



INSTITUTO POLITÉCNICO

NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

TRABAJO TERMINAL II

“Sistema de realidad virtual del cuerpo humano para el estudio del sistema digestivo”

Autor:
Almendarez Perdomo Rodrigo

Directores:
M. en Ing. Moscoso Malagón Yosafat
M. en C. Saucedo Delgado Rafael Norman

Para obtener el título de

Ing. en Sistemas Computacionales

19 de agosto de 2020

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y, por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Agradecimientos

En primer lugar quiero agradecer a mis directores, M. en Ing. Moscoso Malagón Yosafat y el M. en C. Saucedo Delgado Rafael Norman, quienes con sus conocimientos y apoyo me guiaron a través de cada una de las etapas de este proyecto para alcanzar los resultados que buscaba, después de todo el desarrollo de un Trabajo Terminal es el pináculo de la formación de un alumno en la carrera de Ingeniería en Sistemas Computacionales, el cual requiere de mucha dedicación y conocimientos para llegar a su desarrollo exitoso. Especialmente quiero agradecer al profesor Alfredo Rangel Guzmán por sus enseñanzas sobre la vida y el actuar del mundo actual, y el recibir su apoyo incondicional. También quiero agradecer a la Escuela Superior de Cómputo por brindarme todos los recursos, herramientas y conocimientos que fueron necesarios para llevar a cabo el proceso de investigación y desarrollo del Trabajo Terminal. No hubiese podido arribar a estos resultados de no haber sido por su incondicional ayuda. Por último, quiero agradecer a todos mis compañeros y amigos, principalmente a los que me han acompañado desde el inicio de mi carrera y a mi familia y pareja, por apoyarme aún cuando mis ánimos decaían. En especial, quiero hacer mención de mis padres, que siempre estuvieron ahí para darme palabras de apoyo y un abrazo reconfortante para renovar energías y continuar trabajando para finalizar este camino. Muchas gracias a todos.

Rodrigo Almendarez Perdomo

Índice general

Capítulo 1

Introducción

1.1. Introducción

Este documento es el reporte técnico final del trabajo terminal titulado “Sistema de realidad virtual del cuerpo humano para el estudio del sistema digestivo” con número de registro TT: 2019-A104. En el presente capítulo se habla del problema identificado, por qué se considera como tal y cómo es que se ayudó a resolver el problema planteado mediante la ingeniería en sistemas computacionales. También se menciona que se obtiene al concluir con este trabajo terminal, tales como el prototipo del sistema. En el capítulo **II Análisis** se mostrarán todos los diagramas y documentos generados al analizar y generar un diseño del sistema que se estará desarrollando.

Aquí se encuentra la arquitectura general del sistema, y los modelos gráficos de apoyo presentados en el Análisis Estructurado Moderno.

En el capítulo **III Diseño** se describe el trabajo generado en el desarrollo del documento hasta el mes de mayo de 2020 para TT2.

En el capítulo **IV Verificación y Pruebas** se muestran pruebas hechas sobre las implementaciones del sistema siguiendo un guión para la prueba.

En el capítulo **V Conclusión** se muestran los resultados obtenidos y experiencias para mejorar el proceso, así como la vertiente para continuar con el trabajo y las conclusiones del integrante. Finalmente, se encuentran las referencias de todos los recursos empleados para dar soporte y estructura a este Trabajo Terminal, y en los apéndices se anexan elementos extra que dan información más detallada sobre lo que aquí fue realizado.

1.2. Detección del problema

Durante 2019 se entrevistó al Dr. Rios Macias jefe del área de morfología de la Escuela Superior de Medicina del Instituto Politécnico Nacional y comentó que “los medios que se utilizan para el estudio del cuerpo humano principalmente son medios impresos tradicionales, así como el uso de cuerpos para su disección y análisis posterior”.

El uso de cuerpos para su disección tiene un alto costo que incluye el mantenimiento del cuerpo en las instalaciones, el mantenimiento de las instalaciones, y la inhumación de los cuerpos.

1.3. Propuesta de Solución

Se elaboró un sistema de realidad virtual del sistema digestivo del cuerpo humano que permite interactuar con modelos tridimensionales. La intención es sentar las bases para un sistema de apoyo al aprendizaje que sea más práctico [?], sin sustituir a ningún método de estudio tradicional.

1.4. Justificación

El sistema tiene los siguientes beneficios para docentes y alumnos:

- Fuentes confiables. Se usaron textos médicos y sitios web especializados
- Uso de realidad virtual[?].

El sistema es un ejemplo de cómo se pueden actualizar las herramientas educativas, como las que se pueden encontrar en Statista [?], y que representan un mercado de aproximadamente 18.8 mil millones de dólares para el año 2020. Además, Nielsen[?] muestra los datos sobre la expectativa de adopción de la realidad virtual y aumentada en diferentes continentes, como se muestra el Figura ?? y ??.

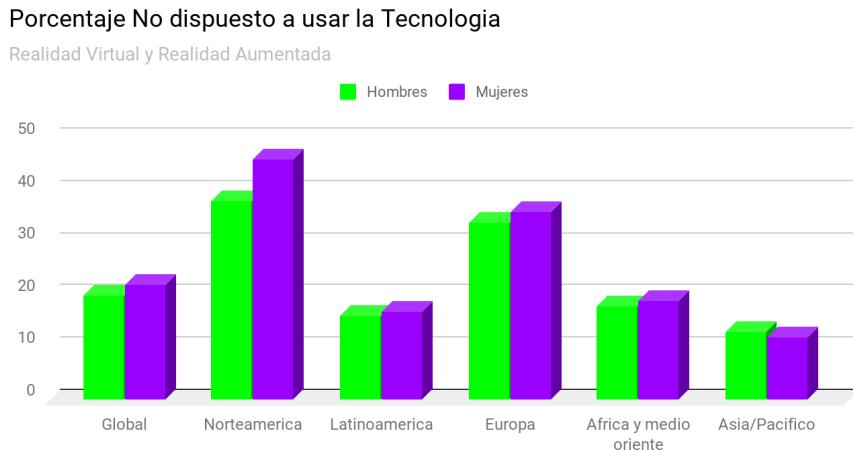


Figura 1.1: Disposición de los consumidores a nivel mundial para usar la realidad virtual y aumentada si esta se encuentra disponible en los próximos 2 años (2020 -2021)

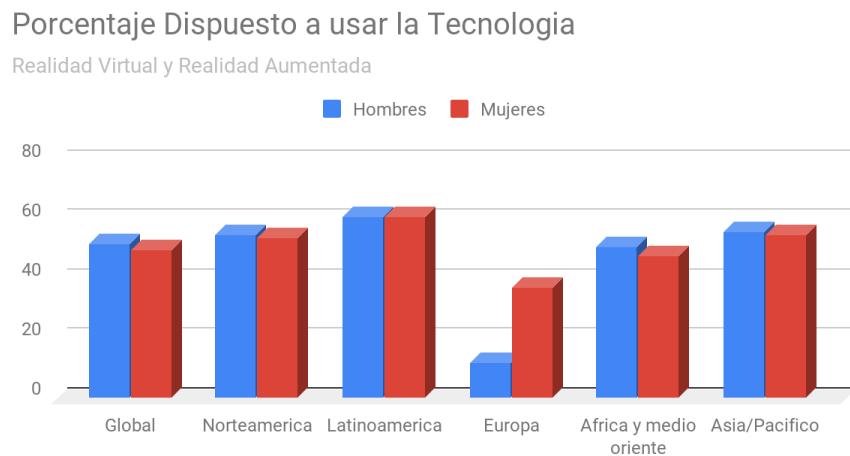


Figura 1.2: Disposición de los consumidores a nivel mundial para no usar la realidad virtual y aumentada si esta se encuentra disponible en los próximos 2 años(2020 -2021)

A medida que crece el consumo de software, multimedia y videojuegos se aumentará la disponibilidad de sistemas como el que se propone en este trabajo terminal.

1.5. Objetivos del Trabajo

Objetivo es analizar, diseñar, desarrollar y probar un sistema de demostración que utiliza la tecnología de Realidad Virtual, para ofrecer una experiencia orientada al estudio de la anatomía y morfología del cuerpo humano, específicamente del sistema digestivo.

1.5.1. Objetivos Específicos

Para lograr el objetivo se identificaron los siguientes objetivos específicos:

- Investigar en fuentes confiables sobre el sistema digestivo.
- Analizar, diseñar, desarrollar y probar un entorno de Realidad Virtual para la interacción.

1.6. Población Objetivo

De acuerdo a un estudio de Ericsson ConsumerLabe[?], en el año 2020 un tercio de los consumidores serán usuarios de Realidad Virtual. Ya desde 2017 se había explorado el nivel de interés de los consumidores en Realidad Virtual[?] con su potencial de reunir gente de todo el mundo y crear una experiencia más profunda, personalizada y enriquecida.

De todos los interesados en la Realidad Virtual se toma a los estudiantes de educación superior del

área de medicina, particularmente a 180[?] estudiantes de la Escuela Superior de Medicina del Instituto Politécnico Nacional de entre 18 y 24 años de edad. Además se espera que los usuarios objetivo tengan las siguientes características[?] específicamente:

- Disposición para el aprendizaje con nuevas tecnologías de la información.
- Conocimientos sólidos en las áreas de biología, física, química; y en forma idónea conocimientos básicos de las etimologías grecolatinas e idioma inglés, que le facilitarán la comprensión y dominio de los conceptos utilizados en las asignaturas básicas y clínicas.
- No ser propenso a sufrir cinetosis.

1.7. Productos logrados

Se logró crear un software (demostración), de una experiencia demostrativa en realidad virtual. El sistema muestra características y elementos del sistema digestivo y permite la participación de un usuario utilizando una misma computadora de forma local. Se usan los controles Oculus Touch ® así como el visor de Realidad Virtual Oculus Rift ®. También se redactó el presente reporte técnico.

1.8. Definiciones

Estas son las definiciones más importantes para el presente trabajo terminal

1.8.1. Realidad Virtual

Algunos autores definen así la Realidad Virtual.

“La realidad virtual (RV) es una simulación tridimensional generada o asistida comúnmente por computadora de algún aspecto del mundo real o ficticio, en el cual el usuario tiene la sensación de pertenecer a ese ambiente sintético o interactuar con él”[?]

Corrado Padila Érica

“Realidad Virtual: gráficos 3D en entornos inmersivos que usan I/O artefactos como guantes, cascos, etc. en busca de mayores grados de iteración con el ambiente virtual”[?]

Lozano Miguel, Calderón

Realidad Virtual es una forma en que los seres humanos puedan visualizar, manipular e interactuar con las computadoras y datos extremadamente complejos”.[?]

Isdale, Jerry

“Un sistema interactivo capaz de crear una simulación que implique a varios de los sentidos del ser humano, generados por una computadora, explorable, visualizable y manipulable en tiempo real; este bajo la forma de imágenes y sonidos, estos, dando la sensación de presencia en el entorno generado”[?]

Levis, Diego

Esta última ha sido la definición que se ha tomado para el desarrollo del proyecto del trabajo terminal, asimismo se puede concluir que todos los autores coinciden en que la realidad virtual es un

un mundo simulado en el que el usuario puede interactuar en tiempo real por medio de dispositivos o computadoras que logran un efecto artificial e inmersivo en el que se pueden manipular objetos.

1.8.2. Producto Multimedia

Los productos multimedia se pueden clasificar en dos categorías: productos interactivos y no interactivos. Los productos no interactivos también se pueden clasificar en productos estáticos como carteles, logotipos, folletos, modelos estáticos 3D, etc., y productos basados en el tiempo. [?, ?].

Los productos multimedia interactivos son aplicaciones de software que contienen productos multimedia[?] (es decir, aplicaciones basadas en eventos como juegos, aplicaciones web basadas en multimedia y materiales de aprendizaje multimedia basados en interactividad). La figura ?? a continuación ilustra los tipos de productos multimedia.

