



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

**SOFTWARE DE APOYO AL ANÁLISIS
RADIOLÓGICO DE TOMOGRAFÍAS
AXIALES COMPUTARIZADAS**

Que para cumplir la opción de titulación curricular en la carrera de:
“Ingeniería en Sistemas Computacionales”

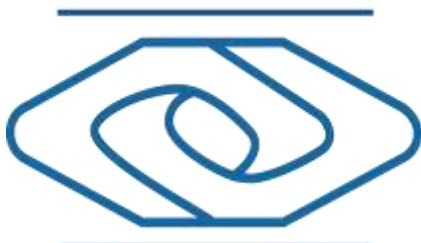
Presentan

**León Díaz Raúl Alberto
Osnaya Gómez Alexis Alan
Ríos López José Alberto
Santiago Nieves Edgar Augusto**

Directores

M. en C. Edgardo Franco Martínez

Dr. Jorge Luis Rosas Trigueros



Índice general

Lista de figuras	5
Lista de tablas	7
1. Introducción	9
1.1. Antecedentes	9
1.2. Planteamiento del problema	11
1.3. Propuesta de solución	11
1.4. Justificación	12
1.5. Objetivos	13
1.5.1. Objetivo General	13
1.5.2. Objetivos Particulares	13
2. Marco teórico	14
2.1. Estado del arte	14
2.2. Imagen	15
2.3. Imágenes médicas	16
2.4. Tomografía computarizada	17
2.5. DICOM	19
2.6. Escala de Hounsfield	20
2.7. Segmentación	21
2.7.1. Umbralización	22
2.7.2. Región creciente	24
2.7.3. Clasificadores	26
2.7.4. Clustering	27
3. Análisis	29
3.1. Comparación de software	29
3.1.1. Sistemas operativos	29

3.1.2. Lenguajes de programación	30
3.1.3. Elección de librerías	32
3.2. Elección de herramientas para el desarrollo	33
3.2.1. Software	33
3.2.2. Metodología	34
3.2.3. Hardware	37
3.3. Análisis de factibilidad	37
3.4. Plan de manejo de riesgos	45
3.4.1. Identificación de riesgos	45
3.4.2. Análisis del riesgo	46
3.4.3. Plan de contención de riesgos	47
3.5. Definición de requerimientos del sistema	48
3.5.1. Requerimientos funcionales	49
3.5.2. Requerimientos no funcionales	50
4. Diseño del sistema	52
4.1. Arquitectura del sistema	52
4.2. Casos de uso	53
4.2.1. Caso de uso general	53
4.2.2. Caso de uso decodificación	54
4.2.3. Caso de uso selección de región	55
4.2.4. Caso de uso tratamiento de imagen	56
4.3. Descripción de casos de uso	57
4.3.1. Caso de uso seleccionar archivos	57
4.3.2. Caso de uso decodificar DICOM	59
4.3.3. Caso de uso obtener arreglo	60
4.3.4. Caso de uso seleccionar región	61
4.3.5. Caso de uso tratar imagen	63
4.3.6. Caso de uso umbralizar	64
4.3.7. Caso de uso segmentar regiones	65
4.3.8. Caso de uso clasificar regiones	66
4.3.9. Caso de uso analizar regiones	67
4.3.10. Caso de uso manipular arreglo	68
4.3.11. Caso de uso mostrar tomografía	69
4.4. Diagramas de secuencia	70
4.4.1. Secuencia: Decodificar DICOM	70
4.4.2. Secuencia: Seleccionar archivos	71
4.4.3. Secuencia: Obtener arreglo	72

4.4.4. Secuencia: Seleccionar regiones	73
4.4.5. Secuencia: Analizar imagen	74
4.4.6. Secuencia: Clasificar regiones	75
4.4.7. Secuencia: Manipular arreglo	76
4.4.8. Secuencia: Segmentar regiones	77
4.4.9. Secuencia: Tratar imagen	78
4.4.10. Secuencia: Umbralizar	79
4.4.11. Secuencia: Mostrar tomografía	80
5. Desarrollo	81
5.1. Decodificación de DICOM	81
5.2. Comunicación entre lenguajes	82
5.3. Umbralización	82
5.4. Segmentación	83
5.4.1. Región creciente	83
5.4.2. Split and merge	83
5.5. Clasificación	83
5.5.1. K-means	84
5.5.2. Fuzzy C-Means	84
6. Pruebas	85
6.1. Umbralización	85
6.2. Contraste	93
6.3. Clasificadores	96
6.4. Segmentación	98
6.5. Representación en 3D	99
7. Conclusiones y trabajo futuro	100
7.1. Conclusiones	100
7.2. Trabajo a futuro	100
A. Código K-Means	102
B. Código Fuzzy C-Means	104
C. Código Split and Merge	107
D. Código Región creciente	109
E. Código generación de contraste	110

Índice de figuras

1.1.	Diagrama del funcionamiento del sistema.	12
2.1.	Pixel y voxel.	16
2.2.	Cortes axiales en TAC.	18
2.3.	Estructura de un archivo DICOM.	20
2.4.	Escala de Hounsfield	21
2.5.	Métodos de características espaciales y región creciente. (a)Histograma con 3 aparentes clases. (b)Característica espacial en 2-D. (c)Ejemplo de región creciente.	23
2.6.	Ejemplo iteración algoritmo Split and Merge.	25
3.1.	Flujo metodología incremental.	35
4.1.	Diagrama de casos de uso general.	53
4.2.	Diagrama caso de uso decodificación.	54
4.3.	Diagrama caso de uso de selección de región.	55
4.4.	Diagrama caso de uso de tratamiento de imagen.	56
4.5.	Secuencia:Decodificar DICOM.	70
4.6.	Secuencia:Seleccionar archivos.	71
4.7.	Secuencia:Obtener arreglo.	72
4.8.	Secuencia:Seleccionar regiones.	73
4.9.	Secuencia:Analizar imagen.	74
4.10.	Secuencia:Clasificar regiones.	75
4.11.	Secuencia:Manipular arreglo.	76
4.12.	Secuencia:Segmentar regiones.	77
4.13.	Secuencia:Tratar imagen.	78
4.14.	Secuencia:Umbralizar.	79
4.15.	Secuencia:Mostrar tomografía.	80
6.1.	Umbralización de fluido cerebro espinal.	86

6.2. Umbralización de grasa.	87
6.3. Umbralización de pulmón.	88
6.4. Umbralización de hueso compacto.	89
6.5. Umbralización de hueso esponjoso.	90
6.6. Umbralización de sangre.	91
6.7. Umbralización de sustancia cerebral gris.	92
6.8. TAC con un contraste completo.	93
6.9. TAC con el contraste recomendado para cerebro.	94
6.10. TAC con el contraste recomendado para hueso.	94
6.11. TAC con el contraste recomendado para partes blandas.	95
6.12. TAC con el contraste recomendado para pulmón.	95
6.13. Algoritmo Fuzzy C-Means aplicado a una TAC.	96
6.14. Algoritmo K-Means aplicado a una TAC.	97
6.15. Región creciente en zona abdominal.	98
6.16. Reconstrucción 3D de un craneo.	99
F.1. Diagrama de clases del sistema.	112

Índice de tablas

3.1.	Tabla comparativa de sistemas operativos	30
3.2.	Tabla comparativa de lenguajes de programación.	31
3.3.	Especificaciones de hardware	37
3.4.	Sistemas operativos a utilizar.	39
3.5.	IDE de desarrollo.	39
3.6.	Herramientas de apoyo para documentación.	40
3.7.	Costos equipos de cómputo.	40
3.8.	Costos sistemas operativos.	41
3.9.	Costos IDE.	41
3.10.	Costos software de apoyo.	42
3.11.	Costos servicios.	42
3.12.	Estimación de sueldos	43
3.13.	Ánalisis PERT del sistema	44
3.14.	Costos sistema	45
3.15.	Análisis de riesgo	47
3.16.	Plan de contención de riesgos	48
3.17.	Requisitos funcionales del sistema	50
3.18.	Requisitos no funcionales del sistema	51
4.1.	Caso de uso seleccionar archivos.	57
4.2.	Caso de uso decodificar DICOM.	59
4.3.	Caso de uso obtener arreglo.	60
4.4.	Caso de uso seleccionar región.	61
4.5.	Caso de uso tratar imagen.	63
4.6.	Caso de uso umbralizar.	64
4.7.	Caso de uso segmentar regiones.	65
4.8.	Caso de uso clasificar regiones.	66
4.9.	Caso de uso analizar regiones.	67
4.10.	Caso de uso manipular arreglo.	68

4.11. Caso de uso mostrar tomografía.	69
5.1. Etiquetas DICOM usadas en el sistema.	82

Capítulo 1

Introducción

1.1. Antecedentes

Los médicos consideran que la imagen médica ha sido y es, con mucha diferencia, el avance técnico que mayor impacto ha tenido en su práctica clínica.

El hombre(como otros mamíferos) es un animal esencialmente visual. Gran parte del cerebro humano está dedicado al procesamiento de la información visual; muchas estrategias mnemotécnicas y de aprendizaje rápido tratan de obtener ventaja de este hecho. Por esa razón, la información diagnóstica que proporcionan los sistemas de imagen es enormemente apreciada por el médico, hasta el punto de que, hoy en día, resulta difícil encontrar situaciones clínicas en las que no se haya hecho uso de una o más técnicas de imagen.

En 1895 se produce el descubrimiento que da lugar a la imagen médica como prueba diagnóstica cuya importancia no ha dejado de crecer hasta nuestros días: Wilhelm Rontgen, experimentando con descargas eléctricas en tubos de Crookes, observó que brillaba una placa de platinocianuro de bario al otro lado de la habitación, a pesar de estar el tubo encerrado en una caja de cartón. Además comprobó que esos "misteriosos rayos"(bautizados por esta razón como rayos X) podían atravesar algunos objetos dejando su sombra en la pantalla. [1]

En el año de 1972 se publicó en la revista *British Journal of Radiology* un artículo del ingeniero Sir Godfrey Newbold Hounsfield sobre una técnica basada en rayos X, llamada tomografía computarizada, la cual hacía uso de métodos matemáticos desarrollados una decada antes por A. M. Cormack. El nuevo método propuesto rea-

lizaba una división de la cabeza en varios cortes los cuales irradiaban a través de sus bordes, por lo cual la radiación emitida podía ser aislada dentro de la misma porción. La ventaja que este método presentaba frente a la técnica convencional de rayos X, era que la información no presentaba alteraciones debidas al material presente a los lados del corte en cuestión.

Con el fin de superar a la radiología convencional, el nuevo método pretendía superar 3 limitantes; la primera, la imposibilidad de mostrar toda la información de una escena tridimensional en una imagen radilógica bidimensional, debido a la superposición de los objetos en la imagen obtenida; la segunda limitante a vencer era la escasa capacidad de distinguir los tejidos blandos; finalmente, no era posible cuantificar las densidades que tenían los tejidos. [2]

El aumento de la utilización de la tecnología en la medicina buscó utilizar las computadoras para hacer diagnósticos automáticos con apoyo de las nuevas modalidades de imágenes médicas, sin embargo factores como la baja eficiencia de los equipos o la falta de accesibilidad a imágenes médicas no lo permitió. Posteriormente en los años 80's surgió un nuevo enfoque en el que se asumió que las salidas proporcionadas por la computadora podrían ser utilizadas por los radiólogos sin reemplazarlos, fue así como surgieron las herramientas de diagnóstico asistido por computadora o CAD(Computer-Aided diagnosis). En las CAD, los radiólogos utilizan la computadora para tener un "segunda opinión" sin embargo son ellos quienes toman la decisión final del diagnóstico. [3]

Con el surgimiento de la tomografía computarizada y otras modalidades de diagnóstico digital con imágenes usadas en las CAD, el ACR(American College of Radiology) y el NEMA(National Electrical Manufacturers Association) se dieron cuenta que se debía crear una forma en que dispositivos de diferentes fabricantes pudieran intercambiar información de las imágenes obtenidas. En 1993 surge DICOM(Digital Imaging and Communication in Medicine) como un estandar ante la problemática planteada. [4] DICOM no es sólo una imagen o formato de archivo, DICOM se encarga de cubrir la transferencia de información, almacenamiento y un protocolo de visualización construido y diseñado para satisfacer todas las necesidades de la medicina contemporanea. Sin duda alguna DICOM es quien gobierna la práctica de la medicina digital. [5]

La innovación tecnológica fue un parte aguas en la historia de la medicina y hoy en día es uno de los más importantes motores para el desarrollo de esta, permitiendo

cada vez mejores y más rápidos diagnósticos de salud.

1.2. Planteamiento del problema

Hoy en día el diagnóstico asistido por computadora es usado principalmente en los métodos de diagnóstico que emplean contraste. Los métodos sin contraste, también llamados simples, no son comúnmente asistidos por computadora después de haber realizado el estudio en el equipo médico especializado.¹

Los equipos necesarios para la generación de estudios como las TAC cuentan con herramientas de software muy potentes como el caso de SOMATOM por parte de la compañía SIEMENS o IQon Spectral de Philips, sin embargo los costos de las licencias así como de los equipos son muy elevados. [6], [7]

Las herramientas de asistencia que son accesibles en muchas ocasiones se limitan a algunas funciones básicas de análisis como rotar, escalar o hacer mediciones a las imágenes; algunas otras ofrecen más funciones como representación en 3D pero sólo es posible cuando las imágenes cuentan más de un tipo de corte sobre el cuerpo humano. [8], [9], [10], [11]

1.3. Propuesta de solución

Para dar solución a la problema anteriormente planteado se propone diseñar y desarrollar una herramienta de visualización y tratamiento de tomografías axiales computarizadas a partir de la información ofrecida por el archivo DICOM que las contiene, la cual permitirá a los especialistas una fácil y mejorada interpretación de los resultados.

El primer paso será la decodificación de un archivo DICOM, una vez decodificada la imagen el usuario podrá observar la información contenida en el archivo DICOM mediante un visor. Adicionalmente, el usuario podrá utilizar diversas herramientas del sistema para que la información sea analizada y así obtener propiedades particulares de la imagen.

En la figura 1.1 se observa el diagrama general de funcionamiento del sistema.

¹Información obtenida a través de una plática con Noé Hernandez, radiólogo en la Clínica de Medicina Familiar División del Norte.

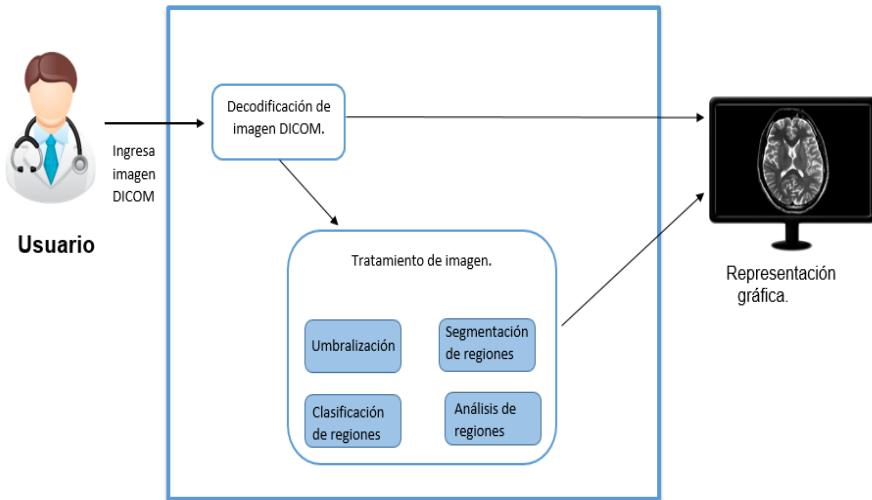


Figura 1.1: Diagrama del funcionamiento del sistema.
Imagen de elaboración propia.

1.4. Justificación

A lo largo de los años la evolución de la ciencia y la tecnología han permitido que el campo de la medicina se desarrolle y avance en un gran nivel, al permitir el desarrollo de medicinas y aparatos quirúrgicos que facilitan y minimizan el error dentro de los procesos, desafortunadamente no en todos los campos del área esto se ha logrado, cuando hablamos de errores médicos debemos tomar en cuenta que las consecuencias que estos pueden traer son muy delicadas pues en muchas ocasiones representan la vida o muerte de los pacientes.

La interpretación de los estudios tomográficos radiológicos se realiza comúnmente con base a un análisis visual de las imágenes adquiridas de los equipos imageneológicos por parte del médico radiólogo, estos análisis son susceptibles a malas interpretaciones debido a error humano, por esto es que decidimos realizar este trabajo con el fin de ayudar a la interpretación de estos estudios.

El uso de algoritmos computacionales para la delineación de estructuras anatómicas y otras regiones de interés del cuerpo humano se han vuelto muy importantes en la asistencia y realización de tareas radiológicas específicas. Estos algoritmos, llamados algoritmos de segmentación de imagen, tienen un papel muy importante en

diversas aplicaciones de imágenes biomédicas , como la cuantificación del volumen de tejidos, diagnóstico, localización de patologías, estudio de estructuras anatómicas, planeación del tratamiento y cirugía computarizada. [12]

Este trabajo terminal beneficia a la sociedad en general pues cualquiera puede necesitar un estudio por medio de una TAC y gracias al software propuesto se podrá hacer un mejor análisis de la imagen.

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar un software de asistencia al especialista para la interpretación adecuada de las tomografías axiales computarizadas.

1.5.2. Objetivos Particulares

- Leer e interpretar la información contenida en una imagen DICOM.
- Umbralizar una imagen médica en formato DICOM.
- Segmentar regiones determinadas dentro de una tomografía axial computarizada.
- Clasificar regiones con base a la distancia Euclíadiana existente entre puntos específicos.
- Analizar distintas propiedades de regiones específicas establecidas por el especialista.
- Facilitar la interpretación de tomografías mediante un visor en 2D.
- Proporcionar una visualización en 3D de las tomografías axiales computarizadas.

Capítulo 2

Marco teórico

En este capítulo se detallarán algunos términos que ayudarán a una mayor comprensión del funcionamiento del sistema.

2.1. Estado del arte

La solución que se propone ante la problemática planteada no es una solución completamente nueva pues se han desarrollado anteriormente otros softwares que poseen una idea similar pero se atacan con un enfoque diferente. A continuación se listan algunos sistemas y trabajos similares a la propuesta realizada.

Image Analysis Software [13]

Algunas características ofrecidas por el software son las siguientes:

- Analiza modelos celulares complejos.
- Gestiona eficazmente grandes cantidades de datos.
- Visualiza modelos de células en 2D o 3D en tiempo real.
- Muestra tejidos y cuantifica células.

3D-Doctor [14]

El software proporciona las siguientes características:

- Puede leer imágenes DICOM, TIFF, BMP, JPEG, Interfile, PNG, PGM, GIF.

- Contiene modelos 3D de exportación STL(ASCII y binario), 3D Studio(3DS)entre otros para la rápida creación de prototipos.
- Mide área, superficie 3D, volumen 3D, distancia, perfil y región de un histograma.
- Representación de diferentes tejidos.
- Imágenes 3D CT/MRI pueden ser cortadas de nuevo facilmente a lo largo de un eje arbitrario.
- Corte de imágenes para corregir las rebanadas de grosor desigual, el redimensionado de volumen y la rotación de imagen.

Análisis digital de imágenes tomográficas sin contraste para la búsqueda de tumores cerebrales [15]

La tesis presentada nos ofrece estas características:

- Análisis digital de imágenes y reconocimiento de patrones para la búsqueda de tumores cerebrales.
- Codificación y extracción de la imagen a partir del archivo digital proporcionada por los sistemas PACS de la actualidad.

Reconstrucción tridimensional de estructuras internas del cuerpo humano a partir de tomografías axiales computarizadas [16]

Las características de este trabajo terminal desarrollado en la Escuela Superior de Cómputo se presentan a continuación:

- Sistema de reconstrucción tridimensional de estructuras internas del cuerpo humano mediante tomografías axiales computarizadas(TAC) en formato DICOM("Digital Imaging and Communication in Medicine"). Dicho sistema proporciona una amplia manipulación del objeto generado en 3D.

2.2. Imagen

Una imagen puede ser definida como una función de dos dimensiones $f(x, y)$, donde x y y son coordenadas espaciales, y a la amplitud de la función f en cualquier par de coordenadas (x, y) se le llama *intensidad* o *niveles de grises* de la imagen en

ese punto. Cuando x , y y el valor de intensidad de la función f son todos valores finitos y discretos, llamamos a la imagen una imagen digital. Notese que una imagen digital está compuesta de un número finito de elementos, cada uno de los cuales tiene una localización específica y un valor. Si cada posición de la imagen tiene una única medida, la imagen es llamada "imagen escalar", si las posiciones tienen más de una medida se les llama "imágenes vectoriales" o "multicanal". En las imágenes discretas en 2D la localización de cada medida es llamada "pixel" y en imágenes 3D se llama "voxel", en la figura 2.1 se observa la diferencia entre ambos términos. [12] [17]

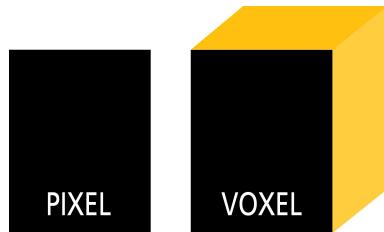


Figura 2.1: Pixel y voxel.
Imagen obtenida de

<https://carenstreamdentalblogdotcom1.wordpress.com/2014/02/19/three-dimensional-basics-pixels-and-voxels/>

2.3. Imágenes médicas

Una imagen médica es un conjunto de técnicas y procesos usados para crear imágenes del cuerpo humano, o partes de él, con propósitos clínicos o para la ciencia médica.

Representa gráficamente una distribución espacial de una o más propiedades físicas o químicas del interior del cuerpo humano. Dos de los parámetros de mayor importancia son el contraste y la resolución.

El contraste se define como la diferencia en la intensidad entre un punto de la imagen y determina lo que podemos ver en ella, en el caso del contraste en una imagen médica es muy importante tener el conocimiento de diversos parámetros físicos o químicos que ayuden a entender lo que está siendo representado.

