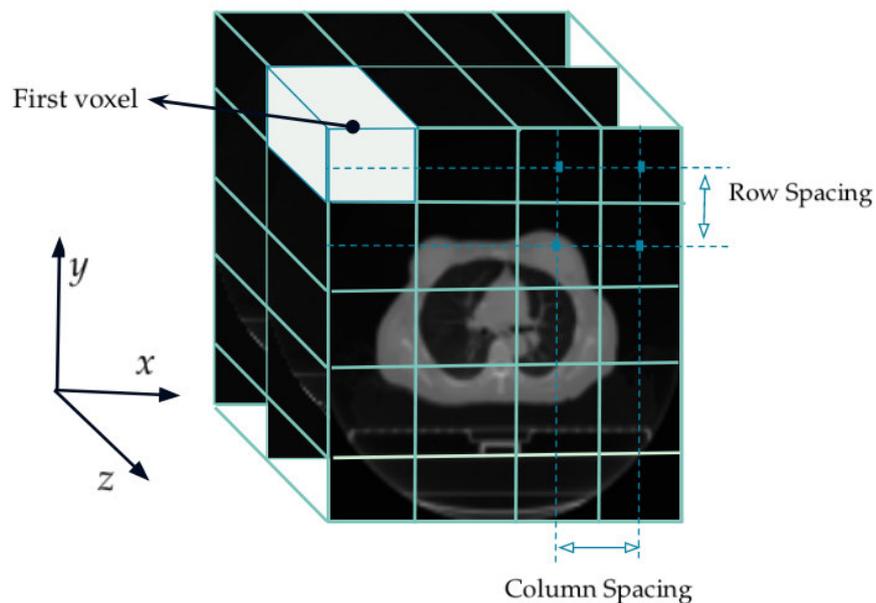
Un examen de TC consta de varias decenas de archivos DICOM. Estos cuentan con atributos similares entre cada fichero, como: las dimensiones de la matriz que contiene a la imagen, los parámetros de cambio de Unidades a Hounsfield, el espaciamiento entre píxeles en el corte axial, etc. Los atributos que se discutieron con anterioridad, excepto la matriz de píxeles, son los mismos para todos los archivos DICOM, dentro de un examen en específico de TC. Asimismo, existen valores particulares de cada corte axial, como es el caso de la posición de la imagen en el marco de referencia, número de corte, la propia matriz de píxeles, entre otros.

Para poder reconstruir una imagen en diferentes planos anatómicos, es necesario obtener una matriz en 3 dimensiones, producto del apilamiento de las matrices  $N \times M$  obtenidas de aislar la matriz de píxeles de cada archivo DICOM, después del proceso de transformación a unidades Hounsfield y de la corrección. Para un examen de  $C$  cortes, la dimensión de la matriz es de  $C \times 512 \times 512$ . En este proyecto se almacenará la matriz como un objeto de Python de la clase **numpy.ndarray** de la librería NumPy, que representa una matriz multidimensional. Para apilar las matri-

ces de forma ordenada se utiliza el atributo Instance Number con tag (0020,0013), que simboliza el orden en el cual fueron obtenidos los archivos por el instrumento [50].

### Escalamiento de ejes

Si bien la matriz de píxeles en HU especifica los valores de los coeficientes de atenuación de cada vóxel, y la posición del coeficiente dentro de la matriz da como resultado la ubicación del vóxel en la imagen, no se precisan las distancias físicas del paciente con respecto de la imagen. Por lo tanto, es necesario utilizar los atributos Pixel Spacing con el tag (0028,0030), que especifica la distancia física entre los centros de cada píxel bidimensional de la imagen axial de TC. Como se muestra en la Figura 3.5, este atributo retorna un arreglo de 2 valores con las distancias entre el centro de los píxeles en ambos ejes  $x$  y  $y$  en  $mm$ . Para el cálculo de la distancia física entre 2 cortes consecutivos (eje  $z$ ), utilizamos el elemento Image Position Patient con tag de (0020,0032). Este atributo detalla los valores de coordenadas  $x, y, z$  de la esquina superior izquierda (centro del primer vóxel transmitido) de la imagen en  $mm$  [50,51]. Si se procede a realizar la resta entre el tercer valor (eje  $z$ ) dado por el atributo Image Position de 2 cortes adyacentes se obtiene el espaciado entre vóxeles, y junto con las distancias entre cada píxel en los ejes  $x$  y  $y$  se puede reconstruir la imagen en los diferentes planos anatómicos mediante la biblioteca Matplotlib de Python, con distancias reales.

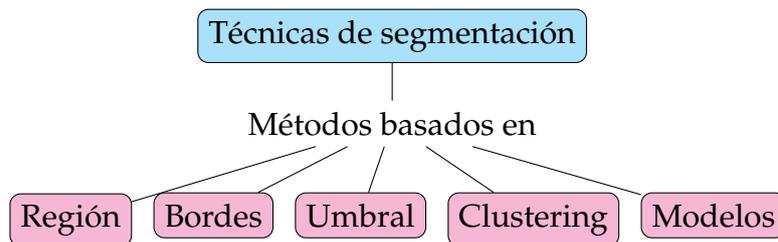


**Figura 3.5:** Esquema de la interpretación de los atributos Pixel Spacing, Image Position y de vóxel en cortes consecutivos.

### 3.5.2. Procesamiento para la segmentación de imágenes de TC

En el procesamiento de imágenes, la segmentación es la técnica que delimita áreas con características análogas dentro de una imagen. Esta técnica divide a la imagen en regiones que no se superponen. Un grupo o colección de píxeles, que tienen propiedades similares, se considera una región. El proceso de segmentación constituye un paso importante previo al desarrollo de sistemas CAD (Diagnóstico Asistido por Computadora), y es utilizada para localizar objetos en una imagen o bien encontrar bordes. La segmentación se fundamenta en las características de los píxeles de la imagen y delimita un área de interés, en términos de vóxeles o píxeles.

La segmentación es una de las tareas más desafiantes y esenciales en cualquier sistema CAD, por la gran complejidad y variabilidad de las apariencias y formas de las estructuras anatómicas en imágenes médicas. Debido a la diversidad de objetos de interés y modalidades de imagen, no existe un conjunto de características universales ni una técnica de segmentación general. Algunas técnicas populares de los diversos métodos de segmentación de imágenes se ilustran en la Figura 3.6 [52–58].



**Figura 3.6:** Técnicas de segmentación de imágenes [59].

En la Figura 3.7 se puede observar un ejemplo de imagen segmentada. Las regiones  $R_1, R_2, R_3$  y  $R_4$  se denominan objetos de la imagen y  $R_5$  es conocido como fondo. Las principales reglas que gobiernan la segmentación basada en regiones son:

- Las regiones combinadas deben ser iguales a la imagen  $I$ , entonces  $\bigcup_{i=1}^n R_i = I$
- Dos regiones no deben superponerse  $R_i \cap R_j = \emptyset$  para  $i \neq j$ .
- $R_i$  es un conjunto conectado para  $i = \{1, \dots, n\}$ .
- $Q(R_i) = \text{TRUE}$  para  $i = 0, 1, 2, \dots$ .
- $Q(R_i \cup R_j) = \text{FALSE}$  para cualquier región adyacente  $R_i$  y  $R_j$ .

Donde  $Q(R_k)$  es un predicado lógico definido sobre los puntos en el conjunto  $R_k$ . Se denomina predicado lógico a un criterio matemático que debe satisfacer cada píxel

de cada región en la imagen. Dos regiones  $R_i$  y  $R_j$  son adyacentes si su unión forma un conjunto conectado [60,61].

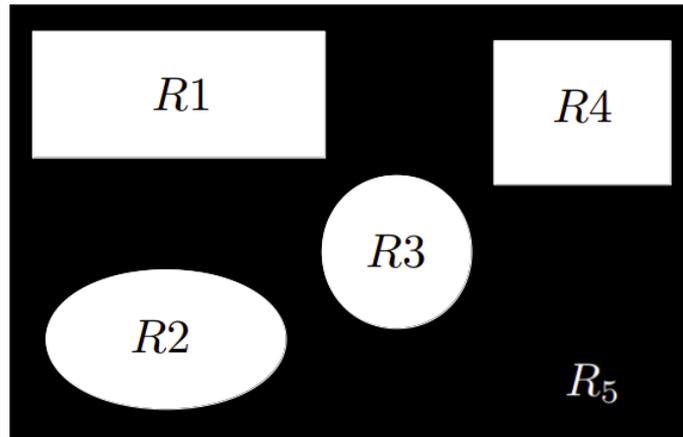


Figura 3.7: Ejemplo de segmentación basada en Regiones.

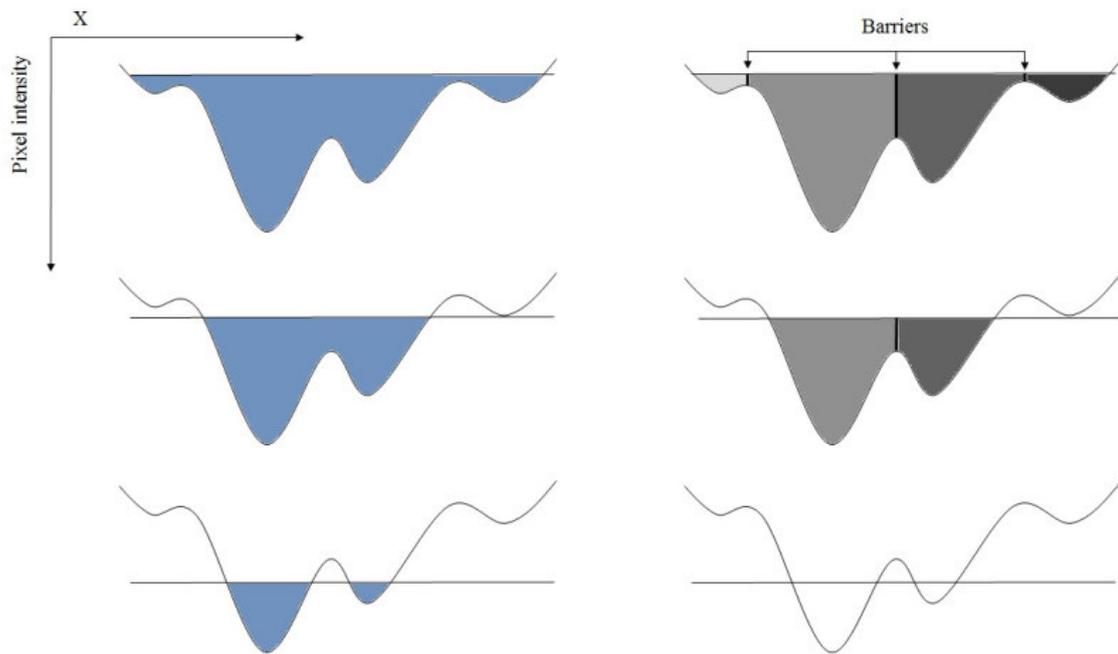
### Segmentación por el Método de Watershed General

El método de Watershed es un tipo de segmentación basada en regiones, y es reconocido como un poderoso proceso de segmentación, debido a sus muchas ventajas, incluida la simplicidad, la velocidad y la división completa de la imagen. Incluso con regiones objetivo que tienen poco contraste y límites débiles. Aunque este método es muy popular, cuenta con algunas deficiencias como la segmentación excesiva y la sensibilidad al ruido. En consecuencia, se han desarrollado y combinado diversos procedimientos y técnicas con el fin de minimizar estos problemas [5,62].

La segmentación por el método de Watershed se fundamenta en considerar el nivel de gris como una superficie topográfica, donde los valores de alta intensidad denotan picos y colinas mientras que los de baja intensidad denotan valles (mínimos locales). La idea principal es comenzar a inundar de agua los alrededores de la superficie topográfica de la imagen. Cuando el agua fusiona los picos, se construyen barreras para evitar esta fusión, hasta que todos los picos estén bajo el agua. Las barreras resultantes dan la segmentación de picos y valles en la imagen, como se ilustra en 3.8 [6,7,36,62,63].

### Segmentación por el Método de Watershed basado en Marcadores

El método de Watershed, basado en marcadores, representa una mejora sobre la transformación original y se ha demostrado que es un método robusto y flexible



**Figura 3.8:** Ilustración del funcionamiento del algoritmo Watershed [6].

para la segmentación de objetos con contornos cerrados, evitando una segmentación excesiva. La idea básica detrás de la segmentación controlada por marcadores es transformar la imagen de entrada de tal manera que las líneas divisorias de la imagen correspondan a límites de objetos significativos. La imagen transformada se llama función de segmentación. En consecuencia, la función de segmentación debe filtrarse antes de utilizar el método de Watershed para eliminar todos los mínimos irrelevantes. Los marcadores se pueden seleccionar de forma manual o automática [64–66], para este proyecto todo el proceso es automático, en especial la generación de los marcadores.

### Operaciones Morfológicas

Las operaciones morfológicas son una colección de operaciones no lineales que cambian la forma, el tamaño y los límites de los objetos en la imagen, mediante elementos estructurantes. Una operación morfológica y un elemento estructurante se definen en términos de conjuntos [60,67].

Un elemento estructurante es una matriz que caracteriza una figura, y se utiliza para sondear o interactuar con una imagen dada. Normalmente se elige un elemento estructurante del mismo tamaño y forma que los objetos que se desea procesar en

la imagen de entrada[68]. A continuación, se detallan las operaciones morfológicas que se utilizarán en este proyecto.

**Dilatación:** La dilatación permite que los píxeles de primer plano de una imagen crezcan o se expandan. Esta operación también rellena pequeños agujeros en un objeto. Además, es utilizada para combinar objetos que están lo suficientemente cerca entre sí pero que no están conectados. La dilatación de la imagen  $I$  con un elemento estructurante  $S$  se denota como  $I \oplus S$  [69,70].

**Erosión:** La erosión se utiliza para disminuir objetos en una imagen eliminando píxeles del límite de ese objeto. Esta operación es opuesta a la dilatación. La erosión de la imagen  $I$  y con un elemento estructurante  $S$  se denomina  $I \ominus S$  [69,70].

**Closing:** Es una operación morfológica compleja. Se obtienen combinando la dilatación y la erosión. Esta operación se puede realizar en imágenes binarias, en escala de grises y en imágenes a color. El closing de una imagen con un elemento estructurante, como un cuadrado o un círculo, suaviza los contornos de los objetos, elimina los pequeños agujeros en los objetos y fusiona los espacios pequeños entre los objetos. El closing se define como la dilatación seguida de la erosión de una imagen[69,70].

$$I \bullet S = (I \oplus S) \ominus S \quad (3.1)$$

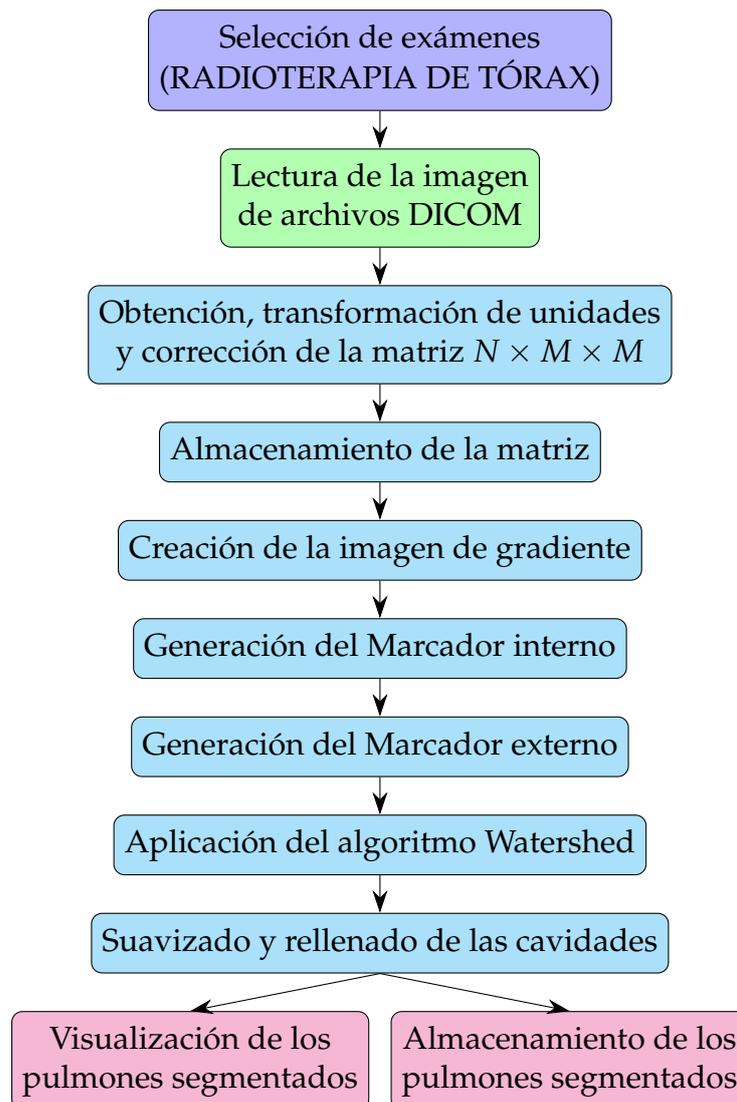
### 3.5.3. Segmentación de pulmones en imágenes de TC

El proceso que se aplicará para la segmentación es un método automático que utiliza principalmente la transformación de Watershed basada en marcadores y operaciones morfológicas.

