



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SOFTWARE DE SIMULACIÓN PARA VISUALIZAR Y
MANIPULAR RECONSTRUCCIÓN TRIDIMENSIONAL DE
DATOS BIOMÉDICOS PARA POTENCIALIZAR EL
APRENDIZAJE DEL PROFESIONAL DEL SECTOR SALUD.

TESIS PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN

**P R E S E N T A :
EDGAR MARTÍNEZ HERNÁNDEZ**

DIRECTORA DE TESIS:
M.I. JIMENA OLVERES MONTIEL



CIUDAD UNIVERSITARIA

2015

AGRADECIMIENTOS

Agradecimientos a mis padres

Agradezco en primer lugar la oportunidad de darme la vida a mí papá Eleazar Martínez y mi mamá Angélica Hernández, porque sin ella no estuviera terminando mi ciclo profesional de licenciatura, gracias por el apoyo moral y económico para afrontar el costo de mi profesión, sé que fue una época difícil y con su gran esfuerzo que dieron a lo máximo pudieron darme una gran ayuda, gracias por aguantar todas las desveladas y todas las molestias por la mañana para levantarme por la madrugada y poder llegar a clase en la universidad, gracias por creer en mí en que podía concluir una carrera, gracias por creer en que sería un buen profesional, gracias por dejarme hacer siempre lo que quiero, gracias por dejarme estudiar una ingeniería, gracias por los consejos tan importantes para afrontar los problemas cotidianos, gracias por educarme con moral y ética, gracias papás por todo.

Agradecimientos a mis hermanos

Agradezco el apoyo que me han dado para concluir este ciclo, siempre es un ejemplo los consejos de los hermanos para tenerlos presentes, fueron de gran ayuda para terminar mi profesión, gracias a mi hermana Edith Martínez que apoyo en comprar la primer computadora que llego a la casa, sin esta computadora que es la principal herramienta de mi profesión no sería el profesional que soy, gracias de todo corazón, espero que mi hermano menor Mario Jesús Martínez tomé mi ejemplo para poder titularse en menos tiempo que yo y concluya sus estudios profesionales, gracias por todo hermanos.

Agradecimientos a mis profesores

Agradezco mucho a mis profesores que han sido una clave para llegar hasta este peldaño de la pirámide de la educación, agradezco a mis queridos profesores de la máxima casa de estudios, quienes fueron los actores principales para formarme como el profesional que actualmente soy, gracias por darme una formación integra como universitario, gracias por la ayuda para convertirme en un profesional con ética, pero sobre todo gracias por compartir su conocimiento y adoptarlo en mí para utilizarlo en bienestar social, gracias por todo el esfuerzo, dedicación y motivación, gracias por mi desarrollo profesional, que se convirtió para resolver problemas y necesidades en beneficio de la sociedad, gracias por su interés por enseñar, gracias por todo profesores.

Agradecimientos a mi directora de tesis

Agradezco mucho a mi directora de tesis por darme la oportunidad de terminar mi tesis y poder obtener el grado de licenciatura, gracias por comprender los problemas por lo que pase en ese momento que causaron un retraso en el avance de mi tesis, gracias por comprender que a veces es primero los problemas familiares que cualquier otra cosas, gracias por tenerme tanta paciencia por los avances, gracias por el interés para desarrollar de una mejor forma el tema de tesis, gracias por darme la oportunidad de ser dirigido por una gran profesionista en la materia de ingeniería biomédica, gracias por todo M.I. Jimena Olveres Montiel.

Agradecimientos a mi universidad

Agradezco mucho pero mucho a mí gran universidad, gracias UNAM por permitirme estudiar en la máxima casa de estudios, primero un bachillerato y por mi buen desempeño como estudiante poder tener la oportunidad de elegir una carrera profesional, gracias UNAM sin tu apoyo como universidad pública hubiera sido más difícil tener una formación profesional, gracias por todo UNAM, gracias universidad prometo utilizar mis conocimientos, mi formación adquirida para el beneficio del país y no en perjuicio del mismo, sino que la nación me lo demande.

Agradecimientos a dios

Agradezco a dios por darme la oportunidad de poder vivir esta gran etapa de mi vida, gracias dios por darme salud y vida para poder terminar mi profesión, gracias dios por darme fuerza para afrontar todo el camino que recorro en mi vida, gracias dios por darme el don del ingenio y creatividad para poder convertir y crear cosas que ayuden a otras personas, gracias dios por darme la oportunidad de que toda mi familia este conmigo para vivir este momento, gracias dios por darme cada día vida, gracias dios por todo.

ÍNDICE

CONTENIDO	Página
RESUMEN	1
INTRODUCCION	5
CAPÍTULO I REALIDAD VIRTUAL	11
1.1 ¿Qué es realidad virtual?.....	13
1.1.1 Características de la realidad virtual.....	14
1.1.2 Implementación de la realidad virtual.....	14
1.1.3 Estructura del software de realidad virtual para visualización CGI 3D.....	16
1.2 Características principales de los entornos de realidad virtual.....	17
1.2.1 Ambientes virtuales inmersivos.....	17
1.2.2 Diferencias con ambientes virtuales no inmersivos.....	17
1.2.3 Ventajas de los ambientes inmersivos sobre los no inmersivos.....	17
1.2.4 Sistemas estereoscópicos.....	18
1.3 Entrada y salida de datos para sistemas de realidad virtual inmersiva CGI 3D.....	18
1.3.1 Visualización estereoscópica CGI 3D inmersiva.....	19
1.3.2 Visualización estereoscópica pasiva CGI 3D.....	20
1.3.3 Visualización estereoscópica activa CGI 3D.....	21
CAPÍTULO II TECNOLOGÍAS Y METODOLOGÍAS DE IMPLEMENTACIÓN	23
2.1 DICOM para obtener los datos de imágenes biomédicas.....	25
2.1.1 Descripción general de las imágenes biomédicas DICOM.....	25
2.2 Reconstrucción volumétrica de estructuras corporales.....	26
2.3 Diseño asistido por computadora 3D.....	27
2.3.1 Transformación geométrica traslación.....	28
2.3.2 Transformación geométrica rotación.....	28
2.3.3 Transformación geométrica escalamiento.....	28
2.3.4 Representación geométrica tridimensional.....	29
2.3.5 Proyección paralela.....	29
2.3.6 Eliminación de líneas ocultas.....	29
2.3.7 Modelado geométrico de la estructura anatómica.....	30
2.4 Diseño asistido por computadora 2D.....	31
2.4.1 Texturizado para geometría tridimensional.....	32
2.5 Motores gráficos para el aprendizaje del profesional del sector salud.....	34
2.5.1 Motores gráficos para el desarrollo de aplicaciones.....	35
2.6 Motor gráfico Quest3D.....	36
2.6.1 Bibliotecas de utilidad del motor gráfico.....	37
2.6.2 Render del motor gráfico.....	37
2.7 Front-End.....	38
2.8 Escenario gráfico / eliminación y optimización.....	38
2.9 Renderizado de bajo nivel.....	39
2.9.1 Métodos de iluminación.....	39
2.9.2 Componente ambiental.....	40
2.9.3 Componente difusa.....	40
2.9.4 Componente especular.....	41
2.10 Interfaces con el dispositivo gráfico.....	42
2.11 Motor de física.....	42

CONTENIDO

Página

CAPÍTULO III DESARROLLO.....	43
3.1 Análisis y definición de requerimientos.....	46
3.1.1 Estudio de factibilidad.....	47
3.1.2 Obtención y análisis de requerimientos.....	47
3.1.3 Especificación de requerimientos.....	49
3.2 Diseño e implementación de software.....	50
3.2.1 Diseño arquitectónico.....	50
3.2.2 Diseño de interfaz gráfica de usuario.....	51
3.2.3 Diseño de componentes de la interfaz gráfica de usuario de interacción.....	51
3.2.4 Diseño de áreas de interacción.....	52
3.2.5 Arquitectura del sistema de software biomédico 3D.....	52
3.2.6 Especificación de base de datos.....	53
3.3 Diseño de las estructuras de datos.....	54
3.3.1 Obtención de las estructuras de datos biomédicos.....	55
3.3.2 Obtención de las estructuras de datos no biomédicos.....	59
3.3.3 Desarrollo de materiales para el texturizado de los datos no biomédicos.....	61
3.3.4 Exportar datos no biomédicos para desarrollar lógica.....	63
3.4 Desarrollo del sistema de software biomédico.....	64
3.4.1 Desarrollo de Front-End del Sistema de software biomédico.....	65
3.5 Desarrollo del sistema de software biomédico 3D.....	65
3.5.1 Obtención de los datos importados al ambiente de desarrollo.....	66
3.5.2 Desarrollo de máquina finita de estados.....	68
3.5.3 Módulo menú barra: Función de pantalla completa del sistema de software biomédico 3D.....	69
3.5.4 Módulo menú barra: Función de reinicio del sistema de software biomédico 3D.....	70
3.5.5 Módulo menú barra: Función de requerimientos del sistema de software biomédico 3D.....	71
3.5.6 Módulo de modelos biomédicos.....	72
3.5.7 Reconstrucción de cerebro con metodología de apilamiento de imágenes.....	73
3.5.8 Módulo de modelos no-biomédicos.....	78
3.5.9 Módulo de estereoscópica pasiva.....	78
3.5.10 Módulo de estereoscópica activa.....	80
3.5.11 Módulo de información de modelos: Función detección de mouse colisión.....	81
3.5.12 Módulo de información de modelos: Función comprobador de colisión por rayo.....	82
3.5.13 Módulo de cortes del cuerpo humano.....	83
3.5.14 Módulo de interactividad visual.....	85
3.5.15 Módulo de interactividad visual: Función de posición de la cámara.....	86
3.5.16 Módulo de interactividad visual: Función de cámara básica.....	88
3.5.17 Módulo de interactividad visual: Función de cámara ortogonal.....	88
3.5.18 Módulo de interactividad visual: Función de cámara de inspección de objeto.....	89
3.5.19 Módulo de interactividad visual: Función de cámara de tercera persona.....	90
3.5.20 Módulo de interactividad visual: Función de auto rotación.....	92
3.5.21 Módulo de interactividad visual: Función de zoom de microscópico.....	93

CONTENIDO

Página

CAPÍTULO IV ANÁLISIS DE RESULTADOS.....	95
4.1 Pruebas de la generación de datos biomédicos.....	97
4.2 Pruebas de la generación de datos no biomédicos	98
4.3 Pruebas de los componentes del sistema de software biomédico 3-D.....	99
4.3.1 Pruebas de los componente de la iluminación.....	99
4.3.2 Pruebas de verificación de la cámara ortogonal.....	100
4.3.3 Pruebas de verificación de la cámara básica.....	100
4.3.4 Pruebas de verificación de la cámara de inspección de objetos.....	101
4.3.5 Pruebas de verificación de la cámara en tercera persona.	101
4.3.6 Pruebas de verificación para la detección de la colisión del mouse.....	102
4.3.7 Pruebas de verificación para la colisión de objetos	102
4.3.8 Pruebas de verificación para la colisión de comprobación por rayo.....	103
4.3.9 Pruebas de verificación para la función estereoscópica pasiva.....	104
4.3.10 Pruebas de verificación para los cortes anatómicos.....	105
CONCLUSIONES.....	107
GLOSARIO.....	111
BIBLIOGRAFÍA.....	115
ANEXO I: HERRAMIENTAS TECNOLÓGICAS PARA EL ANÁLISIS Y DISEÑO.....	119
ANEXO II: DIAGRAMA CASOS DE USO.....	124
ANEXO III: DIAGRAMA DE ACTIVIDAD.....	132
ANEXO IV: INTERFAZ GRÁFICA DE USUARIO.....	144
ANEXO V: HERRAMIENTAS TECNOLÓGICAS.....	150
ANEXO VI: NVIDIA 3D VISION WIRELESS GLASSES.....	154

RESUMEN

El sistema de software biomédico para el aprendizaje del profesional del sector salud puede realizar simulaciones para visualizar y manipular reconstrucciones 3-D de datos biomédicos para potenciar el aprendizaje y la formación del profesional del sector salud. El software biomédico es una aplicación de realidad virtual con usabilidad de sistemas de software biomédico, el desarrollo se realizó con tecnología de motores gráficos (GAME ENGINE), para el tema de anatomía se realizaron consultorías al profesional de la salud y poder obtener una mejor aplicación con metodologías de estructuras corporales reconstruidas en 3-D y representar las estructuras complejas de baja densidad.

El objetivo principal del sistema de software biomédico es el aprendizaje y potenciar las habilidades de los estudiantes, educadores, académicos, clínicos e investigadores que se encuentran envueltos en el sector de la salud. El sistema de software biomédico se puede tener acceso a una amplia base de datos de estructuras de modelos biomédico 3-D de la anatomía humana (cara humana, cráneo, cerebro, tronco cerebral y carótida) obtenidos por tomografía computarizada y reconstrucción 3-D geométrica, además de estructuras corporales modeladas en 3-D que son difíciles de reconstruir en 3-D por su baja densidad (nervio trigémino y nervio ocular) consideradas como estructuras corporales no biomédicas, ya que no son datos reales.

El sistema de software biomédico tiene un módulo de visualización de tecnología estereoscópica 3-D pasiva y activa que permite una mejor visualización, tener la sensación de profundidad e interactuar con las estructuras de una forma inmersiva de una forma fácil e intuitiva. El sistema de software biomédico no solo cuenta con estos módulos, tiene una gran variedad de módulos para que la experiencia del aprendizaje sea completa al utilizarlo. A continuación se muestra una lista de los módulos del sistema de software biomédico:

1. Módulo de Modelos Biomédicos.

- Función de Reconstrucción 3-D de Piel.
- Función de Reconstrucción 3-D de Cabeza.
- Función de Reconstrucción 3-D de Cara.
- Función de Reconstrucción 3-D de Hiodes.
- Función de Reconstrucción 3-D de Cerebro.
- Función de Reconstrucción 3-D de Tallo.
- Función de Reconstrucción 3-D de Carótida.

2. Módulo de Modelos No Biomédicos.

- Función de Modelado 3-D de Rama Oftálmica.
- Función de Modelado 3-D de Rama Maxilar.
- Función de Modelado 3-D de Rama Mandibular.
- Función de Modelado 3-D de Sistema Oftálmico.
- Función de Modelado 3-D de Sistema Óseo.

3. Módulo de Estereoscópica Pasiva.

- Función de Gafas Anáglifo (Rojo/Cyan).
- Función de Gafas Anáglifo (Rojo/Verde).
- Función de Angulo de Ojos (grados).
- Función de Distancia entre Ojos (u).
- Función de Delta X de Imagen.
- Función de Delta Y de Imagen.

4. Módulo de Estereoscópica Activa.

- Función de Gafas 3-D NVIDIA VISION.
- Función de Profundidad.
- Función de Profundidad.
- Función de Visor de Rayos Laser.
- Función de Compatibilidad de Software.
- Función de Convergencia.
- Función de Volumen de Visualización.

5. Módulo de Información de Modelos.

- Función de Terminología de las Estructuras Anatómicas.
- Función de Reproducción de Video.
- Función de Información Teórica.

6. Módulo de Cortes Anatómicos.

- Función de Plano Sagital Izquierdo.
- Función de Plano Sagital Derecho.
- Función de Plano (Axial) Superior.
- Función de Plano (Axial) Inferior.
- Función de Plano (Frontal).
- Función de Plano (Posterior).

7. Módulo de Interactividad Visual.

- Función de Cámara de Inspección.
- Función de Rotación Automática.
- Función de Zoom de Efecto Microscópico.
- Función de Cámara de Navegación.
- Función de Ubicación Anatómica.
- Función de Flechas de Control.

El sistema de software biomédico es una solución al problema de proporcionar cadáveres de aprendizaje utilizado en la enseñanza y en la clínica del sector salud. La mezcla de las profesiones como la ingeniería (aplicaciones matemáticas y tecnologías de la información) y la medicina (consultoría profesional especializada del sector salud para disponer de los conocimientos fundamentales) a un nivel multidisciplinario, se logró el desarrollo del sistema de software biomédico, para la fiabilidad requerida.

El sistema de software biomédico se desarrolló en la Universidad Nacional Autónoma de México (D.F, Ciudad de México) en la Facultad de Ingeniería, la necesidad e impacto de realizar el sistema de software biomédico es para optimizar los recursos para la enseñanza de la anatomía y la fisiología humana, solo por mencionar algunos ejemplos, pero incluso con mayor alcance. El sistema de software biomédico está dirigido a gran parte de los profesionales del sector salud que requiera potencializar el conocimiento de la anatomía humana o interés en el aprendizaje, observación y comprensión de la función anatómica, sin embargo el alcance del sistema de software biomédico es prácticamente ilimitada, ya que se puede realizar el análisis de las estructuras corporales y poder realizar recorridos dentro de las estructuras para planificar cirugías futuras, así como un corredor de la fórmula 1 se entrena en las carreteras virtuales antes de participar en la real, los estudiantes podrían utilizar el sistema de software biomédico como simulador de pacientes para capacitación de alguna patología corporal.

INTRODUCCIÓN

La ingeniería en computación ha producido avances tecnológicos en diferentes áreas profesionales. No obstante la computadora no puede hacer absolutamente nada sin un *software*, es decir, sin un programa que le diga qué es lo que tiene que hacer. Por lo general, un *software* consiste en diversos programas independientes, entre los cuales se encuentran archivos de configuración que se utilizan para ejecutar estos programas, además de un sistema de documentación, que describe la estructura, para que el usuario pueda potencializar el uso del mismo, y de un sitio web que permita a los usuarios descargar la información de versiones recientemente liberadas. Los avances en desarrollo de software evolucionaron de solo poder procesar datos informáticos a procesar imágenes generadas por computadora, había llegado la era de la computación gráfica. Las imágenes generadas por computadora (CGI, Computer Generated Imagery) tomaron al mundo por sorpresa, el uso de esta tecnología se visualizaba como ciencia ficción, pero desde las películas de Hollywood que muestran personajes completamente desarrollados por la imaginación, planetas y entornos desconocidos, todo basado desde la creatividad humana o con la implementación de teorías científicas, fue posible la creación de estos mundos de fantasía y de ciencia ficción. Las imágenes generadas por computadora dejaron de ser ciencia ficción para convertirse en una realidad, la implementación de esta tecnología no sólo se empezó a utilizar para el entretenimiento, sino también como herramienta para resolver casos reales, ya que puede ayudar a visualizar los casos de una forma gráfica y poder hacer una mejor toma de decisiones, algunos de los casos reales donde se empezó a estudiar la posibilidad de usar esta tecnología eran las reconstrucciones de estudios de casos clínicos patológicos.

El proyecto humano visible es el primer intento en la historia para visualizar la anatomía humana a base de imágenes generadas por computadora, el objetivo del proyecto fue obtener por primera vez información real de un cuerpo humano. Para obtener los datos del proyecto humano visible, se creó un conjunto de datos de imágenes digitales obtenidas de los cuerpos de un hombre y de una mujer. El conjunto de datos consiste en imágenes axiales de la cabeza y el cuello con intervalos de 4 mm y secciones longitudinales de todo el cuerpo, también a intervalos de 4mm. La resolución de la imagen por resonancia magnética (MRI, Magnetic Resonance Image) es de 256 píxeles por 256 píxeles. Cada pixel tiene 12 bits de tono gris. Los datos consisten en la tomografía axial computarizada de todo el cuerpo tomadas a intervalos de 1 mm con una resolución de 512 píxeles por 512 píxeles, donde cada píxel se compone de 12 bits de tono gris. Las imágenes anatómicas axiales son de 2048 píxeles por 1216 píxeles en cada pixel definido por 24 bits de color, cada imagen consta de alrededor de 7.5 MB de datos, las secciones transversales también son anatómicas a un 1mm y coinciden con las imágenes axiales para cada modo. Algunos de los archivos que se utilizan en el proceso de obtención de información son [1]:

- La **radiología** ayuda a los profesionales de la salud a ver estructuras del cuerpo. Los médicos que se especializan en la interpretación de estas imágenes se denominan radiólogos. Mediante estas imágenes, el radiólogo u otros médicos con frecuencia puede diagnosticar la causa de síntomas, vigilar que tan bien está respondiendo el cuerpo a un tratamiento que reciba por alguna enfermedad o afección, detectar enfermedades.
- **Tomografía computarizada (TC)**, también conocida como TAC (tomografía axial computarizada), que incluye angiotomografía computarizada.
- Resonancia magnética (RM) y angiografía por resonancia magnética (ARM).
- Radiografías simples, que incluyen radiografía de tórax.
- **Tomografía por emisión de positrones**, también llamada imágenes por TEP o gammagrafía por TEP.
- **Full Color:** Las imágenes originales tienen una resolución de 0.33mm XY, en un formato original y con instrucciones de como abrir estos archivos.
- **70mm:** Imágenes con resolución de 0.17mm XY derivada del análisis de la película original de 70mm en un comprensión sin pérdidas en "Z" en un formato RGB y con las instrucciones sobre cómo abrir estos archivos.
- **PNG:** Las imágenes radiológicas originales e imágenes en color sin pérdidas son directamente legible en el formato PNG y los archivos de texto de la resonancia magnética "MR" y las tomografías computarizadas "CT" son asociadas en las cabeceras.

Para obtener los archivos PNG de información se recomienda, que cada archivo contenga una sola imagen en formato PNG (Portable Network Graphics). Las imágenes anatómicas aparecen directamente cuando se abre el visor. Las imágenes radiológicas (CT, MRI y X-Rays) aparecerán como una pantalla negra. Estas imágenes son de 16 bits en escala de grises. Porque en realidad, es sólo hasta 10 bits de datos en cada pixel, las imágenes parecen ser negras cuando se carga inicialmente en el visor. Para ver las imágenes correctamente, use el ajuste automático de la pantalla o ajustar manualmente el nivel de gamma. La imagen de la TC, MRI o X-Rays serán ahora visibles. [1]

Las imágenes generadas por computadora evolucionaron por el desarrollo e implementación de interfaces graficas de usuario (GUI, Graphics User Interface), con estos sistemas de interfaces graficas de usuario, se logró integrar no sólo las imágenes generadas por computadora, sino también funciones para cargar una secuencia de imágenes con un control grafico para manipularlas, también se implantaron las funciones básicas para interactuar con estos sistemas.

El prototipo Explorador Humano Visible (VHE, Visible Human Explorer) es una interfaz de usuario para navegar por la biblioteca Nacional de Medicina donde están los datos del Humano Visible y demostrar una interfaz de usuario general para la exploración del volumen. La interfaz permite a los usuarios navegar en un humano visible de volumen miniatura, localizar imágenes de interés y recuperar automáticamente las imágenes de resolución completa deseadas a través de internet desde los archivos de la NLM. El prototipo puede también navegar por las imágenes que se han descargado previamente en la maquina local. EL VHE es un programa de trabajo y puede ayudar mucho en la navegación y la recuperación de las imágenes humanas visibles remotas y locales. [1]

En la *figura I.1* se muestra el VHE que puede recuperar y mostrar el humano visible como una herramienta de escaneo con las tomografías computarizada “CT”.

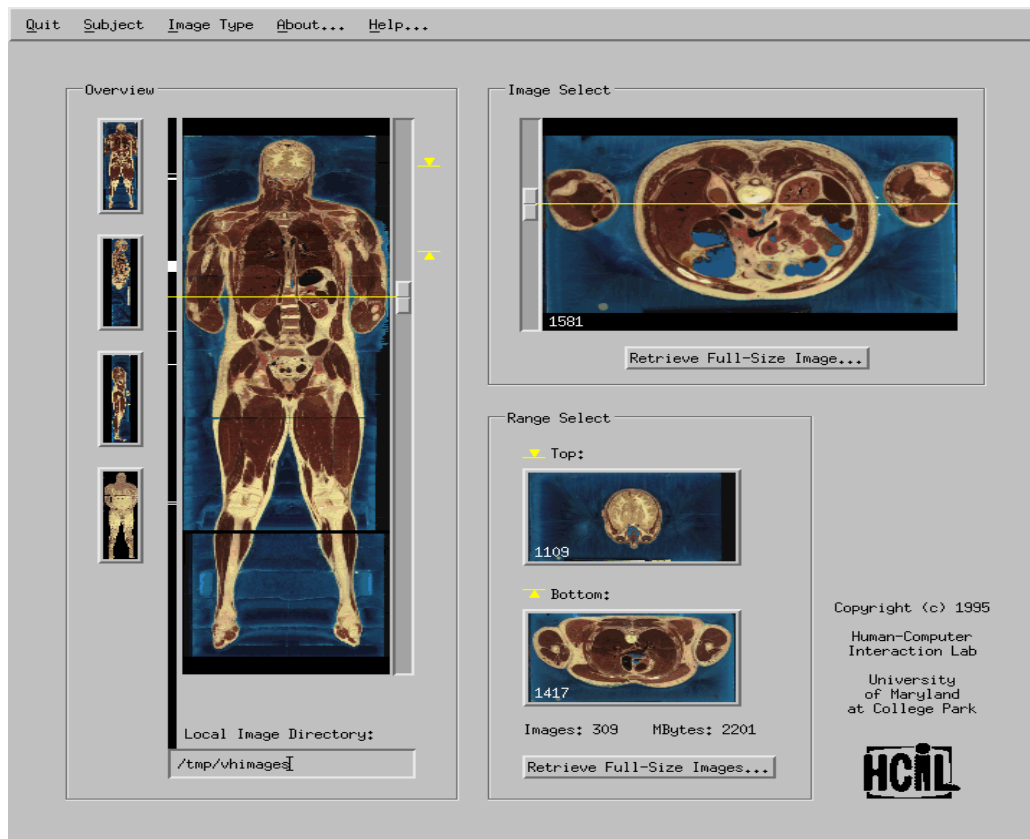


Figura I.1 VHE GUI.

La siguiente evolución de las imágenes generadas por computadora fue dejar de ser solo una secuencia de imágenes por segundo, y convertirse en objetos o gráficos computarizados de una realidad virtual, gracias al desarrollo e implementación de las tecnologías de realidad virtual, donde se puede integrar no solo secuencias de imágenes generadas por computadora, sino también la implementación de múltiples funciones multimedia (Información, Sonido, Imagen, Animación, Video e interactividad), además de poder desarrollar software de realidad virtual con interfaces gráficas de usuario, se hace posible visualizar, interactuar, navegar y modificar las propiedades o atributos físicos de los objetos gráficos o modelos tridimensionales. La implementación de la realidad virtual fue el primer paso de la evolución en los gráficos computarizados, con el desarrollo de plataformas como los motores gráficos, el desarrollo de software gráfico, ahora es posible visualizar formas más complejas y con mejor interacción, gracias a estas nuevas plataformas se pueden tener simuladores que ayuden a mejorar las capacidades humanas como la destreza, aprendizaje, rapidez en solución de problemas y coordinación de toma de decisiones. El desarrollo e implementación de tecnologías en los sectores profesionales y el análisis del área de oportunidad por el gran desaprovechamiento de herramientas tecnológicas computacionales, con la aplicación para potencializar la enseñanza y la formación docente en el área de la salud y la ciencia, se abordará a través de desarrollar y diseñar un software que pueda realizar simulaciones para visualizar, manipular y comprender los datos biomédicos y mejorar la formación y el aprendizaje del profesional del sector salud. Para los efectos de la instrucción asistida por computadora en la enseñanza de la anatomía humana, la información del aprendizaje en las universidades de anatomía humana o la anatomía combinada y cursos de fisiología requieren que los estudiantes utilicen al menos dos habilidades diferentes de aprendizaje [2]:

- a) La adquisición de vocabulario técnico amplio y complejo.
- b) El desarrollo de la capacidad de interpretar y comprender las relaciones 3-D dentro de un cuerpo humano.

La instrucción asistida por computadora (CAI, computer-assisted instruction) puede mejorar el aprendizaje y el logro de puntuaciones globales en una variedad de áreas temáticas y de grupos de edad.

- a) Aumenta la comprensión del estudiante y la interpretación de las relaciones estructurales 3D.
- b) Aumenta el interés de los estudiantes en los cursos de la anatomía humana.
- c) Aumenta la retención de la información anatómica de los estudiantes a nivel de pregrado.

La hipótesis es que la enseñanza asistida por computadora mejora la capacidad del alumno para comprender e interpretar las relaciones estructurales 3D, determinar si la instrucción asistida por computadora utilizada aumenta el interés del estudiante en la materia. Sin embargo, los altos niveles de interés del estudiante en el inicio de estos cursos pueden dificultar esta determinación. La instrucción asistida por computadora es un método eficaz para la enseñanza de la anatomía o anatomía y fisiología, en la *figura 1.2* se muestra un grupo de estudiantes cursando la materia de anatomía. El principal efecto es también importante con respecto a la prueba de los materiales esenciales de aprendizaje de los estudiantes. [2]



Figura I.2 Cyber-Anatomy Operations Manager University of Iowa.

La instrucción asistida por computadora podrá mejorar las oportunidades humanas en las diversas áreas de implementación, el área donde se tiene como objetivo mejorar estas oportunidades, es el sector salud o área de la salud médica, con la implementación de la instrucción asistida por computadora será posible potencializar el aprendizaje, por medio de la visualización e interactividad de objetos tridimensionales de datos biomédicos para el sector salud, para poder visualizar este tipo de información biomédica existen varios métodos como :

- a) **Modelos 3-D de datos biomédicos** (obtenidos por reconstrucción geométrica tridimensional por tomografía axial computarizada “TAC”). En la *figura 1.3* se muestra el proceso para obtener los datos biomédicos y la construcción 3-D.

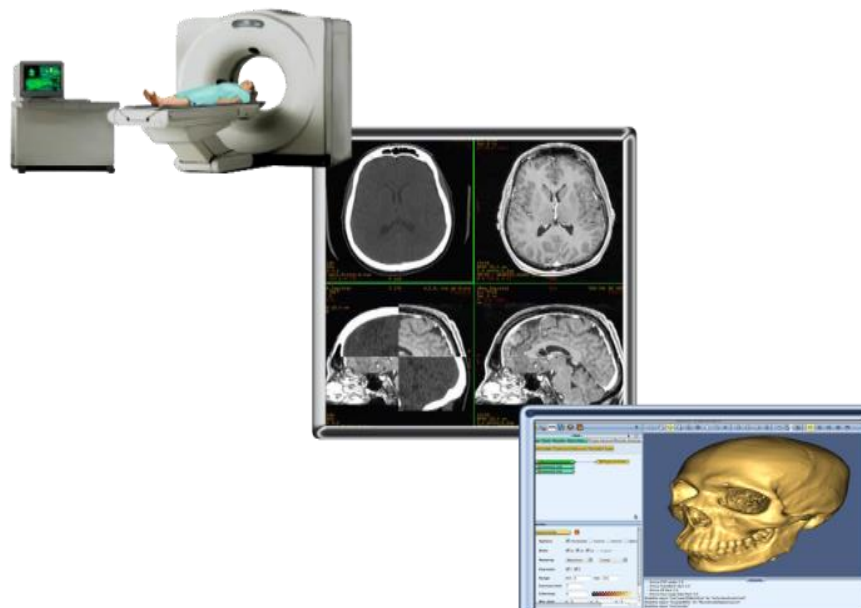


Figura I.3 Modelos tridimensionales de datos biomédicos.

- b) **Modelos 3-D de datos no biomédicos** (obtenidos de modelado geométrico tridimensional). En la figura I.4 se muestra el proceso para obtener los datos no biomédicos con modelado 3-D.

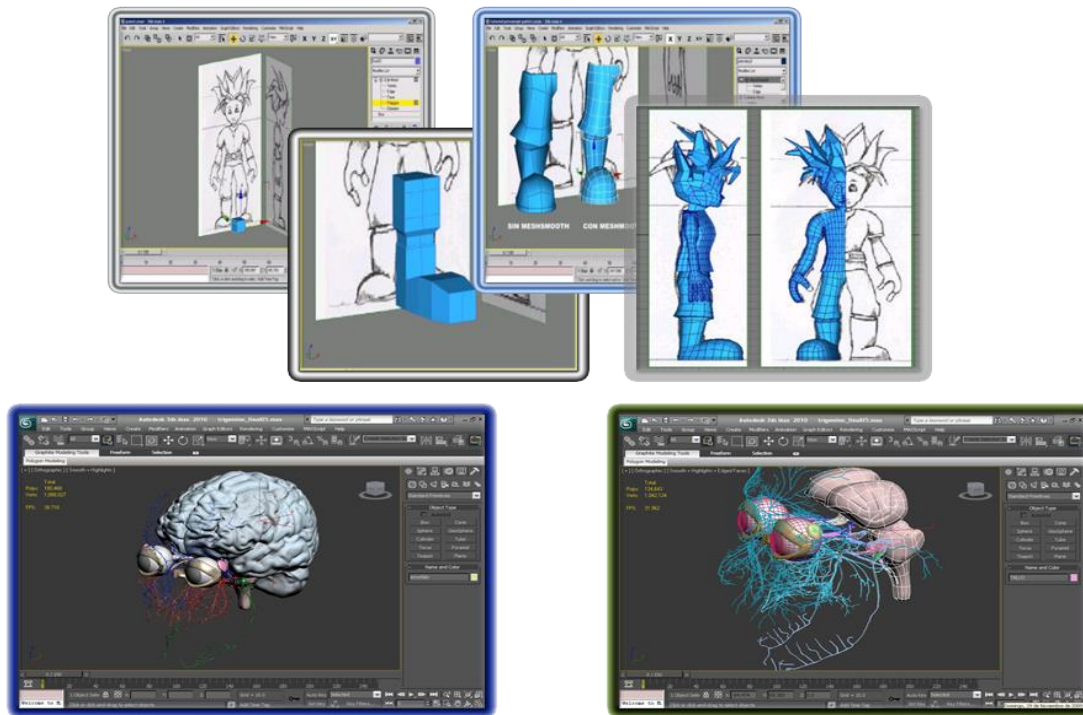


Figura I.4 Modelos digitales de datos no biomédicos.

La instrucción asistida por computadora implementada en un sistema de realidad virtual podrá contar con los servicios necesarios para ofrecer a cualquier profesión, pensemos en un:

- “**Arquitecto**” que no necesitará que sus clientes se imaginen las edificaciones o que solo vean las maquetas, sino que les sea posible "caminar" por las construcciones que aún no existen en el mundo real, observando las estructuras del edificio con una perspectiva de dimensiones reales, mientras es posible realizar modificaciones y mejoras de acuerdo a sus sugerencias.
- “**Médicos**” que pueden analizar y planear con anticipación cirugías completas y estudiantes de medicina que se entrenan con pacientes virtuales y situaciones simuladas en hospitales.
- “**Ingenieros y diseñadores**” que pueden construir y trabajar con prototipos virtuales, mientras prueban su funcionamiento antes de fabricarlos.
- “**Artistas**”, por otro lado, encuentran en este tipo de implementación tecnológica un medio de expresión novedoso para crear no sólo esculturas y pinturas estáticas, sino desarrollar experiencias dinámicas que combinan los mejores elementos de otras manifestaciones artísticas, incluyendo cine, teatro y danza.

En el ámbito empresarial será posible capacitar y entrenar al personal de las empresas mediante procesos laborales simulados, reduciendo los costos de capacitación y optimizando los métodos de trabajo. Gracias a los avances tecnológicos es posible interactuar con reacciones químicas, comportamientos físicos y muchos otros procesos, todos ellos a base de simulaciones que ayudan a potencializar el aprendizaje humano. La evolución de la tecnología y la implementación impactarán la forma como se gestiona el conocimiento, logrando una proyección visual diferente y una prioridad de implementación de las tecnologías en el área del conocimiento. La tecnología ha llegado para cambiar nuestra forma de vida y nuestras profesiones. La implementación de las tecnologías deben ser una prioridad en nuestro entorno de comunidades sociales, en México se tiene mucho trabajo para poder alcanzar a las grandes potencias mundiales y poder ubicarnos en un buen nivel de gestión tecnológica basada con investigación e implementación de innovación para la solución de los problemas y necesidades sociales.

CAPÍTULO I

REALIDAD VIRTUAL

1. REALIDAD VIRTUAL

Las nuevas tecnologías, en particular, la realidad virtual, tendrán un impacto importante en el cuidado de la salud en las próximas décadas. Clínicamente validado, los simuladores médicos poderosos ya están disponibles y en uso en todo el mundo. La cirugía general es líder en el uso de simuladores, la neurocirugía se dirige al uso de éstos y también la cirugía guiada por imágenes. Otras aplicaciones de realidad virtual están siendo utilizadas en la salud mental, anestesia y medicina de urgencias. La rápida evolución de internet y de “e-Learning” son dominios que han acelerado la difusión de técnicas de simulación, imágenes 3D interactivas y cursos estructurados, la amplia difusión está describiendo de una forma masiva a la realidad virtual en la formación quirúrgica, en la planificación y la ejecución de los procedimientos en el aula, a tal grado que se discute el futuro a corto plazo de esta nueva tecnología. Un simulador médico exitoso o sistema quirúrgico basado en la realidad virtual requiere la participación de un equipo de especialistas, incluyendo especialistas en ergonomía y psicología aplicada, ingeniería de software y diseño digital 3D. En 1990, los equipos de investigación, en particular en la Universidad del Norte Carolina y en el Departamento de Defensa de EE.UU., desarrollaron el concepto de los cirujanos del futuro equipados con auriculares de realidad virtual y poder ensayar procedimientos reales utilizando imágenes generadas por computadora. La realidad virtual es una tecnología que nos permitirá planificar procesos quirúrgicos e implementarlos en la práctica, esta metodología permitirá estudiar el resultado obtenido antes de operar al paciente, como por ejemplo en la reconstrucción mamaria, cirugía correctiva maxilofacial, etc. La implementación de la tecnología de realidad virtual, se lograría obtener como resultado la optimización de los métodos quirúrgicos con mejor calidad de servicio en los pacientes. Por tal razón la realidad virtual tiene el objetivo de capacitar al profesional de la salud para un futuro mejorar en las planificaciones quirúrgicas. [10]

Para los cirujanos en formación existe una diferencia sustancial entre tejidos artificiales o tejidos animados y los procedimientos supervisados en pacientes de quirófano, con todas las presiones conocidas, tales como las restricciones de tiempo que gobierna la clínica. Una amplia gama de sistemas de formación de realidad virtual, se han desarrollado, pero no todos son ampliamente disponibles todavía. Un simulador disponible en el mercado para la venopunción con retroalimentación de fuerza para simular la sensación de la cánula en el momento que penetra la piel y las venas. Esto es adecuado para la formación de enfermeras, estudiantes de medicina, flebotomistas y paramédicos. Simuladores más complejos para gastroscopia terapéutica, colangiopancreatografía retrógrada endoscópica y procedimientos endoscópicos están disponibles para los alumnos en gastroenterología. Un simulador radiológico proporciona formación en el cateterismo cardíaco y angiográfico, con el modelado en tiempo real de los parámetros fisiológicos y flujo de sangre. Estos simuladores de realidad virtual ofrecen poder realizar en varias repeticiones los procesos, registrar algunos datos, información para la formación y a menudo sin la necesidad de supervisión de ninguna persona, se puede llevar a cabo todos los procesos necesarios que se requieran simular. Además, algunos simuladores de realidad virtual interactiva que se han desarrollado para procedimientos como la punción lumbar y el drenaje ventricular del cerebro están libremente disponibles para su uso a través de la World Wide Web. Los desafíos para introducir la realidad virtual a la práctica quirúrgica habitual, poco se sabe acerca de la alfabetización informática del profesional de la salud, pero la ansiedad del equipo y la alineación son problemas reales y profesionales de la salud, los profesionales de la salud mayores de 30 años son considerados como la "generación perdida" con respecto a la IT (Tecnologías de la Información). Se está haciendo un importante esfuerzo para introducir la informática en salud a los médicos de los hospitales, ya que se convertirá en la calificación de referencia para futuros puestos de trabajo relacionados con la IT. La amplia aceptación de la cirugía de alta gama tecnológica, como el uso de vídeos en los procedimientos laparoscópicos, ha superado muchos prejuicios quirúrgicos, pero la realidad virtual está siendo vista como trucos o potencialmente peligrosos por la mayoría de los cirujanos. Se requerirán equipos más multidisciplinarios para desarrollar el uso de estas nuevas tecnologías en cirugía. Al igual que con tantos avances en el cuidado de la salud, sin embargo, estas nuevas tecnologías es probable que aumenten los costos, en contra de lo que debe ser la posibilidad de mejorar la capacidad quirúrgica y error médico reducido, con una reducción de la morbilidad y la mortalidad. La evidencia de estos beneficios es probable que tome al menos 5 a 10 años. [10]

1.1 ¿Qué es realidad virtual?

La realidad virtual se describe mejor como un conjunto de tecnologías que permiten a las personas interactuar de manera eficiente con bases de datos informáticas que contienen objetos en forma 3-D en tiempo real utilizando sus sentidos y habilidades naturales. Esta definición evita cualquier referencia a la necesidad de pantallas montadas en la cabeza y la ropa instrumentado como guantes o trajes, así como la tendencia a finales de 1980 y principios de 1990. A pesar de este llamado de la tecnología de inmersión es aún evidente que hoy en día, sólo el 10% de las aplicaciones de realidad virtual garantiza su uso. El punto fuerte de la realidad virtual, ya sea en el diseño o la formación, es que soporta y mejora la interacción en tiempo real por parte del usuario. La realidad virtual en la formación quirúrgica es evidente, la formación quirúrgica tiene un gran costo económico, y la influencia de los sistemas de formación son más accesibles económicamente y con menor tiempo de trabajo, cada vez hay más demanda de alumnos que quieren crecer profesionalmente en la experiencia quirúrgica y la deben obtener fuera de la sala de operaciones. El uso de los sistemas con realidad virtual será un factor crucial para determinar el éxito o fracaso de estos sistemas por parte de los cirujanos en capacitación, la demostración de la calidad del aprendizaje será respaldada por las evaluaciones de formación académica. La evidencia reciente sugiere que este es el caso. La comunidad de la realidad virtual no sólo ha producido sistemas validados experimentalmente para la formación y evaluación de habilidades quirúrgicas, se ha hecho uso de técnicas establecidas que ahora se están convirtiendo reconocidas por las normas internacionales, como la Organización Internacional para la ISO 9241-210 "diseño centrado en el humano de normalización de sistemas interactivos". [10]

1.1.1 Características de la realidad virtual

La realidad virtual, debe ser coordinada o generada por medio de un *software que sea* suficientemente poderoso para sincronizar en *tiempo real*. La finalidad es que el usuario perciba al ambiente virtual de forma similar al de la vida real, en ese sentido, la computadora con el *software* deberán ser capaces de desplegar imágenes de los espacios 3-D a una velocidad de por lo menos 30 cuadros por segundo (*FPS, Frame Per Second*). Por otro lado, para sincronizar los movimientos del usuario con los dispositivos, éstos deben ser muestreados con rapidez y exactitud, de tal manera que se produzca una respuesta rápida y confiable, de acuerdo con la interacción realizada. [12]

La creación de *software* con ambientes virtuales inmersivos conlleva al manejo de elementos complejos, sobre todo por el hecho de que es un área de continua investigación y desarrollo que busca integrar nuevos elementos con el fin de crear ambientes virtuales más reales y creíbles. Podemos mencionar, por ejemplo, la integración de agentes inteligentes con los que el usuario puede interactuar; nuevas formas de cómputo para el manejo del *software*, como arreglo de PCs y compartir ambientes virtuales desde sitios remotos en espacios de colaboración. Actualmente, se investigan nuevas interfaces de interacción más naturales mediante cámaras de video que captan los movimientos del usuario donde ya no es necesario ponerse encima algún dispositivo. [12]

1.1.2 Implementación de la realidad virtual.

Las imágenes 3-D generadas por computadora (CGI, Computer Generated Imagery 3-D) proporciona una de las herramientas y técnicas para crear escenarios virtuales y personajes en el campo de los efectos visuales del cine, películas de animación y video juegos. En contraste, la formación de imágenes por resonancia magnética clínica (RM) proporciona al médico una ventana en el interior del cuerpo del paciente a través de la instrumentación científica. Superficialmente, estos campos profesionales no se comunican y tienen poco en común: CGI 3-D se utiliza para atraer a una audiencia masiva a través del entretenimiento popular, mientras que los usos clínicos de RM un lenguaje restringido, el de la ciencia reduccionista, en la búsqueda de la observación anatómica o patológica y diagnóstico médico. [5]

La tecnología de resonancia magnética (MRI) fue desarrollada en la década de 1970 y entró en servicio clínico en la década de 1980. Esta técnica de imagen es relativamente nueva y produce imágenes transversales monocromáticas del cuerpo humano que proporcionan información de diagnóstico. Hoy en día, el uso de la RM está muy extendida en la asistencia sanitaria moderna, particularmente en los países de primer mundo. En 2003, había aproximadamente 10,000 unidades de MRI en todo el mundo, y aproximadamente 75 millones de imágenes por resonancia magnética efectuados al año. Del mismo modo, desde su creación en la década de 1970, las imágenes 3D generadas por computadora (CGI 3-D) ha visto una creciente exponencial en el desarrollo de *software* y hardware. Su origen está en la investigación espacial militar estadounidense y, con técnicas pioneras tempranas en la animación por computadora en 3-D. [5]

Hoy en día los CGI 3-D tienen acceso a *software* de animación de escritorio y hardware. En un contexto contemporáneo, las tecnologías como Maya y 3ds Max desarrollados por la empresa Autodesk con sede en Estados Unidos de América, ofrecen realizar animaciones, poder realizar modelado geométrico tridimensional con varias herramientas de renderizado y *software* de animación de personajes. Maya y 3ds Max es un paquete de *software* robusto utilizado internacionalmente para realizar películas, comerciales de televisión y desarrollo de video juegos (<http://usa.autodesk.com/>). Estas herramientas permiten a los CGI 3-D poder construir y animar mundos virtuales para el cine, televisión y video juegos. Mientras que el cine y los video juegos son el mercado dominante para CGI 3-D hay otros usos. Estos incluyen visualización de la arquitectura, visualización de productos de ingeniería y el campo de la simulación (como los entrenamientos de vuelo y aplicaciones militares). [5]

Estos mundos virtuales se componen de caracteres digitales, accesorios y bienes, que pueden combinarse en una acción en vivo de la filmación de la película con CGI 3-D. Las herramientas que utilizan, que hace tan solo dos décadas eran del dominio exclusivo de los grandes estudios de cine de Hollywood, ahora se han filtrado a través del camino de los CGI 3-D que trabajan en animaciones médicas. [5]

La animación médica 3D y las funciones de visualización en tres áreas actuales contemporáneas son:

1. La animación biomédica
2. La documentación biomédica
3. La visualización 3-D biomédica.

De estos tres, dos residen dentro de las artes, y uno en las ciencias.

En términos generales, el CGI 3-D que trabaja en la animación biomédica se dedica a una amplia gama de actividades, como la educación para profesionales de la salud, la formación de profesionales médicos forenses, la reconstrucción forense y la presentación farmacéutica de drogas. Del mismo modo, los documentales basados en programas de televisión que ofrecen documentales series o programas con temas biomédicos que proporcionan una diferente forma de educar a un público masivo para que se involucre en las ciencias de la salud a través de documentales. Ejemplos recientes de estos documentales son *Fight for Life* (BBC, 2007) y *la vida antes del nacimiento* (MacDonald, 2005), transmitido en la televisión británica. En contraste, la visualización médica 3D proporciona al profesional médico imágenes 3-D que se generan por la instrumentación de exploración médica, tales como CT y MRI, para el diagnóstico médico, la formación profesional y la educación médica. Estas imágenes se utilizan en áreas tales como la endoscopia virtual y la planificación quirúrgica. [5]

La visualización 3-D en la medicina, utiliza tecnologías 3-D para lograr una visualización aparentemente cercana a la realidad y científicamente precisa a los datos clínicos. Esta instrumentación médica actualmente está muy avanzada. Ya que se pueden capturar conjuntos complejos de datos biomédicos y poder reconstruir para mostrar una visualización 3-D con detalle de los órganos internos y tejidos del cuerpo humano, de manera que no era posible esto anteriormente. La imagen está protegida por medio de un encapsulamiento, en una archivo de compresión de imagen, que solo puede ser decodificada con un conocimiento considerable en el área biomédica y radiológica. La visualización 3-D de la medicina, es, no obstante, específica del paciente, ya que puede ofrecer imágenes altamente detalladas del interior del cuerpo humano que no son representaciones genéricas. El objetivo de esta tecnología es valorar la combinación de datos de resonancia magnética con enfoques interpretativos artesanales digitales 3-D en el campo de la visualización de datos clínicos y su aplicación potencial en el mundo real. [5]

La evolución tecnológica en el procesamiento de datos, ha permitido que la visualización se desarrolló con mejores resultados, logrando procesar datos de resonancia magnética (MRI) que pueden ser post-procesados en imágenes basadas en pantallas 3-D para lograr una reconstrucción 3-D de datos biomédicos. La tecnología de diagnóstico 3-D varía actualmente entre los sistemas que integran la resonancia magnética y hardware para escaneo de tomografía computarizada (CT). En la *figura. 1.1* es un ejemplo de una visualización del volumen 3-D de post-procesado compilado, de datos de la sección transversal de resonancia magnética funcional clínicos 2-D de la cara y el cerebro de un paciente. [5]

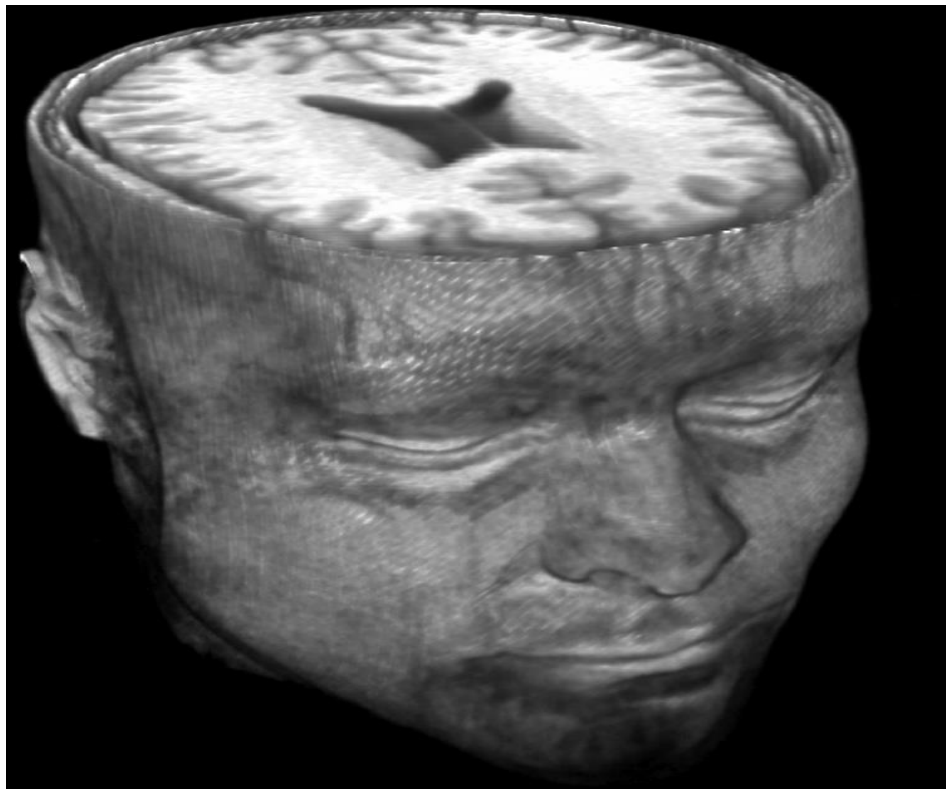


Figura 1.1 Visualización 3-D de la cara y el cerebro de un paciente sobre el software OsiriX, derivado de datos de MRI adquiridos en el Hospital Ninewells, Dundee, Reino Unido.[5]

1.1.3 Estructura del software de realidad virtual para visualización CGI 3-D

La estructura de un software de visualización 3-D es una estructura más compleja que un software de datos, ya que constituye en entornos 3-D que se extiende en múltiples capacidades, mientras que posee una programación específica para el funcionamiento y control de múltiples dispositivos externos, debe ejecutar y modificar propiedades internas para simular comportamientos en tiempo real. Las características de un software de visualización 3-D son una gama sofisticada que se divide en varios procesos principales como: el desarrollo de la aplicación para el diseño de los escenarios 3-D, incorporación de audio, comportamiento de las simulaciones, configuración de los dispositivos, navegación interactiva y manipulación del entorno virtual. [6] La estructura de un software de visualización 3-D propone la creación de un híbrido que comprende el enfoque artesanal CGI 3D con datos clínicos. El acto de crear visualizaciones de ordenador y animación en 3-D sobre la base de datos de la exploración médica, requiere una cantidad significativa de experimentación técnica antes de cualquier intervención creativa. En este caso, el formato de archivo médico DICOM almacena datos de corte transversal de dos dimensiones (2-D), junto con la información detallada sobre el nombre, la edad del paciente, el nombre del escáner y el tipo, entre otros detalles. El objetivo es vincular los datos de imágenes por resonancia magnética con el software de animación Maya o 3ds Max. Exportación de los datos científicos en un paquete de software CGI 3-D que permita al desarrollador aumentar los datos. Durante la fase de experimentación temprana, se estudiaron dos técnicas: la primera fue la reconstrucción del volumen 3-D y el segundo fue "apilamiento de cubos". Ambas técnicas se utilizan comúnmente en visualización médica como un método de post-procesamiento. [5]

La reconstrucción de volumen 3-D es la técnica predominante para la reconstrucción 3-D de la medicina. Esta técnica de reconstrucción de volumen 3-D utiliza píxeles (la unidad más pequeña utilizada para construir una imagen digital) y vóxeles (un pixel de 3-D se utiliza para representar un volumen en pantalla) extraídos de los datos del corte de imagen 2-D para crear una representación tridimensional visual en pantalla de los datos biomédicos. Se trata de un enfoque relativamente rápido y menos intensivo de la unidad central de procesamiento (CPU, Central Processing Unit) que el algoritmo de apilamiento de cubos. El uso de píxeles y vóxeles, no obstante, limita actualmente el número de atributos que pueden ser añadidos por el artista digital. En última instancia, lo que deja menos margen para el trabajo detallado de animación. [5]

La segunda opción para la reconstrucción 3-D es la geometría de malla basada en el algoritmo de apilamiento de cubos. Construido a partir de rodajas de exploración 2-D, el algoritmo del software genera parches de superficie, que luego se acumulan en malla, formando de ese modo la forma 3-D de un objeto. El proceso es más complejo que el volumen de la reconstrucción en 3-D, lo que requiere más recursos computacionales para la construcción de la malla. Como los ordenadores se vuelvan cada vez más potentes y asequibles, sin embargo, este proceso de reconstrucción de polígonos tiene varias ventajas convincentes, la flexibilidad de ser el más fuerte, como la geometría de la superficie permite posteriormente hacer la exportación a otras aplicaciones. [5] En la *figura 1.2* se muestra las etapas para la generación de modelos tridimensionales basadas de resonancia magnética.

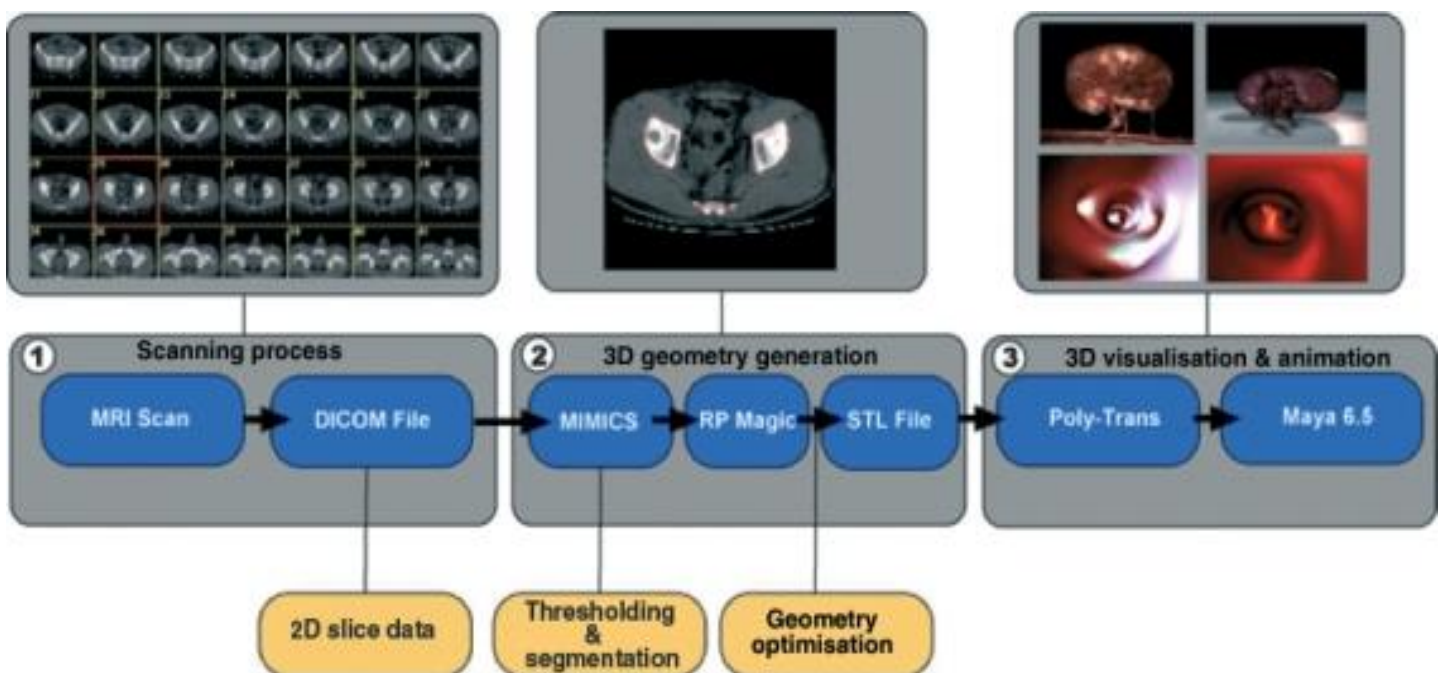


Figure 1.2 Este diagrama ilustra las etapas en escena desarrollada durante la investigación sobre exportación de datos de la resonancia magnética clínica prima hasta la visualización final de 3-D.

1.2 Características principales de los entornos de realidad virtual

La realidad virtual ha derivado en algunas ramas de estudio, entre las que encontramos [12]:

- La realidad inmersiva o intrusiva.
- La realidad proyectada.
- La realidad ampliada o superpuesta.
- y la tele presencia.

1.2.1 Ambientes virtuales inmersivos

La creación de aplicaciones tales como:

- visualizadores para ambientes arquitectónicos,
- visualizadores para tratamiento de fobias en psicología
- visualizadores para enseñanza de medicina

Son espacios inmersivos tridimensionales reales o imaginarios generados por computadora con los que el usuario puede interactuar y que le producen la sensación de estar dentro. La sensación de presencia se genera al integrar varios elementos como son una rápida generación de varias imágenes de alta calidad por segundo desplegadas en un área que cubra un amplio grado de campo de visión del usuario y que son resultado cuando el usuario interactúa al moverse o modificar el espacio y sonido espacial relacionado con el ambiente al que se da vida. La interacción en estos ambientes se busca de la forma más natural posible, por lo que se llega a recurrir al uso de dispositivos especiales que nos permiten una manipulación natural con el ambiente como pueden ser el uso de guantes, sistemas de rastreo de movimiento o interfaces de entrada muy específico estrechamente vinculados con el ambiente en que se trabaja, por ejemplo: en los simuladores de manejo se utilizan volantes y palancas de velocidades, las simulaciones para cirugía se utilizan instrumentos según la cirugía en cuestión. [12]

Las ventajas de poder utilizar ambientes virtuales inmersivos, es poder estar en espacios inaccesibles o de riesgo y modificar los eventos que ahí ocurren, por ejemplo: recorrer libremente ambientes arquitectónicos, ya desaparecidos, diseñar edificios, casas, autos u otros objetos, teniendo una proyección en escala real y realizar modificaciones antes de pasar a la construcción real, recrear ambientes para entrenamiento que sería muy costoso como para el adiestramiento en casos de siniestros, el usuario puede interactuar permitiendo tener fallas, lo que en la vida real podría ser peligroso y de alto costo. [12]

1.2.2 Diferencias con ambientes virtuales no inmersivos

Existen varios sistemas de realidad virtual como son los creados por el lenguaje para modelado de realidad virtual (VRML, Virtual Reality Modeling Language), con alto grado de interactividad y fácil acceso. Estos mundos virtuales se les conoce como no inmersivos, ya que carecen de la sensación de presencia, la cual se logra con el uso de hardware especial de despliegue, dispositivos, audio espacial y la generación de espacios 3-D con escala 1 a 1 con el mundo real. La realidad virtual no inmersiva o también llamada realidad virtual de escritorio es aquella que se crea cuando el participante explora diversos ambientes haciendo uso de los dispositivos de hardware comunes: mouse, monitor, tarjeta de sonido, bocinas y tarjeta de video. [12]

1.2.3 Ventajas de los ambientes inmersivos sobre los no inmersivos

Las ventajas de los ambientes inmersivos virtuales se producen en la capacidad de libertad del movimiento, ya que la experiencia de interactuar produce una sensación de estar dentro del ambiente 3-D generado, además otra sensación para envolver al usuario se produce con el sonido espacial y la retroalimentación táctil en los detalles al visualizar y escala en la que se proyectan los ambientes. Por ejemplo, en los casos de manufactura y arquitectura, poder analizar los objetos en escala real, permitir tomar decisiones, realizar y observar las modificaciones en el espacio del objeto. En aplicaciones como las de psicología o en el entrenamiento médico, se debe tener un despliegue que cubre la mayor parte del campo de visión del usuario, crea la sensación de estar dentro, generando mayor impacto en las sensaciones generadas, que son importantes en esta clase de aplicaciones. [12]

1.2.4 Sistemas estereoscópicos

La alternativa más utilizada para proporcionar al espectador la sensación de imágenes en 3-D consiste en presentar las proyecciones planas de la escena que se obtendrían en dos puntos del espacio situados a una distancia parecida a la que separa los dos ojos (unos 6.5 cm). Para conseguir el efecto de 3-D, el sistema reproductor deberá presentar al ojo derecho la imagen que se ha captado con la cámara de la derecha y al ojo izquierdo la obtenida con la cámara izquierda. La mayoría de sistemas requieren que el espectador utilice algún tipo de lentes u otros elementos ópticos que actúen como filtros separados de las imágenes. Con ello, las imágenes proyectadas en la retina coincidirán con las que hubieran incidido en el espectador si hubiera estado en la posición de la cámara. De hecho, la sensación de espacio conseguida con esta técnica es muy espectacular y sus principios han venido utilizándose desde hace bastantes años. Actualmente, con la posibilidad de sintetizar imágenes mediante ordenador es posible presentar al espectador puntos de vista y ángulos de la escena que serían impensables en una situación real. [13] En la *figura 1.3* se muestra un sistema estereoscópico, este tipo de sistema la UNAM cuenta con dos salas una en el UNIVERSUM y la sala IXTLI.



Figura 1.3 Estos sistemas se utilizan actualmente en salas especiales para visualización en la UNAM como la de UNIVERSUM y la sala IXTLI.

1.3 Entrada y salida de datos para sistemas de realidad virtual inmersiva CGI 3-D.

Los datos de entrada y salida de los sistemas de realidad virtual no inmersiva son generados por dispositivos electrónicos como el ratón y el monitor. Los datos de entrada y salida de los sistemas de realidad virtual inmersiva son generados por dispositivos electrónicos como el joystick, rastreadores, pantallas estereoscópicas y gafas de obturación [7]. Los objetos 3-D son elementos que componen a los entornos virtuales, para visualizar diferentes objetos 3-D en diferentes entornos, estos objetos deben ser almacenados en registros de bases de datos, ya que para generar a los objetos se debe consultar la información que contiene la base de datos, de echo esta información que está almacenada en la base de datos, son las propiedades de los objetos para poder generar al modelo 3-D, algunas de las propiedades que se pueden almacenar son posición, tamaño, orientación y jerarquía, tales que describen acciones, métodos de iluminación, mecanismos de control y soporte de hardware. Una condición para poder visualizar a los objetos 3-D es estar dentro de la perspectiva de la cámara que visualiza al entorno virtual, de otra manera que si esta fuera del límite de la perspectiva, no será posible visualizarlo.

La usabilidad de un sistema de realidad virtual es la clave para que el usuario lo utilice, para que el sistema funcione de mejor forma, es necesario antes de su desarrollo realizar el análisis de las acciones de los objetos, para generar la sincronía que se necesita se debe formar una arquitectura del flujo de datos que estará recorriendo el sistema, por lo regular es formada en módulos de eventos de los objetos (interfaz gráfica de usuario, acciones, funciones, etc.). Los módulos no son independientes por lo regular llegan a procesar de una forma paralela en tiempo real, en una forma más clara de explicar, esto ocurre cuando el usuario esta interactuando con el sistema en una acción de navegación y visualizando imágenes complejas que se procesan al mismo tiempo, realmente el usuario no se da cuenta que en fracciones de segundo se han realizado miles de cálculos matemáticos para procesar audio, movimiento, colisiones, iluminación, etc.

El movimiento de un objeto produce la animación de los estados del objeto cómo puede ser el desplazamiento en un entorno digital, con un script se reconoce correctamente cuando una simulación virtual proporciona el control interactivo del desplazamiento. Es decir, el usuario puede decidir la dirección del movimiento y recorrer un entorno virtual libremente. Esto implica un cambio sustancial, la experiencia de usuario evoluciona de ser un simple espectador, que es pre-determinado (como una película), y se convierte en un participante en el mundo digital, cambiando su experiencia de usuario. Sin embargo, la experiencia de usuario cuando se deja de ser un simple espectador, implica de una gran demanda de procesamiento de información. [8]

La interacción de un desplazamiento de los objetos en un ambiente virtual se generan con el mismo truco que se utiliza en el cine y la TV, una secuencia de imágenes que se propagan rápidamente dando la ilusión de movimiento. Esta técnica toma ventaja de la persistencia del ojo humano, lo que requiere alrededor de 30 imágenes por segundo para percibir totalmente una imagen visual (debido a la memoria de la retina y saturación visual electroquímica) y por lo tanto mantiene la información percibida brevemente. [8]

El movimiento es reconocido por las diferentes posiciones de los objeto en una imagen. Por lo tanto, una secuencia de imágenes donde se almacene el cambio de posición de objeto con una visualización de 30 imágenes por segundo, será interpretada por el ojo como un movimiento fluido. Para que el usuario pueda moverse libremente en un entorno virtual, debe producirse al instante una animación en 3-D. Para lo cual tiene entonces una fracción de segundo para poder generar las imágenes y calcular nuevas posiciones de perspectiva. Permitir que cuando la velocidad y la dirección puedan ser como en un entorno real, esto se conoce como visualización interactiva y de navegación en tiempo real e implica una relación entre la capacidad de los ordenadores, la geometría del modelo y su representación realista. [8] En la *figura 1.4* se muestra un recorrido virtual en tiempo real.

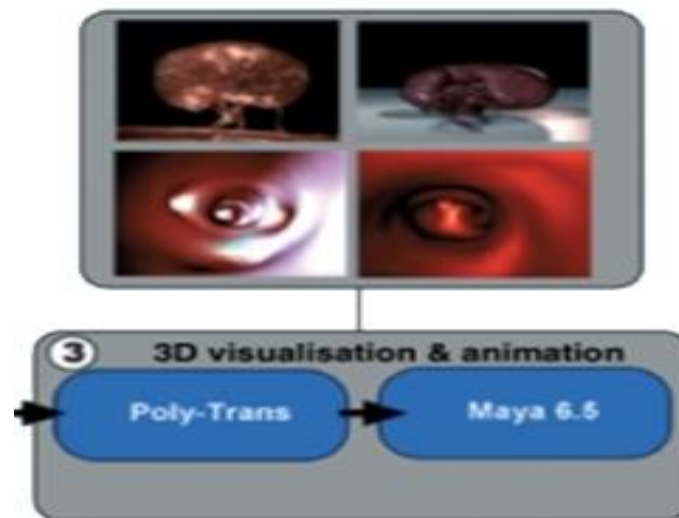


Figure 1.4 Visualización y Animación 3-D

1.3.1 Visualización estereoscópica CGI 3D inmersiva

La visualización en 3-D se realiza utilizando pantallas no inmersiva (2-D tradicional). Los usuarios generan las imágenes en 3-D dentro de los entornos virtuales 3-D inmersiva (3-DI), donde se percibe a sí mismo como estar rodeado de un mundo 3D. Existen tres tipos de visualización:

- Tradicional 2-D no inmersiva (2-DNI)
- El 3-D no inmersiva (3-DNI – Anáglifos Gafas)
- EL 3-DI (cabezal de la pantalla con la posición de la cabeza y el seguimiento de la orientación montada).

La visualización es diferente en los 3-DI y 3-DNI pueden requerir diferentes formas de codificar imágenes. En un entorno 3-DNI, el usuario puede utilizar un marco basado en escenas de la codificación de referencia, mientras en 3-DI podrá fomentar el uso de una trama centrada en el visor de referencia y codificación. La visualización estereoscópica 3-D se confunde a menudo con la visualización convencional de objetos 3-D, existe una percepción mucho más real de la profundidad (la “tercera dimensión”, 3-DI), de diferentes planos de escena y de objetos. [14]

1.3.2 Visualización estereoscópica pasiva CGI 3-D

La visualización estereoscópica pasiva se utiliza para visualizar imágenes multiplexadas en longitud de onda, llamadas comúnmente anáglifos. En una superficie plana se muestra una imagen a partir de la combinación de dos imágenes desplazadas, creadas únicamente con dos colores complementarios, ya sean rojo-azul, rojo-verde o bien ámbar-azul. Estas dos imágenes equivaldrían al par estéreo. La percepción de profundidad en el sistema visual humano de imágenes en superficies planas requiere la ayuda de experiencias previas o de objetos externos, como pueden ser las gafas anáglifo, gafas LCS (Liquid Crystal Shutters – Obturadores de Cristal Líquido) y otros sistemas más modernos, que no afectan al cansancio visual. Las gafas anáglifo están formadas por dos lentes (muy sencillas), cada una con uno de los colores que componen la imagen. De esta manera actúan como filtro y dejan ver a cada ojo sólo el par estéreo que le corresponde. Así pues, por ejemplo, si tuviéramos una imagen creada a partir del desplazamiento de una imagen azul (enfocada para el ojo izquierdo) y otra roja (enfocada para el ojo derecho), necesitaríamos una gafas anáglifo con filtros de los mismos colores: el ojo derecho tendría la lente de color azul y el izquierdo la lente roja, ya que el filtro sólo permite ver la imagen que no sea del mismo color. [15] Las imágenes anáglifo no requieren ningún monitor especial para mostrarse adecuadamente, aumentando el atractivo de la técnica como una solución de 3D estereoscópica de bajo costo.

Otra forma de pasivo 3-D utiliza gafas de visión especial de monitores 3-D polarizadas LCD (Liquid Crystal Display) y pasiva (sin color) vidrios polarizados. La luz polarizada viaja en ondas que ruedan en la misma orientación. Las diferentes orientaciones pueden ser usadas para codificar los canales izquierdo y derecho de un par de imágenes estereoscópicas. Las gafas polarizadas actúan como filtros para la separar los canales, cada lente permite polarizar la luz y coincidir con la orientación, permitiendo conservar la fidelidad del color con éste método. [15] En la *figura 1.5* se muestra la técnica cuando se usan los lentes de estereoscópica pasiva.

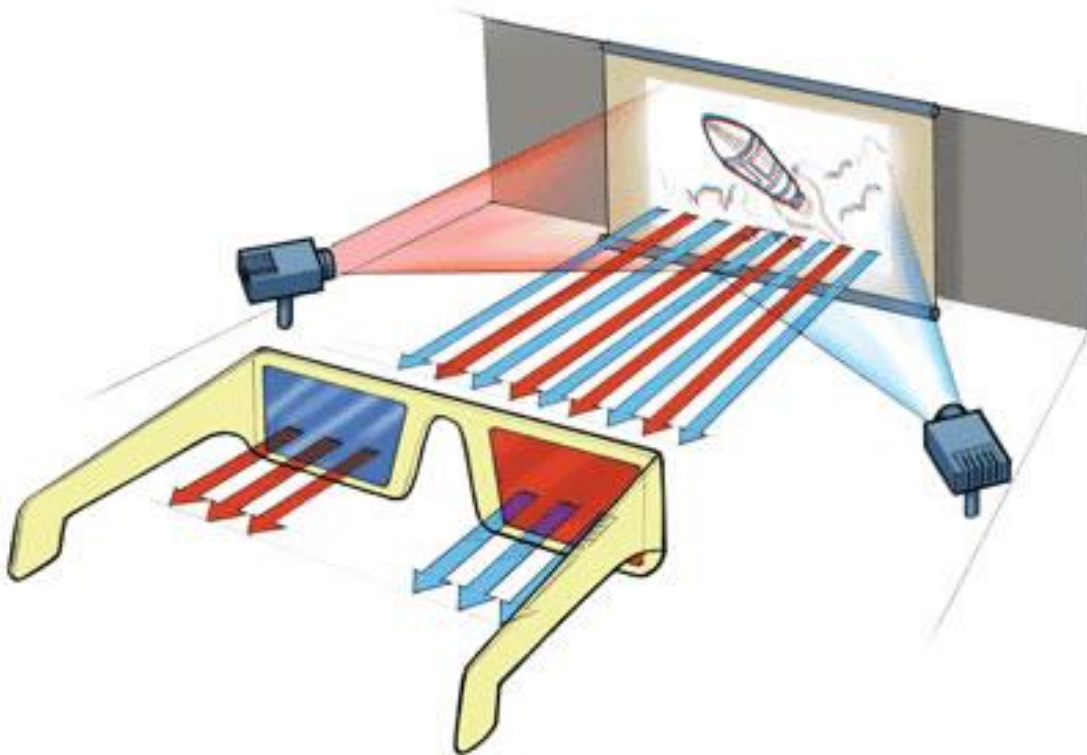


Figura 1.5 Sistema de gafas anáglifos.

1.3.3 Visualización estereoscópica activa CGI 3-D

Las gafas de obturación consisten en gafas con cristal líquido que electrónicamente oscurecen cada ojo, coordinados con el barrido de la imagen del monitor. Al ser la velocidad de obturación a unos 60 FPS, el oscurecimiento no se percibe. De este modo se presenta consecutivamente la imagen izquierda y derecha, y las gafas permiten la visión respectiva obteniendo la visión en profundidad. Esta tecnología utiliza un par de pilas, gafas de obturación LCD inalámbricas junto con una pantalla de alta velocidad de actualización (120 Hz) especial, 3-D Vision. Cada Lente de las gafas de obturador activo se oscurece cuando se aplica voltaje, pero por lo demás es transparente. Las gafas tienen un receptor incorporado de infrarrojos (IR) y se controlan sin cables por un emisor de infrarrojos conectados al ordenador a través de USB (Universal Serial Bus). Se oscurecen alternativamente sobre un ojo y luego el otro, de forma sincronizada con la frecuencia de actualización del monitor, mientras la pantalla muestra alternativamente diferentes perspectivas para cada ojo, usando una técnica llamada secuenciación alternativa-frame. NVIDIA 3-D Vision (Que no debe confundirse con la tecnología NVIDIA 3-D Vision anaglifo) emplea este método para las computadoras compatibles con NVIDIA GPUs. La tecnología ofrece una fidelidad de color. [15] En la **figura 1.6** se muestra los requerimientos necesarios para ejecutar la visualización estereoscópica activa.



Figura 1.6 Requerimientos para el funcionamiento de NVIDIA 3-D Vision.

Los gráficos a nivel de controlador como los dispositivos de NVIDIA 3-D Visión (una versión libre, de gama baja de NVIDIA 3-D Vision se ejecuta en “modo anáglifo”) para NVIDIA GPUs (Unidad de Procesamiento Gráfico) y el controlador (se establece en el modo libre 3D anáglifo con salida DirectX / Direct 3-D). Los principales problemas con los anáglifos son las imágenes fantasmas, provocadas por la retina y la fidelidad de colores. Existen soluciones para reducir el efecto, pero los colores no pueden ser reproducidos correctamente en imágenes de anáglifo. [15] En la **figura 1.7** se muestra como activar las opciones de visualización estereoscópica 3-D.

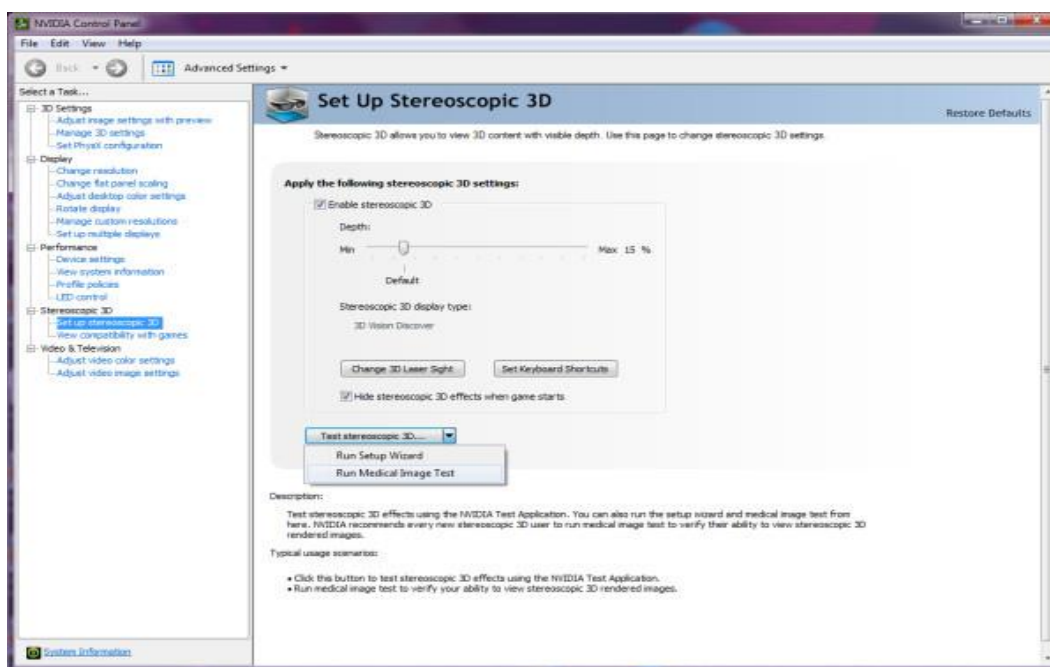


Figura 1.7 NVIDIA Opciones 3-D Estereoscópicas.

CAPÍTULO II

TECNOLOGÍAS Y METODOLOGÍAS DE IMPLEMENTACIÓN

2.1 DICOM para obtener los datos de imágenes biomédicas.

La imagen digital y de comunicación en Medicina (DICOM publicado originalmente por el Colegio Americano de Radiología-Asociación de Fabricantes Eléctricos del Estándar de Imágenes Digitales y Comunicaciones en Medicina, es organizada por el Comité de Estándares de DICOM multi-especialidad), norma o estándar que especifica un conjunto de datos no propietarios del protocolo de intercambio, formato de imagen digital, y estructura de archivos de imágenes biomédicas, que contiene la información relacionada con las imágenes. Los conceptos fundamentales del protocolo DICOM como un mensaje, servicios y objetos de información son base de la funcionalidad de DICOM. [3] Una interfaz DICOM implica mucho más que una simple especificación de hardware, algo así como una toma de corriente eléctrica o el cable paralelo de la impresora. De hecho, DICOM no define un “macho y hembra” en absoluto, sino que define la forma y el flujo de los mensajes electrónicos que transmiten las imágenes e información relacionada entre los ordenadores. En el momento de escribir estas líneas, el estándar DICOM es un volumen de trece conjuntos de información de ingeniería, que se utiliza por los ingenieros como un modelo para las estructuras de información y procedimientos que controlan la entrada y salida de datos de los sistemas de imágenes médicas. Si está bien diseñado (con las especificaciones DICOM), configurada y correctamente, el equipo tiene una interfaz DICOM que se comunicará sin problemas con otros equipos. Dado que las interfaces DICOM están disponibles para casi todos los modelos de equipos de diagnóstico de imagen, los ejecutores del sistema de formación de imágenes ahora tienen la libertad de seleccionar el equipo con base en los méritos y no en consideraciones de propietario. [3] En la *figura 2.1* se muestra el proceso de envío de las imágenes DICOM a una interfaz de computo.

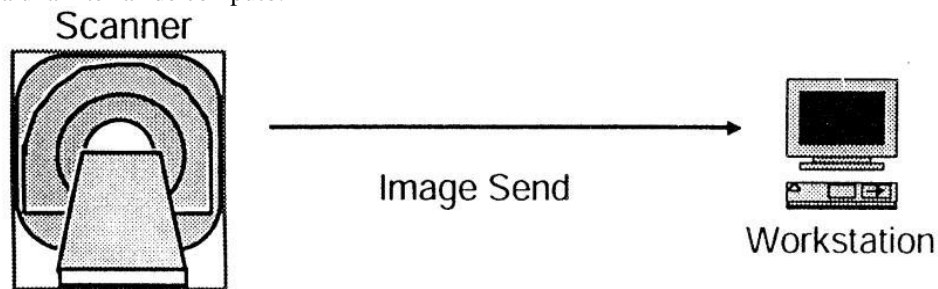


Figura 2.1 Almacenamiento confiable, compromiso de almacenamiento en una extensión básica DICOM.

