



FACULTAD 4

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

## **Herramienta para comparación de estudios sobre imágenes DICOM de Resonancia Magnética.**

**Autores:**

Lester Jesús Alzugaray Prado  
José Ramón Fidalgo García

**Tutor:**

Dr.C Arturo Orellana García



“La sinergia entre la informática y la medicina redefine el paradigma de la atención médica, permitiendo diagnósticos más precisos y tratamientos personalizados.”

Dr. Atul Butte, bioinformático y pediatra.

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo de diploma “Herramienta para comparación de estudios sobre imágenes DICOM de Resonancia Magnética”, concedemos a la Universidad de las Ciencias Informáticas y en especial al Centro de Informática Médica la autorización a hacer uso del mismo en su beneficio.

Para que conste firmamos el presente documento a los 27 días del mes de Noviembre del año 2023

Lester Jesús Alzugaray Prado



Firma del Autor

José Ramón Fidalgo García



Firma del Autor

Dr.C. Arturo Orellana García



Firma del Tutor

**DATOS DE CONTACTO.**

**Dr.C. Arturo Orellana García:** graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Se desempeña como Líder del Grupo de Investigación de Minería de procesos y Asesor de Capacitación, Desarrollo e Investigación del Centro de Soluciones de Informática Médica. Ha liderado proyectos I+D+i de desarrollo de componentes de software a partir de minería de procesos para el análisis de procesos de negocio del entorno hospitalario. Investiga la Ingeniería de comportamiento, la medicina de precisión y el procesamiento de imágenes médicas. Tutora varias tesis de grado, maestrías y doctorados enfocados al análisis de procesos de negocio, la informática médica y otras áreas del conocimiento. Obtuvo el grado de Máster en Informática Aplicada en 2015 desarrollando una herramienta informática basada en técnicas de minería de procesos para identificar problemas en la ejecución de procesos de negocio. Doctor en Ciencias Técnicas desde 2016 presentando un modelo computacional para la detección de variabilidad en procesos de negocio del entorno sanitario aplicando minería de procesos.

Correo electrónico: [aorellana@uci.cu](mailto:aorellana@uci.cu)

**Lester Jesús Alzugaray Prado:** Estudiante de 5to año facultad 4

Correo electrónico: [lesterjap@estudiantes.uci.cu](mailto:lesterjap@estudiantes.uci.cu)

**José Ramón Fidalgo García:** Estudiante de 5to año facultad 4

Correo electrónico: [joserfg@estudiantes.uci.cu](mailto:joserfg@estudiantes.uci.cu)

## AGRADECIMIENTOS

### **Lester Jesús Alzugaray Prado**

Agradezco sinceramente a los profesores de esta institución, cuyo dedicado esfuerzo ha sido fundamental para mi formación académica. Su guía y conocimientos compartidos han contribuido significativamente al desarrollo de esta tesis. A mis compañeros de universidad a los cuales conocí desde los inicios de este proceso. A esos amigos de apartamento que estuvieron presentes en las buenas y a las malas; aquellos que están presentes aquí y a los que estén lejos también.

Mi profundo agradecimiento se dirige especialmente a mi tutor Arturo Orellana, quien no solo me brindó valiosa experiencia sino que también dedicó tiempo y esfuerzo en revisar y orientar cada aspecto de esta investigación. Su asesoría ha sido invaluable y ha dejado una marca indeleble en mi desarrollo académico.

Quisiera expresar mi gratitud a mi querida novia, cuyo apoyo incondicional ha sido mi fuente de inspiración a lo largo de todo el proceso de elaboración de esta tesis. Su paciencia, aliento y comprensión han sido pilares fundamentales que me han permitido superar los desafíos y mantenerme enfocado en mi objetivo.

A mis padres, Magalys y Ramón quiero dedicarles un agradecimiento especial. Su constante presencia, amor y apoyo han sido mi mayor fortaleza. Gracias por ser mis guías, por alentarme en cada paso y por creer en mí incluso cuando yo dudaba. Este logro no habría sido posible sin su inquebrantable respaldo “Gracias mami y papi”.

A todos ustedes, mi más sincero agradecimiento. Este trabajo no solo es el resultado de mi esfuerzo individual, sino también de la contribución invaluable de cada uno de ustedes.

### **José Ramón Fidalgo García**

Agradezco profundamente a mis padres mis hermanos y mi novia por su amor incondicional y apoyo constante a lo largo de este desafiante viaje académico.

Mi más sincero agradecimiento a mi tutor Arturo Orellana, cuya orientación experta y paciencia fueron fundamentales para el desarrollo y éxito de esta tesis.

Quisiera expresar mi gratitud a mis amigos Juan, Pedro, quien siempre se ofrecieron a una partida más aunque solo fuera para salir del estrés de las largas horas de trabajo tanto en las tareas escolares como en la investigación de la tesis. Gracias por aún después de graduarse siguieron jugando una partidas más.

A la institución, agradezco por proporcionar los recursos necesarios que hicieron posible la realización de este trabajo.

Finalmente, a todos aquellos que de alguna manera contribuyeron a este proyecto, mi más sincero agradecimiento por su apoyo y colaboración.

## RESUMEN

Mediante la incursión de la informática moderna en los estudios médicos surge una nueva rama conocida como neuroinformática, encargada de dirigir las investigaciones basadas en imágenes del cerebro humano. Pero no ha bastado con solo la obtención y el procesamiento de dichas imágenes pues se necesita contar con aplicaciones encargadas de la visualización y las mediciones de algunas neuroestructuras para elaborar diagnósticos de distintas condiciones patológicas. El proyecto XAVIA-PACS en conjunto con CNEURO se encarga del procesamiento de estudio de resonancias magnéticas de sus pacientes, surgiendo la necesidad de contar con una herramienta que sea capaz de realizar comparaciones de estudios de una misma persona en distintos años para comprobar su evolución. En la presente investigación se lleva a cabo el proceso de desarrollo de la herramienta de comparación a partir de la fundamentación teórica para su desarrollo, donde se comprenden los principales elementos a tomar de los archivos en formato DICOM obtenidos de resonancias magnéticas. Se seleccionó la metodología AUP-UCI en su escenario 3, se compone de la descripción del modelo de negocio más el modelo conceptual y utiliza el análisis de desarrollo la descripción de requisitos por proceso. La arquitectura seleccionada fue Modelo-Vista-Controlador (MVC) pues este posibilita separar sus componentes según su lógica para facilitar el mantenimiento y escalabilidad de la solución informática. Las técnicas de Caja Negra y Caja blanca aplicadas para la detección de errores mostraron resultados positivos pues se detectan pocas fallas en el funcionamiento las cuales fueron arregladas.

**Palabras clave:** Algoritmos, DICOM, Medición, Resonancias Magnéticas.

## ABSTRACT

Through the incursion of modern computing into medical studies, a new branch known as neuroinformatics arises, responsible for directing research based on images of the human brain. But it has not been enough to simply obtain and process these images since it is necessary to have applications responsible for the visualization and measurements of some neurostructures to make diagnoses of different pathological conditions. The XAVIA-PACS project, together with CNEURO, is responsible for processing the MRI studies of its patients, resulting in the need to have a component that is capable of making comparisons of studies of the same person in different years to check their evolution. In the present investigation, the development process of said component is carried out based on the theoretical foundation for its development, where the main elements to be taken from the DICOM format files obtained from magnetic resonance imaging are understood. The AUP-UCI methodology was selected in its scenario 3, it is composed of the description of the business model plus the conceptual model and uses development analysis to describe requirements by process. The selected architecture was Model-View-Controller (MVC) since this makes it possible to separate its components according to their logic to facilitate the maintenance and scalability of the IT solution. The Black Box and White Box techniques applied to detect errors showed positive results since few malfunctions were detected and were fixed.

**Keywords:** Algorithms, DICOM, Measurement, Magnetic Resonances,

# ÍNDICE

INTRODUCCIÓN.....	1
1. Fundamentación teórica de la investigación .....	6
1.1. Resonancia magnética .....	6
1.2. Estándar DICOM.....	7
1.3. Análisis y comparación de elementos homólogos al problema .....	8
1.4. Algoritmos de calibración de imágenes DICOM para realizar mediciones.....	11
1.4.1 Algoritmo de calibración de imágenes DICOM basada en objetos.....	11
1.4.2. Algoritmos de calibración de imágenes DICOM basada en geometría.....	12
1.4.3. Comparación de los 2 algoritmos a partir de sus ventajas y desventajas.....	13
1.4.4. Algoritmos para la medición de imágenes médicas.....	14
1.5. Metodologías, tecnologías y herramientas.....	24
1.5.1 Ambiente de Desarrollo .....	25
1.6. Conclusiones del capítulo .....	26
2. PROPUESTA DE SOLUCIÓN.....	27
2.1. Descripción de la propuesta de solución .....	27
2.1. Modelo de Dominio .....	28
2.2. Descripción de conceptos en el modelo de dominio.....	29
2.3. Descripción del proceso de negocio.....	30
2.4. Modelado de negocio .....	31
2.5. Especificación de Requisitos .....	32
2.5.1. Requisitos funcionales .....	33
2.5.2. Requisitos no funcionales .....	38
2.6. Definición de los actores.....	39
2.7. Análisis y diseño .....	39
2.7.1. Arquitectura de software.....	39
2.7.2. Descripción de la arquitectura.....	40
2.7.3. Diagrama de clases.....	41
2.7.4. Diagrama de secuencia.....	41
2.7.5. Diagrama de componentes.....	45

Conclusiones del capítulo .....	45
3. IMPLEMENTACIÓN Y PRUEBAS.....	47
3.1. Estándar de Codificación .....	47
3.2. Implementación. ....	47
3.2. Pruebas Funcionales. ....	50
3.2.1. Pruebas de Caja Negra. ....	50
3.2.2. Diseño de Casos de Pruebas.....	51
3.3.1. Prueba del camino básico. ....	52
3.4. Validación requisitos no funcionales.....	53
Validación RNF1. Eficiencia del componente. ....	54
Validación RNF3. Apariencia o Interfaz del componente.....	55
3.5. Comparación con herramientas homólogas en cuanto a precisión .....	55
Conclusiones del capítulo .....	59
Conclusiones: .....	60
Recomendaciones.....	61
REFERENCIAS .....	62

**Índice de Tablas**

Tabla 1 Comparación de estudios Homólogos. Fuente: "Elaboración propia"..... 10

Tabla 2 - Comparación de los algoritmos de calibración para imágenes DICOM..... 13

Tabla 3 - Comparación de los algoritmos analizados. Fuente: Elaboración propia ..... 22

Tabla 4 - Descripción del proceso de negocio. Fuente: Elaboración propia ..... 30

Tabla 5 - Requisitos funcionales ..... 33

Tabla 6 - Descripción del requisito funcional "Importar Imágenes". ..... 36

Tabla 7 - Descripción del requisito funcional "Cargar varias pantallas de imágenes a la vez" ..... 37

Tabla 8 - Descripción del requisito funcional "Sintonizar imágenes"..... 37

Tabla 9 - Diseño de casos de pruebas..... 51

Tabla 10 – Validación del requisito no funcional de eficiencia. .... 54

Tabla 11 - Comparación de precisión de mediciones entre la herramienta desarrollada y Horos ..... 56

Tabla 12 - Comparación de mediciones entre las herramientas ..... 57

**Índice de Figuras**

Figura 1 - Estructura de un Archivo DICOM..... 8

Figura 2 - Espacio Cartesiano para calcular la distancia de un vértice a otro. Fuente: Elaboración propia 15

Figura 3 - Algoritmo de Detección de Puntos de Referencia ..... 16

Figura 4 - Representación del espacio cartesiano y representación del espacio paramétrico. ( Santini & Santini, 2020.) ..... 17

Figura 5 - Implementación Algoritmo Transformada de Hough..... 17

Figura 6 - Proceso de imágenes con Algoritmo Transformada de Hough. Fuente: Elaboración propia ..... 18

Figura 7 - Representación de los valores de píxeles (NIÑO NAVARRETE, 2019)..... 20

Figura 8 - Implementación Algoritmo de Canny. Fuente: Elaboración propia ..... 20

Figura 9 - Proceso de detección de bordes del Algoritmo de Canny. Fuente Elaboración propia..... 21

Figura 10 - Comparación de los algoritmos analizados según precisión. Fuente: Elaboración propia ..... 22

Figura 11- Modelo de dominio para la comparación de varios estudios. Fuente: Elaboración propia. .... 29

Figura 12 - Diagrama del proceso de negocio para hacer comparaciones. Fuente: Elaboración propia..... 32

Figura 13.- Modelo de arquitectura a utilizar. Fuente: Elaboración propia..... 40

Figura 14 - Diagrama de clases del sistema. Fuente: Elaboración propia ..... 41

Figura 15 - Diagrama de secuencia requisito “trazar líneas de medición”. Fuente: Elaboración Propia..... 42

Figura 16 - Diagrama de secuencia requisito “trazar área de medición circular”. Fuente: Elaboración propia. .... 43

Figura 17- Diagrama de secuencia del requisito "importar imágenes". Fuente: Elaboración propia. .... 44

Figura 18- Diagrama de componentes para la herramienta de comparación de estudios. Fuente: Elaboración Propia. .... 45

Figura 19 - Fragmento de código "Trazar línea de medición". Fuente: Elaboración propia..... 48

Figura 20 - Fragmento del código "Replicar mediciones". Fuente: Elaboración propia ..... 49

Figura 21- Fragmento de código "numeroVentanas". Fuente: Elaboración propia ..... 49

Figura 22- Fragmento de código "colocarImágenes". Fuente: Elaboración propia ..... 50

Figura 23- Resultados de los casos de prueba ..... 52

Figura 24- Prueba del camino básico a la funcionalidad modoPintura(). Fuente: Elaboración propia..... 53

Figura 25- Fragmento de código referente a la prueba de tiempo de ejecución. Fuente: Elaboración propia. .... 54

Figura 26- Ventana principal de la interfaz del herramienta para comparación. Fuente: Elaboración propia ..... 55

Figura 27 - Grafico de dispersión obtenido por técnica de Bland-Alman. Fuente: Elaboración propia ..... 58

# INTRODUCCIÓN

La neurociencia comprende una amplia gama de interrogantes acerca de cómo se organiza el sistema nervioso y cómo funciona para generar la conducta. Estos cuestionarios pueden explorarse por medio de las herramientas analíticas de la genética, la biología molecular y celular, la anatomía y la fisiología de los sistemas, la biología conductual y la psicología. (1)

La localización de las funciones cognitivas en el cerebro tiene una larga historia, que se remonta por lo menos a los primeros años del siglo XIX, con la fundación de la frenología. Esta ciencia constituyó una especie de “psicología de las facultades”, empeñada en atribuir características mentales específicas a las diferentes regiones del cerebro, hasta 27 distintas en esa época, en primer lugar, se enfatiza por primera vez la importancia de la superficie cerebral, el córtex. En segundo lugar, se postula que el cerebro no era un órgano unitario e indiferenciado, abriendo la puerta a un estudio más sistemático y, sobre todo, caracterizado por los medios científicos como centro de las funciones cognitivas del cuerpo.(2)

El cambio al siglo XX supuso la eclosión de los estudios neuroanatómicos y neurofisiológicos que fueron perfilando el surgimiento de la Neurociencia, en los años 60, como disciplina de síntesis. En la década de 1990 sucedieron los estudios citoarquitectónicos del córtex, que pusieron de manifiesto la existencia de una compleja organización anatómica, la identificación de varias decenas de áreas compuestas por células nerviosas de morfología diferenciada. Sin embargo, más adelante, algunos científicos cuestionaban que las diferencias morfológicas implicaban diferencias funcionales. A raíz de estos estudios los psicólogos cognitivos empezaban a dirigir sus miradas hacia la neurociencia y la neurociencia empezaba a interesarse más por el tipo de problemas que la psicología se planteaba. La necesidad de una nueva forma de entender las relaciones entre el cerebro y la mente se hacía palpable.(2)

El desarrollo de las TIC ha permitido profundizar en la exploración y el desarrollo de diferentes áreas de las ciencias y otros ámbitos en plena transformación brindando métodos, formas y herramientas para la aplicación de estos descubrimientos. Los programas informáticos se han convertido en una herramienta fundamental en el estudio de un órgano tan complejo como el cerebro y dieron paso al surgimiento de la neuroinformática como uno de los campos más efervescentes y que siguen brindando nuevos descubrimientos. (3)

Los nuevos estudios y descubrimientos en este campo se apoyan de las técnicas

computacionales para obtención y manejo de datos cerebrales. Entre estos datos tienen un impacto mayor los datos de neuroimagen, obtenidos a través de Resonancias Magnéticas y Tomografías Digitales, de ahí la importancia de proponer soluciones sobre el manejo, gestión y procesamiento de datos de neurociencia. Para estos datos poder ser visualizados son convertidos a un formato estándar para su visualización, siendo el caso del estándar en Imagenología Digital y Comunicaciones Médicas (DICOM). El formato DICOM ha sido impulsado por las organizaciones Asociación de Radiólogos Americanos (ACR) y Asociación Nacional de Empresas Eléctricas de Estados Unidos (NEMA) como un estándar para la transmisión de información y almacenamiento de imágenes médicas. Este estándar de archivos está formulado por numerosos elementos, cada cual compuesto por una etiqueta que lo identifica, un valor de representación que indica el tipo de dato almacenado, un campo que indica la longitud del dato y el dato propiamente dicho; este último proporciona información del paciente, del análisis realizado o de la imagen obtenida, dependiendo del valor de la etiqueta. En el archivo los elementos se escriben en orden ascendente de acuerdo al valor de la etiqueta que los identifica.(4)

La resonancia magnética (RM) se ha establecido como una herramienta muy valiosa en el diagnóstico e investigación de muchas áreas en la medicina gracias a su gran capacidad de proveer excelente caracterización y diferenciación de tejidos blandos de múltiples áreas del cuerpo.(5). Por tanto, las resonancias magnéticas son unas de las técnicas más confiables de obtener imágenes cerebrales de pacientes para su posterior análisis.

CNEURO es una institución cubana de investigación y desarrollo dedicado a la investigación traslacional y otras tecnologías médicas, que abarca desde la investigación básica hasta el desarrollo, producción y comercialización de tecnologías y muestra interés en la utilización de software nacional dedicado a la medicina. Entre las investigaciones más destacadas de esta organización se encuentran los estudios de resonancia magnética aplicado a distintos escenarios, entre ellos los diagnósticos y seguimiento de tumores cerebrales, mal formaciones congénitas y afecciones mediante factores externos como contusiones.(6)

Esta entidad lleva a cabo investigaciones en una amplia gama de temas que incluyen neurociencia cognitiva, neuroinformática, neuroimagen funcional, análisis de señal bioeléctrica, modelación matemática, investigación neuroquímica, genética molecular e impresión 3D para dispositivos médicos.(7)

En la Universidad de las Ciencias Informáticas se encuentra en continuo desarrollo

desde su creación del sistema para el almacenamiento, transmisión y visualización de imágenes médicas XAVIA PACS-RIS. Ha sido implantado en 307 estaciones de trabajo con 27 XAVIA PACSServer y 34 servidores de bases de datos. El XAVIA PACSWeb está presente en 20 instituciones, mientras que el XAVIA RIS es utilizado en 15. La Habana constituye la provincia con mayor nivel de introducción del resultado con presencia en 18 instituciones, seguida por Pinar del Río y Villa Clara en tres instituciones; por su parte, Matanzas lo utiliza en dos instituciones.(8)

En el Centro de Neurociencias de Cuba para hacer los análisis longitudinales de los estudios de resonancia magnética se utiliza la herramienta eFilm. Esta es una herramienta privativa la cual los especialistas de CNEURO deben piratear para acceder a sus funcionalidades y algunas de manera incompleta limitando el rango de análisis que pueden generar. En otro sentido en Cuba está aprobado que se utilicen dos herramientas para el diagnóstico por imágenes que son el caso del XAVIA PACS en la Universidad de las Ciencias Informáticas y el Imagis en Santiago de Cuba. En el caso del XAVIA PACS se cuenta con el código fuente, el equipo de desarrollo que permite la sostenibilidad del sistema y es autorizado para ser desplegado en el Occidente y Centro del País. Sin embargo no posee funcionalidades que permitan la comparación de estudios limitando su rango de acción a este tipo de problemáticas.

Por lo antes planteado se identifica como **problema a resolver**: ¿Cómo propiciar el análisis longitudinal de patologías en imágenes de resonancia magnética del cerebro?

Este problema se enmarca en el **objeto de estudio**: el proceso de comparación de estudios longitudinales del cerebro.

Para dar solución al problema se define como **objetivo**: desarrollar una herramienta para comparación de estudios sobre imágenes DICOM de resonancia magnética.

El objeto delimita el **campo de acción**: comparación de estudios del cerebro desde imágenes de resonancia magnética.

Para guiar la investigación se definen las siguientes **tareas a cumplir por los estudiantes**:

1. Elaboración del Marco teórico de la investigación necesario en el desarrollo de la herramienta para la comparación sobre imágenes de resonancia magnética

en formato DICOM.

2. Análisis de herramientas para el manejo y comparación de estudios sobre imágenes de resonancia magnética.
3. Análisis y comparación de los algoritmos utilizados para mediciones sobre imágenes de resonancia magnética en formato DICOM.
4. Validación y pruebas de la herramienta para la comparación de estudios sobre imágenes de resonancia magnética en formato DICOM.