El desarrollo de la herramienta se fundamenta en los códigos encontrados en el sitio web Kaggle [71,72], a los cuáles se les aplicaron ciertas modificaciones que logran una mejor identificación de pulmones en ciertos cortes, y cuyas bases teóricas fueron descritas por primera vez en el artículo “Automatic lung segmentation in ct images using Watershed transform” [5].

La segmentación de tejido pulmonar es una técnica que consta de las siguientes fases. Primero, se realiza la selección de exámenes correspondientes al Protocol

Name RADIOTERAPIA DE TÓRAX y ABDOMEN Y PELVIS, que comprenden las zonas donde se ubican los pulmones. Luego, se procede a lectura de los archivos DICOM. Posteriormente, se realiza el procesamiento de bajo nivel, y finalmente, se desarrolla la visualización de los pulmones segmentados. Alternativamente, se puede guardar la matriz resultante en formato .npy, el cual es un tipo de archivo exportado por la biblioteca NumPy. El flujo de trabajo se puede ver en la Figura 3.9.



**Figura 3.9:** Flujo de trabajo para la segmentación de tejido pulmonar en imágenes de TC de tórax.

### Selección de exámenes

Para realizar la segmentación de tejido pulmonar se utilizarán los exámenes que comprenden la zona torácica. Con este fin, se realizó un filtraje de las imágenes por medio del atributo Protocol name con el tag (0018,1030), un atributo llenado por el

operador con las condiciones bajo las cuales se realizó la Serie [33]. Se eligieron las imágenes con el elemento 10.7 RADIOTERAPIA TORAX, utilizadas en la planificación del tratamiento de radioterapia, y que consta de un total de 17 pacientes como se muestra en la Figura 3.1. También, se utilizará el elemento 6.1 ABDOMEN Y PELVIS de un TC corporal. De este último examen se cuenta solamente con una muestra que contiene 771 imágenes .

### **Lectura de la imagen**

La lectura de los datos de un archivo DICOM para la segmentación sigue el mismo proceso que se realizó para la visualización. Asimismo, se extraen algunos atributos que son necesarios para la reconstrucción de la imagen.

### **Obtención, transformación y corrección de la matriz 3D**

En esta fase del proceso de segmentación se obtiene una matriz de  $N \times M \times M$  dimensiones producto de la lectura de la matriz de píxeles de cada uno de los archivos DICOM de un mismo examen y de su posterior apilamiento, que llamamos matriz 3D. Luego, se realiza el cambio de unidades a Hounsfield y, finalmente se ejecuta la corrección de píxeles de una manera similar a la que se realizó para el proceso de visualización. Así, se obtiene como resultado un objeto `numpy.ndarray` que contiene todas las imágenes obtenidas del examen de tomografía computarizada en forma de una matriz. Este proceso permite una manipulación más accesible de los exámenes de TC y la utilización de la biblioteca NumPy con las ventajas que conlleva utilizar sus funcionalidades.

### **Almacenamiento de la matriz**

Este proceso permite que la matriz sea almacenada en un tipo de archivo liviano que contiene todas las imágenes correspondientes a un examen de TC en unidades Hounsfield, sin pérdida de píxeles. Además, se elimina el hecho de tener que manejar archivos DICOM, por lo que el procesamiento es mucho más rápido. El formato que se utilizará es el NPY. Un tipo de archivo de matriz NumPy creado por el paquete de software Python con la biblioteca NumPy. Los archivos NPY almacenan todos los datos necesarios para reconstruir una matriz en cualquier computadora. Esto incluye información de tipo y forma que permite compartir la información de una manera más fácil, utilizando menos recursos que con los archivos DICOM. Evi-

dentamente, este formato no comprende toda la información que posee un archivo DICOM ya que únicamente permite el almacenamiento de las imágenes de TC y no toda la información que contienen los archivos DICOM. Sin embargo, para procesamiento de imágenes es muy útil[73].

### Creación de la imagen de gradiente

Para esta parte del procesamiento, se calcula el gradiente de la intensidad de la imagen en cada píxel, en la dirección  $x$ ,  $y$  o en ambas direcciones, como se define en la ecuación 3.2. Aquí,  $f$  denota la intensidad de una imagen digital en el punto  $(x, y)$ . Las imágenes resultantes tienen la misma dimensión que la original y muestran cómo varía la función de intensidad de una imagen en cada punto analizado. En consecuencia, se encontrarán los bordes existentes en la imagen original.

$$\nabla f(x, y) \equiv \text{grad}[f(x, y)] \equiv \begin{bmatrix} g_x(x, y) \\ g_y(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \quad (3.2)$$

Para calcular el gradiente en la imagen se utilizará uno de los filtros de primera derivada más populares, el filtro de Sobel. Este filtro será aplicado en la imagen de TC pulmonar original, tanto en la dirección horizontal como en la dirección vertical. El operador usa dos núcleos de  $3 \times 3$  que están convolucionados con la imagen original para calcular aproximaciones de las derivadas. En la ecuación 3.3 se muestran los kernels que utiliza el operador de Sobel, donde  $*$  simboliza la operación de convolución de procesamiento de señal bidimensional y  $A$  representa la imagen original.

$$\mathbf{g}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A} \quad \text{y} \quad \mathbf{g}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad (3.3)$$

El filtro de Sobel generará 2 imágenes con las derivadas parciales en la dirección  $x$  y  $y$ . Estas imágenes son utilizadas para encontrar la magnitud del vector de gradiente en un punto  $(x, y)$  que viene dado por su norma de vector euclidiana, cuyo resultado representa la imagen de gradiente [60,74,75].

$$\|\nabla f(x, y)\| = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \quad (3.4)$$

## Generación del Marcador interno

Para realizar la segmentación de pulmones se utilizará el algoritmo de Watershed basado en marcadores, por lo que es necesario encontrar 2 tipos de marcadores: el interno y el externo. El marcador interno se caracteriza por ser una zona donde se tiene certeza de que pertenece a la región que se desea segmentar o región de interés (ROI), que en este caso corresponde a los pulmones, mientras que el marcador externo corresponde a la zona que se sabe que pertenece al fondo. Al etiquetar la imagen con estas dos regiones se obtendrá una zona de algunos píxeles de anchura, donde se desconoce a que región pertenece y que se encuentra rodeando a los pulmones. Es precisamente en esta región en la que se implementará el algoritmo de Watershed y que producirá una segmentación de pulmones. Mediante este método se logra eliminar mínimos triviales que originan segmentación excesiva.

Para generar el marcador interno se utilizan los valores respectivos en unidades Hounsfield, que corresponden a la zona pulmonar. En las imágenes de TC el aire aparecerá con una intensidad media de aproximadamente  $-1000$  Unidades Hounsfield ( $HU$ ). Mientras que los pulmones tienen valores que van desde  $-600HU$  hasta  $-400HU$  [13]. La pared torácica, la sangre y los huesos serán mucho más densos, muy por encima de  $-400HU$ . Así pues, para especificar los marcadores internos, se seleccionan las regiones con valores de píxeles inferiores a  $-400HU$  [5,76]. Además, se eliminarán todos los objetos que erróneamente se clasifiquen como pulmones y que no lo son, como es el caso de partes de la camilla, el fondo, la tráquea, etc. Este proceso se detalla más adelante, en la sección de tratamiento de imágenes.

## Generación del Marcador externo

El marcador externo se ubica en el fondo. En este caso corresponde a zonas que no pertenecen a la región pulmonar, que incluye la zona torácica y el aire. Por la definición y por como se encontró el marcador interno el borde del pulmón se ubica en el área de una vecindad del marcador interno. Por lo tanto, para encontrar el marcador externo, dilataremos el marcador interno, mediante dos elementos estructurantes circulares de diferentes proporciones. Luego, intersecaremos los resultados de las diferentes dilataciones y el producto lo denominaremos como marcador externo, lo que generará una tira alrededor del pulmón que debe ser de al menos 45 píxeles para cubrir todos los mínimos en el vecindario.

## **Aplicación del algoritmo Watershed**

Una vez que encontramos ambos marcadores procedemos a etiquetar la imagen según las regiones que especifican cada marcador, lo que generará una imagen con 3 zonas principales: el marcador interno, el marcador externo y la zona denominada marcador Watershed, que no es más que la región donde se aplicará el algoritmo. Mediante este proceso se delimita la zona en la cual es difícil identificar a que región pertenece y que el algoritmo tendrá que definir. La transformación de Watershed encuentra los bordes más significativos en la imagen de gradiente entre los marcadores o también llamado marcador de Watershed, logrando así, que no se produzca una sobre segmentación, que es uno de los problemas comunes al utilizar el algoritmo general.

## **Suavizado y relleno de las cavidades**

La región pulmonar segmentada, después de la aplicación del algoritmo Watershed, excluye estructuras densas, como: nódulos pulmonares y vasos hiliares que se encuentran en el borde del pulmón, debido a que estas regiones poseen diferentes coeficientes de atenuación. Esto produce diferencias entre las unidades Hounsfield de los pulmones y de estas estructuras. Por lo tanto, en la imagen de gradiente se crean bordes que al momento de aplicar el algoritmo clasifica estas estructuras como fondo cuando no lo son. En consecuencia, es necesario suavizar y rellenar las cavidades creadas por el algoritmo producto de la exclusión de nódulos y vasos hiliares ubicados en el borde de la región pulmonar[5].

En morfología y procesamiento de imágenes digitales, la transformación Black-Tophat es una operación que se utiliza para extraer pequeños elementos y detalles de imágenes dadas, y se define como la diferencia entre el closing y la imagen de entrada. Así pues, se utilizará esta operación morfológica para incluir las estructuras que fueron omitidas en la segmentación, y que generan hendiduras en el borde del contorno de los pulmones, en consecuencia se suavizan y se rellenan las cavidades existentes en los límites de los pulmones de la imagen.

# Capítulo 4

## Tratamiento de imágenes de TC

### 4.1. Visualización de una imagen de TC

En la visualización de una imagen de TC de archivos DICOM se sigue el flujo de trabajo que se muestra en la Figura 3.4 y se ejecutan los pasos que se detallan en las secciones siguientes.

Primero, se realiza la lectura de los datos necesarios para la reconstrucción de un corte en el plano axial de una tomografía computarizada. Se extrae la matriz de píxeles y ciertos valores que nos permiten definir la imagen mediante la biblioteca pydicom. La biblioteca permite abrir un archivo DICOM como un objeto de la clase `pydicom.dataset.FileDataset` y obtener los atributos del archivo. Para el caso de prueba se obtuvieron los resultados de la tabla 4.1, donde se especifica el tag de grupo y de elemento de los atributos, así como también el keyword, el tipo de dato encontrado y su respectivo valor.

Tag	Keyword	Tipo	Valor
(07FE0, 0010)	PixelData	<code>numpy.ndarray</code>	matriz (512 × 512)
(0028,0010)	Rows	<code>int</code>	512
(0028,0011)	Columns	<code>int</code>	512
(0028,0002)	SamplesPerPixel	<code>int</code>	1

**Cuadro 4.1:** Valores para definir la matriz de píxeles de un archivo DICOM de la base de datos.

Una vez obtenida la matriz de píxeles, esta es guardada como un objeto de la clase `numpy.ndarray`, que representa una matriz multidimensional que para la imagen en cuestión es de dimensión 512 × 512. Luego, se procede al cambio de unidades. Según como se describió en el flujo de trabajo en el capítulo anterior, la transforma-

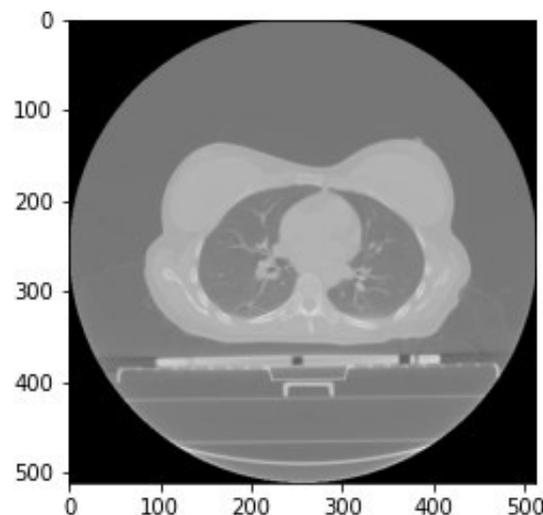
ción que permite el cambio a unidades Hounsfield es la especificada en la ecuación 4.1, cuyas constantes para la transformación son extraídas del archivo DICOM. Los valores para la imagen que estamos estudiando se encuentran tabulados en la tabla 4.2.

$$HU = m \times SV + b = (SV - 1024)[HU] \quad (4.1)$$

Tag	Keyword	Tipo	Valor
(0028,1052)	RescaleIntercept	float	-1024
(0028,1053)	RescaleSlope	float	1
(0028,1054)	RescaleType	str	HU

**Cuadro 4.2:** Valores para definir el cambio de unidades de la matriz de píxeles para un archivo DICOM de la base de datos.

Aplicando la transformación anterior y reemplazando los valores conseguidos del archivo, se obtiene la matriz de píxeles en unidades Hounsfield. Utilizando la biblioteca Matplotlib podemos obtener la imagen de TC, con un escalamiento automático de unidades Hounsfield en escala de grises, ver Figura 4.1. Con esta transformación a los valores de gris, se les asigna propiedades físicas conocidas y que permiten la interpretación del material dentro de una imagen [46].

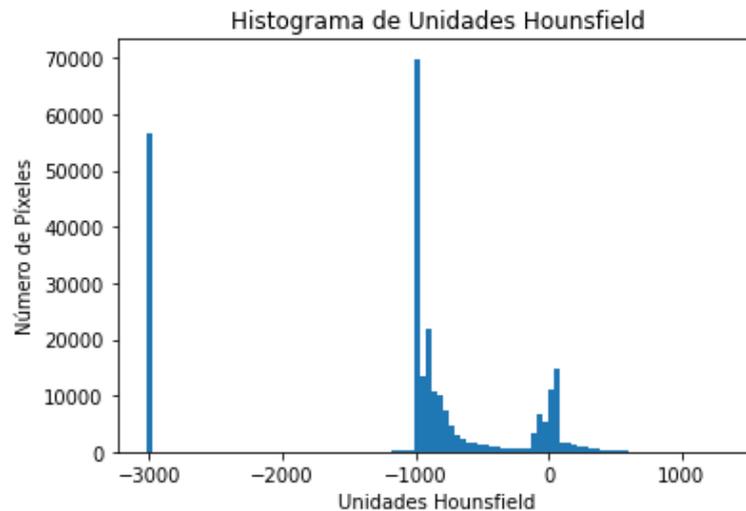


**Figura 4.1:** Visualización de una imagen de TC de tórax en escala de grises, a partir de la matriz de píxeles en unidades Hounsfield, con 512 filas y 512 columnas.

