



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Modelo computacional híbrido basado en  
tecnologías de realidad mixta para la planeación  
quirúrgica de procedimientos de mínima invasión**

**TESIS**

Que para obtener el título de  
**Ingeniera en Computación**

**P R E S E N T A**

María Fernanda Ortiz Figueroa

**DIRECTOR DE TESIS**

Dr. Sergio Teodoro Vite



**Ciudad Universitaria, Cd. Mx., 2022**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Dedicatoria

---

*No es fácil llegar hasta este punto, el cual no es ni el inicio ni el fin, sino una de las diversas maneras que tenemos para formar la única y emocionante oportunidad a la que llamamos vida en este tiempo limitado; por lo que soñar y enfrentar la realidad, son dos cuestiones que van de la mano, en donde hay que tener coraje para seguir a nuestro corazón e intuición.*

*A mi asesor, familia y amigos, dedico este esfuerzo, como testimonio de gratitud y humildad por su apoyo, aliento y estímulo que me han dedicado para posibilitar este pequeño granito de arena con el que contribuyo al mundo.*

# Agradecimientos

---

En estas líneas quisiera expresar mi gratitud que me ha llevado a finalizar el presente trabajo, sabiendo que jamás existirán las palabras suficientes para agradecer una vida de lucha, sacrificio y esfuerzo constante, ya que el éxito no solo se logra con cualidades especiales, al contrario, se necesita de ahínco y deseo, pero sobre todo apoyo como el que he recibido todo este tiempo.

A mis padres, Ignacio Ortiz y Norma Figueroa, por su apoyo incondicional, consejos, regaños y desvelos, que han formado lo que soy ahora.

A mi hermana Beatriz, por su apoyo incondicional, en momentos de lágrimas o risas, a quién no cambiaría por nada ni nadie al ser mi cómplice.

A mi familia en general, por su apoyo incondicional, aliento, experiencia y cariño, que me han permitido valorar el tiempo irremplazable de cada momento que he pasado con ellos, ya que han significado un respiro para seguir adelante.

A mi asesor, Sergio Teodoro Vite, por compartir sus conocimientos y experiencias, así como brindarme sin duda alguna su orientación, seguimiento, motivación y paciencia, fundamental para el desarrollo y culminación de este trabajo, sin su apoyo este trabajo no hubiera sido posible.

Al Dr. Víctor Hugo Ibarra Ramírez por su colaboración y retroalimentación, así como a Ulises y Kevin por su apoyo y sugerencias en el desarrollo de este trabajo.

A mis amigos y compañeros que me han acompañado durante mi carrera y formación, por compartir momentos de felicidad, tristeza, estrés, frustración y satisfacción, sin ellos el camino hubiera sido aburrido y sin sentido. En especial a Roberto, Sergio, Marco, Mario y Angelica, el cariño, respeto y admiración, son solo algunas de las pocas palabras que les puedo dedicar, por el poco o mucho tiempo en este trayecto.

A mis profesores, por sus conocimientos y experiencias, pero sobre todo por sus palabras que en su momento han sido fuente de inspiración y admiración.

A DGAPA a través del programa PAPIME del proyecto PE105820 titulado “Desarrollo de un atlas de cabeza y cuello con maniqués de órganos impresos en 3D y realidad mixta para la

enseñanza de la neuroanatomía humana”, por su apoyo al proporcionarme los estímulos económicos necesarios para realizar este trabajo.

Finalmente, a mi alma máter, la Universidad Nacional Autónoma de México, por haberme permitido formarme en sus diversos espacios, no solo profesionalmente, sino también personalmente, al ser una fuente de conocimiento, sabiduría y gente valiosa en todos los ámbitos, a quién espero retribuir sin duda alguna todo el apoyo que me ha brindado. Gracias por darme la oportunidad en adelante y con orgullo de ser una profesionista egresada de la máxima casa de estudios de México.

# Resumen

---

Los sistemas de realidad virtual y realidad aumentada han permitido que gradualmente dispositivos y desarrollos de software se vayan integrando como soluciones a problemas reales. En el área médica, estas aplicaciones de cómputo han dado paso a un área de investigación y desarrollo que busca impactar directamente en los procesos de atención del paciente, con el fin de mejorar su salud y calidad de vida, por medio de la aplicación de nuevos modelos tecnológicos para el diagnóstico, planeación y asistencia médica principalmente.

La visualización tridimensional de estructuras anatómicas del cuerpo desde hace ya más de una década ha significado un gran avance en el planteamiento de nuevos modelos tecnológicos, ya que representan una mejora importante en los procesos de enseñanza y aprendizaje, y una alternativa novedosa para el diagnóstico médico. Sus usos y metodologías han sido reportados en diversas revistas especializadas de biomedicina y cómputo gráfico, tal que han mostrado ser una herramienta eficiente al momento de tomar decisiones en el diagnóstico y tratamiento de patologías complejas, así como una opción en conjunto con los procedimientos de mínima invasión en intervenciones quirúrgicas. Aunado a ello, tras la declaratoria por la pandemia de la COVID-19, los recursos digitales se han convertido en la mejor alternativa para continuar con los procesos anteriores.

Por tanto, el sistema de visualización web de imágenes médicas es un desarrollo compuesto por un conjunto de bloques de software que tiene como objetivo generar y proporcionar diferentes recursos interactivos por medio de representaciones y elementos gráficos en un navegador web, como una herramienta disponible y accesible desde Internet. Por lo que, esta tesis describe los detalles de la puesta en funcionamiento de la carga y visualización de imágenes y modelos de dos casos médicos, sección del corazón y tórax, a partir del renderizado de gráficos 2D y 3D, de manera tradicional e inmersiva desde un explorador web.

Los resultados obtenidos incluyen: una metodología completa para la adquisición de recursos NRRD y VTK a partir de imágenes de TC en formato DICOM, una metodología de creación gráfica para el sistema de visualización web de imágenes médicas dada la definición de elementos como escenas, renderizadores, cámaras, iluminación, interfaces de interacción gráfica, entre otros efectos y funcionalidades, para la carga de imágenes y modelos 3D, así como las herramientas de cómputo empleadas durante todo el proceso de implementación.

# Abstract

---

Virtual reality and augmented reality systems have allowed devices and software developments to gradually be integrated as solutions to real problems. In the medical area, these computer applications have given way to an area of research and development that seeks to directly impact the patient's care processes, in order to improve their health and quality of life, through the application of new technological models for diagnosis, planning and medical assistance mainly.

The three-dimensional visualization of anatomical structures of the body for more than a decade has meant a great advance in the approach of new technological models, since they represent an important improvement in the teaching and learning processes, and a novel alternative for medical diagnosis. Its uses and methodologies have been reported in various specialized journals of biomedicine and graphic computing, such that they have proven to be an efficient tool when making decisions in the diagnosis and treatment of complex pathologies, as well as an option in conjunction with minimally invasive procedures in surgical interventions.

The Medical Image Web Visualization System is a development composed of a set of software blocks that aims to generate and provide different interactive resources through representations and graphic elements in a Web browser, as a tool available and accessible from the Internet. That is why this thesis describes the details of the implementation of the loading and visualization of images and models of two medical cases, section of the heart and chest, from the rendering of 2D and 3D graphics, in a traditional and immersive way from a Web browser.

The results obtained include: a complete methodology for the acquisition of NRRD and VTK resources from CT images in DICOM format, a graphic creation methodology for the Web visualization system of medical images given the definition of elements such as scenes, renderers, cameras, lighting, graphic interaction interfaces, among other effects and functionalities, for the upload of 3D images and models, as well as the computing tools used throughout the implementation process.

# Presentación de la tesis

---

En México, desde el 2016, la UNAM a través del Instituto de Ciencias Aplicadas y Tecnología (ICAT), el Hospital General de México “Dr. Eduardo Liceaga” (HGM) y el Centro Médico Nacional Siglo XXI (CMNSXXI) del IMSS, han iniciado un esfuerzo para potenciar el uso de tecnologías de realidad virtual y aumentada, para que puedan aplicarse como estándares a nivel nacional e internacional en los procedimientos de planeación quirúrgica de mínima invasión. Sin embargo, este reto requiere de profesionales altamente capacitados en el desarrollo de herramientas gráficas que hagan uso de ambientes interactivos personalizados por paciente.

En los servicios de cardiología y cirugía vascular del HGM y CMNSXXI se ha identificado un aumento en los casos de afección de válvulas cardíacas y adelgazamiento de vascularidades que derivan en aneurismas, condiciones de urgencia que ponen en riesgo la vida del paciente, por lo que se han precisado dos necesidades muy específicas: por un lado, requieren de un software para la visualización interactiva de estructuras cardíacas y vasculares que les permita tener una mejor aproximación del caso a tratar previo a un posible procedimiento, y por otro, requieren de un mapa interactivo de las ramificaciones, que les permita establecer la mejor ruta de acceso a la zona afectada.

Aunado a ello, los hospitales de tercer nivel demandan practicantes con un extraordinario conocimiento en anatomía para poder mejorar la atención y el diagnóstico de los pacientes, pues el objeto de estudio de la medicina se centra y origina en la naturaleza del cuerpo, lo cual implica que los estudiantes de ciencias de la salud, deben tener una formación bastante completa en esta área, al ser valorada y considerada como la base del conocer médico y el diagnóstico clínico (Coll, 2007), no obstante, el estudio de la anatomía humana es considerado un tema complejo, tal que ha sido identificado por el departamento de anatomía de la Facultad de Medicina de la UNAM de forma más puntual en el plan de estudios 2010 de la carrera médico cirujano, debido a la gran cantidad de estructuras anatómicas involucradas y la dificultad para dimensionarlas, y al mismo tiempo, los alumnos tienen cada vez menos oportunidad para interactuar con tejidos *post-mortem* o realizar prácticas en los quirófanos.

Por otro lado, la situación actual por la declaratoria de la pandemia por la COVID-19 ha impactado a diversos sectores, entre ellos el educativo y médico, lo cual ha implicado, entre otras cosas, cambios inminentes en la comunicación, enseñanza y acceso a recursos a través de medios



digitales de manera remota como una vía para continuar estas áreas. En este sentido, hasta el momento, se ha podido evidenciar que las tecnologías de realidad mixta han representado una mejora importante en los procesos de enseñanza y aprendizaje, así como una alternativa novedosa en el área médica.

Por tanto, contar con un sistema gráfico, interactivo, de fácil acceso, intuitivo y de rápida y eficiente implementación, ante la variedad estructuras anatómicas y casos de pacientes, para estudiantes y especialistas de medicina, principalmente en el área de cardiología y neurología, representan un impacto positivo en el campo de las tecnologías de cómputo aplicadas en el ámbito médico de la enseñanza, aprendizaje y entrenamiento de habilidades médicas, así como en el diagnóstico, planeación y asistencia en procedimientos quirúrgicos.

## **Objetivos de la tesis**

El objetivo principal de este trabajo de tesis es desarrollar un sistema de visualización web de imágenes médicas que permita a estudiantes, docentes y especialistas en medicina visualizar e interactuar con estructuras anatómicas, como el corazón y el tórax, a partir de imágenes de tomografía computarizada y modelos tridimensionales virtuales generados por computadora y desplegados por medio de un navegador web, a través de diversas representaciones visuales contenidas en una sola herramienta, empleando tecnología de realidad mixta, como una solución competitiva, sistemática y remota respecto a otras existentes que presentan limitantes para su amplio uso en cuanto a compatibilidad y contenidos. Por lo que este sistema, será una herramienta disponible, accesible, interactiva e intuitiva como complemento en la enseñanza, un medio autodidacta y de reforzamiento para los alumnos, y un recurso en un menor tiempo de implementación para especialistas dado un determinado caso de un paciente.

Por tanto, los objetivos particulares de este trabajo suponen:

1. Ofrecer una descripción documental del estado actual del sistema de visualización.
2. La proposición de una metodología genérica y completa para la adquisición de recursos NRRD y VTK a partir de imágenes de tomografía computarizada.
3. La proposición de una metodología para la creación gráfica del sistema de visualización web de imágenes médicas.

4. La documentación de los trabajos realizados, a partir de la descripción general de las herramientas de cómputo empleadas y la presentación de algunos programas desarrollados.

## **Organización de la tesis**

Basados en los objetivos anteriores, este trabajo está compuesto por cuatro capítulos descriptivos, un capítulo de pruebas, un capítulo de resultados, un capítulo de discusión, un capítulo de conclusión, y cuatro anexos, estructurados de la siguiente manera:

En el capítulo 1, se presenta una introducción que aborda el estado del arte en el ámbito de la realidad mixta, en cuanto a sus antecedentes, así como sus aplicaciones en las ciencias médicas en el mundo y en México. Además, se hace hincapié en los sistemas de visualización como implementaciones y alternativas representativas en medicina.

En el capítulo 2, se presenta una breve descripción de la anatomía del corazón, el panorama global de las enfermedades de cardiovasculares, así como la descripción de lo que es un procedimiento de mínima invasión, su uso en México y lo que representa.

En el capítulo 3, se presenta la descripción generalizada de la metodología para la creación gráfica del sistema de visualización web de imágenes médicas, en donde se define de manera estructurada los procesos, herramientas y conceptos empleados.

En el capítulo 4, se presenta el diseño y desarrollo del sistema, así como la implementación de la carga y visualización de imágenes y modelos para el renderizado de gráficos 2D y 3D, de manera tradicional e inmersiva.

En el capítulo 5, se presenta un estudio para evaluar la experiencia de uso del sistema con un grupo de estudiantes, por lo que se describe el objetivo de estudio, participantes, procedimiento y definición de métricas.

En el capítulo 6, se presenta un análisis de los resultados observados en las pruebas realizadas, considerando las métricas cualitativas, a través del análisis estadístico de las principales demandas psicomotoras del nivel de destreza de los participantes.

En los capítulos 7 y 8, se presentan: una discusión sobre los resultados obtenidos con respecto al estado del arte y las posibles mejoras al sistema de visualización, y las conclusiones derivadas del proceso de desarrollo y validación del modelo híbrido propuesto.

Finalmente, se incluyen cinco anexos: el primero, un desglose detallado de los casos de uso del sistema de visualización; el segundo, la estructura generalizada de un script HTML para una visualización; el tercero, el algoritmo de renderizado empleado con Three.js; el cuarto, la encuesta de evaluación aplicada sobre el sistema; y el quinto, la evidencia de la participación en el concurso InnovaUNAM a través de un proyecto ganador que contempla e incluye el presente desarrollo.

# Índice de contenido

---

Capítulo 1. Introducción .....	1
1.1 Antecedentes.....	2
1.1.1 Historia de la Realidad Mixta .....	4
1.2 Realidad Mixta Aplicada en Medicina .....	7
1.2.1 Realidad Mixta Aplicada en Medicina en el Mundo.....	7
1.2.2 Realidad Mixta Aplicada en la Medicina en México .....	10
1.3 Sistemas de Visualización Web para Imágenes Médicas .....	13
1.3.1 LiverAnatomyExplorer .....	15
1.3.2 MEDICOMWEB.....	16
Capítulo 2. Corazón .....	18
2.1 Morfología y Fisiología del Corazón.....	19
2.2 Procedimientos de Mínima Invasión .....	23
Capítulo 3. Metodología .....	27
3.1 Procesamiento de Imágenes Médicas .....	29
3.1.1 DICOM .....	30
3.1.2 NRRD.....	30
3.1.1 FIJI .....	31
3.1.3 <i>Thresholding</i> .....	32
3.1.4 Window .....	32
3.2 Visualización Web de Imágenes Médicas .....	34
3.2.1 HTML, JS y CSS .....	34
3.2.2 WebGL.....	35
3.2.3 Three.js .....	40
3.3 Visualización Web de Imágenes Médicas en Representación de Volumen.....	41
3.3.1 GIMP .....	42
3.3.2 Unidades y Escala de Hounsfield .....	42
3.3.3 <i>Volume Rendering</i> .....	43
3.4 Visualización Web de Imágenes Médicas Reconstruidas .....	50
3.4.1 Software para la Reconstrucción de Modelos 3D .....	50

3.4.2 Formatos para la Visualización de Reconstrucciones de Modelos 3D .....	52
3.5 Visualización Web Mixta .....	53
3.5.1 WebXR.....	54
3.5.2 Oculus Quest 2.....	54
Capítulo 4. Sistema de Visualización Web de Imágenes Médicas.....	55
4.1 Arquitectura .....	55
4.2 Elementos generales .....	58
4.2.1 Escena, Renderizador y Cámara.....	59
4.2.2 Iluminación .....	61
4.2.3 GUI y Controles.....	62
4.3 Recursos Visuales .....	63
4.3.1 Imágenes TC.....	63
4.3.2 Modelos 3D en Tiempo Real .....	67
4.3.3 Modelos 3D Reconstruidos.....	75
4.4 Implementación de la Visualización Mixta.....	78
Capítulo 5. Pruebas.....	80
5.1 Objetivo del estudio .....	80
5.2 Participantes.....	80
5.3 Procedimiento.....	80
5.4 Definición de métricas.....	81
Capítulo 6. Resultados .....	82
Capítulo 7. Discusión .....	97
Capítulo 8. Conclusiones.....	100
Anexos.....	102
A. Casos de uso desglosados del sistema de visualización web de imágenes médicas .....	102
B. Estructura generalizada del script HTML de una visualización .....	103
C. Algoritmo de renderizado empleado con Three.js.....	105
D. Encuesta de evaluación del sistema de visualización web de imágenes médicas .....	108
E. Reconocimiento por participar en el concurso InnovaUNAM. ....	113
Referencias Bibliográficas .....	115

# Índice de figuras

---

<b>Figura 1.</b> Interacción entre equipos, seres humanos y entornos (Traducido de Microsoft, 2020).	2
<b>Figura 2.</b> Continuo de la virtualidad (Milgram et al., 1994).	3
<b>Figura 3.</b> Ejemplos representativos del continuo de la virtualidad. A. Pontiac, Michigan (Jones, 2020). B. Compras con realidad aumentada (El Universal, 2020). C. Innovación virtual dentro del aula (Hernández, 2020). D. Reunión madre e hija fallecida en realidad virtual (Morales, 2020). E. Holograma Stephen Hawking (Kelion, 2018).	3
<b>Figura 4.</b> Realidad mediada de Mann. (Traducido de Peddie, 2016). A. Taxonomía de realidad de Mann. B. Realidad mixta con realidad mediada.	4
<b>Figura 5.</b> Cirugía de bypass asistido por realidad aumentada. (Cabrilo et al., 2015). A. Paciente y reconstrucción. B. Visualización de vasos reconstruidos a través del ocular del microscopio operatorio, antes y después de un corte. C. Visualización de vasos reconstruidos, vasos en tiempo real y vasos en una tomografía computarizada	8
<b>Figura 6.</b> Sistema de realidad aumentada de proyección láser para cirugía asistida por ordenador. A. Proyector Xar Trax montado en la cámara Polaris y el trípode. B. TC reconstruida y trayecto de craneotomía planificada. C. Patrón de proyección. (Glossop et al., 2003).	9
<b>Figura 7.</b> Cirugía endoscópica asistida por imágenes. (Ukimura & Gill, 2008). A partir de los datos de tomografía computarizada originales basados en un sistema de navegación codificada en cuatro colores, se realiza la sincronización del modelo 3D con la vista quirúrgica real mediante el sistema de realidad aumentada.	10
<b>Figura 8.</b> Sistema de navegación de realidad aumentada. (De Paolis et al., 2020). A. Elementos del sistema: brazalete Myo, casco de realidad virtual Oculus Rift y cámara estéreo OVRVision Pro. B. Escena de realidad aumentada de un modelo reconstruido para la planificación de una cirugía cardiovascular. C. Casco de realidad virtual extendido.	11
<b>Figura 9.</b> Simulador de craneotomía funcionando con la interfaz háptica (Garduño, 2018).	12
<b>Figura 10.</b> Sistema de simulación de cirugía de próstata. (Teodoro, 2008). A. Interfaz mecatrónica completa. B. Simulador gráfico de RTUP en ejecución en tiempo real. C. Simulador de realidad virtual del RTUP completo.	13
<b>Figura 11.</b> Sistemas de visualización web de imágenes médicas. A. Voxel-Man 3D-Navigator: Inner Organs. (VOXEL-MAN, 2021). B. PrimalPictures 3D Human Anatomy (PRIMAL PICTURES,	

2021). C. Anatomy 3D Atlas (Anatomy3DATlas, 2013). D. Functional Neuroanatomy (UBC, 2018). E. AnatomyLearning (AL, 2021). F. MEDVIAS (UNAM, 2021) .....	15
<b>Figura 12.</b> Herramienta de enseñanza quirúrgica basada en WebGL en 2D y 3D (Birr et al., 2013). .....	16
<b>Figura 13.</b> Sistema MEDICOMWEB (Angarita y Beltrán, 2016). .....	16
<b>Figura 14.</b> Ubicación y generalidades del corazón. A. Ubicación del corazón (Wyangarden & Smith, 1998). B. Capas del corazón (Gartner & Hiatt, 2006). .....	20
<b>Figura 15.</b> Anatomía corazón: principales partes que lo conforman (Editado de Gartner & Hiatt, 2006).....	21
<b>Figura 16.</b> Estructuras valvulares del corazón (Editado de Gartner & Hiatt, 2006).....	22
<b>Figura 17.</b> Ciclo cardíaco (Editado de DMCA, 2014). .....	22
<b>Figura 18.</b> Esquema generalizado de la metodología empleada en el desarrollo del sistema de visualización web de imágenes médicas. ....	29
<b>Figura 19.</b> Representación gráfica del procesamiento LUT: cada píxel en la imagen de entrada f se le asigna un píxel correspondiente de la imagen de salida g con la nueva intensidad de píxel tomando un valor especificado por la LUT. ....	33
<b>Figura 20.</b> Transformación de imágenes mediante una tabla de búsqueda en escala de grises. 33	
<b>Figura 21.</b> Arquitectura de WebGL (Editado de Crespo, 2012). .....	36
<b>Figura 22.</b> Pipeline de renderizado de WebGL. ....	38
<b>Figura 23.</b> Proceso de renderizado en un navegador web (Editado de Chromium, 2021).....	40
<b>Figura 24.</b> Estructura general de una aplicación en Three.js (Editado de Three.js Fundamentals, 2021).....	41
<b>Figura 25.</b> Escala de Hounsfield (Aguinaga et al., 2006). .....	43
<b>Figura 26.</b> Aproximación de la Composición Volumétrica (Carmona, 2000).....	47
<b>Figura 27.</b> Ray Casting: por cada centro de píxel de la imagen se lanza un rayo (Carmona, 2000). .....	48
<b>Figura 28.</b> Algoritmo de marching cubes: 15 configuraciones originales (Silvetti et al., 1999).....	49
<b>Figura 29.</b> Volume rendering. A. Representación 3D por MIP. B. Representación 3D por ISO...50	
<b>Figura 30.</b> Proceso de reconstrucción del modelo 3D del corazón: aorta ascendente y descendente. 1. Segmentación general en 3DSlicer. 2. Segmentación específica en 3DSlicer. 3. Reducción de vértices duplicados y ubicación en el origen en Blender. 4. Carga del modelo en MeshLab. 5. Visualización original de vértices en MeshLab. 6. Visualización reducida de vértices en MeshLab. ....	51

<b>Figura 31.</b> Reconstrucción de modelo 3D en 3DSlicer: primeros 4 cuadrantes caso corazón (izquierda) y siguientes 4 cuadrantes caso tórax (derecha), tal que se representa 1. Plano transversal. 2. Vista 3D. 3. Plano Coronal. 4. Plano Sagital. ....	52
<b>Figura 32.</b> Aplicaciones contemporáneas de realidad virtual con dispositivos de interacción en aplicaciones médicas (Adaptado de Teodoro, 2021). A. Sistema interactivo para la planeación quirúrgica. B. Navegación virtual en el interior de estructuras vasculares del corazón en el Centro de Simulación CMSXII del IMSS. C. Sistema de realidad virtual para el entrenamiento de residentes en algología, para procedimiento de punción del ganglio trigémino en el Centro de Destrezas del HGM. ....	54
<b>Figura 33.</b> Arquitectura del sistema de visualización web de imágenes médicas. ....	56
<b>Figura 34.</b> Diagrama de flujo del sistema de visualización web de imágenes médicas. ....	57
<b>Figura 35.</b> Abstracción del caso de uso del sistema de visualización web de imágenes médicas. ....	58
<b>Figura 36.</b> Representación gráfica del sistema de visualización web de imágenes médicas. 1. Barra informativa. 2. Barra de opciones de visualización. 3. Ventana de visualización. 4. Controles. ....	58
<b>Figura 37.</b> Representación gráfica de la GUI de la vista sagital en el sistema de visualización web de imágenes médicas ....	63
<b>Figura 38.</b> Vistas de las imágenes TC del caso de corazón en el sistema de visualización web. A. Vista transversal. B. Vista sagital. C. Vista coronal. D. Vista 3D. ....	67
<b>Figura 39.</b> Relación imágenes TC, píxel, vóxel y texel: donde x, y, z representan las coordenadas ortogonales de un objeto 3D, y v, u, w representan las coordenadas de mapeado de una textura. ....	68
<b>Figura 40.</b> Relación de escalas empleadas para generar una textura. ....	69
<b>Figura 41.</b> Textura "General" creada con base a los valores para tejidos del cuerpo humano en la escala de UH: representación en caso de tórax renderizado. ....	70
<b>Figura 42.</b> Texturas generadas en GIMP para la visualización de mapa de colores con su respectiva representación de volumen renderizado en el caso de corazón. Identificación de texturas: 1. Abstracto1, 2. Abstracto2, 3. Abstracto3, 4. Amanecer, 5. Azules1, 6. Azules2, 7. Cafés, 8. Espectro1, 9. Espectro2, 10. General, 11. Grises, 12. Horizonte1, 13. Horizonte2, 14. Incandescente, 15. Sombra1, 16. Sombra2, 17. Sombra3, y 18. Viridis. ....	71
<b>Figura 43.</b> Vista del mapa de colores del modelo 3D generado en tiempo real del caso de corazón. A. Estilo MIP. B. Estilo ISO. ....	74



<b>Figura 44.</b> Proceso general para generar un modelo 3D en tiempo real en el sistema de visualización web. ....	74
<b>Figura 45.</b> Vista modelo en el sistema de visualización web de imágenes médicas del caso de corazón y tórax: a la izquierda se muestra la visualización en el sistema del modelo 3D reconstruido y las imágenes TC en los diferentes cortes, mientras que a la derecha la visualización detallada de la reconstrucción generada a partir de imágenes TC. ....	78
<b>Figura 46.</b> Visualización con realidad virtual en el explorador web del Oculus Quest 2 del caso de tórax en la vista de mapa de colores. ....	79
<b>Figura 47.</b> Gráfica de esfuerzo. ....	83
<b>Figura 48.</b> Gráfica de demanda mental. ....	85
<b>Figura 49.</b> Gráfica de demanda física. ....	86
<b>Figura 50.</b> Gráfica de demanda temporal. ....	87
<b>Figura 51.</b> Gráfica de rendimiento. ....	89
<b>Figura 52.</b> Gráfica de frustración. ....	91
<b>Figura 53.</b> Casos de uso desglosados del sistema de visualización web de imágenes médicas. ....	102
<b>Figura 54.</b> Reconocimiento por participar en el concurso InnovaUNAM para la Enseñanza y el Aprendizaje de Contenidos Curriculares Prácticos en Ciencias y Humanidades a Distancia con el proyecto: “Atlas anatómico 3D remoto con realidad virtual para la enseñanza-aprendizaje de la anatomía humana y diagnóstico médico usando imágenes de tomografía computarizada”.....	113

# Índice de tablas

---

<b>Tabla 1.</b> Historia de la realidad mixta. ....	5
<b>Tabla 2.</b> Imágenes médicas más comunes (García, 2010). ....	14
<b>Tabla 3.</b> Comparativa de las principales características de las aplicaciones de visualización de contenido médico que se centran en la anatomía humana. ....	17
<b>Tabla 4.</b> Diagrama de Gantt que define el calendario de actividades que contempla la metodología de software para la creación gráfica del sistema de visualización web de imágenes médicas. ....	28
<b>Tabla 5.</b> Estructura general del formato de archivo NRRD. ....	31
<b>Tabla 6.</b> Especificación de dispositivos y plataformas compatibles con el sistema de visualización web de imágenes médicas. ....	55
<b>Tabla 7.</b> Escena y renderizador en Three.js: extracto de la definición de la vista 3D del sistema de visualización web de imágenes médicas. ....	59
<b>Tabla 8.</b> Cámaras virtuales en Three.js: extracto de la definición de la vista de mapa de colores y 3D del sistema de visualización web de imágenes médicas. ....	60
<b>Tabla 9.</b> Fuentes de iluminación en Three.js: extracto de la definición de la vista transversal del sistema de visualización web de imágenes médicas. ....	61
<b>Tabla 10.</b> GUI y controles en Three.js: extracto de la definición de la vista sagital del sistema de visualización web de imágenes médicas. ....	62
<b>Tabla 11.</b> Algoritmo de conversión de imágenes TC DICOM a NRRD en FIJI. ....	64
<b>Tabla 12.</b> Carga de imágenes TC en formato NRRD en Three.js: extracto de la definición de la vista 3D del caso de corazón en el sistema de visualización web de imágenes médicas. ....	65
<b>Tabla 13.</b> Equivalencia de los valores de los tejidos del cuerpo humano dada diversas escalas. ....	69
<b>Tabla 14.</b> Carga de texturas en la vista mapa de colores con base a la implementación en Three.js. ....	72
<b>Tabla 15.</b> Configuración del material, malla y rendering en Three.js para la vista de mapa de colores. ....	73
<b>Tabla 16.</b> Código en Python para convertir un archivo OBJ a VTK. ....	76
<b>Tabla 17.</b> Comparativa de pesos de archivo en el proceso de adquisición, transformación, refinamiento y conversión de reconstrucciones anatómicas del corazón y tórax en formato OBJ y VTK. ....	76

<b>Tabla 18.</b> Implementación de Three.js para cargar la estructura de hueso en formato VTK en la visualización de modelo del caso corazón.....	77
<b>Tabla 19.</b> Resultados de los descriptivos asociados a la evaluación de rendimiento en relación con el rubro de esfuerzo en la escala NASA-TLX. ....	84
<b>Tabla 20.</b> Resultados de los descriptivos asociados a la evaluación de la actividad mental y percepción en relación con el rubro de demanda mental en la escala NASA-TLX. ....	85
<b>Tabla 21.</b> Resultados de los descriptivos asociados a la evaluación de la cantidad de actividad física en relación con el rubro de demanda física en la escala NASA-TLX.....	87
<b>Tabla 22.</b> Resultados de los descriptivos asociados a la evaluación del nivel de precisión temporal sentida entre el requerido y el disponible en relación con el rubro de demanda temporal en la escala NASA-TLX. ....	88
<b>Tabla 23.</b> Resultados de los descriptivos asociados a la evaluación de satisfacción en relación con el rubro de rendimiento en la escala NASA-TLX. ....	90
<b>Tabla 24.</b> Resultados de los descriptivos asociados a la evaluación de inseguridad, estrés, irritación, entre otros aspectos en relación con el rubro de frustración en la escala NASA-TLX. ....	91
<b>Tabla 25.</b> Respuestas de los participantes para futuras versiones del sistema de visualización web de imágenes médicas. ....	92
<b>Tabla 26.</b> Respuestas de los usuarios con comentarios, sugerencias u observaciones generales del sistema de visualización web de imágenes médicas. ....	94
<b>Tabla 27.</b> Estructura generalizada del script HTML de una visualización.....	103
<b>Tabla 28.</b> Algoritmo de renderizado empleado con Three.js. ....	105

---

# Capítulo 1

## Introducción

---

La información es un insumo fundamental para la práctica médica, sin embargo, para obtenerla es necesario procesar y ordenar un conjunto de datos, de tal forma que se pueda adquirir conocimiento de ella. La tecnología de la computación ha tenido una gran influencia en la medicina, al incorporar hardware y software como herramientas, que permitan acercar la información al médico (Lifshitz, 2001), con el fin de aprovecharla mejor y ser de utilidad en cuanto a la salud de las personas. El extraordinario avance que ha experimentado en los últimos 30 años ha propiciado el desarrollo de instrumentos y técnicas eficientes, rápidas y accesibles, en cuanto a disponibilidad y costo, empleadas en diversos grados en la enseñanza, diagnóstico, planeación, asistencia, tratamiento, prevención e investigación en todas las áreas de la salud, lo que ha significado un cambio en la medicina del futuro.

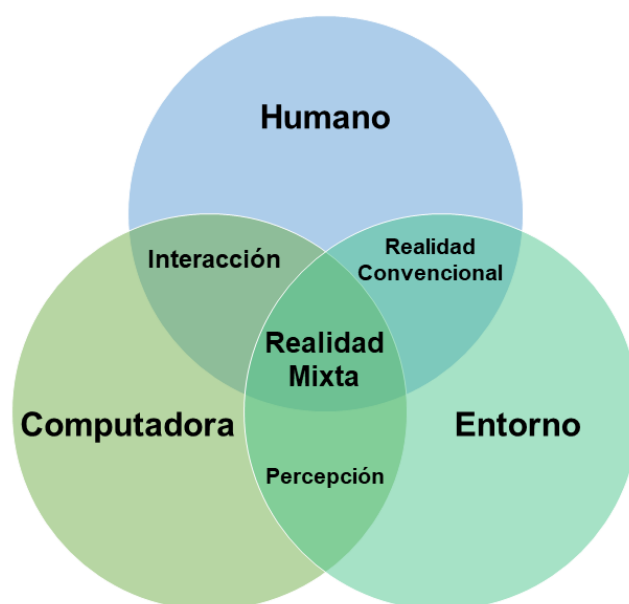
Una de las áreas que ha tenido un gran impacto en este campo ha sido la computación gráfica, ya que a partir del desarrollo, implementación y mejora continua de un conjunto de técnicas y algoritmos, ha sido posible representar y manipular datos de imagen generados por computadora, con el objetivo de generar, principalmente, modelos, visualizaciones y simulaciones genuinas, con efectos realistas y experiencias de usuario completas e inmersivas, lo cual ha jugado un papel importante en la medicina al poner a prueba varias aplicaciones, pues de esta manera los expertos disponen de mejores herramientas para desempeñar sus capacidades de una forma más rápida, además de brindar la oportunidad de realizar tareas que antes eran imposibles, riesgosas o en extremo imprecisas.

Los gráficos por computadora, son una forma de visualización de la información como un medio para poder obtener una visión global de las posibles relaciones entre los datos y las conclusiones que se pueden extraer mediante el análisis (Castro et al., 2010), lo que ha provocado que emerjan propuestas de interacción e interfaces entre las personas y ordenadores, tal es el caso de las interfaces de realidad mixta, las cuales buscan generar un entorno o ambiente para el usuario a partir de la correcta fusión de la información del mundo real y la información sintética y manejada por algún equipo.

## 1.1 Antecedentes

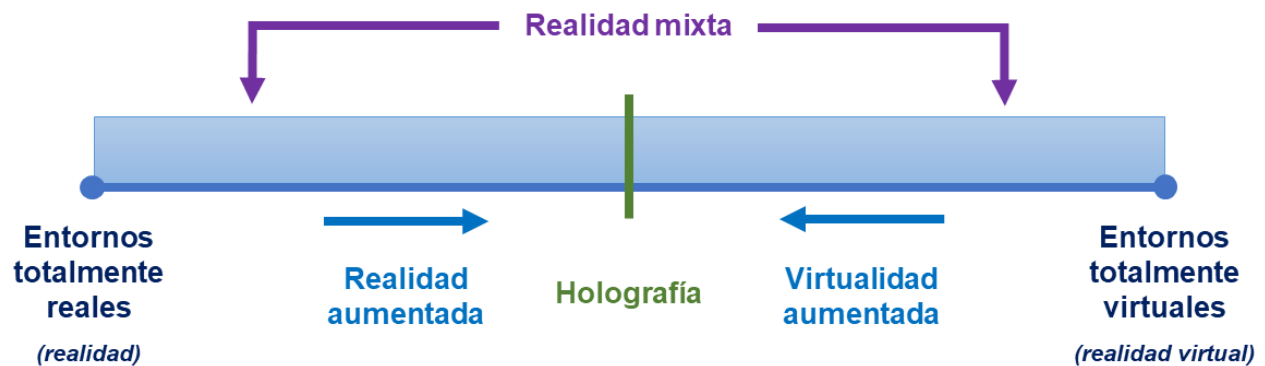
---

La interacción humano computadora, ha evolucionado desde hace siglo y medio, en donde el proceso de comunicación se ha vuelto más natural con el paso del tiempo. Actualmente, los sistemas de realidad mixta suponen la siguiente fase de medios de intercambio de información en distintos niveles, al combinar el procesamiento de los equipos, la entrada del usuario y la recepción del entorno (Figura 1), para crear auténticas experiencias, en donde se presenta la convergencia de la realidad física virtualmente mejorada y el espacio virtual físicamente persistente, creando un espacio colectivo (Rubio y Gertrudix, 2016), mejor conocido como metaverso.



**Figura 1.** Interacción entre equipos, seres humanos y entornos (Traducido de Microsoft, 2020).

El término de metaverso fue acuñado por primera vez en la novela *Snow Crash* (Stephenson, 1992) publicada en 1992, en donde Neal Stephenson se adelanta a su tiempo con este concepto al describir la relación entre la realidad y la virtualidad como un espacio colectivo y compartido, en donde los seres humanos interactúan, como avatares, unos con otros mediante agentes de software, en un espacio tridimensional que emplea una metáfora del mundo real (Peddie, 2016). Más tarde, en 1994, Paul Milgram y Fumio Kishino, definen el metaverso como el continuo de la virtualidad (Figura 2), el cual especifica que los ambientes totalmente reales y virtuales se encuentran en extremos opuestos, tal que representan los límites del continuo, por consiguiente, entre ellos existen todas las posibilidades de combinar ambos conceptos (Milgram et al., 1994).



**Figura 2.** Continuo de la virtualidad (Milgram et al., 1994).

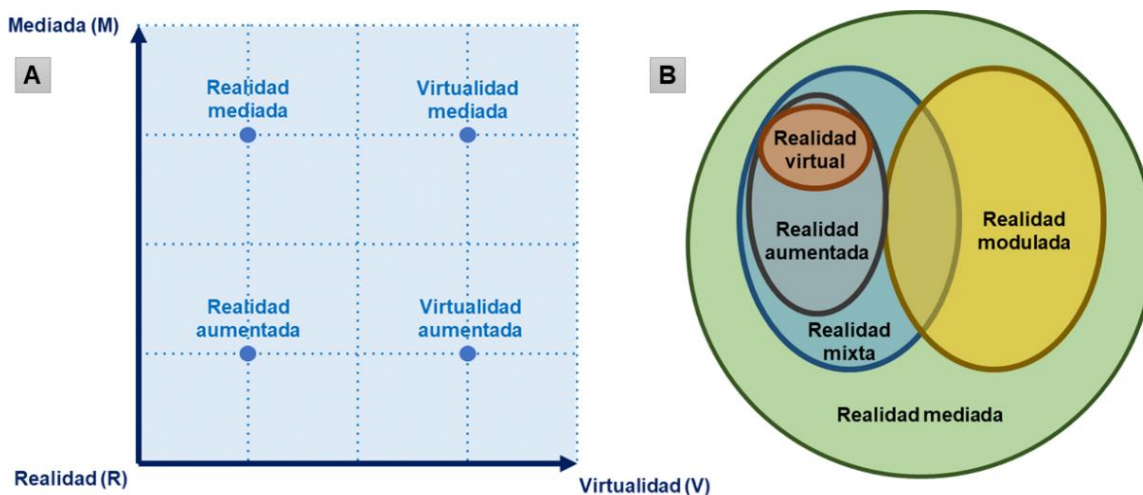
Dicho lo anterior, se tiene que la realidad (Figura 3 A) hace referencia a aquellos ambientes que están conformados solo por elementos reales, los cuales pueden o no ser observados a través de algún medio, mientras que la realidad virtual (Figura 3 D) hace referencia a aquellos ambientes que están conformados solamente por elementos virtuales, usualmente sintetizados por computadora. En el punto medio de esta línea continua se tienen entornos con una presencia proporcional de elementos del mundo real y virtual, lo que describe a la holografía (Figura 3 E) (Beléndez, 1996). Sin embargo, los medios conformados por elementos reales y virtuales que se superponen y combinan sin ninguna distinción, representan a la realidad mixta (Figura 3 B y C), término que logró definirse, por primera vez, a partir de la propuesta de Milgram y Kishino.



**Figura 3.** Ejemplos representativos del continuo de la virtualidad. A. Pontiac, Michigan (Jones, 2020). B. Compras con realidad aumentada (El Universal, 2020). C. Innovación virtual dentro del aula (Hernández, 2020). D. Reunión madre e hija fallecida en realidad virtual (Morales, 2020). E. Holograma Stephen Hawking (Kelion, 2018).

La realidad mixta, es una expresión que aún no es bien definida, ya que existen varias clases de ambientes que se pueden encontrar, por lo que este término no es comúnmente utilizado, el más usual es el de realidad aumentada. No obstante, la realidad mezclada establece dos subconjuntos: el primero, la virtualidad aumentada, la cual inserta vistas u objetos del mundo real en una escena virtual, y el segundo, la realidad aumentada, la cual inserta vistas u objetos virtuales en una escena del mundo real (Azuna et al., 2001), de tal manera que el continuo que existe entre estos dos grupos cubre los gráficos mejorados por vídeo y el vídeo mejorado por gráficos.

El continuo de la virtualidad, se limita a ir entre lo real y lo virtual, sin embargo, no considera la existencia de una amplia gama de dispositivos que modifican la percepción humana (Mann, 2001) de forma deliberada o accidental, y que no solo se agregan elementos a ella (Peddie, 2016). Por este motivo, Steve Mann, también en 1994, propone redefinir el metaverso como un marco general para la modificación artificial de la percepción humana por medio de dispositivos para aumentar (Starnier et al., 1997), disminuir deliberadamente, y más generalmente, para alterar de otra manera la entrada sensorial, al tener un mundo mediado por el aparato que se emplea (Mann, 1994; Mann, 1999), dando lugar a la realidad mediada, un continuo que va entre la realidad y la virtualidad (Figura 4).



**Figura 4.** Realidad mediada de Mann. (Traducido de Peddie, 2016). A. Taxonomía de realidad de Mann. B. Realidad mixta con realidad mediada.

### 1.1.1 Historia de la Realidad Mixta

Describir los orígenes de la realidad mixta, implica mencionar los desarrollos y avances de la realidad aumentada y la realidad virtual, debido a que estos conceptos están relacionados, y a

menudo se utilizan indistintamente. A continuación, se describen algunos de los eventos más relevantes:

**Tabla 1.** Historia de la realidad mixta.

Año	Desarrollo y/o avance
1901	Frank L. Baum define, en su novela <i>The Wonderful Wizard of Oz</i> , la idea de gafas electrónicas que superponen datos sobre personas que visualizan, al cual denominó <i>Charater Maker</i> (Baum, 1900).
1962	Mortin Heilig desarrolla el primer sistema de inmersión, conocido como <i>Sensorama</i> , capaz de mostrar imágenes estereoscópicas tridimensionales en gran angular con sensaciones visuales, aurales, olfativos y táctiles, con el objetivo de sumergir al usuario en un mundo alternativo lo más real posible (Carmigniani & Furht, 2011).
1965	Iván Sutherland estableció las bases de un sistema multisensorial basado en ordenador llamado <i>The Ultimate Display</i> (Sutherland, 1965).
1968	Iván Sutherland fabricó la <i>Sword of Damocles</i> , primer casco de realidad aumentada, capaz de generar imágenes de objetos tridimensionales y censar los movimientos del usuario para modificar la orientación de los objetos respecto a él; por lo que este dispositivo fue el precursor de los cascos de realidad virtual (Sutherland, 1968).
1974	Myron Krueger crea <i>Videoplace</i> , un laboratorio de realidad virtual que permite a los usuarios interactuar con objetos virtuales por primera vez (Krueger, 1997).
1980	Steve Mann crea la primera computadora portátil, un sistema de visión por ordenador con texto y superposiciones gráficas sobre una realidad mediada fotográficamente (Mann, 1997).
1981	Dan Reitan geomapea espacialmente múltiples imágenes de un radar meteorológico a mapas de realidad virtual de la Tierra y símbolos abstractos para transmisiones meteorológicas en televisión (Pandey et al., 2020).
1987	Douglas George y Robert Morris crean un prototipo funcional de un sistema de realidad aumentada basado en un telescopio astronómico, que se superponía en la lente del telescopio, por encima de las imágenes del cielo, de las estrellas y cuerpos celestiales, para ofrecer información adicional (George & Morris, 1989).
1989	Jaron Lanier, CEO de VPL, acuño el término de realidad virtual para reunir una amplia variedad de proyectos virtuales bajo una sola rúbrica (Steuer, 1992).
1990	Tom Caudell y David Mizzel, acuñan el término de realidad aumentada (Berryman, 2012) al referirse al mecanismo por el que los operarios electrónicos de las fábricas de aeronáutica recibían más información, proyectando un esquema de montaje (Caudell & Mizell, 1992).