Los métodos científicos utilizados en esta investigación son:

- 1) **Histórico Lógico:** Analizar las bibliografías existentes referentes a la visualización y comparación de estudios sobre imágenes de resonancia magnética en formato DICOM. La investigación se inicia con un estudio del estado del arte de la problemática planteada y los requisitos necesarios para la herramienta a desarrollar. Además se realiza un análisis de los algoritmos utilizados para la medición en imágenes de resonancia magnética.
- 2) **Analítico Sintético:** para descomponer en elementos más concretos el problema de investigación. Analizar las herramientas homólogas, los algoritmos de calibración de imágenes médicas, algoritmos de medición y realizar la propuesta de solución.
- 3) **Inductivo Deductivo:** A partir del proceso para la comparación de las estructuras en imágenes DICOM y las estadísticas más utilizadas en las mismas se evalúa la problemática existente, para determinar los aspectos particulares y desarrollar una propuesta de solución para esta.
- 4) **Modelación:** para confeccionar modelos y diagramas exigidos en la metodología para la comprensión de los procesos a desarrollar como parte de la investigación.
- 5) **Experimento:** para la validación de los resultados obtenidos por la herramienta para la comparación de estudios sobre imágenes de resonancia magnética en formato DICOM.

Resultados esperados:

Obtener una herramienta informática para realizar la comparación de estudios sobre imágenes DICOM el cual permita realizar mediciones precisas de áreas u objetos en estas imágenes junto con la posibilidad de cargar varios estudios a la vez de un mismo paciente y replicar las ediciones en toda la serie de imágenes.

**Estructura del contenido**

### **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN**

Este capítulo contiene los principales conceptos asociados a la investigación. Aborda un análisis de herramientas y componentes homólogos como soluciones a la visualización y comparación de imágenes médicas de resonancia magnética. Como parte de la revisión bibliográfica se detallan los pasos para realizar proceso de comparación de estudios sobre imágenes obtenidas a través de resonancia magnéticas y llevadas al estándar DICOM. Se define la metodología a utilizar para el diseño y análisis del componente. Se escogen las herramientas que se utilizarán para representar el diseño e implementar la solución informática.

### **CAPÍTULOS 2: PROPUESTA DE SOLUCIÓN**

En este capítulo se aborda el diseño del componente de comparación de estudios sobre imágenes DICOM de resonancia magnética. Se realizan y detallan los requisitos necesarios para cada una de las funciones del componente y los diagramas necesarios para el entendimiento del desarrollo de este. Se analiza y selecciona el modelo arquitectónico para el desarrollo de la herramienta. Se generan los artefactos para el elaborar el diseño del software.

### **CAPÍTULO 3: VALIDACIÓN Y PRUEBAS DEL COMPONENTE DE COMPARACIÓN DE ESTUDIOS SOBRE IMÁGENES DICOM**

En este capítulo se realiza la implementación de la solución informática propuesta, en el cual se toma en cuenta las principales herramientas y librerías necesarias para garantizar el funcionamiento de cada una de las partes del software. Como parte esencial en el desarrollo de cualquier software es necesario validar su correcto funcionamiento, el mismo se realizará aplicando las pruebas funcionales en cada uno de los requisitos del mismo. Se utilizarán las técnicas de caja negra y caja blanca para las pruebas.

### **1. Fundamentación teórica de la investigación**

Al ser realizadas las resonancias magnéticas se generan imágenes bidimensionales del paciente, estas se almacenan en formato dcm perteneciente al estándar DICOM. Este estándar permite organizar y guardar distintas informaciones sobre los estudios de un paciente. Siendo importante la posibilidad de visualizar y comparar estos estudios. A partir de la metodología AUP-UCI en su escenario 3 se describen los requisitos funcionales por proceso para un mejor entendimiento. Se analizan las herramientas homólogas, se seleccionan el lenguaje de programación y las tecnologías a utilizar para el desarrollo de la solución informática.

Este capítulo contiene los principales conceptos asociados a la investigación. Aborda un análisis de herramientas y componentes homólogos como soluciones a mostrar y comparar imágenes médicas de resonancia magnética. Como parte de la revisión bibliográfica se detallan los pasos para realizar proceso para la comparación de estudios sobre imágenes obtenidas a través de resonancia magnéticas y llevadas al estándar DICOM. Además de definir la metodología más adecuada para la descripción y entendimiento de las funcionalidades necesarias. Se seleccionan las herramientas para el diseño e implementación que sean aplicables y útiles a la solución informática propuesta para el Centro de Neurociencias de Cuba y el Sistema XAVIA-PACS de la UCI.

#### **1.1. Resonancia magnética**

La resonancia magnética es un examen auxiliar imagenológico que brinda la posibilidad de realizar un análisis más exhaustivo sobre las patologías de un paciente brindando un mejor diagnóstico de la misma, teniendo como principal característica la capacidad de obtener con mayor nitidez imágenes de tejido blando a diferencia de otros métodos imagenológicos. Tomando esta ventaja en cuenta la resonancia magnética puede contribuir de forma decisiva en la identificación de las características morfológicas de las diversas patologías que afectan a todo tipo de paciente.(9, 10)

La resonancia magnética se fundamenta en la absorción de energía por un núcleo magnéticamente activo, que debe tener dos características fundamentales: un número par de protones y un espín; el núcleo más utilizado es el hidrógeno por su presencia abundante en el cuerpo humano.(11) En las resonancias magnéticas todas las imágenes obtenidas son producidas usando una secuencia de pulso las cuales son

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

almacenadas en el scanner de la computadora. Las secuencias más utilizadas son T1 y T2 (Tiempos de relajación). La primera secuencia se emplea para evaluar las estructuras anatómicas normales, mientras que la segunda secuencia en T2 permite detectar la presencia de alguna patología(12). En el caso de T1 este constituye el tiempo de magnetización longitudinal que es el intervalo de tiempo en el cual la magnetización longitudinal recupera el 63% de su estado de equilibrio. La grasa presenta un T1 corto y es hiperintensa (brillante) en las imágenes de resonancia magnética ponderadas en T1(13). El tiempo de magnetización transversal o T2 es el tiempo en el cual la magnetización transversal desciende en un 63% de su fuerza máxima. El agua pura y otros líquidos son hiperintensos (brillantes) mientras que la grasa se muestra hipertensa en las imágenes ponderadas en T2(13).

A partir de realizados exámenes se pueden ver claramente cada una de las estructuras del cerebro del paciente. Estas imágenes se obtiene en formato MAYO/Analyze que utiliza NIFTI (Neuroimaging Informatics Technology Initiative) para añadir nuevos campos de información a la cabecera de las imágenes lo cual permite almacenarlas en un solo archivo. Se convierten y estandarizan al formato DICOM o .dcm a fin de tener un formato uniforme en todos los cortes cerebrales obtenidos durante el proceso. La correcta utilización de los espectros de color aplicados mediante la obtención de la imagen garantizan la correcta visualización de la misma.

### **1.2. Estándar DICOM**

El estándar DICOM define protocolos de comunicación de red, almacenamiento e intercambio de datos fuera de línea, protocolos semánticos para el intercambio de datos a niveles de conformidad. Estos estándares especifican protocolos de comunicación para dispositivos que reclaman conformidad. Además, una sintaxis y semántica de la información que se puede intercambiar en cada uno de sus protocolos. (14)

Para introducirse en el complejo entorno de la medicina, DICOM usa su propio lenguaje, basado en su modelo propio del mundo real. Se interpreta como todos los datos físicos o descriptivos como por ejemplo el nombre del paciente, el tipo de estudio, el dispositivo médico, los parámetros de la adquisición, la imagen digital, entre otros que son vistos por DICOM como elementos con sus respectivos atributos y propiedades. De esta forma se establece una jerarquía entre los datos que permite realizar una clasificación según el contenido de la información por grupos, facilitando la identificación, el acceso a las variables y los parámetros de interés dentro de un mismo archivo.(15, 16)

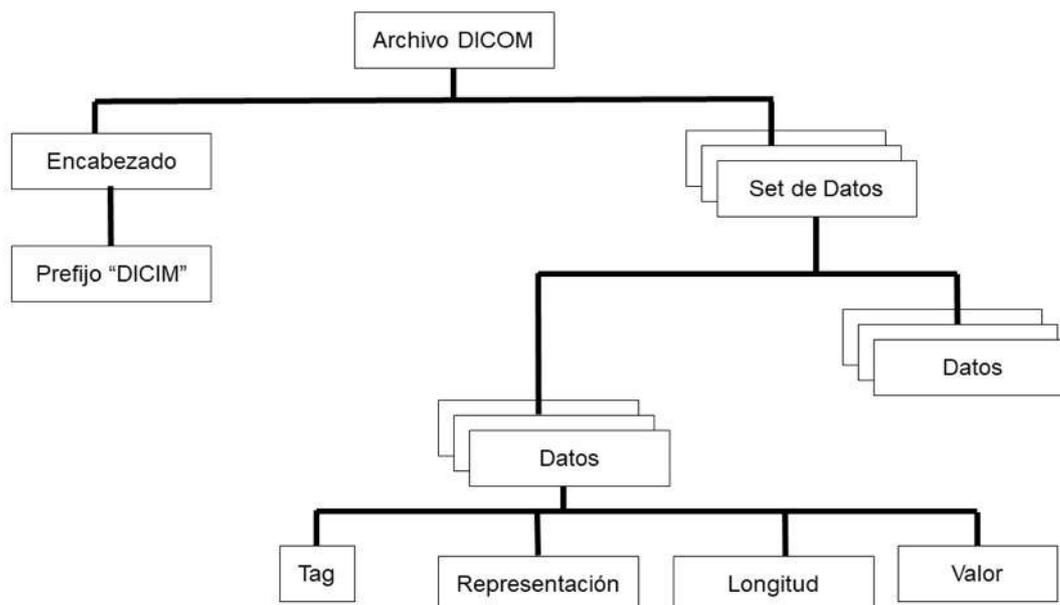


Figura 1 - Estructura de un Archivo DICOM.

Por lo general un archivo DICOM es reconocible por su extensión \*.dcm, sin embargo, esto no es una exigencia del Estándar, por lo que la forma de diferenciarlo es por medio de su encabezado (HEADER) que consta de 128 bytes de archivos de preámbulo y 4 bytes de prefijo "DICM". El preámbulo puede estar en blanco o contener información sobre la aplicación principal con la que debe ser ejecutado.(16)

Cada una de las estructuras de estos archivos se pueden tratar de manera independiente mediante el apoyo de librerías como pydicom para el tratamiento de estos, enfocada principalmente a la visualización o modificación en cualquiera de las series. Estas se pueden tratar de manera independiente lo cual propicia poder sincronizar dos estudios distintos de un mismo paciente en la misma serie, a este punto se realizan las observaciones previas y se replican en el resto de las imágenes. A partir de esta estructura resulta cómodo el trabajo con estos datos asegurando la accesibilidad a la información contenida en las series. Dado que el Centro de Neurodatos de Cuba y el sistema XAVIA-PACS trabaja las resonancias magnéticas y radiografías en el estándar DICOM se busca que exista la homogenización en el manejo de los mismos.

### 1.3. Análisis y comparación de elementos homólogos al problema

Para el desarrollo del software se realiza un análisis de herramientas similares u homólogas con el fin de tener una noción sobre las principales funciones con las que cuentan estas para la visualización y comparación de los estudios en imágenes en

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

formato DICOM obtenidas mediante resonancia magnética. Se toma como principal criterio la capacidad de cargar estudios completos de un paciente y las realización de las mediciones en distintas áreas necesarios para la emisión de un diagnóstico por parte del especialista. Este análisis cumple con la finalidad de establecer las funcionalidades comunes con los que cuentan los principales softwares que actualmente son más competentes en el mercado, a parte también encontrar las posibles funcionalidades a realizar en cada uno de los requisitos para poder contar con el resultado esperado en el desarrollo.

**OsiriX:** Es una herramienta de visualización y procesamiento de imágenes médicas basada en formato DICOM desarrollada específicamente para ser utilizada en entornos clínicos y de investigación. Esta permite la visualización de imágenes médicas en 2D y 3D, lo cual proporciona una presentación detallada de las estructuras anatómicas. Ofrece herramientas para realizar segmentación que permiten la delineación de estructuras específicas en las imágenes médicas. Permite realizar mediciones y cálculos de área en estructuras cerebrales. Es capaz de integrarse con otras tecnologías y trabaja con la modalidad de imágenes de resonancia magnéticas. Se integra a entornos PACS para acceder y gestionar imágenes médicas desde diferentes fuentes. Se encuentra disponible para el sistema operativo macOS contando con una versión básica gratuita y una comercial con características adicionales y soporte técnico.(17, 18)

**RadiAnt DICOM Viewer:** Es un software de visualización de imágenes médicas en formato DICOM. Es aplicable en el campo de la radiología y otras disciplinas médicas para visualizar y analizar imágenes obtenidas en la modalidad de resonancias magnéticas. Cuenta con una interfaz de usuario sencilla e intuitiva. Adapta la pantalla para distintos entornos y permite mostrar varias imágenes de un mismo estudio. Ofrece visualización de datos en 2D y 3D de las imágenes en formato DICOM. Permite desplazarse a través de cada uno de los cortes de la secuencia, realizar zoom, panorámicas y rotación. Proporciona herramientas de medición entre las que se encuentran las de distancia, ángulos y áreas. Se pueden realizar anotaciones en cada uno de los estudios para describir hallazgos de alguna patología. Es compatible con Windows y macOS, no requiere instalación y no se integra a sistemas PACS.(19, 20)

**Horos:** Es un software de visualización de imágenes médicas basado en el formato DICOM. Es una plataforma de código abierto que se desarrolló a partir del código fuente de OsiriX. Permite la visualización de imágenes médicas en 2D y 3D mediante la cual se puede explorar y navegar a través de las imágenes utilizando la herramienta de zoom, panorámica y rotación. Cuenta con herramientas para realizar mediciones

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

precisas ya sea de manera lineal, angulares, de área y de densidad. Es compatible con resonancias magnética. Se puede integrar con los sistemas PACS para acceder y gestionar imágenes médicas desde diferentes fuentes. Es de código abierto lo cual permite aumentar su portabilidad. Está disponible solamente para el sistema operativo macOS.(21–23)

**eFilm:** Es una aplicación de software de visualización de imágenes médicas de alta calidad desarrollada por la compañía Merge. Esta aplicación está diseñada para visualizar varios formatos de imágenes entre los que se incluye DICOM. Admite la visualización de imágenes obtenidas en la modalidad de resonancia magnética. Presenta una interfaz intuitiva y sencilla para el usuario. Ofrece herramientas para realizar mediciones lineales, de área y de ángulos con la capacidad de aplicar filtros sobre las imágenes. Cuenta con la funcionalidad de agregar varias pantallas para la comparación de estudios. No se integra a entornos PACS, es una herramienta independiente.(24)

Tabla 1 Comparación de estudios Homólogos. Fuente: "Elaboración propia"

Herramientas	Cargar varios estudios	Mediciones lineales	Mediciones de área	Software libre	Integración a entornos PACS	Compatible formato DICOM
OsiriX:	No	Sí	Sí	No	Sí	Sí
RadiAnt DICOM Viewer:	No	Sí	Sí	No	Sí	Sí
Horos:	Sí	Sí	Sí	Sí	Sí	Sí
eFilm:	Sí	Sí	Sí	No	No	Sí

A partir del análisis de estas herramientas se identifican para el componente de comparación de estudios la capacidad de poder cargar en las ventanas de visualización varios estudios en distintas etapas de un mismo paciente con la capacidad de replicar las mediciones en ambos estudios a la vez, estableciendo así la ocurrencia de cambios en la condición médica del paciente y dando la posibilidad de realizar un seguimiento evolutivo de la enfermedad.

### **1.4. Algoritmos de calibración de imágenes DICOM para realizar mediciones**

Para la realización de mediciones de manera exacta sobre imágenes médicas es necesario realizar la calibración de todas las imágenes de manera individual debido a la diferencia de escalas entre las distancias de los píxeles de una imagen y las mediciones a escalas reales. La calibración consta de la estructuración de una escala relacionada entre el tamaño de la imagen en pantalla y el tamaño de las estructuras u objetos de la imagen en tamaño real, mediante la investigación y análisis de varias investigaciones, trabajos y artículos referentes a algoritmos con esta función se han seleccionado dos posibles algoritmos que pueden ser aplicados en el desarrollo de la propuesta de solución. Estos algoritmos son calibración geométrica de imágenes DICOM y algoritmo de calibración por objeto.(25, 26)

#### **1.4.1 Algoritmo de calibración de imágenes DICOM basada en objetos.**

La calibración de imagen DICOM basada en objeto es un método de calibración que utiliza un objeto de referencia con dimensiones conocidas para establecer una relación entre los valores de píxeles y las medidas físicas en una imagen DICOM. Este método de calibración se utiliza comúnmente en imágenes médicas para realizar mediciones precisas.(27)

Para calibrar una imagen DICOM utilizando este método, se debe seleccionar un objeto de referencia con dimensiones conocidas que se encuentre en la imagen. La medida física del objeto de referencia se ingresa en el sistema de imágenes, que luego utiliza esta información para calibrar la imagen.(27)

Los objetos de referencia comúnmente utilizados incluyen objetos de calibración con dimensiones conocidas, como bloques de plástico con orificios de diámetros conocidos o varillas de metal con longitudes conocidas. Estos objetos de referencia generalmente vienen con una certificación que indica su precisión y precisión en las medidas.(28, 29)

Es importante tener en cuenta que la selección del objeto de referencia adecuado es crítica para garantizar una calibración precisa de la imagen. Además, es importante seguir las instrucciones de calibración específicas proporcionadas por el fabricante del sistema de imágenes para garantizar una calibración precisa y confiable.(29)

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En resumen, la calibración de imagen DICOM basada en objeto es un método comúnmente utilizado para realizar mediciones precisas en imágenes médicas. Este método utiliza objetos de referencia con dimensiones conocidas para establecer una relación entre los valores de píxeles y las medidas físicas en una imagen DICOM.

Analizando el funcionamiento de este algoritmo en su estructura obliga al especialista a introducir manualmente en el software los datos de tamaño del objeto de referencia. Este tipo de calibración trae consigo inexactitud en el momento de introducción de las mediciones pues cabe la posibilidad de que se introduzcan valores erróneos y restando exactitud en la calibración. Una calibración mal elaborada produce que el resto de mediciones estén condicionadas a este error.

### **1.4.2. Algoritmos de calibración de imágenes DICOM basada en geometría**

La calibración basada en la geometría es un algoritmo utilizado en la calibración de imágenes médicas en formato DICOM para la realización de mediciones con mayor precisión. Este algoritmo utiliza la geometría de la imagen para calcular el tamaño de los píxeles y la escala de la imagen, lo que permite medir distancias y áreas con mayor precisión en la imagen.(30)

La geometría de la imagen se refiere a la posición y orientación de la imagen en el espacio tridimensional. Para utilizar este algoritmo, se requiere que la imagen tenga información sobre la posición y orientación en el espacio tridimensional. Esta información se almacena en los campos de "*Image Position (Patient)*" y "*Image Orientation (Patient)*" en el archivo DICOM.(30)

El algoritmo de calibración basado en la geometría utiliza la información de posición y orientación para calcular el tamaño de los píxeles y la escala de la imagen. Este algoritmo funciona seleccionando dos puntos en la imagen que están separados por una distancia conocida en el mundo real. Luego, utiliza la información de posición y orientación de la imagen para calcular la distancia entre los dos puntos en el mundo real y la distancia entre los dos puntos en la imagen. El tamaño de los píxeles se calcula utilizando la fórmula: tamaño de los píxeles = distancia en el mundo real / distancia en la imagen. Finalmente, se utilizan los valores de tamaño de los píxeles para calcular la escala de la imagen utilizando la información de *rescale slope* y *rescale intercept*, que se encuentran en la información de la imagen DICOM.(31, 32)

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

Una vez que se ha realizado la calibración basada en la geometría, se pueden medir distancias y áreas con mayor precisión en la imagen. Sin embargo, es importante tener en cuenta que este algoritmo puede ser sensible a la precisión de la información de posición y orientación de la imagen y a la elección de los puntos de referencia utilizados para la calibración. Además, la precisión de la calibración también puede verse afectada por factores como la resolución de la imagen y el ruido de la imagen.(32)

### 1.4.3. Comparación de los 2 algoritmos a partir de sus ventajas y desventajas

Tabla 2 - Comparación de los algoritmos de calibración para imágenes DICOM.

Algoritmo	Ventajas	Desventajas
<b>Basado en objeto</b>	No depende de la información de posición y orientación de la imagen	Requiere la utilización de un objeto de referencia físico
	Es menos sensible a la resolución de la imagen y al ruido.	La precisión de la calibración puede verse afectada por la iluminación y las condiciones ambientales.
	Puede ser más preciso en la calibración de imágenes con distorsiones geométricas	Puede ser más costoso y más difícil de implementar
<b>Basado en geometría</b>	Utiliza la información de precisión y orientación de la imagen para calcular el tamaño de los píxeles y la escala de la imagen.	Puede ser sensible a la precisión de la información de posición y orientación de la imagen
	Es preciso para la calibración de imágenes con geometría regular.	La precisión de la calibración puede verse afectada por la resolución y el ruido de la imagen
	Es más fácil de implementar y no requiere objetos de referencia físicos.	Puede ser menos preciso en la calibración de imágenes con distorsiones geométricas.

Ambos algoritmos tienen sus propias ventajas y desventajas. El algoritmo basado en objeto no depende de la información de posición y orientación de la imagen, es menos sensible a la resolución y el ruido de la imagen, y puede ser más preciso en la calibración de imágenes con distorsiones geométricas. Sin embargo, requiere la utilización de un objeto de referencia físico, puede ser más costoso y más difícil de implementar, y la precisión de la calibración puede verse afectada por la iluminación y las condiciones ambientales.