Si observamos la imagen obtenida en la Figura 4.1, se puede apreciar un círculo alrededor del cuerpo del paciente, ya que se almacenaron valores que no correspon-

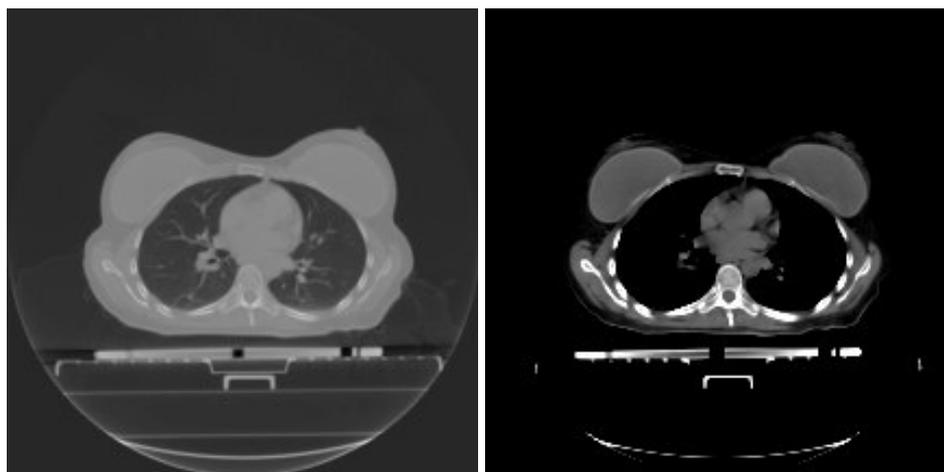
den a una medida hecha por el instrumento. Este valor es arbitrariamente puesto por el dispositivo, porque esta zona se encuentra fuera del rango de medición. El círculo aparece debido a que algunos escáneres tienen límites de escaneo cilíndricos, pero la imagen de salida es cuadrada [71]. El valor se encuentra fuera de los valores típicos de unidades Hounsfield en una imagen de TC, que son de alrededor de  $-2000HU$  a  $4000HU$  [17,77]. De ahí que la imagen necesita una corrección a ciertos valores dentro de la matriz. A esta fase del procesamiento la llamamos corrección de píxeles.

Para identificar el valor almacenado que se encuentra fuera del rango se realizó un histograma con los valores de píxel en unidades Hounsfield de la imagen de TC ver Figura 4.2, donde se encontró que el elemento fuera del rango es exactamente de  $-3024HU$ . Dicho valor no corresponde a alguno medido por el tomógrafo.



**Figura 4.2:** Histograma con los valores de píxel en unidades Hounsfield de una imagen de TC. Se demuestra que existe una gran cantidad de píxeles en el valor de  $-3024HU$ , que está fuera del rango de medida de un tomógrafo.

Una vez encontrado el valor, se procedió a reemplazarlo con el valor de atenuación del aire, que es de alrededor de  $-1024HU$ . Para especificar que esta región pertenece a aire y así no perder escalas de grises al momento de la visualización, la escala de grises comprende los valores  $[0, 250]$ .



(a) Rango completo de HU.

(b) Rango seleccionado de HU.



(c) Imagen de TC obtenida del software iQ-LITE 3.0.0

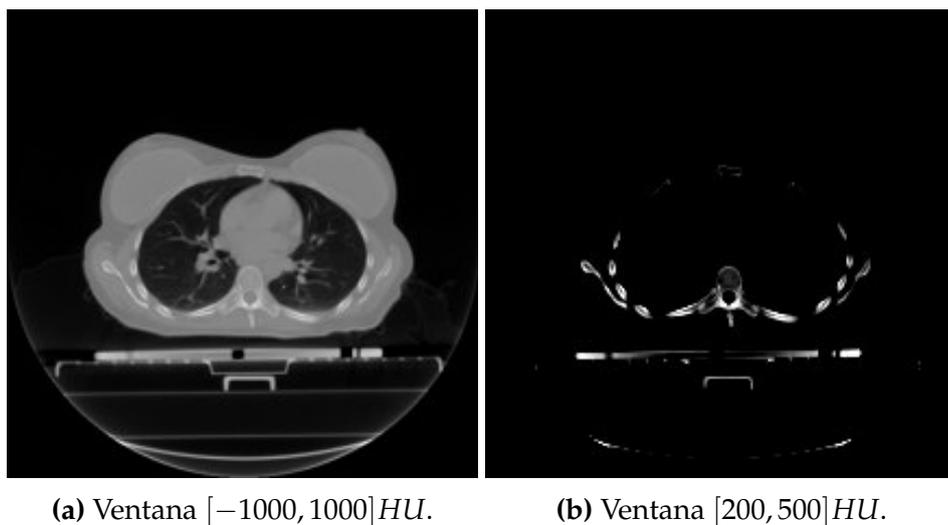
**Figura 4.3:** Visualización de una imagen de TC de tórax con corte axial de una paciente femenina de 48 años de edad, mediante diferentes métodos de visualización. (a) Visualización de la imagen utilizando el rango completo de unidades Hounsfield y adaptándolos automáticamente a la escala de grises. (b) Visualización de la imagen utilizando la ventana con valores de  $[-100HU, 300HU]$ . (c) Visualización a través del software incorporado en el Instrumento de TC (iQ-LITE 3.0.0).

En la Figura 4.3 se muestra una imagen de TC con corrección de píxel por 3 métodos diferentes de visualización. La imagen de tomografía computarizada utilizada para la Figura 4.3a comprende los valores de  $[-1460HU, 1349HU]$ , por lo que la escala Hounsfield se extiende a lo largo de 2809 unidades. Cuando se procede a realizar la visualización, la biblioteca Matplotlib distribuirá la escala Hounsfield de manera uniforme, para que comprenda todos los valores que abarca la escala de grises. Por lo tanto, cada material dentro de la imagen es difícilmente distinguible, ya que cada tonalidad de gris en la imagen comprende 11 unidades Hounsfield, lo

que implica la pérdida de información al momento de la visualización. Para evitar esta pérdida de información, la imagen puede ser representada mediante una escala de grises en un fragmento de los valores de unidades Hounsfield. Esta técnica es llamada visualización por ventana. Las ventanas, permiten en definitiva, extraer la información de la matriz de píxeles mostrando sólo rangos específicos que comprendan los valores del órgano o tejido que se desee estudiar [78].

En la imagen 4.3a la visualización se realiza mediante la biblioteca Matplotlib de Python. Se utilizan todos los rangos que existen de unidades Hounsfield en la imagen, mientras que en 4.3b se utiliza la ventana que contiene rangos que van desde  $[-100HU, 300HU]$ , los valores por debajo del mínimo valor del rango, pasan a ser el valor mínimo en la escala de grises y los valores que sobrepasen  $300HU$  se convertirán en el valor máximo de la escala, los valores entre el rango de  $-100HU$  y  $300HU$  se convertirán automáticamente a su correspondiente en la escala de grises. La imagen 4.3c corresponde a la visualización realizada por el software incorporado en el instrumento de TC.

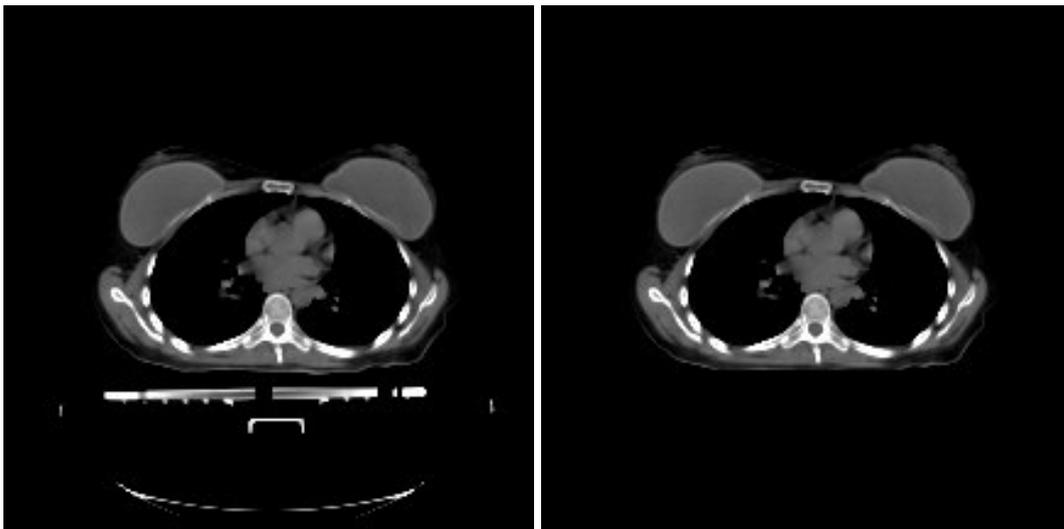
El corte de los rangos para la visualización que se realizó en la imagen 4.3b, se llevó a cabo con el fin de contrastar la imagen producida por el Software de TC (iQ-LITE 3.0.0) y la imagen producto de la manipulación de la matriz de píxeles obtenida del archivo DICOM. Cabe destacar que el corte de rangos afecta únicamente a la visualización, manteniendo la información de la imagen (matriz de píxeles) íntegra y sin modificación después del proceso de anonimización y de escalamiento a unidades Hounsfield.



**Figura 4.4:** Visualización de una imagen de TC de tórax con corte axial, con diferentes ventanas. (a) Visualización de la imagen utilizando la ventana de  $[-1000, 1000]HU$ . (b) Visualización de la imagen utilizando la ventana de  $[200, 500]$ .

En la Figura 4.4 se aplicaron diferentes ventanas a una imagen de TC. La imagen del lado izquierdo muestra una ventana que engloba los rango de  $-1000HU$  a  $1000HU$ . Esta gama cubre los materiales aire, tejido blando y hueso. La imagen de la derecha muestra exactamente el mismo corte. El rango mostrado ahora es de  $200HU$  a  $500HU$ , que muestra el hueso cortical [46].

Por motivos de visualización, y para el posterior procesamiento de las imágenes, es necesario eliminar cualquier elemento que no pertenezca a estructuras anatómicas. Para esto, se visualizaron todas las imágenes del examen de un paciente, y se encontró que el único elemento que no pertenece a estructuras anatómicas es la camilla. Por lo tanto, se procedió a removerla y a reemplazarla con aire ( $-1024HU$ ). El método de remoción de camilla se centra en identificar la ubicación de los píxeles que corresponden a la misma, mediante la visualización de una de las imágenes se descubrió que los píxeles que se encuentran dentro del rango en  $y = \{370, \dots, 512\}$  pertenecen a la camilla por lo que en esta zona se asignarán unidades Hounsfield correspondientes al aire. En la Figura 4.5 se observa el resultado de este proceso.



(a) Imagen Original de TC

(b) Imagen de TC recortada la camilla

**Figura 4.5:** Comparación entre la imagen original y la imagen final producto del proceso de remoción de la camilla para una imagen de TC. La visualización sigue el estilo del software iQ-LITE 3.0.0.

El código realizado para la remoción de la camilla permite obtener las imágenes que se muestran en la Figura 4.5b. Siguiendo el flujo de trabajo detallado con anterioridad, se diseñaron diferentes funciones que realizan cada una de las fases: lectura de la imagen y datos del archivo DICOM, obtención de la matriz, transformación de unidades, corrección de píxeles y la visualización. Todas estas funcio-

nes fueron guardadas en el archivo `imprimir_informacion.py`. Además, este script permite que el usuario pueda escoger la ventana (rangos de datos *HU*) que desea visualizar.

#### 4.1.1. Visualización Plano Axial

Para la visualización de las imágenes de TC en diferentes planos anatómicos se implementaron los 4 primeros pasos que se realizaron para el proceso de visualización de una sola imagen. La diferencia para este caso radica en que el desarrollo es iterativo y debe ser aplicado para todos y cada uno de los archivos DICOM que pertenecen a la misma serie. La obtención, transformación y corrección de la matriz de píxeles es llevada a cabo para cada una de las imágenes de un mismo examen. Luego, se procede a generar la matriz 3D; que es guardada como un objeto `numpy.ndarray`, que permite acceder de una forma fácil a cualquiera de los planos mediante las funciones que posee `numpy` para la manipulación de matrices multidimensionales. Para este examen se tiene una dimensión de  $63 \times 512 \times 512$ . Además, se obtienen las relaciones entre los ejes que se precisó en la sección escalamiento de ejes. Para el examen que se está analizando se obtuvieron los valores que se detallan en el cuadro 4.3.

	Tag	Keyword	Valor
Primera imagen	(0028,0030)	PixelSpacing	$[0,976562, 0,976562]$ [mm]
	(0020,0032)	ImagePositionPatient	$[-250,000, -250,000, 200,000]$ [mm]
	(0020,0013)	InstanceNumber	1
Segunda imagen	(0028,0030)	PixelSpacing	$[0,976562, 0,976562]$ [mm]
	(0020,0032)	Image Position	$[-250,000, -250,000, 195,000]$ [mm]
	(0020,0013)	InstanceNumber	2

**Cuadro 4.3:** Elementos que definen los valores de escalamiento en imágenes consecutivas para dos archivos DICOM de la base de datos, el atributo `InstanceNumber` determina el orden de las imágenes respecto de su adquisición.

Generalmente, la distancia entre píxeles en los ejes  $x$  y  $y$  son iguales como se observa en el cuadro 4.3, con los valores  $[0,976562, 0,976562]$  [mm], respectivamente para  $x, y$ . Estas coordenadas corresponden a los cortes axiales, a diferencia de la vista sagital y coronal, ya que la distancia entre cortes (espaciamento entre píxeles eje  $z$ ) suele variar dependiendo del tipo de examen que se realice, siendo mucho menor para exámenes que se basen en diagnóstico y un valor mayor para exámenes de planificación de terapia. El espaciamento en el eje  $z$  también depende de la zona en que se practique la tomografía y es calculado mediante la resta del tercer elemento

del atributo Image Position que detalla la posición en el eje  $z$ , del centro del primer vóxel transmitido. Esto da como resultado que la distancia entre píxeles en el eje  $z$  del examen analizado sea de  $5mm$ .

Para el caso de la tabla 4.3, se tiene que cada píxel tiene una dimensión de  $0,976562[mm]$  en los ejes  $x$  y  $y$ , por lo que la imagen en total representa  $50cm \times 50cm$ , para un corte axial. Para el eje  $z$ , se tiene que la distancia entre cortes consecutivos es de  $5[mm]$  y tiene un total de 63 cortes, por lo que la dimensión física de la imagen es de  $31,5cm$ . Las relaciones de los ejes para el caso estudiado son los siguientes:

$$\begin{aligned} \frac{x}{y} &= 1 \\ \frac{z}{x} &= \frac{5}{0,976562} = 5,120 \\ \frac{z}{y} &= \frac{5}{0,976562} = 5,120 \end{aligned} \quad (4.2)$$

Las correlaciones que se expresan en 4.2, son específicas de este examen y para este paciente, aunque la mayoría de los exámenes estudiados presentan las mismas correlaciones. El código que se implementó extrae automáticamente los valores y los calcula para cada examen, por lo que el algoritmo es independiente del tipo de relaciones que se tenga entre cada eje.