- 
- 1992 Louis Rosenberg desarrolla uno de los primeros sistemas de realidad aumentada en funcionamiento llamado *Virtual Fixtures*, en el Laboratorio de Investigación de la Fuerza Aérea de los Estados Unidos, el cuál consistía en un exoesqueleto y un visor que componía el espacio del operador y un brazo robótico con una cámara que conformaba el entorno remoto (Rosenberg, 1992).
- 1992 Steven Feiner, Blair MacIntyre y Doree Seligmann, diseñan *KARMA*, un prototipo de realidad aumentada (Feiner et al., 1993).
- 1993 Loral WDL, en colaboración con el Departamento de Defensa de los Estados Unidos, enseña la primera demostración de tecnología de realidad aumentada combinada con vehículos pilotados en vivo y simuladores tripulados (Mellado, 2011).
- 1994 Julie Martin crea *Dancing in Cyberspace*, la primera producción teatral con acróbatas bailando alrededor de objetos virtuales en un escenario (Singh, 2012).
- 1995 Ravela y otros colaboradores en la Universidad de Massachusetts, introducen un sistema visual usando cámaras monoculares para rastrear objetos, a través de vistas para la realidad aumentada (Ravela et al., 1994).
- 1999 Hirpkazu Kato desarrolla *ARToolKit*, una biblioteca que permite la creación de aplicaciones de realidad aumentada en el *HitLab* del Instituto de Ciencia y Tecnología de la Nasa (Kato, 1999).
- 2000 Bruce H. Thomas desarrolla *ARQuake*, primer juego al aire libre con dispositivos móviles de realidad aumentada (Thomas et al., 2000).
- 2004 *Trimble Navigation* y el *Human Interface Technology Laboratory*, muestran un sistema de realidad aumentada montando en casco para exteriores (Reitmayr & Schmalstieg, 2004).
- 2008 *RA Wikitude*, una guía de viaje con realidad aumentada es lanzada para *smartphones (Android G1)* (Chapman et al., 2009).
- 2009 *AR Toolkit* es portado a *Adobe Flash (FLARToolkit)* por Saqoosha, por lo que la realidad aumentada llega al navegador web (Koyama, 2009).
- 2015 Microsoft anuncia *Windows Holographic* y *HoloLens*, unas gafas de realidad aumentada, las cuales utilizan varios sensores y una computadora para mezclar hologramas de alta definición con el mundo real (Ludlow, 2015).
- 2016 Niantic pone disponible para dispositivos móviles *iOS* y *Android Pokemon GO*, videojuego que permite a través de la cámara visualizar animaciones de objetos superpuestos en el vídeo del mundo real, así como permitir la interacción por medio de la pantalla del dispositivo (Bond, 2016).
-

Es importante destacar, que el crecimiento y progreso de la realidad mixta ha sido notable a partir de finales de la década de 1990, ya que comenzaron diversas conferencias como el taller y simposio internacional sobre realidad aumentada, el simposio internacional sobre realidad mixta y diseño de realidad aumentada, y el taller de diseño de ambientes de realidad aumentada. Al mismo tiempo, se formaron sociedades como el Laboratorio de Sistemas de Realidad Mixta (*MRLab*) en Japón y el Consorcio *Arvika* en Alemania; y entidades ya existentes como la *IEEE Computer Society*, la *IEEE Visualizations and Graphics Technical Committee* y la *Association for Computing Machinery*, muestran un mayor interés en tecnologías específicas de realidad extendida (Azuma, 2001).

## **1.2 Realidad Mixta Aplicada en Medicina**

---

La aplicación de la realidad mezclada en el campo de la medicina, como se había mencionado anteriormente, es muy amplia, no obstante, las principales áreas en las que ha destacado su uso va dirigida al entrenamiento de habilidades médicas, la planeación de procedimientos quirúrgicos y la asistencia médica (León, 2016), con el objetivo de disponer de herramientas para la mejora de las competencias de los médicos (Aguilar et al., 2016), reducir la duración y el material de las intervenciones quirúrgicas, evitar sistemas de radiación, localizar afecciones anatómicas con mayor facilidad y planificar intervenciones de manera más precisa y exacta (Roberts et al., 2014), ya que la realidad mediada permite la flexibilidad para crear escenarios dinámicos y específicos, lo cual permite el seguimiento y minimización de errores, para automatizar diversos procesos médicos.

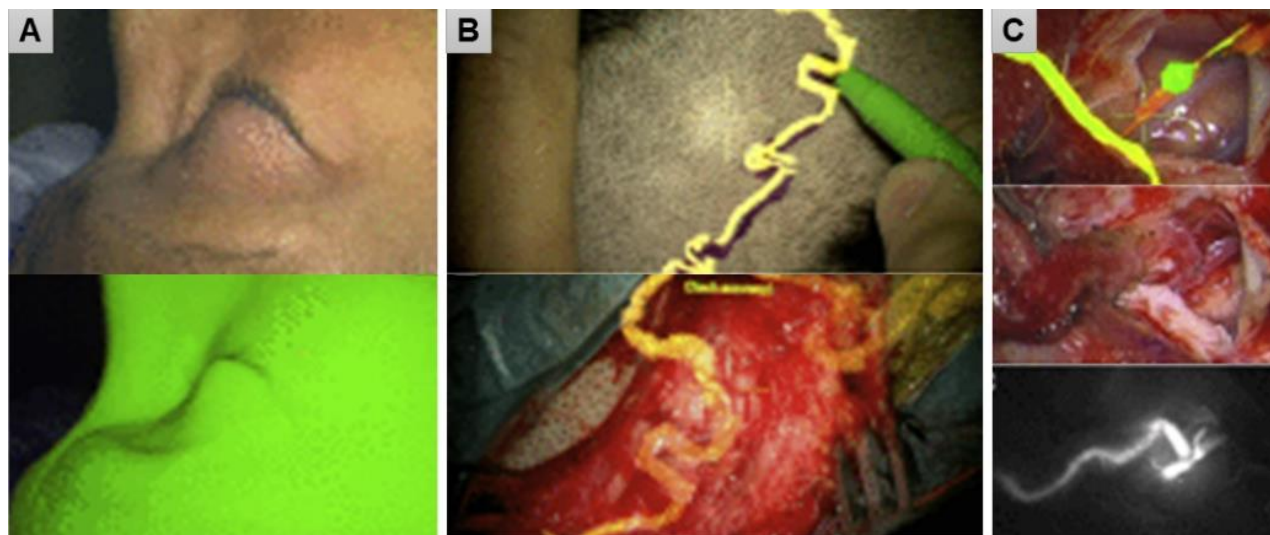
El reto de los desarrollos que emplean realidad mixta busca la representación real de la anatomía, así como la creación de entornos tridimensionales reales y hasta cierto punto inmersivos, por tanto, los siguientes casos de estudio que se exponen, corresponden a solo algunos de los diversos y bastos reportes de sistemas creados, desarrollados e implementados tanto en México como en el mundo.

### **1.2.1 Realidad Mixta Aplicada en Medicina en el Mundo**

#### **i. Sistema de Realidad Aumentada para Cirugía Cerebral Asistida**

El sistema de cirugía de derivación cerebral asistida por realidad aumentada (Cabrilo et al., 2015), es un desarrollo realizado en la División de Neurocirugía del Centro Médico de la Universidad de

Ginebra, Suiza en 2015, en donde Ivan Cabrilo, Karl Schaller y Philippe Bijlenga proponen evaluar la utilidad de una configuración basada en realidad aumentada, para recolectar vasos donantes a través de su localización precisa en tiempo real. La aplicación de este método considero a cuatro pacientes seleccionados con algún padecimiento en las arterias cerebrales, de tal forma que la anatomía de la cabeza, cráneo y vasos extracraneales e intracraneales de cada uno de los pacientes fueron segmentados preoperatoriamente a partir de conjuntos de imagen en tres dimensiones, y posteriormente inyectados intraoperatoriamente en el ocular del microscopio operativo para la orientación de imágenes, lo cual permitió optimizar el flujo de trabajo en este tipo de procedimientos mínimamente invasivos, al proporcionar información anatómica esencial, totalmente integrada en el campo quirúrgico. La Figura 5 muestra el sistema de realidad aumentada en funcionamiento.

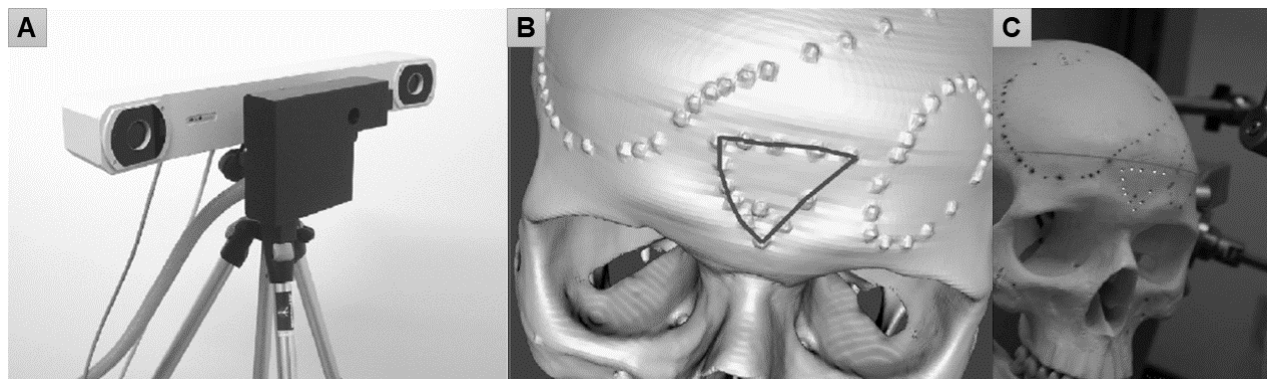


**Figura 5.** Cirugía de bypass asistido por realidad aumentada. (Cabrilo et al., 2015). A. Paciente y reconstrucción. B. Visualización de vasos reconstruidos a través del ocular del microscopio operatorio, antes y después de un corte. C. Visualización de vasos reconstruidos, vasos en tiempo real y vasos en una tomografía computarizada.

## ii. Sistema de Realidad Aumentada de Proyección Láser

El sistema de realidad aumentada de proyección láser para cirugía asistida por computadora (Glossop et al., 2003), fue presentado en la conferencia internacional *Medical Image Computing and Computer – Assisted Intervention (MICCAI)*, llevada a cabo en Montreal, Canadá en 2003, en donde Glossop y colaboradores presentaron la evaluación de un nuevo dispositivo de realidad aumentada, el cual utiliza láseres infrarrojos escaneados y visibles para proyectar información generada por una computadora, con el fin de poder realizar diversos planes quirúrgicos que

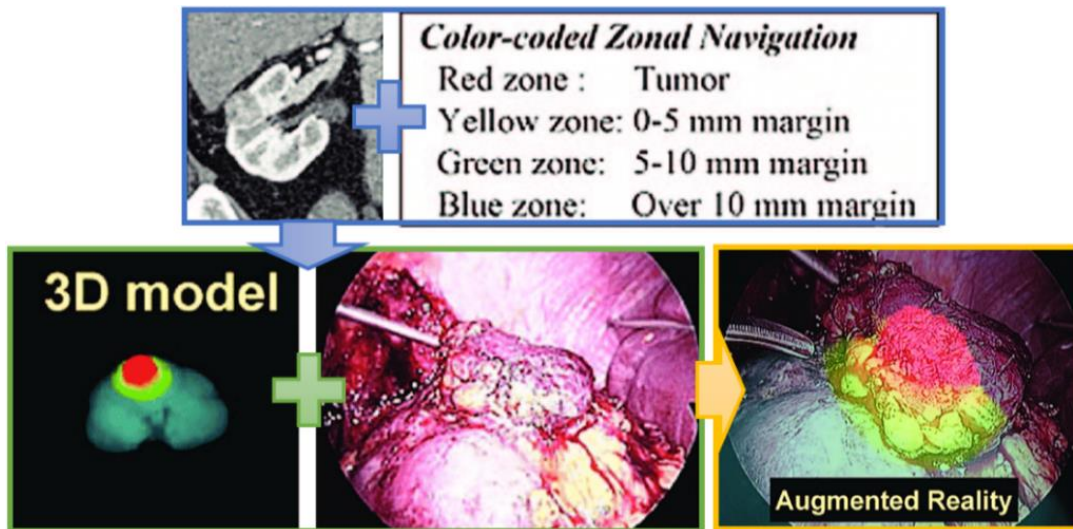
incluyen la identificación de estructuras anatómicas críticas, visualización de rutas y objetivos preplanificados y teledirigida, directamente sobre el paciente, al integrar una cámara 3D para realizar mediciones en tiempo real y el uso de una propia interfaz (API). La Figura 6 muestra el sistema de realidad aumentada en funcionamiento.



**Figura 6.** Sistema de realidad aumentada de proyección láser para cirugía asistida por ordenador. A. Proyector Xar Trax montado en la cámara Polaris y el trípode. B. TC reconstruida y trayecto de craneotomía planificada. C. Patrón de proyección. (Glossop et al., 2003).

### iii. Sistema de Visualización de Realidad Aumentada en Urología

El sistema de cirugía endoscópica asistida por imágenes (Ukimura & Gill, 2008), empleado en el Instituto de Urología Glickman en Cleveland, Ohio en 2008, supone el primer uso clínico de la tecnología de realidad aumentada en urología, por lo que Osamu Ukimura y Inderbir Gill describen la experiencia inicial de la asistencia de imagen por computadora en procedimientos de mínima invasión en esta área (Figura 7), al emplear un sistema compuesto por una equipo y un localizador que permiten la ubicación espacial de la posición de los diversos instrumentos quirúrgicos, al utilizar un sensor magnético y óptico, así como la aplicación clínica de un sistema de visualización de realidad aumentada que ayuda al cirujano laparoscópico a entender las anatomías 3D del cáncer o vitales más allá de la vista quirúrgica, con el fin de aumentar la precisión y la confianza del cirujano, al proporcionar datos oncológicos preoperatorios y la comprensión de las anatomías vitales circundantes.



**Figura 7.** Cirugía endoscópica asistida por imágenes. (Ukimura & Gill, 2008). A partir de los datos de tomografía computarizada originales basados en un sistema de navegación codificada en cuatro colores, se realiza la sincronización del modelo 3D con la vista quirúrgica real mediante el sistema de realidad aumentada.

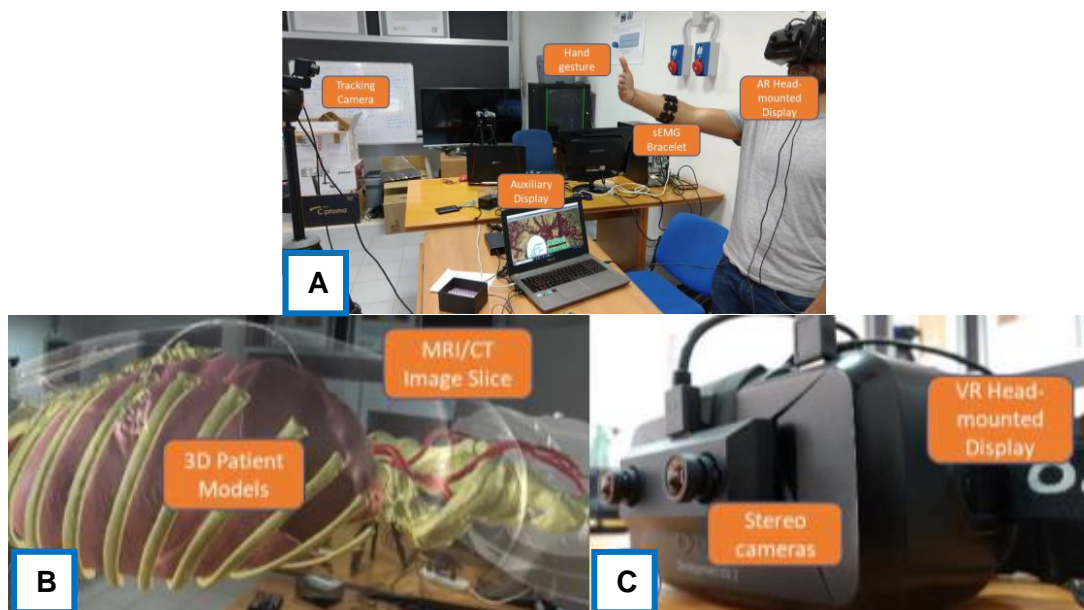
## 1.2.2 Realidad Mixta Aplicada en la Medicina en México

### i. Sistema de Navegación de Realidad Aumentada

El sistema de realidad aumentada con navegación basada en gestos manuales para aplicaciones en cirugía guiada por imágenes (De Paolis et al., 2020), fue desarrollado en conjunto por el Instituto de Ciencias Aplicadas y Tecnológica (ICAT) de la UNAM en México y el Departamento de Ingeniería para la Innovación de la Universidad de Salento en Italia, el cuál plantea un sistema de planificación quirúrgica basada en imágenes para cirugías de mínima invasión (Figura 8), en donde es posible realizar la visualización de una escena aumentada de estructuras anatómicas de interés, reconstruidas a partir de estudios de resonancia magnética (MRI) y tomografía computarizada (TC), así como la navegación y exploración de los modelos anatómicos en tiempo real, en otras palabras, se tiene un sistema en donde el usuario puede visualizar modelos 3D de órganos e interactuar con ellos.

La visualización, navegación e interacción de este sistema es posible por el proceso de seguimiento, alineación y renderizado, a través de la interacción de tres dispositivos, los cuales son: brazalete *Myo*, permite reconocer los gestos de los usuarios y proporcionar al sistema los comandos empleados para manipular las transformaciones de los modelos virtuales; el casco *Oculus Rift*, permiten proporcionar una visualización de una escena aumentada al usuario al definir

una referencia entre el espacio de los modelos virtuales y la posición de la cabeza del usuario, y cámara estéreo *OVRVision Pro*, permiten obtener una vista del entorno real en donde se encuentra el usuario.



**Figura 8.** Sistema de navegación de realidad aumentada. (De Paolis et al., 2020). A. Elementos del sistema: brazaletes Myo, casco de realidad virtual Oculus Rift y cámara estéreo *OVRVision Pro*. B. Escena de realidad aumentada de un modelo reconstruido para la planificación de una cirugía cardiovascular. C. Casco de realidad virtual extendido.

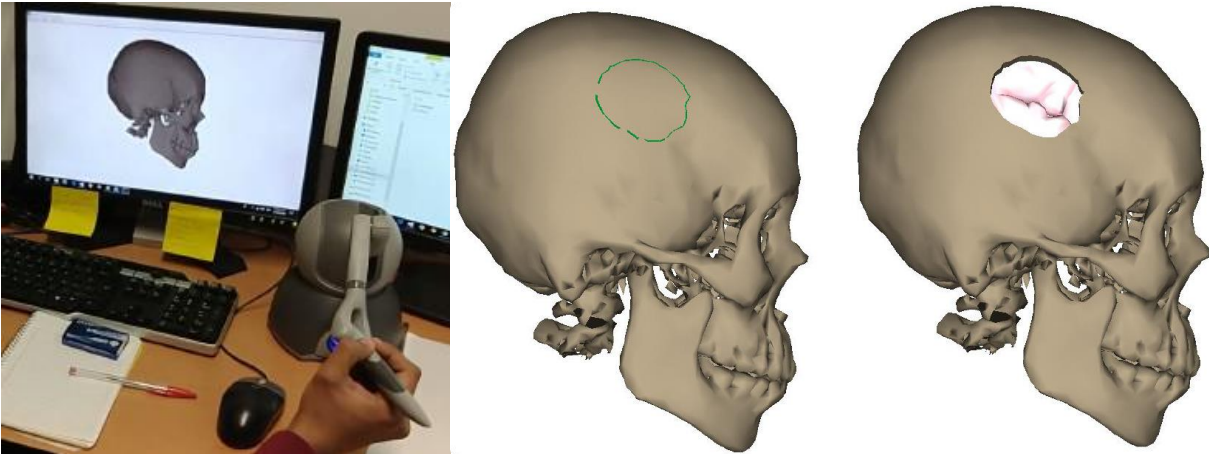
## ii. Simulación Computacional de Cortes

En el 2018, Mauricio Garduño desarrolló un simulador computarizado para el entrenamiento médico en el procedimiento de corte de hueso craneal (craneotomía) con retroalimentación háptica (Garduño, 2018), el cual fue evaluado por especialistas del Hospital General de México (HGM) y la Unidad de Investigación y Desarrollo Tecnológico de la UNAM (UIDT). El trabajo plantea una plataforma de software que representa un escenario idealizado, conformado por un modelo tridimensional del cráneo, reconstruido a partir de imágenes de angiografía por tomografía computarizada (ATC), y el modelo de herramientas quirúrgicas necesarias para la simulación, de tal forma que el conjunto de acciones realizadas son controladas por el flujo del programa, a través de dispositivos de interacción, que se encargan de censar cada uno de los movimientos del usuario, para proveer al sistema información en tiempo real (Figura 9).

Este sistema fue desarrollado con lenguaje C++, el *SDK OpenHaptics* y el dispositivo háptico *Geomagic Phantom Omni* de la compañía *3DSYSTEM*. Cabe destacar que este simulador



está estructurado en un conjunto de bloques de software funcionales, con el fin de retroalimentar al usuario con estímulos a partir de la interacción: visual, se procesan elementos visuales para desplegar el contenido gráfico en un monitor; aural, se proporciona un estímulo auditivo al usuario, y háptica, se proporciona un estímulo táctil al usuario, entre la interacción del modelo virtual, el dispositivo y las fuerzas que se ven reflejadas.



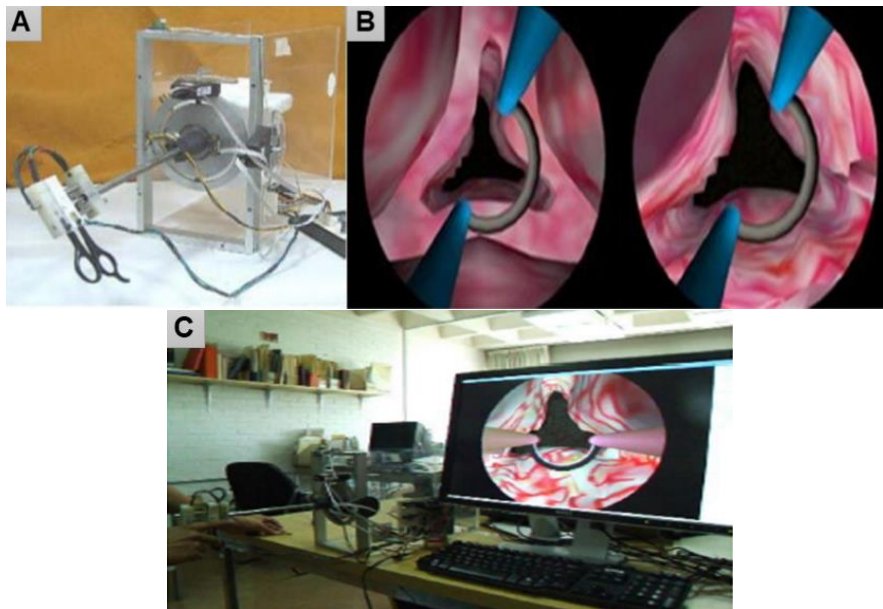
**Figura 9.** Simulador de craneotomía funcionando con la interfaz háptica (Garduño, 2018).

### iii. Sistema de Simulación de Cirugía de Próstata.

El grupo de Análisis de Imágenes y Visualización del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), de la UNAM en 2008 (Teodoro, 2008), desarrolló un sistema de simulación del procedimiento de Resección Transuretral de Próstata (RTUP), con el fin de proporcionar una herramienta para entrenamiento individual, así como un ambiente de enseñanza colectivo en el observatorio de visualización IXTLI, con los elementos necesarios para llevar a cabo la exploración endoscópica, al aportar dos metodologías: una para la reconstrucción de modelos tridimensionales a partir de imágenes anatómicas, y otra para la creación de gráfica para la aplicación sobre el simulador; así como algunas técnicas para efectos visuales. Este sistema está compuesto tres secciones (Figura 10):

- Interfaz mecatrónica, simula los efectos que produce una herramienta de corte quirúrgico real, al estar conformada por un dispositivo mecánico que emula un resectoscopio, y una interfaz electrónica que convierte las señales analógicas y digitales de los sensores situados en el dispositivo, en una señal digital con la información, a través de un protocolo de comunicación serial con el ambiente gráfico, para transportarla a la estación de trabajo.

- Estación de trabajo, transporta el mundo real a un mundo virtual al realizar el cálculo de las deformaciones, optimización de cálculos con GPU/CPU y procesos recurrentes que se tienen en el sistema.
- Estación gráfica, permite al usuario observar un ambiente virtual que simula el procedimiento de RTUP, el cual está basado en un sistema gráfico de modelos 3D con *OpenGL* y algoritmos de deformación de malla de volumen y el procesamiento de todas las variables físicas que intervienen en el proceso de simulación en tiempo real.



**Figura 10.** Sistema de simulación de cirugía de próstata. (Teodoro, 2008). A. Interfaz mecatrónica completa. B. Simulador gráfico de RTUP en ejecución en tiempo real. C. Simulador de realidad virtual del RTUP completo.

### 1.3 Sistemas de Visualización Web para Imágenes Médicas

---

Las imágenes médicas desempeñan un papel crucial en la atención sanitaria en todos los niveles, pues sin ellas los médicos tendrían que recurrir a métodos de diagnóstico y planeación invasivos con más frecuencia (Salmon, 1993), y el progreso durante el tratamiento sería más difícil o imposible, además de que no se tendrían referencias suficientes (Marín et al., 2019). El diagnóstico médico basado en imágenes hasta principios de 1970 era realizado por radiografía convencional (Mahmoudi et al., 2010), no obstante, se tenían numerosas limitaciones, ya que no se podía representar en una película de dos dimensiones toda la información de las estructuras sin que esta



quedará superpuesta; dado que solo se tiene la capacidad de registrar en una placa radiográfica ciertos tejidos debido a su densidad (Arguiñarena et al., 2015).

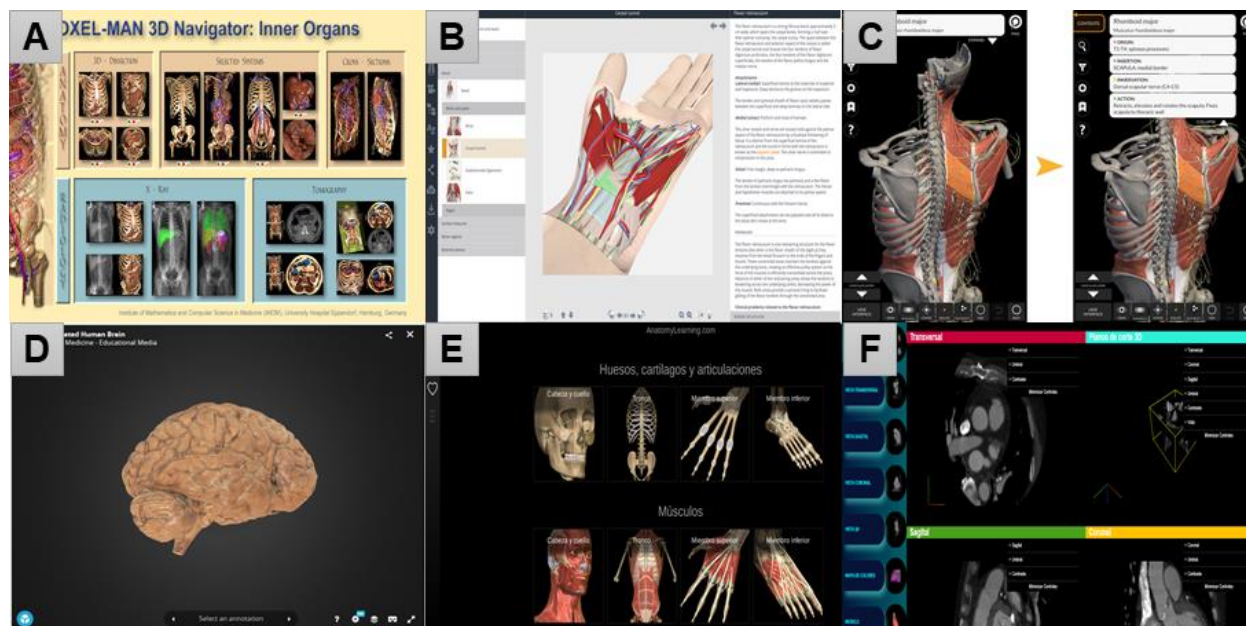
Además, el manejo y almacenamiento de las imágenes en medios impresos (papel o placas), no eran los más apropiados por la demanda que con el tiempo iba incrementando. En tal sentido, el nacimiento de la tomografía computarizada (TC) introdujo un mejor medio para el diagnóstico basado en imágenes digitales (Adame et al., 2000), lo que provocó el desarrollo de diferentes técnicas de obtención de imágenes, la generación de diferentes tipos de imágenes médicas digitales para el diagnóstico y medios de almacenamiento más apropiados. En la Tabla 2 se muestra la relación de imágenes médicas más comunes.

**Tabla 2.** *Imágenes médicas más comunes (García, 2010).*

Año	Imagen médica	Técnica	Dispositivo
1895	Radiografía	Radiación ionizante en una proyección	Tubo de rayos X
1950	Fluoroscopia	Radiación ionizante con medios de contraste en una proyección	Fluoroscopio
1961	Tomografía por Emisión de Positrones (PET)	Radiación gamma	Emisor de positrones
1970	Ultrasonido	Uso de ondas acústicas de alta frecuencia	Ecógrafo
1972	Tomografía Computarizada (TC)	Radiación ionizante con/sin medios de contraste en múltiples proyecciones	Tomógrafo
1976	Imagen por Resonancia Magnética (MRI)	Campos magnéticos	Equipo de MRI

Por su parte, la Web es un sistema que funciona a través de Internet, por lo que es considerada como una plataforma independiente que permite crear y compartir (Roberts, 2014), entre otras cosas, visualizaciones con presencia en todas partes y todo el tiempo, lo que hace posible llevar a cabo varias formas de colaboración según se requiera. Al considerar, la relevancia de la Web como un medio de comunicación, así como las diversas herramientas, bibliotecas, interfaces de programación de aplicaciones (API) y entornos de trabajo (*frameworks*), disponibles para crear contenido web, ha permitido el desarrollo de sistemas de visualización web de imágenes médicas, los cuáles se centran en la visualización anatómica en la enseñanza médica y en el

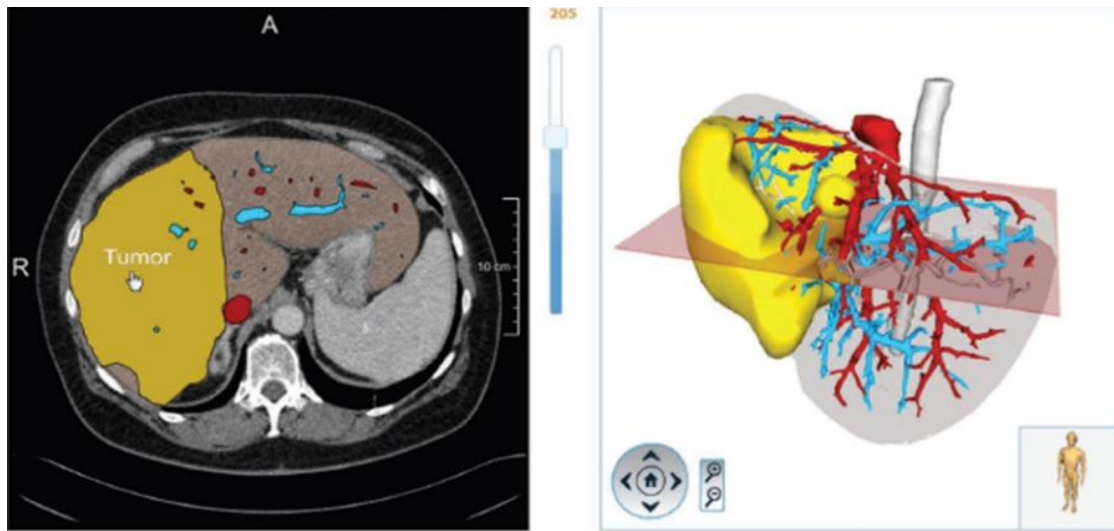
entrenamiento básico para cirugía (Birr et al., 2013), algunos ejemplos se muestran en la Figura 11, y a continuación, se especifican dos implementaciones con más detalle:



**Figura 11.** Sistemas de visualización web de imágenes médicas. A. *Voxel-Man 3D-Navigator: Inner Organs*. (VOXEL-MAN, 2021). B. *PrimalPictures 3D Human Anatomy* (PRIMAL PICTURES, 2021). C. *Anatomy 3D Atlas* (Anatomy3DAtlas, 2013). D. *Functional Neuroanatomy* (UBC, 2018). E. *AnatomyLearning* (AL, 2021). F. *MEDVIAS* (UNAM, 2021)

### 1.3.1 LiverAnatomyExplorer

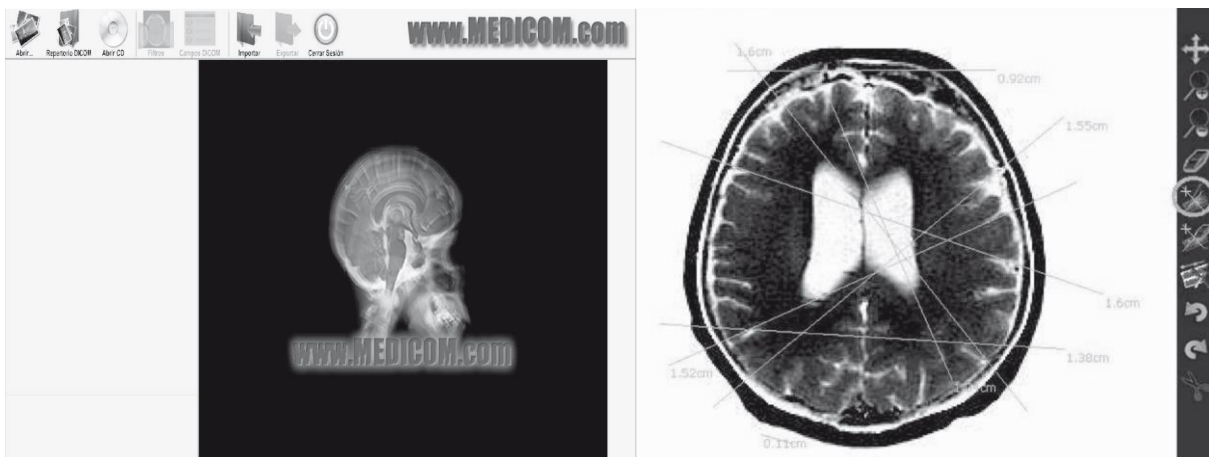
El sistema *LiverAnatomyExplorer* (Birr et al., 2013), es una herramienta de enseñanza quirúrgica en tiempo real basada en tecnologías Web de vanguardia como SVG (*Scalable Vector Graphics*), X3D (*Extensible3D*) y WebGL (*Web Graphics Library*), la cual combina imágenes clínicas 2D tradicionales (TC y MRI) con modelos 3D interactivos basados en la Web derivados de datos de imágenes específicos del paciente (Figura 12). Además, la herramienta se ve reforzada por vídeos quirúrgicos, autoevaluación y creación en línea, de tal forma que los instructores pueden administrar los estudios de caso presentados y crear cuestionarios de opción múltiple.



**Figura 12.** Herramienta de enseñanza quirúrgica basada en WebGL en 2D y 3D (Birr et al., 2013).

### 1.3.2 MEDICOMWEB

El sistema *MEDICOMWEB* es un desarrollo orientado a la Web (Angarita y Beltrán, 2016) bajo una estructura de datos basada en el modelo del estándar DICOM, un conjunto de herramientas (aumento, medidas y filtros) para visualización y análisis, un sistema de exploración y navegación intuitiva de la colección de imágenes, accesible a través de la Web (Figura 13) con cualquier navegador y otros valores agregados, implementado con HTML, AJAX, *JavaScript* (JS), Java y Toolkit.



**Figura 13.** Sistema *MEDICOMWEB* (Angarita y Beltrán, 2016).

Actualmente, las aplicaciones de este tipo que se han realizado a través de la Web son limitadas por el alcance, contenido y disponibilidad libre para acceder a ellas, y si extrapolamos la

incorporación de conceptos como realidad mixta a los mismos, se tiene un número reducido de desarrollos, sin embargo, representan una solución competitiva y remota con respecto a soluciones existentes en el estado del arte, por lo que en la Tabla 3 se muestra una comparativa de las principales características de las aplicaciones ejemplo descritas anteriormente y las indicadas en la Figura 11.

**Tabla 3.** Comparativa de las principales características de las aplicaciones de visualización de contenido médico que se centran en la anatomía humana.

Característica	Liver Anatomy Explorer	Medicom Web	Voxel-Man	Primal Pictures	Anatomy 3D Atlas	Functional Neuroanatomy	Anatomy Learning	MEDVIAS
Plataforma Web	✓	✓	✓	✓	✗	✓	✓	✓
Aplicación móvil	✗	✗	✗	✓	✓	✗	✓	✓
Realidad Mixta	✗	✗	✓	✗	✗	✗	✗	✓
Tipo contenido anatómico	Específico	Variado	Variado	Variado	Variado	Específico	Variado	Variado

De modo que, cada uno de los aspectos que se han descrito en las secciones anteriores, sustentan la importancia de los sistemas de realidad extendida y de visualización web en la medicina, para fines de la presente tesis, al considerar las respectivas mejoras de cada uno de los sistemas, dadas las implementaciones que se tienen hasta el momento, así como el nuevo reto, para integrar ambas ideas, y de esta manera desarrollar un sistema más robusto, accesible, aumentar la calidad del contenido y con alcances en otras áreas de la medicina como la planeación quirúrgica.

---

## Capítulo 2

### Corazón

---

Las enfermedades cardiovasculares son un conjunto de trastornos del corazón y de los vasos sanguíneos, las cuales desde hace 20 años son la principal causa de muerte a nivel mundial, al representar el 16% del total de muertes debidas a todas las causas, puesto que al menos una de cada tres personas en el mundo pierde la vida por alguna patología relacionada a este tipo de afecciones y de forma prematura en personas menores de 70 años; en donde se estima que para el año 2030 aumentará a un 20%, debido al incremento de su prevalencia desproporcional en países de ingresos económicos bajos y medios, ya que suponen un 75% de estas defunciones (OMS, 2021); en los países más desarrollados, en cambio, la mortalidad por dichas enfermedades está descendiendo debido a la mejora de la asistencia sanitaria (Corella y Ordovás, 2007) .

En México, este tipo de padecimientos son el principal grupo de causas de muerte desde 1990, en donde la mayor mortalidad se ha observado en los hombres y son consideradas las principales causas de años de vida perdidos en el país (Gómez et al, 2016). Se ha visto que más del 60% de la población adulta en el país presenta al menos algún factor de riesgo para padecer alguna afección asociada a una enfermedad cardiovascular, que junto al proceso de envejecimiento de la población genera, por sí solo, un incremento de enfermedades asociadas a la edad (Sánchez et al, 2014), lo cual supone un panorama futuro en que la mortalidad para estas causas tendrá una tendencia al aumento en los próximos años.

Las principales enfermedades cardiovasculares tienen relación con afecciones de los vasos sanguíneos, el músculo y válvulas cardíacas, problemas en el ritmo cardíaco, defectos de nacimiento, así como infecciones en el corazón. Los síntomas más comunes en cardiopatías consisten en: dolor o molestias en el pecho, dolor o molestias en los brazos, hombro izquierdo, mandíbula o espalda, dificultad para respirar, náuseas o vómitos, dificultad para caminar, mareos o desmayos, pérdida de equilibrio o coordinación, sudores fríos y palidez, aparición súbita, generalmente unilateral, de entumecimiento en la cara, piernas o brazos, confusión, dificultad para hablar o comprender lo que se dice, y/o problemas visuales en uno o ambos ojos (OMS, 2021).

Ante un problema de tal magnitud los esfuerzos no deben centrarse sólo en la mejora de los tratamientos, sino también en la prevención, puesto que el 80% de este tipo de enfermedades se pueden prever, ya que en ellas interaccionan la susceptibilidad genética del individuo con los

factores ambientales a los que se halla expuesto a lo largo de su vida, sobre todo, la dieta malsana, la inactividad física, el consumo de tabaco y alcohol, aunado a factores de riesgo manifestados en personas con hipertensión arterial, hiperglucemia, hiperlipidemia, sobrepeso u obesidad (Sánchez et al., 2016). Además, una serie de determinantes subyacentes en este tipo de padecimientos son un reflejo de las principales fuerzas que rigen los cambios sociales, económicos y culturales: la globalización, la urbanización y el envejecimiento de la población. Por tanto, el diagnóstico y tratamiento oportuno representan una vía para atacar este problema de salud; y en caso de requerir algún tipo de intervención quirúrgica, los procedimientos de mínima invasión han supuesto una manera de disminuir el riesgo de sangrado, morbilidad y mortalidad en los pacientes (Sánchez et al., 2004).

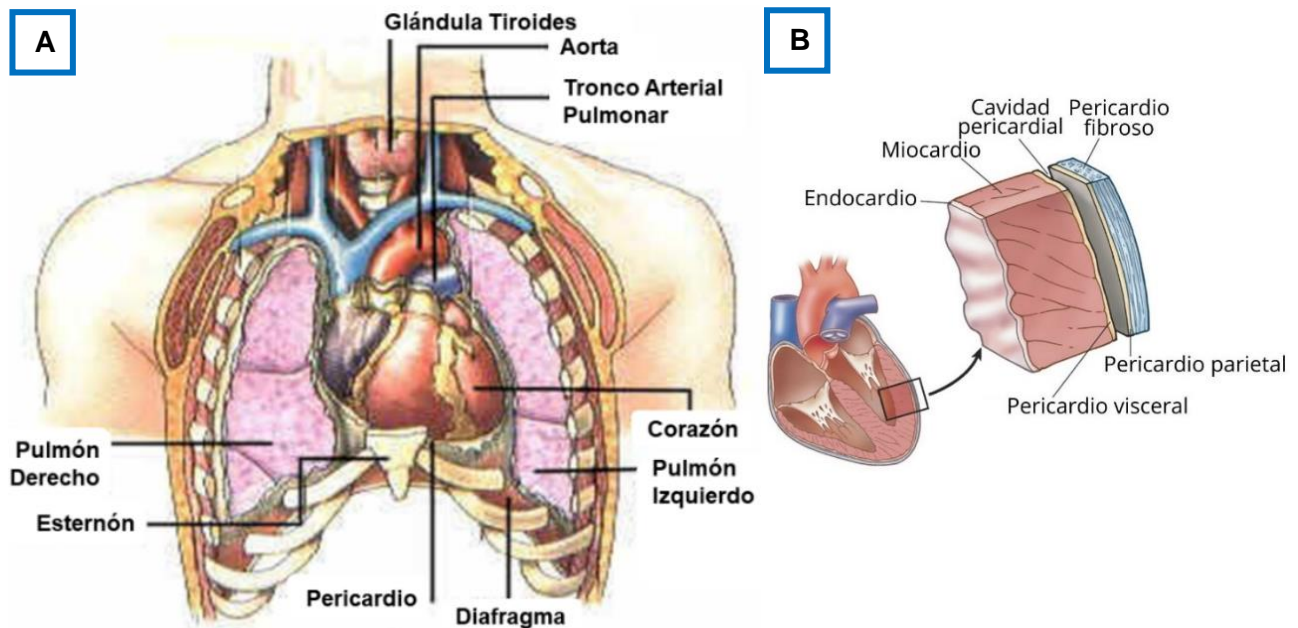
## 2.1 Morfología y Fisiología del Corazón

---

El corazón es un víscera muscular hueco y el órgano central del sistema cardiovascular, localizado en la cavidad torácica, en el mediastino inferior medio, situado a través del pericardio sobre el centro tendinoso del diafragma (Figura 14 A), en donde su importancia es tal que es el encargado de bombear la sangre oxigenada a todos los rincones del organismo y la sangre desoxigenada a los pulmones a través de los vasos sanguíneos, el cuál presenta un tamaño variable respecto a la edad, sexo y superficie corporal del individuo, no obstante, en una persona adulta el promedio oscila entre los 220–300 gramos (De los Nietos, 2007).

Se encuentra envuelto por el pericardio, una bolsa serosa que constituye la capa externa del corazón, y dos hojas, visceral y parietal, entre las cuales existe una pequeña cantidad de líquido lubricante que permite entre ellas su deslizamiento, así como los movimientos cardíacos. La pared del corazón se compone de tres capas (Figura 14 B) (Anderson et al., 2004):

- *Epicardio*: Cubre el corazón, envuelve las raíces de los grandes vasos sanguíneos y adhiere la pared del corazón a un saco protector.
- *Miocardio*: Músculo cardíaco que constituye la mayor parte del grosor de la pared y es responsable del bombeo de sangre.
- *Endocardio*: Reviste las cavidades internas del corazón y cubre las válvulas cardíacas.