Por otro lado, el algoritmo basado en geometría utiliza la información de posición y orientación de la imagen para calcular el tamaño de los píxeles y la escala de la imagen,

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

es preciso para la calibración de imágenes con geometría regular, es más fácil de implementar y no requiere objetos de referencia físicos. Sin embargo, puede ser sensible a la precisión de la información de posición y orientación de la imagen, la precisión de la calibración puede verse afectada por la resolución y el ruido de la imagen, y puede ser menos preciso en la calibración de imágenes con distorsiones geométricas.

Para la solución informática propuesta se utilizará el algoritmo de calibración geométrica pues es el más sencillo y exacto aplicable en las posibles calibraciones a realizar en caso de ocurrir error en las resonancias magnéticas. También prevé el posible cambio que puede ocurrir en la introducción de datos dentro de un archivo DICOM que se vaya a almacenar y actualiza las variables de escala en la imagen.

### **1.4.4. Algoritmos para la medición de imágenes médicas.**

Se consultó bibliografía actual y variada en bases de datos referenciadas como PubMed, ISI, Frontiers, entre otras para identificar y obtener información asociada a los algoritmos necesarios para la realización de mediciones de distancia en imágenes médicas. Este análisis propició seleccionar el algoritmo más adecuado en la realización de mediciones sobre imágenes de resonancias magnéticas cerebrales en formato DICOM. Los algoritmos para mediciones de distancia en imágenes fueron la Transformada de Hough (33), Detección de puntos de referencia (34), Algoritmo de Canny (35).

Para las siguientes comparaciones se utilizó el set de datos de imágenes medicas de resonancia magnética en formato DICOM proporcionado por CNEURO. Estas imágenes se sometieron a un tratamiento de anonimización a fin de mantener la integridad de los pacientes a los cuales pertenecen estos estudios. Las imágenes presentes en este set de datos fueron tomadas bajo las mismas medidas, condiciones y se calibraron debidamente por los expertos en radiología del Centro de Neurociencias de Cuba.

#### **1.4.4.1 Algoritmo de detección de puntos de referencia.**

Este algoritmo se basa en la distancia que se encuentra entre dos puntos de una imagen. Estos puntos unidos por una recta que se encuentra en un plano de dos dimensiones, en este caso una imagen. Una imagen bidimensional se encuentra representada por un conjunto de píxeles formando una matriz de dos dimensiones. Cada una de estas dimensiones se representa en largo y ancho como los ejes de referencia X y Y en la geometría matemática.(34). Para la realización de este cálculo de distancia en línea recta se utiliza la fórmula euclidiana.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

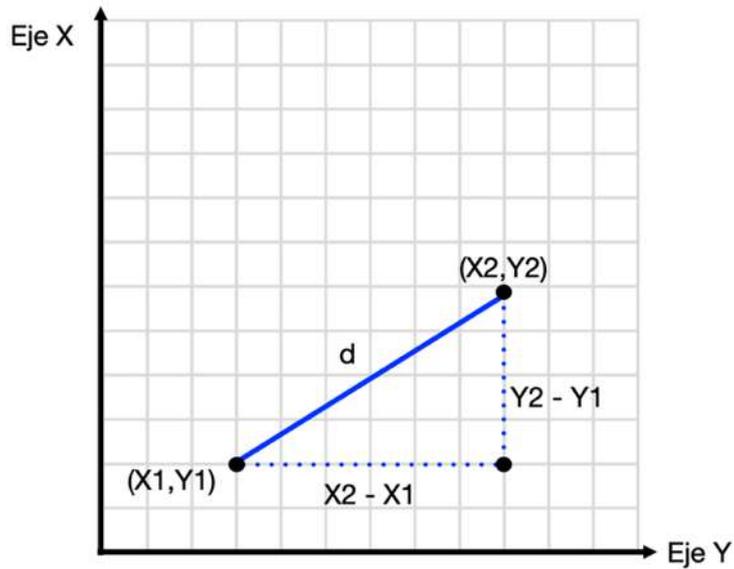


Figura 2 - Espacio Cartesiano para calcular la distancia de un vértice a otro. Fuente: Elaboración propia

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (\text{I})$$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (\text{II})$$

Una vez despejada la ecuación se obtiene la distancia en píxeles desde un punto al otro de la imagen, pero se hace necesario conocer la distancia real fuera del eje de coordenadas. Por lo que es necesario multiplicar el resultado por la cantidad de filas de píxeles contenidas en la imagen o multiplicar por el valor de escala de la imagen.(34)

Para las imágenes DICOM se utiliza el valor *PixelSpacing* contenido en los *Tags* que proporciona el propio archivo(36, 37). Así se logra transformar la medición de unidades euclidianas a unidades métricas reales. Lo anterior se resume en la fórmula.

$$\text{Distancia} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} * \text{PixelSpacing} \quad (\text{III})$$

Al realizar el cálculo sustituyendo las variables por los valores se obtiene la distancia de un punto hasta el otro. La exactitud de esta medición depende de la manera en que se obtuvieron, trataron y calibraron los valores de la imagen. En las figura 3 se muestra la implementación del algoritmo.

```

1 def DetecciónPuntosRef(self):
2     if self.inicio and self.fin:
3         self.canvas.delete("linea_anterior")
4         self.canvas.create_line(self.inicio[0], self.inicio[1], self.fin[0],
5                                 self.fin[1],
6                                 fill="red", width=1, tags="linea_anterior")
7         self.longitud = math.sqrt((self.fin[0] - self.inicio[0])**2
8                                 + (self.fin[1] - self.inicio[1])**2)
9         print (self.longitud)
10
11 PixelSpacing = self.serie[self.indice_actual].PixelSpacing
12 self.longitud = (self.longitud * PixelSpacing[0]) / 10
13 self.valor_longitud.config(text=f"Longitud: {self.longitud:.2f} cm")

```

Figura 3 - Algoritmo de Detección de Puntos de Referencia

En la implementación se obtienen los datos de posición del punto de referencia inicial y final de la línea de medición. Estos datos se sustituyen en la ecuación para el cálculo de distancia en el plano cartesiano de la matriz de píxeles. El resultado de la medición es multiplicado por la variable PixelSpacing contenida en las Tags de la imagen. Se crea la línea en el espacio de la imagen y se devuelve el valor de la medición.

#### 1.4.4.2. Algoritmo de medición por la transformada de Hough

Es un método propuesto por Hough (1962), para encontrar la ecuación de una línea que pase por un conjunto de puntos en el plano xy. Una línea que une una secuencia de puntos o píxeles puede expresarse con la ecuación de la recta:  $y = mx + b$  siendo b la pendiente de esta recta. (33).(38)

Se puede describir de manera analítica un segmento de línea en varias formas. Sin embargo, una ecuación conveniente para describir un conjunto de líneas es la notación paramétrica o normal.

$$\rho = x \cos \theta + y \sin \theta \quad (IV)$$

Donde  $\rho$  es la longitud normal desde el origen hasta la línea y  $\theta$  es el ángulo de  $\rho$  con respecto al eje x.(39)

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

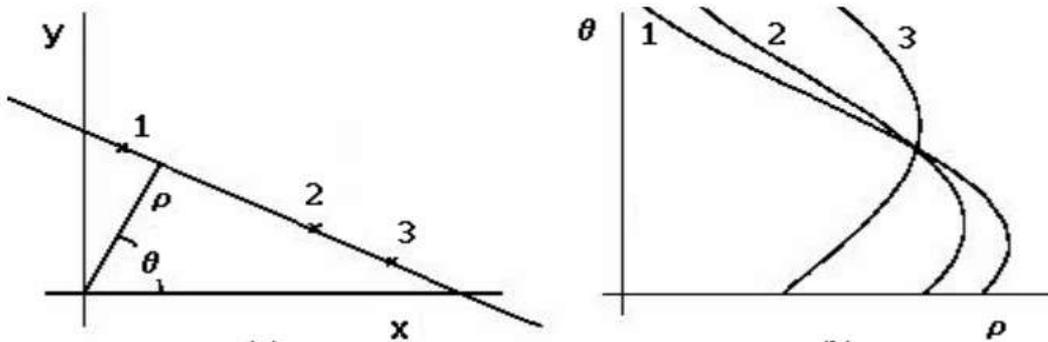


Figura 4 - Representación del espacio cartesiano y representación del espacio paramétrico. (Santini & Santini, 2020.)

En la figura 3 se muestra:

El espacio cartesiano que representa la línea físicamente visible.

El espacio paramétrico del algoritmo de Hough, en el cual la línea se comporta como una curva dependiendo del valor de los píxeles cercanos.

Al ser representadas estas líneas en forma de matriz se evidencia el camino seguido a partir de su pendiente correspondiente y el ángulo de cada uno de los píxeles seguidos. (33). Finalizada la detección estas líneas rectas representadas en la imagen funcionan como guía para la realización de las mediciones a fin de dar mas exactitud en estas; dada la capacidad de mostrar las posibles estructuras en la imagen con estas formas geométricas(40).

Para llevar a la implementación el algoritmo anterior se utilizó la librería OpenCV perteneciente al lenguaje Python (41–43). En la figura 5 se muestra su implementación.

```
1 def Hough(imagenz):
2     gris = cv2.cvtColor(imagenz, cv2.COLOR_BGR2GRAY)
3     bordes = cv2.Canny(gris, 50, 150, apertureSize=3)
4     kernel = np.ones((2,2),np.uint8)
5     bordes_dil = cv2.dilate(bordes,kernel, iterations=1)
6
7     minLineLength = 100
8     maxLineGap = 10
9
10    lines = cv2.HoughLinesP(bordes_dil, 1, np.pi / 180, 50, minLineLength, maxLineGap)
11
12    for line in lines:
13        for x1, y1, x2, y2 in line:
14            cv2.line(img, (x1, y1), (x2, y2), (0, 0, 255), 2)
15
16    return img
```

Figura 5 - Implementación Algoritmo Transformada de Hough

En esta implementación la imagen cargada se somete a los distintos filtros. En su primera la imagen es convertida en una cadena de tonos grises para resaltar las

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

tonalidades en los bordes. Se le aplica el filtro para la detección de bordes en su segundo paso. Estos bordes entran al filtro de para detectar líneas y finalmente se superpone en la imagen final. En la figura 6 se muestra el resultado por etapas de la Transformada de Hough.

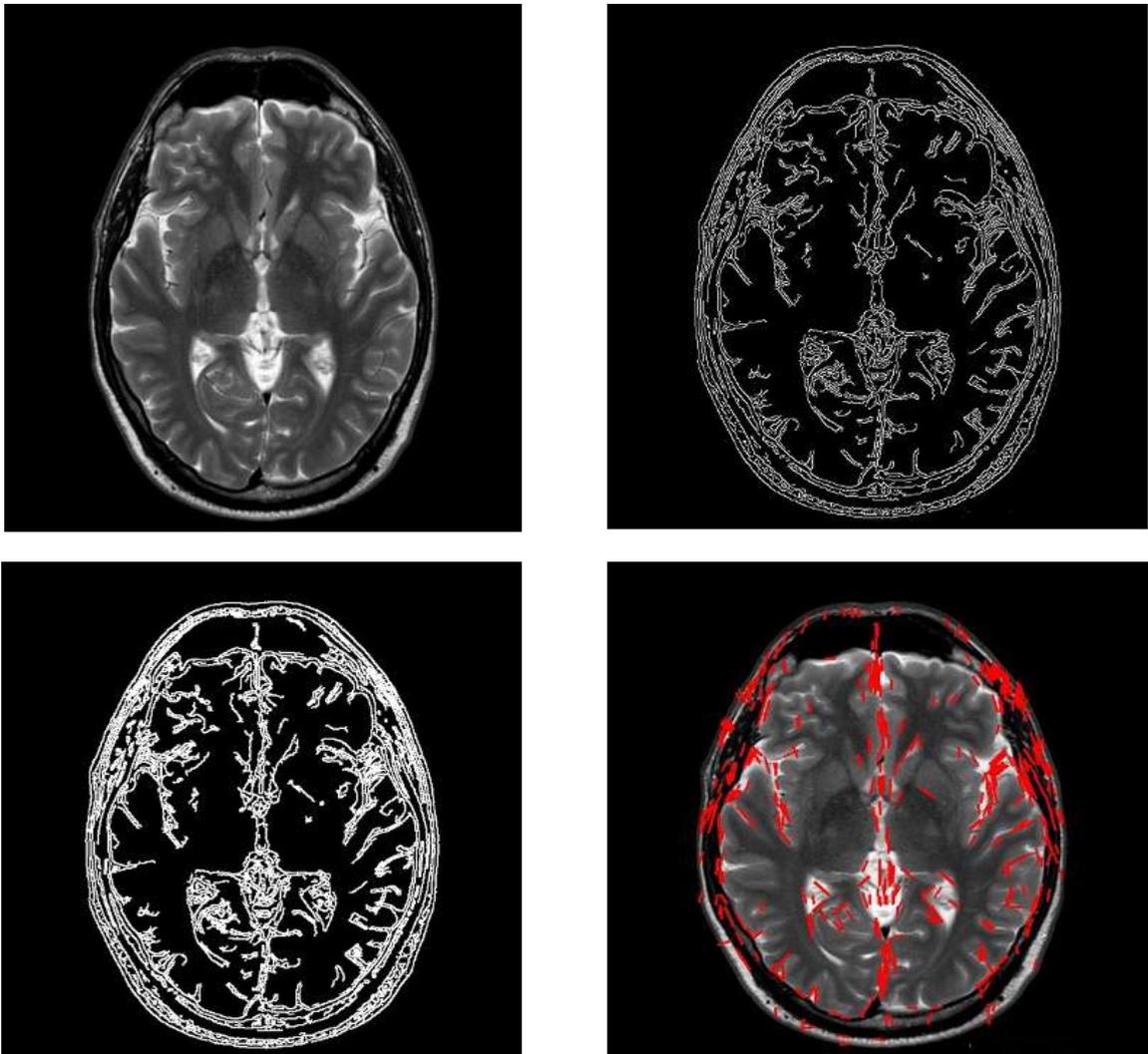


Figura 6 - Proceso de imágenes con Algoritmo Transformada de Hough. Fuente: Elaboración propia

Aplicado el algoritmo sobre una imagen se resalta la existencia de estas líneas rectas contenidas en la matriz de píxeles(44). Al ser trazadas estas líneas apoyan la funcionalidad de poder trazar las mediciones sobre estas líneas estructurales a fin de mejorar la exactitud en el momento de la medición.

### 1.4.4.3. Algoritmo para la detección de bordes. Algoritmo de Canny.

El algoritmo de detección de bordes o algoritmo de Canny se conoce como filtro de alto paso por su cualidad para detectar los límites entre dos regiones con un nivel de gris relativamente distinto de propiedades. En este los bordes se consideran son píxeles en

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

los que la función de brillo cambia bruscamente, siendo los detectores de bordes una colección de métodos de procesamiento de imágenes locales muy importantes que se utilizan para localizar cambios nítidos en la función de intensidad.(35, 45–47)

La ecuación para esta representación está argumentada por:

$$(G) = \sqrt{G_x^2 + G_y^2} \quad (V)$$

$$Angle (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (VI)$$

A partir de la fórmula del gradiente de color de cada pixel se establece el peso de los mismos y aplicando la fórmula para el ángulo de la evaluación de gradiente a fin de trazar el camino a seguir. Esto da como resultado que los bordes son discontinuidades repentinas en una imagen, que pueden surgir de la superficie normal, el color de la superficie, la profundidad, la iluminación u otras discontinuidades(35, 48, 49). Estos bordes son importantes por dos razones fundamentales.

La mayor parte de la información semántica y de formas se puede deducir de ellos, por lo que se puede realizar el reconocimiento de objetos y analizar las perspectivas y la geometría de una imagen.

Es una representación más compacta que los píxeles. Se puede señalar dónde se producen los bordes desde el perfil de intensidad de una imagen a lo largo de una fila o columna de la imagen. Dondequiera que haya un cambio rápido en la función de intensidad indica un borde.

Como parte de su procesamiento típico se aplica el filtro de la siguiente manera visual.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

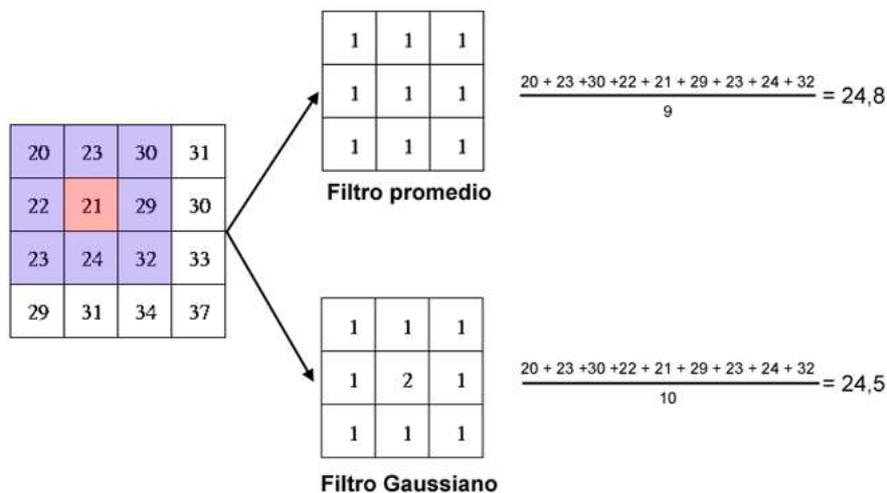


Figura 7 - Representación de los valores de píxeles (NIÑO NAVARRETE, 2019)

Se le adjudica la prioridad al pixel central al cual se le hace el cálculo de gradiente para detectar si existe un cambio brusco de coloración de un pixel a otro. Utilizando la librería OpenCV se implementa y prueba el algoritmo de Canny en lenguaje python y realizar la prueba de detección en una imagen cualquiera a escalas de grises(41–43). En la figura 8 se muestra la implementación del algoritmo.

```
1 def CannyM(imagenz):
2
3     gris = cv2.cvtColor(imagenz, cv2.COLOR_BGR2GRAY)
4
5     gauss = cv2.GaussianBlur(gris, (1,1), 0)
6
7     canny = cv2.Canny(gauss, 50, 150)
8
9     (contornos, _) = cv2.findContours(canny.copy(),
10                                     cv2.RETR_EXTERNAL,
11                                     cv2.CHAIN_APPROX_SIMPLE)
12
13     cv2.drawContours(imagenz, contornos, -1, (0,0,255), 2)
14
15     return imagenz
```

Figura 8 - Implementación Algoritmo de Canny. Fuente: Elaboración propia

El Algoritmo pasa la imagen por distintos filtros. En la primera instancia pasa por el filtro de grises para resaltar los cambios de coloración. En su segunda instancia entra al filtro Gaussiano, este reduce la resolución de la imagen en un margen de (1,1) para entrar al filtro de detección de bordes. En la tercera instancia la imagen entra en el filtro de

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

detección de contornos y superpone los bordes detectados sobre la imagen inicial. En la figura 9 se ve el proceso después de pasar por cada uno de los filtros.

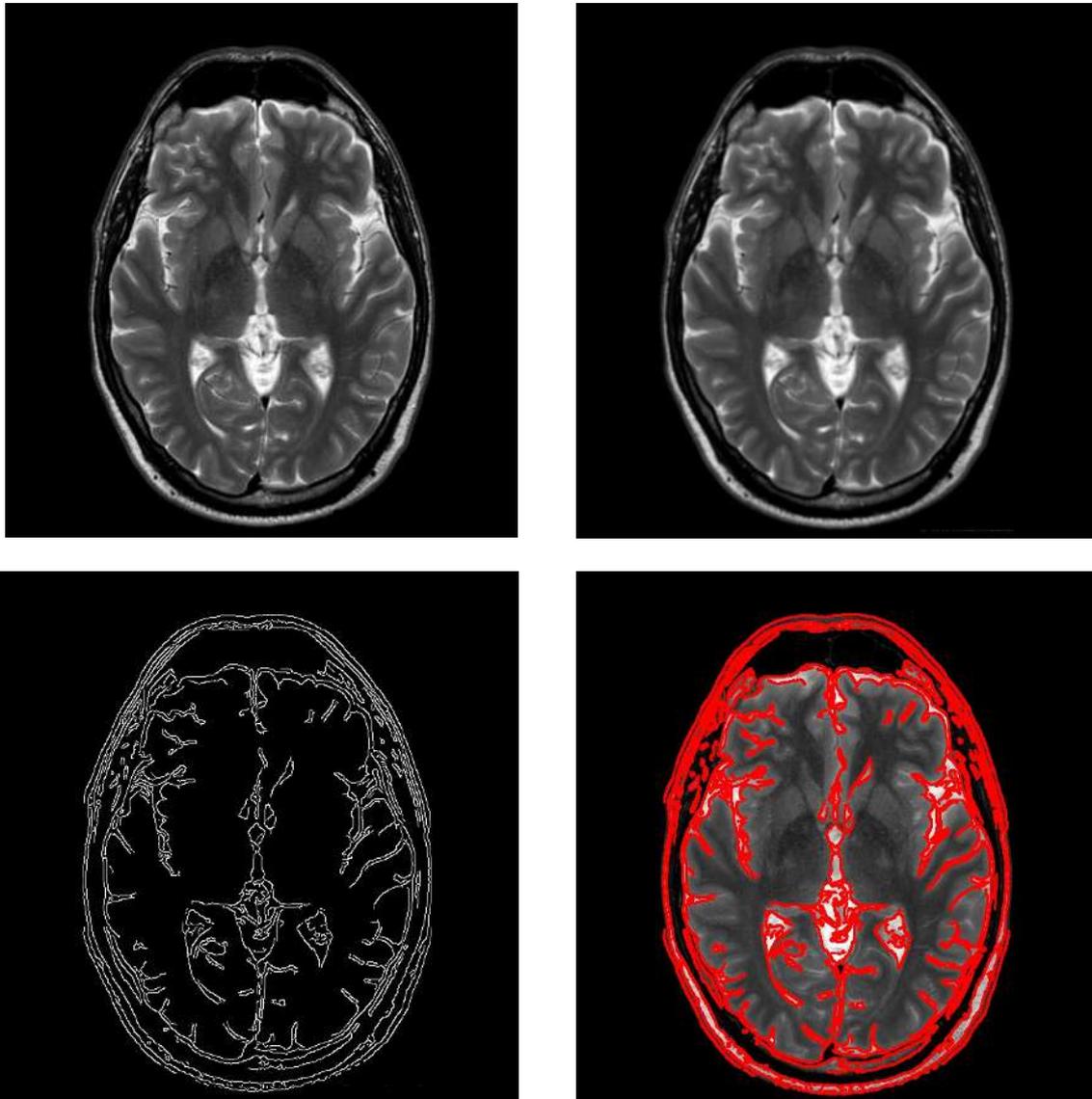


Figura 9 - Proceso de detección de bordes del Algoritmo de Canny. Fuente Elaboración propia.