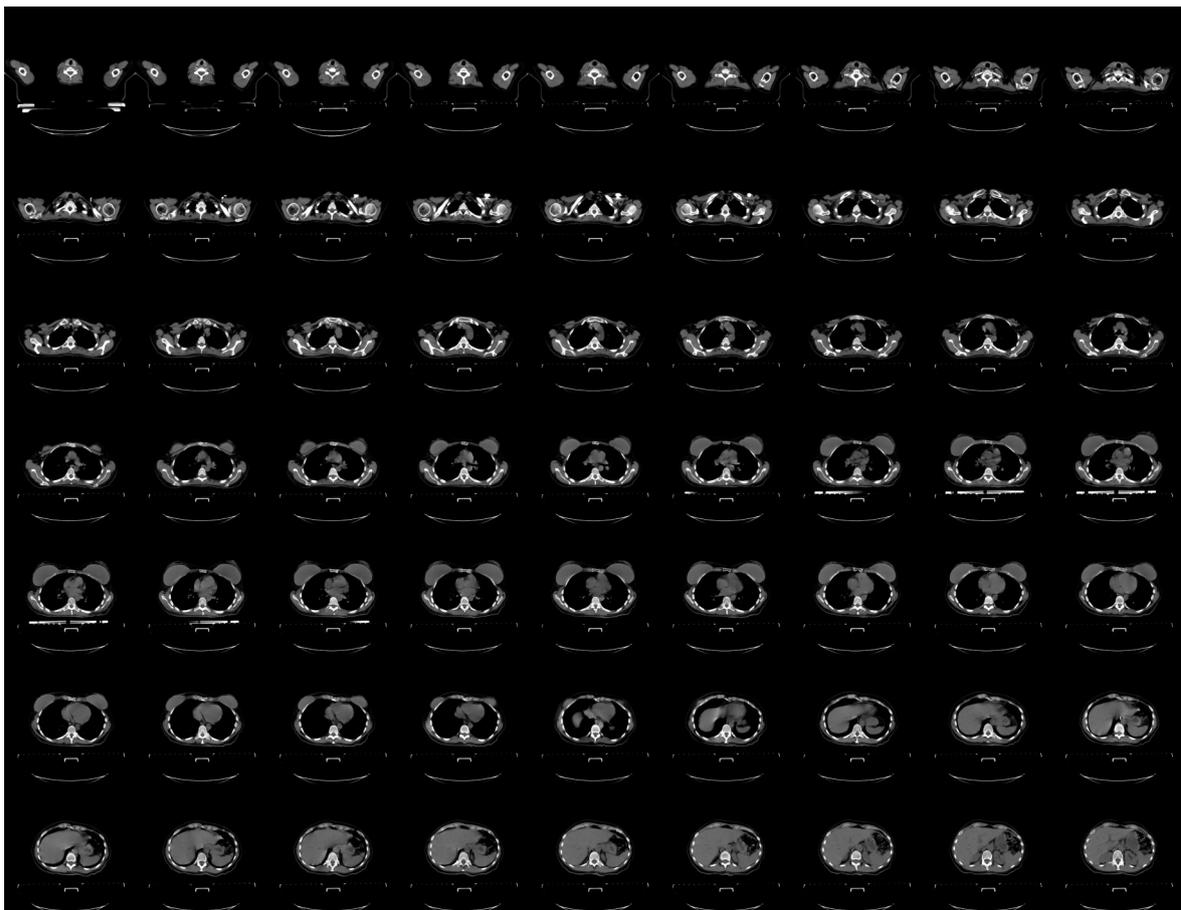
Una vez que se obtiene la matriz en 3D como objeto `numpy.ndarray`, se procede a graficar mediante la biblioteca `Matplotlib`. Para el plano axial se toman los ejes  $x$  y  $y$ . Luego, se le aplica el escalamiento de los ejes, que en este caso es  $x/y = 1$ . Este proceso es también automatizado mediante el código y sigue el flujo de trabajo de la Figura 3.4 para la generación de la visualización en el plano axial. Cabe señalar que el código se lo realizó de tal manera que todos los datos son adquiridos de los archivos. Por lo tanto, puede ser utilizado para cualquier imagen de TC, y no está asociado precisamente a los archivos que se utilizaron para este proyecto, sino que puede usarse de un modo general para cualquier examen de TC.

Con el fin de simular a los paquetes de software que se implementan en los dispositivos de TC, el objetivo principal de este método consiste en no disminuir el tamaño de la imagen, y que sea guardada sin reducir la información, con el fin de poder manipular la imagen en un formato común como es el PNG. Este proceso permite exportar la imagen para la visualización por personal médico. La imagen no pierde el número de píxeles iniciales. Además, este procesamiento transforma los cortes pertenecientes a un examen, que en este caso es de 63 cortes, en una sola

imagen, distribuyendo los cortes en 9 columnas por 7 filas. La distribución de las imágenes puede ser elegida por el usuario, pero también depende del número de imágenes de las que se dispone.

La imagen final es guardada automáticamente por el programa en formato PNG. Para el caso estudiado en la Figura 4.6, se obtuvo una imagen con un tamaño de  $4608 \times 3584$  píxeles que corresponde exactamente a la multiplicación de  $512 \times 9$  y  $512 \times 7$ , que es la distribución de cada uno de los cortes en la imagen final. Por lo tanto, se puede decir que al realizar esta visualización no se ha perdido información con respecto al número de píxeles.

En la Figura 4.6 se puede encontrar el resultado de la implementación del proceso de visualización de un examen de TC en el plano axial para un examen de 63 imágenes, con las propiedades que están descritas en el cuadro 4.3 y con la relación entre ejes calculada en la ecuación 4.2 para el eje  $x/y$ .

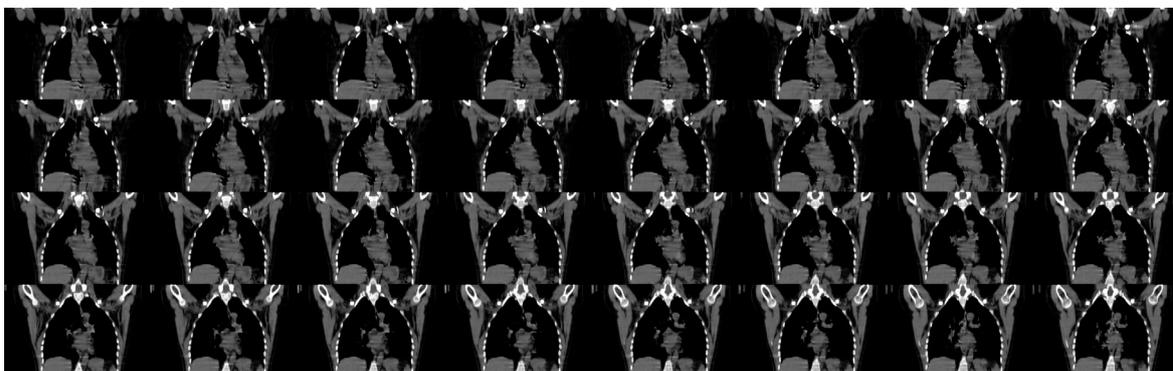


**Figura 4.6:** Plano axial de un examen de TC a partir de archivos DICOM con un total de 63 cortes en el plano axial. La visualización de la imagen sigue el estilo del software iQ-LITE 3.0.0.

### 4.1.2. Visualización Plano Coronal

En esta imagen el flujo de trabajo es el mismo que para el plano axial. La principal diferencia radica en que los ejes que se tomarán en cuenta para la gráfica son el eje  $x$  y el eje  $z$ . También, se debe implementar la relación entre los ejes que se encuentra descrita en la ecuación 4.2, específicamente, la relación  $z/x = 5,120$ . Finalmente, mediante la biblioteca Matplotlib, se obtendrá la vista de los cortes en el plano coronal como se puede observar en la Figura 4.7.

Para la presentación de este plano no se toman en cuenta todas las imágenes generadas, debido a que son un total de 512. El total de imágenes corresponde al número de píxeles en el corte axial para el eje  $y$ . Además, muchos de los cortes no contienen ningún tipo de fisiología o información que pueda ser útil. Para la imagen que se muestra en 4.7, se grafica solamente el 6,25% del total que corresponde a 32 imágenes distribuidas en 8 columnas por 4 filas. El código diseñado posibilita que el usuario pueda introducir el porcentaje de imágenes que va a obtener como salida. También, permite seleccionar los rangos de unidades Hounsfield para la presentación de la imagen sin afectar la información dentro de la matriz, tal y como se describió en la sección 3.5.1. (Procesamiento de imágenes DICOM para la visualización). Todo este proceso también aplica para el plano sagital.



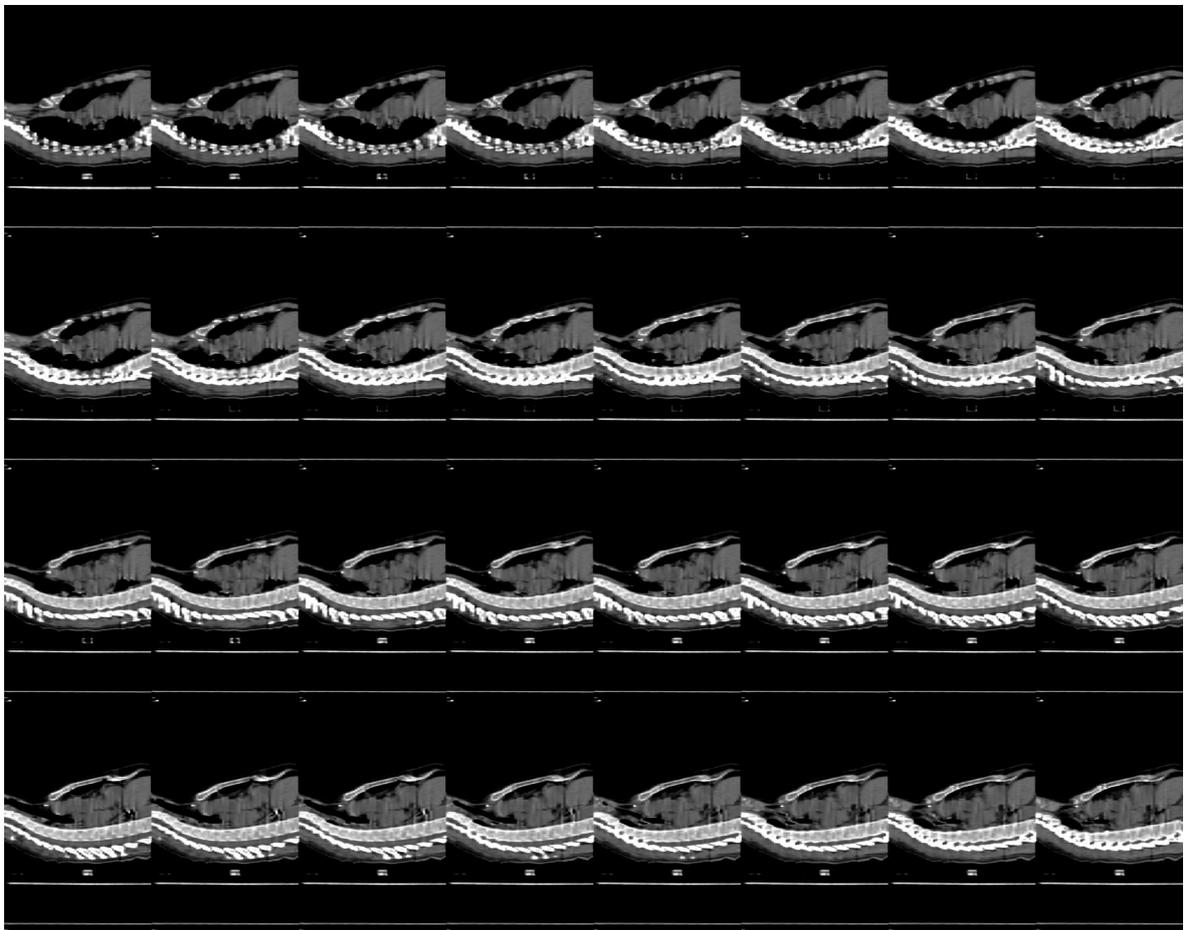
**Figura 4.7:** Plano coronal de un examen de TC generado a partir de archivos DICOM con un total de 63 cortes en el plano axial y 512 cortes en el plano coronal, utilizando la ventana con valores entre  $-100HU$  a  $300HU$ .

### 4.1.3. Visualización Plano Sagital

El tratamiento de las imágenes para el proceso de visualización en el plano sagital es el mismo que ya se describió para el plano axial y para el plano coronal. Una

vez que se ha obtenido la matriz de píxeles y se ha encontrado la relación que existe entre los ejes  $y$  y  $z$  se procede a graficar las imágenes obtenidas mediante las matrices que corresponden a este plano. Al momento de realizar la presentación de las imágenes mediante la biblioteca Matplotlib, se aplican las relaciones entre los ejes que para este caso es de  $z/y = 5,120$ , de manera similar al empleado en los planos anteriores.

Al igual que en el plano coronal, en este caso no se muestran todas las imágenes ya que muchas de ellas principalmente las que se encuentran fuera del centro no contienen anatomía interna del paciente. Las imágenes seleccionadas para la Figura 4.8 son un total de 32 cortes que se sitúan en el centro del eje coordenado.



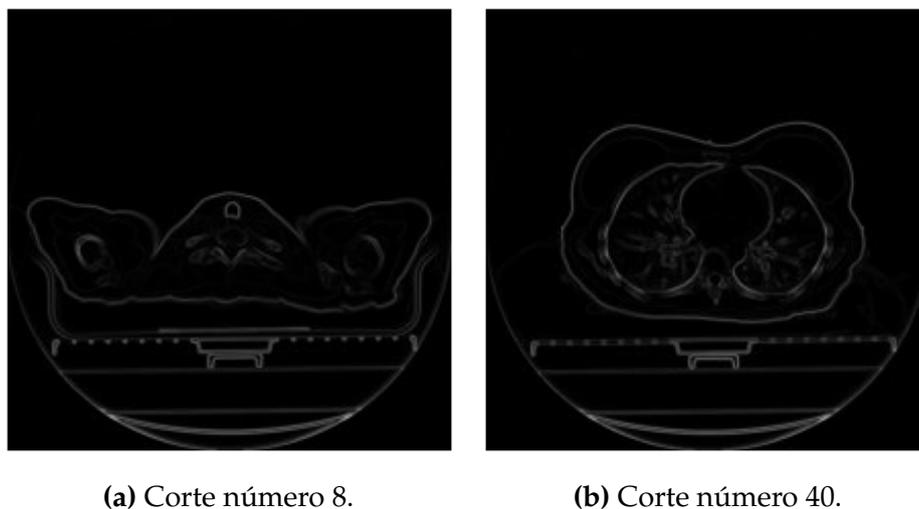
**Figura 4.8:** Plano sagital de un examen de TC generado a partir de archivos DICOM con un total de 63 cortes en el plano axial y 512 cortes en el plano sagital, utilizando la ventana con valores entre  $-100HU$  a  $300HU$ .

## 4.2. Segmentación de tejido pulmonar

La segmentación de pulmones sigue el flujo de trabajo que se describió en la sección anterior, ver Figura 3.9. Así como en la visualización, en la segmentación es necesario encontrar la matriz de píxeles y transformarla en una medida física que represente una magnitud real, como son las unidades Hounsfield. Por lo que, desde el desarrollo, hasta la obtención de la matriz y su posterior transformación es el mismo para ambos procesamientos. De tal manera que se utilizarán las funciones que se diseñaron en la visualización para la lectura de la imagen de archivos DICOM, para la obtención, la transformación de unidades, y para la corrección de la matriz  $N \times M \times M$ . Luego de obtener la matriz se procedió a guardarla en el formato NPY.

Para el análisis se utilizarán cortes seleccionados que se adapten al tipo de estudio que se realiza. Primero, utilizaremos los cortes 8 y 40 del examen de prueba y procedemos a generar la imagen de gradiente, debido a que el algoritmo de Watershed generalmente se calcula en este tipo de imágenes.

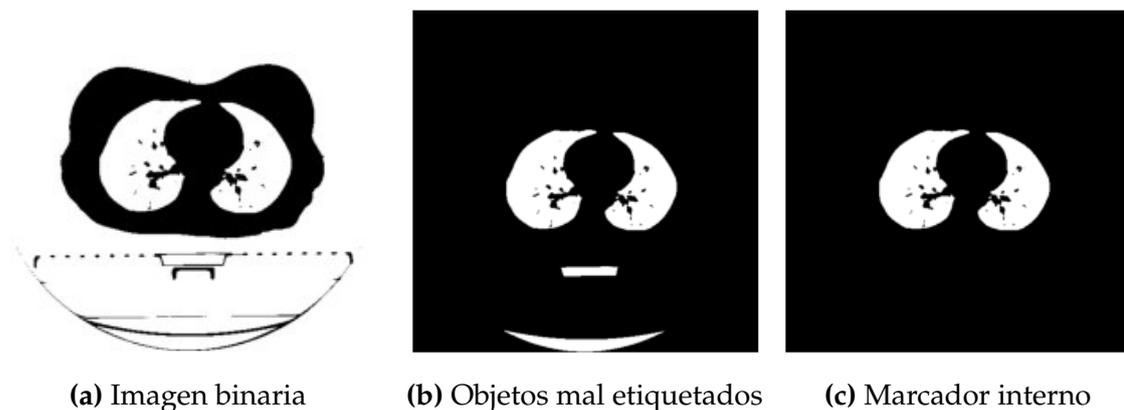
La imagen de gradiente se obtuvo mediante uno de los filtros más populares, como es el filtro de Sobel. Este filtro es aplicado en las direcciones  $x$  y  $y$ . Posteriormente, se encontró la magnitud del vector gradiente y se normalizó la matriz resultante con respecto a la magnitud máxima de la escala de grises y se graficó. La figura 4.9 presenta las imágenes de gradiente de los cortes seleccionados, respectivamente, de la muestra de datos.