La resolución por su parte se define como la nitidez con la que se percibe una imagen observada por un instrumento óptico. Nos ayuda a encontrar todas las ca-

racterísticas de la imagen para así distinguir y analizar detalles. [18]

Modalidades de las imágenes médicas. [19]

- Radiología: Es el uso médico de la radiación para diagnosticar y tratar diversos problemas de salud. A partir de la utilización de rayos gamma, rayos X y otras clases de rayos, es posible obtener imágenes internas del organismo.
- Medicina nuclear: Es una especialidad médica que realiza diagnósticos y tratamientos mediante la utilización de trazadores o radiofármacos. Realiza estudios de órganos y sistemas desde el punto de vista funcional. La Medicina Nuclear se diferencia de las otras técnicas de imagen en que realiza estudios fisiopatológicos. Es decir da una visión de cómo funciona el organismo.
- Ecografía: Utiliza ondas sonoras de alta frecuencia para observar órganos y estructuras al interior del cuerpo.
- Resonancia magnética: Se considera como una técnica no invasiva, ya que no requiere la introducción de herramientas o elementos en el cuerpo ni tiene consecuencias para el paciente. La información que se obtiene a través de la resonancia magnética es convertida en imágenes en una computadora, permitiendo que el profesional observe, de este modo, el interior del organismo.
- Endoscopia: Es un procedimiento que permite que el médico vea el interior del cuerpo. Utiliza un instrumento llamado endoscopio o tubo visor. Los endoscopios tienen una cámara diminuta unida a un tubo largo y delgado.

2.4. Tomografía computarizada

El descubrimiento y el desarrollo de la TC (Tomografía Computarizada) revolucionó el diagnóstico por imagen en medicina. La TC es una técnica digital y matemática de diagnóstico por imagen, que origina cortes tomográficos en los que cada capa no está contaminada por estructuras borrosas procedentes de la anatomía adyacente. Lo más importante, la TC permite la diferenciación y cuantificación de los tejidos duros y blandos. De este modo, por primera vez en las técnicas de imagen en medicina, el radiólogo podía visualizar los tejidos duros y blandos sobre una imagen, sin llevar a cabo un procedimiento invasivo sobre el paciente, como puede ser la inyección de medios de contraste.

La TC produce imágenes axiales de la anatomía de un paciente. Dichas imágenes se obtienen de forma perpendicular al eje mayor del cuerpo como se muestra en la figura 2.2.

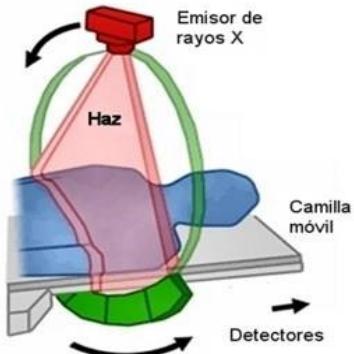


Figura 2.2: Cortes axiales en TAC.

Imagen obtenida de <https://www.fda.gov/radiation-emittingproducts/radiationemittingproductsandprocedures/medicalimaging/medicalx-rays/ucm115318.htm>

La TC es una técnica de diagnóstico para imagen digital. La fuente de rayo X se une rígidamente a un dispositivo detector de la geometría en abanico de los rayos, que rota 360 grados alrededor del paciente y recoge los datos. El detector de imagen presenta un estado gaseoso o sólido, lo que produce señales que sirven como datos de entrada para un ordenador exclusivo. Dicho ordenador procesa los datos mediante técnicas de retroproyección con algoritmo de Fourier, desarrolladas por primera vez por Hounsfield para dar lugar a imágenes de TC. Las imágenes de TC son en sí mismas tridimensionales, de 512 X 512 pixeles típicamente, con un espesor definido por la separación de cortes de la técnica de imagen. Cada elemento de la imagen de TC se denomina voxel, el cual presenta un valor, al cual se le refiere en unidades de Hounsfield, que describe la densidad de la imagen de TC en dicho punto. [20]

El resultado final de la reconstrucción por la computadora, es una matriz de números, que no es conveniente para su visualización en pantalla, por lo que un procesador se encarga de asignar a cada número o rango de números, un tono gris adecuado. Los valores numéricos de la imagen de tomografía computada, están relacionados con los coeficientes de atenuación, debido a que la disminución que sufre el haz de rayos X, al atravesar un objeto, depende de los coeficientes de atenuación

lineales locales del objeto. [21]

Algunas ventajas que ofrece esta técnica es el tiempo de exploración del orden de segundos en las modernas generaciones de aparatos, unido a su baja inocuidad (capacidad de hacer daño), lo que permite repetirlo cuantas veces sea necesario, así como su alta fiabilidad diagnóstica de certeza media, que alcanza el 85 %, superando con creces los resultados de todas las técnicas exploratorias clásicas juntas, aunque, justo es decirlo, no las desplaza en absoluto, sino que todas ellas se complementan mutuamente. [22]

Los posibles usos de este método diagnóstico, son los siguientes: anormalidades del cerebro y medula espinal, tumores cerebrales y accidentes cerebro vasculares, sinusitis, aneurisma de aorta, infecciones torácicas, enfermedades de órganos como el hígado, los riñones y los nódulos linfáticos del abdomen y muchos otros más. [21]

2.5. DICOM

DICOM (Digital Imaging and Communications in Medicine) es un estándar propuesto y administrado por la National Electrical Manufacturers Association (NEMA) en 1992. Especifica mecanismos de codificación, almacenamiento y transmisión de imágenes médicas; para llevar a cabo un análisis digital de imágenes médicas, generalmente se utilizan visores DICOM que implementan el estándar ya que con ellos es posible visualizar y exportar las imágenes a formatos de imagen digital comunes (JPG, PNG, BMP, etc.). [23] El formato genérico del archivo de DICOM consiste en dos partes diferenciadas:

1. Una cabecera (header) con multitud de campos estandarizados que especifican datos administrativos (datos del paciente, hospital donde se realizó, entre otros), datos sobre el estudio y la sintaxis de transferencia UID que especifica la codificación y la compresión del conjunto de datos que le sigue.
2. Un conjunto de datos (data set) de DICOM, que contiene la imagen o las imágenes especificadas que pueden estar comprimidas con distintos estándares, en la figura 2.3 se muestra la estructura básica de un archivo DICOM.

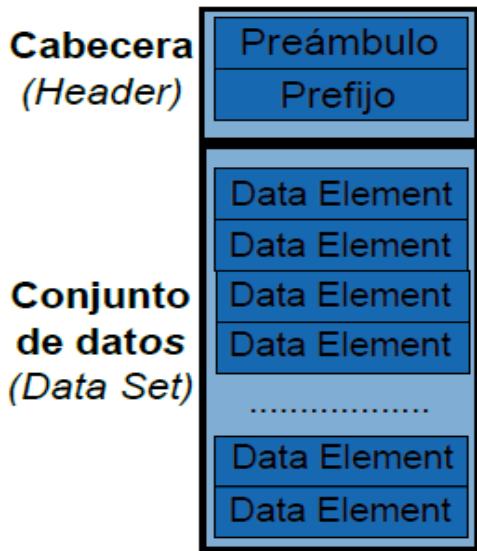


Figura 2.3: Estructura de un archivo DICOM.
Imagen obtenida de [15]

2.6. Escala de Hounsfield

El elemento individual de la imagen de TC es el vóxel, que tiene el valor, referido en unidades de Hounsfield, que describe la densidad de la imagen de TC en ese punto. Cada vóxel contiene 12 bits de datos y va desde -1000(aire) hasta las +1000 de unidades Hounsfield. Los escáneres de TC tienen un valor estandarizado de Hounsfield de 0 para el agua. La escala de densidad de los TC es cuantitativa y significativa en cuanto a la identificación y diferenciación de las estructuras y los tejidos. [24]

La escala de Hounsfield (HU) es una transformación lineal de la medida original del coeficiente de atenuación, basada en la radiodensidad del agua destilada, establecido en el STP (Estándar presión y temperatura) y se define como igual a 0HU, mientras que la radiodensidad del aire en STP se define como -1000HU; lo anterior proporciona al tejido óseo más denso (hueso compacto) valores cercanos a +1000HU. La figura 2.4 muestra los valores aproximados para algunos tejidos y órganos comúnmente estudiados.

La escala Hounsfield se extiende a lo largo de 2000 unidades que difícilmente

serían distinguibles si se le asignara a cada unidad un nivel de brillo distinto en un monitor de vídeo, esto debido a que el ojo humano no es capaz de distinguir más de 40 tonalidades de brillo diferentes, representar en una imagen toda la gama de valores de la escala de Hounsfield conlleva a no poder visualizar una gran cantidad de información. Por lo tanto, solo se representa mediante una escala de grises un sector parcial de los valores de la TC con el fin de solo visualizar el órgano o tejido estudiado y su detalle. [25]

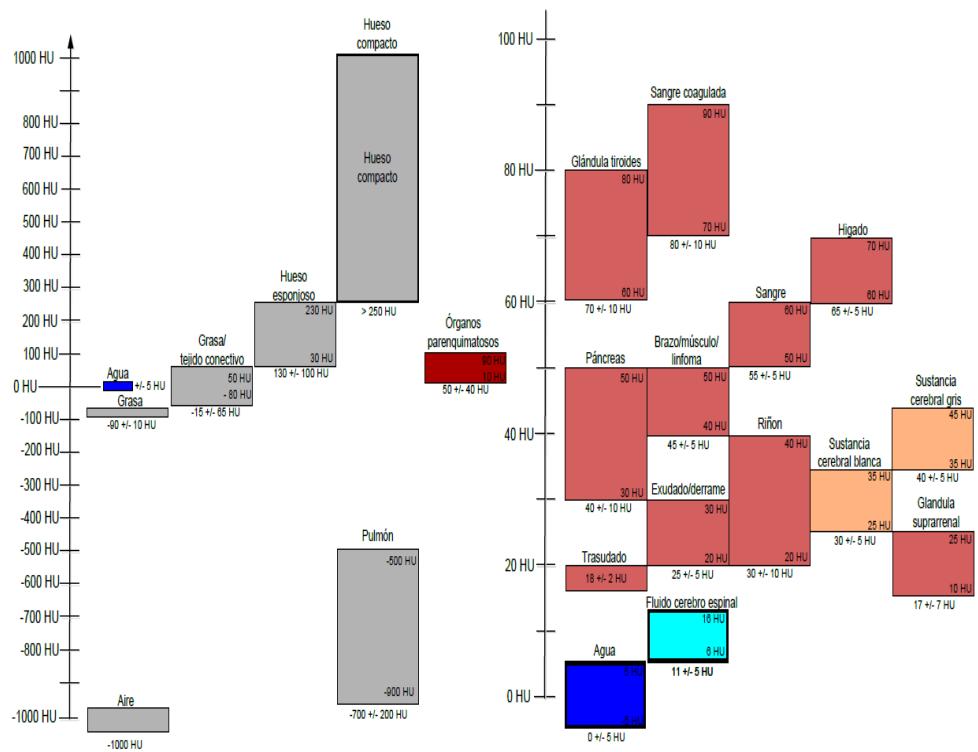


Figura 2.4: Escala de Hounsfield
Imagen obtenida de [25]

2.7. Segmentación

La segmentación de imágenes está definida como el particionado de una imagen en regiones constituyentes que no se solapan y las cuales son homogéneas con respecto a alguna característica como intensidad o textura. Si el dominio de la imagen está dado

por Ω , el problema de la segmentación es determinar cuales conjuntos $S_k \subset \Omega$, cuya unión, es el dominio Ω completo. Así, los conjuntos que componen una segmentación deben satisfacer

$$\Omega = \bigcup_{k=1}^K S_k \quad (2.1)$$

donde $S_k \cap S_j = \emptyset$ para $k \neq j$ y cada S_k está conectado. En un funcionamiento ideal, los métodos de segmentación encuentran aquellos conjuntos que corresponden a distintas estructuras anatómicas o regiones de interés en la imagen.

Si no tomamos en cuenta la restricción que indica que las regiones estén conectadas el proceso de determinar los conjuntos S_k es llamado "clasificación de pixeles", y los conjuntos se llaman "clases". La clasificación de pixeles es una meta deseable en las imágenes médicas, especialmente cuando regiones desconectadas que pertenezcan a la misma clase de tejido deben ser identificadas. Determinar el número total de clases K en la clasificación de pixeles puede ser un gran problema. A menudo, suponemos que el valor de K se conoce con base al conocimiento previo de la anatomía considerada. Por ejemplo, en la segmentación de una imagen cerebral obtenida a partir de una resonancia magnética, es común asumir que $K = 3$, correspondiente a las clases de los tejidos de materia gris, materia blanca y el fluido cerebroespinal.

El etiquetado es el proceso de asignar un significado importante a cada región o clase, este procedimiento se puede llevar a cabo como un proceso separado de la segmentación. En este proceso se mapea el índice numérico k del conjunto S_k a una denominación anatómica. En la imagenología médica, las etiquetas por lo general son visualmente detectables con obviedad y pueden ser determinadas en una revisión por un físico o un técnico. El etiquetado automatizado por computadora es deseable cuando las etiquetas no son obvias y en sistemas con procesos automatizados. Una típica situación que involucra etiquetado sucede en las mastografías digitales, en la cual la imagen es segmentada en diferentes regiones, las cuales después son clasificadas como sanas o tejido con tumor. [12]

2.7.1. Umbralización

Las técnicas de umbralización segmentan imágenes escalares al generar una partición binaria a través de las intensidades de la imagen. La figura 2.5a muestra el histograma de una imagen escalar que aparentemente posee 3 clases, correspondientes a las 3 modas. El procedimiento de umbralización trata de determinar un valor de intensidad, llamado "umbral", el cual separa las clases deseadas. Logramos la seg-

mentación cuando agrupamos los pixeles con intensidades mayores a las del umbral en una clase y todos los otros en otra clase. Dos umbrales potenciales se observan en la figura 2.5a en el valle del histograma. Determinar más de un valor umbral es un proceso llamado multiumbralización.

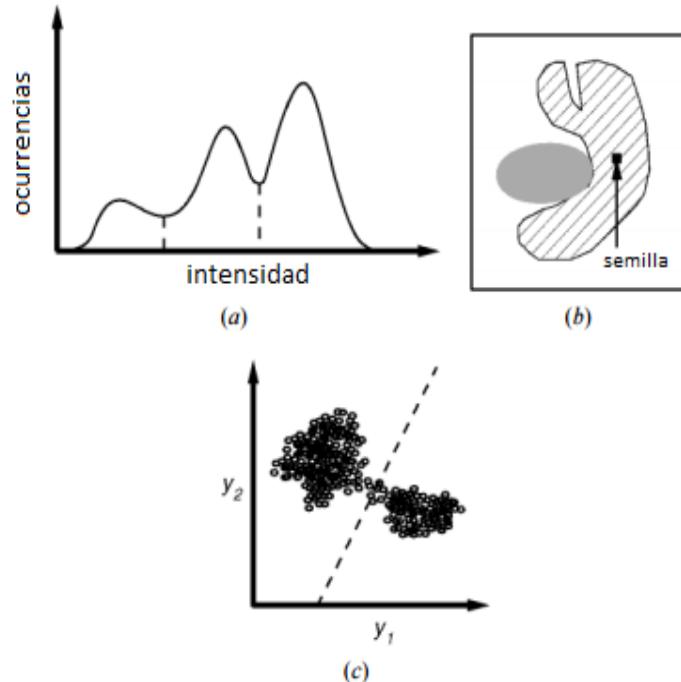


Figura 2.5: Métodos de características espaciales y región creciente. (a)Histograma con 3 aparentes clases. (b)Característica espacial en 2-D. (c)Ejemplo de región creciente.

Imagen obtenida de [12]

La umbralización es un procedimiento simple pero efectivo para segmentar imágenes en las cuales diferentes estructuras tienen intensidades contrastantes u otras características cuantificables. A menudo la partición de la imagen se genera en interacción con el usuario aunque también existen métodos automatizados.

La umbralización a menudo es utilizada como la primera de las operaciones en una secuencia de procesamiento de imágenes. Se ha aplicado en mastografías digita-

les, en las cuales usualmente hay presentes dos clases de tejidos –sanos y con tumores. Su principal limitación, es que en su forma más simple, sólo se generan dos clases y no se puede aplicar a imágenes multicanales. Además, la umbralización comúnmente no toma en cuenta las características espaciales de la imagen. Esto causa que sea sensible a las inhomogeneidades de ruido e intensidad, lo cual puede pasar en la resonancia magnética. Ambos sucesos corrompen de manera significante el histograma de la imagen haciendo más difícil la separación. Por estas razones, se han propuesto variaciones a la umbralización clásica aplicada en imágenes médicas basadas en intensidades y conectividad. [12]

2.7.2. Región creciente

El método de región creciente es una técnica para extraer una región de la imagen que está conectada basada en un criterio predefinido. Este criterio puede estar basado en la intensidad y/o bordes de la imagen. En su forma más simple, la región creciente requiere una semilla que es seleccionada manualmente por el operador y extrae todos los pixeles conectados a la semilla inicial basado en un criterio predefinido.

Por ejemplo, un posible criterio puede ser hacer crecer una región hasta que se tope con el borde de la imagen.

Esto es representado en la figura 2.5b, en el cual una región creciente ha sido usada para aislar una de las estructuras.

Como la umbralización, la región creciente raramente es usada por si sola, generalmente es usada dentro de un conjunto de operaciones de procesamiento de imagen, particularmente para la delineación de pequeñas estructuras como tumores o lesiones. La principal desventaja de la región creciente es que requiere una interacción manual para obtener la semilla. Aparte, para cada región que necesita extraerse, una semilla debe plantarse. El algoritmo Split-Merge (dividir y juntar) está relacionado con región creciente.

La región creciente puede ser sensible al ruido, causando así que las regiones extraídas puedan tener hoyos o incluso puedan ir desconectadas. De forma inversa, los efectos de volumen parcial pueden lograr que regiones separadas se conecten entre sí. Para ayudar a resolver estos problemas, se propuso un algoritmo homotópico de región creciente, para preservar la topología entre una región inicial y una región extraída.

Split and Merge. Las técnicas de Split and Merge intentan solucionar los problemas de ruido y los falsos bordes usando una medida controlada de homogeneidad. El objetivo es segmentar automáticamente la imagen en un mínimo número de regiones que intentan representar áreas de uniformidad, produciendo bordes con características relacionadas con la resolución de la imagen. La alternativa aproximada de región creciente inicia con un valor de semilla e intenta encontrar la extensión de una región local que obedece al criterio de homogeneidad dado por la semilla.

Normalmente el algoritmo comienza con la hipótesis de que la imagen completa es una única región, entonces analiza la homogeneidad de la misma (mediante un cierto criterio y propiedades). Si existe homogeneidad, la imagen se encuentra ya segmentada, si no es así, entonces la región es dividida en 4 regiones. Este proceso se repite para cada una de las regiones generadas hasta que el proceso de división no puede llevarse a cabo.

Una vez que se ha llevado a cabo el proceso de división, se comprueba para cada región generada, si es posible unirla con una región adyacente (lógicamente si satisfacen el criterio de homogeneidad establecido). El proceso termina cuando no se pueden fusionar más regiones. [26]

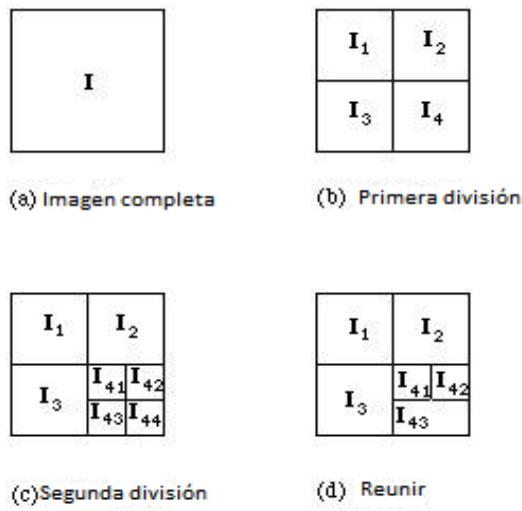


Figura 2.6: Ejemplo iteración algoritmo Split and Merge.
Imagen obtenida de https://users.cs.cf.ac.uk/Dave.Marshall/Vision_lecture/node34.html

2.7.3. Clasificadores

Los métodos clasificadores son técnicas de reconocimiento de patrones que buscan una partición a una característica espacial dada por una imagen usando datos con etiquetas conocidas. Una característica espacial es un rango de espacio de cualquier función de una imagen, siendo la característica espacial más común las intensidades de la imagen. Un histograma, como se muestra en la figura 3a, es un ejemplo de una característica espacial de una dimensión. La figura 3c muestra un ejemplo de una característica espacial en 2-d con aparentemente dos clases. Así como una figura espacial pudo haberse sido generada de un eco dual de una resonancia magnética, en la cual un eje representa las intensidades de un protón de una imagen densamente ponderada y otro eje representa las intensidades de la ponderación T2 de la imagen. Todos los pixeles con sus características asociadas son agrupadas en una clase en el lado izquierdo de la partición. Los clasificadores son conocidos como métodos supervisados ya que requieren de datos entrenados que son segmentados manualmente y se usan como referencia para segmentar automáticamente nuevos datos. Hay muchas maneras en la cual los datos entrenados puedan ser aplicados a métodos clasificadores. Un clasificador simple es el clasificador de vecino más cercano, en el cual cada pixel es clasificado en la misma clase donde estan los datos entrenados con la intensidad más cercana. El clasificador del k-ésimo vecino mas cercano es considerado un clasificador no paramétrico porque no hace suposiciones subyacentes sobre la estadística de la estructura de los datos. Otro clasificador no paramétrico es la ventana de Parzen, en la cual la clasificación está hecha por un proceso de decisión ponderada dentro de una ventana predefinida de característica espacial, centrado en la intensidad de pixeles no etiquetados.

Un clasificador usado comunmente es el de máxima similitud o clasificador Bayes. Este asume que la intensidad de los pixeles son muestras independientes de una mezcla de distribución de probabilidad, usualmente Gaussiana. Este mezcla, es llamada modelo de mezcla finita y esta dada por la siguiente función de probabilidad:

$$f(y_j; \theta, \pi) = \sum_{k=1}^K \pi_k f_k(y_j; \theta_k) \quad (2.2)$$

donde y_j es la intensidad del pixel j , f_k es el componente de probabilidad de la función de densidad parametrizada por θ_k , y $\theta = [\theta_1, \dots, \theta_k]$. Las variables π_k son coeficientes mezclados donde miden la contribución para cada densidad de la función y $\pi = [\pi_1, \dots, \pi_k]$. Los datos entrenados son recolectados a traves de obtener muestras representativas de cada componente del modelo mezclado y el estimado para cada theta.