Este Algoritmo realiza la detección y trazado de todos los contornos en la imagen cerebral, pero por consiguiente ignora la formación de algunas estructuras u otras formas tumorales que se asemejan a estructuras básicas del cerebro, por ende, ayuda con las mediciones pero no tiene en cuenta la existencia de estas. Además, al existir ruido en la imagen, este no es capaz de detectar los cambios para trazar los contornos.

### **1.4.4.4. Comparación de los algoritmos aplicables a la solución informática.**

Para realizar las pruebas se seleccionaron al azar un grupo de cortes cerebrales de distintos estudios del dataset utilizado. Los cortes son de distintas áreas y regiones a fin

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

de obtener resultados aplicables a cada zona cerebral. Todas las imágenes seleccionadas fueron sometidas a la aplicación de estos algoritmos.

Al realizar el análisis comparativo de los algoritmos utilizados para las mediciones en imágenes cerebrales los autores obtuvieron los siguientes resultados, significativos para la presente investigación.

Tabla 3 - Comparación de los algoritmos analizados. Fuente: Elaboración propia

Algoritmos	Pruebas realizadas	Observaciones	Precisión
Detección de puntos de referencia	10	Permite realizar las mediciones a partir de los datos de matriz de la imagen y la variable de escala real (PixelSpacing) obtenida en los metadatos de la imagen.	97.8%
Transformada de Hough	10	Permite la detección de estructuras lineales en las imágenes de médicas a fin de realizar mediciones más exactas.	95.3%
Algoritmo de Canny	10	Permite detectar los contornos de cada una de las estructuras anatómicas del cerebro a partir de los bordes de cada estructura.	85.5 %

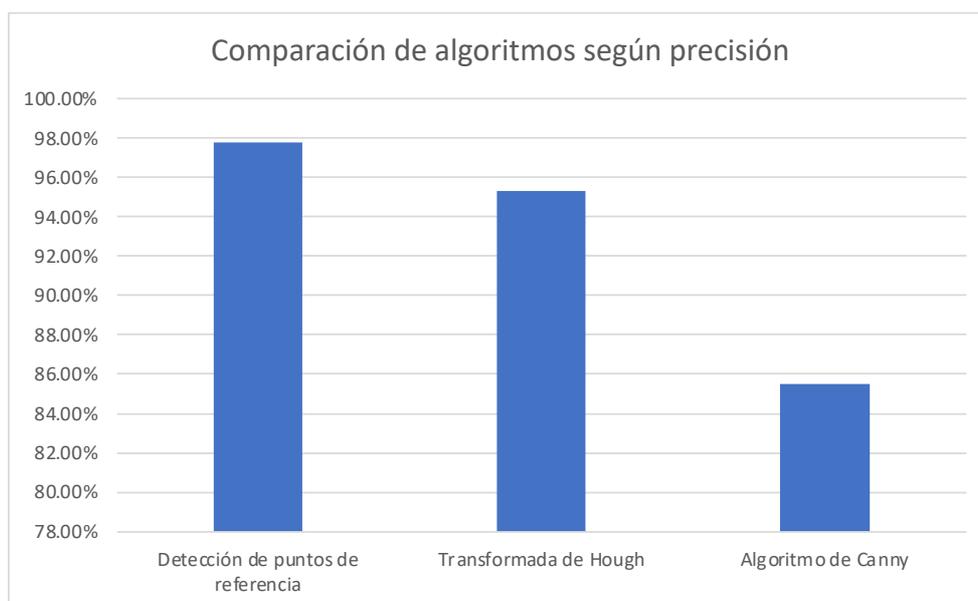


Figura 10 - Comparación de los algoritmos analizados según precisión. Fuente: Elaboración propia

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

En la tabla anterior se aprecia la diferencia de precisión entre los algoritmos analizados. En los casos de prueba a los que se sometieron se utilizaron los mismos estudios y las mismas condiciones a fin de obtener resultados más justos. Se utilizó el software eFilmWorkstation para comparar y verificar la precisión de las mediciones realizadas con los algoritmos(24).

### **1.4.4.5. Resultados de la comparación de algoritmos para la medición**

Al ejecutar el análisis de los algoritmos de medición localizados en la literatura revisada se pudo determinar la precisión de cada uno de estos para la medición. Se analizó la utilidad de los mismos en cada caso de prueba aplicado. Se describe en los siguientes puntos.

- La transformada de Hough es aplicable para resonancias magnéticas en distintas zonas del cuerpo pero no es aplicable de manera precisa para realizar mediciones en imágenes cerebrales. En los casos de prueba utilizados detecta muy pocas líneas en las imágenes del cerebro, por este motivo no es viable para estas mediciones.
- El algoritmo de Canny, utilizado para la detección de contornos detecta áreas del cerebro en las cuales sus contornos son resaltantes, pero no detecta las demás en las cuales se difumina levemente el borde. No es aplicable pues no detecta todos los contornos al existir distintos degradados o zonas perfectamente resaltadas. Al realizar las mediciones sobre este y no detectar todos los contornos no muestra utilidad en las mediciones.
- La detección de puntos de referencia para las mediciones mostró ser el algoritmo más preciso en las pruebas para las mediciones. Este utiliza para su estimación de distancia la fórmula euclidiana y la variable de escala PixelSpacing contenida en los tags de la imagen DICOM. En este caso las mediciones se realizan de manera manual y dependen de la exactitud en que se seleccionen los puntos para medir.

El algoritmo seleccionado para la solución informática es el algoritmo de detección de puntos de referencia. Al realizar las pruebas de mediciones mostró resultados similares en un 97.8 % respecto a las realizadas con el software eFilmWorkstation(24).

### 1.5. Metodologías, tecnologías y herramientas.

La selección adecuada de las metodologías a utilizar a la hora del desarrollo del software es fundamental. En este caso las mismas utilizadas fueron definidas por el Centro de Informática Médica (CESIM) para el desarrollo de software en el proyecto XAVIA-PACS

**Metodología de Desarrollo:** Agile Unified Process (AUP) versión UCI (AUP-UCI) describe una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. En el caso de la variación definida para la UCI se propone para el ciclo de vida de los proyectos las fases: Inicio, Ejecución y Cierre. Cuenta con 4 escenarios posibles en los proyectos de desarrollo de software:(50)

Escenario 1: proyectos que modelan el negocio con casos de uso del negocio (CUN) solo pueden modelar el sistema con casos de uso del sistema (CUS).

Escenario 2: proyectos que modelan el negocio con modelo conceptual (MC) solo pueden modelar el sistema con casos de uso del sistema (CUS).

Escenario 3: proyectos que modelan el negocio con descripción de proceso de negocio (DPN) solo pueden modelar el sistema con descripción de requisitos de procesos (DRP)

Escenario 4: proyectos que no modelen negocio solo pueden modelar el sistema con historias de usuario (UH).

El escenario a utilizar para el siguiente desarrollo del componente de comparación de estudios sobre imágenes DICOM de resonancia magnética es el escenario 3 que es el actual definido por el proyecto de desarrollo XAVIA por lo que se hace necesario realiza modelos de negocio a partir de la descripción de los procesos de negocio y modelar los procesos a partir de la descripción de los requisitos de procesos. Para el desarrollo de este componente se definen además un conjunto de herramientas, tecnologías, lenguaje, arquitectura y metodología. Las decisiones generales empleadas para esta tarea se derivan de las actuales necesidades del sistema XAVIA-PACS. De esta forma se mantiene la integridad tecnológica en la solución general. En este epígrafe se definen estas.

### 1.5.1 Ambiente de Desarrollo

**Visual Studio Code versión 1.48.2:** es un editor de código fuente desarrollado por Microsoft, es ligero pero potente que está disponible para distintos sistemas operativos. Tiene la capacidad de adaptarse a cualquier flujo de trabajo y da la posibilidad de personalizar cada uno de sus paquetes, temas y extensiones. Cuenta además con la capacidad de integrarse a distintas tecnologías como Git para el control de versiones, IntelliSense para la finalización de código de y la navegación. Además, posee la capacidad de adicionarle cientos de extensiones lo que le permite agregar nuevas funcionalidades y copilotos de codificación para la corrección de muchos errores de sintaxis y es compatible con distintos lenguajes de programación.(51)

**Git versión 2.40.1:** es un sistema de control de versiones distribuido diseñado para manejar proyectos de cualquier tamaño y tipo con velocidad y eficiencia. Es fácil de aprender, tiene una huella pequeña y un rendimiento rápido. Permite a varios desarrolladores trabajar en el mismo proyecto simultáneamente, incluso si no están en la misma ubicación física, y está diseñado para manejar proyectos que contienen cientos de miles de archivos y terabytes de datos. Git cuenta con una amplia gama de características y herramientas, como ramificación y fusión de código, control de versiones, integración con herramientas de desarrollo y personalización y extensibilidad para adaptarse a las necesidades específicas de los desarrolladores. En general, Git se ha convertido en una herramienta esencial para los desarrolladores de todo el mundo.(52)

**Herramienta CASE Visual Paradigm versión 8.0:** es una herramienta de modelado de software líder en la industria que permite a los desarrolladores modelar, diseñar y documentar software utilizando varios lenguajes de modelado. Ofrece integración con diversas herramientas y tecnologías, como Java, .NET y C++, y admite la integración con herramientas de gestión de proyectos y control de versiones para trabajar en equipo de manera efectiva. También cuenta con características como la simulación y validación de modelos, la generación de informes y documentación detallada, y la personalización y extensibilidad para adaptarse a las necesidades específicas de los desarrolladores. En general, Visual Paradigm es una herramienta esencial para los desarrolladores de software de todo el mundo.

#### Lenguaje de Programación

**Python versión 3.11.6:** es un lenguaje de programación interpretado y de alto nivel que se utiliza en una amplia variedad de aplicaciones, desde el desarrollo web hasta el

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

análisis de datos y la inteligencia artificial. Ofrece numerosas bibliotecas y frameworks, como NumPy, pydicom, para realizar tareas de manera eficiente. Además, su sintaxis clara y concisa hace que sea fácil de aprender y leer, y es un lenguaje multiplataforma que se puede ejecutar en diferentes sistemas operativos. Python también cuenta con una comunidad activa y una amplia gama de recursos en línea, lo que lo convierte en una herramienta esencial para los desarrolladores de todo el mundo.(53) Utilizaremos de este lenguaje distintas librerías entre las que figura pydicom para el trabajo con las imágenes en formato DICOM y *pyQt* para el desarrollo de las interfaces gráficas basadas en *QtCore*, *QtGUI* y *QtWidget*.

### 1.6. Conclusiones del capítulo

Al analizar las bibliografías referentes a la comparación de estudios sobre imágenes de resonancia magnética en formato DICOM se arribaron a las siguientes conclusiones:

- El análisis de los elementos homólogos a la solución informática identificó como requisitos importantes la capacidad de comparar imágenes a partir de mediciones precisas sobre estas. Además la capacidad de cargar múltiples estudios de un paciente para realizar estas comparaciones.
- De los algoritmos comparados para la calibración de imágenes de resonancia magnética se seleccionó el algoritmo de calibración por geometría pues este utiliza los datos propios de la imagen para realizar este proceso.
- Se seleccionó el algoritmo de detección de puntos de referencia para realizar las mediciones por ser el más preciso en los resultados.
- Se aplicará la metodología ágil AUP-UCI en su escenario 3 pues esta es la utilizada para documentar el Sistema XAVIA-PACS, a fin de mantener la documentación homogénea para su continuo desarrollo.
- Se definió como lenguaje de programación el lenguaje Python. Este lenguaje cuenta con la librería *PyDICOM* para facilitar el trabajo y correcto tratamiento con imágenes en formato DICOM.

### **2. PROPUESTA DE SOLUCIÓN.**

En este artículo se abordan los principales elementos para elaborar los artefactos necesarios en el desarrollo del componente informático. Entre estos se comprende el levantamiento de los requisitos y la descripción de los mismos, la descripción del modelo de negocio, la propuesta de la arquitectura y los diagramas de secuencia necesarios para entender el funcionamiento de las principales funcionalidades.

En este capítulo se aborda el diseño del componente de comparación de estudios sobre imágenes DICOM de resonancia magnética. Se realizan y detallan los requisitos necesarios para cada una de las funciones del componente y los diagramas necesarios para el entendimiento del desarrollo de este. Se generan los artefactos necesarios a fin lograr el entendimiento para el desarrollo del componente de comparación. Se propone un modelo arquitectónico cómodo y asequible a las necesidades para la implementación de la solución informática propuesta.

#### **2.1. Descripción de la propuesta de solución**

Esta herramienta de comparación de estudios de imágenes DICOM de resonancia magnética está concebida principalmente para el uso de los profesionales de la medicina, específicamente para los neurólogos, pero es aplicable además en otras áreas de la medicina en las cuales se utilicen imágenes de resonancia magnética. Cuyas características se enfocan en las comparaciones de medidas entre algunos de los elementos encontrados en dichas imágenes médicas; ya sea por la ocurrencia de tumores cerebrales, contusiones, derrames cerebrales, entre otras afectaciones que se puedan presentar en este órgano.(54)

Una de sus principales funciones radica en la posibilidad de visualizar datos de imagen en formato DICOM lo cual ayuda al análisis de dichos archivos. Una vez accedido a los mismos permite desplazarse en cada uno de los cortes cerebrales, para analizar cada sección de manera independiente o para buscar de manera más exacta en las áreas afectadas.

Debe permitir cargar varias pantallas a la vez, realizando la sintonización de estas para posicionarla en el mismo número de corte y en distintos ángulos de visión de los mismos, siendo útil en gran medida la funcionalidad de girar una imagen. En cada una de las pantallas permite realizar la replicación de las mediciones en cualquiera de las demás pantallas.

Como función principal de medición contará con la capacidad de poder trazar líneas sobre las imágenes con las medidas en tamaño real, también la capacidad de trazar otras por encima de esta línea para poder calcular otras zonas afectadas relacionadas como por ejemplo con la existencia de tumores cerebrales. Otra herramienta muy destacada para la realización de mediciones mediante trazos le corresponde al trazado de un área circular, el cual mide el área de una zona afectada de manera radial junto con la posibilidad de editar y cambiar el tamaño de dicha área.

Otra de las funciones del mismo es la capacidad de realizar el desplazamiento automático de todo el set de imágenes de manera que estas se muestren en una secuencia ordenada generando un video fluido que permite apreciar de mejor manera cada una de las secciones y mostrar los cambios existentes en cada una de las áreas. Haciendo posible que se sincronicen distintos estudios o set de imágenes en la misma área con tan solo seleccionar una de las secciones transversales de imágenes.

### **2.1. Modelo de Dominio**

Las clases de dominio y las relaciones entre ellas corresponden al modelo de Dominio. En este caso el modelo de Dominio correspondiente para el componente de comparación de estudios de imagen en formato DICOM muestra la relación entre el especialista en neurociencia y los métodos para la comparación de las imágenes incluyendo la visualización detallada de la misma. Se realiza con el propósito de sentar las bases más importantes en el desarrollo del software pero no lo define completamente. Estas imágenes por parte del centro del cual provienen ya deben estar asimiladas al estándar con el cual se está trabajando en ese momento y el cual se ha mantenido a lo largo de todos los estudios. Este modelo se muestra en la figura 2.

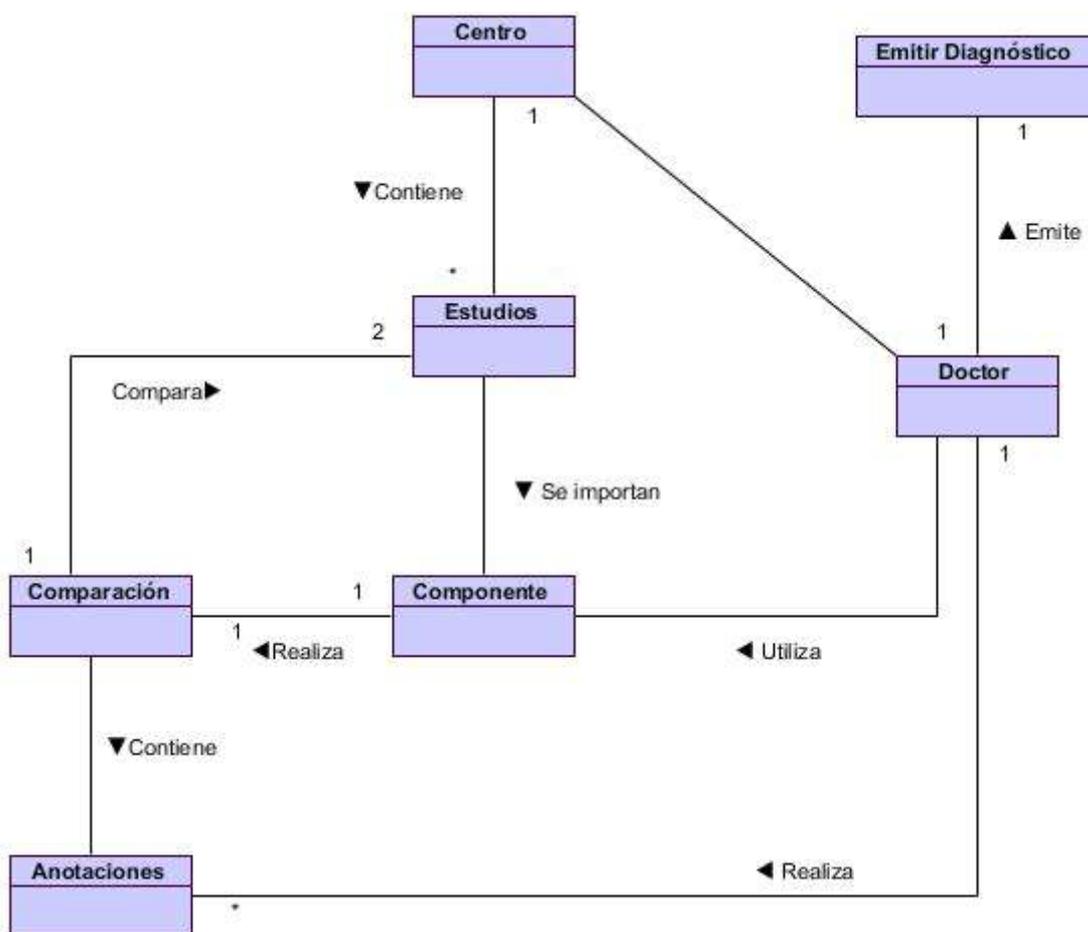


Figura 11- Modelo de dominio para la comparación de varios estudios. Fuente: Elaboración propia.

## 2.2. Descripción de conceptos en el modelo de dominio.

**Doctor:** Especialista encargado en analizar los estudios sobre imágenes de resonancia magnética en formato DICOM.

**Centro:** CNEURO es el Centro de Neurociencias de Cuba, en este se realizan los distintos estudios cerebrales del país. Abarcan desde la obtención de datos cerebrales de neuroimagen hasta las investigaciones referentes al tema para la detección de patologías en los pacientes.

**Estudio:** Son los estudios de resonancia magnética de un paciente. De cada paciente se realizan varios estudios anuales para llevar seguimiento de su estado y condición médica.

**Componente:** Es la herramienta utilizada para llevar a cabo la carga de los estudios y realizar las mediciones de las estructuras cerebrales.

**Comparación:** La comparación se realiza entre imágenes de varios estudios de un paciente.

**Anotaciones:** El especialista en neurología realiza anotaciones específicas que ayuden a llevar un seguimiento de las condiciones médicas del paciente analizado.

**Emitir Diagnóstico:** Al ser comparados los estudios de un paciente es emitido el diagnóstico supervisado por el especialista en neurología al frente del estudio.

### 2.3. Descripción del proceso de negocio.

La descripción de los procesos de negocio incluye información más amplia del procesos y de los elementos que intervienen, por ejemplo, personas que los ejecutarán funciones, datos e información. Con este se ofrece una comunicación más amplia posibilitando el entendimiento de los mismos para el desarrollo.(55) A continuación se hace la descripción del proceso de negocio para el desarrollo del componente.