**Figura 4.9:** Imágenes de gradiente de diferentes cortes axiales de un examen de TC de tórax.

A continuación de obtener la imagen de gradiente, se encuentran los marcadores

interno y externo. A fin de encontrar el marcador interno realizamos un filtraje de los píxeles en la imagen original. El filtraje se basa en etiquetar dos regiones: la que comprende a pulmones y la que corresponde a fondo, generando de esta forma una imagen binaria. La región pulmonar engloba los valores menores a  $-400HU$ , ya que los pulmones tiene un valor de alrededor de  $-400HU$  a  $-600HU$ , mientras que el aire, que se encuentra dentro de los pulmones tiene aproximadamente  $-1000HU$ . Las cantidades que sobrepasan estos valores corresponden a la pared torácica, la sangre, los huesos, etc. Este proceso genera una matriz binaria en la que todos los elementos que cumplen con la condición (píxel  $< -400$ ) tendrán como salida TRUE o 1, mientras que los píxeles que no cumplan con la condición tendrán un valor de FALSE o 0. En la figura 4.10a se puede observar el resultado de graficar la matriz binaria generada por la condición de filtrado para el corte número 40 de la muestra.

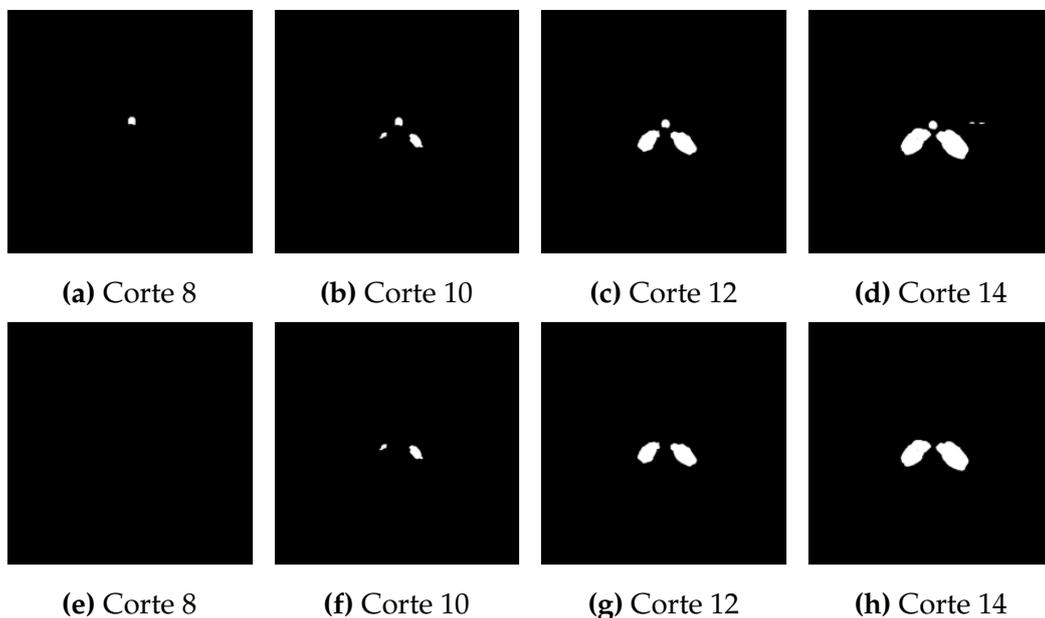


**Figura 4.10:** Proceso para la generación del marcador interno en el corte 40 del examen de TC de tórax de prueba. (a) Imagen binaria resultado del filtraje por unidades Hounsfield con la condición (píxel  $< -400HU$ ). (b) Pulmones y parte de la camilla que se etiquetaron como región pulmonar (c) Marcador interno, después de eliminar todos los objetos que se etiquetaron erróneamente como región pulmonar al momento de realizar el filtraje por unidades Hounsfield.

Como el fondo de la imagen corresponde a aire, tenemos como resultado una imagen con una región que rodea al cuerpo que según la condición aplicada pertenece a la zona de los pulmones, ver figura 4.10a. Evidentemente esta región no se refiere a pulmones, y en consecuencia se procede a eliminarla mediante la función *segmentation.clear\_border* de la biblioteca **skimage**, que borra los objetos conectados al borde de la imagen. Asimismo, en la imagen 4.10b se observa que partes de la camilla son etiquetadas como zonas que pertenecen a pulmones ya que se encuentran dentro de los rangos que se aplicó para el filtraje. Para solucionar este problema,

implementamos la función que se ejecutó en la sección de visualización donde eliminamos la camilla por medio de selección de píxeles. En la figura 4.10c se observa el resultado de eliminar las zonas que fueron mal etiquetadas en el proceso de filtraje, generando así el marcador interno final para este corte.

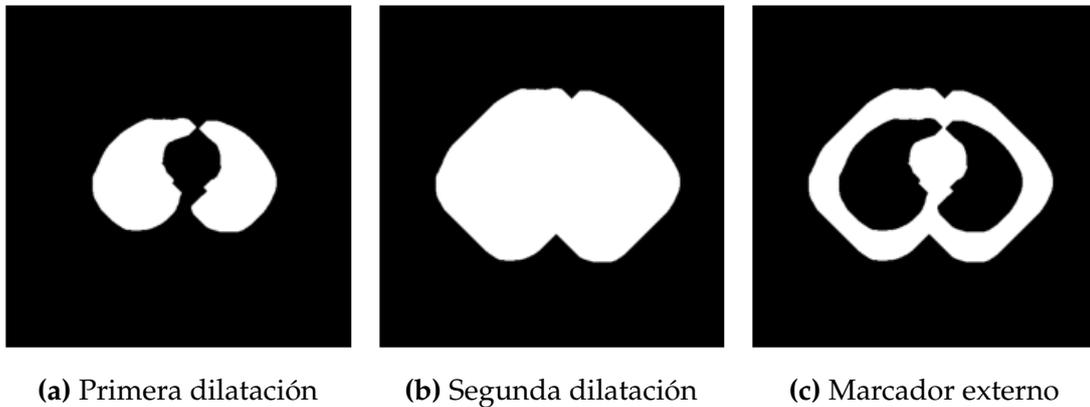
Además de los objetos mencionados anteriormente se tiene que la tráquea también aparece como tejido pulmonar, ya que esta zona contiene aire. Para eliminar la tráquea de la imagen binaria del marcador interno, se etiqueta cada uno de los elementos. La ubicación de la tráquea se identifica buscando el objeto circular grande cerca del centro en los primeros cortes, teniendo en cuenta esta característica se diseñaron varios métodos para eliminarla. El proceso para eliminar la tráquea toma en cuenta el tamaño y la ubicación de la misma con respecto a la de los pulmones. Así pues, por medio de funciones que retornan las características de las regiones en la imagen se cuantificó la dimensión en número de píxeles que comprende la tráquea. Luego, se eliminan todos los elementos que abarquen estas dimensiones y que se ubiquen en las posiciones dadas por los primeros cortes donde se determinó la tráquea. En la figura 4.11 se observa el resultado del código aplicado en los primeros cortes del examen.



**Figura 4.11:** Cortes antes y después del proceso de eliminación de la tráquea. (a)(b)(c)(d) Primeros cortes del examen de TC de toráx, donde se aprecia la tráquea, que corresponde al círculo en el centro del corte. (e)(f)(g)(h) Marcador interno después del proceso de remoción de la tráquea y de todos los objetos que no pertenecen a los pulmones.

Una vez encontrado el marcador interno de las imágenes, procedemos a utilizar

su matriz binaria para deducir el marcador externo mediante la dilatación de la matriz, a través de dos elementos estructurantes circulares de diferentes proporciones con el propósito de expandir las formas en la imagen. Así, se genera una franja de al menos 45 píxeles por medio de la intersección entre los resultados de las operaciones morfológicas.

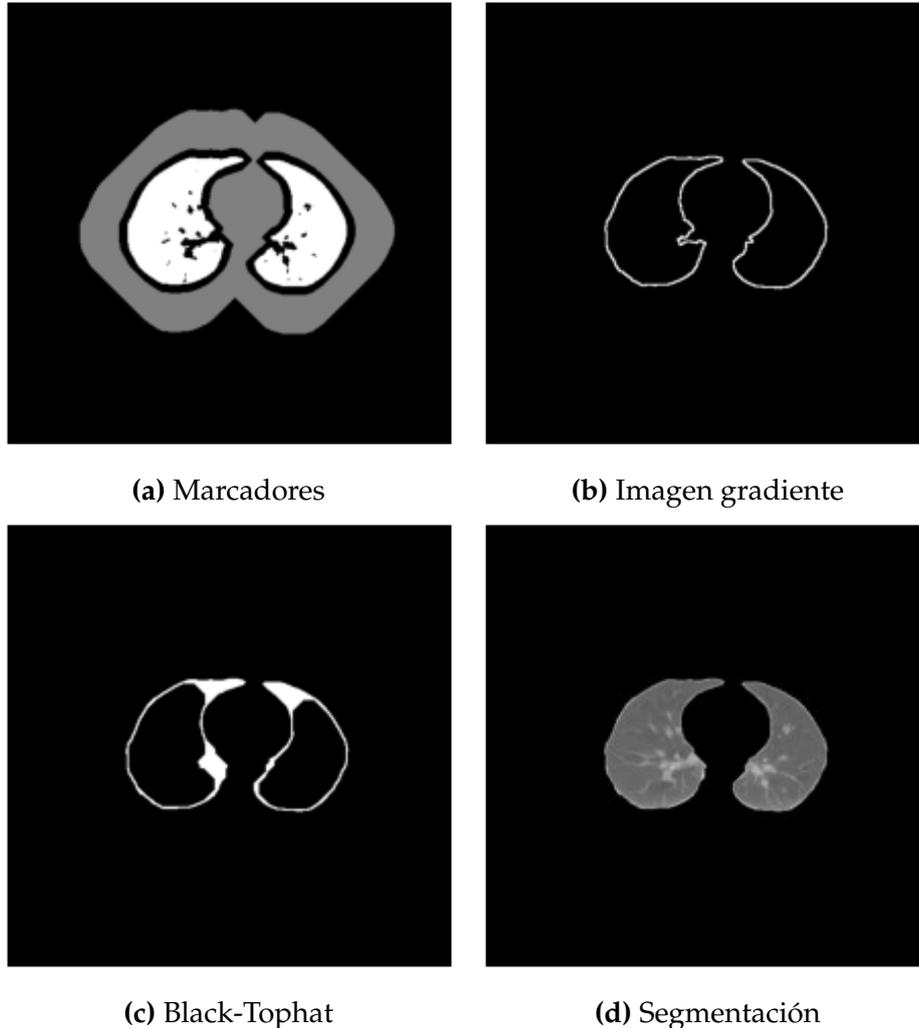


**Figura 4.12:** Proceso para la generación del marcador externo en el corte 40 del examen de TC de tórax de prueba. (a). Primera dilatación con 10 interacciones. (b) Segunda dilatación con 55 interacciones. (c) Marcador externo, la franja generada es de 45 píxeles.

Por el proceso realizado tenemos que el borde del pulmón se encuentra en la vecindad del marcador interno (zona de color blanco en la figura 4.13a) y del marcador externo (área color gris en la figura 4.13a). Por lo expuesto anteriormente, la aplicación del algoritmo de Watershed será en la tira generada entre ambos marcadores, que llamaremos marcador de Watershed (tira de 45 píxeles color negra alrededor del marcador interno, ver figura 4.13a). Posteriormente, aplicamos el algoritmo de Watershed basado en marcadores en la imagen de gradiente. Al aplicar el algoritmo utilizamos la imagen de gradiente de la figura 4.9b con sus respectivos marcadores interno y externo. Luego, obtenemos una imagen binaria en donde se representa con color blanco la región pulmonar y con negro el fondo. A continuación, encontramos el contorno de los pulmones mediante la aplicación del gradiente morfológico en la imagen binaria, como se muestra en la figura 4.13b. El gradiente morfológico se obtiene mediante la diferencia entre una dilatación y una erosión de la imagen segmentada con un elemento estructurante dado.

Al utilizar el algoritmo de Watershed la imagen segmentada descarta estructuras densas pero que pertenecen a los pulmones. Consecuentemente, es imprescindible incluir estas estructuras en la imagen final. Con este fin se aplicará la transformada Black-Tophat, que provoca que los bordes en la imagen segmentada se suavicen y

se rellenen las cavidades que se generaron al aplicar el algoritmo. En la figura 4.13c se puede ver el resultado de aplicar la transformada Black-Tophat sobre los bordes de la imagen de gradiente después del proceso de segmentación por Watershed.

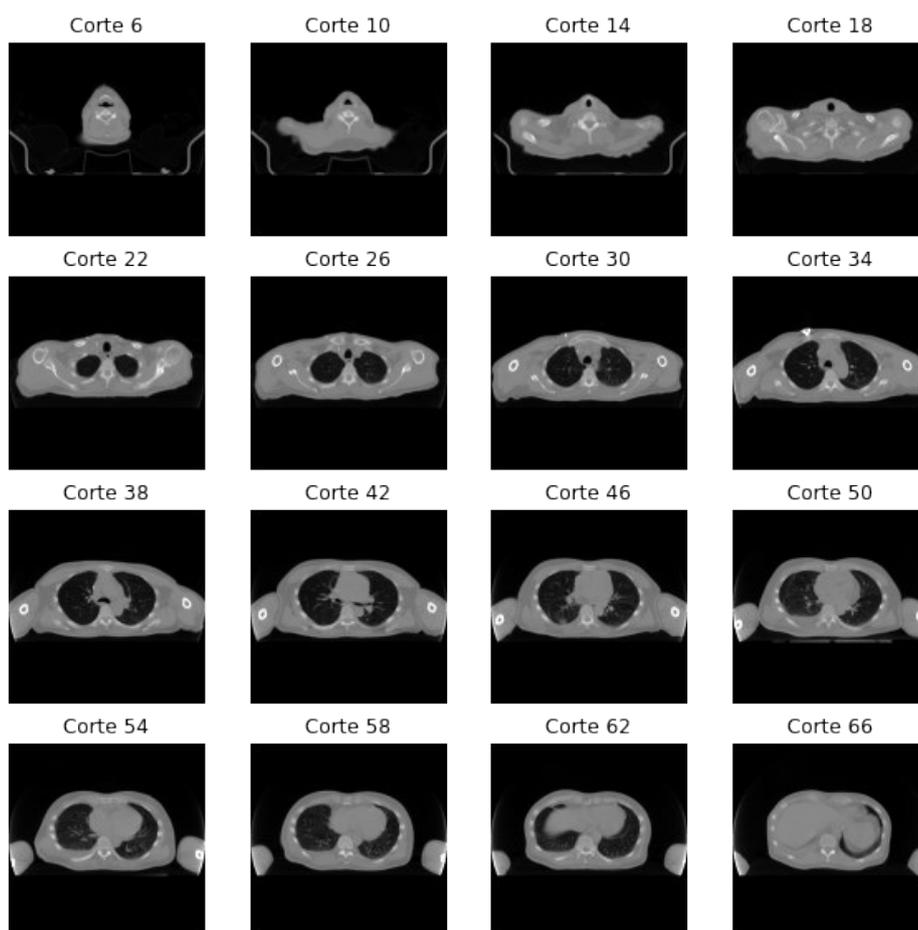


**Figura 4.13:** Aplicación de algoritmo de Watershed y Black-Tophat para la segmentación en el corte 40. (a) Imagen con los marcadores etiquetados para la aplicación del algoritmo de Watershed. La zona gris representa el marcador externo, la región blanca el marcador interno y la franja negra entre ambos marcadores es el marcador de Watershed. (b) Imagen de gradiente morfológico, aplicado al resultado de la segmentación de Watershed. (c) Reincorporación de estructuras densas mediante la transformación Black-Tophat, que suaviza y rellena cavidades. (d) Pulmones segmentados.