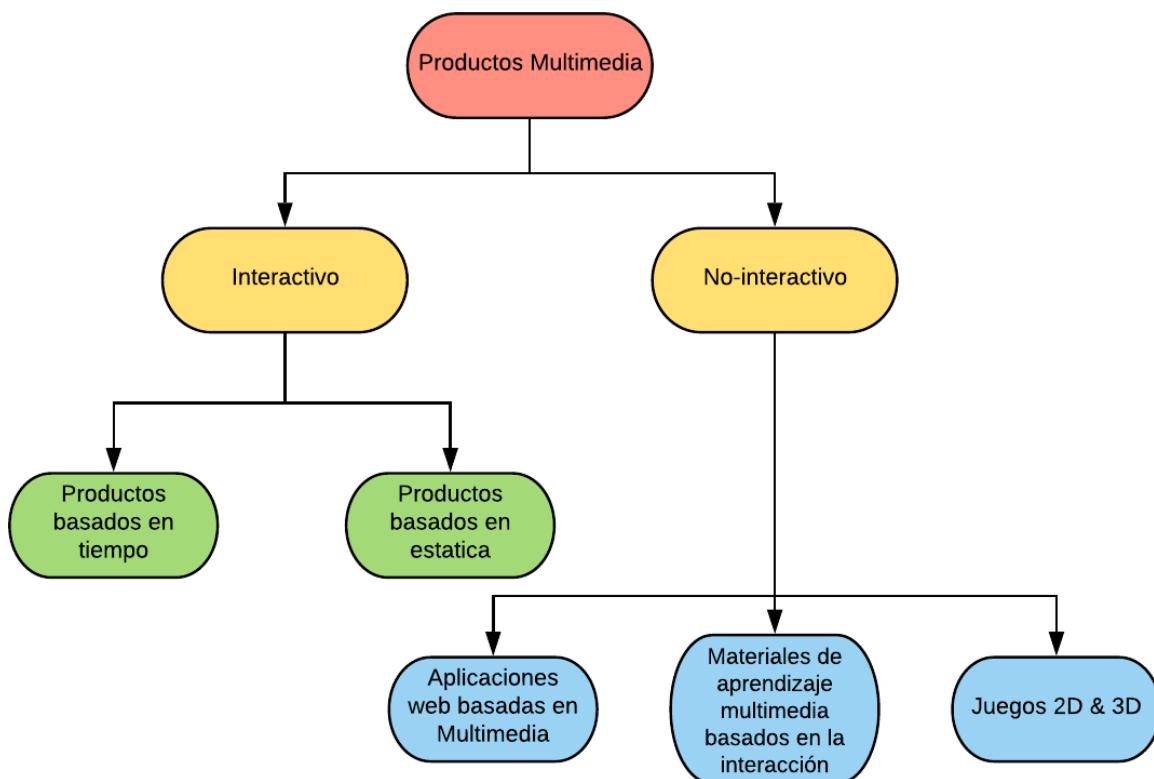


Figura 1.3: Tipos de productos multimedia

Un sistema multimedia se puede analizar desde tres puntos:

- La vista externa. La forma en que el usuario interactúa con el producto.
- Flujo de acciones. El orden en que se muestran los marcos de los modelos a los usuarios. [?, ?].
- Los roles de los usuarios. La interacción con productos multimedia.

El cuadro ?? resume los tipos de productos multimedia y sus características.

Tipos de productos multimedia	Características Multimedia		
	Vista	Flujo	Roles de los Usuarios
<i>Productos estáticos</i>	Un cuadro	Sin acciones	Pasivo (ver/leer)
<i>Productos basados en tiempo</i>	Múltiples cuadros	Secuencia	Pasivo(mirar)
<i>Productos interactivos</i>	Impulsado por eventos	Secuencia, Selectiva, Interactiva e Impulsada por eventos	Activo(Realiza eventos)

Cuadro 1.1: Características de los productos multimedia.

Para este trabajo, el sistema que se propone es una producto interactivo.

1.9. Estado del Arte

A continuación se muestran algunos de trabajos académicos desarrollados en México y fueron comparados con el trabajo planeado. Como comparativa y de forma ilustrativa del sector académico.

1. TT No. 2014-A058 “Sistema para la orientación de los efectos sobre la espalda humana en pacientes con sobrepeso”[?]
2. TT No. 2012-B055 “Laboratorio Virtual del cuerpo humano 3D con asistente de ayuda en línea para el nivel superior bajo el paradigma de Educación Basada en Web con tecnologías de Web Semántica”[?]
3. TT No. 2014-B035 “Simulación en Tercera Dimensión del Sistema Circulatorio de los Cánidos para el uso Educativo”[?]
4. TT No. 2014-B039 “Simulación de una Línea del Metro con Realidad Virtual”[?]
5. Tesis que para optar por el grado de Maestro en Ciencia e Ingeniería de la Computación, Sistema de seguimiento de movimiento de las extremidades superiores basado en sensores iniciales para rehabilitación en realidad virtual.[?]
6. Adecuación educativa de la realidad virtual como herramienta didáctica para el proceso enseñanza-aprendizaje / tesis que para obtener el título de Licenciado en Pedagogía, presenta María de la O García Noriega; asesor Lucina Moreno Valle Suárez.[?]

Así mismo se ha encontrado software propietario desarrollado por empresas privadas los cuales son los siguientes.

- The Body VR: Anatomy Viewer es la única herramienta de visualización de Realidad Virtual disponible en el mercado que se basa en datos médicos específicos del paciente (por ejemplo, MRI, CT, PET) y cumple con los estándares DICOM. Proporciona simulaciones de R.V. anatómicas

en tiempo real para visualizar diagnósticos médicos, ilustrar el impacto de los procedimientos y tratamientos, y crear una toma de decisiones más educada.



Figura 1.4: Software “The Body VR” en uso.

- Anatomyou VR: Estructuras anatómicas fotorrealistas, modeladas en colaboración con RenderArea, validadas por expertos clínicos y certificadas por personal capacitado en Tecnologías Médicas de la Universidad de Las Palmas de Gran Canaria.



Figura 1.5: Pancarta promocional de “Anatomyou VR”

- Biodigital Anatomy: El cuerpo tridimensional más completo, científicamente preciso e interactivo jamás ensamblado. Anatomía masculina y femenina, en los detalles básicos (gratuitos) y profesionales. Cada sistema está completamente segmentado, etiquetado y direccionable para una fácil configuración que satisface cualquier necesidad educativa.

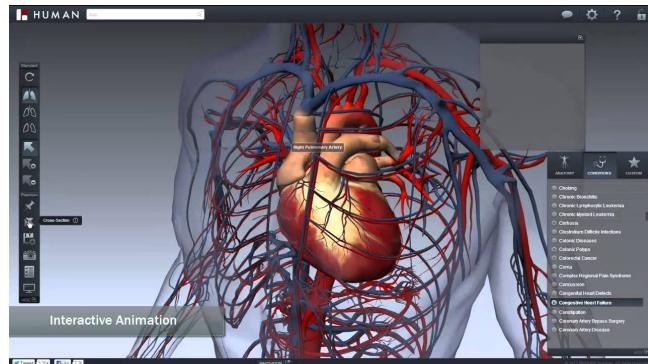


Figura 1.6: Interfaz del software de Biomedical Anatomy

- 3D Organon VR Anatom: 3D Organon es un completo atlas anatómico que presenta los 15 sistemas del cuerpo humano. Incluye más de 4,000 estructuras y órganos anatómicos realistas y más de 160 correlaciones clínicas encontradas por sistema del cuerpo.



Figura 1.7: Interfaz del software de 3D Organon VR Anatom

1.10. Metodologías

Se siguió como modelo de desarrollo a la metodología OpenUp[?], que está basada en RUP. OpenUP es extensible y se usó como base, ya que se adaptó al desarrollo de este proyecto. La figura ?? muestra el ciclo de vida iterativo de OpenUP.

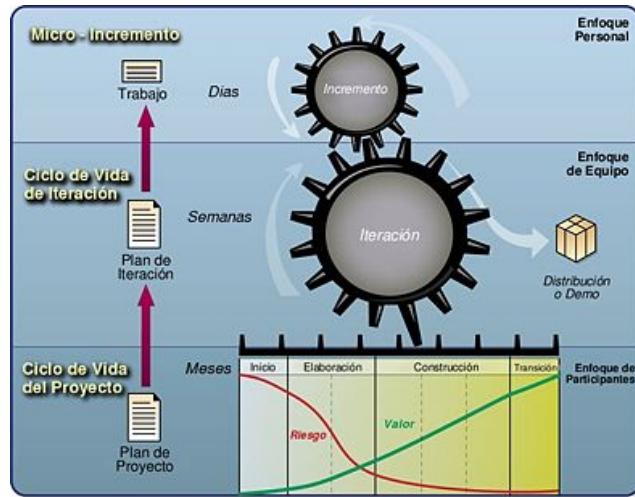


Figura 1.8: Ciclo de Vida Iterativo de OpenUP

OpenUP está organizado en contenido y proceso, con cuatro fases concepción o creación del proyecto, preparación o elaboración detallada del proyecto, construcción y transición. Además también se consideró la Metodología de ingeniería de software multimedia que tiene como fases, requisitos y preproducción, diseño y producción, validación y Postproducción y evolución, como se puede ver en la figura ??

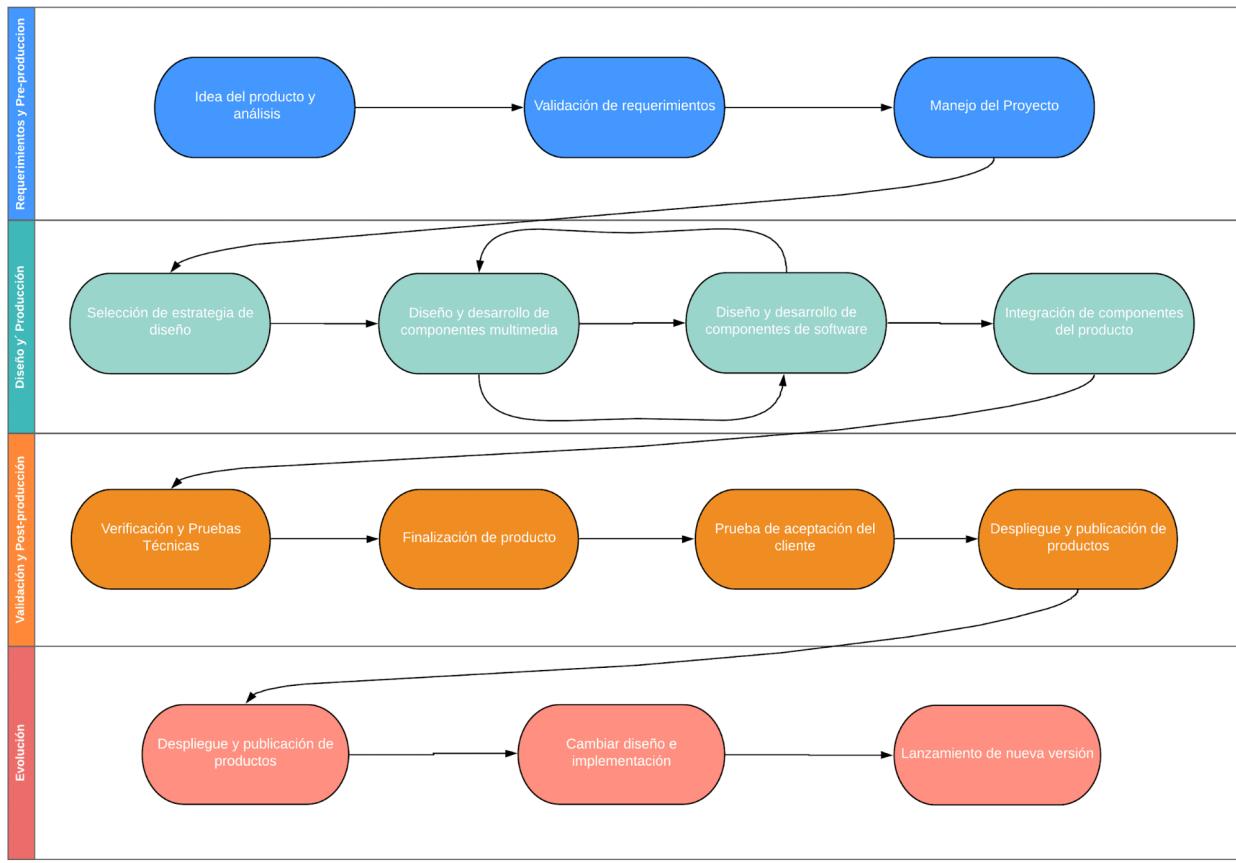


Figura 1.9: Fases de la Metodología de ingeniería de software multimedia

También se tomó como referencia a Git-flow[?] que es un modelo de ramificación que consiste en separar cada característica en su propia rama de desarrollo. Se utiliza una versión simplificada del modelo[?] compuesta por tres tipos principales de ramas.