2.1.1 Descripción general de las imágenes biomédicas DICOM.

DICOM proporciona información detallada de ingeniería que se puede utilizar en las especificaciones de interfaz. Las interfaces DICOM están disponibles para la conexión de cualquier combinación de las siguientes categorías de dispositivos de imágenes digitales [3]:

- a) Un equipo de adquisición de imágenes (TAC, MR, RXC, ecografía y escáner de medicina nuclear).
- b) Los archivos de imágenes.
- c) Los dispositivos de procesamiento de imágenes y estaciones de trabajo de visualización de imágenes.
- d) Los dispositivos de salida en papel (por ejemplo, transparencias fotográficas y las impresoras de papel).

DICOM es una especificación completa del contenido de la información, la estructura, la codificación y protocolos de comunicación para el intercambio electrónico de imágenes diagnósticos y terapéuticas e información relacionada con las imágenes. Desde 1995, todas las principales modalidades de imágenes de diagnóstico han sido estandarizadas. Esta lista incluye la tomografía computarizada (TC), resonancia magnética (MRI), radiografía computarizada, ultrasonido, medicina nuclear, radiofluoroscopia, angiografía de rayos X, y la captura secundaria (para vídeo digitalizado). DICOM off-line de gestión de almacenamiento de medios permite a los usuarios intercambiar archivos de forma manual DICOM en medios de almacenamiento extraíbles. Un archivo DICOM puede incluir no solo imágenes, sino también la información relacionada que distingue una imagen de otra (por ejemplo, los detalles pertinentes del procedimiento realizado, texto de interpretación, o la configuración de formato para la impresión). La posibilidad de enviar información relacionada con las imágenes es una de las características más importantes que distinguen a DICOM de los muchos estándares de archivo de imagen que se limitan a los datos de imagen por sí solo. DICOM define un formato de archivo y un directorio de archivos de imágenes e información relacionada. Los usuarios pueden especificar tipos preferidos de los medios físicos para transportar archivos DICOM para un contexto de imagenología clínica particular (por ejemplo, la angiografía coronaria, general de diagnóstico ecografía, endoscopia gastrointestinal). Por ejemplo, la comunidad de Cardiología de Estados Unidos ha especificado el almacenamiento de imágenes de coronariografía y cateterismo cardiaco en disco compacto de solo lectura utilizando un grupo de imágenes fotográficas expertas con una compresión sin pérdida conjunta. Otras posibles aplicaciones de DICOM off-line de gestión de almacenamiento son medios de transporte de las imágenes de diagnóstico a partir de una unidad de impresión portátil a un departamento de consultoría o de una estación de trabajo de diagnóstico a una sala de conferencias clínicas. Una de las razones por las que DICOM ha tenido éxito en una amplia variedad de contextos de imágenes clínicas, es que la norma especifica una declaración de conformidad, que mejora la comunicación de las especificaciones del software para equipos de imágenes. [3]

2.2 Reconstrucción volumétrica de estructuras corporales

La capacidad de comprensión de estructuras encefálicas está íntimamente relacionada con la habilidad para visualizarlas. Tradicionalmente, las distintas modalidades de técnicas de diagnóstico por la imagen han compartido una limitación común, la bidimensionalidad de las representaciones. La visualización e interpretación de imágenes anatómicas, obtenidas a partir de técnicas de diagnósticas, supone una dificultad añadida debido a su complejidad y a las diferencias individuales en su morfología (variabilidad biológica).

El aprendizaje de la anatomía humana requiere la comprensión de las características morfológicas y relaciones espaciales de vecindad entre las diferentes estructuras corporales. La visualización volumétrica optimiza la presentación de la compleja información contenida en las imágenes morfológicas obtenidas por las técnicas diagnósticas. No sólo se mejora la capacidad sintética de visualización, sino también el grado de interactividad y realismo, superando las restricciones inherentes a las representaciones 2-D tradicionales.

El proceso de reconstrucción 3-D o generación volumétrica consiste en la adquisición de una o más series de imágenes tomográficas, la posterior segmentación o identificación y etiquetado de las estructuras anatómicas corporales a reconstruir, la extracción de contornos/superficies en secciones transversales adyacentes, y el renderizado del modelo 3-D.

La utilización de recursos visuales, en formato 3-D, resulta ser un método atrayentes e innovadores para la docencia y el aprendizaje de los estudiantes, siendo además, unas herramientas reutilizables y útiles para fomentar y motivar el aprendizaje de los alumnos de en cualquier disciplina.

La metodología consta de utilizar imágenes médicas procedentes de la técnica de diagnóstico de imágenes gráficas computarizadas. A partir de este procedimiento se generan las estructuras anatómicas en 3-D, mediante el software Amira, que permite la reconstrucción de objetos 3-D, como los huesos que se muestran en la *figura 2.2*, y representarlos como una malla volumétrica de superficie triangular o tetrahedral a partir de los datos que representan el volumen de una imagen (vóxeles).

Amira permite cargar cualquier conjunto de datos de imagen médica (Resonancia Magnética, Tomografía Computarizada, etc.) soportando distintos formatos de imagen digital, incluyendo el formato estándar para imágenes radiológicas DICOM y empleando en el manejo, almacenamiento y transmisión de imágenes médicas.

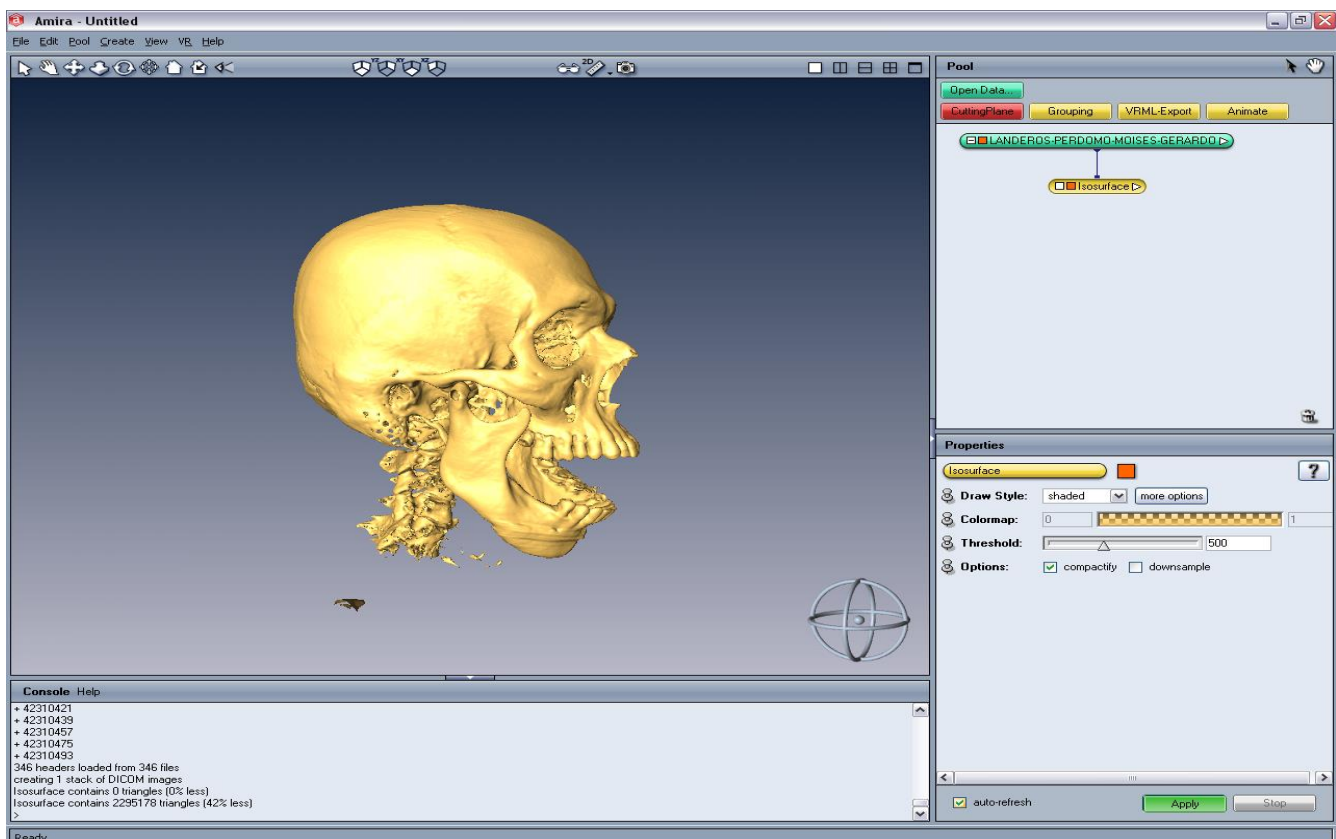


Figura 2.2 Reconstrucción volumétrica de la cabeza.

2.3 Diseño asistido por computadora 3-D

El diseño asistido por computadora 3-D, son modelos tridimensionales que corresponden a escenarios con clases diferentes de objetos, tales como plantas, arquitectura, personas, vehículos, anatomía, etc. Por esto, es necesario utilizar geometrías que representen con precisión las características de los objetos en cuestión. Los métodos como los fractales y los sistemas de partículas, permiten a su vez, representar elementos naturales como nubes, montañas, vegetales, glóbulos rojos, entre otras formas. Difícilmente habrá un solo método para describir cabalmente todos los objetos que existen en la naturaleza y que incluyan todas sus características. Básicamente se tienen dos modalidades para representar la geometría [8]:

- Representaciones de frontera que describe los objetos 3-D, como el conjunto de superficies externas.
- Representaciones de divisiones de espacio, en que se subdivide el ambiente en pequeñas unidades volumétricas (voxels: volumen element), determinando la región espacial que ocupa cada objeto.

En los inicios de la computación gráfica las descripciones geométricas se estructuraban sólo en puntos o líneas con dirección (vectores), pero actualmente la mayoría de los programas de CAD y RV se basan en la representación por fronteras. De este modo la descripción de una geometría 3-D en la computadora se realiza, generalmente, debido a los polígonos externos de las formas, localizados en un espacio determinado por tres ejes cartesianos, que representan el espacio, con cierto tamaño del universo y medidas. En estas descripciones gráficas las posiciones de cada punto se identifican por sus coordenadas. Con dos puntos externos se pueden indicar una línea y a su vez tres puntos definen un plano. De este modo, un objeto 3-D se define como un conjunto de superficies poligonales (denominadas caras), en el cual se identifican sus aristas y vértices, en la **figura 2.3** se muestra las representaciones gráficas. También se reconoce en cada polígono, una normal, que indica la orientación de la cara, estableciendo uno de los lados como visible. [8]

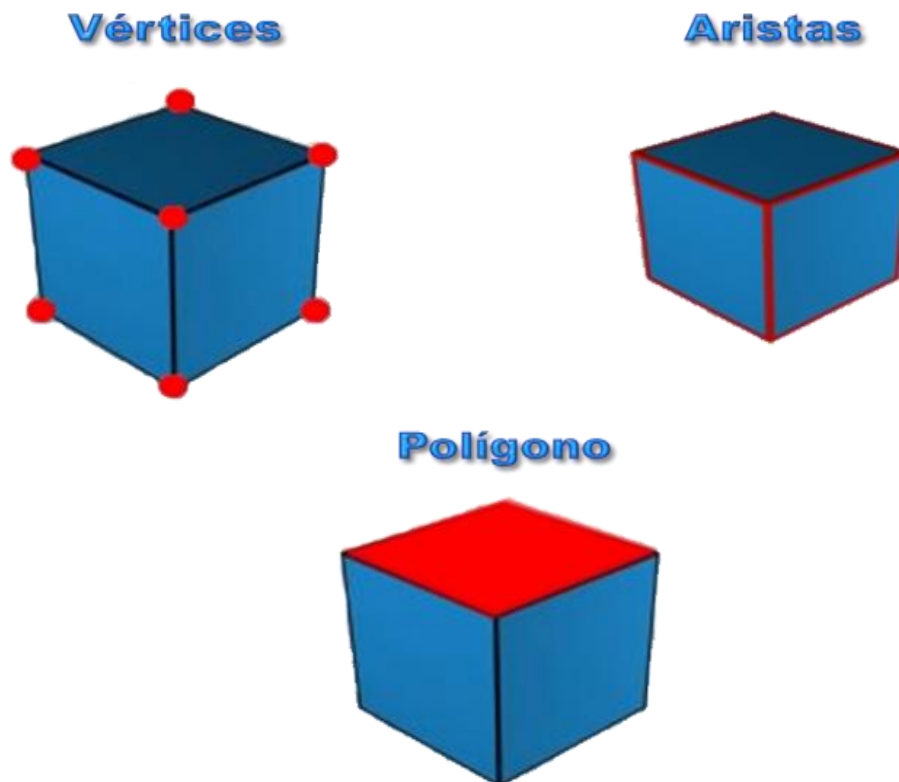


Figura 2.3 Representación gráfica de vértices, aristas y polígonos.

2.3.1 Transformación geométrica traslación

La traslación es el cambio de posición de un punto $p_1(x_1, y_1, z_1)$ a la posición $p_2(x_2, y_2, z_2)$. La función que representa el cálculo para el cambio de posición se representa en la *figura 2.4* mediante la siguiente matriz: [8]

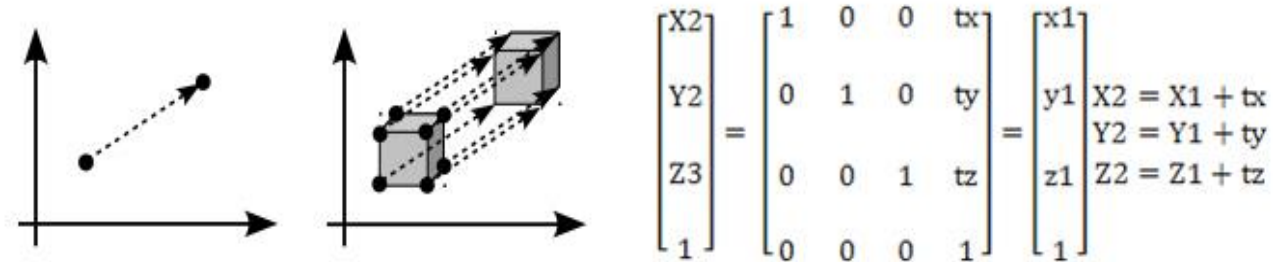


Figura 2.4 Función de transformación geométrica de traslación.

Las variables t_x, t_y y t_z son la distancia de traslación que se incrementó o decremento de los ejes X, Y, Z. La función de traslación es completamente válida para todos los puntos de los objetos 3-D, por lo que un objeto 3-D se traslada por cada uno de los puntos de posición inicial de los objetos 3-D, en concreto los objetos 3-D son representados por superficie o polígonos que son compuestos a su vez por vértices, estos vértices son los puntos de traslación en el cambio de distancia y al realizar el cambio de posición de los vértices se vuelve a generar o trazar los polígonos que componen al objeto 3-D. [8]

2.3.2 Transformación geométrica rotación

La función de rotación se realiza mediante la asignación de un eje R(X, Y, Z) el cual se girará el objeto y el valor de la rotación, dicho valor se modifica en radianes. Los ángulos de rotación positivos producen giros en sentido opuesto a las manecillas del reloj con respecto al eje de una de las coordenadas. En la *figura 2.5* se muestra el giro con respecto al eje z la matriz. [8]

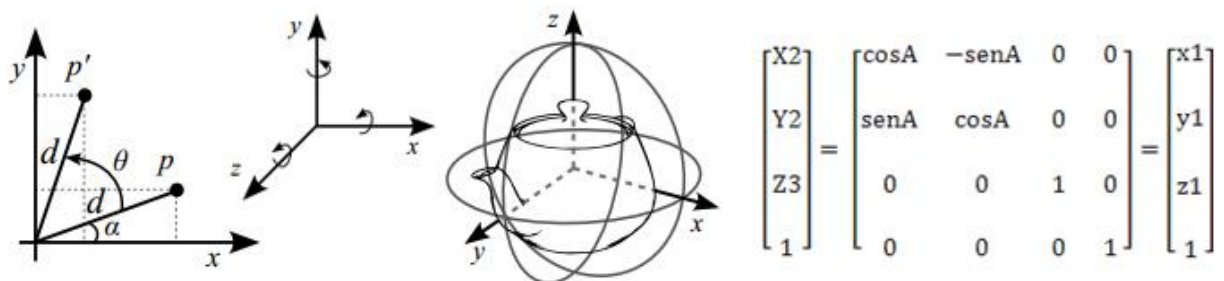


Figura 2.5 Función de Transformación Geométrica de Rotación.

2.3.3 Transformación geométrica escalamiento

La función de escalar un objeto 3-D cambia el tamaño y/o vuelve a posicionar con respecto al origen de las coordenadas. Si todos los parámetros de transformación son iguales se cambian las dimensiones relativas del objeto. En la *figura 2.6* se muestra la matriz que representa el escalamiento. [8]

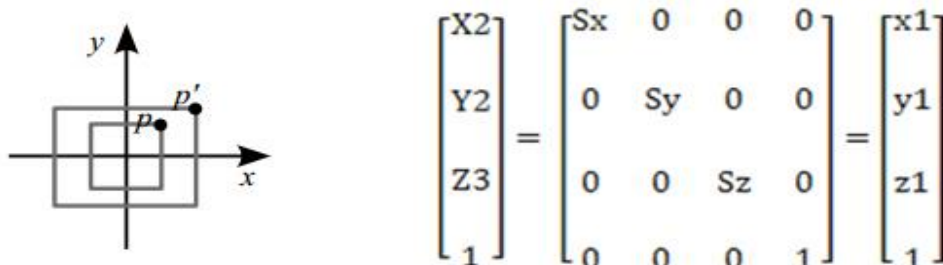


Figura 2.6 Las variables S_x, S_y, S_z son los parámetros de escalamiento con respecto al origen.

2.3.4 Representación geométrica tridimensional

Existen varios métodos de representación tridimensional para obtener la imagen de una escena 3-D, primero debemos establecer una posición (usualmente en coordenadas cartesianas) para el observador o cámara, que son los conceptos utilizados para denominar al punto de vista desde el cual se ve el modelo computacional. Frente a éste se establece un plano de visión, en el cual se proyecta la información geométrica de las formas 3-D. En principio los objetos son representados como formas transparentes (wireframe) y luego se aplican técnicas de iluminación y presentación de superficies para colorear y sombrear las áreas visibles. Así la computadora genera las imágenes a partir de la posición de la cámara con respecto a los objetos que ésta debiera estar visualizando. [8] En la *figura 2.7* se muestra la técnica para generar la vista de objeto 3-D.

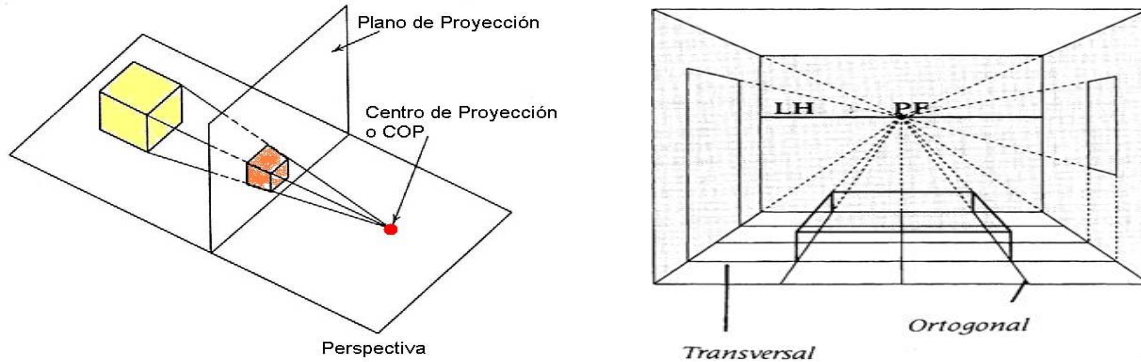


Figura 2.7 Representación geométrica tridimensional ó perspectiva.

2.3.5 Proyección paralela

Este método consiste en proyectar puntos del objeto a través de líneas paralelas al eje de visión. Es así, como en diferentes posiciones de la cámara, se proyectan las superficies visibles del objeto sobre el plano de visualización para obtener diversas vistas (superiores, laterales, isométricas) que posean medidas proporcionales del objeto. [8] En la *figura 2.8* se muestran las proyecciones.

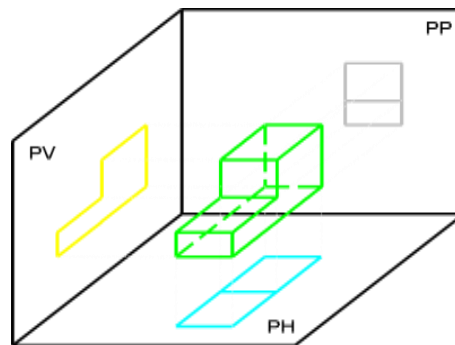


Figura 2.8 Proyección paralelas u ortogonal.

2.3.6 Eliminación de líneas ocultas

La eliminación de líneas ocultas, consiste en calcular las regiones o píxeles del frente y la parte de atrás de la imagen que ocupa cada objeto del modelo. Básicamente hay dos tipos de algoritmos para la eliminación de líneas ocultas. El algoritmo con precisión de imagen, se trata de determinar que objeto es visible en el centro de cada píxel y el algoritmo con precisión de objetos, se trata de determinar que parte de las caras o que parte de las aristas de un modelo son visibles. [8] En la *figura 2.9* se muestra la eliminación de líneas ocultas.

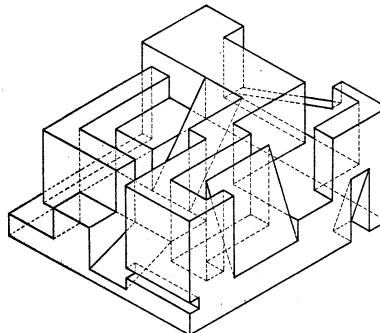


Figura 2.9 Líneas Ocultas en una estructura tridimensional.

2.3.7 Modelado geométrico de la estructura anatómica

La visualización anatómica es el principal objetivo para el aprendizaje, una vez obtenida la imagen médica o los datos de los modelos anatómicos que se requieren, se debe realizar el método de modelado que constituye la obtención de superficies o conjuntos de estructuras 3D anatómicas y se modelaron a partir de herramientas tecnológicas como 3D Studio Max.

Las estructuras anatómicas modeladas se generaron con datos teóricos y no reales, estas estructuras se le considera como datos no biomédicos, ya que se obtienen de información teórica como dibujos, diagramas, esquemas, fotos, modelos reales de plástico o en conservación, de la imaginación o de investigaciones teóricas. El método de modelado geométrico de una estructura anatómica consiste en usar las distintas formas geométricas o primitivas geométricas (cubo, esfera, etc.) que se pueden manipular, unir entre varias estructuras para modificar, conformar objetos complejos y posteriormente tener una estructura sumamente real.

Las estructuras anatómicas se examinaron para determinar su realismo, estas estructuras se fueron ensamblando conforme a la estructura anatómica, cada estructura genera información espacial (propiedades como posición geométrica espacial, tamaño, iluminación, etc.) que posteriormente se requiere para manipular en el entorno virtual, este proceso posteriormente se explicará en el desarrollo como se almacena en una base de datos. En la **figura 2.10** se muestra el modelado geométrico de la estructura anatómica.

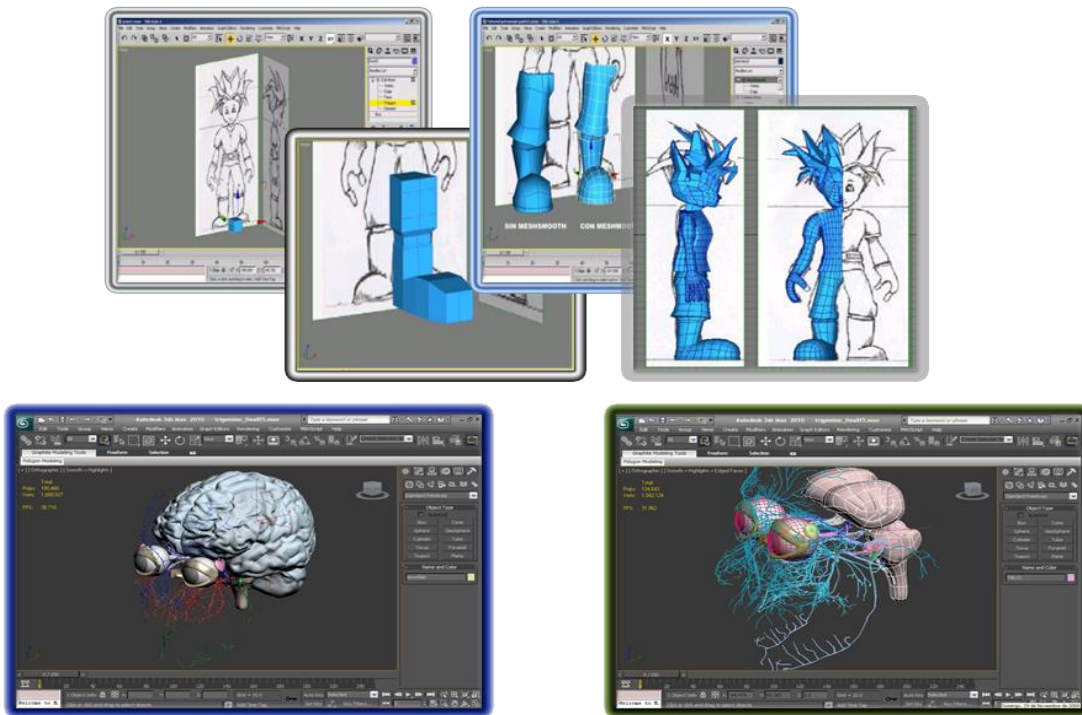


Figura 2.10 Modelado geométrico de la estructura anatómica.

2.4 Diseño asistido por computadora 2D

El diseño asistido por computadora 2-D (CAD, Computer Aided Design) se maneja en el computador, básicamente de dos formas: como un mapa de puntos (bitmaps) o como una geometría vectorial. Las imágenes de puntos se utilizan en el procesamiento de fotografías digitalizadas o en dibujos sencillos y consisten en una trama de puntos de color (pixel) que representan la imagen. Se caracterizan por su facilidad de manipulación cambiando la totalidad de los puntos, pero están limitadas en resolución y precisión, además que son planas. El archivo digital se puede conservar en una diversidad de formatos, entre los que podemos mencionar: BMP, GIF, PCX, TIFF o JPG. La geometría guarda la información precisa de las formas gráficas, a través de un sistema de coordenadas cartesianas $P(X, Y)$. En la **figura 2.11** se identifica por sus puntos extremos y pueden ser representadas a cualquier escala o calidad de imagen. [8]

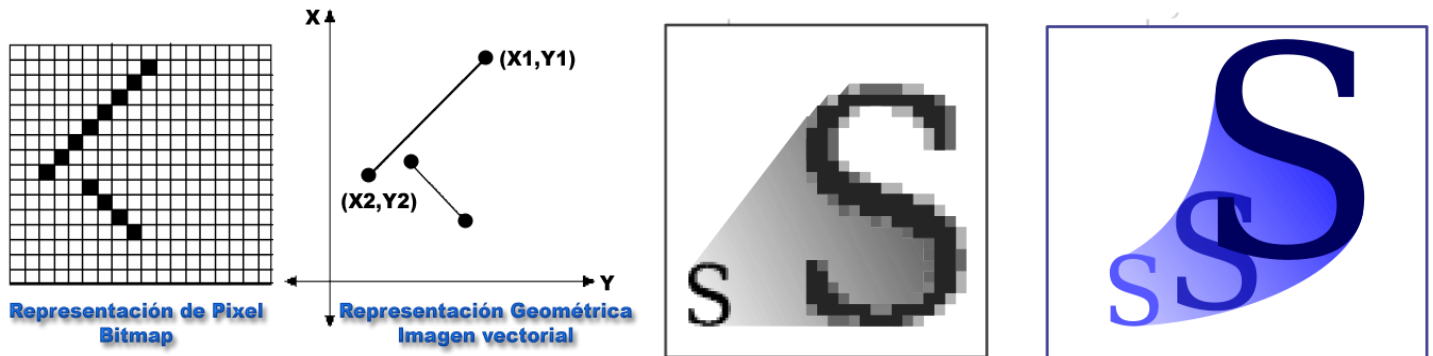


Figura 2.11 Representación de pixel bitmap y geometría de imagen vectorial.

El diseño de las imágenes vectoriales es más complejo, pero permite trabajar con dimensiones exactas e incluso con formas 3-D, basta con agregar un tercer eje de coordenadas $P(X, Y, Z)$. Estos programas son denominados CAD (Computer Aided Design, Diseño Asistido por Computadora). Por ello en la actualidad es fácil encontrar diseños de construcciones, automóviles, computadores, Anatómicos y otros, realizados con programas CAD. La información gráfica consiste en una base de datos numérica almacenada en el computador, que define cualquier tipo de forma o volumen. Para poder visualizarlos, es necesario interpretarlos y generar una imagen en la pantalla. Aunque la geometría sea 3-D, se genera una vista plana de la información. Ocasionalmente esta vista reúne varias caras de las formas (perspectiva) otorgando una representación de la volumetría. Esta información es coherente, en el sentido que la alteración de una parte de la geometría 3-D, altera el conjunto, por tanto cada vista presentará la geometría modificada, asegurando la sensación de manejar un modelo completo. [8] En la siguiente **figura 2.12** se muestra un diseño asistido por computadora 2D que se utiliza para el diseño de los botones de la GUI.

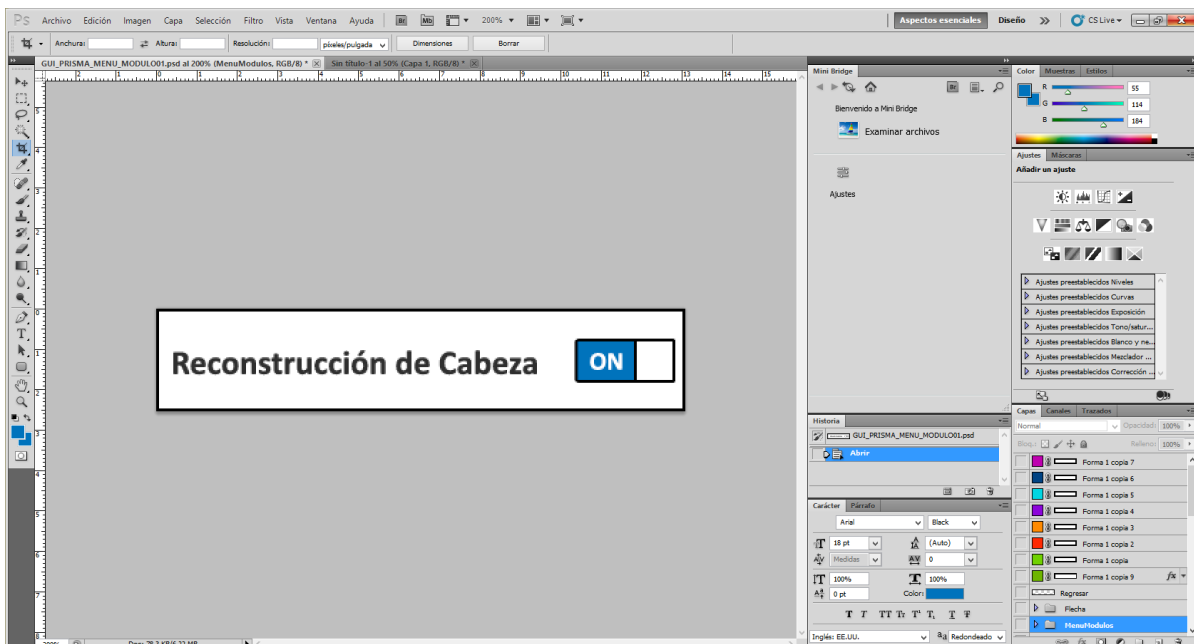


Figura 2.12 Diseño de botones de la GUI.

2.4.1 Texturizado para geometría tridimensional

Para lograr que los modelos 3-D alcancen una apariencia realista, es necesario incorporar a la geometría dos capacidades relevantes, superficies texturizadas y fórmulas de iluminación. Estas dos propiedades complementan la geometría y generan una mejor reproducción de la situación real que representa el modelo computacional. [8] En la *figura 2.13* se muestra las geometrías que están texturizadas.

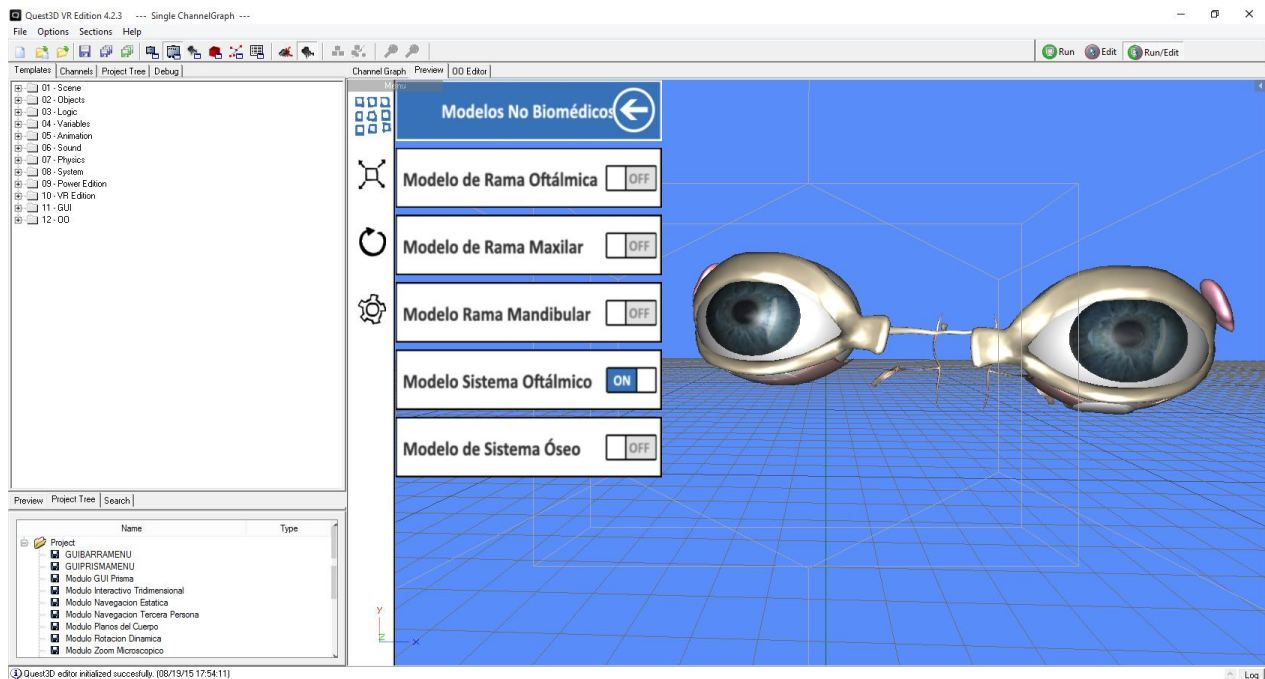


Figura 2.13 Planos y geometrías texturizadas.

Los objetos geométricos pueden ser representados con colores básicos. Los colores se asignan de acuerdo con patrones RGB (Red, Green, Blue) o HSL (Tono, Intensidad, Brillo), en combinaciones proporcionales de cualquier de estos tres valores. Como los colores están representados por intensidad luminosa, los valores mayores son más claros. Los tres indicadores al máximo producen el blanco, al mínimo generan el negro, como ausencia de color adicionalmente se incorporan un valor para la transparencia (alpha-channel). [8] En la *figura 2.14* se muestra los canales RGB y A.

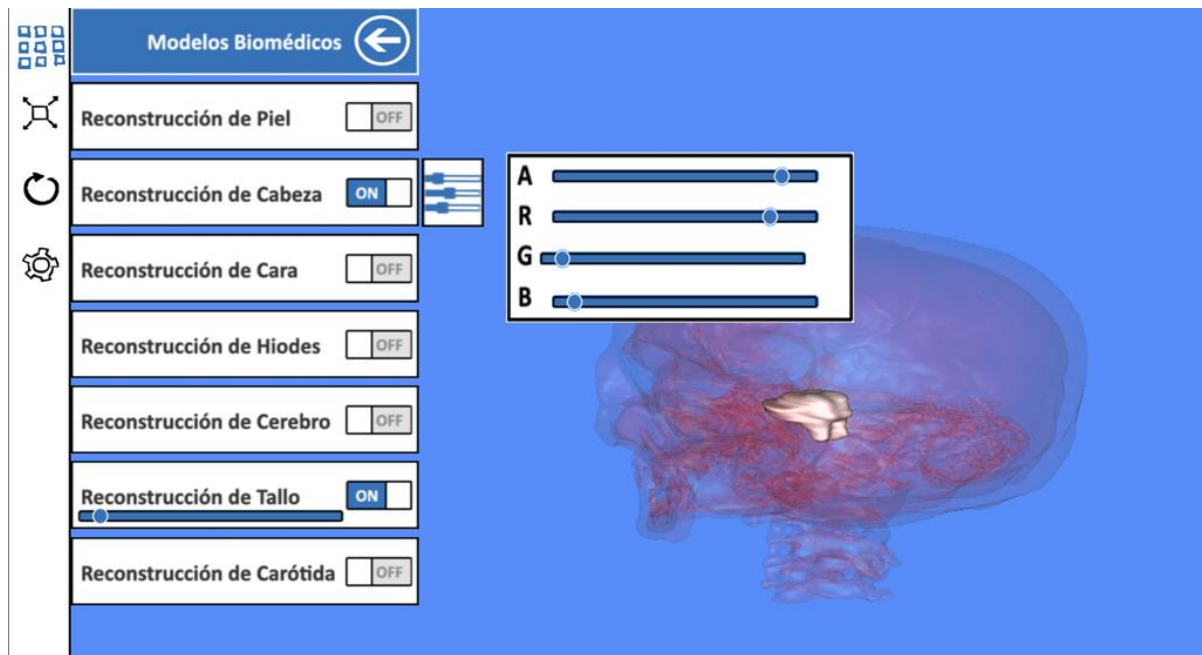


Figura 2.14 Objetos geométricos representados con los canales RGB y Alpha.

Sin embargo, se ha reconocido que la variedad de colores, por muy amplia que sea, no alcanza a otorgar un gran realismo visual, debido a que en los ambientes reales, los colores varían de tono en las superficies. Una de las técnicas más eficaces para mejorar la apariencia realista de las superficies, ha sido la incorporación de texturas digitalizadas. La utilización de texturas consiste en utilizar fotografías de materiales reales, que se aplican como imágenes planas en algunas superficies. La técnica de aplicación es similar a la proyección de mapas, por lo que se denomina “mapeo”. Si la forma es distinta a la imagen, ésta simplemente se recorta. Si el tamaño de la imagen es muy pequeño se puede repetir en la superficie, o se puede agrandar, aunque esto último muestra notoriamente los puntos de la imagen (pixelado). [8] En la **figura 2.15** se muestra la implementación del mapeo UV.

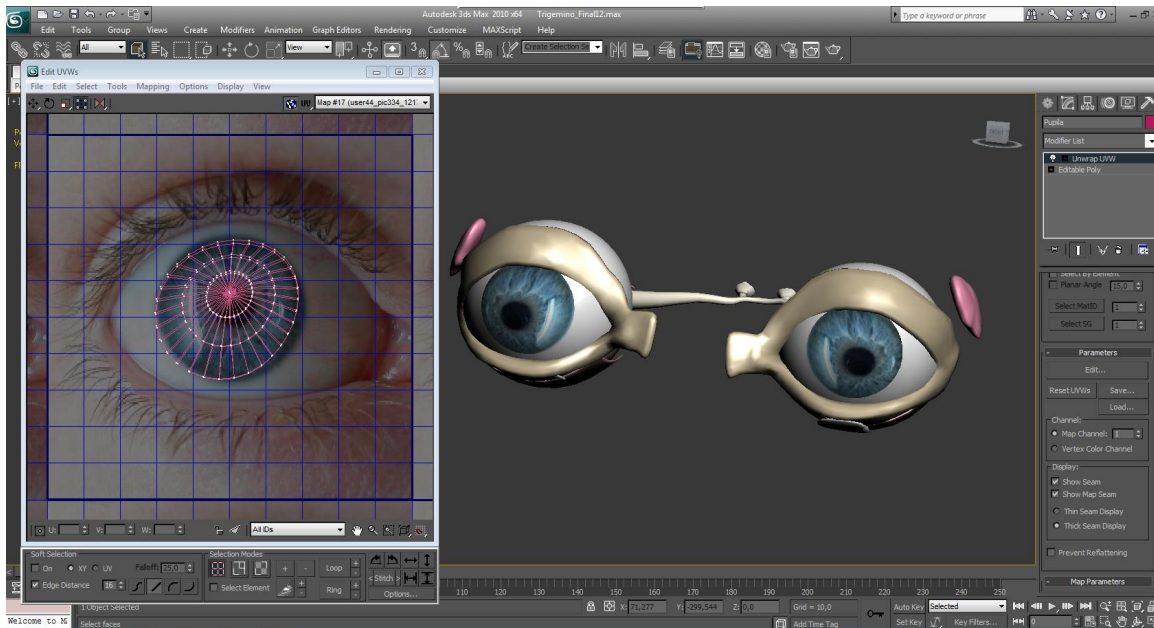


Figura 2.15 Mapeo UV de textura en objetos geométricos para lograr un realismo visual.

Las texturas pueden tener un valor de rugosidad (bumping mapping), en que las sombras adquieren profundidad. De hecho, es más eficaz en algunas formas pequeñas, no modelarlas 3-D sino simplemente texturizarlas con una imagen real, con el fin de ahorrar procesamiento geométrico en objetos que tienen un rol secundario en la escena. También algunas formas naturales difíciles de modelar apropiadamente, es preferible representarlas con una fotografía, en que se asigna un valor de transparencia al color de fondo, para generar un perfil recortado. [8] Esta técnica no se utilizó pero se menciona para potencializar la calidad de realismo de las estructuras no biomédicas que posteriormente se requieran modelar. En la **figura 2.16** se muestra la técnica de Bumping Mapping.

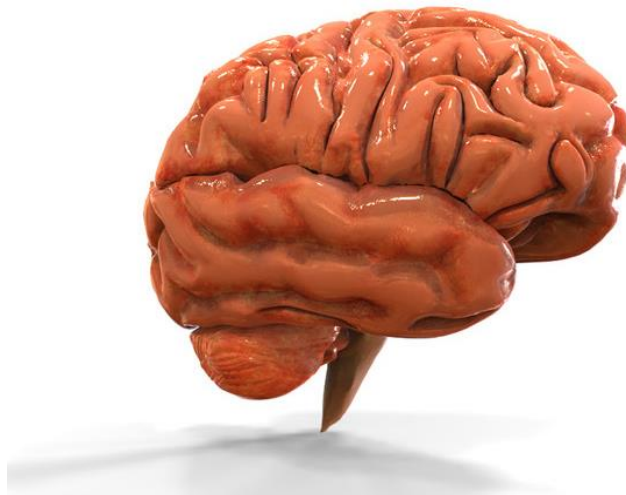


Figura 2.16 Bumping mapping en objetos geométricos para adquirir profundidad y mejor realismo.

2.5 Motores gráficos para el aprendizaje del profesional del sector salud.

Los motores gráficos son principalmente para desarrollar aplicaciones de entretenimiento, pero para fines del objetivo se utiliza para desarrollar el sistema de software biomédico para la enseñanza electrónica (e-Learning) médica del profesional de la salud. La adopción de la tecnología tiende en aumento de los conocimientos respetivamente del usuario, el aprendizaje auto-dirigido y el trabajo colaborativo son los resultados pedagógicos asociados a esta tecnología.

A pesar del impulso y la aceleración en el desarrollo tecnológico actual, la adopción tecnológica de los motores gráficos para desarrollar mundos virtuales como innovación de aprendizaje electrónico en el área médica, no ha sido fundamental, que sin alguna respuesta del porque no a la adopción inmediata de la tecnología. Sin duda una de las áreas donde se discute más esta adopción y tiene gran importancia en la enseñanza de los profesionales de la medicina. Al igual que la mayoría de los problemas básicos de la educación, esta cuestión plantea problemas en diferentes niveles de pensamiento, y es beneficioso para hacer frente mientras se adopta la tecnología en la educación profesional de la salud. El valor asociado con el aprendizaje presenta las variables que influyen en la tasa de adopción. Las supuestas cualidades beneficiosas de los desarrollos en motores gráficos, tales como inmersión, oportunidades de rol, simulación e interacción personal asociados a la tecnología, su influencia en la formación y los resultados del aprendizaje van creciendo. [16]

La Web es un término que se refiere al aprendizaje electrónico (e-Learning):

- Mejora la comunicación y colaboración entre las personas a través de las tecnologías.
- Mejora la comunicación entre las aplicaciones de software independientes a través de estándares web abiertos para la descripción y acceso a los datos.
- Mejora las interfaces que imitan la capacidad de respuesta en tiempo real de aplicaciones de escritorio dentro de una ventana del navegador.

Existen universidades médicas virtuales en todo el mundo, pero si hubiera la adopción de aplicaciones desarrolladas en motores gráficos, se podrían tener una mejor meta educativa, así como la interacción de la tecnología puede ser por medio virtual, también tendría la opción en algunos entornos virtuales específicos para mejorar la enseñanza, utilizar un avatar para mejorar la inmersión y educación del profesional de la salud.

El término avatar es una antigua palabra sánscrita que retrata a una deidad que adopta una forma humana. Se trata de figuras animadas, que el usuario podrá realizar varias funciones como moverse, navegar, volar, etc. Esta combinación puede ofrecer una gran cantidad de oportunidades para todos los grupos de edad, porque los juegos se pueden motivar a los usuarios finales para acceder y divertirse mientras se aprende. [16]

El potencial educativo y de investigación de las aplicaciones desarrolladas con motores gráficos se está exponiendo, sin embargo, la investigación educativa que implica el uso y la eficacia de estas tecnologías innovadoras ha estado en su infancia. Se necesita más investigación sobre los resultados de la educación. Sin embargo, se han reportado ventajas para que los estudiantes participen en estas nuevas tecnologías. Estudiantes interactúan activamente con el contenido y con el papel de habilidades de las aplicaciones desarrolladas con motores gráficos relacionados con su profesión. [16]

Al permitir a los estudiantes interactuar con un avatar en un ambiente seguro, simulado, una disminución en la ansiedad, el aumento de la competencia en el aprendizaje de una nueva habilidad, el estímulo a cooperar y colaborar, así como resolver conflictos, sería posible. El aprendizaje activo se lleva a cabo debido a otros participantes que están en el mismo mundo virtual y la construcción de objetos para representar las ideas que puedan mejorar la auto-reflexión y el conocimiento. Las investigaciones hacen un punto para identificar y evaluar las tecnologías incipientes que tienen un efecto sobre la enseñanza y el aprendizaje en la educación superior y prevé un plazo para la posible adopción en la educación. Los mundos virtuales se están utilizando actualmente como espacios educativos y continuos, creciendo en popularidad en los campus y las empresas de todo el mundo. [16]

2.5.1 Motores gráficos para el desarrollo de aplicaciones.

El término motor de gráfico (Game-Engines) surgió a mediados de los años 90 con la aparición del famosísimo juego de acción en primera persona Doom, desarrollado por la compañía id software. Esta afirmación se sustenta sobre el hecho de que Doom fue diseñado con una arquitectura orientada a la reutilización mediante una separación adecuada en distintos módulos de los componentes fundamentales, como el sistema de renderizado gráfico, el sistema de detección de colisiones, el sistema de audio y los elementos más artísticos como, los escenarios virtuales o las reglas. Este planteamiento facilitaba enormemente la reutilización de software y el concepto de motor gráfico se hizo más popular a medida que otros desarrolladores comenzaron a utilizar diversos módulos para generar los propios. En otras palabras, era posible diseñar y desarrollar un juego serio sin modificar el núcleo o motor, sino que el esfuerzo se podía dirigir directamente a la parte artística y a las reglas del mismo. Este enfoque ha ido evolucionando y se ha expandido, desde la generación de módulos por desarrolladores independientes o amateurs hasta la creación de una gran variedad de herramientas, bibliotecas e incluso lenguajes que facilitan el desarrollo de videojuegos serios utilizando motores o herramientas pertenecientes a terceras partes, debido a que les resulta más rentable y generalmente obtienen mejores resultados. [17] Resulta especialmente importante destacar la existencia de motores gráficos, como por ejemplo:

- JMonkey Engine (abreviado jME)
- Java3D
- Quest3D, se muestra en la *figura 2.17*
- Virtools
- Unity3D
- VTK (Visualization ToolKit)

Este tipo de herramientas, junto con técnicas específicas, optimización de metodologías, patrones de diseño, entre otros, conforman un aspecto esencial al desarrollar una aplicación en un motor gráfico. Al igual que ocurre en otros aspectos relacionados con la Ingeniería del Software, desde un punto de vista general resulta aconsejable el uso de todos estos elementos para agilizar el proceso de desarrollo y reducir errores potenciales. En otras palabras, **“no es necesario, ni productivo, reinventar la rueda cada vez que se afronta un nuevo proyecto”**. El desarrollo de aplicaciones en motores gráficos es un proceso complejo debido a los distintos requisitos que ha de satisfacer y a la integración de distintas disciplinas que intervienen en dicho proceso. Desde un punto de vista general, una aplicación desarrollada en motor gráfico es una aplicación gráfica en tiempo real en la que existe una interacción explícita mediante el usuario y la propia aplicación. En este contexto, el concepto de tiempo real se refiere a la necesidad de generar una determinada tasa de “frames” o imágenes por segundo, típicamente 30 o 60, para que el usuario tenga una sensación continua de realidad. Por otra parte, la interacción se refiere a la forma de comunicación que existe entre el usuario y la aplicación. Normalmente, esta interacción se realiza mediante joysticks o mandos, pero también es posible llevarla a cabo con otros dispositivos como por ejemplo teclados, ratones, cascos o incluso mediante el propio cuerpo a través de técnicas de visión por computador o de interacción táctil. Los ingenieros son los responsables de diseñar e implementar el software que permite la ejecución de la aplicación desarrollada en el motor gráfico, así como las herramientas que dan soporte a dicha ejecución. Resulta bastante común encontrar perfiles de ingenieros especializados en programación gráfica o en scripting. Sin embargo, el concepto de ingeniero transversal es bastante común. En el mundo del desarrollo es bastante probable encontrar ingeniero sénior responsable de supervisar el desarrollo desde un punto de vista técnico, de manera independiente al diseño y generación de código. No obstante, este tipo de role suelen estar asociados a la supervisión técnica, la gestión del proyecto e incluso a la toma de decisiones vinculadas a la dirección del proyecto. El diseño de la aplicación desarrollada en un motor gráfico, se destaca la evolución del mismo desde el principio hasta el final, la secuencia, las reglas, los objetivos. Evidentemente, todos los aspectos de diseño están estrechamente ligados. Al igual que ocurre en otras disciplinas en el campo de la informática, el desarrollo de las aplicaciones en motores gráficos se ha beneficiado de sus herramientas que facilitan dicho desarrollo, automatizando determinadas tareas y ocultando la complejidad inherente a muchos procesos de bajo nivel, en otras palabras hacen la vida más sencilla a los desarrolladores. [17]

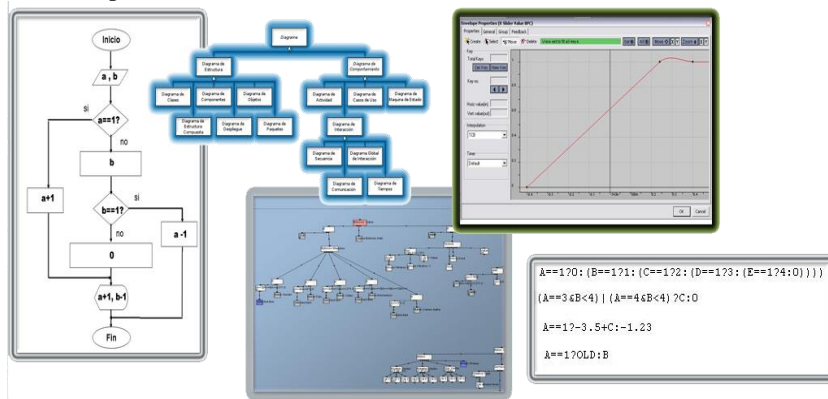


Figura 2.17 Motor de juegos (Game Engines Quest3D).

2.6 Motor gráfico Quest3D

Quest3D es un motor gráfico con una plataforma de desarrollo, generalmente se usa para arquitectura, diseño de producto, software de entrenamiento y simuladores médicos. Los datos y animaciones son importados de paquetes CAD tales como Maya, 3D Studio Max y AutoCAD, a Quest3D donde son utilizados para la creación de aplicaciones interactivas 3D en tiempo real. Quest3D es un producto desarrollado por Act-3D B.V. El por qué desarrollar el sistema de software biomédico en el motor gráfico Quest3D, fue para cumplir con el objetivo de tener estructuras corporales reales de calidad, ya que este motor fue el único que tiene el potencial de importar objetos tridimensionales de más de 1, 600,000 polígonos, en los demás motores gráficos su límite es de 60,000 a 75,000 polígonos por objeto, cantidad que no podría visualizar una estructura real.

Entorno de desarrollo

Una de las características más importantes de Quest3D es la metodología de programación. Una forma totalmente diferente a los habituales lenguajes de programación, las librerías que contiene está basado en C++, el entorno de desarrollo de Quest3D se programa de forma visual y codificada (cannel Editor). Otra característica destacable es el hecho de que el programador puede modificar la aplicación mientras esta se ejecuta. Esto significa que no existe compilación de código como en los entornos de programación habituales. El entorno de Quest3D consiste en diferentes editores en la creación de la aplicación:

- Editor de árbol de canales (Channels Tree)
- Modificación de características de los objetos 3D
- Animaciones
- Programación Hight Level Shading Language(HLSL)
- Programación LUA Script entre otros.

Lógica de las aplicaciones

Las aplicaciones Quest3D se desarrollan conectando componentes funcionales (cajas negras), denominadas "Channels", vinculados componen una estructura de árbol, que representa la estructura del programa que se implementa. El árbol de cajas negras se ejecuta por completo una vez (al menos) por frame, invocando a cada "channel". Lo que se obtiene como resultado es una aplicación 3D en tiempo real. Como no hay fase de compilación, o interpretación de un lenguaje de scripting, ya que los "Channels" son cajas con su código precompilado (implementadas en Dynamic Link Libraries), el rendimiento de las aplicaciones es el mismo en fase de diseño que en ejecución, características muy apreciada cuando se desarrollan aplicaciones en tiempo real.

Orientación a objetos

Quest3D ha evolucionado en su versión 4.0 y posteriores, permitiendo implementar aplicaciones siguiendo un paradigma de diseño orientado a objetos. Haciendo uso de su nuevo editor de interfaces y clases, permite de una manera bastante intuitiva el encapsulamiento de subárboles de "channels" en "Objetos", que contienen métodos y propiedades. Esta característica aumenta la potencia del entorno, permitiendo aplicaciones mucho más dinámicas.

Publicación de aplicaciones

Las aplicaciones finalizadas pueden ser publicadas en diferentes formatos, para permitir su visualización en diferentes medios:

- Fichero ejecutable "standalone" (plataforma Microsoft Windows)
- Visor Web basado en control ActiveX. Los navegadores soportados son Internet Explorer, Firefox y Chrome

Se puede adquirir diferentes licencias para el uso tanto comercial como educacional.

Requerimientos del sistema

Algunas funcionalidades del motor gráfico 3D requieren hardware más especificado.

- Windows 2000, Windows XP, Windows Vista (64 or 32 bit) y Direct X 9.
- 256 MB RAM.
- Procesador de 1Ghz.
- Tarjeta gráfica compatible con DirectX y 32 MB de memoria gráfica.
- 400MB de espacio en disco duro.

Aplicaciones

- Realidad virtual
- Video juegos para pc
- Visualización arquitectónica
- Serious games
- Simuladores
- TV
- Cine

Enlace externo: <http://www.quest3d.com/>

2.6.1 Bibliotecas de utilidad del motor gráfico

Las principales bibliotecas vinculadas a las utilidades que dan soporte al motor gráfico, son específicas del ámbito del motor gráfico pero otras son comunes a cualquier tipo de proyecto de software que tenga una complejidad significativa. [17] A continuación se enumeran algunos de las bibliotecas:

- **Biblioteca matemática:** Responsable de proporcionar al desarrollador diversas utilidades que faciliten el tratamiento de operaciones relativas a vectores, matrices, operaciones vinculadas a líneas, rayos, esferas y otras figuras geométricas. Las bibliotecas matemáticas son esenciales en el desarrollo de una aplicación en un motor gráfico, ya que éstos tienen una naturaleza inherentemente matemática.
- **Estructuras de datos y algoritmos:** Responsable de proporcionar una implementación más personalizada y optimizada de diversas estructuras de datos, como por ejemplo listas enlazadas o árboles binarios, y algoritmos, como por ejemplo búsqueda u ordenación. Este subsistema resulta especialmente importante cuando la memoria de las plataformas sobre las que se ejecutará el motor está limitada.
- **Gestión de memoria:** Responsable de garantizar la asignación y liberación de memoria de una manera eficiente.

2.6.2 Render del motor gráfico

Debido a que el componente gráfico es una parte fundamental, junto con la necesidad de mejorarlo continuamente, el motor de renderizado es una de las partes más complejas de cualquier motor gráfico. Al igual que ocurre con la propia arquitectura de un motor gráfico, el enfoque más utilizado para diseñar el motor de renderizado consiste en utilizar una arquitectura de capas. [17] En la *figura 2.18* se muestra la arquitectura general de un motor de rendering.

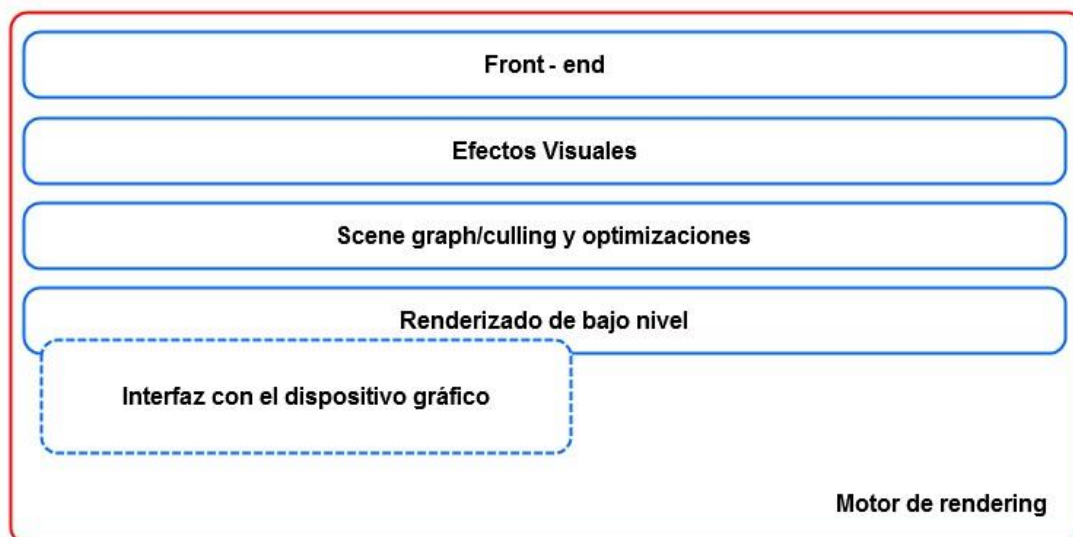


Figura 2.18 Arquitectura general de un motor de rendering.

2.7 Front-End

Suele estar vinculada a funcionalidad relativa a la superposición de contenido 2-D sobre el escenario 3-D. Por ejemplo, es bastante común utilizar algún tipo de módulo que permita visualizar el menú de un juego o la interfaz gráfica que permite conocer el estado de un evento u objeto. [17] En la **figura 2.19** se muestra la funcionalidad para el contenido 2-D.

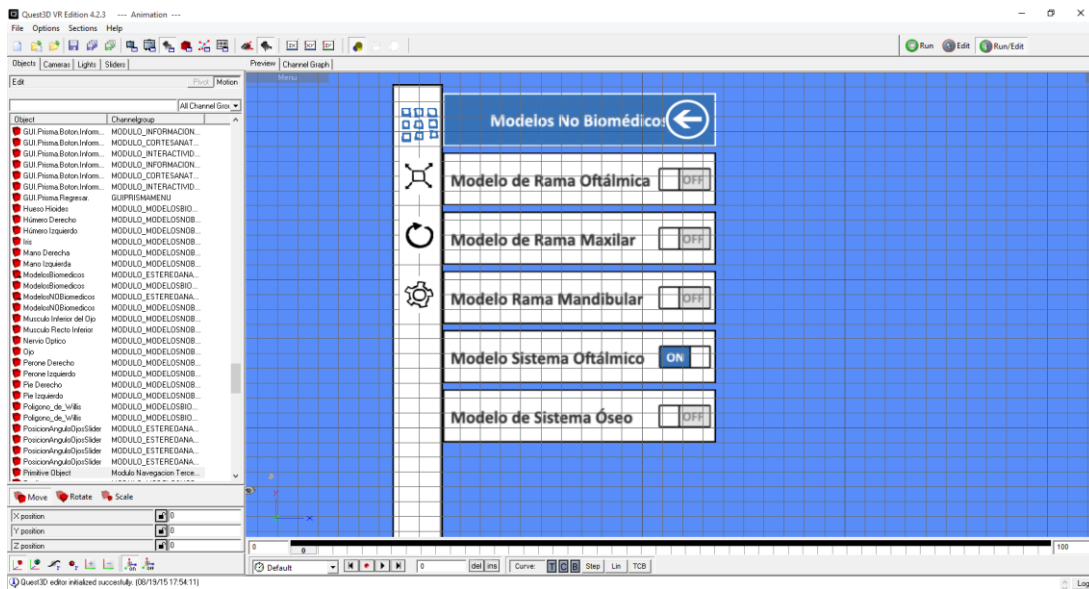


Figura 2.19 Módulo que permita visualizar el menú de un juego o la interfaz gráfica.

2.8 Escenario gráfico / eliminación y optimización

Es la responsable de seleccionar qué parte o partes de la escena se enviarán a la capa de render. Esta selección, u optimización, permite incrementar el rendimiento del motor de render, debido a que se limita el número de primitivas geométricas. Aunque en la capa de render sólo se dibujan las primitivas que están dentro del campo de visión de la cámara, es decir, dentro de la ventana de visión, es posible aplicar más optimizaciones que simplifiquen la complejidad de la escena, obviando aquellas partes de la misma que no son visibles desde la cámara. Este tipo de optimizaciones son críticas con el objetivo de obtener tasas de frames por segundo aceptables. Una de las optimizaciones típicas consiste en hacer uso de estructuras de datos de subdivisión espacial para hacer más eficiente el render, gracias a que es posible determinar de una manera rápida el conjunto de objetos potencialmente visibles. Dichas estructuras de datos suelen ser árboles, aunque también es posible utilizar otras alternativas. Tradicionalmente, las subdivisiones espaciales se conocen como escenario gráfico, aunque en realidad representan un caso particular de estructura de datos. [17] En la **figura 2.20** se muestra la eliminación de geometría que está dentro del campo de visión.

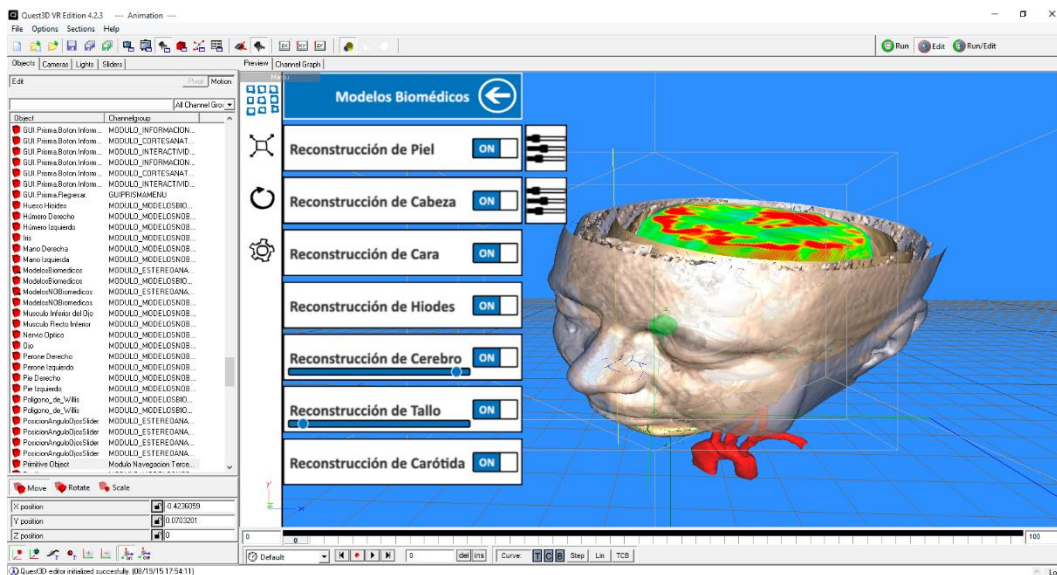


Figura 2.20 Eliminación geométrica por campo de visión.

2.9.2 Componente ambiental

Si suponemos que la luz ambiental afecta de la misma forma a todas las superficies desde todas direcciones, nuestra ecuación de iluminación ambiental será $I = I_a K_a$, I_a es la intensidad de la luz ambiental que se supone constante para todos los objetos. La cantidad de luz ambiental reflejada por la superficie de un objeto está determinada por K_a , el coeficiente de reflexión ambiental, que varía entre 0 y 1. El coeficiente de reflexiones ambientales una propiedad del material. Como sucede con otras propiedades, el coeficiente de reflexión ambiental es una conveniencia empírica y no corresponde directamente a ninguna propiedad física de los materiales reales. [8] En la *figura 2.22* se muestra las direcciones de iluminación de la componente ambiental.

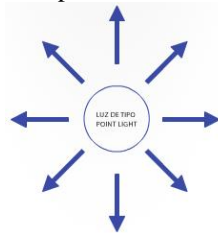


Figura 2.22 Componente ambiental.

2.9.3 Componente difusa

Si consideramos la iluminación de un objeto a partir de una fuente luminosa puntual, cuyos rayos emanan uniformemente en todas direcciones a partir de un solo punto. La brillantez del objeto varía de una parte a otra como se muestra en la *figura 2.23*, dependiendo de la dirección y la distancia con respecto a la fuente luminosa.



Figura 2.23 Componente difusa.

La ecuación de iluminación difusa es $I = I_p K_d \cos \theta$ I_p es la intensidad de la fuente luminosa puntual, el coeficiente de reflexión difusa k_d del material es una constante entre 0 y 1 que varía de un material a otro. El ángulo θ debe estar entre 0° y 90° para que la luz tenga efecto directo en el punto que sombrea, esto quiere decir que la luz emitida detrás del objeto, no iluminará las caras frontales, y viceversa. Suponiendo que los vectores \vec{N} y \vec{L} se normalizaron, podemos describir la ecuación usando el producto punto $I = I_p K_d (\vec{N} \cdot \vec{L})$ Si una fuente luminosa puntual se encuentra a distancia suficiente de los objetos que se sombreen, formará en esencia el mismo ángulo con todas las superficies que comparten la misma normal. En este caso, la luz se conoce como fuente luminosa direccional y L es una constante para la fuente luminosa. Los objetos iluminados de esta manera se ven con demasiado contraste entre las regiones de luz y sombra; por lo tanto, es común añadir un término de ambiente para obtener una ecuación de iluminación más realista $I = I_a K_a + I_p K_d (\vec{N} \cdot \vec{L})$. [8] En la *figura 2.24* se muestra la componente difusa del motor gráfico implementada en una estructura geométrica anatómica.

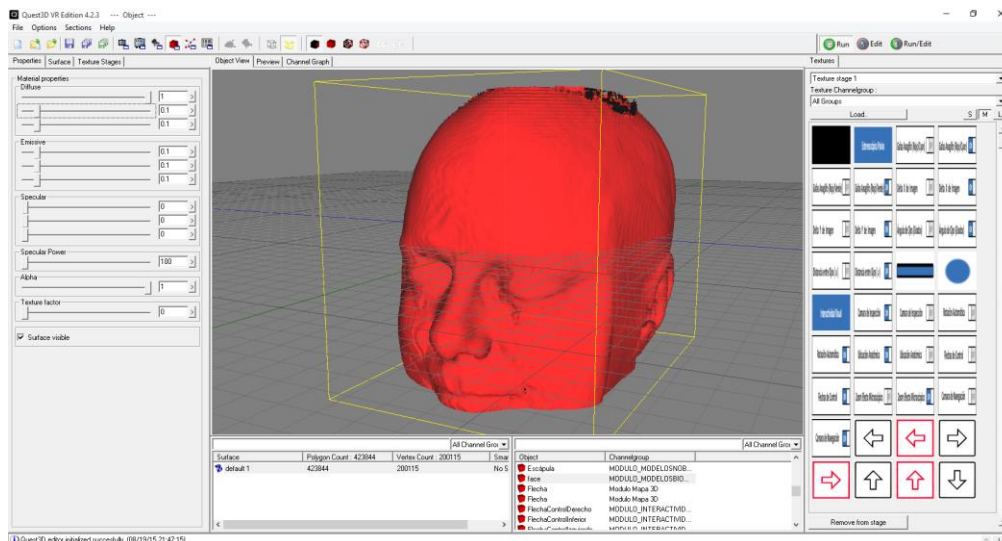


Figura 2.24 Componente difusa en motor gráfico.

2.9.4 Componente especular

La reflexión especular se puede observar en cualquier superficie brillante. El punto de máximo brillo es ocasionado por la reflexión especular, mientras que la luz reflejada en el resto del objeto es el resultado de la reflexión difusa. El color del punto de máximo brillo, así como el de la luz reflejada de forma especular, es del color de la luz incidente y no el que corresponde al color del objeto. Si movemos el punto de vista, también se moverá el punto de máximo brillo, esto sucede porque las superficies brillantes reflejan la luz en forma desigual en distintas direcciones. En la figura 2.25 se muestra la reflexión de la componente especular.

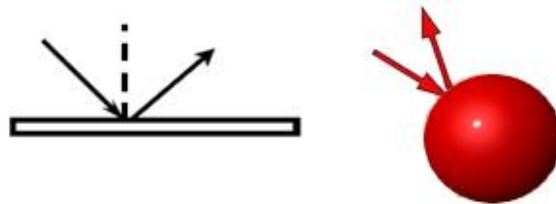


Figura 2.25 Reflexión de la componente especular.

La luz se refleja sólo en la dirección de reflexión \mathbf{R} , que es \mathbf{L} invertida con respecto de \mathbf{N} . De esta manera, el observador puede ver la luz reflejada especularmente en un espejo sólo si el ángulo α es cero; α es el ángulo entre \mathbf{R} y la dirección \mathbf{V} al punto de observación. La máxima reflectancia especular ocurre cuando α es cero y decrece rápidamente conforme aumenta α . Esta caída rápida es aproximada por $\cos \alpha^n$, donde n es el exponente de reflexión especular del material. Los valores típicos de n varían entre uno y valores muy grandes, dependiendo del material de superficie que simule. Un valor de 1 proporciona una caída amplia, suave, mientras que los valores más elevados simulan un punto de máximo brillo definido y enfocado. Para un reflector perfecto, n sería infinita, en el caso de que el $\cos \alpha$ sea negativo, será considerado como 0.

La ecuación que define la iluminación especular, utilizando el modelo Phong será

$$I_s = I_a K_a + K_d I_p (\vec{N} \cdot \vec{L}) + K_s (|\mathbf{N} \cdot \mathbf{H}|)^n \quad \mathbf{H} = (\vec{L} + \vec{V})/2.$$

El vector intermedio \mathbf{H} , llamado así porque su dirección es intermedia entre las direcciones de la fuente luminosa y el observador, también se le conoce como la dirección de reales máximos. Si la superficie se orientara de manera que su normal estuviera en la misma dirección que \mathbf{H} , el observador vería el punto máximo de brillo especular, ya que \mathbf{R} y \mathbf{V} también apuntarían en la misma dirección. [8] En la figura 2.26 se muestra la reflexión de la componente especular.

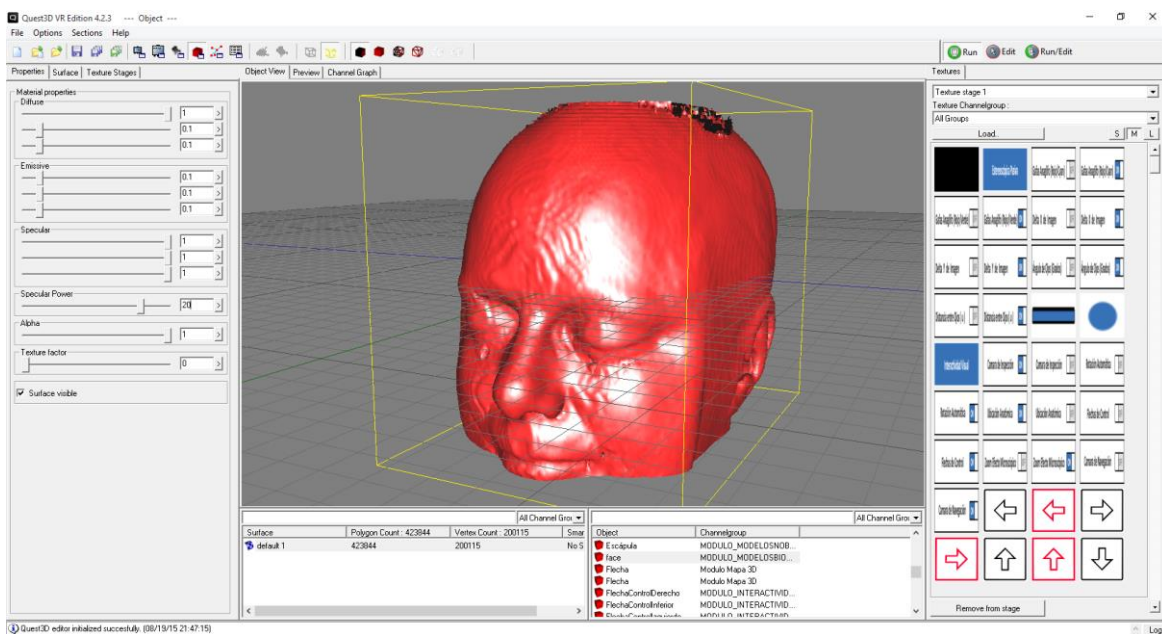


Figura 2.26 Reflexión de la componente especular en motor gráfico.

2.10 Interfaces con el dispositivo gráfico

En cualquier tipo de aplicación gráfica desarrollada con motor gráfica es necesario un módulo que ofrezca una abstracción respecto a la interacción del usuario, es decir, un módulo que principalmente sea responsable de procesar los **eventos de entrada** del usuario. Típicamente, dichos eventos estarán asociados a la pulsación de una tecla, al movimiento del ratón como en la *figura 2.27*, el uso de un joystick, entre otros. Desde un punto de vista más general, el módulo de interfaces de usuario también es responsable del **tratamiento de los eventos** de salida, es decir, aquellos eventos que proporcionan una retroalimentación al usuario. Debido a que este módulo gestiona los eventos de entrada y de salida, se suele denominar comúnmente componente de entrada/salida. El módulo de interfaces de usuario actúa como un puente entre los detalles de bajo nivel del hardware utilizado para interactuar y el resto de controles de más alto nivel. Este módulo también es responsable de otras tareas importantes, como la asociación de acciones o funciones lógicas con el sistema de control, es decir, permite asociar eventos de entrada y acciones lógicas de alto nivel. En la gestión de eventos se suelen utilizar patrones de diseño, de manera que cuando se detecta un evento, éste se traslada a la entidad adecuada para llevar a cabo su tratamiento. [17]

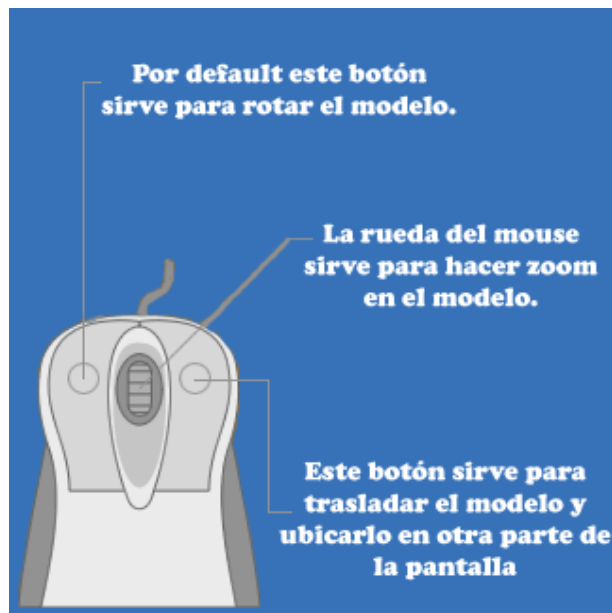


Figura 2.27 Módulo de interacción del usuario.

2.11 Motor de física

La detección de colisiones y su posterior tratamiento resultan esenciales para dotar de realismo al mismo. Sin un mecanismo de detección de colisiones, los objetos se traspasarían unos a otros y no sería posible interactuar con ellos. Desde un punto de vista general, el sistema de detección de colisiones es responsable de llevar a cabo las siguientes tareas:

- La **detección de colisiones**, cuya salida es un valor lógico indicando si hay o no colisión.
- La **determinación de la colisión**, cuya tarea consiste en calcular el punto de intersección de la colisión.
- La **respuesta a la colisión**, que tiene como objetivo determinar las acciones que se generarán como consecuencia de la misma.

Debido a las restricciones impuestas por la naturaleza de tiempo real, los mecanismos de gestión de colisiones se suelen aproximar para simplificar la complejidad de los mismos y no reducir el rendimiento del motor. Por ejemplo, en algunas ocasiones los objetos 3D se aproximan con una serie de líneas, utilizando técnicas de intersección de líneas para determinar la existencia o no de una colisión. También es bastante común hacer uso de árboles BSP para representar el entorno y optimizar la detección de colisiones con respecto a los propios objetos. Por otra parte, algunos desarrollos incluyen sistemas realistas o semi-realistas de simulación dinámica, estos sistemas se suelen denominar **sistema de física** y están directamente ligados al sistema de gestión de colisiones. [17]

CAPÍTULO III

DESARROLLO

El concepto ingeniería de software se propuso originalmente en 1968, a lo largo de las décadas de 1970 y 1980 se desarrolló una variedad de técnicas y métodos de ingeniería de software, tales como la programación estructurada, el encubrimiento de información y el desarrollo orientado a objetos. La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software. Las actividades fundamentales de la ingeniería de software son la especificación, desarrollo, validación y evolución del software. No obstante, cuando se habla de ingeniería de software, esto no sólo se refiere a los programas en sí, sino también a toda la documentación asociada y los datos de configuración requeridos para hacer que estos programas operen de manera correcta. El desarrollo de la World Wide Web es básicamente un almacén de información universal accesible que tuvo efecto sobre los sistemas de software, dichos sistemas de software corrían en computadoras locales. Los sistemas de software basados en la web podían desarrollarse donde se tuviera acceso a un navegador web, en lugar de una interfaz de usuario de propósito específico. Esta situación condujo al desarrollo de una gran variedad de nuevos productos de sistemas de software, a los cuales se integraba desde la web. Así como estos sistemas de software, el desarrollo de navegadores web que procesan programas y realizan procesos locales condujo a una evolución en los sistemas de software. En lugar de elaborar sistemas de software e implementarlo en la PC de los usuarios, se implementaba en un servidor web. [9] El enfoque sistemático que se usa en la ingeniería de software se conoce en ocasiones como proceso de software. Un proceso de software es una secuencia de actividades que conducen a la elaboración de un sistema de software. Existen cuatro actividades fundamentales que son comunes a todos los procesos de software, y éstos son:

- **Especificación:** Se definen el software que se producirá y las restricciones en su operación.
- **Desarrollo:** Se diseña y programa el software.
- **Validación:** Se verifica el software para asegurar que sea lo que el cliente requiere.
- **Evolución:** Se modifica el software para reflejar los requerimientos cambiantes.