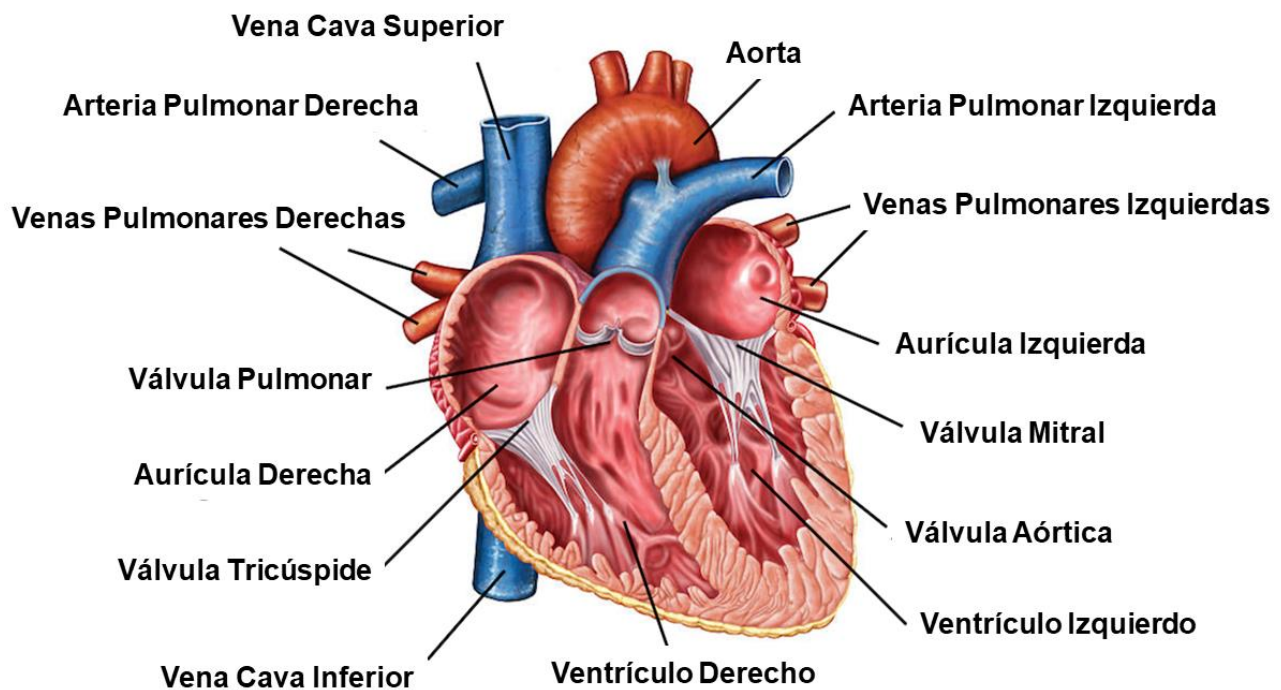
**Figura 14.** Ubicación y generalidades del corazón. A. Ubicación del corazón (Wyngarden & Smith, 1998). B. Capas del corazón (Gartner & Hiatt, 2006).

El interior del corazón está formado por cuatro cavidades:

- Dos *ventrículos*: Tienen la función de expulsar la sangre fuera del corazón a través de las arterias.
- Dos *aurículas*: Tienen la función de recibir la sangre que viene fuera del corazón a través de las venas.

Cabe destacar que el corazón presenta bilateralidad, de forma que tiene una aurícula y un ventrículo a la derecha, y otro par de cámaras a la izquierda, las cuales no se comunican. Las primeras, bombean la sangre desde la circulación sistémica, a través de las venas cavas superior e inferior, hasta la circulación pulmonar; mientras que las últimas, bombean la sangre que llega desde la circulación pulmonar a la circulación sistémica (Xin et al., 2013). Estas cavidades están separadas entre sí, externamente por el surco interauricular e interventricular que se unen en la cruz del corazón, e internamente por el septum o tabique interauricular e interventricular que constan de una parte muscular y otra fibrosa.



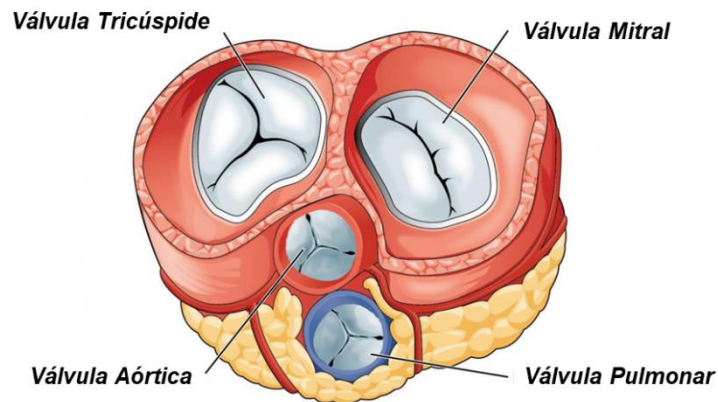


**Figura 15.** Anatomía corazón: principales partes que lo conforman (Editado de Gartner & Hiatt, 2006).

Presenta cuatro estructuras valvulares que regulan y mantienen el flujo de la sangre por el corazón y hacia afuera del mismo, en donde cada válvula está formada por un grupo de pliegues, o valvas, que se abren y cierran mientras el corazón se contrae y se dilata (Guindo, 2004), las cuales son:

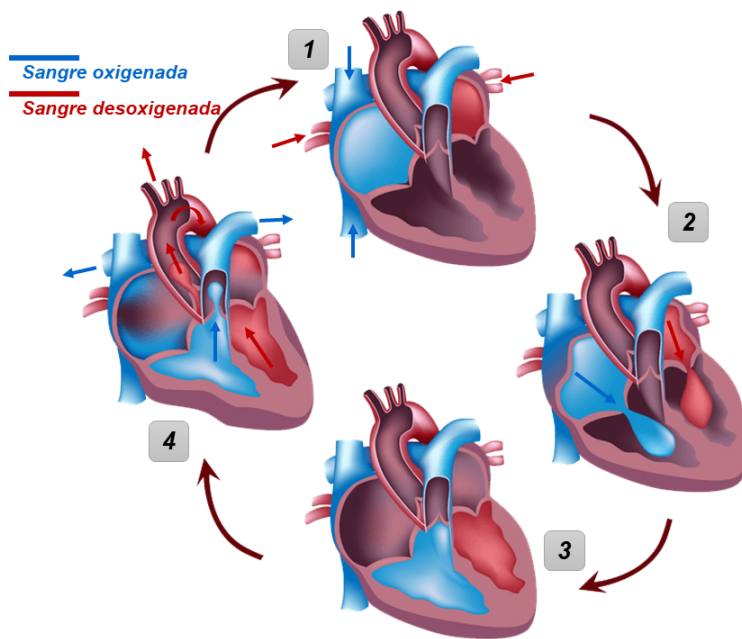
- Dos *auriculoventriculares*: Comunican cada aurícula con su ventrículo.
  - *Mitral*, tiene dos valvas y se encuentra entre aurícula y ventrículo izquierdo.
  - *Tricúspide*, tiene tres valvas y se encuentra entre aurícula y ventrículo derecho.
  
- Dos *sigmoideas*: También conocidas como semilunares.
  - *Aórtica*, tiene tres valvas y se encuentra la salida del ventrículo izquierdo, por lo que se encarga de manejar el flujo de sangre desde este hacia la aorta.
  - *Pulmonar*, tiene tres valvas y se encuentra la salida del ventrículo derecho, por lo que se encarga de manejar el flujo de sangre desde este por el tronco de la arteria pulmonar hacia las arterias pulmonares.





**Figura 16.** Estructuras valvulares del corazón (Editado de Gartner & Hiatt, 2006).

La circulación de la sangre a través de las cámaras cardiacas sucede por los movimientos rítmicos y automáticos de contracción y relajación del músculo cardiaco (Figura 17), llamados:



**Figura 17.** Ciclo cardíaco (Editado de DMCA, 2014).

1. La sangre entra en las aurículas (**Diástole auricular**) dilatadas que se llenan completamente.
2. Cuando las aurículas se llenan completamente, se contraen (**Sístole auricular**) y la sangre pasa a los ventrículos (**Diástole ventricular**) gracias a la apertura de las válvulas auroventriculares.
3. Una vez que los ventrículos se han llenado de sangre, se contraen (**Sístole ventricular**), las válvulas permanecen cerradas.
4. La sangre sale por las arterias forzando la apertura de las válvulas semilunares, al mismo tiempo, las aurículas se dilatan (**Diástole auricular**), y vuelve a iniciar el ciclo.

- **Sístole:** El miocardio se contrae, de tal manera que los ventrículos realizan la misma acción, reduciendo su tamaño y volumen, por lo que la sangre contenida en ellos es expulsada hacia sus respectivas arterias.
- **Diástole:** El miocardio se relaja, lo que provoca un aumento de volumen acompañado de una caída de presión, por consiguiente, la sangre que hay en las aurículas tiende a pasar libremente hacia los ventrículos para igualar la presión entre ambas cámaras, lo que llena otra vez los ventrículos y reinicia el ciclo de nuevo (Dini et al., 2013).

## 2.2 Procedimientos de Mínima Invasión

---

Los procedimientos o cirugías de mínima invasión son métodos para realizar operaciones mayores a través de incisiones pequeñas, a menudo utilizando sistemas de imagen en miniatura, de alta tecnología, para reducir el traumatismo de la exposición quirúrgica (Spight et al., 2014). Los primeros procedimientos de mínima invasión fueron descritos desde el siglo X, cuando el médico árabe Abulcasis utilizó la refracción de la luz para visualizar el cuello uterino de una paciente (Filipi et al., 1991; Spaner & Warnock, 1997); sin embargo, no fue hasta los siglos XIX y XX cuando este tipo de cirugías se desarrollaron formalmente y desde entonces han presentado una evolución vertiginosa (Guerrero et al., 2014), convirtiéndose en los procedimientos de elección para una gran diversidad de procedimientos quirúrgicos de diversas disciplinas.

En México, en 1990 en la capital el Dr. Leopoldo Gutiérrez realizó el primer procedimiento de este tipo, el cuál consistió en una colecistectomía por laparoscopia en el Hospital General de Zona No. 8 del Instituto Mexicano del Seguro Social. Así mismo, otros precursores de la cirugía de invasión mínima fueron los doctores Jorge Cueto García y Alejandro Weber Sánchez, miembros fundadores de la Asociación Mexicana de Cirugía Endoscópica junto con el doctor Alberto Chousleb Kalach, uno de los pioneros de la laparoscopia en el ISSSTE (AMCE, 2021).

En la actualidad, diferentes estudios han comunicado de manera consistente que los procedimientos de mínima invasión son seguros, presentan resultados iguales o superiores a la cirugía convencional y proporcionan ventajas cosméticas evidentes (Schmitto et al., 2012), puesto que hay: menor sangrado y necesidad de transfusión; menor dolor posquirúrgico y menor necesidad, por tanto, de analgésicos; menor incidencia de infección de herida quirúrgica; recuperación más rápida y reincorporación más precoz a la actividad habitual, y menor coste hospitalario (Bautista, 2014).

Así pues, dotar al médico con herramientas que le permitan interactuar con componentes reales y simulados a través de diversos elementos, como lo son: estudios preoperatorios, modelos anatómicos tridimensionales y contenido multimedia de asistencia antes, durante y después de una cirugía, resulta de gran relevancia en la medicina. Por lo que, la descripción de las características anatómicas principales del corazón, abordadas en lo anterior, son un marco de referencia de las estructuras anatómicas de interés implementadas para generar cada una de las vistas en el sistema de visualización web propuesto. Mientras que los aspectos generales

relacionados a los procesos de mínima invasión especificados fundamentan la importancia del presente desarrollo no solo como una herramienta de aprendizaje en la medicina, sino como una aplicación que permita el diagnóstico a partir de imágenes médicas para la prevención y planeación quirúrgica de manera accesible y en un menor tiempo.

---

## Capítulo 3

### Metodología

---

Para lograr un sistema de visualización web de imágenes médicas eficiente, fluido y accesible, se necesitó considerar la representación 2D y 3D de estructuras anatómicas de interés a través de vistas que muestren los planos de corte (transversal, sagital y coronal), ya que estas son las posiciones anatómicas que se toman en cuenta para estudiar el cuerpo humano, y modelos tridimensionales renderizados en tiempo real, los cuales son generados al momento o a partir de reconstrucciones previas, tal que pueden ser manipulados por medio de diversos controles que permiten entre otras cosas: trasladar, escalar, rotar y segmentar. De este modo, se emplearon diversos paquetes de software de distribución libre, los cuales cuentan con robustos algoritmos que permiten aplicar técnicas para adquirir los recursos necesarios; la implementación adecuada de API's y bibliotecas con el fin de generar cada una de las visualizaciones propuestas, así como el análisis de los formatos y resoluciones idóneas para el renderizado 2D y 3D a través de un navegador web.

Además, se utilizó como base un conjunto de etapas parcialmente ordenadas con el fin de obtener un producto de software de calidad, tal que las necesidades identificadas en esta tesis son traducidas a requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código probado, documentado y certificado para su uso operativo (Jacobson, 1998), a través de la implementación de una metodología de software en cascada con ciclo iterativo, con el objetivo de poder realizar una evolución sostenida de la solución desde el primer momento. Tal que se consideró el siguiente proceso de ingeniería de software (Tabla 4):

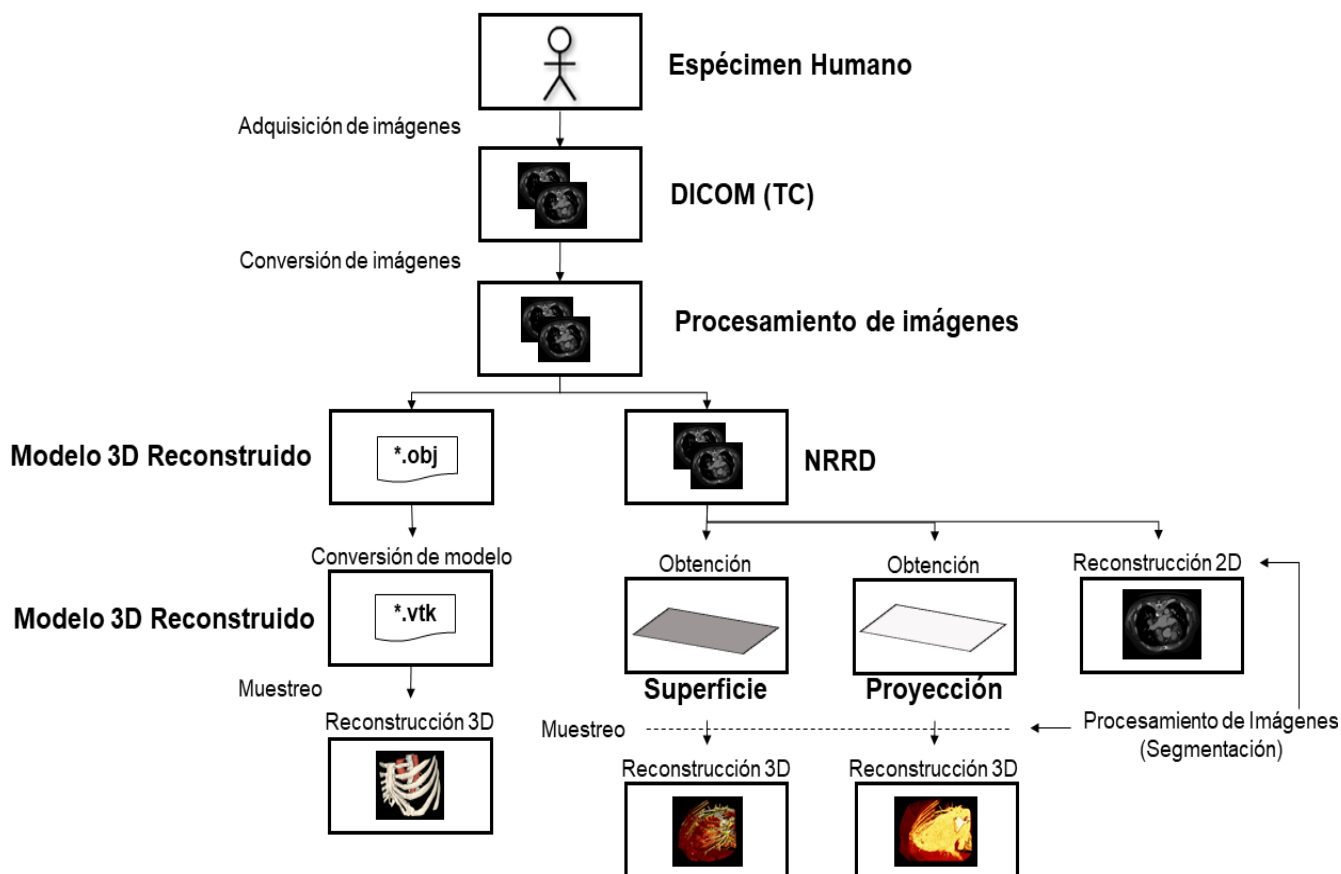
- Planificación, análisis y especificación de las necesidades identificadas e implementadas en una metodología de diseño de software, así como definición de la arquitectura del sistema de visualización.
- Segmentación y reconstrucción tridimensional de la anatomía de 2 pacientes con caso de falla de válvula cardiaca, siguiendo los protocolos médicos vigentes al momento del estudio, mediante equipo especializado del CMNSXXI y software de análisis de imágenes, en donde se pondrá énfasis en 2 estructuras principales: corazón y tórax.

- Diseño e implementación de un ambiente de realidad mixta para el aprendizaje anatómico, así como el diagnóstico y planificación quirúrgica.
- Diseño, planificación e implementación de pruebas de integración en la exploración de soluciones de interacción humano – computadora para determinar cuál tecnología ofrece una mejor inmersión al usuario a través de un navegador web.
- Diseño, planificación e implementación de pruebas de integración en una serie de evaluaciones con un especialista y docente en el área con el fin de determinar la mejor solución y particularidades del sistema para la enseñanza y aprendizaje anatómico, así como la toma de decisiones para el diagnóstico y planeación quirúrgica usando realidad extendida en un determinado escenario.
- Diseño e implementación de pruebas de uso del modelo de computación híbrido con alumnos de la carrera médico cirujano de la Facultad de Medicina de la UNAM.
- Análisis de los resultados de la experiencia de los usuarios obtenidos en las pruebas.
- Finalización del desarrollo de la tesis para la obtención de grado, así como la documentación de los esfuerzos en cuanto al desarrollo de sistemas de software interactivos con realidad mixta aplicado en las ciencias de la salud en México.

**Tabla 4.** Diagrama de Gantt que define el calendario de actividades que contempla la metodología de software para la creación gráfica del sistema de visualización web de imágenes médicas.

Actividad	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6
Diseño de la arquitectura del sistema.	■					
Selección y segmentación de imágenes TAC de pacientes.	■	■	■	■		
Implementación de ambiente de realidad mixta.		■	■	■	■	
Pruebas de integración con dispositivos de interacción.				■	■	
Pruebas de integración con un especialista y docente.			■	■	■	
Pruebas de uso con alumnos de medicina de la UNAM			■	■	■	
Análisis de resultados de experiencia de usuario.					■	■
Escritura de la tesis.				■	■	■

Es importante mencionar que la Figura 18 aborda y describe de manera generalizada una metodología de creación gráfica del sistema de visualización web de imágenes médicas, como una forma de abstraer las etapas clave en la que se centra el funcionamiento principal del software.



**Figura 18.** Esquema generalizado de la metodología empleada en el desarrollo del sistema de visualización web de imágenes médicas.

### 3.1 Procesamiento de Imágenes Médicas

A fin de representar gráficamente una imagen médica a través de un navegador web, se consideró un procesamiento de la información en dos fases: la primera, previa a la implementación, en la que se realizó el análisis del formato, tipo de dato y resoluciones adecuadas para los recursos que se necesitan emplear para ello, de tal manera que se llevó a cabo el procesamiento de un conjunto de imágenes extraídas por TC en formato DICOM a través la herramienta *Fiji* con el propósito de obtener recursos en formato NRRD considerando las características en particular que se requerían; la segunda, durante la implementación, al considerar disponer de un control interactivo para el usuario, el cual le permita manipular las imágenes o modelo, según el tipo de visualización,

al dividirla en partes constituyentes hasta un nivel de subdivisión en el que se puedan aislar regiones de interés a través de los procesos de segmentación simples de *thresholding* y *window*.

La elección de estas contemplaciones se retomará, explicará y especificará en el capítulo siguiente, por lo que a continuación se detallará la herramienta de procesamiento previo, los formatos manejados, así como los métodos de segmentación puestos en funcionamiento en el sistema de visualización web del presente trabajo.

### **3.1.1 DICOM**

DICOM es el acrónimo en inglés de Imágenes Digitales y Comunicaciones en Medicina, el cuál define el estándar internacional de comunicación entre sistemas de información y un formato de almacenamiento de imágenes médicas. El estándar fue desarrollado por el *American College of Radiology* (ACR) y el *National Electrical Manufacturers Association* (NEMA) y publicado por primera vez en 1993 (OTech, 2021). En otras palabras, DICOM es un protocolo de comunicaciones estándar empleado para capturar, almacenar y transmitir imágenes médicas y datos relacionados (DICOM, 2021), al actuar como un modelo para las estructuras de información y procedimientos que controlan la entrada y salida de datos en sistemas de imágenes médicas.

### **3.1.2 NRRD**

NRRD por sus siglas *Nearly Raw Raster Data* es una biblioteca y un formato de archivo diseñado para la visualización científica y el procesamiento de imágenes que involucran datos ráster n-dimensionales, es decir, matrices de celdas o píxeles, organizadas en filas y columnas, en la que cada celda contiene un valor que representa información (NRRD, 2021). El formato de archivo NRRD es conveniente, simple, flexible y portátil, al estar conformado por un encabezado y datos (sin procesar o procesados) en formato ASCII, gzip, bzip o raw, los cuales pueden estar en el mismo archivo o por separado. La estructura general del formato de archivo NRRD se especifica en la Tabla 5.

Cabe destacar que este formato es compatible con claves o valores personalizados en el encabezado que no están definidos en la especificación NRRD. Además, facilita las tareas de procesamiento que a menudo preceden a la visualización o segmentación, así como el análisis básico para comprender la estructura de un conjunto de datos, ya que este puede ser leído desde

un simple editor de texto, en comparación de otros formatos (PNG, JPG, DICOM, entre otros más) con los cuales no es posible.

**Tabla 5.** Estructura general del formato de archivo NRRD

Descripción	Formato
<b>Línea mágica</b>	NRRD000 <b>X</b>
<ul style="list-style-type: none"> <li>NRRD: Identificación como archivo NRRD</li> <li>000 X: Versión del formato de archivo</li> </ul>	<code>&lt; campo &gt;: &lt; desc &gt;</code> <code>&lt; campo &gt;: &lt; desc &gt;</code>
<b>Especificaciones de campo</b>	<code># &lt; comentario &gt;</code>
<ul style="list-style-type: none"> <li>Necesarios: Tamaño, dimensión, tipo de dato, codificación</li> <li>Otros: Contenido, min, max, salto de línea, unidades de muestra, entre otros más.</li> </ul>	<code>...</code> <code>&lt; campo &gt;: &lt; desc &gt;</code> <code>&lt; clave &gt;: = &lt; valor &gt;</code> <code>&lt; clave &gt;: = &lt; valor &gt;</code> <code>&lt; clave &gt;: = &lt; valor &gt;</code>
<b>Pares clave / valor</b>	<code># &lt; comentario &gt;</code>
<ul style="list-style-type: none"> <li>Especificaciones personalizadas</li> </ul>	
<b>Espacio que separa el encabezado de los datos</b>	<code>&lt; datos &gt; &lt; datos &gt; &lt; datos &gt; &lt; datos &gt; ...</code>
<b>Datos</b>	

### 3.1.1 FIJI

Fiji por su acrónimo en inglés *Fiji is just a ImageJ*, es un paquete de procesamiento y análisis de imágenes de código abierto bajo la licencia GNU *General Public License* y basado en el núcleo de ImageJ2, que tiene la licencia permisiva BSD de 2 cláusulas, el cual fue desarrollado en el 2012 por Johannes Schindelin, Ignacio Arganda, entre otros más, y es mantenido por el grupo Eliceiri/LOCI en la Universidad de Wisconsin-Madison, y los laboratorios de Jug y Tomancak en el MPI-CBG en Dresde (Schindelin et al., 2012).

El objetivo principal de Fiji es proporcionar diversos complementos y componentes actualizables automáticamente, los cuales poseen sus propias licencias, en una misma herramienta, así como ofrecer una extensa documentación, al emplear prácticas modernas de ingeniería de software para combinar potentes bibliotecas con una amplia gama de lenguajes de secuencias de comando, de tal forma que se permita la creación rápida de prototipos de algoritmos de procesamiento de imágenes, por lo que busca mejorar las herramientas disponibles para que las ciencias de la vida procesen y analicen datos (ImageJ, 2021).



### 3.1.3 Thresholding

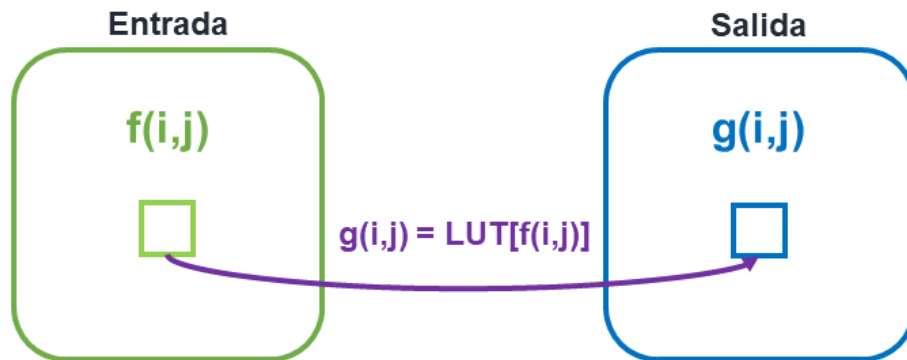
En el procesamiento de imágenes el *thresholding* es un método simple de segmentación, al considerar la propiedad de intensidad que comparten los píxeles de una región, el cuál crea y emplea umbrales para componer imágenes binarias, separarlas en regiones claras (valores de fondo) y oscuras (valores de primer plano), a partir de una en escala de grises. De esta manera, se establece que si la intensidad de un píxel  $I(x, y)$  de una imagen  $P(x, y)$  está por debajo del valor del umbral  $T$  establecido se convierte a negro (valor de bit cero), mientras que si está por encima se convierte a blanco (valor de bit uno) (Morse, 2000), lo anterior se representa de la siguiente manera:

$$P(x, y) = \begin{cases} \mathbf{1} & I(x, y) \geq T \\ \mathbf{0} & I(x, y) < T \end{cases} \quad [E0]$$

El umbral de intensidad de la imagen, es decir, la luminosidad relativa de la imagen interpretada comúnmente como brillo, se establece manualmente en un valor específico o automáticamente mediante una aplicación; Sezgin y Sankur (2004), clasifican los métodos de creación de umbrales en función de la información que manipulan: histograma (análisis de picos, valles y curvaturas), agrupamiento (modelación gaussiana), entropía (regiones), atributos de un objeto (formas o bordes), espaciales (distribución de probabilidad), o locales (adaptación umbral).

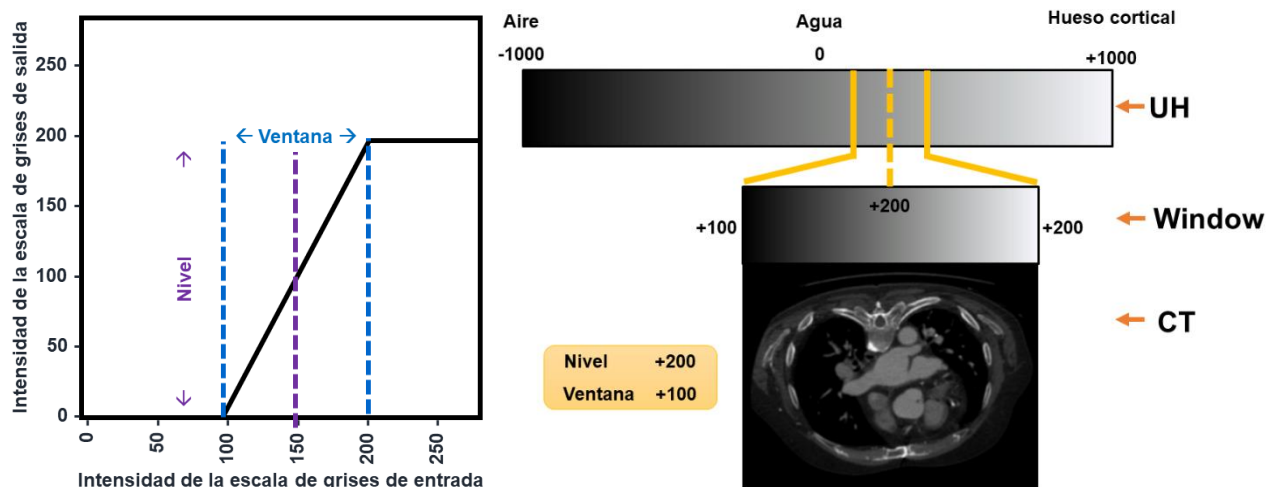
### 3.1.4 Window

*Window*, al igual que *thresholding*, es un método simple de segmentación, también conocido como nivel, densidad, corte de nivel de gris o LUT (mnemónico para *lookup table*); el cual aplica una función de transformación de escala de grises lineal en forma de una tabla de búsqueda (LUT) especificada por dos parámetros, ventana y nivel, lo que da como resultado resaltar las intensidades de píxeles correspondientes a un subconjunto de todo un rango dinámico, tal que los píxeles pueden caer fuera del rango especificado (Figura 19).



**Figura 19.** Representación gráfica del procesamiento LUT: cada píxel en la imagen de entrada  $f$  se le asigna un píxel correspondiente de la imagen de salida  $g$  con la nueva intensidad de píxel tomando un valor especificado por la LUT.

Una forma de establecer una función de transformación lineal en escala de grises (Figura 20), es definir el ancho de LUT donde la pendiente es distinta de cero (ventana) y el centro de ese mismo segmento de la LUT (nivel), por lo que mantener la ventana constante mientras se ajusta el nivel tiene el efecto de mover la parte de pendiente distinta de cero de la transformada hacia la izquierda o derecha. Del mismo modo, alterar el parámetro de la ventana amplia o reduce el rango de intensidad de píxeles que se resaltará. Es importante destacar que implementar este método se interpreta como una variación de contraste de una imagen, en otras palabras, un ajuste entre los colores más claros y oscuros de la imagen (Mugmiroh et al, 2018).



**Figura 20.** Transformación de imágenes mediante una tabla de búsqueda en escala de grises.

## 3.2 Visualización Web de Imágenes Médicas

---

Para poder realizar la renderización de gráficos 2D y 3D interactivos en un navegador web, se hizo uso de WebGL en el marco de Three.js, dado que este encapsula la interfaz de bajo nivel de WebGL en una de alto nivel, lo cual permite abstraer detalles nativos y simplificar la creación de escenas 3D comunes en la programación de gráficos al descomponerlas en mallas, materiales, fuentes de luz, entre otros, como objetos *JavaScript*, así como incorporar elementos útiles para dotar los desarrollos con efectos visuales y funcionalidades más enriquecedoras de manera simple e intuitiva. De esta manera, una vez generados los recursos al procesar las imágenes, se procedió a desarrollar las visualizaciones propuestas para el sistema del presente escrito, a través de estas herramientas y elementos de HTML5, JS y CSS, las cuales se describen a continuación de manera más detallada.

### 3.2.1 HTML, JS y CSS

HTML por sus siglas en inglés *HyperText Markup Language*, es un lenguaje de marcado que permite definir y codificar contenido de forma textual y estructurada en un navegador, al incorporar etiquetas o marcas que contienen información adicional acerca de la estructura o presentación de este. Fue propuesto en 1991 por el físico e investigador del CERN (Organización Europea para la Investigación Nuclear) Tim Beners Lee como un nuevo sistema de hipertexto para compartir documentos y basado en el lenguaje de marcado SGML (*Standard Generalized Markup Language*), sin embargo, la primera propuesta oficial para convertir HTML en un estándar se realizó en 1993 por parte del organismo IETF (*Internet Engineering Task Force*). HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado *World Wide Web Consortium (W3C)* (UM, 2016).

CSS por sus siglas en inglés *Cascading Style Sheets* es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Las hojas de estilo aparecieron poco después que el lenguaje de marcado SGML, alrededor del año 1970, ya que se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a documentos electrónicos, por lo que se presentaron dos propuestas ante el organismo W3C: la CHSS (*Cascading HTML Style Sheets*) por Hakon Wium Lie y la SSP (*Stream-based Style Sheet Proposal*) por Bert Bos. Por consiguiente,

entre 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta, al cual nombraron CSS (Eguiluz, 2006).

*JavaScript* (JS) es un lenguaje de programación ligero e interpretado de scripts, el cual ejecuta secuencia de comandos, basado en prototipos que permiten emular la programación orientada a objetos; desarrollado por Brendan Eich en 1995 y normalizado por *ECMA International* como el nombre *ECMAScript Language Reference*. Este lenguaje es empleado principalmente del lado del cliente como parte de un navegador web, lo que permite mejorar la interfaz para el usuario y crear contenidos dinámicos (Menéndez, 2021).

### 3.2.2 WebGL

*Web-based Graphics Library*, mejor conocido como WebGL, es un estándar Web multiplataforma libre de regalías para una API de gráficos 3D de bajo nivel basado en *OpenGL ES 2.0* y expuesto a través del elemento canvas de HTML5 como interfaces DOM<sup>1</sup>. Por lo tanto, al ser una API, ofrece una librería especializada en renderizado 3D, que hace posible la creación de gráficos 3D que se actualizan en tiempo real directamente en el navegador, sin incluir complementos a la Web, y haciendo uso del hardware para desempeñar algunas funciones mucho más rápido de lo que es posible en un software corriendo en la CPU de propósito general, al emplear la GPU de la tarjeta gráfica para procesar grandes cargas de gráficos (Khronos, 2021).

Cabe resaltar que *OpenGL ES 2.0* (*OpenGL for Embedded Systems*) es una variante simplificada de *OpenGL 2.0*, el cual es manejado, al igual que *OpenGL*<sup>2</sup>, por el grupo tecnológico *Khronos Group*<sup>3</sup>. El primer prototipo de WebGL nació como un experimento que comenzó en Mozilla por Vladimir Vukicevic en 2006. A principios del 2009 fue estandarizado por *Khronos Group* y comenzó el *WebGL Working Group* con la participación inicial de Apple, Google, Mozilla y Opera, con el fin de verificar que el contenido WebGL pueda correr tanto en equipos de escritorio como dispositivos móviles (Crespo, 2012).

---

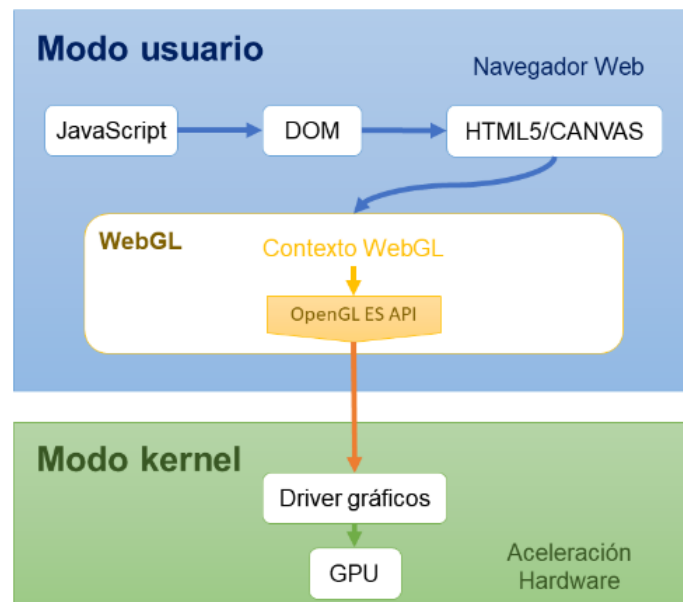
<sup>1</sup> *Document Object Model* (Modelado de Objetos del Documento), es una API multiplataforma para representar e interactuar con objetos en los documentos HTML, XHTML y XML.

<sup>2</sup> Especificación estándar que define una API multiplataforma para escribir aplicaciones que contengan gráficos 2D y 3D, para dispositivos integrados.

<sup>3</sup> Consorcio industrial sin fines de lucro encargado de crear estándares abiertos para permitir la creación y aceleración de computación paralela, gráficos y medios dinámicos en una variedad de plataformas y dispositivos.

## i. Arquitectura

La arquitectura de WebGL, se especifica a través de *JavaScript* tal que se obtiene el DOM del elemento canvas de HTML5, posteriormente se define el contexto WebGL, por medio del cual se accede a la API de WebGL, finalmente este último se encarga de comunicarse con el controlador de la tarjeta gráfica, de tal forma que se pueda realizar la aceleración por hardware en la GPU, como se muestra en la Figura 21.



**Figura 21.** Arquitectura de WebGL (Editado de Crespo, 2012).

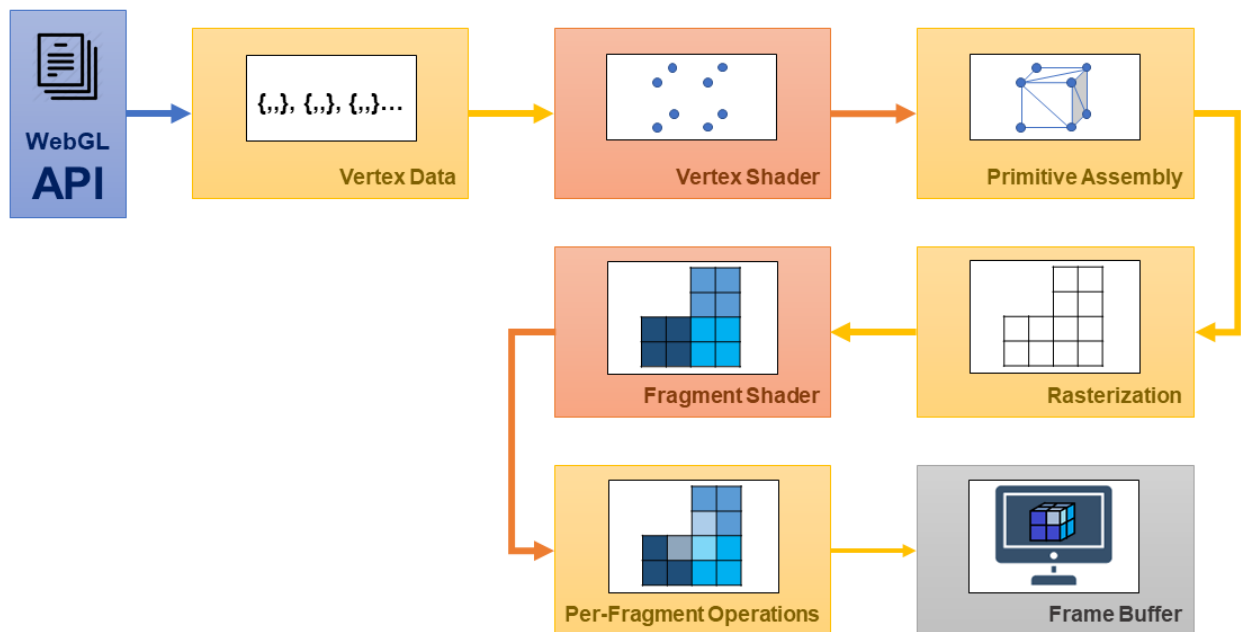
## ii. Pipeline de renderizado

El pipeline de WebGL es el mismo que se define en *OpenGL ES 2.0*, el cual implementa un pipeline gráfico basado en dos especificaciones: *OpenGL ES 2.0 API specification*, descripción del sistema de gráficos donde se indica qué es, cómo actúa y lo que se requiere para implementarlo; y *OpenGL ES SL specification*, descripción del *Shading Language*, también conocido como GLSL, donde se define el lenguaje de sombreado estandarizado de alto nivel con sintaxis basada en el lenguaje de programación C para usarse con *OpenGL*, pues unifica el procesamiento de vértices y fragmentos en un solo conjunto de instrucciones, a partir de la programación de *shaders*<sup>4</sup>, para modificar su comportamiento, lo que permite tener un control sobre pipeline de renderizado. De esta manera,

<sup>4</sup> Conjunto de instrucciones (programa) que se ejecutan en la GPU.

en la Figura 22 se observa el pipeline de renderizado de WebGL, tal que cada etapa describe lo siguiente:

- *Vertex Data*, se reciben los vértices: *Vertex Buffer Object* (VBO) y *Vertex Array Object* (VAO).
- *Vertex Shader*, se procesan un conjunto de operaciones (programa GLSL) que transforman cada uno de los vértices (traslación, rotación, escalado, perspectiva, entre otras) y se calculan algunos de sus atributos.
- *Primitive Assembly*, se ensamblan las primitivas, las cuales son objetos geométricos como triángulos, líneas o puntos con sus respectivas posiciones, y se implementa el *face culling*, es decir, se guarda el estado de si la forma geométrica está dentro o no del campo de visión en la pantalla, de no estarlo se descarta y no se trata en las etapas posteriores.
- *Rasterization*, se lleva a cabo la conversión de primitivas a un conjunto 2D de fragmentos, a partir de la interpolación de datos para todos los píxeles cubiertos por la primitiva, es decir, se lleva a cabo la conversión de formas geométricas a unidades de píxel.
- *Fragment Shader*, se procesan un conjunto de operaciones (programa GLSL) que se aplicarán a cada uno de los fragmentos generados, en donde se determina el color de cada uno de ellos.
- *Per-Fragment Operations*, a cada fragmento (píxel) se le aplica un conjunto de operaciones como:
  - *Pixel Ownership Test*, prueba que permite que el sistema de ventanas descarte o no el píxel.
  - *Scissor Test*, prueba que permite definir áreas donde no hay que pintar, por lo que se puede determinar si el fragmento está dentro o fuera para descartarlo.
  - *Stencil y Depth Test*, pruebas que permiten decidir a partir del valor del buffer de profundidad y el buffer de platilla si se rechaza o no el fragmento.
  - *Blending*, técnica en donde se aplican operaciones para crear transparencias con los fragmentos que componen un objeto.
  - *Dithering*, técnica que permite mejorar la resolución de los colores.
- *Frame Buffer*, se genera un array bidimensional de píxeles, en donde se almacenan los colores de los fragmentos calculados que representa lo que se va a ver en pantalla.



**Figura 22.** Pipeline de renderizado de WebGL.

### iii. Funcionamiento interno

La implementación de WebGL varía de un navegador a otro, ya que no todos adoptan la misma estructura como sistema de renderizado, al no emplear una plataforma idéntica para aplicaciones como base para el navegador. Sin duda alguna, conocer el funcionamiento interno de un navegador respecto a WebGL, es necesario para poder adoptar medidas que permitan adaptar una determinada aplicación al sistema local, en momentos de alta carga gráfica y procesado, de tal forma que se tenga un óptimo rendimiento. A continuación, se describe como el navegador Google Chrome, uno de los más empleados a nivel mundial, trabaja internamente para ofrecer WebGL, el cual está basado en el desarrollo de Chromium<sup>5</sup> y el uso de Webkit<sup>6</sup>.

- *Proceso de renderizado en Chromium (Chromium, 2021)*

En Webkit están definidos diversos procesos encargados de controlar diferentes aspectos del navegador, unos de los más relevantes son los controladores de HTML, CSS y JS. En el momento, en el que alguno de los controladores quiera renderizar algo en el navegador, iniciará un *Render*

<sup>5</sup> Proyecto de navegador web de código abierto y de participación comunitaria bajo el ámbito de Google Code, a partir del cual se basa el código fuente de Google Chrome.

<sup>6</sup> Plataforma para aplicaciones como base para el navegador, en donde uno de los componentes más famoso es el WebCore para diseñar y renderizar. así como la librería DOM para HTML y SVG.

*Process*, que actuará como cliente, de tal forma que se encargará de serializar las instrucciones y depositarlas en un *buffer* de comandos (*RCP Buffer*<sup>7</sup>), en una memoria compartida (*Shared Memory*), para realizar las peticiones a la unidad de procesamiento gráfico (GPU) de manera asíncrona. Se puede señalar que el *buffer* tiene un tamaño fijo, en caso de que este sea sobrepasado se ejecutara un *Sync Flush*, proceso donde el *Render Process* se vaciará automáticamente para liberar recursos y atender las siguientes peticiones, lo que implica que dicho proceso no fluya a medida que se ejecuta al llevar al máximo el navegador.

El navegador se encargará de recoger las peticiones del *buffer* para transferirlas a un proceso propio del navegador llamado *GPU Process*. El *GPU Process* recoge los datos serializados y los parsea, para ejecutarlos apropiadamente en la GPU, a través de la comunicación que se da mediante los drivers del propio hardware. Cabe destacar que en esta comunicación existen dos piezas fundamentales de traducción, que a su vez permiten que el contexto gráfico acceda directamente a las librerías nativas del sistema operativo GL: *ANGLE* por sus siglas en inglés *Almost Native Graphics Layer Engine*, la cual es una capa de proceso que traduce comandos *OpenGL ES 2.0* a *DirectX 9*, en caso de que sea usado en *Windows*; y *SwiftShader* para máquinas que no tengan drivers de hardware adecuados, por lo que se realizará una rasterización vía software.

El proceso se ejecuta exclusivamente en un *SandBox*, es decir, en un proceso aislado, mediante el cual se pueden llevar a cabo distintos programas con seguridad y de manera separada, empleado para ejecutar código nuevo o software de dudosa confiabilidad, con el objetivo de evitar la corrupción de datos del sistema en donde estos se ejecutan; de tal forma que el proceso de renderizado no tenga acceso directo a las llamadas 3D suministradas por el sistema operativo. Casi todas las llamadas de *OpenGL* no tienen valores de retorno al cliente y el servidor puede trabajar asíncronamente para mantener un buen rendimiento y no tener que validar ningún retorno. En algunos modos de *debug* o llamadas específicas es necesario controlar retornos para obtener ciertos valores del servidor, a través del mecanismo *IPC (Inter-Process Communication)*, el cual es un conjunto de métodos para intercambiar datos entre hilos o procesos de ejecución.