Tabla 4 - Descripción del proceso de negocio. Fuente: Elaboración propia

<b>Objetivo</b>	Se describe el proceso de comparación de estudios a través de imágenes en el estándar DICOM de resonancia magnética.
<b>Evento(s) que lo genera(n)</b>	Para realizar la comparación de estudios primero se deben elegir las imágenes tomadas en distintas fechas.
<b>Pre condiciones</b>	Se selecciona un paciente del cual se van a tomar las imágenes
<b>Marco legal</b>	N/A
<b>Reglas del negocio</b>	
<b>Responsable</b>	Personal de salud
<b>Clientes internos</b>	Personal de salud
<b>Clientes externos</b>	N/A
<b>Entradas</b>	Neuroimágenes en el estándar DICOM
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1. Adicionar estudio	El Doctor importa en el componente de comparación los estudios que van a ser comparados.
2. Adicionar la cantidad de pantallas con imágenes a comparar	El Doctor adiciona en pantalla la cantidad de imágenes que va a necesitar de los estudios importados para poder realizar la comparación.
3. Realizar las mediciones de las zonas afectadas	El Doctor hace uso de la herramienta de medición para marcar el área afectada en ambos estudios para realizar el seguimiento de la situación del paciente.
4. Redactar las notas	

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

El Doctor realiza las anotaciones referentes a las observaciones detectadas durante la medición y comparación de la condición médica en ambos estudios.
5. Emitir diagnóstico A partir de la comparación de ambos estudios el doctor emite el diagnóstico en dependencia de los resultados obtenidos durante el proceso de análisis.
<b>Pos-condiciones</b>
1. Se realizaron las anotaciones a partir de los resultados de la comparación de estudios.
<b>Salidas</b>
1. Se emite el diagnóstico a partir de los criterios obtenidos en la comparación.
<b>Flujos paralelos</b>
N/A
<b>Pos-condiciones</b>
1. N/A
<b>Salidas</b>
1. N/A
<b>Flujos alternos</b>
1. N/A
<b>Pos-condiciones</b>
1. N/A
<b>Asuntos pendientes</b>
N/A

### 2.4. Modelado de negocio

El diagrama de procesos del negocio describe la lógica de los pasos de un proceso de negocio, especificando sus actividades, roles, las reglas y los objetos de datos que pueden ser de entrada o salida. Además, facilita el entendimiento entre clientes, desarrolladores y usuarios finales. En este se especifica el proceso para realizar las comparaciones de estudios realizados por el especialista de la salud en los estudios.

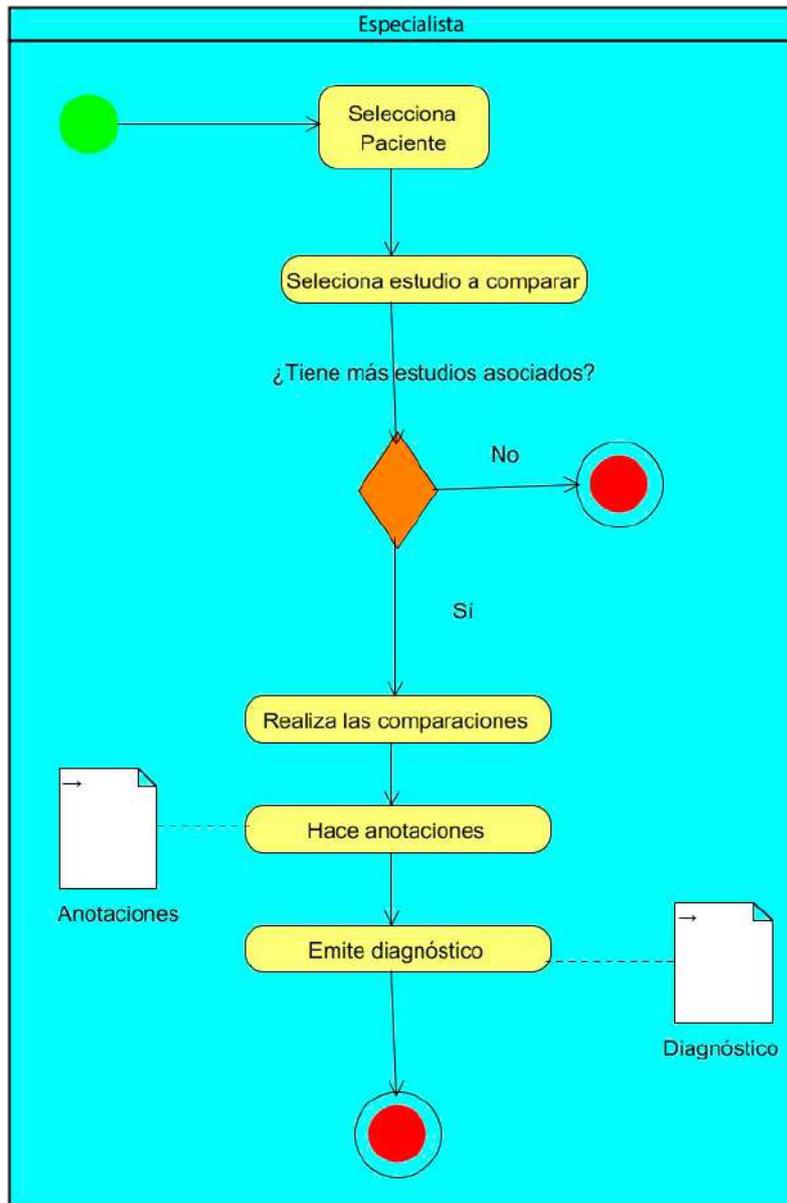


Figura 12 - Diagrama del proceso de negocio para hacer comparaciones. Fuente: Elaboración propia.

## 2.5. Especificación de Requisitos

A continuación, se muestra la tabla con los requisitos funcionales que responden al cumplimiento de las necesidades de cada una de las funcionalidades del software. Las descripciones detalladas de las funcionalidades que debe realizar el sistema, sus propiedades esenciales y con el objetivo de cumplir las necesidades de los usuarios finales a los que corresponden los requisitos del sistema.(56).

### 2.5.1. Requisitos funcionales

Unos de los requisitos funcionales de más alto nivel para este componente de comparación de estudios lo constituyen el ser capaz de cargar y redimensionar las imágenes en formato DICOM junto a la propiedad de poder trabajar con las distintas vistas de la misma. El resto de las funciones giran en torno a la realización de mediciones con las cuales se analiza cada uno de los elementos del estudio en cuestión.

Tabla 5 - Requisitos funcionales

<b>No.</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Prioridad</b>	<b>Complejidad</b>
<b>RF1</b>	<i>Importar imágenes</i>	El componente debe permitir importar imágenes hacia la pantalla de visualización	<i>Alta</i>	<i>Alta</i>
<b>RF2</b>	<i>Superponer imágenes</i>	El componente debe poder superponer imágenes para realizar comparaciones	<i>Alta</i>	<i>Alta</i>
<b>RF3</b>	<i>Cargar varias pantallas de imágenes a la vez</i>	El componente debe poder cargar varias pantallas con distintas imágenes a la vez	<i>Alta</i>	<i>Alta</i>
<b>RF4</b>	<i>Hacer zoom en las imágenes</i>	El componente debe permitir acercar y alejar distintas áreas de la imagen	<i>Alta</i>	<i>Alta</i>
<b>RF5</b>	Desplazar sobre las áreas de la imagen	El componente debe permitir que el usuario se desplace por distintas áreas de la imagen	Alta	Alta
<b>RF6</b>	Realizar anotaciones	El componente debe permitir al usuario redactar, guardar y exportar anotaciones sobre una imagen.	Alta	Alta
<b>RF7</b>	Cambiar Opacidad	El componente debe permitir al usuario cambiar opacidad de la imagen a fin de que se resalten algunas áreas	Alta	Alta
<b>RF8</b>	Sintonizar imágenes	El componente debe permitir al usuario posicionar varias imágenes a la vez en el mismo fotograma	Alta	Alta

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

<b>RF9</b>	Realizar seguimiento de estudios	El componente debe permitir al usuario marcar un área y que esta se seleccione a su vez en el resto de las pantallas que tienen imágenes cargadas	Alta	Alta
<b>RF10</b>	Trazar línea de medición	El componente debe permitir al usuario trazar las líneas de medición sobre la imagen y hacer que estas cambien de color una vez que se encuentren en un ángulo de 90 grados	Alta	Alta
<b>Rf11</b>	Trazar área de medición circular	El componente debe permitir al usuario trazar un área circular para realizar mediciones de un área definida y a su vez la capacidad de editar las dimensiones de dicha área circular.	Alta	Alta
<b>Rf12</b>	Girar la imagen	El componente debe permitir al usuario girar la imagen en distinta posición y en la dirección requerida	Alta	Alta
<b>RF13</b>	Modo Cine	El componente debe permitir al usuario hacer que las imágenes se desplacen de manera automática en cada uno de los paquetes de imagen DICOM	Alta	Alta
<b>RF14</b>	Desplazamiento manual sobre las imágenes	El componente debe permitir al usuario cambiar cada una de las imágenes de cada paquete DICOM mediante el rodillo del ratón.	Alta	Alta

Las tablas a continuación contienen la descripción de los requisitos obtenidos a partir del análisis.

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Tabla 4 - Descripción del requisito funcional “Trazar línea de medición”.

<b>Descripción textual</b>	El componente debe permitir al usuario trazar las líneas de medición sobre la imagen y hacer que estas cambien de color una vez que se encuentren en un ángulo de 90 grados
<b>Actores</b>	Especialista
<b>Precondición</b>	Cargar archivo DICOM en la herramienta
<b>Flujo de eventos</b>	
<b>Flujo básico “Trazar línea de medición”</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> selecciona la opción trazar línea</li> <li>2) El <b>Componente</b> activa la opción de trazado</li> <li>3) El <b>Especialista</b> traza la línea de medición</li> </ol>	
<b>Pos-condiciones</b>	
<ol style="list-style-type: none"> <li>1) Se traza la línea mostrando el tamaño de esta</li> </ol>	
<b>Flujos alternos</b>	
<b>Trazar alternativo “Trazar segunda línea”</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> traza una segunda línea</li> <li>2) El <b>Sistema</b> detecta si estas son perpendiculares y hace que cambien de color</li> </ol>	
<b>Interfaz</b>	

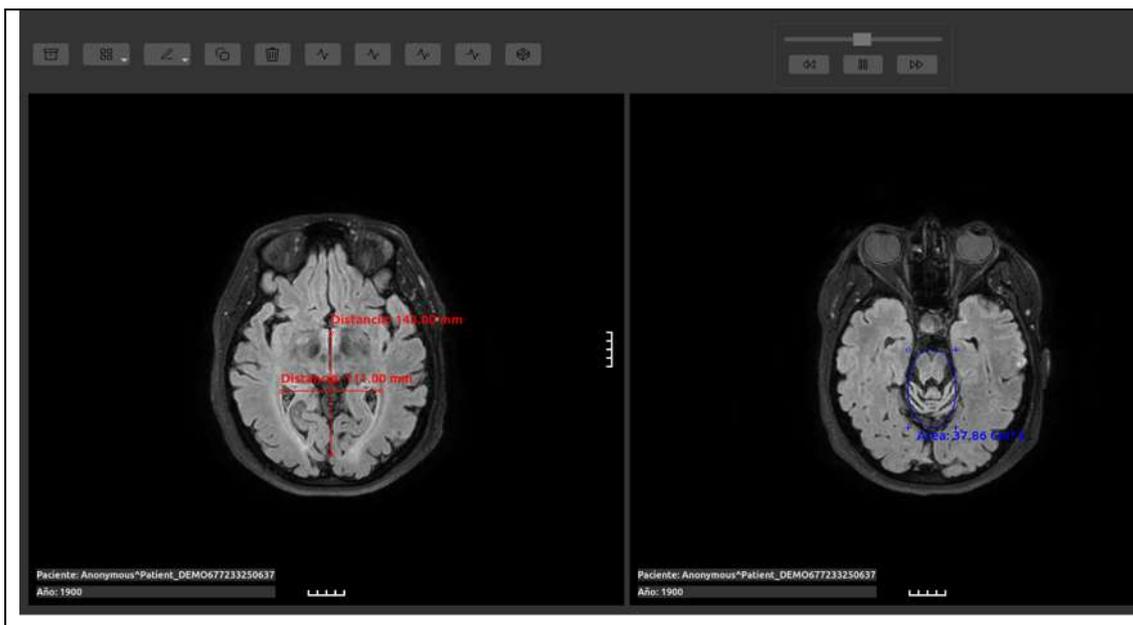


Tabla 6 - Descripción del requisito funcional “Importar Imágenes”.

<b>Descripción textual</b>	El componente debe permitir que se carguen los estudios y las imágenes contenidas en el mismo para realizar las mediciones
<b>Actores</b>	Especialista
<b>Precondición</b>	Tener el componente cargado
<b>Flujo de eventos</b>	
<b>Flujo básico “Importar imágenes”</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> selecciona la opción cargar estudios</li> <li>2) El <b>Especialista</b> selecciona el estudio dentro del buscador</li> <li>3) El <b>Especialista</b> selecciona la opción aceptar para cargar el estudio</li> </ol>	
<b>Pos-condiciones</b>	
<ol style="list-style-type: none"> <li>1) Se cargan los estudios en la aplicación</li> </ol>	
<b>Flujos alternos</b>	
<ol style="list-style-type: none"> <li>1) No se encuentra el estudio en la base de datos</li> <li>2) Pasar de estudio</li> </ol>	

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Tabla 7 - Descripción del requisito funcional “Cargar varias pantallas de imágenes a la vez”

<b>Descripción textual</b>	El componente debe permitir cargar varias pantallas de imágenes a la vez
<b>Actores</b>	Especialista
<b>Precondición</b>	Tener el componente cargado
<b>Flujo de eventos</b>	
<b>Flujo básico “Cargar varias pantallas de imágenes a la vez”</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> selecciona la opción de cargar varias pantallas de imágenes.</li> <li>2) El <b>Especialista</b> selecciona la cantidad de pantallas que desea que se muestren en el componente.</li> <li>3) El <b>Especialista</b> selecciona cada una de las imágenes que quiere que se muestren</li> </ol>	
<b>Pos-condiciones</b>	
<ol style="list-style-type: none"> <li>1) Se cargan las distintas pantallas en el componente</li> </ol>	
<b>Flujos alternos</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> selecciona nuevamente la opción de cargar distintas pantallas.</li> <li>2) El <b>Especialista</b> selecciona mostrar una sola pantalla.</li> </ol>	

Tabla 8 - Descripción del requisito funcional “Sintonizar imágenes”

<b>Descripción textual</b>	El componente debe permitir sintonizar distintos estudios en la misma imagen de la serie
<b>Actores</b>	Especialista
<b>Precondición</b>	Tener el componente cargado con varios estudios en la pantalla
<b>Flujo de eventos</b>	
<b>Flujo básico “Sintonizar imágenes”</b>	
<ol style="list-style-type: none"> <li>1) El <b>Especialista</b> selecciona la opción sintonizar estudios</li> <li>2) El <b>Sistema</b> sintoniza los estudios</li> </ol>	

<b>Pos-condiciones</b>
1) Se sintonizan los estudios
<b>Flujos alternos</b>
1) No se encuentran las imágenes a sintonizar.

### 2.5.2. Requisitos no funcionales

#### RNF1- Requisito no funcional de Eficiencia

El componente debe ser capaz de cargar los estudios a analizar en un plazo inferior a 2 segundos junto con cada una de las transacciones que se efectuaran para hacer los cambios de imagen y los trazados de mediciones.

#### Características de hardware:

CPU: Intel Core 2 DUO 1.8 GHz

RAM: 4GB o superior

Espacio en Disco: 80GB

#### RNF2- Requisito de portabilidad

El componente se adapta de manera correcta a distintas resoluciones de pantalla, en las cuales no sufren distorsiones tanto el acceso a sus distintos botones como la información en formato de imagen DICOM que se muestra en cada una de sus pantallas. Estas resoluciones de pantallas abarcan desde 1024x720 hasta 1920x1080.

#### RNF3- Requisito no funcional de Apariencia o Interfaz

El componente muestra una interfaz simple y amigable para su uso por el especialista. Utiliza colores en tono de grises para mantener centrada la atención en la información precisada en las imágenes que se muestran.

### 2.6. Definición de los actores.

En la ingeniería de diseño de un software un actor representa a cualquier ente externo que interactúa con el sistema, pueden ser humanos u otros sistemas. En este caso se presenta un solo actor el cual va a acceder a las funcionalidades del componente.

**Actor:** Especialista.

**Descripción:** Médico de la rama de las neurociencias u aplicable a otra especialidad que interactúe con el componente informático para la comparación de estudios de imagen DICOM de resonancia magnética.

### 2.7. Análisis y diseño

El modelo de diseño constituye el conjunto de diagramas que describen el diseño lógico de un sistema. Comprende los diagramas de clases de software, diagramas de interacción, como quiere el modelador.

#### 2.7.1. Arquitectura de software.

Para la solución informática propuesta y según las tecnologías utilizadas se define utilizar la arquitectura Modelo Vista Controlador (MVC). Debido a que el componente se programó utilizando el lenguaje Python y utilizando del mismo las librerías *pydicom* para el trabajo con imágenes DICOM estableciendo un modelo con los datos del mismo archivo y *pyQt* para la creación de las vistas y el manejo de los componentes de la misma a partir del controlador que se utiliza en la librería.(57, 58)

La arquitectura Modelo-Vista-Controlador (MVC) es un patrón de diseño de software ampliamente utilizado en el desarrollo de aplicaciones. (58)Esta arquitectura se divide en tres componentes principales: el modelo, la vista y el controlador, cada uno con una responsabilidad específica.

El componente del modelo es responsable de representar los datos y la lógica de negocios de la aplicación. Encapsula los datos y la lógica de la aplicación para que el resto de la aplicación pueda acceder a ellos de manera consistente. El modelo también puede contener métodos para realizar operaciones en los datos y para validar los datos antes de almacenarlos o recuperarlos.(59)

El componente de la vista es responsable de mostrar los datos al usuario y de proporcionar una interfaz para que el usuario interactúe con la aplicación. La vista puede ser cualquier cosa, desde una página web hasta una interfaz de línea de comandos. La vista no realiza ninguna operación en los datos, solo muestra los datos al usuario y envía las acciones del usuario al controlador.(59)

El componente del controlador actúa como intermediario entre el modelo y la vista. El controlador recibe las acciones del usuario desde la vista y las procesa. El controlador interactúa con el modelo para obtener o actualizar los datos y luego actualiza la vista con los datos actualizados. (58)

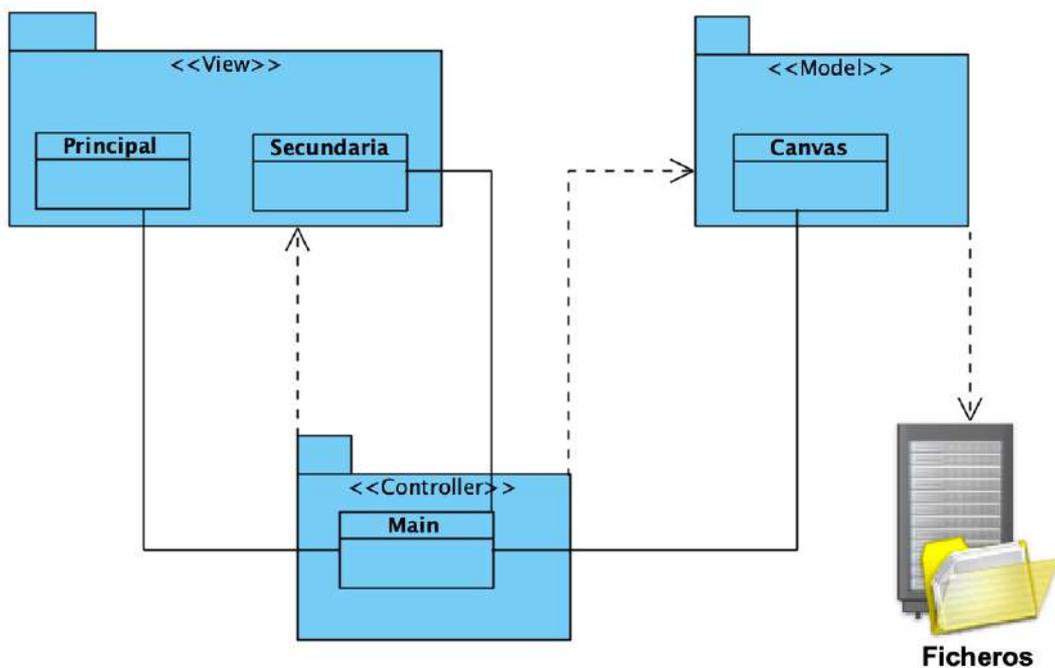


Figura 13.- Modelo de arquitectura a utilizar. Fuente: Elaboración propia.

### 2.7.2. Descripción de la arquitectura.

La arquitectura propuesta para el desarrollo de la solución informática permite un mejor entendimiento de cada uno de los componentes que estructuran al componente de comparación de estudios. Partiendo desde una vista principal la cual muestra y contiene las opciones de las funcionalidades. El elemento *main* o controlador que asegura la llamada y ejecución de cada uno de los métodos y finalmente el Canvas que es el modelo de datos de este software donde se almacena la información de la imagen y devuelve los datos visuales y cálculos de cada una de las operaciones a realizar. Para

esta herramienta los datos se obtienen alojados en ficheros que contienen las imágenes DICOM.

### 2.7.3. Diagrama de clases.

El diagrama de clases UML permite etiquetar las clases y las asociaciones entre estas clases mediante el uso de estereotipos. El diagrama de clases UML de la propuesta de solución representa las clases a utilizar en el desarrollo del componente de comparación, con sus atributos y métodos, junto con las asociaciones entre dichas clases.