En la práctica éstas son actividades complejas e incluyen sub-actividades tales como la validación de requerimientos, el diseño arquitectónico, la prueba de unidad, etc. También existen actividades de soporte al proceso, como la documentación y el manejo de la configuración del software. Cuando los procesos se discuten y describen, por lo general se habla de actividades como especificar un modelo de datos, diseñar una interfaz de usuario, etc. El proceso que se utiliza para explicar los diferentes enfoques del desarrollo de software, se puede considerar, que se extiende y se adapta para crear procesos más específicos de ingeniería de software. El modelo de proceso que se utiliza para cumplir con el objetivo es *el modelo en cascada, este toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución y luego los representa como fases separadas del proceso, tal como especificación de requerimientos, diseño de software, implementación, pruebas, etc.* Las principales etapas del modelo en cascada reflejan directamente las actividades fundamentales del desarrollo, en la *figura 3.1* se muestra el modelo. [9]

- **Análisis y definición de requerimientos.** Los servicios, las restricciones y las metas se definen y sirven como especificación.
- **Diseño del sistema y del software.** Asigna requerimientos, implica identificar y describir las abstracciones y sus relaciones.
- **Implementación y prueba de unidad.** Consiste en verificar que cada unidad cumpla con su especificación.
- **Integración y prueba de sistema.** Las unidades se integran y se prueban como un sistema completo.
- **Operación y mantenimiento.** Se instala y se pone en práctica, incluye corregir errores, mejorar e incrementar servicios.

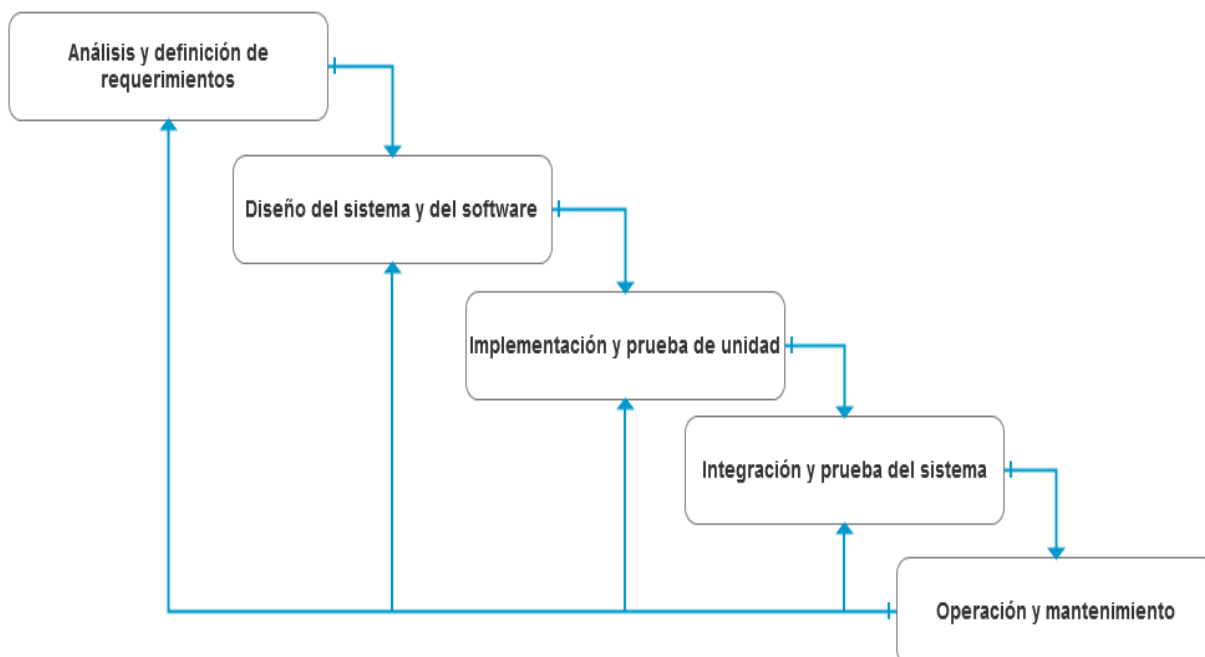


Figura 3.1 Modelo en cascada

En la práctica, dichas etapas se traslapan y se nutren mutuamente de información. Durante el diseño se identificaron los problemas con los requerimientos. En la codificación se descubren problemas de diseño y así sucesivamente. El proceso de software no es un simple modelo lineal, sino que implica retroalimentación de una fase a otra. Durante la fase final del ciclo de vida (operación y mantenimiento), el software se pone en servicio. Se descubren los errores y las omisiones en los requerimientos originales del software. Surgen los errores de programa y diseño, y se detecta la necesidad de nueva funcionalidad. Por lo tanto, el sistema debe evolucionar para mantener útil. Hacer tales cambios (mantenimiento de software) puede implicar la repetición de etapas anteriores del proceso. [9]

3.1 Análisis y definición de requerimientos

La evolución e implementación de las tecnologías de la información en todos los sectores ha crecido considerablemente, y analizando el área de oportunidad en el sector profesional de la salud, por el desaprovechamiento que se observa de las tecnologías de la información con aplicación didáctica y educativa, para potencializar la enseñanza en el área de la salud y científica, que principalmente es el estudio del caso para gestionar y resolver la necesidad. Por estas razones se visualiza solucionar el problema, de no usar las tecnologías de la información para la formación de la población de profesionales del sector salud, en otras palabras el objetivo es reducir la brecha digital que se radica en esta población de profesionales. La solución al problema se resuelve gestionando, el desarrollo y diseño de un *sistema de software de simulación anatómica humana para mejorar la formación, capacitación, entrenamiento y aprendizaje del profesional del sector salud*. La anatomía es la principal herramienta que debe tener el profesional de la salud, ya que es tan importante tener bien sólidos estos conocimientos que serán primordiales para el éxito de un diagnóstico o planeación de una cirugía.

El sistema de software se clasifica, como desarrollo de un sistema de software de información para modelado y simulación, un sistema cuyo principal propósito es gestionar y dar acceso a una base de datos de información (datos de estructuras Biomédicas), los conflictos para este tipo de sistemas de software, serán la seguridad, usabilidad, privacidad y mantenimiento de la integridad de los datos de las estructuras biomédicas. El sistema de software se ejecuta en forma local, pero también funciona en la web para alcanzar mayor distribución de la población a la que se está orientando el desarrollo del sistema de software. Se ejecuta en Sistema Operativo Windows y navegadores como Internet Explorer, por ser el sistema y navegador con mayor uso del sector o población del profesional del sector salud.

El sistema de software impactará en la optimización de recursos, ya que será útil en la planeación para los conocimientos necesarios, para la enseñanza de la fisiología y anatomía del cuerpo humano, y posteriormente para la planeación de una cirugía, son sólo ejemplos por mencionar pero tiene alcances aún mayores. El sistema de software está orientado a todo profesional de la salud que requiera potencializar los conocimientos anatómicos o con interés en el estudio, visualización y análisis de las funciones anatómicas humanas. Sin embargo el alcance de este tipo de sistema de software es virtualmente ilimitado, ya que si queremos visualizar el uso que podrían hacer los profesionales del sector salud con el sistema de software, sería para analizar patologías y poder planear anticipadamente cirugías completas.

Para el análisis y diseño del sistema de software se utilizó la gestión y administración de proyectos, metodologías UML (Lenguaje Unificado de Modelado, Unified Modeling Language), Usabilidad de Sistemas, Arquitectura de la Información, Análisis de Sistemas, Diseño de Sistemas, Diseño de Interfaces de Usuario y Diseño de Modelos Tridimensionales. Los procesos para poder gestionar el desarrollo del sistema de software, son secuencias entrelazadas de actividades, con la meta general de especificar, diseñar, implementar y probar el sistema de software. Para los procesos de actividades del Análisis y Diseño del software, se usaron diferentes herramientas tecnológicas. Por mencionar las herramientas tecnológicas son:

- Sparx Systems Enterprise Architect
- Balsamiq Mockups
- Axure RP (Interactive Wireframe Software & Mockup Tool)

Para mayor información de la descripción de las herramientas tecnológicas que se utilizaron para la gestión del Análisis y Diseño, en el *Anexo I se encuentran toda la Información necesaria.*

Existen cuatro actividades principales en el proceso de ingeniería de requerimientos:

- Estudio de factibilidad
- Obtención y análisis de requerimientos
- Especificación de requerimientos.
- Validación de requerimientos.

3.1.1 Estudio de factibilidad

La gestión del sistema de software para los datos biomédicos requiere de sistemas integrados de información que a menudo no están disponibles para el profesional del sector salud, ya que no se cuenta con disponibilidad de invertir en la calidad y la mejora. Por lo tanto, el apoyo de gestión para la mejora de software puede ser difícil de obtener. De hecho, parece que hay poca comprensión en las organizaciones del profesional del sector salud, ya que deben fomentar y permitir la adopción de la ingeniería de software y soluciones tecnológicas. Una posibilidad es utilizar indagación apreciativa para ayudar a descubrir facilitadores y motivadores que le ayudarán a métodos de ingeniería de software para ganar mayor aceptación en el sector. El sector salud tiene una cultura rígida, esta funcionalidad no soporta de manera óptima la aplicación de herramientas que trascienden, sobre todo cuando se inducen cambios significativos en las relaciones tradicionales, ya que puede generar escepticismo y miedo. Esto pone de relieve la importancia de considerar las cuestiones sociales y la comunicación en la evaluación e implementación de un sistema de software. [18] En la **figura 3.2** se muestra el modelo de despliegue del sistema de software con funcionamiento web.

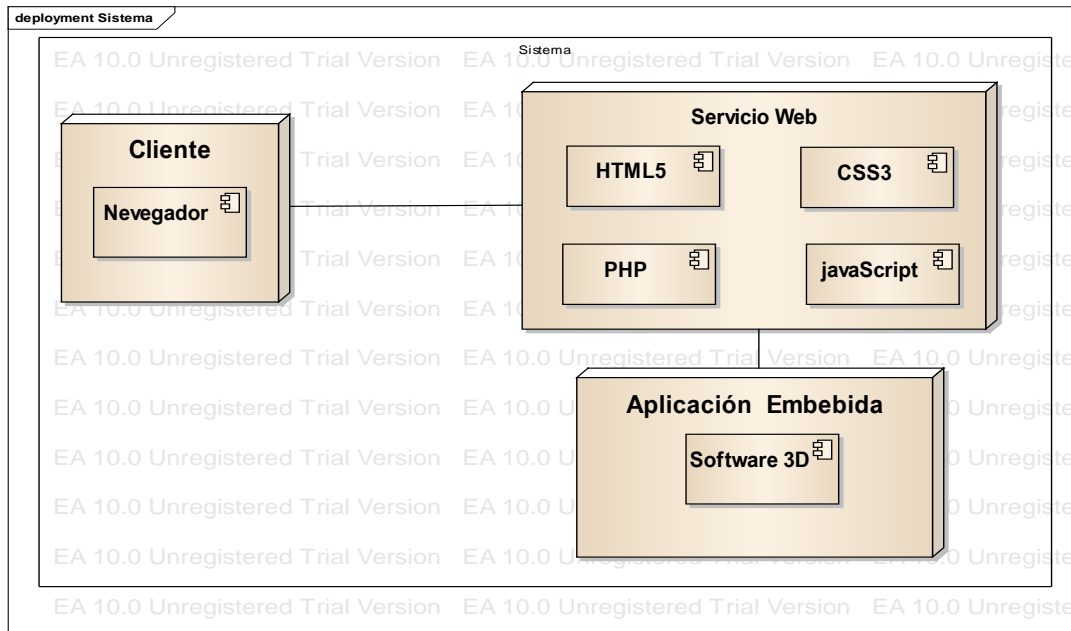


Figura 3.2 Modelo de despliegue del sistema de software.

Son muy pocos los profesionales de la salud que están capacitados para pensar analíticamente acerca de cómo debe funcionar los sistemas de software tecnológicos para el sector salud. Por lo tanto, a menudo es difícil para estos profesionales conocer las aportaciones que los enfoques y análisis de ingeniería de software pueden aportar. Por el contrario, los profesionales de la ingeniería de software a menudo tienen poca educación en el área del sector salud. [18] En el futuro, los sistemas de software proporcionarán capacitación y oportunidades para el desarrollo de la implementación de la ingeniería de software y tener sistemas de software que solucionen las necesidades de la enseñanza de los profesionales del sector salud.

3.1.2 Obtención y análisis de requerimientos

El modelado de sistemas es la actividad para identificar las características más relevantes, se analiza para conocer y mejorar el comportamiento del sistema original. Este proceso es diferente del modo basado en la investigación médica. El proceso de un ingeniero de software es comprometerse en un proyecto que mejoró la enseñanza para el profesional del sector salud. [18] Hay seis pasos fundamentales, que se realizan iterativamente:

- Definir el propósito y el alcance del sistema.
- Especificar, recopilar y elaborar los datos necesarios a través de métodos de recolección de datos.
- Diseñar, validar y verificar los modelos de sistema adecuados.
- Utilice el modelo para aprender sobre el comportamiento del sistema para encontrar la mejor alternativa de diseño.
- Utilice los resultados para determinar la forma de configurar el sistema para un mejor rendimiento.
- Desarrollar planes de ejecución, evaluación y coordinar su actuación.

Los diagramas de casos de uso son uno de los tipos de diagramas de UML que se utilizan para modelar los aspectos dinámicos de un sistema de software. Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. Los diagramas de caso de uso se emplean para modelar la vista de casos de uso de un sistema de software. Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Estos diagramas facilitan que los sistemas, subsistemas y clases sean abordables y comprensibles, al presentar una vista externa de cómo pueden utilizarse estos elementos en un contexto dado. [20] En la **figura 3.3** se muestra a continuación el diagrama de caso de uso del home del sistema de software, para mayor información se puede consultar en el **Anexo II todos los casos de uso del sistema de software**.

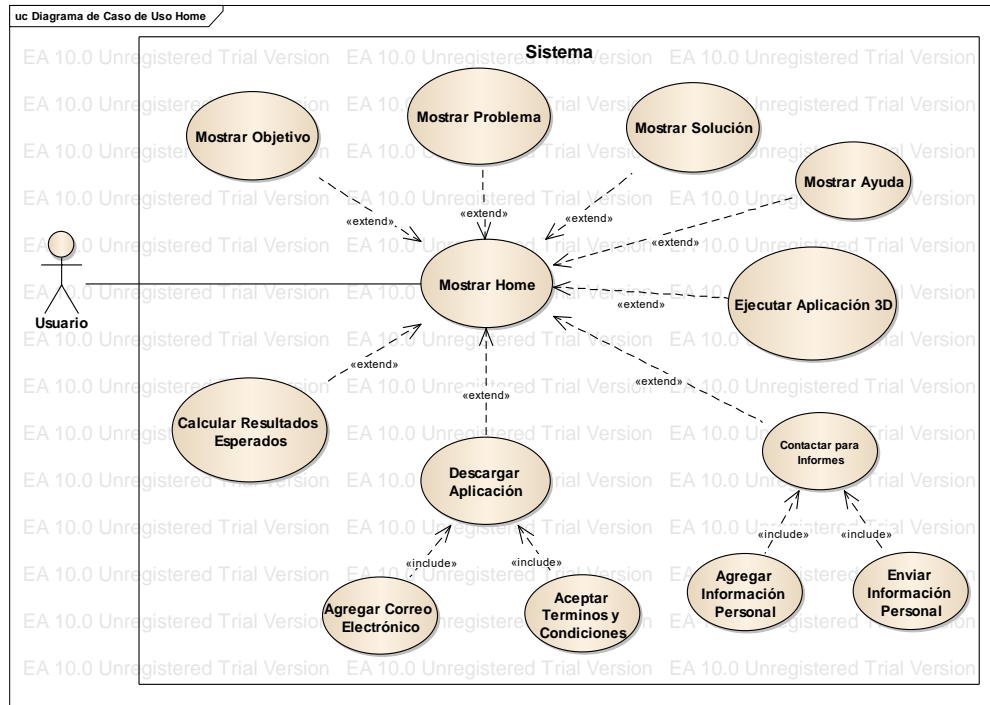


Figura 3.3 Diagrama de caso de uso del home del sistema de software.

Los diagramas de casos de uso son importantes para probar sistemas de software ejecutables a través de ingeniería directa y para comprenderlos a través de ingeniería inversa. [20] En la **figura 3.4** se describe el diagrama de caso de uso para el sistema de software 3D.

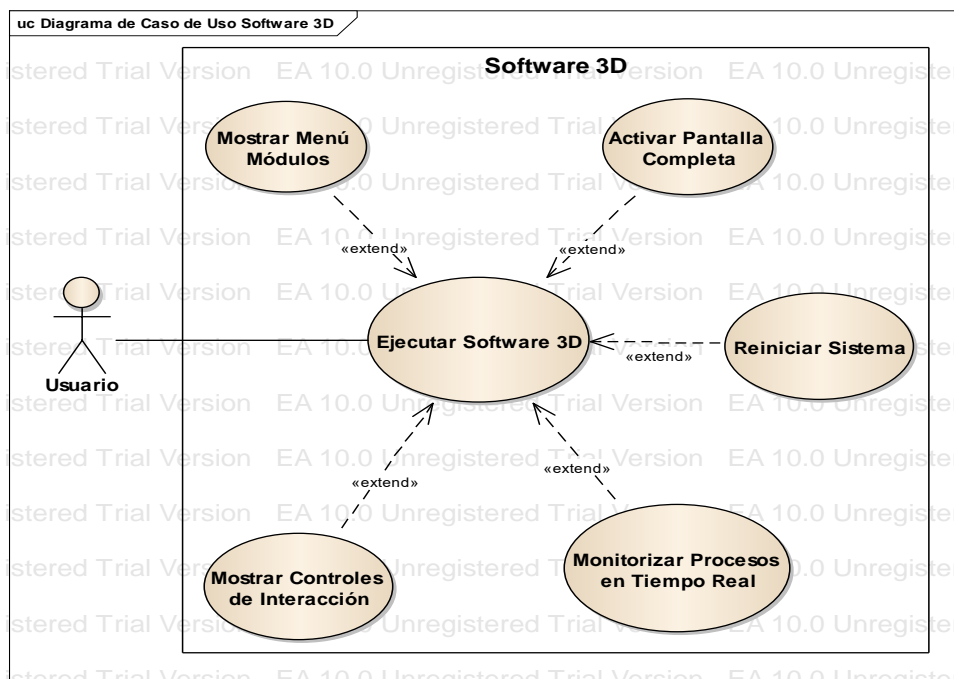


Figura 3.4 Diagrama de caso de uso del sistema de software 3D.

3.1.3 Especificación de requerimientos.

Los diagramas de actividades son uno de los tipos de diagrama UML que se utilizan para el modelado de los aspectos dinámicos de los sistemas. Un diagrama de actividades es fundamentalmente un diagrama de flujo que muestra el flujo de control entre actividades. Al contrario que un diagrama de flujo clásico, un diagrama de actividad muestra tanto la concurrencia como las bifurcaciones del control. Los diagramas de actividades se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar los pasos secuenciales de un proceso computacional. Con un diagrama de actividades también se puede modelar el flujo de datos entre los distintos pasos. Los diagramas de actividades pueden servir para visualizar, especificar, construir y documentar la dinámica de una sociedad de objetos, o pueden emplearse para modelar el flujo de control de una operación. Mientras que los diagramas de interacción destacan el flujo de control entre objetos, los diagramas de actividades destacan el flujo de control entre los distintos pasos. Una actividad es una ejecución estructurada en curso de un comportamiento. La ejecución de una actividad se descompone finalmente en la ejecución de varias acciones individuales, cada una de las cuales puede cambiar el estado del sistema o puede comunicar mensajes. Los diagramas de actividades no son sólo importantes para modelar los aspectos dinámicos de un sistema, sino también para construir sistemas ejecutables a través de ingeniería directa e inversa. [9] En la **figura 3.5** se muestra el diagrama de caso de actividad del home del sistema de software, en el **Anexo III** se encuentran todos los casos de actividad del sistema de software.

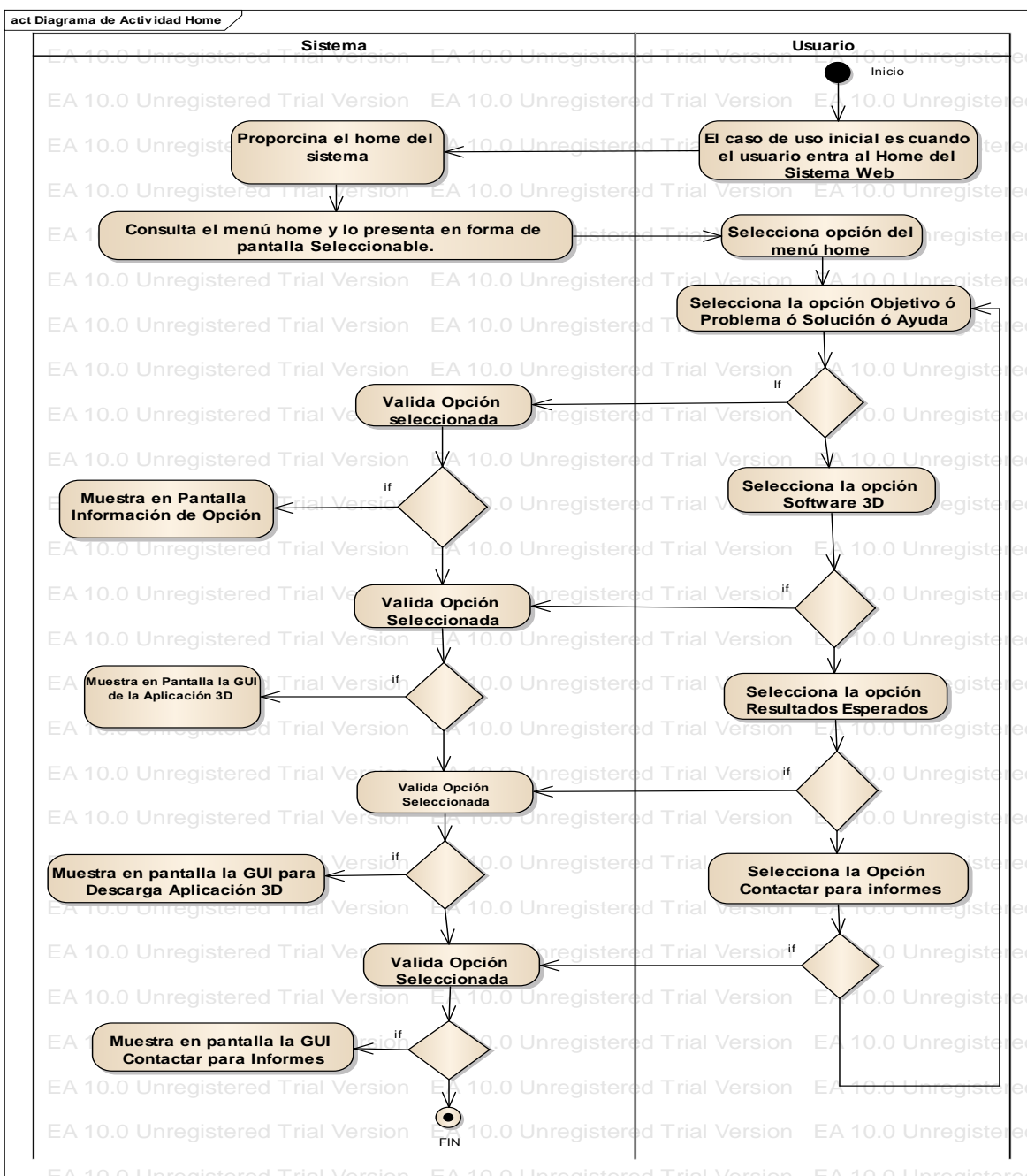


Figura 3.5 Diagrama de actividad del home del sistema de software.

3.2 Diseño e implementación de software

Un diseño de software, es una descripción de la estructura del software que se va implementar, los modelos y las estructuras de datos utilizados por el sistema de software, las interfaces entre componentes y, en ocasiones, los algoritmos usados. El diagrama sugiere que las etapas del proceso de diseño son secuencias, las actividades del proceso de diseño están vinculadas. En todos los procesos de diseño es inevitable la retroalimentación de una etapa a otra y la consecuente reelaboración del diseño. [9]

3.2.1 Diseño arquitectónico.

Es la estructura global del sistema de software, los principales componentes (llamados módulos), sus relaciones y cómo se distribuyen. El diseño arquitectónico se interesa por entender cómo debe organizarse un sistema de software cómo tiene que diseñarse la estructura global, es la primera etapa en el proceso del diseño del software, es el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes, estructurales y la relación entre ellos. Un modelo arquitectónico describe la forma en que se organiza el sistema de software como un conjunto de componentes en comunicación, es un proceso creativo en el cual se diseña una organización del sistema de software que cubrirá los requerimientos funcionales y no funcionales, ya que es un proceso creativo, las actividades dentro del proceso dependen del tipo de sistema, así como de los requerimientos específicos. Es útil pensar en el diseño arquitectónico como un conjunto de decisiones a tomar en vez de una secuencia de actividades. [9] En la *figura 3.6* se muestra el diseño arquitectónico del sistema de software biomédico.

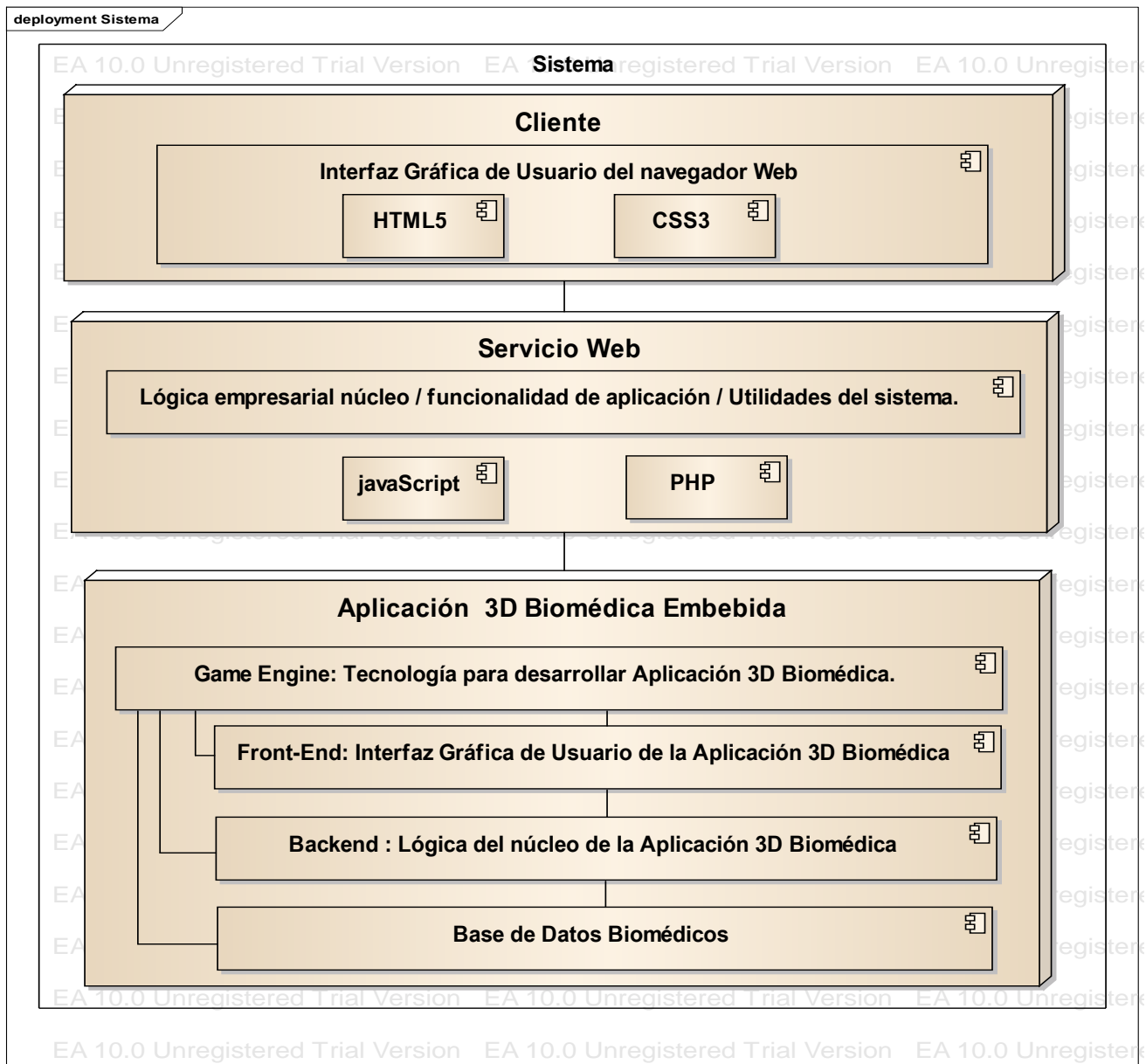


Figura 3.6 Diseño arquitectónico del sistema de software biomédico.

3.2.2 Diseño de interfaz gráfica de usuario.

El diseño de interfaz gráfica de usuario son las interfaces del sistema de software que contienen los componentes con los cuales se podrá interactuar, no deben presentar ambigüedades, deben ser interfaces precisas y los componentes deben ser factibles para los usuarios para evitar una curva de aprendizaje. [9] Es recomendable usar la metodología de Arquitectura de la Información e Usabilidad de sistemas de software para lograr una interfaz ideal. Una vez que se acuerdan las especificaciones de interfaz, los componentes se diseñan y se desarrollan de forma ocurrente. En la **figura 3.7** se muestra la interfaz graficas de usuario del home del sistema de software biomédico, en el **Anexo IV** se encuentran todas las interfaces graficas usuario del sistema de software biomédico.

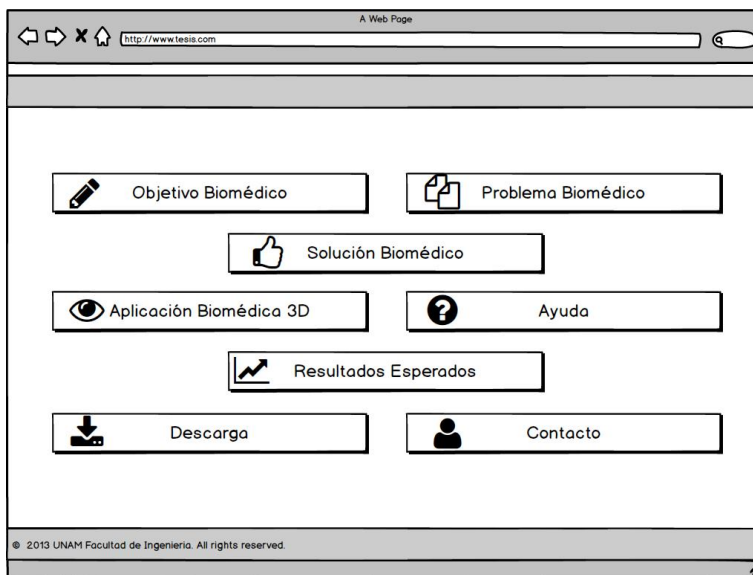


Figura 3.7 Diseño de interfaz gráfica de usuario para el home del sistema de software Biomédico.

3.2.3 Diseño de componentes de la interfaz gráfica de usuario de interacción.

El diseño de componentes son representaciones gráficas para indicar la funcionalidad de la acción que realiza el usuario, solo se toma como referencia de dato para representar la funcionalidad que se implementa en el sistema de software biomédico se muestra en la **figura 3.8**, al desarrollar el componente es normal que se deje un diseño específico, ya que para modificar el aspecto (*look and feel*) del sistema de software se tendrá que iniciar el requerimiento del componente con otro diseño. La navegación del sistema de software biomédico son componentes que permiten moverse por las diferentes secciones y poder re-direccionar, sin la sensación de haberse perdido. [17] Para conseguir la sensación de no haberse perdido, el sistema de software biomédico contempla un **menú de secciones** (*zona* donde está dividida la información contenida en el sistema de software biomédico) y **botones de acción** (elementos que permiten realizar acciones).

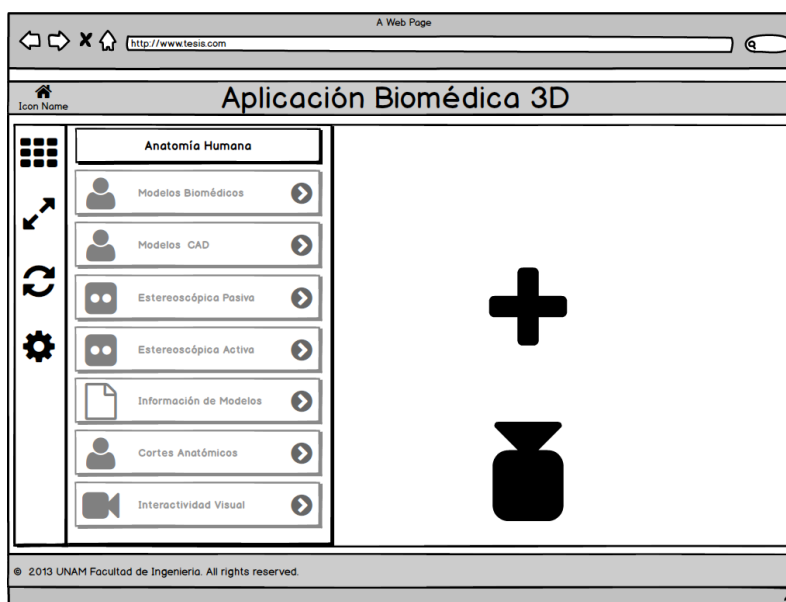


Figura 3.8 Diseño de interfaz gráfica de usuario del sistema de software biomédico.

3.2.4 Diseño de áreas de interacción

Son las zonas para realizar acciones o funciones en el sistema de software, la interacción, en este sentido, va desde acciones menores que pueden ser enlaces, uno de los elementos que se debe tener en cuenta, es que **las zonas de interacción son las que muestran botones**, es necesario que en las zonas de interacción haya la mínima información posible y que siempre sea la necesaria para llevar a cabo en forma adecuada la acción a que se refiera la interfaz. [17] El sistema de software biomédico 3-D, cuenta con los datos mínimos para visualizar. En la **figura 3.9** se muestra la interacción para activar la información de nombres anatómicos del sistema de software biomédico 3-D.

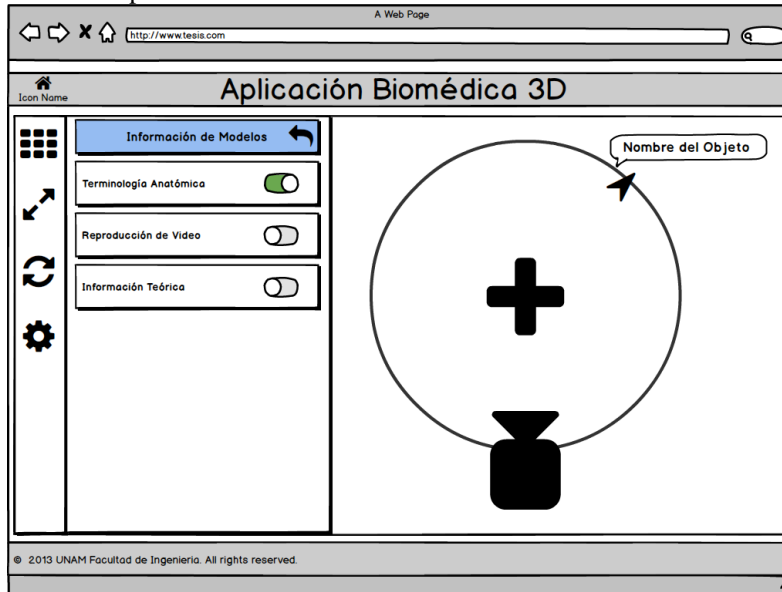


Figura 3.9 Diseño de interfaz gráfica de usuario para la interacción de nombres anatómicos.

3.2.5 Arquitectura del sistema de software biomédico 3-D

La arquitectura del sistema de software biomédico 3-D que se muestra en la **figura 3.10**, se organiza en capas de procesos que deben ser realizadas para el desarrollo del sistema de software biomédico.

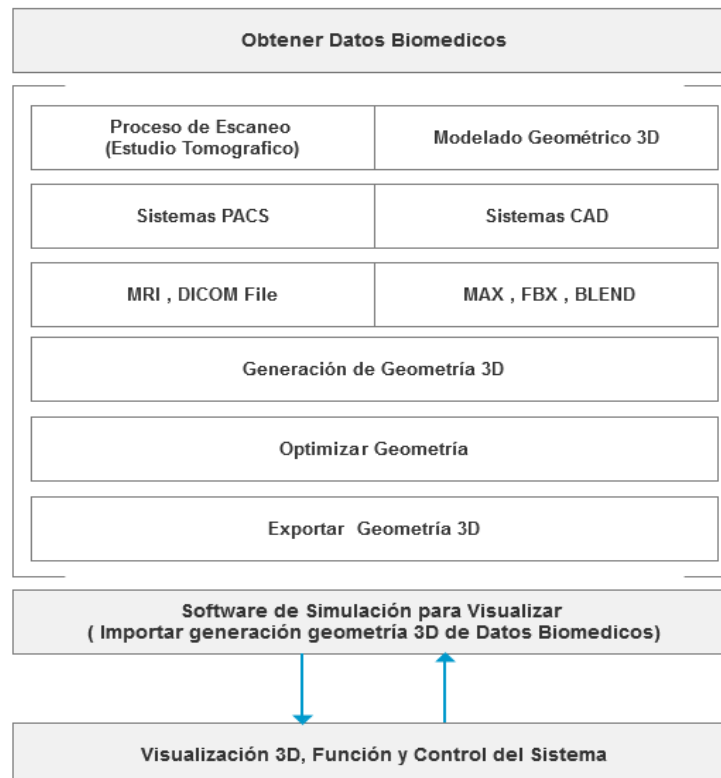


Figura 3.10 Arquitectura del sistema de software biomédico 3-D.

3.2.6 Especificación de base de datos.

El diseño de la base de datos, es en donde se diseñan las estructuras del sistema de datos y se representan en una base de datos, estas actividades conducen a un conjunto de salidas de diseño, el detalle y la representación del diseño de la base de datos varían considerablemente. Si se usa un enfoque dirigido por un modelo, dichas salidas serían sobre todo diagramas. [9] En la *figura 3.11* se muestra el diseño de la estructura de la base de datos para el sistema de software biomédico 3-D.

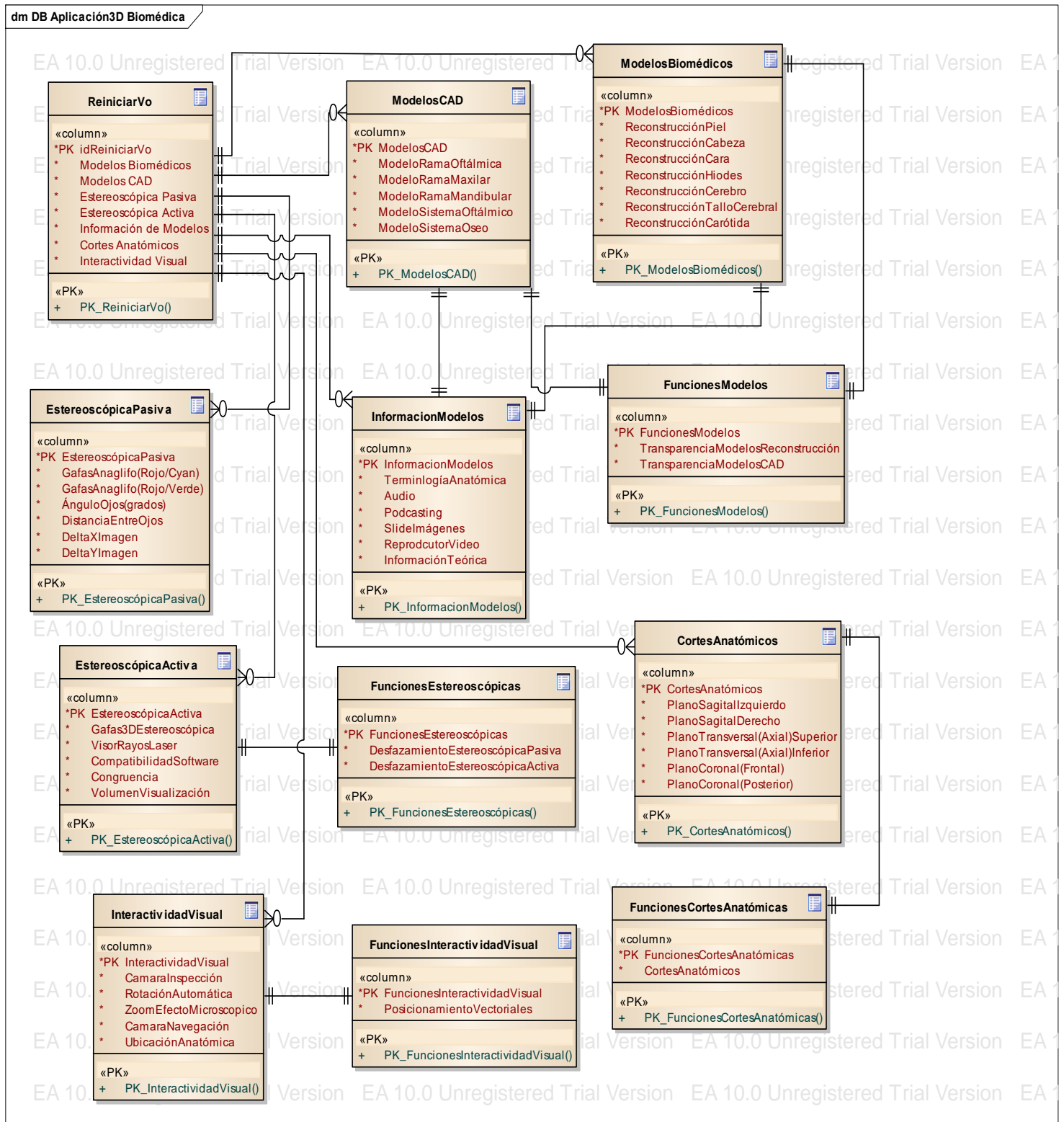


Figura 3.11 Diseño de base de datos del sistema de software biomédico 3-D.

3.3 Diseño de las estructuras de datos

El diseño de las estructuras de datos, es donde se especifican las estructuras necesarias para implementar en el sistema de software biomédico, a continuación se muestra las diferentes estructuras de datos para obtener los modelos geométricos tridimensionales requeridos:

1. **Estructuras de datos para modelos biomédicos** (Obtenidos de Reconstrucción Geométrica por Tomografía Axial Computarizada, “TAC”). En la *figura 3.12* se muestra la estructura de datos biomédicos.

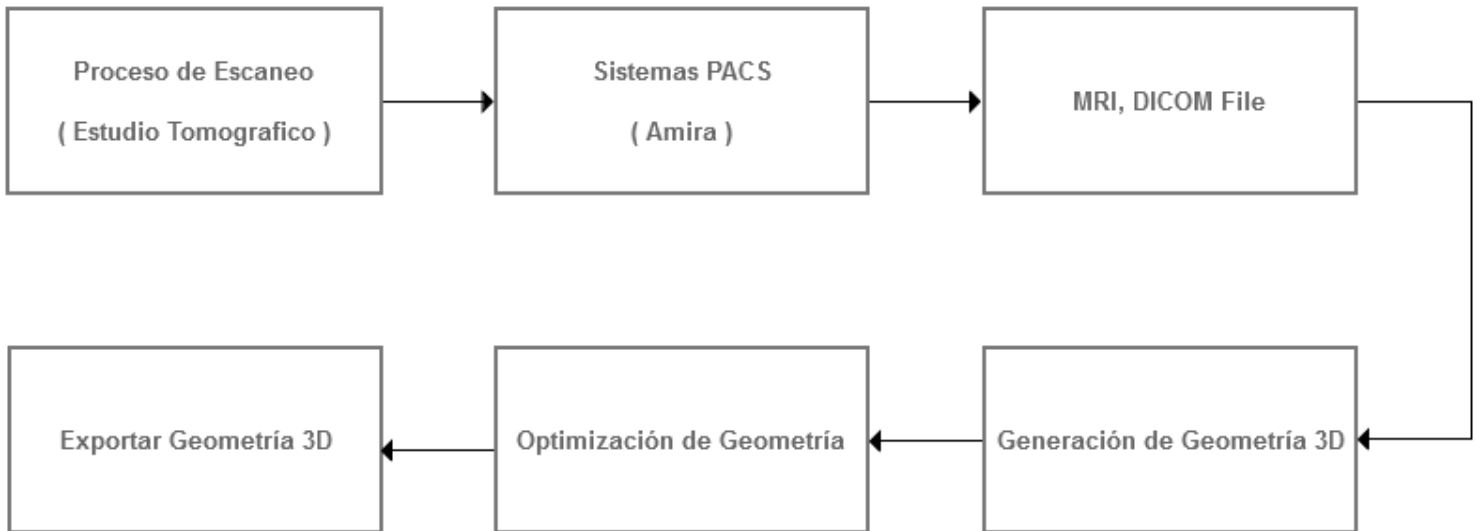


Figura 3.12 Estructura de datos biomédicos.

2. **Estructuras de datos para modelos no biomédicos** (Obtenidos de Diseño Asistido por Computadora, “CAD”). En la *figura 3.13* se muestra la estructura de datos no biomédicos.

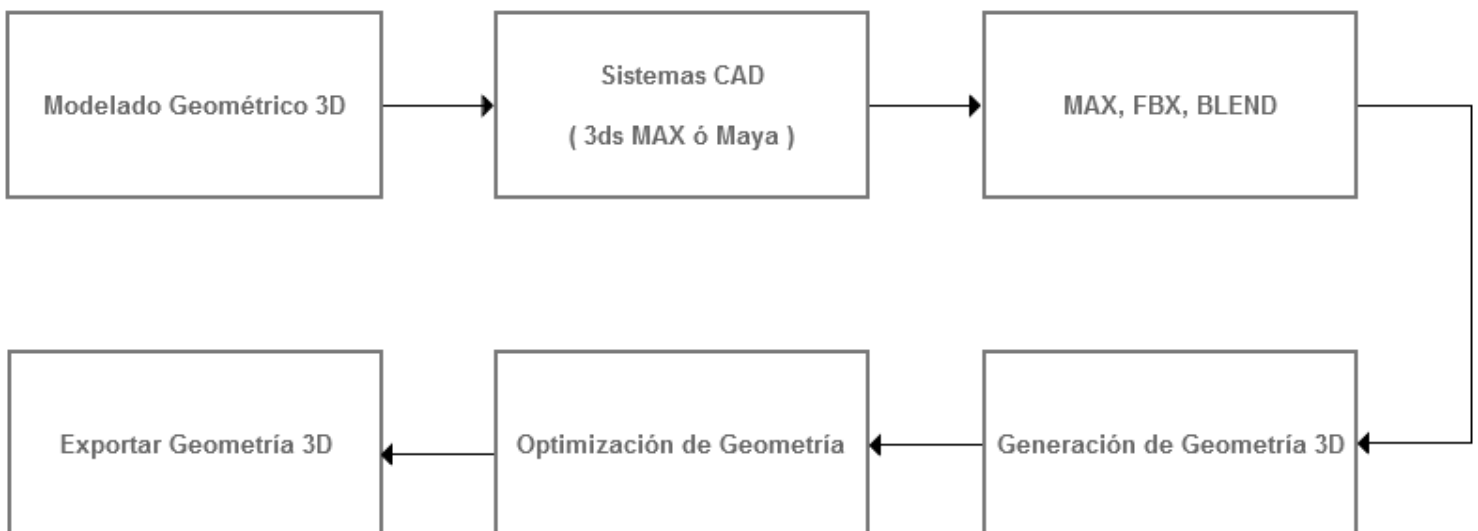


Figura 3.13 Datos no biomédicos.

3.3.1 Obtención de las estructuras de datos biomédicos.

El proceso de datos biomédicos se obtiene con datos requeridos de un estudio tomográfico, que es una serie de imágenes médicas digitales almacenadas como un archivo DICOM (en sus siglas en inglés, *Digital Imaging and Communication in Medicine*), este archivo representa la zona principal a reconstruir, zonas representadas en color blanco y negro, las zonas de color blanco representan las partes reales anatómicas con mayor densidad y las zonas en color negro representan el vacío o las partes anatómicas con menor densidad. Cuando la tomografía muestra diferentes zonas anatómicas y existe información de éstas que no se quiere reconstruir, se debe realizar un filtro de la información que no se requiera, el filtrado es la edición en cada tomografía donde aparece la información no requerida. Para la obtención de los datos biomédicos se utilizó los datos reales de un paciente de sexo masculino, que se le realizaron 500 cortes de aproximadamente 1mm de distancia entre corte, anteriormente al sujeto de estudio se le administro un medio de contraste, sustancia líquida que refleja los rayos del tomógrafo, mejora la obtención de la información y posteriormente una reconstrucción 3D de calidad, con dimensiones y proporciones reales.

La reconstrucción 3-D, se basa en usar el Sistema de Archivado y Transmisión de Imágenes (*PACS, en sus siglas en inglés, Picture Archiving and Communication System, en este caso "Amira 4.0"*), su principal función es generar superficie 3-D a partir de las imágenes médicas (archivos en formato DICOM, etc.), contiene funciones para visualizar las reconstrucción 3-D y poder generar la superficie deseada, este proceso se logra debido a que precarga en un módulo del PACS que ejecuta el algoritmo de reconstrucción 3-D (*Volumen Rendering, representación de Volumen*), esté algoritmo realiza la gestión de superficie de las áreas correspondientes a generar y el producto final es la digitalización de objetos o estructuras biomédicos 3-D. En la *figura 3.14* se muestra el proceso de obtención de datos biomédicos que se utilizó para la gestión de la aplicación tridimensional del sistema web biomédico, en el *Anexo V se encuentran la descripción de la herramienta Amira para la obtención de los datos biomédicos*.

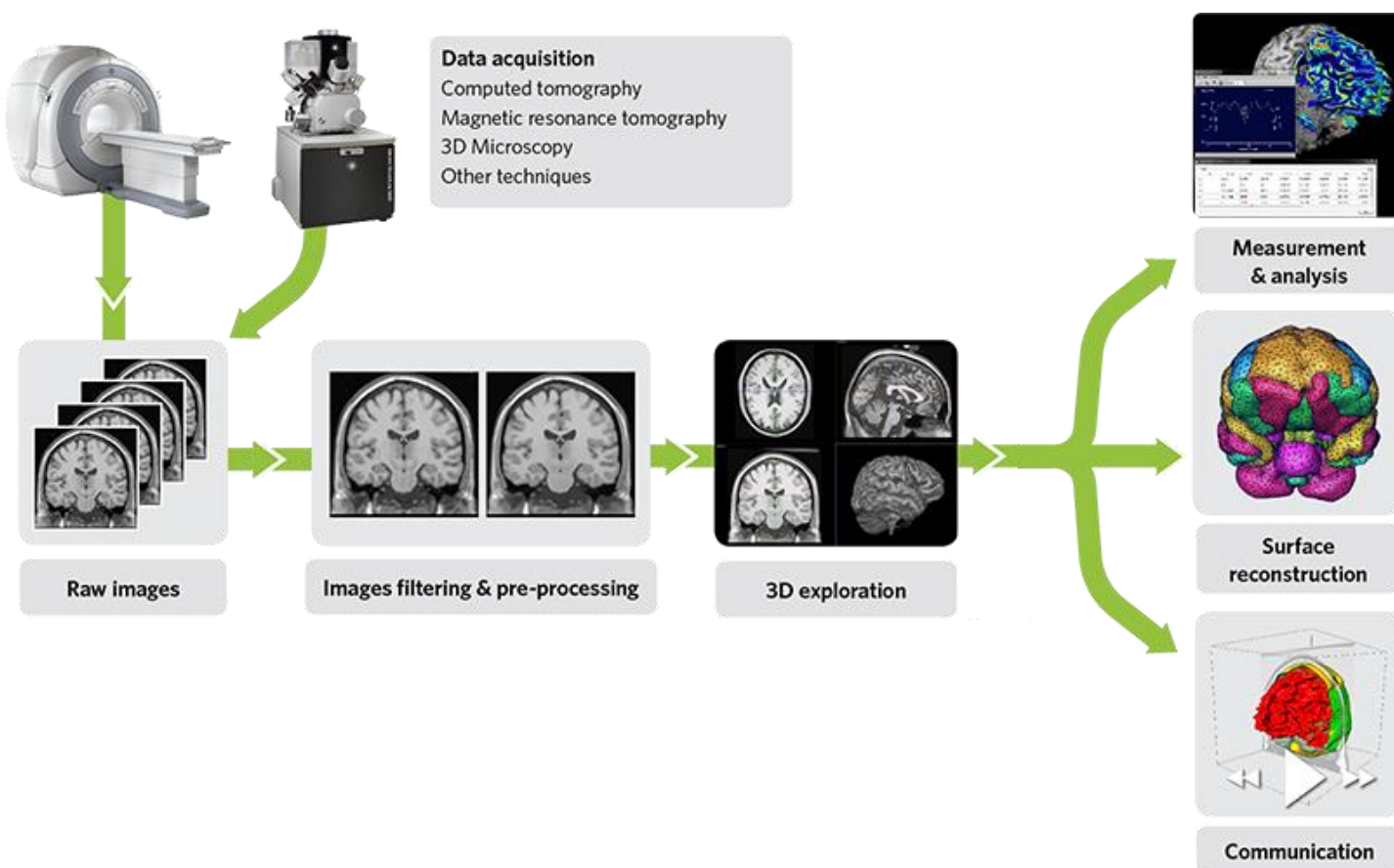


Figura 3.14 Proceso de obtención de datos biomédicos.

El proceso para obtener los datos biomédicos, se realiza mediante la ejecución de “*Amira 4.0*”, una vez que el programa este cargado completamente, nos dirigimos a abrir la carpeta que contiene las “*imágenes DICOM*” y con el bloque de construcción llamado OPENDATA, seleccionamos todas las imágenes DICOM y posteriormente descargamos, este proceso requiere un cierto tiempo para descargar las imágenes, depende del número de imágenes DICOM y los recursos del computador. En la *figura 3.15* se muestra la descarga de archivos DICOM en AMIRA 4.0.

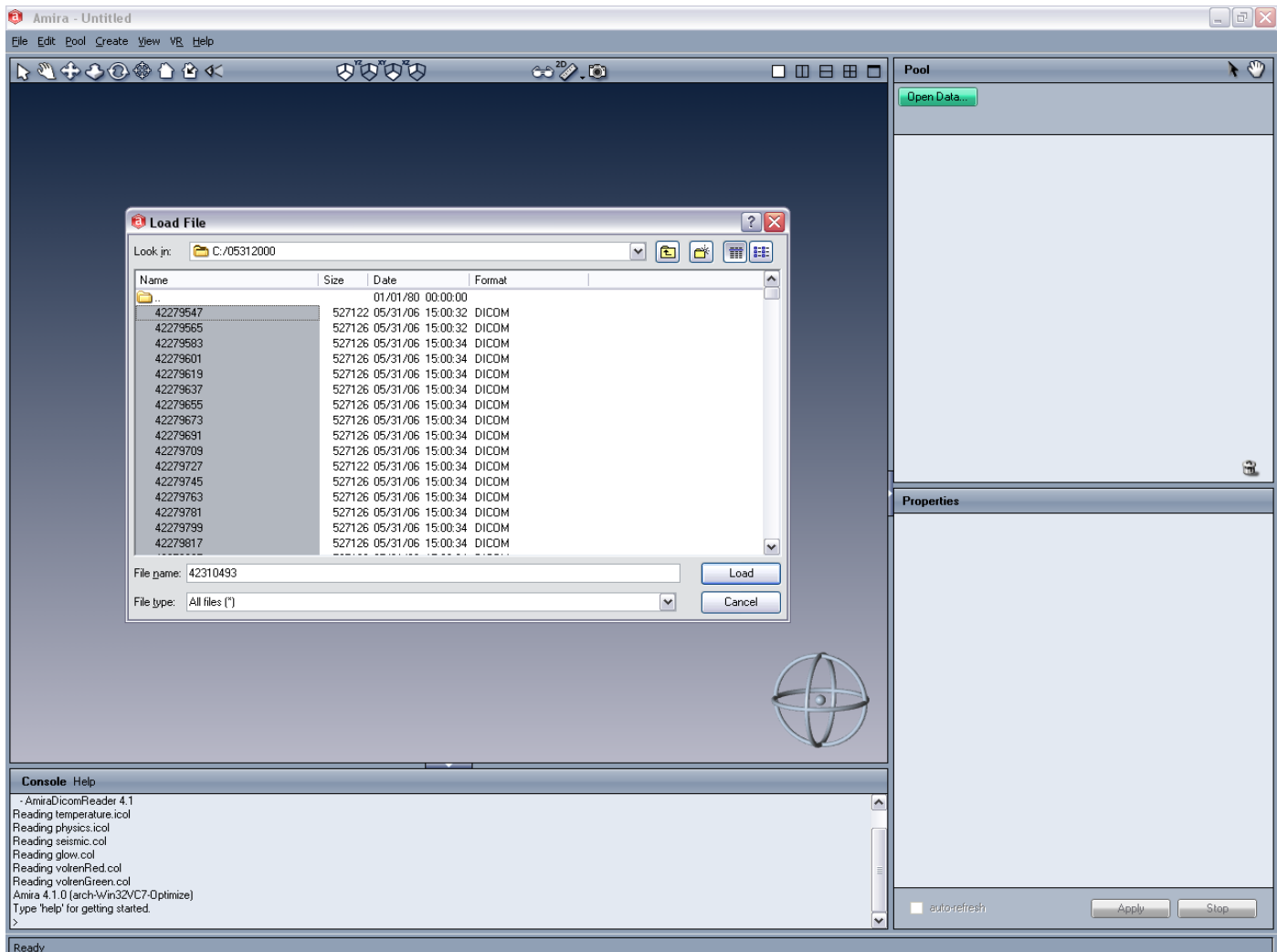


Figura 3.15 AMIRA 4.0, descarga de archivos DICOM.

Cuando el proceso de carga de imágenes DICOM termina, se debe activar el bloque que contiene la información de todas las imágenes DICOM en forma de pila, una vez activado posteriormente se debe activar otro bloque llamado “*ISOSURFACE*” el cual se dedica a la construcción de superficie, este bloque debe ser asociado con el bloque principal, cuando ambos bloques se encuentran asociados, debemos configurar algunos parámetros (Draw Style : shaded, Theshold : 500, Options: compactily ON, Auto-refresh: ON) y filtros (función de optimización de polígonos, se logró reducir la cantidad de polígonos sin perder la calidad) que le pertenecen al bloque Isosurface, esta configuración nos permite generar la superficie 3D de calidad deseada logrando disminuir el tamaño del dato biomédico, una interacción de estructura sin tener problemas de consumo de procesamiento en el momento de realizar los cálculos gráficos. En la *figura 3.16* se muestra la reconstrucción de datos biomédicos 3D.

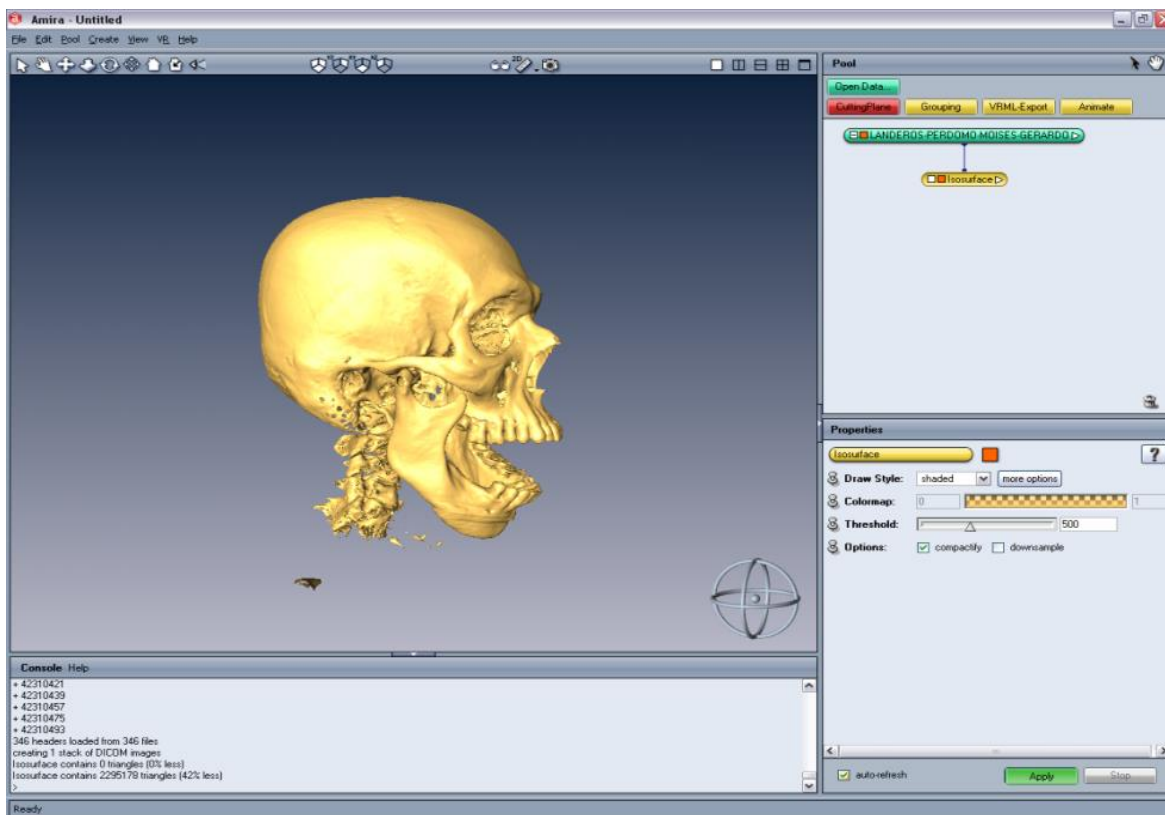


Figura 3.16 AMIRA 4.0, reconstrucción de datos biomédicos 3-D.

Cuando el proceso de generar el dato biomédico esté terminado, se prosigue al proceso de exportar el dato generado, para esto se debe activar el módulo de **VRML Exporter** (con extensión **.Obj**, **VRML**, etc.), una vez activado es necesario asociar el bloque al de **ISOSURFACE**, una vez asociados, en el bloque VRML Exporter aparece las propiedades de selección de extensión y forma de exportar los datos de reconstrucción biomédicos. En la **figura 3.17** se muestra como se exportan los datos biomédicos 3D.

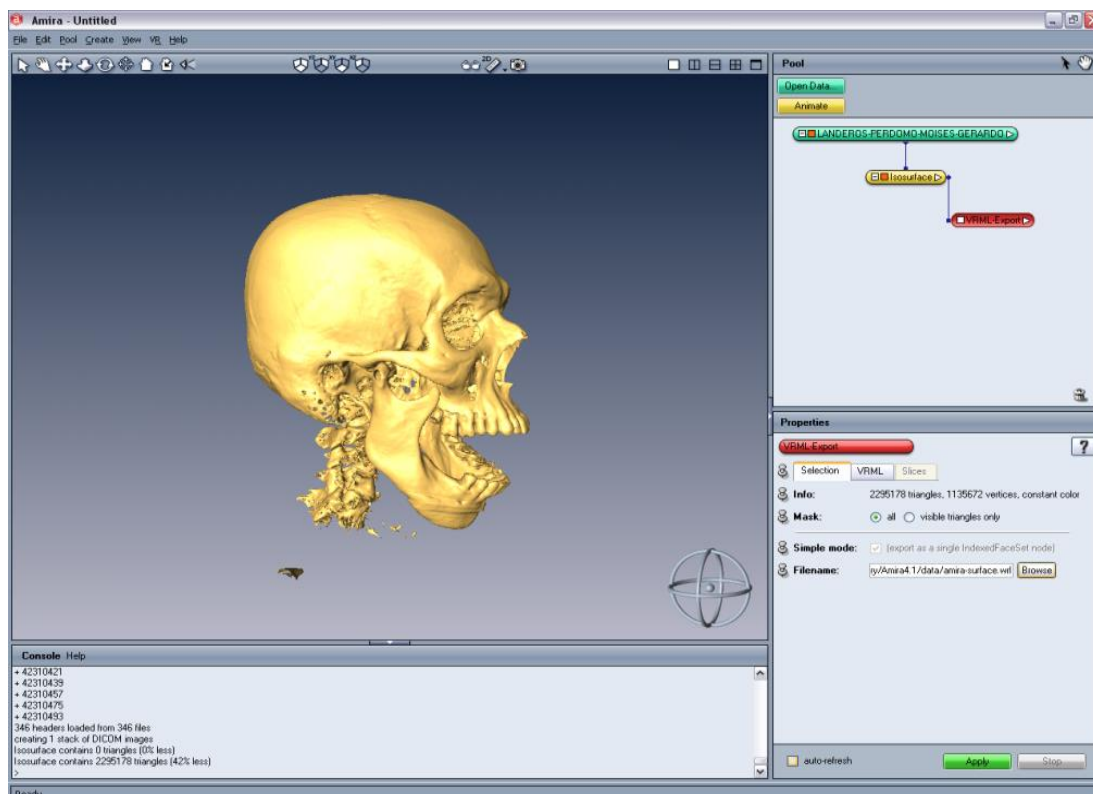


Figura 3.17 AMIRA 4.0, exportar datos biomédicos 3-D.

Una vez exportados los datos generados, podemos notar en la carpeta donde se guardó, que es un archivo con extensión .obj, este archivo que contiene los datos biomédicos son importados en otra herramienta tecnológica orientada al Diseño Asistido por Computadora (**CAD**, *por sus siglas en inglés, Computer Aided Desing, 3dsMax*), se utiliza para realizar la eliminación de segmentos no deseados, ejecutar otra capa de optimización de polígonos, modificaciones en la reconstrucción 3-D, etc. En la **figura 3.18** se muestra la importación de los datos biomédicos en 3ds Max.

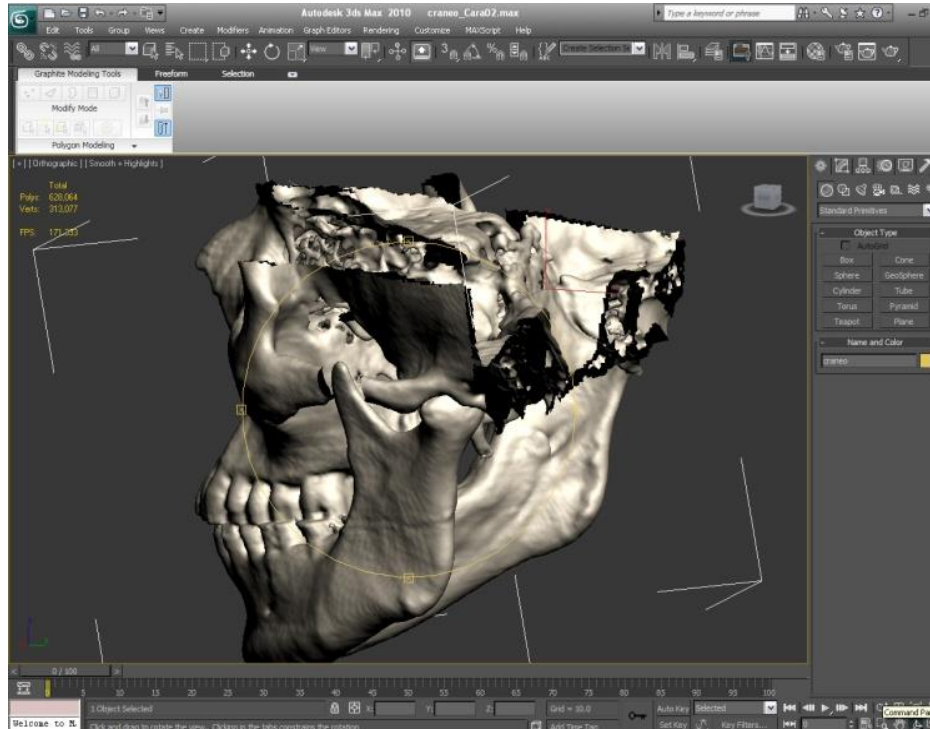


Figura 3.18 3ds Max, modificaciones en los datos biomédicos 3-D.

Una vez terminadas las modificaciones, el objeto se exporta nuevamente a un archivo de tipo "**DirectX**" con extensión .x , que posteriormente se importará y usará como bloques de construcción en un entorno o la plataforma para desarrollo (**GAME ENGINE**) y poder programar la aplicación que realiza la ejecución grafica (GPU, Graphics Processing Unit) de estos datos biomédicos. En la **figura 3.19** se muestra los datos biomédico listos para ser exportados al motor gráfico para la programar el sistema de software.

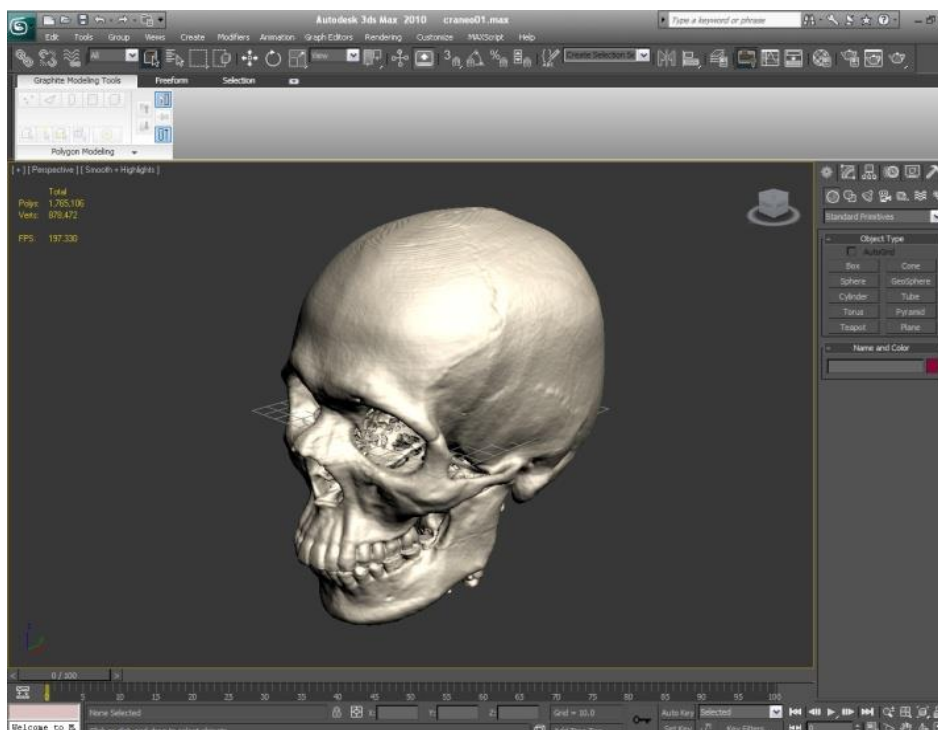


Figura 3.19 3ds Max, dato biomédicos 3-D para exportar.

3.3.2 Obtención de las estructuras de datos no biomédicos

El proceso para el desarrollo de datos no biomédicos, se realiza mediante diseño asistido por computadora (por sus siglas en inglés, CAD Computer Aided Design) de las zonas anatómicas con baja densidad, que son partes complicadas de reconstruir biomédicamente en 3-D, por esta complejidad se utiliza la metodología CAD. La generación de las estructuras geométricas, se obtienen a partir de información como: *Dibujos, Diagramas, Esquemas, Fotos, Modelos reales, Imaginación, Microscopía e Investigación teórica*. Esta información se utilizan como referencia, base o fundamento para la reconstrucción geométrica con la mayor calidad y realidad posible, que consiste en reconstruir superficie poligonal de cada uno de las zonas de interés, estas estructuras geométricas se realizan mediante formas geométricas simples como *planos, cajas, esferas, cilindros, etc.* o mediante formas compuestas, realizando operaciones booleanas, extrusiones, etc. La unidad mínima para definir un plano es un triángulo y sirve para medir la perfección de cualquier estructura geométrica, es decir, que mientras un objeto tenga más triángulos, mayor será el detalle de la estructura geométrica. Lo único que hay que cuidar al realizar la estructura geométrica, es encontrar el balance entre la calidad o apariencia de las estructuras y la eficiencia debido al número de polígonos que tengan todas las estructuras geométricas que en conjunto formen el mundo virtual. Es muy importante que las estructuras geométricas se vean lo mejor posible, sin que afecte el rendimiento en el tiempo de interacción en tiempo real. En la *figura 3.20* se muestra el proceso de obtención de datos No biomédicos que se utilizó para la gestión del sistema web biomédico, en el *Anexo V se encuentran la descripción de la herramienta 3dsMaxStudio para la obtención de los datos No biomédicos*.

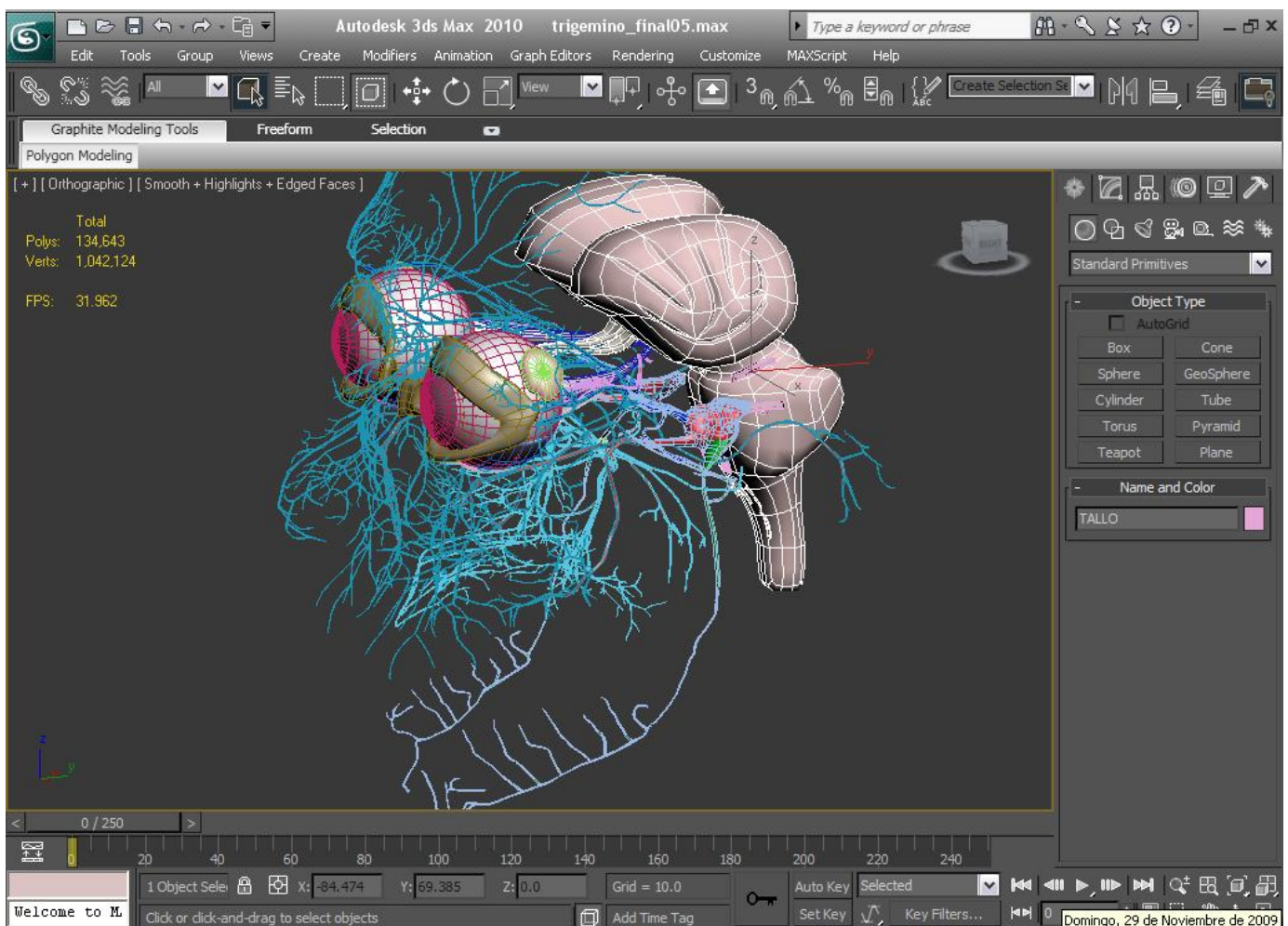


Figura 3.20 Datos no biomédicos (CAD, Computer Aided Design).

El desarrollo de los datos no biomédicos se inicia generando superficie geométrica como se comentó anteriormente, se utiliza la metodología de Diseño Asistido por Computadora, con la herramienta tecnológica 3ds Max Studio, ya que contiene una variedad de herramientas digitales para interacción y generación de superficie poligonal, que permite trabajar de mejor forma con modelos 3-D que tienen miles y millones de polígonos, que son necesarios para generar las estructuras más reales, existen una variedad de tecnología para modelar (3dsMax, Maya y Blender), pero la que contiene mayor usabilidad para trabajar con modelos médicos y que esto hace más complejo manipular en un entorno tridimensional, 3ds max permite esa flexibilidad para desarrollar las estructuras geométrica corporales con una complejidad real al requerimiento del objetivo.

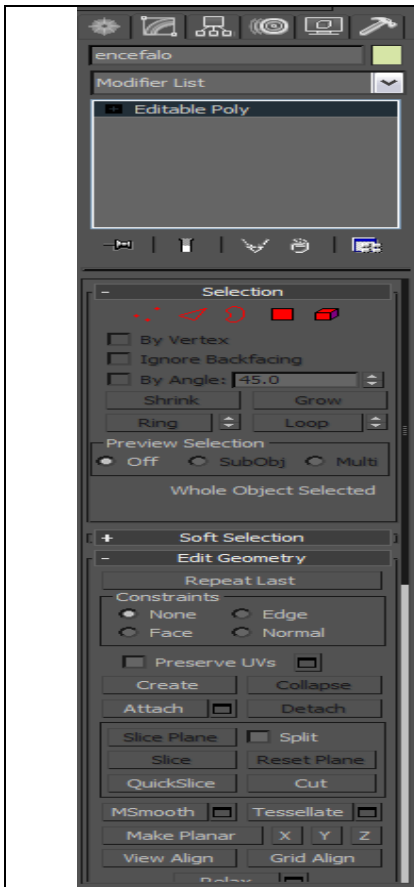


Figura 3.21 Interfaz del panel de comandos para desarrollar los datos no biomédicos.

Los controles del panel de comandos, se muestran en la **figura 3.21**, varían según el tipo de objeto creado. Algunos controles siempre están presentes y otros son comunes a diversos tipos de objetos. El panel de comandos para crear cuenta con un conjunto de características básicas comunes a todos los tipos de objetos, las cuales son:

Categoría: Los botones superiores acceden a las siete categorías básicas de objetos. El campo de texto desplegable que se encuentra debajo de los botones de categoría accede a las distintas sub categorías dependiendo de la categoría seleccionada.

Tipo de Objeto: Se muestra el conjunto de objetos básicos que hay en cada categoría.

Nombre: Asignación de nombre del objeto para identificarlo.

Color: Asignación de color para trabajar el objeto.

Método de Creación: En muchas ocasiones hay más de una forma de crear un objeto, tal es el caso de crear un objeto mediante el centro a la arista del mismo. Si se desea cambiar el método, ha de hacerse antes de crearlo.

Parámetros: La mayoría de los objetos presentan esta persiana, donde se muestran los parámetros de creación, es posible definir algunos parámetros antes de crear el objeto, mientras que otros sólo permiten ajustarlos en los ya creados.

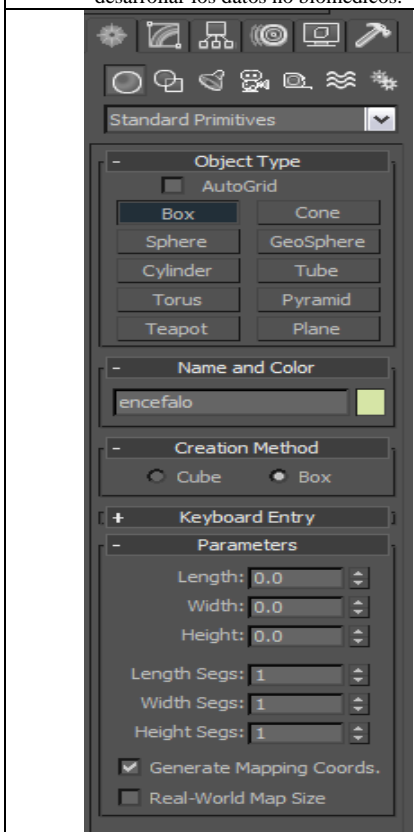


Figura 3.22 Categorías de objetos.

Categorías de Objetos: En el panel de comandos se muestra en la **figura 3.22**, se presentan serie de categorías básicas, de las cuales algunas se utilizaron.

Geometría: Presenta una serie de sub categorías como: primitivas estándares (cajas, esferas, prismas, cilindros, conos, objetos de composición, etc.).

Formas: Sólo incluyen una sub categoría, que es splines, donde se incluyen objetos 2D como línea, círculo, arco, etc. que al unirlos se crean objetos 3D mediante extrusión.