Finalmente, luego de aplicar todo el proceso explicado anteriormente se realiza de manera automática la segmentación de pulmones en la imagen de prueba. Mediante la imagen binaria de 4.13c seleccionamos en la imagen original los píxeles que representan fondo y los que se seleccionarán como tejido pulmonar, cuyo resultado

se puede evidenciar en la figura 4.13d.

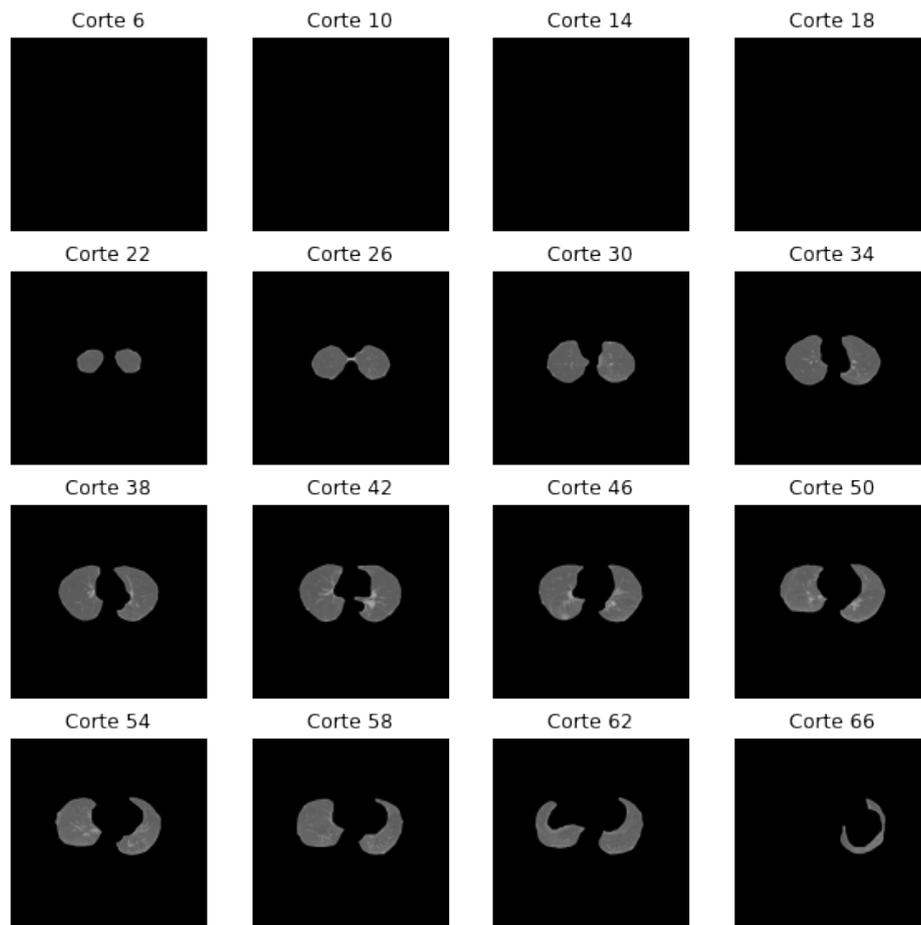
A continuación, aplicamos el algoritmo a un examen completo. Con el fin de examinar los datos con los que se está trabajando, de manera que se generó una imagen que es incrustada en el notebook o que puede ser exportada en formato PNG. La diferencia entre este método y el que se discutió en la sección 4.1.1 (Visualización Plano Axial) es el número de cortes que se generan para la visualización y la calidad con las que son exportadas las imágenes en cada caso. El método que se utiliza en este escenario es de forma exploratoria, por lo que, en consecuencia la imagen pierde algo de resolución, debido a que el tamaño de la imagen en píxeles disminuye radicalmente. Este proceso de visualización se limita a imágenes seleccionadas, y se lo realiza con el fin de priorizar la optimización del tiempo empleado y dejar de lado la calidad de la imagen generada. La figura 4.14 muestra un fragmento del examen de TC de tórax de prueba, que utiliza esta lógica de visualización.



**Figura 4.14:** Muestra de imágenes de un examen de TC de radioterapia de Tórax, el examen cuenta con 75 cortes, la visualización utiliza la ventana con valores entre  $-1000HU$  a  $1000HU$ .

El código para segmentación de pulmones fue aplicado a los cortes mostrados en la figura 4.14, el cual representa un examen de TC de tórax de 75 archivos DICOM. El resultado puede ser observado en la figura 4.15. En la mayoría de los cortes se presentan resultados visualmente aceptables, en los cuales se segmenta pulmones junto con vasos, mientras que en algunos cortes además de seleccionar la región de interés se encuentran presentes tráquea, huesos y tejidos blandos, según el criterio de la Dra. Elizabeth Zamora médica radióloga que desempeña actividades profesionales en el área de Radiología en la Clínica Novaclínica Santa Cecilia.

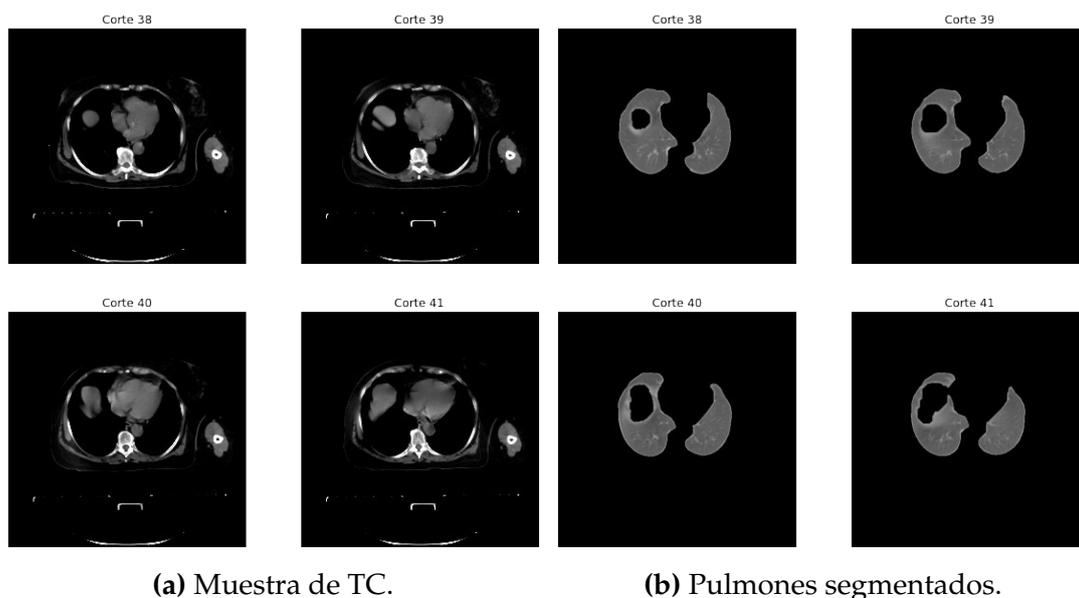
Independientemente de las imágenes obtenidas, en este caso, el resultado más importante es la generación de la matriz en 3D que contendrá las imágenes segmentadas de los pulmones en unidades Hounsfield. De igual forma, como se almacenaron las imágenes de TC aquí se utiliza el formato NPY para guardar los cortes de pulmones segmentados del examen completo.



**Figura 4.15:** Pulmones segmentados de la muestra de imágenes de un examen de TC de radioterapia de Tórax, presentados en la figura 4.14 que contiene un total de 75 cortes.

El algoritmo de segmentación, finalmente fue optimizado y convertido en una función de Python. El cual permite automatizar el proceso de segmentación para que sea muy fácil de utilizar. Además, mediante la función implementada se pueden modificar ciertos parámetros para que los resultados sean más eficientes y no dependan del tipo de camillas utilizadas en el proceso de TC.

Aquí es importante destacar un resultado particular encontrado en algunos de los últimos cortes de ciertos exámenes de TC. En la figura 4.17b se observa que en los pulmones segmentados aparecen unos orificios. Esta parte es descartada del tejido pulmonar por el algoritmo de segmentación, ya que pertenece a zonas donde existen líquidos u otro tipo de tejido que no corresponde a pulmones. Estas estructuras solamente aparecen en los últimos cortes, en donde el tejido pulmonar comienza a desaparecer y se manifiestan estos nuevos tejidos, esto, de acuerdo al criterio profesional de la Dra. Manola Pazmiño, médica radióloga quien validó los resultados.



**Figura 4.16:** (a) Muestra de los últimos cortes de un examen de TC de radioterapia de Tórax. (b) Pulmones segmentados de la muestra de TC presentada en (a).

La función de segmentación de pulmones fue aplicada a las 18 muestras de exámenes de TC que contienen la zona del tórax. El algoritmo funciona perfectamente para la mayoría de los casos, a diferencia de ciertos exámenes con características particulares en los cuales la segmentación no se realiza de manera eficaz. De acuerdo a las pruebas realizadas se extraen los siguientes comentarios:

- La posición del paciente debe de ser HFS (Head First-Supine) o FFS (Feet First-Supine), caso contrario el algoritmo no funcionará de manera exacta en los

primeros cortes (cortes donde aparezca la tráquea), porque la identificación de la tráquea depende de la ubicación de la misma en la imagen.

- Las imágenes deben pertenecer a la zona del tórax, dado que si el algoritmo es aplicado a otras partes del cuerpo, puede generar resultados inexactos, porque al realizar la selección del marcador interno se utiliza un umbral de  $-400HU$ , que abarca varias zonas, como es el caso de la cabeza (boca que contiene aire), en donde se puede identificar tejido pulmonar, aunque esto no sea correcto. Por lo tanto, la mejor opción es eliminar imágenes que no pertenezcan a la zona del tórax para evitar este problema. Este paso de eliminación, además de que excluye posibles causas de errores, hace que el algoritmo sea mucho más rápido, puesto que no se analizan imágenes innecesarias. Cabe decir que, este es un proceso muy sencillo ya que se usa el objeto `np.ndarray` y las funciones implementadas de visualización para realizarlo.

#### 4.2.1. Resultados de Segmentación de Pulmones

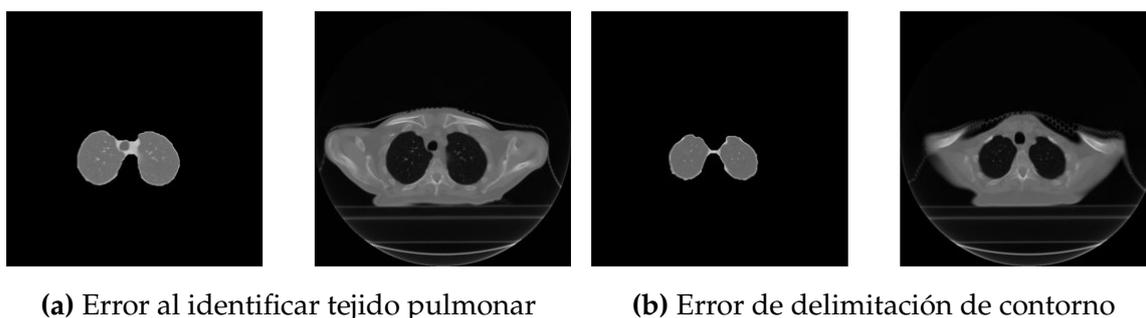
Para el análisis de los resultados obtenidos, se eliminaron todas las imágenes que no pertenecen a la zona del tórax. Además, se descartaron los exámenes en donde los pacientes no se encuentran en las posiciones HFS o FFS. Por lo que, se obtuvo un total de 17 pruebas finales, con un número muy variado de cortes por muestra.

N.º Examen	N.º Cortes	N.º Cortes segmentados (con errores)	N.º Examen	N.º Cortes	N.º Cortes segmentados (con errores)
1	25	7	10	50	10
2	29	5	11	60	15
3	29	7	12	45	6
4	44	8	13	60	9
5	50	12	14	40	7
6	9	2	15	61	18
7	240	82	16	38	6
8	47	28	17	30	12
9	50	8	-	-	-

**Cuadro 4.4:** Resultados de la segmentación de pulmones en diferentes exámenes. N.º Examen (Identificador del examen), N.º Cortes (Número de cortes por examen), N.º Cortes segmentados (con errores) (Número de cortes con errores en la segmentación de pulmones).

En el cuadro 4.5 se presentan los resultados del proceso de segmentación en cada uno de los cortes de los 17 exámenes de prueba. Una descripción más detallada de cada examen se encuentra en el Apéndice D.4.

Para la selección de cortes que contenían errores en la segmentación, se tomaron en cuenta todas las imágenes en las cuales la tráquea, huesos y tejidos blandos son confundidos con tejido pulmonar. Asimismo, los cortes en donde los pulmones no se encuentran bien delimitados en todo su contorno han sido considerados dentro del grupo que posee errores en el proceso de segmentación.



**Figura 4.17:** Cortes con errores de segmentación de pulmones (a) Tráquea identificada como parte de los pulmones. (b) Los bordes del pulmón no son delimitados correctamente.

En esta investigación se estudiaron 17 pacientes, con un promedio de 53 cortes por examen, que constituyen un total de 907 imágenes de TC de tórax en total. De los 907 cortes segmentados, se registraron 249 imágenes de pulmones con problemas en contornos y una identificación imperfecta de tejido pulmonar, lo que representa un 26.68 % del total de pruebas realizadas.

	N.º Cortes Segmentados	N.º Cortes Segmentados (con errores)
N.º Cortes totales	907	249
Media	53	14
Desviación estándar	50.08	18.46
Mínimo	9	2
Máximo	240	82

**Cuadro 4.5:** Resultados cuantitativos de la segmentación de pulmones del número total de cortes, correspondientes a todos los exámenes.

Todas las pruebas realizadas al conjunto de datos final arrojaron buenos resultados de segmentación en la mayoría de los casos. El algoritmo no confunde zonas de la camilla con tejido pulmonar en el total de pruebas realizadas. Además, la identi-

ficación de tráquea y pulmones se lleva a cabo de una mejor manera, a diferencia de los códigos base en los cuales se apoya este proyecto.

Es preciso insistir que todos los archivos creados y utilizados, tanto para la extracción y visualización de datos e imágenes en archivos DICOM, así como también para el proceso de segmentación de pulmones se encuentran alojados en el repositorio: <https://github.com/jennifer95/visualizacion-segmentacion-pulmonar-DICOM-python>. Además, encontrará todos los cuadernos de Jupyter Notebook generados para explicar los procesos y resultados descritos en este capítulo. Así pues, del mismo repositorio se puede descargar la herramienta que se creó para el proceso de anonimización de información y visualización de datos e imágenes de archivos DICOM.

# Capítulo 5

## Conclusiones

El formato DICOM proporciona una gran cantidad de datos del proceso de tomografía computarizada, que ayuda a sustentar la información dada únicamente por la imagen. Además, permite desarrollar cortes en diferentes planos anatómicos, así como también la obtención de las imágenes bajo diferentes ventanas y niveles de grises, a través de la manipulación, obtención de la matriz de píxeles (matriz con los coeficientes de atenuación de cada vóxel) y demás atributos, que hacen posible una buena interpretación de las estructuras anatómicas del paciente.

Trabajar directamente con archivos DICOM resulta desafiante, ya que es necesario software especializado para este formato y que trabaje bajo este estándar. Por esta razón, se generó la necesidad de crear un software que decodifique y permita visualizar la información almacenada en los archivos, y que además tenga la capacidad de poder guardar los datos extraídos de una manera rápida, de fácil acceso y que a su vez pueda ser exportada a otras plataformas.

El lenguaje de programación Python es una excelente opción para el tratamiento informático de imágenes médicas en formato DICOM. Ya que, Python cuenta con una gran cantidad de librerías tanto para manipulación de archivos DICOM así como también para procesamiento de imágenes. Este lenguaje de alto nivel permitió desarrollar el software para comprender, analizar y tratar una imagen de tomografía computarizada de una manera más sencilla. Además, mediante Python fue posible desarrollar una herramienta que elimina y reemplaza todos los datos del perfil de confidencialidad, para no comprometer información personal o delicada del paciente a la cual también se tiene acceso al momento de manejar archivos DICOM. Por otra parte, Python y el entorno de programación Jupyter Notebook son de acceso libre, lo que posibilita que cualquier usuario lo pueda utilizar, independientemente

de su sistema operativo (Windows, GNU Linux, Mac OS), y de ser el caso modificar la herramienta e implementarla de acuerdo a sus requerimientos.