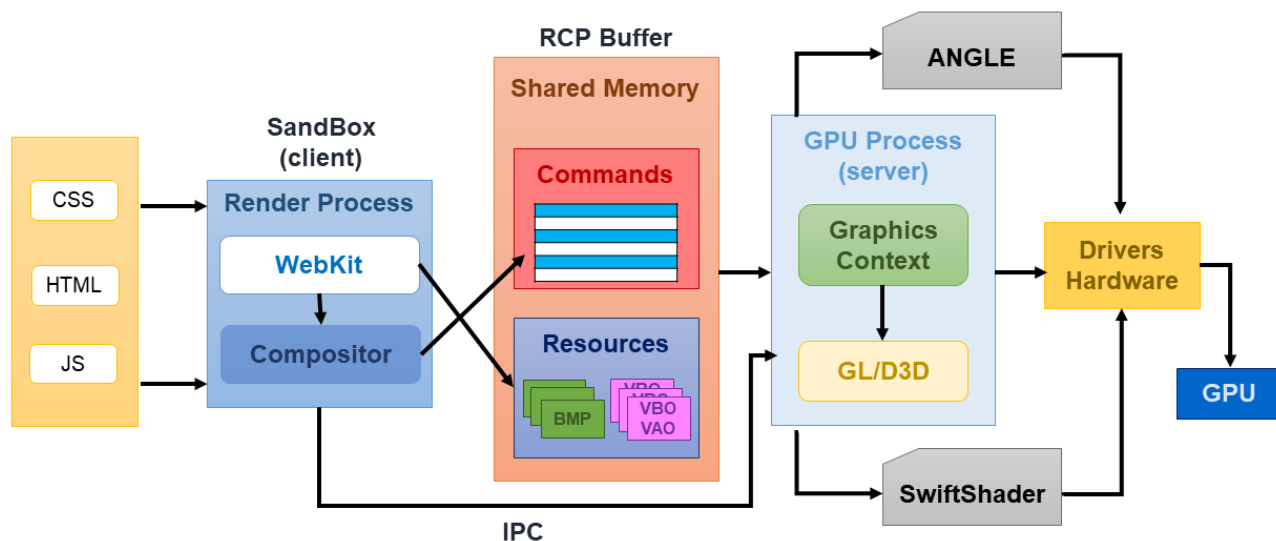
La ejecución del *GPU Process* de manera aislada permite: seguridad, al trabajar de manera aislada la lógica de renderizado al no compartir recursos básicos; robusteza, dado que el funcionamiento o interrupción de este proceso no interfiere en el resto de los procesos y no para

---

<sup>7</sup> Remote Procedure Call Buffer.



la ejecución del navegador; y uniformidad, ya que *OpenGL ES 2.0* es una API de renderizado estandarizada que permite mantener un único código para las diferentes distribuciones de Chromium. De esta manera, es importante mencionar que el proceso descrito anteriormente y mostrado en la Figura 23 es compartido entre todas las ejecuciones del navegador que quieran renderizar algo, por lo que el comportamiento de WebGL y WebKit es similar.



**Figura 23.** Proceso de renderizado en un navegador web (Editado de Chromium, 2021).

### 3.2.3 Three.js

Three.js es una biblioteca ligera y de alto nivel escrita en *JavaScript* para crear y mostrar gráficos animados en 3D a través de un navegador web, creada y liberada en GitHub por el español Ricardo Cabello en abril de 2010, conocido por su seudónimo de *Mr. Doob*. Esta biblioteca fue creada sobre WebGL, al encapsular la interfaz de bajo nivel en objetos *JavaScript* para simplificar de manera intuitiva la creación de escenas 3D comunes para la programación de gráficos, lo que asegura que el código es compatible en varios navegadores, considerando estos elementos y otros que no son soportados por WebGL. Las principales contribuciones de Three.js involucran: renderizadores, efectos, escenas, mallas, cámaras, animación, luces, materiales, *shaders*, objetos, geometría, cargadores de datos, utilidades, exportación e importación. En la Figura 24 se describe la estructura general que presenta una aplicación Three.js (Threejs Fundamentals, 2021).

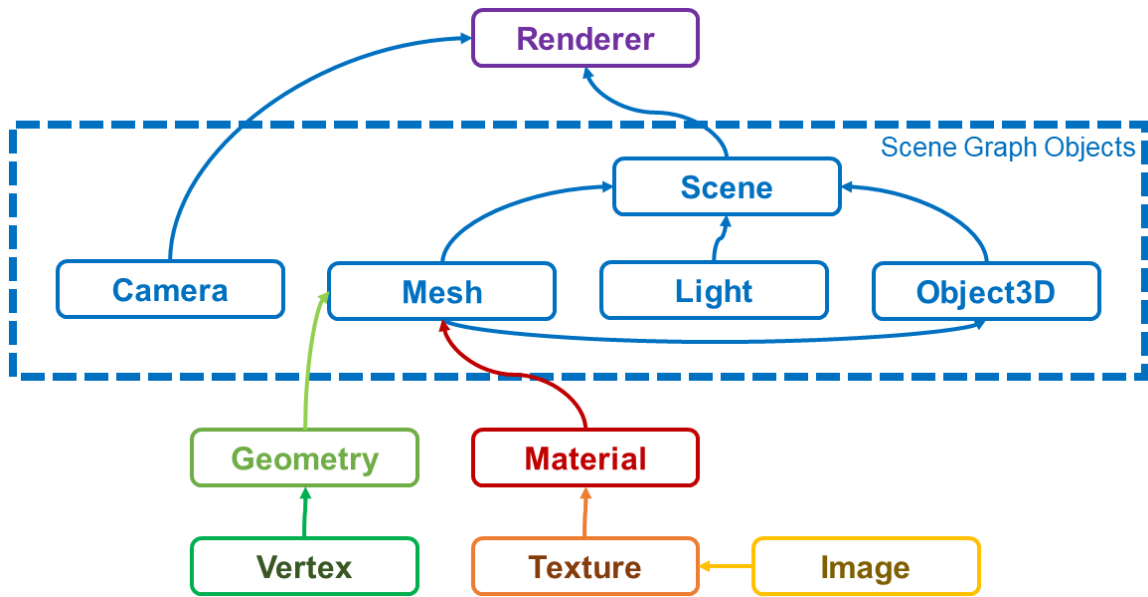


Figura 24. Estructura general de una aplicación en Three.js (Editado de Three.js Fundamentals, 2021).

### 3.3 Visualización Web de Imágenes Médicas en Representación de Volumen

Con la intención de representar el renderizado volumétrico de modelos generados en tiempo real de una estructura anatómica de interés, se hizo uso de Three.js, al emplear el cargador de archivos NRRD, *shaders*, gráficos 3D Web, controles de órbita y la interfaz gráfica de usuario, de tal manera que a partir del recurso generado del procesamiento y la implementación de estos elementos se genere la visualización deseada. Es importante señalar que, para lograr dinamismo en esta representación, fue necesario diseñar y crear imágenes como mapas de colores por medio de GIMP, para generar texturas que permitan dotar al volumen reconstruido de un color particular acentuando de esta forma detalles de la anatomía, así como variar el estilo de renderizado al obtener una superficie (isosuperficie, ISO) o una proyección (proyección de máxima intensidad, MIP), con base al mapa de colores y las Unidades Hounsfield (UH) de la imagen médica en NRRD, visualizadas por medio de un algoritmo de *ray casting*, con el fin de generar un gráfico en 3D. En lo siguiente, se presenta la descripción de la herramienta de edición de imágenes, así como algunos conceptos de interés que sustentan la implementación anterior.

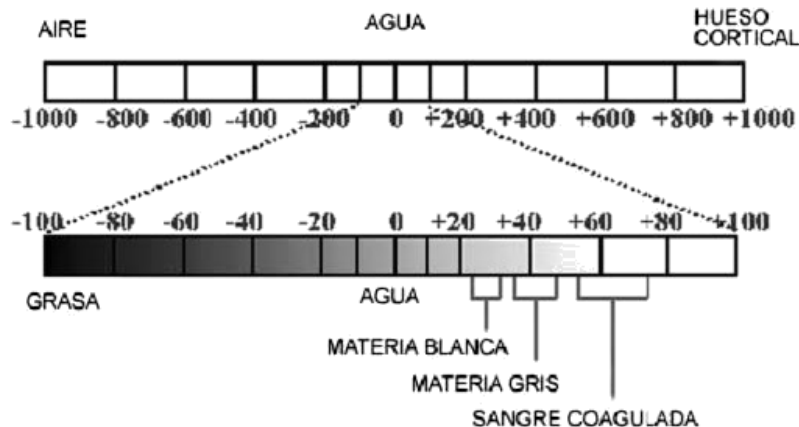
### 3.3.1 GIMP

GIMP (*GNU Image Manipulation Program*) es un software libre y multiplataforma englobado dentro del proyecto GNU y disponible bajo esta licencia pública y la de *GNU Lesser General Public License*, el cual sirve para procesar gráficos y fotografías digitales en formato JPG, GIF, PNG, PCX, TIFF, BMP, PIX, PSD, XCF, y PDF, con el fin de crear, recortar, modificar, combinar y alterar en forma de mapa de bits. Así mismo, lee y escribe la mayoría de los formatos de ficheros gráfico. Este programa fue desarrollado en 1995 por Spencer Kimball y Peter Mattis de la Universidad de Barkeley (GIMP, 2021).

### 3.3.2 Unidades y Escala de Hounsfield

El proceso de TC comienza con la emisión de rayos X, los cuales son atenuados por el paciente en función del grosor del corte y de las estructuras que estén en el camino del rayo a través él, de tal forma que el área explorada se convierte en un volumen conformado por una matriz de volúmenes más pequeños denominados vóxeles. Los detectores absorben los fotones que salen del paciente y son transformados en un signo electrónico y ampliados, para ser posteriormente convertidos en un número según su intensidad (Aguinaga et al., 2006). La imagen capturada por el tomógrafo representa una matriz conformada por cuadros dispuestos en filas y columnas, en donde cada cuadro es un píxel, al cual se le asigna un valor numérico que corresponde a un valor promedio de la atenuación correspondiente del vóxel en la escala de unidades arbitrarias llamadas Unidades Hounsfield (UH).

La escala UH fue nombrada en honor a Sir Godfrey Newbold Hounsfield, en ella se propone un rango de valores que varía de +1000 a -1000 que conforman una escala de grises, en donde al valor superior se le asigna el color blanco (hueso cortical), al valor central el color gris (agua), y al valor inferior el color negro (aire), de tal forma que a los otros tejidos se le asigna un número de acuerdo a su densidad relativa (López et al., 2009), como se muestra en la Figura 25. Es importante señalar que esta escala es abierta, tal que el rango de valores puede representar 4096 niveles de gris mediante un número de 12 bits ( $2^{12} = 4096$ ).



**Figura 25.** Escala de Hounsfield (Aguinaga et al., 2006).

### 3.3.3 Volume Rendering

El renderizado de volumen mejor conocido por su término en inglés *volume rendering* representa una colección de métodos utilizados en gráficos por computadora y visualización científica para crear una proyección 2D a partir de un conjunto de datos 3D muestreados discretamente, en otras palabras, visualizar campos escalares tridimensionales. El conjunto de datos del espacio 3D es llamado generalmente como *dataset*, de tal manera que cada muestra tomada de este como una partición elemental es conocido como vóxel (Watt & Watt, 1992).

Existen diversas estrategias para abordar el renderizado de volúmenes, entre las que se pueden destacar las técnicas indirectas, en donde se procesa el *dataset* con el objeto de encontrar superficies significativas que puedan renderizarse empleando técnicas usuales de visualización, no obstante, no se toma en cuenta que el *dataset* puede originarse a partir de fluidos y otros materiales que pueden ser transparentes y deben modelarse como tal, por ejemplo, las imágenes médicas; y las directas, a través de segmentación difusa utilizando una función de transferencia, tal que se genera una imagen tridimensional a partir de la proyección de las muestras discretas de un volumen (Zhou & Tönnies, 2003), de tal forma que se resuelve la limitante anterior, sin embargo, implica una complejidad mayor computacionalmente, pero con resultados de mayor calidad.

Los métodos de *volume rendering* empleados usualmente, están descritos por los siguientes algoritmos: *ray casting*, calcula la proyección trazando rayos hacia el espacio de volumen; *shear-warp*, emplea un esquema similar al anterior, pero en este caso se omiten las regiones de imagen opacas y vóxeles transparentes; *slicing*, representa el volumen mediante corte

de texturas 3D como una superposición de capas semitransparentes de volumen, y *splatting*, muestra el volumen como una matriz de funciones de base superpuesta, comúnmente gaussianas con acumulación de la evaluación de la contribución ponderada de cada vóxel en varios píxeles (Carmona., 2015). Por otro lado, la teoría que sustenta al *volume rendering*, en el que se basan los algoritmos, describe el modelo físico de una simulación aproximada de la propagación de la luz a través de un medio participante representado por un volumen.

- *Propiedades del volumen*

En términos físicos, cuando por un volumen fluye la luz, esta puede ser absorbida, esparcida y emitida; adicionalmente, se pueden producir otros fenómenos, sin embargo, en el renderizado de volumen no se tiene como objetivo simular todos ellos, ya que la modelación de este comportamiento requiere del uso de la Teoría del Transporte Radioactiva, lo que lo hace un tanto complejo, sin embargo, en la visualización científica el modelo físico a considerar está basado únicamente en la emisión y absorción, como una forma de representar la interacción de la luz con el volumen, y determinar la cantidad de luz alcanzada en cada píxel de la imagen.

- *Ecuación de volume rendering (Carmona, 2015)*

Peter Williams y Nelson Max, especificaron dos modelos que resultan en el mismo modelo matemático para obtener la intensidad final en un píxel de la imagen. El primero, llamado Modelo de Partículas, define que el volumen es modelado como una nube de partículas que absorben y emiten luz; mientras que el segundo, el Modelo Continuo, establece que el volumen es obtenido como un medio continuo incandescente. Basados en estos dos trabajos, a continuación, se presenta la derivación de la ecuación de visualización volumétrica [E1], como una ecuación integral que no tiene antiderivada para funciones lineales a trozos o de orden superior, la cual puede ser evaluada numéricamente a partir de una aproximación simple apoyada en Series de Riemann:

$$I(D) = I_0 e^{-\int_0^D \tau(\lambda') d\lambda'} + \int_0^D L(\lambda) \tau(\lambda) e^{-\int_\lambda^D \tau(\lambda') d\lambda'} d\lambda \quad [E1]$$

En donde, el primer término de la ecuación calcula la cantidad de luz  $I_0$  entrante al volumen, atenuada exponencialmente según la absorción de las partículas en el trayecto del rayo; el segundo término agrega la cantidad de luz emitida de cada punto a lo largo del rayo, considerando

la extinción desde el punto de emisión, a una distancia  $\lambda$  del origen del rayo, hasta el final de este. El rayo de luz puede ser reparametrizado, desde el ojo al considerar  $\lambda = 0$  hasta el punto de entrada de la luz  $\lambda = D$  [E2]; así mismo, si se considera despreciable la intensidad de luz entrante o atenuada a la nube, la ecuación se simplifica a [E3], tal que el término de atenuación exponencial (extinción), también llamada transparencia, es considerada como la cantidad de luz que no es bloqueada por partículas entre el ojo y  $\lambda$ , por tanto, se encuentra la contribución de luz emitida  $L(\lambda)$  a cada distancia  $\lambda$ , atenuada por su absorción  $\tau(\lambda)$  y adicionalmente por la transparencia acumulada hasta  $\lambda$  a lo largo del rayo que atraviesa la nube.

$$I(D) = I_0 e^{-\int_0^D \tau(\lambda') d\lambda'} + \int_0^D L(\lambda) \tau(\lambda) e^{-\int_0^\lambda \tau(\lambda') d\lambda'} d\lambda \quad [E2]$$

$$I(D) = \int_0^D L(\lambda) \tau(\lambda) e^{-\int_0^\lambda \tau(\lambda') d\lambda'} d\lambda \quad [E3]$$

La ecuación [E3] genera una imagen monocromática, de modo que la intensidad  $L(\lambda)$  representa sólo la luminosidad. Para extender a imágenes de color real correspondientes al espacio de color RGB, se considera que en el modelo una partícula absorbe totalmente la luz que colisiona con la misma, por lo que la absorción  $\tau(\lambda)$  será igual para todas las longitudes de onda [E4], en donde  $c(\lambda)$  y  $C$  son tuplas RGB, por lo que la extinción se puede interpretar como la transparencia  $T(\lambda)$  de la nube hasta la distancia  $\lambda$ ; basada en la transparencia, se puede calcular la opacidad  $\alpha$  acumulada en la trayectoria del rayo a cualquier distancia como [E5].

$$C = \int_0^D c(\lambda) \tau(\lambda) e^{-\int_0^\lambda \tau(\lambda') d\lambda'} d\lambda \quad [E4]$$

$$\alpha(\lambda) = 1 - T(\lambda)$$

$$\alpha(\lambda) = 1 - e^{-\int_0^\lambda t(s(x(\lambda'))) d\lambda'} \quad [E5]$$

- *Evaluación de la ecuación de volume rendering*

La asignación de un color y absorción, mejor conocido como clasificación, en cada punto de intersección del rayo con el volumen se puede realizar mediante: segmentación, en este caso el volumen puede ser particionado en sus estructuras específicas que lo conforman para asignarle en distintas partes diferentes valores; o por manipulación de funciones de transferencia, las cuales pueden ser unidimensionales o multidimensionales, siendo las primeras las más típicas, ya que asignan las propiedades ópticas a cada vóxel basándose en el valor escalar de una muestra del mismo. Por otro lado, existen dos tipos de clasificación, las cuales pueden producir resultados distintos: pre-clasificación, ocurre cuando se aplica la función de transferencia a cada vóxel antes de que estos sean interpolados; y post-clasificación, ocurre cuando se aplica la función de transferencia al resultado de la interpolación de las muestras escalares del volumen.

Las funciones de transferencia unidimensionales suelen ser funciones lineales a trozos, en donde a cada muestra  $s$  es aplicada a un color  $c(s) = RGB$  y una absorción  $\tau(s) = A$ , para conformar una tupla RGBA, con el fin de mapear estos valores, sin embargo, [E4] no contempla lo anterior, puesto que el color y la absorción están en función de  $\lambda \in [0, D]$ ; si se considera el análisis por post-clasificación, la ecuación puede ser reescrita como [E6], tal que  $x(\lambda)$  es la parametrización del rayo evaluada en  $\lambda$ , lo cual da como resultado un punto  $(x, y, z)$  evaluado en un campo escalar  $s(x, y, z)$  del volumen para obtener el valor muestra  $s$ .

$$C = \int_0^D c(s(x(\lambda)))\tau(s(x(\lambda)))e^{-\int_0^\lambda \tau(s(x(\lambda'))d\lambda'} d\lambda \quad [E6]$$

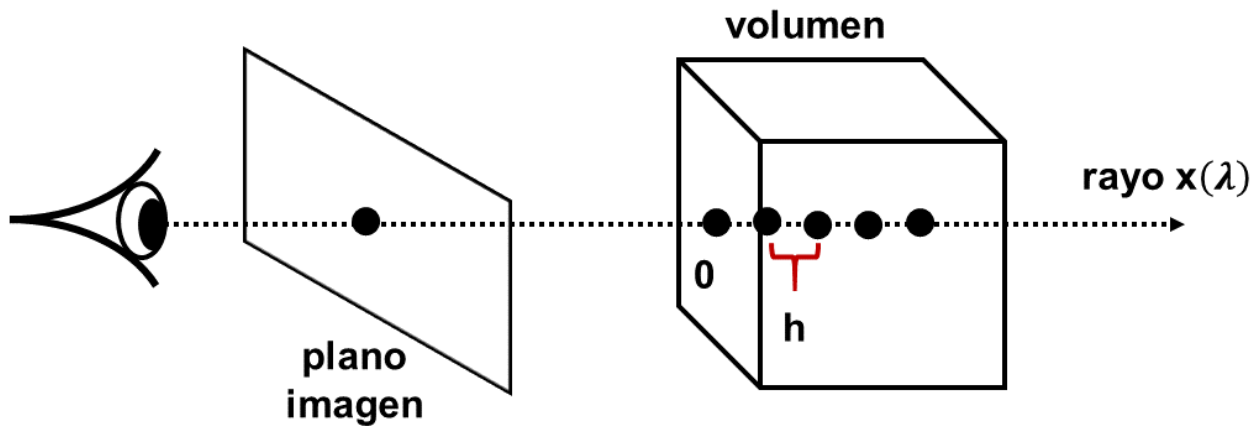
Para facilitar la evaluación de la ecuación, el rayo es dividido en segmentos  $n$ , con paso fijos de longitud  $h$ ; por lo que, al aplicar diversas consideraciones, sustituciones y simplificaciones, así como, contemplar la aproximación de la composición volumétrica (Lacroute, 1995)(Figura 26), la cual nos indica como determinar un píxel a partir de las contribuciones de color y opacidad de las muestras equidistantes y comúnmente constantes que se encuentran a lo largo del rayo que atraviesa el volumen, en donde la longitud  $h$  de cada segmento, se asume constantes la emisión y la absorción, la ecuación anterior puede escribirse de manera aproximada como [E7], tal que  $c_i = c(s_i)$  es el color y  $a_i = 1 - e^{h\tau(s_i)}$  es la opacidad.

$$C \approx \sum_{i=0}^{n-1} \prod_{j=0}^{i-1} (1 - a_j) a_i c_i \quad [E7]$$

Con todo esto, para evaluar esta expresión, se emplea un esquema de composición que obtiene el valor del color y opacidad, el cual puede utilizar un operador digital iterativamente en el orden *front to back* [E8] (muestra más cercana hasta la más lejana al ojo), o *back to front* [E9] (desde la muestra más lejana hasta la más cercana al ojo).

$$c := (1 - a) a_i c_i + c \quad a := (1 - a) a_i + a \quad [E8]$$

$$c := a_i c_i + (1 - a_i) c \quad a := a_i + (1 - a_i) a \quad [E9]$$



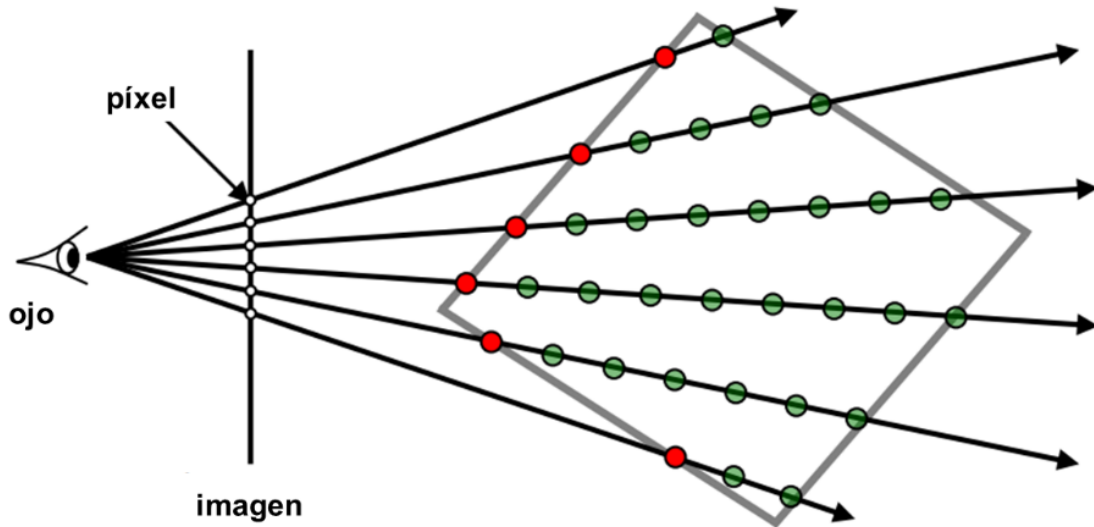
**Figura 26.** Aproximación de la Composición Volumétrica (Carmona, 2000).

### i. Ray Casting

El *ray casting* consiste en el lanzamiento de un rayo desde el ojo por cada centro de píxel de la imagen (Figura 27), para evaluar directamente la ecuación [E7]. Cada rayo es discretizado una vez que entra al volumen, este acumula un color y opacidad, empleando un paso  $h$  constante, en donde cada posición discreta del rayo, se muestrea el volumen, se clasifica la muestra y se compone con las muestras que la preceden; un filtro suele ser aplicado para obtener las interpolaciones requeridas. Existen varios algoritmos de *ray casting*, los cuales pueden variar en la forma en la que se calcula la intersección de los rayos con el volumen: el primer caso, calcula de manera individual la intersección de un rayo con la caja delimitadora (*bounding box*) del volumen mediante el algoritmo general de intersección rayo – caja; el segundo caso, calcula la intersección



de los rayos con el volumen mediante la rasterización por software del *bounding box* del volumen, al utilizar la interpolación lineal con corrección perspectiva.



**Figura 27.** Ray Casting: por cada centro de píxel de la imagen se lanza un rayo (Carmona, 2000).

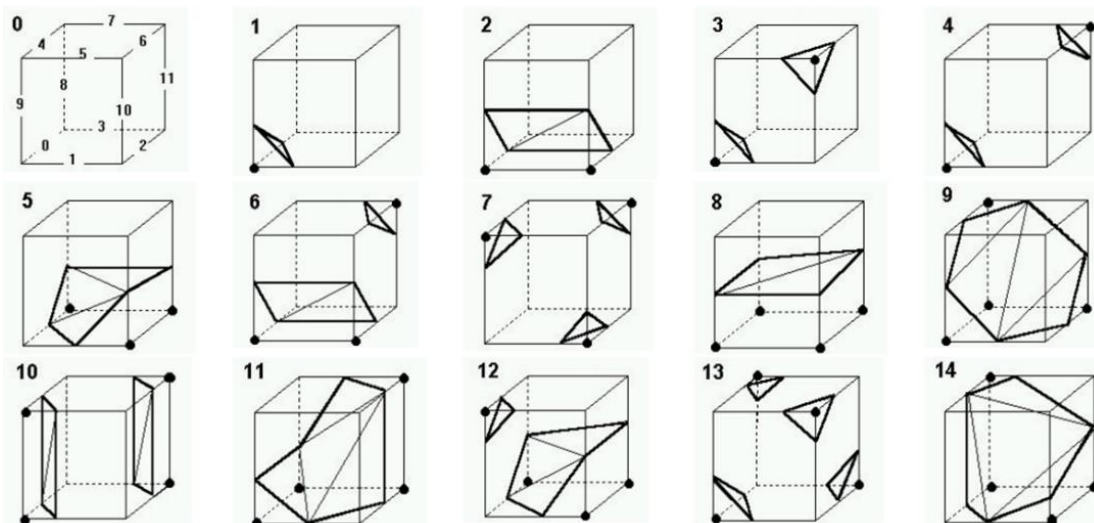
El algoritmo más común es el de Marc Levoy (Lacroute, 1995), propuesto en 1998, el cual describe el siguiente procedimiento: primero, se realiza el muestreo de cuadrículas y se aumenta el número de vóxeles en el eje para que estén alineados de manera adecuada al disponer de los datos; posteriormente, se lleva a cabo el sombreado para generar el color de cada vóxel, empleando el modelo estándar de sombreado de *Phong* (Fischer et al., 2005); después, se implementa la clasificación para precisar la opacidad de cada vóxel; luego, se configura el volumen para ejecutar el *ray casting*, de tal forma que se proyecta un rayo por cada píxel a través del volumen en la pantalla; como último paso, se aplica la composición, con el fin de calcular el color final del píxel en coordenadas de pantalla.

## ii. Estilo de Renderizado por Isosuperficie

El renderizado por isosuperficie (ISO) consiste en procesar capa por capa de un volumen de datos que representan un objeto tridimensional, en función de un determinado valor umbral (Figura 29 B). De esa manera, entre dos capas adyacentes se pueden vincular los contornos para establecer un esqueleto de polígonos, en donde el conjunto de polígonos encontrados entre todas las capas procesadas representa un sólido con una estructura intermedia, es decir, una superficie. Los algoritmos más comunes de este tipo de renderizado son: *marching cubes*, *opaque cubes*,

*marching tetrahedra* y *dividing cubes*. En seguida, se describe el primer algoritmo, pues resulta de importancia para este trabajo.

El algoritmo de *marching cubes* (Silvetti et al., 1999), descrito por Lorensen y Cline en 1987, busca extraer una superficie umbral a partir de una matriz volumétrica de datos escalares. De tal manera que se tiene un vóxel en el espacio delimitado por el valor de sus ocho vértices, lo que permite clasificarlo dado estos valores respecto al valor umbral; un vóxel contiene un trozo de la superficie umbral si por lo menos uno de sus vértices está por debajo del valor umbral y por lo menos otro está por encima, por lo que se puede asumir un valor por debajo o por encima del umbral. En total se tienen  $2^8 = 256$  posibilidades, pero por cuestiones de simetría por la reflexión y rotación se reducen en principio a solo 15 (Figura 28), ya que existe ambigüedades en la construcción de la superficie umbral entre vóxeles vecinos, por lo que a partir de la idea general de este algoritmo se han propuesto soluciones que varían la implementación.

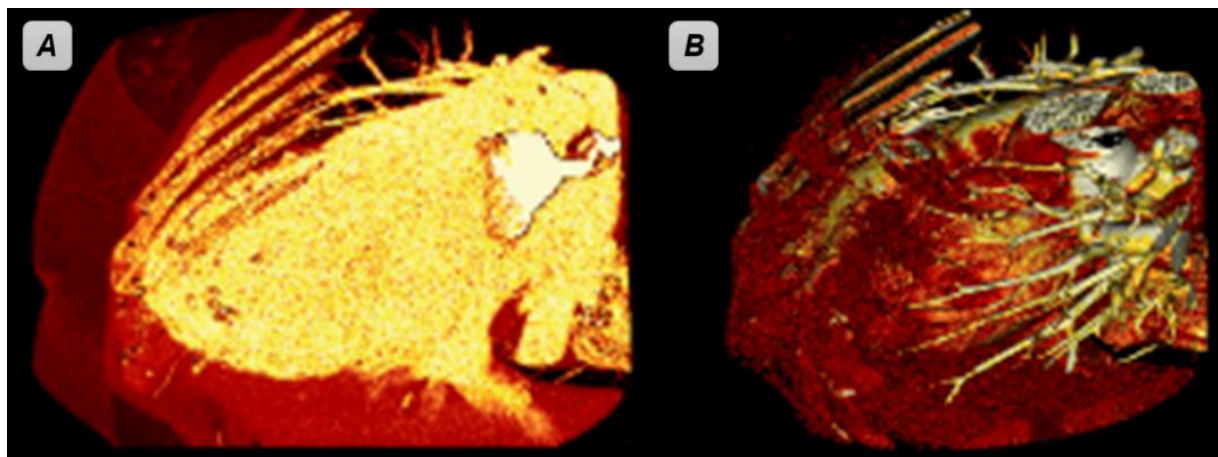


**Figura 28.** Algoritmo de *marching cubes*: 15 configuraciones originales (Silvetti et al., 1999).

### iii. Estilo de Renderizado por Proyección de Máxima Intensidad

La Proyección de Máxima Intensidad (MIP) es un método de reconstrucción 2D que permite realzar las estructuras con el mayor valor de muestra interpolado (Figura 29 A), es decir, de máxima atenuación, tal que a lo largo del rayo se usa para determinar el valor de cada píxel, lo que facilita una visualización rápida y efectiva de estructuras densas, como los vasos contrastados y huesos, lo cual resulta útil para visualizar densidades máximas dentro de datos de volumen (Napel et al., 1993), sin embargo, MIP no proporciona sombreado ni información de profundidad, a pesar de que

es posible crear una representación tridimensional al sumar una imagen bidimensional una tras otra, por tanto, las relaciones espaciales anatómicas pueden malinterpretarse e información clínica relevante puede ser descartada (Person et al, 2004).



**Figura 29.** Volume rendering. A. Representación 3D por MIP. B. Representación 3D por ISO.

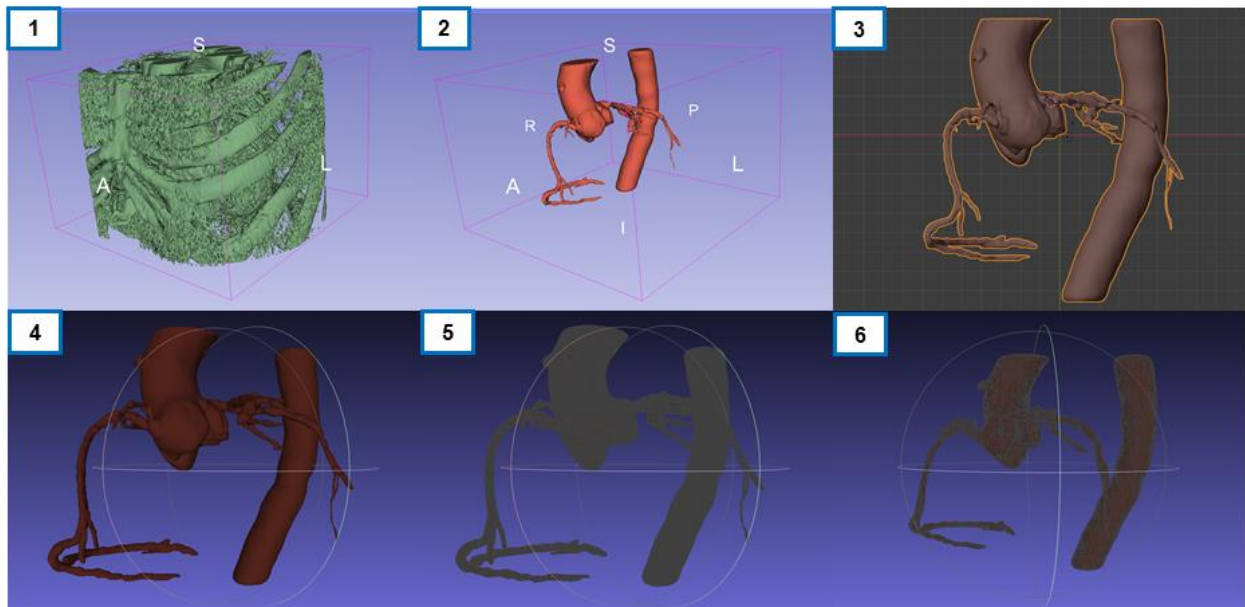
### 3.4 Visualización Web de Imágenes Médicas Reconstruidas

---

La reconstrucción de modelos 3D a partir de imágenes TC, es un proceso común para la representación de estructuras anatómicas, por lo que fue considerado implementar para crear los recursos necesarios y así integrarlos al sistema de visualización web. Para ello, se llevó a cabo el proceso de reconstrucción de la sección específica del corazón y del tórax, a través de varios softwares comerciales y libres que cuentan con algoritmos robustos para aplicar algunas técnicas que generen, editen o refinan mallas; así como el manejo adecuado de los formatos de visualización de modelos, de tal manera que estos puedan ser usados por las herramientas y el sistema mismo.

#### 3.4.1 Software para la Reconstrucción de Modelos 3D

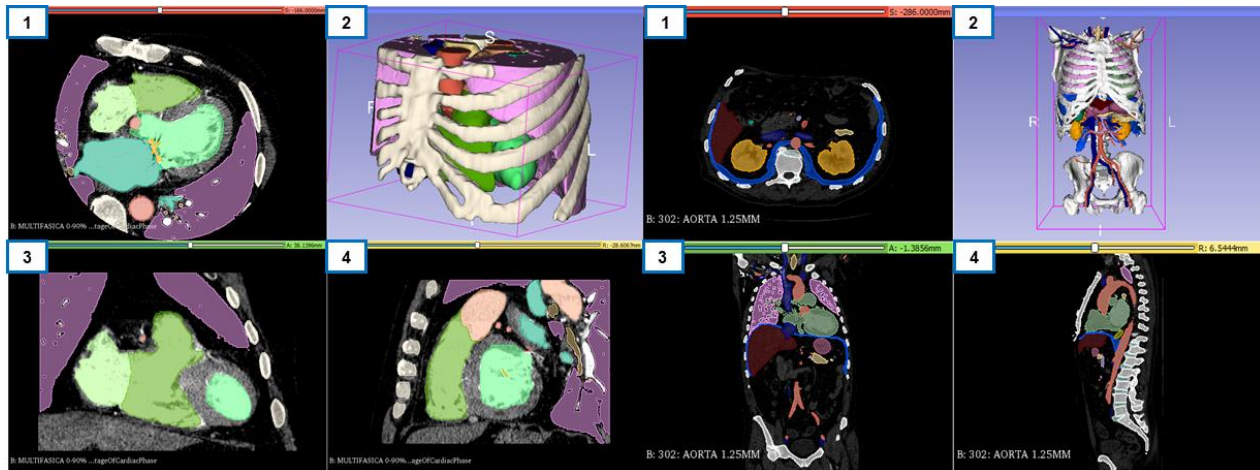
Para el proceso de reconstrucción del modelo del corazón y tórax (Figura 30) se empleó *3DSlicer*, como principal software para obtener un modelo 3D a partir de la selección y segmentación de un conjunto de imágenes extraídas por TC; después, el modelo fue editado en el software de *Blender* para eliminar los vértices duplicados e innecesarios en la malla, así como centrarlo en el origen y aplicar otras transformaciones básicas para su visualización en el sistema; por último, se refino la malla para aligerar el modelo a través del software de *MeshLab*, a partir de la simplificación de superficies triangulares, conservando los detalles geométricos e información de textura.



**Figura 30.** Proceso de reconstrucción del modelo 3D del corazón: aorta ascendente y descendente. 1. Segmentación general en 3DSlicer. 2. Segmentación específica en 3DSlicer. 3. Reducción de vértices duplicados y ubicación en el origen en Blender. 4. Carga del modelo en MeshLab. 5. Visualización original de vértices en MeshLab. 6. Visualización reducida de vértices en MeshLab.

#### i. 3DSlicer

3DSlicer es un paquete de software libre, de código abierto y multiplataforma, que permite cargar y manipular imágenes DICOM, para el análisis de imagen y visualización interactiva (imágenes volumétricas de vóxeles y mallas poligonales) científica, procesar tensores de difusión, servir de interfaz para dispositivos externos de guía por imagen, renderización de volúmenes por GPU, edición manual, segmentación automática, entre otras capacidades. Presenta una organización modular lo que permite añadir nuevas funcionalidades adaptadas a necesidades específicas de cada usuario. Este programa fue desarrollado inicialmente como un proyecto de tesis de maestría en 1998 (Slicer, 2021). En la Figura 31 se muestra la implementación de 3DSlicer para la reconstrucción del corazón y tórax, generados para el presente sistema.



**Figura 31.** Reconstrucción de modelo 3D en 3DSlicer: primeros 4 cuadrantes caso corazón (izquierda) y siguientes 4 cuadrantes caso tórax (derecha), tal que se representa 1. Plano transversal. 2. Vista 3D. 3. Plano Coronal. 4. Plano Sagital.

## ii. Blender

*Blender* es un programa informático multiplataforma, libre y de código abierto, dedicado especialmente al modelado, iluminación, renderizado, animación, simulación, composición y seguimiento de movimiento, edición de vídeo y creación de gráficos tridimensionales. Este programa fue desarrollado por Ton Roosendaal y lanzado por primera vez en 1994, además emplea una interfaz que utiliza *OpenGL*, lo que permite proporcionar una experiencia consistente (Blender, 2021).

## iii. MeshLab

*MeshLab* es un software libre, multiplataforma y de código abierto para el procesamiento y edición de mallas triangulares no estructuradas en 3D, a través de sus herramientas para edición, optimización, inspección y renderizado de mallas, tal que permite remover vértices duplicados, homogeneizar una malla, subdividir superficies, reconstruir mallas, entre otros aspectos más; desarrollado por el centro de investigación ISTI – CNR en el 2005 y empleado ampliamente bajo el contexto de investigación, académico y desarrollo, en diversas áreas de ciencias e ingeniería (Cignoni et al., 2008).

### 3.4.2 Formatos para la Visualización de Reconstrucciones de Modelos 3D

Los formatos empleados para manejar archivos de imágenes en 3D permiten almacenar datos en cuanto a coordenadas, archivos, mapa de texturas, información de colores, entre otras



características, lo que posibilita emplear los modelos 3D entre diversas herramientas de software e implementaciones que se desarrollen. En este trabajo, durante el proceso de reconstrucción de modelos tridimensionales, así como su implementación, implicó trabajar del formato OBJ al VTK.

#### i. OBJ

El formato OBJ es conocido como *Wavefront 3D Object File*, el cual fue desarrollado por *Wavefront Technologies* en 1990, e indica un formato de archivo estándar de definición de geometrías, empleado principalmente para representar objetos tridimensionales, que contiene las coordenadas 3D de cada vértice, la posición UV de cada vértice de coordenadas de textura, las normales de los vértices, entre otra información. Por defecto, los vértices se almacenan en orden contrario a las agujas del reloj y las líneas de información están escritas en ASCII. Además, en los archivos OBJ se especifican otros dos formatos: MTL (*Material Template Library*), describe las propiedades de sombreado de la superficie de los objetos; y STL (*Standard Triangle Language*), describe sólo la geometría de la superficie de un objeto 3D sin ninguna representación de color, textura u otro atributo (LOC, 2020).

#### ii. VTK

El Kit de Herramientas de Visualización (VTK), por sus siglas en inglés, es un conjunto de herramientas de código abierto con licencia BSD, creado en 1993 por los investigadores de *GE Corporate R&D*, para gráficos por computadora en 3D, modelado, procesamiento de imágenes, reproducción de volumen, visualización científica y trazado en 2D. Por tanto, esta herramienta proporciona, entre otras cosas, un formato de archivo que ofrece un esquema de representación de datos consistentes para una variedad de tipos de conjuntos de datos, y proporcionar un método simple para comunicar datos entre software, el cual consta de: versión e identificador, encabezado, formato de archivo (ASCII o binario), estructura del conjunto de datos (describe la geometría y topología del *dataset*), así como la descripción de los atributos del conjunto de datos (VTK, 2021).

### 3.5 Visualización Web Mixta

---

Implementar experiencias de realidad virtual, aumentada o mixta, permiten proveer sistemas más atractivos e interactivos, que en la visualización de estructuras anatómicas ha resultado de gran utilidad e interés. Para ello, a través de Three.js se aplicó el estándar *WebXR*, con el fin de poder representar la vista 3D, el modelo de mapa de colores y el modelo 3D en realidad virtual por medio

del navegador del *Oculus Quest 2*, dispositivo empleado en este caso para realizar una prueba de concepto de esta funcionalidad en el sistema de visualización web desarrollado.

### 3.5.1 WebXR

*WebXR* es un estándar para hacer accesible la realidad virtual y aumentada (XR) a través de la Web, lanzado por el departamento de *Google Creative Lab*, al ofrecer una API que proporciona las interfaces necesarias para permitir a los desarrolladores crear aplicaciones inmersivas, atractivas, cómodas y seguras en la Web en una variedad de factores de forma de hardware, tal que su representación está a cargo de *WebGL*, de manera directa o a través de marcos de trabajo de *JavaScript*, como *Three.js*, lo que proporciona un contenido virtual alineado con el mundo real antes de mostrarse al usuario. Actualmente, solo está disponible, y por ende compatible, en *Microsoft Edge* y *Chrome* (W3, 2020).

### 3.5.2 Oculus Quest 2

El *Oculus Quest 2* es un visor de realidad virtual, desarrollado por *Oculus* una marca de Facebook, capaz de funcionar como un auricular independiente con un sistema operativo basado en Android y con un software de realidad virtual compatible con *Oculus* que se ejecuta en una computadora de escritorio en una conexión a través de USB o Wi-Fi, el cual cuenta con 6 GB de memoria, de 64 GB a 256 GB de almacenamiento interno, un procesador Snapdragon Qualcomm XR2 y una resolución de 1832 x 1920 píxeles, como características distintivas de este dispositivo (Oculus, 2020), tal que uso resulta diverso como en el caso de aplicaciones médicas (Figura 32).



**Figura 32.** Aplicaciones contemporáneas de realidad virtual con dispositivos de interacción en aplicaciones médicas (Adaptado de Teodoro, 2021). A. Sistema interactivo para la planeación quirúrgica. B. Navegación virtual en el interior de estructuras vasculares del corazón en el Centro de Simulación CMSXII del IMSS. C. Sistema de realidad virtual para el entrenamiento de residentes en algología, para procedimiento de punción del ganglio trigémino en el Centro de Destrezas del HGM.

---

# Capítulo 4

## Sistema de Visualización Web de Imágenes Médicas

---

El sistema de visualización web de imágenes médicas está compuesto por un conjunto de bloques de software, tal que sus funcionalidades están dirigidas a generar y proporcionar diversos recursos por medio de representaciones y elementos gráficos en un navegador web. Por tanto, se trata de un medio disponible y accesible desde Internet, en donde se presenta una interfaz interactiva con diferentes tipos de vistas de dos casos médicos, sección del corazón y tórax, la cuál puede ser manipulada por el usuario con fines de enseñanza, aprendizaje, diagnóstico o planeación quirúrgica. En este capítulo, se presenta el diseño y desarrollo del sistema, así como la implementación de la carga y visualización de imágenes y modelos, para el renderizado de gráficos bidimensionales y tridimensionales, de manera tradicional e inmersiva.

### 4.1 Arquitectura

---

El sistema se desarrolló en lenguaje de marcas HTML y el lenguaje de programación *JavaScript*, en la especificación de *OpenGL Web* (WebGL) a través del marco de Three.js, de tal forma que la compatibilidad resulta diversa en cuanto a dispositivos y plataformas, en donde el soporte requiere como requisito mínimo ciertas características en cuanto al procesamiento gráfico, las cuales se muestran en la siguiente tabla.

**Tabla 6.** Especificación de dispositivos y plataformas compatibles con el sistema de visualización web de imágenes médicas.