- **master:** El código fuente refleja un estado listo para producción.
- **develop:** La última entrega con cambios de desarrollo.
- **feature:** Nuevas características para el próximo lanzamiento.

Otros factores a considerar son que tomando en cuenta el libro Análisis Estructurado Moderno de Edward Yourdon “debe usarse cualquier modelo que se adapte a la situación en la que se encuentra.”[?]. Aunque la metodología de ingeniería de software multimedia no propone un diagrama específico como modelo gráfico para la visualización del proceso general de la realización del sistema de forma general, para este trabajo se considera apropiado emplear diagramas adicionales a manera de tener una mejor visión del proyecto.

- **Diagramas de Flujo de Datos:** Estos nos permiten comprender cómo se mueven y procesan los datos en el sistema, visualizando el sistema como una red de procesos.
- **Diagramas de Flujo.** Describe gráficamente la lógica de un procedimiento específico.

Capítulo 2

Análisis

2.1. Sistema operativo y plataforma de desarrollo

EL cuadro ?? es la comparativa de las plataformas que se analizaron para la elaboración del trabajo terminal.

Descripción	Software	Plataforma Windows		Operatividad
		Costo		
Sistema Operativo	Windows 10 Pro	MXN\$5,199.00		Ideal para pequeñas empresas o usuarios que necesiten una funcionalidad mejorada.
Sistema Operativo	Windows 10 Home	MXN\$3,599.00		Ideal para uso personal o doméstico.
Sistema Operativo	Windows 10 Pro para Workstations	MXN\$7,899.00		Ideal para los usuarios avanzados y pequeñas empresas que buscan funcionalidad mejorada con la capacidad de calcular cargas de trabajo intensivas.
Sistema Operativo	Windows Server 2019	MXN\$10,814.69		Sistema operativo que une los entornos on-premises con Azure y agrega capas adicionales de seguridad a la vez que te ayuda a modernizar tus aplicaciones e infraestructura.

Cuadro 2.1: Sistemas operativos de plataforma Windows

El cuadro ?? muestra la comparativa de las plataformas de desarrollo que se analizaron para la elaboración del trabajo terminal.

Plataforma de desarrollo			
Descripción	Software	Costo	Operatividad
Motor de Desarrollo	Unity Free	Gratis	Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas
Motor de Desarrollo	Unreal Engine 4	Gratis	Unreal Engine es un motor de juego creado por la compañía Epic Games, mostrado inicialmente en el shooter en primera persona Unreal en 1998. Aunque se desarrolló principalmente para los shooters en primera persona, se ha utilizado con éxito en una variedad de otros géneros.

Cuadro 2.2: Plataforma de desarrollo.

Tomando en cuenta la infraestructura que posee la Escuela Superior de Medicina del Instituto Politécnico Nacional, específicamente el Departamento de Computación se optó por usar el sistema operativo Windows 10 Pro y Unity ®.

2.2. Hardware

De acuerdo a las especificaciones del software seleccionado se recomienda usar:

- Tarjeta Gráfica: NVIDIA GTX 1060 / AMD Radeon RX 480 o mejor
- Tarjeta Gráfica Alternativa: NVIDIA GTX 970 / AMD Radeon R9 290 o mejor

- CPU: Intel i5-4590 / AMD Ryzen 5 1500X o mejor
- Memoria: 8GB+ RAM
- Salida de Video: Compatible HDMI 1.3 video output
- USB puertos: 3x USB 3.0 y 1x USB 2.0

Y en caso de no contar con estos elementos, el mínimo es:

- Tarjeta Gráfica: NVIDIA GTX 1050Ti / AMD Radeon RX 470 o mejor
- Tarjeta Gráfica Alternativa: NVIDIA GTX 960 / AMD Radeon R9 290 o mejor
- CPU: Intel i3-6100 / AMD Ryzen 3 1200, FX4350 o mejor
- Memoria: 8GB+ RAM
- Salida de Video: Compatible HDMI 1.3 video output
- Puertos USB: 1x USB 3.0 y 2x USB 2.0

2.3. Viabilidad

También se analizó la factibilidad del proyecto en general. Desde el punto de vista técnico se realizó una evaluación de la tecnología actual existente y la posibilidad de utilizarla en el desarrollo del sistema. Además de DirectX versión 11 el cuadro ?? muestra los recursos técnicos necesarios para la ejecución correcta del software:

Cantidad	Recursos	Características
1	Computadora Personal de Escritorio	Tarjeta gráfica discreta, RX480 Memoria RAM 16 Gb 5 Puertos USB Procesador de 4 núcleos o mayor
1	Sistema de realidad Virtual Oculus Rift	Visor HMD, controles Touch Sensores Touch.

Cuadro 2.3: Recursos Técnicos.

Económicamente, se determinaron los recursos para desarrollar el sistema así como la comparativa con el uso de cuerpos para su examinación y estudio. Después de un análisis e investigación de los costos con la dirección del Área de Morfología en la Escuela Superior de Medicina bajo la asesoría del Dr. Macias Rios se determinó que el costo que se tiene para el traslado, mantenimiento, uso e inhumación de los cuerpos es de \$40,000.00 c/u, como se ve en el cuadro ??.

Costo de uso de cuerpos.	
Traslado, mantenimiento, uso e inhumación	\$40,000.00 c/u
Total	\$40,000.00 c/u

Cuadro 2.4: Costo de cálculo de uso de cuerpos.

En el caso del desarrollo e implementación del proyecto se consideró la depreciación, como se observa en el cuadro ??.

Depreciaciones del Proyecto								
Equipos de Cómputo				Depreciación				
Cantidad	Equipos	Monto original de Inversión	Valor actual del equipo	Valor a depreciar	% anual	% mensual	Depresión mensual	Depreciación anual
1	Computadora de escritorio armada	\$25,054.63	\$20,000.00	\$5,054.63	33.33 %	2.78 %	\$ 140.52	\$ 1,545.72
1	Laptop HP	\$9,999.00	\$6,999.00	\$3,000.00	33.33 %	2.78 %	\$ 83.40	\$ 917.40
				Total:				\$ 2,463.12

Cuadro 2.5: Depreciaciones del proyecto.

Para ofrecer una experiencia aceptable al momento del uso del equipo de Realidad Virtual y el software se proponen los elementos del cuadro ??.

Pendiente tabla costos

Además, el sistema de Realidad Virtual con sus componentes tiene un costo que se muestra en el cuadro ??.

Sistema de Realidad Virtual	
Producto	Producto
Visor Oculus Rift	Incluido en el paquete
Controles Touch Oculus x 2	Incluido en el paquete
Sensores Oculus x 2	Incluido en el paquete
Anexos	Incluido en el paquete
Total:	\$ 8,821.74

Cuadro 2.6: Costos y contenido del sistema de Realidad Virtual.

Se estimaron los sueldos de programador y modelado, como se observa en el cuadro ??.

Sueldos				
Puesto	Sueldo Mensual individual	Cantidad de personal	Sueldos mensuales totales	6 meses
Programador	\$25,296.00	1	\$25,296.00	\$151,776.00
Modelador 3D	\$25,296.00	1	\$25,296.00	\$151,776.00
Total				\$303,522.00

Cuadro 2.7: Cálculo de Sueldos.

Servicios		
Concepto	Mensual	11 Meses
Luz (kw Consumidos por costo Unitario)	\$430	\$4,730
Agua (Lt consumidos por costo unitario)	\$200	\$2,200
Teléfono e Internet (renta mensual fija)	\$ 450	\$4,850
Total:		\$11,780

Cuadro 2.8: Cálculo de Costo por Servicios.

Los servicios estimados se muestran en el cuadro ?? y en el cuadro ?? se muestra la suma total y como resultado se obtiene el costo total del proyecto, que se estima en: \$326,316.86.

Costos del Proyecto	
Concepto	Costo
Servicios	\$ 11,780
Sueldos	\$303,522.00
Depreciaciones	\$2,463.12
Equipo extra.	\$ 8,821.74
Total	\$ 326,316.86

Cuadro 2.9: Costos finales del proyecto

En resumen, el costo de usar nueve cuerpos sería de \$360,000.00 y el del proyecto de \$326,318.00, con lo cual se puede considerar viable económicoamente.

2.4. Análisis de la plataforma Unity ®

Unity ® es un motor de desarrollo de videojuegos multiplataforma desarrollado por Unity ® Technologies. Con él se pueden crear videojuegos, simuladores, software de realidad virtual y aumentada. Puede generar código para computadoras de escritorio, portátiles, consolas de videojuegos, Smart TV y otros dispositivos móviles. Ofrece una API de scripting en C#. En la figura ?? se puede ver el entorno en general.

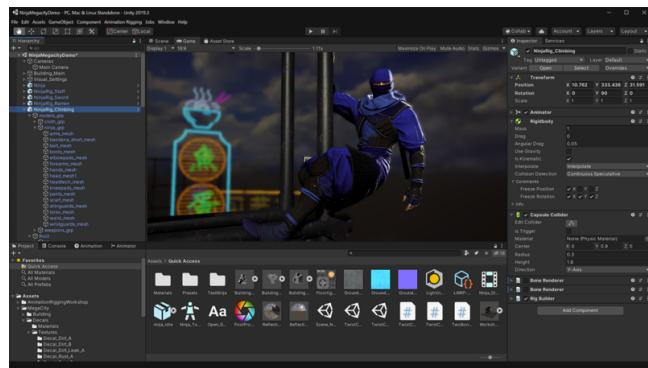


Figura 2.1: Interfaz de trabajo de Unity ®3D en 2020 [?]

2.5. Sistema de Seguimiento

Cada dispositivo rastreador tiene una “constelación” predefinida de LED infrarrojos ocultos debajo del plástico externo, que puede ver resaltado en la figura ???. La luz IR es invisible para el ojo humano.



Figura 2.2: Muestra de el diseño de la “constelación” incluida en el Oculus Rift ®

Los sensores, que son esencialmente cámaras con filtros para ver sólo la luz IR, envían cuadros a la PC del usuario a través de un cable USB a 60 Hz. La PC procesa cada cuadro, identificando la posición de cada LED IR y, por lo tanto, la posición relativa de cada objeto.

El software puede reconocer fácilmente qué LED está viendo porque conoce la forma de la “constelación”, recuerda dónde estaba el objeto en el cuadro anterior y conoce su dirección de aceleración (desde el acelerómetro) y su rotación (desde giroscopio). Cada LED IR también parpadea a una frecuencia específica para identificarse.

Estas innovaciones le provieron a “Constellation” de una ventaja sobre los sistemas de seguimiento anteriores.



Figura 2.3: Visor Oculus Rift ®, controles Oculus Touch ® y sensores Oculus Sensor ®

Para admitir el movimiento rápido, los auriculares Rift y los controladores táctiles se comunican de forma inalámbrica con un chip inalámbrico en el sensor cada vez que están a punto de pulsar sus LED. Esto permite que el obturador de la cámara se dispare exactamente como lo hacen los LED, y permite que la exposición sea corta.

2.6. Análisis de Información médica

El cuerpo humano es una estructura compleja y altamente organizada, formada por células que trabajan juntas para realizar funciones específicas necesarias para mantener la vida.[?].