Las metodologías implementadas en este proyecto proveen la capacidad de poder trabajar directamente con los valores de atenuación lineal de los tejidos (unidades Hounsfield). Además, al operar directamente con la matriz de píxeles la imagen no depende del tipo de ventana con la que se trabaje, lo que representa una gran ventaja con respecto al procesamiento sobre imágenes en escala de grises. Asimismo, las técnicas implementadas permiten utilizar valores tabulados de unidades Hounsfield referentes a tejidos, para el procesamiento de los cortes. De esta manera, se pueden identificar varias estructuras anatómicas con base en sus valores de atenuación lineal.

La herramienta de software implementada es independiente de los diferentes tipos de dispositivos que se utilicen para generar los archivos DICOM. Dicho de otro modo, es independiente del tipo de técnica que se use para la generación de las imágenes. Esto, para el caso de anonimización y visualización de datos en archivos DICOM. Ahora bien, para la visualización de la imagen ocurre de manera diferente, porque para imágenes de tomografía computarizada se trabaja con un tipo de cantidad específica: unidades Hounsfield, una característica particular de esta técnica. Todo el proceso de obtención de la imagen se basa en transformaciones utilizando unidades Hounsfield, además los tags de los atributos que identifican a cada elemento en los archivos DICOM son diferentes entre cada una de las técnicas. Teniendo en cuenta esto, de momento, no es posible que la herramienta pueda funcionar para otros tipos de modalidades de obtención de imágenes médicas (Resonancias Magnéticas, ecografía, etc.), sino solamente para imágenes producto del proceso de tomografía computarizada. Cabe destacar que la herramienta no depende del número de imágenes en el examen, tipo de protocolo, etc. Mientras que el examen sea de tomografía computarizada y se utilice la atenuación lineal de los tejidos como base para generar la imagen médica la herramienta brinda buenos resultados.

La segmentación de pulmones en imágenes de cortes axiales de TC de tórax fue desarrollada mediante una técnica que involucra los siguientes conceptos: unidades Hounsfield, el algoritmo de Watershed basado en marcadores, operaciones morfológicas en imágenes y la transformación Black-Tophat. El proceso utilizado para la segmentación de pulmones permite rellenar y suavizar los bordes de la imagen segmentada, ya que al utilizar el algoritmo de Watershed se excluyen ciertas estructuras densas que pertenecen a pulmones, pero que no están dentro de los valores típicos de atenuación lineal de los tejidos pertenecientes a dicho órgano, como es el caso

de los nódulos pulmonares ubicados en los bordes del pulmón. Tomando en cuenta todo lo mencionado anteriormente, se desarrolló un código en Python en un cuaderno de Jupyter Notebook para probar el algoritmo de segmentación en un total de 17 exámenes con el elemento 10.7 RADIOTERAPIA TORAX, correspondientemente al atributo Protocol Name. Además, se utilizó un examen con el atributo 6.1 ABDOMEN Y PELVIS de un TC corporal, por lo que en total se aplicó el algoritmo en 18 exámenes diferentes, con un número variable de imágenes por examen. La segmentación arrojó resultados visualmente aceptables para 17 del total de muestras. Sin embargo, en uno de los exámenes no se realiza una segmentación eficaz en los primeros cortes debido a la posición en la que se encuentra el paciente.

Las modificaciones que se realizaron a los códigos en los cuales se apoya este proyecto lograron que la segmentación sea más óptima y efectiva. Los algoritmos base confundían pulmones con tráquea en cortes donde el tamaño del pulmón y de la tráquea eran semejantes (primeros cortes). Por lo tanto, se tomaron otras características para la eliminación de la tráquea. La ubicación y el tamaño de la tráquea resultaron ser buenos parámetros para que el proceso de identificación de pulmones retorne mejores resultados. Al utilizar la localización de la tráquea como base para identificarla, hace que la técnica de segmentación dependa de la posición del paciente dentro del instrumento. En consecuencia, el algoritmo genera buenos resultados cuando el paciente se encuentra en la posición HFS (Head First-Supine), que es la más común para los exámenes de TC. Algo semejante ocurre con los contornos cerrados llenos de aire en partes de la camilla, de ahí que el proceso de remoción de la camilla genera buenos resultados y el problema desaparece.

Las pruebas realizadas demuestran que la técnica de segmentación pulmonar como paso previo al procesamiento de sistemas CAD, es independiente del tipo de protocolo utilizado para generar las imágenes, además de que es precisa y rápida. También se confirma que la herramienta realiza la segmentación aunque los bordes no sean muy fuertes en la imagen.

# Apéndice A

## Anonimización de Imágenes DICOM

Para el proceso de eliminar datos sensibles en imágenes DICOM se crearon 3 scripts. El archivo `anonimizacion.py` que contiene varias funciones que son utilizadas para anonimizar los datos confidenciales en imágenes DICOM. El archivo `manejo_tags.dat` permite un fácil manejo de los atributos a partir de sus identificadores de grupo y elemento. El documento con los atributos que se utilizaron para proteger la identidad del paciente se denomina `private_tags.dat`.

### A.1. Script `anonimizacion.py`

Este script cuenta con 3 funcionalidades, la primera es para la anonimización de una única imagen DICOM. La segunda funcionalidad permite anonimizar un examen completo de tomografía computarizada, con la ventaja de que cada imagen que pertenezca a un mismo examen tendrá un identificador único (nombre del paciente), a diferencia de algunas bibliotecas de Python creadas con este fin, que asignan nombres aleatorios a cada imagen. Finalmente la tercera funcionalidad, en donde se consigue una automatización completa, ya que permite anonimizar una carpeta con diferentes exámenes y diferentes pacientes logrando así optimizar mucho tiempo.

### A.2. Script `manejo_tags.py`

Este script permite el uso y manejo fácil de los atributos de las imágenes DICOM según sus identificadores de grupo y de elemento.

### **A.3. Archivo `private_tags.dat`**

Este archivo contiene todos los atributos del perfil de confidencialidad que se utilizaron en el script `anonimizacion.py`. Estos elementos son utilizados para: la protección, la eliminación y reemplazo de Atributos dentro de un Conjunto de Datos DICOM que potencialmente pueden resultar en la filtración de Información de Identificación Individual sobre el paciente u otras personas u organizaciones involucradas en la adquisición de la imagen. Los elementos fueron extraídos de [30] y de [31]. El archivo cuenta con 3 columnas, en las 2 primeras se especifican los valores correspondientes a sus identificadores de grupo y de elemento, respectivamente. Para este caso, la tercera columna irá vacía debido a que está reservada para atributos donde se especifiquen unidades.

# Apéndice B

## Visualización de Atributos DICOM

Para la visualización de la metadata de archivos DICOM es necesario la utilización de 2 scripts `imprimir_informacion.py` y `manejo_tags.py`, este último ya descrito y utilizado en el proceso de anonimización. Además es imprescindible un documento con tres columnas donde se especifiquen los identificadores de grupo y de elemento para las dos primeras columnas y para la tercera columna las unidades del atributo que se desee obtener, para este proyecto se generaron 2 documentos. El archivo `private_tags.dat`, utilizado ya con anterioridad y el archivo `CT_tags.dat` que tiene el mismo formato pero con los atributos correspondientes a parámetros de adquisición de un archivo DICOM de TC, que fue utilizado en la Figura 3.2.

### B.1. Script `imprimir_informacion.py`

Este script permite la visualización de los metadatos en archivos DICOM, para este caso se implementaron 2 archivos `.dat` donde se especifican los identificadores de los atributos y sus unidades, pero si el usuario desea visualizar otros elementos que no están especificados en los archivos ya implementados, lo puede hacer, mediante el ingreso de un archivo con la estructura indicada.

### B.2. Archivo `CT_tags.dat`

Archivo con los parámetros de adquisición de una imagen de CT, los elementos fueron adquiridos de [79].

## B.3. Ejemplos de Visualización de datos DICOM

A continuación se muestra el funcionamiento del script `imprimir_informacion.py`. Este algoritmo permite la visualización de la metadata dentro de las imágenes DICOM. Específicamente, los atributos pertenecientes al perfil de confidencialidad. Además, en las figuras B.1a y B.1b se presenta una comparativa entre la metadata de las imágenes antes y después de su proceso de anonimización, respectivamente. Los datos en la visualización antes del proceso de anonimizar fueron censurados.

Tag	Name	Keyword	Value
(0008, 0050)	Accession Number	AccessionNumber	
(0040, 2017)	Filler Order Number / Imaging Service Request	FillerOrderNumberImagingServiceRequest	
(0010, 0020)	Patient ID	PatientID	
(0010, 0030)	Patient's Birth Date	PatientBirthDate	26/09/1956
(0010, 0010)	Patient's Name	PatientName	
(0010, 0040)	Patient's Sex	PatientSex	F
(0008, 0090)	Referring Physician's Name	ReferringPhysicianName	
(0008, 0020)	Study Date	StudyDate	28/07/2020
(0020, 0010)	Study ID	StudyID	39052
(0008, 0030)	Study Time	StudyTime	13:56:23
(0010, 0021)	Issuer of Patient ID	IssuerOfPatientID	SITE_DEFAULT_LOCAL
(0010, 1040)	Patient's Address	PatientAddress	
(0010, 1010)	Patient's Age	PatientAge	63Y
(0040, 0254)	Performed Procedure Step Description	PerformedProcedureStepDescription	
(0040, 0253)	Performed Procedure Step ID	PerformedProcedureStepID	PPS ID 39052
(0040, 0244)	Performed Procedure Step Start Date	PerformedProcedureStepStartDate	28/07/2020
(0040, 0245)	Performed Procedure Step Start Time	PerformedProcedureStepStartTime	13:56:23
(0008, 1050)	Performing Physician's Name	PerformingPhysicianName	
(0040, 0275)	Request Attributes Sequence	RequestAttributesSequence	
(0032, 1033)	Requesting Service	RequestingService	Unspecified
(0008, 103e)	Series Description	SeriesDescription	ARTERIAL, TC CORPORAL
(0008, 1030)	Study Description	StudyDescription	
(0020, 0052)	Frame of Reference UID	FrameOfReferenceUID	
(0008, 3010)	Irradiation Event UID	IrradiationEventUID	
(0020, 000e)	Series Instance UID	SeriesInstanceUID	
(0008, 0018)	SOP Instance UID	SOPInstanceUID	
(0020, 000d)	Study Instance UID	StudyInstanceUID	
(0008, 0023)	Content Date	ContentDate	28/07/2020
(0008, 0033)	Content Time	ContentTime	14:07:33
(0008, 0022)	Acquisition Date	AcquisitionDate	28/07/2020
(0008, 0032)	Acquisition Time	AcquisitionTime	
(0018, 1030)	Protocol Name	ProtocolName	6.1 ABDOMEN Y PELVIS*
(0008, 0021)	Series Date	SeriesDate	28/07/2020
(0008, 0031)	Series Time	SeriesTime	14:06:06
(0018, 1000)	Device Serial Number	DeviceSerialNumber	*
(0008, 0080)	Institution Name	InstitutionName	

(a) Metadata de imagen DICOM sin anonimizar, los datos sensibles fueron censurados.

Tag	Name	Keyword	Value
(0010, 0030)	Patient's Birth Date	PatientBirthDate	26/09/1956
(0010, 0010)	Patient's Name	PatientName	Paciente Anónima
(0010, 0040)	Patient's Sex	PatientSex	F
(0008, 0020)	Study Date	StudyDate	28/07/2020
(0008, 0030)	Study Time	StudyTime	13:56:23
(0010, 0021)	Issuer of Patient ID	IssuerOfPatientID	SITE_DEFAULT_LOCAL
(0010, 1040)	Patient's Address	PatientAddress	Entidad Anónima
(0010, 1010)	Patient's Age	PatientAge	63Y
(0040, 0254)	Performed Procedure Step Description	PerformedProcedureStepDescription	TC CORPORAL
(0040, 0244)	Performed Procedure Step Start Date	PerformedProcedureStepStartDate	28/07/2020
(0040, 0245)	Performed Procedure Step Start Time	PerformedProcedureStepStartTime	13:56:23
(0008, 1050)	Performing Physician's Name	PerformingPhysicianName	Entidad Anónima
(0032, 1033)	Requesting Service	RequestingService	Unspecified
(0008, 103e)	Series Description	SeriesDescription	ARTERIAL, TC CORPORAL
(0008, 1030)	Study Description	StudyDescription	
(0008, 0023)	Content Date	ContentDate	28/07/2020
(0008, 0033)	Content Time	ContentTime	14:07:33
(0008, 0022)	Acquisition Date	AcquisitionDate	28/07/2020
(0008, 0032)	Acquisition Time	AcquisitionTime	14:0250,908430
(0018, 1030)	Protocol Name	ProtocolName	6.1 ABDOMEN Y PELVIS*
(0008, 0021)	Series Date	SeriesDate	28/07/2020
(0008, 0031)	Series Time	SeriesTime	14:06:06
(0018, 1000)	Device Serial Number	DeviceSerialNumber	*
(0008, 0080)	Institution Name	InstitutionName	Entidad Anónima
(0008, 1010)	Station Name	StationName	ct99

(b) Metadata de imagen DICOM anonimizada.

**Figura B.1:** Visualización de la metadata correspondiente al perfil de confidencialidad dentro de los archivos DICOM.

# Apéndice C

## Visualización de imagen de TC

### C.1. Script `imprimir_imagen.py`

Este script contiene todas las funciones necesarias para la visualización de una imagen DICOM, y para la visualización de un examen de TC en diferentes planos anatómicos. Con este fin, se implementaron varias funciones las cuales permiten la lectura de archivos DICOM, transformación a unidades Hounsfield, la corrección de píxeles y la visualización de una imagen de TC. El rango de unidades Hounsfield puede ser seleccionado y ajustado a la escala de grises por el usuario. Adicionalmente, para un examen de TC con varios archivos DICOM, se llevó a cabo la obtención de la matriz en 3D, el escalamiento de ejes para la representación de los archivos en diferentes planos anatómicos (axial, sagital, coronal) y finalmente la obtención de la imagen con los cortes en los distintos planos.

### C.2. Notebook Visualización de imágenes DICOM

Es un archivo `.ipynb` donde se documentan los procesos y consideraciones llevados a cabo para realizar el script `imprimir_imagen.py`, que permite la visualización de las imágenes en diferentes planos anatómicos. Este archivo también contiene explicaciones bastante detalladas de los códigos mediante celdas de texto Markdown (en formato LaTeX) que generan un documento muy amigable al usuario, y permite explicar el código con texto y gráficas. Además, en el Notebook se puede acceder y ejecutar las funciones que se crearon para visualizar las imágenes de archivos DICOM, como módulos de Python.

# Apéndice D

## Segmentación de tejido pulmonar

### D.1. Script `segmentacion_pulmonar.py`

En este archivo se encuentran guardadas todas las funciones que se desarrollaron para realizar una segmentación de pulmones en imágenes DICOM, mediante el algoritmo de Watershed basado en marcadores. Las funciones implementadas van desde la generación de la imagen de gradiente, marcadores internos y externos, la aplicación del algoritmo, hasta funciones que permiten automatizar y simplificar los diferentes procesos que se llevan a cabo en la segmentación.

### D.2. Notebook Segmentación de tejido pulmonar

Es un cuaderno de Jupyter, en el cual se implementan las funciones que son necesarias para segmentación de pulmones, de una manera interactiva y de fácil entendimiento. Al poder implementar texto y ejecutar código, esta clase de archivos resulta muy útil para personas que deseen comprender las técnicas realizadas, ya que el código viene acompañado de explicaciones y pruebas realizadas a los diferentes exámenes.

### D.3. Notebook Resultados (Segmentación de Pulmones)

Es un cuaderno de Jupyter en donde se encuentran todas las pruebas de segmentación de pulmones que se realizaron a las 18 muestras de TC.