Dispositivo	Sistema Operativo	Característica	Navegador
Escritorio	Windows	Versiones posteriores, iguales o similares a controladores NVIDIA 257.21, ATI/AMD 10.6 e Intel 2010	Microsoft Edge, Google Chrome, Firefox, Safari y Opera
	Mac	Versiones posteriores o iguales a Mac OS 10.6	
Móvil	iOS	PowerVR SGX (535, 543 y 554) y versiones posteriores	
	Android	Android 2.2 (API nivel 8) y versiones posteriores	

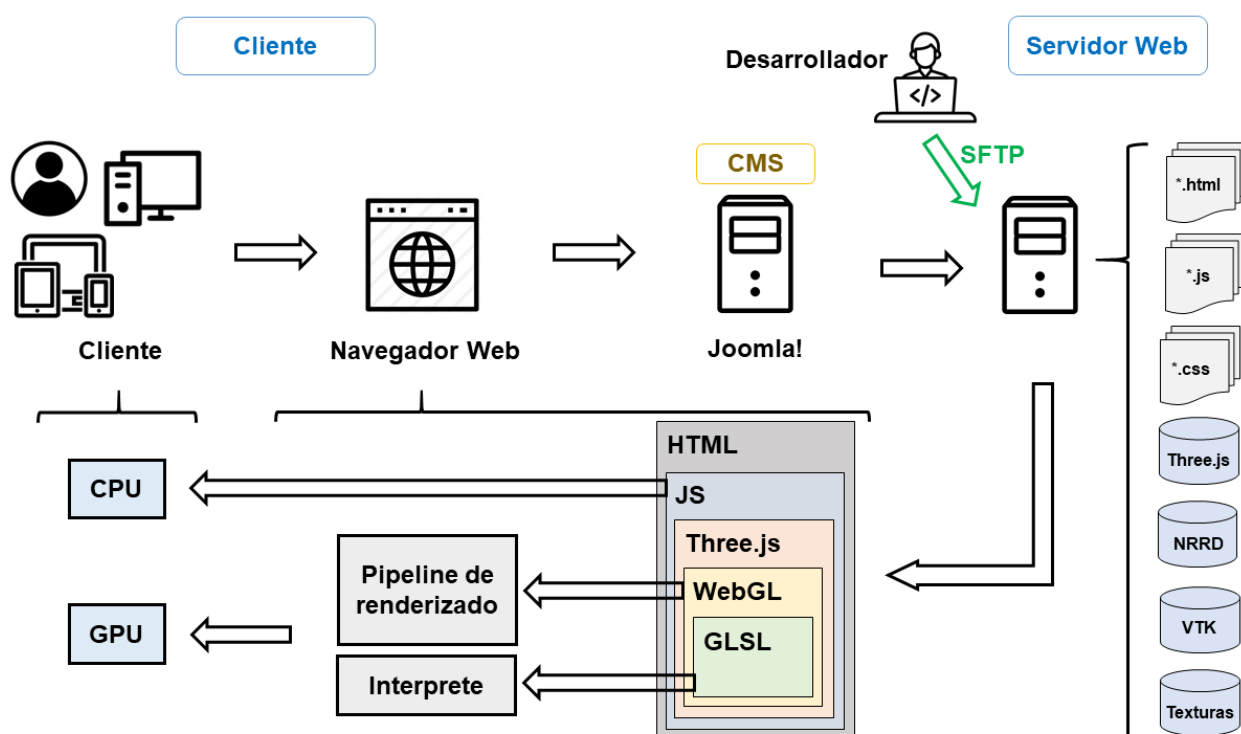
\* Para más detalle consultar *Graphics Driver and Feature Blacklists and Whitelists* en <https://www.khronos.org/webgl/wiki/>

La implementación del sistema web se realizó bajo la arquitectura cliente – servidor, por lo que se puede especificar (Figura 33): primero, el usuario realiza peticiones a través del navegador



web; después, las peticiones son recibidas y gestionadas por Joomla (Joomla, 2021), un sistema de gestión de contenidos (CMS) que permite desarrollar sitios web dinámicos e interactivos al presentar una arquitectura modelo-vista-controlador (MVC) para crear, modificar o eliminar contenido de forma sencilla en él, de tal forma que el CMS interactúa con el servidor web, el cual contiene los diversos scripts, bibliotecas y recursos necesarios para que funcione el sistema; finalmente, una vez identificados los recursos solicitados, son devueltos al CMS para que se regrese una respuesta al usuario a través del navegador.

Cabe destacar que en el último punto especificado, las operaciones relacionadas a la presentación de la información, involucradas principalmente con los scripts HTML, JS y CSS serán resueltas en la CPU del cliente, mientras que las operaciones relacionadas con WebGL y GLSL, que describen procesos de renderización pesados, serán resueltas en la GPU del usuario por medio del pipeline de renderizado y la interpretación que se requiere, según como se especificó en el capítulo anterior, con el fin de generar la visualización de las imágenes médicas con su respectiva funcionalidad respecto a lo que se ha solicitado.

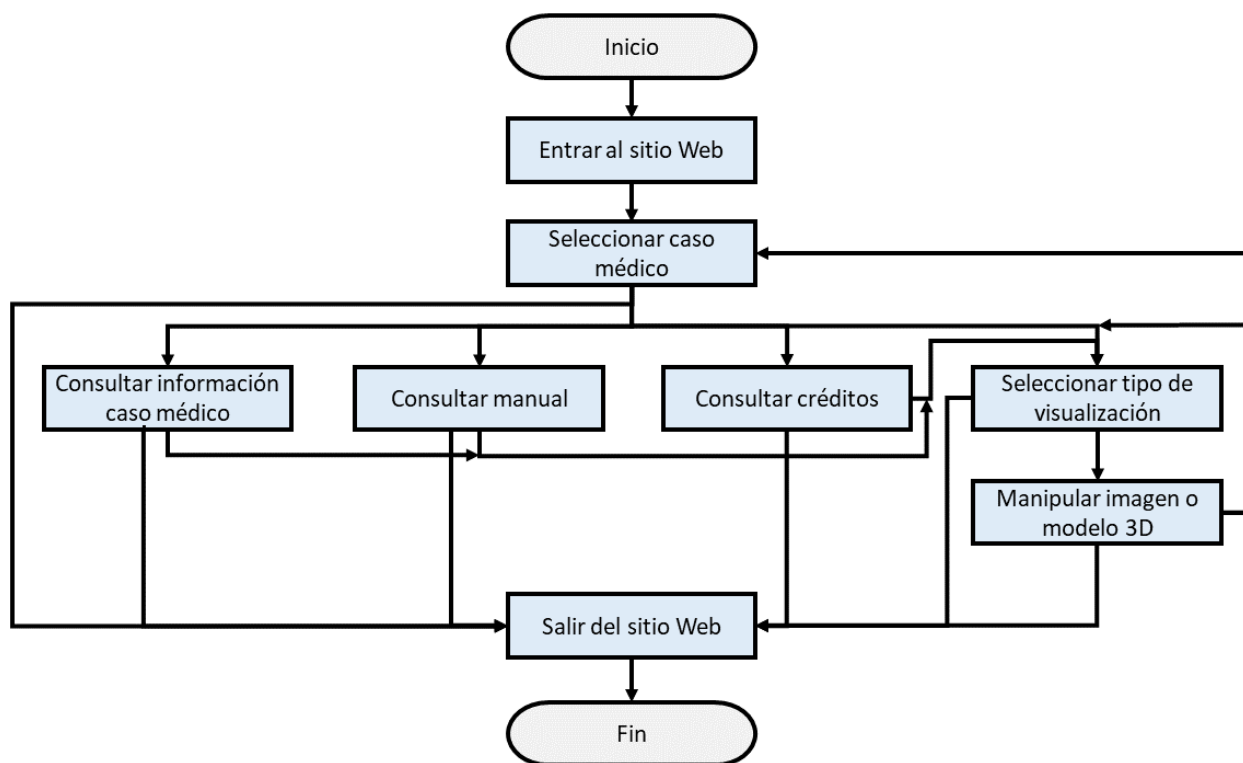


**Figura 33.** Arquitectura del sistema de visualización web de imágenes médicas.

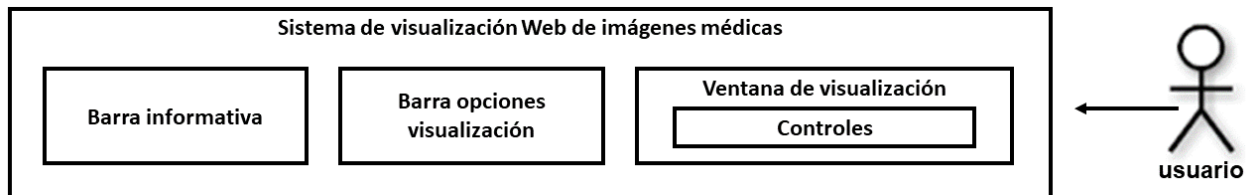
Por otro lado, es importante indicar que el manejo de los recursos desarrollados y requeridos para que estén disponibles en el servidor Web y el sistema de visualización web

funcione, se realizó mediante el protocolo SFTP, por sus siglas en inglés Protocolo de Transferencia Segura de Archivos, el cuál es un método estándar para transferir archivos o datos entre un equipo local y un espacio web que se dispone en el hosting empleado en un servidor Web a través de *Secure Shell* (SSH); por lo que, se empleó FileZilla (FZ, 2021), como una herramienta para realizar la respectiva configuración como clientes y establecer una conexión con el servidor y de esta manera transferir la información necesaria, durante el desarrollo del presente sistema que se describe.

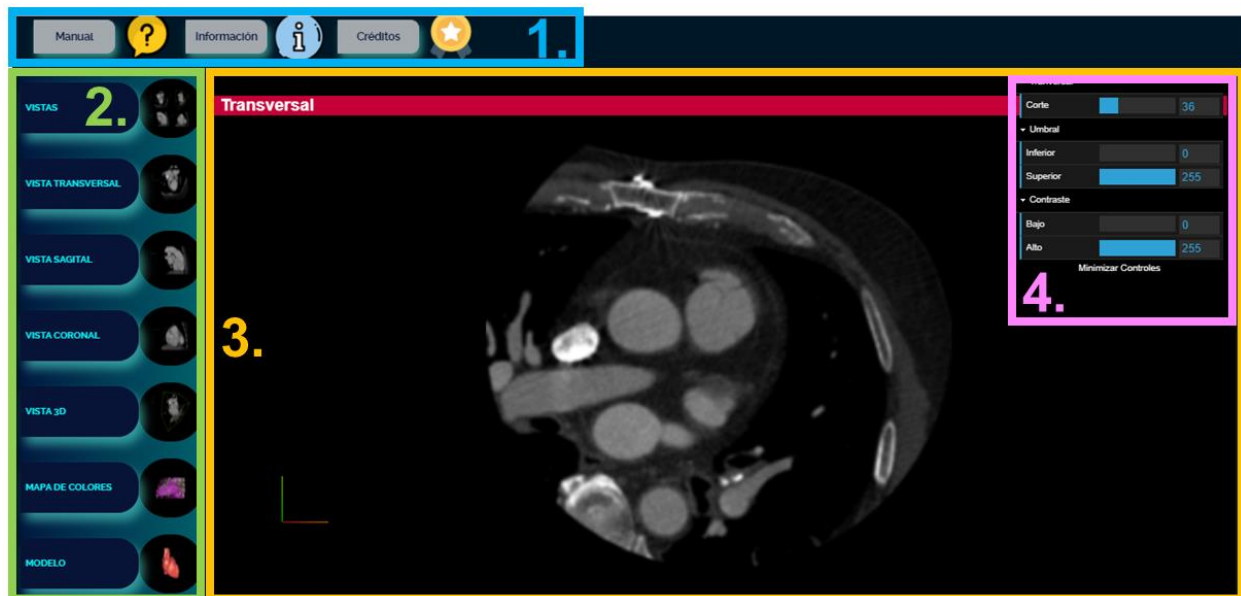
El flujo del sistema de visualización web, representado en la Figura 34, muestra las acciones de forma general que puede realizar un usuario desde que ingresa al sitio web hasta la obtención de los resultados. Mientras que en la Figura 35 se presenta una abstracción de los subsistemas que conforman al sistema por medio de un caso de uso que describe las interacciones entre las acciones o actividades que un usuario puede realizar, tal que en el Anexo A se desglosa cada uno de ellos. De esta manera, en la Figura 36 se expone la representación gráfica del sistema de visualización que describen el diagrama de flujo y caso de uso. En las secciones siguientes, se analizarán y describirán los elementos y funcionalidades que definen el presente desarrollo por medio de estos esquemas.



**Figura 34.** Diagrama de flujo del sistema de visualización web de imágenes médicas.



**Figura 35.** Abstracción del caso de uso del sistema de visualización web de imágenes médicas.



**Figura 36.** Representación gráfica del sistema de visualización web de imágenes médicas. 1. Barra informativa. 2. Barra de opciones de visualización. 3. Ventana de visualización. 4. Controles.

## 4.2 Elementos generales

Para poder generar cada una de las vistas que distinguen al sistema de visualización web, es decir, la visualización bidimensional y tridimensional de una TC en formato NRRD, así como la representación 3D por medio de modelos renderizados a partir de reconstrucciones o creados en tiempo real, se desarrollaron diversos scripts HTML que definen cada una de ellas, de modo que se importó la biblioteca de Three.js por medio de los módulos necesarios a incluir dentro de la definición de la página web, en donde se configuraron y emplearon las siguientes características en común:

## 4.2.1 Escena, Renderizador y Cámara

Con el fin de visualizar un gráfico a través de un navegador con Three.js, es necesario contar con 3 elementos principales: escena, renderizador y cámara, por lo tanto, se establece la relación para renderizar la escena con la cámara. De esta manera, se tiene que la escena, es un espacio o contenedor en donde se van a importar todos los elementos necesarios para renderizarlos, los cuales serán los únicos que puedan ser visualizados; el renderizador, es el motor que genera la representación gráfica; mientras que la cámara, es lo que permite indicar la sección que se va a renderizar de la escena. A continuación, se especifica la definición y configuración de la escena y el renderizador.

**Tabla 7.** Escena y renderizador en Three.js: extracto de la definición de la vista 3D del sistema de visualización web de imágenes médicas.

Definición
<pre>// Se importa el módulo de THREE.JS import * as THREE   from '../.../three.js-master/build/three.module.js';  // Se declaran las respectivas variables a emplear var sceneVV, rendererVV, containerVV;  // Se crea la escena para la vista seleccionada sceneVV = new THREE.Scene();  // Se crea el renderizador para dibujar la escena rendererVV = new THREE.WebGLRenderer();  // Se establece la proporción de píxeles de la ventana para evitar una salida borrosa rendererVV.setPixelRatio( window.devicePixelRatio );  // Se establece el tamaño de la ventana de salida (ancho y alto), // considerando la proporción de píxeles rendererVV.setSize( window.innerWidth, window.innerHeight );  // Se crea un contenedor tipo DIV y se configura para que pueda ser añadido al DOM // con el fin de que se cree el canvas containerVV = document.createElement( 'div' ); document.body.appendChild( containerVV ); containerVV.appendChild( rendererVV.domElement );</pre>

En el programa se hace uso de las vistas de proyección ortográfica y en perspectiva, a través de una cámara virtual que proyecta, discretiza y muestra un objeto dibujado computacionalmente de una manera que se asemeja más al modo en el que visualizamos un objeto físico con los ojos. En la Tabla 8, se indica la definición en Three.js de la cámara en las respectivas proyecciones y en que visualizaciones se emplea.

**Tabla 8.** Cámaras virtuales en Three.js: extracto de la definición de la vista de mapa de colores y 3D del sistema de visualización web de imágenes médicas.

Tipo de proyección	Definición	Vista que la emplea
Ortográfica	<pre> // Se importa el módulo de THREE.JS import * as THREE   from '../.../three.js-master/build/three.module.js';  // Se definen las respectivas variables var cameraVV; const viewSize = 512; // Altura del frustum const aspectRatio = window.innerWidth / window.innerHeight;  // Se crea una instancia de una cámara en proyección ortográfica THREE.OrthographicCamera (left, right, top, bottom, plane, near, far) cameraVV = new THREE.OrthographicCamera(   - viewSize * aspectRatio / 2,   viewSize * aspectRatio / 2,   viewSize / 2, - viewSize / 2, 0.001, 3000 );  // Se configura la posición inicial y la dirección de giro de la cámara dada su orientación cameraVV.position.set( 300, -300, -300 ); cameraVV.up.set(0, 0, -1);  // Añadimos la cámara a la escena sceneVV.add( cameraVV ); </pre>	<p>Transversal</p> <p>Sagital</p> <p>Coronal</p> <p>Mapa de Colores</p>
Perspectiva	<pre> // Se importa el módulo de THREE.JS import * as THREE   from '../.../three.js-master/build/three.module.js';  // Se define la variable var cameraVV;  // Se crea una instancia de la cámara en proyección en perspectiva THREE.PerspectiveCamera(fov, aspect, near, far) cameraVV = new THREE.PerspectiveCamera(   60, (window.innerWidth / window.innerHeight),   0.01, 1e10 );  // Se configura la posición inicial y la dirección de giro de la cámara dada su orientación cameraVV.position.z = -300; cameraVV.position.y = 300; cameraVV.position.x = 300; cameraVV.up.set(0, 0, -1);  // Añadimos la cámara a la escena sceneVV.add( cameraVV ); </pre>	<p>3D</p> <p>Modelo</p>

Es importante destacar, que el modo de proyección ortográfico permite que el tamaño del objeto en la imagen renderizada permanezca constante independientemente de la distancia de la cámara, lo que resulta útil para renderizar escenas 2D y elementos de interfaz de usuario, por el contrario, el modo de proyección en perspectiva está diseñado para imitar la forma en la que ve el

ojo humano, por lo que es comúnmente empleado para renderizar una escena 3D. No obstante, en la visualización de la vista de mapa de colores, que corresponde al modelo 3D en tiempo real, no cumple con lo anterior, ya que se emplea una cámara que utiliza proyección ortográfica, sin embargo, su uso está ligado al proceso de *volume rendering* implementado y soportado por Three.js.

#### 4.2.2 Iluminación

Dotar una escena con mayor realismo y efectos visuales requiere de la representación de ciertos fenómenos físicos de forma computacional, de tal manera que en este caso se emplearon dos modelos de iluminación: el primero, una fuente de luz hemisférica, con el objeto de representar lo más cercano a la luz natural, permitiendo así crear un efecto de luminosidad más normal para escenas interiores, simulando superficies reflectantes y un clima débil, por lo que no afecta a las sombras; y el segundo, una fuente de luz direccional, también conocida como luz infinita, con el propósito de crear una fuente de luz paralela como la luz solar, la cual afecta a las sombras. De esta manera, la implementación de estas dos fuentes de iluminación, se indica en la siguiente tabla.

**Tabla 9.** Fuentes de iluminación en Three.js: extracto de la definición de la vista transversal del sistema de visualización web de imágenes médicas.

Definición
<pre>// Se define y crea una instancia de la fuente de luz hemisférica var hemiLightVT = new THREE.HemisphereLight( 0xffffff, 0x000000 , 1 ); // Se agrega la fuente de luz a la escena sceneVT.add( hemiLightVT );  // Se define y crea una instancia de la fuente de luz direccional var dirLightVT = new THREE.DirectionalLight( 0xffffff, 0.5 ); // Se indica la posición de la fuente de luz en la escena dirLightVT.position.set( 200, 200, 200 ); // Se agrega la fuente de luz a la escena sceneVT.add( dirLightVT );</pre>

En el caso de la visualización de mapa de colores, en donde se renderiza el modelo 3D en tiempo real, no se especifican directamente los modelos de iluminación, ya que la iluminación es tratada en el *shader* empleado para aplicar el *volume rendering*, es decir, se emplea un *baked lights* o cocinado de luces, de tal manera que se calcula previamente la iluminación antes del tiempo de ejecución.

### 4.2.3 GUI y Controles

Definir la interacción entre los elementos gráficos mostrados en la visualización y el usuario, se logró al implementar una interfaz gráfica de usuario, mejor conocida como GUI, con el fin de definir un elemento gráfico, un menú, para facilitar y simplificar las acciones posibles que puede realizar un usuario basadas en la interrelación a través del mouse o su equivalente, según el dispositivo que se emplee, para poder comunicarse con el navegador; este último punto, a partir de la puesta en funcionamiento de controles de órbita, elemento de Three.js que permite que la cámara orbite alrededor de un objeto. Por lo que, la configuración de ambos elementos dentro de cada visualización posibilita la operatividad del sistema web (Figura 37), en la Tabla 10 se indica la definición empleada de estos elementos, en donde se muestra que la configuración de las opciones del menú está ligada, en este caso, a las propiedades de la imagen o modelo de una determinada vista, aspectos que se retomaran en secciones posteriores.

**Tabla 10.** GUI y controles en Three.js: extracto de la definición de la vista sagital del sistema de visualización web de imágenes médicas

#### Definición

```
// Se importa el módulo de INTERFAZ GRÁFICA DE USUARIO y CONTROLES DE ÓRBITA
import { GUI } from '../../../three.js-master/examples/jsm/libs/dat.gui.module.js';
import { OrbitControls } from '../../../three.js-master/examples/jsm/controls/OrbitControls.js';

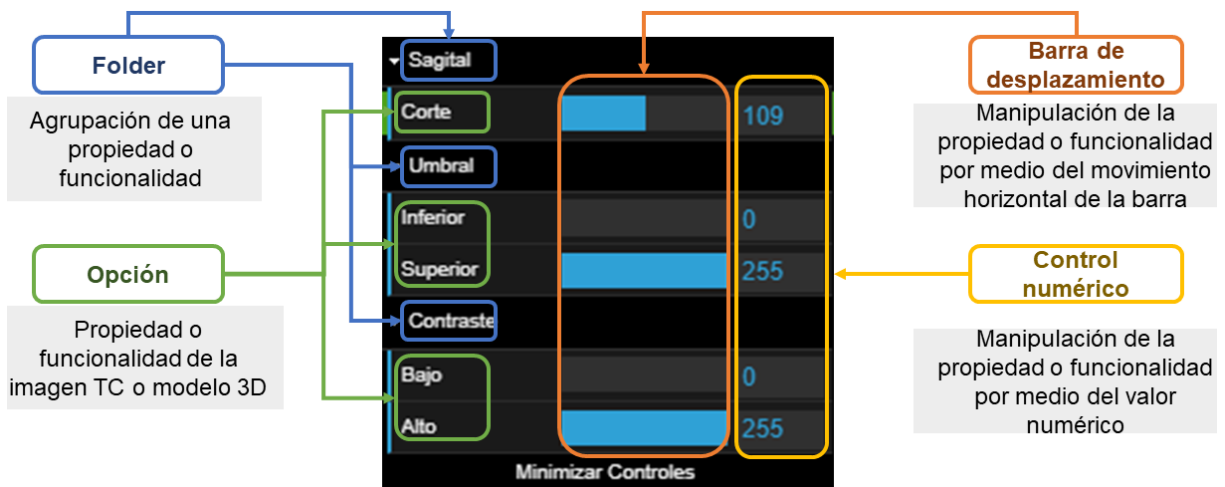
// Se declaran las variables necesarias para manejar el GUI y los controles
var guiVS, corteSagitalFolderVS, umbralFolderVS, contrasteFolderVS, controlsVS;

// Se crea y configura una instancia de los controles
controlsVS = new OrbitControls( cameraVS, rendererVS.domElement );
controlsVS.rotateSpeed = 0;
controlsVS.minZoom = 0.8;
controlsVS.maxZoom = 3;
controlsVS.panSpeed = 0.3;
controlsVS.zoomSpeed = 0.3;

// Se actualizan los controles al realizar una transformación manual de la cámara
controlsVS.update();

// Se crea una instancia del GUI, tal que se agrega al canvas
guiVS = new GUI();
// Se crean folders para especificar y agrupar las opciones en el menú
corteSagitalFolderVA = guiVS.addFolder('Transversal');
umbralFolderVS = guiVS.addFolder('Umbral');
contrasteFolderVS = guiVS.addFolder('Contraste');

. . .
// En el apartado de carga de la imagen TC en formato NRRD se configuran y
agregan las opciones del menú, con base a las propiedades de la imagen
// Control de visualización transversal de los slices
corteAxialFolderVA.add(
    sliceZVA, "index", 0, volume.RASDimensions[ 2 ], 1 ).name( "Corte" ).onChange(
        function () { sliceZVA.repaint.call( sliceZVA ); }
);
. . .
```



**Figura 37.** Representación gráfica de la GUI de la vista sagital en el sistema de visualización web de imágenes médicas

Las definiciones anteriores no solo se configuran de manera general dentro de un elemento script de HTML, puesto que, de ser así, no se podría visualizar nada en el navegador, por lo que la estructura que se requiere precisar en el programa que define cada visualización se muestra en el Anexo B, con el objetivo de conjuntar de manera adecuada la especificación de los elementos previos.

### 4.3 Recursos Visuales

El sistema de visualización desarrollado se destaca por implementar y poner a disposición del usuario 3 tipos de recursos por medio de las diversas vistas: imagen TC, modelo 3D en tiempo real y modelo 3D reconstruido. En lo siguiente se describirá de manera detallada las particularidades de cada recurso.

#### 4.3.1 Imágenes TC

Para poder cargar imágenes TC en el navegador, se procedió a seguir el siguiente procedimiento:

##### i. Conversión de Formato de Imágenes TC

El primer paso, consta de convertir las imágenes TC en formato DICOM a formato NRRD empleando FIJI, ya que Three.js proporciona un cargador de este modelo de archivos, como un medio para tratar con este tipo de recursos, por lo que el algoritmo usado para dicha acción se especifica en la Tabla 11 con base al software y particularidades indicadas en el capítulo 3, en



donde cabe precisar que el tipo de dato seleccionado se determinó al analizar los metadatos del archivo DICOM con las imágenes de TC y los admitidos por el cargador, mientras que exportar las imágenes en diferentes resoluciones se realizó con dos objetivos: el primero, determinar cuál era el más conveniente como un recurso solicitado al servidor, puesto que esto implica un peso de archivo y por ende un tiempo de carga en el navegador diferente dada la resolución; y el segundo, verificar la calidad de la imagen mostrada en la visualización.

**Tabla 11.** Algoritmo de conversión de imágenes TC DICOM a NRRD en FIJI

<b>Algoritmo</b>
<pre> Inicio    resolución ← [128, 256, 512, 1024]    Para Cada i ∈ resolución Hacer     Importar la secuencia de imágenes TC DICOM     Cambiar el tipo de dato a 8 bits de las imágenes     Cambiar la escala de las imágenes a una resolución i de píxeles     Voltear las imágenes horizontalmente     Exportar las imágenes en formato NRRD   Fin Para  Fin </pre>

El anterior punto, llevó a definir que lo más ideal para las vistas de imagen TC es una resolución de 128 píxeles, mientras que para el caso del modelo 3D en tiempo real, que igual implementa un archivo NRRD, es de 256 píxeles. Así mismo, se consideró manejar imágenes cuadradas, es decir, con resolución en potencia de 2, como una característica que permite optimizar ciertas configuraciones en el hardware de gráficos. Por otro lado, voltear las imágenes de forma horizontal, se realizó para mostrar en la posición adecuada la anatomía, es decir, si estamos en el plano coronal en una vista anterior del corazón, este debe observarse del lado derecho, como una forma de restar trabajo en cuanto a operaciones computacionales, si se aplicarán las transformaciones por medio de código a la respectiva matriz con los datos de las imágenes.

## ii. Configuración de Script para Cargar, Visualizar y Manipular las Imágenes TC

El segundo paso, consta de especificar en el script de las vistas que proveen el recurso de imagen TC las configuraciones necesarias para que el archivo NRRD pueda ser cargado, visualizado y manipulando. Lo que permitió generar 4 tipos de vistas: transversal, sagital, coronal y 3D, tal que se las primeras 3 proporcionan una visualización individual de los planos en los que se estudia el cuerpo humano, mientras que la última, provee una visualización tridimensional de los tres cortes, a partir del manejo de la representación geométrica del caso de estudio y un determinado material

para generar una malla, que es renderizada como una imagen en el navegador. Así mismo, la manipulación a través del uso de la GUI y los controles facilito definir: el control del corte y plano a visualizar, pues el conjunto de imágenes cargadas representa un volumen; el control de la escala *thresholding* y *window*, cuyo funcionamiento se especificó en el capítulo anterior; así como el control de vistas *default*, con el fin de colocar, en el caso de la vista 3D, en una determinada posición las imágenes tridimensionales sin la necesidad de trasladar o rotar manualmente.

En este sentido, la Tabla 12 se muestra la definición implementada en una de las representaciones, misma que siguen las demás; mientras que en la Figura 38 se expone el resultado final de las opciones de visualización de imágenes TC, en donde cabe señalar que se concentró en una opción de vista todas ellas, identificada en el sistema web en la barra de opciones de visualización como “vistas”, a fin de dotar diversas representaciones visuales de una imagen TC al momento de requerir analizarla para enseñar, aprender, diagnosticar o planificar.

**Tabla 12.** Carga de imágenes TC en formato NRRD en Three.js: extracto de la definición de la vista 3D del caso de corazón en el sistema de visualización web de imágenes médicas.

Definición
<pre> // Se importa el módulo del CARGADOR DE ARCHIVOS NRRD (Nearly Raw Raster Data) import { NRRDLoader }   from '../.../three.js-master/examples/jsm/loaders/NRRDLoader.js';  // Se crea una instancia del cargador NRRD var loader = new NRRDLoader();  // Se cargan las imágenes TC en formato NRRD, tal que se pasa como argumento una función que al mismo tiempo se ejecuta, la cual se encarga de verificar los metadatos del archivo para obtener la información y parámetros clave, tal que se podrán manipular de diversas formas para definir diferentes funcionalidades loader.load( "../modelos/nrrd/pruebaCorazon1-8B-128P-F.nrrd", function ( volume ) {    //Se definen las variables para manejar los planos que define la TC   var sliceZVV, sliceYVV, sliceXVV;    // Se crea una instancia para manejar la representación geométrica del caso   var geometryVV =     new THREE.BoxBufferGeometry( volume.xLength, volume.yLength, volume.zLength );    // Se crea una instancia para manejar un material que dibuje el caso de manera sencilla   y sombreada (no se ve afectado por las luces)   var materialVV = new THREE.MeshBasicMaterial( { color: 0x00ff00 } );    // Se crea una caja auxiliar para visualizar el volumen en la vista 3D   var cubeVV = new THREE.Mesh( geometryVV, materialVV );   cubeVV.visible = false;    var boxVV = new THREE.BoxHelper( cubeVV );   sceneVV.add( boxVV );    boxVV.applyMatrix4( volume.matrix );   sceneVV.add( cubeVV );    // Se definen las variables para controlar la visibilidad de los cortes </pre>

---

```

const visibilityControlXVV, visibilityControlYVV, visibilityControlZVV =
  { visible: true };

// Se obtienen el corte de un determinado plano y se agrega a la escena
// Plano X vista Sagital de la TC
sliceXVV = volume.extractSlice( 'x', Math.floor( volume.RASDimensions[ 0 ] / 2 ) );
sceneVV.add( sliceXVV.mesh );

// Se aplica lo mismo para la vista coronal (RASDimensions[ 1 ] / 2)
// a la vista transversal (RASDimensions[ 0 ] / 2)
. . .

// Se definen y agregan las diferentes opciones de la GUI
// Control de visualización sagital de los slices
corteSagitalFolder.add( sliceXVV, "index", 0,
volume.RASDimensions[ 0 ], 1 ).name( "Corte" ).onChange(
  function () {
    sliceXVV.repaint.call( sliceXVV );
  }
);

// Control de la visibilidad del plano X vista Sagital de la CT
corteSagitalFolder.add( visibilityControlXVV, "visible" ).name( "Visible" ).onChange(
  function () {
    sliceXVV.mesh.visible = visibilityControlXVV.visible;
    rendererVV.render( sceneVV, cameraVV );
  }
);

// Se aplica lo mismo, pero para el caso de la vista coronal y transversal
. . .

// Control de la escala de umbral inferior (Threshold)
umbralFolder.add( volume, "lowerThreshold", volume.min, volume.max/2, 0.0001 )
.name( "Inferior" ).onChange(
  function () {
    volume.repaintAllSlices();
  }
);

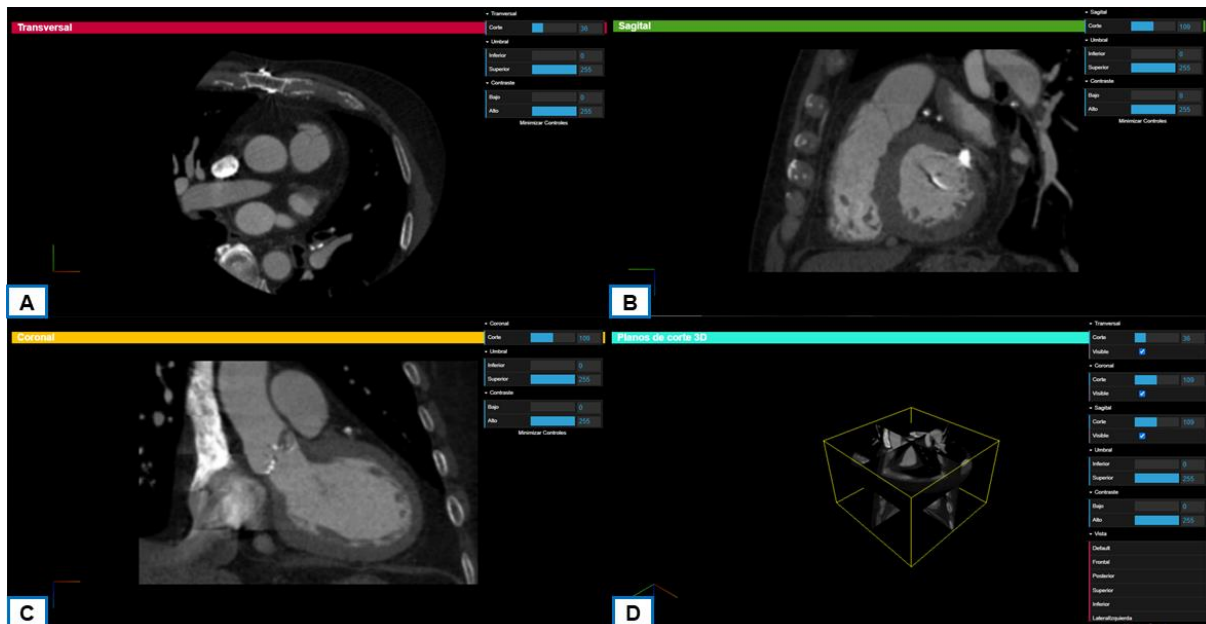
// Control de la escala de umbral superior (Threshold)
umbralFolder.add( volume, "upperThreshold", volume.max/2, volume.max, 0.0001 )
.name( "Superior" ).onChange(
  function () {
    volume.repaintAllSlices();
  }
);

// Control de la escala de contraste bajo (Window Low)
contrasteFolder.add( volume, "windowLow", volume.min, volume.max, 0.0001 )
.name( "Bajo" ).onChange(
  function () {
    volume.repaintAllSlices();
  }
);

// Control de la escala de contraste alto (Window High)
contrasteFolder.add( volume, "windowHigh", volume.min, volume.max, 0.0001 )
.name( "Alto" ).onChange(
  function () {
    volume.repaintAllSlices();
  }
);
} );

```

---



**Figura 38.** Vistas de las imágenes TC del caso de corazón en el sistema de visualización web. A. Vista transversal. B. Vista sagital. C. Vista coronal. D. Vista 3D.

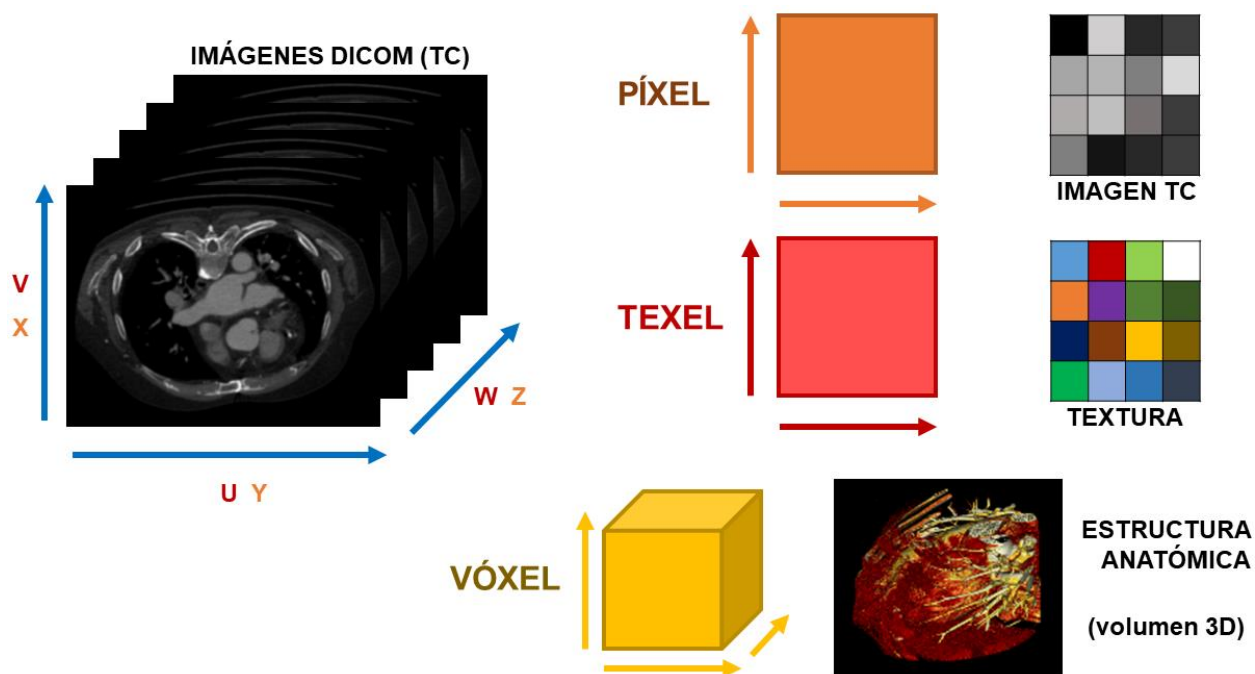
### 4.3.2 Modelos 3D en Tiempo Real

Para poder generar un modelo 3D en tiempo real en un navegador, se requiere, por un lado, la información correspondiente a las imágenes TC, como se describió en el apartado anterior de conversión, y por otro lado, generar un conjunto de texturas, tal que en conjunto con las imágenes TC se pueda configurar un script para cargar, visualizar y manipular estas, a partir de la implementación que ofrece Three.js de *volume rendering*, donde se maneja un algoritmo de *ray casting*, el cual considera dos formas de renderizado: isosuperficie y proyección de máxima intensidad, conceptos explicados en el capítulo 3; por tanto, a continuación se desglosa de manera extensa el procedimiento que se siguió:

#### i. Generación de Texturas

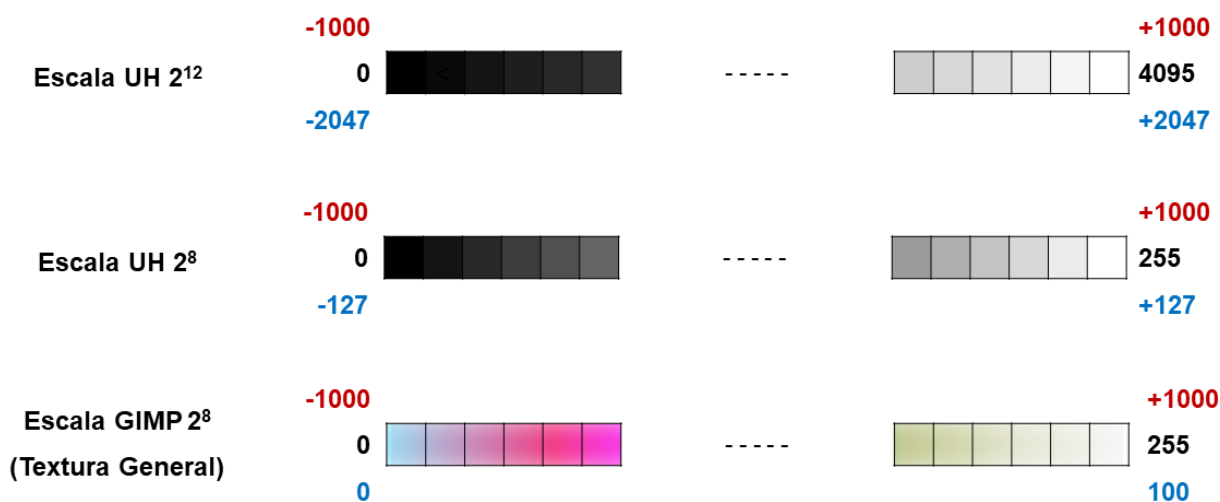
El propósito de generar una textura en el presente desarrollo contempla dotar al volumen renderizado de un determinado color para resaltar ciertas particularidades de la anatomía, al emplear una textura como un mapa de colores que permita dibujar la estructura anatómica en la escena del navegador. Por tanto, se parte de la relación del valor de la escala de Unidades Hounsfield asignado a cada píxel, el cual es la unidad mínima direccionable en pantalla, que en conjunto conforman una imagen de TC, que a su vez conforman un volumen en tres dimensiones que representa una estructura anatómica, en donde la unidad más pequeña que se visualiza en el

espacio 3D es un vóxel. Por lo que, a la superficie de una imagen, se le puede aplicar una textura, tal que cada píxel pueda ser dibujado a partir de un determinado texel, unidad mínima de una textura (Figura 39. Relación imágenes TC, píxel, vóxel y texel: donde x, y, z representan las).



**Figura 39.** Relación imágenes TC, píxel, vóxel y texel: donde x, y, z representan las coordenadas ortogonales de un objeto 3D, y v, u, w representan las coordenadas de mapeado de una textura.

Por tanto, al considerar que cada valor en la escala de UH representa un determinado tejido del cuerpo humano, se buscó representar la misma escala a partir de una textura, para así dibujar la anatomía no solo en una gama de escala de grises, sino que en una de colores. Para ello se partió del rango de valores teóricos de -1000 a +1000 de la escala de UH representados en un número de 12 bits ( $2^{12} = 4096$ ), el cual nos da una escala de -2047 a +2047 dado este número y el rango en la escala, de tal forma que se obtuvo su equivalente a un número de 8 bits ( $2^8 = 256$ ), puesto que las imágenes TC se modificaron al convertirlas a NRRD; para generar la textura se empleó GIMP, tal que se creó un lienzo con una resolución de 1 x 256 píxeles, en donde cabe destacar que para ubicarnos en cada píxel se maneja una escala de 0 a 100, por lo que en la Figura 40 se muestra la relación entre las escalas mencionadas.



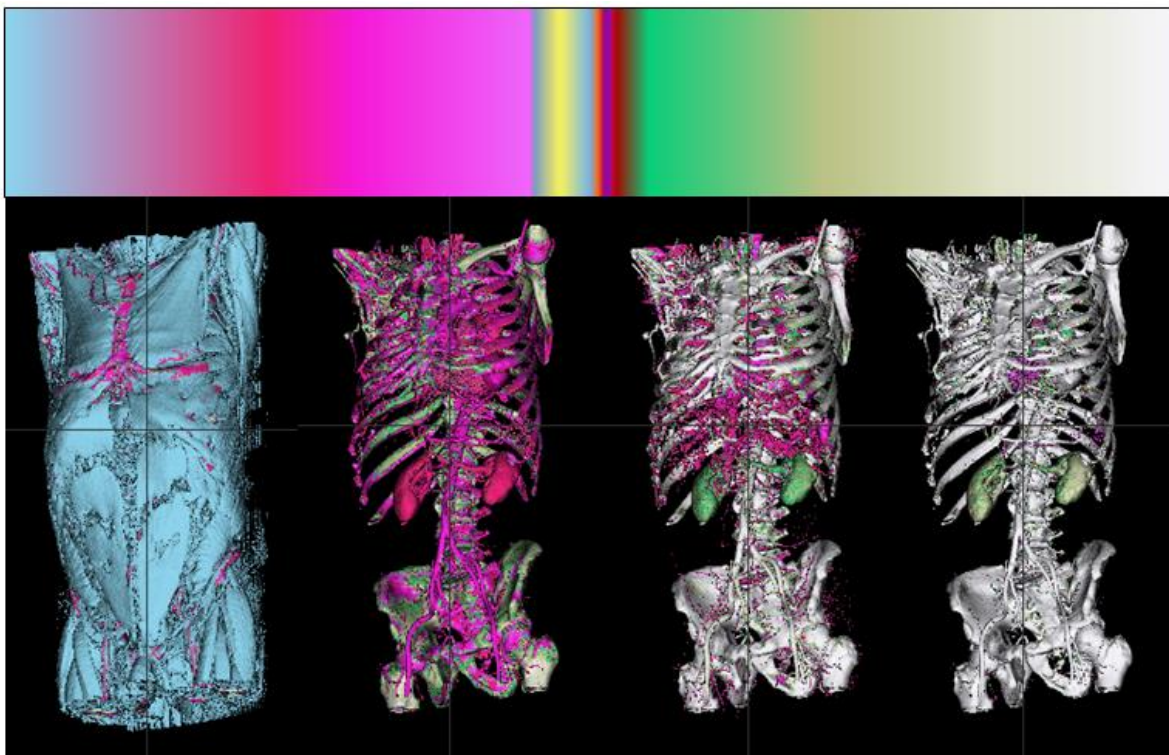
**Figura 40.** Relación de escalas empleadas para generar una textura.