La biología del cuerpo humano incluye:

- Fisiología (cómo funciona el cuerpo)
- Anatomía (cómo se estructura el cuerpo)

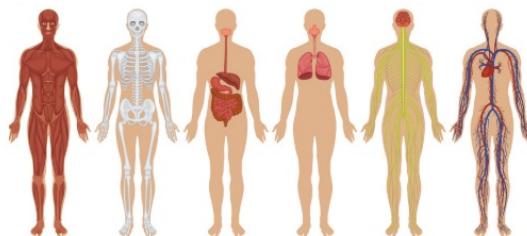


Figura 2.4: Gráfico ejemplificando los sistemas del cuerpo humano.[?]

2.6.1. Cavidad abdominal

Para el desarrollo del trabajo terminal interesa la cavidad abdominal, que es el espacio corporal que ocupa la región del abdomen, ubicada entre el diafragma y la abertura de la pelvis. Es la cavidad más grande del cuerpo humano y contiene los principales órganos del aparato digestivo, urinario y genital.

Para su estudio y evaluación clínica en el campo de la medicina, el abdomen debe ser dividido topográficamente de forma externa en 9 cuadrantes o regiones, utilizando cuatro líneas trazadas imaginariamente, dos verticales y dos horizontales.[?]

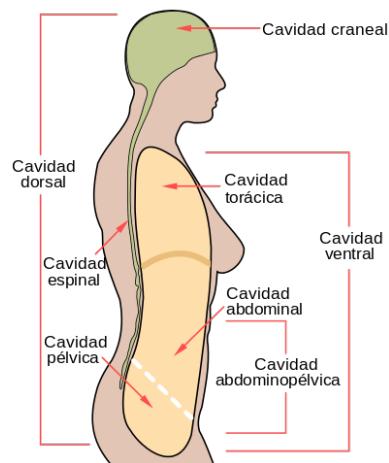


Figura 2.5: Diagrama que muestra las diferentes cavidades del cuerpo humano.

A continuación se muestra una representación del sistema digestivo, está contiene la información de cuáles son los órganos y elementos que lo conforman, estos serán varios ya que se han tomado como guía, estos elegidos mediante a una entrevista de estudiantes de medicina como material que se ha utilizado para el estudio del cuerpo humano, para la realización de modelos en 3D, asimismo, ejemplifica el material que dispone pero no limitado a el alumnado para el estudio del sistema en cuestión.

Aunado a esto se incorporan la mayoría de los elementos del sistema digestivo los cuales se encuentran en la cavidad abdominal estos expuestos por autores de los diferentes materiales en los cuales, pero no limitados a estos, se realizó la investigación del sistema.



Figura 2.6: Órganos digestivos in situ (El epiplón mayor ha sido parcialmente eliminado o reflejado)[?]

1. Hígado
2. Estómago
3. Colon transverso
4. Intestino delgado
5. Ciego con apéndice vermiciforme

Para efectos de este Trabajo Terminal solamente se desarrolló el sistema digestivo, ubicado en la cavidad abdominal, para su interacción mediante realidad virtual con el mismo.

2.6.2. Sistema Digestivo

El tubo digestivo se compone de diferentes secciones: boca, esófago, estómago, intestino delgado, intestino grueso, recto y ano. Cada una de estas estructuras tiene elementos en común y algunos otros que las diferencian entre ellas.[?]

Cabe mencionar que el sistema digestivo también incluye órganos situados fuera del tubo digestivo los cuales son, el páncreas, el hígado y la vesícula biliar. Para los efectos del trabajo terminal el sistema digestivo será resumido en los elementos contenidos en la siguiente lista:[?].

- Glándulas Salivales
- Cavidad Oral
- Faringe
- Esófago
- Estómago
- Intestino Delgado
 - Hígado
 - Páncreas
 - Vesícula Biliar
- Intestino Grueso
- Ano

2.7. Casos de Uso

El planteamiento de casos de uso en un sistema no tradicional como este puede generar dificultades al momento de expresar cuál es la actividad que podrá realizar el actor, ya que las posibilidades son virtualmente infinitas sólo limitadas por las acciones del sistema mismo, pero se pueden denotar en acciones específicas que queremos que el usuario pueda realizar en el sistema.

Ahora bien en la figura ?? se muestran los casos de uso que dan parte a el uso del software y las posibilidades generales de interacción en un software de Realidad Virtual. Aunque las interacciones pueden ser prácticamente infinitas estas se ven limitadas hasta cierto punto por las capacidades integradas dentro del software, más adelante se describirán las interacciones con los modelos 3D de los sistemas elegidos, en este caso el sistema de realidad virtual.

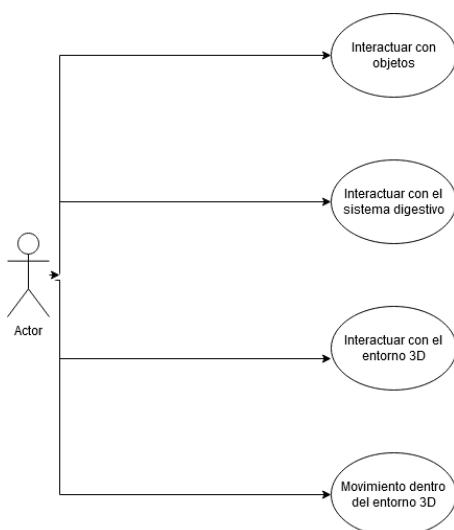


Figura 2.7: Diagrama de casos de uso del sistema

2.8. Complicaciones del desarrollo sin metodología

El proyecto comenzó su desarrollo sin tomar en cuenta la metodología de desarrollo Open UP, lo cual detonó en que el trabajo se viera mermado en cuanto a su documentación y desarrollo.

El desarrollo de un sistema mediante Open UP puede ser una herramienta poderosa para el desarrollo de software ya que dentro de ella se desarrollan micro incrementos en el desarrollo del software y además cumple con los principios del Manifiesto Ágil[?].

La problemática surge cuando se intenta implementar en un proyecto de software diferente al cual usualmente se desarrolla, por ejemplo un Sistema de Calendarización de Trabajos terminales o un Generador de Páginas Web, difiere en cuanto a su desarrollo ya que se usan herramientas y elementos diferentes, tales como el desarrollo en el motor Unity ®, el diseño de los modelos 3D y el uso de elementos propietarios de Oculus ® dificultan la tarea de documentación y desarrollo bajo lineamientos por más ligeros que sean debido a la naturaleza del proyecto.

Es por eso que se buscó como alternativa una metodología que tome en cuenta las características inherentes de un software que su núcleo está arraigado a la realidad virtual haciéndolo un software multimedia.

Debido al cambio de enfoque de desarrollo de la metodología las tareas de reingeniería son suplementadas por más tiempo de desarrollo y refinamiento de los elementos anteriores a estos dentro del cronograma de actividades.

Capítulo 3

Diseño

En este capítulo se desarrolla los apartados dedicados a la planeación del trabajo a realizar, comenzando con el diseño utilizando elementos de la metodología de Metodología de ingeniería de software multimedia y los elementos de apoyo de la metodología estructurada de Yourdon cuando estos sean necesarios para exemplificar diseño o acciones específicas del software.

3.1. Selección de estrategia de diseño

Con respecto a la realidad virtual, una de las cosas básicas a considerar es la forma en que nuestros sentidos sirven como la entrada que nuestro cerebro utiliza para construir una comprensión del mundo que nos rodea. La vista, el oído, el tacto, el olfato y el gusto son el conjunto de estímulo externo más ampliamente aceptado que percibe el cuerpo humano.

Estos sentidos y nuestras reacciones a ellos son el resultado de milenios de selección natural y hay varias consecuencias de esto incorporadas en nuestro instinto. Todo esto es un conocimiento relativamente común y parece que no es necesario reiterar aquí, pero lo importante es afirmar que nosotros, como humanos, tenemos ciertos resultados predecibles basados en ciertos conjuntos de entradas. Esencialmente, es instinto, naturaleza humana.

Un sitio web bien diseñado utilizará de manera similar el color, la distancia y la tipografía para comunicar claramente un propósito y, a menudo, persuadir algún tipo de acción.

Para que todo esto sea efectivo, se deben implementar y descubrir principios de diseño razonables. Existen varios principios para el diseño que pueden traducirse de otros medios. El diseño de impresión, el diseño web, la arquitectura, el diseño de interiores, el teatro, los gráficos en movimiento, etc., tienen elementos que pueden considerarse relevantes y adoptados.[36]

Al mismo tiempo, el medio de la realidad virtual como propiedades, como la capacidad de intersección del contenido, son únicas.

Es por esto que el diseño de un sistema de realidad virtual presenta retos los cuales son difíciles de sobrellevar ya que se tiene que crear una experiencia para el usuario en el sistema mismo lo cual conlleva a la selección de una estrategia de diseño centrada en la UX del usuario.

3.2. Requisitos para el desarrollo de software para proveer una experiencia de realidad virtual optima.

3.2.1. Los cuatro núcleos del diseño UX para RV

Se tomaron en cuenta dos consideraciones centrales para el diseño de experiencias de realidad virtual:

1. Que fuera interactivo
2. Que fuera reactivo.
3. Que fuera cómodo
4. Que fuera fácil de usar.

3.3. Flexibilidad del Diseño

La Metodología de ingeniería de software multimedia provee la libertad para elegir el desarrollo de los elementos de “Diseño y desarrollo de componentes multimedia” y “Diseño y desarrollo de componentes de software” y así alternar el desarrollo y diseño de los múltiples elementos que se utilizan dentro del proyecto. Para un documento técnico más ordenado se incluirán primeramente los componentes multimedia y posteriormente los componentes de software aunque estos hayan sido desarrollados en diferentes momentos, como la metodología da la libertad de realizarlo.

3.4. Componentes multimedia

En general, independientemente de la disciplina, el proceso de modelado es una simplificación de un objeto para su posterior estudio o representación. Así, podemos hablar de modelos matemáticos que simplifican fenómenos físicos, o modelos meteorológicos para la predicción del tiempo atmosférico, etc. Un modelo geométrico define la información sobre la forma (geometría) de un determinado objeto. Las simplificaciones que se realicen en su definición vendrán determinadas por diferentes factores como el método de representación utilizado, operadores empleados o nivel de detalle.[?]

Se puede definir el proceso de modelado geométrico tridimensional como el encargado de crear modelos consistentes que puedan ser manejados algorítmicamente en un computador. Este proceso de construcción se aborda en diferentes etapas, partiendo típicamente de entidades básicas y aplicando una serie de operadores sobre ellas. Estas entidades básicas pueden ser primitivas geométricas (calculadas de forma algorítmica o mediante una ecuación matemática) u obtenidas mediante un dispositivo de captura (escáner 3D).

Existen multitud de técnicas de modelado 3D. En una primera taxonomía de alto nivel podemos hacer una categorización dependiendo de si el modelado se centra en definir únicamente las características del contorno del objeto, los siguientes son los más usados:

- **Modelado Sólido:** también conocidos como Geometría Sólida Constructiva (CSG Constructed Solid Geometry). Los modelos sólidos definen el volumen del objeto que representan, y en muchos casos indican incluso el centro de masas, la densidad del material interna, etc. Se utilizan en fabricación por computador y en aplicaciones médicas e industriales.
- **Modelado de Contorno:** también conocidos como Representación de Contorno (B-Rep - Boundary Representation). Los modelos de contorno únicamente representan la superficie límite del objeto (de forma conceptual, la “cáscara”). Son más fáciles de definir y modificar. Además, lo interesante para la representación del objeto es su apariencia exterior (en los casos donde interesa el interior simplemente se aproxima, como en el caso del SubSurfaceScattering). Prácticamente todos los paquetes de diseño y animación (incluido Blender) empleados en síntesis de imagen y en aplicaciones interactivas emplean este tipo de modelos.

Para cubrir las necesidades de los modelos 3D de los órganos del sistema digestivo se ha optado por que estos fueran realizados en el modelado de contorno por su facilidad de desarrollo y ligereza de carga en el renderizado en el momento de la implementación de estos en el sistema de realidad virtual.

3.5. Generacion de Entorno 3D

El entorno 3D es en donde el usuario se encontrará al ingresar al sistema de realidad virtual, para, este se ha realizado para dar la sensación de encontrarse en un ambiente médico.