Para mezclas Gaussianas, esto significa un estimado de K-medias, covarianzas y coeficientes mezclados. La clasificación de los nuevos datos es obtenida asignando cada pixel a la clase con la más alta probabilidad posterior. Cuando los datos en verdad siguen una mezcla finita Gaussiana, el clasificador de máxima similitud puede realizarse de forma correcta y es capaz de proveer una segmentación suave compuesta por probabilidades posteriores.

Los clasificadores estandar requieren que las estructuras que van a ser segmentadas posean características cuantificables distintas. Como los datos entrenados pueden etiquetarse, los clasificadores pueden transferir estas estiquetas a nuevos datos siempre y cuando su característica espacial se distinga lo suficiente en cada etiqueta. Siendo notorio, los clasificadores son relativamente computacionalmente eficientes, no como los métodos de umbralización que pueden ser aplicados para imágenes multicanal. Una desventaja de los clasificadores es que por lo general no realizan ningún modelado espacial. Otra desventaja es que requieren una interacción manual para obtener los datos de entrenamiento. Se pueden adquirir conjuntos de entrenamiento para cada imagen que requiere ser segmentada, pero esto puede consumir mucho tiempo y puede ser laboriosos. Por el otro lado, usar un conjunto datos de entrenamiento para un gran número de escaneo de numeros puede llevar a resultados parciales que no toman en cuenta la variedad anatómica y fisiológica entre diferentes sujetos. [12]

2.7.4. Clustering

Los algoritmos de clustering escenciamente realizan la misma función de los métodos clasificadores pero sin usar datos entrenados. Aparte, son metodos no supervisados. Para compensar la falta de datos entrenados, los métodos de clustering iteran alternadamente entre imagenes segmentada y caracterízan las propiedades de cada clase. En cierto sentido, los métodos de clustering se entrena a si mismos usando los datos disponibles. Los tres algoritmos de clustering mas comunes son el K-means o ISODATA algorithm, el fuzzy c-means algorithm y el expectation-maximization algorithm. [26]

Algoritmos de clustering

- A. *Algoritmo K-medias.* En el clustering de K-medias, particiona una colección de datos en k números de grupos de datos. Clasifica un conjunto de datos ya dados en k números de cluster desarticulados. El algoritmo de K-medias es separarado dos fases. En la primer fase se calcula el centro k y en la segunda fase se traslada cada punto al cluster en el cual tiene el centro mas cercano desde su respectivo

punto de dato. Hay diferentes métodos que definen la distancia entre el centro más cercano y uno de los métodos mas utilizados es la distancia Euclidiana. Una vez que el agrupamiento ya se realizó se recalcula un nuevo centro para cada cluster y basado en ese centro, una nueva distancia Euclidiana es calculada entre cada centro y cada punto y asigna los puntos al cluster que tiene la mínima distancia Euclidiana. Cada cluster en la partición es definido por sus objetos miembros y por su centro. El centro de cada cluster es el punto en el cual la suma de distancias de todos los objetos en el cluster es minimizada. Por lo tanto K-medias es un algoritmo iterativo en el cual se minimiza la suma de las distancias de cada objeto hacia el centro del cluster. [27]

- B. *Algoritmo Fuzzy C-Media.* Itera entre las probabilidades posteriores y calcula la máxima probabilidad estimada de la media, covarianzas y coeficientes mezclados de un modelo mixto.
- C. *Algoritmo EM.* EM es uno de los algoritmos más comúnmente usados para la estimación de densidad en puntos de datos en una configuración sin supervisar. El algoritmo depende en encontrar estimaciones de máxima verosimilitud de los parámetros cuando un modelo de dato depende de ciertas variables latentes. En el algoritmo EM, se realizan alternaciones en los pasos de expectación y maximización iterativamente hasta que el resultado converga. El paso de E calcula una expectación de una verosimilitud incluyendo las variables latentes como si fueran observadas, y el paso de maximización calcula la máxima verosimilitud estimada por los parámetros maximizando la verosimilitud esperada encontrada en el paso E. Los parametros encontrados en el paso de maximizacion son usados despues por otro paso E y el proceso se repite hasta que converja. [28]

Capítulo 3

Análisis

3.1. Comparación de software

En esta sección se analizan algunas características que se considerarán para el desarrollo del sistema, estas características son el sistema operativo y los lenguajes de programación, esto con el fin de elegir las herramientas que puedan ser más útiles para la construcción del sistema.

3.1.1. Sistemas operativos

La elección del sistema operativo es primordial para el desarrollo de un sistema ya que con base a este se pueden encontrar librerías específicas para el desarrollo del mismo. En la tabla 3.1 se hace una comparación entre dos sistemas operativos detallando algunas características que se considerarán para el desarrollo del sistema.

Sistema operativo	Ventajas	Desventajas
Linux	Es un sistema operativo de software libre. Cuenta con una gran estabilidad. Seguridad porque es un sistema operacional diseñado con la idea de Cliente-Servidor con permisos de acceso y ejecución a cada usuario. Las vulnerabilidades son detectadas y corregidas más rápidamente que cualquier otro sistema operativo. [29]	A pesar de ser un sistema operativo de software libre hay muchas aplicaciones que no corren en Linux. Linux no cuenta con una empresa que lo respalde por lo que no cuenta con soporte concreto. La configuración no es tan trivial. Tiene una gran cantidad de distribuciones. [29]
Windows	Es el sistema operativo más conocido en el mundo por lo que cuenta con gran cantidad de aplicaciones. Interfaz gráfica amigable. Cuenta con muchos asistentes de configuración. La mayoría de los médicos están más familiarizados con Windows. [29]	Cuenta con más brecha de seguridad ya que muchos virus están hechos para este sistema operativo. Costo elevado. Usuario no tiene acceso al código. Baja estabilidad. [29]
Mac OS	Interfaz intuitiva. Su vulnerabilidad ante malware es muy baja. Gran rendimiento. [30]	El hardware es caro. No existe una gran cantidad de software para Mac. La mayoría de los accesorios deben ser de la misma compañía. [30]

Tabla 3.1: Tabla comparativa de sistemas operativos.

3.1.2. Lenguajes de programación

Para el desarrollo del sistema se deben de tomar en cuenta las características que te brindan algunos lenguajes de programación, ya que todos los lenguajes tienen sus ventajas así como sus desventajas. A continuación en la tabla 3.2 se listan tres lenguajes de programación junto con sus ventajas y desventajas.

Lenguaje de programación	Ventajas	Desventajas
Java	El compilador del lenguaje de Java restringe ciertas operaciones para corregir errores. Estandariza muchas estructuras y operaciones como listas, manejo de conexiones de interconexiones (network connections) y provee interfaces gráficas de usuario(graphical user interfaces). Es orientado a objetos. Manejo automático de la memoria. [31]	Al tener que ser ejecutado mediante una maquina virtual (JVM) hace que no sea tan rápido como otras tecnologías, un ejemplo C++. [31]
C++	Usa múltiples paradigmas de programación (POO, estructurado). Puede compilar y ejecutar código en C. Provee un desempeño y manejo de memoria eficiente. Alto nivel de abstracción. [32]	Uso de DLLs (librerías dinámicas) muy complejos. No tiene un recolector de basura. No es seguro porque tiene apuntadores, funciones amigas y variables globales. [32]
Python	Lenguaje de programación multiparadigma, permite varios estilos de programación: orientado a objetos, estructural y funcional. Gran soporte e integración con otros lenguajes y herramientas. Lenguaje estable, confiable y fácil de aprender. Te permite realizar aplicaciones con menos código que con otros lenguajes. Cuenta con una gran cantidad de Frameworks que se pueden utilizar. [33]	Ya que es un lenguaje interpretado es lento. No es la mejor opción para el manejo de memoria. [33]
C#	Declaraciones en el espacio de nombres. Tipos de datos: Existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java. Atributos: Cada miembro de una clase tiene un atributo de acceso del tipo público, protegido, interno, interno protegido ^[31] y privado. Pase de parámetros: Se puede declarar los métodos para que acepten un número variable de parámetros.	Se tiene que conseguir una versión reciente de Visual Studio .NET. Se tiene que tener algunos requerimientos mínimos del sistema para poder trabajar adecuadamente como contar con Windows NT 4 o superior. Se requiere mucho espacio de almacenamiento para su instalación. [34]

Tabla 3.2: Tabla comparativa de lenguajes de programación.

3.1.3. Elección de librerías

El uso de librerías facilitan el desarrollo del sistema ya que hay mucho código que se puede utilizar. En este sección se describen las librerías que se van a implementar.

Descripción general de PYDICOM

PYDICOM es un paquete puro de python para trabajar con archivos DICOM tales como imágenes médicas, reportes y objetos de radioterapia. PYDICOM facilita leer estos archivos complejos en estructuras naturales de lenguaje python para su fácil manipulación. Se modifica su dataset que puede ser escrito de nuevo en un formato de archivo DICOM. PYDICOM no se trata de solo ver las imágenes, esta diseñado para manipular elementos de datos en archivos DICOM con código python. Además es fácil de instalar y usar, y ya que es un paquete de python, debe correr en cualquier lugar donde corre un programa en python. Una limitación presente en esta librería es que los datos comprimidos de píxeles (por ejemplo JPEG) no se pueden alterar de una manera inteligente como es posible con los píxeles que no se encuentran comprimidos. PYDICOM tiene una licencia basada en la licencia de MIT. La librería pydicom es soportada por varios sistemas operativos tales como: Linux, Mac OS, Windows. [35]

Dataset es el objeto base en el modelo de objetos de pydicom. Es el objeto principal con el cual se trabajará directamente. Dataset es un derivado del *dict de python* (diccionario de python), así que este hereda los métodos de dict. En otras palabras es una colección de la paridad llave:valor, donde la llave es la etiqueta DICOM y el valor es una instancia del elemento de datos. [35]

PYDICOM no es una librería exigente, los únicos requisitos para utilizarla son:

- Tener una versión de compilador mayor a la 2.4.
- Tener la librería NumPy para la manipulación de los datos de los píxeles.

Descripción general de DCM4CHE

DCM4CHE es un conjunto de aplicaciones utilizadas mundialmente por profesionales de la salud, proyectos de investigación , aplicaciones de software libre y software comercial. DCM4CHE es una implementación de software libre de alto rendimiento del estandar DICOM. Está desarrollado en el lenguaje de programación Java. Esta herramienta es un sistema JEE(Java Enterprise Edition) y JMX(Java Management Extensions) que se despliega dentro del servidor de aplicaciones JBoss para proveer

diversos servicios clínicos. Se puede usar para diversos propósitos, los más populares son:

1. Manejo y almacenamiento de archivos DICOM.
2. PACS(Sistema de archivo y transmisión de imágenes) cuando se usa con un visor.

En un principio DCM4CHE fue diseñado con la intención de ser utilizado por Sun como un JSR(Java Specification Request) para cualquier aplicación de DICOM en Java. Con eso en mente, la herramienta fue separada en una aplicación de interfaz y otra de aplicación. [36]

Algunas de las características más importantes son:

- Provee una interfaz de usuario basada en la web.
- Permite un manejo de objetos DICOM a través de la conversión a estándares del sistema.
- Uso del servidor HL7.
- Acceso web a diversas fuentes de información.
- Exportación a CD.

3.2. Elección de herramientas para el desarrollo

Con base a lo expuesto anteriormente en esta sección se hace la selección de las herramientas a utilizar para el desarrollo del sistema.

3.2.1. Software

Lenguajes Python No es un secreto que Python es uno de los lenguajes de programación más populares, durante los últimos 5 años, Python se ha sostenido como el lenguaje de programación número uno. Python es uno de los favoritos entre los desarrolladores debido al énfasis que pone en su legibilidad y su eficiencia, especialmente cuando se compara con otros lenguajes como Java, PHP o C++. Python permite crear más funciones en una menor cantidad de líneas de código. Python es rápido de aprender para cualquiera.

La belleza de Python -además de su simplicidad- se basa en las reglas de programación sobre las cuales está creado el lenguaje. Algunos de estos principios son:

- La legibilidad es muy importante.
- Menos es más.
- Lo complejo está bien, pero no complicado.
- Lo claro es mejor que lo implícito.

Justificación.

Se escogió este lenguaje debido a las librerías con las que cuenta además del posible manejo de la información que permite, además la sencillez que este lenguaje presenta fue determinante para su elección.

C# Es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C. Visual Studio admite Visual C# con un editor de código , plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas.

Justificación Se eligió este lenguaje debido a las herramientas de diseño que proporciona gracias al framework .NET, la facilidad de accesibilidad a otros lenguajes y el ser un lenguaje nativo de Windows.

Librería: **PYDICOM** PYDICOM es una librería presente en Python la cual permite la obtención y manejo de archivos DICOM. Es de código abierto. PYDICOM permite la reescritura del dataset de los archivos DICOM. Tiene soporte en diferentes sistemas operativos, entre ellos Windows, Mac OS y distintas distribuciones Linux.

Justificación.

Es una herramienta que implementa muchas funciones para el manejo de imágenes médicas además de estar disponible en las versiones actuales de Python por lo cual se mantiene actualizada y presenta un continuo mantenimiento.

3.2.2. Metodología

La metodología de trabajo a utilizar en el desarrollo de este sistema será la metodología incremental. El modelo incremental está basado en la idea de desarrollar una implementación inicial, exponiéndose así a los comentarios del usuario y mejorándolo mediante varias versiones hasta que se ha desarrollado un sistema adecuado (Figura 3.1). Las especificaciones, el desarrollo y la validación de actividades están intercaladas en lugar de estar separada, con una rápida retroalimentación entre las actividades. [37]

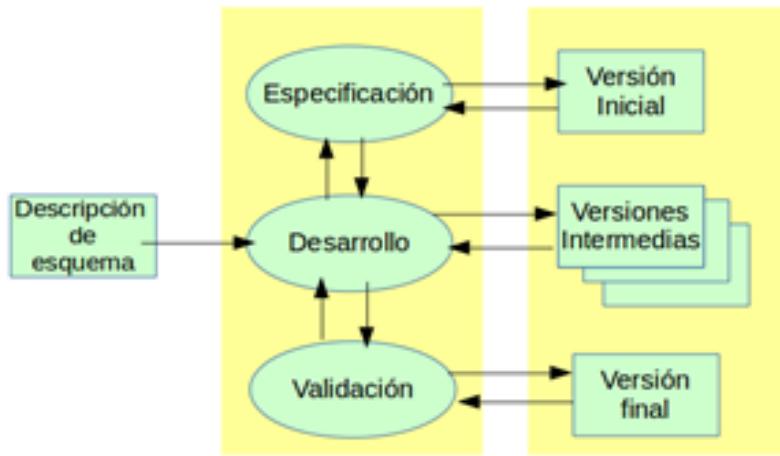


Figura 3.1: Flujo metodología incremental.
Imagen obtenida de [37]

La metodología incremental, que es una pieza fundamental para enfoques ágiles, es mejor que la metodología en cascada que está más enfocada al negocio, al e-commerce y a sistemas personales. La metodología incremental refleja la manera en la que resolvemos problemas. Raramente trabajamos la solución de un problema por adelantado, pero nos movemos a través de una serie de pasos para la solución, retroalimentándonos cuando nos damos cuenta de que un error fue cometido. Desarrollar un sistema de forma incremental es barato y con un fácil manejo de cambios mientras este está siendo desarrollado. [37]

Cada incremento o versión del sistema incorpora una funcionalidad que necesita el cliente. Generalmente, los primeros incrementos del sistema incluyen las funcionalidades más importantes o las más urgentemente requeridas. Esto significa que el cliente puede evaluar el sistema en una etapa relativamente inicial en el desarrollo para observar si se le va a hacer entregado lo que requirió. En caso de no ser así, entonces solo el incremento tiene que cambiarse y, posiblemente se definen nuevas funcionalidades para incrementos posteriores.

La metodología incremental tiene tres importantes beneficios:

1. El costo de acomodar los cambios de requerimientos del cliente es reducido.

2. Es más fácil obtener la retroalimentación del cliente en el desarrollo que se ha estado haciendo. Los clientes pueden comentar con base a las demostraciones del sistema y pueden ver cuanto se ha implementado. Los clientes encuentran difícil juzgar el progreso de los documentos del diseño.
3. Es posible una mayor entrega y muestra del sistema al cliente, a pesar de que no todas las funcionalidades estén incluidas.

De alguna manera la metodología incremental es el enfoque más común para el desarrollo de aplicaciones. Este enfoque puede ser tanto dirigido por el plan como ágil o usualmente como una mezcla de enfoques. En el enfoque dirigido por el plan, los incrementos del sistema son identificados por adelantado; si un enfoque ágil es adoptado, los primeros incrementos son identificados pero el desarrollo de incrementos posteriores depende del progreso y las prioridades del cliente. [37]

Desde un punto de vista administrativo, la metodología incremental tiene dos problemas.

1. El progreso no es visible. Los administradores necesitan entregas para medir el proceso. Si el sistema es desarrollado rápidamente, no es costosamente efectivo producir documentos que reflejan cada versión del sistema.
2. La estructura del sistema se tiende a degradar cuando se agregan nuevos incrementos. A menos que el tiempo y el dinero se gaste en mejorar el sistema, cambios regulares tienden a corromper la estructura del sistema. Incorporar nuevos cambios en el software se puede convertir difícil y costoso.

Se dividió el sistema en 3 módulos, los cuales se listan y describen a continuación.

Módulo 1. **Decodificación de DICOM.** Se decodifica un archivo DICOM para obtener la información necesaria de la imagen.

Módulo 2. **Tratamiento de imagen.** Con base a la información encontrada en el archivo DICOM se aplican diversas técnicas para detectar propiedades particulares en las tomografías. Dentro de este módulo se pueden encontrar 4 procesos principales.

- a) Umbralización.
- b) Segmentación de regiones.
- c) Clasificación de regiones.
- d) Análisis de regiones.

Módulo 3. **Visualización.** Se presenta de manera gráfica la información contenida en un archivo de tipo DICOM permitiendo visualizar las estructuras internas del cuerpo.

3.2.3. Hardware

Para el desarrollo del sistema se cuenta con 4 equipos de cómputo portátiles en los cuales se realizarán las implementaciones de código, pruebas y la documentación que se requiera. En la tabla 3.3 se hace una breve descripción de las características de hardware los equipos.

Nombre del equipo	Procesador	Memoria RAM	Sistema Operativo
Toshiba Satellite S40DT	AMD A8-5545 a 1.70 GHz	6 GB	Windows 10 Home Single Language, Elementary O.S. 0.4 Loki. 64 bits
Lenovo ideapad S1 5100	Intel Core i3-3227U a 1.90 GHz	4 GB	Windows 10 Home Single Language, Ubuntu 15.04 lts. 32 bits
Toshiba Satellite	Inter Core i3-M380 a 2.53GHz	3 GB	Windows 8.1 pro, Ubuntu 14.04 lts. 64 bits
Dell Inspiron 15R	Intel Core i7	6 GB	Windows 10 Home Single Language de 64 bits.

Tabla 3.3: Especificaciones de hardware

3.3. Análisis de factibilidad

Como en todo proyecto se deben considerar ciertos factores referentes a los recursos y algunas restricciones que podrían presentarse durante el desarrollo del sistema. Se hará un estudio de factibilidad desde 3 perspectivas distintas:

- **Factibilidad técnica:** Se basa principalmente en la evaluación de los equipos

tanto en software como en hardware para la realización del proyecto. Se hace también una evaluación de la infraestructura con que se cuenta para el proyecto.

- **Factibilidad operativa:** Permite saber si es viable poner el proyecto en operación.
- **Factibilidad económica:** Ayuda a determinar desde la perspectiva financiera si el proyecto se puede realizar, se justifican los costos y tiempos fijados con base a los beneficios que el trabajo aporta.

Estudio de factibilidad operacional

El sistema tiene una interfaz sencilla lo que permite a los usuarios tener un manejo óptimo de la herramienta para un mayor aprovechamiento de sus funciones. El sistema ofrece una visualización de estructuras internas del cuerpo con gran calidad en un tiempo eficiente.

Se ha diseñado el sistema de manera que el usuario haga uso de este mediante una computadora portátil o de escritorio, lo cual evita el adquirir equipo de alto costo y complejidad para su manejo.

Con base a los usuarios potenciales de este sistema, se toma como hecho que quien haga uso de este sistema tiene conocimientos sobre las TAC, anatomía del cuerpo humano y manejo básico de computadoras.

Estudio de factibilidad técnica

Con base a las herramientas seleccionadas para la realización del sistema, en la tabla siguiente se listan y describen los sistemas operativos sobre los cuales se trabajará el desarrollo del sistema.

Sistema	Descripción
Windows 8.1 Pro	Cuenta con una interfaz que permite hacer cualquier cambio sobre la PC, incluye funciones como Control de escritorio remoto, encriptado de archivos y dispositivos, Hyper-V para la creación y uso de máquinas virtuales, etc. [38]
Windows 10 Single Language	Sistema simple que recupera la sencillez de su interfaz, mantiene las mejores características de Windows 8 además de integrar nuevas herramientas como la visualización de múltiples escritorios y la inclusión de la bash para desarrolladores. [39]

Tabla 3.4: Sistemas operativos a utilizar.

Tomando en cuenta los sistemas a utilizar y las herramientas se eligió utilizar el IDE Visual Studio para el desarrollo del sistema debido a la integración del framework .NET y el manejador de versiones.

IDE	Descripción
Visual Studio	Desarrolla aplicaciones para Android, iOS, Mac, Windows, la web y la nube. Escribe código con rapidez. Depurar y emitir diagnósticos fácilmente, realiza pruebas periódicas y publicar rápidamente, extender y personalizar según sus preferencias, colaboración de manera eficiente. [40]

Tabla 3.5: IDE de desarrollo.

Finalmente para el desarrollo de la documentación del sistema así como del manual del usuario utilizaremos Texmaker, GIMP y StarUML.

Herramienta	Descripción
Texmaker	Es un moderno editor de textos en Latex multiplataforma la cual conjunta una gran cantidad de funcionalidades para crear documentos en Latex en una sola aplicación. [41]
GIMP	Programa de distribución libre para el retoque de fotografías, edición de la composición y creación de imágenes. [42]
StarUML	Un sofisiticado modelador de software basado en las reglas de UML. [43]

Tabla 3.6: Herramientas de apoyo para documentación.