De esta manera, al investigar en diversas fuentes los valores en la escala de UH para cada tejido, se determinó considerar un rango mínimo y máximo de cada uno de ellos, ya que los diversos valores encontrados se deben a que la escala no es absoluta al depender de las condiciones ambientales, el tomógrafo o instrumento empleado, así como el mismo espécimen de donde se obtienen las imágenes, y por ende los datos. Por consiguiente, en la Tabla 13. Equivalencia de los valores de los tejidos del cuerpo humano dada diversas escalas, se presentan los valores considerados y sus respectivos equivalentes dadas las escalas anteriores.

**Tabla 13.** Equivalencia de los valores de los tejidos del cuerpo humano dada diversas escalas.

Tejido	Escala UH 2 <sup>12</sup>		Escala UH 2 <sup>12</sup>		Escala UH 2 <sup>8</sup>		Escala GIMP 2 <sup>8</sup>	
	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
<b>Aire</b>	<b>-1000</b>	<b>-1000</b>	<b>-2047</b>	<b>-2047</b>	<b>-127</b>	<b>-127</b>	<b>0</b>	<b>0</b>
<b>Pulmones</b>	-400	-550	-818.8	-1125.85	-50.8	-69.85	30	22.5
<b>Tejido adiposo</b>	-50	-100	-102.35	-204.7	-6.35	-12.7	47.5	45
<b>Grasas</b>	-50	-100	-102.35	-204.7	-6.35	-12.7	47.5	45
<b>Agua</b>	0	0	0	0	0	0	50	50
<b>Musculo</b>	10	40	20.47	81.88	1.27	5.08	50.50	52
<b>Líquido cefalorraquídeo</b>	15	15	30.705	30.705	1.905	1.905	50.75	50.75
<b>Materia blanca</b>	20	36	40.94	73.692	2.54	4.572	51	51.5
<b>Riñones</b>	20	35	40.94	71.645	2.54	4.445	51	51.75
<b>Páncreas</b>	20	40	40.94	81.88	2.54	5.08	51	52
<b>Hígado</b>	35	60	71.645	122.82	4.445	7.62	51.5	53
<b>Bazo</b>	30	60	61.41	122.82	3.81	7.62	51.5	53
<b>Materia gris</b>	36	43	73.692	88.021	4.572	5.461	51.75	52.15
<b>Corazón</b>	40	50	81.88	102.35	5.08	6.35	52	52.5
<b>Sangre</b>	43	60	88.021	122.82	5.461	7.62	52.15	53
<b>Medios de contraste</b>	100	300	204.7	614.1	12.7	38.1	55	65
<b>Hueso</b>	400	1000	818.8	2047	50.8	127	70	100
<b>Hueso trabecular</b>	700	700	1432.9	1432.9	88.9	88.9	85	85
<b>Hueso cortical</b>	<b>1000</b>	<b>1000</b>	<b>2047</b>	<b>2047</b>	<b>127</b>	<b>127</b>	<b>100</b>	<b>100</b>

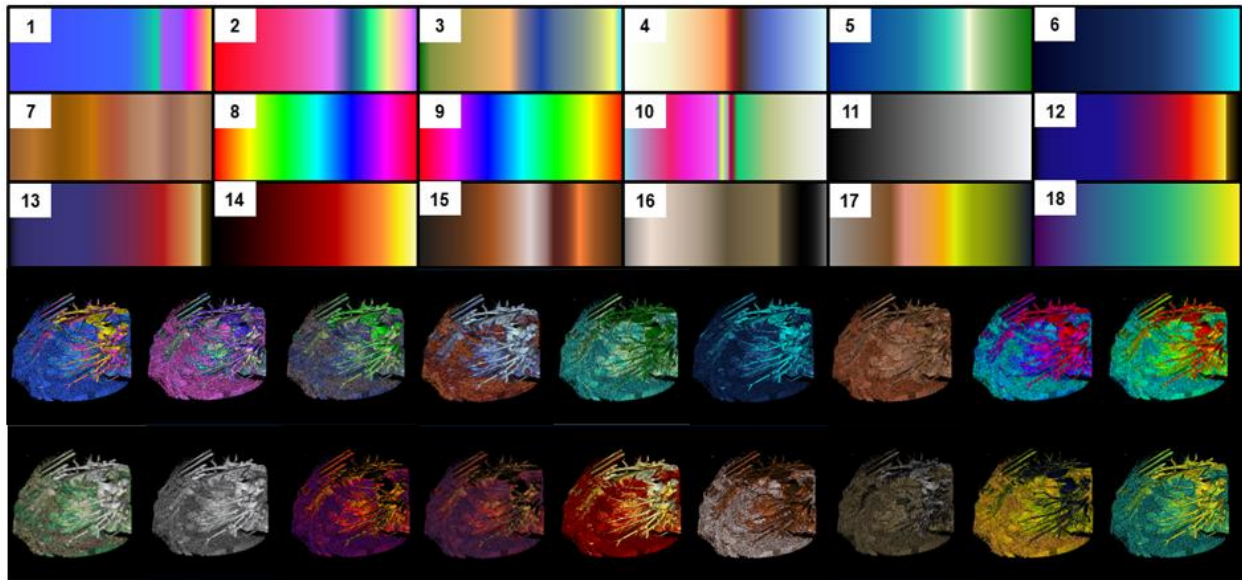
Con base en los valores obtenidos para la escala en GIMP y seleccionar un determinado color para cada tejido, se procedió a pintar la textura, a la cual se denominó "General", resultado que se muestra en la Figura 41. Textura "General" creada con base a los valores para tejidos del cuerpo humano, en donde es posible visualizar en la parte superior la textura final generada, y en la parte inferior la aplicación de la misma en el volumen renderizado del caso del tórax, tal que se puede distinguir, por ejemplo, aire en color azul, tejido adiposo y grasa en color rosa, riñones en color rosa u amarillo, y hueso en color blanco.



**Figura 41.** Textura "General" creada con base a los valores para tejidos del cuerpo humano en la escala de UH: representación en caso de tórax renderizado.

Si bien, la anterior textura permite acentuar diversas particularidades en la estructura de la anatomía del cuerpo humano con un color específico como comúnmente son identificadas, se determinó generar otras texturas con base a escalas de colores básicas empleadas por herramientas de software de edición, como GIMP, por tanto se crearon otras 17 texturas, con el fin de dotar de varias opciones de color posibles para dibujar en la escena del navegador una determinada estructura, las cuales se muestran en la Figura 42, en conjunto con su representación visual para el caso del corazón.





**Figura 42.** Texturas generadas en GIMP para la visualización de mapa de colores con su respectiva representación de volumen renderizado en el caso de corazón. Identificación de texturas: 1. Abstracto1, 2. Abstracto2, 3. Abstracto3, 4. Amanecer, 5. Azules1, 6. Azules2, 7. Cafés, 8. Espectro1, 9. Espectro2, 10. General, 11. Grises, 12. Horizonte1, 13. Horizonte2, 14. Incandescente, 15. Sombra1, 16. Sombra2, 17. Sombra3, y 18. Viridis.

## ii. Carga de Texturas

Para poder cargar las texturas anteriores como parte de la visualización de mapa de colores en el sistema de visualización, se parte de la creación de una instancia que permita cargar las imágenes TC en formato NRRD, para su posterior carga, como se especificó en el caso del recurso de imágenes TC; después, se procedió a crear una instancia de una textura, la cual se aplicará a la superficie que se renderizará, de tal forma que toma las dimensiones y datos del caso cargado, luego, se le asigna un formato para tratar los datos desde el canal rojo al ser escalares, un tipo de dato byte sin signo con base a las especificaciones de los metadatos del archivo NRRD, a continuación, se especifica el tipo de filtro para configurar como será mostrada la textura cuando un texel cubra más o menos un píxel, así como los requisitos de alineación para el inicio de cada fila de pixeles en la memoria.

Por otro lado, se crea una variable, la cual contendrá cada una de las texturas, tal que se especifica la ubicación para que la textura pueda ser cargada, así como la función que se empleará para ello, en este caso, será la de *render*, en donde se le especifica al renderizador que muestre los respectivos cambios afectando la escena y la cámara. Lo descrito anteriormente, se muestra implementado en la siguiente tabla, en donde se puede destacar que los nombres con los cuáles



se hace referencia a cada textura, está relacionado al nombre especificado en otra variable que permite implementar y manejar la GUI y los controles respectivos.

**Tabla 14.** Carga de texturas en la vista mapa de colores con base a la implementación en Three.js

Definición
<pre>// Se crea una instancia del cargador NRRD var loader = new NRRDLoader();  // Se cargan las imágenes TC en formato NRRD loader.load( "../modelos/nrrd/pruebaCorazon1-8B-256P-F.nrrd", function ( volume ) {   // Se crea una textura para mantener el volumen (visualización de un volumen)   const texture =     new THREE.DataTexture3D( volume.data, volume.xLength, volume.yLength, volume.zLength );    // Al tener escalares en el caso cargado se tratan los datos desde el canal rojo   texture.format = THREE.RedFormat;    // Con base en las especificaciones en los metadatos del caso se configura la textura   texture.type = THREE.UnsignedByteType;   texture.minFilter = texture.magFilter = THREE.LinearFilter;   texture.unpackAlignment = 1;    // Se especifican las texturas que se emplearan para la propiedad de colorMap   cmtextures = {     abstracto1: new THREE.TextureLoader().load( '../paletas/Abstracto1.png', render ),     abstracto2: new THREE.TextureLoader().load( '../paletas/Abstracto2.png', render ),     abstracto3: new THREE.TextureLoader().load( '../paletas/Abstracto3.png', render ),     amanecer: new THREE.TextureLoader().load( '../paletas/Amanecer.png', render ),     ...     // Así sucesivamente con las demás texturas que se generaron   };    // Configuraciones para renderizar volumen   ... });</pre>

### iii. Configuración de Materiales, Malla y Rendering

Para poder renderizar un volumen en la escena a través del navegador web, se emplea la implementación que ofrece la biblioteca Three.js para generar un dibujo tridimensional en la escena a partir de la implementación de una técnica de *volume render* a nivel del sombreador de fragmentos (*fragment shader*), el cual pone en práctica en un algoritmo la aplicación de *ray casting*, isosuperficies y MIP (mapeo de color); dicho algoritmo se muestra en el Anexo C. Las configuraciones realizadas en el programa de la visualización de mapa de colores, abstraen las configuraciones a bajo nivel, por lo que solo es necesario, importar el módulo para emplear un sombreador de volumen, para posteriormente especificar el *shader* y las variables globales para manejarlo (*uniforms*), tal que permitirá generar un material para poder crear una malla en conjunto con una geometría con base a los datos que representan las imágenes TC cargadas, con el fin de

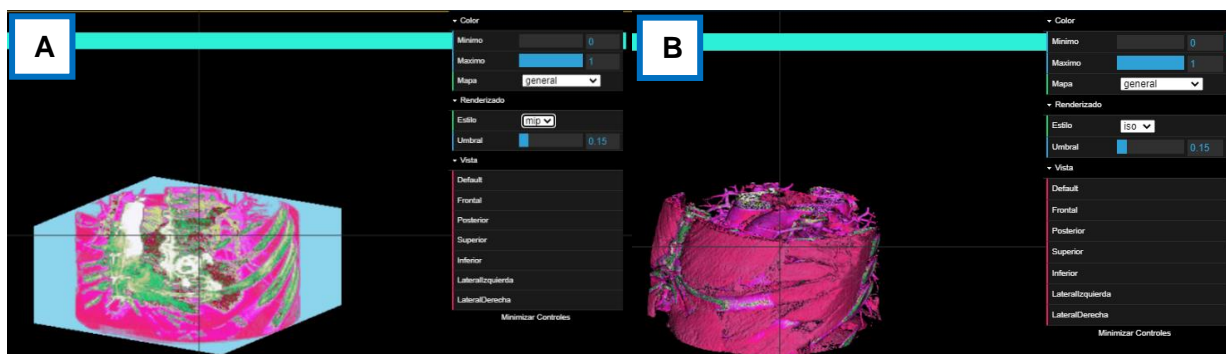
que pueda ser agregada a la escena y se renderice para su visualización, como se muestra en la Tabla 15.

**Tabla 15.** Configuración del material, malla y rendering en Three.js para la vista de mapa de colores.

Definición
<pre>// Se importa el módulo de SHADERS import { VolumeRenderShader1 } from '../.../three.js-master/examples/jsm/shaders/VolumeShader.js';  ...  // Se crea una instancia del cargador NRRD var loader = new NRRDLoader();  // Se cargan las imágenes TC en formato NRRD loader.load( "../modelos/nrrd/pruebaCorazon1-8B-256P-F.nrrd", function ( volume ) {   // Configuraciones textura   ...    // Material   // Se crea la variable tipo shader   const shader = VolumeRenderShader1;    // Se crea una variable para manejar el shader   const uniforms = THREE.UniformsUtils.clone( shader.uniforms );    // Se configuran las variables globales del shader con base a las propiedades   especificadas y configuradas en la GUI   uniforms[ "u_data" ].value = texture;   uniforms[ "u_size" ].value.set( volume.xLength, volume.yLength, volume.zLength );   uniforms[ "u_clim" ].value.set( volconfig.Minimo, volconfig.Maximo );   // 0: MIP, 1: ISO   uniforms[ "u_renderstyle" ].value = volconfig.Estilo == 'mip' ? 0 : 1;   // Para el estilo de renderizado ISO   uniforms[ "u_renderthreshold" ].value = volconfig.Umbral;   uniforms[ "u_cmdata" ].value = cmtextures[ volconfig.Mapa ];    // Se crea y configura una instancia que se empleará como material   material = new THREE.ShaderMaterial( {     uniforms: uniforms,     vertexShader: shader.vertexShader,     fragmentShader: shader.fragmentShader,     // El sombreador de volumen usa el reverso como su punto de referencia     side: THREE.BackSide   } );    // Mesh   // Se crea una geometría con base a los datos del caso cargado   const geometry =     new THREE.BoxBufferGeometry( volume.xLength, volume.yLength, volume.zLength );   // Se aplica una transformación para su correcta visualización en la escena   geometry.translate( volume.xLength / 2, volume.yLength / 2, volume.zLength / 2 );   // Se crea una instancia de una malla empleando la geometría y material   const mesh = new THREE.Mesh( geometry, material );   // Se agrega la malla a la escena   scene.add( mesh );   // Se realiza la respectiva renderización   render(); } );</pre>

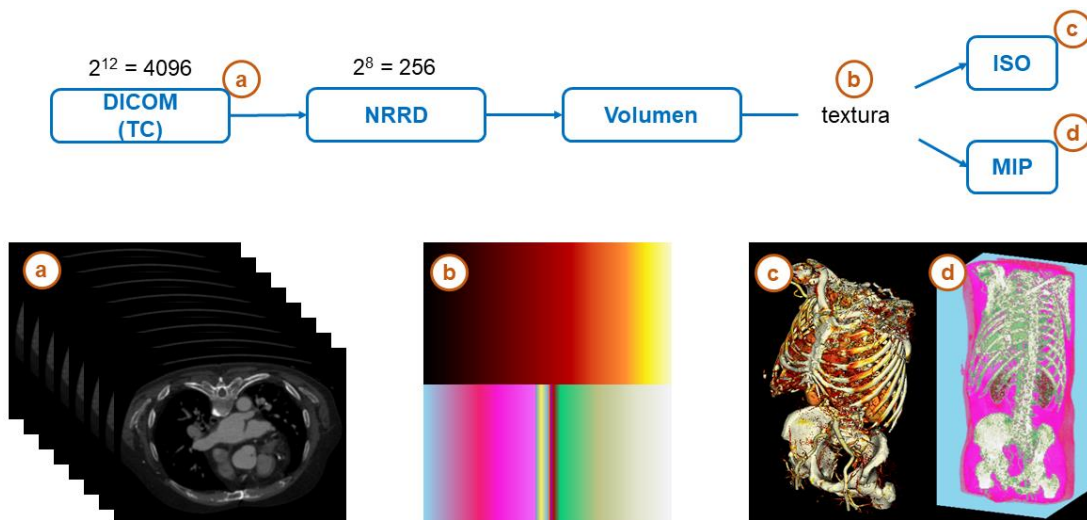
Es importante mencionar que la visualización de mapa de colores (Figura 43) proporciona de igual forma manipulación e interacción a través de la GUI y controles, tal que se definió: el

control de color, a partir de las opciones “Mínimo” y “Máximo”, lo que permitirá posicionarnos en un rango del espectro de la textura, es decir, se podrá considerar todos los colores definidos en ella o solo una sección conforme a la implementación para obtener el color en el *shader*, el control de mapa, cuyo funcionamiento es poder cambiar entre las diversas texturas generadas y proporcionadas en esta vista; el control de estilo de renderizado, el cual permitirá escoger entre la opción ISO o MIP, y en caso de seleccionar la primera se podrá hacer uso del control de “Umbral”, con el objetivo de manejar que tanta densidad de la estructura anatómica se quiere visualizar; así como el control de vistas *default*, de tal manera que se pueda colocar en una determinada posición el modelo 3D sin la necesidad de trasladar o rotar manualmente.



**Figura 43.** Vista del mapa de colores del modelo 3D generado en tiempo real del caso de corazón. A. Estilo MIP. B. Estilo ISO.

De esta manera, una vez explicado lo anterior en relación con el recurso de modelo 3D en tiempo real, en la Figura 44 se muestra una abstracción del proceso empleado y desglosado en el sistema de visualización web de imágenes médicas, para proporcionar este recurso.



**Figura 44.** Proceso general para generar un modelo 3D en tiempo real en el sistema de visualización web.

### 4.3.3 Modelos 3D reconstruidos

Para poder renderizar un modelo 3D reconstruido en un navegador, se partió del proceso implementando para generar la vista 3D con imágenes TC, descrito en las secciones anteriores, con el fin de proporcionar la tomografía computarizada de la anatomía en conjunto con su respectiva reconstrucción tridimensional, ya que en comparación con el modelo volumétrico generado en tiempo real, la definición particular de cada estructura anatómica es más ideal, definida y modular al realizar una segmentación previa a detalle, lo que permite empalmar ambos recursos para su posterior uso, según sea lo más conveniente.

Por tanto, para poder generar esta visualización, se requirió, por un lado, obtener un modelo 3D reconstruido a partir de imágenes TC, por lo que se procedió a seguir un proceso de segmentación, tanto semiautomático como manual, con el software de *3DSlicer* para generar un modelo 3D en formato OBJ, para su posterior procesamiento de transformación y refinamiento en *Blender* y *MeshLab*, procedimiento abordado en el capítulo anterior; por otro, cargar la reconstrucción generada al navegador con Three.js, en donde cabe destacar, que a pesar de que esta herramienta cuenta con un cargador OBJ, se trabajó con archivos VTK, cargador que también es proporcionado por Three.js, al ser un recurso más ligero para el tipo de implementación que se busca con el sistema de visualización. De esta manera, a continuación, se desglosa de manera extensa este último punto:

#### i. Carga de modelos 3D reconstruidos a partir de imágenes CT

Una vez que se tienen los modelos transformados y refinados de la reconstrucción de las diversas estructuras anatómicas de interés en formato OBJ, se procede a realizar la transformación de cada uno de ellos al formato VTK a través de un programa desarrollado en *Python*, como se muestra en la **Tabla 16**. Es importante mencionar que se empleó la distribución de *Anconda3 – Spyder 4.0.1*, así como la versión 4.2 del módulo VTK, pues versiones posteriores, en cuanto al formato VTK, generan archivos que no son compatibles con el cargador VTK en Three.js. En la **Tabla 17**, se muestra una comparativa de los pesos de los archivos generados en el proceso de obtención y conversión de las respectivas reconstrucciones, con el fin de ratificar la relevancia del formato VTK como un recurso solicitado a través de un navegador y renderizado en una escena en el sistema de visualización web de imágenes médicas.

**Tabla 16.** Código en Python para convertir un archivo OBJ a VTK.

Definición
<pre> # -*- coding: utf-8 -*- ## Se importa la biblioteca VTK import vtk  ## Se especifica la ruta del archivo OBJ fileOBJ = 'nombreArchivo.obj' ## Se especifica la ruta para el archivo VTK que se generará fileVTK = 'nombreArchivo.vtk'  ## Se crea una instancia para leer archivos OBJ reader = vtk.vtkOBJReader() ## Se especifica la ruta del archivo OBJ reader.SetFileName(fileOBJ) ## Se actualiza la instancia reader.Update() ## Se obtienen los datos poligonales del archivo OBJ obj = reader.GetOutput()  ## Se crea una instancia de un objeto que escribe archivos de datos poligonales a formato VTK writer = vtk.vtkPolyDataWriter() ## Se especifica la ruta del archivo VTK writer.SetFileName(fileVTK)  ## Se genera el archivo VTK a partir de los datos del archivo OBJ if vtk.VTK_MAJOR_VERSION &lt;= 5:     writer.SetInput(obj) else:     writer.SetInputData(obj)  ## Se escriben los datos en el archivo writer.Write() </pre>

**Tabla 17.** Comparativa de pesos de archivo en el proceso de adquisición, transformación, refinamiento y conversión de reconstrucciones anatómicas del corazón y tórax en formato OBJ y VTK

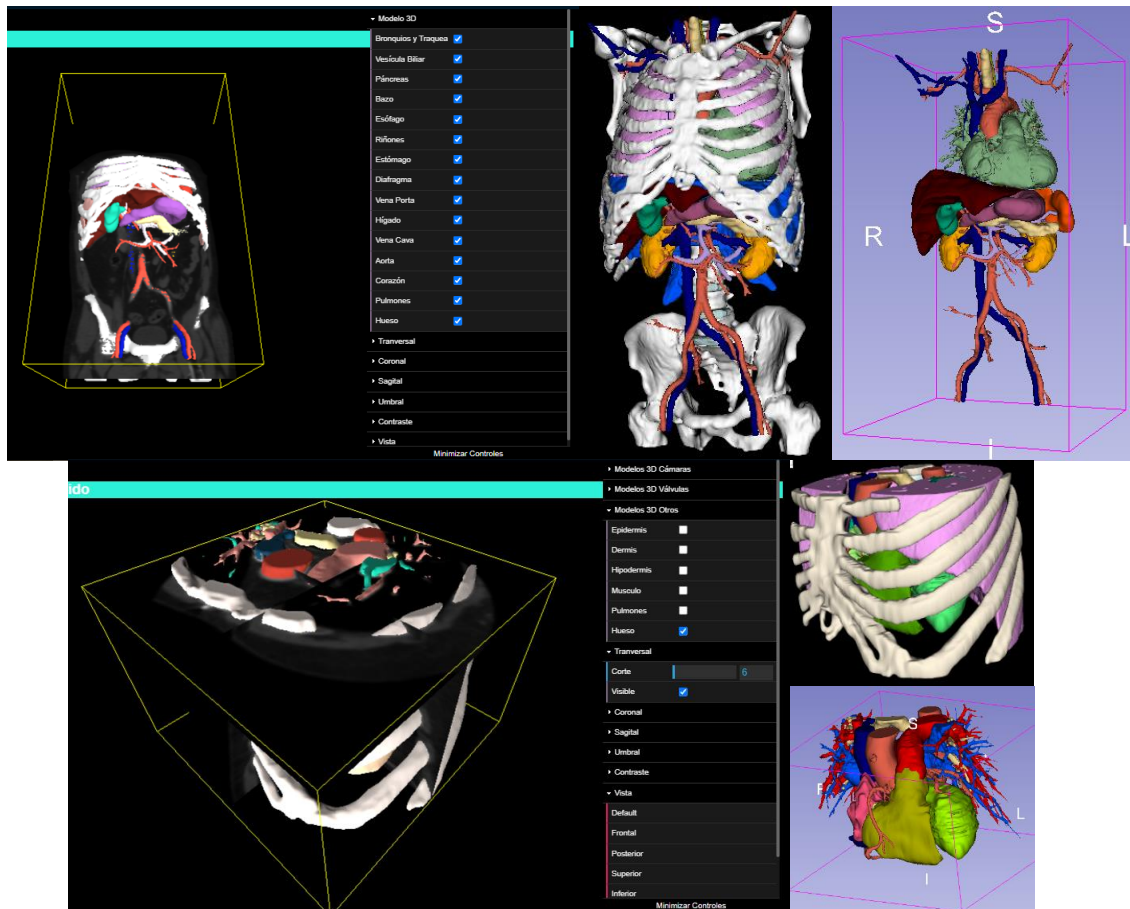
Caso	3DSlicer - OBJ	Blender - OBJ	MeshLab - OBJ	VTK
Corazón	573 MB	545 MB	64.4 MB	44.4 MB
Tórax	615 MB	595 MB	28.6 MB	16.5 MB

Una vez que se han obtenido las diversas estructuras anatómicas en formato VTK, se procede a implementar las configuraciones necesarias para generar la visualización del modelo 3D, en donde se abstraen las configuraciones de bajo nivel, por lo que solo es necesario importar el módulo para cargar archivos en formato VTK, crear un material característico para cada estructura reconstruida, cargar y especificar los datos geométricos contenidos en el archivo VTK, crear una malla y añadirla en la escena, tal y como se especifica en la Tabla 18. Es importante mencionar que la visualización del modelo (Figura 45), proporciona de igual forma la misma

manipulación e interacción a través de la GUI y controles, especificados en el recurso de imágenes TC, sin embargo, en esta también se definió la funcionalidad de visibilidad de cada modelo 3D reconstruido que conforman cada caso, de la forma que se pueda controlar que estructura se visualiza en todo momento en la escena del navegador, y se tenga un mejor control dado el uso que se requiera implementar.

**Tabla 18.** Implementación de Three.js para cargar la estructura de hueso en formato VTK en la visualización de modelo del caso corazón.

Definición
<pre> // Se importa el cargador VTK import { VTKLoader }   from '../.../three.js-master/examples/jsm/loaders/VTKLoader.js';  ...  // Se crea un material para la respectiva anatomía const vtkmaterialHueso = new THREE.MeshLambertMaterial( { wireframe: false, morphTargets: false, side: THREE.DoubleSide, color: 0xFBEEEE6 } );  // Se crea una instancia del cargador VTK const vtkloaderHueso = new VTKLoader();  // Se carga la reconstrucción de la respectiva anatomía y se configura vtkloaderHueso.load( "../modelos/vtk/corazon/vm_huesos_v.vtk", function ( geometry ) {    // Se configura la geometria   geometry.computeVertexNormals();   geometry.translate(-29.5, -12.5, 0);    // Se crea una malla con base al material y datos geométricos de la respectiva anatomía   const mesh = new THREE.Mesh( geometry, vtkmaterialHueso );    // Se agrega la malla a la escena   sceneVV.add( mesh );    // Se configura la propiedad de visibilidad   const visibilityControl = {     visible: true   };    // Se configura el GUI para controlar la visibilidad de la respectiva geometría   modeloFolderO.add( visibilityControl, "visible" ).name( "Hueso" ).onChange(     function () {       mesh.visible = visibilityControl.visible;       rendererVV.render( sceneVV, cameraVV );     }   );    // Se realiza lo mismo para cada una de las estructuras anatómicas del respectivo caso   ...  } ); </pre>



**Figura 45.** Vista modelo en el sistema de visualización web de imágenes médicas del caso de corazón y tórax: a la izquierda se muestra la visualización en el sistema del modelo 3D reconstruido y las imágenes TC en los diferentes cortes, mientras que a la derecha la visualización detallada de la reconstrucción generada a partir de imágenes TC.

#### 4.4 Implementación de la Visualización Mixta

El sistema de visualización desarrollado y especificado hasta este momento, cuenta con diversas visualizaciones y funcionalidades interactivas desde un navegador web a través de diversos dispositivos, no obstante, al haber empleado Three.js para diseñar y crear lo anterior, se tiene la posibilidad de habilitar la experiencia de la tecnología *WebXR*, es decir, la experiencia de realidad virtual y aumentada a través de la Web, de tal manera que los recursos visuales implementados en el sistema sean más convenientes y ampliamente accesibles desde otra perspectiva. Para ello, se realizaron las configuraciones necesarias en el flujo de los scripts de cada una de las visualizaciones para proporcionar el recurso de realidad virtual a través del navegador web, como se especifica a continuación:



- Importar la biblioteca de VR al proyecto.

```
import {VRButton} from 'three/examples/jsm/webxr/VRButton.js';
```

- Crear un botón para indicar al usuario la compatibilidad con la visualización VR, a través del cual se inicia una sesión de realidad virtual, si este es activado.

```
document.body.appendChild( VRButton.createButton( renderer ) );
```

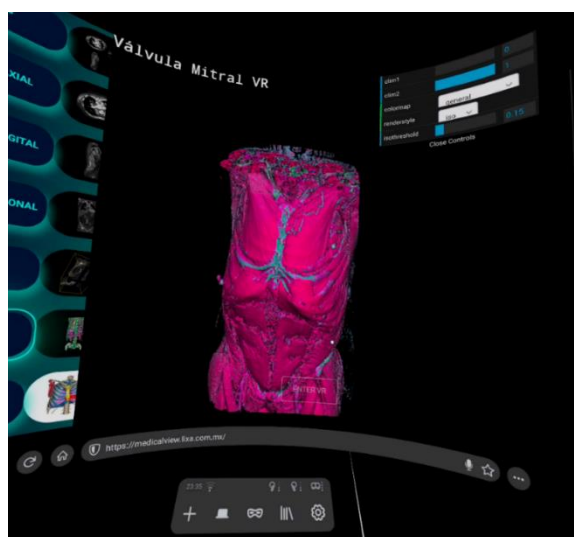
- Indicar a la instancia de WebGLRenderer que habilite el renderizado XR.

```
renderer.xr.enabled = true;
```

- Ajustar el ciclo de animación.

```
renderer.setAnimationLoop( function(){ renderer.render( scene, camera ); } );
```

Lo anterior permite habilitar la visualización de las imágenes TC, modelo 3D en tiempo real y modelo 3D reconstruido a través de realidad virtual, sin embargo, se requiere configurar los controles y otro tipo de funcionalidades, no abordadas en el presente desarrollo, para que la interactividad propuesta en el sistema de visualización sea más intuitiva y natural con los dispositivos para experiencias de realidad virtual; no obstante, la implementación realizada, se generó con el objetivo de probar el concepto y funcionalidad de esta tecnología por medio de unos lentes de realidad virtual como los *Oculus Quest 2* a través del navegador web del casco como se muestra en la Figura 46.



**Figura 46.** Visualización con realidad virtual en el explorador web del Oculus Quest 2 del caso de tórax en la vista de mapa de colores.



---

# Capítulo 5

## Pruebas

---

### 5.1 Objetivo del estudio

---

El objetivo del estudio fue evaluar la experiencia de uso del sistema de visualización web de imágenes médicas mediante una encuesta cualitativa basada en la escala *NASA-Task Load Index* (NASA-TLX), la cual es un instrumento estandarizado de evaluación de la carga mental de trabajo desarrollado por Hart y Steveland (1988), con la finalidad de observar el comportamiento y percepción de la aplicación web, así como la retroalimentación ante posibles mejoras o implementaciones en futuras nuevas versiones del sistema de visualización que se busquen o logren llevar a cabo para poder establecer escenarios adecuados como un medio de aprendizaje, diagnóstico, planeación y asistencia en el área médica.

### 5.2 Participantes

---

La evaluación se realizó con un grupo piloto de 85 alumnos del primer año de la carrera de médico cirujano de la Facultad de Medicina de la Universidad Nacional Autónoma de México (UNAM), con recién y/o poco conocimiento en el estudio anatómico.

### 5.3 Procedimiento

---

El plan de pruebas se programó en dos sesiones grupales de 30 – 40 minutos aproximadamente por cada grupo de diferentes individuos de manera virtual y remota por medio de la herramienta Zoom, tal que la primera sesión estuvo conformada por 42 participantes, y la segunda por 43. De esta manera, la sesión se dividió en tres partes: 1) se proporcionó un enlace al sistema de visualización web para que los participantes exploraran y se familiarizaran con la aplicación, mientras se les daba contexto y una breve explicación del contenido y funcionalidad del desarrollo; 2) con ayuda del docente a cargo del grupo, se solicitó la identificación de tres estructuras anatómicas en un intervalo de 2 – 3 minutos en un determinado caso y en los principales recursos

de visualización (imagen TC, *volume rendering* y reconstrucción), en donde se permitió en cada caso a algún participante compartir su pantalla con el fin de que el docente corroborará la respuesta con todo el grupo; finalmente, 3) se aplicó una encuesta a través de un formulario anexo en la página en donde se presentó el sistema de visualización web.

## 5.4 Definición de métricas

---

Para evaluar la experiencia de uso del sistema de visualización desarrollado se empleó la escala NASA-TLX, la cual es utilizada y validada en numerosas publicaciones internacionales (Martins, et al., 2015; Harris, et al., 2020; Wilson, et al., 2011; Charlton, 2002; Marras & Karwowski, 2006; Rutledge, et al., 2009) al ser una valoración de auto reconocimiento y clasificación del nivel de destreza del participante para las principales demandas psicomotoras: *esfuerzo* (grado de esfuerzo mental y físico que tiene que realizar el sujeto para obtener su nivel de rendimiento), *demanda mental* (cantidad de actividad mental y perceptiva que requiere la tarea), *demanda física* (cantidad de actividad física que requiere la tarea), *demanda temporal* (nivel de presión temporal sentida entre el requerido y el disponible), *rendimiento* (satisfacción con el nivel de rendimiento), y *nivel de frustración* (inseguridad, estrés, irritación, entre otros, durante la realización de una tarea).

De esta manera, se diseñó una encuesta cualitativa dividida en tres secciones (Anexo D): uno, información y experiencia previa del participante, en donde se incorporan preguntas relacionadas a la experiencia previa del participante en cuanto al sistema propuesto; dos, NASA-TLX, tal que se abordan diferentes cuestiones para evaluar cada demanda psicomotora del participante; y tres, nivel de satisfacción, la cual busca agregar preguntas subjetivas sobre la experiencia de aprendizaje a través del uso del sistema de visualización web de imágenes médicas. Es importante señalar, que la escala NASA-TLX propone una escala con valores del 0 al 100 en intervalos de 10 como forma directa de representar un porcentaje en las respuestas. Por cuestiones prácticas se establecieron los valores en una escala de 1 a 10 en intervalos unitarios equivalentes a la original.

---

# Capítulo 6

## Resultados

---

Para asegurar la valoración completa de los datos recolectados de la experiencia de uso del sistema de visualización web de imágenes médicas a través de la encuesta aplicada a estudiantes de medicina, se consideró realizar un análisis estadístico y una representación visual conforme a las secciones propuestas, de tal manera que a continuación se abordan los resultados obtenidos.

El primer grupo de preguntas relacionadas a la sección de información y experiencia previa, se realizaron con el fin de obtener una primera aproximación de la noción y opinión general de las tecnologías y recursos visuales con los que se han involucrado los estudiantes de ciencias de la salud. De este modo, se tiene que los participantes se encontraban en un rango de 17 a 28 años, en donde el 64.4% tenía 18 años, tal que el 98.8% considera útiles las tecnologías de realidad virtual, lo cual indica el interés en cuanto al uso de herramientas que implementan tecnologías de realidad extendida como lo es la realidad virtual al ser estudiantes de medicina.

Por su parte, el 43.5% considera accesibles los sistemas o herramientas web para el aprendizaje médico relacionado a la visualización de imágenes TC y modelos 3D anatómicos, de este modo es evidente cierta presencia de accesibilidad a recursos visuales, pero no en su mayoría, lo que sin duda es indicador de que algunos sistemas o herramientas actuales presentan algún tipo de limitante. En lo esencial, se tiene que los estudiantes han tenido una mayor interacción con recursos como lo son imágenes en general (48.8%), mientras que recursos como lo son las imágenes DICOM (8.3%) no son comunes o el nombre no lo asocian a imágenes de MRI o TC, y los modelos 3D interactivos (42.9%) a pesar de precisar un valor cercano a las imágenes en general, es más usual encontrarse con imágenes en libros de texto o vía Web.

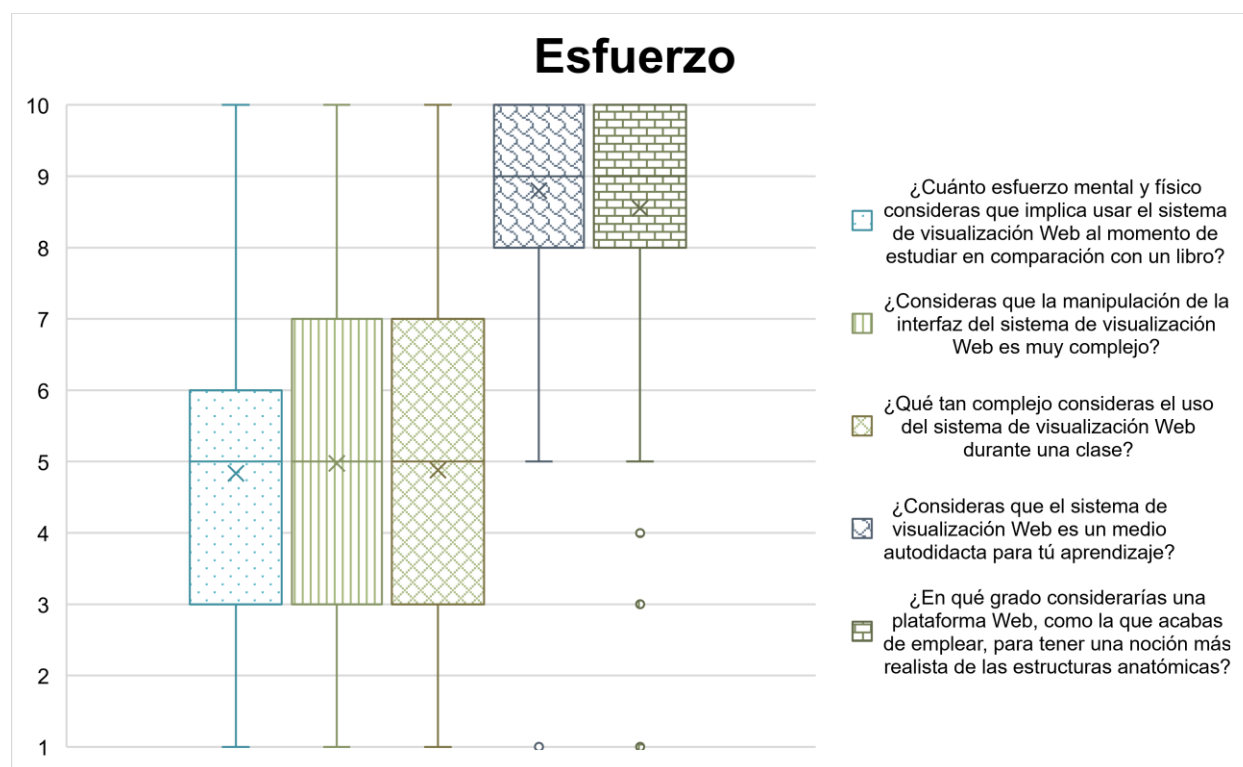
En cuanto a la siguiente sección, conformada por varios grupos de preguntas relacionadas a la escala NASA-TLX, se desglosan a continuación:

- **Esfuerzo**

El primer grupo de preguntas de esta sección fueron etiquetadas como esfuerzo (E), considerando determinar el desempeño mental a través del rendimiento del participante al emplear el sistema de visualización, de acuerdo con las preguntas de la Figura 47 y la Tabla 19, para poder determinar el impacto y complejidad de uso y manipulación del de cada uno de los recursos visuales con los

que interactuaron en el software como un medio autodidacta de aprendizaje y de noción realista de estructuras anatómicas.

En el gráfico de la Figura 47, con base a la ponderación establecida, se tiene que dada la pregunta de izquierda a derecha uno implica el peor de los casos: poco esfuerzo, totalmente desacuerdo, poco complejo, totalmente desacuerdo y poco; mientras que diez implica el mejor de los casos: mucho esfuerzo, totalmente de acuerdo, muy complejo, totalmente de acuerdo y mucho.



**Figura 47.** Gráfica de esfuerzo.

Pese a todo, de forma particular se observa que para las primera tres preguntas se tienen resultados que no son totalmente concluyentes al ser muy amplios, no obstante, la tendencia que se presenta es cercana al valor medio de la escala, lo que indica la mejora en cuanto a controles, funcionalidad y uso entre los diversos dispositivos, de tal forma que para el estudiante no sea difícil manejar el sistema. Por otro lado, las últimas dos preguntas, nos permiten observar la aceptación de considerar al sistema como una herramienta de aprendizaje autodidacta para tener una mejor noción en cuanto a anatomía.

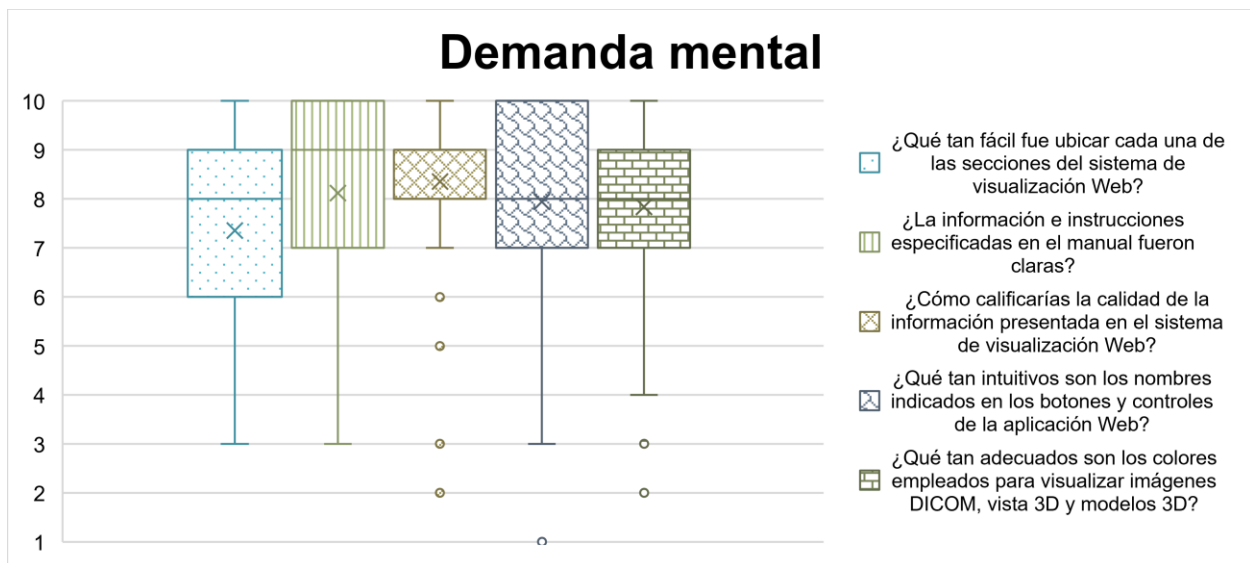
**Tabla 19.** Resultados de los descriptivos asociados a la evaluación de rendimiento en relación con el rubro de esfuerzo en la escala NASA-TLX.

Descriptivo	¿Cuánto esfuerzo mental y físico consideras que implica usar el sistema de visualización Web al momento de estudiar en comparación con un libro?	¿Consideras que la manipulación de la interfaz del sistema de visualización Web es muy complejo?	¿Qué tan complejo consideras el uso del sistema de visualización Web durante una clase?	¿Consideras que el sistema de visualización Web es un medio autodidacta para tú aprendizaje?	¿En qué grado considerarías una plataforma Web, como la que acabas de emplear, para tener una noción más realista de las estructuras anatómicas?
Media	4.83	4.97	4.88	8.80	8.55
Mediana	5.00	5.00	5.00	9.00	9.00
Moda	5.00	4.00	2.00	10.00	10.00
Desviación estándar	2.04	2.44	2.56	1.56	1.70
Varianza de la muestra	4.18	5.99	6.58	2.44	2.89
Rango	9.00	9.00	9.00	9.00	9.00
Mínimo	1.00	1.00	1.00	1.00	1.00
Máximo	10.00	10.00	10.00	10.00	10.00
Cuenta	85.00	85.00	85.00	85.00	85.00

**- Demanda mental**

El segundo grupo de preguntas de esta sección fueron etiquetadas como demanda mental (DM), teniendo en cuenta la cantidad de actividad mental y perceptiva que se requiere para hacer una tarea específica, de acuerdo con las preguntas de la Figura 48 y Tabla 20, para poder determinar qué tan intuitiva y clara resulta usar la herramienta visual desarrollada a partir de la distribución de cada uno de los recursos y la forma de interactuar con ellos con base a la información proporcionada, nombres y colores empleados.

En el gráfico de la Figura 48, con base a la ponderación propuesta, se tiene que dada la pregunta de izquierda a derecha uno implica el peor de los casos: muy difícil, nada claras, muy mala, nada intuitivos y nada adecuado; mientras que diez implica el mejor de los casos: muy fácil, muy claras, muy buena, muy intuitivos y muy adecuado.



**Figura 48.** Gráfica de demanda mental.

En particular, las primeras dos preguntas y la cuarta presentan resultados que no son totalmente concluyentes al ser muy amplios, sin embargo, la tendencia que se tiene es cercana al valor correspondiente al mejor de los casos, lo que es señal que emplear el sistema para realizar ciertas actividades es adecuado considerando mejorar ciertos aspectos en cuanto a ubicación, información y presentación de los recursos. Mientras que, en relación con la calidad visual de las imágenes y modelos fueron considerados con una buena percepción, a partir de los resultados de la tercera y última pregunta.