Se utilizaron modelos ya realizados por un autor adquiriendo los derechos de uso ya que la realización de estos no se consideran parte integral del desarrollo del Trabajo Terminal, escrito esto no se quiere demeritar la necesidad de hacer el usuario ya que, como se ha mencionado en secciones anteriores, se tiene énfasis en la experiencia del usuario para que la inmersión del usuario sea la mayor posible.

A continuación se muestran capturas del entorno 3D como fue implementado dentro el motor de desarrollo Unity.

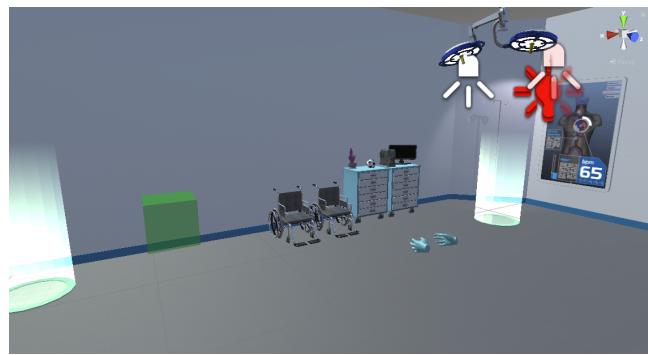


Figura 3.1: Entorno 3D vista normal

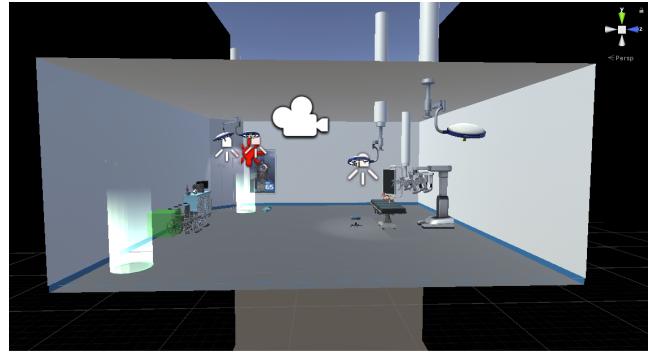


Figura 3.2: Entorno 3D vista externa de la escena

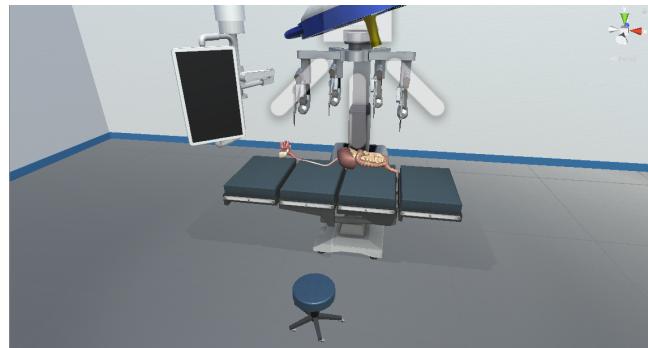


Figura 3.3: Entorno 3D vista principal

3.6. Generacion de Modelos 3D

Los componentes multimedia a desarrollar en modelos 3D los cuales son miembros del sistema digestivo del ser humano, el sistema digestivo incluye a los órganos del tubo alimenticio y glándulas de secreción exocrina y endocrina.

3.6.1. Glándulas Salivales

A continuación se muestran las figuras del resultado final del desarrollo de las glándulas salivales del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.

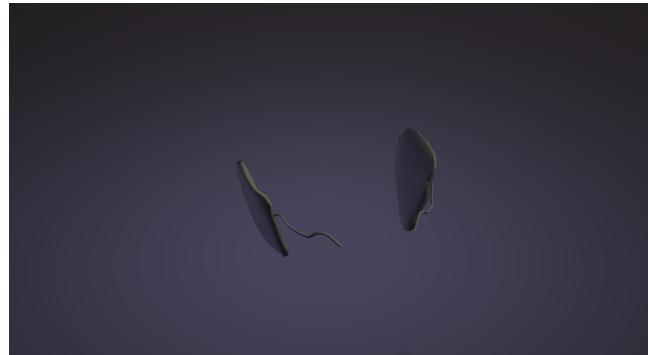


Figura 3.4: Modelo 3D de las glándulas salivales

3.6.2. Cavidad oral y faringe

A continuación se muestran las figuras del resultado final del desarrollo de la cavidad oral del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.

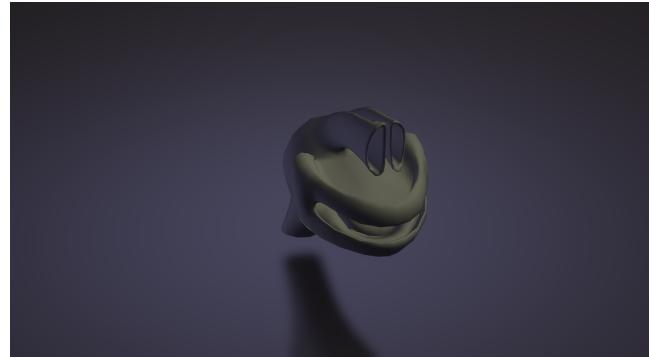


Figura 3.5: Modelo 3D de la cavidad oral

3.6.3. Esófago

A continuación se muestran las figuras del resultado final del desarrollo del esófago del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.

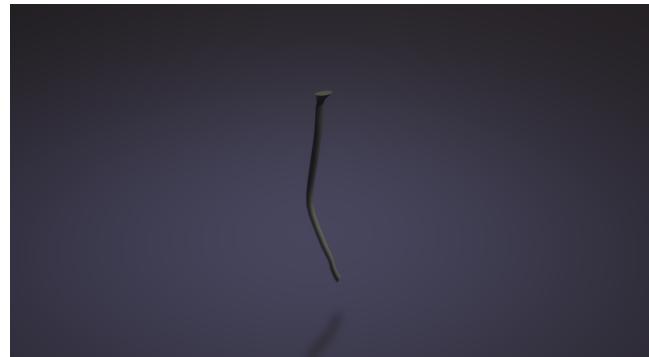


Figura 3.6: Modelo 3D del esófago

3.6.4. Estómago

A continuación se muestran las figuras del resultado final del desarrollo del estómago del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.



Figura 3.7: Modelo 3D del estómago

3.6.5. Intestino delgado

A continuación se muestran las figuras del resultado final del desarrollo del intestino delgado del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.



Figura 3.8: Modelo 3D del intestino delgado

3.6.6. Hígado

A continuación se muestran las figuras del resultado final del desarrollo del hígado del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.



Figura 3.9: Modelo 3D del hígado

3.6.7. Páncreas

A continuación se muestran las figuras del resultado final del desarrollo del páncreas del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.



Figura 3.10: Modelo 3D del páncreas

3.6.8. Vesícula Biliar

A continuación se muestran las figuras del resultado final del desarrollo de la vesícula biliar del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.

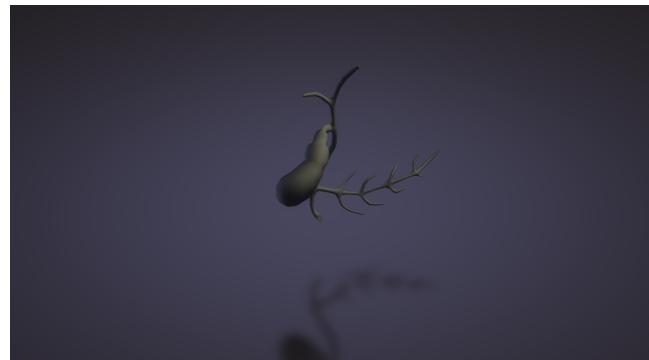


Figura 3.11: Modelo 3D de la vesícula biliar

3.6.9. Intestino Grueso y Ano

A continuación se muestran las figuras del resultado final del desarrollo del intestino grueso y ano del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado basado en el material anteriormente provisto.



Figura 3.12: Modelo 3D del intestino grueso

3.7. Modelo del sistema digestivo unificado

A continuación se muestran las figuras del resultado final del desarrollo del sistema digestivo en el software de modelado en 3D llamado “Blender”, este fue realizado reuniendo todos los modelos de órganos y elementos individuales creados con anterioridad.

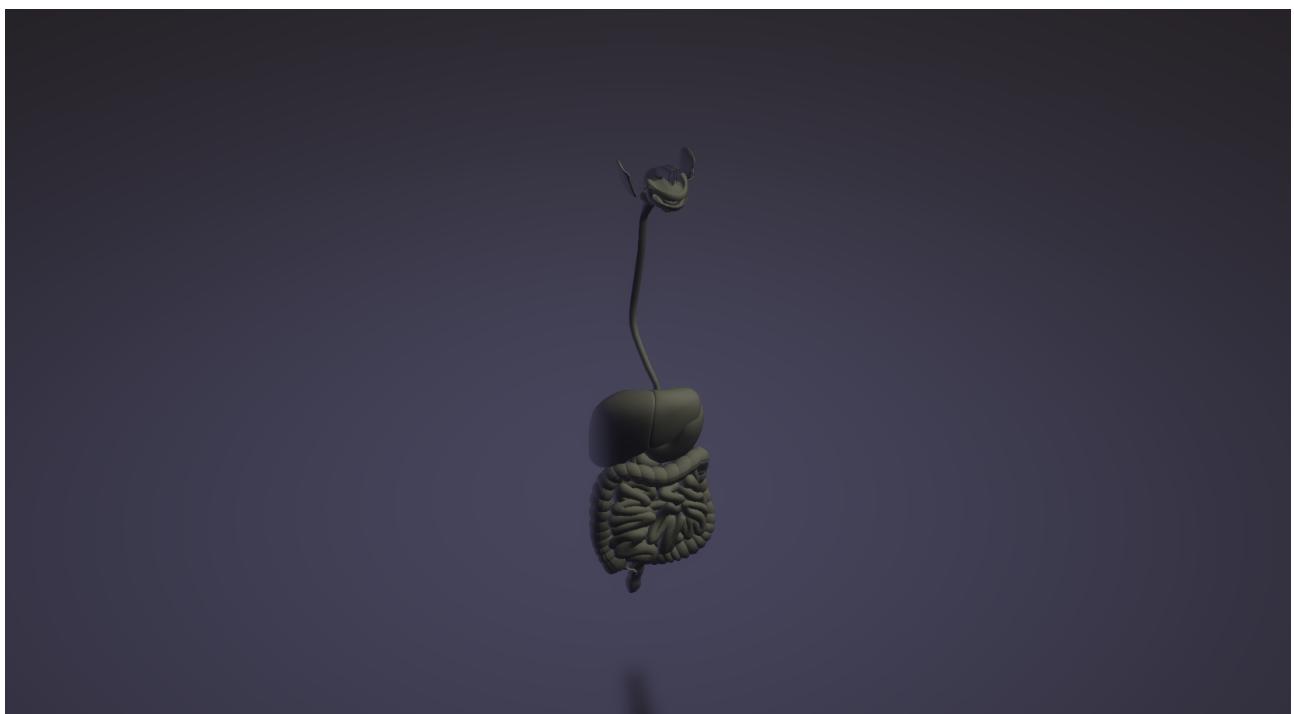


Figura 3.13: Modelo 3D del Sistema digestivo

3.8. Evaluación de modelos 3D por personal calificado

Debido al tiempo de desarrollo, el cual tomó más de lo planteado, de los modelos antes expuestos en la sección anterior no fue posible concretar una cita para su evaluación con el personal calificado de la Escuela Superior de Medicina en las fechas previamente planteadas.

Se esperaba poder tener una reunión en fechas posteriores pero la situación epidémica que se ha desarrollado en el país y limitaciones impuestas por las autoridades hicieron imposible la evaluación de los modelos desarrollados.

Esto no significa que no se haya hecho bajo rigor alguno, sólo se utilizaron materiales de medicina impresos, así como referencias en video de disecciones del sistema digestivo, esto para estar lo más familiarizado posible, como estudiante de ingeniería en sistemas computacionales, al momento de desarrollar dichos modelos.