Estudio de factibilidad económica

Para conocer el costo del sistema de deben contemplar diversos factores como el costo de los equipos para trabajar, las licencias usadas, sueldos y servicios que se requerirán durante el tiempo que se desarrolle el sistema. Es importante mencionar que ya se cuentas con los equipos de computo y algunas de las herramientas de software que se utilizarán, sin embargo se incluirán es este plan para hacer una estimación más cercana al precio.

Equipo	Precio
Toshiba Satellite S40DT	MXN \$14,000.00
Lenovo ideapad S1 5100	MXN \$8,000.00
Toshiba Satellite	MXN \$11,000.00
Dell Inspiron 15R	MXN \$15,000.00
Total	MXN \$48,000.00

Tabla 3.7: Costos equipos de cómputo.

La siguiente tabla lista los costos de los sistemas operativos que se utilizarán.

Sistema operativo	Precio
Windows 10 Single Language	MXN \$2,499.00
Windows 8.1 pro	MXN \$790.00
Total	MXN \$8,287.00

Tabla 3.8: Costos sistemas operativos.

En la siguiente tabla se muestra el precio del IDE a utilizar.

IDE	Precio
Visual Studio	Software de distribución libre en su versión Community. [40]
Total	MXN \$0.00

Tabla 3.9: Costos IDE.

Se listan los precios de los softwares de apoyo junto con sus precios.

Software	Precio
Texmaker	Software de distribución libre. [41]
StarUML	MXN Software de distribución libre. [43]
GIMP	Sistema de distribución libre. [42]
Total	MXN \$0.00

Tabla 3.10: Costos software de apoyo.

En el desarrollo de un sistema es importante tener en cuenta los pagos de los servicios así como la renta del espacio donde se trabajará. Se hará un estimado de los gastos para un periodo de 10 meses que es el tiempo en que se tiene planeado terminar el sistema. Los costos se han obtenido con base a un promedio de los gastos de una casa con 4 habitantes.

Servicio/Producto	Costo mensual	Costo subtotal
Energía eléctrica	MXN \$350.00	MXN \$3,500.00
Agua	MXN \$300.00	MXN \$3,000.00
Servicio de telefonía e internet	MXN \$420.00	MXN \$4,200.00
Papelería	MXN \$150.00	MXN \$1,500.00
Renta	MXN \$6,500.00	MXN \$65,000.00
Total	MXN \$7720.00	MXN \$77,200.00

Tabla 3.11: Costos servicios.

Ahora se muestran los gastos referentes a salarios para los desarrolladores y líder de proyecto. Con base en datos recolectados del Observatorio Laboral y la CONASAMI se establecieron los siguientes salarios para el desarrollo del proyecto.

Empleado	Numero de empleados	Salario mensual	Total de salarios Mensuales	Salarios por día
Lider de Proyecto	1	MXN \$15,583.00	MXN \$15,583.00	MXN \$519.44
Programador Junior	2	MXN \$10,080.00	MXN \$20,160.00	MXN \$672.00
Tester	1	MXN \$8,532.00	MXN \$8,532.00	MXN \$284.40
Limpieza	1	MXN \$2,000.00	MXN \$2,000.00	MXN \$67.00
Total			MXN \$46,275.00	MXN \$1,542.84

Tabla 3.12: Estimación de sueldos

Ahora para calcular el costo del proyecto en cuanto a salarios se refiere haremos un cálculo PERT.

El método denominado PERT "Program Evaluation and Review Technique" puede ser catalogado como un método cuantitativo de planificación. El resultado final de la aplicación de este algoritmo será un cronograma para el proyecto, en el cual se podrá conocer la duración total del mismo, y la clasificación de las actividades según su criticidad.

Nació a finales de 1957, como resultado de un encargo de la oficina de proyectos especiales de la armada estadounidense a la división de sistemas de lockheed y a la empresa de consultoría Booz Allen & Hamilton.

El PERT actúa como una herramienta para definir y coordinar lo que hay que hacer para llevar a cabo, con éxito y a tiempo, de los objetivos de un proyecto. Su campo de aplicación es tan amplio como el número de actividades susceptibles de planificación. El PERT es un instrumento que ayuda a tomar decisiones, pero no las toma; sólo aporta información para tomarlas. Es por ello muy interesante conocer esta técnica y de ser capaz de utilizar su información, y con este fin hemos redactado este documento. [44]

Se consideran jornadas laborales de 8 horas, cuando se considere un tiempo de urgencia se aumentará el sueldo. Se muestra a continuación la tabla del cálculo PERT.

Actividad	Marcador Actual	Marcador Anterior	Tiempo Normal	Tiempo Urgencia	Costo normal MXN	Costo urgencia MXN
Desarrollo Introducción	A		7	5	\$10,799.88	\$15,428.40
Desarrollo Marco Teórico	B	A	9	6	\$13,885.56	\$18,514.08
Diseño (Módulo decodificación imagen DICOM)	C	B	15	12	\$23,142.60	\$37,028.16
Implementación (Módulo decodificación imagen DICOM)	D	C	7	5	\$10,799.88	\$15,428.40
Realización de pruebas (Módulo decodificación imagen DICOM)	E	C,D	5	3	\$7,714.20	\$9,257.04
Diseño (Módulo tratamiento de imagen)	F	E	42	36	\$64,799.28	\$111,084.48
Implementación (Módulo tratamiento de imagen)	G	F	28	22	\$43,199.52	\$67,884.96
Realización de pruebas (Módulo tratamiento de imagen)	H	F,G	18	10	\$27,771.12	\$30,856.80
Diseño (Módulo visualización)	I	H	25	21	\$38,571.00	\$64,799.28
Implementación (Módulo visualización)	J	I	18	15	\$27,771.12	\$46,285.20
Realización de pruebas(Módulo visualización)	K	I,J	13	8	\$20,056.92	\$24,685.44
Costo Total					\$288,511.08	

Tabla 3.13: Análisis PERT del sistema

Para obtener el valor completo del sistema se suman los costos obtenidos.

Concepto	Costo
Hardware	MXN \$48,000.00
Sistemas operativos	MXN \$8,287.00
Software	MXN \$0.00
Servicios	MXN \$77,200.00
Análisis PERT	MXN \$288,511.08
Total	MXN \$421,998.08

Tabla 3.14: Costos sistema

Obteniendo una utilidad del 20 % se fijaría un costo de MXN \$506,397.70.

3.4. Plan de manejo de riesgos

En esta sección hablaremos sobre los riesgos y dificultades que se pueden sufrir en el desarrollo del sistema, hay que considerar cada riesgo identificado y realizar un juicio acerca de la probabilidad y gravedad de dicho riesgo y encontrar las medidas que se necesitan para poder lograr los objetivos.

3.4.1. Identificación de riesgos

Los riesgos identificados en la realización del sistema del trabajo terminal son:

- No cumplir con los objetivos principales.
- Un integrante del equipo esté ausente por enfermedad en un momento crítico.
- Las herramientas usadas (software) sean obsoletas.
- Las herramientas de software no puedan trabajar en una forma íntegra.
- Fallos en equipos de cómputo.
- Los algoritmos no sean los adecuados conforme a lo requerido.

- El trabajo no esté completo para la fecha de entrega.
- No tener suficientes archivos.
- Subestimar la complejidad del sistema.

3.4.2. Análisis del riesgo

No es posible hacer valoraciones precisas y numéricas de la probabilidad y gravedad de cada riesgo. Habrá que asignar una probabilidad del riesgo con base al criterio del equipo, a continuación se darán valores de probabilidad según la gravedad.

- Muy baja (menor del 10 %).
- Baja (del 10 al 25 %).
- Moderada (del 25 al 50 %).
- Alta (del 50 al 75 %).
- Muy alta (mayor del 75 %).

En la tabla 3.4 se muestra la probabilidad y el impacto que puede tener el trabajo terminal.

Riesgo	Probabilidad	Impacto
No cumplir con los objetivos principales.	Muy baja.	Catastrófico.
Un integrante del equipo esté ausente por enfermedad en un momento crítico.	Alta.	Tolerable.
Las herramientas usadas (software) sean obsoletas.	Baja.	Serio.
Las herramientas de software nos puedan trabajar en una forma íntegra.	Moderada.	Serio.
Fallos en equipos de cómputo.	Alta.	Catastrófico.
Los algoritmos no sean los adecuados conforme a lo requerido.	Baja.	Serio.
El trabajo no esté completo para la fecha de entrega.	Baja.	Catastrófico.
No tener suficientes archivos.	Moderada.	Serio.
Subestimar la complejidad del sistema.	Baja.	Serio.

Tabla 3.15: Análisis de riesgo

3.4.3. Plan de contención de riesgos

El proceso de planeación del riesgo considera cada uno de los riesgos clave identificados y desarrolla estrategias para manejarlos. Para cada uno de los riesgos, se deberá considerar las acciones que puede tomar para minimizar la perturbación del trabajo terminal.

En la tabla 3.13 se muestran los riesgos y la prevención que se deben de considerar.

Riesgo	Prevención
No cumplir con los objetivos principales.	Tener una buena comunicación con los directores y mostrar avances del trabajo terminal continuamente.
Un integrante del equipo esté ausente por enfermedad en un momento crítico.	No se puede prevenir una enfermedad.
Las herramientas usadas (software) sean obsoletas.	Investigar si la herramienta tiene mantenimiento continuamente.
Las herramientas de software nos puedan trabajar en una forma íntegra.	Cuando se quiera usar una herramienta tener en cuenta si se puede integrar con las herramientas ya usadas o con herramientas que se pueden usar en el futuro.
Fallos en equipos de cómputo.	Tener un mantenimiento continuo de las computadoras.
Los algoritmos no sean los adecuados conforme a lo requerido.	Hacer una investigación a fondo de los algoritmos que se deben de usar e igual tener una variedad de algoritmos por si uno falla se tenga otros en la reserva.
El trabajo no esté completo para la fecha de entrega.	Cada integrante del equipo debe de hacer lo que le corresponda para poder llegar al objetivo.
No tener suficientes archivos.	Tener una cantidad adecuada de archivos.
Subestimar la complejidad del sistema.	Tener un avance continuo del proyecto y una constante revaluación del avance general del trabajo terminal.

Tabla 3.16: Plan de contención de riesgos

3.5. Definición de requerimientos del sistema

El objetivo principal es desarrollar una herramienta capaz de tomar un archivo DICOM almacenado en un disco donde vienen los estudios de un paciente y permita realizar un análisis y clasificación detallada de tejidos con base en los resultados obtenidos a partir del archivo DICOM.

La herramienta debe ser manipulada por el usuario (especialista) de manera muy básica, es decir, puede umbralizar, segmentar, obtener un punto creciente, clasificar (clustering) sobre un imagen DICOM.

3.5.1. Requerimientos funcionales

Los requerimientos funcionales para un sistema explican lo que el sistema debe hacer. Los requerimientos dependen del tipo de software en desarrollo, de los usuarios y del enfoque general cuando se escriben los requerimientos. Al ser los requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. [45]

En la siguiente tabla 4.1 se enlistan los requerimientos funcionales del sistema.

Identificador	Nombre	Descripción
RF1	Seleccionar un directorio.	El usuario deberá seleccionar un directorio donde este almacenado el archivo DICOM.
RF2	Visualizar imagen DICOM en 2D.	A partir del conjunto de imágenes DICOM se podrá ver la imagen sin ningún tratamiento.
RF3	Segmentar.	Para poder segmentar una parte de la imagen se hace un tratamiento adecuado para poder hacerlo.
RF4	Umbralizar.	Se umbraliza una imagen DICOM, el umbral se obtiene con base a la escala de Hounsfield para poder clasificar los tejidos.
RF5	Obtener un punto creciente.	El usuario dará una semilla (punto de origen) para buscar en su vecindad 8(celdas adyacentes a su posición), se podrá mover si hay una diferencia menor o igual a Épsilon.
RF6	Clustering.	Es otra técnica para poder clasificar los diferentes tejidos que contiene la imagen DICOM.
RF7	Visualizar imagen DICOM con tratamiento.	A partir de una imagen DICOM y con un tratamiento adecuado se visualizaran los tejidos.

Tabla 3.17: Requisitos funcionales del sistema

3.5.2. Requerimientos no funcionales

Los requerimientos no funcionales, como su nombre lo dice, son requerimientos que no se vincula directamente con las funciones específicos que el sistema proporciona. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y capacidad de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usados en las interfaces con otros sistemas. [45]

En la siguiente tabla 4.2 se enlistan los requerimientos no funcionales del sistema.

Identificador	Nombre	Descripción
RNF1	Desarrollado en Python y C#	Se desarrolló en estos lenguajes por dos razones la decodificación del archivo DICOM es muy sencilla en Python y por la escalabilidad que tiene C# con otros lenguajes aunado a las herramientas de diseño proporcionadas.
RNF2	Compatibilidad de lenguaje.	El compilado puede ser Python 2 o Python 3.
RNF3	Intuitivo.	El sistema debe ser sencillo para el usuario que sea fácil su manejo.
RNF4	Eficiente.	El tratamiento o el análisis deben ser eficiente.
RNF5	Tipo de archivo.	El sistema únicamente soportara archivos DICOM.
RNF6	Rapidez.	El usuario tendrá un tiempo de respuesta muy bajo cuando se le aplique una técnica a la imagen.
RNF7	Actualizaciones.	El sistema debe de ser escalable para nuevas mejoras.

Tabla 3.18: Requisitos no funcionales del sistema

Capítulo 4

Diseño del sistema

4.1. Arquitectura del sistema

En esta sección describimos la entrada, núcleo y salida del sistema en un buen funcionamiento.

Entrada del sistema.

Para comenzar la operación del sistema se parte de una o varias imágenes DICOM con las cuales se obtiene la información necesaria para generar una visualización de un estudio tomográfico.

Núcleo del sistema.

Aquí se llevan a cabo todos los algoritmos sobre las regiones seleccionadas por el usuario para un mejor análisis y obtención de propiedades.

Salida del sistema.

Finalmente el sistema ofrece una imagen en 2D ya analizada de la región seleccionada por el usuario, lo cual permite una mejor visualización de las estructuras.

4.2. Casos de uso

4.2.1. Caso de uso general

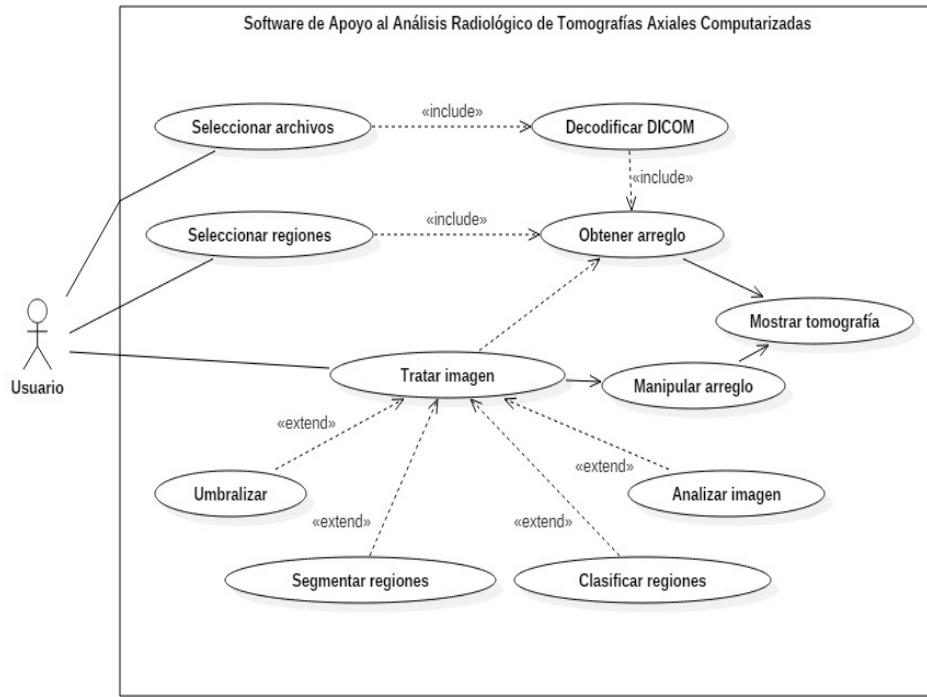


Figura 4.1: Diagrama de casos de uso general.
Imagen de elaboración propia.

En la figura 5.1 se observa el caso de uso general del sistema, el sistema tiene 3 interacciones principales con el usuario: selección de archivos, selección de región y tratamiento de imagen.

En la primera interacción el usuario a través de una interfaz selecciona uno o varios archivos en formato DICOM, una vez cargado el archivo el sistema se encarga de decodificar la información contenida para así poder generar el arreglo de valores de Hounsfield.

La segunda interacción principal del usuario parte de la primera, una vez que el usuario puede observar la tomografía tiene la opción de elegir una región específica, una vez acotada la región el sistema determina los valores del arreglo de Hounsfield

correspondientes a la nueva región y genera una nueva matriz, posteriormente se despliega la imagen de la nueva región.

La tercera y última interacción que tiene el usuario con el sistema es el tratamiento de imágenes, para esto es necesario haber obtenido el arreglo de una región acotada, el usuario podrá realizar distintos tipos de manipulación de la imagen como umbralización, segmentación de las regiones, análisis de las regiones, incluso el análisis de la propia imagen. Una vez realizadas las operaciones necesarias el sistema se encargará de manipular el arreglo para que se puedan observar los cambios producidos sobre el estudio tomográfico.

4.2.2. Caso de uso decodificación

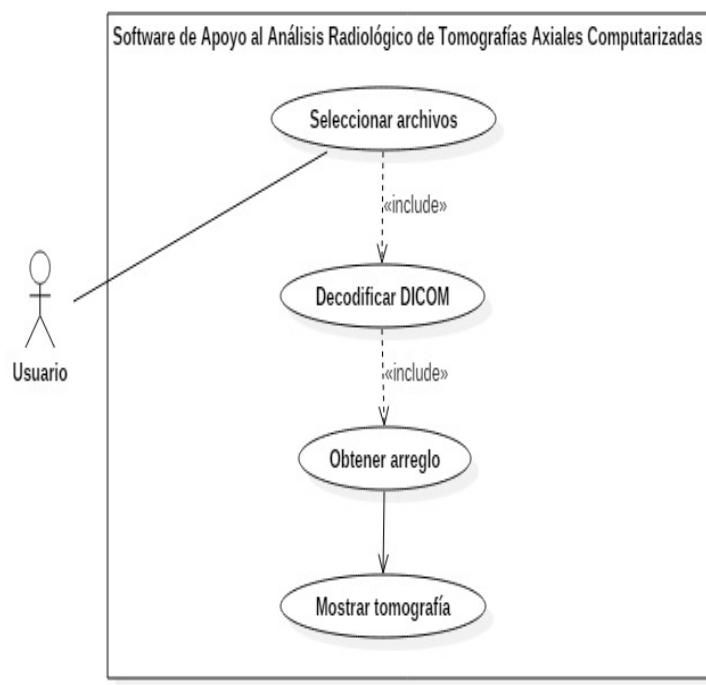


Figura 4.2: Diagrama caso de uso decodificación.
Imagen de elaboración propia.

La figura 5.2 muestra el diagrama de caso de uso de decodificación donde el usuario elige el archivo DICOM, una vez en el sistema el archivo se decodifica y se

obtiene la matriz que contiene los valores de Hounsfield y con base a esta última se muestra en 2D la tomografía.

4.2.3. Caso de uso selección de región

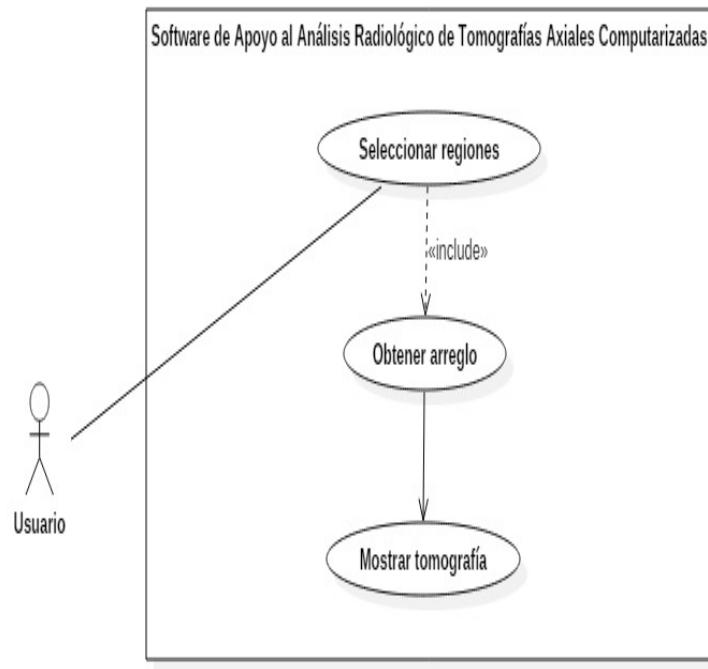


Figura 4.3: Diagrama caso de uso de selección de región.
Imagen de elaboración propia.

En el caso de uso de selección de región mostrado en la figura 5.3 el usuario selecciona una región determinada para su análisis, el archivo ya se ha decodificado con anterioridad por lo que no es necesario hacerlo de nuevo, con base a la región elegida se obtiene la matriz de valores de Hounsfield correspondiente y se muestra en 2D la región seleccionada.