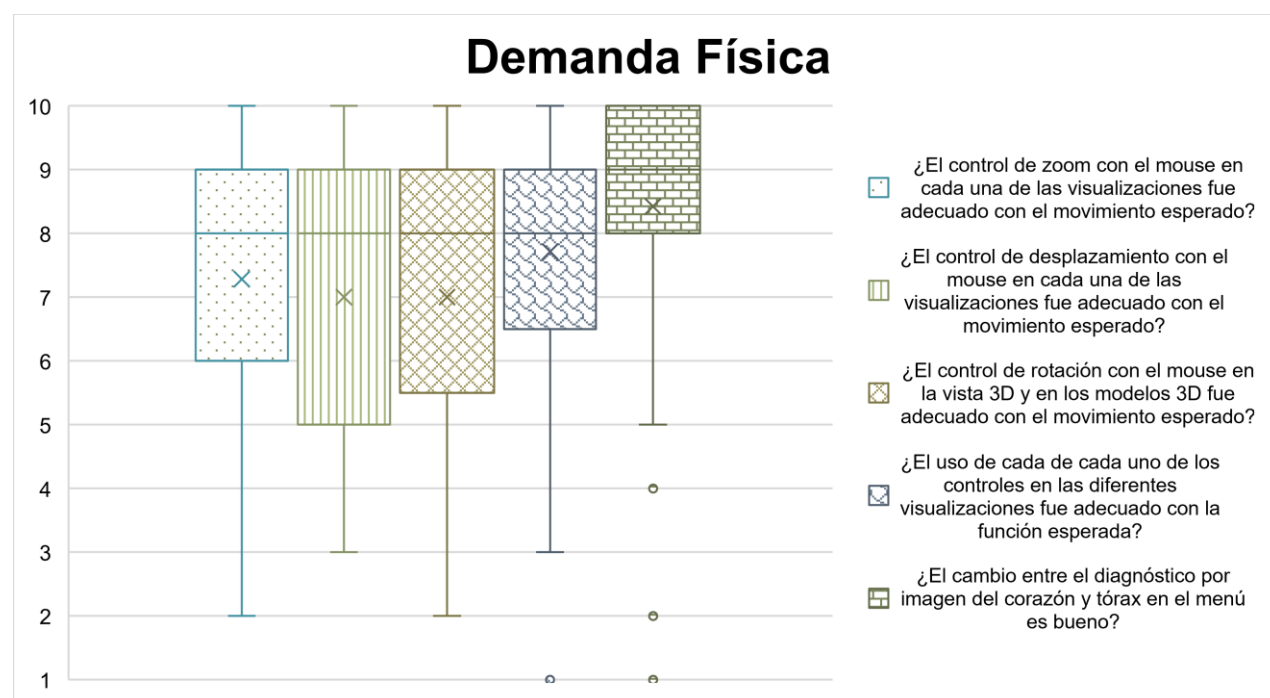
**Tabla 20.** Resultados de los descriptivos asociados a la evaluación de la actividad mental y percepción en relación con el rubro de demanda mental en la escala NASA-TLX.

Descriptivo	¿Qué tan fácil fue ubicar cada una de las secciones del sistema de visualización Web?	¿La información e instrucciones especificadas en el manual fueron claras?	¿Cómo calificarías la calidad de la información presentada en el sistema de visualización Web?	¿Qué tan intuitivos son los nombres indicados en los botones y controles de la aplicación Web?	¿Qué tan adecuados son los colores empleados para visualizar imágenes DICOM, vista 3D y modelos 3D?
Media	7.35	8.11	8.35	7.94	7.83
Mediana	8.00	9.00	9.00	8.00	8.00
Moda	8.00	10.00	9.00	10.00	9.00
Desviación estándar	1.93	1.76	1.52	1.99	1.96
Varianza de la muestra	3.75	3.10	2.32	3.96	3.87
Rango	7.00	7.00	8.00	9.00	8.00
Mínimo	3.00	3.00	2.00	1.00	2.00
Máximo	10.00	10.00	10.00	10.00	10.00
Suma	625.00	690.00	710.00	675.00	666.00
Cuenta	85.00	85.00	85.00	85.00	85.00

## - Demanda física

El tercer grupo de preguntas de esta sección fueron etiquetadas como demanda física (DF), de modo que se pudiera obtener la cantidad de actividad física que requiere hacer una tarea, de acuerdo con las preguntas de la Figura 49 y Tabla 21, para poder determinar la correcta funcionalidad de los diversos controles con los que cuenta el sistema de visualización con el propósito de interactuar de forma fluida.

En el gráfico de la Figura 49, con base a la ponderación establecida, se tiene que dada la pregunta uno implica el peor de los casos: totalmente desacuerdo; mientras que diez implica el mejor de los casos: totalmente de acuerdo.



**Figura 49.** Gráfica de demanda física.

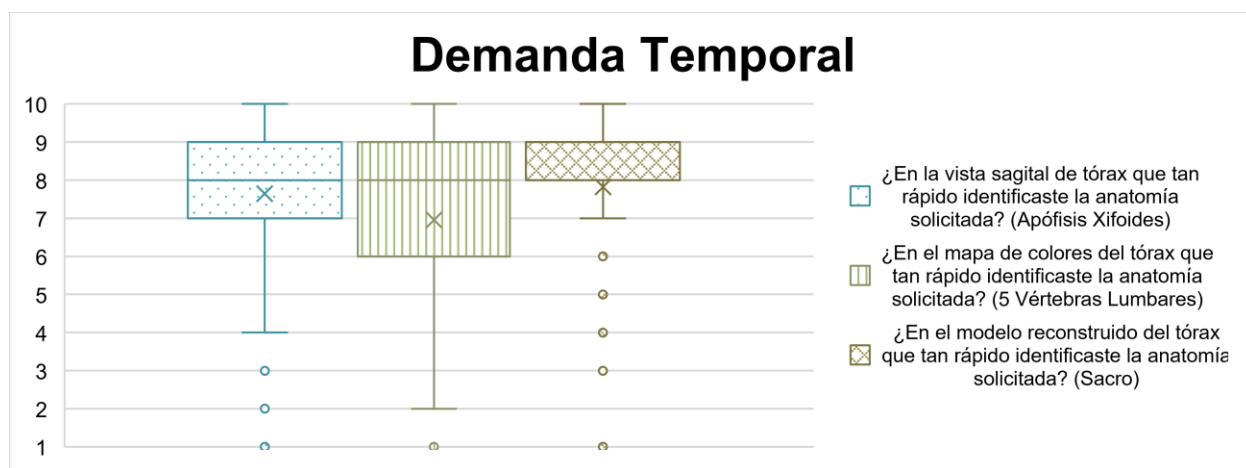
Por consiguiente, en las primeras cuatro preguntas se tiene que los resultados no son totalmente concluyentes al ser muy amplios, a pesar de ello se observa que la tendencia de los datos obtenidos es por arriba de la media de la escala empleada, lo que implica que la interacción con el sistema por medio de las interfaces de interacción y controles es adecuada, aunque requiere precisar cada una de las acciones para obtener una funcionalidad más exacta. Es importante mencionar que presentar más de un recurso anatómico no dificulta la navegación e interacción con la herramienta, lo cual se puede observar en la última pregunta.

**Tabla 21.** Resultados de los descriptivos asociados a la evaluación de la cantidad de actividad física en relación con el rubro de demanda física en la escala NASA-TLX.

Descriptivo	¿El control de zoom con el mouse en cada una de las visualizaciones fue adecuado con el movimiento esperado?	¿El control de desplazamiento con el mouse en cada una de las visualizaciones fue adecuado con el movimiento esperado?	¿El control de rotación con el mouse en la vista 3D y en los modelos 3D fue adecuado con el movimiento esperado?	¿El uso de cada de cada uno de los controles en las diferentes visualizaciones fue adecuado con la función esperada?	¿El cambio entre el diagnóstico por imagen del corazón y tórax en el menú es bueno?
Media	7.28	7.00	7.00	7.71	8.42
Mediana	8.00	8.00	8.00	8.00	9.00
Moda	8.00	8.00	8.00	9.00	10.00
Desviación estándar	2.19	2.23	2.33	1.98	1.94
Varianza de la muestra	4.80	4.97	5.45	3.94	3.79
Rango	8.00	7.00	8.00	9.00	9.00
Mínimo	2.00	3.00	2.00	1.00	1.00
Máximo	10.00	10.00	10.00	10.00	10.00
Cuenta	85.00	85.00	85.00	85.00	85.00

#### - Demanda temporal

El cuarto grupo de preguntas de esta sección fueron etiquetadas como demanda temporal (DT), visto que se requiere determinar el nivel de presión temporal sentida, por medio de las preguntas de la Figura 50 y la Tabla 22, para poder medir la razón entre el tiempo requerido y el disponible al realizar una determinada tarea en el software, de tal manera que sea posible vislumbrar la velocidad para emplear los diversos recursos visuales que componen al sistema. En el gráfico de la Figura 50, con base a la ponderación establecida, se tiene que dada la pregunta uno implica el peor de los casos: muy lento; mientras que diez implica el mejor de los casos: muy rápido.



**Figura 50.** Gráfica de demanda temporal.



Por lo que se refiere, de manera particular, la segunda pregunta presenta resultados que no son totalmente concluyente al ser muy amplios, sin embargo, la tendencia de la puntuación es hacia el mejor de los casos, lo que indica que hay que mejorar la visualización del mapa de colores, es decir, el modelo 3D generado en tiempo real, de tal manera que sea más eficiente cuando se emplee. Mientras que, para el resto de las preguntas, los resultados implican que los recursos visuales como imágenes DICOM y modelos 3D reconstruidos son más fluidos y amigables de manipular por parte de un usuario.

**Tabla 22.** Resultados de los descriptivos asociados a la evaluación del nivel de precisión temporal sentida entre el requerido y el disponible en relación con el rubro de demanda temporal en la escala NASA-TLX.

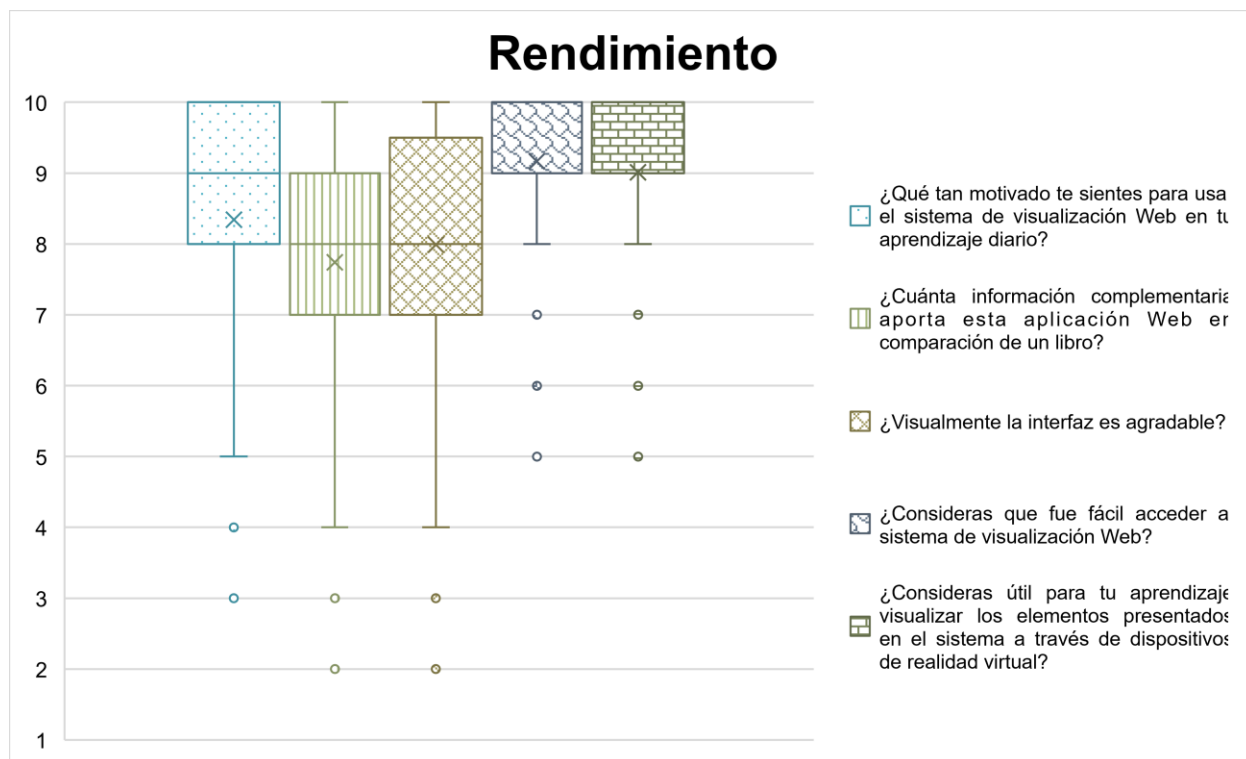
Descriptivo	¿En la vista sagital de tórax que tan rápido identificaste la anatomía solicitada? (Apófisis Xífoides)	¿En el mapa de colores del tórax que tan rápido identificaste la anatomía solicitada? (5 Vértebras Lumbares)	¿En el modelo reconstruido del tórax que tan rápido identificaste la anatomía solicitada? (Sacro)
Media	7.64	6.95	7.82
Mediana	8.00	8.00	8.00
Moda	8.00	8.00	8.00
Desviación estándar	2.06	2.41	1.88
Varianza de la muestra	4.25	5.83	3.55
Rango	9.00	9.00	9.00
Mínimo	1.00	1.00	1.00
Máximo	10.00	10.00	10.00
Cuenta	85.00	85.00	85.00

#### - Rendimiento

El quinto grupo de preguntas de esta sección fueron etiquetadas como rendimiento (R), teniendo en cuenta hasta qué punto un individuo se siente satisfecho con su nivel de rendimiento al emplear la herramienta visual, de acuerdo con las preguntas de la Figura 51 y la Tabla 23, para poder determinar la motivación para emplear la herramienta como un medio de aprendizaje considerando la usabilidad, accesibilidad y visualización gráfica, en comparación con medios impresos, y por medio de herramientas de despliegue virtual.

En el gráfico de la Figura 51, con base a la ponderación establecida, se tiene que dada la pregunta de izquierda a derecha uno implica el peor de los casos: poco motivado, poco y

totalmente desacuerdo para las últimas tres preguntas; mientras que diez implica el mejor de los casos: muy motivado, mucho y totalmente de acuerdo para las últimas tres preguntas.



**Figura 51.** Gráfica de rendimiento.

Por tanto, de manera particular, en este caso se observa y se podría indicar que los resultados en su mayoría son concluyentes al no estar tan dispersos, sin embargo, para las primeras tres preguntas, es posible señalar que se tiene una tendencia hacia el valor del mejor de los casos, lo cual significa que usar el sistema por un usuario genera una buena satisfacción como un medio de aprendizaje en comparación de la que se obtiene al emplear un libro y considerando la interfaz presentada. No obstante, las últimas dos preguntas, precisan resultados muy puntuales en relación con el aspecto atractivo del presente desarrollo al considerarlo accesible y con la posibilidad de visualización por medio de tecnologías de vanguardia que contempla la realidad mixta.

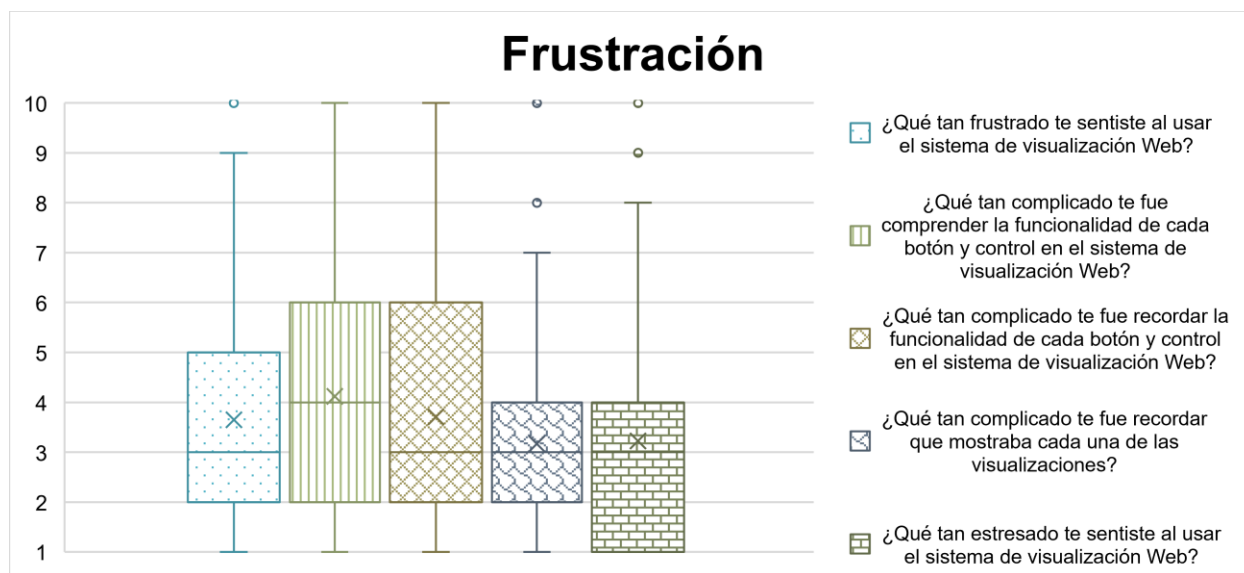
**Tabla 23.** Resultados de los descriptivos asociados a la evaluación de satisfacción en relación con el rubro de rendimiento en la escala NASA-TLX.

Descriptivo	¿Qué tan motivado te sientes para usar el sistema de visualización Web en tu aprendizaje diario?	¿Cuánta información complementaria aporta esta aplicación Web en comparación de un libro?	¿Visualmente la interfaz es agradable?	¿Consideras que fue fácil acceder al sistema de visualización Web?	¿Consideras útil para tu aprendizaje visualizar los elementos presentados en el sistema a través de dispositivos de realidad virtual?
Media	8.34	7.74	7.98	9.16	9.01
Mediana	9.00	8.00	8.00	10.00	9.00
Moda	10.00	8.00	9.00	10.00	10.00
Desviación estándar	1.69	1.74	1.94	1.17	1.32
Varianza de la muestra	2.87	3.05	3.77	1.37	1.74
Rango	7.00	8.00	8.00	5.00	5.00
Mínimo	3.00	2.00	2.00	5.00	5.00
Máximo	10.00	10.00	10.00	10.00	10.00
Cuenta	85.00	85.00	85.00	85.00	85.00

#### - Frustración

El sexto, y último, grupo de preguntas fueron etiquetadas como frustración (F), con la intención de determinar hasta qué punto un sujeto usuario del sistema se siente inseguro, estresado, irritado, desconectado, entre otros aspectos más, al realizar una determinada tarea de acuerdo con las preguntas de la Figura 52 y la Tabla 24, para poder determinar cuanta frustración, estrés y complejidad implica usar el software al comprender y recordar cada uno de los recursos presentados a través de las diversas vistas, interfaces de interacción y controles.

En el gráfico de la Figura 52, con base a la ponderación establecida, se tiene que dada la pregunta de izquierda a derecha uno implica el mejor de los casos: nada frustrado, nada complicado para las siguientes 3 preguntas y nada estresado; mientras que diez implica el peor de los casos: nada frustrado, muy complicado para las siguientes tres preguntas y muy estresado.



**Figura 52.** Gráfica de frustración.

En tal sentido, de manera particular, se observa que las primeras tres preguntas, así como la última presentan resultados que no son totalmente concluyentes al ser muy amplios, sin embargo, la tendencia que se presenta es hacia el valor del mejor de los casos, lo que nos indica que los estudiantes sienten cierta frustración, estrés y complejidad al emplear el sistema, aunque al considerar las mejoras de los grupos anteriores de preguntas, los resultados obtenidos pueden mejorar. No obstante, la cuarta pregunta, señala que los estudiantes en su mayoría recuerdan que muestra cada visualización, lo que hace evidente la identificación de los diversos recursos con los que se ha dotado al sistema de visualización de imágenes médicas.

**Tabla 24.** Resultados de los descriptivos asociados a la evaluación de inseguridad, estrés, irritación, entre otros aspectos en relación con el rubro de frustración en la escala NASA-TLX.

Descriptivo	¿Qué tan frustrado te sentiste al usar el sistema de visualización Web?	¿Qué tan complicado te fue comprender la funcionalidad de cada botón y control en el sistema de visualización Web?	¿Qué tan complicado te fue recordar la funcionalidad de cada botón y control en el sistema de visualización Web?	¿Qué tan complicado te fue recordar que mostraba cada una de las visualizaciones?	¿Qué tan estresado te sentiste al usar el sistema de visualización Web?
Media	3.64	4.11	3.70	3.16	3.22
Mediana	3.00	4.00	3.00	3.00	3.00
Moda	2.00	2.00	3.00	2.00	1.00
Desviación estándar	2.44	2.34	2.45	2.03	2.20
Varianza de la muestra	5.96	5.50	6.04	4.13	4.86
Rango	9.00	9.00	9.00	9.00	9.00
Mínimo	1.00	1.00	1.00	1.00	1.00
Máximo	10.00	10.00	10.00	10.00	10.00
Cuenta	85.00	85.00	85.00	85.00	85.00

En última instancia, se tiene la sección de nivel de satisfacción, en donde se comprende obtener la opinión de los estudiantes a partir de su análisis, evaluación y criterios individuales con relación a la experiencia posterior de uso del sistema de visualización a partir de preguntas subjetivas. Cabe destacar, que al ser cuestiones más abiertas, se tiene la oportunidad de contestar con mayor libertad y creatividad, lo que permite contrastar con los resultados anteriores, de tal forma que se tiene una retroalimentación más completa para posteriores mejoras de la herramienta, así como el análisis actual del presente desarrollo.

Por tanto, se tiene que el 93.5% de los estudiantes volverían a usar el sistema de visualización, mientras que el 95.3% considera oportuno recomendar a otro compañero de curso la utilización de la herramienta. Así mismo, el 94.1% consideran apropiado proponer a un profesor que emplee este software para que imparta su clase. De esta manera, es evidente que el desarrollo realizado presentó una buena aceptación y consideración para un posterior uso, contemplando que las discrepancias obtenidas tienen relación con algunas cuestiones puntuales identificadas en la Tabla 25, la cual nos da un panorama acerca de lo que esperan encontrar en futuras versiones del sistema, mientras que en la Tabla 26, se concentran los comentarios, opiniones y sugerencias de la experiencia de uso de la aplicación.

**Tabla 25.** Respuestas de los participantes para futuras versiones del sistema de visualización web de imágenes médicas.

¿Qué te gustaría observar en futuras versiones en el sistema de visualización web de imágenes médicas?
Histología.
Tal vez una ligera explicación de que es cada cosa o sus funciones.
Gráficos más pulidos.
Videos explicando la información sobre anatomía.
Más sistemas.
Cerebro.
Breve información relacionada con cada estructura de cuerpo humano.
Los diferentes sistemas y el cráneo
Otras regiones del cuerpo.
Texturas y alguna introducción breve.
Cerebro.
Regiones mejor explicadas o señaladas, y más partes del cuerpo.
Poder mover la estructura sin necesidad de apretar el botón derecho, poder cambiar a blanco el fondo para apreciar más algunas estructuras, poner información escrita resumida.
Órganos o sistemas en específico.
Más estructuras del cuerpo y las imágenes un poco más nítidas
Que fuera un poco más estable y accesible con todos los dispositivos para facilitar su uso.

Me gustaría seguir viendo esta parte anatómica mejor desarrolla y con más variedad.
Podría ser poner un pequeño concepto de la estructura y partes que la conforman de una parte del cuerpo.
Mejor interfaz y controles más eficientes
Aun no sé sobre todos los modelos que hay, pero creo que cualquiera que se relacione con anatomía estarían bien.
Que fueran un tanto más claras o definidas para facilitar la identificación de estructuras.
Más colores.
Me gustaría ver más modelos anatómicos y si cabe la posibilidad el nombre de cada una de ellas.
Información desplegada de las estructuras observadas al seleccionarlás, así como imágenes más claras y con mayor variedad.
Tal vez más regiones del cuerpo y el sistema esquelético.
Que se incluya el nombre de la estructura y algunas patologías.
Creo que está muy completo.
El cerebro.
Nombre de las partes que se quieran visualizar, sería buena opción dar una breve información sobre aquella estructura.
Más estructuras de todo el cuerpo humano
Más estructuras
Mayor fluidez, mejor movilidad de controles y una interfaz un poco más amigable
Ya desarrolladas las estructuras mejor definidas y con señalamientos.
Poder separar cada estructura en el modelo 3D.
Más cosas, como solo las articulaciones o músculos.
Tal vez que se eliminen las líneas y que el movimiento sea más fluido, de igual forma que pueda controlarse sin un mouse.
Que la pagina sea más rápida.
Una mejor señalización y control.
Me gustaría que fuera más manipulable para dispositivos móviles, me imagino que por las dimensiones de la pantalla se vuelve difícil de visualizar y por eso desaparecen algunas opciones.
Un sistema de realidad virtual para celulares, ya que muchas aplicaciones tienen esto; es un modelo en 3D que la persona, por medio de su cámara, la puede ver en frente de él.
Más imágenes del cuerpo humano para poder identificar las partes que lo conforman, de modo que se pueda repasar y sea materiales complementarios para cada alumno que ingrese a la página.
Cráneo.
Órganos de forma específica.
El cerebro.
Cada órgano.
Que coloquen la textura o características.
Varias estructuras y su funcionalidad.
Podría ser más estructuras, no solo de tórax y corazón.
Más facilidad de encontrar diferentes estructuras.
Imágenes un poco más realistas y colores que permiten diferenciar mejor cada sistema anatómico.
Sistema óseo, articulaciones y músculos.
Miembros superiores e inferiores
Irrigación, porque se me dificulta mucho.
Más estructuras.
Poder saber para qué sirve para que sirve cada botón, algo como una descripción.
La posibilidad de que al seleccionar una estructura de la región aparezca el nombre y de otro color.
Músculos.

Tal vez me gustaría observar más los nervios y de más sistemas del cuerpo.
Cráneo
Cerebro
Me gustaría observar más contenido y que hubiera un manejo un poco más sencillo de los colores.
Tejidos, órganos y estructuras específicas.
Que al darle clic a una estructura en cualquiera de las visualizaciones te diera el nombre de la estructura e información básica de la misma.
Imágenes un poco más claras, sin "restos" de otras estructuras.
Más vistas aparte de la vista de corazón.
Por ejemplo, en el tórax, en el modelo 3D; que al poder tocar los órganos se pudieran separar para verlos individualmente, o como se verían en la vida real cada órgano, hueso, músculo, etc.
El oído.
Órganos a mayor detalle.
Con sonidos.
Casi todas las partes del cuerpo humano.
Mejor fluidez y mayor rendimiento.
Otros sistemas y que se pueda enfocar a sólo un órgano específicamente.
Más partes del cuerpo.
Estaría muy bien tener los nombres de cada estructura a su lado
Todo el cuerpo
Sistema osteomioarticular.
Que se aplique una guía interactiva para entender todo el material que se presenta, y lo que puedes hacer con las distintas herramientas.
Me gustaría observar el cuerpo completo y poder verlo también por distintos sistemas, ya sea por músculos, huesos, nervios, vasos sanguíneos, etc.
Otras estructuras del cuerpo.

**Tabla 26.** Respuestas de los usuarios con comentarios, sugerencias u observaciones generales del sistema de visualización web de imágenes médicas.

<b>Comentarios, sugerencias u observaciones generales del sistema de visualización web de imágenes médicas</b>
Todo excelente.
Recomendaría hacer más atractiva la plataforma, en general los colores, de igual forma que al momento de rotar las imágenes no fuera tan lento. Hacer más sencillos los menús de vistas.
Es un poco difícil el manejo de la imagen de tórax, no se es igual de fácil de mover a diferencia que del corazón, y no pude centrar ninguno de los modelos.
Tal vez los colores no tan brillantes.
Es muy buena idea y aplicación, sólo son pequeños detalles. ¡Felicidades!
Considero que el fondo de color negro cansa la vista.
Estoy consciente de que esto es el comienzo en el desarrollo de esta visualización y me agrada mucho tanto el concepto como la forma tan práctica del menú.
Que lo hicieras compatible con todos los dispositivos.
Me gustaría que en el apartado de vista se reacomode los nombres de los planos, ya que tapa un poco la imagen, pero en general todo está bien.
Un poco más colorido, con una pequeña explicación de lo que se muestra
Me gustó mucho, espero que pronto sea completamente funcional en todos los dispositivos para poder usarlo de manera óptima, creo que es muy bueno y útil.
Buen trabajo.

Me parece una excelente idea emplear este tipo de herramientas.
Es muy buena idea para la enseñanza, es más didáctico y puede ayudar a comprender mejor tanto las posiciones anatómicas como la posición de los órganos o huesos.
Me parece un buen trabajo y una gran iniciativa.
Que la plataforma y los controles sean más accesibles para equipos con una interfaz más baja, fuera de eso creo que es muy buena idea este sistema y puede ser muy útil para profesores y alumnos.
Me pareció que el programa es una gran herramienta para los estudiantes porque permite observar las estructuras en 3D.
Nada por el momento.
Estuvo muy entretenido y fue divertido utilizarlo.
Solo en los colores del modelo, para mi gusto quería mejor algo no tan fuerte en los colores.
Hasta ahora está bien, aunque puede mejorar por el momento todo bien.
Gracias por dejarnos probar el sistema.
Que se termine el contenido anatómico.
Solamente el problema de que tarda en cargar la imagen, pero realmente fue por mi procesador.
Es muy buena, pero tardan en cargar las imágenes.
En mi caso algunas veces cuando pasaba de la sección de corazón a tórax, se quedaba en la sección de corazón. Las imágenes al desplazarlas algunas veces se volvían muy lento.
Me gustaron bastante las imágenes de tomografía, serán de gran utilidad. Solamente me quedo insatisfecha por lo difícil de manipular que es en el celular y porque desaparecen botones de la pantalla.
En realidad, ninguna.
Me parece una idea genial para poder repasar los conocimientos teóricos.
Está muy bien la página.
No puedo acceder a mapa de colores.
Esta increíble y muy entendible.
Todo perfecto, aunque siempre puede mejorar.
Es muy buen programa y una gran idea para poder reforzar los conocimientos de anatomía
No pude ver la parte de anatomía de colores en mi iPad.
Bastante productivo.
Poner más herramientas correspondientes al temario de 1ro.
Todo me pareció muy bueno.
Muy buena plataforma.
Rotar cuesta un poco de trabajo.
Hacer más eficaz el mouse de rotación.
En algunos planos hace falta un poco más de detalle o claridad, pero realmente está muy bien el programa.
Es una web que nos ayudará a los estudiantes, sin embargo, sería bueno tener un inicio más llamativo, así como una explicación de cada botón.
Mejorar el movimiento y desplazamiento de las imágenes.
La página pienso que en general cargo muy lento, tal vez sea más pesada de lo que espere.
Permitir la funcionalidad de mapa de colores en ciertos dispositivos.
Podrían incluir un tutorial de uso al ingresar la plataforma.
Hacer un poco más fáciles de manejar los controles, así como la rotación con el mouse.
Está lenta la rotación de imágenes por ejemplo se está de cabeza el sujeto, para rotarlo se siente muy lento
El sistema está muy bien hecho, se ve que será muy útil. La única sugerencia sería solucionar el problema de que el modelo y el mapa de colores no cargan en ciertos dispositivos.
Considero que puede ser muy útil, pero también que puede haber mejoras respecto a los controles, que sean un poco más claro, y que el uso del mouse sea más rápido, para no tardar tanto "girando" una estructura.
Hacer más ágil la rotación con el mouse.



---

El apartado "mapa de colores" no funcionó en mi dispositivo (iPad) pero si funcionó en la computadora. Al principio de la opción "modelo" tardó un poco pero muy mínimo.

---

Todo bien.

---

Todo está bien, a excepción del mapa de colores es lento.

---

Van muy bien, pero necesitamos más facilidad en las configuraciones de la interfaz.

---

La herramienta es muy útil y me pareció increíble, sólo es un poco complicado rotar los planos.

---

Me pareció una herramienta super útil, está bien elaborada y es muy fácil de utilizar.

---

Creo que es un muy complejo y detallado, pero al seleccionar las opciones para cambiar de color o forma, tenía un poco de problema porque la misma barra de herramienta no me dejaba visualizar del todo la imagen.

---

Vista en celular reducida, algunos dispositivos no admiten lo de colores.

---

Falla en mapa de colores.

---

El mapa de colores no es tan fluido al momento de mover el tórax y también al modificar las barras de color y renderización.

---

Me gustó mucho esta plataforma, solo le agregaría más sistemas.

---

Que sea compatible con todos los dispositivos.

---

---

# Capítulo 7

## Discusión

---

De acuerdo con las bases que sustentan el presente trabajo, una de las áreas que ha tenido un mayor impacto por el desarrollo y evolución de herramientas tecnológicas ha sido el entrenamiento de habilidades médicas, lo que supone que estudiantes y practicantes, futuros profesionistas, necesiten contar con extraordinarios conocimientos anatómicos para su día a día. En este sentido, el uso de nuevas tecnologías en relación a visualizaciones de imágenes médicas y representaciones 3D de conceptos básicos anatómicos desde hace una década se ha concentrado en usuarios jóvenes (Zhou et al., 2019; Birr et al., 2013) y ha ido en aumento, lo cual es evidente por el grupo que se consideró para evaluar el sistema al encontrarse en un rango de edad entre 17 a 28 años, lo que permite observar una tendencia de interés y familiaridad que representan este tipo de recursos.

No obstante, grupos de personas de edad más avanzada, como lo son médicos con una amplia experiencia y que muchas veces practican la docencia, muestran una mayor dificultad intuitiva ante tecnologías inmersivas y nuevas, a pesar de ello manifiestan interés en usar gráficos 3D interactivos y modelos 3D como un complemento para la enseñanza (Birr et al., 2013) y estar a la vanguardia; hecho que también ha impulsado a docentes, ingenieros y diversas instituciones médicas y educativas en México para potenciar el uso de recursos de punta por medio de realidad aumentada, virtual o mixta.

Así mismo, con la información previa que se obtuvo de los participantes en la encuesta, es posible vislumbrar que los antecedentes relacionados con recursos visuales médicos en su mayoría han sido a través de imágenes médicas bidimensionales (Birr et al., 2013; Angarita y Beltrán, 2016), en donde el termino asociado como DICOM, no resulta tan familiar en individuos que recién comienzan su formación en el área médica; además, las implementaciones hasta este momento explotan ampliamente este tipo de materiales, sin embargo, como se ha mencionado, en los últimos años, el desarrollo de elementos interactivos 3D se ha abordado en gran medida, por lo que cierto porcentaje de usuarios los conocen. No obstante, para el caso de sistemas de realidad mixta, no resulta tan común en cuanto a experiencias y asociación a recursos gráficos de

renderización e inmersión, pero que es de interés para explorar como una alternativa y complemento, como expresaron los estudiantes en la encuesta.

Por otro lado, el sistema representa un medio autodidacta para el aprendizaje de un alumno al ser una plataforma que proporciona una noción más realista de las estructuras anatómicas, de tal forma que se tiene una herramienta que despliega escenarios de realidad virtual obtenidos de un paciente real lo que permite tener simulaciones más realistas en pruebas de conceptos (Teodoro, 2021; Birr et al., 2013; Zhou et al., 2019). Además, el entrenamiento para la adquisición de conocimientos y habilidades en ambientes inmersivos resulta una gran alternativa en comparación con entrenamientos tradicionales que muchas veces por cuestiones adversas no es posible acceder a ellos, como lo ha representado la pandemia por COVID-19, y significan un medio para adquirir experiencia lo más cercano a la realidad; tal que la curva de aceptación va en aumento en cuanto a su uso, capacidad de aprendizaje y práctica, así como disfrute de diversos grupos de personas en relación con su edad, lo que los hace más eficientes (Zhou et al., 2019).

Considerando este punto, se observó que los estudiantes no tuvieron dificultades para recordar los diversos recursos con los que se dotó al software, lo que resultó atractivo por las diferentes visualizaciones, la calidad de los despliegues gráficos bidimensionales y tridimensionales, los colores empleados, así como la navegación entre los dos casos clínicos empleados para proporcionar dos diferentes estructuras anatómicas. A pesar de ello, es evidente que se requiere hacer mejoras para hacer más robusta la aplicación, al considerar otros aspectos que permitirán elevar la curva de aceptación, al considerar agregar información de apoyo y referencia, mejorar el diseño para hacerlo aún más atractivo, mejorar las interfaces y controles de interacción para hacerlos más precisos y exactos, así como considerar otras funcionalidades de apoyo para la manipulación rápida y fluida de la herramienta.

Así mismo, la identificación de estructuras anatómicas específicas en las imágenes 3D, el modelo 3D renderizado en tiempo real y el modelo 3D renderizado a partir de una reconstrucción, ha sido rápido, fluido y transparente para los alumnos, lo que permite demostrar que emplear este sistema como complemento en la enseñanza, en donde el docente lo emplea como guía en la impartición de una clase, y posteriormente, el estudiante la utiliza para reforzar y explorar los conocimientos adquiridos, significan una gran alternativa, lo cual es señalado en las respuestas obtenidas al considerar usar la aplicación a fin de prepararse para presentar sus evaluaciones o subsanar nociones que no pueden dimensionar y adquirir a través de medios impresos u otro tipo

de materiales similares, como también ha sido señalado en otros desarrollo similares (Birr et al., 2013).

En este paso, una limitante del sistema sigue siendo la compatibilidad de uso en cuanto a navegadores y dispositivos, a pesar de haber disminuido esta brecha al realizar una implementación Web, puesto que esta vía ha sido considerada como una solución eficiente, accesible, y a un costo menor en comparación con otras implementaciones por los recursos que se requieren (Angarita y Beltrán, 2016). En este caso fue posible observar dos aspectos muy importantes: el primero, al proporcionar dos casos clínicos como recursos en el sistema se encontraron tiempos de latencia diferentes entre uno y otro, tal que el caso de corazón resulto más liviano para cargar en comparación con el caso de tórax, lo cual indica para futuras versiones, considerar segregar y clasificar los contenidos para homologar la cargar de los recursos, así como la viabilidad de enriquecer la herramienta para agregar más estructuras anatómicas a partir de casos clínicos reales; y el segundo, a pesar de considerar diferentes tipos de visualizaciones y una solución Web, el acceso y uso total en todos los navegadores y dispositivos, aun no es posible alcanzar, debido a que ciertas funcionalidades en cuanto a gráficos requieren de configuraciones y hardware específico, que se sigue mejorando y optimizando, sin embargo, el espectro para un mayor uso del sistema es realmente mayor en comparación con otras implementaciones de realidad virtual o aumentada que se han desarrollado o están disponibles en el mercado.

Las discrepancias identificadas como casos aislados observados en los gráficos de cada uno de los rubros de la encuesta, hacen evidente inconvenientes muy puntuales en cuanto a no poder acceder por el dispositivo, principalmente en la visualización de mapa de colores, lo que implica compatibilidad con la GPU, que es donde se realiza el proceso de esta vista, de tal forma que habría que optimizar y validar esta cuestión de manera previa para proporcionar una retroalimentación del inconveniente y dar otra alternativa, situación que se encuentra en otros trabajos (Arguiñarena et al., 2015). Así mismo, mejorar las configuraciones y consideraciones en cuanto a los recursos que se cargan en la herramienta para disminuir y mejorar los tiempos de latencia de las visualizaciones. Por otro lado, realizar una mejor implementación para dispositivos móviles, puesto que la interacción en comparación con la que se tiene en un equipo de escritorio presenta grandes diferencias. No obstante, la respuesta en cuanto a seguir usando la herramienta individualmente, en clase, e incluso recomendarla, ha sido satisfactoria.

---

# Capítulo 8

## Conclusiones

---

En este trabajo se presentó el desarrollo y validación de la primera versión de un nuevo sistema de visualización web de imágenes médicas como una herramienta disponible, accesible, interactiva e intuitiva para la enseñanza, aprendizaje, diagnóstico y planeación quirúrgica. La aplicación incorpora tres recursos a partir de diferentes visualizaciones como lo son las imágenes DICOM de TC, modelos 3D renderizados en tiempo real y modelos 3D renderizados a partir de reconstrucciones previas, de dos casos clínicos reales (corazón y tórax), lo que permite proveer una diversidad de materiales para la exploración e identificación de ciertas estructuras anatómicas según como se requiera.

La metodología propuesta e implementada para la adquisición de recursos en formato NRRD y VTK a partir de TC, se detalló y realizó un comparativo para demostrar la eficiencia de uso, de tal forma que representa una vía para mejorar los tiempos de latencia al cargar recursos gráficos a través de un navegador web, así como un punto de partida para compararla con otras alternativas, puesto a que no es la única.

Así mismo, la metodología propuesta e implementada para la creación del software, supone una reducción de los tiempos de segmentación de estructuras anatómicas, lo cual tiene como resultado el proporcionar materiales a estudiantes y docentes de una forma más rápida, así como poner a disposición de los profesionales de la salud modelos 3D que corresponden con casos clínicos reales para el diagnóstico y planeación quirúrgica; dado que la segmentación a través de otras herramientas implican un trabajo de un par de semanas, mientras que con esta metodología se reduce el proceso a un par de horas, lo cual demuestra una gran diferencia de tiempos, y resulta crucial al brindar la atención a un paciente.

El sistema de visualización muestra una prueba de concepto para mostrar a través de realidad extendida, más precisamente realidad virtual, cada una de las visualizaciones implementadas, lo que permite constatar el término de metaverso definido por Mann, un mundo mediado no solo por lo real y virtual, sino también por dispositivos; por lo que, para futuras versiones, el desarrollo y configuración de manera más amplia para este tipo de inmersión, resulta para abordar y robustecer la herramienta.

De acuerdo con los resultados obtenidos de los participantes en la evaluación de experiencia de uso de la aplicación se demuestra que el sistema de visualización fue bien aceptado, al significar un medio con diversos materiales interactivos y accesibles que resultan fluidos, intuitivos y accesibles, que no se encuentran de manera conjunta en otras implementaciones, e incluso por cuestiones de éticas se presentan ciertas limitaciones.

Además, se documenta de forma puntual y general, por un lado, los desarrollos en cuanto a la realidad mixta, y por otro, los sistemas de visualización, al sustentar sus bases y seguir la vía de implementación en el área médica, de tal forma que la herramienta de visualización web demuestra la importancia de aplicar diversos conceptos para crear y desarrollar herramientas tecnológicas de vanguardia que estén en un continuo mejoramiento para enriquecerlas con el fin de que puedan ser empleadas en la vida real.

Finalmente, es importante mencionar que este trabajo fue seleccionado como ganador en el “Concurso InnovaUNAM para la Enseñanza y el Aprendizaje de Contenidos Curriculares Prácticos en Ciencias y Humanidades a Distancia” con el proyecto titulado “Atlas Anatómico 3D con Realidad Virtual para la Enseñanza-Aprendizaje de la Anatomía Humana y Diagnóstico Médico usando Imágenes de Tomografía Computarizada” (Anexo E), lo cual permitirá implementar la solución propuesta en los contenidos curriculares de la carrera de Médico Cirujano de la Facultad de Medicina de la UNAM, contribuyendo al fortalecimiento del proceso enseñanza-aprendizaje de la Anatomía Humana.

# Anexos

## A. Casos de uso desglosados del sistema de visualización web de imágenes médicas

Desglose de los subsistemas por medio de casos de uso que conforman al sistema de visualización Web de imágenes médicas, tal que se describen las interacciones entre las acciones o actividades que un usuario puede realizar en él.

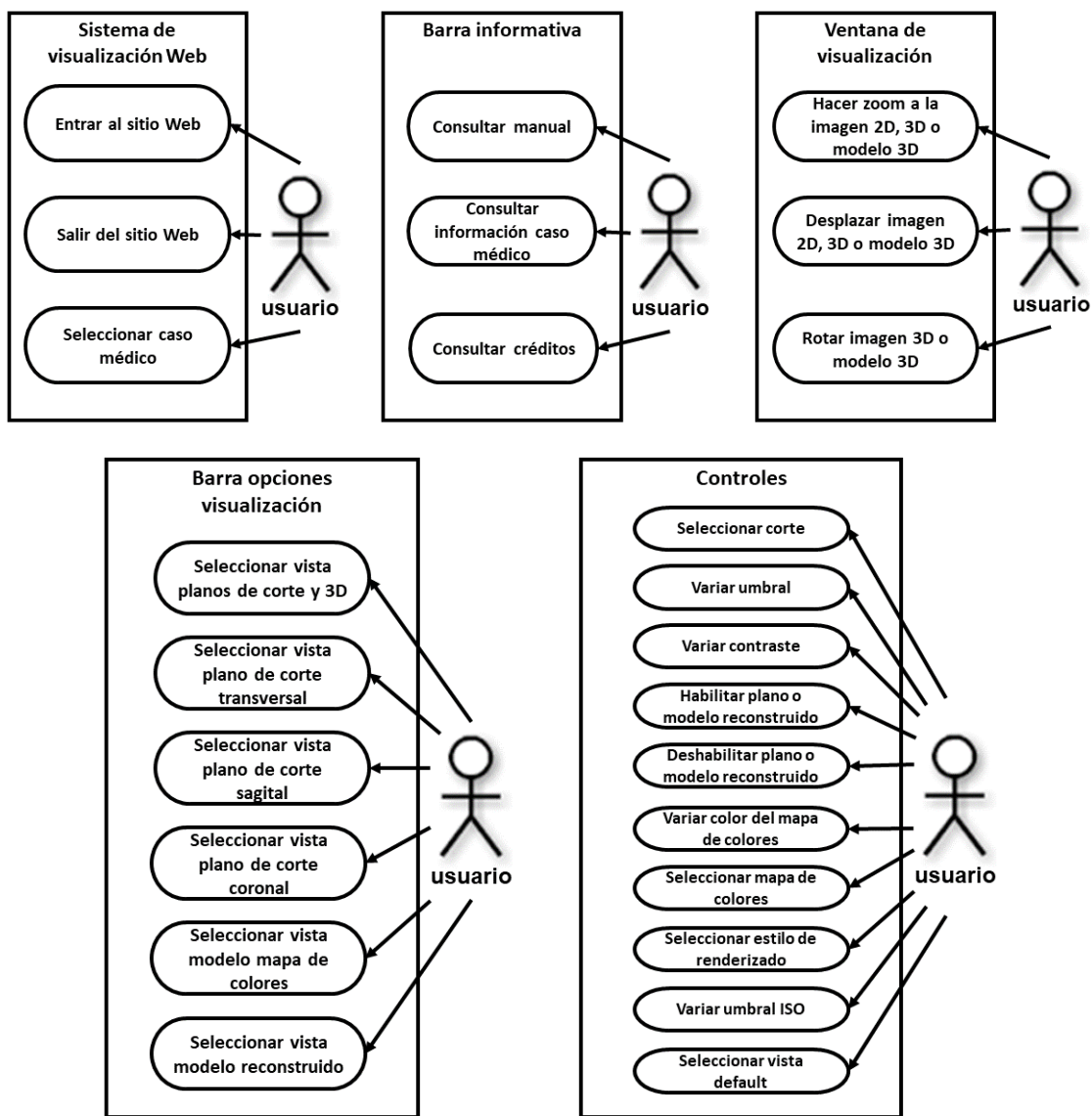


Figura 53. Casos de uso desglosados del sistema de visualización web de imágenes médicas.