Luces: Sólo tiene una sub categoría que es predeterminado. Se presentan algunos tipos como: **Omni**, direccional, Foco con objetivo y Foco libre.

Cámaras: Al igual que las luces tiene una sola sub categoría. Las cámaras proporcionan un punto de vista sobre la escena. Puede ser **free camera o target camera**. Algo importante es la creación de luces y cámaras dentro de 3D Studio MAX, solo se tienen como finalidad de apoyo. La integración final de estos objetos se desarrollará con el Game Engine que será el entorno para el desarrollo de la aplicación 3D Biomédica.

SUBCATEGORÍAS: Dependiendo de la categoría previamente seleccionada, va a depender el número y tipos de sub categorías que se tengan como opciones. Para la categoría de geometría, se hace uso de la sub categoría primitivas estándar para la creación de objetos primitivos (cajas, cilindros, esferas, conos, etc.).

3.3.3 Desarrollo de materiales para el texturizado de los datos no biomédicos.

EL desarrollo de los datos no biomédicos se utiliza el proceso de texturizado, que consiste en aplicar a cada triángulo (plano o polígono) un aspecto, que se denomina textura. Las tecnologías de reconstrucción geométrica incorporan diferentes texturas para simular casi cualquier superficie. Al aplicar una textura, a cada píxel de cada triángulo o polígono se le asigna un color RGB (Red - Rojo, Green - Verde y Blue - Azul).

Las tecnologías de reconstrucción geométrica (3d studio max, maya, blender, etc.) codifican cada píxel con cuatro valores, tres de los cuales son el color y el cuarto aporta la información de transparencia, denominado “canal alpha”. Ese cuarto valor es especialmente indicado para producir efectos de transparencia, para superficies que se necesite ver del otro lado.

El desarrollo de materiales para los datos no Biomédicos se utilizó una herramienta tecnológica para la edición de imágenes *Photoshop CS5*, donde se diseña los materiales o texturas que son necesarias para alcanzar el realismo requerido, esta herramienta es seleccionada porque los resultados son de nivel tan alto que con ningún otro programa (gimp, etc) se lograría alcanzar, es el más completo, permite manejar más fácilmente sus herramientas y permite resultados inmediatos de una mejor calidad de gráficos .

El desarrollo de materiales son aplicaciones informáticas en forma de pintura y fotografía, se puede editar las imágenes, retocar fotografías y poder utilizarlas como material, el formato de salida de los materiales pueden ser *jpeg, png, bitmap, etc.* En la *figura 3.23* se muestra el proceso de obtención de materiales para texturizado de los datos no biomédicos que se utilizó para la gestión del sistema web biomédico, en el *Anexo V se encuentran la descripción de la herramienta PhotoShop CS5.*

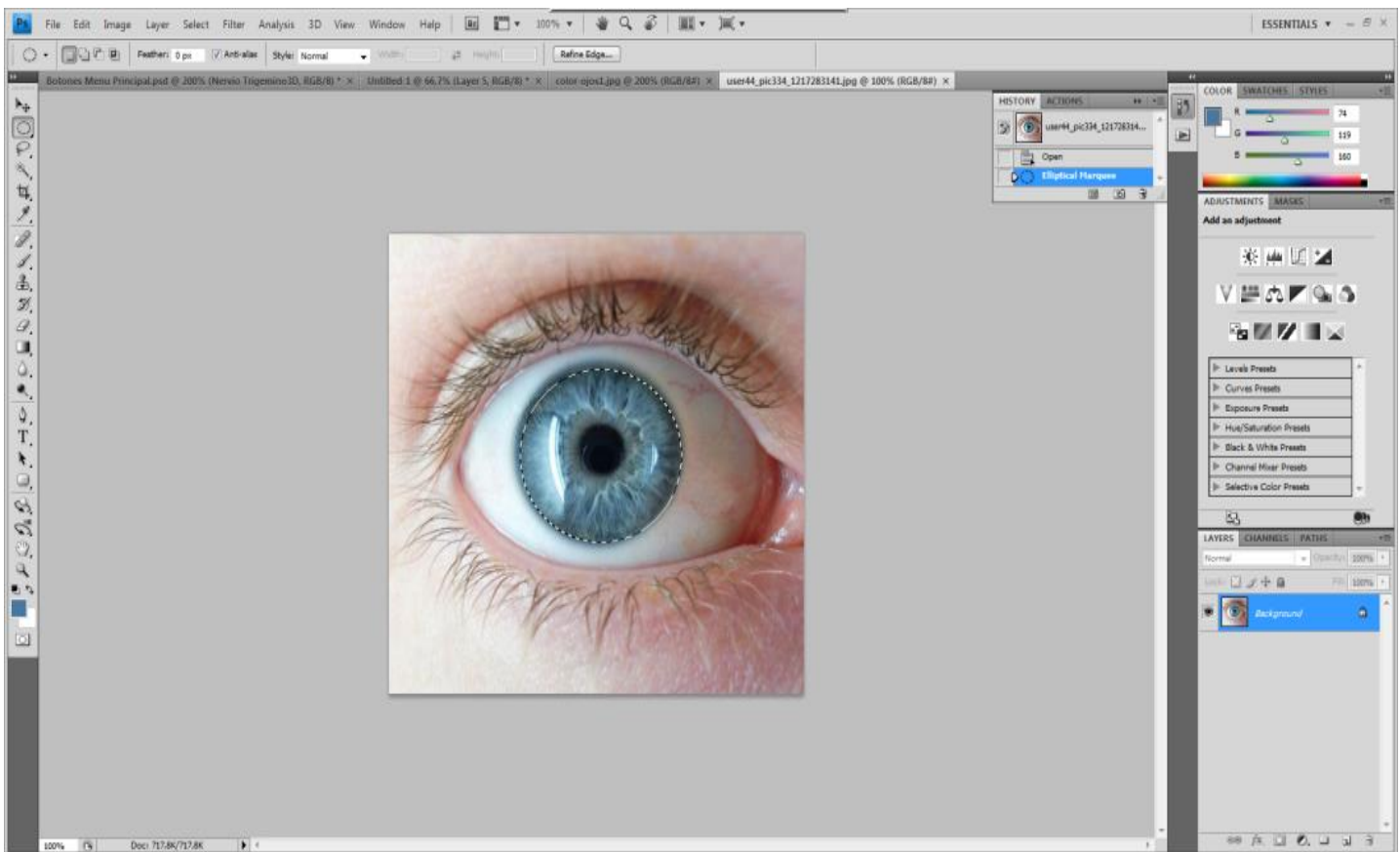


Figura 3.23 Desarrollo de materiales para texturizado de los datos no biomédicos.

La aplicación de las coordenadas de mapeo de un material gráfico 2-D a un objeto 3-D, se utiliza el modificador Unwrap UVW que contiene 3d Studio Max, este controla el mapeo y procesa los materiales que aparecen sobre la superficie del objeto 3D. Las coordenadas de mapeado se utilizan para definir el modo en que la textura de un mapa se alinea y se distribuye en el objeto. Este tipo de coordenadas se identifica con las letras U, V y W, correspondientes a la dirección horizontal, la dirección vertical y la profundidad respectivamente.

Las coordenadas de mapeo pueden controlarse mediante varios modificadores pero el que se utilizó fue Unwrap UVW. Este modificador, incluido en la categoría UV, permite asignar mapas planos y editar independientemente las coordenadas UVW de su objeto.

Las características principales son que este modificador tiene un editor para que el modelo pueda convertirse en un tipo de malla plana como si fuera un pedazo de piel, teniendo de esta forma la geometría se pueden seleccionar los vértices y acomodar la posición respectivamente a la textura, para tener un efecto adecuado de mapeo. En la **figura 3.24** se muestra el proceso de texturizado geométrico.

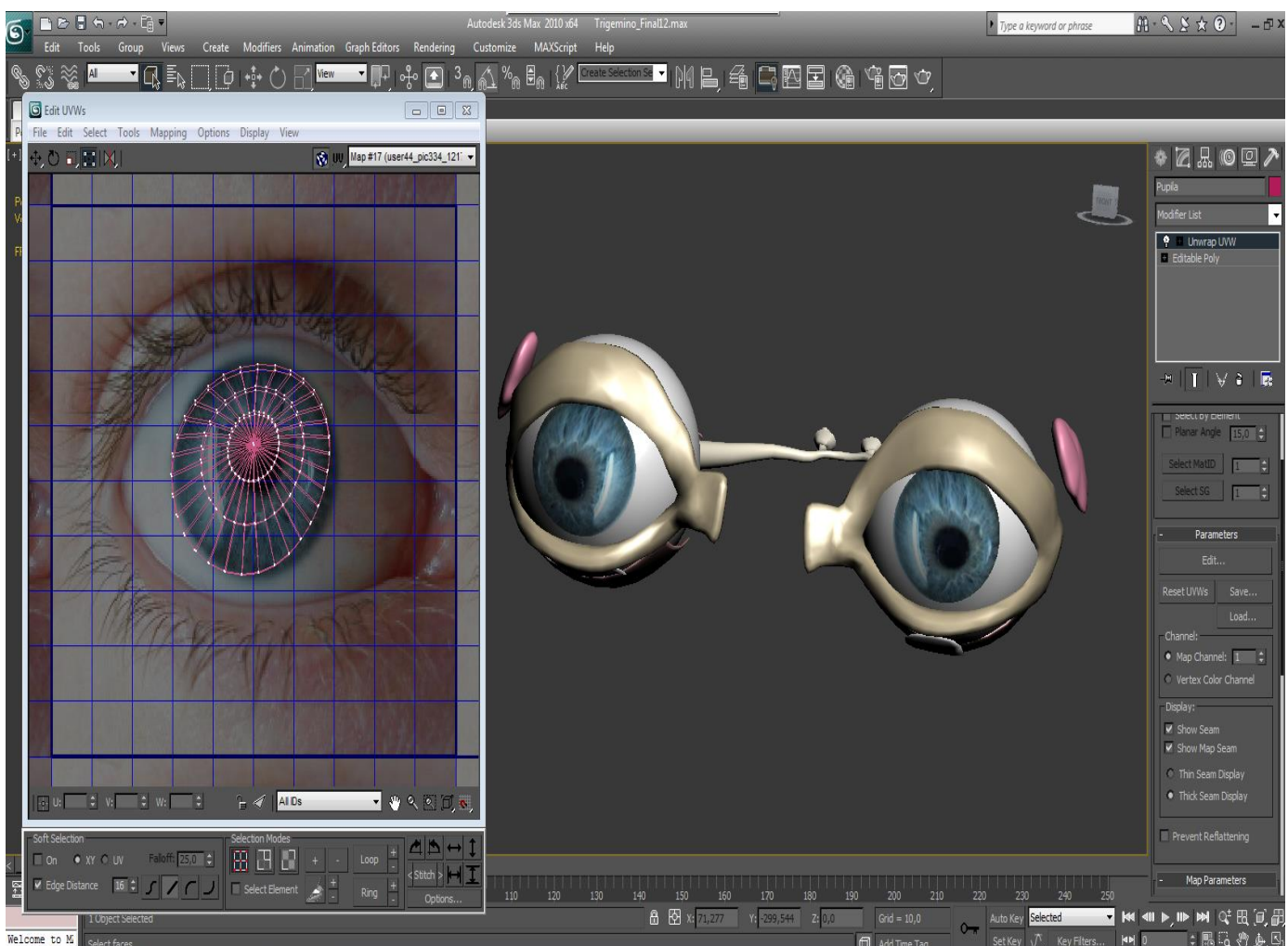


Figura 3.24 Desarrollo de materiales para texturizado de los datos no biomédicos.

3.3.4 Exportar datos no biomédicos para desarrollar lógica

La exportación para desarrollo de los datos no Biomédicos se realiza desde la herramienta tecnológica de 3d Studio Max a un entorno de desarrollo **GameEngine**, este entorno posteriormente utiliza los datos para generar el sistema de software 3-D biomédico.

Los formatos de archivos que el entorno de desarrollo que utiliza son varios (*.3ds (Pertenece a 3dsMax), *.X (Pertenece a DirectX), etc), en ocasiones para poder exportar estos formatos para que el entorno de desarrollo los pueda abrir o importar es necesario descargar los plug-in correspondientes para exportar el desarrollo de los datos no biomédicos, estos plug-ins los podemos encontrar en la web para realizar la descarga.

Una vez instalados los plug-ins con los modelos terminados (malla completa, texturas con mapeo), podemos exportar nuestros modelos 3-D eligiendo la opción exportar del menú archivo (File) y guardando el proyecto como archivo (*.3ds (Pertenece a 3dsMax), *.X (Pertenece a DirectX), etc). En la **figura 3.25** se muestra la exportación de los datos al motor gráfico.

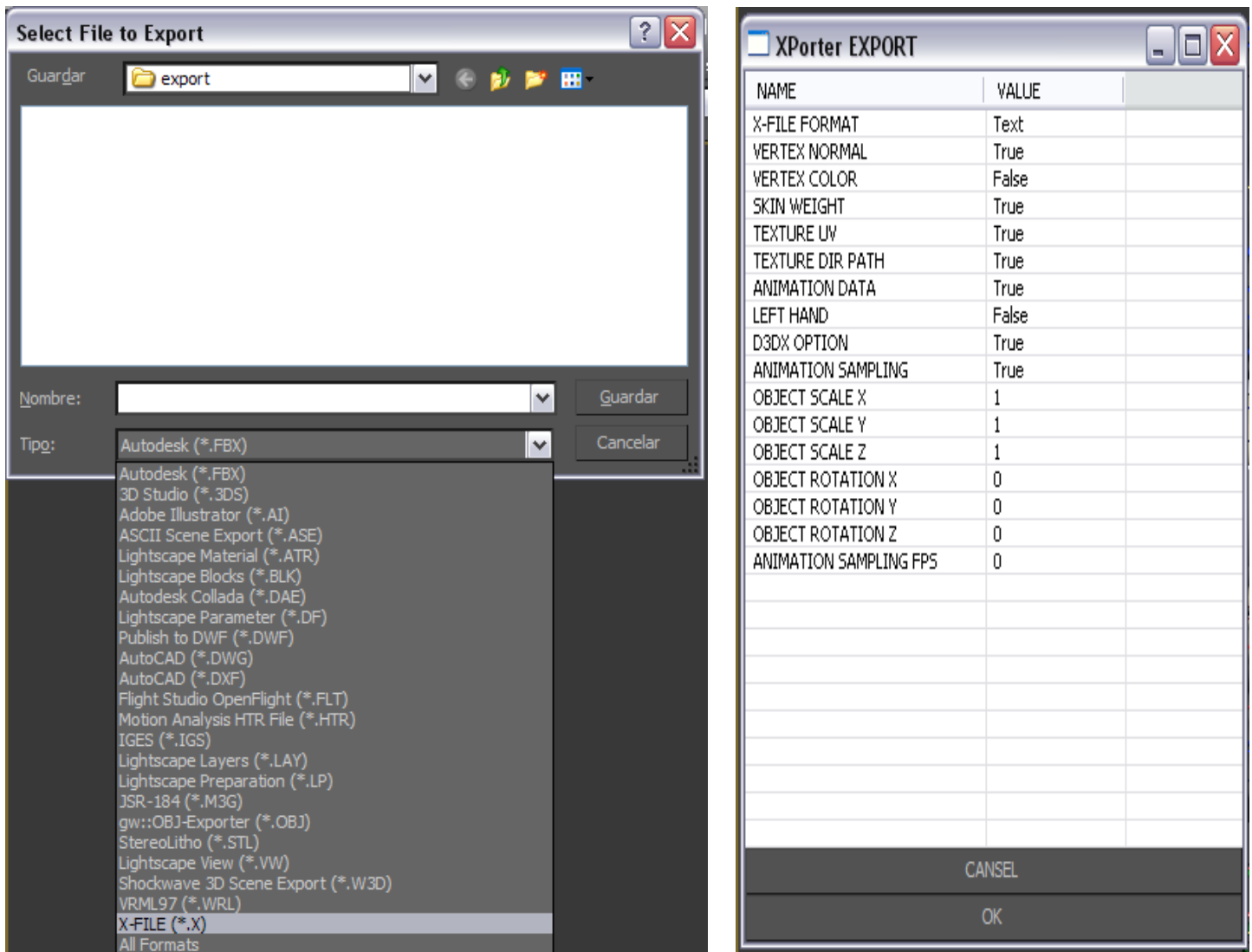


Figura 3.25 Exportación del los datos no biomédicos para desarrollo de lógica.

3.4 Desarrollo del sistema de software biomédico.

Las herramientas de desarrollo pueden combinarse en un marco llamado ambiente o entorno de desarrollo integrado (IDE por las siglas de Interactive Development Environment). Esto ofrece un conjunto común de facilidades, que usan las herramientas para comunicarse y operar con mayor destreza en una forma integrada. [19] En la *figura 3.26* se muestra el proceso de codificación que se utilizó para el desarrollo del sistema de software biomédico, en el *Anexo V se encuentran la descripción de los entornos de desarrollo*.

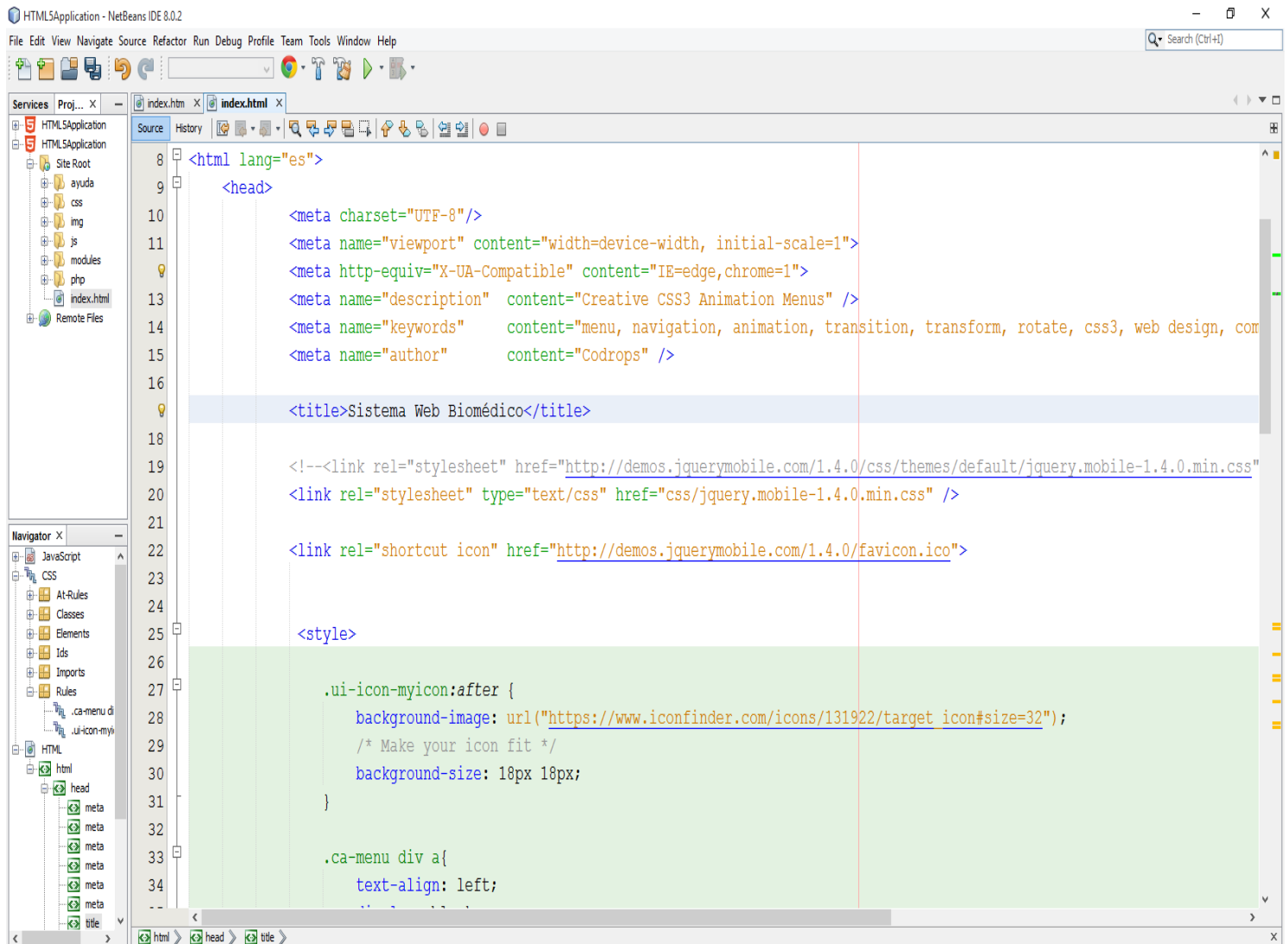


Figura 3.26 Entorno de desarrollo del sistema de software biomédico.

3.4.1 Desarrollo de Front-End del sistema de software biomédico.

El desarrollo del Front-End del sistema de software biomédico se gestiona a partir de la codificación o construcción con varios lenguajes de programación, a partir de las interfaces de usuario que se gestionaron en el capítulo IV y que se encuentra la descripción completa de estas en el Anexo IV, se empieza a desarrollar la primer pantalla “home del sistema de software biomédico” con respecto al diseño arquitectónico del sistema de software biomédico que se encuentra en el capítulo IV.

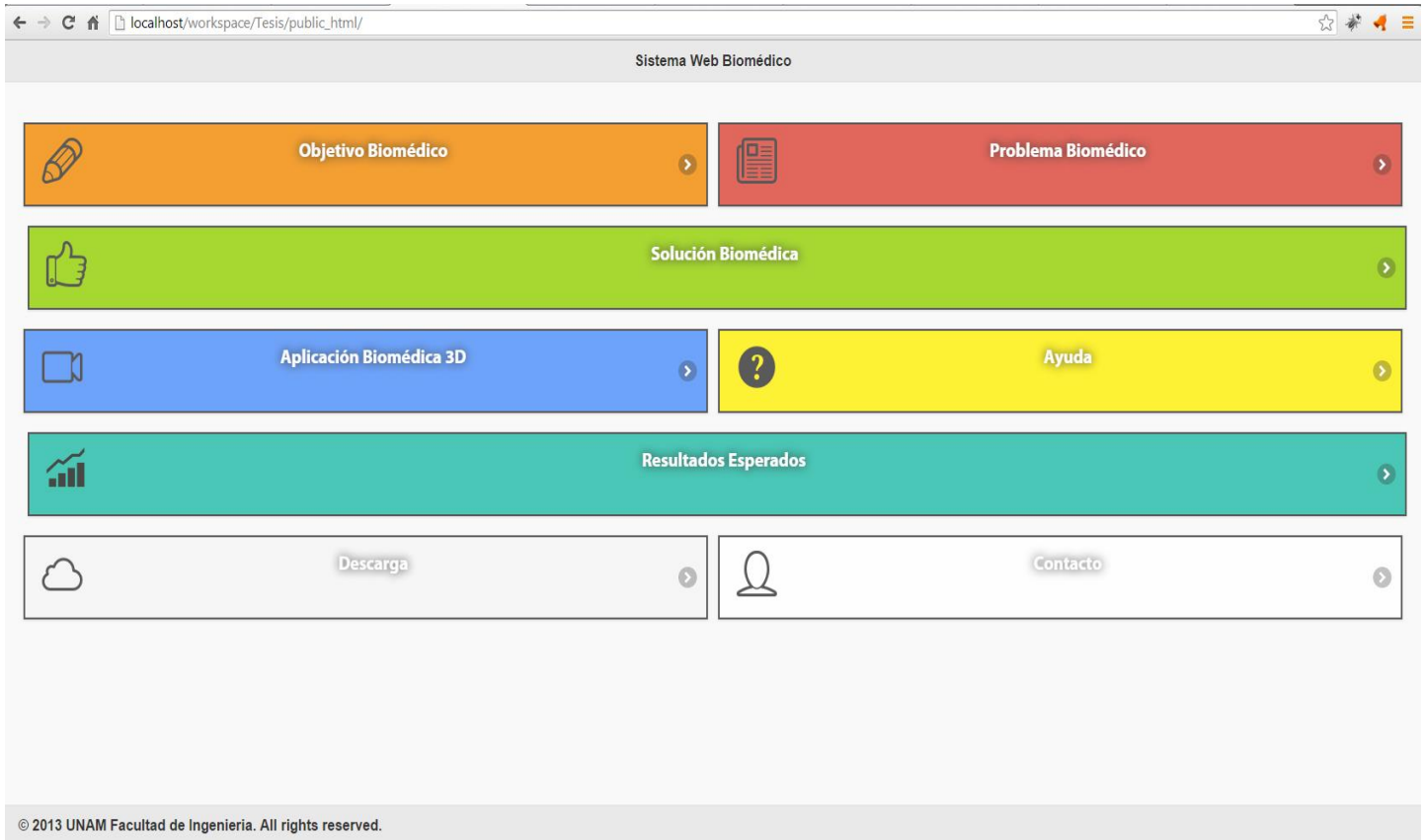


Figura 3.27 Home, sistema de software biomédico.

En el desarrollo del Front-End del sistema web biomédico se genera a partir de la codificación o programación de la interfaz gráfica de usuario, para empezar a construcción se requiere desarrollar las pantallas, los botones, el fondos e iconografía, en esta fase se utilizó hojas de estilo CSS3 en donde se codifica los estilos de la interfaz gráfica de usuario de esta pantalla, con respecto al diseño iconográfico que se desarrolló a partir de diseño 2-D utilizando herramientas de diseño como Photoshop CS5, ya que el programa tiene la capacidad de crear imágenes, filtros, transparencia, vectores gráficos de imágenes y tener múltiples imágenes gracias al sistema de capas que tiene integrado, con estas aplicaciones del programa se desarrolla todo el diseño gráfico 2D con formato de imágenes JPEG y PNG. El desarrollo del Front-End del sistema se genera la capa grafica de usuario donde esté tendrá la primera interacción con el sistema, es desarrollada buscando una experiencia de usuario fácil y de rápida curva de aprendizaje, cada módulo del sistema es representado por un botón con un estilo ligero y muy llamativo representado por un color, texto, iconografía e indicador de interacción, para esta se implementaron funciones lógicas desarrolladas con JavaScript para su re-direccionamiento e uso del módulo seleccionado por el usuario, principalmente el módulo con más desarrollo de interacción y al cual se está enfocado el objetivo es Aplicación Biomédica 3-D.

3.5 Desarrollo del sistema de software biomédico 3-D.

En el desarrollo del sistema de software biomédico 3-D se utilizó métodos para gestionar proyectos de aplicaciones de software con motores gráficos, requieren un perfil de conocimiento de diferentes áreas profesionales (programación, diseño gráfico, edición de sonidos, etc.). Cada área, trabaja con diferentes tipos de datos en diferentes tipos de formatos, para el desarrollo de la aplicación se implementaron ciertos conocimientos de estas áreas y poder lograr el objetivo. Las herramientas que nos permitan la construcción automática del proyecto, la integración de sus diferentes componentes y la coordinación son de gran ayuda en un entorno tan heterogéneo. Desde el punto de vista de la gestión de la aplicación, una tarea esencial es la automatización del proceso de compilación y de construcción de los programas. Una de las tareas que más frecuentemente se realizan mientras se desarrolla y depura un programa es la de compilación y construcción. Cuanto más grande y complejo sea un programa, mayor es el tiempo que se pierde en esta fase. Por tanto, un proceso automático de construcción de software ahorrará muchas pérdidas de tiempo en el futuro.

3.5.1 Obtención de los datos importados al ambiente de desarrollo

El ambiente de desarrollo donde se realiza la programación y que tiene la capacidad de importar geometría 3-D, además de contar con un lenguaje de programación orientado a objetos de carácter tridimensional, que nos permita manejar los diferentes eventos que ocurren en el mundo virtual y que nos permite personalizar cada una de las interfaces requeridas, manejar en paralelo un entorno multimedia (Texto, Sonido, Imagen, Animación, Video e Interactividad), es la tecnología Quest3D que cuenta con reconocimientos por parte de siggraph como una de las mejores herramientas para el desarrollo de aplicaciones de realidad virtual y visualización científica, además de que contamos con varias fuentes de información a nuestro alcance (Tutoriales en la Web, Foros de discusión, Cursos, etc.). Otra de las razones para la elección de éste ambiente de desarrollo es la utilización del plug-in para importar geometría 3D, los plug-ins que se utilizaron son desarrollados por Quest3D (almacenado en archivo *.CGR) y DirectX (almacenado en archivo *.X), el plug-in de Quest3D se utilizó para importar geometría de pocos polígonos (obtenidos de modelado geométrico 3D) y el plug- de DirectX se utilizó para importar geometría de muchos polígonos (obtenidos de reconstrucción geométrica por tomografía axial computarizada TAC). En la *figura 3.28* se muestra el motor gráfico para programar el sistema de software biomédico 3-D.

The screenshot displays the Quest3D development environment. On the left, a 'Project Tree' shows a hierarchy of objects and logic channels. The main workspace shows a scene graph starting with 'Start3DScene', leading to 'Project Start', which branches into 'Orbit Camera', 'DX8 Light', and a 'Lua' channel. Below the Lua channel, a table lists data for X, Y, Z, H, P, B, and Timer Value. On the right, the 'Lua Script Properties (Lua)' window shows a script that calculates trigonometric values and updates object positions and rotations. At the bottom right, a 'Last error message' window shows 'No error to report'.

X	Y	Z	H	P	B	Timer Value
7.93	-10.98	-2.49	0.57	0.54	-0.47	323.18

```

8 local yRot = channel.GetChild(4)
9 local zRot = channel.GetChild(5)
10
11 local timerValueChannel = channe
12 local boxObject = channel.GetChi
13
14 local value = 0
15 local cosValue = 0
16 local sinValue = 0
17 local timerValue = timerValueCha
18
19 --SIN(A*B*0.001)*A*0.1 + SIN(SIN(B*
20 --COS(A*B*0.001)*SIN(A*B*0.001)*A*0
21 --COS(A*B*0.001)*A*0.01
22 --SIN(COS(A*B*0.001))*A*0.01
23 --COS(A*B*0.001)*SIN(A*B*0.001)*A*0
24
25 while i<100 do
26   cosValue = math.cos(i*timerVa
27   sinValue = math.sin(i*timerVa
28   value = math.cos(i*timerValue
29
30   xPos:SetValue(cosValue*i*0.1 +
31   yPos:SetValue(sinValue*i*0.1 +
32   zPos:SetValue(cosValue*sinValue
33   xRot:SetValue(cosValue*i*0.01
34   yRot:SetValue(math.sin(cosVal
35   zRot:SetValue(cosValue*sinVal
36   boxObject:CallChannel()
37   i = i + i
38 end
39 end
  
```

Figura 3.28 Motor gráfico Quest3D para desarrollo del sistema de software 3-D.

El desarrollo de obtención de los datos a importar, se realiza del conjunto del modelo híbrido de datos reales y datos no reales, el formato de archivo se encuentra en 3ds formato de diseño asistido por computadora, esta herramienta tecnológica nos ofrece un plug-in para realizar la conversión de formato de archivo, que se requiere para poder realizar la manipulación de estos datos por medio de programación y lógica de flujos que contiene el ambiente de desarrollo (Quest3D), la conversión se realiza y el formato de archivo obtenido es DirectX formato que el ambiente de desarrollo Quest3D puede importar los datos trabajados anteriormente con tecnología CAD. En la **figura 3.29** se muestra el proceso para realizar la importación de datos.

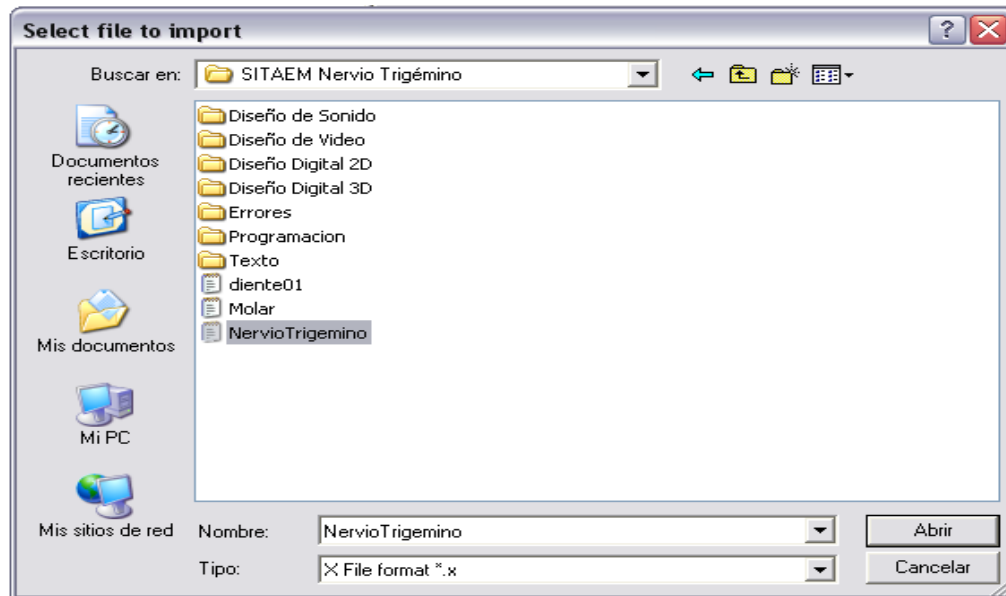


Figura 3.29 En Quest3D buscamos el archivo el cual vamos a importar con las extensión *.X y lo seleccionamos.

Inmediatamente el ambiente de desarrollo muestra la pantalla de configuración del modelo, donde ponemos ver qué tipo de importación se está realizando, como se va a descargar los modelos, la forma que se requiere el sistema de coordenadas y si requiere de una búsqueda de texturas para poder direccionarlas en el modelo. En la **figura 3.30** se muestra la configuración para importar los datos geométricos.

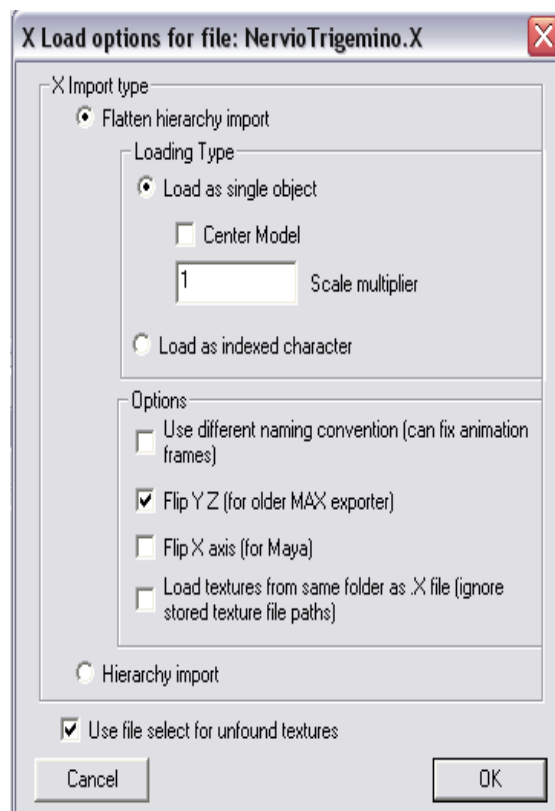


Figura 3.30 En Quest3D pantalla de configuración para importar los datos de la geometría 3-D.

3.5.2 Desarrollo de máquina finita de estados.

Este es el principal desarrollo del flujo lógico de la estructura del programa, aquí está el funcionamiento principal de la aplicación biomédica 3-D, se crea una estructura compleja mediante la elaboración de nodos de enlace para controlar los módulos de interacción de la aplicación. La máquina de estados finitos es un método para manejar estructuras complejas, como una estructura de menús donde se activan los estados de la lógica de las escenas. Por ejemplo, una estructura de menús puede tener un estado de pantalla y cambiar a otros estados de pantalla en determinadas condiciones. En la **figura 3.31** se muestra la función de la máquina finita de estados.

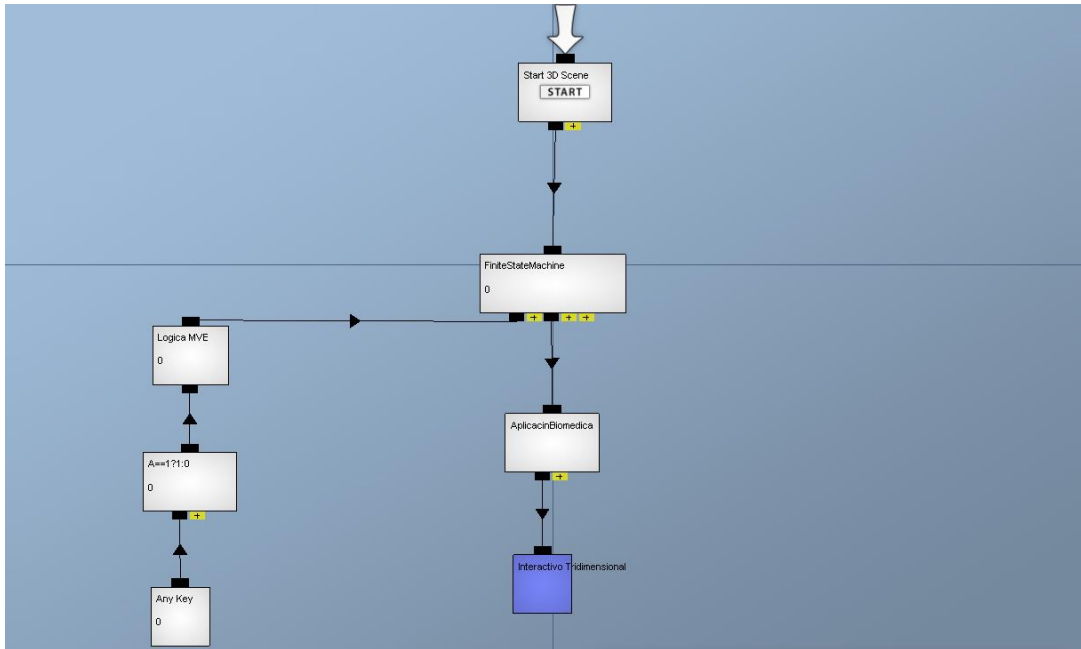


Figura 3.31 Programación de máquina finita de estados.

Otro ejemplo es la lógica para controlar un personaje. Si un personaje tiene funciones complejas de comportamiento como querer huir y buscar un área de agrupación, las condiciones para cambiar a un determinado comportamiento y manejarlo es muy compleja y requiere de una gran cantidad de mucha lógica, una máquina de estados finitos puede ser utilizada para pintar el cerebro de los personajes y controlar su comportamiento. Las funciones más complejas realmente pueden ser usadas para un montón de estructuras simples. Por ejemplo cuando el usuario hace clic en un botón de un texto determinado y lo cambia de pantalla después de 5 segundos o si el usuario pulsa algún objeto en el espacio. Estas estructuras lógicas son más fáciles de crear con las máquinas de estados finitos. Se puede utilizar el mayor número de máquinas de estados finitos en el proyecto según sea necesario. En la **figura 3.32** se muestra los estados del autómata.

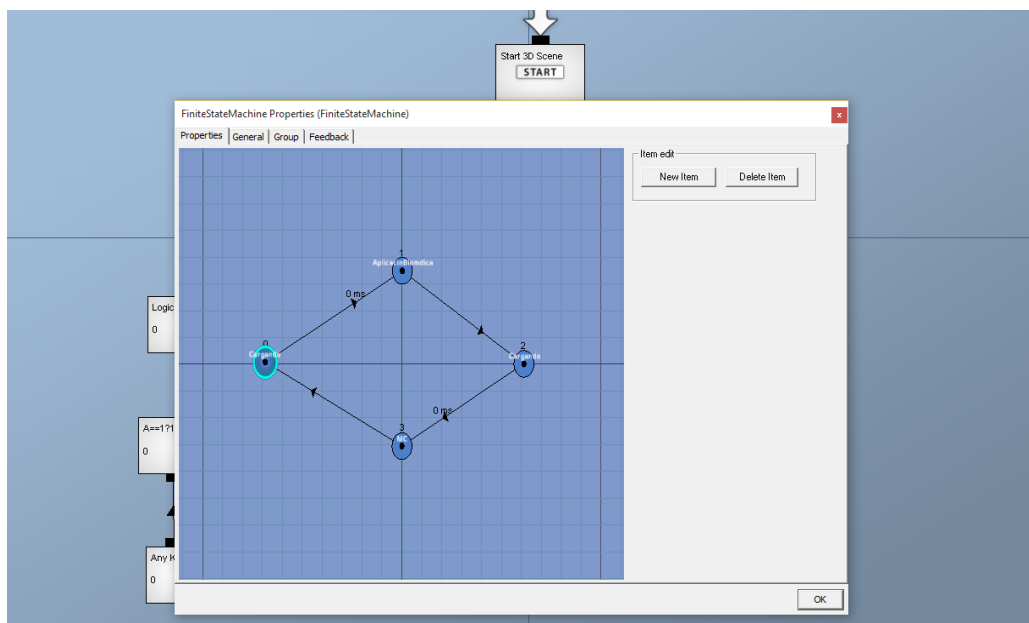


Figura 3.32 Programación de máquina finita de estados.

3.5.3 Módulo menú barra: Función pantalla completa del sistema de software biomédico 3-D.

Esta función se desarrolló con el canal ir a pantalla completa (en sus siglas en inglés, Go Full Screen), el canal es capaz de cambiar a una sesión de pantalla completa y volver al modo de ventana. En la *figura 3.33* se muestra la función de pantalla completa.

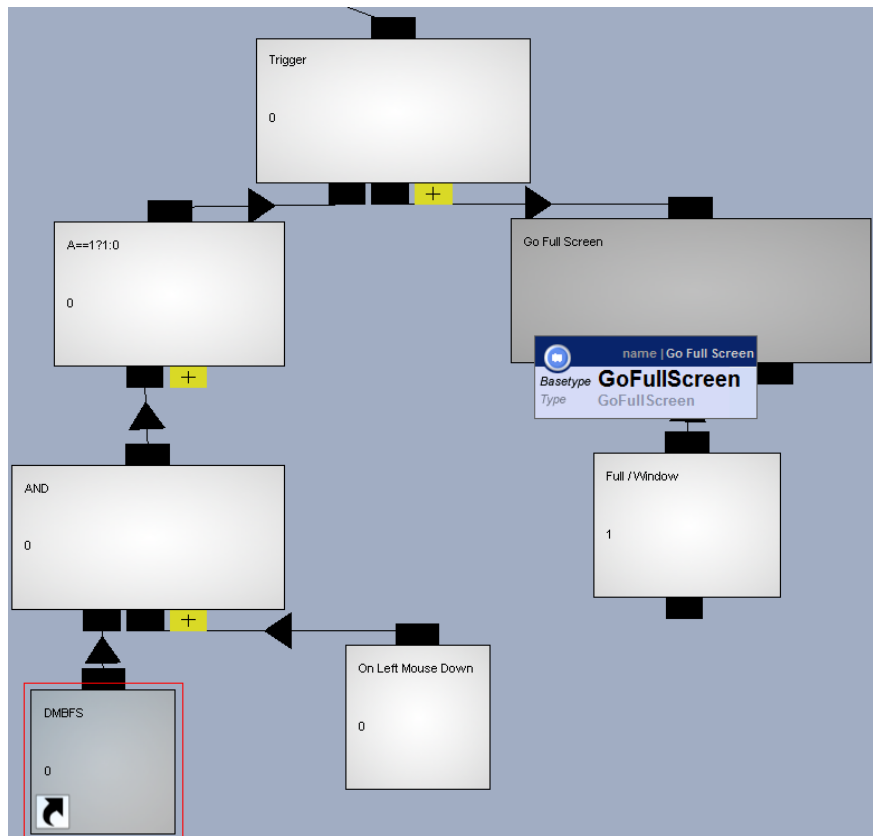


Figura 3.33 Función de ir a pantalla completa.

Con una entrada de valor al canal se puede establecer el modo deseado. Cuando desea pre-definir el tamaño de la pantalla a ventana completa se puede usar como entrada a la función las variables de altura y anchura. Se tiene que verificar la configuración de las variables a una resolución de pantalla de 800X600 o 1024X768. Con el canal activado se puede seleccionar un modo de resolución de pantalla y usar la pantalla completa. Además se puede seleccionar un adaptador de tarjeta gráfica, ya que algunos PCS tienen más de una. Para la interacción gráfica se vincula el siguiente cuadro de diálogo. En la *figura 3.34* se muestra la configuración de la pantalla completa.

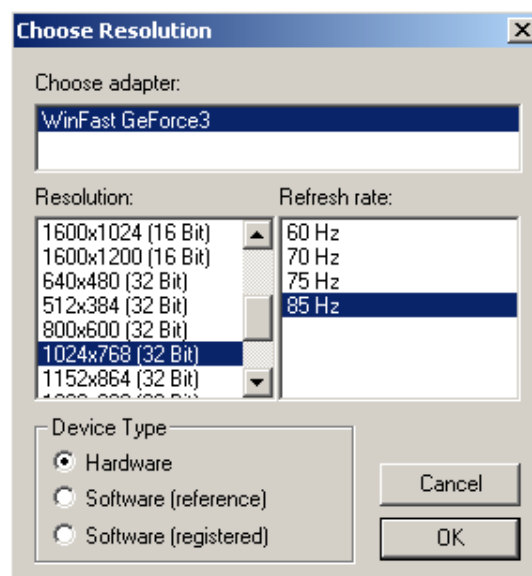


Figura 3.34 Componente gráfico para seleccionar resolución.

3.5.4 Módulo menú barra: Función reinicio de sistema de software biomédico 3-D.

Esta función se desarrolló con el canal valor ajustado (en sus siglas en inglés, Set Value), el canal es capaz de cambiar el valor de la variable a un valor asignado. Se activa cuando se dispara el llamado al canal, ocasionando que las variables ligadas al canal cambien el valor actual por el valor de la variable valor (en sus siglas en inglés, value) consiguiendo regresar al valor actual de todas las funciones del sistema de software biomédico 3-D. En la *figura 3.35* se muestra la función para reiniciar al valor inicial de las variables.

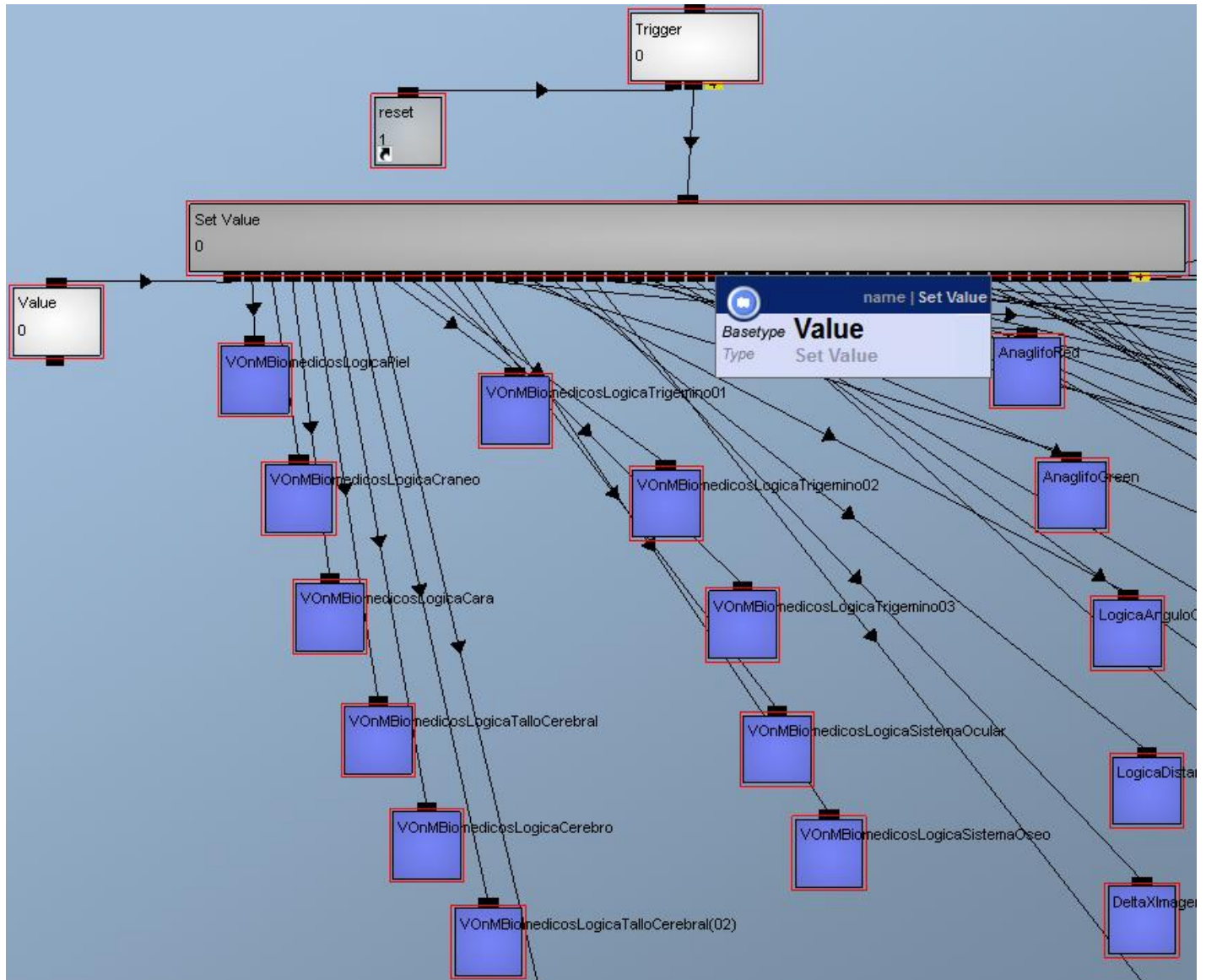


Figura 3.35 Función valor ajustado.

3.5.5 Módulo menú barra: Función de requerimientos del sistema de software biomédico 3-D.

Este canal muestra en texto 2-D de la pantalla 3-D la obtención de los fotogramas por segundo (en sus siglas en ingles FPS, Frame Per Seconds), Memoria que se está usando en el sistema para procesar el sistema de software biomédico 3-D y la resolución gráfica de la aplicación que se esté ejecutando. En la *figura 3.36* se muestra la función que obtiene los requerimientos consumidos del sistema de software.

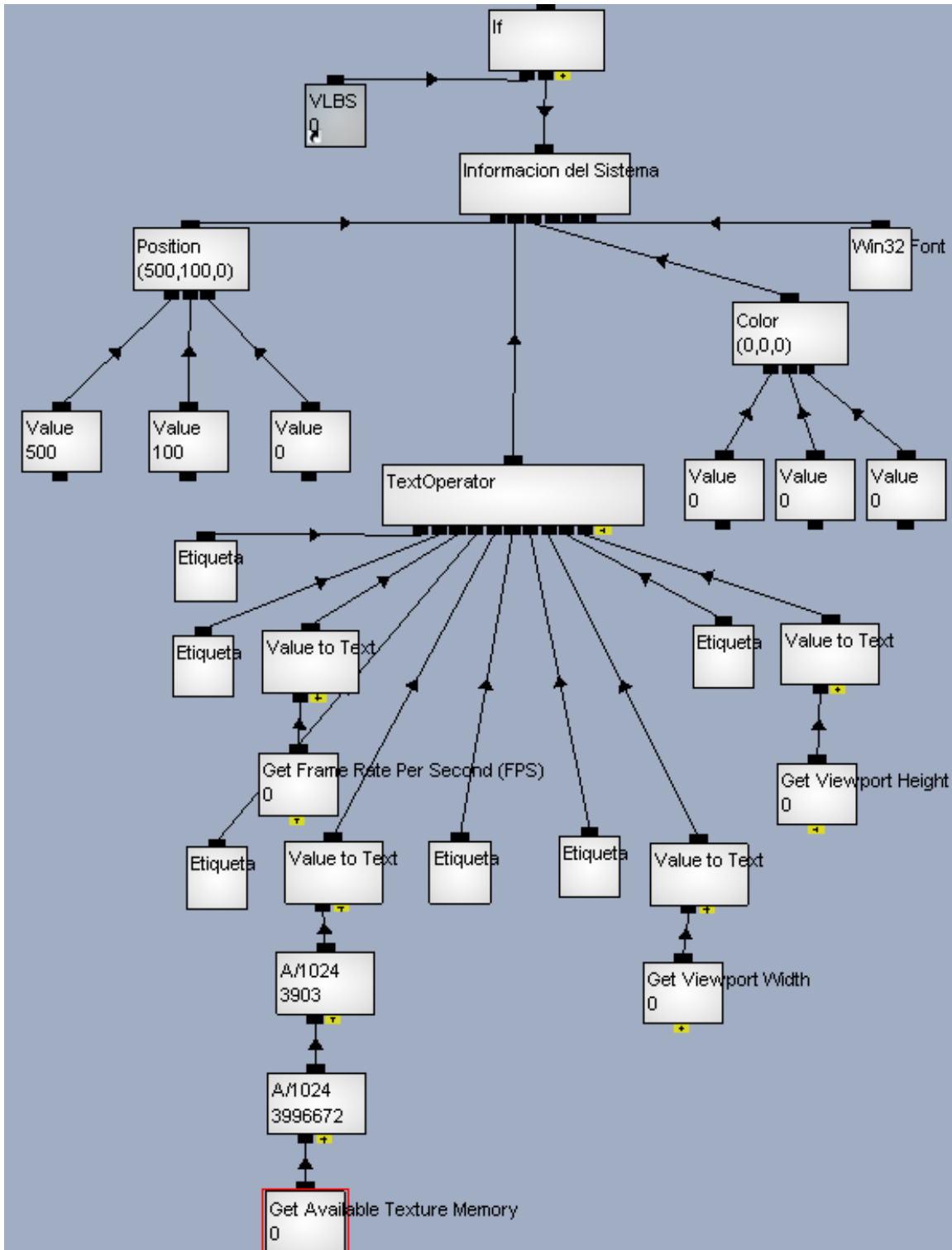


Figura 3.36 Función de configuración del sistema de software biomédico 3-D.

3.5.6 Módulo de modelos biomédicos.

El proceso de funcionamiento del módulo de modelos biomédicos se puede activar para ejecutar la visibilidad del modelo reconstruido (piel, cabeza, cara, hiodes, Cerebro, Tallo, Carótida), además de interactuar con las propiedades dinámicas de cada modelo, las propiedades como interacción con el modelo (funciones traslación, rotación, escalamiento, transparencia, etc.). En la *figura 3.37* se muestra las propiedades del módulo de modelos biomédicos.

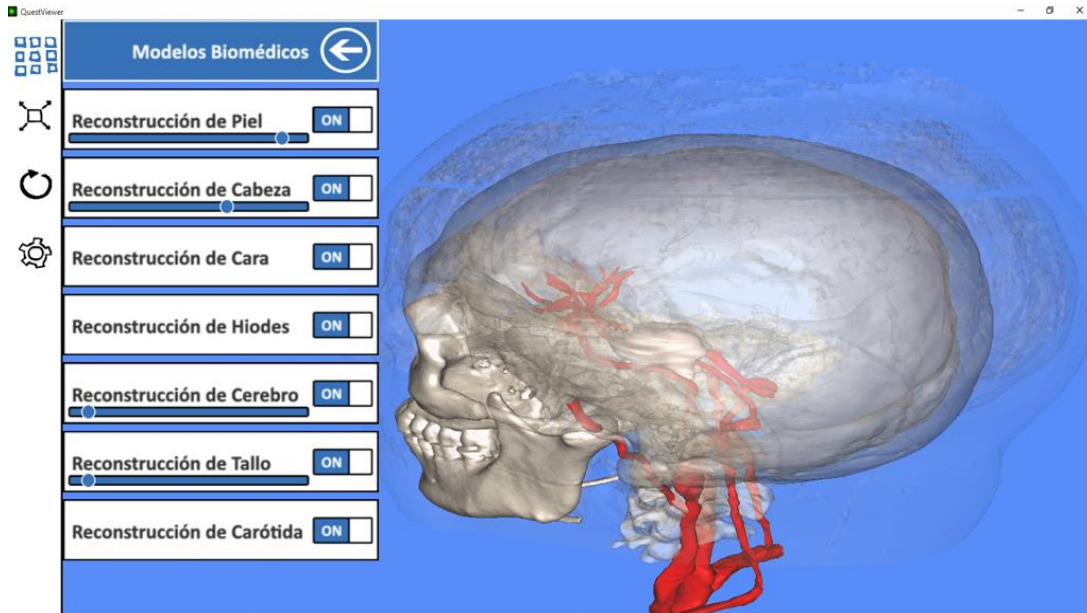


Figura 3.37 Propiedades de los modelos biomédicos.

La transparencia funciona con un componente de interfaz de usuario de tipo deslizador o conocido como control deslizador, al mover de posición el indicador del componente se visualizan en el modelo seleccionado los cambios de la propiedad transparencia. Este cambio se procesa con una función matemática que utiliza como entrada el valor desde 0% a 100% del canal alfa versus a la posición del indicador del componente que se ejecuta arrastrando el indicador del componente de derecha a izquierda para obtener la transparencia o la solidez, respectivamente, en el modelo biomédico seleccionado. En la *figura 3.38* se muestra la función lineal del canal de transparencia.

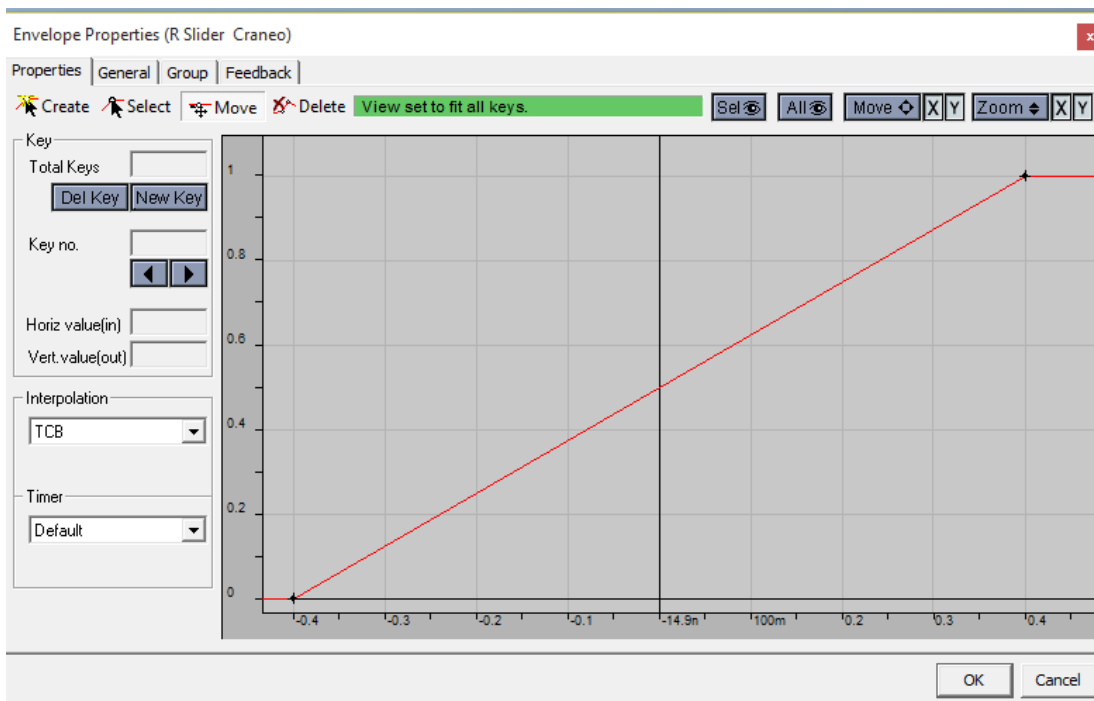


Figura 3.38 Función de canal alfa o transparencia.

3.5.7 Reconstrucción de cerebro con metodología de apilamiento de imágenes.

Las imágenes de multi-resolución, consiste en que cada imagen se compone en una pirámide jerárquica de pequeños cuadros tipo azulejos, se dividen en azulejos organizados y conocidos como quad-árboles. En la **figura 3.39** se muestra el método de apilamiento de imágenes.

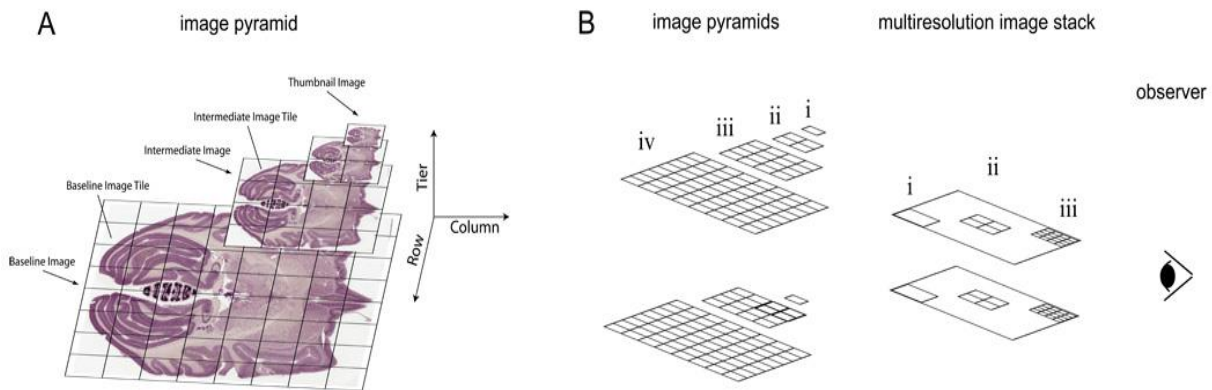


Figura 3.39 Las imágenes de multi-resolución consisten en una pirámide de imágenes compuesta por pequeñas baldosas, que permite la navegación rápida a través de imágenes muy grandes cargando solamente los azulejos de imagen que se está visualizando.

Un quad-árbol, es un modelo de una región bidimensional de forma recursiva dividiéndolo en cuadrantes. Estos cuadrantes son conocidos como hijos del principal. Por lo tanto, cada baldosa padre tiene hasta cuatro hijos. Por pila de imágenes, nos referimos a un conjunto de dos o más imágenes dispuestas de tal manera que sus planos son paralelos entre sí. [11]

Metodología

El método depende de abordar correctamente los cuadros de imagen de carga, de forma progresiva, desde el multi-resolución de pilas de imágenes. En concreto, para una determinada perspectiva adoptada por el observador (o usuario final) hay un corte óptimo (o partición) del árbol del patio de una imagen dada multi-resolución tal que seleccione texels (elementos de textura) de cada baldosa imagen cargada para que no proyecte a más de uno o dos píxeles en la pantalla del ordenador. Peticiones de baldosas de imagen se cargan con el fin de llegar adaptativamente al corte óptimo del árbol cuádruple. Este método puede ser caracterizado como visualización interactiva en árbol cuádruple imagen Pila multi-resolución con base utilizando un criterio basado proyección Texel, y sirve para diferenciarlo de otros métodos que se basan en imágenes multi-resolución no basados en árbol cuádruple o criterios alternativos, como distancia de la cámara basada en lugar de proyección basado Texel. El resultado es que texels más cerca del observador tienden a provenir de niveles de mayor resolución en la pirámide de la imagen correspondiente a una imagen en la imagen de pila multi-resolución. De una lista de cuadros de imagen cargados actualmente, tomamos aquel cuya texels proyectar en la mayor área de la pantalla y buscar a los hijos, si es que existen, de la teja de los padres, la difusión de las recuperaciones en diversos fotogramas. Usando este método, es posible visualizar e interactuar rápidamente en 3D con alta resolución de la multi-resolución de pilas de imágenes. Desde la perspectiva del observador, cuadros de imagen aparecen aproximadamente el mismo tamaño a pesar de que se accede a ellos desde diferentes niveles dentro de las imágenes multi-resolución que comprenden la pila de imágenes. [11] En la **figura 3.40** se muestra la multi resolución de imágenes.

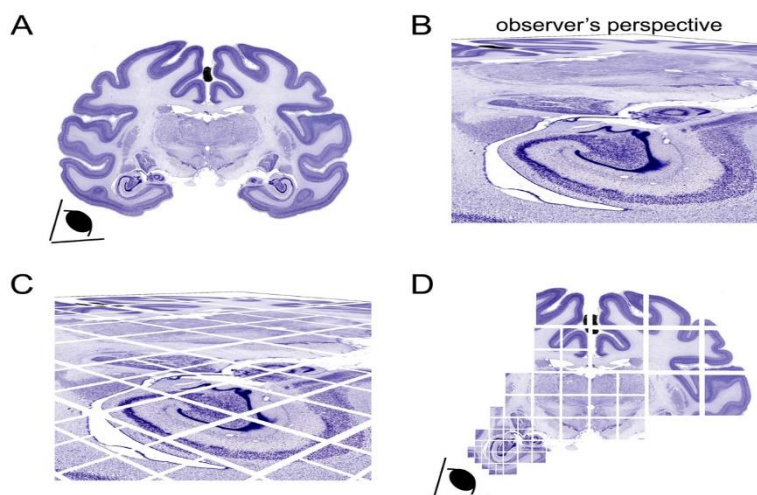


Figura 3.40 Ejemplo de visualización de una sola imagen de multi-resolución de una pila de imágenes.

Implementación

En la figura A y B se muestran la aplicación del apilamiento de imágenes de multi-resolución donde el espectador visualiza la reconstrucción tridimensional. En la **figura 3.41** se muestra la visualización de una pila de imágenes.

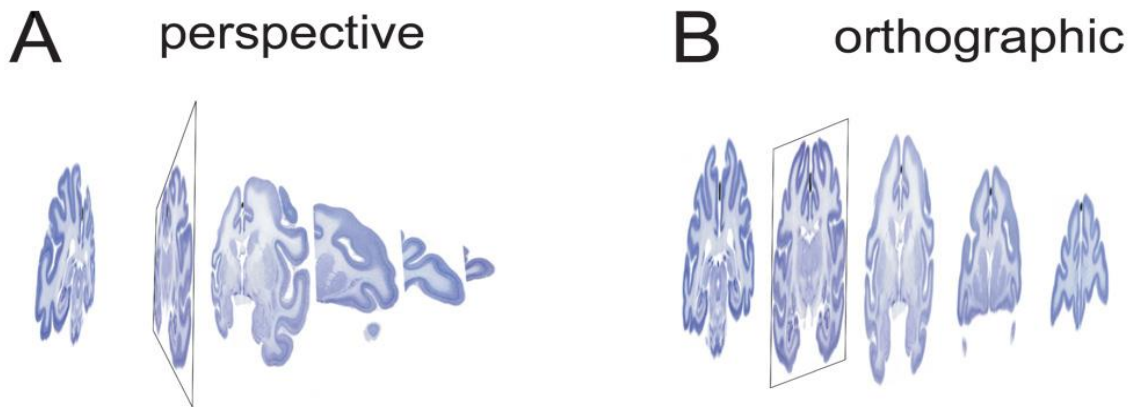


Figura 3.41 Visualización de una pila de imágenes que utilicen cualquiera de (A) la perspectiva o (B) proyecciones ortográficas. La oclusión de secciones usando la proyección en perspectiva (A) no está presente cuando se utiliza la proyección ortográfica (B).

Por cada descarga, está cargando imágenes de mosaico en el disco de 256×256 píxeles en formato JPEG y las utiliza para crear un mapa de textura. A continuación, recoge de forma recursiva los hijos de azulejos cargados cuyas texels que proyecta de cierta forma a los píxeles en la pantalla (por ejemplo, si un Texel se proyecta a más de un píxel en la pantalla). En la lista de cuadros de imagen cargados actualmente, toma el azulejo cuyo texels proyecta en la mayor área de la pantalla y obtiene sus cuatro hijos. En la **figura 3.42** se muestra la ejecución de la programación del apilamiento de imágenes.

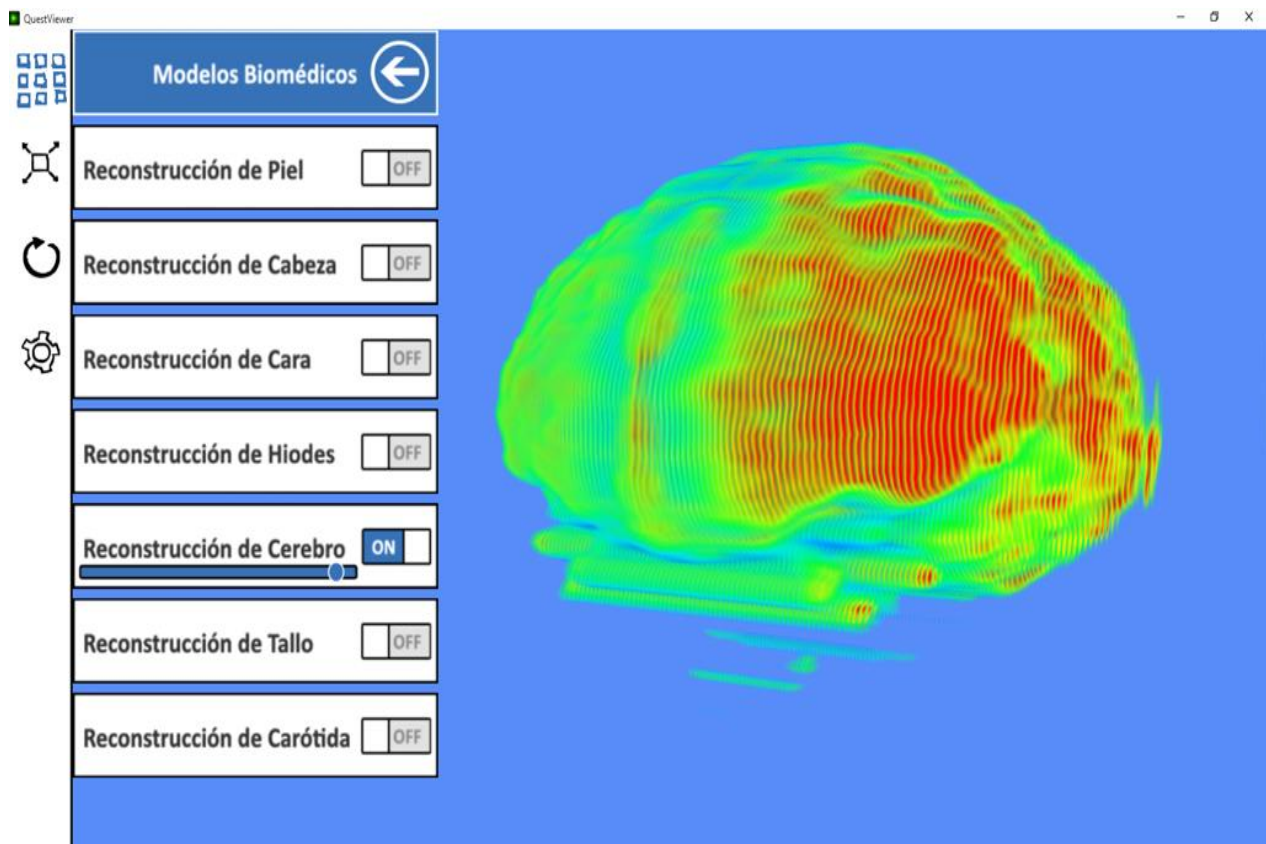


Figura 3.42 Visualización de una pila de imágenes utilizando el software de simulación para la visualización y manipulación de la reconstrucción tridimensional de datos biomédicos para potenciar sector de la salud profesional de aprendizaje con vistas en perspectiva o proyecciones ortográficas.

Al procesar el árbol del patio de baldosas, utiliza solamente un recorte del árbol. Por ejemplo, hace que sea una baldosa o de sus hijos, pero no ambos. Debido a que las imágenes son semi-transparente; un artefacto visual resulta si un azulejo se juntó con algunos de sus hijos. [11] En la **figura 3.43** se muestra la visualización de imágenes para la reconstrucción del cerebro.

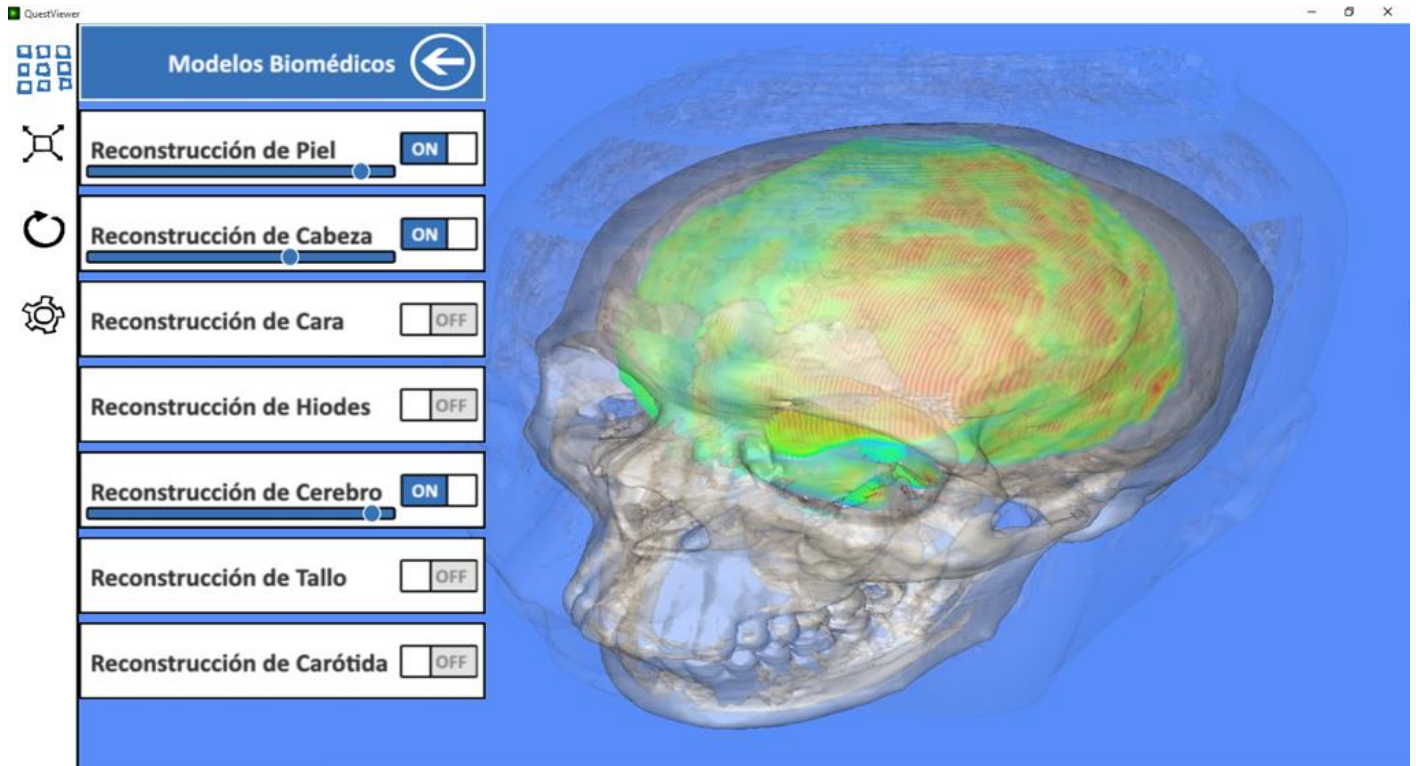


Figura 3.43 Visualización de una pila de imágenes para la reconstrucción del cerebro.

El proceso para ejecutar el método de apilamiento de imágenes se programó mediante una relación de funciones, la primera función contiene un For Loop con tres parámetros principales (valor actual = 0, iteración: incremento ó decremento y canal a llamar: objeto 3-D). En la **figura 3.44** se muestra la función for loop.

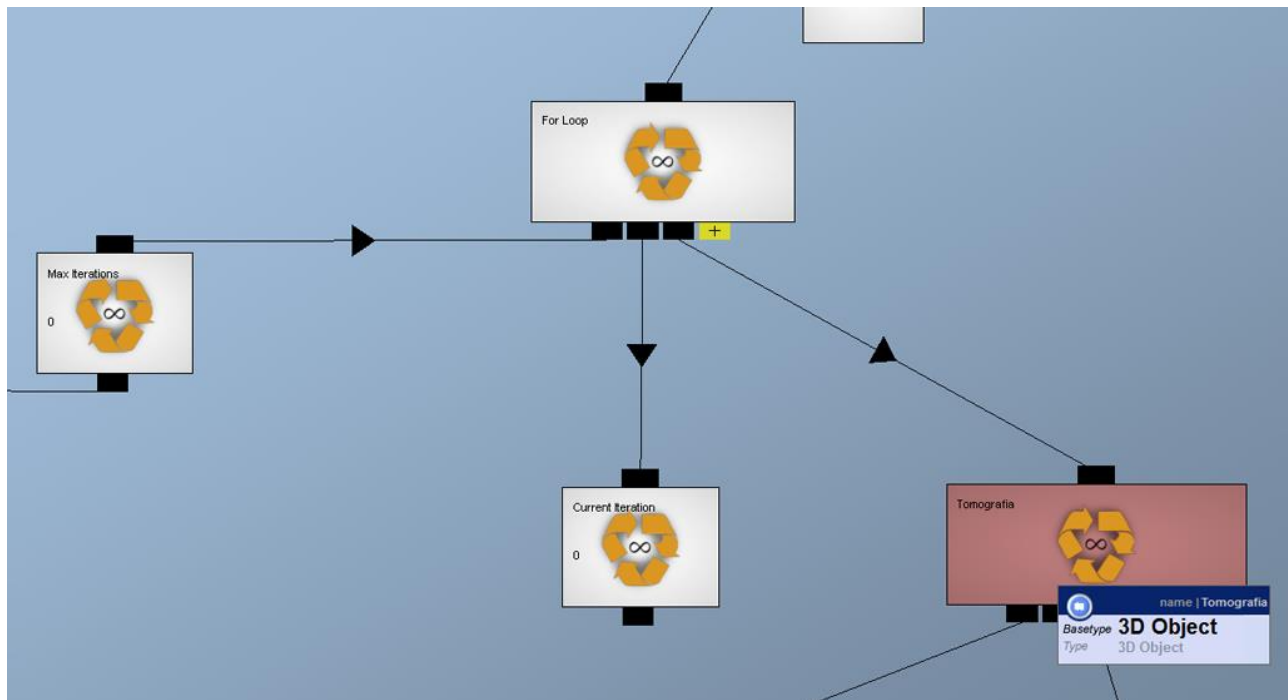


Figura 3.44 Función for loop.

La segunda función es la interacción de incremento ó decremento basado con la interfaz gráfica slider, esté componente interactúa con la función para obtener el valor de la iteración y está desarrollada para mostrar la cantidad de imágenes que se requiera. En la *figura 3.45* se muestra la función lineal para mostrar la pila de imágenes.

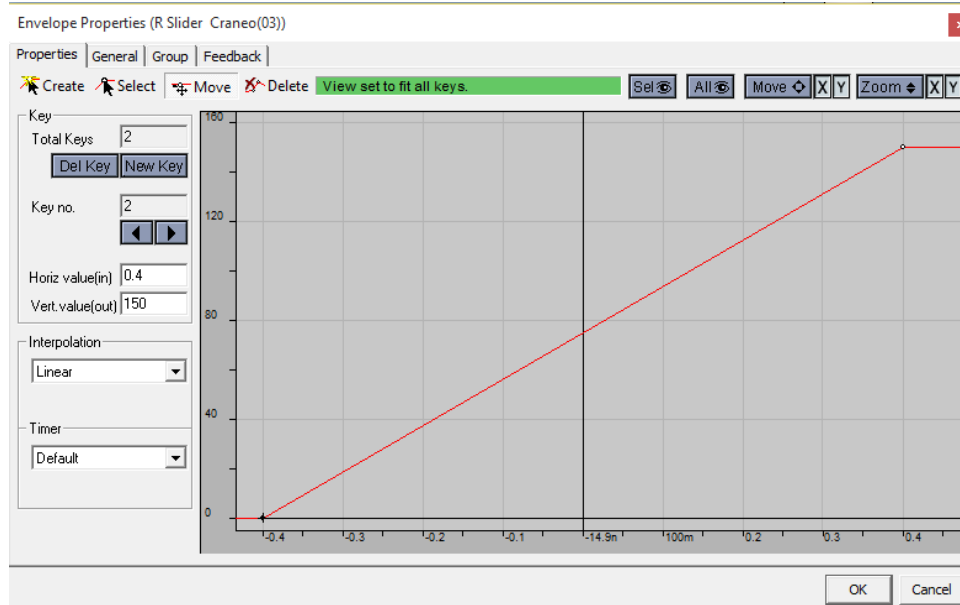


Figura 3.45 Función de iteración para mostrar la cantidad de imágenes.

La tercera función es el incremento o decremento de la posición que tendrá cada clon del objeto 3d, en este caso se usa un objeto 3d primitiva de tipo plano, que posteriormente se agrega una textura de tipo de formato “png” para generar transparencia y poder realizar la reconstrucción con el apilamiento de imágenes o texturas, esta función genera el espacio entre cada plano que contendrá cada imagen, de esta forma se va generando la metodología para visualizar la reconstrucción. En la *figura 3.46* se muestra la función de incremento o decremento para mostrar la pila de imágenes.