3.9. Diseño y desarrollo de componentes de software

Los componentes de software a diseñar y desarrollar el cual interactúa con el entorno en 3D y modelos 3D.

Hubo puntos clave desarrollados los cuales tuvieron que ser desarrollada para llevar a cabo la mejor UX. Los desarrollos son incrementales, en cuanto a el grado de interacción que se logra.

Los componentes desarrollados para este software son:

- Seguimiento de HMD y controles
- Locomoción y ergonomía
 - Teleportación
 - Puntos de teleportación
 - Giros rápidos, entradas personalizadas, oclusión del usuario
- Presencia e interacción de las manos
 - Agregar manos
 - Interactuando con entorno
 - Interacciones manuales adicionales

Todos estos componentes forman la base para que la experiencia del software de realidad virtual se “sienta” lo más “natural” posible y estos mismos deben de estar refinados para que al integrarse con los componentes multimedia la interacción con estos se lleve de manera fluida.

3.10. Creando el proyecto en Unity ®

Se optó por el uso de Unity ® en su versión 2018.4.14f1 LTS ya que esta misma será soportada por más tiempo y es ideal para un desarrollo en el cual se necesite solamente usar una versión estable del editor sin que haya cambios dentro de este que perjudique el desarrollo del sistema; así mismo es la versión que Oculus[?] recomienda dentro de sus requerimientos y recomendaciones de desarrollo.

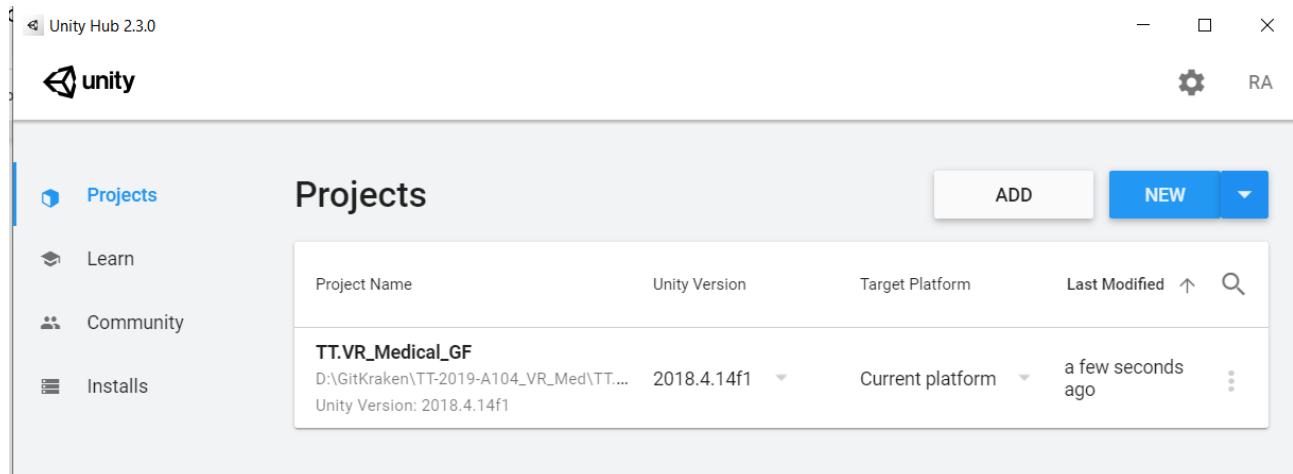


Figura 3.14: Unity Hub 2.3.0

3.11. Seguimiento de HMD y controles

En este apartado se realizó el rastreo del HMD y de los controles y que estos se vean reflejados como movimientos dentro del software y replicados en el mismo HMD el cual está siendo usado por el usuario y así puede percibir el movimiento. Esto es la base para que el software provea una experiencia en la cual el usuario pueda estar inmerso.

Utilizando el SDK que provee Oculus para su desarrollo con el motor Unity se realiza el elemento OVRCameraRig y TrackedAlias.

Se abrió el prefabricado OVRCameraRig , así como el objeto secundario TrackingSpace en la jerarquía. Se selecciona OVRCameraRig y luego se navega hasta el script LinkedAliasAssociationCollection en el inspector. Se adaptan los siguientes activos en las entradas apropiadas en el Script LinkedAliasAssociationCollection:

- TrackingSpace a Play Area
- CenterEyeAnchor a el HMD
- CenterEyeAnchor a la cámara del HMD
- LeftHand Anchor al controlador izquierdo
- RightHandAnchor al controlador derecho

Realizando estos pasos se logra el rastreo de los controles y el HMD exemplificando con la figura ??.

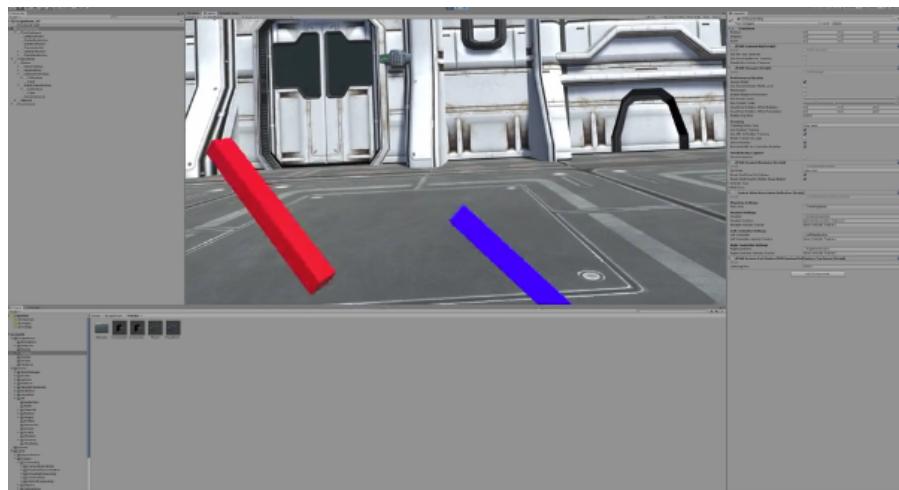


Figura 3.15: Entorno de desarrollo Unity mostrando el rastreo del HMD y los controles

3.12. Locomoción y ergonomía

El modo más simple de locomoción es el movimiento físico y este es el más confortable que se puede tener debido a que cuando uno decide moverse en la realidad el usuario también se moverá dentro del entorno virtual, pero este viene con limitaciones inherentes y retos que no son plenamente obvios al momento de la concepción y desarrollo, se exponen algunos a continuación .

- Los usuarios poseen áreas de interacción de diferentes tamaños, no todos poseen un área grande lo cual les permite más libertad de movimiento, mientras más pequeño sea el espacio que requiera el usuario más usuarios podrán hacer uso del sistema. Se podría ajustar el contenido del sistema, pero esto implicaría mayor trabajo y tiempo, esto no se traduce en una experiencia de usuario mejor que si este se hubiera diseñado con un área de interacción pequeña en primer lugar.
- Se toma en cuenta que con el movimiento físico es probable que algunos usuarios no tengan la posibilidad de moverse físicamente dentro de su área, pueden estar limitados por una discapacidad, lo cual implicaría en que el movimiento sería una distracción para el usuario asumiendo que de algún modo este pudiera interactuar de alguna manera.

Otros dos tipos más de locomoción que se consideraron para el desarrollo del sistema uno de estos es el movimiento guionizado y movimiento de avatar, el primero se refiere a cuando la perspectiva se mueve a través de un camino predeterminado, este es usualmente utilizado cuando se realiza una experiencia que no requiera movimientos del usuario, fácilmente ejemplificado por un viaje en una montaña rusa en un software de realidad virtual este puede llegar a causar vección, generalmente este tipo de movimiento suele ser poco tolerable en largos períodos de tiempo.

Por otro lado el movimiento de avatar es el clásico movimiento que solemos encontrar en videojuegos o en experiencias pasadas de realidad virtual, al mover una palanca del mando el avatar comenzará su movimiento. Este fue un acercamiento más que válido pero con el progreso de la tecnología la manera de implementar el movimiento ha evolucionado.

La teleportación se refiere a una mecánica instantánea o casi instantánea en la cual el usuario aparece en un lugar seleccionado, típicamente funciona apuntando hacia al lugar deseado y presionando un botón, aunque hay múltiples variaciones de este acercamiento.

La teleportación ocurre en un instante lo cual reduce a un mínimo la posibilidad de generar una molestia comparación de los tipos de movimiento expuestos anteriormente.

Tomando en cuenta los tipos de movimiento pasado, sus deficiencias y beneficios se tomó la decisión de implementar la teleportación como método de movimiento en el entorno virtual del sistema.

3.12.1. Teleportación

En este apartado se desarrollaron los movimientos que el usuario pudiera tener dentro del entorno en 3D, estos dan la posibilidad de “movimiento” dentro del mismo así como las limitaciones que se implementan para que el mismo usuario no pueda acceder a partes que no queramos o estén disponibles para este.

El desarrollo se realizó de la manera siguiente:

Se cambió el nombre del Prefab a una convención de nombres, en este caso `Teleport.Curved`. respectivamente, y se realizó una copia para la mano derecha para tener un objeto distinto para la mano derecha.

Se configuró el script de fachada de puntero adjunto a cada objeto arrastrando el Alias de controlador correspondiente de la Jerarquía al campo `FollowSource`.

Se crearon dos nuevos GameObjects vacíos para el manejo de las entradas del control y adjunto el script OVRInputTouchAction a cada uno de ellos.

Se estableció el valor de la propiedad táctil de ambos scripts OVRInputTouchAction en Thumbstick primario. Estableciendo el valor de la propiedad del controlador en L Touch y R Touch respectivamente.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Zinnia.Action;

public class OVRInputTouchAction : BooleanAction
{
    public OVRInput.Controller controller = OVRInput.Controller.Active;
    public OVRInput.Touch touch;

    void Update() {
        Receive(OVRInput.Get(touch, controller));
    }
}
```

Para cada objeto de ObjectPointer.Curved, se le asigna el objeto TeleportCurved correspondiente de la Jerarquía con un script OVRInputTouch al campo Acción de activación en el componente Fachada de puntero.

De esta manera al tocar cualquier thumbstick este invocará una curva la cual nos mostrará dónde será la teleportación una vez esta sea implementada.

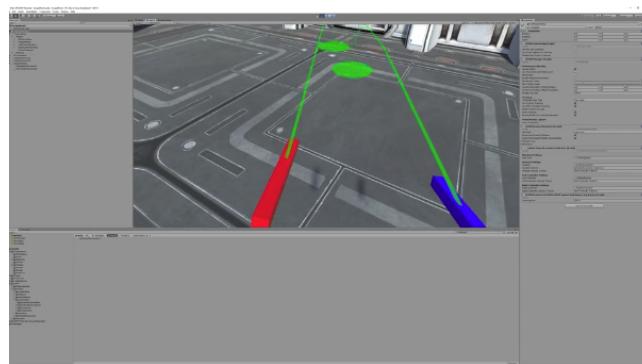


Figura 3.16: Curva que muestra la ubicación de la teleportación al tocar los thumbstick de los controles.

Para la adición de la acción de teleportación, la cual es la que permitirá el movimiento del usuario dentro del entorno de realidad virtual. Se tuvo la opción de implementar dos tipos de acciones de teleportación, Teleport.Instant y Teleport.Dash, se eligió la primera ya que supone una menor probabilidad de causar vección en el usuario ya que no se percibe movimiento alguno en la acción de teleportación.

Se implementó VRTK/Locomotion/Teleporter y arrastrando el Teleporter.Prefab instantánea a la escena.

Se estableció el campo de validez de la cámara del script de TeleporterFacade en el objeto de escena SceneCameras.

Se crearon dos nuevos GameObjects vacíos para adjuntar las entradas de los botones y se adjunto el script OVRInputButtonAction a cada uno de ellos.

Se estableció el valor de la propiedad táctil de ambos scripts OVRInputButtonAction en Thumstick primario. Se estableció el valor de la propiedad del controlador en L.Touch y R.Touch respectivamente.