4.2.4. Caso de uso tratamiento de imagen

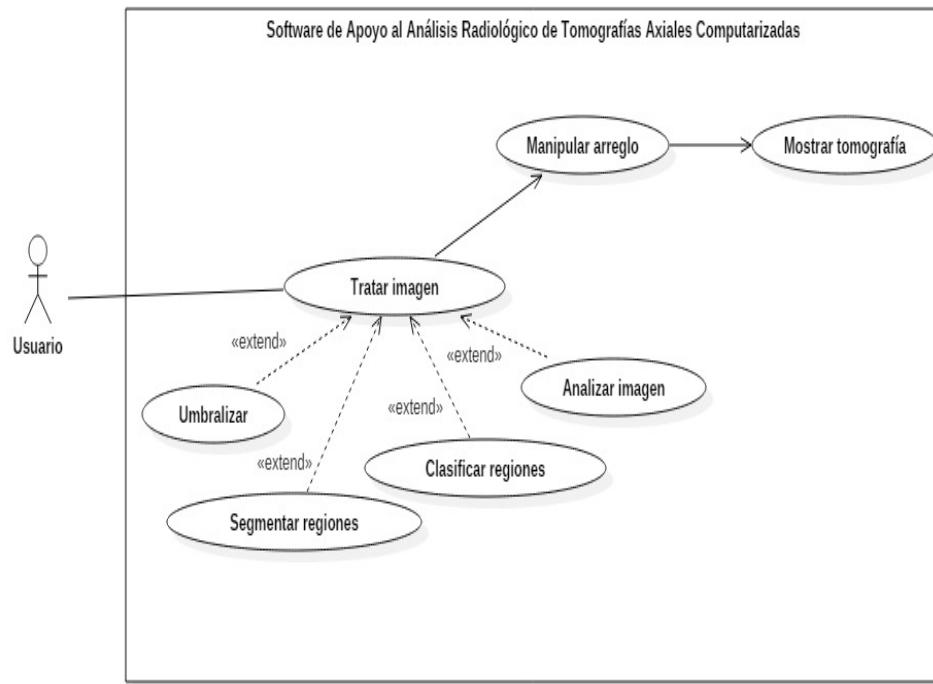


Figura 4.4: Diagrama caso de uso de tratamiento de imagen.
Imagen de elaboración propia.

El caso de uso de tratamiento de imagen que se muestra en la figura 5.4 el usuario tiene la posibilidad de hacer diversas operaciones sobre la región analizada, principalmente se podrán aplicar algoritmos para umbralizar, segmentar regiones, analizar regiones y el análisis de imagen, algunas operaciones que podrían entrar dentro del análisis de imagen son rotación, escalado, medición de segmentos, entre otros.

4.3. Descripción de casos de uso

4.3.1. Caso de uso seleccionar archivos

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Dar al sistema uno o más archivos DICOM para trabajar.
Resumen	El usuario podra elegir uno o más archivos de tipo DICOM que se deseen estudiar.
Entradas	Archivos de tipo DICOM.
Salidas	Ruta absoluta del archivo o archivos.
Precondiciones	Se elige al menos un archivo de tipo DICOM.
Postcondiciones	Los archivos seleccionados deben ser tipo DICOM, ningún otro es válido.
Tipo	Primario.
Módulo	1.Módulo de decodificación de DICOM.

Tabla 4.1: Caso de uso seleccionar archivos.

Flujo principal caso de uso seleccionar archivos

1. El usuario elige la pestaña *Archivo* y da clic en *Abrir archivo*.
2. Se muestra una ventana donde el usuario puede navegar entre las carpetas y archivos existentes en el equipo.
3. El usuario selecciona el archivo o archivos DICOM.
4. El usuario da click en *Aceptar*.
5. El sistema recibe la ruta del archivo de uno o varios archivos DICOM.[Alternativa A]

6. Fin del caso de uso.

Flujo alternativo A caso de uso seleccionar archivos. El usuario no selecciona ningún archivo.

A1. No se recibe ninguna ruta.

A2. Se manda un mensaje de error.

A3. Se despliega la ventana con las carpetas y archivos del equipo.

A4. El usuario selecciona el archivo o archivos DICOM.

A5. El usuario da click en *Aceptar*.

A6. El sistema recibe la ruta del archivo de uno o varios archivo DICOM.

A7. Fin del caso de uso.

4.3.2. Caso de uso decodificar DICOM

Atributo	Descripción
Versión	1.0
Actor(es)	Sistema
Propósito	Decodificar un archivo de tipo DICOM para obtener los valores de Hounsfield.
Resumen	El sistema decodifica el archivo o archivos DICOM seleccionados por el usuario anteriormente.
Entradas	Archivos DICOM.
Salidas	Valores de Hounsfield.
Precondiciones	El archivo cargado es tipo DICOM.
Postcondiciones	Los valores obtenidos están dentro de los rangos de la escala de Hounsfield.
Tipo	Primario.
Módulo	1. Módulo de decodificación de DICOM.

Tabla 4.2: Caso de uso decodificar DICOM.

Flujo principal caso de uso decodificar DICOM

1. El sistema accede a la ruta absoluta del archivo.
2. El sistema valida que el archivo sea tipo DICOM.[Ruta alternativa A]
3. Se crea un objeto DICOM.
4. Se almacena la información contenida en el archivo DICOM.
5. Fin del caso de uso.

Flujo alternativo A caso de uso decodificar DICOM. El usuario no selecciona ningún archivo.

- A1. El archivo no es de tipo DICOM.

- A2. Se manda un mensaje de error.
- A3. Se despliega la ventana con las carpetas y archivos del equipo.
- A4. El usuario selecciona el archivo o archivos DICOM.
- A5. Se lee y verifica que el archivo sea tipo DICOM.[Ruta alternativa A]
- A6. Se crea un objeto DICOM.
- A7. Se almacena la información contenida en el archivo DICOM.
- A8. Fin del caso de uso.

4.3.3. Caso de uso obtener arreglo

Atributo	Descripción
Versión	1.0
Actor(es)	Sistema
Propósito	Ordenar los valores de Hounsfield.
Resumen	El sistema almacena en un arreglo los valores de Hounsfield obtenidos de la imagen.
Entradas	Conjunto de valores decodificados.
Salidas	Arreglo con los valores de Hounsfield.
Precondiciones	El archivo DICOM fue decodificado.
Postcondiciones	El arreglo contiene únicamente valores dentro de los rangos de la escala de Hounsfield.
Tipo	Primario.
Módulo	1.Módulo de decodificación de DICOM.

Tabla 4.3: Caso de uso obtener arreglo.

Flujo principal caso obtener arreglo.

1. El sistema lee los valores obtenidos en la decodificación.
2. Se obtienen los valores correspondientes a cada segmento de la tomografía.
3. Se organizan los valores en una matriz.
4. Fin del caso de uso.

4.3.4. Caso de uso seleccionar región

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Acotar una región para su tratamiento.
Resumen	El usuario elige acotar una región y mediante una herramienta proporcionada por el sistema se selecciona lo que se desea analizar.
Entradas	Conjunto de coordenadas que comprenden la región.
Salidas	Conjunto de valores de Hounsfield correspondientes a la región.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	El arreglo contiene únicamente valores dentro de los rangos de la escala de Hounsfield.
Tipo	Primario.
Módulo	1.Módulo de decodificación de DICOM.

Tabla 4.4: Caso de uso seleccionar región.

Flujo principal caso seleccionar región.

1. El usuario activa la opción *Acotar región*.

2. El sistema proporciona al usuario una herramienta para la delimitación de un área.
3. El usuario delimita la región que desea analizar.
4. El sistema valida que la región esté dentro de los límites de la tomografía.[Flujo alternativo A.]
5. Se dan los valores correspondientes al área acotada.
6. Fin del caso de uso.

Flujo alternativo A caso de uso seleccionar región. El usuario no selecciona ningún archivo.

- A1. Uno de los límites de la región sale de al área válida.
- A2. Se manda un mensaje de error.
- A3. Se reposiciona la herramienta de acotación.
- A4. El usuario delimita la región que desea analizar.
- A5. El sistema valida que la región esté dentro de los límites de la tomografía.
- A6. Se dan los valores correspondientes al área acotada.
- A7. Fin del caso de uso.

4.3.5. Caso de uso tratar imagen

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Permitir al usuario realizar un conjunto de operaciones sobre la imagen.
Resumen	El usuario elige la operación que se hará sobre la imagen.
Entradas	Operación a realizar.
Salidas	Valores tratados.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	Los valores tratados respetan la escala de Hounsfield.
Tipo	Secundario.
Módulo	2.Tratamiento de imagen.

Tabla 4.5: Caso de uso tratar imagen.

Flujo principal caso tratar imagen.

1. El usuario elije el tratamiento que desea aplicar.
2. El sistema aplica los algoritmos requeridos a cada proceso.
3. Se generan los valores correspondientes obteniendo nuevos datos.
4. Fin del caso de uso.

4.3.6. Caso de uso umbralizar

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Umbralizar la imagen para obtener propiedades no visibles.
Resumen	El sistema aplica algoritmos de umbralización con base a las propiedades de la imagen.
Entradas	Arreglo a umbralizar.
Salidas	Imagen umbralizada.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	Los valores están dentro de la escala de Hounsfield.
Tipo	Primario.
Módulo	2.Tratamiento de imagen.

Tabla 4.6: Caso de uso umbralizar.

Flujo principal caso umbralizar.

1. El usuario selecciona la opción *Umbralizar*.
2. El sistema lee la matriz de la región a umbralizar.
3. Se aplica una serie de algoritmos.
4. Se obtienen los valores umbralizados.
5. Fin del caso de uso.

4.3.7. Caso de uso segmentar regiones

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Segmentar una o varias regiones.
Resumen	Se aplican los algoritmos requeridos para una segmentación.
Entradas	Arreglo de valores de Hounsfield.
Salidas	Valores segmentados.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	Valores tratados respetan la escala de Hounsfield.
Tipo	Primario.
Módulo	2.Tratamiento de imagen.

Tabla 4.7: Caso de uso segmentar regiones.

Flujo principal caso segmentar regiones.

1. El usuario selecciona la opción *Segmentar*.
2. El sistema aplica los algoritmos requeridos.
3. Se obtienen los valores de la segmentación.
4. Fin del caso de uso.

4.3.8. Caso de uso clasificar regiones

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Clasificar las regiones con base a distintos criterios.
Resumen	Se aplican los algoritmos requeridos para una clasificación dependiendo el criterio.
Entradas	Arreglo de valores de Hounsfield.
Salidas	Valores clasificados.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	Valores tratados respetan la escala de Hounsfield.
Tipo	Primario.
Módulo	2.Tratamiento de imagen.

Tabla 4.8: Caso de uso clasificar regiones.

Flujo principal caso clasificar regiones.

1. El usuario selecciona la opción *Clasificar*.
2. El sistema aplica los algoritmos dependiendo el criterio de clasificación.
3. Se clasifican los distintos valores.
4. Fin del caso de uso.

4.3.9. Caso de uso analizar regiones

Atributo	Descripción
Versión	1.0
Actor(es)	Usuario, Sistema
Propósito	Aplicar operaciones específicas sobre la imagen.
Resumen	Se aplican operaciones de menor complejidad.
Entradas	Arreglo de valores de Hounsfield y dependiendo la operación los parámetros necesarios.
Salidas	Valores manipulados.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	La imagen mantiene su integridad.
Tipo	Secundario.
Módulo	2.Tratamiento de imagen.

Tabla 4.9: Caso de uso analizar regiones.

Flujo principal caso analizar regiones.

1. El usuario selecciona la operación deseada, como pueden ser *Rotar*, *Escalar*, *Medir*, entre otras.
2. El usuario proporciona los parámetros necesarios.
3. Se aplica la operación.
4. Se devuelve el resultado de aplicar la operación.
5. Fin del caso de uso.

4.3.10. Caso de uso manipular arreglo

Atributo	Descripción
Versión	1.0
Actor(es)	Sistema
Propósito	Asignar los nuevos valores a la matriz sin perder la información.
Resumen	Se asignan los valores nuevos a las matrices cuidando la integridad de las demás.
Entradas	Valores de Hounsfield.
Salidas	Arreglo manipulado.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	La imagen mantiene su integridad.
Tipo	Primario.
Módulo	2.Tratamiento de imagen.

Tabla 4.10: Caso de uso manipular arreglo.

Flujo principal caso manipular arreglo.

1. Se trata la imagen.
2. Se envía el conjunto de valores ya tratados.
3. Se guardan en la nueva matriz.
4. Fin del caso de uso.

4.3.11. Caso de uso mostrar tomografía

Atributo	Descripción
Versión	1.0
Actor(es)	Sistema
Propósito	Mostrar la tomografía al usuario.
Resumen	Con base a los valores de Hounsfield se genera la visualización de la tomografía en 2D.
Entradas	Arreglo con valores de Hounsfield.
Salidas	Imagen en 2D de la tomografía.
Precondiciones	La imagen ya fue decodificada y el arreglo obtenido.
Postcondiciones	La matriz de Hounsfield no se desecha, se guarda para usos posteriores.
Tipo	Primario.
Módulo	3.Visualización.

Tabla 4.11: Caso de uso mostrar tomografía.

Flujo principal caso mostrar tomografía.

1. Se recibe el arreglo de valores de Hounsfield.
2. Se genera la visualización de la tomografía en 2D.
3. Fin del caso de uso.

4.4. Diagramas de secuencia

4.4.1. Secuencia: Decodificar DICOM

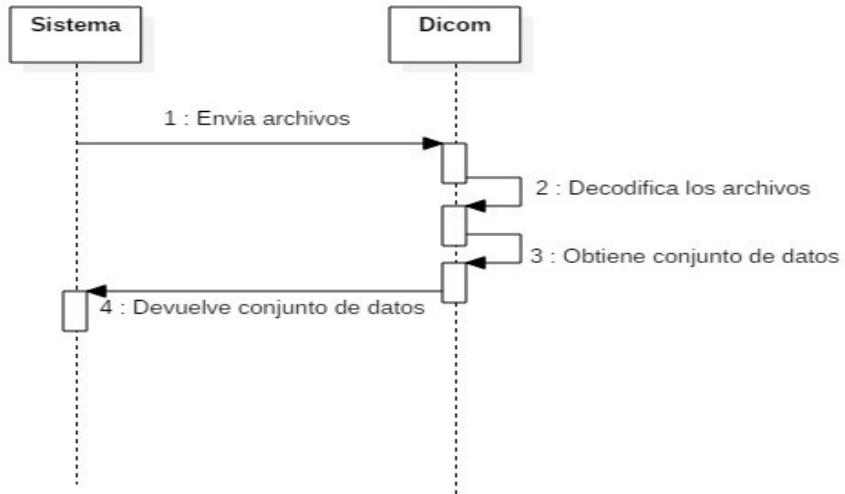


Figura 4.5: Secuencia:Decodificar DICOM.

Imagen de elaboración propia.

4.4.2. Secuencia: Seleccionar archivos

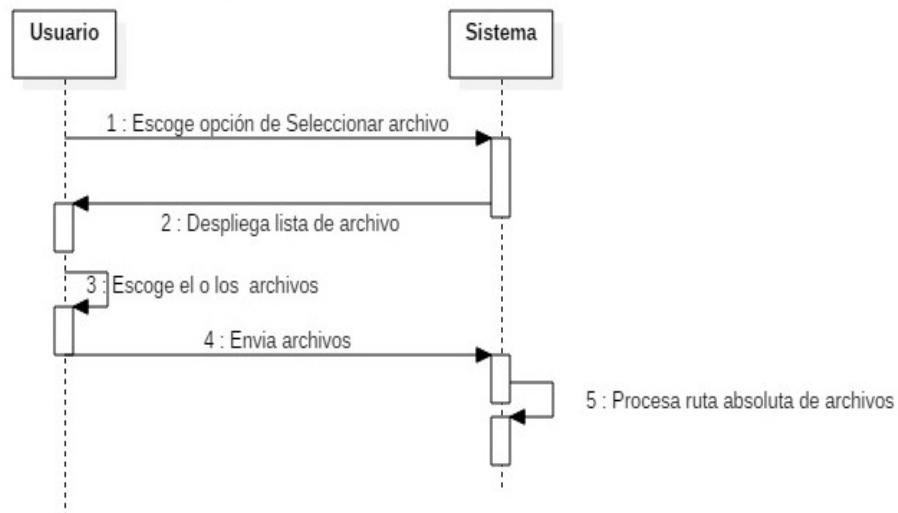


Figura 4.6: Secuencia:Seleccionar archivos.
Imagen de elaboración propia.

4.4.3. Secuencia: Obtener arreglo

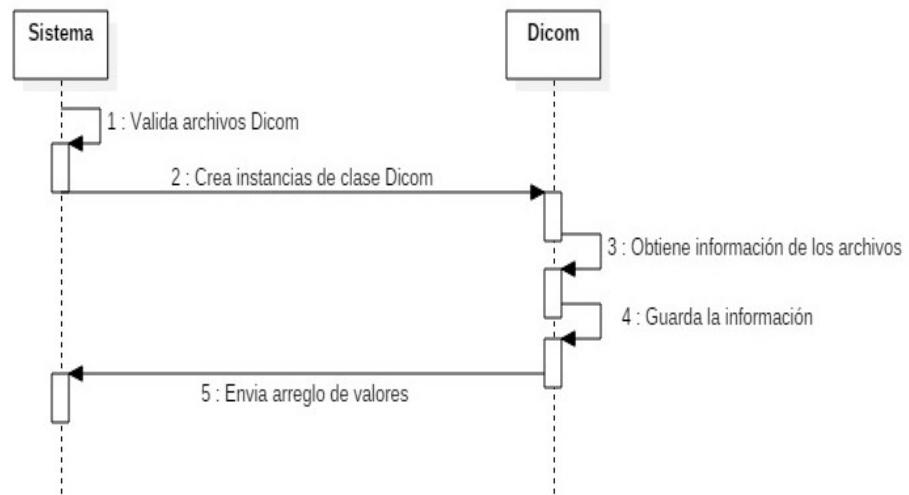


Figura 4.7: Secuencia:Obtener arreglo.
Imagen de elaboración propia.

4.4.4. Secuencia: Seleccionar regiones

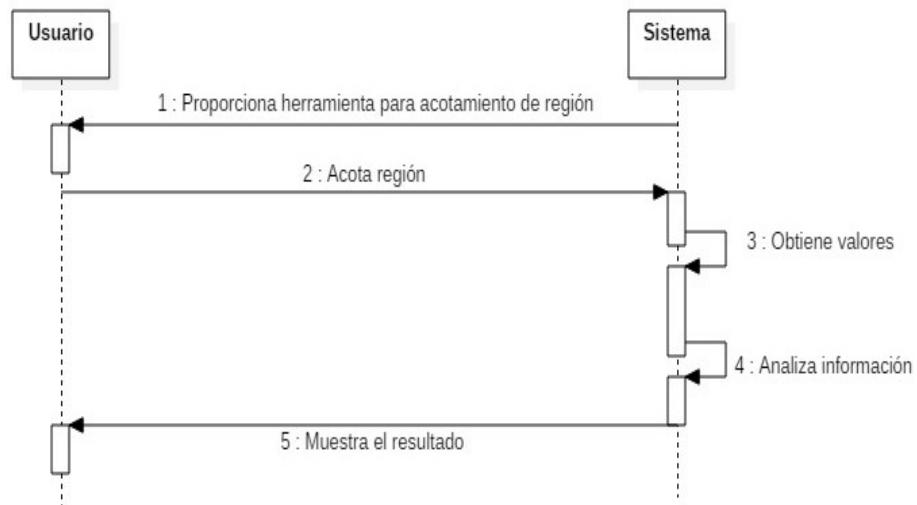


Figura 4.8: Secuencia: Seleccionar regiones.
Imagen de elaboración propia.

4.4.5. Secuencia: Analizar imagen

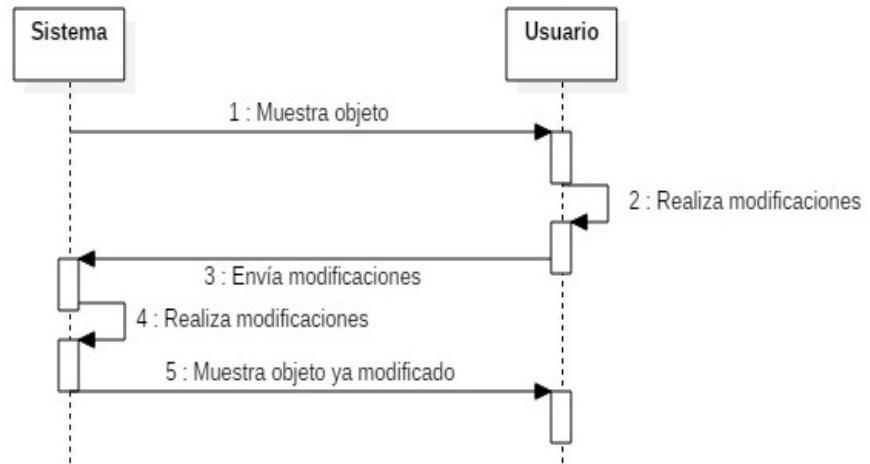


Figura 4.9: Secuencia:Analizar imagen.
Imagen de elaboración propia.

4.4.6. Secuencia: Clasificar regiones

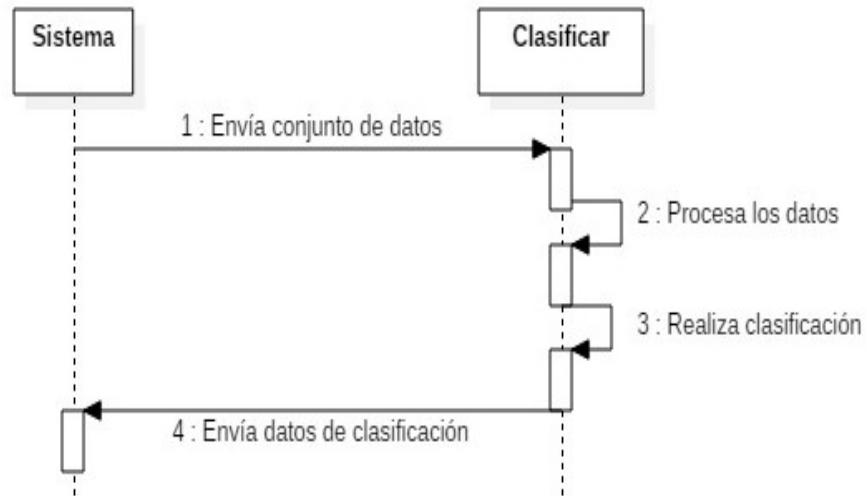


Figura 4.10: Secuencia: Clasificar regiones.
Imagen de elaboración propia.