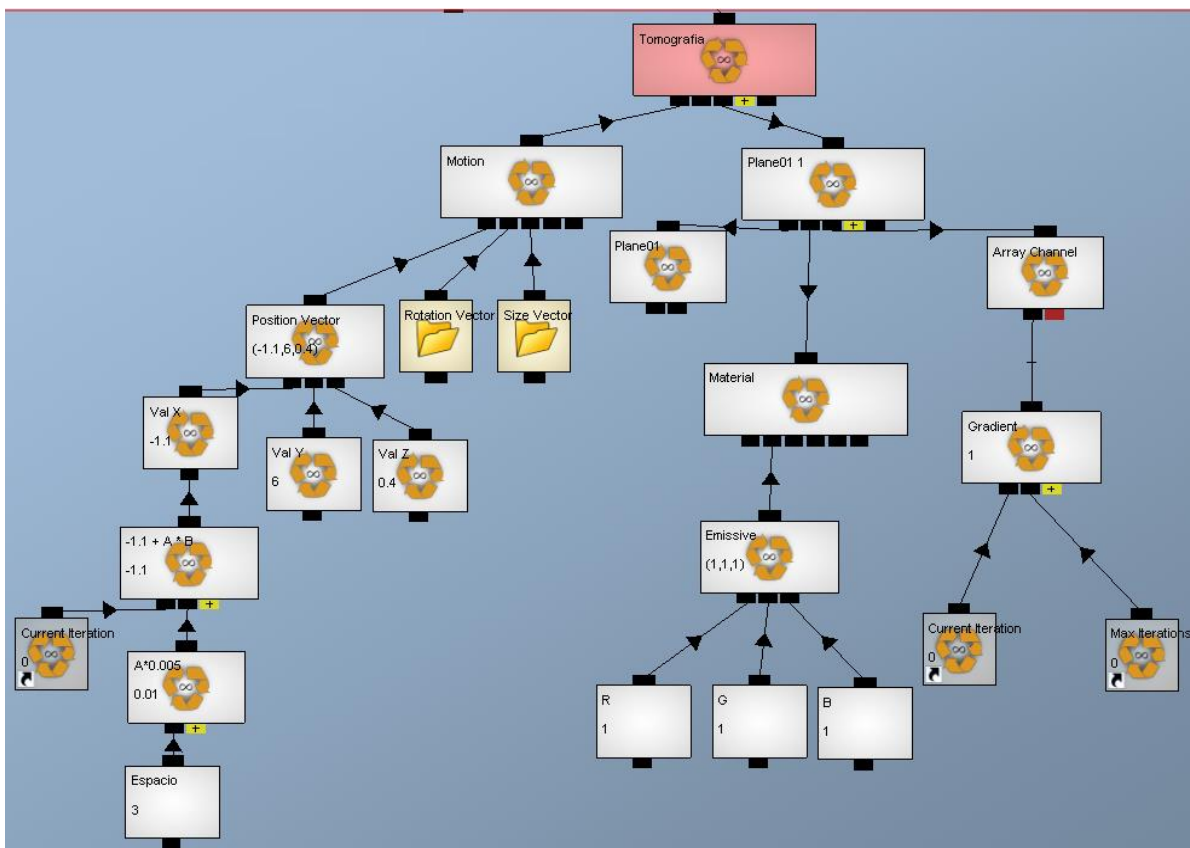


Figura 3.46 Función de incremento o decremento para la posición de planos con imágenes.

La cuarta función es la creación de un campo en la base de datos para almacenar las imágenes médicas, que se usan para desarrollar la reconstrucción tridimensional basada en apilamiento de imágenes médicas. En la *figura 3.47* se muestra la función de mapeo para el apilamiento de imágenes médicas.

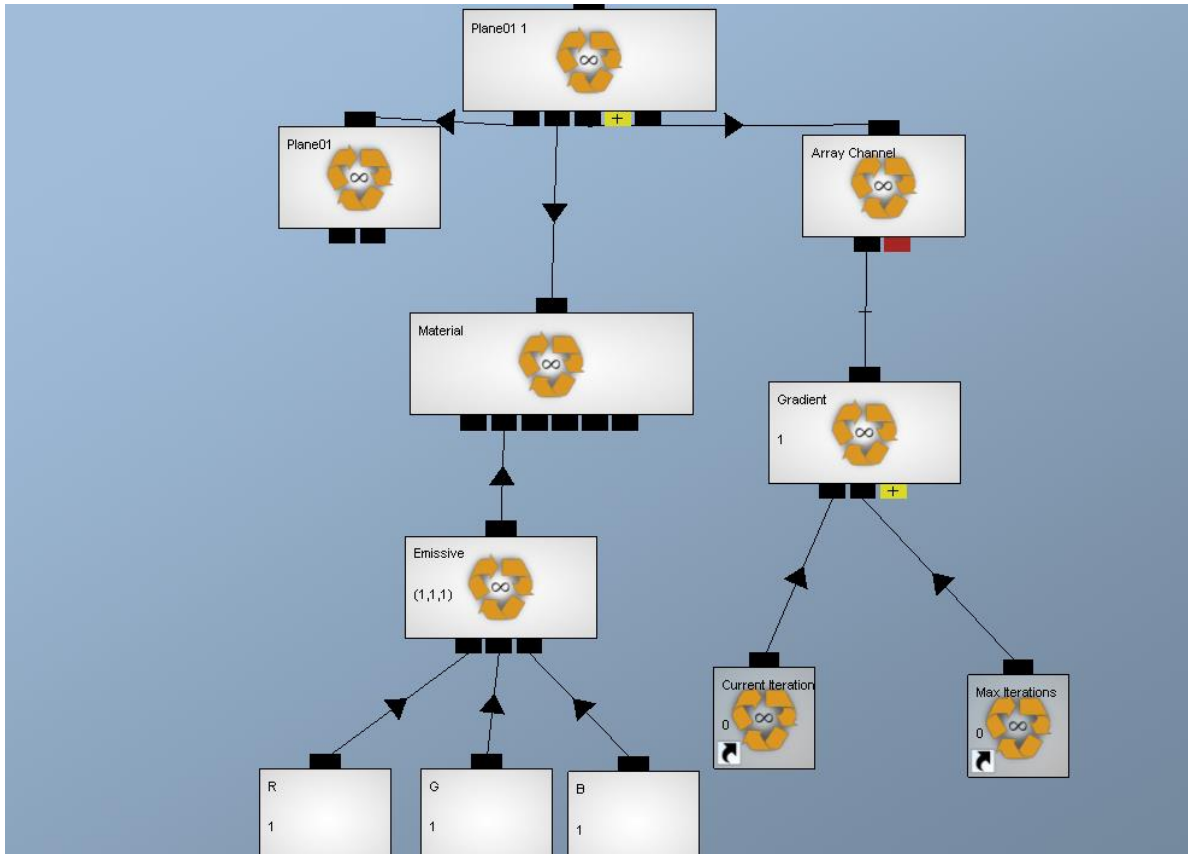


Figura 3.47 Función de imágenes médicas para mapear en los planos 3D.

Una vez creado el campo para las tomografías se van generando registros para llamar cada imagen que corresponde a cada plano, de esta forma la imagen se mapea al plano 3-D para poder visualizar la reconstrucción tridimensional. En la *figura 3.48* se muestra la base de datos donde se almacenan el apilamiento de imágenes médicas.

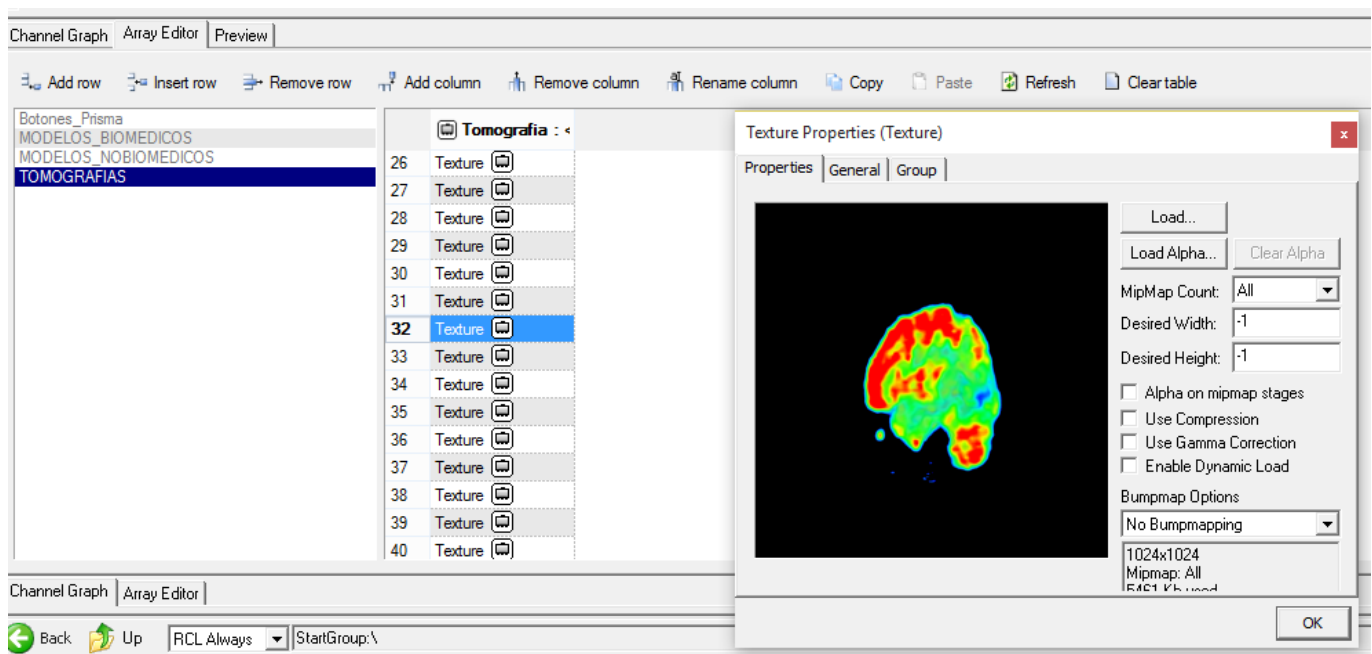


Figura 3.48 Base de datos, campo de almacenamiento de tomografías.

3.5.8 Módulo de modelos no biomédicos.

La función de estructuras corporales modeladas llamadas “no biomédicos” por ser modelos no reales, se activa para tener el control y decidir qué modelo No biomédico (rama oftálmica, rama maxilar, rama mandibular, sistema oftálmico y sistema óseo) visualizar desde la aplicación biomédica 3-D, al activar el modelo No biomédico se podrá interactuar con las propiedades dinámicas, en este caso las propiedades son la interactividad y la transparencia de la superficie. En la *figura 3.49* se muestra la función que activa los modelos no biomédicos.

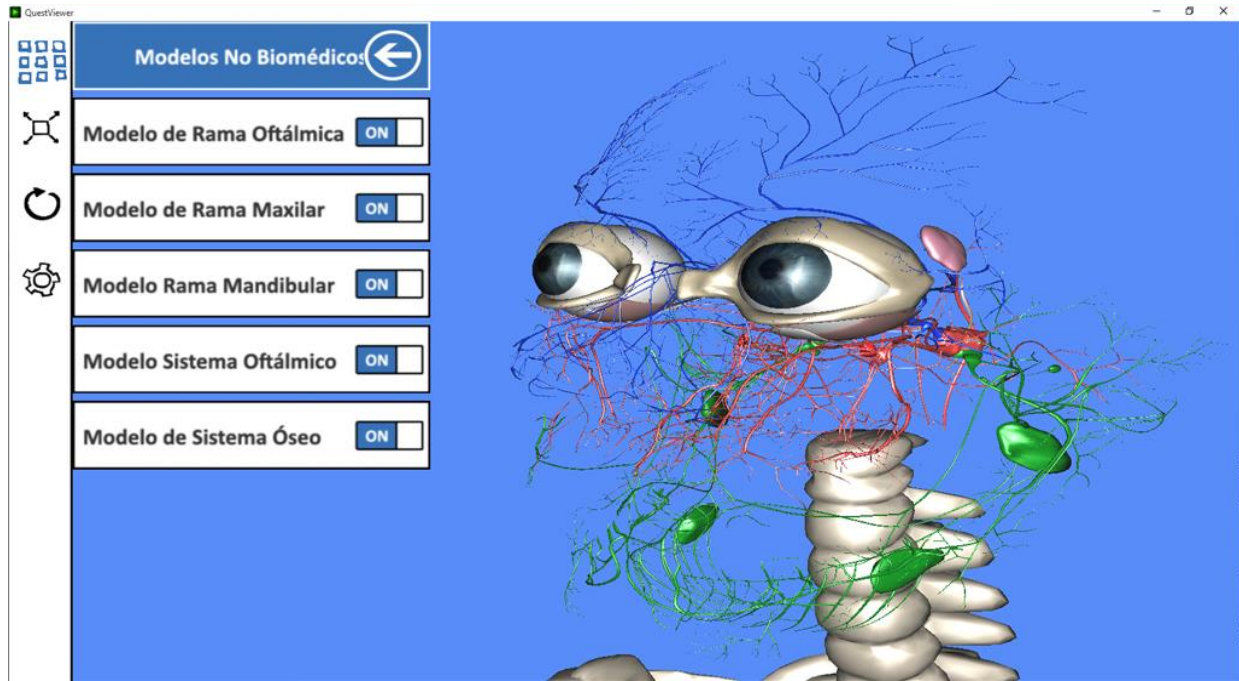


Figura 3.49 Función de modelos no biomédicos.

3.5.9 Módulo de estereoscópica pasiva

La función de estereoscopía se ejecuta para mantener la activación y el control de la técnica de inmersión 3-D usando gafas anáglifo. La estereoscopía es una técnica de imágenes tomadas o tratados con diferentes filtros de color superpuestas en una sola imagen. En la *figura 3.50* se muestra la función del render estéreo.

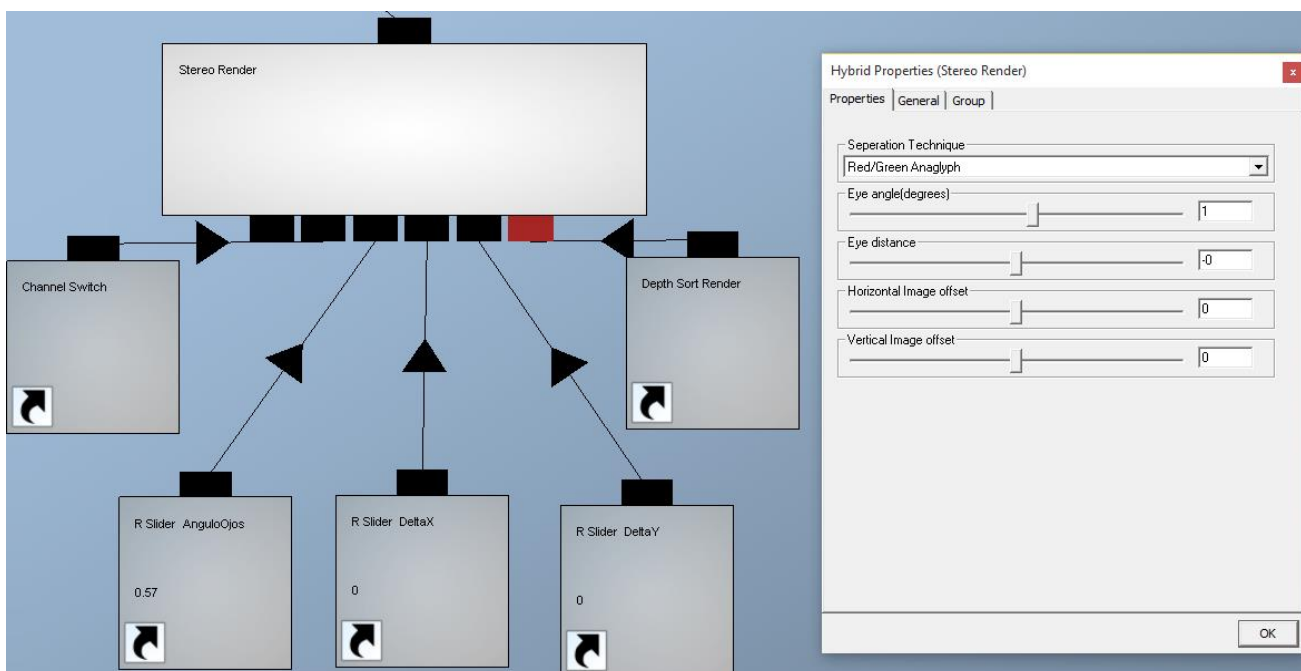


Figura 3.50 Función render estéreo.

Visto a través de las lentes denominadas gafas anáglifo y que tienen un filtro de color diferente para cada ojo. La misión de estos filtros es llegar a cada ojo sólo la imagen que se merece. Esto logra imágenes Filtro y lograr el efecto deseado y necesario para que el cerebro interprete la imagen 3-D que se merece. En la *figura 3.51* se muestra la función de estéreo anáglifo (rojo/cyan).

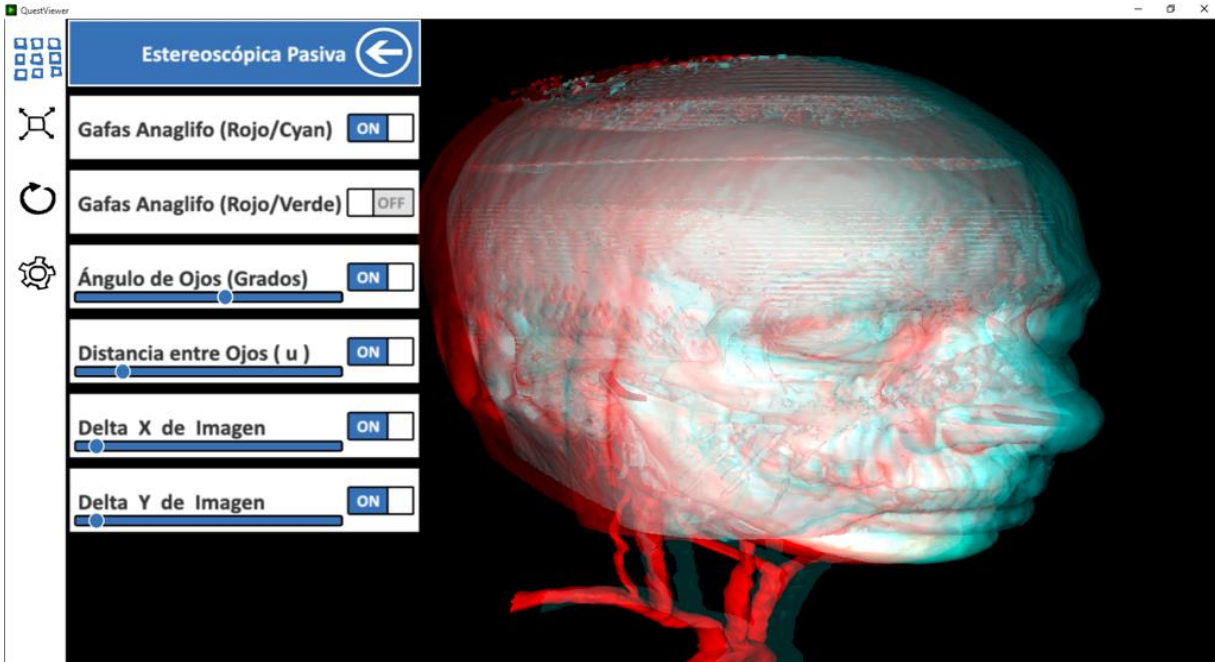


Figura 3.51 Función render estéreo anáglifo (rojo/cyan).

La distancia entre los ojos tiene funcionamiento estático y dinámico, cuando se activa el slider se utiliza el modo estático donde se obtiene el valor para la distancia entre los ojos, el modo dinámico es cuando la cámara se mueve hacia adelante o hacia atrás, la operación que se realiza es una distancia entre dos puntos para que se calcule el mismo valor constante para la distancia entre los ojos y que no se produzca fantasmas al momento de visualizar, la operación la realiza el modulo llamado distancia cámara Apuntador. En la *figura 3.52* se muestra la función de constante distancia de cámara al apuntador.

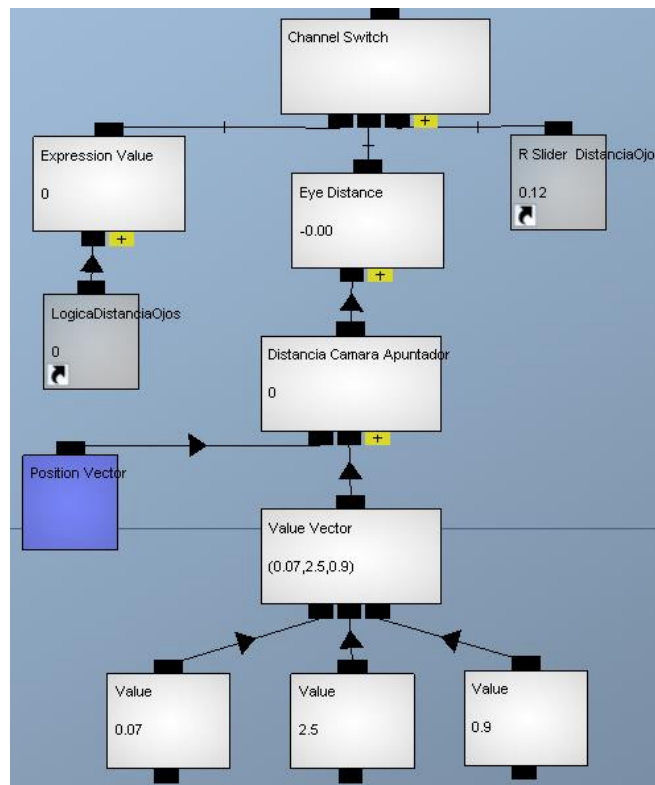


Figura 3.52 Función distancia cámara apuntador.

3.5.10 Módulo de estereoscópica activa.

Función de la estereoscopia activa opera para controlar el estéreo activo que se ejecuta a través de la unidad de procesamiento de gráficos y controlar la configuración del NVIDIA VISION GAFAS 3-D. En la *figura 3.53* se muestra la función de gafas 3-D NVIDIA.

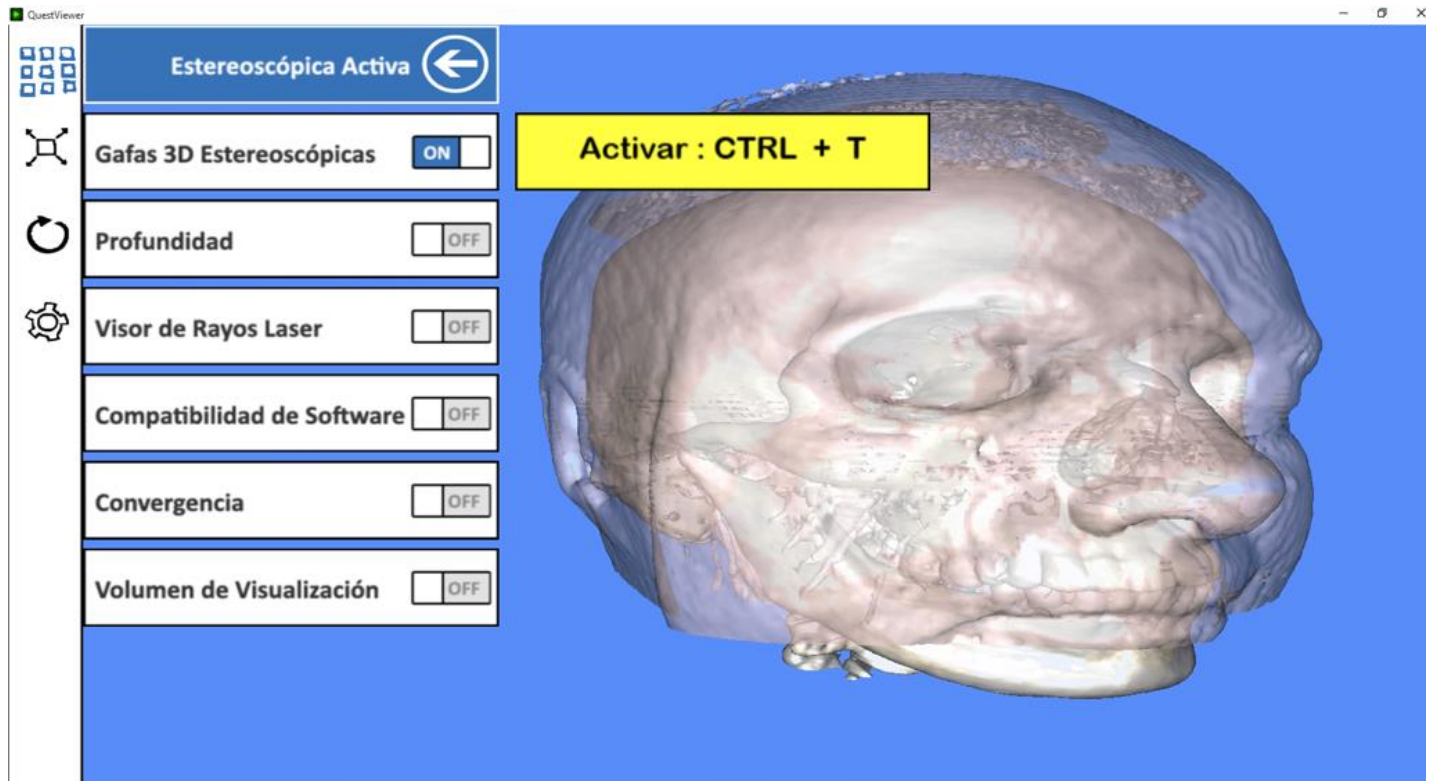


Figura 3.53 Función de gafas 3-D estereoscópicas activas.

Las NVIDIA VISION GAFAS 3-D consiste en unas gafas de cristal líquido que se oscurecen electrónicamente cada ojo, coordinado con el barrido de la imagen del monitor. Cuando la velocidad de obturación barre alrededor de 120 FPS, el oscurecimiento no se percibe, presenta así consecutivamente imagen de la izquierda y la derecha y permitir la visualización, respectivamente, obteniendo la profundidad. En la *figura 3.54* se muestra los NVIDIA Geforce 3DVISION.



Figura 3.54 NVIDIA Geforce 3DVISION.

3.5.11 Módulo de información de modelos: Función detección de mouse colisión

El módulo de información de modelos consiste en mostrar toda información que tenga una primitiva geométrica y se representa en forma de texto en la pantalla, para desarrollar las funciones para las primitivas 2-D y 3-D se usaron diferentes métodos de detección de geometría, conocido técnicamente como estructuras de detección de colisiones. El desarrollo de la colisión realiza la detección de objetos en el espacio activando detección con un 1 lógico y desactivando con un 0 lógico, los objetos que detecta esta función de colisión pueden ser componentes de interfaz gráfica de usuario, usando un método de mapeo en un objeto plano para realizar funciones de interacción del diseño de interfaz dentro del espacio vectorial, también la función de colisión funciona para detectar modelos tridimensionales, ya sea primitivas como modelos complejos, que es el caso del objetivo, la colisión también se utiliza como función de guía de pared en algún recorrido que se active en el modo tercera persona, sin la colisión las funciones mencionadas anteriormente no realizarían su tarea para detectar objetos seleccionados, por tal razón la importancia de tener un buen desarrollo en la estructura de implementación de la función para detección de colisiones. Como anteriormente se comentó, no existe solo una función de colisión, hay diferentes funciones de colisiones, a continuación se describen los casos que fueron necesarios en la aplicación biomédica 3-D:

- La función Detección de Colisión por Ratón (En sus siglas en inglés, DetectMouseCollision) se activa con un 1 lógico cuando el cursor del ratón esta sobre un objeto. En esta función se pueden tener enlace con diferentes formas de objetos (planos, esferas, objetos complejos, etc.) esta función detecta si el cursor del ratón esta colisionando con el objeto o no. Si el cursor del ratón esta colisionando con el objeto 3D la función dispara o envía un dato lógico de 1 y si no existe colisión dispara o envía un dato de 0. En la **figura 3.55** se muestra la función de detección de colisión del mouse con una superficie plana.

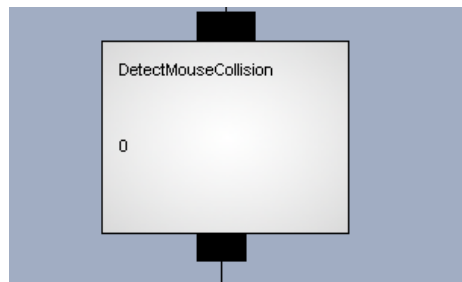


Figura 3.55 Función de detección de mouse colisión.

- La función se implementó en la aplicación 3-D para detectar la interfaz con la que el usuario activa otras funciones que contiene la aplicación, los componentes que se detectan con la colisión son sliders, botones, mostrar etiquetas, mostrar estado, etc. En la **figura 3.56** se muestra la función de detección de mouse colisión en el sistema de software biomédico 3D.

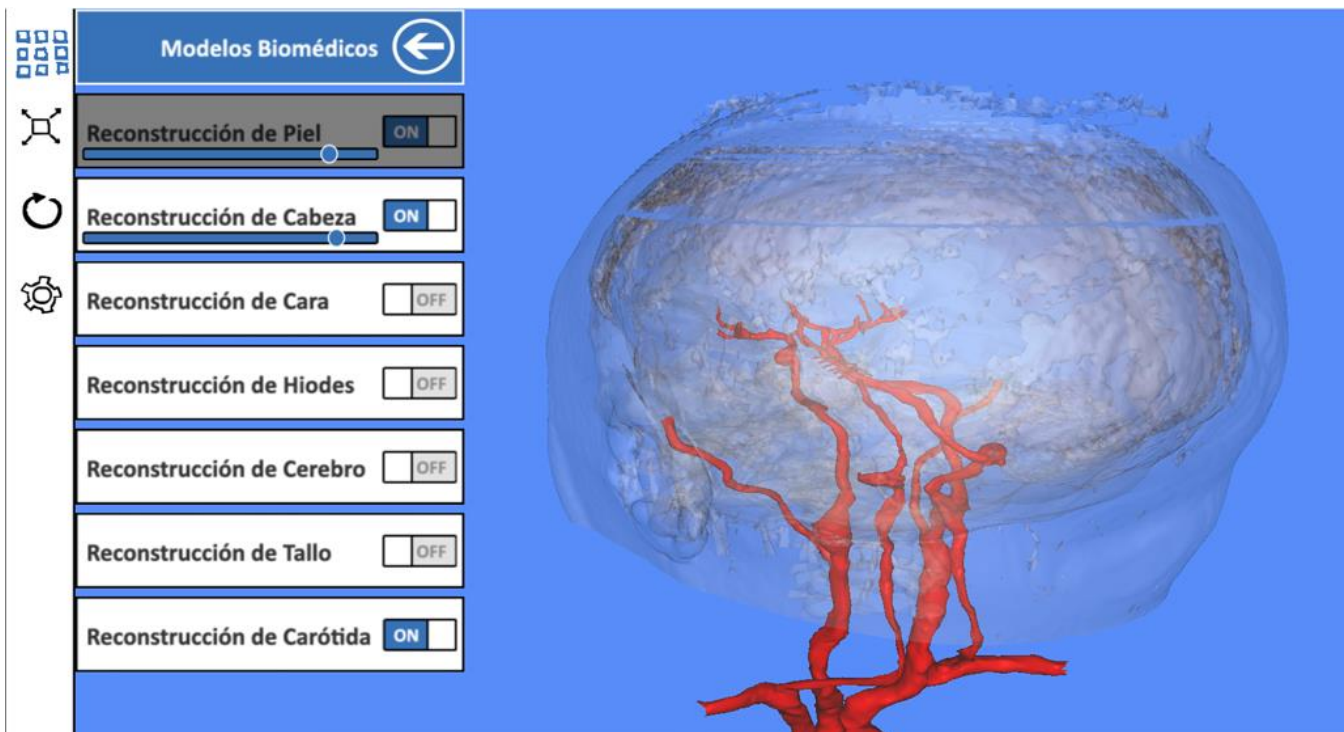


Figura 3.56 Detección de mouse colisión.

3.5.12 Módulo información de modelos: Función comprobador de colisión por rayo

La función *collisionRayCheck* activa la colisión mediante la ejecución de un rayo que intersecta la superficie contra la que esta colisionando el objeto intersectado. Los parámetros a configurar de la función son **Posición Original (relación vectorial)**: Posición del rayo, **Movimiento del vector (valor del vector de)**: Movimiento del rayo, **Salida: Punto de Intersección (valor del vector de)**: Punto de Intersección, **Salida: Dirección de Intersección (valor del vector de)**: Dirección de Intersección, **Canal de Colisión de Objetos (colisión de objetos)**: Objetos que necesitan ser comprobados por la colisión por rayo. En la *figura 3.57* se muestra la función de comprobación de colisión por rayo. En la *figura 3.57* se muestra la función de comprobación de colisión por rayo.

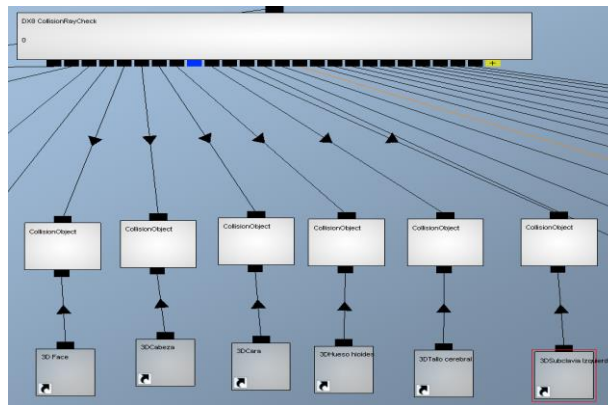


Figura 3.57 Función de comprobación de colisión por rayo.

La función detecta la colisión del objeto que tiene colisión con el rayo, este puede ser configurado con los parámetros anteriormente mencionados. Cuando la colisión es detectada, la función activa un valor entero para cada uno de los objetos relacionados con la función de colisión (iniciando el valor entero con 1) y quedando con el valor entero 0 cuando no es detectada una colisión, el valor entero sirve como id de objetos, así cuando se detecta el objeto la función puede saber que objeto se está colisionando. La función ejecuta un rayo que se determina por parámetros como la posición vectorial original y el vectorial de movimiento. Cuando la colisión es detectada se activa la intersección del punto del vector que será la posición en 3-D en el espacio donde la colisión fue detectada. La dirección de intersección será la normal vectorial del polígono sobre el cual la colisión fue detectada. Para detectar colisiones de objetos anidados, se debe tener una jerarquía de posición de identificador de colisiones de los objetos, ya que al desactivar el objeto la función buscará automáticamente el siguiente por jerarquía, de esta forma se puede calcular todos los puntos de colisión del rayo y regresar la colisión más cercana. En la *figura 3.58* se muestra la comprobación de colisión por rayo en los datos biomédicos del sistema de software biomédico 3-D.

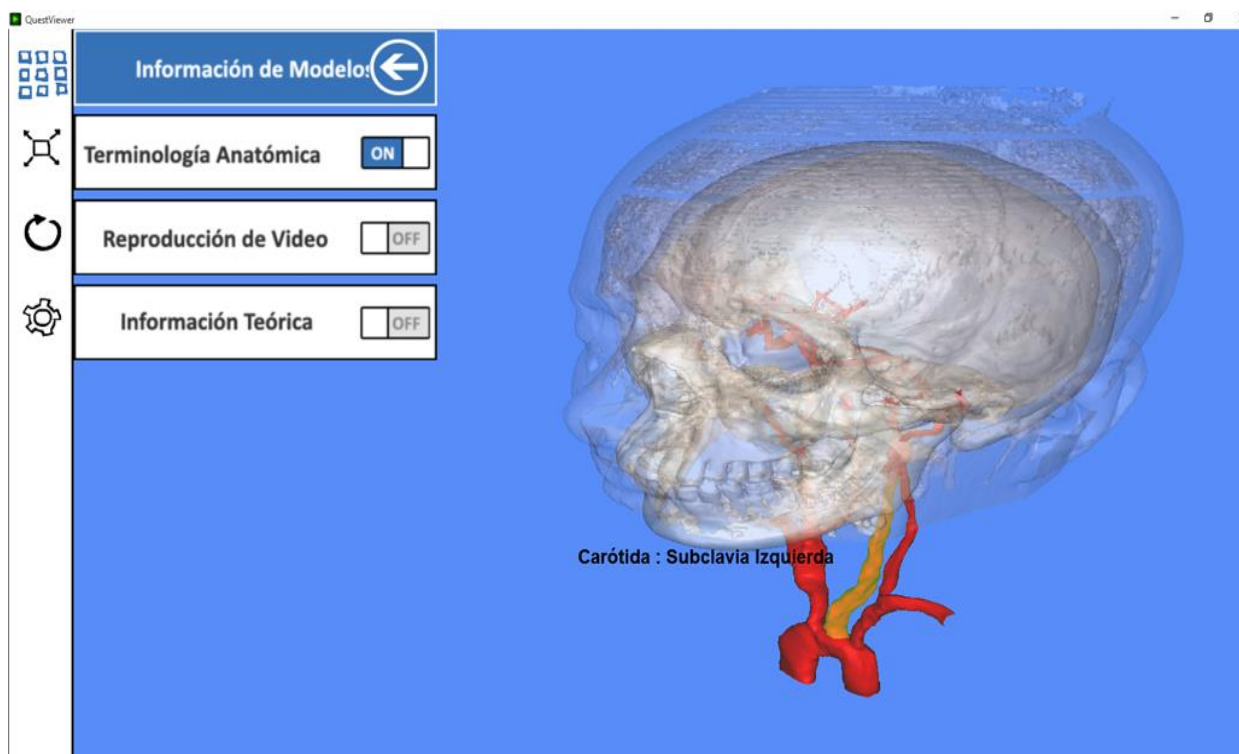


Figura 3.58 Comprobador por rayo de colisión.

3.5.13 Módulo de cortes del cuerpo humano

La función de cortes del cuerpo se ejecuta para tomar el control en los diferentes cortes existentes como:

- a) **Corte plano sagital.** En la *figura 3.59* se muestra el plano sagittal izquierdo.

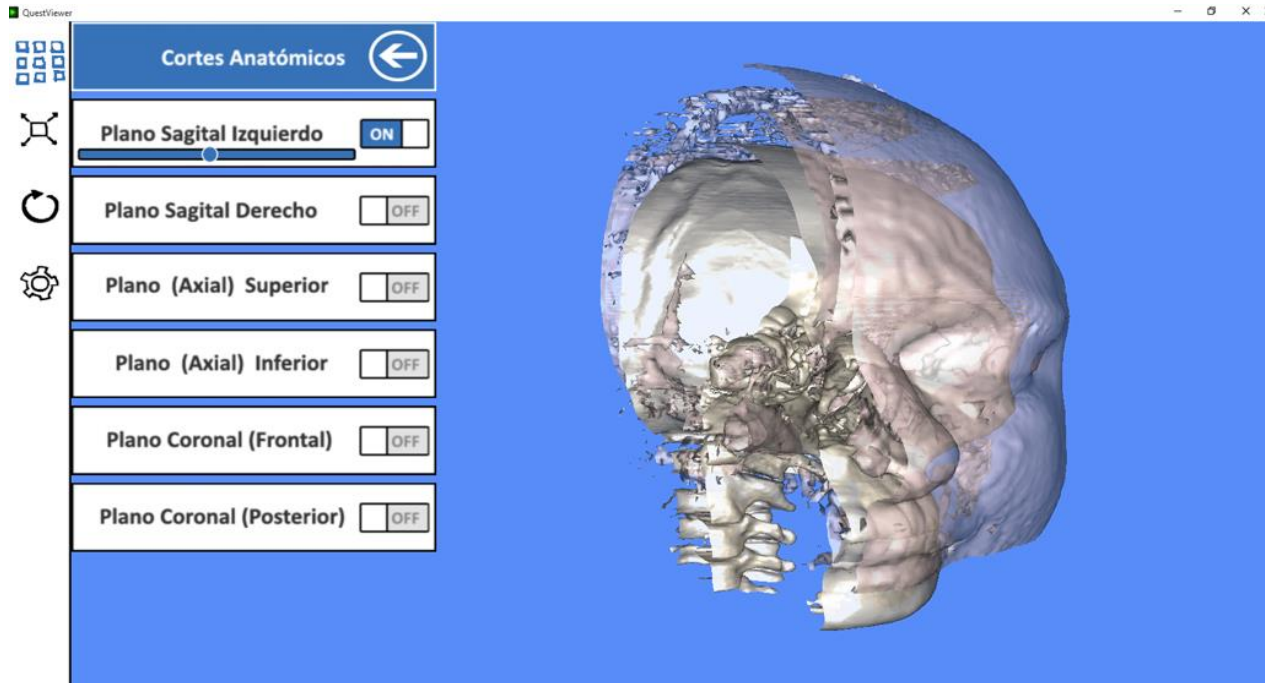


Figura 3.59 Plano sagital izquierdo.

- b) **corte plano coronal.** En la *figura 3.60* se muestra el plano coronal posterior.

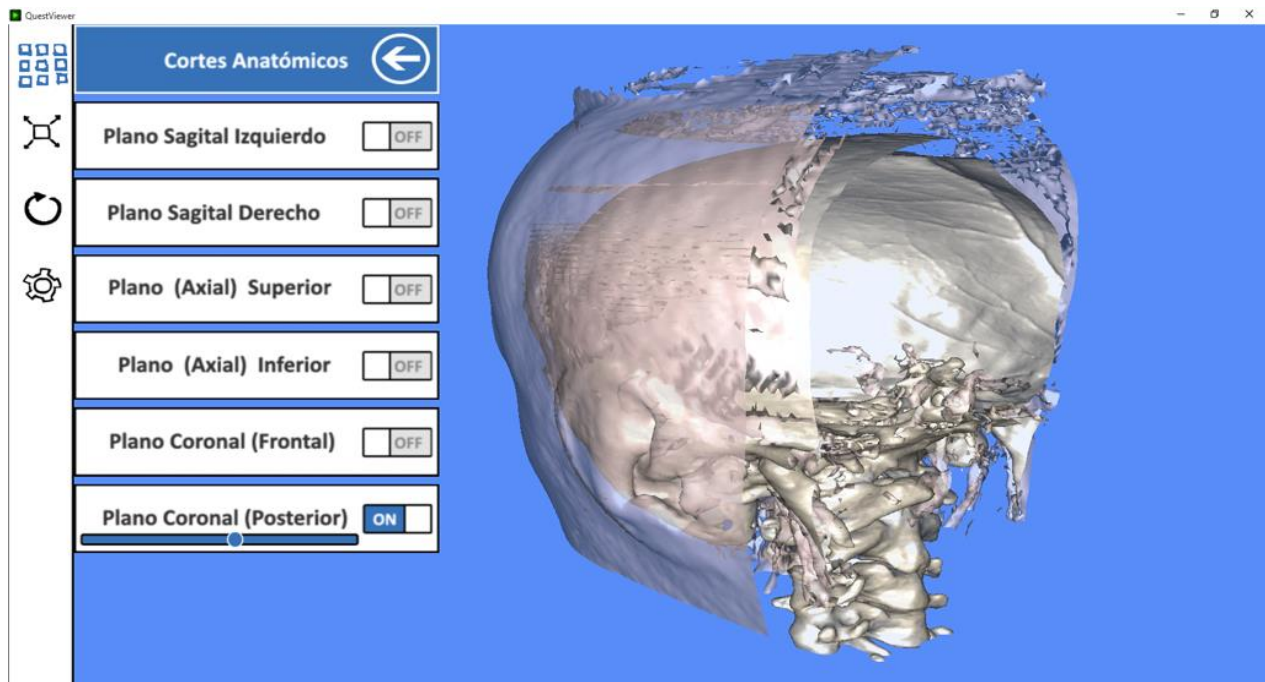


Figura 3.60 Plano coronal posterior.

c) Corte plano transversal. En la *figura 3.61* se muestra el plano coronal posterior.

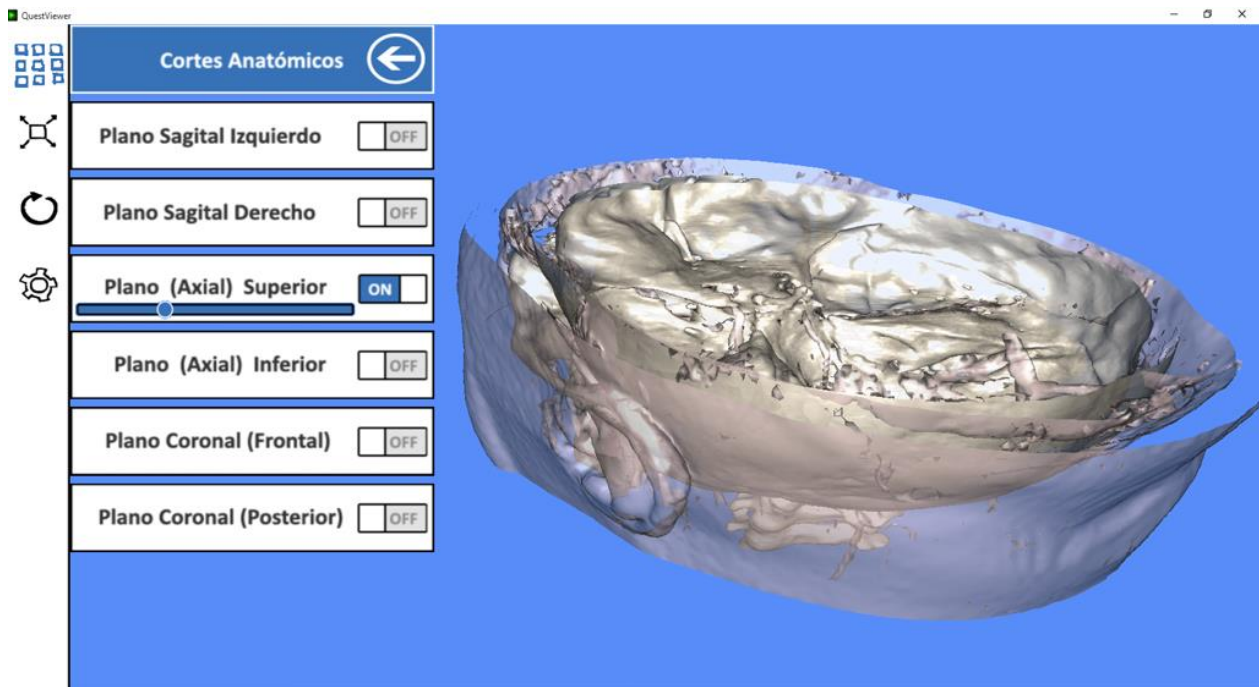


Figura 3.61 Plano coronal posterior.

La función se ejecuta en tiempo real para ambos modelos, biomédicos y no biomédicos, el método para realizar el corte se utiliza un módulo de render que genera un plano con movimiento, a este módulo se le conoce en los video juegos como plano de recorte (en sus siglas en inglés, Clipping Plane). En la *figura 3.62* se muestra la función del plano de corte.

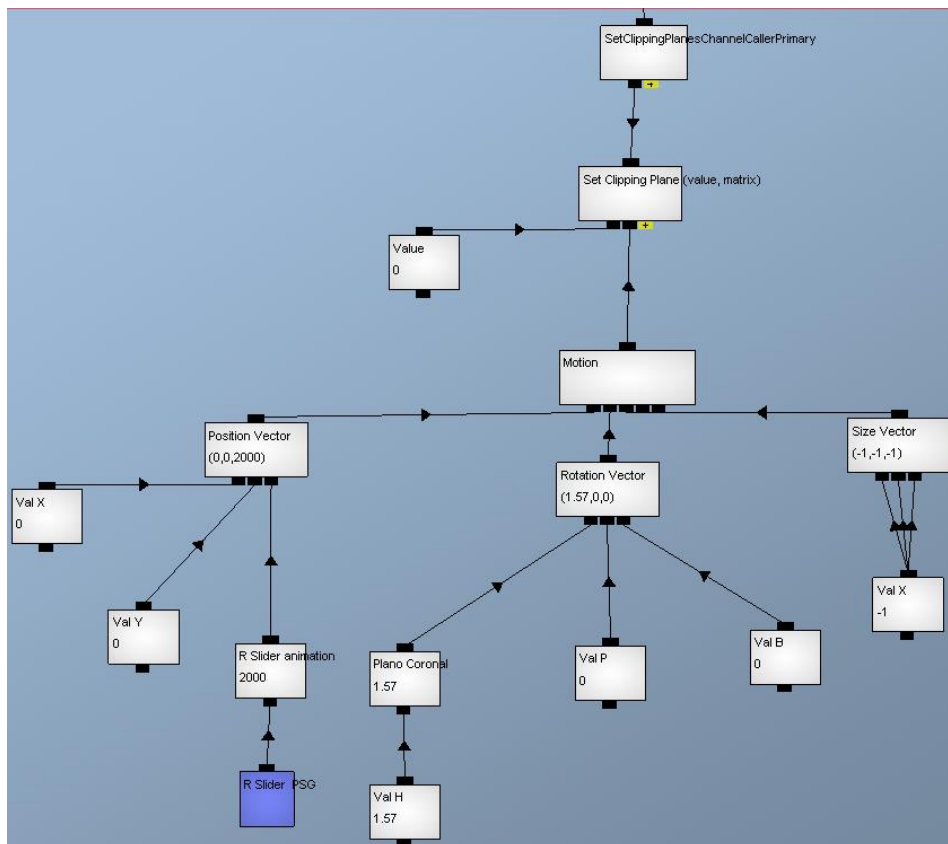


Figura 3.62 Función de plano de recorte (en sus siglas en inglés, Clipping Plane).

Este método controla el movimiento del corte gráfica en el modelo biomédico o no biomédico, para tener el control del plano de corte se implementa el componente deslizante (en sus siglas en inglés slider), enlazado con la implementación de una función que calcula el control de movimiento en X, Y y Z. En la **figura 3.63** se muestra la función lineal que controla el plano de corte.

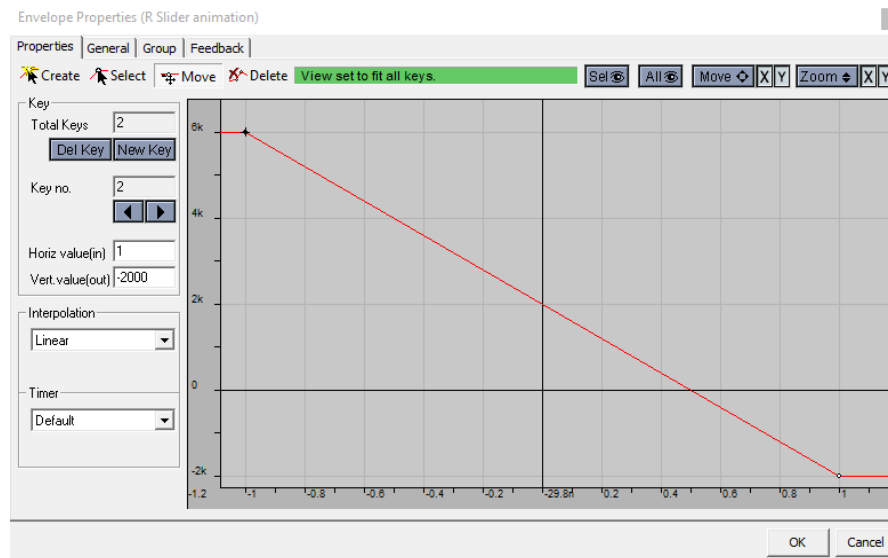


Figura 3.63 Función de control del plano de corte (en sus siglas en inglés, Clipping Plane).

3.5.14 Módulo de interactividad visual

La interactividad es la interacción entre la computadora y el usuario, que consiste en la programación de todas las funciones que afectan a los modelos 3-D, como la visualización, interacción, navegación y modificación de las propiedades de los modelos biomédicos (obtenidos de reconstrucción geométrica por tomografía axial computarizada) y de modelos no biomédicos (obtenidos por diseño asistido por computadora, modelado geométrico). La interacción con los objetos 3-D se ejecuta a través de una función donde el usuario puede rotar, mover y acercar los modelos 3-D. En la **figura 3.64** se muestra el módulo de interactividad visual.

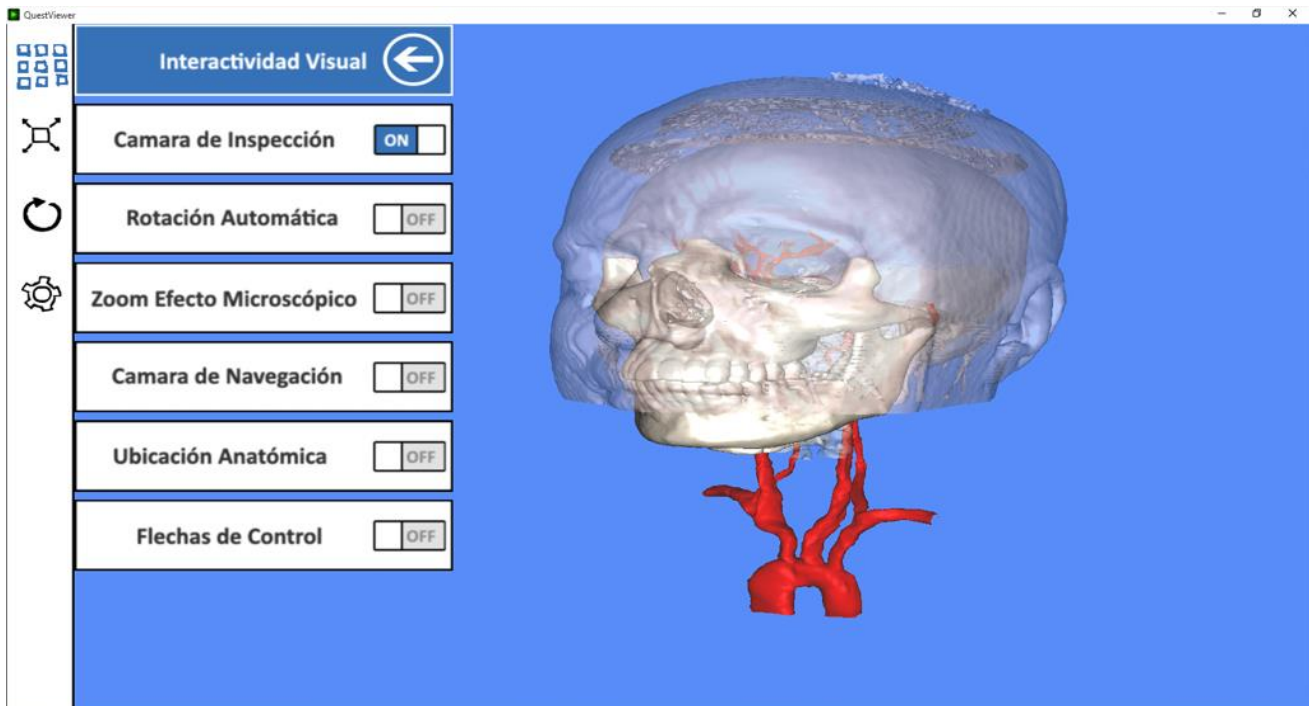


Figura 3.64 Modulo de interactividad visual

Para activar la función de interacción se asignaron entradas de datos por medio de los botones del ratón. La primera función es cuando se ejecuta un clic sostenido en el botón izquierdo y realizando un desplazamiento consecutivo en el eje X, se tiene como respuesta el control de rotación en el eje Y del visor; asimismo, cuando el desplazamiento consecutivo es el eje Y, se tiene como respuesta el control de rotación en el eje X del visor. La segunda función es cuando se ejecuta un clic sostenido en el botón derecho y realizando un desplazamiento consecutivo en el eje X, se tiene como respuesta el control de traslación en el eje X del visor; asimismo, cuando el desplazamiento consecutivo es el eje Y, se tiene como respuesta el control de traslación en el eje Y del visor. La tercera función es cuando se ejecuta un desplazamiento hacia adelante en el botón scroll, se tienen como respuesta un acercamiento en el eje Z del visor; asimismo, cuando el desplazamiento es hacia atrás en el botón scroll, se tiene como respuesta un alejamiento en el eje Z del visor. Existen varias funciones para interactuar con los objetos 3D, debido a varios datos de interactividad (valores escalares, valores vectoriales y valores matriciales), de las funciones de interacción, fue necesario desarrollar un modelo de datos para la interactividad.

3.5.15 Módulo de interactividad visual: Función de posición de la cámara

La primer entrada o hijo del bloque que determina la posición de la cámara, indica la orientación de la misma. La preparación de los cálculos de la perspectiva de la cámara es terminada automáticamente. El valor solo necesita hacer un zoom del factor. Un largo valor de aumento y un pequeño valor de disminución. Algunos necesitan preparar los cálculos más precisos de la perspectiva. Un ejemplo de esto es cuando se quiere especificar un lejano y cercano plano de recorte. El canal que prepara los cálculos de la perspectiva es el bloque de Projection Matrix. Los datos biomédicos como no biomédicos son objetos tridimensionales que no se podrán visualizar si no se realiza un render y este es imposible sin una cámara, pero se puede usar múltiples cámaras en un render. En la *figura 3.65 y 3.66* se muestra la cámara que se utiliza para visualizar las estructuras biomédicas en el sistema de software.

1. Necesitamos una cámara para poder ver nuestro mundo.

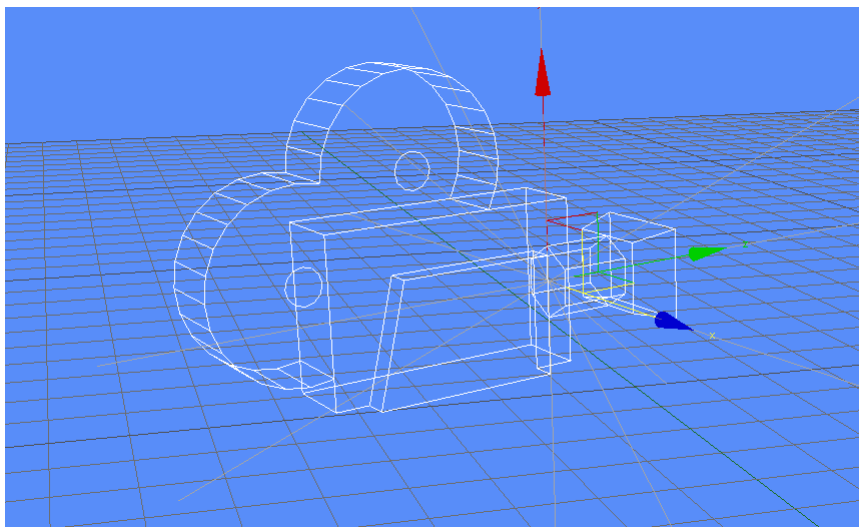


Figura 3.65 Cámara de inspección

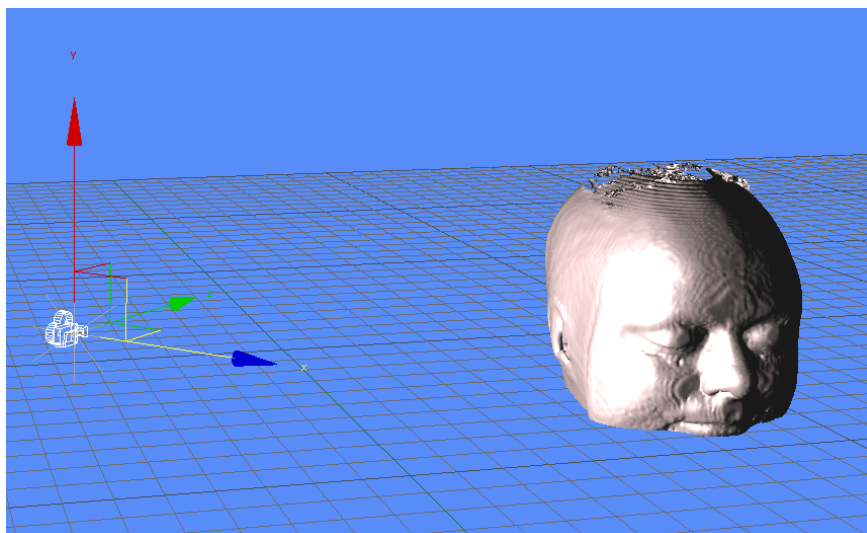


Figura 3.66 Cámara de inspección con estructura biomédica

2. La cámara se crea con una matriz, la matriz especifica la posición de la cámara, la rotación de la cámara y tamaño de la cámara. En la *figura 3.67* se muestra la matriz de la posición de la cámara. Algunas características de la cámara son:

- **Camera Matriz – Matriz:** La orientación y la posición de la cámara.
- **Fog On Off – Value:** Enciende o Apaga el fog (Off = 0, on = 1).
- **Fog Mode: 0 – Lin 1 – Exp 2 – Exp 2 – Value:** El modo Fog (Linear = 0, etc.)
- **FogStart Distancia – Value:** Inicio del Fog (solo usado si el fog está en modo linear)
- **FogEnd Distancia – Value:** Final del fog (solo usado si el fog está en modo linear)
- **FogDensity – Value:** Densidad del fog (solo usado si el fog esté en modo linear)
- **Fog Color R – Value:** El componente Rojo del color del fog.
- **Fog Color G – Value:** El componente Verde del color del fog.
- **Fog Color B – Value:** El componente Azul del color del fog.
- **Zoom Factor – Value:** Se puede usar esté para un aumento y disminución de la lente.
- **Projection Matriz – Matriz:** Se puede conectar una matriz de proyección aquí para lograr otras proyecciones (isométricas por ejemplo).

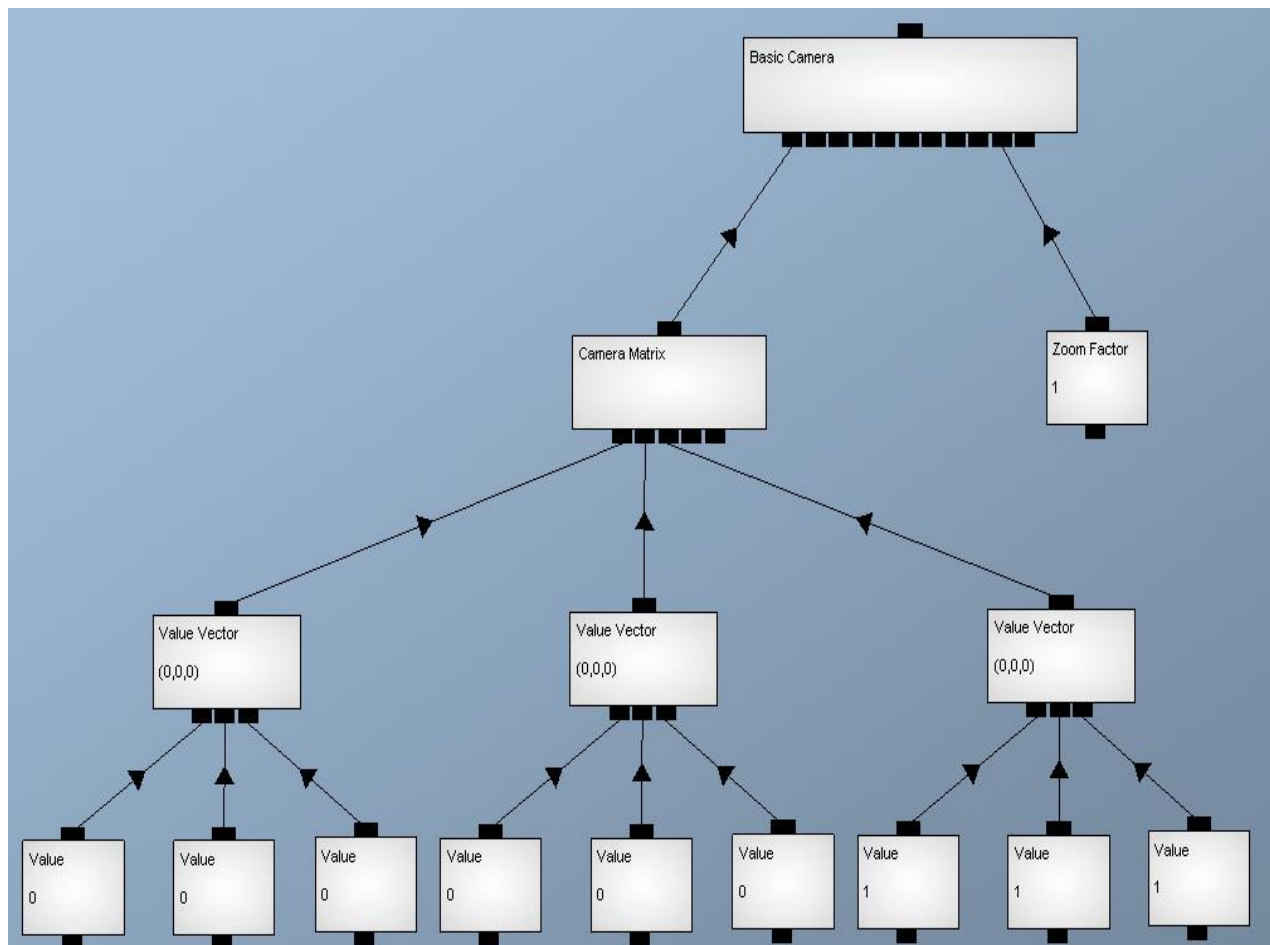


Figura 3.67 Matriz de posición de cámara.

3.5.16 Módulo de interactividad visual: Función de cámara básica

La cámara básica puede especificar el punto de vista para el render de objetos, generalmente el vector de posición, el vector de rotación y el vector de escalamiento son fijos, este tipo de cámara se utilizó para visualizar objetos estáticos como las Interfaces gráficas de usuario (Menús, Fondos, etc.). La cámara realiza una proyección con perspectiva tomando en cuenta la distancia Z de un objeto, lo que permite dar el efecto de una imagen en 3-D, ya que a los objetos más alejados los disminuye de tamaño. En la *figura 3.68* se muestra la proyección con perspectiva.

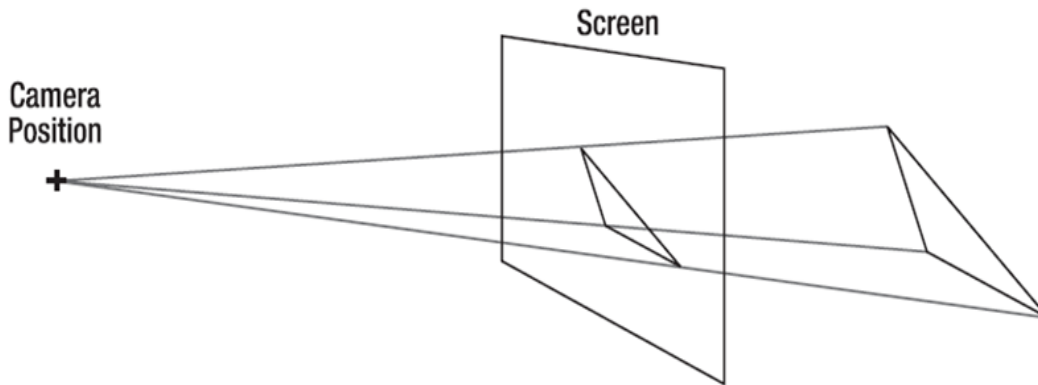


Figura 3.68 Proyección con perspectiva.

3.5.17 Módulo de interactividad visual: Función de cámara ortogonal

La cámara ortogonal puede especificar el punto de vista para el render de objetos, generalmente el vector de posición, el vector de rotación y el vector de escalamiento son fijos, lo que la hace diferente a la cámara básica es la proyección ya que en esta cámara la componente $Z = 0$. Esta cámara fue principalmente usada para visualizar el menú prisma, donde se necesitaba visualizar solo las caras ortogonales. En la *figura 3.69* se muestra la cámara de posición ortogonal.

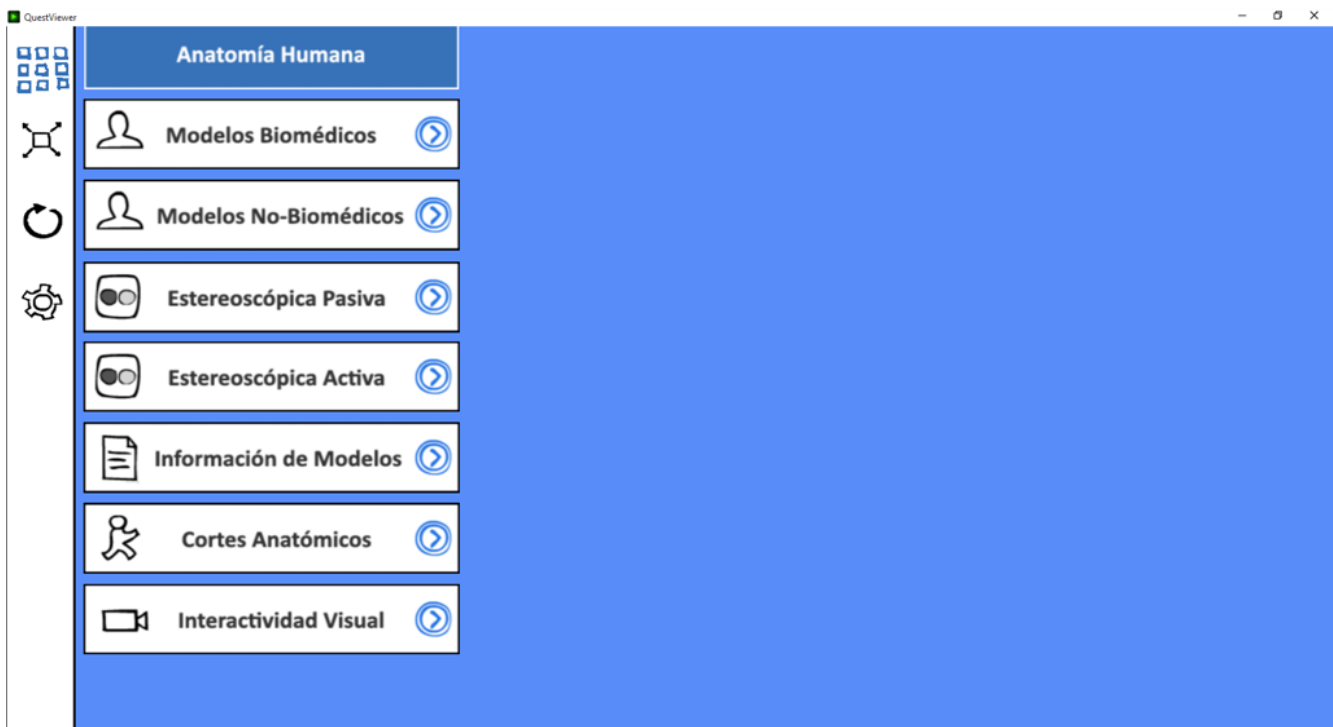


Figura 3.69 Cámara de posición ortogonal.

La cámara realiza una proyección ortogonal, donde la componente Z del objeto no es tomada en cuenta, los objetos más alejados no se ven más pequeños ni los más cercanos más grandes. En la figura se muestra la proyección ortogonal. En la *figura 3.70* se muestra la proyección ortogonal.

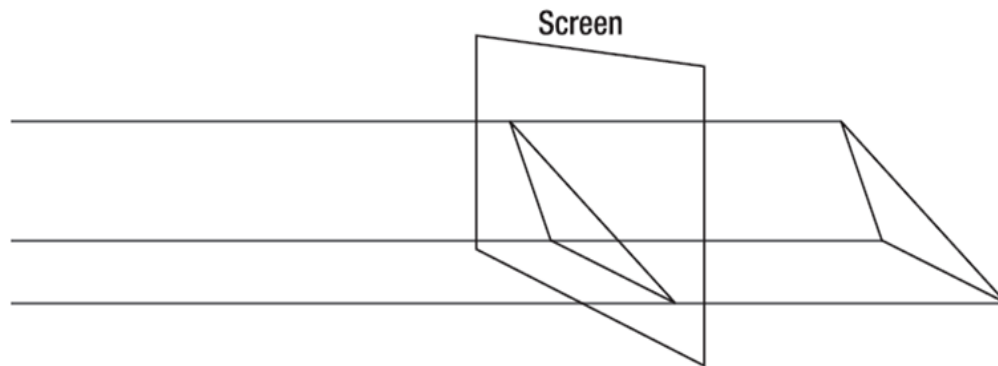


Figura 3.70 Proyección ortogonal.

3.5.18 Módulo de interactividad visual: Función de cámara de inspección de objeto

La cámara de inspección de objetos puede especificar el punto de vista para el render de objetos con un objetivo. Esta cámara podrá rotar en el eje X a 360°, en el eje Y a 360° realizando la sensación de rotación del objeto 3-D. También podrá trasladarse junto con el objetivo en el eje X en ambos sentidos (+ y -) y en el eje Y en ambos sentidos (+ y -) realizando la sensación del movimiento del objeto 3-D y por último se puede mover en el eje Z en ambos sentidos (+ y -) realizando la sensación de alejar y acercar el objeto 3-D. En la *figura 3.71* se muestra las funciones de la cámara de inspección de objetos.

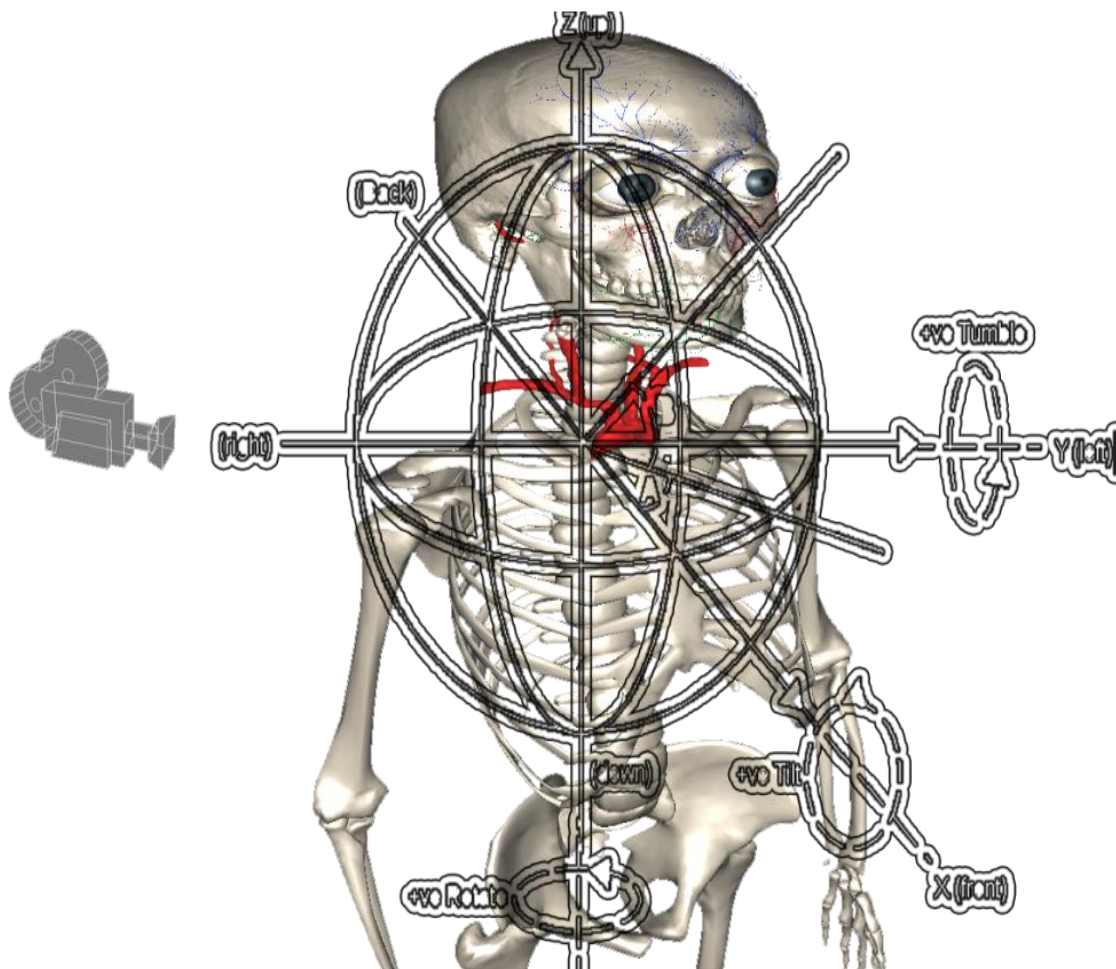


Figura 3.71 Funciones de cámara de inspección de objetos.

Esta cámara fue principalmente usada para visualizar los objetos 3-D almacenados en el programa. En la figura 3.72 se muestra la cámara de inspección activada en el sistema de software biomédico 3-D. En la **figura 3.72** se muestra la cámara de inspección de objetos.

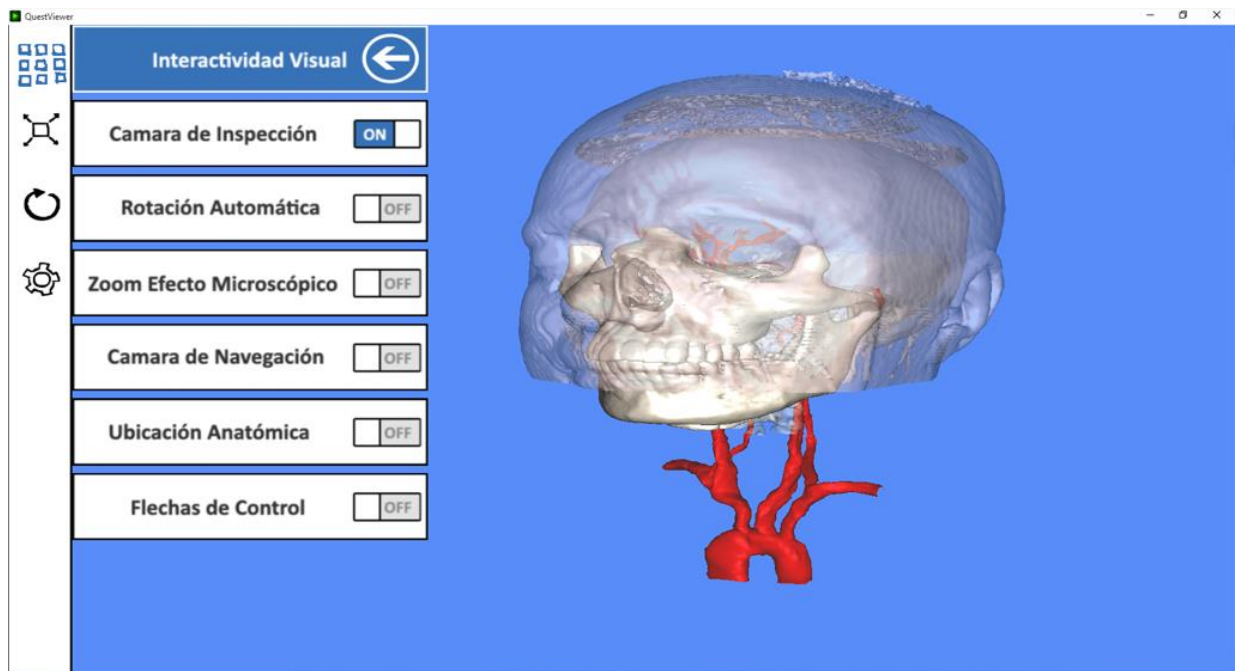


Figura 3.72 Cámara de inspección de objetos.

3.5.19 Módulo de interactividad visual: Función de cámara en tercera persona.

La cámara en tercera persona puede especificar el punto de vista conocido como tercera persona, frecuente en los videojuegos tridimensionales de estilo aventura, rol, etc. Esta vista tiene como característica que el personaje o *avatar* que se controla se ve de cuerpo entero y generalmente de espaldas. Para lograr esta técnica existe una cámara detrás del personaje, la cual lo sigue constantemente a una cierta distancia, ángulo de inclinación y altura. En la mayoría de los juegos se utiliza con frecuencia esta vista para lograr una mayor sensación de inmersión en los espacios tridimensionales. La ventaja de utilizar esta técnica es que es posible visualizar directamente los alrededores del personaje. En la **figura 3.73** se muestra la cámara en tercera persona.