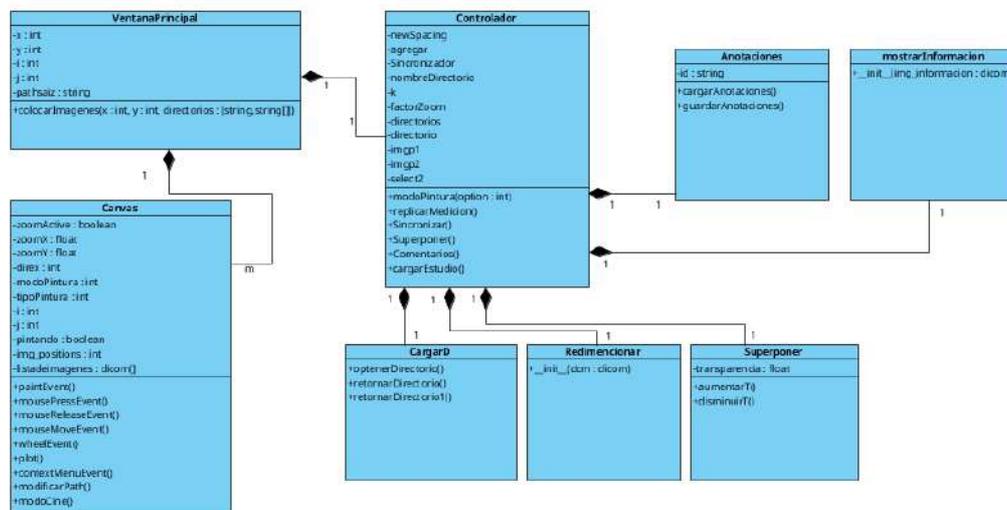


Figura 14 - Diagrama de clases del sistema. Fuente: Elaboración propia

En la figura anterior se modelan las clases existentes en el código. Se muestran las relaciones entre estas clases y las dependencias existentes entre ellas, las cuales concuerdan con los requisitos propuestos en la solución informática.

### 2.7.4. Diagrama de secuencia.

Buscando un mayor entendimiento de las actividades que se llevan a cabo en las descripciones de requisitos, se realizaron los diagramas de secuencia del diseño, los cuales permiten representar las interacciones entre ellos en orden temporal. En la figura 4 se muestra el diagrama de secuencia del diseño referente al requisito trazar línea de medición donde la clase principal realiza las llamadas al controlador y este a su vez actúa sobre los métodos contenidos en la clase Canvas que representa al modelo.(60)

#### 2.7.4.1. Diagrama de secuencia trazar línea de medición.

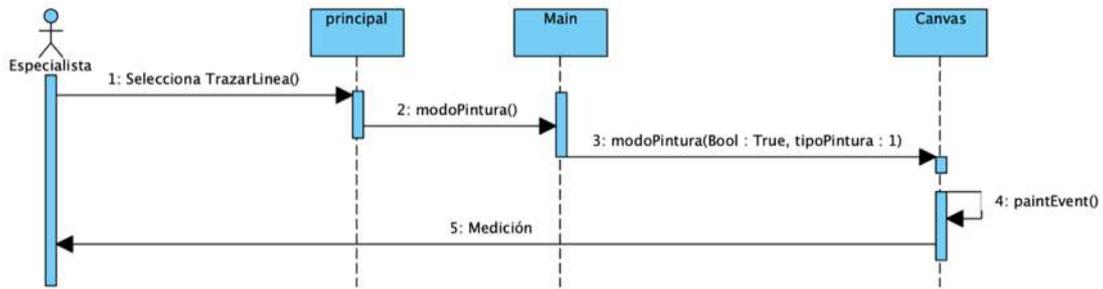


Figura 15 - Diagrama de secuencia requisito “trazar líneas de medición”. Fuente: Elaboración Propia

#### 2.7.4.2. Descripción del diagrama de secuencia

El especialista debe seleccionar la opción Trazar línea de medición, para que el sistema permita trazar una línea con la cual es posible medir la estructura seleccionada. Esta función pedirá a la clase controladora que ejecute el método encontrado en el Canvas para poder realizar el trazado y medición.

- a. El especialista selecciona la opción **TrazarLinea()** en la ventana principal correspondiente a la vista.
- b. La ventana principal realiza la solicitud de la opción **modoPintura()** hacia la clase controladora.
- c. La clase controladora solicita el **modoPintura()** al Canvas pasando por parámetros el atributo True para validar la opción y el atributo tipo para seleccionar el tipo de medición que se realizará.
- d. La clase Canvas ejecuta el método **paintEvent()** encargado del trazado de la línea de medición y posteriormente el cálculo de la distancia que ocupa dicha medición.
- e. El Canvas muestra en la ventana principal y envía el valor obtenido durante la realización de esta.

### 2.7.4.3. Diagrama de secuencia trazar área de medición circular.

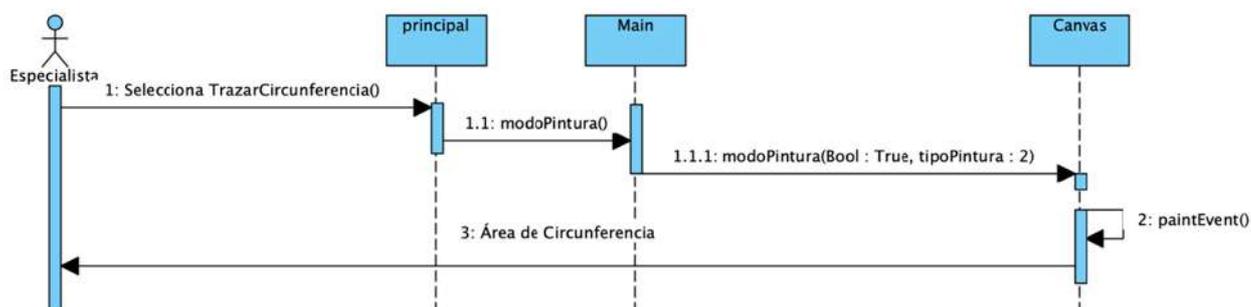


Figura 16 - Diagrama de secuencia requisito “trazar área de medición circular”. Fuente: Elaboración propia.

### 2.7.4.4. Descripción diagrama de secuencia.

El especialista debe seleccionar la opción Trazar Circunferencia, para que el sistema permita trazar una circunferencia para calcular el área de la estructura seleccionada. Esta función pedirá a la clase controladora que ejecute el método encontrado en el Canvas para poder realizar el trazado y cálculo de área.

- El especialista selecciona la opción **TrazarCircunferencia()** en la ventana principal correspondiente a la vista.
- La ventana principal realiza la solicitud de la opción **modoPintura()** hacia la clase controladora.
- La clase controladora solicita el **modoPintura()** al Canvas pasando por parámetros el atributo True para validar la opción y el atributo tipo para seleccionar el tipo de medición que se realizará.
- La clase Canvas ejecuta el método **paintEvent()** encargado de trazar la circunferencia de medición y posteriormente el cálculo del área de misma.
- El Canvas muestra trazada la circunferencia en la ventana principal y el valor obtenido a través del cálculo de área.

### 2.7.4.5. Diagrama de secuencia importar imágenes.

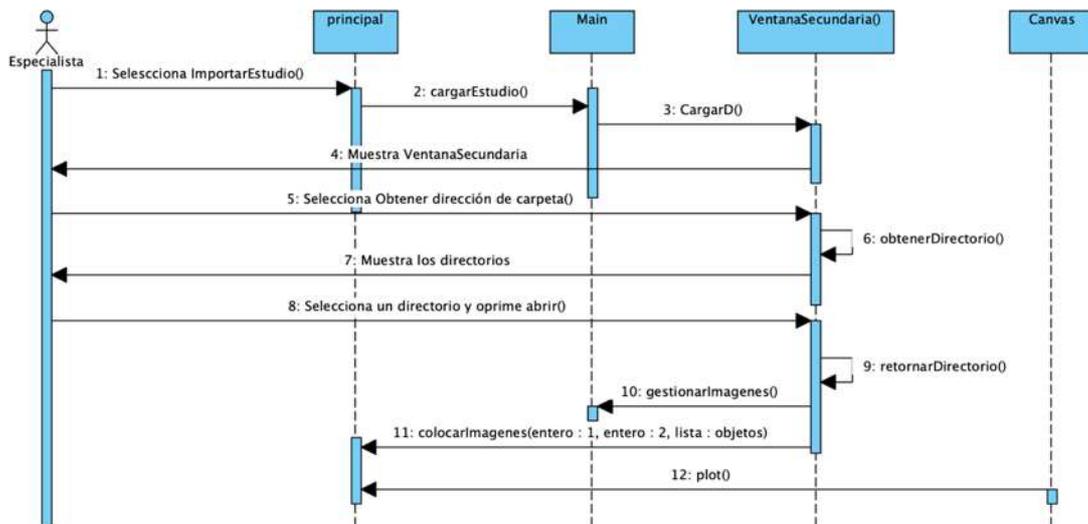


Figura 17- Diagrama de secuencia del requisito "importar imágenes". Fuente: Elaboración propia.

### 2.7.4.6. Descripción diagrama de secuencia.

El especialista debe seleccionar la opción importar imágenes la cual le va mostrar la ventana para seleccionar el directorio que quiere importar de las imágenes. Esta función se encarga de listar cada uno de los datos incluidas las imágenes que se encuentran contenidos en el estudio.

- a) El especialista selecciona la opción **ImportarEstudio()** en la ventana principal del componente.
- b) La ventana principal realiza la solicitud **cargarEstudio()** a la clase controladora.
- c) La clase controladora solicita **cargarD** a la clase **VentanaSecundaria()** se muestra la ventana secundaria.
- d) El especialista selecciona la opción **Obtener dirección de carpeta()** a la ventana Secundaria.
- e) La ventana secundaria ejecuta **ObtenerDirectorio()** y muestra al especialista los directorios de los estudios.
- f) El especialista selecciona un directorio y oprime en la opción **abrir()**.
- g) La ventana Secundaria ejecuta **retornarDirectorio()** y solicita al **main gestionarImagenes()**.
- h) La ventana secundaria solicita a la clase principal el método **colocarImagenes()**.
- i) La clase Canvas pinta las imágenes en la ventana Principal.

### 2.7.5. Diagrama de componentes

El Diagrama de Componentes incluye como los elementos deben empaquetarse para su despliegue a manera de entidades independientes y ejecutables. Representa las estructuras físicas del código siendo utilizado para estructurar el Modelo de implementación a manera de subsistemas y modelar la vista estática del sistema, describiendo sus elementos físicos a manera de relaciones entre estos. Representa el sistema dividido en componentes y evidencia la organización y las dependencias lógicas entre estos. En la figura 18 se muestra el diagrama de componentes que representa la estructura de la herramienta para comparación de estudios.

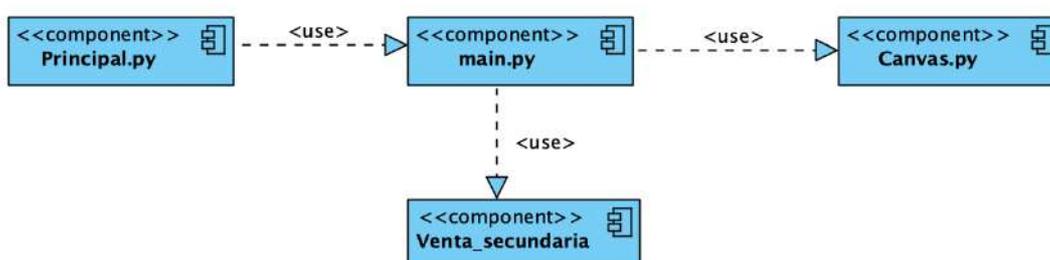


Figura 18- Diagrama de componentes para la herramienta de comparación de estudios.  
Fuente: Elaboración Propia.

Los componentes que componen la herramienta para comparación de estudios son:

- **Principal.py:** Este componente muestra la interfaz principal del software. Es el que contiene los botones para utilizar las funcionalidades del software.
- **Main.py:** En este componente se encuentra implementado el controlador de la herramienta. Se encarga de realizar todas las operaciones del software.
- **Canvas.py:** Este componente contiene el modelo de datos y todas las funcionalidades para el trabajo con las imágenes de resonancia magnética.
- **Ventana\_secundaria.py:** Este componente se encarga de hacer la búsqueda de los estudios en los ficheros del ordenador.

### Conclusiones del capítulo

Luego del análisis y diseño de la herramienta para la comparación de estudios sobre imágenes de resonancia magnética se obtuvieron los siguientes resultados:

- Se proponen los requisitos funcionales de la herramienta para la comparación de estudios sobre imágenes de resonancia magnética en formato DICOM, entre

## CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

estos se destacan trazar líneas de medición, hacer cálculos de área y replicar mediciones en distintos estudios.

- Se seleccionó la arquitectura Modelo-Vista-Controlador pues muestra de manera más organizada la presencia de cada uno de los componentes de la herramienta. Para esta se tienen todas las interfaces en el paquete de vistas, la clase principal en el controlador y en el modelo la estructura del estudio DICOM cargados.
- Se define el diagrama de clases para un mayor entendimiento de la estructura del componente. Los diagrama de secuencia para representar el flujo de llamadas de los métodos y el diagrama de componentes para definir la estructura de software de la herramienta informática.

### 3. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo se describe la implementación de la herramienta para comparación de estudios sobre imágenes DICOM de resonancia magnética, se define el estándar de código utilizado como notación para la declaración de métodos y variables. Se describen las pruebas utilizadas para evaluar el correcto funcionamiento de la herramienta y se analizan sus resultados. Se realiza la validación de los requisitos no funcionales, se valida la precisión del algoritmo de medición en comparación a la herramienta homóloga Horos.

#### 3.1. Estándar de Codificación

Los estándares de codificación tienen una función principal para el entendimiento de la codificación y en la utilización de cada uno de los métodos y variables; aumentando así la uniformidad del código correspondiente a la herramienta a desarrollar. Para este componente se utilizó el siguiente estándar de codificación.(61)

- Utilizar nombres descriptivos para los nombres de las clases o funciones, propiedades y métodos.
- Se utiliza la notación *CamelCase* para declarar los nombres de los atributos de las clases con letra inicial minúscula. Si es un nombre compuesto se utiliza minúsculas y mayúsculas.
- Se utiliza la notación *PascalCase* para declarar los nombres de los métodos, clases y valores enumerables en mayúscula. Si son palabras compuestas se escribe mayúsculas al inicio de cada palabra.
- Escribir comentarios en las clases y en los métodos, así como a las instrucciones, que sean de mediano a alto nivel de complejidad.

#### 3.2. Implementación.

La fase de implementación muestra los pseudocódigos que corresponden a los algoritmos para trazar línea de medición, replica de mediciones y cargar varias vistas. Se realiza la descripción de los procesos reflejados para entender su funcionamiento.

### 3.2.1. Implementación del requisito "Trazar línea de medición".

En la figura 19 se muestra el pseudocódigo:

```

1  if self.modopintura :
2
3      painter = QPainter(self)
4
5      if self.start_pos is not None and self.end_pos is not None:
6
7
8          pen = self.pen
9
10
11         pen.setWidth(1)
12         painter.setPen(pen)
13         painter.drawLine(self.start_pos, self.end_pos)
14         painter.drawEllipse(self.start_pos,2,2)
15         painter.drawEllipse(self.end_pos,2,2)
16
17         if self.start_pos2 is not None and self.end_pos2 is not None:
18             pendientes = 3
19             if self.start_pos.x() < self.end_pos.x()+2 and self.start_pos.x() > self.end_pos.x()-2:
20                 if self.start_pos2.y() < self.end_pos2.y()+2 and self.start_pos2.y() > self.end_pos2.y()-2:
21                     pendientes = -1.000
22                 elif self.start_pos.y() < self.end_pos.y()+2 and self.start_pos.y() > self.end_pos.y()-2:
23                     if self.start_pos2.x() < self.end_pos2.x()+2 and self.start_pos2.x() > self.end_pos2.x()-2:
24                         pendientes = -1.000
25                 elif self.start_pos2.y() < self.end_pos2.y()+2 and self.start_pos2.y() > self.end_pos2.y()-2:
26                     if self.start_pos.x() < self.end_pos.x()+2 and self.start_pos.x() > self.end_pos.x()-2:
27                         pendientes = -1.000
28                 elif self.start_pos2.x() < self.end_pos2.x()+2 and self.start_pos2.x() > self.end_pos2.x()-2:
29                     if self.start_pos.y() < self.end_pos.y()+2 and self.start_pos.y() > self.end_pos.y()-2:
30                         pendientes = -1.000
31             else:
32                 pendientes = (self.end_pos.y()-self.start_pos.y())/(self.end_pos.x()-self.start_pos.x())*
33                     ((self.end_pos2.y()-self.start_pos2.y())/(self.end_pos2.x()-self.start_pos2.x()))
34
35
36             if pendientes > -1.015 and pendientes < -0.985 :
37
38                 self.pen = QPen(Qt.red)
39             else:
40                 self.pen = QPen(QColor(255, 165, 0))
41             pen = self.pen
42             painter.setPen(pen)
43             painter.drawLine(self.start_pos2, self.end_pos2)
44             painter.drawEllipse(self.start_pos2,2,2)
45             painter.drawEllipse(self.end_pos2,2,2)
46             distance2 = np.sqrt((self.start_pos2.x() - self.end_pos2.x())**2 +
47                 (self.start_pos2.y() - self.end_pos2.y())**2)
48
49             # Calcular la distancia en milímetros
50
51             distance_mm2 = distance2 * self.pixel_spacing_x
52             painter.drawText(QPoint(self.start_pos2.x(), self.start_pos2.y() - 10),
53                 f"Distancia: {distance_mm2:.2f} mm")
54
55             distance = np.sqrt((self.start_pos.x() - self.end_pos.x())**2 +
56                 (self.start_pos.y() - self.end_pos.y())**2)
57
58             # Calcular la distancia en milímetros
59
60             distance_mm = distance * self.pixel_spacing_x
61             painter.drawText(QPoint(self.start_pos.x(), self.start_pos.y() - 10),
62                 f"Distancia: {distance_mm:.2f} mm")

```

Figura 19 - Fragmento de código "Trazar línea de medición". Fuente: Elaboración propia

En este algoritmo implementado primero se verifica que el modo pintura esté activo, luego se comprueba que tipo de dibujo se hará, si es círculo o de tipo lineal. Posteriormente nos aseguramos de que las variables necesarias para trazar la línea ya hayan sido inicializadas para comenzar a dibujar el segmento. Finalmente se realiza el cálculo de distancia basado en los puntos recolectados en los eventos del mouse y se incorporan a la imagen junto con el valor de distancia calculado multiplicado por la variable *PixelSpacing* obtenida del archivo DICOM y de la relación de aspecto proporcional de la pantalla respecto a la imagen.

### 3.1.2. Implementación del requisito funcional “Replicar mediciones”.

En la figura 20 se muestra el pseudocódigo:

```

1 def replicarMedicion(self):
2     if self.VP.i is not None and self.VP.j is not None:
3         x = self.VP.x
4         y = self.VP.y
5         for i in range(x):
6             for j in range(y):
7
8
9                 self.VP.grid_layout.itemAtPosition(i,j).widget().line = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().line
10                self.VP.grid_layout.itemAtPosition(i,j).widget().start_pos = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().start_pos
11                self.VP.grid_layout.itemAtPosition(i,j).widget().end_pos = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().end_pos
12                self.VP.grid_layout.itemAtPosition(i,j).widget().start_pos2 = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().start_pos2
13                self.VP.grid_layout.itemAtPosition(i,j).widget().end_pos2 = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().end_pos2
14                self.VP.grid_layout.itemAtPosition(i,j).widget().modoPintura = True
15
16
17                self.VP.grid_layout.itemAtPosition(i,j).widget().circle = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().circle
18                self.VP.grid_layout.itemAtPosition(i,j).widget().startCircle = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().startCircle
19                self.VP.grid_layout.itemAtPosition(i,j).widget().largo = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().largo
20                self.VP.grid_layout.itemAtPosition(i,j).widget().ancho = self.VP.grid_layout.itemAtPosition(self.VP.i,self.VP.j).widget().ancho
21
22                self.VP.grid_layout.itemAtPosition(i,j).widget().update()
23
24                pass

```

Figura 20 - Fragmento del código "Replicar mediciones". Fuente: Elaboración propia

Este algoritmo obtiene las variables de los puntos iniciales y finales de las líneas de medición de la vista seleccionada. Actualiza las variables en el resto de los Canvas de cada vista. Luego traza las líneas utilizando los valores obtenidos de los puntos iniciales y finales de la vista seleccionada.

### 3.1.3. Implementación del requisito “Cargar varias pantallas”

En la figura 21 se muestra el pseudocódigo:

```

1 def numeroVentanas(self,num):
2     while self.VP.grid_layout.count():
3         item = self.VP.grid_layout.takeAt(0)
4         widget = item.widget()
5         if widget is not None:
6             widget.deleteLater()
7             print("Borrando widget")
8         if num>2:
9             self.VP.colocarImagenes(2,int(num/2),self.directorios)
10        else:
11            self.VP.colocarImagenes(1,num,self.directorios)
12
13        pass

```

Figura 21- Fragmento de código "numeroVentanas". Fuente: Elaboración propia

En este fragmento de código se toma como parámetro el número de ventanas que se quieren mostrar, a partir de este se eliminan las ventanas y se llama al método colocar imágenes de *Principal.py*. En la figura 22 se muestra el pseudocódigo del método *colocarImagenes()*.

```

1  def colocarImagenes(self,x,y,directorios):
2
3      self.x = x
4      self.y = y
5      medidas = self.grid_layout.sizeHint()
6      ancho = medidas.height() / x
7      largo = medidas.width() / y
8      for i in range(x):
9          for j in range(y):
10             ventana_imagen = self.Canvas(self,directorios,i,j)
11             self.grid_layout.addWidget(ventana_imagen,i,j)
12
13         self.adjustSize()
14         self.setWindowState(Qt.WindowMaximized)
15
16     pass
17

```

Figura 22- Fragmento de código "colocarImagenes". Fuente: Elaboración propia

Este método recibe la distribución de las vistas, las muestra y distribuye el tamaño para que abarquen el mismo espacio en la pantalla principal de la herramienta. Carga una serie distinta en cada vista y permite cambiar entre las vistas Axial, Sagital y Coronal contenidas en el estudio de resonancia magnética según sea seleccionada.