## B. Estructura generalizada del script HTML de una visualización

Para poder definir cada visualización en el sistema se empleó la siguiente estructura en cada script HTML desarrollado, en donde se puede indicar que los comentarios que hacen referencia a ruta se refieren a la ubicación de los archivos \*.css y \*.js en el servidor respecto a la del programa en él.

**Tabla 27.** Estructura generalizada del script HTML de una visualización.

Definición
<pre>&lt;!DOCTYPE html&gt; &lt;html lang="en"&gt;   &lt;head&gt;     &lt;!-- Nombre que hace referencia al caso que se está visualizando --&gt;     &lt;title&gt; &lt;!-- nombreCaso --&gt; &lt;/title&gt;     &lt;meta charset="utf-8"&gt;     &lt;meta       name="viewport"       content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum- scale=1.0"&gt;     &lt;!-- Hoja de estilos empleada por los módulos de THREE.JS --&gt;     &lt;link type="text/css" rel="stylesheet" href="&lt;!-- ruta --&gt;"&gt;     &lt;!-- Hoja de estilos propios --&gt;     &lt;link rel="stylesheet" type="text/css" href="&lt;!-- ruta --&gt;" media="screen" /&gt;   &lt;/head&gt;    &lt;body&gt;     &lt;!-- Definición de una barra que indica la visualización que se ha seleccionado --&gt;     &lt;div id='info'&gt;&lt;p id='vv'&gt;&lt;b&gt;&lt;!-- nombreVisualizacion --&gt;&lt;/b&gt;&lt;/p&gt;&lt;/div&gt;     &lt;!-- Definición del contenedor principal del campo de visualización --&gt;     &lt;div id='vistaVolumen'&gt;&lt;/div&gt;      &lt;!-- Definición del script con la funcionalidad de la visualización seleccionada --&gt;     &lt;script type="module"&gt;       // Se importan los módulos necesarios de THREE.JS para generar       // la visualización seleccionada       import * as &lt;!-- nombreAliasModulo --&gt;         from &lt;!-- ruta --&gt;;        import { &lt;!-- nombreAliasModulo --&gt; }         from &lt;!-- ruta --&gt;;        ...        // Se definen las variables necesarias para generar la visualización seleccionada       var &lt;!-- nombreVariable --&gt;, . . . , &lt;!-- nombreVariable --&gt;;       const &lt;!-- nombreVariable --&gt;, . . . , &lt;!-- nombreVariable --&gt;;       let &lt;!-- nombreVariable --&gt;, . . . , &lt;!-- nombreVariable --&gt;;        // Se ejecuta la función de inicio       init();        // Se ejecuta la función de animación       animate();        // Se define la función de INICIO: Se encarga de la inicialización de cámaras,       // escenas, luces, imágenes, render y controles, así como las configuraciones       // necesarias para generar la visualización seleccionada       function init() {         ...       }     &lt;/script&gt;   &lt;/body&gt; &lt;/html&gt;</pre>



---

```
    window.addEventListener( 'resize', onWindowResize, false );
}

// Se define la función de ANIMACIÓN: Se encarga de definir un bucle de animación,
// es decir, un ciclo que permitirá que el renderizador dibuje la escena cada vez
// que se actualice la escena; en una pantalla típica esto significa 60 veces por
// segundo.

// Así mismo se especifican configuraciones de la cámara, render y controles, según
// la visualización que se haya seleccionado, puesto que requieren de un refresh por
// la acción que se ejecuta en esta función
function animate() {
    requestAnimationFrame( animate );
    ...
}

// Se define la función de REDIMENSIÓN: Se encarga de realizar diversos ajustes en
// caso de redimensionar la ventana de trabajo respecto a la cámara, el render y los
// controles, según la visualización que se haya seleccionado
function onWindowResize() {
    ...
}

// En algunas visualizaciones se definen funciones que realizan una acción en
// específico, las cuales se mandan a llamar en la función de inicio
function <!-- nombreFuncion -->(){
    ...
}

</script>
</body>
</html>
```

---

## C. Algoritmo de renderizado empleado con Three.js

Para poder implementar *volume rendering* por medio del sombreador de fragmentos en Three.js se hace uso del *shader* “VolumenShader.js”, el cual pone en práctica un algoritmo de la aplicación de *ray casting*, ISO y MIP, tal que se muestra a continuación el algoritmo simplificado del mismo.

**Tabla 28.** Algoritmo de renderizado empleado con Three.js.

Algoritmo
<pre>Inicio  VolumeRenderShader = {    uniforms : {     Configurar los valores iniciales de la estructura uniforms     (u_size, u_renderstyle, u_renderthreshold, u_climb, u_data, u_cmdata)   }    vertexShader : [     Definir las variables necesarias para manejar el vertexShader      Función Main ()       Preparar transformaciones para mapear a vista de cámara        // Se da un paso hacia atrás (en las coordenadas de la leva) hasta el plano de       corte cercano y proyecta hacia atrás, esto no da toda la información para       calcular el rayo y truncarlo al cono de visión       Proyectar la coordenada del vértice local a la posición de la cámara y al plano       de recorte lejano        // (z = -1 en clip coords)       Realizar la intersección de rayo y plano de recorte cercano        // (z = +1 en clip coords)       Realizar la intersección de rayo y plano de recorte lejano        Establecer variaciones y posición de salida     Fin Función   ]    fragmentShader : [     // Se establece como distancia máxima sqrt(3) a través del volumen de renderizado     Definir las variables necesarias para manejar el fragmentShader      Función Main ()       Normalizar la información del plano de recorte        Calcular el vector unitario que apunta en la dirección de la vista a través de       este fragmento        // La posición es la cara posterior del cuboide, por lo que la distancia inicial       calculada en el punto el producto de abajo es la distancia desde el plano del       clip cercano a la parte posterior del cuboide       Calcular la distancia (negativa) a la superficie frontal o cerca del plano de       recorte        Definir la posición inicial en la superficie frontal        Establecer cuantos pasos se darán        Obtener la ubicación inicial y el vector de pasos en coordenadas de textura        // Esto ayuda a establecer si los rayos están correctamente orientados       Mostrar el número de pasos</pre>

---

```

        Si renderizado = 0 Entonces
            caso_MIP (sl, s, ns, vr)
        Si no Si renderizado = 1 Entonces
            caso_ISO (sl, s, ns, vr)
        Fin Si
Fin Función

Función sample1 (texcoords)
    // Muestra el valor flotante de una textura 3D, asume datos de intensidad
    Obtener coordenadas de textura
Fin Función

Función apply_colormap (val)
    Obtener color textura con base a un CLIM entre el mínimo y máximo de color para
    un determinado dato
Fin Función

Función caso_MIP (start_loc, step, nsteps, view_ray)
    Definir variables necesarias para manejar caso MIP

    // En WebGL 1 el índice de bucle no se puede comparar con expresión no constante,
    entonces usamos un máximo codificado de forma rígida y una condición adicional
    dentro del bucle

    // Ingresar a un bucle de ray casting
    Para i <- 0 Hasta numMaxPaso Con Pazo 1 Hacer
        Verificar número de paso

        Definir muestra de la textura 3D

        // Aplicar operación MIP
        Si VAL > MAX_VAL Entonces
            MAX_VAL <- VAL
            MAX_I <- ITER
        Fin Si

        Avanzar la ubicación más profundamente en el volumen
    Fin Para

    // Proporciona imágenes más nítidas
    Refinar la ubicación

    Resolver el color final
Fin Función

Función caso_ISO (start_loc, step, nsteps, view_ray)
    // Se define init transparente, color final y paso a la deriva de la muestra.
    // Se define low_threshold
    Definir variables necesarias para manejar caso ISO

    // En WebGL 1 el índice de bucle no se puede comparar con expresión no constante,
    entonces usamos un máximo codificado de forma rígida y una condición adicional
    dentro del bucle

    // Ingresar a un bucle de ray casting
    Para i <- 0 Hasta numMaxPaso Con Pazo 1 Hacer
        Verificar número de paso

        Definir muestra de la textura 3D

        // Aplicar operación ISO
        Si VAL > LOW Entonces
            Toma el último intervalo en pasos más pequeños

            Para a <- 0 Hasta numPasoRefinamiento Con Pazo 1 Hacer
                Definir muestra de la textura 3D

                Si VAL > RENDERTHRESHOLD Entonces

```

---

---

```

        add_lighting(v, il, ds, vr)
    Fin Si
  Fin Para
Fin Si

  Avanzar la ubicación más profundamente en el volumen
Fin Para
Fin Función

Función add_lighting (val, loc, step, view_ray)
  // Se calcula el color incorporado de iluminación

  Obtener la dirección de vista (rayo) normalizada

  Calcular el vector normal a partir del gradiente

  Definir la magnitud del gradiente

  Normalizar

  Invertir normal para que apunte hacia el espectador

  // Realizar según el número de luces (luces múltiples)
  Para i <- 0 Hasta numMaxLuces Con Pazo 1 Hacer
    // Asegurarse de evitar la visión cero
    Obtener la dirección de la luz

    Calcular las propiedades de la iluminación

    Calcular máscara

    Calcular colores con base a la máscara y las propiedades de la iluminación
  Fin Para

  Calcular el colore final componiendo diferentes componentes

Fin Función
]
}

Exportar VolumeRenderShader

```

---

## D. Encuesta de evaluación del sistema de visualización Web de imágenes médicas

Mediante la presente encuesta cualitativa, basada en la escala NASA-TLX, la cual es un instrumento estandarizado de valuación de la carga mental de trabajo, se pretende evaluar la experiencia de uso del sistema de visualización Web de imágenes médicas.

### SECCIÓN I: Información y experiencia previa del participante

En esta sección se incorporan preguntas relacionadas con la experiencia previa del participante en cuanto al sistema propuesto.

- Edad: \_\_\_\_\_

- ¿En qué año de estudios te encuentras?

<input type="checkbox"/> 1ro	<input type="checkbox"/> 2do	<input type="checkbox"/> 3ro	<input type="checkbox"/> 4to	<input type="checkbox"/> 5to	<input type="checkbox"/> 6to
------------------------------	------------------------------	------------------------------	------------------------------	------------------------------	------------------------------

- ¿Consideras que las tecnologías de realidad virtual son útiles para el aprendizaje?

<input type="checkbox"/> Sí	<input type="checkbox"/> No
-----------------------------	-----------------------------

- ¿Cómo consideras que es el acceso a sistemas o herramientas Web para el aprendizaje médico relacionado a la visualización de imágenes y modelos 3D anatómicos?

Accesible 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------

 No accesible

- ¿Has empleado algún sistema de visualización Web de imágenes médicas?

<input type="checkbox"/> Sí	<input type="checkbox"/> No
-----------------------------	-----------------------------

En caso afirmativo, en dicho sistema, ¿empleaste alguno de los siguientes elementos?

- Imágenes DICOM
- Modelos 3D interactivos
- Imágenes 3D
- Otro

Especificar cual: \_\_\_\_\_

### SECCIÓN II: NASA-TLX (Task Load Index)

En esta sección se busca evaluar cada demanda psicomotora del participante.

- **Esfuerzo (E)**

E1. ¿Cuánto esfuerzo mental y físico consideras que implica usar el sistema de visualización Web al momento de estudiar en comparación con un libro?

Poco esfuerzo 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Mucho esfuerzo

E2. ¿Consideras que la manipulación de la interfaz del sistema de visualización Web es muy complejo?

Totalmente desacuerdo 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Totalmente de acuerdo

E3. ¿Qué tan complejo consideras el uso del sistema de visualización Web durante una clase?

Poco complejo 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Muy complejo

E4. ¿Consideras que el sistema de visualización Web es un medio autodidacta para tú aprendizaje?

Totalmente desacuerdo 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Totalmente de acuerdo

E5. ¿En qué grado considerarías una plataforma Web, como la que acabas de emplear, para tener una noción más realista de las estructuras anatómicas?

Poco 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Mucho

- **Demanda Mental (DM)**

DM1. ¿Qué tan fácil fue ubicar cada una de las secciones del sistema de visualización Web?

Muy difícil 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Muy fácil

DM2. ¿La información e instrucciones especificadas en el manual fueron claras?

Nada claras 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Muy claras

DM3. ¿Cómo calificarías la calidad de la información presentada en el sistema de visualización Web?

Muy mala 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Muy buena

DM4. ¿Qué tan intuitivos son los nombres indicados en los botones y controles de la aplicación Web?

Nada intuitivos 

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

 Muy intuitivos

DM5. ¿Qué tan adecuados son los colores empleados para visualizar imágenes DICOM, vista 3D y modelos 3D?

Nada adecuado 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Muy adecuado

• **Demanda Física (DF)**

DF1. ¿El control de zoom con el mouse en cada una de las visualizaciones fue adecuado con el movimiento esperado?

Totalmente desacuerdo 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Totalmente de acuerdo

DF2. ¿El control de desplazamiento con el mouse en cada una de las visualizaciones fue adecuado con el movimiento esperado?

Totalmente desacuerdo 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Totalmente de acuerdo

DF3. ¿El control de rotación con el mouse en la vista 3D y en los modelos 3D fue adecuado con el movimiento esperado?

Totalmente desacuerdo 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Totalmente de acuerdo

DF4. ¿El uso de cada uno de los controles en las diferentes visualizaciones fue adecuado con la función esperada?

Totalmente desacuerdo 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Totalmente de acuerdo

DF5. ¿El cambio entre el diagnóstico por imagen del corazón y tórax en el menú es bueno?

Totalmente desacuerdo 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Totalmente de acuerdo

• **Demanda Temporal (DT)**

DT1. ¿En la vista sagital de tórax que tan rápido identificaste la anatomía solicitada?  
(Apófisis Xifoides)

Muy lento 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Muy rápido

DT2. ¿En el mapa de colores del tórax que tan rápido identificaste la anatomía solicitada?  
(5 vértebras lumbares)

Muy lento 

□1	□2	□3	□4	□5	□6	□7	□8	□9	□10
----	----	----	----	----	----	----	----	----	-----

 Muy rápido

DT3. ¿En el modelo reconstruido del tórax que tan rápido identificaste la anatomía solicitada?  
(Sacro)

Muy lento 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy rápido

• **Rendimiento (R)**

R1. ¿Qué tan motivado te sientes para usar el sistema de visualización Web en tu aprendizaje diario?

Poco motivado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy motivado

R2. ¿Cuánta información complementaria aporta esta aplicación Web en comparación de un libro?

Poco 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Mucho

R3. ¿Visualmente la interfaz es agradable?

Totalmente desacuerdo 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Totalmente de acuerdo

R4. ¿Consideras que fue fácil acceder al sistema de visualización Web?

Totalmente desacuerdo 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Totalmente de acuerdo

R5. ¿Consideras útil para tu aprendizaje visualizar los elementos presentados en el sistema a través de dispositivos de realidad virtual?

Totalmente desacuerdo 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Totalmente de acuerdo

• **Nivel de Frustración (NF)**

NF1. ¿Qué tan frustrado te sentiste al usar el sistema de visualización Web?

Nada frustrado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy frustrado

NF2. ¿Qué tan complicado te fue comprender la funcionalidad de cada botón y control en el sistema de visualización Web?

Nada complicado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy complicado

NF3. ¿Qué tan complicado te fue recordar la funcionalidad de cada botón y control en el sistema de visualización Web?

Nada complicado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy complicado



NF4. ¿Qué tan complicado te fue recordar que mostraba cada una de las visualizaciones?

Nada complicado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy complicado

NF5. ¿Qué tan estresado te sentiste al usar el sistema de visualización Web?

Nada estresado 

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7	<input type="checkbox"/> 8	<input type="checkbox"/> 9	<input type="checkbox"/> 10
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

 Muy estresado

### SECCIÓN III: Nivel de satisfacción

En esta sección se incorporan preguntas subjetivas sobre la experiencia de aprendizaje con el uso del sistema de visualización Web de imágenes médicas.

- Si tuvieras la oportunidad, ¿volverías a usar este sistema de visualización Web de imágenes médicas?

<input type="checkbox"/> Sí	<input type="checkbox"/> No
-----------------------------	-----------------------------

- ¿Recomendarías usar este sistema de visualización Web de imágenes médicas a otro compañero de curso?

<input type="checkbox"/> Sí	<input type="checkbox"/> No
-----------------------------	-----------------------------

- ¿Recomendarías usar este sistema de visualización Web de imágenes médicas a un profesor para que imparta su clase, cuando así se requiera?

<input type="checkbox"/> Sí	<input type="checkbox"/> No
-----------------------------	-----------------------------

- ¿Qué te gustaría observar en futuras versiones en el sistema de visualización Web de imágenes médicas?

\_\_\_\_\_

- Comentarios, sugerencias u observaciones generales del sistema de visualización Web de imágenes médicas.

\_\_\_\_\_

## E. Reconocimiento por participar en el concurso InnovaUNAM.



La Universidad Nacional Autónoma de México

otorga el presente

# Reconocimiento

a

María Fernanda Ortiz Figueroa

por su participación en el proyecto

*“Atlas anatómico 3D remoto con realidad virtual para la enseñanza-aprendizaje de la anatomía humana y diagnóstico médico usando imágenes de tomografía computarizada” dentro del Concurso InnovaUNAM para la Enseñanza y el Aprendizaje de Contenidos Curriculares Prácticos en Ciencias y Humanidades a Distancia.*

**“Por mi Raza hablará el Espíritu”**

Ciudad Universitaria, CDMX a 20 de Septiembre de 2021

Dr. Enrique Graue Wiechers  
Rector



**Figura 54.** Reconocimiento por participar en el concurso InnovaUNAM para la Enseñanza y el Aprendizaje de Contenidos Curriculares Prácticos en Ciencias y Humanidades a Distancia con el proyecto: “Atlas anatómico 3D remoto con realidad virtual para la enseñanza-aprendizaje de la anatomía humana y diagnóstico médico usando imágenes de tomografía computarizada”.



---

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE105820 “Desarrollo de un atlas de cabeza y cuello con maniquíes de órganos impresos en 3D y realidad mixta para la enseñanza de la neuroanatomía humana”

---

# PAPIME

---

# Referencias Bibliográficas

---

- Adame, M., Valdés, B., García, M., Galeana, H., y Bahena, D. (2000). *Sistema de capacitación orientado a médicos para uso del sistema de visualización de imágenes médicas PACS*. (Tesis de pregrado). Facultad de Ingeniería – UNAM, México.
- Aguilar, E., Vivas, A., y Sabater, J. (2016). Una aproximación a la realidad aumentada y sus aplicaciones quirúrgicas. *Entre Ciencia e Ingeniería*. 12(24). 15-24. <https://doi.org/10.31908/19098367.3811>
- Aguinaga, H., Rivera, J., Tamayo, L., Osorno, C., y Tobón, R. (2006). Tomografía axial computarizada y resonancia magnética para la elaboración de un atlas de anatomía segmentaria a partir de criosecciones axiales del perro. *Revista Colombiana de Ciencias Pecuarias*. 19(4). 451-9.
- AL. (2021). *Anatomy Learning*. Recuperado de <https://anatomylearning.com/> el 25 de abril de 2021.
- AMCE. (2021). *Asociación Mexicana de Cirugía Endoscópica*. Recuperado de <https://amce.com.mx/historia/> el 12 de marzo de 2021.
- Anatomy3DATlas. (2013). *Anatomy 3D Atlas*. Recuperado de <https://anatomy3datlas.com/> el 25 de abril de 2021.
- Anderson, R., Razavi, R., & Taylor, A. (2004). Cardiac anatomy revisited. *Journal of Anatomy*. 205(3). 159-177.
- Angarita, C., y Beltrán, N. (2016). Aplicación web para la visualización de imágenes médicas medicomweb. *Respuestas*. 13(2). 32- 37. <https://doi.org/10.22463/0122820X.540>
- Arguñarena, E., del Fresno, M., Massa, J., Escobar, P., y Santiago, M. (2015). Visualización de imágenes médicas de alta resolución mediante una aplicación Zero Footprint. *Tecnología y Ciencia*. 13(27). 299 – 306.
- Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*. 21(6). 34-47.
- Azuna, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S. & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*. 21(6). 34-47.
- Baum, L. (1900). *The Wonderful Wizard of Oz*. Chicago, Estados Unidos: George M. Hill Company.
- Bautista, V. (2014). Cirugía de mínima invasión. *Cirugía Cardiovascular*. 21(2). 107-110.
- Beléndez, A. (1996). *Fundamentos de óptica para ingeniería informática*. España: Servicio de Publicaciones de la Universidad de Alicante.
- Berryman, D. (2012). Realidad aumentada. *Servicios de referencia médica trimestral*. 31(2). 212-218.
- Birr, S., Mönch, J., Sommerfeld, D., Preim, U., & Preim, B. (2013). The LiverAnatomyExplorer: a WebGL-Based surgical teaching tool. *IEEE Computer Graphics and Applications*. 33(5). 48-58. <https://doi.org/10.1109/MCG.2013.41>

- Blender. (2021). *About*. Recuperado de: <https://www.blender.org/> el 5 de abril de 2021.
- Bond, S. (2016). *After the success of 'Pokémon GO': how will augmented reality impact*. Recuperado de <https://www.forbes.com/sites/drsarahbond/2016/07/17/after-the-success-of-pokemon-gohow-will-augmented-reality-impact-archaeological-sites/?sh=76ba1ab042a1> el 2 de marzo de 2021.
- Cabrilo, I., Schaller, K., & Bijlenga, P. (2015). Augmented reality assisted bypass surgery: embracing minimal invasiveness. *World neurosurgery*. 83(4). 596–602.
- Carmigniani, J., & Furht, B. (2011). *Augmented Reality: An Overview. Handbook of Augmented Reality*. Nueva York, Estados Unidos: Springer. [https://doi.org/10.1007/978-1-4614-0064-6\\_1](https://doi.org/10.1007/978-1-4614-0064-6_1)
- Carmona, R. (2000). *Introducción al rendering directo de volúmenes*. Venezuela: Lecturas en Ciencias de la Computación. ISSN 1316-6239.
- Castro, S., Flores, D., y Martig, S. (2010). *Realidad aumentada en visualización. Argentina: XII Workshop de Investigadores en Ciencias de la Computación*. Recuperado de [http://sedici.unlp.edu.ar/bitstream/handle/10915/19501/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/19501/Documento_completo.pdf?sequence=1) el 5 de febrero de 2021.
- Caudell, T., & Mizell, D. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. *Boeing Computer Services, Research and Technology*. 2(1). 659-669. <https://doi.org/10.1109/HICSS.1992.183317>
- Chapman, R., Riddle, D., & Merlo, J. (2009). Techniques for supporting the author of outdoor mobile multimodal augmented reality. *Proceedings of the Human Factors and Ergonomics Society 53rd Annual Meeting*. 53(27). <https://doi.org/10.1177/154193120905302707>
- Charlton, S.G. (2002). Measurement of cognitive states in test and evaluation. En, S.G. Charlton y F.G.O'Brien (eds.), *Handbook of Human Factors Testing and Evaluation*. Londres: Lawrence Erlbaum Associates.
- Chromium. (2021). *GPU accelerated compositing in Chrome*. Recuperado de <https://www.chromium.org/developers/design-documents/gpu-accelerated-compositing-in-chrome> el 3 de abril de 2021.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2008). MeshLab: an Open-Source Mesh Processing Tool *Sixth Eurographics Italian Chapter Conference*. 129-136.
- Coll, C. (2007). Las competencias en la educación escolar: algo más que una moda y mucho menos que un remedio. *Aula de Innovación Educativa*. 161. 34 – 39.
- Corella, D., y Ordovás, J. (2007). Genes, dieta y enfermedades cardiovasculares. *Investigación y Ciencia*. 77(13). 74-83.
- Crespo, J. (2012). *Diseño y desarrollo de un juego en WebGL*. (Tesis de pregrado). Universidad Politécnica de Cataluña, España.

- De los Nietos, M. (2007). Nociones básicas de anatomía, fisiología y patología cardíaca: bradiarritmias y taquiarritmias. *Enfermería en Cardiología*. 14(1). 7-20.
- De Paolis, L., Teodoro, S., Padilla, M., Domínguez, C., Muscatello, S. & Hernández, A. (2021). An augmented reality platform with hand gestures-based navigation for applications in image-guide surgery: prospective concept evaluation by surgeons. *International Journal of Human-Computer Interaction*. <https://doi.org/10.1080/10447318.2021.1926116>
- DICOM. (2021). *About DICOM: Overview*. Recuperado de <https://www.dicomstandard.org/about> el 25 de enero de 2021.
- Dini, F., Guarini, G., Ballo, P., Carluccio, E., Maiello, M., Capozza, P., Innelli, P., et al. (2013). The left ventricle as a mechanical engine: from Leonardo da Vinci to the echocardiographic assessment of peak power output-to-left ventricular mass. *Journal of Cardiovascular Medicine*. 14(3). 214-20.
- DMCA. (2014). *Ciclo cardíaco del corazón sistema circulatoria de la sístole cuerpo humano*. [Ilustración]. DMCA. Recuperado de <https://www.freepng.es/png-ls6yid/> el 5 de marzo de 2021.
- Eguiluz, J. (2006). *Introducción a CSS*. Recuperado de <https://uniwebisidad.com/libros/css> el 28 de marzo de 2021.
- El Universal. (2020, 25 agosto). *Realidad aumentada*. [Fotografía]. El Universal. Recuperado de <https://www.eluniversal.com.mx/techbit/compras-con-realidad-aumentada> el 16 de febrero de 2021.
- Feiner, S., Macintyre, B., & Sellgmann, D. (1993). Knowledge based augmented reality. *Communications of the ACM*. 36(7). 53-62. <https://doi.org/10.1145/159544.159587>
- Filipi, C., Fitzgibbons, R., & Salerno, G. (1991). Historical Review: Diagnostic Laparoscopy to Laparoscopic Cholecystectomy and Beyond. *St Louis Quality Medical Publishing*.
- Fischer, J., Bartz, D., y Straber, W. (2005). *Intuitive and lightweight user interaction for medical augmented reality*. Alemania: Visual Computing for Medicine.
- FZ. (2021). *Overview*. Recuperado de <https://filezilla-project.org/> el 15 de abril de 2021.
- García, I. (2010). *Aportaciones a la segmentación y caracterización de imágenes médicas 3D*. Sevilla, España: Universidad de Sevilla.
- Garduño, M. (2018). *Simulación computacional de cortes en objetos rígidos con retroalimentación háptica para entrenamiento en micro neurocirugía*. (Tesis de pregrado). Facultad de Ingeniería - UNAM, México.
- Gartner, L., & Hiatt, J. (2006). *Color textbook of histology*. Estados Unidos: Elsevier Health Sciences.
- George, D., & Morris, R. (1989). A computer-driven astronomical telescope guidance and control system with superimposed star field and celestial coordinate graphics display. *Journal of the Royal Astronomical Society of Canada*. 83(1). 32-41.
- GIMP. (2021). *Gimp*. Recuperado de <http://www.gimp.org/es/> el 4 de abril de 2021.

- Glossop, N., Wedlake, C., Moore, J., Peters, T. & Wang, Z. (2003). Laser projection augmented reality system for computer assisted surgery. *In Medical Image Computing and Computer-Assisted Intervention-MICCAI*. 239-246. [https://doi.org/10.1007/978-3-540-39903-2\\_30](https://doi.org/10.1007/978-3-540-39903-2_30)
- Gómez, D., Fullman, N., Lamadrid, H., Cahuana, L., Darney, B., Ávila, B., et al. (2016). Dissonant health transition in the states of Mexico, 1990 – 2013: a systematic análisis for the Global Burden of Disease Study 2013. *Lancet*. 388(10058). 2386 – 2402.
- Guerrero, J., Ortiz, S., Castillo, R., y Salazar, C. (2014). La cirugía de invasión mínima: antecedente histórico; presente y perspectivas futuras en el ISSTE. *Revista de Especialidades Médico-Quirúrgicas*. 19(3). 375-386.
- Guindo, J. (2004). *Apuntes de Cardiología*. Barcelona: J&C Ediciones Médicas.
- Harris, D., Wilson, M., & Vine, S. (2020). Development and validation of a simulation workload measure: the simulation task load index (SIM-TLX). *Virtual Reality*. 24. 557-566. <https://doi.org/10.1007/s10055-019-00422-9>
- Hart, S. & Staveland, L. (1988). Development of Nasa-Tlx (Task Load Index): result of empirical and theoretical research. *Human Mental Workload*. 139 – 183.
- Hernández, D. (2020, 29 septiembre). *Realidad virtual: una apuesta de innovación dentro del aula*. [Fotografía]. Tecnológico de Monterrey. Recuperado de <https://tec.mx/es/noticias/aguascalientes/educacion/realidad-virtual-una-apuesta-de-innovacion-dentro-del-aula> el 16 de febrero de 2021.
- ImageJ. (2021). *Introduction*. Recuperado de <https://imagej.net/learn/> el 25 de enero de 2021.
- Jacobson, I. (1998). *Applying UML in the inified proces*. Rational Software.
- Jones, A. (2020, 19 mayo). *Pontiac, Michigan Water Quality Report*. [Fotografía]. Epicwater Filters. Recuperado de <https://www.epicwaterfilters.com/blogs/news/pontiac-michigan-water-quality-report> el 16 de febrero de 2021.
- Joomla. (2021). *About Joomla!*. Recuperado de: <https://www.joomla.org/about-joomla.html> el 15 de abril de 2021.
- Kato, H. (1999). *ARToolKit*. Washington: HITLab. Recuperado de <http://ns1.hitl.washington.edu/artoolkit/> el 26 de febrero de 2021.
- Kelion, L. (2018, 7 noviembre). *Holograma Stephen Hawking*. [Fotografía]. BBC News. Recuperado de <https://www.bbc.com/mundo/noticias-46103146> el 16 de febrero de 2021.
- Khronos. (2021). *WebGL*. Recuperado de <https://www.khronos.org/webgl/> el 2 de abril de 2021,
- Koyama, T. (2009). *Introduction to flartoolkit*. Adobe System Incorporated.
- Krueger, M. (1997). Responsive environments. *National Computer Conference*. 77(2). 423-433. <https://doi.org/10.1145/1499402.1499476>



- Lacroute, P. (1995). *Fast Volume Rendering Using Shear-Warp Factorization of the Viewing Transformation*. Estados Unidos: Reporte Técnico CSL-TR-95-678, Universidad de Standford.
- León, D. (2016). *Realidad aumentada mediante gráficos 3D, seguimiento y lentes de realidad virtual*. (Tesis de maestría). CCADET-UNAM, México.
- Lifshitz, A., (2001). *Informática y medicina*. México: FACMED. Recuperado de [http://www.facmed.unam.mx/eventos/seam2k1/2001/ponencia\\_abr\\_2k1.htm](http://www.facmed.unam.mx/eventos/seam2k1/2001/ponencia_abr_2k1.htm) el 1 de febrero de 2021.
- LOC. (2020). *Wavefront OBJ File Format*. Recuperado de: <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml> el 5 de abril de 2021.
- López, G., Rudolph, M., y Guzmán, C. (2009). Valoración digital de índices de atenuación radiológica de estructuras anatómicas normales y materiales dentales observables en imágenes panorámicas. *Revista Facultad de Odontología Universidad Antioquia*. 20(2). 119-128.
- Ludlow, B. (2015). Virtual reality: emerging applications and future directions. *Rural Special Education Quarterly*. 34(3). 3-10. <https://doi.org/10.1177/875687051503400302>
- Mahmoudi, S., Akhondi, A., Rahmani, R., Faghieh, S., Taimouri, V., Sabouri, A., & Soltanian, H. (2010). Web-Based interactive 2D/3D medical image processing and visualization software. *Computer Methods and Programs in Biomedicine*. 98(2). 172-182.
- Mahmoudi, S., Akhondi, A., Rahmani, R., Faghieh, S., Taimouri, V., Sabouri, A., & Soltanian, H. (2010). Web-Based interactive 2D/3D medical image processing and visualization software. *Computer Methods and Programs in Biomedicine*. 98(2). 172-182.
- Mann, S. (1994). *Mediated reality (Technical Report 260)*. Cambridge, MA: MIT Media Lab, Perceptual Computing Group.
- Mann, S. (1997). An historical account of the 'WearComp' and 'WearCam' inventions developed for applications in 'personal imaging'. Digest of Papers. *First International Symposium on Wearable Computers*. 66-73. <https://doi.org/10.1109/ISWC.1997.629921>
- Mann, S. (1999). Mediated reality: university of Toronto RWM project. *Linux Journal*. 59.
- Mann, S. (2001). *Intelligent image processing*. Nueva York, Estados Unidos: IEEE John Wiley and Sons.
- Marín, J., Peñaloza, M., Rodríguez, J., Ruiz, P., Ostaiza, I., Vinueza, A., Galarza, M., Añez, R., y Bermúdez, V. (2019). Importancia de las imágenes médicas en la valoración de la anatomía y función cardiovascular. *Revista Latinoamericana de Hipertensión*. 14(5). 629-628.
- Marras, W., & Karwowski, W. (2006). *Fundamentals and assessment tools for occupational ergonomics*. Estados Unidos, Florida: Taylor & Francis.
- Martins, V., Kirner, T., & Kirner, C. (2015). Subjective usability evaluation criteria of augmented reality applications. *Virtual, Augmented and Mixed Reality (VAMR) 2015: lecture notes in computer science*. 9179. 39-48. [https://doi.org/10.1007/978-3-319-21067-4\\_5](https://doi.org/10.1007/978-3-319-21067-4_5)



- Mellado, M. (2011). *Development and implementation of a mobile AR-based assistance system on the android-platform for the smart factory*. Kaiserslautern: Ricardo Campos García.
- Menéndez, R. (2021). *Lenguaje de programación JavaScript*. Universidad de Murcia. Recuperado de <https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf> el 28 de marzo de 2021.
- Microsoft. (2020, 26 agosto). *The interactions between computers, humans, and environments*. [Ilustración]. Microsoft. Recuperado de <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality> el 16 de febrero de 2021.
- MIK. (2002). *SSH and File Transfers*. Recuperado de: [https://docstore.mik.ua/oreilly/networking\\_2ndEd/ssh/ch03\\_08.htm](https://docstore.mik.ua/oreilly/networking_2ndEd/ssh/ch03_08.htm) el 9 de abril de 2021.
- Milgram, P., Takemur, H., Utsumi, A., & Kishino, F. (1994). *Augmented Reality: A class of displays on the reality-virtuality continuum*. Kyoto, Japón: Telemulator and Telepresence Technologies.
- Morales, B. (2020, 19 febrero). *La reacción de madre al reunirse con su hija fallecida en realidad virtual*. [Fotografía]. Concierto 88.5. Recuperado de <https://www.concierto.cl/2020/02/la-reaccion-de-madre-al-reunirse-con-su-hija-fallecida-en-realidad-virtual/> el 16 de febrero de 2021.
- Morse, B. (2000). *Lecture 4: Thresholding*. Escocia: Brigham Young University.
- Napel, S., Rubin, G., & Jeffrey, R. (1993). STS-MIP: a new reconstruction technique for CT of the chest. *Journal of Computer Assisted Tomography*. 17(5). 832-8.
- NRRD. (2021). *NRRD*. Recuperado de <http://teem.sourceforge.net/nrrd/> el 25 de enero de 2021.
- Oculus. (2020). *Quest 2*. Recuperado de: [https://www.oculus.com/quest-2/?locale=es\\_ES](https://www.oculus.com/quest-2/?locale=es_ES) el 9 de agosto de 2021.
- OMS. (2021). *Enfermedades no transmisibles*. Recuperado de: <https://www.who.int/es/news-room/fact-sheets/detail/noncommunicable-diseases> el 15 de abril de 2021.
- OTech. (2021). *DICOM: Introducción*. Recuperado de [https://www.otechimg.com/\\_getDoc.cfm?id=43&started=1](https://www.otechimg.com/_getDoc.cfm?id=43&started=1) el 25 de enero de 2021.
- Pandey, H., Maurya, A., Prajapati, R., Pandey, A. & Nagve, V. (2020). *Augmented Reality in Agriculture*. Manchester: EasyChair.
- Peddie, J. (2016). *Augmented Reality: Where We Will All live*. Estados Unidos: Springer International Publishing.
- Persson, A., Dahlström, N., Engellau, L., Larsson, E., Brismar, T., & Smedby, Ö. (2004). Volume Rendering Compared with Maximum Intensity Projection for Magnetic Resonance Angiography Measurements of the Abdominal Aorta. *Acta Radiológica*. 45(4). 453-459.
- PRIMAL PICTURES. (2021). *Primal Pictures Powering Anatomy*. Recuperado de <https://www.primalpictures.com/> el 25 de enero de 2021.
- Ravela, S., Weiss, R., Draper, B., Pinette, B., Hanson, A., & Riseman, E. (1994). *Stealth navigation: planning and behaviors*. Amherst: University of Massachusetts.

- Reitmayr, G., & Schmalstieg, D. (2004). Collaborative augmented reality for outdoor navigation and information browsing. *Vienna University of Technology*. 31-41.
- Roberts, J., Ritsos, P., Badam, S., Brodbeck, D., Kennedy, J. & Elmqvist, N. (2014). Visualization beyond the Desktop – the Next Big Thing. *IEEE Computer Graphics and Applications*. 34(6). 26-34. <https://doi.org/10.1109/MCG.2014.82>
- Roberts, J., Ritsos, P., Badam, S., Brodbeck, D., Kennedy, J. & Elmqvist, N. (2014). Visualization beyond the Desktop – the Next Big Thing. *IEEE Computer Graphics and Applications*. 34(6). 26-34. <https://doi.org/10.1109/MCG.2014.82>
- Rosenberg, L. (1992). *The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments*. Brooks City-Base: Armstrong Laboratory.
- Rubio, J. L., y Gertrudix, M. (2016). Realidad Virtual (HMD) e Interacción desde la perspectiva de la construcción narrativa y la comunicación: propuesta taxonómica. *Revista ICONO14: Revista científica de comunicaciones y tecnologías emergentes*. 14(2). 1-24. <https://doi.org/10.7195/ri14.v14i2.965>
- Rutledge, T., Stucky, E., Dollarhide, A., Shively, M., Jain, S., & Wolfson, T., (2009). A real-time assessment of work stress in physicians and nurses. *Health Psychology*. 28. 194-200.
- Salmon, M. (1993). *Utilidad de la imagen tridimensional para la planeación quirúrgica en las disrrafias espinales*. (Tesis maestría). UNAM, México.
- Sánchez, A., Bobadilla, M., Dimas, B., Gómez, M. y González, G. (2016). Enfermedad cardiovascular: primera causa de morbilidad en un hospital de tercer nivel. *Revista Mexicana de Cardiología*. 27(3). 98-102.
- Sánchez, H., Gozález, J., Jiménez, D., y Martínez, F. (2014). Enfermedades cardiovasculares, respiratorias y renales. *Medicine*. 11(62). 3677 – 3690.
- Schindelin, J., Arganda, I., & Frise, E. (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*. 9(7). 676-682. <https://doi.org/10.1038/nmeth.2019>
- Schmitto, J., Mokashi, C., & Cohn, L. (2012). Minimally invasive valve surgery. *Journal American College of Cardiology*. 56(6). 455-462.
- Sezgin, M., y Sankur, B. (2004). Encuesta sobre técnicas de umbralización de imágenes y evaluación cuantitativa del rendimiento. *Journal of Electronic Imaging*. 13(1). 146-165.
- Silvetti, A., Castro, S.M., y Delrieux, C. (1999). *Marching cubes: una implementación eficiente que contempla todos los casos*. Argentina: Workshop de Investigadores en Ciencias de la Computación.
- Singh, A. (2012). Study of comparison of channel satisfaction among employees across the various retail stores. *International Journal of Innovative Research and Development*. 1(4). 78- 94
- Slicer. (2021). *What is 3D Slicer*. Recuperado de: <https://www.slicer.org/> el 5 de abril de 2021.
- Spaner, S., & Warnock, G. (1997). A brief history of endoscopy, laparoscopy, and laparoscopic surgery. *Journal of Laparoendoscopic & Advanced Surgical Techniques*. 7(6). 369-73.

- Spight, D., Hunter, J., Jobe, B., et al. (2014). *Schwart's Principles of Surgery*. México: McGraw-Hill Education.
- Starner, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R. & Pentland, A. (1997). Augmented reality through wearable computing. *Presence*. 6(4). 386-398.
- Stephenson, N. (1992). *Snow Crash*. Nueva York, Estados Unidos: Bantam Books.
- Steuer, J. (1992). Defining virtual reality: dimensions determining telepresence. *Journal of Communication*. 42(4). 73-93. <https://doi.org/10.1111/j.1460-2466.1992.tb00812.x>
- Sutherland, I. (1965). The Ultimate Display. *Proceedings of the IFIPS Congress*. 65(2). 506-508.
- Sutherland, I. (1968). A head-mounted three dimensional display. *Fall Joint Computer Conference*. 68(1). 757-764. <https://doi.org/10.1145/1476589.1476686>
- Teodoro, S. (2008). *Modelado de un ambiente virtual para un sistema de simulación de cirugía de próstata*. (Tesis pregrado). Facultad de Ingeniería – UNAM, México.
- Teodoro, S. (2021). *Simulación de reparación de aneurismas usando realidad virtual para el entrenamiento médico en microcirugía cerebrovascular*. (Tesis doctoral). Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas – UNAM, México.
- Thomas, B., Close, B., Donoghue, J., Squires, J., De Bond, P., Morris, M., & Piekarski, W. (2000). ARQuake: an outdoor/indoor augmented reality first person application. *Digest of Papers, Fourth International Symposium on Wearable Computers*. 139-146. <https://doi.org/10.1109/ISWC.2000.888480>
- Threejs Fundamentals. (2021). *Three.js*. Recuperado de <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html> el 3 de abril de 2021.
- UBC. (2018). *Functional Neuroanatomy*. Recuperado de <https://www.neuroanatomy.ca/> el 25 de abril de 2021.
- Ukimura, O., & Gill, I. (2008). Imaging-assisted endoscopic surgery: Cleveland clinic experience. *Journal of Endourology*. 22(4). 803–810. <https://doi.org/10.1089/end.2007.9823>
- UM. (2016). *Desarrollo de aplicaciones Web. Introducción general a HTML y CSS*. Recuperado de <https://www.um.es/docencia/barzana/DAWEB/Introduccion-a-html-y-css.html> el 28 de marzo de 2021.
- UNAM. (2021). *MEDVIAS*. Recuperado de <http://papime.amazing-graphics.com/index.php/diagnostico-por-imagen/corazon> de 25 de enero de 2021.
- VOXEL-MAN. (2021). *Voxel-Man 3D-Navigator: Inner Organs*. Recuperado de <https://www.voxel-man.com/3d-navigators/inner-organs/> el 25 de enero de 2021.
- VTK. (2021). *About VTK*. Recuperado de: <https://vtk.org/> el 6 de abril de 2021.
- W3. (2020). *WebXR Device API*. Recuperado de: <https://www.w3.org/TR/webxr/> el 15 de abril de 2021.
- Watt, M., & Watt, A. (1992). *Advanced Animation and Rendering Techniques: Theory and Practice*. Estados Unidos: ACM Press.

- Wilson, M., Poolton, J., Malhotra, N. et al. (2011) Development and validation of a surgical workload measure: the Surgery Task Load Index (SURG-TLX). *World J Surg* 35. 1961. <https://doi.org/10.1007/s00268-011-1141-4>
- Wynngarden, J. & Smith, L. (1998). *Tratado de medicina interna*. México: Interamericana.
- Xin, M., Olson, E. & Bassel, R. (2013). Mending broken hearts: cardiac development as a basis for adult heart regeneration and repair. *Nature Reviews Molecular Cell Biology*. 14(8). 529-541.
- Zhou, J., & Tönnies, K. (2003). *State of the art for volume rendering*. Alemania: Universidad de Magdeburg.
- Zhou, Z., Jiang, S., Yang, Z., y Zhou, L. (2019). Personalized planning and training system for bronchotherapy base don virtual reality. *Virtual Reality*. 23. 347-361. <https://doi.org/10.1007/s10055-018-0350-7>