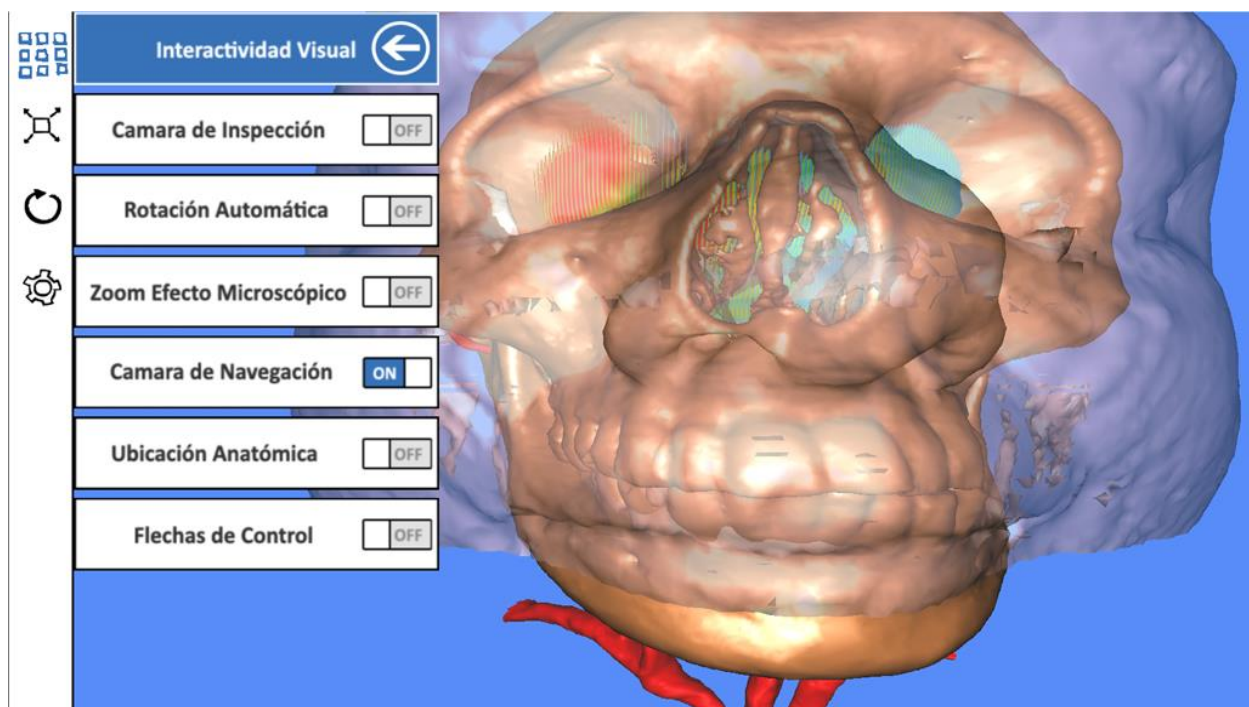


Figura 3.73 Cámara en tercera persona.

Para poder lograr esta navegación se utilizaron varias funciones matemáticas, como el cálculo matricial del movimiento de personaje (traslación y rotación), la interacción del control del avatar para realizar un recorrido virtual se activa mediante el botón respectivo, que es la misma para la cámara, solo que la jerarquía de control de movimiento es avatar - padre y la cámara - hijo. El software tiene programado un personaje que en este caso representa a un glóbulo rojo, pero se pueden modificar o almacenar más personajes con sus animaciones. En la *figura 3.74* se muestra la función desarrollada de la cámara de tercera persona.

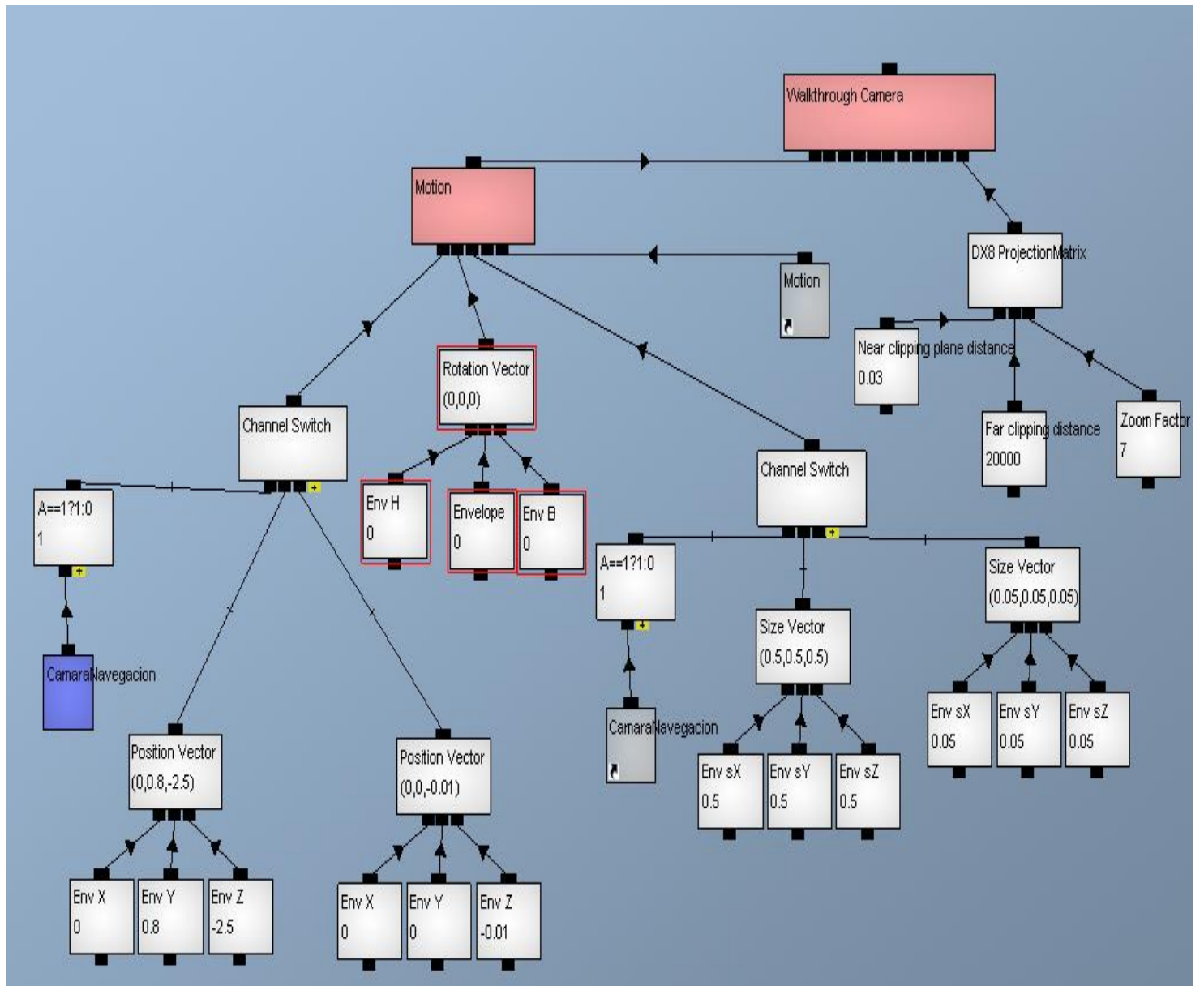


Figura 3.74 Cámara en tercera persona.

3.5.20 Módulo de interactividad visual: Función de auto rotación.

La función de auto rotación se ejecuta para tener movimiento de rotación constante automático de los modelos biomédicos y no biomédicos, sin tener ningún tipo de interacción con la aplicación, proporcionando un tiempo para que el usuario realice las actividades que requiera sobre los modelos (teórica anatómica, análisis anatómico, etc.), el usuario puede ver los modelos en todas sus perspectiva en un ángulo de 360 grados. En la *figura 3.75* se muestra la función de rotación automática.

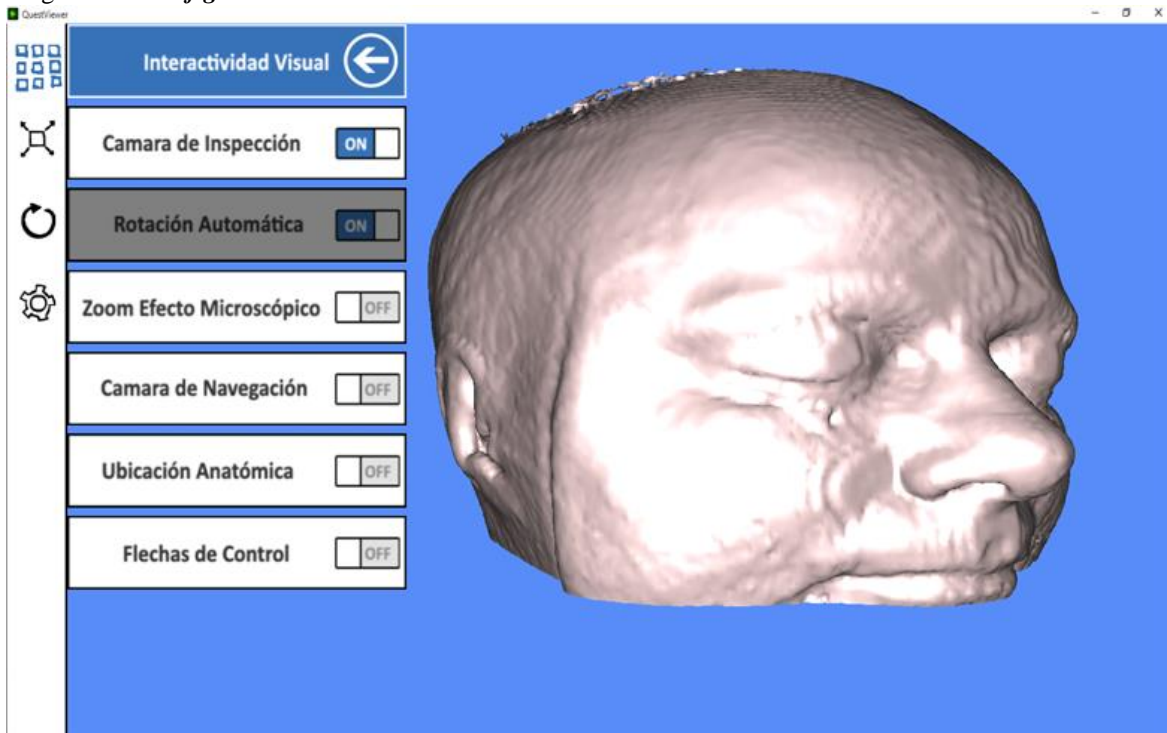


Figura 3.75 Función de rotación automática.

La función de movimiento de rotación constante automático funciona con un módulo de inercia que desencadena un bucle infinito, el valor es positivo o negativa dependiendo de la variable que se ingrese al módulo de inercia, el cálculo del movimiento se realiza en radianes. En la *figura 3.76* se muestra la función de inercia.

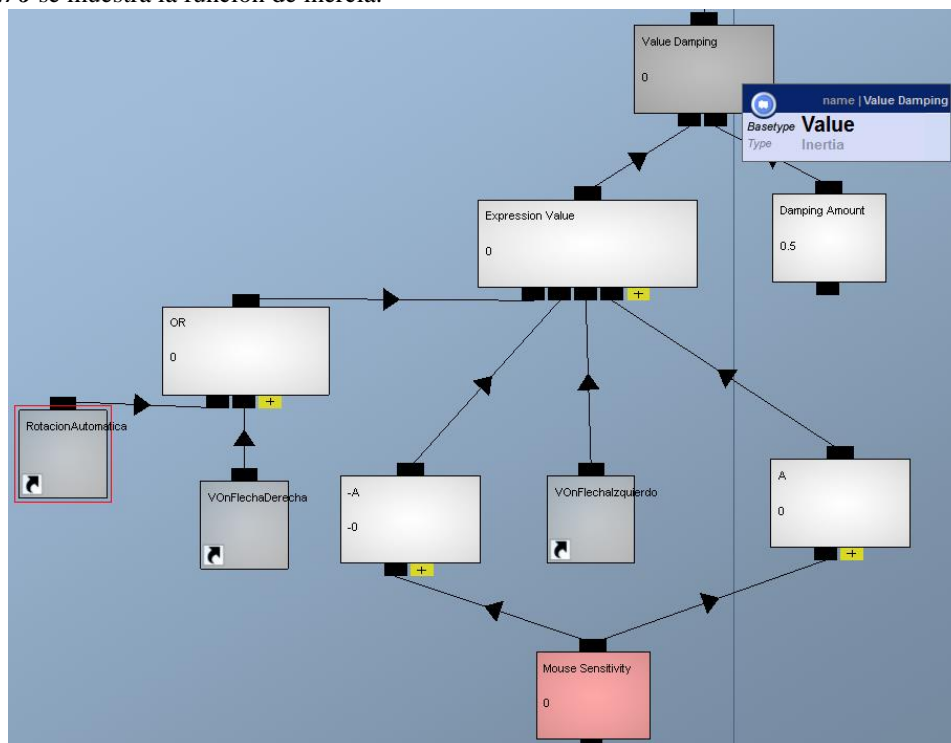


Figura 3.76 Función de inercia.

3.5.21 Módulo de interactividad visual: Función de zoom microscópico.

La función de zoom de microscópico se ejecuta para conseguir el control de una simulación de microscopio de laboratorio, con esta función se amplifica la visión del usuario en una escala de 1x a 20x. En la *figura 3.77* se muestra la función de efecto microscópico.

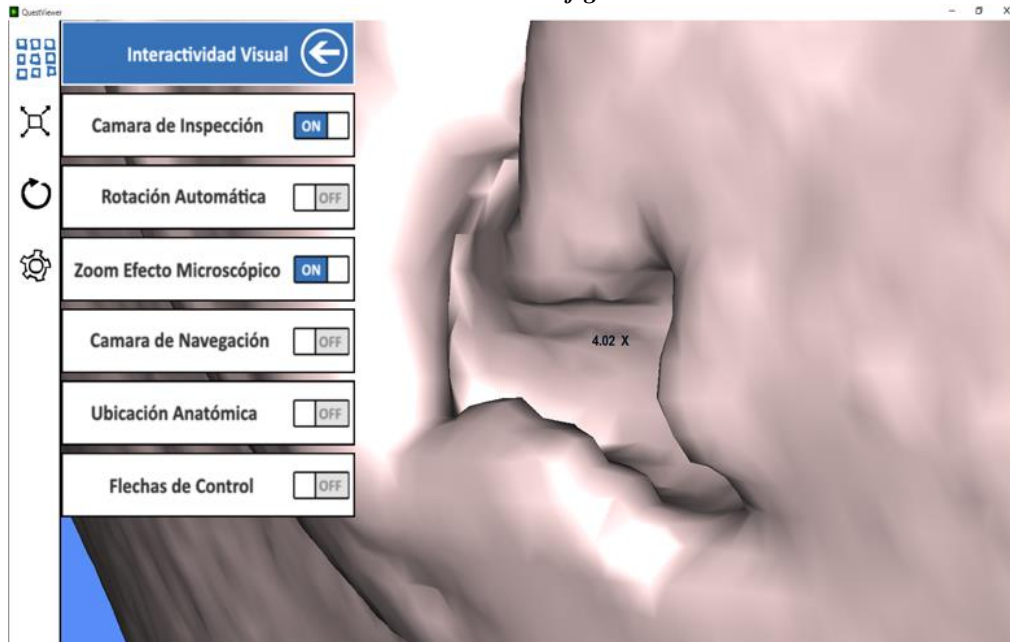


Figura 3.77 Función de efecto microscópico.

La función de zoom de efecto microscópico se realiza mediante un módulo de operador de valor (value operator) donde las propiedades básicas a modificar en el objeto cámara enlazan el valor de zoom. En la *figura 3.78* se muestra la función de la operación para el zoom.

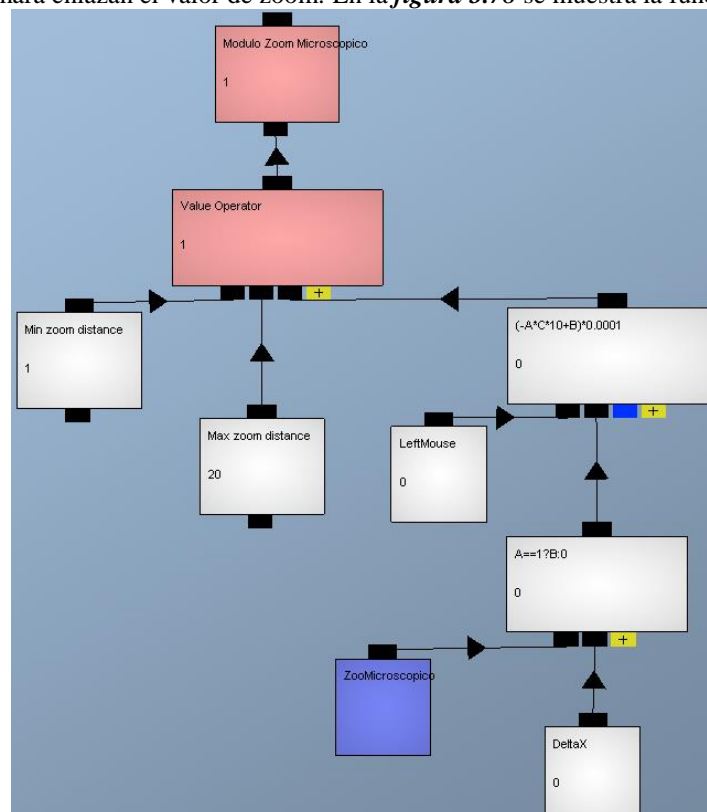


Figura 3.78 Función operación de zoom.

Para activar este cambio de variable dinámica se implementa una función donde el orden de entrada de datos es respecto al desplazamiento offset, si se trata de un desplazamiento hacia delante los datos son positivo, si no los datos son negativos, estos datos se pueden aumentar o disminuir la visión.

CAPÍTULO IV

ANÁLISIS DE RESULTADOS

Al probar el software, se ejecutaron datos para analizar el funcionamiento, las pruebas que se realizaron son para demostrar que el software cumple con los requerimientos del objetivo, así como descubrir defectos en el software antes de tener la versión de liberación. Hay que verificar los resultados de las pruebas que se opera para buscar errores, anomalías o información de atributos no funcionales del software. Al realizar las pruebas se encontraron situaciones donde el comportamiento del software era incorrecto, indeseable o no estaba de acuerdo con su especificación. Tales situaciones son consecuencia de defectos del software. Estas pruebas de defectos tuvieron la finalidad de erradicar el comportamiento indeseable del software, como cálculos incorrectos y corrupción de datos. Las pruebas del software biomédico se enfocaron en el desarrollo y funcionalidad de los módulos que generan objetos:

- La generación de datos biomédicos.
- La generación de datos no biomédicos.

4.1 Pruebas de la generación de datos biomédicos.

Las pruebas de unidad se enfocaron a que cumplieran con la validación y verificación del software biomédico, la validación es adquirir un *modelo biomédico tridimensional* de calidad con una estimación a la realidad, al verificar si cumplía con la validación se realizaron pruebas como:

- Varios estudios de tomográficos para adquirir el modelo con la calidad de reconstrucción que se validó.

Para verificar si el modelo que se desarrolló cumplía con la calidad se tomaron algunos parámetros para su validación:

- La distancia de cortes transversales, los cortes fueron de 1 [cm], .5 [cm], 1[mm] y 0.5 [mm].
- Con respecto a estos cortes se probaron los modelos si el ambiente de desarrollo tenía la capacidad para procesar la información.
- Para comprobar esta fase, el modelo biomédico pasa por varios filtros, como se explicó en el capítulo de desarrollo.
- El filtro que se hace referencia es al método de importación en ambiente para desarrollar la programación, esto se realizó para tener en cuenta que numero polígonos puede procesar el render del ambiente de programación.

El resultado de las pruebas en la unidad de datos biomédicos fue que un modelo biomédico de más de 2,000,000 de polígonos, el ambiente de desarrollo no puede procesar, son demasiados polígonos por objeto, el ambiente tiene un límite para procesar polígonos por objeto, esta prueba en la unidad de datos biomédicos ayudo para validar el corte de 1[mm] que al reconstrucción de volumen se obtiene un modelo biomédico reconstruido con 1,600,000 polígonos, se verifico la reconstrucción validada en el ambiente de programación, este número de polígonos puede procesar el render del ambiente de programación, por lo que el límite de polígonos por objeto es de 1,600,000 polígonos. En la *figura 4.1* se muestra los datos biomédicos.

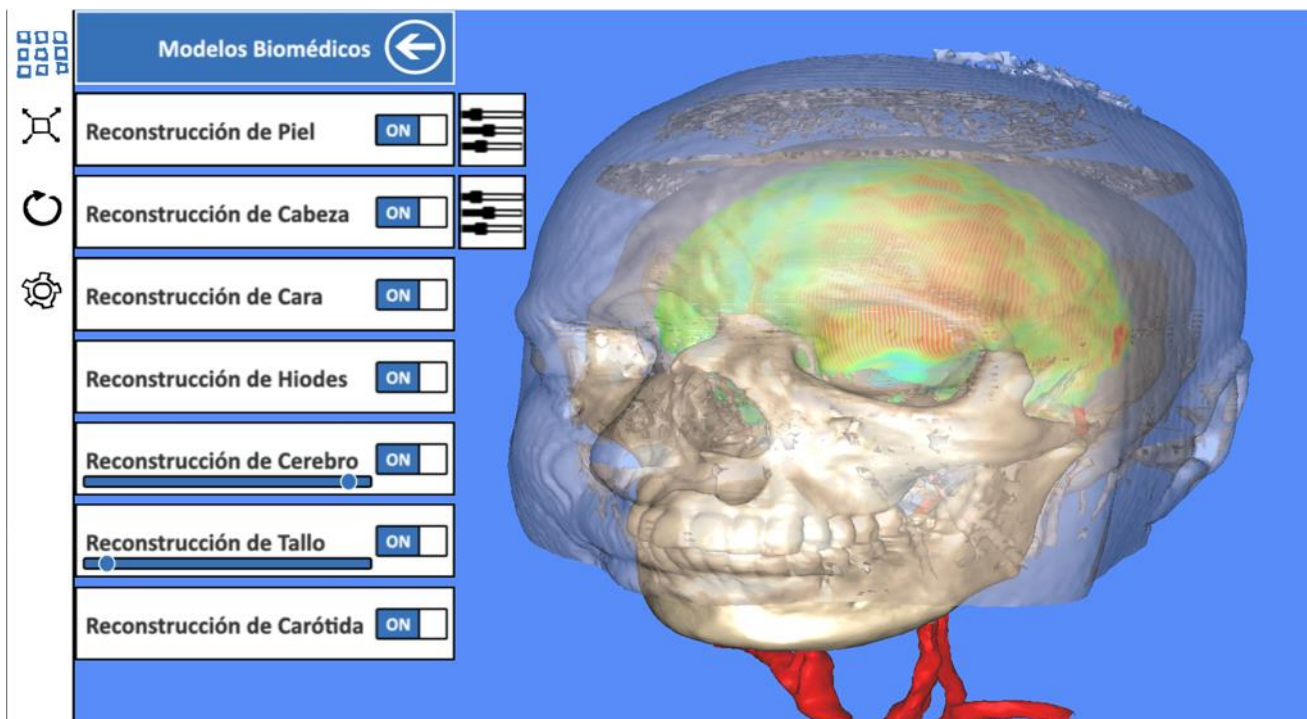


Figura 4.1 Resultado de la generación de datos biomédicos.

4.2 Pruebas de la generación de datos no biomédicos.

Las pruebas de unidad se enfocaron que cumplieran con la validación y verificación del software biomédico, la validación es adquirir un modelo *no biomédico tridimensional* de calidad con una estimación a la realidad, en la **figura 4.2** se muestra el resultado de la generación de datos no biomédicos, al verificar si cumplía con la validación se realizaron pruebas como:

- Adquirir modelos no biomédicos tridimensionales por medio de generación de estructuras tridimensional con CAD cercanos a la realidad.
- Los modelos no biomédicos como se explicó en el capítulo de desarrollo son estructuras generadas por medio de modelado tridimensional pero logrando simular la realidad, se usa este método, ya que no se pueden adquirir por medio de tomografía por su baja densidad.

Para verificar si el modelo que se desarrolló cumplía con la calidad se tomaron algunos parámetros para su validación:

- Se realizó pruebas de simulación de la realidad de los modelos no biomédicos, para esta verificación se consultó a especialistas con los conocimientos de anatomía médica.
- Se realizó pruebas de procesamiento en el ambiente de desarrollo para verificar que el ambiente no tiene problemas de procesamiento de los modelos, no se presentó ningún problema al procesar los modelos no biomédicos, ya que el número de polígonos son máximo 500,000 polígonos, con este número no hay consecuencia para poder desarrollar con los modelos.
- Se realizó pruebas de tamaño de los modelos biomédicos, como no son de reconstrucción se debe cambiar el tamaño conforme al ensamblaje más realista
- por último se realizó pruebas de color y textura lo más cercano a las partes humanas.

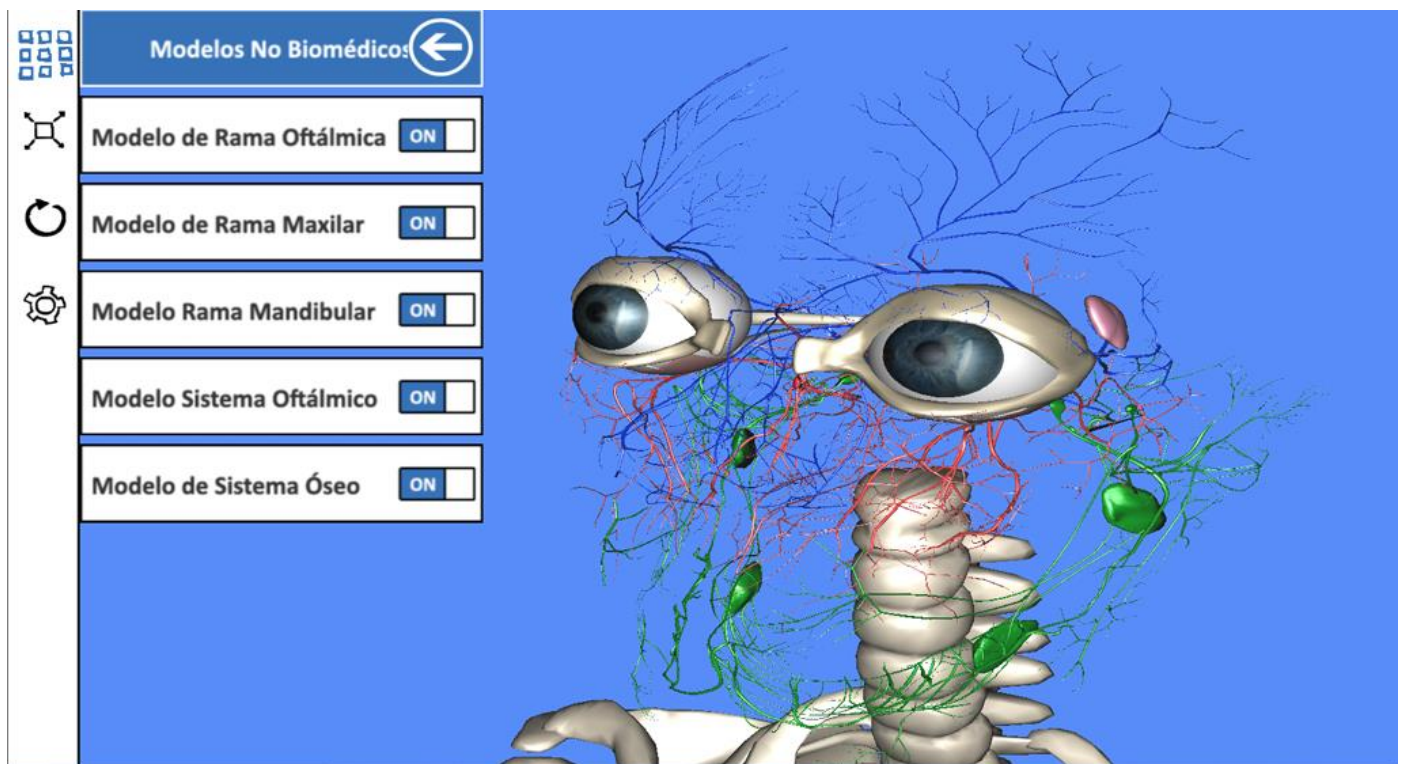


Figura 4.2 Resultado de la generación de datos no biomédicos.

4.3 Pruebas de los componentes del sistema de software biomédico 3-D

Las pruebas de componentes del sistema biomédico se realizaron en los módulos de interacción de la aplicación biomédica, en estas pruebas se verificó el desarrollo de los módulos que componen al sistema biomédico, los módulos están especificados en el capítulo de desarrollo.

Las funciones de los componentes que conforman el sistema biomédico se especificó en la verificación de la interacción, se requirió un poco de tiempo, los componentes que conforman el sistema ejecutan funciones matemáticas que realizan la simulación de los modelos tridimensionales, si los componentes presentan un bug, el usuario tendría problemas al ejecutar adecuadamente la simulación en el sistema biomédico, esto llevaría a un mal uso del sistema provocando posiblemente una mala usabilidad, el sistema se validó si es amigable, usable y si presenta un buen funcionamiento, la versión de liberación del sistema biomédico se debe validar y verificar en una versión estable, esto se consigue con las respectivas validaciones y verificaciones de los módulos que componen el sistema.

4.3.1 Pruebas del componente de la iluminación.

Las pruebas de componente para la iluminación de la aplicación biomédica se verificaron las posiciones vectoriales de las luces, al inicio del desarrollo las posiciones estaban establecidas, pero conforme se realizaban las pruebas de simulación de la visualización no se cumplía con las validaciones de visibilidad, estas posiciones iniciales no cumplían con el objetivo validado, tener un impacto realista, aunque se tenga un modelo complejo pero luces deficientes, no se podrá tener el realismo que se requiere, la luz es muy importante para el efecto tridimensional, como se mencionó anteriormente las luces estaban en una posición estable.

El resultado para resolver y cumplir con la validación se solucionó en convertir la posición de las luces de estáticas a dinámicas, esto se realizó especificando padres e hijos de los objetos de iluminación, la luz se convirtió en el hijo de los objetos ejecutando un anclaje de la luz con la cámara de visualización de los modelos biomédicos, verificando este método de iluminación se consiguió validar el realismo de simulación de los modelos biomédicos. En la *figura 4.3* se muestra el resultado de la componente de iluminación.

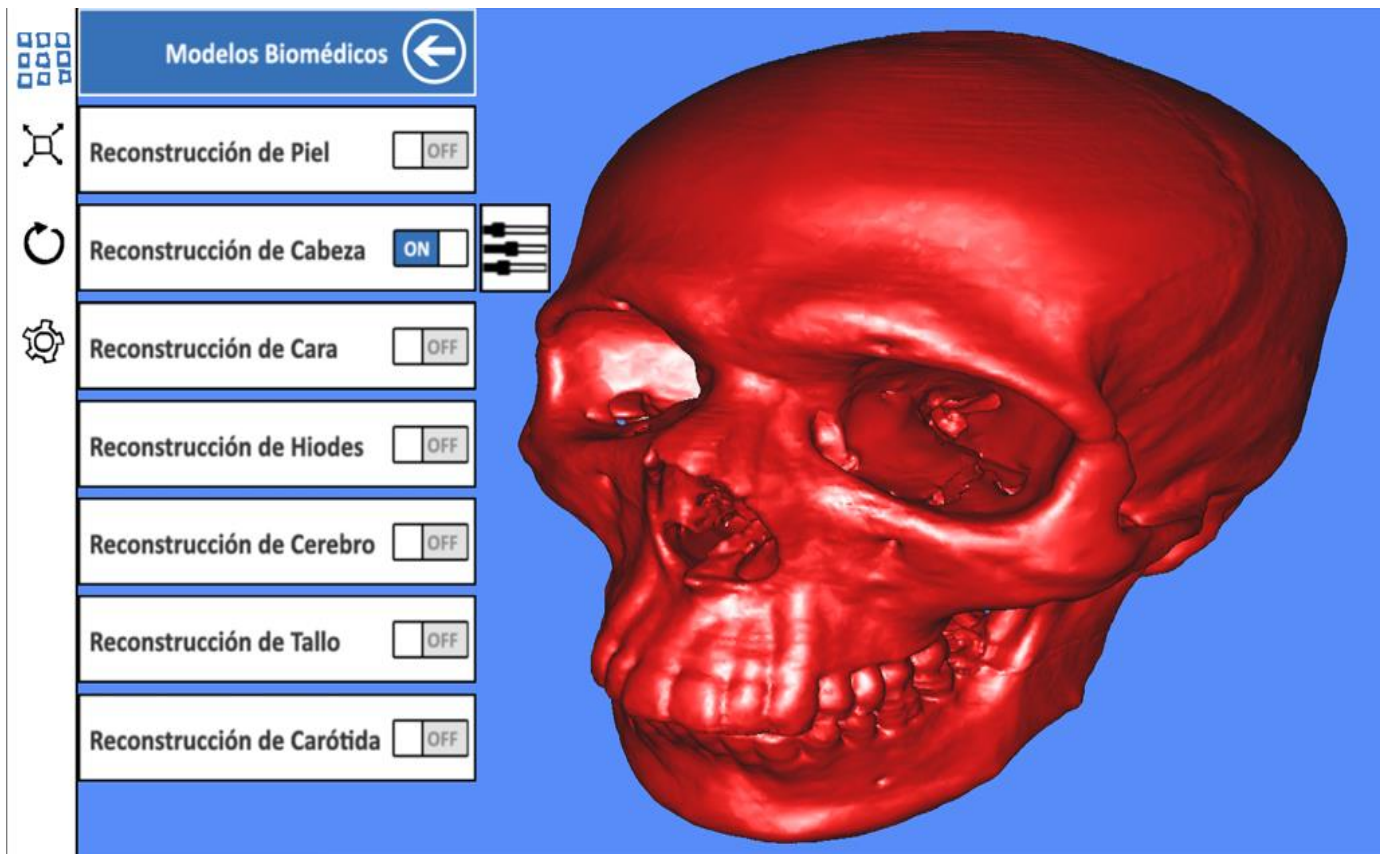


Figura 4.3 Resultado del componente de la iluminación.

4.3.2 Pruebas de verificación de la cámara ortogonal

Las pruebas de componente para la cámara ortogonal se enfocaron en cumplir con la eliminación de la perspectiva, ya que se necesitaba eliminar esta propiedad de la cámara, para no visualizar las caras laterales del menú con forma de prisma, solo es necesario visualizar el plano ortogonal, así podemos interpretar un prisma como una interfaz menú pero en realidad es un prisma con el efecto ortogonal y perspectiva con valor de variable perspectiva igual a cero. En la *figura 4.4* se muestra el resultado de la cámara ortogonal.



Figura 4.4 Resultado de la cámara ortogonal.

4.3.3 Pruebas de verificación de la cámara básica

Las pruebas de componente para la cámara básica se enfocaron que cumpliera con la distancia entre la cámara y las interfaces gráficas de usuario, ya que la funcionalidad principal de esta cámara es la visualización de todas las interfaces gráficas de usuario. En la *figura 4.5* se muestra el resultado de la cámara básica.



Figura 4.5 Resultado de la cámara básica.

4.3.4 Pruebas de verificación de la cámara de inspección de objetos

Las pruebas de componente para la cámara de inspección de objeto se enfocaron que cumpliera con la visualización en 360°, el método de funcionamiento se explicó anteriormente en el capítulo de desarrollo, en esta parte se verificaron las interacciones que se usan para realizar el método de movilidad en 360°, se verifico el movimiento en los ejes (X, Y) y el acercamiento o alejamiento en el eje Z, para resolver esta parte de verificaron varias veces el proceso calibrando las funciones hasta llegar a cumplir la validación de esta cámara. En la *figura 4.6* se muestra el resultado de la cámara de inspección de objeto.

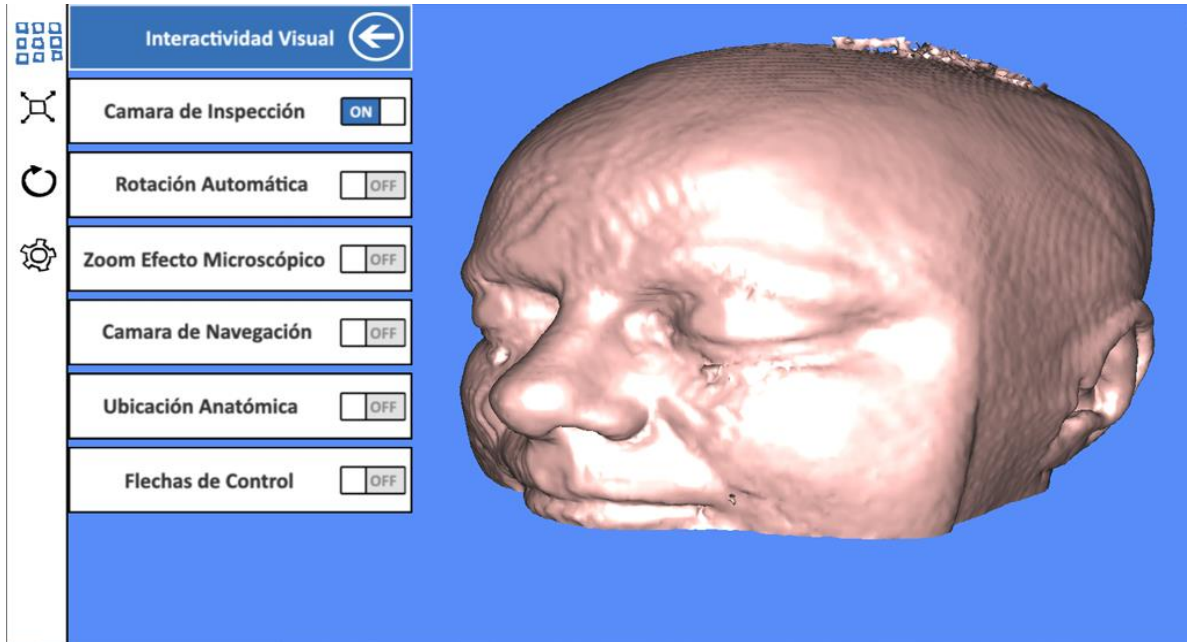


Figura 4.6 Resultado de la cámara de inspección de objeto.

4.3.5 Pruebas de verificación de la cámara en tercera persona.

Las pruebas de componente para la cámara en tercera persona se enfocaron que cumpliera con las funciones de rotación para especificar la dirección del viaje, el avance y retroceso del punto de vista, para visualizar internamente los modelos biomédicos, simulando un viaje en el interior en los modelos biomédicos, se realizaron varias verificaciones en los cálculos matemáticos hasta tener la validación del módulo de interacción, consiguiendo la similitud de una cámara en tercera persona que tienen los video juegos, se logró la validación, tener una cámara con una buena usabilidad. En la *figura 4.7* se muestra el resultado de la cámara de tercera persona.

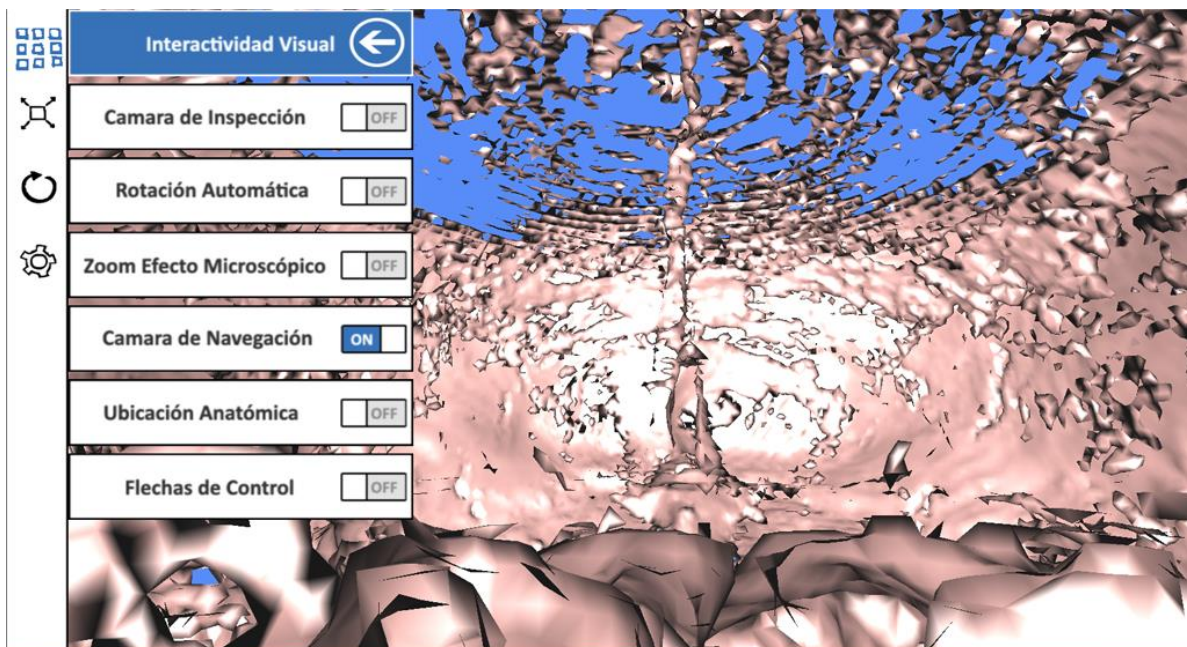


Figura 4.7 Resultado de la cámara en tercera persona.

4.3.6 Pruebas de verificación para la detección de la colisión del mouse.

Las pruebas de componente para la detección de la colisión del mouse se enfocaron en que funcionara correctamente la interacción del mouse con todas las interfaces de usuario, el comportamiento de esta colisión no causo problemas y el resulta es una interacción sin problemas con las interfaces que ejecutan los diferentes componentes del software. En la *figura 4.8* se muestra el resultado de la detección de la colisión del mouse.

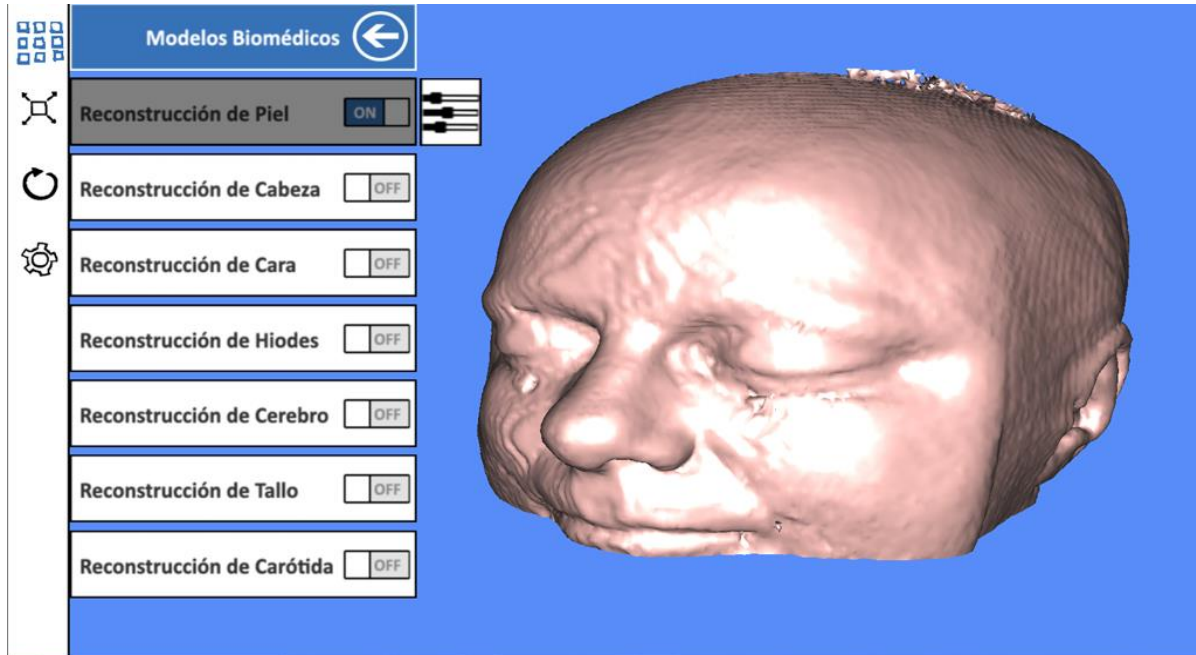


Figura 4.8 Resultado de la detección de la colisión del mouse.

4.3.7 Pruebas de verificación para la colisión de objetos

Las prueba de componente para la colisión de objetos se enfocaron en la detección correcta del mouse sobre todos los objetos tipo primitiva geométrica, esto se refiere a los objetos que tienen volumen en el espacio, la verificación es que se detecte la colisión del mouse con un polígono o varios polígonos que conforman a los modelos biomédicos o no biomédicos de la aplicación. En la *figura 4.9* se muestra el resultado de la detección de la colisión de objetos.

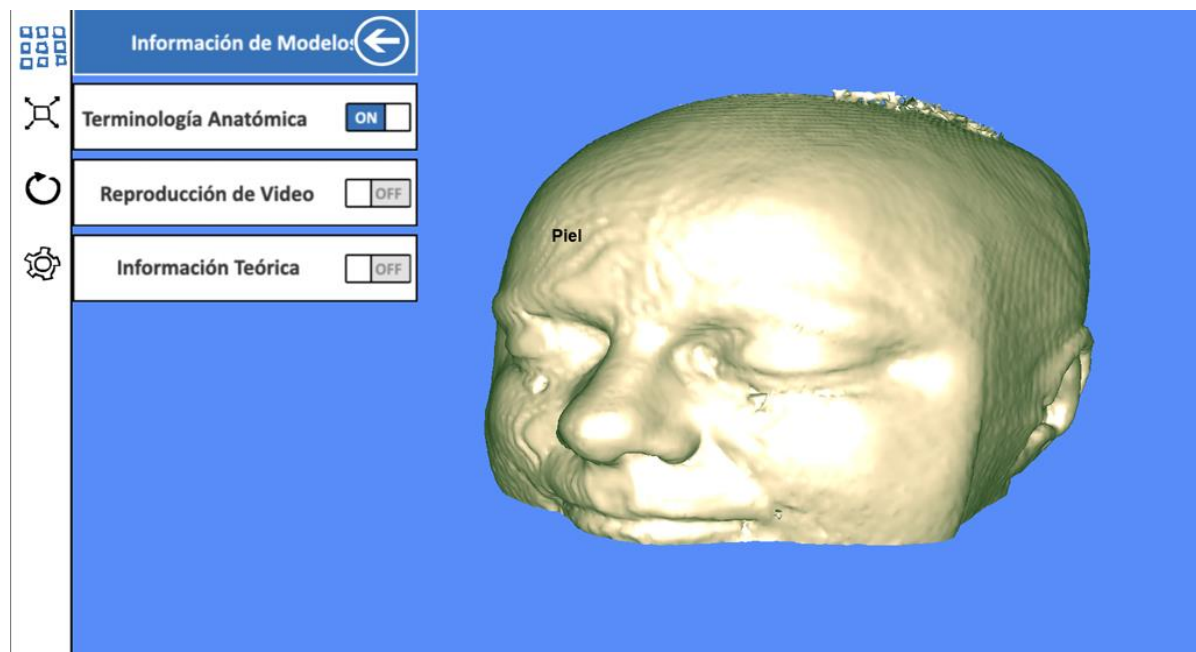


Figura 4.9 Resultado de la detección de la colisión de objetos.

4.3.8 Pruebas de verificación para la colisión de comprobación por rayo

Las prueba de componente para la colisión de comprobación por rayo se enfocaron en la detección por jerarquía de objetos, la verificación consiste en realizar pruebas con un rayo que emite la librería “collisionraycheck” con el que se detecta las capas geométricas de los modelos biomédico y no biomédicos, esto quiere decir que si un objeto tiene otro adentro, el primero que debe detectar es el externo y el interno lo detecta hasta quitar el externo, la verificación de este tipo de colisión se invirtió tiempo para validar, con el cual se identifican los modelos biomédicos y no biomédicos, al poder identificar a los objetos generamos un id por cada objeto con un valor numérico que se utiliza para enlazar el valor numérico con un arreglo que contiene cadenas de texto, que se refieren al nombre de etiqueta que le corresponde para nombrar a los modelos biomédicos y no biomédicos. En la *figura 4.10* se muestra el resultado de la detección de la colisión de comprobación por rayo.

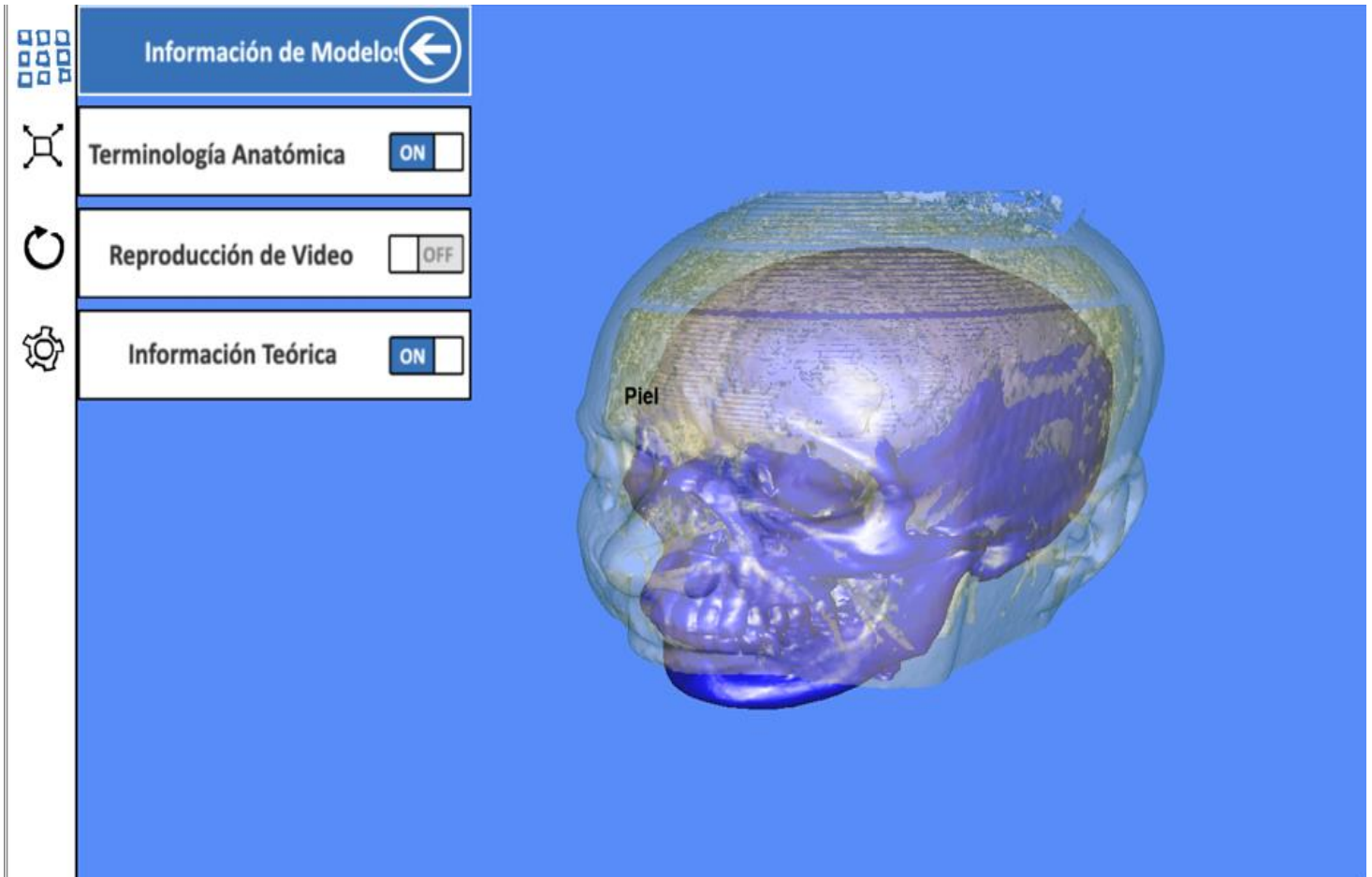


Figura 4.10 Resultado de la detección de la colisión de comprobación por rayo.

4.3.9 Pruebas de verificación para la función estereoscópica pasiva

La prueba de componente para estereoscópica pasiva se enfocaron en activar y controlar la técnica de inmersión 3-D usando gafas anáglifo. El problema que se encontró fue un problema de la separación de la distancia entre los ojos, la solución para este desfasamiento que provocaba fantasma se realizó un funcionamiento en forma estática y dinámica, cuando se activa el slider se utiliza el modo estático donde se obtiene el valor para la distancia entre los ojos, el modo dinámico es cuando la cámara se mueve hacia adelante o hacia atrás, la operación que se realiza es una distancia entre dos puntos para que se calcule el mismo valor constante para la distancia entre los ojos y que no se produzca fantasmas al momento de visualizar, la operación la realiza el modulo llamado Distancia Cámara Apuntador. En la *figura 4.11* se muestra los resultados de la estereoscópica pasiva.

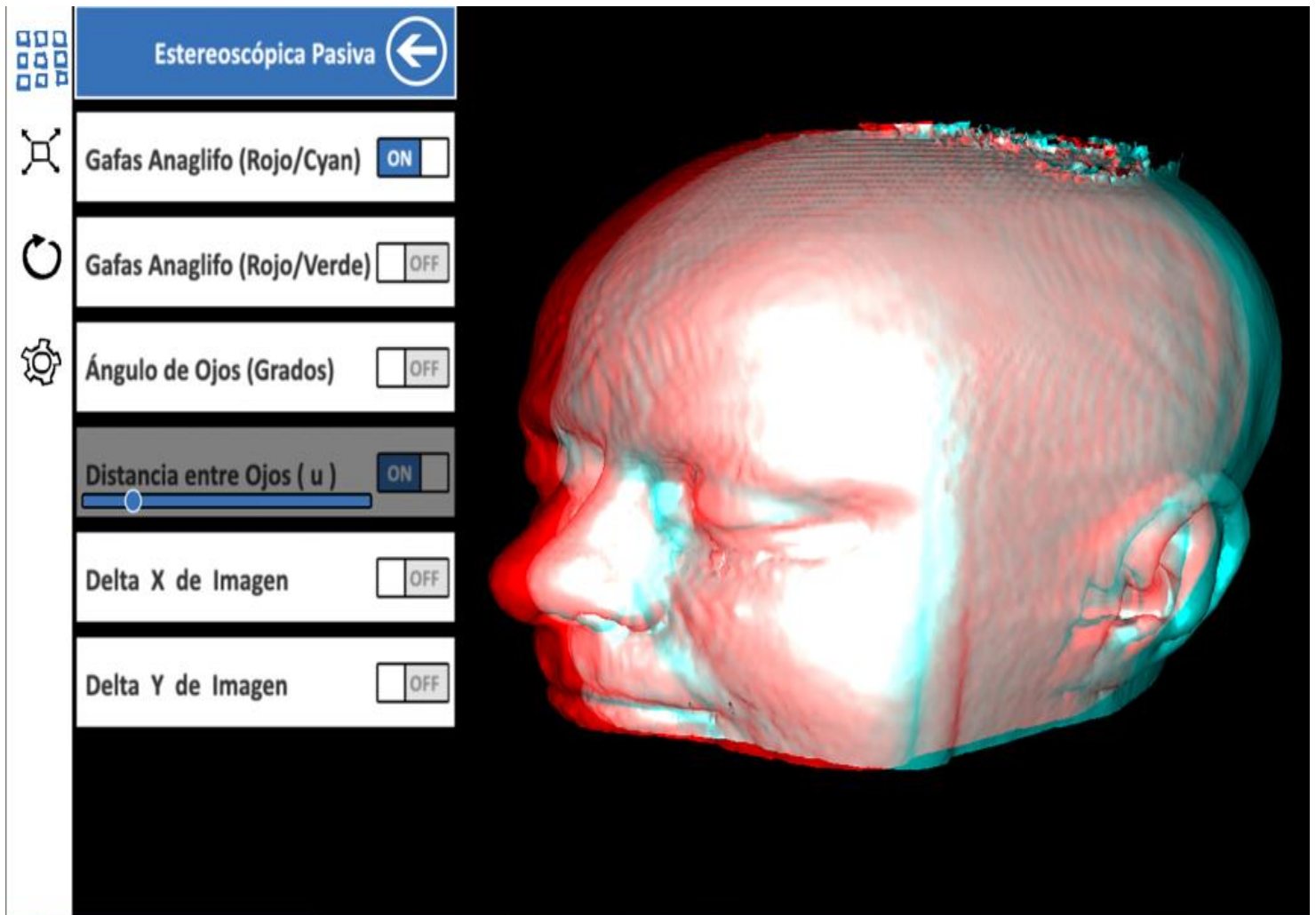


Figura 4.11 Resultado de estereoscópica pasiva.

4.3.10 Pruebas de verificación para los cortes anatómicos

La prueba de componte para los cortes anatómicos se enfocó en el funcionamiento de los diferentes cortes (corte plano sagital, corte plano coronal y el corte plano transversal), se tuvieron varios problemas para tener el resultado esperado, ya que la función de los cortes tendría que realizarse en tiempo real para todo modelo sea biomédico y no biomédico. El resultado con el cual se resolvió fue utilizando un módulo de render que genera un plano con movimiento, a éste modulo se le conoce como plano de recorte (en sus siglas en inglés, Clipping Plane). En la *figura 4.12* se muestra los resultados de los cortes anatómicos.

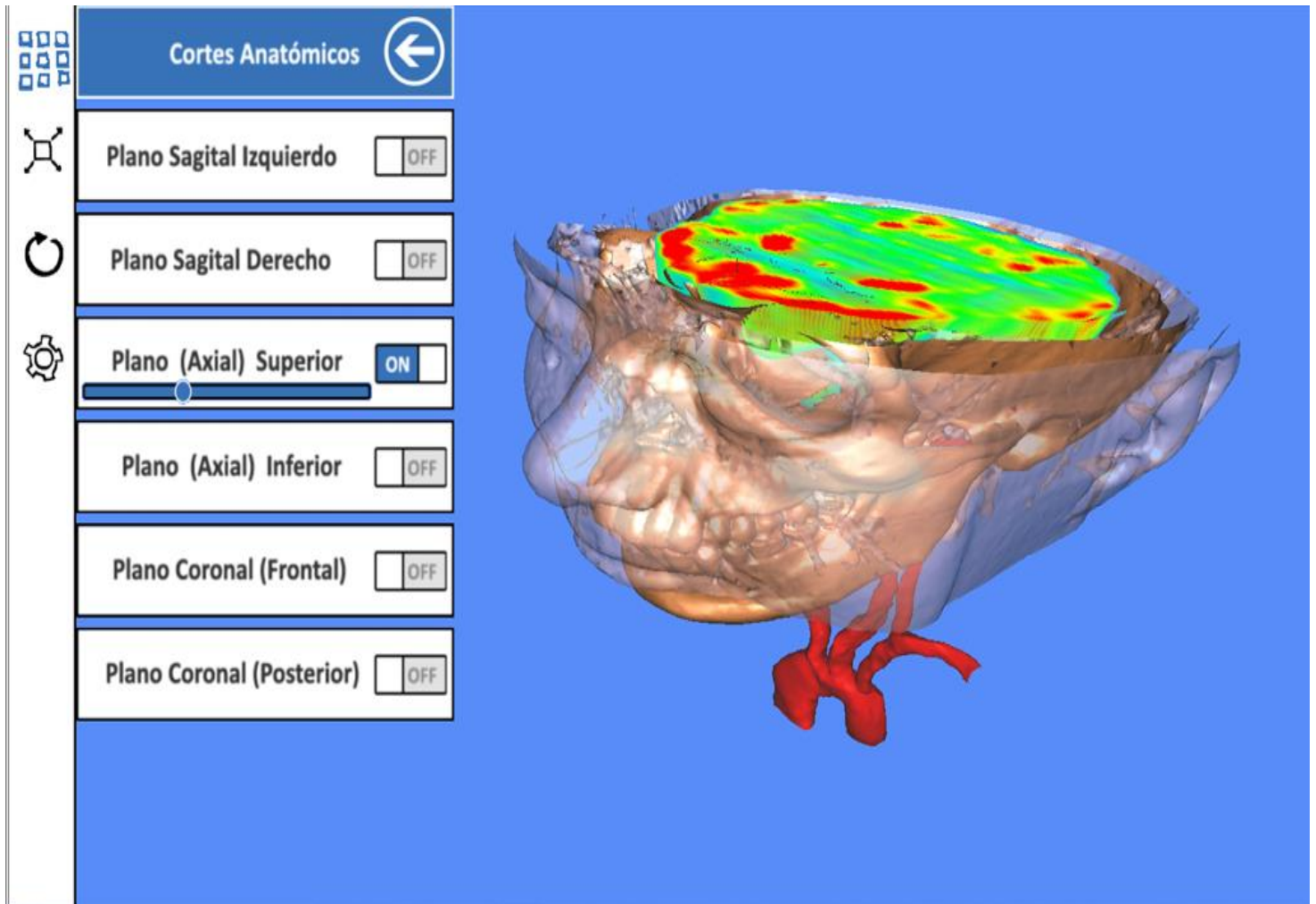


Figura 4.12 Resultado de Cortes Anatómicos.

El desarrollo del software no se detiene cuando se entrega, sino que continúa a lo largo de la vida de éste. Después de distribuir un sistema, inevitablemente debe modificarse, con la finalidad de mantenerlo útil. Las expectativas del usuario generan nuevos requerimientos para el software existente. Por lo general, los sistemas de software tienen una vida muy larga. Por ende, la ingeniería de software se debe considerar como un proceso en espiral, con requerimientos, diseño, implementación y pruebas continuas, a lo largo de la vida del sistema, esta continuidad de pruebas y modificaciones que es conocida como evoluciones del software.

La evolución del software puede volverse evidente incluso antes de que se entregue, de manera que las futuras versiones estarían en desarrollo antes de que se libere la versión actual. La evolución es la fase donde es posible hacer cambios significativos a la arquitectura y la funcionalidad del software. Durante el servicio, los únicos cambios que se realizan son relativamente pequeños. Durante la evolución, el software se usa con éxito y hay un flujo constante de propuestas de cambios a los requerimientos.

CONCLUSIONES

Justo Sierra *“Nacionalizar la ciencia, mexicanizar el saber” y “que en la medida en una nación genere su propia tecnología, podrá realizar una verdadera revolución, desarrollándose en todos los aspectos y alcanzando un mejor nivel de vida para la mayor parte de su sociedad”.*

El sistema de software biomédico fue desarrollado en la Facultad de Ingeniería de la UNAM con la dirección de la M.I Jimena Olveres Montiel, nos encontramos en el camino con paradigmas, ya que la tecnología en cierta forma puede causar miedo y tensión para algunos educadores o estudiantes cuando se introducen a una nueva tecnología específica que puede influir en su capacidad intelectual, que tenga características propias para mejorar las oportunidades del aprendizaje autorregulado, muchos educadores a veces pasan por un mal momento, y la idea de dominar una nueva tecnología como herramienta de enseñanza – aprendizaje, que potenciaría el aprendizaje del estudiante, puede crear más tensión. Las barreras comienzan a desarrollarse entre el educador y la nueva herramienta de aprendizaje porque no hay tiempo para aprender y entender cómo utilizar la nueva herramienta. A menudo hay muy poca evidencia de resultados positivos de una nueva tecnología de reciente introducción, las ventajas se pueden introducir, sin embargo, el por qué, qué, y ¿qué pasa con" cuestiones relativas a la tecnología deben ser respondidas para que la tecnología sea aceptada, experimentada, y aprobada.

El desarrollo del sistemas de software biomédico para el aprendizaje en un entorno 3-D requiere complejas y elementos altamente involucrados de ingeniería de software y la informática. Para muchos educadores, un entorno de aprendizaje 3-D no sería percibido como una ventaja debido a que la "plasticidad" asociado con los mundos virtuales, en otras palabras, la facilidad de construcción y cambiando el entorno de aprendizaje según sea necesario, no sería entendido hasta que el educador jugará y experimentará con los mundos virtuales. Las valiosas herramientas de comunicación integradas en un sistema de software biomédico 3-D, y la facilidad asociada a la zona de aprendizaje para proporcionar orientación y asistencia, no pueden realizarse plenamente. Además, algunos educadores siguen sin ver e-learning más allá de aprender los sistemas de gestión y no han tenido una exposición a los mundos virtuales. Por lo tanto, un problema de baja compatibilidad se presenta en el medio académico.

Los medios de adaptación y aprendizaje, se realizan con la interacción con un entorno que nos rodea, las teorías del aprendizaje tratan de explicar la complejidad del aprendizaje, y muchos educadores se están alejando del desarrollo del conocimiento y están adoptando un cambio de paradigma que es "un proceso de caminar en mayor o menor grado de alineación con un entorno dinámico". Por lo tanto, la teoría del aprendizaje encaja perfectamente en el sistema de software biomédico para el aprendizaje en un entorno 3-D, apoyando a la comunicación y el compartir el conocimiento, la comunicación es un aspecto esencial en la educación. Los beneficios asociados con el desarrollo del sistema de software biomédico para el aprendizaje en un entorno 3-D son transferir habilidades que le permitirán mejorar el trabajo a los estudiantes, ya que se sienten más satisfechos con su trabajo si están involucrados y continuar desarrollando estas habilidades en un ambiente de aprendizaje que permiten la transferencia de competencias en el sistema de software biomédico para el lugar de trabajo y tal vez el desarrollo de habilidades para la vida.

Por lo tanto, el uso del sistema de software biomédico para la enseñanza en un entorno 3-D en los estudiantes de la salud puede ser cuestionable hasta que se realicen más investigaciones, dos de los principales retos relacionados con el sistema de software biomédico en la educación es el tiempo invertido en la creación de espacios de aprendizaje en un entorno virtual, así como los costos involucrados. El sistema de software biomédico se reproduce en tiempo real, el encanto de un entorno colorido dinámico puede distraer la atención del estudiante de los objetivos de aprendizaje. Por supuesto, estos son todos los puntos de vista que pueden ser debatidos.

Es evidente que se necesita investigación empírica para el uso futuro del sistema de software biomédico en la formación médica y la educación general. Los desafíos que enfrenta el sistema de software biomédico con fines educativos son titularidad de trabajo colaborativo y la certificación de la autoría. Estas dos cuestiones representan un problema para la evaluación de los resultados del aprendizaje. La transformación actual de los mundos virtuales es sensible al tiempo, y la interactividad es la clave para el aprendizaje. La experiencia se construye activamente a través de la interacción directa con el mundo, y que, de hecho, el conocimiento es la experiencia". El uso de la Realidad Virtual (VR) como herramienta educativa de los estudiantes para el conocimiento de las estructuras anatómicas antes y después de una experiencia de VR, se indica que la provisión de simulaciones en 3-D de entornos virtuales de aprendizaje presenta el potencial de mejorar el aprendizaje.

GLOSARIO

Bitmap: Un bitmap es una imagen producida por una matriz fija de píxeles coloreados, como un mosaico. Se puede usar los bitmaps como las texturas para los materiales.

Booleano: Combina dos objetos, realizando una operación Booleana sobre ellos. Los dos objetos originales son los operandos y el objeto Booleano es el resultado de la operación.

CAD: Computer-Aided Design (Diseño Asistido por Computadora).

Cámara: Es uno de los nodos básicos del entorno en 3D. La cámara se considera como la vista del usuario en una escena en 3D. Sin una cámara, el usuario no puede ver el mundo en 3D.

Canal Alfa: Hace referencia a la información almacenada dentro de un mapa de bits cuyo objetivo es controlar la cantidad de transparencia por pixel. A los canales alfa se hace referencia utilizando su profundidad de bits, que describe cuantos niveles de transparencia son capaces de admitir. El canal alfa utilizado en nuestras imágenes fue de 8 bits de profundidad y proporciona 256 niveles de transparencia.

Coordenadas de Mapeo: Las coordenadas de mapeo especifican la colocación, orientación y escala de un mapa en la geometría. Si se aplica un material mapeado a un objeto que no tiene ninguna coordenada de mapeo, el render asigna las coordenadas de mapeo predefinidas. Las coordenadas de mapeo incorporadas se diseñan para cada tipo objeto. Las coordenadas de mapeo caja colocan un mapa doble en cada uno de sus seis lados. Para el cilindro, la imagen se envuelve una vez alrededor de sus lados. Una esfera tiene la imagen envuelta una vez alrededor de la esfera.

Coordenadas del Mundo: Cuando calculamos situaciones en un mundo en 3D, tendremos que medir desde un punto de partida. Si este punto de partida no es el origen del mundo en 3D, tal como un nodo, las medidas que estamos tomando se encuentran en coordenadas locales.

Detección de Colisión: Un proceso que exige mucha memoria y que comprueba si los modelos están o no intersectándose.

Escalamiento: Uno de los tres tipos principales de transformaciones utilizado para aumentar o disminuir el tamaño de la geometría.

Formato de archivo EXE: Formato de un archivo correspondiente a una aplicación o programa ejecutable.

Formato de archivo JPG: Formato de imagen comprimida, por lo que no ocupa mucho espacio en el disco y por lo tanto se descarga con mayor velocidad.

Formato de archivo MAX: Es un formato de archivo nativo de 3D Studio Max, el cuál sirve para almacenar modelos tridimensionales, luces, cámaras, ayudantes, animaciones, etc.

Formato de archivo PNG (Portable Network Graphics): Es un formato de archivo de imagen desarrollado para el uso con la Internet y Web. Con la característica principal que cuenta con 8 bits del canal alfa para poder asignarle un canal de transparencia a la respectiva imagen.

Gizmo: Muchos modificadores desplegados en los puertos de vista como una caja que inicialmente rodea al objeto seleccionado llamado gizmo. Un gizmo actúa como un tipo de recipiente que transfiere la modificación hacia el objeto al cual está ligado. Se puede mover, escalar y rotar un gizmo como se haría sobre cualquier objeto, alterando el efecto del modificador en el objeto.

HMD. Head Mounted Display. Interfase de los sistemas de RV que genera la inmersión del usuario. Consiste en un casco con pantallas integradas.

Luz Ambiental. Se utiliza para aproximarse a un nivel general de la luz en un entorno. Concretamente, el uso de la luz ambiental es un intento de aproximarse a la cantidad de luz reflectada dentro de una escena.

Materiales. Es un conjunto de datos que se asignan a la superficie o caras de un objeto para que este aparezcan algo cuando son renderizados. Los materiales afectan el color de los objetos, su brillo y su opacidad. Un material estándar contiene una componente ambiental, difusa, y especular. Se pueden asignar mapas a las distintas componentes de un material estándar.

Materiales Compuestos. Los materiales compuestos permiten crear un material compuesto o integrado por dos o más submateriales. El material compuesto Multi/subobjeto permite asignar los diferentes materiales a los subobjetos diferentes, al nivel del subobjeto de su geometría.

Malla (Mesh). Es un tipo especial de recurso aplicado a los modelos que permite construir y deformar la geometría en sus distintos niveles; vértice, arista, cara, polígono y objeto.

Mapas. Las imágenes que se asignan a los materiales son llamadas mapas. Los programas de software de imágenes proveen varios tipos de mapas diferentes. Ellos incluyen bitmaps estándares (como archivos .bmp, .jpg, o .tga), mapas procesales, como el Checker o Marble, y sistemas de procesamiento de imágenes como composiciones y sistemas de enmascarado.

Mip – Mapping. Proceso que crea varias versiones sucesivamente más pequeñas de una textura. Esto permite al render en 3D elegir una versión de la textura que tenga el tamaño adecuado, dado que el tamaño de la textura varía dependiendo de su ubicación en el entorno en 3D.

Modelo. Es uno de los nodos principales de un entorno en 3D. Los modelos se utilizan para representar los objetos de una escena.

Objeto Tiempo de Espera. Objeto especial de código que actúa como temporizador. Se utiliza para disparar un evento determinado una vez que ha transcurrido una cierta cantidad de tiempo.

Pixel. Es el elemento más pequeño de una imagen, es un solo punto en una imagen gráfica. Los monitores gráficos despliegan imágenes dividiendo la pantalla en miles (o millones) de pixeles, arreglados en filas y columnas.

Plug-In. Es una característica o funcionalidad proporcionada para un programa independiente o componente, es decir, extiende la funcionalidad de un programa. Estos pueden ser proporcionados por distintos medios, por ejemplo por diseñadores de software independientes.

Primitivos. Son los cinco tipos de recursos básicos incorporados tanto a los modeladores tridimensionales como a Lingo, concretamente, estos cinco primitivos son: cajas, cilindros, esferas, planos y sistemas de partículas.

Profundidad de Bits. Es un término utilizado para hacer referencia al número de colores disponibles para un monitor, un mapa de bits o un programa.

Renderización (Render). Es un proceso de representación de los entornos en 3D, donde se observan los modelos geométricos texturizados, con iluminación, etcétera, vistos desde el punto de vista de la cámara.

Rotación. Es uno de los tres tipos de transformación. Es el proceso de girar los nodos de un elemento geométrico respecto a otro nodo dentro del entorno 3D.

RV. Abreviatura de Realidad Virtual.

Shader. Es un elemento de los entornos en 3D que se utiliza para definir las cualidades de la superficie, como su color o su brillo.

SubObject. El subobjeto es un subconjunto de la geometría del un objeto. Muchos objetos tienen varios tipos de subobjetos que se pueden trabajar independientemente. Por ejemplo, un subobjeto de un objeto malla editable son vértices, bordes, caras, polígonos, y elementos. Para acceder a los subobjetos, hay que acceder por medio del panel Modificar. En el nivel subobjeto se puede seleccionar los subobjetos, transformar las selecciones, aplicar modificadores, y así sucesivamente.

Textura. Elemento de los entornos en 3D que se utiliza en combinación con un shader para definir las cualidades materiales de un modelo.

Transformación. Término que se utiliza para hacer referencia al proceso que orienta, coloca y escala los nodos en el entorno en 3D. Cuando se crea cualquier objeto, se graba la información de su posición, rotación, y escala en una tabla interna llamada matriz de transformación. Subsecuentemente los ajustes de posición, rotación, y escalamiento son llamadas transformaciones. Cada transformación geométrica tiene como referencia un sistema de coordenadas ubicado en cualquier elemento o región geométrica, así como en el sistema de coordenadas predefinido. Un objeto puede tener solamente una transformación geométrica en un mismo instante de tiempo.

Traslación. Es uno de los tres tipos de transformación. Es el proceso de posicionar los nodos dentro del entorno 3D.

Visión estereoscópica. Es la visión binocular (dos ojos) que produce la sensación de una imagen en tres dimensiones, al ser procesadas por el cerebro, a la vez, las dos imágenes que captan las retinas oculares. Puesto que los campos de visión están superpuestos en gran parte para obtener este efecto, nada más el área superpuesta permite la visión tridimensional.

UVW Mapping (Mapeo UVW). Es el proceso de aplicar un shader a un modelo. Los shaders se aplican a los modelos en el espacio en 3D relativo a estos. Para evitar la confusión entre las coordenadas de la aplicación de mapas y las coordenadas normales, se utilizan las letras UVW, aunque técnicamente se trata de las coordenadas XYZ. La mayoría de los mapas de materiales son planos 2D asignados a una superficie 3D. Por consiguiente, el sistema de coordenadas usado para describir la colocación y transformación de los mapas es diferente de los ejes X, Y, y Z coordenadas usadas en el espacio 3D. La coordenada de profundidad W para un plano 2D, se utiliza para poder voltear la orientación de un mapa, relativo a su geometría. La coordenada de W también tiene un significado para los materiales de 3 dimensiones.

BIBLIOGRAFÍA

- [1] Michael Ackerman, “**The Visible Human Project**,” Slate, 1986, <http://www.nlm.nih.gov/research/visible/visible_human.html> (November 1995). <http://www.cs.umd.edu/hcil/visible-human/vhe.shtml>.
- [2] Gilbert St. “**The Effects of Computer-Assisted Instruction in Teaching Human Anatomy**” Slate, Iowa City, 2010, <http://cyber-anatomy.com/product_CAHA_med.php> (April 2010). http://www.cyberscience3d.com/pdf/NSF_Study_by_Tallitsch.pdf.
- [3] W. Dean Bidgood, Jr., MD, MS, Steven C. Horii, MD, [...], and Donald E. Van Syckle, “**Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging**”, USA, 1997, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC61235/>> (May-Jun 1997).
- [4] Nicola H Strickland, London, 2000, “**PACS (Picture Archiving and communication Systems): Filmless Radiology**”, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1718393/>> (July 2000).
- [5] John McGhee, Great Britain and Ireland, 2010, “**3D Visualization and Animation Technologies in Anatomical Imaging**”, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2815947/#!po=43.7500>> (November 2009).
- [6] Juan Carlos Parra Márquez, Rodrigo García Alvarado, Iván Santelices Malfanti, [Practical introduction to virtual reality], Universidad Del Bío-Bío, Concepcion (Chile), pg127 (2001).
- [7] Grigore C. Burdea, Philippe Coiffet, [Virtual Reality Technology], John Wiley and Sons, New York (United States of American), pg444 (1993).
- [8] Watt Alan, [3D Computer Graphics], Addison-Wesley, University of California (United States of American), pg570 (2000).
- [9] SOMMER VILLE Ian, [Ingeniería de Software], Addison-Wesley Iberoamericana, Estado de México (México), pg792 (2011).
- [10] Rory McCloy, Robert Stone, University of Manchester, “**Virtual Reality in Surgery**”, Manchester England, 2001, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1121442/>> (October 2001).
- [11] Issac Trotts, Shawn Mikula, Edward G.Jones, “**Interactive Visualization of Multiresolution Image Stacks in 3D**” Slate, Sullivan & Baker, 1994, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2492583/>> (January 2007).
- [12] Grigore C. Burdea, Philippe Coiffet, [Virtual Reality Technology], John Wiley and Sons, New York (United States of American), pg444 (1993).
- [13] Francesc Tarrés Ruis, [Sistemas Audiovisuales I. Televisión Analógica y Digital], Universitat Politècnica de Catalunya, Catalunya (España), pg454 (2004).
- [14] Maria Kozhevnikov, Ruoli P. Dhond, “**Understanding Immersivity:Image Generation and Transformation Processes in 3D Immersive Environments**”, Harvard Medical School, Charlestown, MA, USA, 2012, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3415688/>> (July 2012).
- [15] Maged N Kamel Boulos, Larry R Robinson, “**Web GIS in practice VII: stereoscopic 3D solutions for online maps and virtual globes**”, Faculty of Health, University of Plymouth Wisconsin USA, 2009, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2772849/>> (2009 Oct).
- [16] Reviewed by Scott Erdley, Rod Ward, and Bill Perry, “**Versatile, Immersive, Creative and Dynamic Virtual 3-D Healthcare Learning Environments: A Review of the Literature**”, School of Nursing, University of San Francisco, San Francisco, CA, USA, 2008, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2626432/>> (2008 Jul-Sep).
- [17] David Ballejo Fernández & Cleto Martín Angelina, [Desarrollo de Videojuegos 1: Arquitectura del Motor de Videojuegos], Universidad de Castilla la Mancha, pg314 (2013).
- [18] Renata Kopach-Konrad, MSc, Mark Lawley, PhD, Mike Criswell, “**Applying Systems Engineering Principles in Improving Health Care Delivery**”, School of Biomedical Engineering, Regenstrief Center for Healthcare Engineering, Purdue University, West Lafayette, IN USA, 2007, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2150611/#!po=10.8696>> (2007 Dec).
- [19] Huff, C. C. “**Elements of a realistic CASE tool adoption budget. Comm. ACM**”, 35 (4), 45-54. <<http://www.SoftwareEngineering-9.com/Web/CASE/>> (1992).
- [20] Richard S. Snell, [Neuro-Anatomía Clínica], George Washington University School of Medicine and Health Science, Washington D.C (USA), pg 636 (2001).
- [21] Travessera de Gràcia, [Diccionario Terminológico de Ciencias médicas], Barcelona (España), pg 1319 (2004).

ANEXO I:

HERRAMIENTAS TECNOLÓGICAS PARA EL ANÁLISIS Y DISEÑO

Sparx Systems Enterprise Architect

Es una herramienta de modelado visual y diseño basado en el OMG UML. La plataforma soporta: el diseño y construcción de sistemas de software, modelado de procesos de negocio, y la industria de modelado de dominios. Es utilizado por las empresas y organizaciones que modelan la arquitectura de sus sistemas, para procesar la aplicación de estos modelos a través del ciclo de vida de desarrollo de aplicaciones. Sistemas de modelado con UML que proporciona una base para el modelado de todos los aspectos de la arquitectura de la organización, junto con la capacidad de proporcionar una base para el diseño e implementación de nuevos sistemas o cambiar los sistemas existentes.

Los aspectos que pueden ser cubiertos por este tipo de gama de modelos para trazar las arquitecturas organizativas o sistemas, el proceso de reingeniería de negocios, análisis de negocio y arquitecturas orientadas a servicios y modelos web, a través de la aplicación y diseño de base de datos y re-ingeniería, y desarrollo de sistemas embebidos. Junto con el modelado de sistemas, Enterprise Architect cubre los aspectos básicos del ciclo de vida de desarrollo de aplicaciones, desde la gestión de requisitos hasta el diseño, las fases de construcción, prueba y mantenimiento, con el apoyo de la trazabilidad, gestión de proyectos y control de cambios de estos procesos, así como, instalaciones para el desarrollo dirigido por modelos de código de la aplicación utilizando una plataforma de desarrollo integrado interno.

La base de usuarios oscila entre los programadores y analistas de negocio a través de arquitectos de la empresa, en organizaciones que van desde empresas pequeñas para desarrolladores, corporaciones multinacionales y organizaciones gubernamentales a través de los organismos internacionales de estándares de la industria.