### 3.2. Pruebas Funcionales.

Las pruebas funcionales se centran en revisar si un programa, sistema o aplicación cumple exitosamente con cada uno de los requisitos funcionales planteados para su desarrollo y las tareas específicas en cuanto a su diseño. Estas se realizan desde una perspectiva fuera de su programación centrándose en la funcionalidad que se puede observar a medida que el usuario va utilizando cada una de las opciones. Esto incluye la interacción interfaz usuario, las respuestas entradas por el usuario y las salidas mostradas por el software.

#### 3.2.1. Pruebas de Caja Negra.

Las pruebas de Caja Negra son técnicas de pruebas de software empleadas para comprobar la funcionalidad del software sin tomar en cuenta la estructura interna del código o algún otro detalle en la implementación u otros escenarios de ejecución. Para la elaboración y planificación de estas es necesario conocer cuales son las especificaciones y requerimientos funcionales del software de ahí comparar las respuestas que arroja el software respecto al resultado esperado en cada uno de los casos de pruebas. Para la presente investigación se tomaron dos estudios del banco de datos proporcionados por CNEURO con los cuales se realizaran las mediciones para detectar las posibles anomalías que puedan ocurrir y la exactitud necesaria en las mediciones. Además en estas pruebas se determinan los posibles falsos positivos y

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

falsos negativos que puedan arrojar estas pruebas y a partir de estas comenzar con las correcciones en las funcionalidades del software.(62, 63).

A partir de seleccionada la muestra de imágenes a utilizar para las pruebas se toma para constatar la exactitud de los resultados se utilizó de referencia los resultados arrojados por la herramienta eFilm Workstation por ser de las más exactas en las mediciones junto con las comparaciones de las medidas reales tomadas por los especialistas del Centro de Neurociencia de Cuba (CNEURO).(64, 65)

### 3.2.2. Diseño de Casos de Pruebas.

Las pruebas de software se encargan de probar la existencia de errores en la programación de un software. En este caso se aplica la técnica de partición de equivalencia. Por tanto, se trata de encontrar ciertos datos de entrada, casos de pruebas, normalmente en un conjunto pequeño de ellos, que aseguren una confianza aceptable. Siguiendo esta técnica de particiones de equivalencia, se consiguen casos de prueba dividiendo el espacio completo de cada entrada en una serie de bloques, llamados particiones o bloques de equivalencia, que contienen valores para los que el programa a probar se comporta de manera idéntica. Entonces, tomando un único dato como representante de cada bloque se tiene el mismo grado de confianza que se obtendría probándolo para todos los valores del mismo.(66)

Tabla 9 - Diseño de casos de pruebas

Casos de pruebas				
Escenario	Descripción	Campo	Respuesta del Sistema	Resultado de la prueba
1.1 Trazar línea de medición correctamente	El usuario traza una línea de medición	Trazar línea	Se traza una línea azul de medición que muestra la distancia entre sus puntos.	Satisfactoria
1.2 Trazar línea de medición fuera de área	El usuario intenta trazar una línea de medición fuera del área de la imagen	Trazar línea	No se traza la línea fuera del área.	Satisfactoria
1.3 Trazar segunda línea fuera de ángulo	El usuario traza la segunda línea de medición fuera de ángulo	Trazar línea	Ambas líneas se mantienen de color azul indicando que no se cortan de forma perpendicular	Satisfactoria

1.4	Trazar segunda línea en el ángulo de 90 grados	El usuario traza la segunda línea de medición en un ángulo de 90 grados	Trazar línea	Ambas líneas se vuelven de color rojo indicando que se cortan de manera perpendicular	Satisfactoria
-----	--	---	--------------	---	---------------

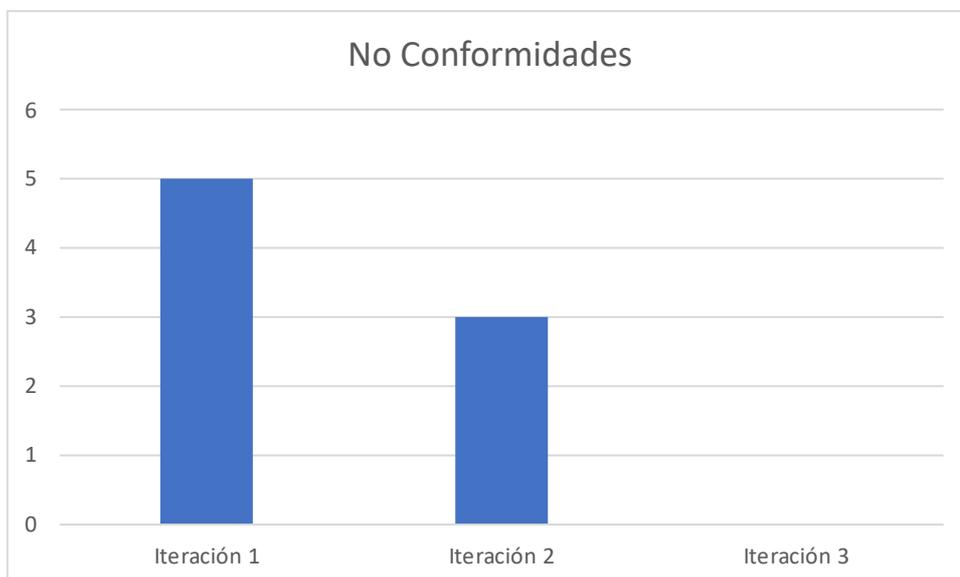


Figura 23- Resultados de los casos de prueba

### 3.3. Pruebas de Caja Blanca.

Las pruebas de caja blanca basan su diseño en el a partir del control procedimental para derivados. Mediante estas pruebas el desarrollador puede obtener casos de pruebas directamente en cada una de las funcionalidades mediante la ejecución por lo menos de una vez de cada una de las sentencias del código implementado en el software. Se centra en el estudio minucioso de las sesiones operativas considerando los detalles procedurales para la lógica del sistema. En estos se garantizan todos los caminos independientes de las llamadas de cada uno de los métodos presentes en el programa. Se demuestran mediante las respuestas verdaderas o falsas generadas en las llamadas de cada uno de los casos de pruebas. Para esta se seleccionó la prueba del camino básico.(67)

#### 3.3.1. Prueba del camino básico.

La prueba del camino básico es una técnica de prueba de Caja Blanca. Utilizando esta se obtienen medidas de complejidad lógica de un diseño y usar esta medida como una guía para la definición de un conjunto básico de los posibles caminos. En este caso en particular se hará para controlar que cada uno de los componentes llamados desde la

clase principal se ejecuten de forma tal que se ejecuten de manera correcta sin tener caídas o que los mismos no sean encontrados.(67)

Para la realización de esta prueba se utilizó la librería *Pytest* utilizada para las pruebas unitarias y para la prueba aplicada, camino básico. Esta prueba consta de programación orientada al seguimiento de las llamadas de los métodos entre la vista, el modelo y el controlador de tal forma que se cumpla el flujo de llamada entre los componentes que componen la aplicación. En la figura 24 se muestra el resultado de las pruebas

```

1 import apport
2 import pytest
3
4 from main import Controlador
5 from Principal import VentanaPrincipal
6 from PyQt5.QtWidgets import QApplication
7 from TablaDeInformacion import mostrarInformacion as tablaInformacion
8 from Ventana_secundaria import CargarD as VentanaSecundaria
9 from Sincronizador import Sincronizador
10 from model.Canvas import Canvas
11 import sys
12 import os
13
14 def initialize_controlador():
15     app = QApplication(sys.argv)
16     Model = Canvas
17     Control= Controlador
18     ventana_principal = VentanaPrincipal(Model,Control)
19     ventana_principal.show()
20
21     return ventana_principal
22
23 def test_modo_pintura():
24     controlador = initialize_controlador()
25
26     controlador.i = 0
27     controlador.j = 0
28     controlador.Controlador.modoPintura(1)
29     assert controlador.grid_layout.itemAtPosition(0,0).widget().modoPintura == True
30     assert controlador.grid_layout.itemAtPosition(0,0).widget().tipoPintura == 1
31
32
33 def test_modo_velocidadCine():
34     controlador = initialize_controlador()
35     controlador.i = 0
36     controlador.j = 0
37     controlador.Controlador.velocidadCine()
38     assert controlador.grid_layout.itemAtPosition(0,0).widget().velocidad == controlador.VP.slider.value * 10
39
40 if __name__ == "__main__":
41     pytest.main(["-s", "--html=report.html"])
42

```

Figura 24- Prueba del camino básico a la funcionalidad modoPintura(). Fuente:  
Elaboración propia

Una vez realizada la prueba los resultados obtenidos son satisfactorios. Esta fue la respuesta esperada como resultado de la prueba demostrando que las conexiones y llamadas de los elementos de la herramienta responden correctamente entre ellos.

### 3.4. Validación requisitos no funcionales.

La validación de los requisitos no funcionales de la solución informática se realizó a partir de las propuestas en el capítulo anterior. Una vez concluida la fase de implementación y pruebas de software se hicieron pruebas en distintos dispositivos a fin de garantizar el correcto cumplimiento.

**Validación RNF1. Eficiencia del componente.**

Para realizar esta validación se utilizaron tres dispositivos de cómputo con los recursos propuestos en la descripción del requisito. De estos, dos eran equipos físicos y una máquina virtual. En estos casos solo difería la resolución de pantalla. En la figura 25 se muestra el resultado obtenido del tiempo de ejecución de este requisito.

```

1  def colocarImagenes(self,x,y,directorios):
2
3      tiempoInicial = time.time()
4
5      self.x = x
6      self.y = y
7      medidas = self.grid_layout.sizeHint()
8      ancho = medidas.height() / x
9      largo = medidas.width() / y
10     for i in range(x):
11         for j in range(y):
12             ventana_imagen = self.Canvas(self,directorios,i,j)
13             self.grid_layout.addWidget(ventana_imagen,i,j)
14
15     tiempoFinal = time.time()
16
17     # Calcular la diferencia de tiempo en segundos y milisegundos
18     diferencia_segundos = tiempoFinal - tiempoInicial
19     diferencia_milisegundos = (tiempoFinal - tiempoInicial) * 1000
20
21     print("El proceso tomó", diferencia_segundos, "segundos")
22     print("El proceso tomó", diferencia_milisegundos, "milisegundos")
23
24
25
26     self.adjustSize()
27     self.setWindowState(Qt.WindowMaximized)
28
29     pass

```

Figura 25- Fragmento de código referente a la prueba de tiempo de ejecución. Fuente: Elaboración propia.

Tabla 10 – Validación del requisito no funcional de eficiencia.

Computadoras	Tiempo estudio 1	Tiempo estudio 2	Tiempo estudio 3
Acer Procesador: Intel Pentium RAM: 4GB Disco:	1.3	1.5	1.4
MacBook Pro Procesador: i7 4ta RAM: 16GB Disco: 80GB	0.7	1	0.9
Máquina virtual Procesador: Intel Pentium	1.7	1.8	1.6

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

RAM: 4GB			
Disco: 80GB			

Una vez concluidas las pruebas se obtiene que los 3 equipos de cómputo obtuvieron resultados distintos en cuanto a la eficiencia en que se cargaron los estudios. En todos los casos ninguno obtuvo un tiempo mayor a 2 segundos. Se valida de manera satisfactoria el requisito

### Validación RNF3. Apariencia o Interfaz del componente.

La herramienta cumple con las normas establecidas para el desarrollo. Este muestra una interfaz cómoda e intuitiva para el usuario. Se respetaron los colores para las ventanas y las imágenes que se muestran en las pantallas generadas.

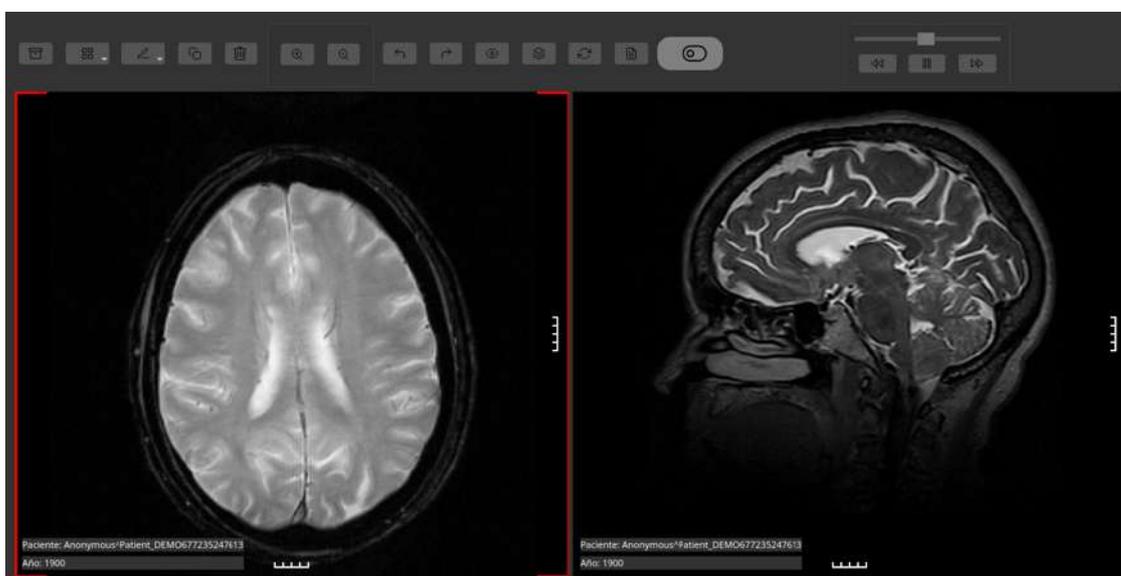


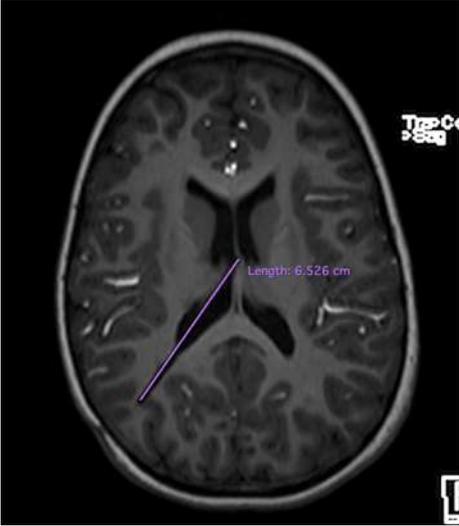
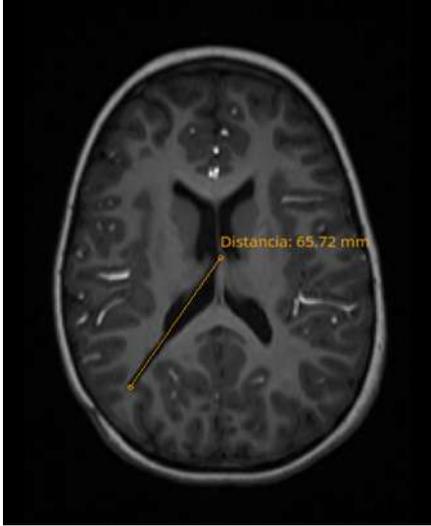
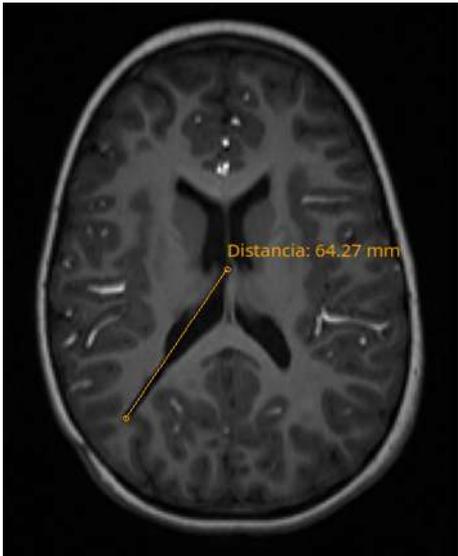
Figura 26- Ventana principal de la interfaz del herramienta para comparación. Fuente: Elaboración propia

### 3.5. Comparación con herramientas homólogas en cuanto a precisión

La comparación para la precisión en las mediciones del algoritmo se realizó con la herramienta desarrollada y el software Horos. Como umbral de diferencia se toma que las medidas son manuales en el caso 1 y en el segundo caso se utilizaron las coordenadas de los puntos iniciales y finales de las líneas de medición en ambas herramientas. En la tabla 11 se muestran los resultados de las mediciones.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Tabla 11 - Comparación de precisión de mediciones entre la herramienta desarrollada y Horos

Medición en Horos.	Medición en la solución de los autores.
Mediciones realizadas a mano alzada con la funcionalidad trazar línea de medición.	
	
Medición: 6.526 cm Conversión: 65.26 mm	Medición: 65.72 mm
Diferencia de medición: 0.46 mm	
Mediciones realizadas con las coordenadas iniciales y finales de las líneas de medición	
	
Coordenada de inicio: (128,125) Coordenada de fin: (92, 178)	Coordenada de inicio: (128,125) Coordenada de fin: (92, 178)
Medición: 6.445 cm	Medición: 64.27 mm

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Conversión: 64.45 mm	
Diferencia de medición: 0.18 mm	

En las mediciones realizadas se detecta un umbral de diferencia de 0.46 mm en las mediciones manuales y 0.18 mm en las mediciones con las coordenadas iniciales y finales de las líneas de medición. Para comprobar la precisión de la herramienta desarrollada por los autores se seleccionó el la técnica de Bland-Alman para calcular el porcentaje de concordancia entre ambas herramientas. Esta técnica se basa en examinar las diferencias entre las mediciones de ambas herramientas y representarlas en un gráfico de dispersión. Para su aplicación es necesario calcular la diferencia absoluta de las mediciones obtenidas en cada instancia y calcular el promedio de diferencias entre los dos métodos(68).

$$DA = |m1 - m2| \tag{VII}$$

$$PD = \frac{\sum D}{No} \tag{VIII}$$

En estas fórmulas DA es la diferencia absoluta, m las mediciones, PD es el promedio de diferencias, D la diferencia y No es el número de observaciones. Para aplicar el gráfico de dispersión es necesario calcular los límites superiores e inferiores  $\pm 1.96$  la desviación estándar de las mediciones.(68)

$$DED = \sqrt{\sum \frac{(D - PD)^2}{No}} \tag{IX}$$

$$Ls = PD + 1.96 * DED \tag{X}$$

$$Li = PD - 1.96 * DED \tag{XI}$$

Para aplicar la técnica se utilizó el set de datos proporcionado por CNEURO, con los datos de los pacientes anonimizados. Se seleccionó uno de los estudios de los pacientes y se sometieron a 10 mediciones de cada herramienta. En la tabla 12 se muestra el proceso de mediciones y los resultados.

Tabla 12 - Comparación de mediciones entre las herramientas

Casos	Herramienta Horos	Solución de los autores	Diferencia absoluta	Porcentaje de diferencia
-------	----------------------	----------------------------	------------------------	-----------------------------

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

Caso 1	64.45 mm	64.27 mm	0.18 mm	0.34%
Caso 2	67.47 mm	67.32 mm	0.15 mm	0.29%
Caso 3	34.02 mm	34.22 mm	0.20 mm	0.38%
Caso 4	42.28 mm	42.45 mm	0.17 mm	0.32%
Caso 5	83.89 mm	83.63 mm	0.26 mm	0.50%
Caso 6	42.78 mm	42.90 mm	0.12 mm	0.23%
Caso 7	74.16 mm	73.94 mm	0.22 mm	0.42%
Caso 8	95.85 mm	95.67 mm	0.18 mm	0.34%
Caso 9	4.68 mm	4.84 mm	0.16 mm	0.31%
Caso 10	13.59 mm	13.78 mm	0.19 mm	0.36%

Al aplicar la técnica de Bland-Altman se calculan los porcentajes de precisión de la solución de los autores comparado al Horos. Los resultados de esta comparación se observan en el gráfico de dispersión figura 27.