```
using System.Collections.Generic;
using UnityEngine;
using Zinnia.Action;

public class OVRInputButtonAction : BooleanAction {
    public OVRInput.Controller controller = OVRInput.Controller.Active;
    public OVRInput.Button button;

    void Update() {
        Receive(OVRInput.Get(button, controller));
    }
}
```

Para cada objeto prefabricado Curva de puntero de objeto, se asignó los GameObjects correspondientes con el script OVRInputButtonAction al campo Acción de selección en el componente Fachada de puntero.

Para cada puntero de ObjectPointer.Curved, se agregó un nuevo objeto EventData en la lista Datos de acción seleccionados.

Se asignó el objeto Teleporter.Instant al campo de objeto ubicando y seleccionando TeleporterFacade y luego Teleport.

Implementando los procesos pasados se logra que el usuario pueda teleportarse dentro del entorno virtual.

De esta manera al tocar cualquier thumbstick este invocará una curva la cual nos mostrará dónde será la teleportación una vez esta sea implementada.

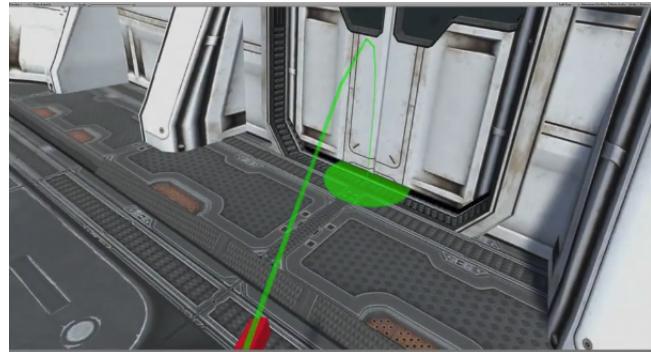


Figura 3.17: Curva que muestra la ubicación de la teleportación al tocar los thumbstick del control izquierdo.



Figura 3.18: Curva que muestra la ubicación de la teleportación al tocar los thumbstick de los controles.

En la figura ?? se puede notar un oscurecimiento, el cual es representativo del desvanecimiento que se implementó para reducir la vección del usuario y evitar cinetosis en los usuarios que pudieran llegar a ser propensos a ella.

Se procedió a integrar un indicador de flecha el cual servirá para indicar a de cara a que posición se estará cuando se realice la teleportación, así ampliando las posibilidades de interacción del usuario y facilitando las mismas.

Se importó un modelo de flecha en la vista Proyecto y fue arrastrado a la escena debajo del objeto de escena ValidContainer ubicado dentro de la jerarquía de objetos de escena de Destino para cada prefab ObjectPointerCurved.

Se ajustó la transformación del nuevo objeto de flecha para que sea visible dentro de la escena y asigne material PointerDefaultValid al renderizador de malla.

Se crearon dos nuevos GameObjects para cada control y obtener la dirección del stick del control vacíos y integró el script OVRInputAxis2DAction a cada uno de ellos.

Se estableció el valor de la propiedad táctil de ambos scripts OVRInputAxis2DAction en el control primario así como el valor de la propiedad del controlador en L Touch y R Touch respectivamente.

Se implementa el prefabricado AxisRotator a la escena.

Se cambia el nombre del Prefab a una convención de nombres AxisRotator.L y se realiza una copia para la mano derecha ya se que implemento de la misma manera con el nombre AxisRotator.R

Se crearon dos nuevos GameObjects vacíos debajo de cada uno de los objetos OVRInputAxis2DAction creados anteriormente y se agrego el script Vector2ToFloat a los cuatro.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Zinnia.Action;

public class OVRInputAxis2DAction : Vector2Action
{
    public OVRInput.Controller controller = OVRInput.Controller.Active;
    public OVRInput.Axis2D axis;

    void Update() {
        Receive(OVRInput.Get(axis, controller));
    }
}
```

Se estableció el valor del eje del script Vector2ToFloat en el eje correcto, en este caso, una X y una Y para cada objeto OVRInputAxis2DAction.

En cada secuencia de comandos OVRInputAxis2DAction, se agregaron dos nuevos objetos Vector2 a la lista de valores cambiados de Vector2.

Se asignaron los objetos Vector2ToFloat a cada campo. Usando el menú desplegable, seleccione Vector2ToFloat y luego DoTransform.

Use el botón .^agregar componente.^en el Inspector para agregar el componente de script FloatAction a cada objeto Vector2ToFloat.

En cada scriptVector2ToFloat, agrege un nuevo objeto de acción individual a la lista de acciones individuales transformadas y asigne el objeto primario.

Usando el menú desplegable, seleccione FloatAction y luego Recibir.

Se configuró cada objeto AxisRotator asignando un objeto FloatAction en los campos Eje lateral lateral y Eje longitudinal correspondientes en el script AxisRotatorFacade para los ejes X e Y de cada mano, respectivamente.

En la secuencia de comandos de TeleporterFacade, establezca el valor de Uso de desplazamiento en Offset siempre con rotación de destino.

Implementando esto se logra que un puntero aparezca en una flecha en la dirección indicada para que el usuario pueda elegir su dirección de orientación al elegir la ubicación de teleportación como se

muestra en la figura ??.

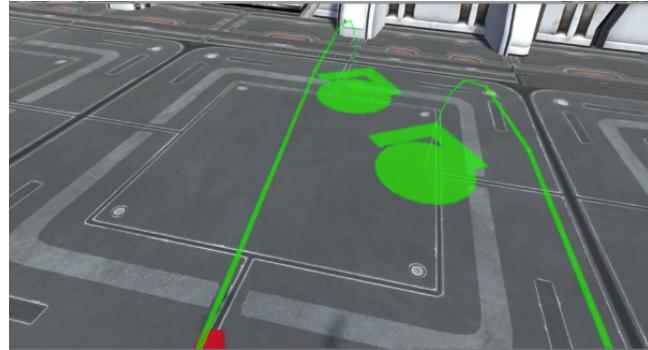


Figura 3.19: Curva que muestra la ubicación de la teleportación y posición del stick al interactuar con los thumbstick de los controles

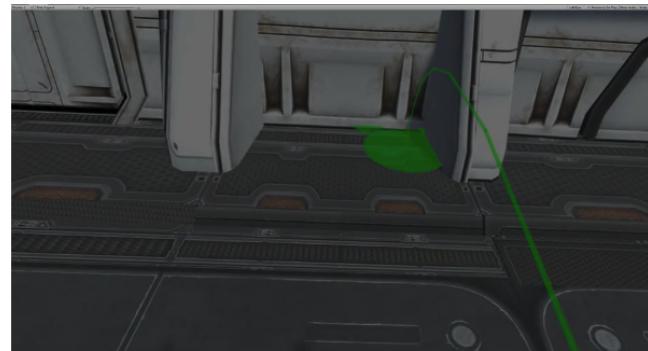


Figura 3.20: Curva que muestra la ubicación de la teleportación y posición del stick al interactuar con los thumbstick de los controles

3.12.2. Puntos de teleportación

Los puntos de teleportación son los cuales resaltan una locación dentro del entorno virtual la cual se quiere o requiere que el usuario vaya con facilidad hacia ellas.

Se agregó una nueva capa llamada "Teleportable" se estableció el objeto FloorCollider en la nueva capa.

Se creó un nuevo GameObject vacío y se agregaron componente en el Inspector para adjuntar el script AnyLayerRule.

Se estableció el campo LayerMask en el script AnyLayerRule en la capa Teleportable.

```
namespace VRTK.Prefabs.Locomotion.DestinationLocations
{
    using UnityEngine;
```

```

using UnityEngine.Events;
using Malimbe.MemberChangeMethod;
using Malimbe.MemberClearanceMethod;
using Malimbe.XmlDocumentationAttribute;
using Malimbe.PropertySerializationAttribute;
using Zinnia.Rule;
using Zinnia.Data.Attribute;

/// <summary>
/// The public interface into the DestinationLocation Prefab.
/// </summary>
public class DestinationLocationFacade : MonoBehaviour
{
    #region Location Settings
    /// <summary>
    /// Determines if the location is in the locked and unusable state.
    /// </summary>
    [Serialized]
    [field: Header("Location Settings"), DocumentedByXml]
    public bool IsLocked { get; set; }
    /// <summary>
    /// Whether to apply the rotation of the custom destination location to the selected a
    /// </summary>
    [Serialized]
    [field: DocumentedByXml]
    public bool ApplyDestinationRotation { get; set; } = true;
    /// <summary>
    /// Allows to optionally determine which <see cref="SurfaceData"/> sources can affect
    /// </summary>
    [Serialized, Cleared]
    [field: DocumentedByXml]
    public RuleContainer SourceValidity { get; set; }
    #endregion

    #region Location Events
    /// <summary>
    /// Emitted when the Destination Location is entered for the first time.
    /// </summary>
    [Header("Location Events"), DocumentedByXml]
    public UnityEvent HoverActivated = new UnityEvent();
    /// <summary>
    /// Emitted when the Destination Location is entered.
    /// </summary>
    [DocumentedByXml]
    public DestinationLocation.SurfaceDataUnityEvent Entered = new DestinationLocation.Sur
    /// <summary>
    /// Emitted when the Destination Location is exited.
    /// </summary>
    [DocumentedByXml]

```

```

public DestinationLocation.SurfaceDataUnityEvent Exited = new DestinationLocation.Surf
/// <summary>
/// Emitted when the Destination Location is exited for the last time.
/// </summary>
[DocumentedByXml]
public UnityEvent HoverDeactivated = new UnityEvent();
/// <summary>
/// Emitted when the Destination Location is activated.
/// </summary>
[DocumentedByXml]
public DestinationLocation.TransformDataUnityEvent Activated = new DestinationLocation.
/// <summary>
/// Emitted when the Destination Location is deactivated.
/// </summary>
[DocumentedByXml]
public UnityEvent Deactivated = new UnityEvent();
#endregion

#region Reference Settings
/// <summary>
/// The linked Internal Setup.
/// </summary>
[Serialized, Cleared]
[field: Header("Reference Settings"), DocumentedByXml, Restricted]
public DestinationLocationConfigurator Configuration { get; protected set; }
#endregion

/// <summary>
/// Called after <see cref="IsLocked"/> has been changed.
/// </summary>
[CalledAfterChangeOf(nameof(IsLocked))]
protected virtual void OnAfterIsLockedChange()
{
    Configuration.SetLockedState(IsLocked);
}

/// <summary>
/// Called after <see cref="ApplyDestinationRotation"/> has been changed.
/// </summary>
[CalledAfterChangeOf(nameof(ApplyDestinationRotation))]
protected virtual void OnAfterApplyDestinationRotationChange()
{
    Configuration.LocationController.ApplyDestinationRotation = ApplyDestinationRotati
}

/// <summary>
/// Called after <see cref="SourceValidity"/> has been changed.
/// </summary>
[CalledAfterChangeOf(nameof(SourceValidity))]

```

```
        protected virtual void OnAfterSourceValidityChange()
        {
            Configuration.LocationController.SourceValidity = SourceValidity;
        }
    }
}
```

Se estableció el campo TargetValidity en el script TeleporterFacade en AnyLayerRule script gameobject.

Para cada ObjectPointerCurved, se asigno AnyLayerRule al campo TargetValidity del script PointerFacade.

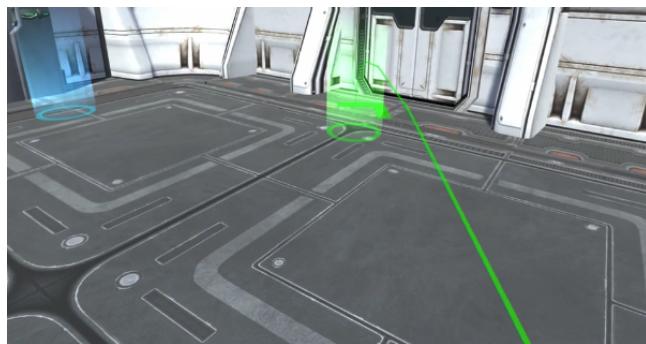


Figura 3.21: Desvanecimiento al realizar la teleportación y posición del stick al interactuar con los thumbstick del control

3.12.3. Giros rápidos, entradas personalizadas, oclusión del usuario

En esta sección se implementan giros de la cámara para facilitar al usuario la interacción en el entorno virtual así como la personalización y mejoras de interacción con los controles las cuales pretenden hacer más intuitivo el movimiento del usuario en el sistema de realidad virtual, además se implementan occlusiones en determinados lugares para evitar que el mismo usuario pueda ver a través de modelos 3D no deseados mermando así la experiencia de usuario.