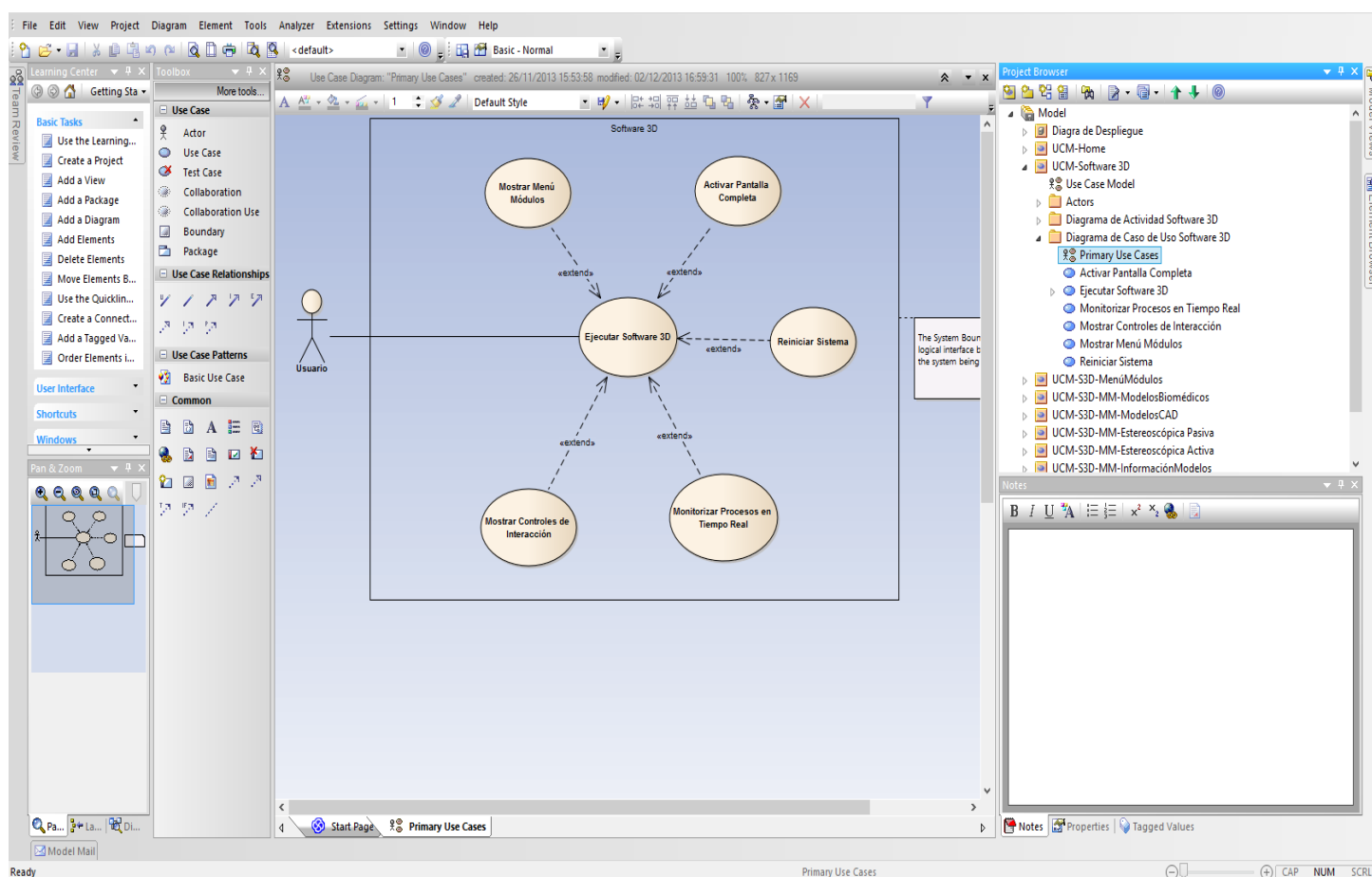


Figura AI.1 Sparx Systems Enterprise Architect.

Balsamiq Mockups

Es una herramienta que nos permite realizar Wireframes para webs fácilmente. Un Wireframe (aplicado a la web) es una representación esquemática de la solución que llevaremos adelante, sin entrar en etapas posteriores como el diseño gráfico o la programación web. Nos permite acordar con el cliente aspectos claves de la solución a desarrollar, como la distribución general de los elementos, sus jerarquías y la navegación de los mismos. Balsamiq Mockups nos provee de representaciones de todos los elementos utilizados para la construcción de una web, como pantallas de navegadores, títulos, menús, imágenes, videos, etc. Haciendo uso de ellos, sólo debemos organizarlos en un documento y ya podemos tener una primera aproximación de la solución a desarrollar. Esta es una herramienta que puede ser usada tanto por clientes como por desarrolladores. Los clientes pueden hacer uso de ello sin tener ningún tipo de conocimiento técnico especial. Gracias a ello, pueden comunicar de una manera más eficiente sus ideas y necesidades al grupo de trabajo que realiza los desarrollos técnicos. Los desarrolladores pueden usarlo con el mismo propósito, pero al revés. Para comunicar rápidamente las propuestas de solución, sin invertir demasiada cantidad de tiempo en esta primera etapa.

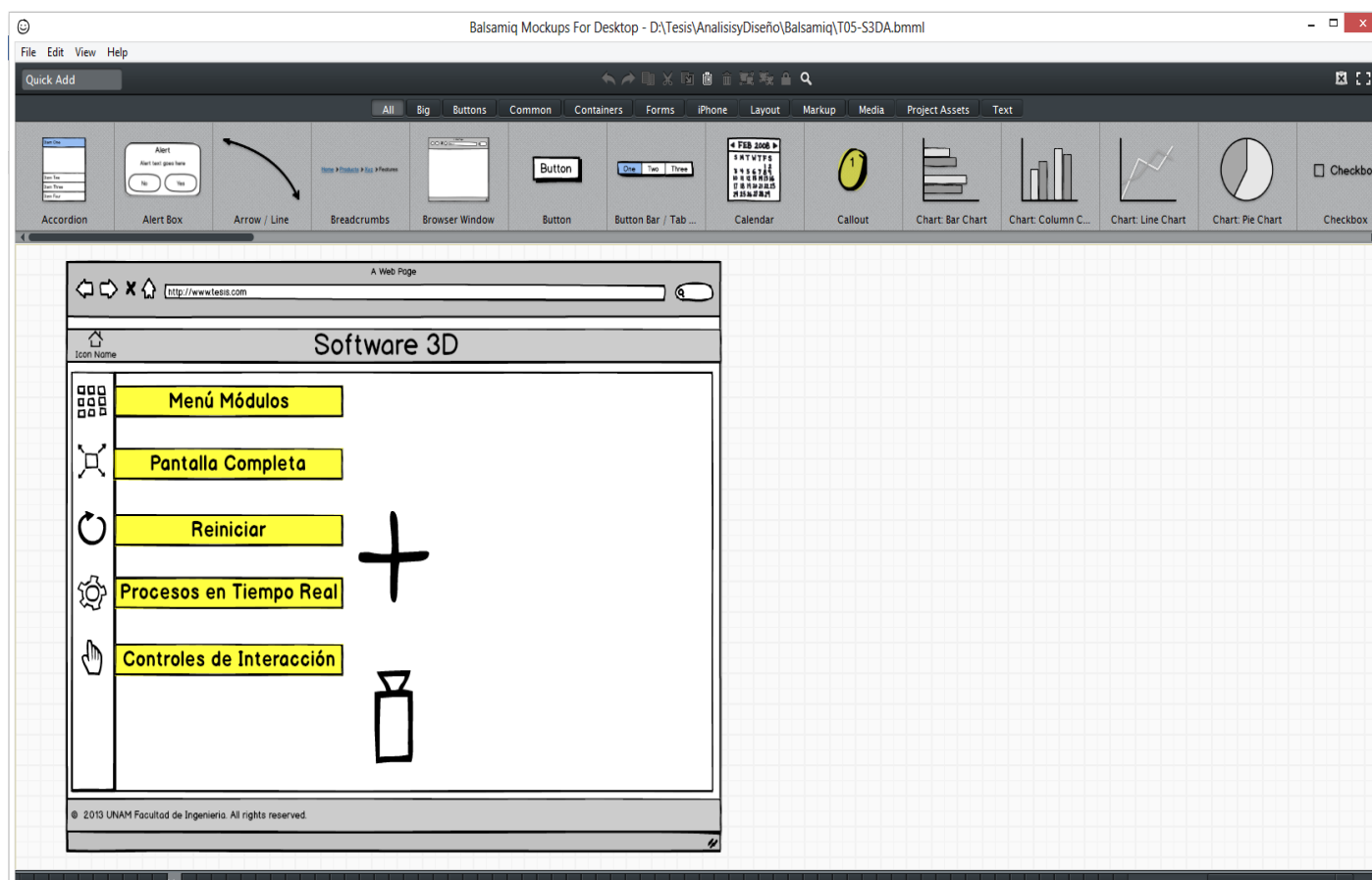


Figura AI.2 Balsamiq Mockups.

Axure RP (Interactive Wireframe Software & Mockup Tool)

Es una herramienta orientada a diseñar wireframes, interfaces graficas de usuario y prototipos básicos o avanzados de forma fácil, incluso con aspecto de sketch y en la medida de los conocimientos y experiencia del proceso de la arquitectura de información, incluye funciones para simular de una mejor forma las interacciones propuestas en el proyecto. Estas son las fortalezas: la interactividad que simula el comportamiento de las interfaces de usuario gracias a su fortaleza como herramienta de prototipado y lo mejor de todo es que los prototipos funcionan en cualquier navegador. Otra de sus fortalezas tiene que ver con los procesos de documentación. El arquitecto de información puede exportar el prototipo en HTML (ideal para documentar funcionamiento y para realizar pruebas de usuario), o en un documento Word que recoge, además de las interfaces diseñadas, todas las anotaciones que se hagan sobre el funcionamiento de cada página, módulo o componente. Esta herramienta tiene por defecto una amplia gama de widgets con los cuales se pueden diseñar fácil y rápidamente interfaces para proyectos de sitios web o de aplicaciones móviles. Igualmente el arquitecto de información puede crear otros widgets personalizados y reutilizables, los cuales pueden simular funcionalidades o módulos que se pueden necesitar para describir mejor un proceso, interacción o contenido. Además, gracias a que la comunidad de diseñadores UX y arquitectos de información que usan Axure cada vez es más grande, se pueden encontrar en internet diversas librerías de gadgets, módulos, íconos, grillas y demás elementos de diseño que le pueden aportar muchos elementos a tu propio proyecto.

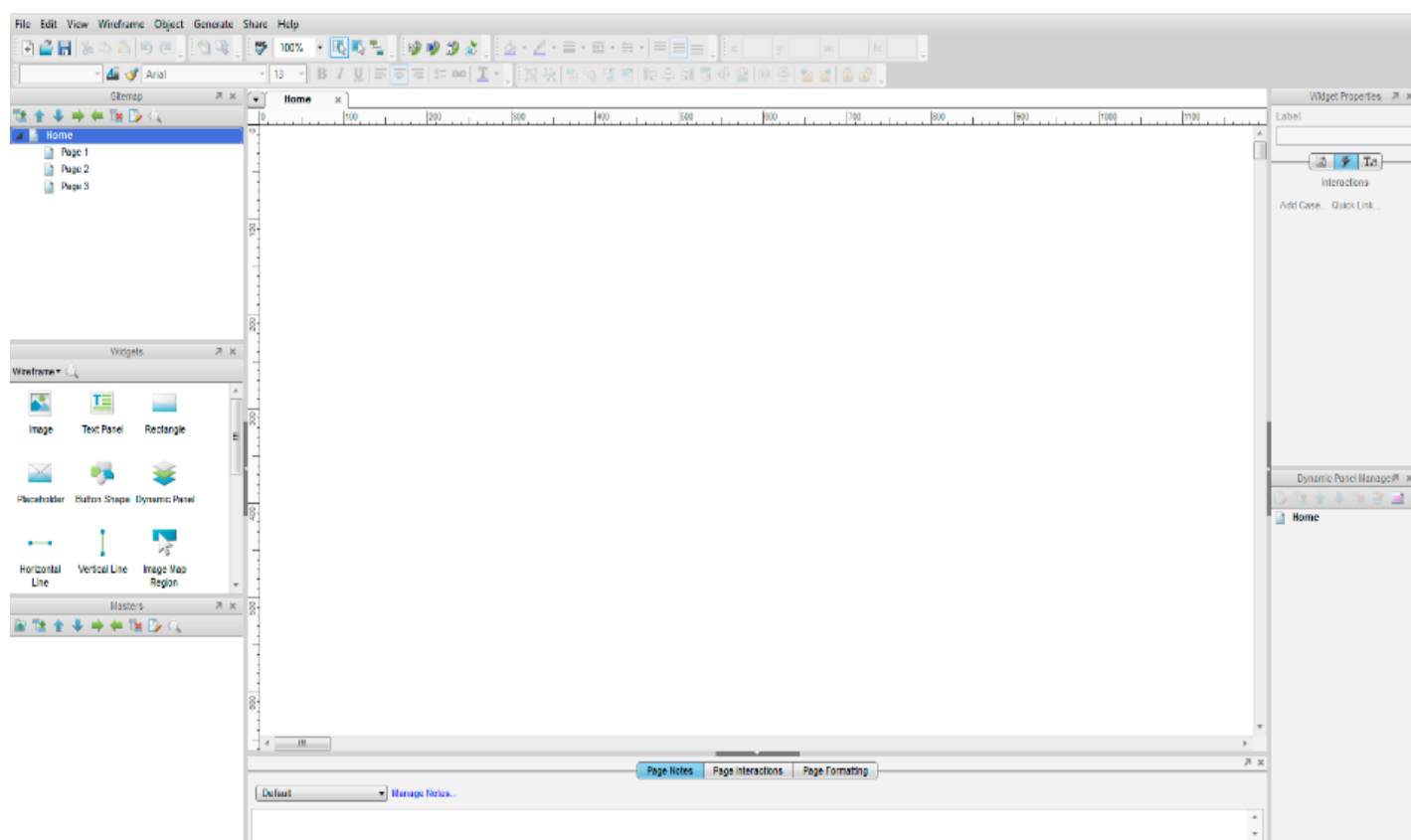


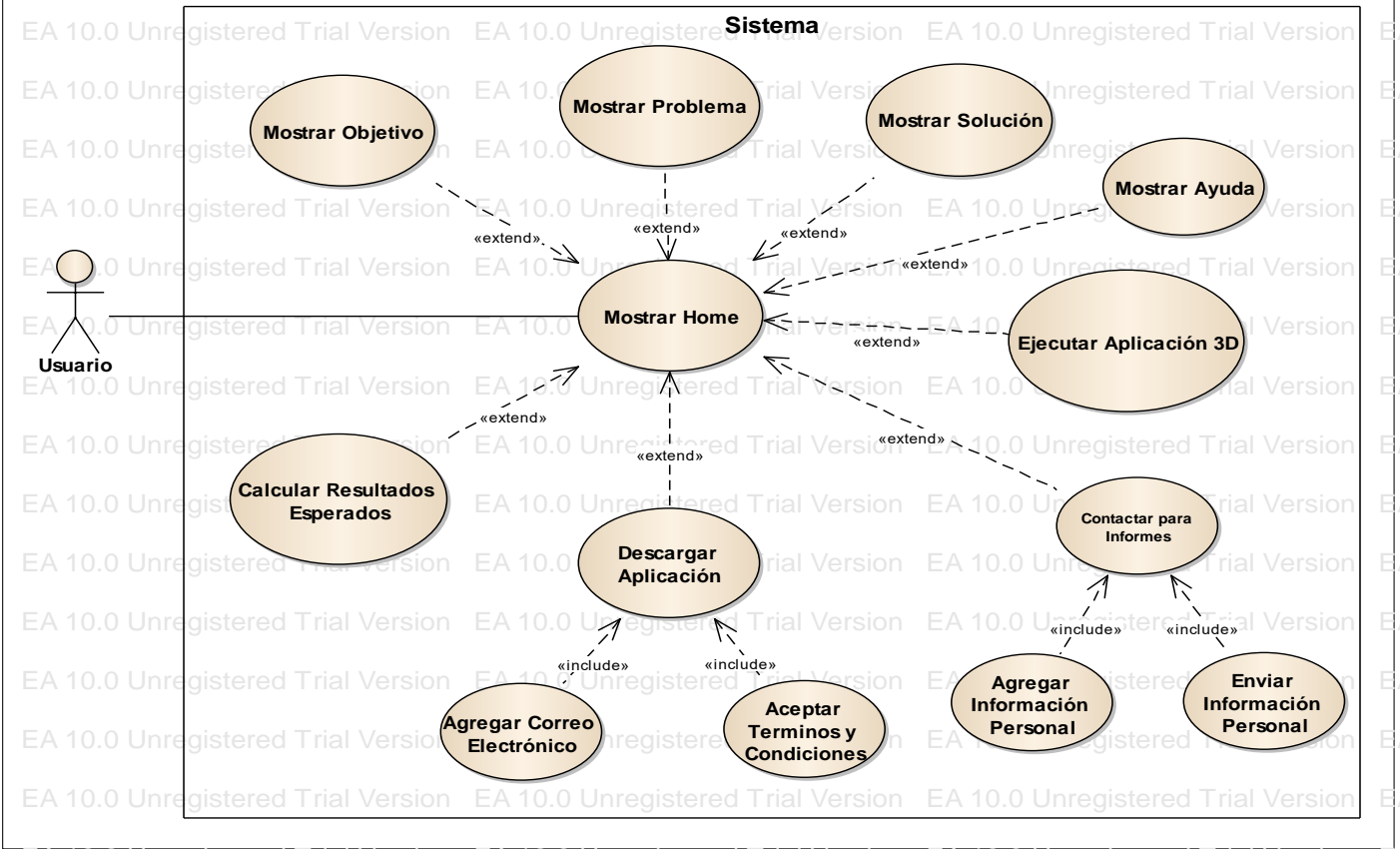
Figura AI.3 Axure RP (Interactive Wireframe Software & Mockup Tool).

Las cuatro actividades básicas como el proceso de especificación, desarrollo, validación y evolución se organizan de diversa manera en diferentes procesos de desarrollo. En el modelo en cascada se organizan en secuencia, mientras que se entrelazan en el desarrollo incremental. La forma en que se llevan a cabo estas actividades depende del tipo de software, del personal y de la inclusión de estructuras organizativas. En la programación extrema, por ejemplo, las especificaciones se escriben en tarjetas. Las pruebas son ejecutables y se desarrollan antes del programa en sí. La evolución incluye la reestructuración o refactorización sustancial del sistema. [9]

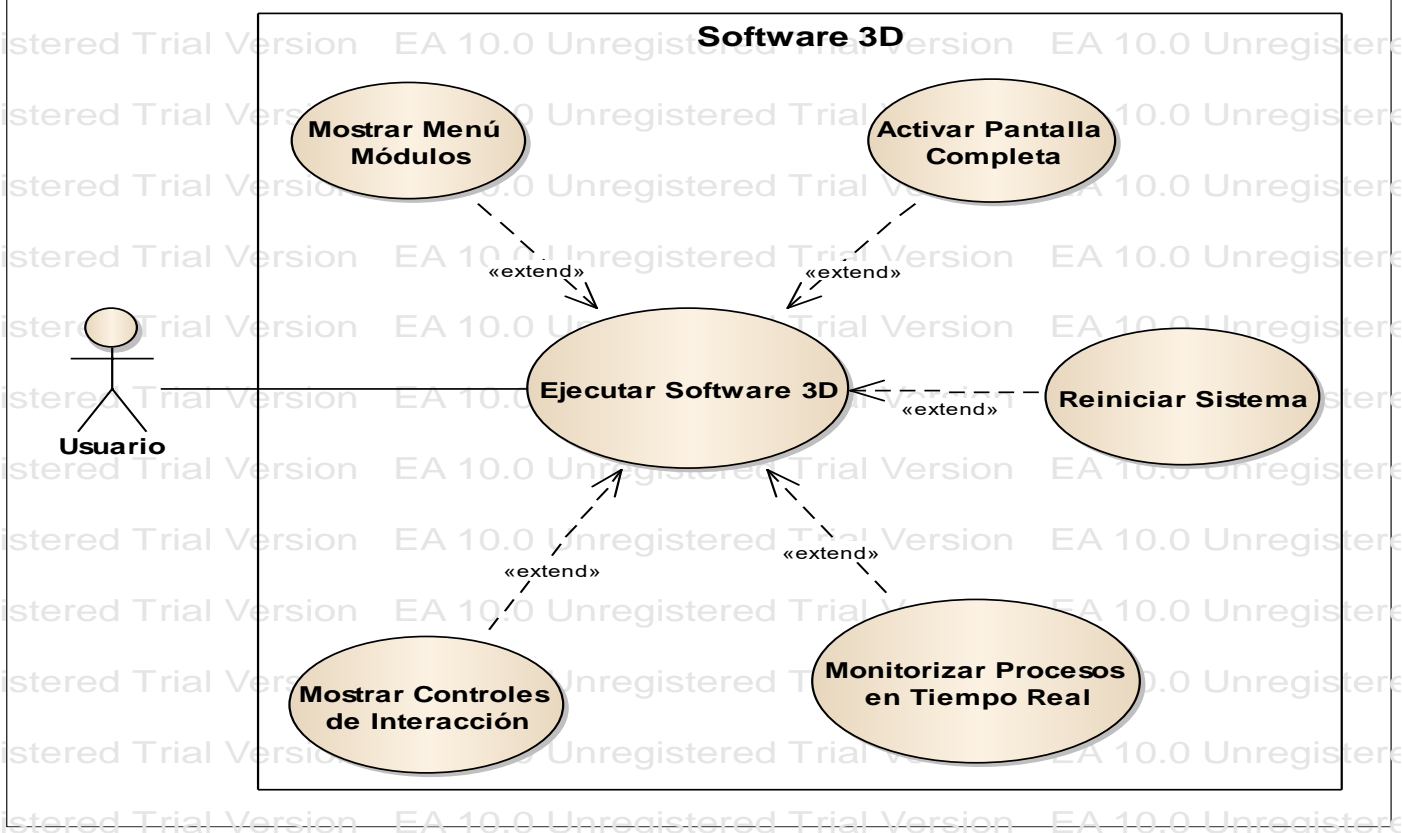
ANEXO II:

DIAGRAMA CASOS DE USO

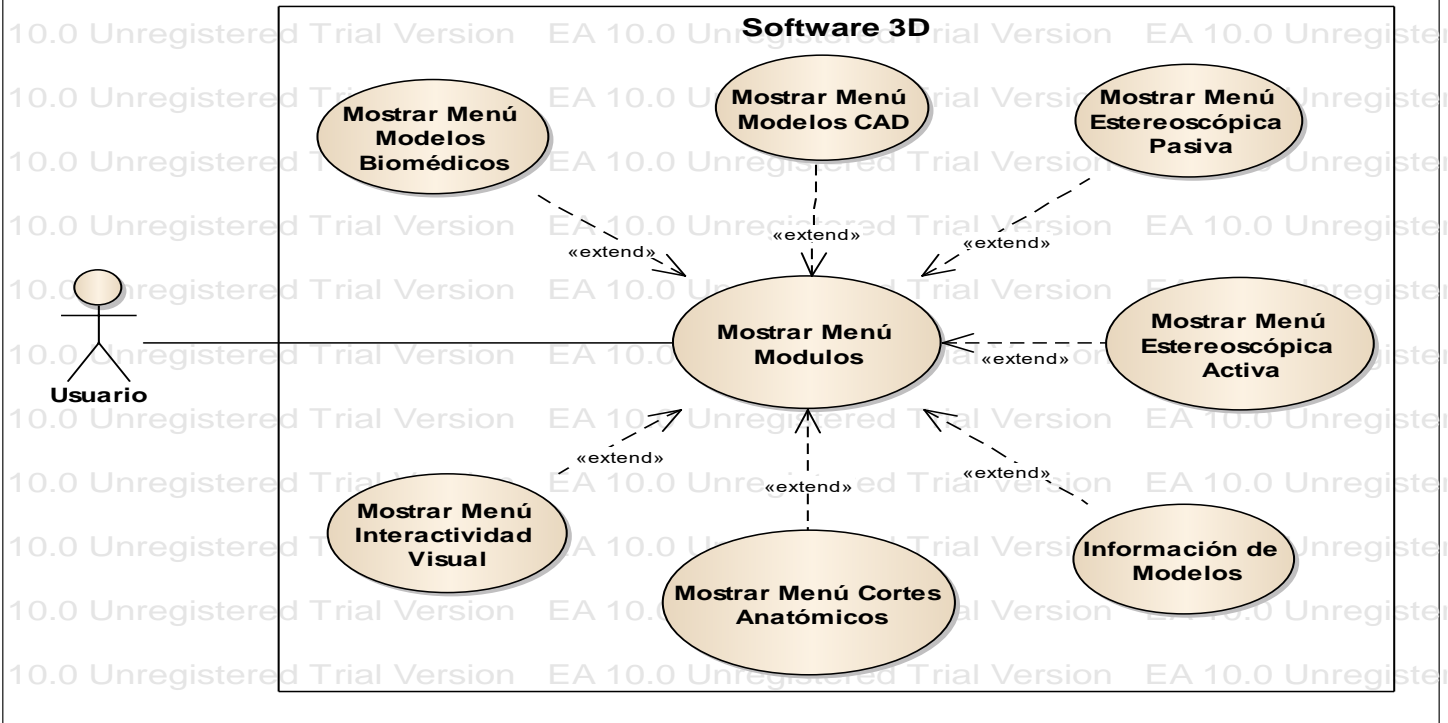
uc Diagrama de Caso de Uso Home



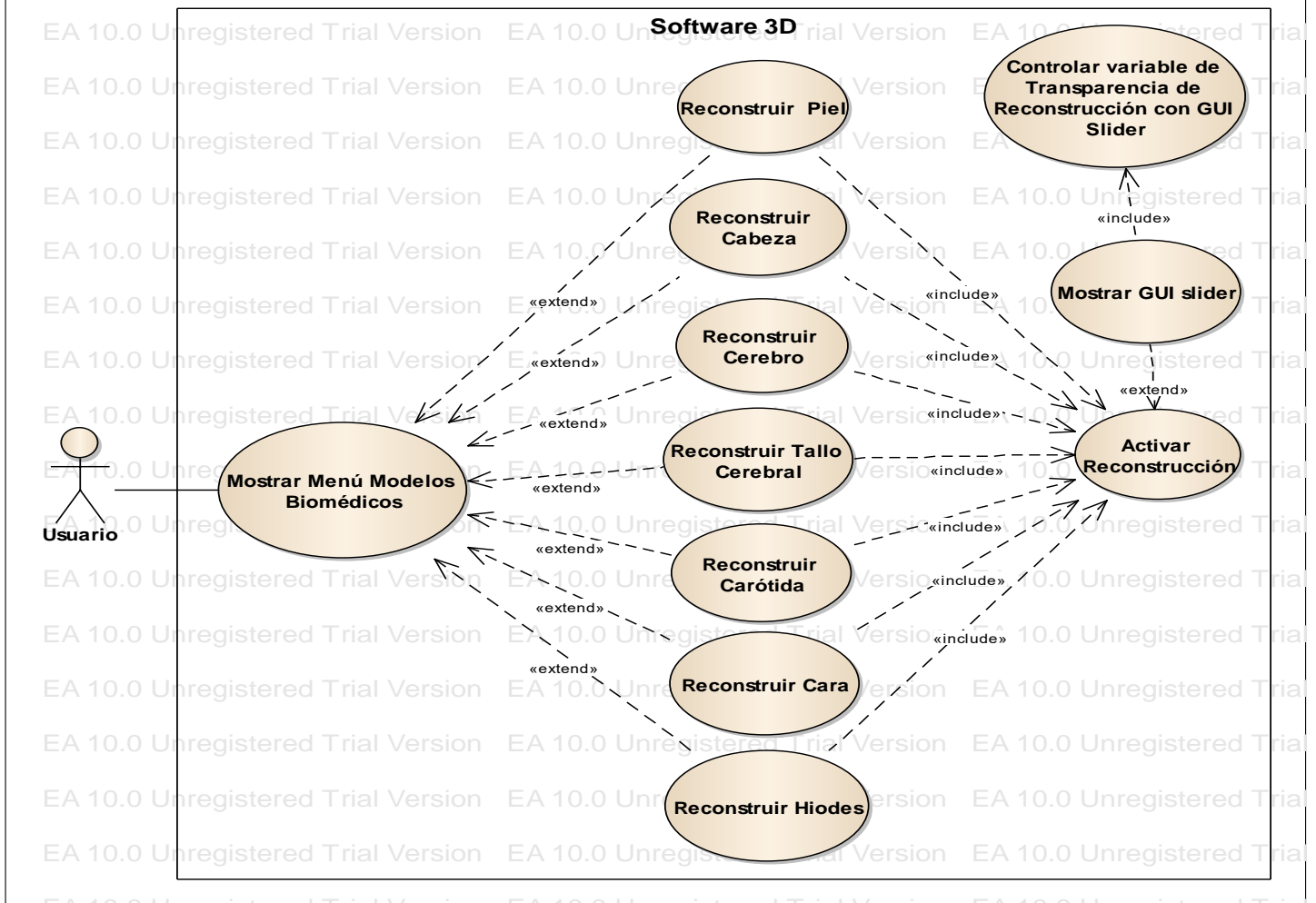
uc Diagrama de Caso de Uso Software 3D



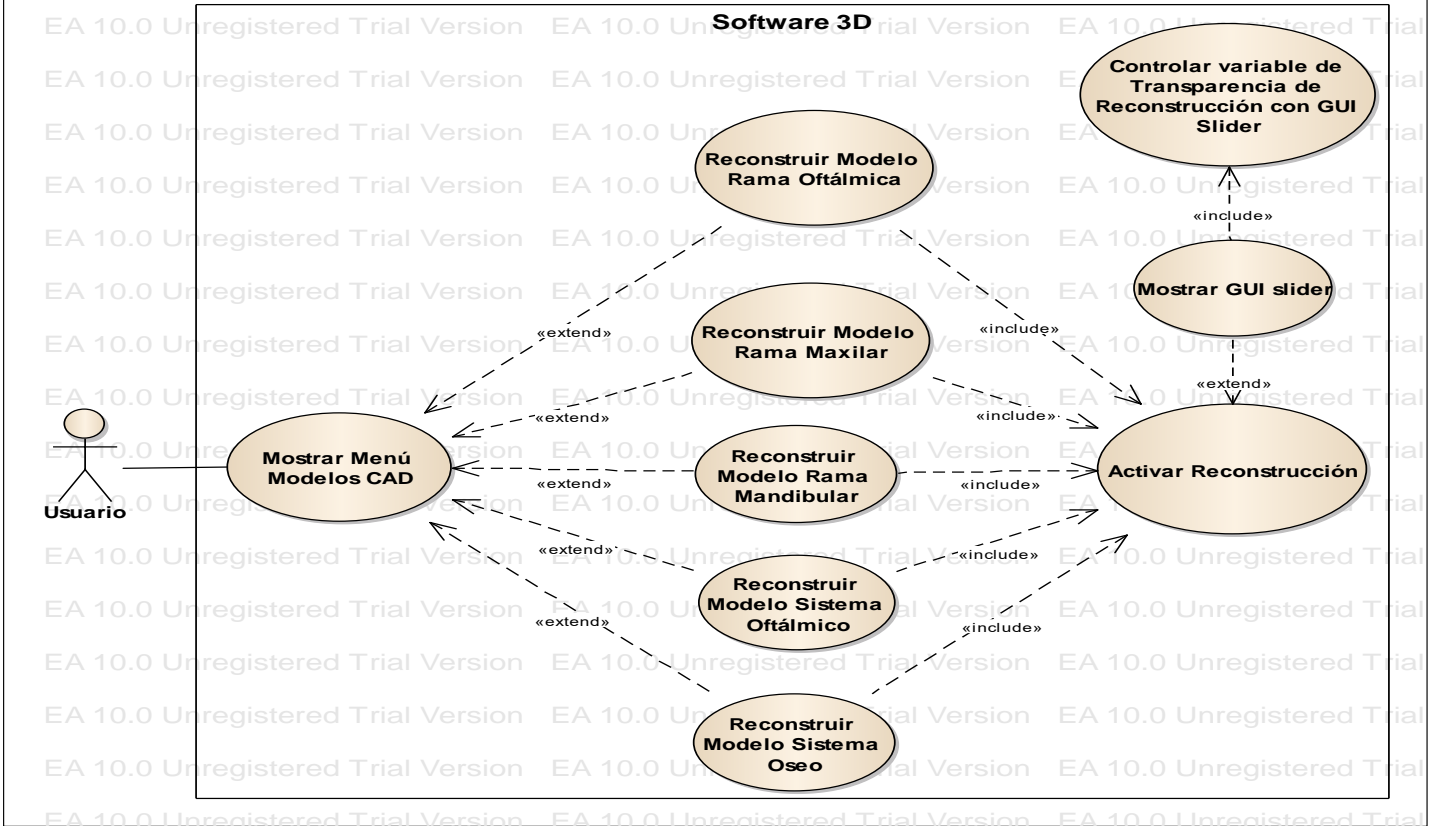
uc Diagrama de Caso de Uso Menú Módulos



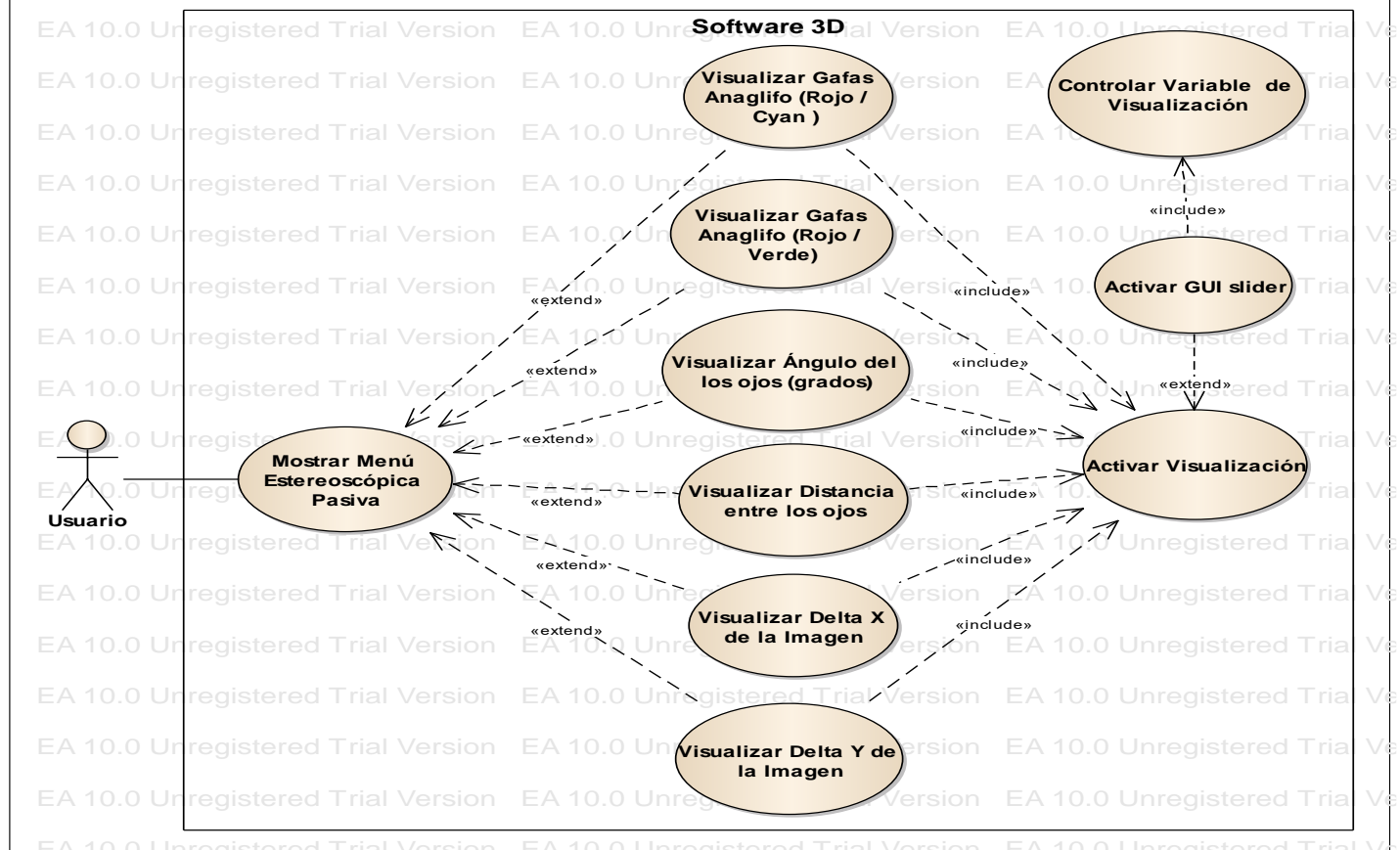
uc Diagrama de Caso de Uso Modelos Biomédicos



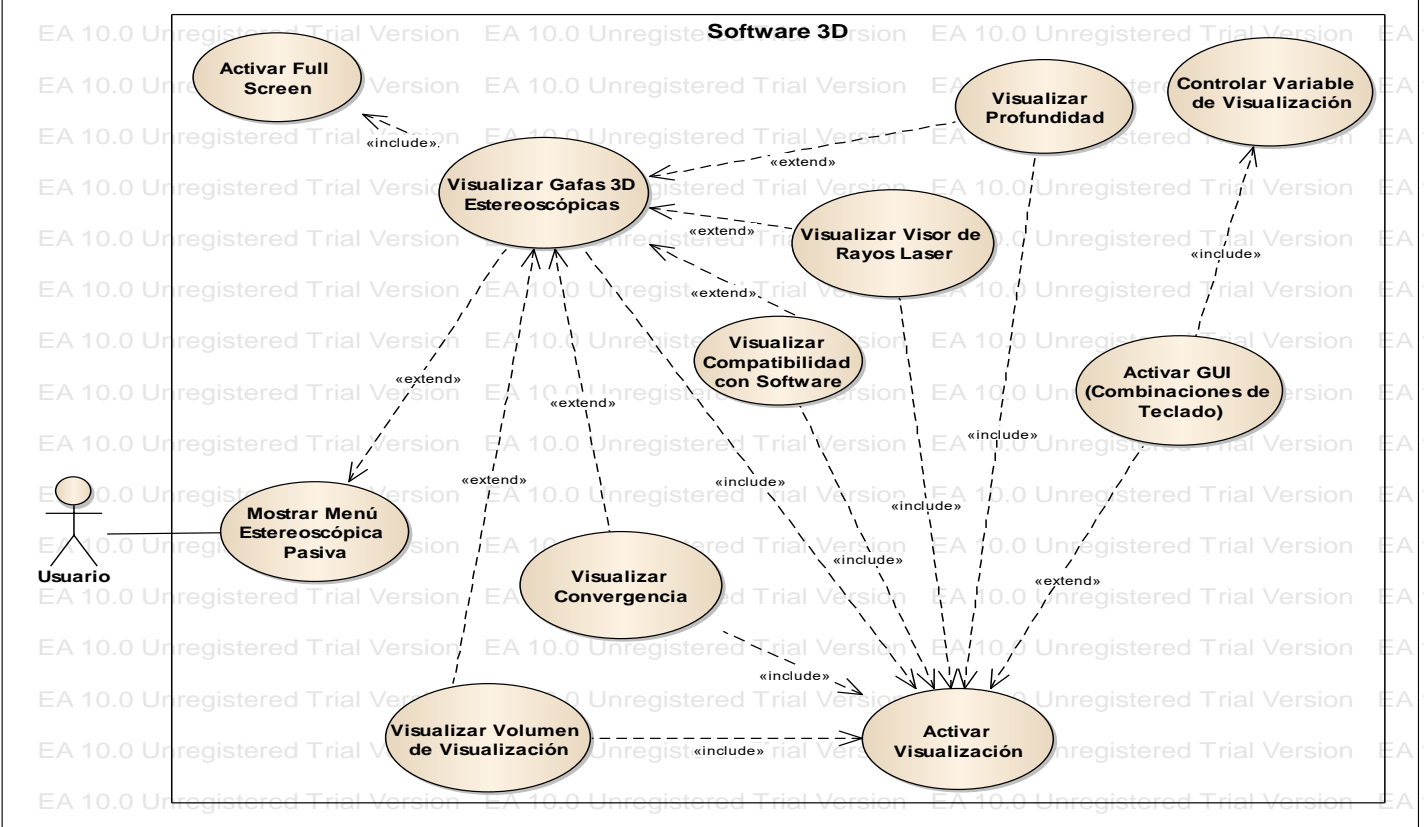
uc Diagramas de Caso de Uso CAD



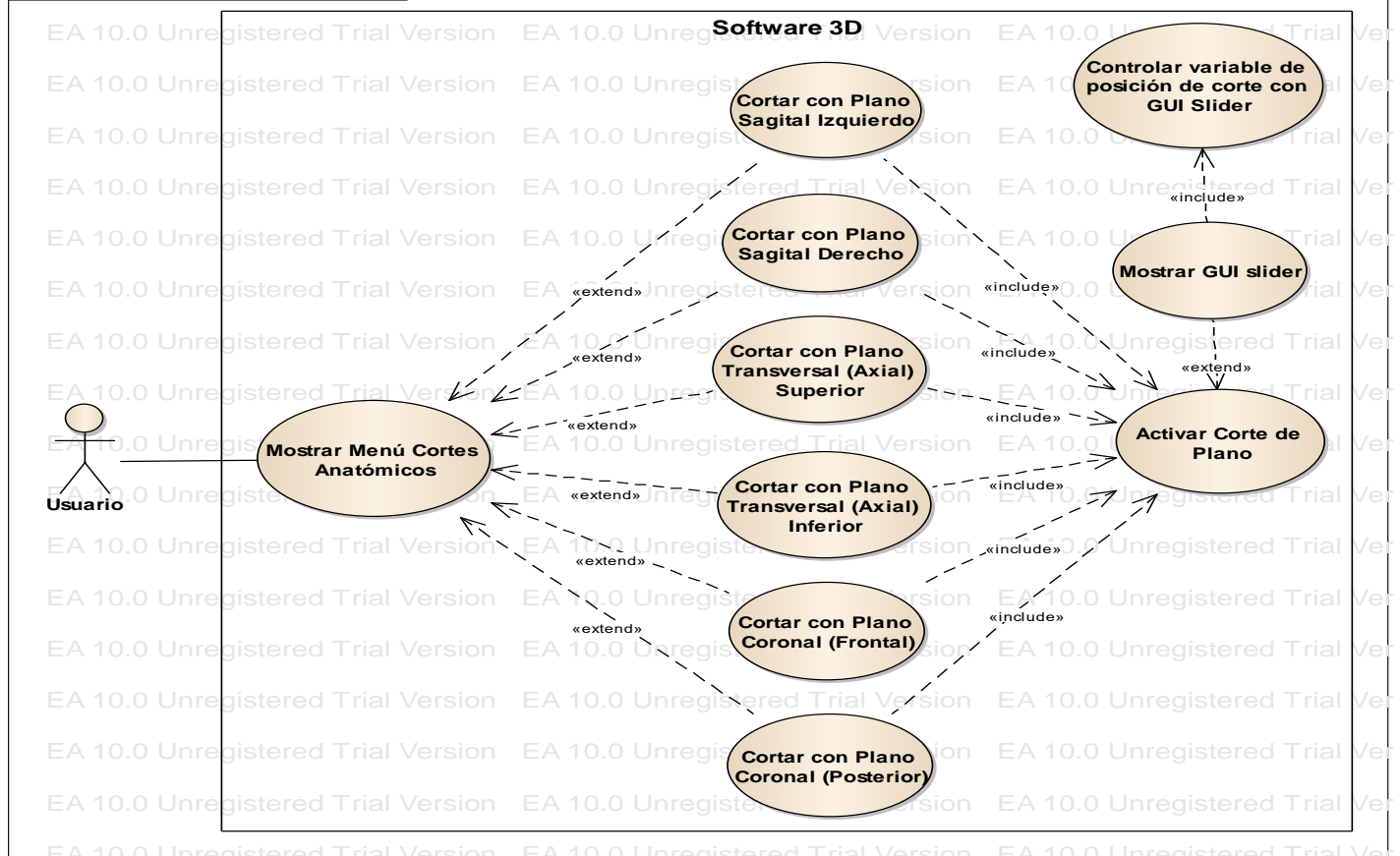
uc Diagrama de Caso de Uso Estereoscópica Pasiva



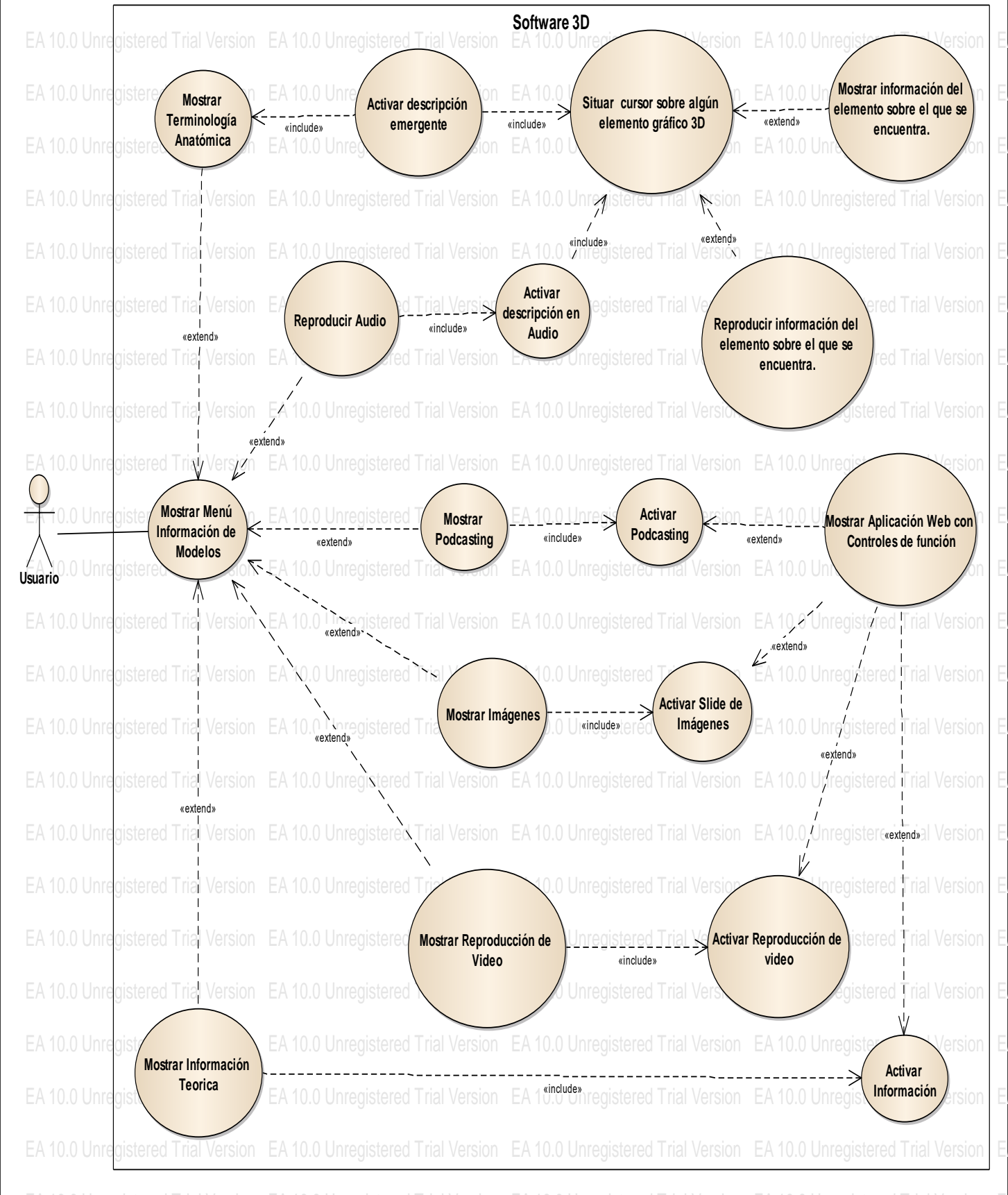
uc Diagrama de Caso de Uso Estereoscópica Activa



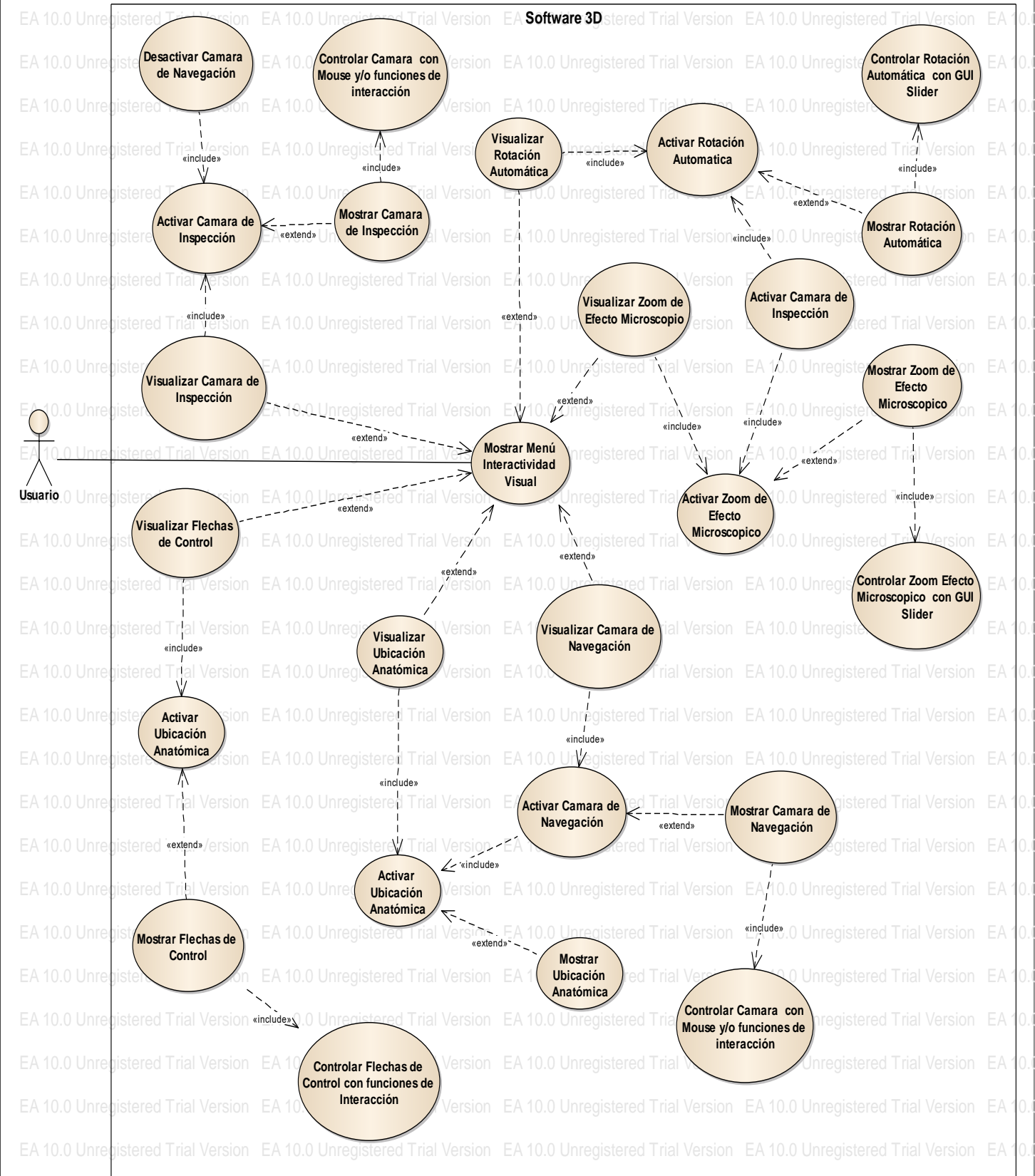
uc Diagrama de Caso de Uso Cortes Anatómicos



uc Diagrama de Caso de Uso Información de Modelos



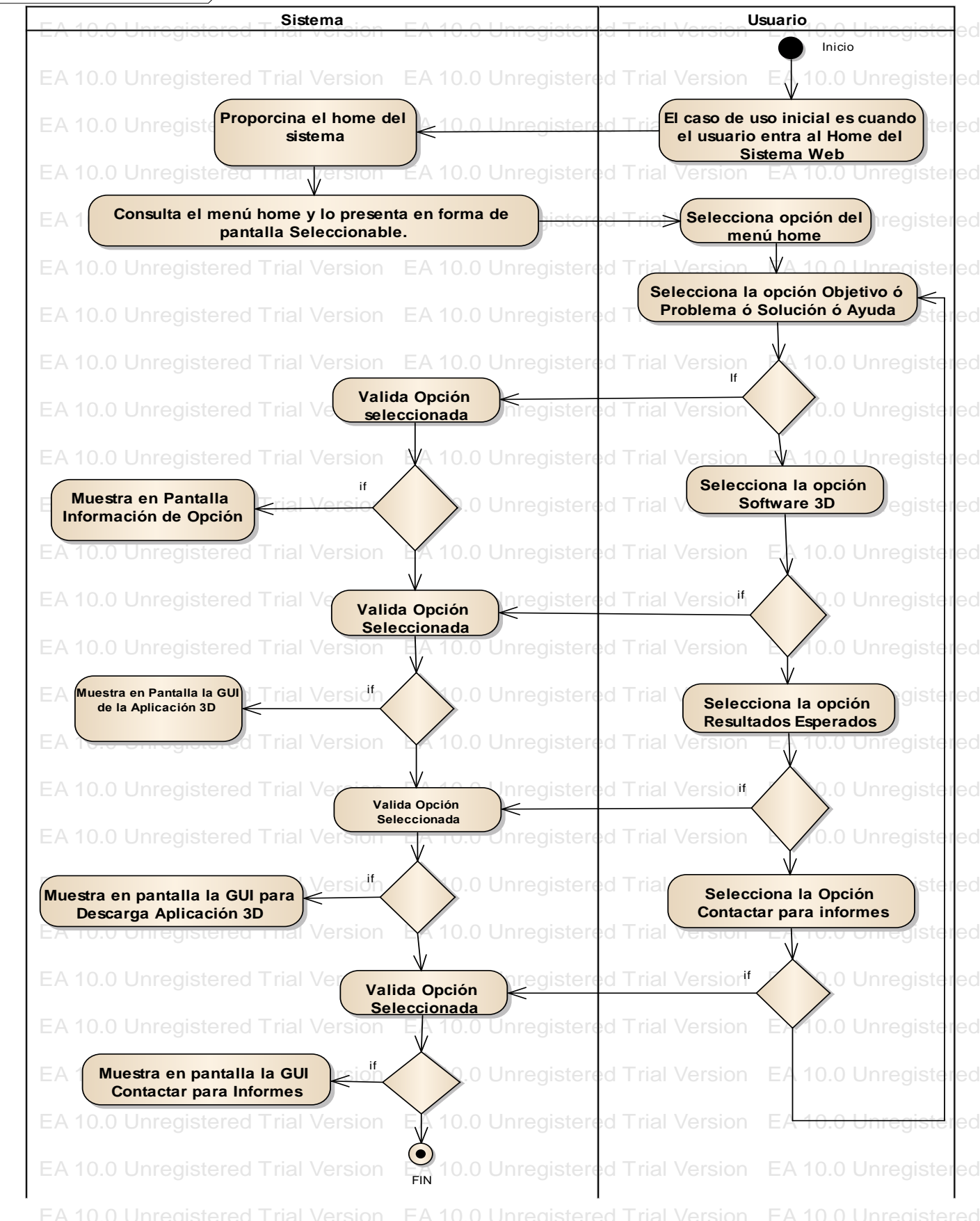
uc Diagrama de Caso de Uso Interactividad Visual



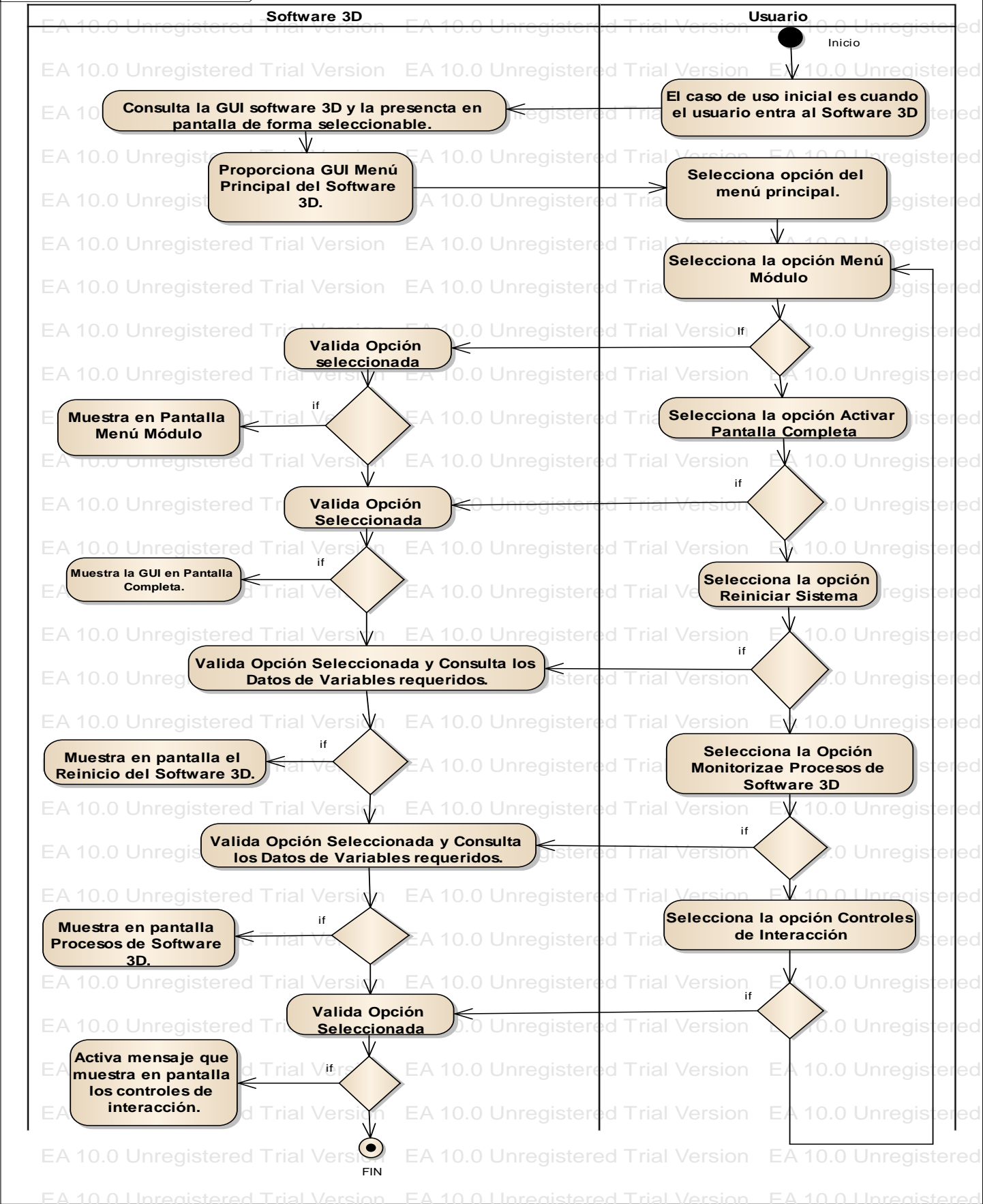
ANEXO III:

DIAGRAMA DE ACTIVIDAD

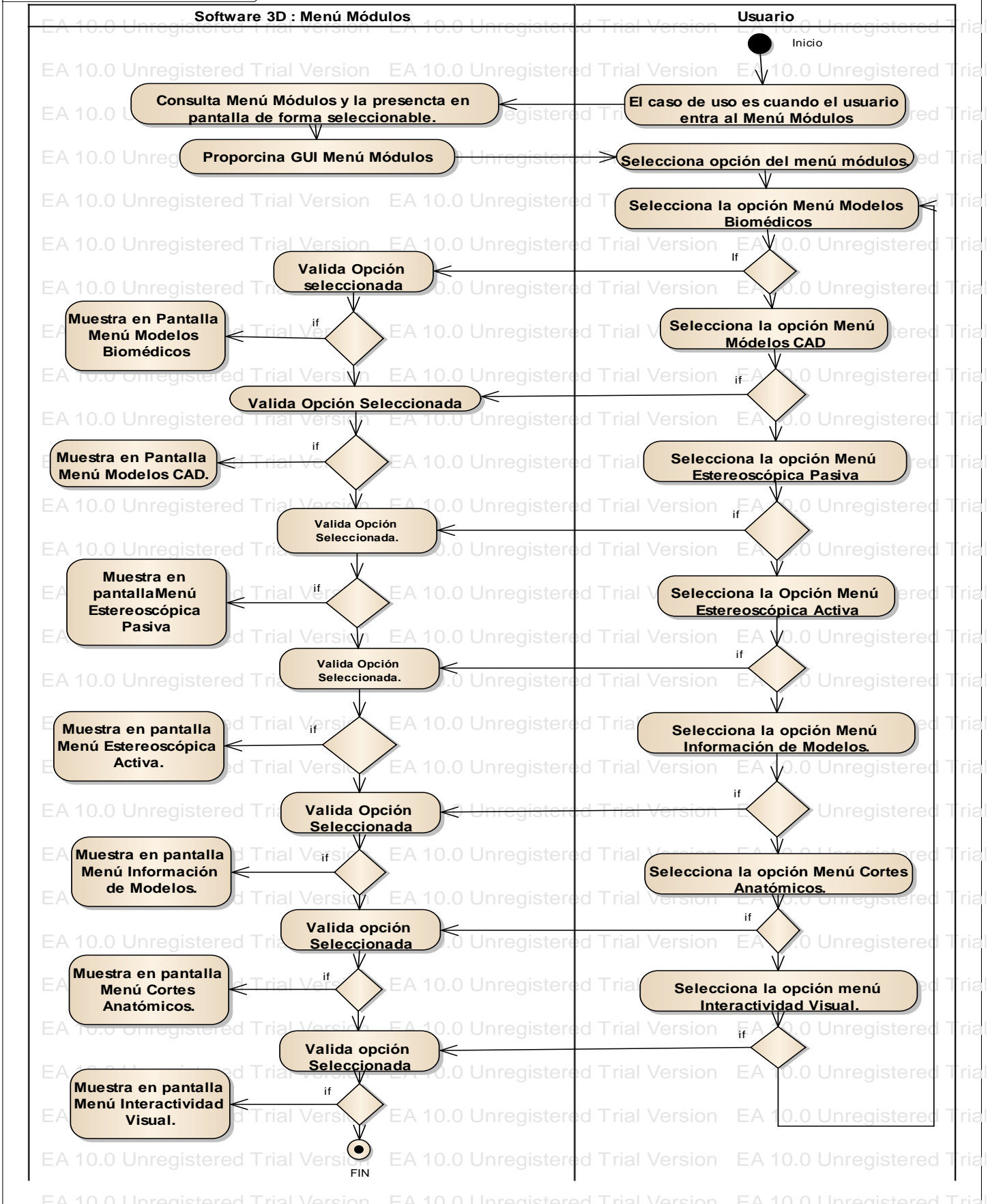
act Diagrama de Actividad Home



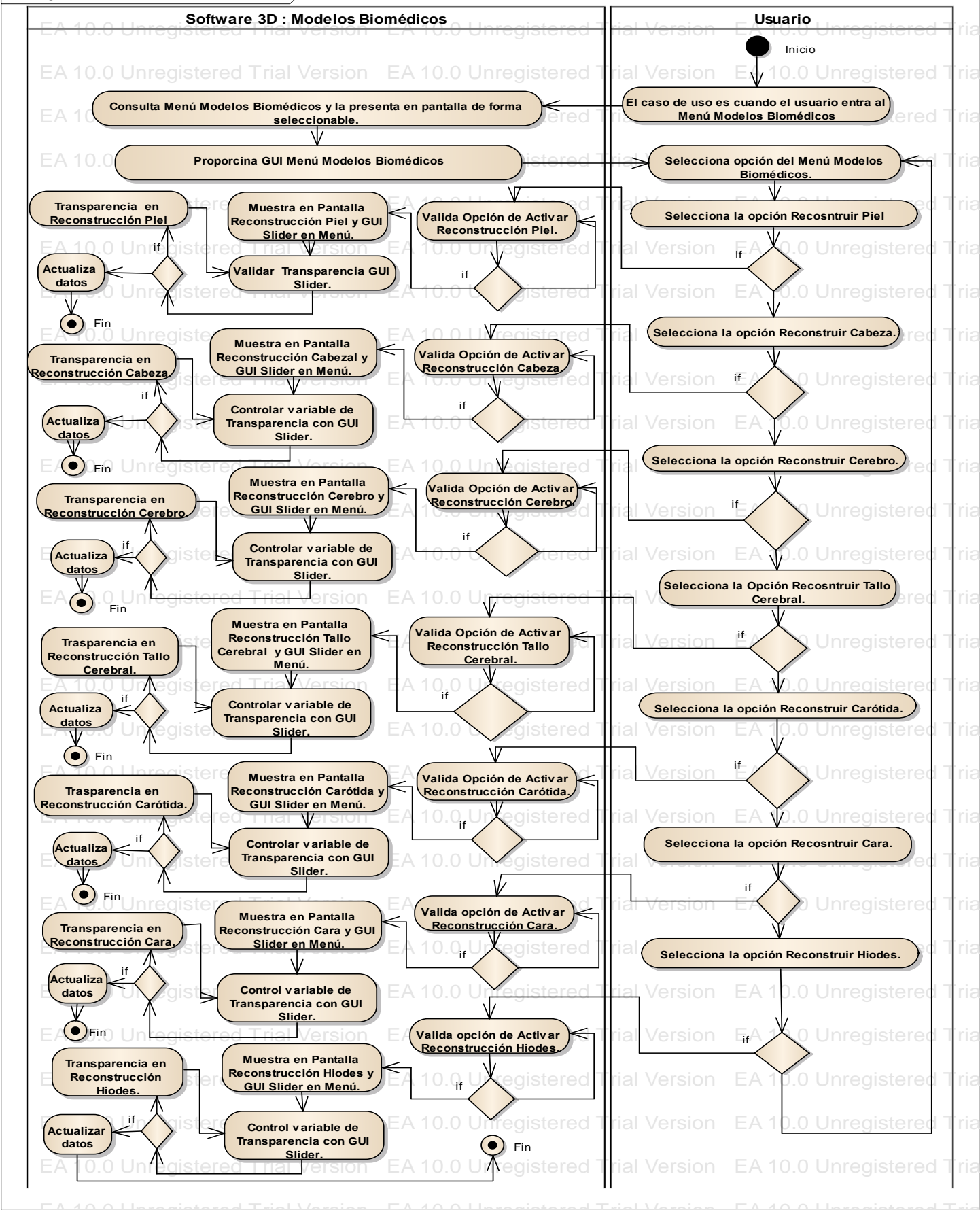
act Diagrama de Actividad Software 3D



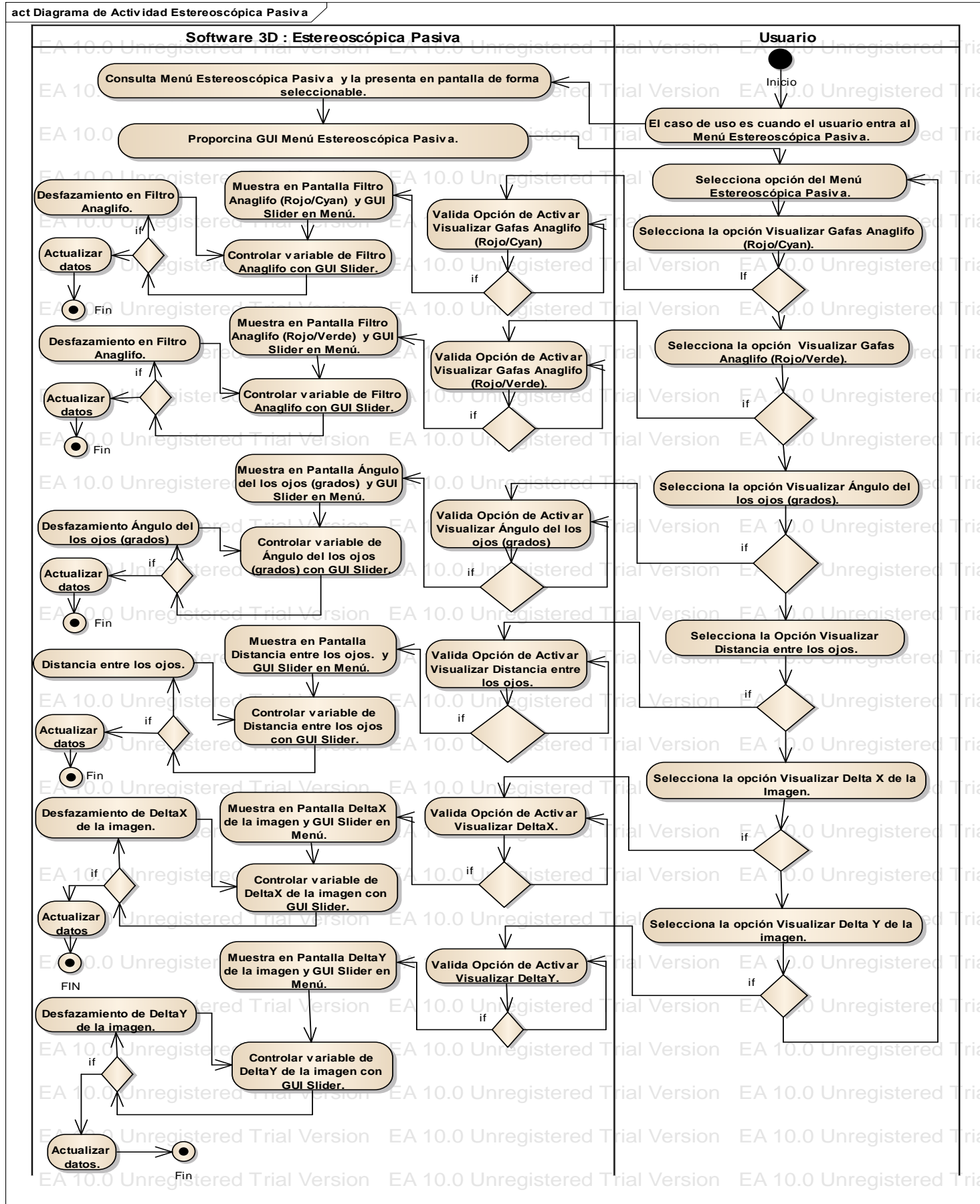
act Diagrama de Actividad Menú Módulos



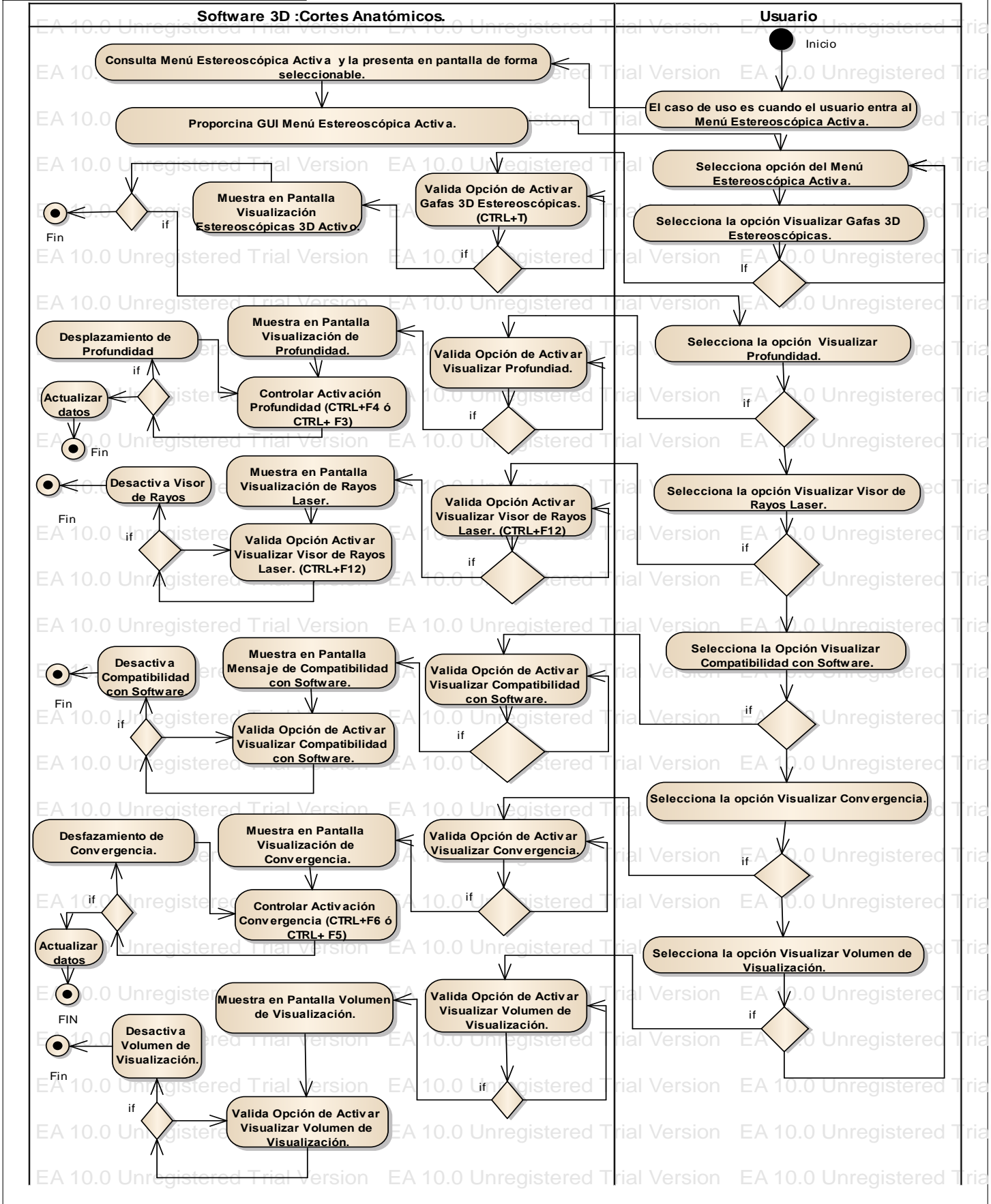
act Diagrama de Actividad Modelos Biomédicos



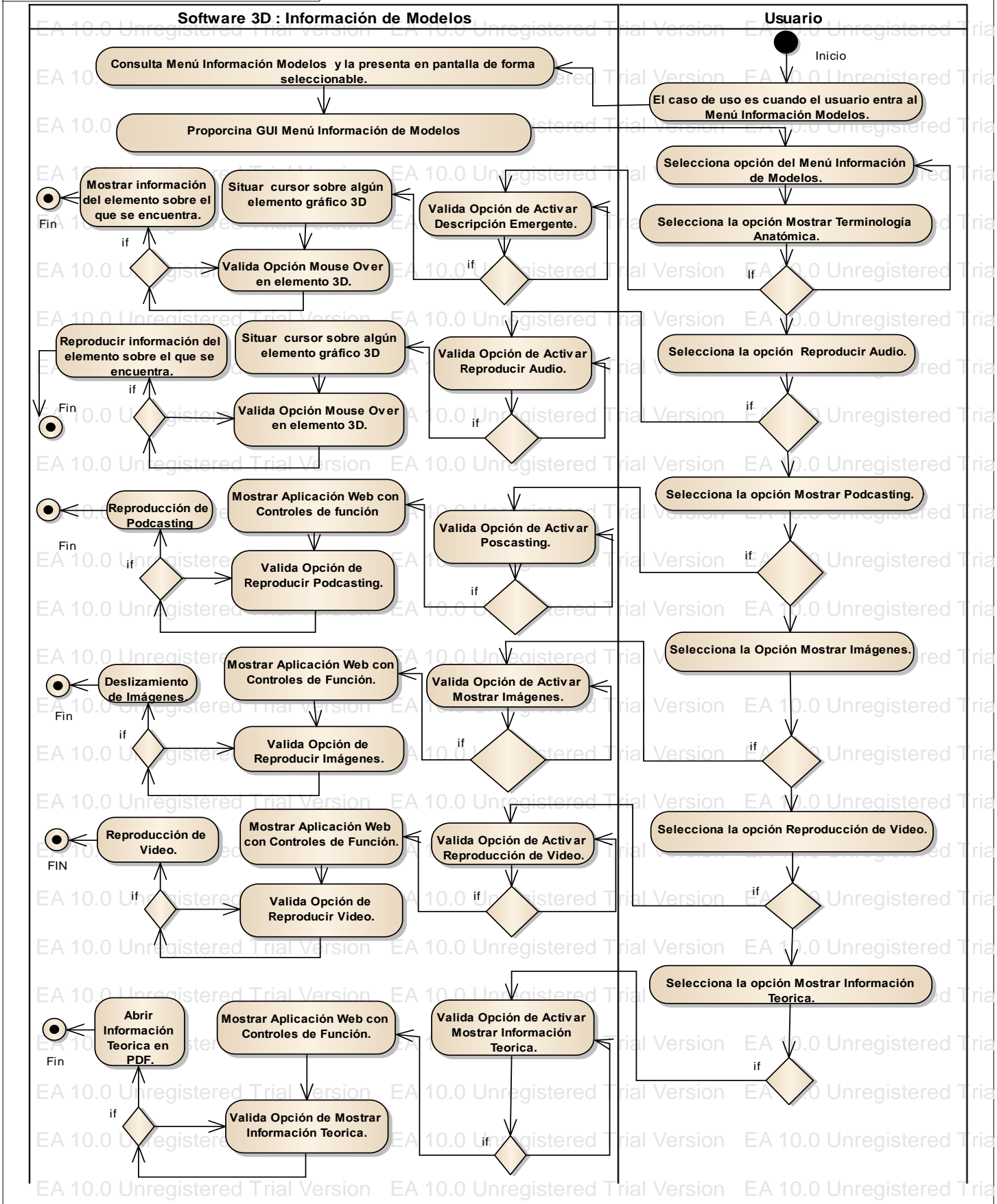
act Diagrama de Actividad Estereoscópica Pasiva



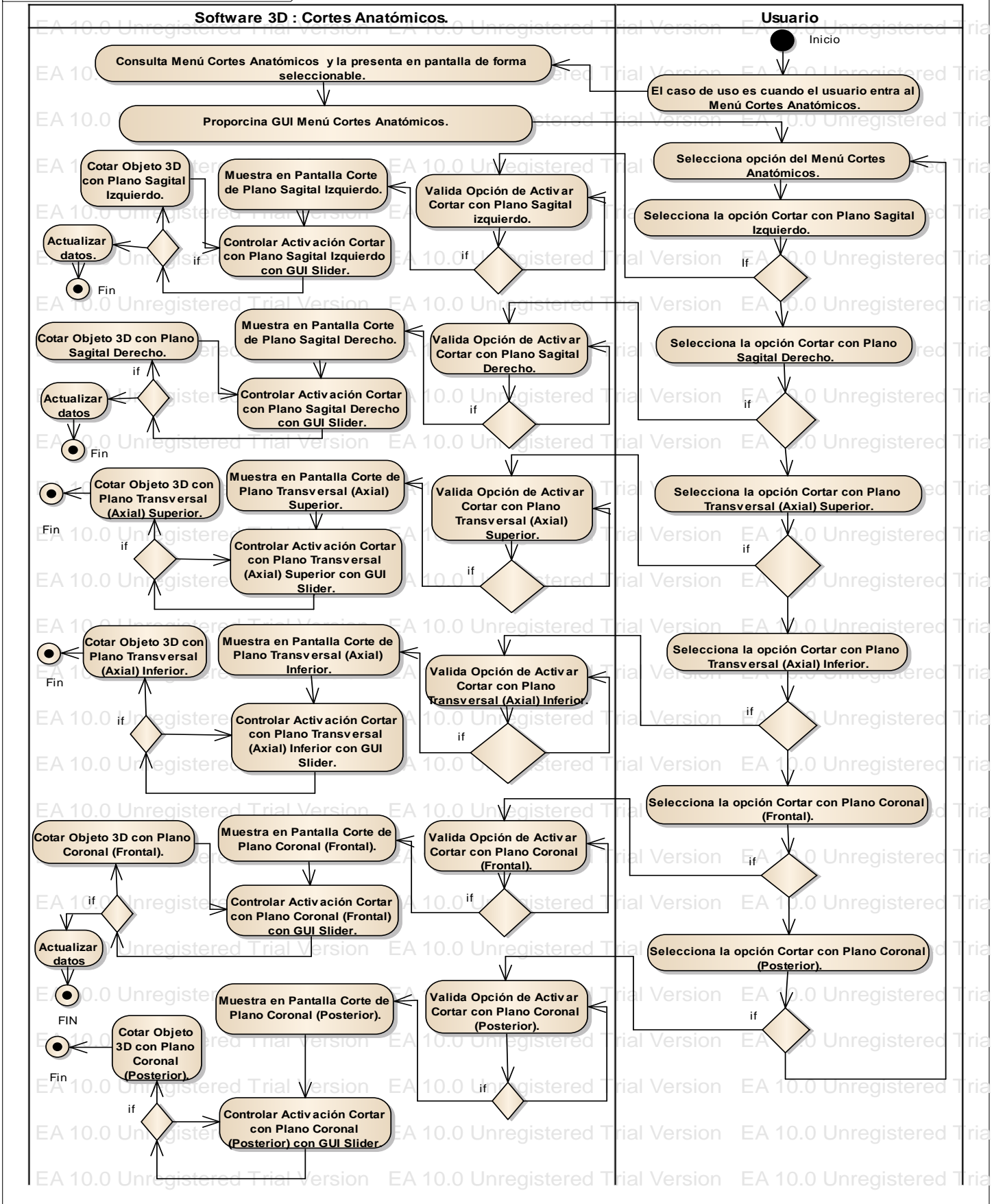
act Diagrama de Actividad Estereoscópica Activa



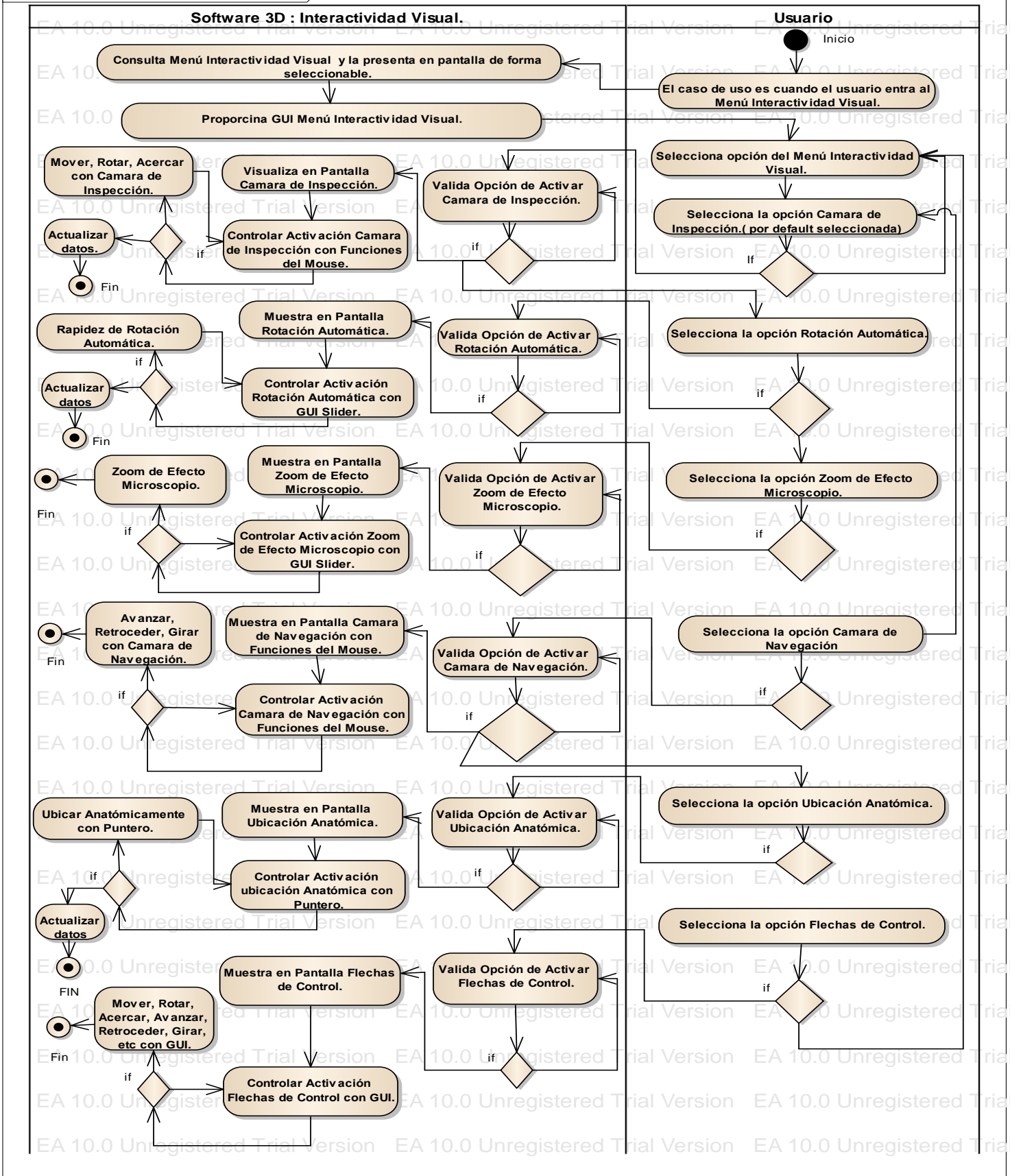
act Diagrama de Actividad Información de Modelos



act Diagrama de Actividad Cortes Anatómicos.