#### D.4. Descripción de los exámenes de tórax para segmentación

Patient	Number of Images	Patient Sex	Study Description	Protocol Name	Modality	Exposure Time [msec]	Patient Position	Manufacturer	Manufacturer ModelName
0	25	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
1	29	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
2	29	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
3	44	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
4	50	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
5	9	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
6	240	F	TCC	6.1 AP	CT	800	FFS	GEMS	LightSpeed VCT
7	47	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
8	50	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
9	50	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
10	60	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
11	45	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
12	60	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
13	40	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
14	61	F	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
15	38	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT
16	30	M	PT	10.7 RT	CT	1000	HFS	GEMS	LightSpeed VCT

**Cuadro D.1:** Características de los exámenes usados para las pruebas de segmentación de pulmones. 10.7RT (10.7 RADIOTERAPIA TORAX), PT (PLANEAMIENTO DE TERAPIA), GEMS (GE MEDICAL SYSTEMS), 6.1 AP (6.1 ABDOMEN Y PELVIS), TCC (TC CORPORAL), HFS (HEAD FIRST-SUPINE), FFS (FEET FIRST-SUPINE).

# Bibliografía

- [1] Suzanne Amador Kane and Boris A. Gelman. *Introduction to physics in Modern Medicine*, chapter 5, pages 189–262. CRC Press, Taylor et Francis Group, 3rd edition, 2020.
- [2] NEMA. Does dicom work with other standards-development organizations? URL: <http://dicom.nema.org/dicom/geninfo/Brochure.pdf>. [Online; accessed 15-April-2021].
- [3] D. L. Bailey. *Nuclear medicine physics: a handbook for teachers and students*, chapter 1, pages 1–48. IAEA, Vienna, 2015.
- [4] Worldwide cancer data: World cancer research fund international. URL: <http://www.wcrf.org/dietandcancer/worldwide-cancer-data/>, May 2021. [Online; accessed 1-July-2021].
- [5] R. Shojaii, J. Alirezaie, and P. Babyn. Automatic lung segmentation in ct images using watershed transform. *IEEE International Conference on Image Processing 2005*, 2005.
- [6] Thomas Heeneman and Mark Hoogendoorn. Lung nodule detection by using deep learning. *University of Amsterdam, Research Paper*, 2018.
- [7] Fernand Meyer. The watershed concept and its use in segmentation : a brief history. *ArXiv*, abs/1202.0216, 2012.
- [8] Amy S. Tidwell. Advanced imaging concepts: A pictorial glossary of ct and mri technology. *Clinical Techniques in Small Animal Practice*, 14(2):65–111, 1999.
- [9] Faiz M. Khan and John P. Gibbons. *Khan's the Physics of Radiation Therapy*, chapter 12, pages 195–199. LWW (PE), 5th edition, 2014.
- [10] John R. Haaga and Daniel Boll. *Ct and Mri of the whole body, 2-volume set*, chapter 1, pages 3–7. Elsevier - Health Sciences Div, 6th edition, 2016.

- [11] K. Kirk Shung, Michael B. Smith, and Benjamin M.W. Tsui. *Principles of Medical Imaging*, chapter 1, pages 1–77. Academic Press, San Diego, 1992.
- [12] Perry Sprawls. *Physical principles of medical imaging*, chapter 23, pages 343–355. Medical Physics Pub., Madison, WI, 2nd edition, 1995.
- [13] Juan Carlos Ramirez Giraldo, Carolina Arboleda Clavijo, and Cynthia H. McCollough. TOMOGRAFÍA COMPUTARIZADA POR RAYOS X: FUNDAMENTOS Y ACTUALIDAD. *Revista Ingeniería Biomédica copyrightdica*, 2:54 – 66, 12 2008.
- [14] Ervin B. Podgorsak. *Radiation physics for medical physicists*, chapter 7, pages 277–355. Springer, Cham, third edition, 2016.
- [15] William R. Hendee and E. Russell Ritenour. *Medical imaging physics*, chapter 4, pages 45–69. Wiley, New York, NY, 4th edition, 2002.
- [16] Lee W. Goldman. Principles of ct and ct technology. *Journal of Nuclear Medicine Technology*, 35(3):115–128, 2007.
- [17] Chris Guy and Dominic Ffytche. *An introduction to the principles of Medical Imaging*, chapter 4, pages 115–150. Imperial College Press, London, 2005.
- [18] Jerrold T. Bushberg, J. Anthony Seibert, Edwin Marion Leidholdt, John M. Boone, and Craig K. Abbey. *The essential physics of medical imaging*, chapter 4, pages 87–91. Wolters Kluwer, Philadelphia, 3rd edition, 2012.
- [19] Lucas L. Geyer, U. Joseph Schoepf, Felix G. Meinel, John W. Nance, Gorka Bastarrika, Jonathon A. Leipsic, Narinder S. Paul, Marco Rengo, Andrea Laghi, Carlo N. De Cecco, and et al. State of the art: Iterative ct reconstruction techniques. *Radiology*, 276(2):339–357, 2015.
- [20] Alfredo Tirado-Ramos, Jingkun Hu, and K P Lee. Information object definition-based unified modeling language representation of dicom structured reporting: a case study of transcoding dicom to xml. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC349388/>, 2002. [Online; accessed 22-April-2021].
- [21] Otech Inc. Dicom. URL: [https://www.otechimg.com/\\_getDoc.cfm?id=43&starred=1](https://www.otechimg.com/_getDoc.cfm?id=43&starred=1). [Online; accessed 18-May-2021].
- [22] Oleg S. Pianykh. *Digital Imaging and Communications in Medicine (DICOM): A practical introduction and survival guide*, chapter 5, pages 27–80. Springer Open, Berlin, 2nd edition, 2012.

- [23] NEMA. Dicom ps3.1 2021b - introduction and overview. URL: <http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>, 2021. [Online; accessed 19-May-2021].
- [24] NEMA. Ps3.3 dicom ps3.3 2021b - information object definitions. URL: <http://dicom.nema.org/medical/dicom/current/output/pdf/part03.pdf>, 2021. [Online; accessed 18-May-2021].
- [25] Sven Siekmann. *Analysis of new concepts and definitions in DICOM second generation RADIOTHERAPY objects based on an experimental implementation*. PhD thesis, Verlag nicht ermittelbar, Erscheinungsort nicht ermittelbar, 2012.
- [26] NEMA. Ps3.5 dicom ps3.5 2021b - data structures and encoding. URL: <http://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf>, 2021. [Online; accessed 20-May-2021].
- [27] NEMA. Ps3.6 dicom ps3.6 2021b - data dictionary. URL: <http://dicom.nema.org/medical/dicom/current/output/pdf/part06.pdf>, 2021. [Online; accessed 13-April-2021].
- [28] NEMA. A.3 computed tomography image iod. URL: [http://dicom.nema.org/dicom/2013/output/chtml/part03/sect\\_A.3.html](http://dicom.nema.org/dicom/2013/output/chtml/part03/sect_A.3.html), 2013. [Online; accessed 22-April-2021].
- [29] Jiantin Yuan, Wenqiang Zhang, Wei Wu, Xiaoyi Lv, Xue Bai, Chen Chen, and Yue Hong. Desensitization and cleaning technology of ct image of parotid tumor based on dicom. *Journal of Physics: Conference Series*, 1576:012027, 2020.
- [30] Wayne Newhauser, Timothy Jones, Stuart Swerdloff, Warren Newhauser, Mark Cilia, Robert Carver, Andy Halloran, and Rui Zhang. Anonymization of dicom electronic medical records for radiation therapy. *Computers in Biology and Medicine*, 53:134–140, 2014.
- [31] NEMA. E attribute confidentiality profiles. URL: [http://dicom.nema.org/dicom/2013/output/chtml/part15/chapter\\_E.html#table\\_E.1-1](http://dicom.nema.org/dicom/2013/output/chtml/part15/chapter_E.html#table_E.1-1), 2013. [Online; accessed 10-May-2021].
- [32] IMAGE Information Systems Ltd. | IMAGE Information Systems Europe GmbH. *iQ-LITE USER MANUAL*. London, UK, 2016.

- [33] LLC Innolitics. Dicom standard browser. URL: <https://dicom.innolitics.com/ciods/respiratory-waveform/general-series/00181030>. [Online; accessed 10-May-2021].
- [34] LLC Innolitics. Dicom standard browser. URL: <https://dicom.innolitics.com/ciods/raw-data/general-series/00185100>. [Online; accessed 20-July-2021].
- [35] Ravishankar Chityala and Sridevi Pudipeddi. *Image Processing and Acquisition using Python*, chapter 3, pages 37–57. Taylor & Francis, 2nd edition, 06 2020.
- [36] Nouredine Alaa. Image processing with python: An introduction. Technical report, Cadi Ayyad University, 03 2021.
- [37] Darcy Mason. Pydicom. URL: <https://pydicom.github.io>, Jun 2020. [Online; accessed 1-July-2021].
- [38] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [39] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. Scikit-image: Image processing in python. *PeerJ*, 2, 2014.
- [40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020.
- [41] Fernando Pérez. Jupyter. URL: <https://bids.berkeley.edu/research/project-jupyter>, Aug 2020. [Online; accessed 1-July-2021].
- [42] Fernando Pérez. Jupyterhub. URL: <https://datahub.berkeley.edu/hub/>. [Online; accessed 1-July-2021].
- [43] Fernando Pérez. Jupyterhub. URL: <https://statistics.berkeley.edu/computing/jupyterhub>. [Online; accessed 1-July-2021].
- [44] John T. Hathcock and Russ L. Stickle. Principles and concepts of computed tomography. *Veterinary Clinics of North America: Small Animal Practice*, 23(2):399–415, 1993.

- [45] Jerry L. Prince and Jonathan M. Links. *Medical Imaging Signals and Systems*, chapter 7, pages 186–213. Pearson Education, Upper Saddle River, NJ, 2nd edition, 2015.
- [46] Andreas Maier, Stefan Steidl, Vincent Christlein, and Joachim Hornegger. *Medical imaging systems: an introductory guide*, chapter 7, pages 119–186. Springer Open, Cham, Switzerland, 2018.
- [47] NEMA. C.7.6.3 image pixel module (current). URL: [http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect\\_C.7.6.3.html](http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect_C.7.6.3.html), 2018. [Online; accessed 10-May-2021].
- [48] NEMA. C.7.6.3 image pixel module. URL: [http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect\\_C.7.6.3.html#table\\_C.7-11c](http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect_C.7.6.3.html#table_C.7-11c), 2018. [Online; accessed 10-May-2021].
- [49] NEMA. Dicom ps3.3 2018d - information object definitions. URL: [http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect\\_C.11.html](http://dicom.nema.org/medical/Dicom/2018d/output/chtml/part03/sect_C.11.html), 2018. [Online; accessed 10-May-2021].
- [50] NEMA. Dicom ps3.3 2021b - information object definitions. URL: [http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect\\_C.7.6.2.html](http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.2.html), 2021. [Online; accessed 10-May-2021].
- [51] NEMA. 10.7 basic pixel spacing calibration macro. URL: [http://dicom.nema.org/dicom/2013/output/chtml/part03/sect\\_10.7.html](http://dicom.nema.org/dicom/2013/output/chtml/part03/sect_10.7.html), 2013. [Online; accessed 10-May-2021].
- [52] Candace Makeda Moore. Segmentation: Radiology reference article. URL: <https://radiopaedia.org/articles/segmentation>. [Online; accessed 20-May-2021].
- [53] Yuheng Song and Hao Yan. Image segmentation techniques overview. *2017 Asia Modelling Symposium (AMS)*, 2017.
- [54] Gagandeep Kaur and Jyotirmoy Chhaterji. A survey on medical image segmentation. *International Journal of Science and Research*, 6:1305–1311, 2017.
- [55] Sujata Saini and Komal Arora. A study analysis on the different image segmentation techniques. *International Journal of Information & Computation Technology*, 4(14):1445–1452, 2014.

- [56] S. Umaa Mageswari, M. Sridevi, and C. Mala. An experimental study and analysis of different image segmentation techniques. *Procedia Engineering*, 64:36–45, 2013.
- [57] Ahmed Elnakib, Georgy Gimel'farb, Jasjit S. Suri, and Ayman El-Baz. Medical image segmentation: A brief survey. *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, page 1–39, 2011.
- [58] Khalid Nazim Sattar. Tadoc : Tool for automated detection of oral cancer. *International Journal of Advanced Computer Science and Applications*, 11(3), 2020.
- [59] Jigna.J Patel. Various segmentation techniques in image processing. *International Journal of Engineering Science and Computing*, 6:12781–12783, 06 2017.
- [60] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*, chapter 3,9,10. Pearson, New York, NY, 4th edition, 2018.
- [61] NASA. Hierarchical image segmentation - logical predicate segmentation. URL: <https://science.gsfc.nasa.gov/606.3/TILT0N/logical-predicate-segmentation.html>. [Online; accessed 19-July-2021].
- [62] EA Zanaty and Said Ghoniemy. Medical image segmentation techniques: An overview. *International Journal of informatics and medical data processing (JIMDP)*, 1:16–37, 2016.
- [63] Dilpreet Kaur and Yadwinder Kaur. Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, 3(5):809–814, 2014.
- [64] K. Parvati, B. S. Prakasa Rao, and M. Mariya Das. Image segmentation using gray-scale morphology and marker-controlled watershed transformation. *Discrete Dynamics in Nature and Society*, 2008:1–8, 2008.
- [65] D. Wang and P. Vallotton. Improved marker-controlled watershed segmentation with local boundary priors. *2010 25th International Conference of Image and Vision Computing New Zealand*, 2010.
- [66] Pierre Soille. *Morphological image analysis: Principles and applications*, chapter 9, pages 267–290. Springer, Berlin, 2nd edition, 2003.
- [67] Skitimage. Morphological filtering. URL: <https://scikit-image.org/docs/dev/autoexamples/applications/plotmorphology.html>. [Online; accessed 18-April-2021].

- [68] MathWorks. Strel. URL: <https://la.mathworks.com/help/images/structuring-elements.html>. [Online; accessed 18-April-2021].
- [69] John C. Russ. *The Image Processing Handbook*, chapter 8, pages 463–469. CRC Press, 7th edition, 2016.
- [70] William K. Pratt. *Digital Image Processing: PIKS scientific inside*, chapter 14, pages 442–465. Wiley, Hoboken, NJ, 4th edition, 2007.
- [71] Gzuidhof. Full preprocessing tutorial. URL: <https://www.kaggle.com/gzuidhof/full-preprocessing-tutorial>, Feb 2017. [Online; accessed 1-July-2021].
- [72] Ankasor. Improved lung segmentation using watershed. URL: <https://www.kaggle.com/ankasor/improved-lung-segmentation-using-watershed>, Jan 2017. [Online; accessed 1-July-2021].
- [73] Sharpened Productions. Extensión de archivo .npy. URL: <https://fileinfo.com/extension/npy#:~:text=An%20NPY%20file%20is%20a,incluces%20dtype%20and%20shape%20information>. [Online; accessed 22-April-2021].
- [74] Wilhelm Burger and Mark J. Burge. *Principles of Digital Image Processing Fundamental Techniques*, chapter 6, pages 131–142. Springer London, London, 2009.
- [75] Jason M. Kinser. *Image operators: image processing in Python*, chapter 13, pages 189–193. CRC Press/Taylor & Francis Group, Boca Raton, FL, 2019.
- [76] S. Hu, E.A. Hoffman, and J.M. Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images. *IEEE Transactions on Medical Imaging*, 20(6):490–498, 2001.
- [77] Katherine, R Rulaningtyas, and K Ain. Ct scan image segmentation based on hounsfield unit values using otsu thresholding method. *Journal of Physics: Conference Series*, 1816(1):012080, 2021.
- [78] Edgardo Franco and Adriano Pennacchia. Análisis digital de imágenes tomográficas sin contraste para la búsqueda de tumores cerebrales. tesis presentada para obtener el grado de maestro en ciencias en computación. Master’s thesis, Instituto Politécnico Nacional, México D.F., 2011.
- [79] NEMA. Dicom ps3.3 2016a - information object definitions. URL: [http://dicom.nema.org/medical/dicom/2016a/output/chtml/part03/sect\\_C.8.2.html](http://dicom.nema.org/medical/dicom/2016a/output/chtml/part03/sect_C.8.2.html), 2016. [Online; accessed 10-May-2021].