4.4.7. Secuencia: Manipular arreglo

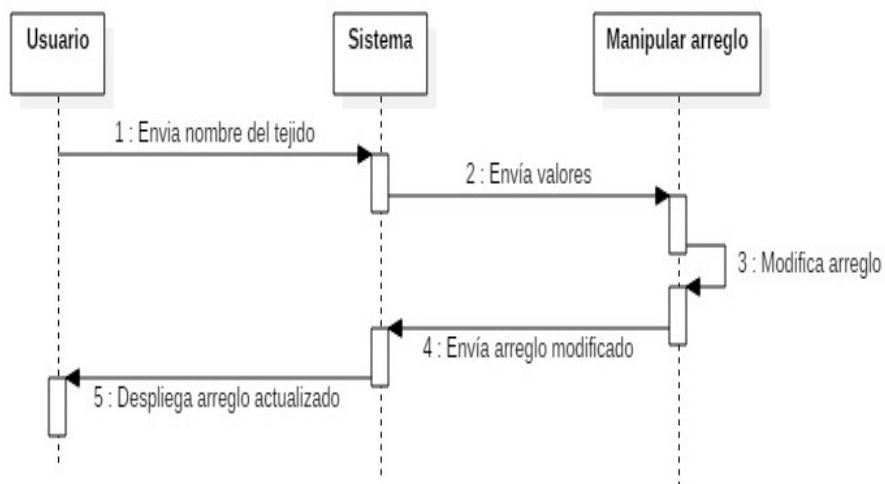


Figura 4.11: Secuencia:Manipular arreglo.
Imagen de elaboración propia.

4.4.8. Secuencia: Segmentar regiones

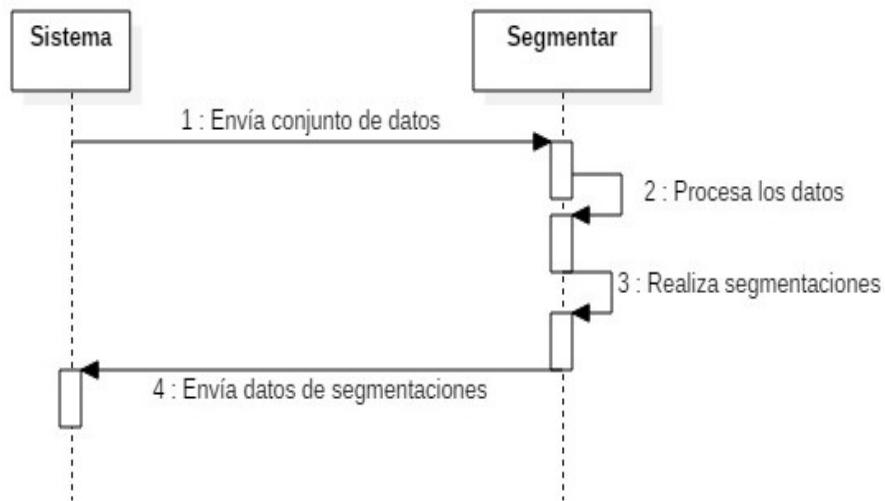


Figura 4.12: Secuencia:Segmentar regiones.
Imagen de elaboración propia.

4.4.9. Secuencia: Tratar imagen

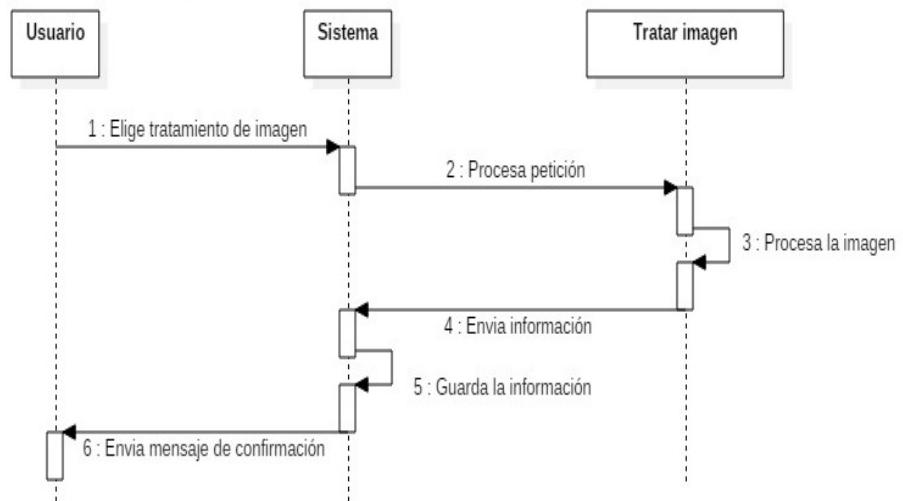


Figura 4.13: Secuencia:Tratar imagen.
Imagen de elaboración propia.

4.4.10. Secuencia: Umbralizar

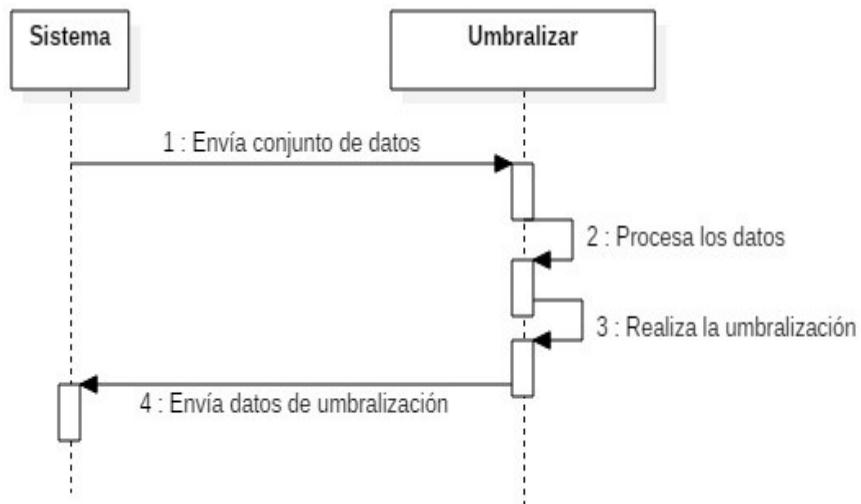


Figura 4.14: Secuencia:Umbralizar.

Imagen de elaboración propia.

4.4.11. Secuencia: Mostrar tomografía

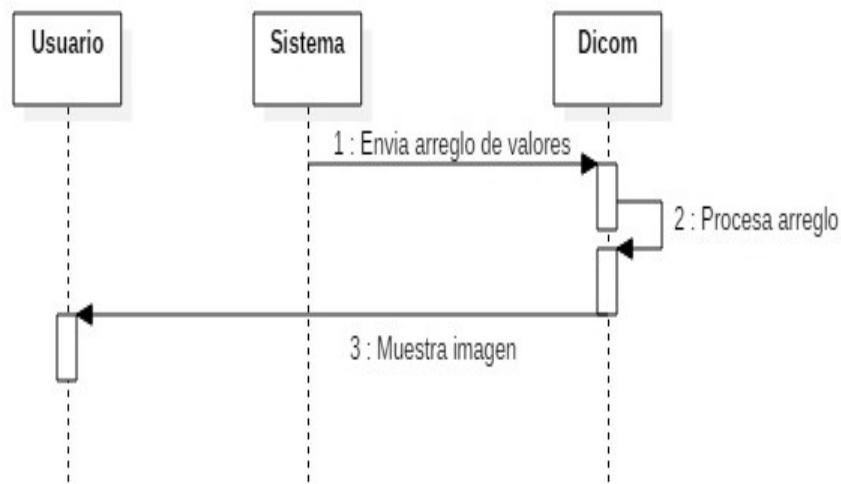


Figura 4.15: Secuencia:Mostrar tomografía.
Imagen de elaboración propia.

Capítulo 5

Desarrollo

En este capítulo se explican cuales fueron los algoritmos elegidos para el sistema, donde se utilizan, se explica como fue su implementación y cómo se modifican los parámetros en cada uno para obtener diferentes resultados.

5.1. Decodificación de DICOM

Para la decodificación de los archivos DICOM se hace uso de la función *dicom.read_file* encontrada en la librería *PYDICOM* del lenguaje Python, la cual al enviarle la ruta de la carpeta donde se encuentran los archivos DICOM nos permite obtener toda la información de cada uno. Una de las etiquetas propias de la librería es el atributo *pixel_array*, el cual nos devuelve la matriz de densidades. Utilizando las etiquetas estandarizadas en DICOM podemos obtener los datos que deseemos. Algunos ejemplos de las etiquetas utilizadas son:

Etiqueta	Parámetro obtenido
0X28, 0X30	Distancia entre centros de pixeles
0X10, 0X10	Nombre del paciente
0X10, 0X1010	Edad del paciente
0X10, 0X40	Sexo del paciente
0X8, 0X20	Fecha de toma del estudio
0X18, 0X50	Espesor de tomografía
0X08, 0X08	Hospital o clínica donde fue tomado el estudio

Tabla 5.1: Etiquetas DICOM usadas en el sistema.

5.2. Comunicación entre lenguajes

Para una mejor implementación del sistema se usan 2 lenguajes, lo cual nos lleva a la tarea de enviar información entre ellos. El framework .NET nos facilita esta tarea mediante el uso de procesos, con funciones basadas en arquitectura tipo *pipelines*, esta arquitectura permite que los procesos creados accedan a otro programa y el flujo de datos generado en ese programa se conserve en el proceso padre.

La arquitectura se pudo implementar gracias a la librería *System.Diagnostics* que nos da acceso a objetos de tipo *ProcessStartInfo* dentro del framework .NET, se hace uso de funciones como *UseShellExecute* para iniciar la ejecución de un programa desde los procesos creados y *RedirectStandardOutput* para leer el flujo de datos.

5.3. Umbralización

La naturaleza de la TAC nos permite obtener una matriz de densidades. Gracias a la clasificación propuesta por Hounsfield se parametrizan los órganos y tejidos teniendo un valor y una ventana de tolerancia en cada uno. Tomando los valores de la escala de Hounsfield se procedió a implementar una umbralización por definición. Una vez que se ha elegido el tejido que se desea visualizar se recorre toda la matriz de densidades y se hace un mapeo con un color distinto cuando el valor de pixel se

encuentra dentro del rango definido por la escala.

Los tejidos que es posible visualizar en el sistema son *agua*, *aire*, *fluido cerebroespinal*, *sustancia cerebral blanca*, *sustancia cerebral gris*, *grasa*, *hueso compacto*, *hueso esponjoso*, *hígado*, *páncreas*, *pulmón*, *riñón*, *sangre y sangre coagulada*.

5.4. Segmentación

La segmentación en este sistema tiene como intención el poder aislar los tejidos entre si. Para estos procedimientos se recurrió a dos técnicas de segmentación distintas, la primera de ellas es la técnica *Región creciente*, la cual requiere de una interacción con el usuario; la otra técnica utilizada es *Split and merge* la cual no requiere de ninguna interacción para su funcionamiento.

5.4.1. Región creciente

El algoritmo *Región creciente* nos permite encontrar la continuidad de un tejido seleccionado por el usuario. Una vez que el usuario selecciona el punto, se inicia una búsqueda de amplitud con parámetros de similitud basados en distancia Euclíadiana y proximidad al punto de origen, encontrando así pixeles con características similares entre si con el fin de formar un objeto.

5.4.2. Split and merge

Split and merge nos permite agrupar objetos(tejidos) con características de similitud basadas en su UH y la proximidad entre los pixeles. Una de las ventajas de utilizar este método a diferencia del algoritmo de *Región creciente* es que aquí no se tiene un punto de partida sobre el cual hacer la búsqueda por lo que tenemos una mayor área para encontrar objetos sin embargo esto puede significar un problema para la delimitación de diferentes objetos debido a la abundante cantidad de datos presente.

5.5. Clasificación

Los clasificadores nos permiten agrupar pixeles con características similares, se hicieron 2 implementaciones para cada uno de los algoritmos elegidos, una supervisada y una no supervisada, en ambas implementaciones el usuario tiene la opción de

definir el número de grupos que desea encontrar en la TAC. En la implementación supervisada el usuario define los centros de cada grupo.

5.5.1. K-means

Se implementó el algoritmo *K-means* para la creación de grupos dentro de la TAC, se usa la distancia Euclidiana entre pixeles como parámetro de similitud. La distancia Euclidiana es calculada con base a las densidades de los tejidos. Este algoritmo es muy útil cuando se buscan grupos cuya definición no se encuentra dentro del conjunto de información.

5.5.2. Fuzzy C-Means

El algoritmo *Fuzzy C-Means* trabaja de manera similar al algoritmo anterior pero en este la diferencia entre pixeles es menos tolerable dando así un valor más definido de similitud. Además se cuenta con una matriz de pertenencia la cual nos ayuda a identificar el cluster al que pertenece el pixel analizado. La matriz da un valor a cada cluster para cada pixel y se elige el que mayor pertenencia tiene.

Capítulo 6

Pruebas

En este capítulo se muestran los diferentes resultados obtenidos durante el funcionamiento del sistema. Para poder llevar acabo estas pruebas de funcionamiento se utilizaron TAC proporcionadas por uno de nuestros directores. En los estudios podemos observar desde el cerebro hasta los pies de diversos pacientes lo cual enriquece estas pruebas y nos permite tener una mayor perspectiva al momento de hacer una autoevaluación del sistema.

6.1. Umbralización

Las imágenes mostradas a continuación corresponden a diferentes umbralizaciones realizadas con el sistema.

En la umbralización de fluido cerebro espinal se utilizó un TAC obtenida de la zona de la cabeza de un paciente. Teniendo del lado derecho la tomografía sin ningún tratamiento y del lado izquierdo la imagen tratada el fluido puede observarse con un tono verde.

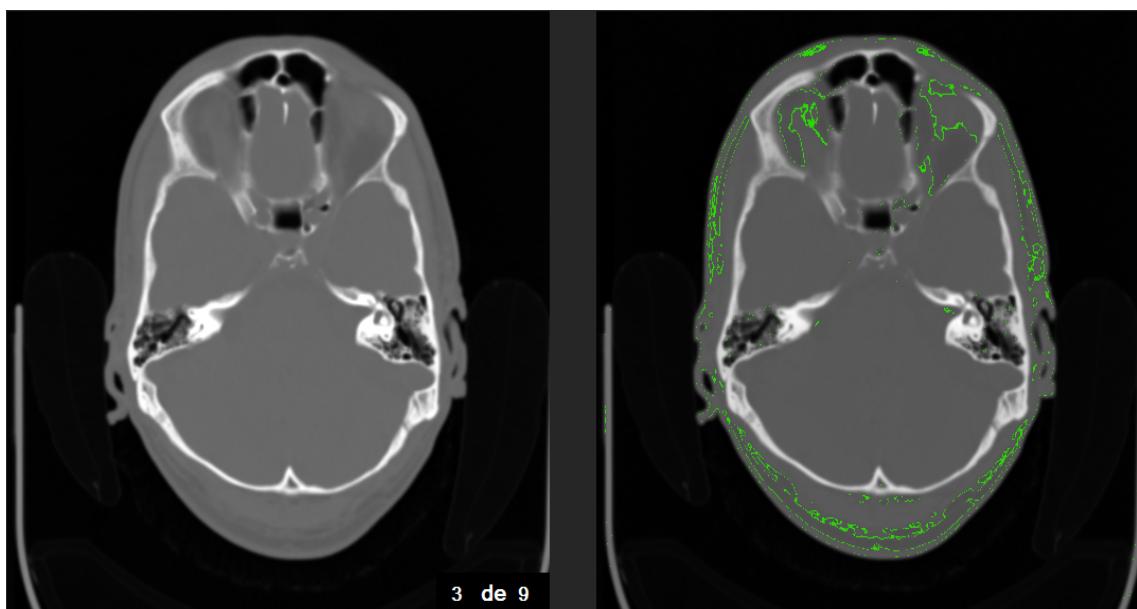


Figura 6.1: Umbralización de fluido cerebro espinal.
Captura de pantalla del sistema en funcionamiento.

Para la umbralización del tejido graso se utilizó una tomografía de la zona abdominal, en esta zona es donde mayor acumulación de grasa hay regularmente, en otras zonas del cuerpo podemos observar este tejido en la piel.

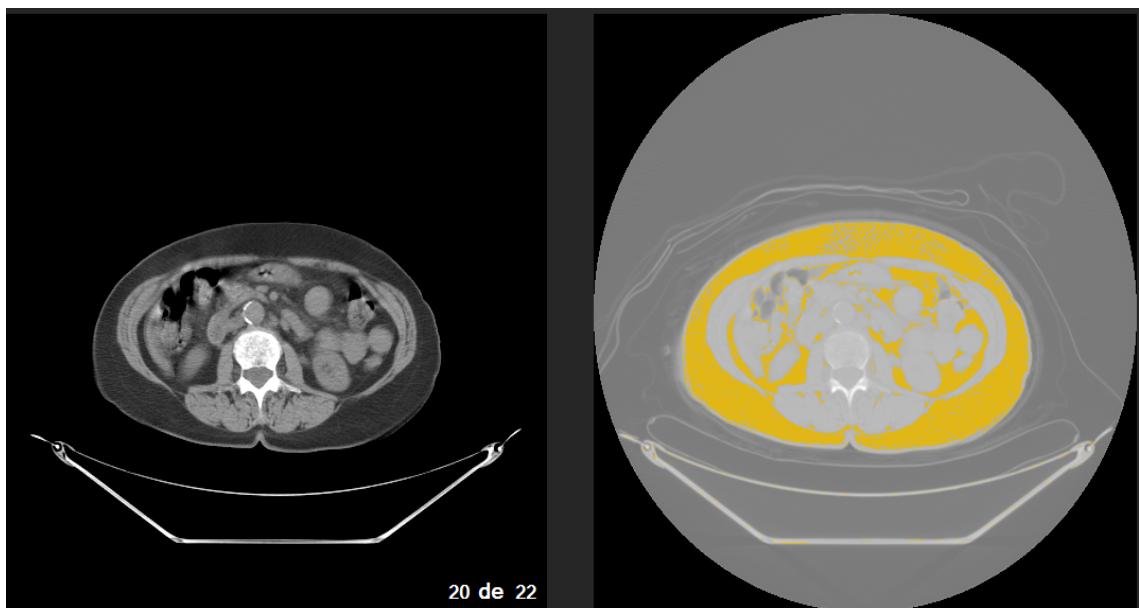


Figura 6.2: Umbralización de grasa.
Captura de pantalla del sistema en funcionamiento.

En la umbralización de los pulmones podemos observar que no es una figura muy sólida y el tejido es muy similar al aire debido al contenido de estos órganos.

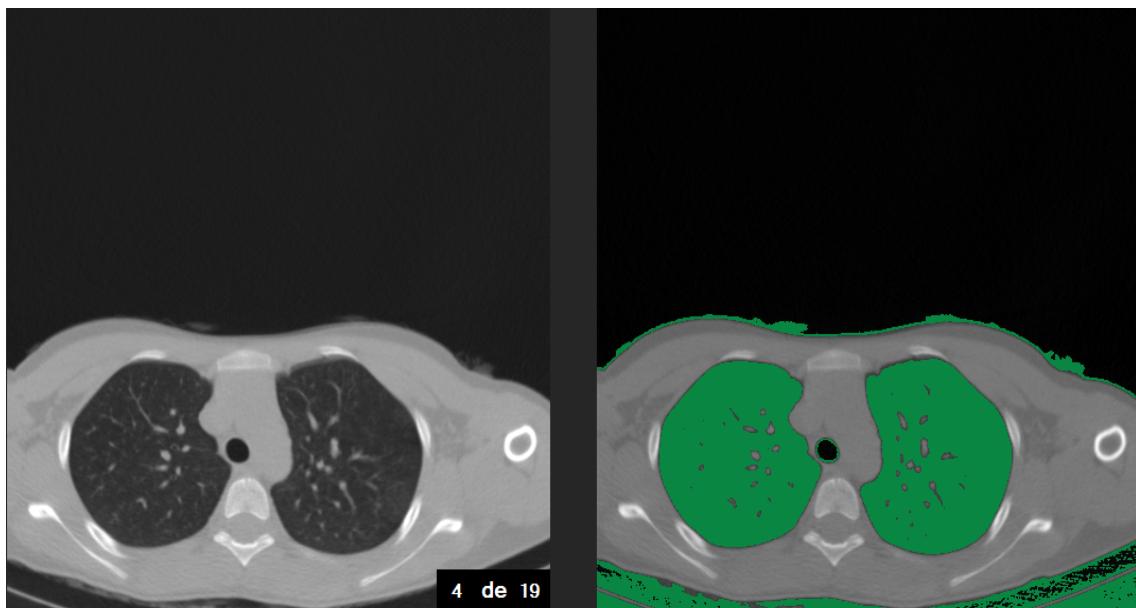


Figura 6.3: Umbralización de pulmón.
Captura de pantalla del sistema en funcionamiento.

Continuando con la tomografía de zona abdominal se realizó una umbralización para hueso compacto, donde únicamente podemos ver lo correspondiente a la columna vertebral con un tono rosaceo del lado izquierdo.

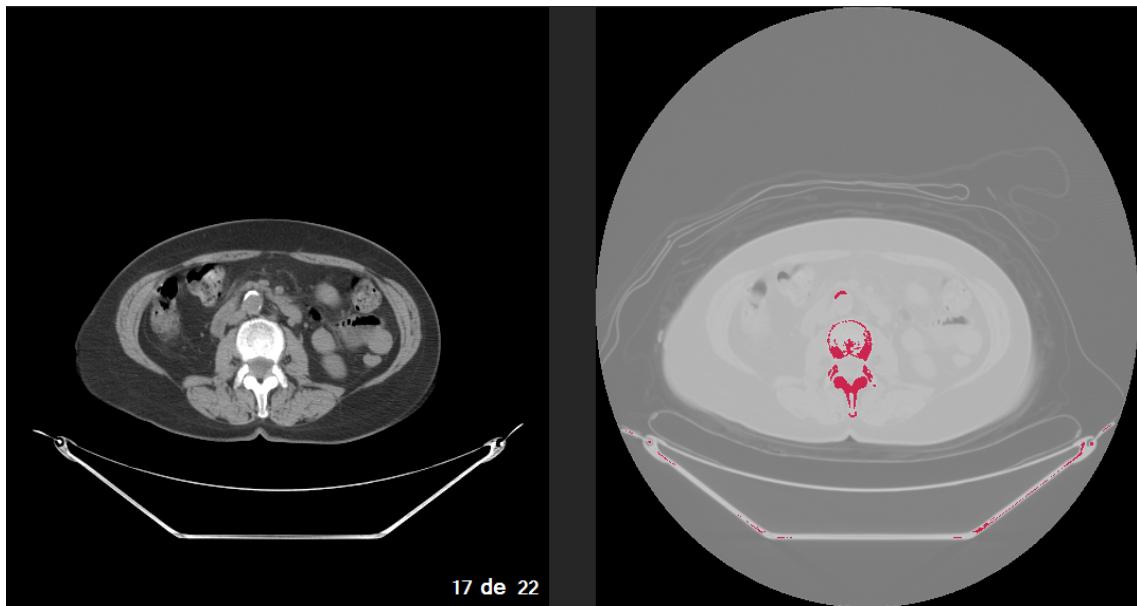


Figura 6.4: Umbralización de hueso compacto.
Captura de pantalla del sistema en funcionamiento.