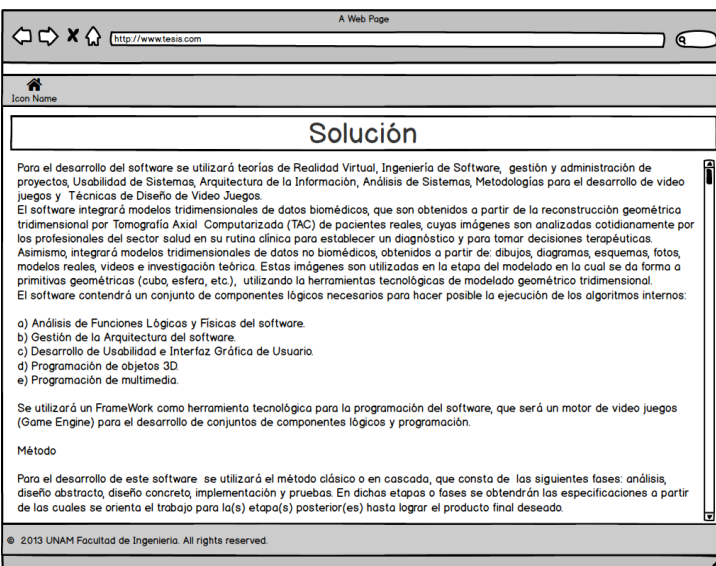
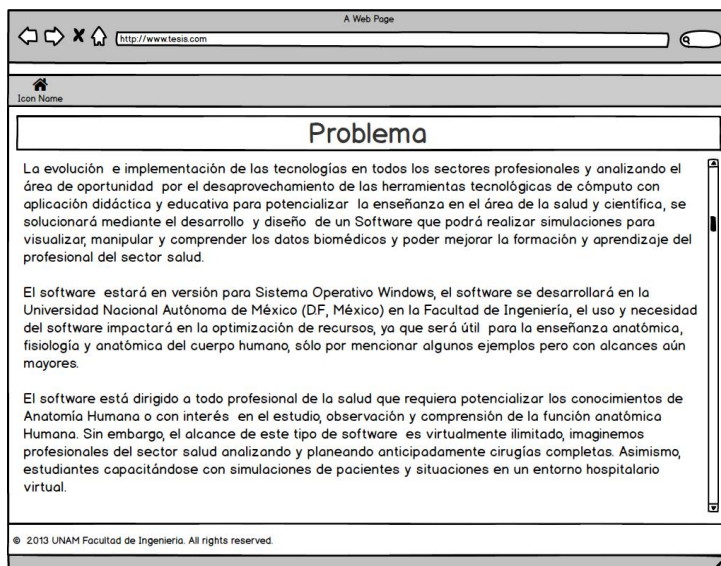
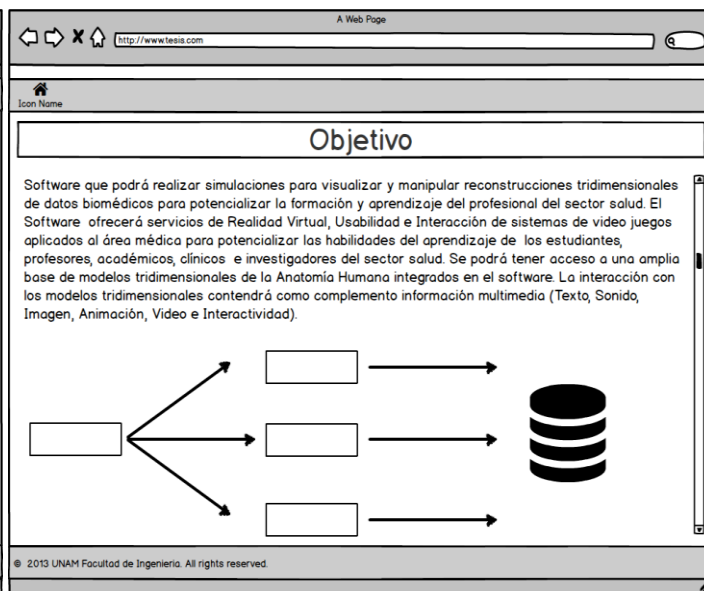
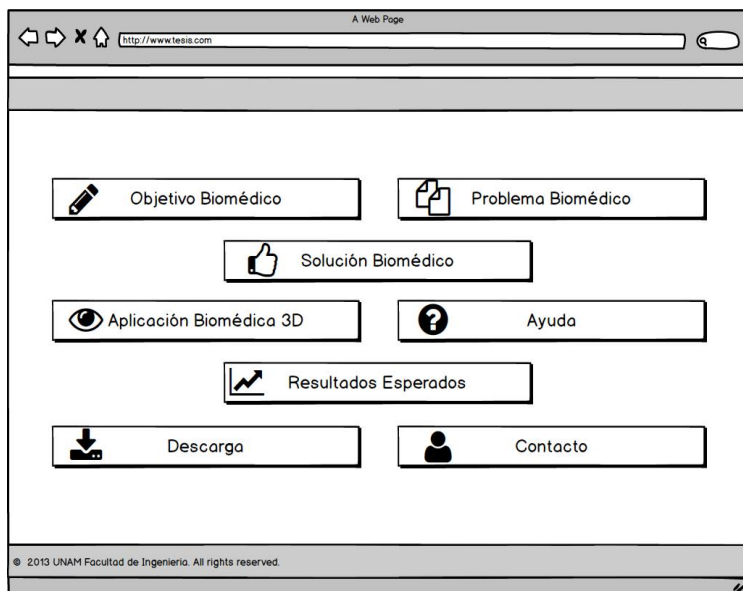


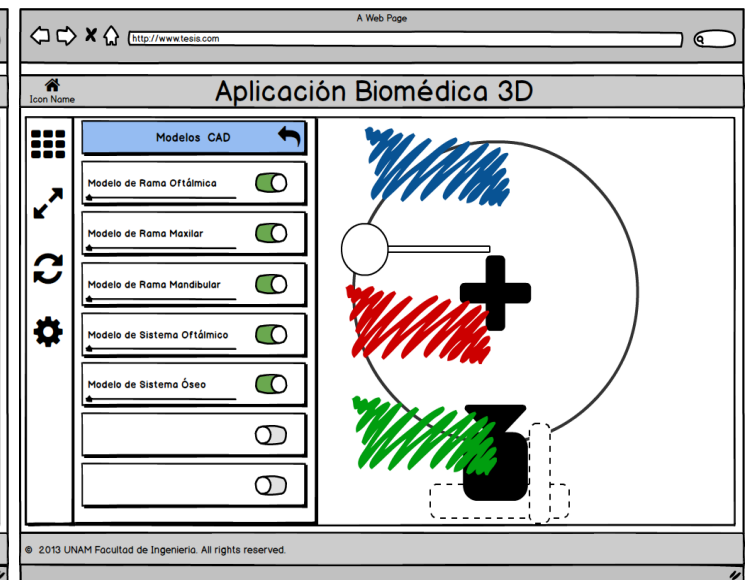
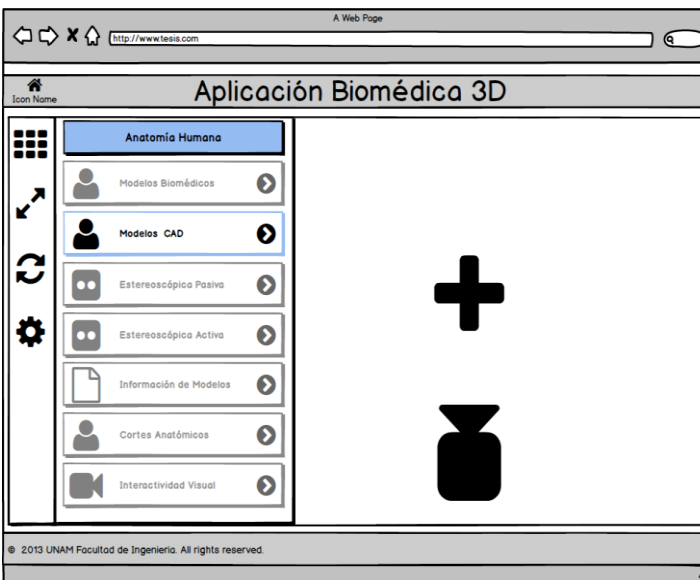
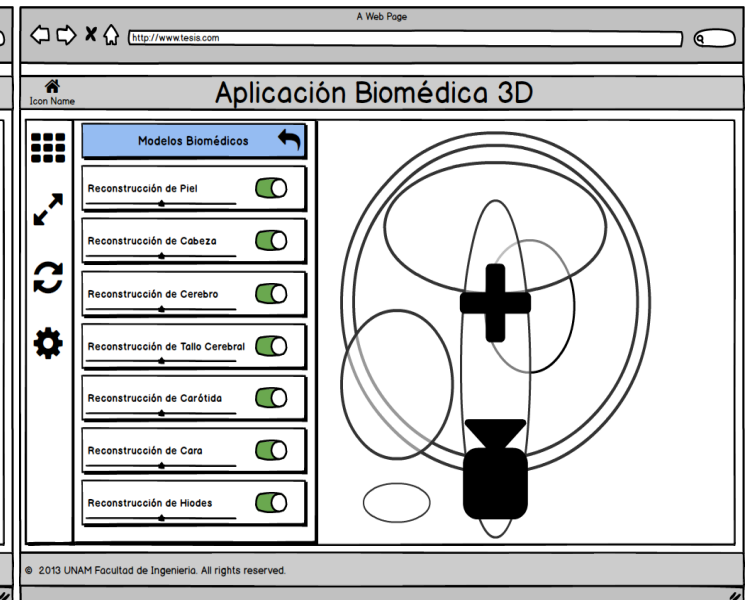
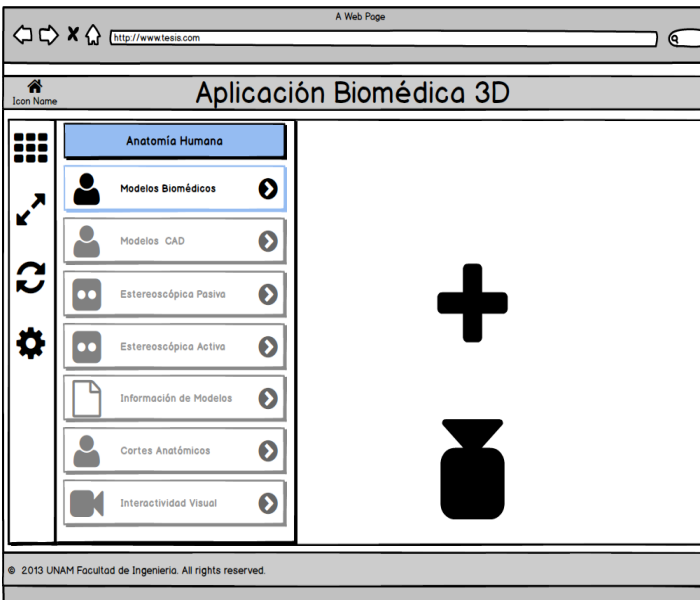
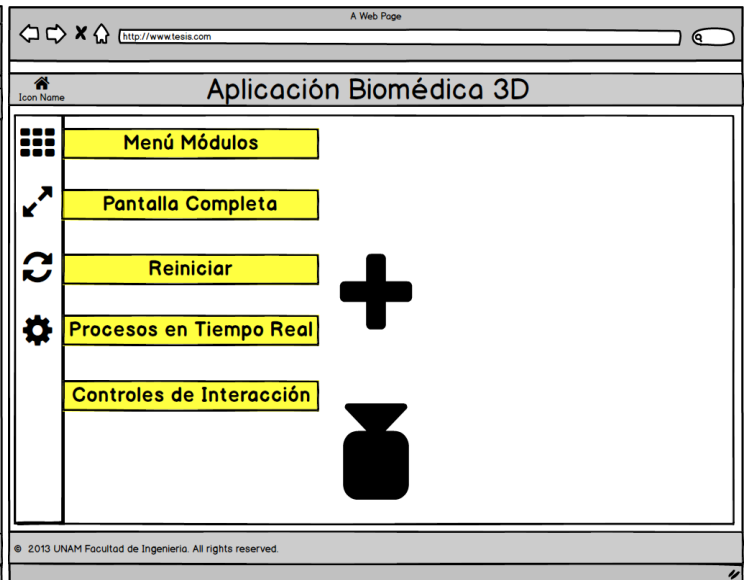
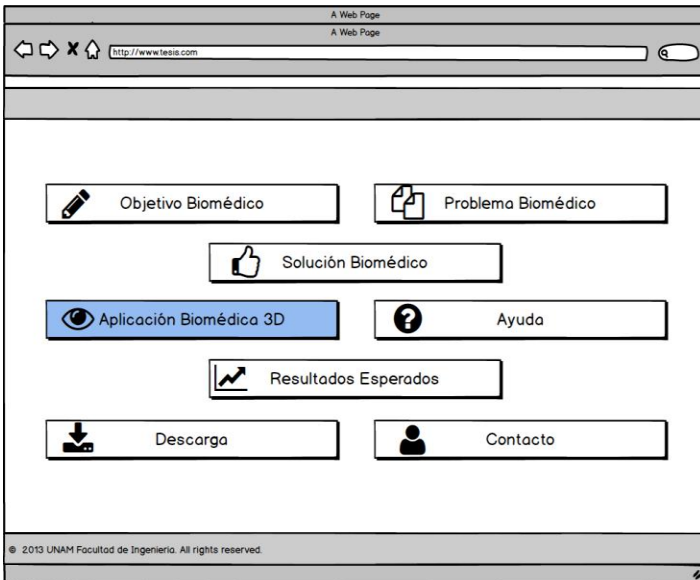
act Diagramas de Actividad Interactividad Visual

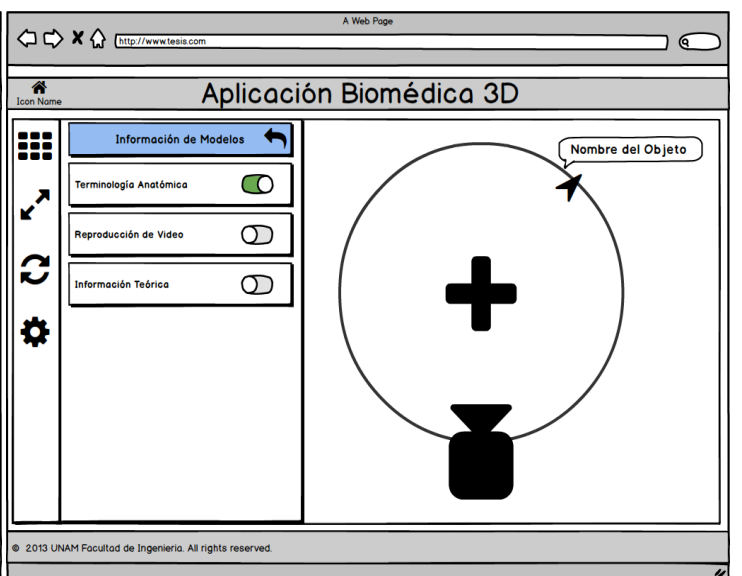
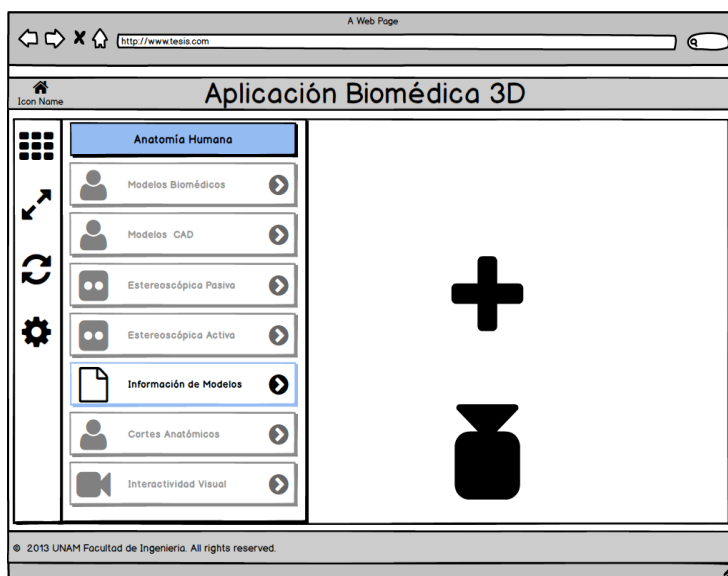
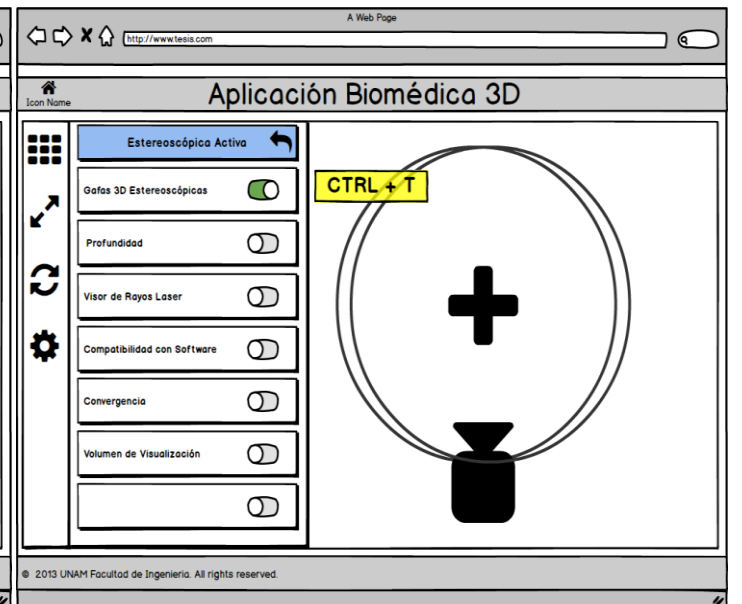
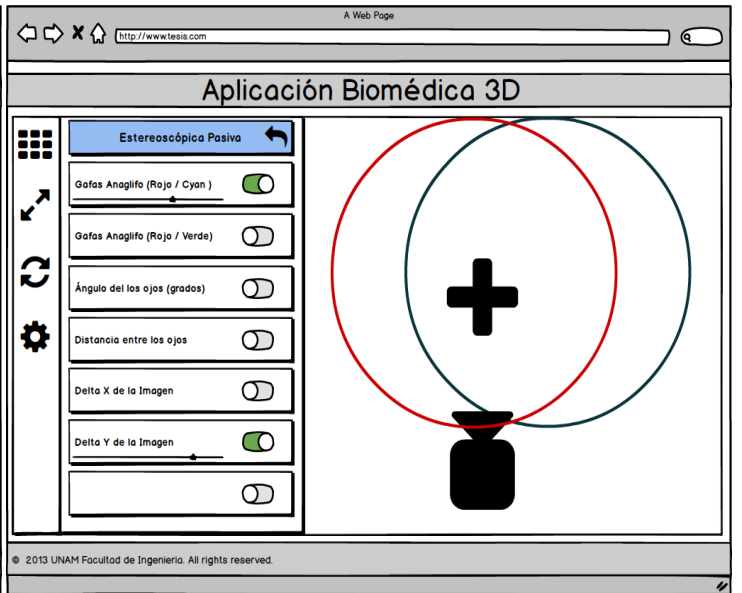
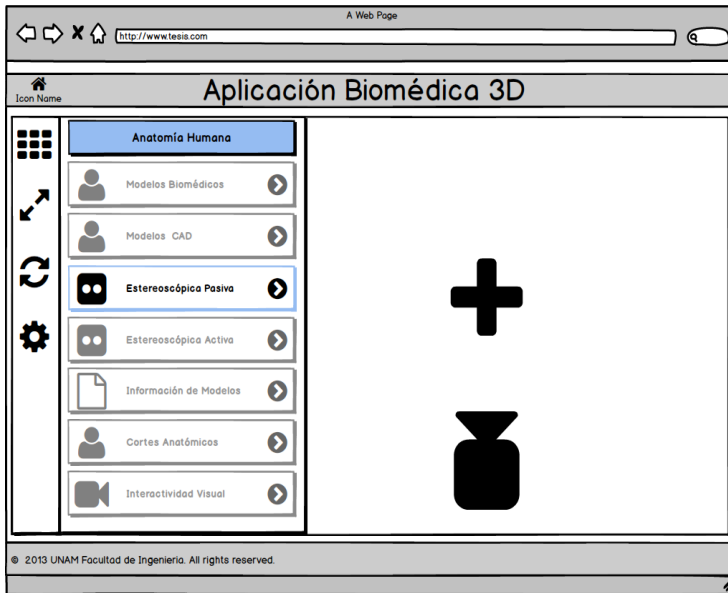


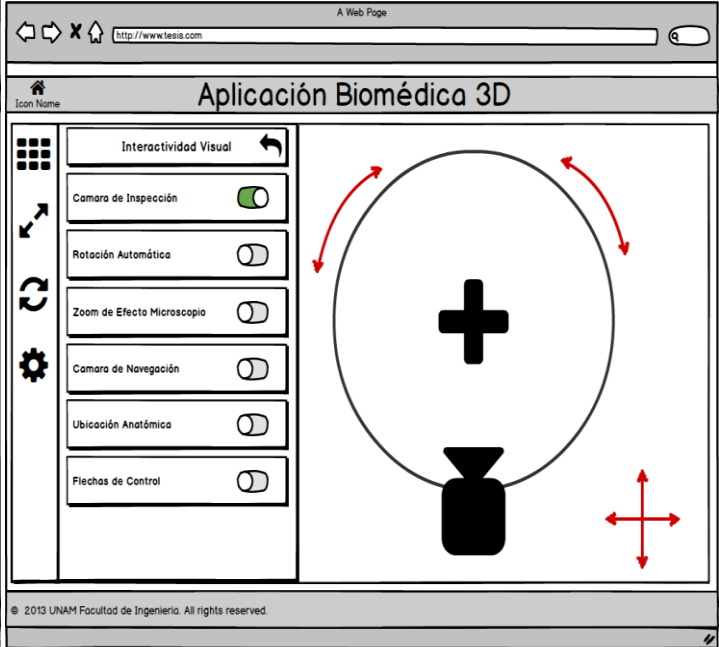
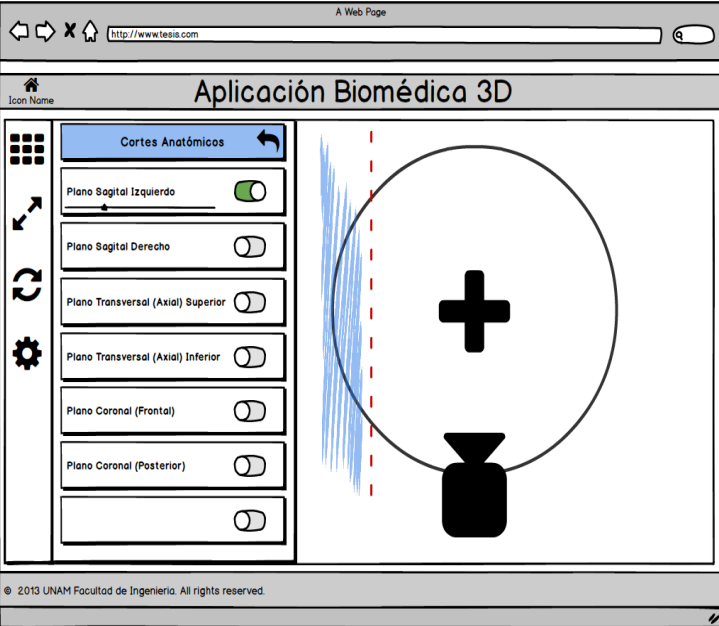
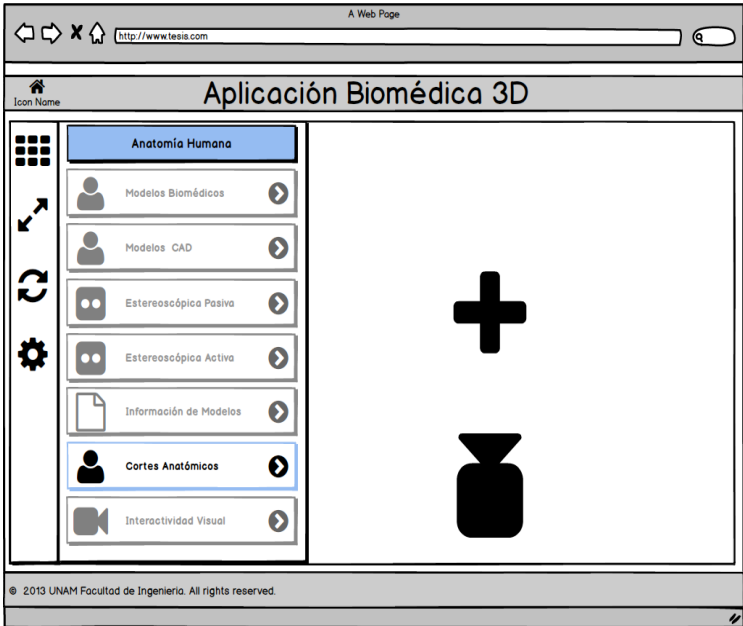
ANEXO IV:

**INTERFAZ GRÁFICA DE
USUARIO**









ANEXO V:

HERRAMIENTAS

TECNOLÓGICAS

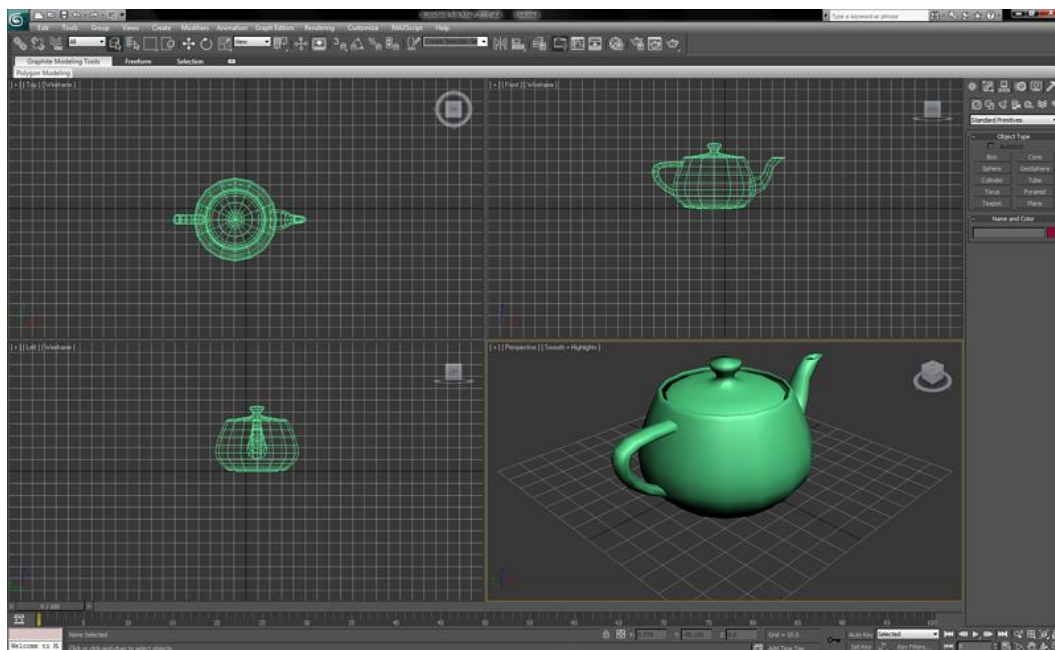
Adobe photoshop

Adobe Photoshop es un editor de gráficos rasterizados desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos, su nombre en español significa literalmente "taller de fotos". Es líder mundial del mercado de las aplicaciones de edición de imágenes y domina este sector de tal manera que su nombre es ampliamente empleado como sinónimo para la edición de imágenes en general. Actualmente forma parte de la familia Adobe Creative Suite y es desarrollado y comercializado por Adobe Systems Incorporated inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows. Su distribución viene en diferentes presentaciones, que van desde su forma individual hasta como parte de un paquete, siendo éstos: Adobe Creative Suite Design Premium y Versión Standard, Adobe Creative Suite Web Premium, Adobe Creative Suite Production Studio Premium y Adobe Creative Suite Master Collection.



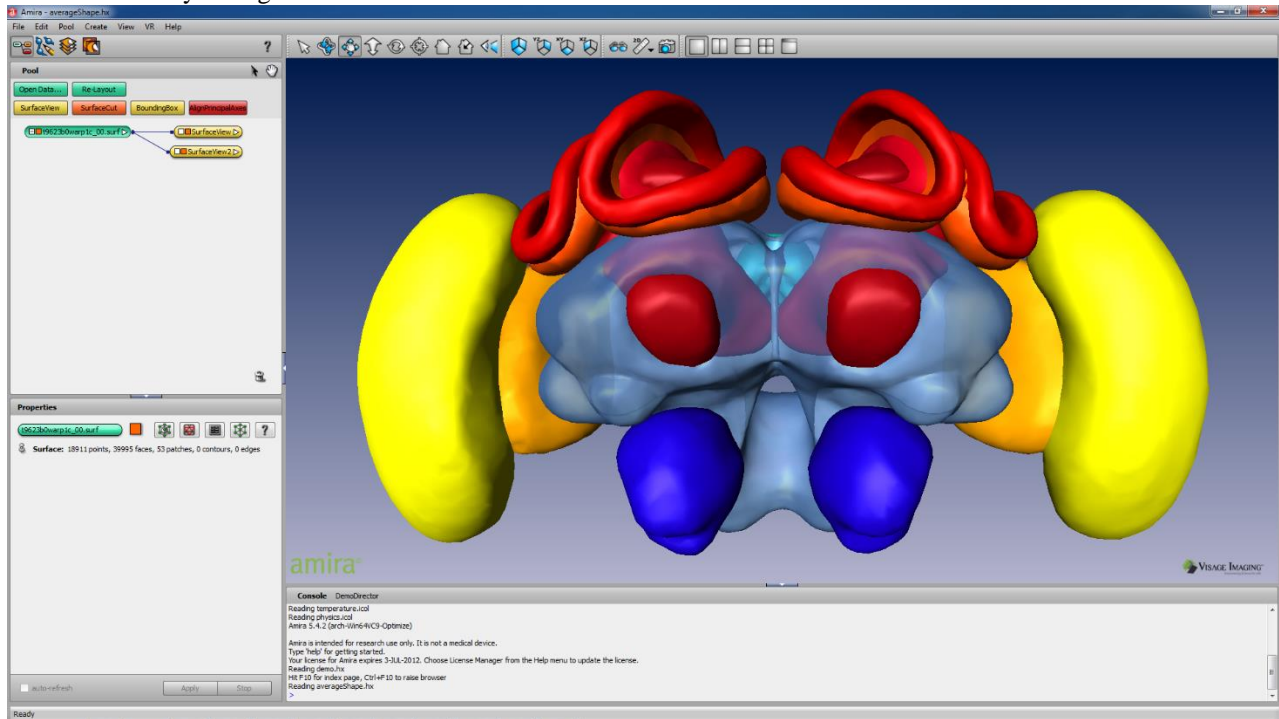
Autodesk 3ds Max

Autodesk 3ds Max (anteriormente 3D Studio Max) es un programa de creación de gráficos y animación 3D desarrollado por Autodesk, en concreto la división Autodesk Media & Entertainment (anteriormente Discreet). Creado inicialmente por el Grupo Yost para Autodesk, salió a la venta por primera vez en 1990 para DOS. 3ds Max, con su arquitectura basada en plugins, es uno de los programas de animación 3D más utilizado, especialmente para la creación de videojuegos, anuncios de televisión, en arquitectura o en películas.



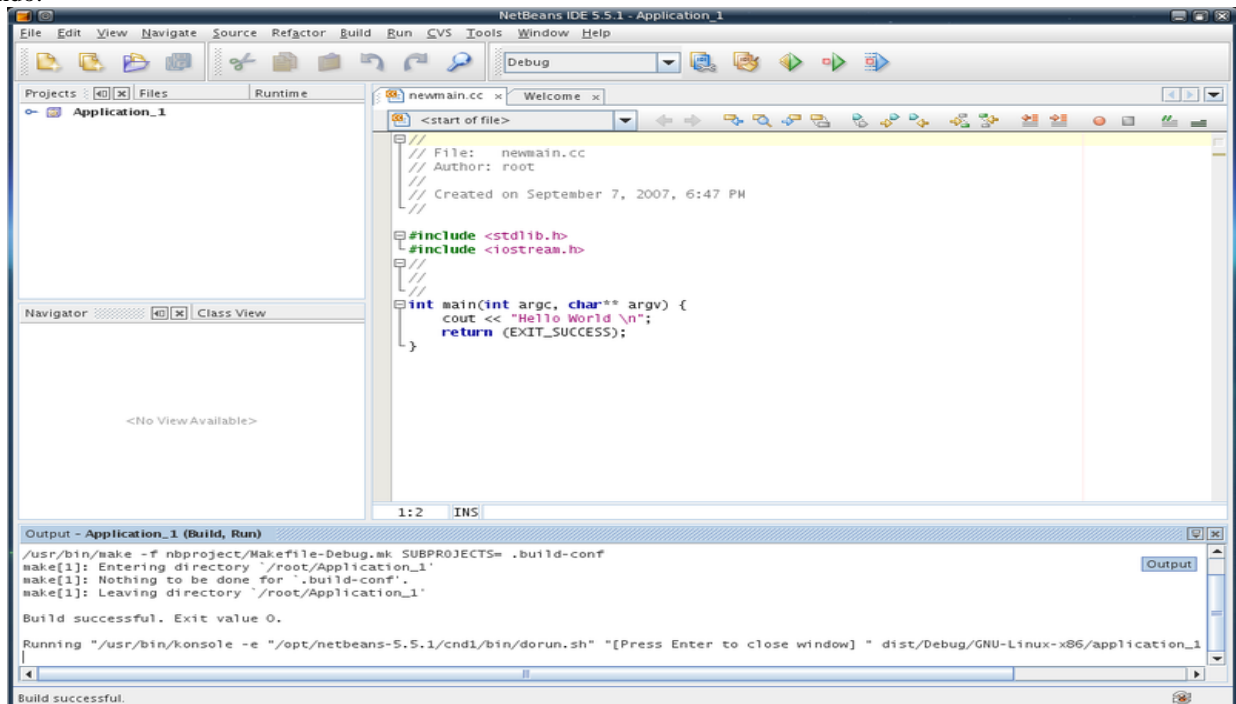
Amira 3D

Amira es una poderosa y multifacética plataforma de software 3D para visualizar, manipular, y la comprensión de los datos de la tomografía computarizada, la microscopía, MRI, y muchas otras modalidades de imagen. Con la velocidad y flexibilidad increíble, Amira permite flujos de trabajo avanzados de imagen 3D para especialistas en las áreas de investigación que van desde la biología molecular y celular de la neurociencia y bioingeniería.



NetBeans

Es un entorno de desarrollo integral libre, hecho principalmente para lenguajes de programación para web. Existe además un número importante de módulos para extenderlos. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo.



ANEXO VI:

NVIDIA 3D VISION

WIRELESS GLASSES

REQUERIMIENTOS DE MINIMOS DE NVIDIA 3D VISION

NVIDIA GeForce 3D Vision, tiene la experiencia más inmersiva para una PC. Los requisitos mínimos del sistema para asegurar que una PC está configurada correctamente para disfrutar de la experiencia GeForce 3D Vision.

a) Requerimientos mínimos de Sistema:





- Microsoft Windows Vista 32-bit a 64-bit.
- Intel Core 2 Duo ó AMD Athlon X2 CPU o superior.
- 1GB de memoria en sistema (2GB son recomendables).
- 100 MB de espacio libre en disco.

b) Compatibilidad NVIDIA GeForce en CPU:

- GeForce GT 630, 640, 720, 730,740.
- GeForce GTX 650, 650 Ti, 650 Ti BOOST, 660, 660 Ti, 670, 680
- GeForce GTX 750, 750 Ti, 760, 770, 780,780 Ti
- GeForce GTX 950,960,970,980,980 Ti
- GeForce GTX TITAN Black, X, Z.

d) Compatibilidad de Pantalla 3D:

Asegúrese de que está ejecutando su aplicación en las resoluciones que soporta la tarjeta GeForce 3D Vision. Con el fin de apoyar GeForce 3D Vision, su aplicación debe estar configurado para funcionar a la configuración correcta para su visualización. Si se tiene:

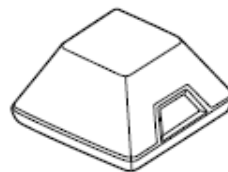
	<ul style="list-style-type: none"> • 120 Hz LCD: Seleccione una resolución en la aplicación de 1680X1050. Utilice el asistente de NVIDIA o ajustes en el juego para establecer una frecuencia de refresco de 100Hz o superior.
	<ul style="list-style-type: none"> • 60 Hz HDTV DLP: Seleccione una resolución de 1920X1080, 1768X992 o 1280X1024 y establecer la frecuencia de actualización de 60 Hz.
	<ul style="list-style-type: none"> • 120 Hz HD 3D del proyector: Seleccione una resolución de 1280X720. Utilice el Asistente de NVIDIA o ajustes en el juego para establecer una frecuencia de refresco de 120Hz.
	<ul style="list-style-type: none"> • Ejecutar la aplicación en modo de pantalla completa: GeForce 3D Vision actualmente sólo es compatible con las aplicaciones de pantalla completa y no se puede ejecutar en modo de ventana. Asegúrese de seleccionar el modo de pantalla completa de los ajustes en la aplicación.
	<ul style="list-style-type: none"> • Permita que su 120 Hz se caliente: Para conseguir la mejor calidad de imagen 3D estereoscópica, cuando se utiliza una frecuencia de 120 Hz LCD, deje por favor su monitor se caliente. Este período de calentamiento es normal para monitores LCD.

EQUIPO

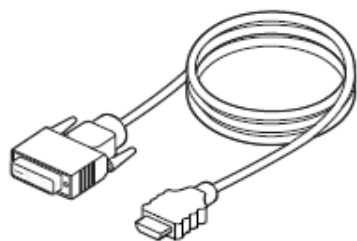
Las siguientes partes y ensambles son incluidas en el NVIDIA GeForce 3D Vision:



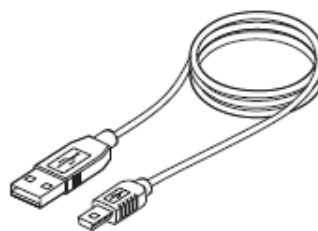
Wireless Glasses



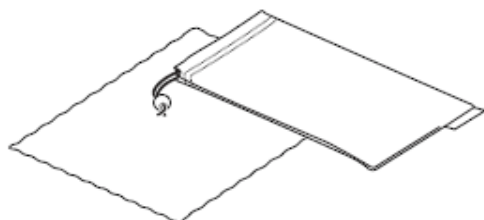
IR Emitter



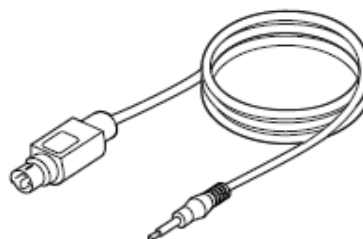
DVI-to-HDMI Cable
(For use with DLP HDTVs)



Two USB Cables
(6 foot and 10 foot)



Cleaning Pouch
Cleaning Cloth



VESA 3-pin Stereo Cable

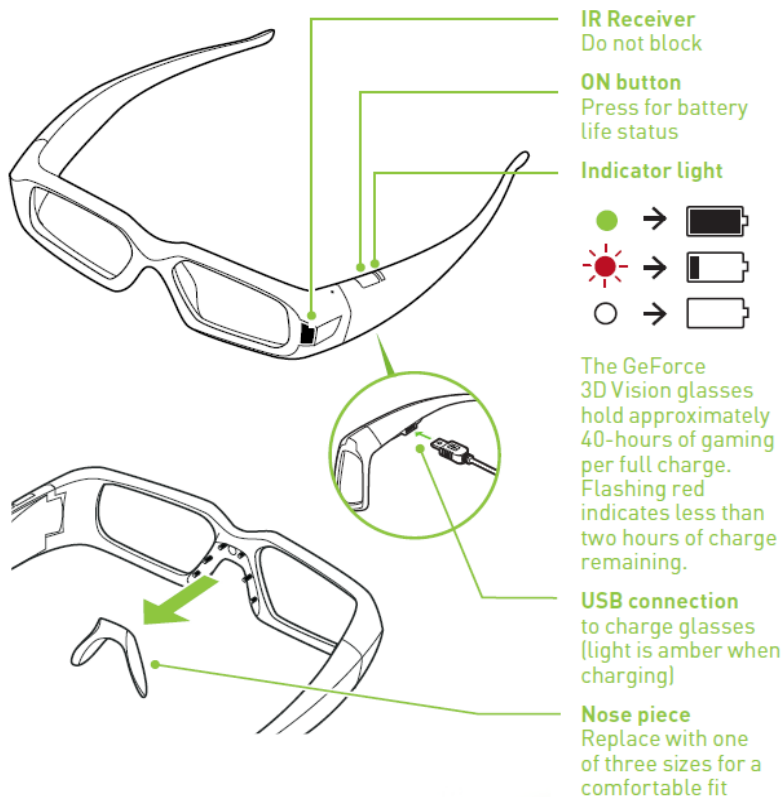


Software and Manuals CD with
GeForce 3D Vision User Guide

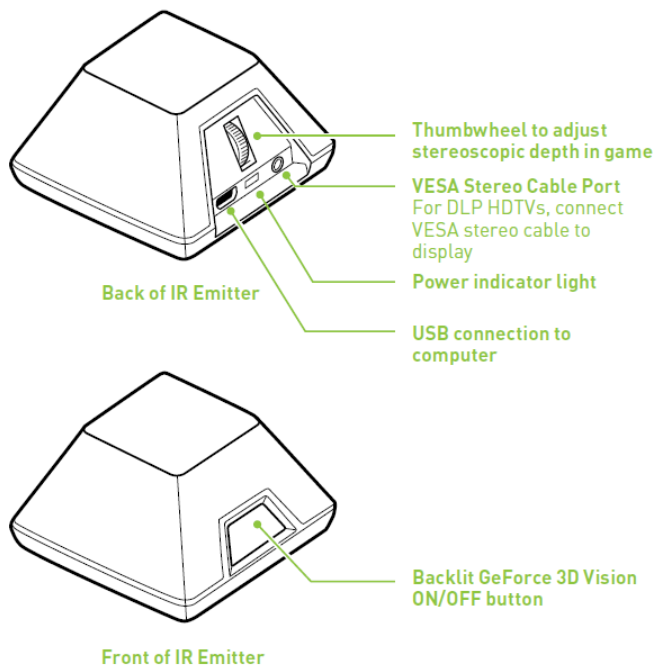


Interchangeable
Nosepieces

GAFAS INALÁMBRICAS

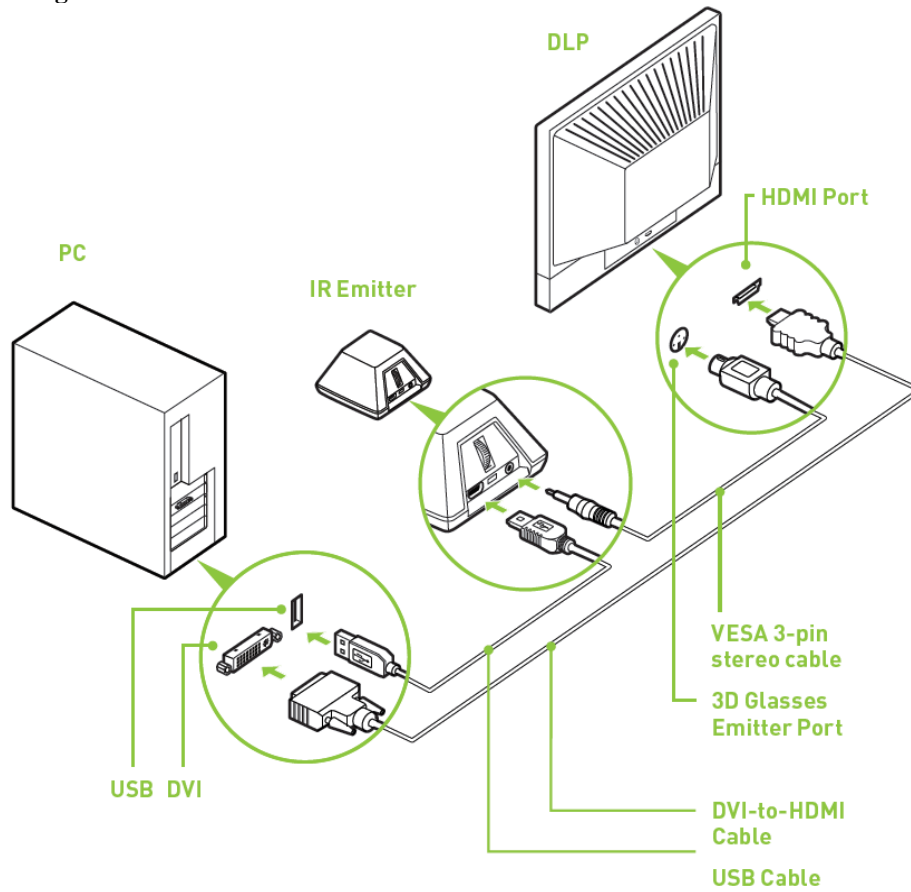


EMISOR IR

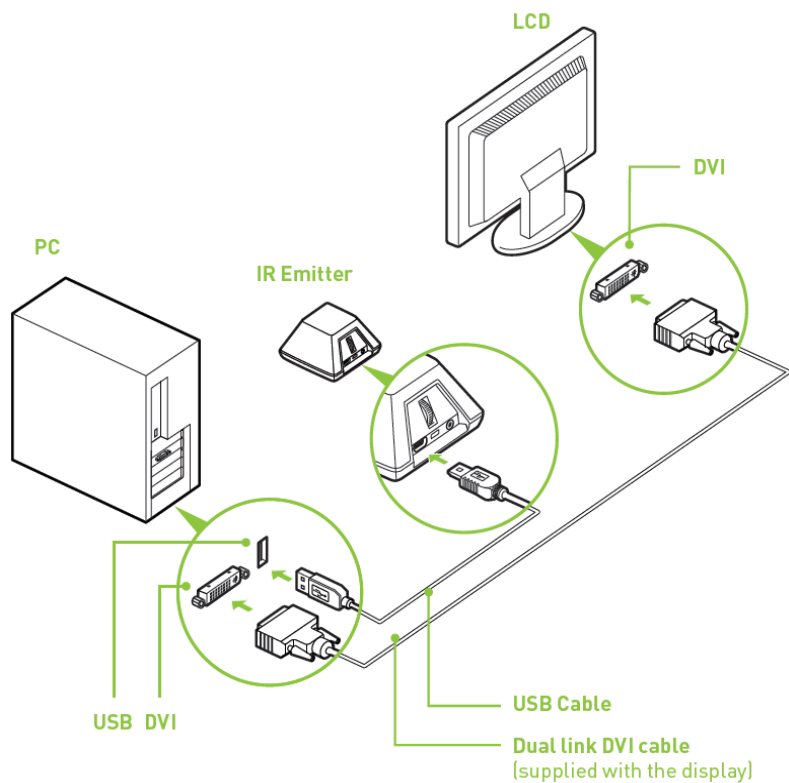


CONFIGURACIONES

a) DLP HDTV Diagrama de conexión



b) 120 Hz Diagrama de conexión del LCD



INSTALACIÓN Y CONFIGURACIÓN

- 1) Antes de comenzar la instalación, debe eliminar el controlador de gráficos actual del equipo.

Vaya a Inicio -> Panel de control -> Programas y características

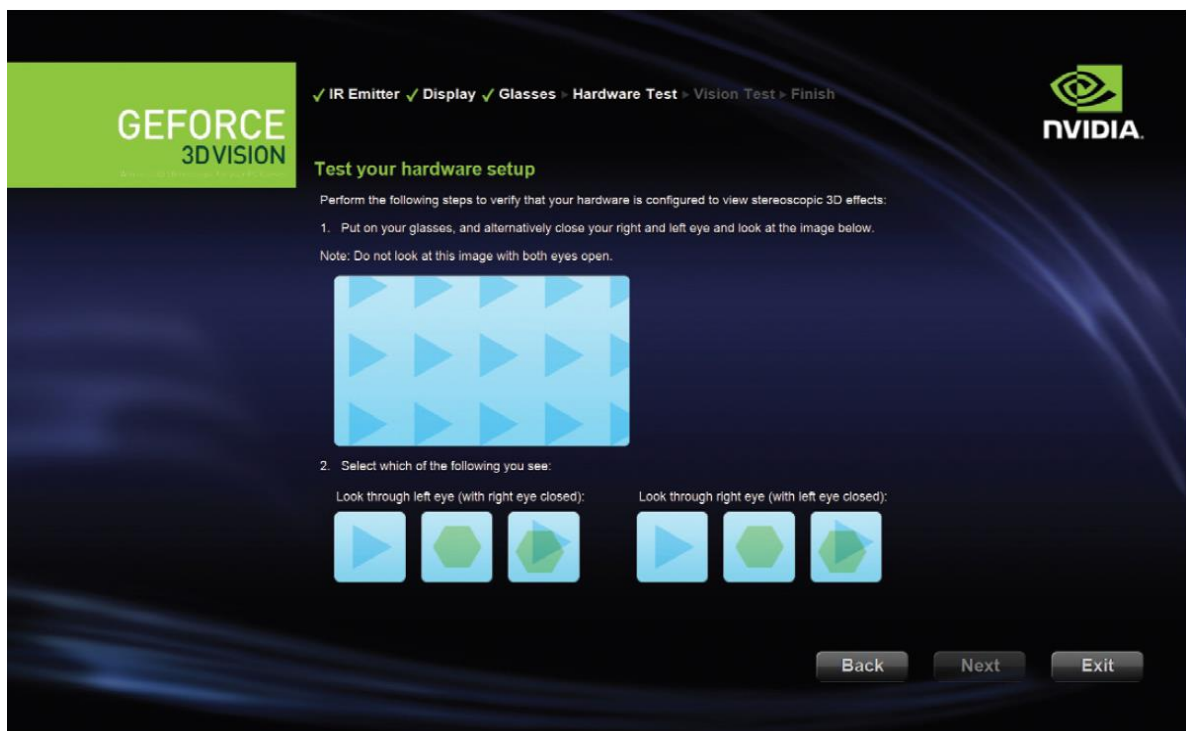
Si tiene una tarjeta gráfica NVIDIA, haga doble clic en controlador de NVIDIA.

- Seleccione la opción quitar y solo seguir.
- Seleccione el controlador de pantalla NVIDIA.
- Haga clic en quitar
- Reinicie su equipo

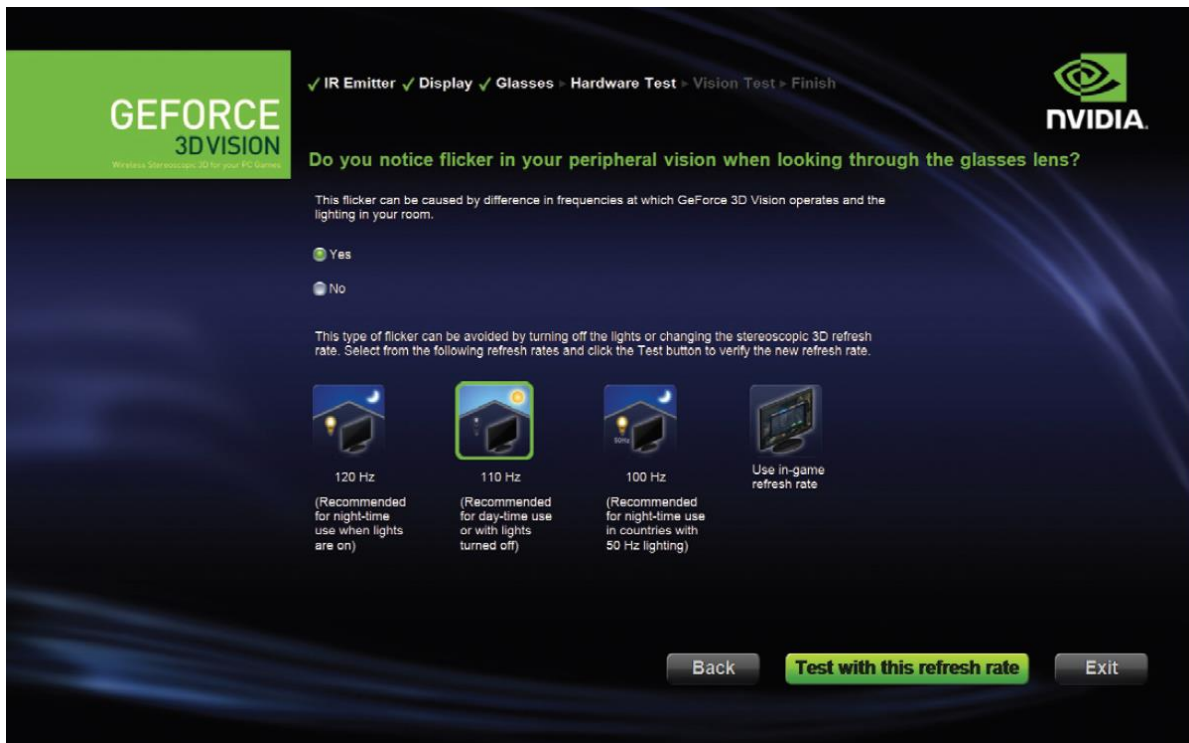
* Si va a reemplazar una tarjeta gráfica no NVIDIA, asegúrese de retirar el controlador de gráficos y apagar el sistema para reemplazar la tarjeta antes de continuar.

- 2) Inserte el Software GeForce 3D Vision y los controladores del Manual CD.
- 3) Seleccione instalar los controladores de la tarjeta gráfica GeForce.
- 4) Seleccione sí, quiero reiniciar mi equipo ahora cuando se le solicite. El proceso de instalación continuará automáticamente después del reinicio.
- 5) Seleccione siguiente para instalar los controladores del GeForce 3D Vision.
- 6) Seleccione terminar para completar la instalación. El asistente de configuración GeForce 3D Vision comienza automáticamente cuando el InstallShield se completa. El Asistente de configuración guiará a través de la instalación del emisor de infrarrojos, la configuración de la pantalla, y la configuración de las gafas.
- 7) Seleccione siguiente para iniciar el asiste de instalación. El asistente es muy intuitivo y fácil de seguir.
- 8) Conecte el emisor IR cuando se le pida.
- 9) Seleccione su entorno de juego.
- 10) Búsqueda de la pantalla 3D disponible.
- 11) Encienda las gafas.
- 12) Prueba tu configuración de hardware.

Esta pantalla se utiliza para verificar que su hardware está configurado correctamente para ver los efectos 3D estereoscópicos y que sus gafas y emisor IR están funcionando correctamente. Siga las instrucciones en la pantalla.



Si no puede ver las imágenes adecuadas, el asistente de configuración no continuará hasta que el problema de hardware se resuelve. Utilice el botón atrás para revisar sus gafas, emisor y configuración de hardware. (Para pantallas LCD solamente) ¿Parpadeo en la visión periférica cuando se mira a través de las gafas? Si usted está viendo el parpadeo excesivo en su visión periférica, esta pantalla le permite cambiar la frecuencia de actualización de la pantalla. Si selecciona Sí, se muestran las opciones de actualización de frecuencia.



Selecciona el botón de prueba con frecuencia de actualización para verificar la selección de frecuencia con el parpadeo de los lentes.

13) Verifica la disponibilidad para visualizar contenido estereoscópico 3D.

Ahora que sabemos que el hardware puede interpretar contenido estereoscópico 3D, necesitamos asegurarnos que pueda hacerlo. La incapacidad para ver la imagen de prueba médica en 3D estereoscópico puede deberse a:

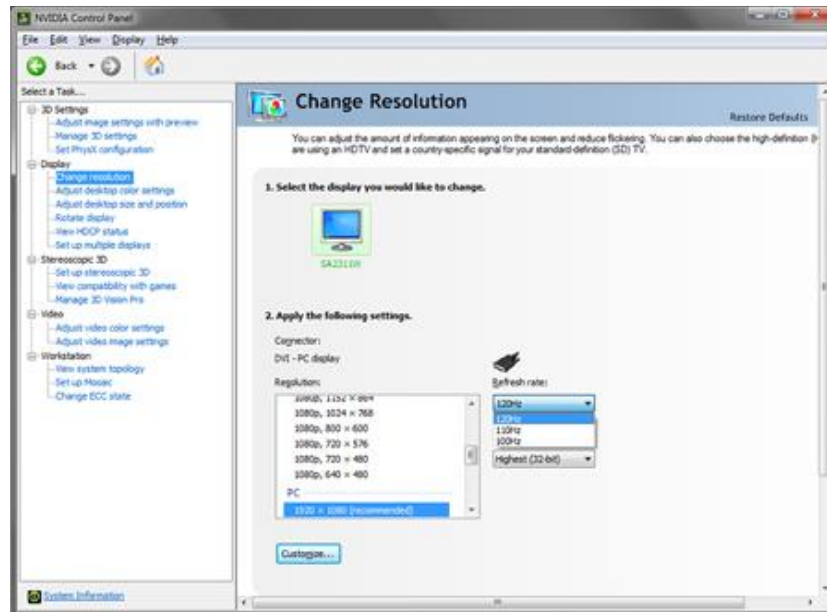
- Los ajustes necesarios para mejorar la visualización en 3D estereoscópico. Seleccione atrás y compruebe que puede ver en 3D estereoscópico en la prueba de la pantalla de configuración del hardware.
- Los problemas visuales. Consulte las normativas de seguridad incluidas en la guía del usuario de precauciones. Consulte a un médico si sospecha de un problema físico para ver estereoscópico 3D.

14) Instalación Completa

Ahora estás listo para ver en estereoscópica 3D, para detalles de información en el panel de control NVIDIA, la guía de usuario GeForce 3D Vision se encuentra en el CD.

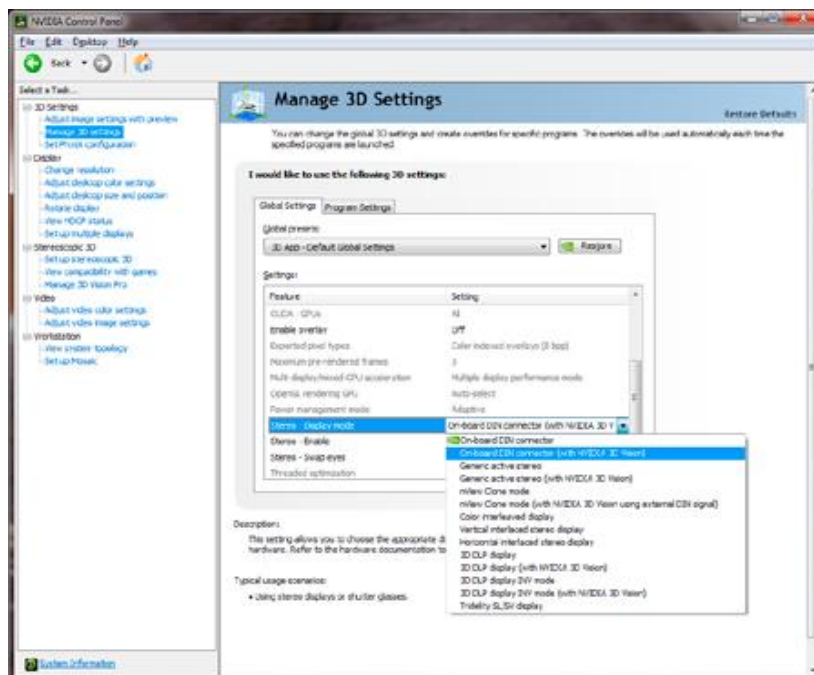
CONFIGURACION DE ESTEREO (SOLÓ PARA TARJETA NVIDIA QUADRO)

1. Compruebe la velocidad de refresco de pantalla. A menos que su dispositivo de pantalla está diseñado para apoyar estéreo a una menor frecuencia de actualización, asegúrese de seleccionar una frecuencia de refresco de 100 Hz o superior (120 Hz).



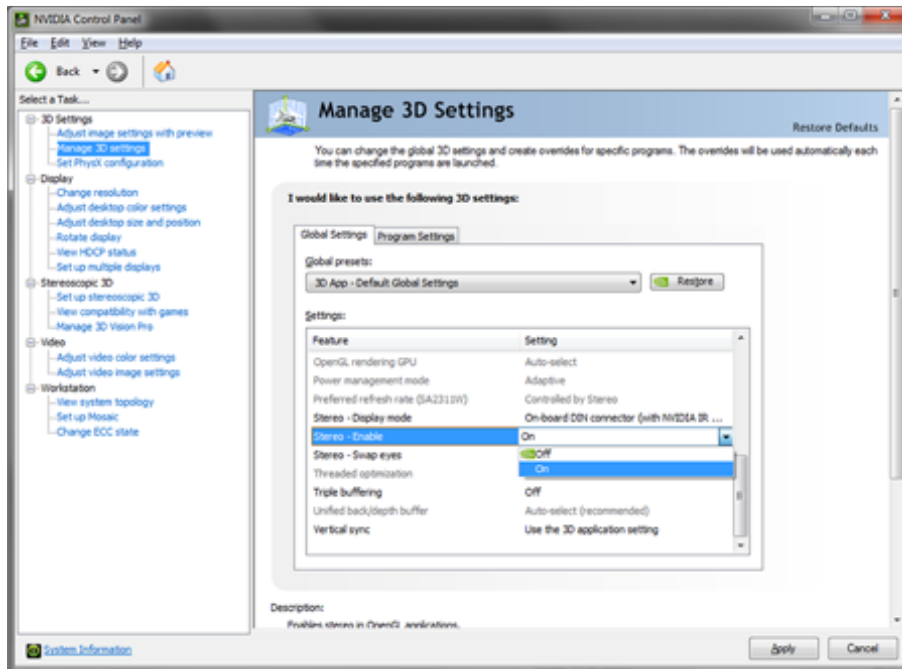
Dependiendo de la iluminación del entorno puede experimentar una pulsación de la imagen cuando se mira a través de los lentes. Si se nota esto, se puede ajustar la frecuencia de actualización a una alineación e incluso en múltiples con su frecuencia de actualización de iluminación (por ejemplo, la iluminación de 60 Hz -> 120 Hz actualización de la exhibición, iluminación 50Hz -> 100Hz de refresco de pantalla).

2. Seleccione el formato de estéreo de salida en Administrador de configuración 3D. Seleccione el formato de salida correcto para su dispositivo.



- **Los paneles LCD o proyectores de Canal Individual:** On-Board DIN Connector (with NVIDIA 3D Vision) or Active Stereo (with NVIDIA 3D Vision).
- **Dual Pipe / Modo Clon:** nView Clone Mode (with NVIDIA 3D Vision using external DIN signal).
- **Proyección real DLP de TV:** 3D DLP display (with NVIDIA 3D Vision)

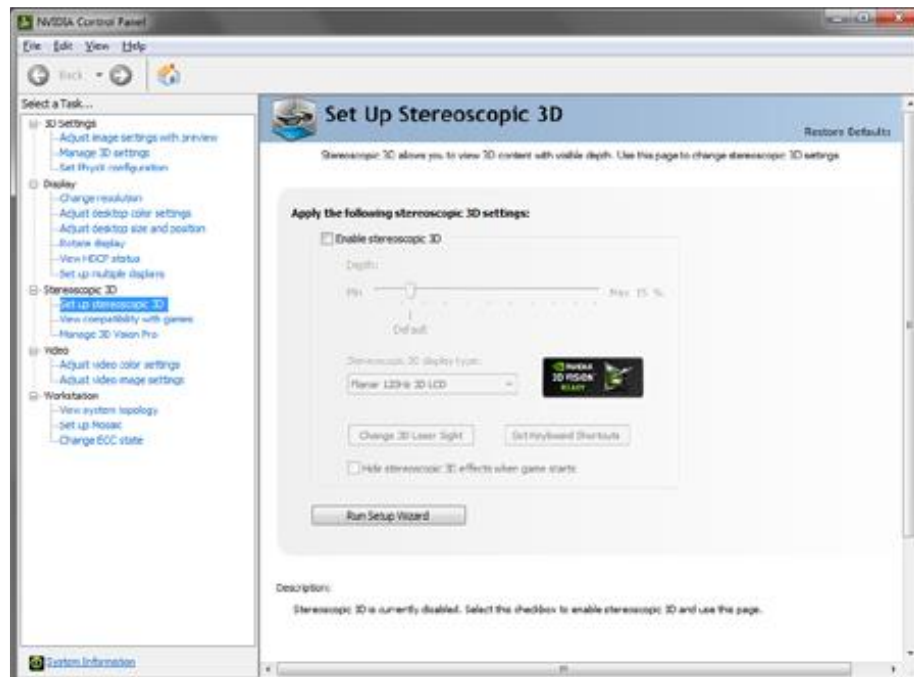
3. Habilitar el Estéreo. Encienda el estéreo para aplicaciones compatibles.



CONFIGURACION DE DIRECTX ESTEREO

El conductor estéreo DirectX no necesita estar habilitado. Cuando permita habilitar la compatibilidad de aplicaciones, incluyendo juegos y aplicaciones profesionales compatibles.

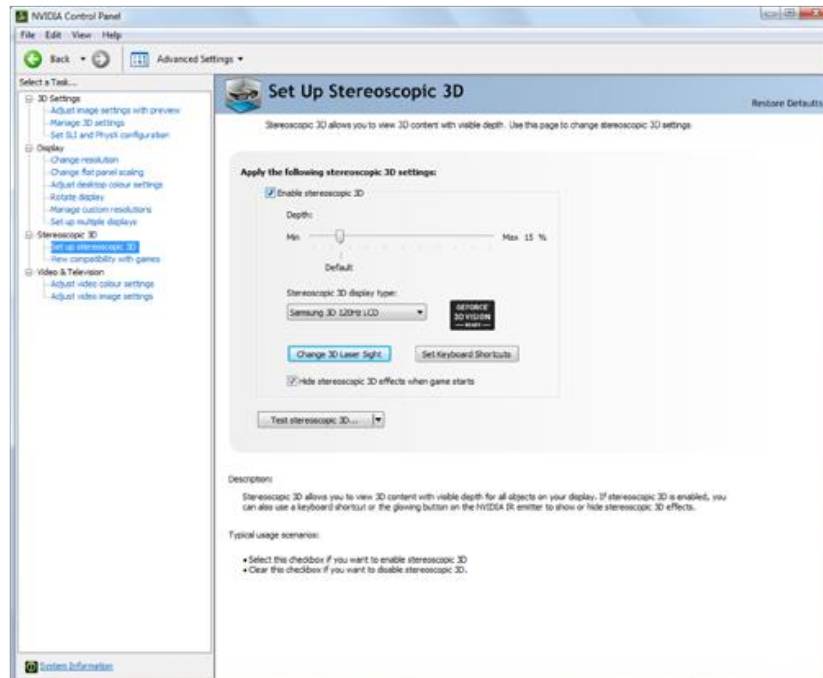
1) Ejecute el Asistente de Configuración.



NVIDIA proporciona herramientas para personalizar la experiencia 3D Vision para aplicaciones basadas en DirectX. Para acceder a los menús de herramientas, ir al panel de control de NVIDIA.

2) Acceso al panel de control de NVIDIA

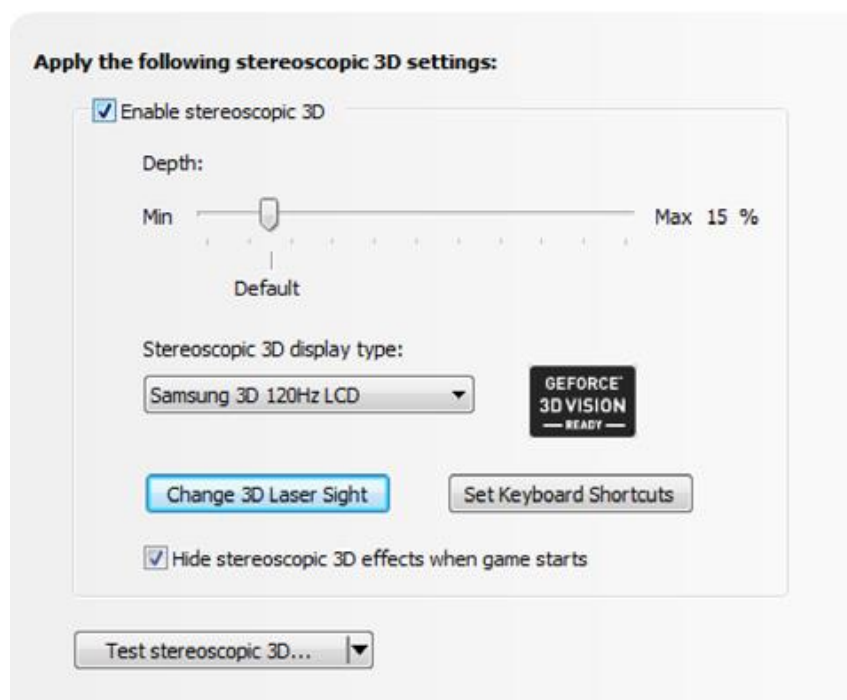
Para abrir el panel de control de NVIDIA, haga click derecho en el escritorio y seleccione el panel de control de NVIDIA. También puede ir a Inicio->NVIDIA Corporation -> NVIDIA panel de control. Seleccione estereoscópica 3D a partir de los temas en el panel izquierdo de la ventana.



Es posible que tenga que hacer clic en el signo + delante de 3D estereoscópico.

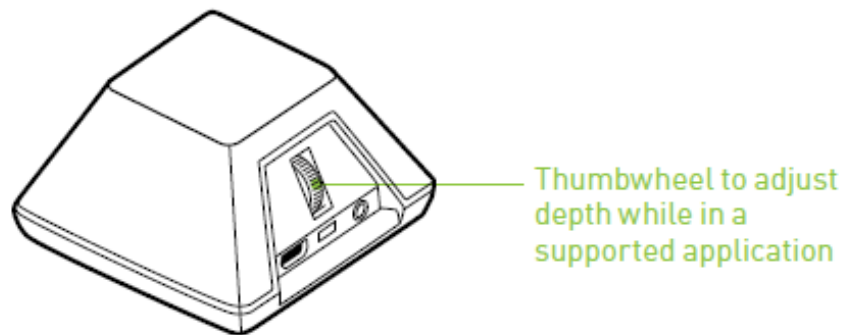
3) Configuración del Controlador del DirectX 3D estereoscópico

Esta sección del panel de control de NVIDIA proporciona todos los ajustes necesarios para mejorar su experiencia. Cada uno de los ajustes de esta pantalla se comenta.



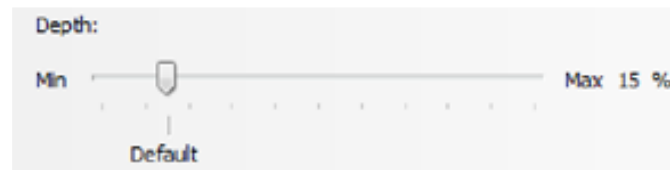
Ajuste de la profundidad

La cantidad de profundidad en una aplicación es la profundidad que el objeto más lejano se coloca en una escena. Cuando el 3D Vision es instalado por primera vez en el sistema, la profundidad se establece en el valor predeterminado de 15%. Para las aplicaciones 3D, el 15 % es un buen punto para comenzar. La visualización de la profundidad a un ajuste más alto puede ser incómodo para algunos usuarios. A medida que utilice los 3D Vision, sus ojos se convertirán más acostumbrados a la visión 3D estereoscópica y puedes gradualmente aumentar la cantidad de profundidad y sin fatiga visual. Establecer la configuración 3D estereoscópica en la pantalla, permite ajustar la cantidad de la profundidad ajustándola con el control deslizante en la cantidad deseada. El ajuste de profundidad en este control deslizante indica la cantidad de profundidad para todas las aplicaciones compatibles cuando se ponen en marcha. También puede cambiar la cantidad de profundidad de mediante el ajuste de la rueda en la parte posterior del Hub. Cualquier ajuste que realice en la ruedecilla se refleja en el control deslizante de profundidad en el NVIDIA panel de control.



Otra forma de cambiar la profundidad es usando las teclas de acceso directo del teclado.

- Para disminuir la profundidad, use Ctrl + F3 (teclas de acceso directo por defecto).
-
- Para aumentar la profundidad, use Ctrl + F4 (teclas de acceso directo por defecto).



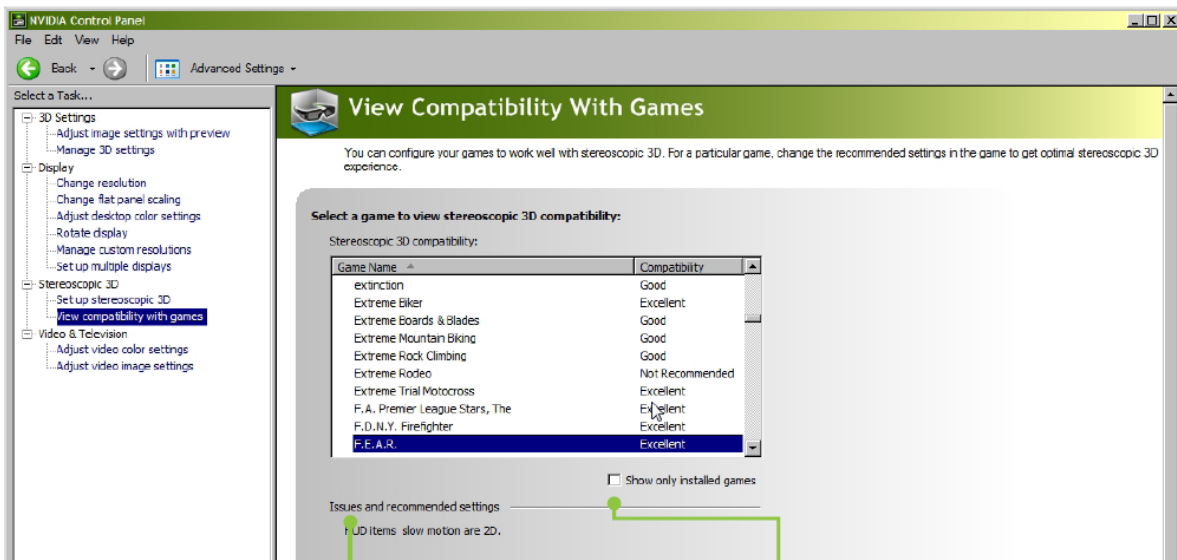
Cualquier ajuste que realice utilizando los atajos de las teclas es reflejada en el control deslizante de la profundidad en el panel de control de NVIDIA.

4) Estereoscópica 3D Tipo de pantalla

El tipo de visualización estereoscópica 3D está en una lista desplegable que contiene todos los Vision 3D que fueron encontrados por el Asistente de configuración. Esta lista es generada por el asistente. Si tuviera más de un 3D Vision conectado en pantalla durante la instalación, se reflejará en la lista.

COMPATIBILIDAD

NVIDIA tiene una prueba y proporciona perfiles de controladores para muchos juegos para asegurarse del funcionamiento correcto con GeForce 3D Vision. Todos los juegos que han sido probados se muestran en el panel de control de NVIDIA con su clasificación de compatibilidad. Para abrir el Panel de control de NVIDIA, haga clic derecho en el escritorio y seleccione el Panel de control de NVIDIA. También puede ir a Inicio > NVIDIA Corporation > Panel de control de NVIDIA. Seleccione Ver la compatibilidad con los juegos de los temas en el panel izquierdo de la ventana. Esta opción muestra una lista de todos los juegos que han sido probados por NVIDIA utilizando el software 3D estereoscópico. Cada juego ha sido calificado con Excelente, Bueno, Regular, no se recomienda. Estas calificaciones se basan en lo bien que los juegos funcionan en 3D estereoscópico y si hay algún problema en el juego que pueden interferir con su experiencia GeForce 3D Vision. Para ver un partido en la mejor calidad posible, debe seguir los temas y los ajustes recomendados señaladas por NVIDIA. No seguir estas opciones podría dar lugar a que el juego no se ejecuta en 3D estereoscópico.



Displays the recommended settings for the game selected in the list. Use the shortcut keys **Ctrl+Alt+Insert** to display the information as an overlay while in the game.

Check this box to see only those games loaded on your system. Unchecked, all games tested by NVIDIA are listed.