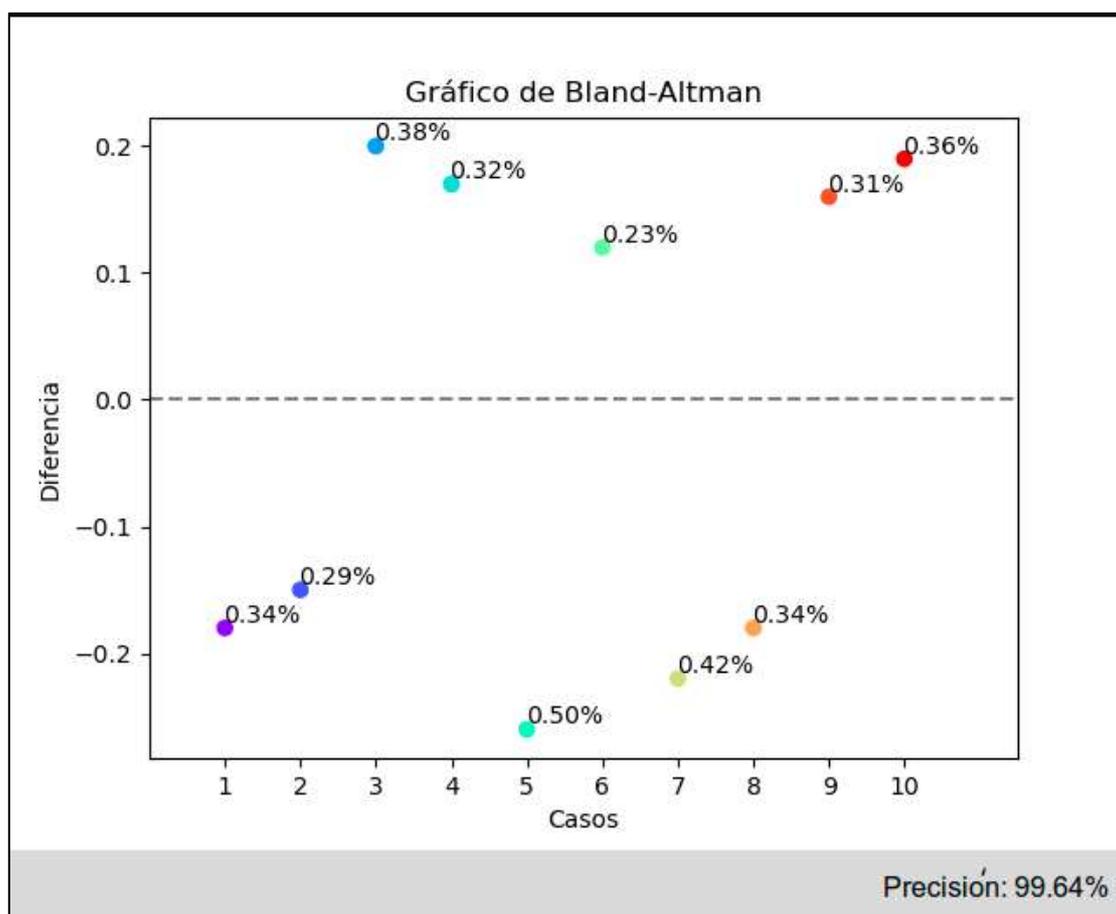


Figura 27 - Gráfico de dispersión obtenido por técnica de Bland-Altman. Fuente: Elaboración propia

Al aplicar la técnica de Bland-Altman se obtienen los valores expresados en porcentos de las diferencias entre las mediciones de ambas herramientas. El valor de precisión

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

obtenido por la propuesta de solución de los autores es de 99.64%. respecto a la comparación con su homólogo Horos.

### **Conclusiones del capítulo.**

Para dar solución a la propuesta diseñada se implementaron las funcionalidades de la herramienta a partir del estándar de código seleccionado a fin de lograr una programación simple y entendible. Las pruebas de software posibilitaron la detección y mitigación de no conformidades en el funcionamiento de la herramienta, entre las de alto impacto se encuentran el requisito funcional trazar línea de medición y replicar mediciones en los estudios. La técnica de Bland-Alman posibilitó la validación de la precisión de la herramienta mediante la comparación de homólogos a la solución con un valor diagnóstico del 99.64%.

## CONCLUSIONES:

Con la realización del presente trabajo de diploma se ha cumplido con el objetivo general propuesto, así como con las tareas de la investigación, arribándose a las siguientes conclusiones.

- La elaboración del marco teórico de la investigación permitió trazar la estrategia para el análisis de los elementos relacionados con la comparación de estudios sobre imágenes de resonancia magnética, facilitó las vías para el trabajo con archivos en formato DICOM
- El análisis de las herramientas para el manejo y comparación de estudios sobre imágenes de resonancia magnética, posibilitó sentar las bases para el desarrollo de la propuesta de solución.
- Los algoritmos identificados en la literatura constituyen variantes para las mediciones de estructuras cerebrales. A partir de la comparación de precisión se seleccionó el algoritmo de detección de puntos de referencia para trazar las líneas de medición en las imágenes de resonancia magnética.
- Las pruebas realizadas a la herramienta para comparación de estudios permitieron mostrar que la propuesta de solución cumple con las funcionalidades identificadas. La técnica de Bland-Alman comprobó mediante comparación de homólogos el valor diagnóstico de la solución que presentó un 99.64% de precisión en las mediciones.

## Recomendaciones

Para dar continuidad a la presente investigación se recomienda:

- Integrar la herramienta para comparación de estudios sobre imágenes de resonancia magnética en formato DICOM al sistema XAVIA PACS-RIS
- Implementar en la herramienta un modelo entrenado de redes neuronales para la identificación y medición automática de estructuras anómalas en el cerebro

## REFERENCIAS

1. PURVES, Dale, AUGUSTINE, George J., FITZPATRICK, Davis, HALL, William C. and LAMANTIA, Anthony S. *Neurociencia*. . Editorial médica panamericana, 2007.
2. ESCERA, Carles. Aproximación histórica y conceptual a la Neurociencia Cognitiva. *Cognitiva*. 2004. Vol. 16, no. 2, p. 141–61.
3. MARIA, Mamani Castillo Carla. La Inteligencia Artificial en la Neurociencia. .
4. FORERO, M. G., SARMIENTO, W. J. and DÍAZ, G. M. Aplicación para la lectura y conversión de formatos DICOM no estándar y génesis. In: *II Congreso Latinoamericano de Ingeniería Biomédica*. 2001.
5. RIVERA, Diego Miguel, PUENTES, Sofía and CABALLERO, Ligia. Resonancia magnética cerebral: secuencias básicas e interpretación. *Universitas Médica*. 2011. Vol. 52, no. 3, p. 292–306.
6. CUBA, Centro de Neurociencias de. *Centro de Neurociencias de Cuba*. Online. 2020. Available from: <https://www.cneuro.cu/>
7. Empresa Centro de Neurociencias de Cuba – Empresa Centro de Neurociencias de Cuba. Online. [Accessed 1 March 2023]. Available from: <https://www.cneuro.cu/>
8. ORELLANA GARCÍA, Arturo, VEGA IZAGUIRRE, Leodan, LÓPEZ COSSIO, Filiberto, MOLINA HERNÁNDEZ, Yenisel, SOTO PELEGRÍN, Lissette, CERUTO MARRERO, Gerardo, MÉNDEZ MEDEROS, Arianne and DÍAZ RUIZ, Marien. Impactos de la plataforma XAVIA PACS-RIS en hospitales del centro y occidente de Cuba. *Revista de Información científica para la Dirección en Salud. INFODIR; No. 41 (2023): mayo - agosto*. Online. 25 June 2023. Available from: <https://revinfodir.sld.cu/index.php/infodir/article/view/1402/1696>
9. LOPEZ, J, CHIMENOS, E, BLANCO, A, RESELLO, X and JANE, E. Diagnóstico por la imagen de los trastornos de la articulación craneomandibular. *Avances en Odontoestomatología*. 2005. Vol. 21, no. 2, p. 71–88.
10. LÉVANO LOAYZA, Sandro Alexander, SOVERO GASPAR, Abell Temistocles, LÉVANO LOAYZA, Sandro Alexander and SOVERO GASPAR, Abell Temistocles. Evaluación anatómica de la articulación temporomandibular mediante resonancia magnética. Artículo de revisión. *Revista Estomatológica Herediana*. October 2020. Vol. 30, no. 4, p. 285–293. DOI 10.20453/reh.v30i4.3882.
11. GODOY, N and MONTOYA, M. *Resonancia magnética nuclear y resonancia magnética funcional*. 2009. Published: Informe técnico, Escuela de Ingeniería de Antioquia - Instituto de ciencias de la salud
12. CHAVHAN, Govind, BABYN, Paul, THOMAS, Binu, SHROFF, Manohar and HAACKE, Ewart. Principles, techniques, and applications of T2\*-based MR imaging and its special applications. *Radiographics*. 2009. Vol. 29, no. 5, p. 1433–1449.

13. HAN, MC and KIM, CW. *Cortes anatómicos correlacionados con TC y RM*. . 3rd. Madrid, Spain : Editorial Marban, 1998.
14. TWA, Michael D. and JOHNSON, Chris A. Digital Imaging and Communication Standards. *Optometry and Vision Science*. 2022. Vol. 99, no. 5, p. 423.
15. LAROBINA, Michele. Thirty Years of the DICOM Standard. *Tomography*. 2023. Vol. 9, no. 5, p. 1829–1838.
16. PIEDRAHITA, JORGE HERNANDO RIVERA, SERNA, WALTER SERNA and LEMUS, JUAN PABLO TRUJILLO. Descripción del estándar DICOM para un acceso confiable a la información de las imágenes médicas. *Scientia et technica*. 2010. Vol. 16, no. 45, p. 289–294.
17. ROSSET, Antoine, SPADOLA, Luca and RATIB, Osman. OsiriX: an open-source software for navigating in multidimensional DICOM images. *Journal of digital imaging*. 2004. Vol. 17, p. 205–216.
18. *OsiriX DICOM Viewer \textbar The world famous medical imaging viewer*. Online. 2023. [Accessed 30 May 2023]. Available from: <https://www.osirix-viewer.com/>
19. AGBETOBA, Abib, LUONG, Amber, SLOW, Jin Keat, SENIOR, Brent, CALLEJAS, Claudio, SZCZYGIELSKI, Kornel and CITARDI, Martin J. Educational utility of advanced three-dimensional virtual imaging in evaluating the anatomical configuration of the frontal recess. In : *International forum of allergy & rhinology*. Wiley Online Library, 2017. p. 143–148. ISBN 2042-6976.
20. *RadiAnt DICOM Viewer \textbar ES*. Online. 2009. [Accessed 30 May 2023]. Available from: <https://www.radiantviewer.com/>
21. SANO, Atsushi. Virtual bronchoscopy using Horos. *Lung India*. 2020. Vol. 37, no. 5, p. 457–458.
22. *Our Mission - Horos Project*. Online. 2023. [Accessed 30 May 2023]. Available from: <https://horosproject.org/about/>
23. ZEPPA, Pietro, NEITZERT, Luca, MAMMI, Marco, MONTICELLI, Matteo, ALTIERI, Roberto, CASTALDO, Margherita, COFANO, Fabio, BORRÈ, Alda, ZENGA, Francesco and MELCARNE, Antonio. How reliable are volumetric techniques for high-grade gliomas? A comparison study of different available tools. *Neurosurgery*. 2020. Vol. 87, no. 6, p. E672–E679.
24. *eFilm Solutions*. Online. 2023. [Accessed 1 March 2023]. Available from: <https://estore.merge.com/na/index.aspx>
25. AL-MAYAH, Adnan, AL-JANABI, Abdulhadi and MOHAMMED, Fatimah. Calibration of X-ray computed tomography systems: A review. *Measurement*. 2020. Vol. 157, p. 107741. DOI 10.1016/j.measurement.2019.107741.
26. J. P. HORNAK. *Medical Imaging Signals and Systems*. . 2nd edition. 2015.

Pearson.

27. J. P. HORNAK,. “*The Basics of MRI*” (3rd edition). . 3ra. 2012.
28. J. P. HORNAK,. *Medical Imaging Signals and Systems*. . 2nd edition. 2015. Pearson.
29. NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION (NEMA). Digital Imaging and Communications in Medicine (DICOM). .
30. AL-MAYAH, Adnan, AL-JANABI, Abdulhadi and MOHAMMED, Fatimah. Calibration of X-ray computed tomography systems: A review. *Measurement*. 2020. Vol. 157, p. 107741. DOI 10.1016/j.measurement.2019.107741.
31. HURTADO, Juan M, MONTILLA, Alberto and ROJAS, Juan A. A novel calibration method for medical imaging systems based on physical measurements. *Medical Physics*. 2021. Vol. 48, no. 1, p. 164–174. DOI 10.1002/mp.14531.
32. ZHANG, Ming, LI, Xiang and PENG, Yufeng. A comprehensive review and comparison of different calibration methods for camera-based non-contact measurement systems. *Optics and Lasers in Engineering*. 2020. Vol. 133, p. 106047. DOI 10.1016/j.optlaseng.2020.106047.
33. VÁZQUEZ-BAUTISTA, Olivia. Aplicación de la recta para detectar líneas en imágenes digitales con la transformada Hough. *Con-Ciencia Boletín Científico de la Escuela Preparatoria No. 3*. 2021. Vol. 8, no. 15, p. 19–22.
34. ESENG-BILOGO, Fortunato-C. Visualizador de imágenes médicas dicom. Online. 2016. [Accessed 3 November 2023]. Available from: <https://crea.ujaen.es/handle/10953.1/3624>
35. NIÑO NAVARRETE, MANUEL. DISEÑO DE UNA HERRAMIENTA WEB PARA MANIPULAR IMÁGENES MÉDICAS. Online. 2019. [Accessed 5 November 2023]. Available from: <http://rinacional.tecnm.mx/handle/TecNM/2733>
36. DECOURT, Colin and DUONG, Luc. Semi-supervised generative adversarial networks for the segmentation of the left ventricle in pediatric MRI. *Computers in Biology and Medicine*. 2020. Vol. 123, p. 103884.
37. FLORKOW, Mateusz C., WILLEMSSEN, Koen, ZIJLSTRA, Frank, FOPPEN, Wouter, VAN DER WAL, Bart C. H., VAN DER VOORT VAN ZYP, Jochem R. N., VIERGEVER, Max A., CASTELEIN, René M., WEINANS, Harrie, VAN STRALEN, Marijn, SAKKERS, Ralph J. B. and SEEVINCK, Peter R. MRI-based synthetic CT shows equivalence to conventional CT for the morphological assessment of the hip joint. *Journal of Orthopaedic Research*. April 2022. Vol. 40, no. 4, p. 954–964. DOI 10.1002/jor.25127.
38. MONCHO SANTONJA, María, MICÓ-VICENT, Bàrbara, DEFEZ, Beatriz, JORDÁN, Jorge and PERIS-FAJARNES, Guillermo. Hough Transform Sensitivity Factor

Calculation Model Applied to the Analysis of Acne Vulgaris Skin Lesions. *Applied Sciences*. 2022. Vol. 12, no. 3, p. 1691.

39. SANTINI, Fernando Oscar and SANTINI, Carlos Ariel. Herramienta de Software para el Diagnóstico de Estudios Radiológicos Digitales del Raquis bajo Norma DICOM. In : *Escuela y Workshop Argentino en Ciencias de las Imágenes*. Online. 2021. p. 43. [Accessed 3 November 2023]. Available from:

[https://www.academia.edu/download/51607541/Actas\\_ECImag2012.pdf#page=47](https://www.academia.edu/download/51607541/Actas_ECImag2012.pdf#page=47)

40. PICO-VELÁSQUEZ, Jean Alejandro, MORENO-TARAZONA, Alejandra, GUAYACÁN-CHAPARRO, Luis Carlos, BAUTIST-ROZO, Lola Xiomara and MARTÍNEZ-CARRILLO, Fabio. A fast right ventricle segmentation in cine-MRI from a dense hough representation. *Revista Politécnica*. 2022. Vol. 18, no. 35, p. 84–97.

41. KRITSKIY, Dmitriy, POHUDINA, Olha, KOVALEVSKYI, Mykhailo, TSEGELNYK, Yevgen and KOMBAROV, Volodymyr. Powder Mixtures Analysis for Laser Cladding Using OpenCV Library. In : NECHYPORUK, Mykola, PAVLIKOV, Vladimir and KRITSKIY, Dmitriy (eds.), *Integrated Computer Technologies in Mechanical Engineering - 2021*. Online. Cham : Springer International Publishing, 2022. p. 924–937. Lecture Notes in Networks and Systems. [Accessed 6 November 2023]. ISBN 978-3-030-94258-8.

42. GOLLAPUDI, Sunila. OpenCV with Python. In : GOLLAPUDI, Sunila, *Learn Computer Vision Using OpenCV*. Online. Berkeley, CA : Apress, 2019. p. 31–50. [Accessed 6 November 2023]. ISBN 978-1-4842-4260-5.

43. SIGUT, Jose, CASTRO, Miguel, ARNAY, Rafael and SIGUT, Marta. OpenCV basics: A mobile application to support the teaching of computer vision concepts. *IEEE Transactions on Education*. 2020. Vol. 63, no. 4, p. 328–335.

44. SIVKOV, Stepan, NOVIKOV, Leonid, ROMANOVA, Galina, ROMANOVA, Anastasia, VAGANOV, Denis, VALITOV, Marat and VASILIEV, Sergey. The algorithm development for operation of a computer vision system via the OpenCV library. *Procedia Computer Science*. 2020. Vol. 169, p. 662–667.

45. RADHAKRISHNAN, Meera, PANNEERSELVAM, Anandan and NACHIMUTHU, Nandhagopal. Canny edge detection model in MRI image segmentation using optimized parameter tuning method. *Intell. Autom. Soft Comput.* 2020. Vol. 26, no. 4, p. 1185–1199.

46. PENG, Wei and ZHAO, Yuan. Liver CT image segmentation based on modified Canny algorithm. In : *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. Online. IEEE, 2019. p. 1–5. [Accessed 6 November 2023]. Available from:

<https://ieeexplore.ieee.org/abstract/document/8966082/>

47. EUCLIDIANA, Distancia. Actividad 12: Lectura Capítulo 7. Online. 2019. [Accessed 6 November 2023]. Available from: <https://cmaspUBLIC.ihmc.us/rid=1M1Z77ZMV-KZD0FC-221K/Actividad%2012%20lectura%207.pdf>
48. BARRIOS, José M. González and RUEDA, Raúl. Identificación de la dimensión en una matriz de distancias euclidianas. Online. 2012. [Accessed 6 November 2023]. Available from: [https://miscelaneamatematica.org/welcome/default/download/tbl\\_articulos.pdf2.be7368ea5ed14dd3.52756564615f722e706466.pdf](https://miscelaneamatematica.org/welcome/default/download/tbl_articulos.pdf2.be7368ea5ed14dd3.52756564615f722e706466.pdf)
49. BERMEJO, Angel Javier Carrillo. *Clasificación Morfométrica de Tumores Cerebrales Utilizando Aprendizaje Supervisado*. Online. PhD Thesis. Universidad Autónoma de Yucatán, 2019. [Accessed 6 November 2023]. Available from: <https://www.cilir-lab.org/images/tesis/2018-Angel%20Javier%20Carrillo%20Bermejo-Maestria.pdf>
50. RODRÍGUEZ SÁNCHEZ, T. *Metodología de desarrollo para la Actividad productiva de la UCI*. . La Habana : Universidad de las Ciencias Informáticas, 2015.
51. *Visual Studio Code*. Online. June 2023. Microsoft. Available from: <https://code.visualstudio.com/docs>
52. GIT. *About Git*. Online. 2023. Available from: <https://git-scm.com/about>
53. PYTHON SOFTWARE FOUNDATION. *About Python*. Online. 2021. Available from: <https://www.python.org/about/>
54. KANG, Eunju, MIN, Jung Hwan, LEE, Joon, PARK, Hyeonjin, KIM, Byoung-Nam and LEE, Jong-Min. Deep learning-based image reconstruction algorithm for reducing acquisition time in 3-T magnetic resonance imaging of the knee: A multicenter prospective study. *Investigative Radiology*. 2019. Vol. 54, no. 9, p. 587–594. DOI 10.1097/RLI.0000000000000583.
55. CAMARA, RICARDO ARMANDO BARRERA. Gestión de procesos de negocio. *Inventio. La génesis de la cultura universitaria en Morelos*. Online. 2019. [Accessed 22 October 2023]. Available from: <http://riaa.uaem.mx/handle/20.500.12055/761>
56. PRESSMAN, Roger S. *Software engineering: a practitioner's approach*. . Palgrave macmillan, 2005. ISBN 0-07-301933-X.
57. FREEMAN, A. and ROBSON, J. *Head First Design Patterns: A Brain-Friendly Guide*. . O'Reilly Media, 2015.
58. SHALLOWAY, A. and TROTT, J. R. *Design Patterns Explained: A New Perspective on Object-Oriented Design*. . Addison-Wesley Professional, 2019.
59. FOWLER, M. *Patterns of Enterprise Application Architecture*. . Addison-Wesley

Professional, 2004.

60. NIKANDER, Jussi, NADSCHLÄGER, Stefan and AUER, Dagmar. Improving the architecture of agricultural knowledge processing systems using design patterns. *Journal of Agricultural Informatics*. 2019. Vol. 10, no. 1, p. 1–11.
61. PALLEP. Capitalization - Microsoft Style Guide. Online. 24 June 2022. [Accessed 24 May 2023]. Available from: <https://learn.microsoft.com/en-us/style-guide/capitalization> Microsoft style uses sentence-style capitalization. That means everything is lowercase except the first word and proper nouns.
62. GARCÍA, Anays Gómez, HERNÁNDEZ, Mercedes Sosa, MARCOS, Sandra Verona and DAPENA, Martha Dunia Delgado. Buenas prácticas de la ingeniería de software: pruebas de software. *Revista Cubana de Transformación Digital*. 2023. Vol. 4, no. 2, p. 205: 1–13.
63. PRESSMAN, Roger S. *Software engineering: a practitioner's approach*. . Palgrave macmillan, 2005. ISBN 0-07-301933-X.
64. eFilm Solutions. Online. [Accessed 1 March 2023]. Available from: <https://estore.merge.com/na/index.aspx>
65. ZIMMERER, René E., AFRICA, Robert E., WESTENHAVER, Zack K. and MCKINNON, Brian J. Bisphosphonate therapy in otosclerosis: A scoping review. *Laryngoscope investigative otolaryngology*. 2022. Vol. 7, no. 1, p. 242–249.
66. LARREA, Juan José Olarte. TC2: una herramienta de apoyo para las pruebas del software. In : *Contribuciones científicas en honor de Mirian Andrés Gómez*. Online. Universidad de La Rioja, 2010. p. 411–415. [Accessed 22 October 2023]. Available from: <https://dialnet.unirioja.es/descarga/articulo/3217862.pdf>
67. PRESSMAN, Roger S. A practitioner's approach. *Software Engineering*. 2010. Vol. 2, p. 41–42.
68. GALARZA ZAMBRANO, Jorge Javier. *Herramienta de visualización de datos provenientes del análisis masivo de imágenes médicas en la plataforma de QUIBIM Precision*. Online. PhD Thesis. Universitat Politècnica de València, 2019. [Accessed 22 November 2023]. Available from: <https://riunet.upv.es/handle/10251/127867>