A diferencia del estudio utilizado para el hueso compacto en el hueso esponjoso se utilizó un estudio cerebral donde podemos ver la protección que el cráneo da al cerebro, el hueso esponjoso regularmente se combina con hueso compacto.

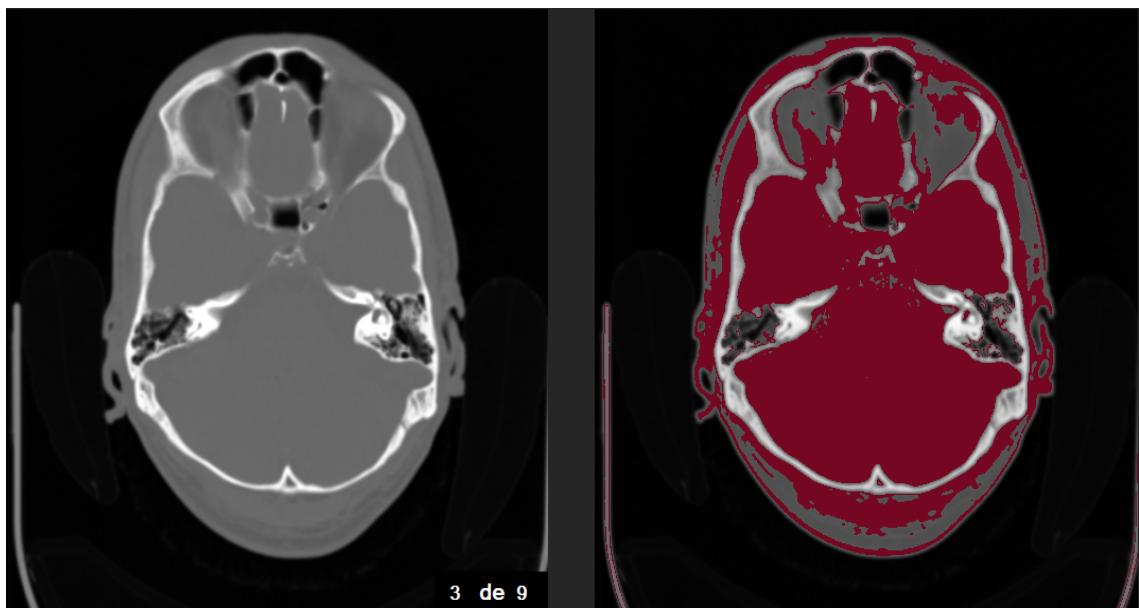


Figura 6.5: Umbralización de hueso esponjoso.
Captura de pantalla del sistema en funcionamiento.

La umbralización de sangre se realizó sobre una tomografía de cerebro, el cerebro está inundado de diferentes fluidos por lo cual no observaremos una gran continuidad como se puede ver en el hueso esponjoso.

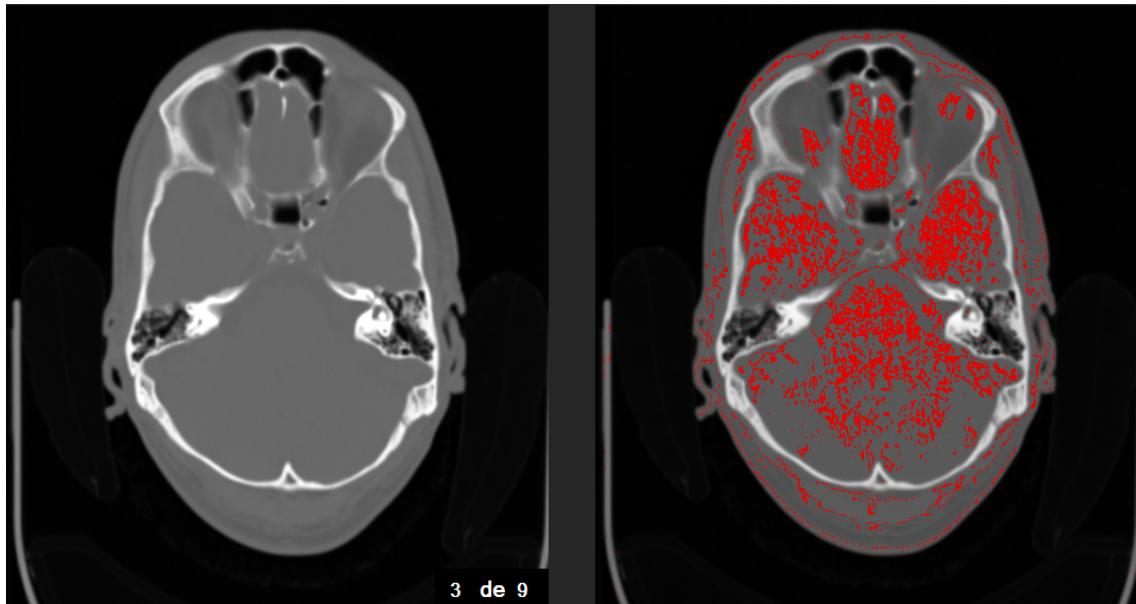


Figura 6.6: Umbralización de sangre.
Captura de pantalla del sistema en funcionamiento.

Otro fluido importante que encontramos en el cerebro es la sustancia cerebral gris, esta se encuentra en una menor proporción que la sangre.

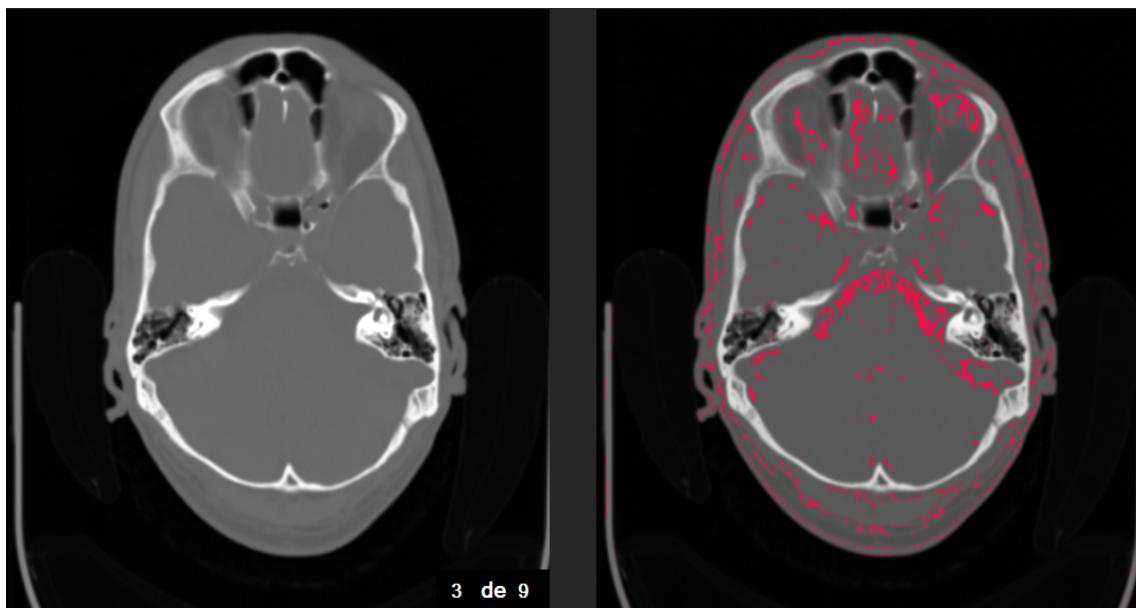


Figura 6.7: Umbralización de sustancia cerebral gris.
Captura de pantalla del sistema en funcionamiento.

6.2. Contraste

Las herramientas de contraste en los visores DICOM son muy importantes pues permiten obtener un mayor detalle de las estructuras internas del cuerpo sin hacer un tratamiento sobre la TAC, en este caso se hace un mapeo con la escala de gris para dar más o menos valores a cada tejido.

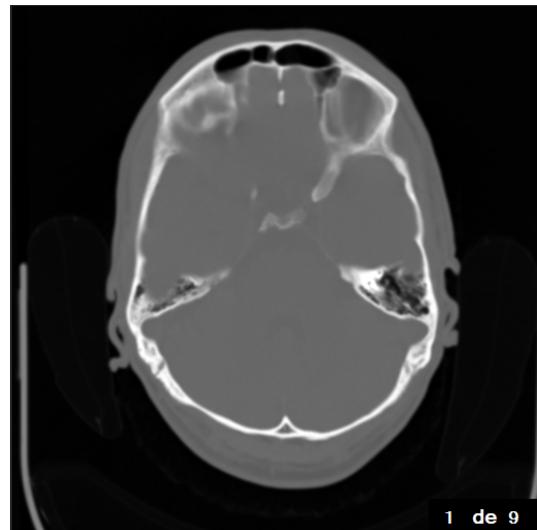


Figura 6.8: TAC con un contraste completo.
Captura de pantalla del sistema en funcionamiento.



Figura 6.9: TAC con el contraste recomendado para cerebro.
Captura de pantalla del sistema en funcionamiento.

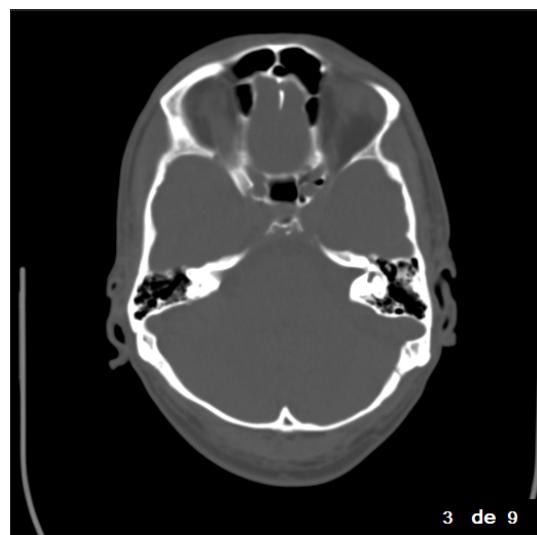


Figura 6.10: TAC con el contraste recomendado para hueso.
Captura de pantalla del sistema en funcionamiento.

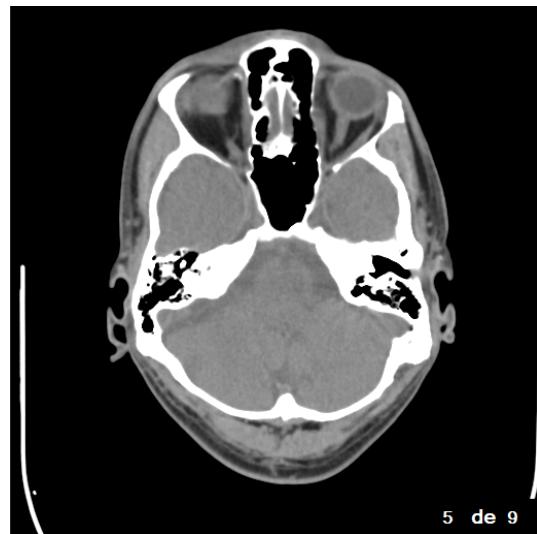


Figura 6.11: TAC con el contraste recomendado para partes blandas.
Captura de pantalla del sistema en funcionamiento.



Figura 6.12: TAC con el contraste recomendado para pulmón.
Captura de pantalla del sistema en funcionamiento.

6.3. Clasificadores

A continuación se muestran los resultados obtenidos con las técnicas de clasificación utilizadas.

La técnica Fuzzy C-Means se implementa sobre una TAC en la zona abdominal donde podemos encontrar una gran acumulación de diferentes tejidos. Se pueden observar los valores máximos y mínimos de cada grupo y el color con el que está representado. Adicionalmente se puede observar la implementación de la herramienta de zoom para ver con más detalles algunos tejidos.

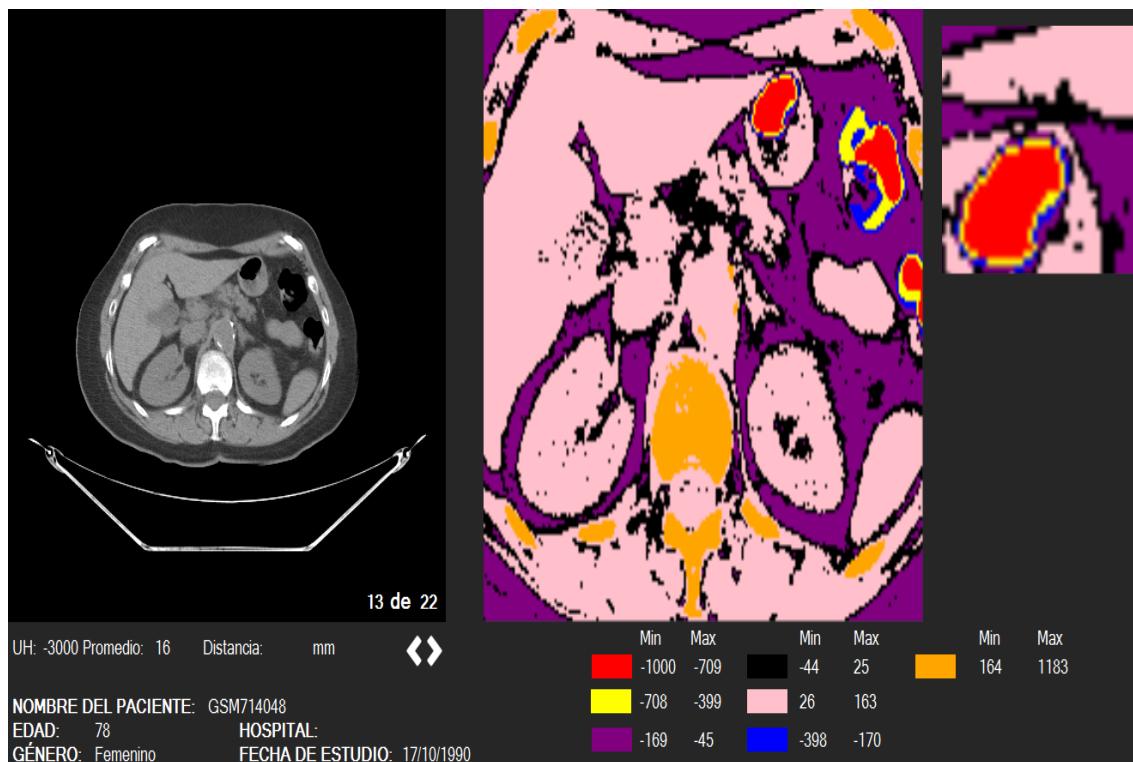


Figura 6.13: Algoritmo Fuzzy C-Means aplicado a una TAC.
Captura de pantalla del sistema en funcionamiento.

La técnica K-Means se implementa sobre una TAC de la zona torácica.

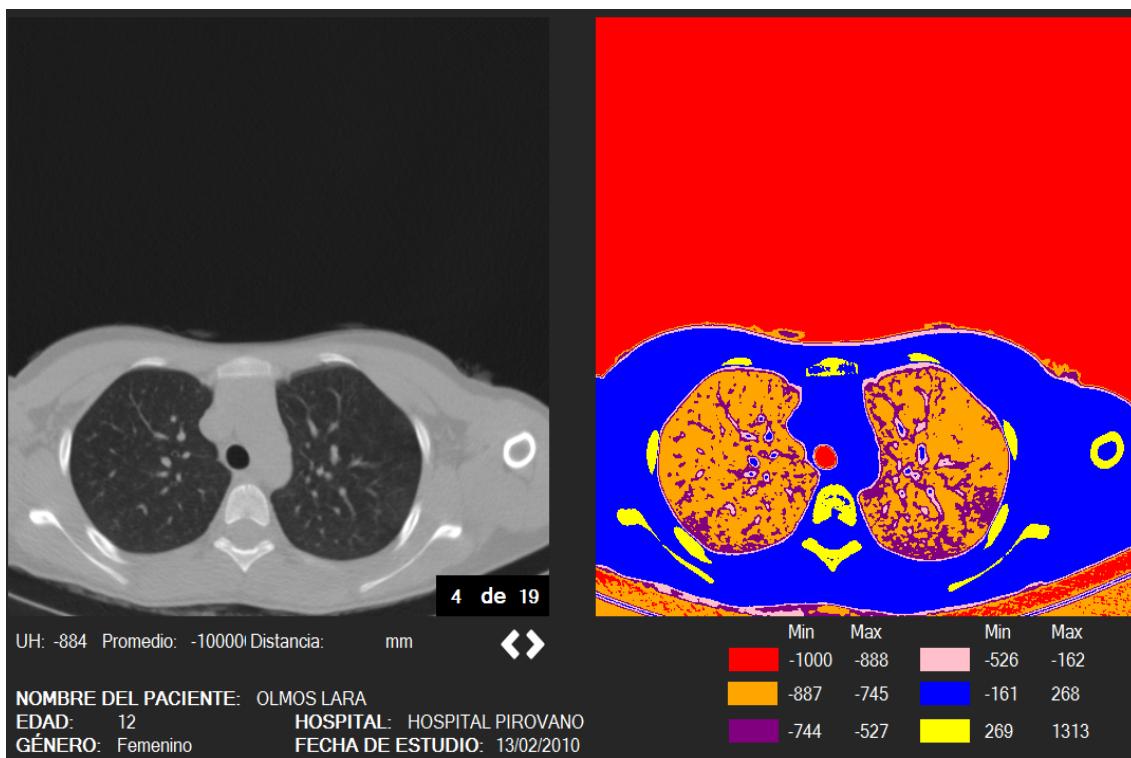


Figura 6.14: Algoritmo K-Means aplicado a una TAC.
Captura de pantalla del sistema en funcionamiento.

6.4. Segmentación

La técnica de Región creciente se implementa sobre una TAC de zona abdominal y intenta aislar órganos de hueso y fluidos.

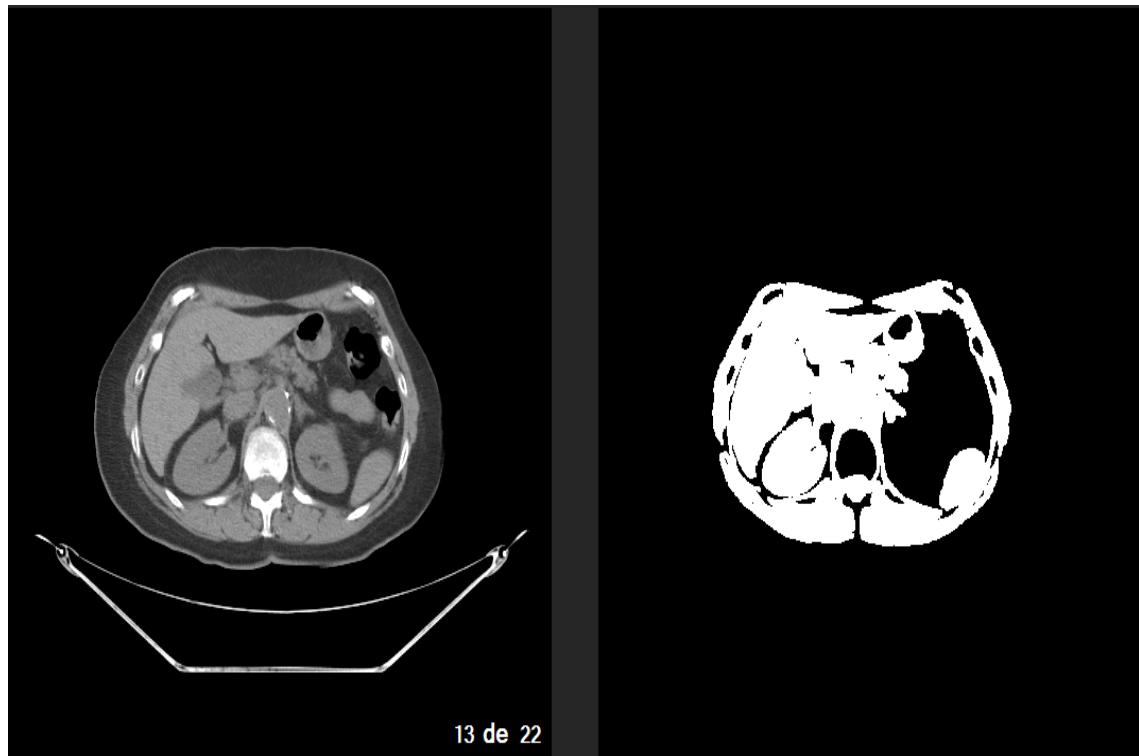


Figura 6.15: Región creciente en zona abdominal.
Captura de pantalla del sistema en funcionamiento.

6.5. Representación en 3D

Se utiliza una TAC de la zona craneal para mostrar al funcionamiento de la reconstrucción en 3D.

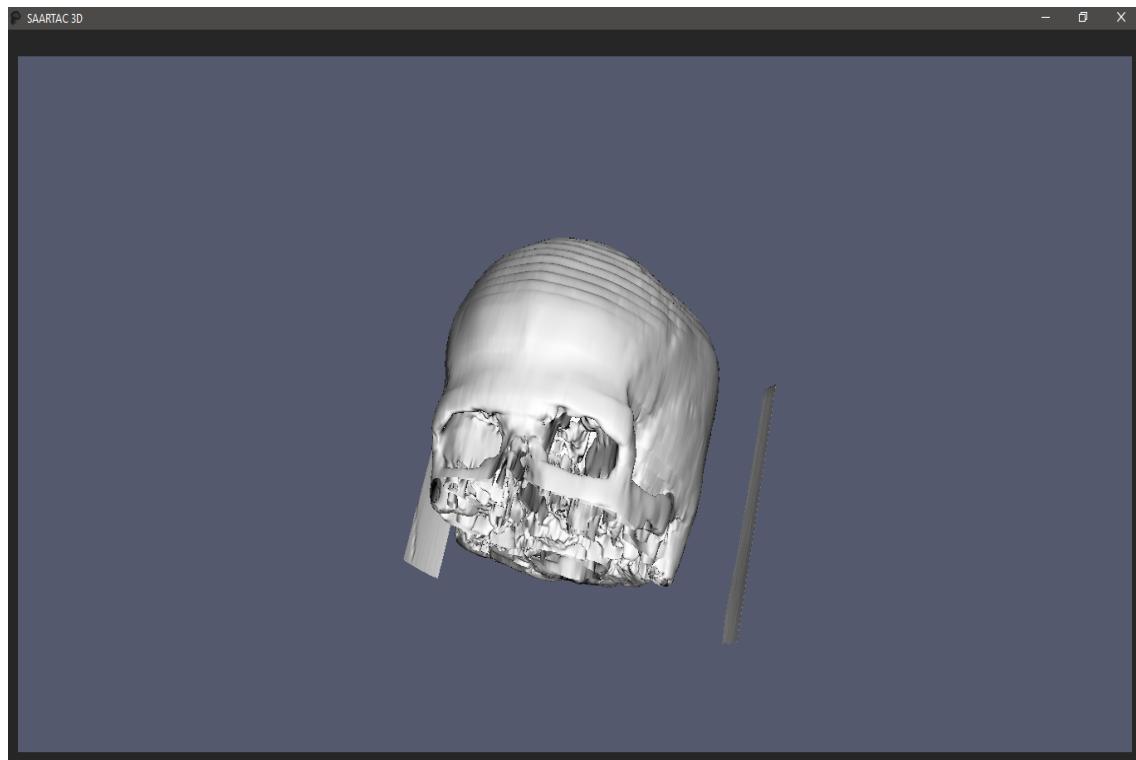


Figura 6.16: Reconstrucción 3D de un cráneo.
Captura de pantalla del sistema en funcionamiento.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Una vez terminado el sistema se llegó a las siguientes conclusiones:

- Se cumplieron los objetivos planteados para el desarrollo del sistema.
- La extensión del sistema cubre aspectos que las herramientas actuales no proporcionan sin embargo sigue siendo una herramienta de menor nivel que los softwares ofrecidos en los equipos médicos especializados.
- El hacer una implementación propia de los algoritmos nos ayudó pues pudimos orientarlos a nuestra arquitectura y hacer un manejo integral de los datos, quitando las limitaciones que en muchas ocasiones tienen las librerías.
- Cuando el especialista encuentra incongruencias dentro de la TAC el uso de técnicas no supervisadas puede ser muy útil como una segunda opinión de los resultados obtenidos.

7.2. Trabajo a futuro

Las propuestas de trabajo futuro para este sistema son las siguientes:

- Incluir más técnicas de clasificación y segmentación para el análisis de TAC.
- Implementar métodos exploratorios en las TAC.
- Hacer una implementación propia para la representación en 3D de TAC.

- Incluir un módulo de comparación entre las tomografías analizadas e históricos de la medicina.
- Agregar más herramientas que faciliten el manejo de las TAC y complementen el análisis.
- Permitir al especialista observar todos los campos disponibles en la cabecera del archivo DICOM.
- Desarrollar una API de la aplicación para permitir su uso a terceros.

Apéndice A

Código K-Means

Esta función genera centros aleatorios.

```
1  public void generarCentros(){
2      centros = new List<Double>();
3      rnd = new Random();
4      for (int i = 0; i < numerosK; i++) {
5          int numero = rnd.Next((IgnorarAire ? IGNORAR : -1000),
6                                 1400);
7          centros.Add((double)numero);
8      }
}
```

Esta función Euclíadiana entre dos densidades.

```
1  public void distanciaEuclidiana(int i, int j){
2      if (IgnorarAire && datos[i][j] < IGNORAR) return;
3      int indc = 0;
4      conjunto.Clear();
5      foreach (Double indice in centros){
6          Double resta = Math.Abs(datos[i][j] - indice);
7          conjunto.Add(resta);
8      }
9      for (int k = 1; k < conjunto.Count; k++)
10         if (conjunto[indc] > conjunto[k])
11             indc = k;
12     clases[i, j] = indc + 1;
13 }
```

La función promedio saca el promedio de las densidaes en el cluster y genera nuevos centros con base en el promedio calculado.

```
1     public void promedio(){
2         centros.Clear();
3         double[] sumas = new double[numerosK];
4         double[] contador = new double[numerosK];
5         for (int i = 0; i < numerosK; i++)
6             sumas[i] = contador[i] = 0;
7         for (int i = 0; i < datos.Count; i++){
8             for (int j = 0; j < datos[i].Length; j++){
9                 if (IgnorarAire && datos[i][j] < IGNORAR)
10                     continue;
11                 sumas[clases[i, j] - 1] += datos[i][j];
12                 contador[clases[i, j] - 1]++;
13             }
14             if (reporte_progreso.CancellationPending)
15                 return;
16             operaciones_cargando++;
17             reporte_progreso.ReportProgress((90 *
18                 operaciones_cargando) / operaciones_total);
19         }
20         centros.Clear();
21         for (int i = 0; i < numerosK; i++) {
22             if (contador[i] == 0) {
23                 centros.Add(ObtenPixelRandom());
24             } else {
25                 centros.Add(sumas[i] / contador[i]);
26             }
27         }
28     }
```

Apéndice B

Código Fuzzy C-Means

Se generan los centros aleatorios.

```
1     public void generarCentros(){
2         centros = new List<Double>();
3         rnd = new Random();
4         for (int i = 0; i < numerosK; i++) {
5             int numero = rnd.Next((IgnorarAire ? IGNORAR : -1000),
6                                   1400);
7             centros.Add((double)numero);
8         }
}
```

Se calcula la distancia con base a la diferencia de densidades.

```
1     public void GenerarDistancias(){
2         for (int i = 0; i < N; i++){
3             if (reporte_progreso.CancellationPending)
4                 return;
5             for (int j = 0; j < M; j++){
6                 if (IgnorarAire && datos[i][j] < IGNORAR)
7                     continue;
8                 for (int k = 0; k < numerosK; k++){
9                     double dist = datos[i][j] - centros[k];
10                    distancias[i, j, k] = dist * dist;
11                }
12            }
13        }
14    }
```

Se crea la matriz de pertenencia, dando un valor de pertenencia a cada densidad para cada cluster.

```
1     public void ActualizarPertenencia(){
2         for (int i = 0; i < N; i++){
3             operaciones_cargando++;
4             reporte_progreso.ReportProgress((operaciones_cargando *
5                 90) / operaciones_total);
6             if (reporte_progreso.CancellationPending)
7                 return;
8             for (int j = 0; j < M; j++){
9                 if (IgnorarAire && datos [i] [j] < IGNORAR)
10                     continue;
11                 for (int k = 0; k < numerosK; k++){
12                     double sum = 0.0;
13                     for (int l = 0; l < numerosK; l++)
14                         sum += Math.Pow(distancias [i , j , k] /
15                             distancias [i , j , l] , 2.0 / (m - 1.0));
16                     pertenencia [i , j , k] = 1.0 / sum;
17                 }
18             }
19         }
20     }
```

Se generan los nuevos centros para las iteraciones siguientes.

```
1     public void GeneraNuevosCentros(){
2         for (int k = 0; k < numerosK; k++){
3             long aa = 0;
4             long bb = 0;
5             for (int p = 0; p < N; p++){
6                 if (reporte_progreso.CancellationPending)
7                     return;
8                 for (int i = 0; i < M; i++){
9                     if (IgnorarAire && datos [p][i] < IGNORAR)
10                         continue;
11                     double valor = Math.Round(Math.Pow(pertenencia [
12                         p, i, k], m), 5);
13                     if (valor <= 0.00001)
14                         continue;
15                     aa += (long)(Math.Round(valor * datos[p][i], 5)
16                         * 100000));
17                     bb += (long)(valor * 100000);
18                 }
19                 if (bb <= 0.0001) {
20                     centros [k] = ObtenPixelRandom();
21                 } else {
22                     centros [k] = (double)aa / (double)bb;
23                 }
24             }
25 }
```

Apéndice C

Código Split and Merge

Esta función verifica para cada región si existe similitud en ella, si no se cumple la condición, las regiones se separan.

```
1     private void EvaluacionRegion(int y1, int x1, int y2, int x2) {  
2         if (y1 > y2 || x1 > x2)  
3             return;  
4         double promedio = EvaluacionPromedio(y1, x1, y2, x2);  
5         if (Desviacion(y1, x1, y2, x2, promedio)) {  
6             SeparaRegiones(y1, x1, y2, x2);  
7         } else {  
8             EstablecerValores(y1, x1, y2, x2, promedio);  
9         }  
10    }  
11 }
```

Esta función separa la imagen en 4 regiones para su evaluación.

```
1     private void SeparaRegiones(int y1, int x1, int y2, int x2) {  
2         int PMY = (y1 + y2) / 2;  
3         int PMX = (x1 + x2) / 2;  
4         EvaluacionRegion(y1, x1, PMY, PMX);  
5         EvaluacionRegion(y1, PMX + 1, PMY, x2);  
6         EvaluacionRegion(PMY + 1, x1, y2, PMX);  
7         EvaluacionRegion(PMY + 1, PMX + 1, y2, x2);  
8     }
```

Esta función junta las regiones separadas.

```
1  private void JuntaRegiones(int yy, int xx, int valor) {
2      Queue busqueda = new Queue();
3      busqueda.Enqueue(new Tuple<int, int>(yy, xx));
4      visitados [yy, xx] = 1;
5      while (busqueda.Count > 0) {
6          Tuple<int, int> actual = (Tuple<int, int>)busqueda.
7              Dequeue();
8          cuadrosTotales++;
9          sumaTotal += resultado [actual.Item1, actual.Item2];
10         for (int i = 0; i < 4; i++) {
11             int y = actual.Item1 + movimientos [i, 0];
12             int x = actual.Item2 + movimientos [i, 1];
13             if (DentroMapa(y, x)) {
14                 if (visitados [y, x] == 1 || Math.Abs(resultado
15                     [y, x] - resultado [actual.Item1, actual.
16                     Item2]) > EPS_merge)
17                     continue;
18                 visitados [y, x] = 1;
19                 busqueda.Enqueue(new Tuple<int, int>(y, x));
20             }
21         }
22     }
23 }
```

Apéndice D

Código Región creciente

Con esta función se genera la segmentación del objeto con base a una semilla dada por el usuario.

```
1     public int[,] ObtenerRegion(int calidad = 0) {
2         int[,] salida = new int[N, M];
3         Queue busqueda = new Queue();
4         busqueda.Enqueue(new Tuple<int, int>(origenY, origenX));
5         salida[origenY, origenX] = 1;
6         while(busqueda.Count > 0) {
7             Tuple<int, int> actual = (Tuple<int, int>) busqueda.
8                 Dequeue();
9             for(int i = 0; i < 8; i++){
10                 int y = actual.Item1 + movimientos[i, 0];
11                 int x = actual.Item2 + movimientos[i, 1];
12                 if(DentroMapa(y, x)) {
13                     if (salida[y, x] == 1 || Math.Abs(mapa[origenY,
14                         , origenX] - mapa[actual.Item1, actual.Item2
15                         ]) > precision[calidad])
16                         continue;
17                     salida[y, x] = 1;
18                     busqueda.Enqueue(new Tuple<int, int>(y, x));
19                 }
20             }
21         }
22         return salida;
23     }
```

Apéndice E

Código generación de contraste

Esta función genera un mapeo de los valores de gris y los valores de Hounsfield.

```
1     public void obtenerImagenConVentana(ParametroContraste o) {
2         int[,] matriz = o.matriz;
3         int limiteInferior = o.inicio;
4         int limiteSuperior = o.fin;
5         int N = matriz.GetLength(0);
6         int M = matriz.GetLength(1);
7         Bitmap imagen = new Bitmap(N, M);
8         int tam = limiteSuperior - limiteInferior + 1;
9         double porcion = 255.0 / tam;
10
11        for (int i = 0; i < N; i++) {
12            for (int j = 0; j < M; j++) {
13                int valorGris = (int)(porcion * (double)(matriz[i,
14                    j] - limiteInferior + 1));
15                if (matriz[i, j] < limiteInferior)
16                    valorGris = 0;
17                if (matriz[i, j] > limiteSuperior)
18                    valorGris = 255;
19                Color color = Color.FromArgb(valorGris, valorGris,
20                    valorGris);
21                imagen.SetPixel(i, j, color);
22            }
23        }
24        imagenes[o.indice] = imagen;
25    }
```

Apéndice F

Diagrama de clases

La siguiente imagen muestra las clases utilizadas en el sistema. El paradigma de la programación orientada a objetos nos ayudó a llevar un mejor control sobre los diferentes algoritmos y herramientas implementadas.

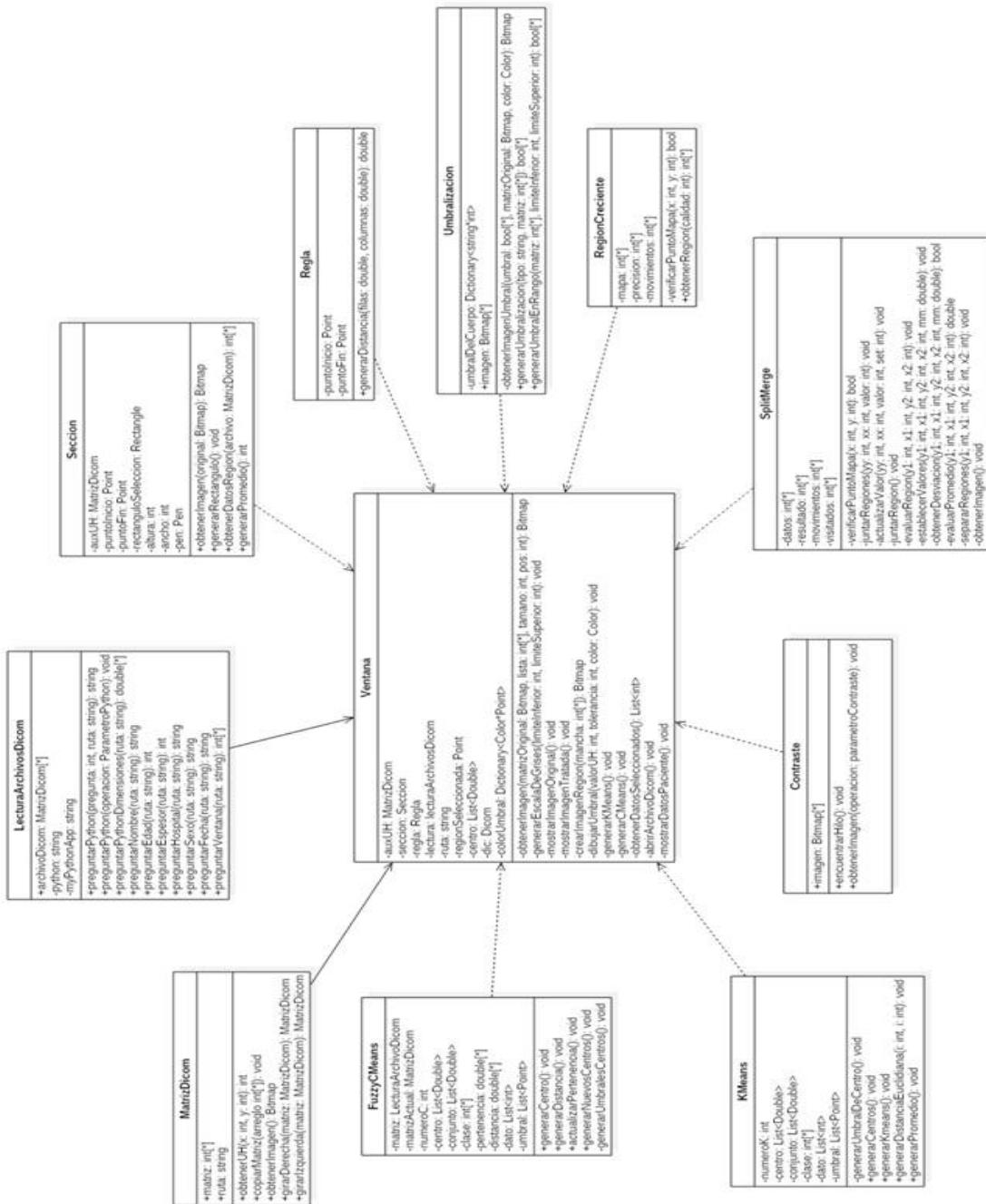


Figura F.1: Diagrama de clases del sistema.
Imagen de elaboración propia.

Bibliografía

- [1] P. A. Carrión, J. Ródenas, and J. J. Rieta, *Ingeniería biomédica. Imágenes médicas.* Ediciones de la Universidad de Castilla-La Mancha, 2006.
- [2] J. C. R. Giraldo, C. A. Clavijo, and C. McCollough, “Tomografia computarizada por rayos x; fundamentos y actualidad,” *Ingeniería Biomédica*, vol. 2, no. 4, p. 13, 2008.
- [3] K. Doi, “Computer-aided diagnosis in medical imaging: Historical review, current status and future potential,” *Computerized Medical Imaging and Graphics*, vol. 31, no. 4, p. 198, 2007.
- [4] NEMA, <http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>, 2016.
- [5] O. Pianykh, *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide.* Springer Berlin Heidelberg, 2009.
- [6] B. Imaging. (2014) Ct scanner price guide. [Online]. Available: <https://info.blockimaging.com/bid/84432/ct-scanner-price-guide>
- [7] Medwow. (2013) Scanner ct - toshiba - aquilion 32. [Online]. Available: <http://es.medwow.com/used-ct-scanner/toshiba/aquilion-32/716334511.item>
- [8] I. D. Imaging. (2017) Occiput.io. [Online]. Available: <https://idoimaging.com/programs/332>
- [9] ——. (2017) Niftyrec. [Online]. Available: <https://idoimaging.com/programs/331>
- [10] O. Viewer. (2014) Osirix hd user manual. [Online]. Available: <http://www.osirix-viewer.com/Manual/>

- [11] D. Laboratory. (2017) 3dimviewer (for windows and mac os x). [Online]. Available: <http://www.3dim-laboratory.cz/en/software/3dimviewer>
- [12] D. L. Phalm, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual Review of Biomedical Engineering*, vol. 2, p. 315, 2000.
- [13] PerkinELmer. (2016) Image analysis software. [Online]. Available: <http://www.perkinelmer.com/category/image-analysis-software>
- [14] A. S. Corp. (2016) 3d-doctor. [Online]. Available: <http://www.ablesw.com/3d-doctor/index.html>
- [15] E. F. Martínez, “Análisis digital de imágenes tomográficas sin contraste para la búsqueda de tumores cerebrales,” Master’s thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2011.
- [16] H. F. Mondragón and K. P. Hernández, “Reconstrucción tridimensional de estructuras internas del cuerpo humano a partir de tomografías axiales computarizadas,” Escuela Superior de Cómputo - Instituto Politécnico Nacional, México, Tech. Rep., 2012.
- [17] R. C. González and R. E. Woods, *Digital Image Processing*. Pearson, 2008.
- [18] J. Bushberg, *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2002.
- [19] E. Carver and B. Carver, *Medical imaging: Techniques, reflection and evaluation*. Churchill Livingstone, 2012.
- [20] C. E. Misch, *Prótesis Dental Sobre Implantes*. Elsevier, 2007.
- [21] D. N. C. Pereira, *Tomografía Axial Computada*. Universidad de la República Oriental del Uruguay, 2004.
- [22] J. Jiménez, *Fundamentos Morfológicos de la Tomografía Axial Computarizada Craneo - Encefálica*. Universidad de Sevilla, 1981.
- [23] O. S. Pianykh, *Digital imaging and communication in medicine(DICOM) a practical introduction and survival guide*. Springer Science and Business Media, 2009.
- [24] C. E. Misch, *Implantología contemporánea*. Elsevier, 2009.

- [25] M. Hofer, *Manual práctico de TC*. Médica Panamericana, 2005.
- [26] I. N. Manousakas, P. E. Undrill, G. G. Cameron, and T. W. Redpath, “Split-and-merge segmentation of magnetic resonance medical images: Performance evaluation and extension to three dimensions,” *Computer and biomedical research*, vol. 31, p. 393, 1998.
- [27] N. Dhanachandra, K. Manglem, and Y. J. Chanu, “Image segmentation usin k-means clustering algorithm and substractive clustering algorithm,” *Procedia Computer Science*, vol. 54, p. 764, 2015.
- [28] S.Tatiraju and A. Mehta, “Image segmentation using k-means clustering, em and normalized cuts.”
- [29] A. S. Tanenbaum, *Sistemas operativos Modernos*. Pearson, 2009.
- [30] J. Levin, *Mac OS X and iOS internals: To the Apple’s Core*. Wrox, 2012.
- [31] B. Eckel, *Piensa en Java*. Prentice Hall, 2007.
- [32] H. M. Deitel and P. J. Deitel, *Cómo programar en C/C++ y Java*. Pearson, 2004.
- [33] A. B. Downey, *Think Python: How to think like a computer scientist*. O'Reilly, 2016.
- [34] P. I. (2009) Ventajas del csharp y desventajas. [Online]. Available: <http://programacion1abundiz.blogspot.mx/2009/09/ventajas-del-c-y-desventajas.html>
- [35] PYDICOM. (2012) Pydicom documentation. [Online]. Available: <http://pydicom.readthedocs.io/en/stable>
- [36] dcm4che.org. dcm4che.org wiki. [Online]. Available: <http://dcm4che.atlassian.net/wiki/display/proj/dcm4che.org+Wiki>
- [37] I. Sommerville, *Software Engineering*. Pearson, 2006.
- [38] T. Northrup, *Windows 8.1 Inside Out*. Microsoft Press, 2015.
- [39] Microsoft. (2017) Características disponibles solo en windows 10. [Online]. Available: <https://www.microsoft.com/es-mx/windows/features>

- [40] Microsoft. (2017) Visual studio. [Online]. Available: <http://visualstudio.com/es>
- [41] P. Brachet. (2015) Texmaker: The universal latex editor. [Online]. Available: <http://www.xm1math.net/texmaker/>
- [42] G. Team. Gnu image manipulation program. [Online]. Available: <https://www.gimp.org/>
- [43] MKLab. (2016) Staruml. [Online]. Available: <http://staruml.io/>
- [44] D. X. M. Triadó, “El pert: un método eficaz para la planificación de actividades,” *UNIVERSITAT DE BARCELONA*, vol. 1, pp. 2–5, 2012.
- [45] I. Sommerville, *Ingenería de software*. Pearson, 2011.