Se creó un nuevo GameObject vacío para obtener cuando el stick es tocado y adjunto el script de TeleporterActivation y configurando el campo Controlador en R Touch.

Se creó otro nuevo GameObject vacío y adjunto la secuencia de comandos TeleporterSelection y configuro el campo Controlador en R Touch.

Se crearon dos nuevos GameObjects vacíos para ser agregados al InputHandler bajo el script TeleporterSelection GameObject y adjuntó el script FloatAction a cada uno.

Se agregó el componente en el Inspector para agregar los objetos del juego de script `FloatAction` a los campos `Extraer X` e `Extraer Y` del script `TeleporterSelection`.

Se asignaron los campos Eje lateral y Eje longitudinal en la secuencia de comandos AxisRotator-Facade a las dos nuevas secuencias de comandos de FloatAction GameObjects.

Fueron deshabilitados los objetos de juego de script OVRInputTouch, OVRInputButtonAction y OVRInputAxis2DAction para la mano derecha.

Fueron reasignados los campos Acción de activación y Acción de selección en la secuencia de comandos de PointerFacade de la derecha a la secuencia de comandos TeleporterActivation y TeleporterSelection GameObjects, respectivamente.

Se inhabilitó el prefabricado ObjectPointerCurved de la izquierda en la escena y el script OVRInputTouch, OVRInputButtonAction GameObjects para la mano izquierda.

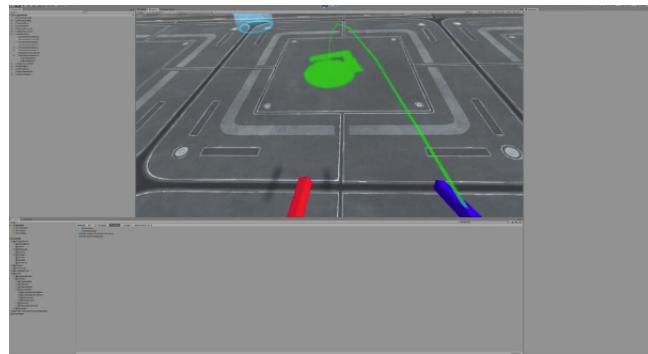


Figura 3.22: Curva de teleportación habilitada solamente para el control derecho

En el script AxesToVector3Facade adjunto al PrefabObject, estableciendo el Tipo de uso del eje Direccional.

Se estableció el campo Eje lateral en el valor del eje X FloatAction script GameObject ubicado debajo de la mano izquierda OVRInputAxis2DAction script GameObject.

Se estableció el valor del campo Multiplicador de velocidad lateral en 45 y el valor del campo Multiplicador de velocidad longitudinal en 0.

Agregando una secuencia de comandos Vector3Cache a la secuencia de comandos AxesToVector3Facade GameObject.

Se agregó una nueva acción Vector3 a la lista de acciones Vector3 procesadas en el script AxesToVector3Facade y fueron agregadas al GameObject padre. Estableciendo las funciones en Vector3Cache/CachedData.

Fue agregado un script CameraColorOverlay al script GameObject de AxesToVector3Facade.

Se agregó un nuevo Vector3 a la lista de Vector3 modificado en el script Vector3Cache y se agregó el objeto del juego principal. Estableciendo las funciones en CameraColorOverlay/link.

En el script CameraColorOverlay, se asignó el objeto de escena SceneCameras al campo Validez de la cámara.

Se implementó el material de TeleportFade en el proyecto y asignando al campo Material de superposición en el script CameraColorOverlay y cambie el valor del campo Eliminar duración a 0.5.

Se creó un nuevo GameObject vacío y se agrego el componente para agregar el script Vector3ToVector2 y un script Vector2ToVector3.

Fue agregada una nueva acción Vector3 a la lista de acciones Vector3 procesadas en el script AxesToVector3Facade y agregando el GameObject del script Vector3ToVector2. Se estableció la función en Vector3ToVector2/DoTransform.

Se agregó una nueva acción Vector2 a la lista de acciones Transform(Vector2) en el script Vector3ToVector2 y agregó el GameObject principal. Se establecieron las funciones en Vector2ToVector3.DoTransform.

Se establecieron dentro del campo Mapa de coordenadas en X a Y e Y a X excluyendo Z en el script Vector3ToVector2. Se agregó el script TransformEulerRotationMutator al objeto de juego de script Vector3ToVector2.

En el script Vector2ToVector3, fue agregada una nueva acción Vector3 a la lista de acciones Transform(Vector3). Asignando el GameObject principal y estableciendo el menú desplegable en TransformEulerRotationMutator/DoIncrementProperty.

Estableciendo el campo Destino en el script TransformEulerRotationMutator para el objeto de escena PlayAreaAlias. Se activan las opciones usar valores locales y mutar en el eje Y e inhabilitar las opciones mutar en el eje X y mutar en el eje Z.

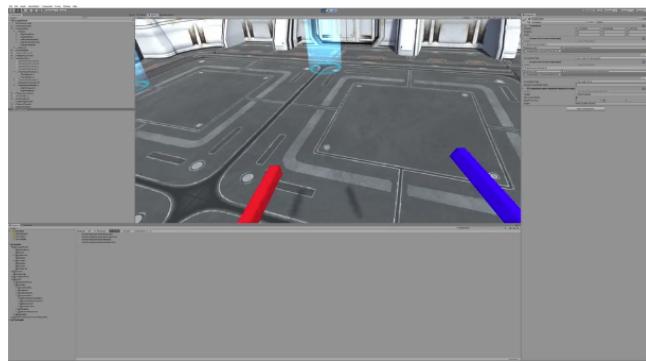


Figura 3.23: Captura previa de giro rápido con stick de control izquierdo



Figura 3.24: Captura posterior de giro rápido con stick de control izquierdo

Deshabilitando el objeto SnapToFloor debajo del objeto Teleporter_..InstantPrefab en la escena.

Utilizando el prefabricado CollisionFader en el proyecto y arrastrándolo a la escena debajo del objeto de escena HeadsetAlias.

En el script CameraColorOverlay adjunto al prefabricado CollisionFader, se estableció el campo CameraValidity en el objeto de escena SceneCameras.

Fueron creadas dos nuevas capas llamadas FadeCamera y FadeOut. Colocando el objeto prefabricado CollisionFader en la capa FadeCamera.

Se colocó el objeto de la escena principal en la capa FadeOut. Se cambió la matriz de colisión para que la capa FadeCamera solo colisione con la capa FadeOut.

Cambiando el valor del campo ClearFlags en los scripts de la cámara bajo el GameObject de escena OVRCameraRig a color sólido y configurando el color en negro.

3.13. Presencia e interacción de las manos

Desde la introducción de los controles Oculus Touch para el sistema Oculus Rift y similares, la presencia de las manos ha sido la manera principal de proveer una experiencia de realidad virtual, los usuarios pueden encontrar manos virtuales que simulan las que poseen en realidad, esto nos ayuda a utilizar el conocimiento de los usuarios del mundo real y trasladarlo a el entorno virtual en lugar de tener que crear y asignar botones y crear controles complicados ya que las manos en la realidad virtual son tan importantes como en la realidad.

La implementación de manos en el sistema de realidad virtual se realizó de la manera siguiente:

3.13.1. Agregar manos

Se integraron los prefabricados CustomHandLeft y CustomHandRight en el proyecto y se adicionaron a los prefabricados LeftControllerAlias y RightControllerAlias correspondientes en la escena.

Deshabilitando el script OVRGrabber en los objetos de escena CustomHandLeft y CustomHandRight.

Deshabilite el componente MeshRenderer para los objetos Cube en las jerarquías LeftControllerAlias y RightControllerAlias.

Se establecieron la capa de los objetos de escena CustomHandLeft y CustomHandRight para ignorar Raycast.

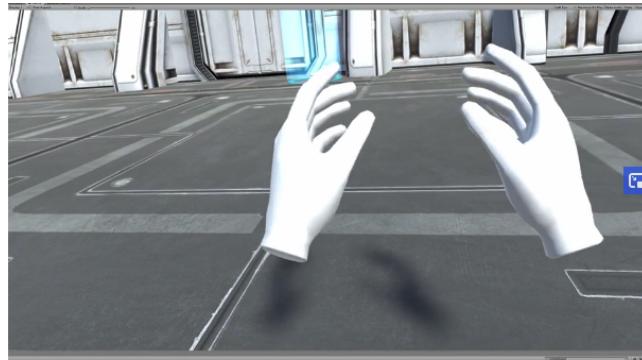


Figura 3.25: Cambio de placeholders por modelos de manos en 3D

Un método de configuración y refinamiento de la posición de las manos es utilizando el HMD y se compara la posición de sus manos reales con la de las manos virtuales, otro método que se utilizó fue cambiar los modelos dentro del entorno mediante prueba y error, comprobando qué tan similar es la localización de los dedos conforme a la realidad comparada a lo que se ve en el entorno virtual.

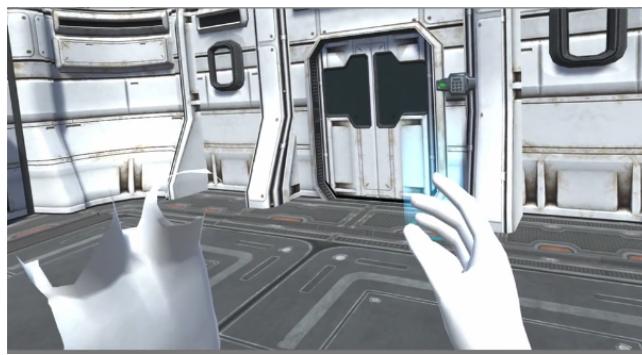


Figura 3.26: Oclusión deshabilitada para modelo de manos 3D



Figura 3.27: Refinado de localización de manos en el entorno virtual

Cabe mencionar que debido a la diferencia de complejión de los humanos no se puede realizar una configuración universal de estos elementos.

Se localizan los modelos OculusTouchRift..Left y OculusTouchRift..Right ubicados en la escena LeftControllerAlias y RightControllerAlias respectivamente para hacer una representación del control en el entorno virtual, si se decide utilizarse en un futuro.

3.13.2. Interactuando con entorno

Se inhabilitó el MeshRenderer en el objeto ExampleAvatar en cada una de las jerarquías de objetos de Interactor.

Creando dos nuevos GameObjects para agregarlo al InputHandler y poder controlar los gatillos posteriormente y se agrega el componente en el inspector para agregar los scripts OVRInputAxis1DAction y FloatToBoolean a cada objeto.

En los scripts FloatToBoolean, se establezca el campo límites positivos entre 0,75 y 1. Esto para tener una sensación de agarre conforme se va presionando el botón y no sea cuando ya está presionado completamente el gatillo.

En cada secuencia de comandos OVRInputAxis1DAction, se establezca el campo controlador en los valores L Touch y R Touch correspondientes y establezca el campo axis en Disparador manual primario. Se agrega componente en el inspector para agregar el script BooleanAction a cada objeto.

En los scripts OVRInputAxis1DAction, se agregó un nuevo objeto de acción individual en la lista de acciones individuales de ValueChanged. Agregando el GameObject principal y se estableció el script en FloatToBoolean/DoTransform.

En los scripts FloatToBoolean, se agregó un nuevo objeto de acción booleana en la lista de acciones booleanas transformadas. Agregando el GameObject principal y estableciendo en BooleanAction/Receive.

En los scripts de InteractorFacade en los prefabricados de Interactor en la escena, estableciendo el campo GrabAction en los GameObjects de script BooleanAction correspondientes.

Se utiliza el prefab Interactable.Primary_Grab.Secondary Swap en el proyecto y se agregó debajo de la jerarquía de objetos de escena y se hizo coincidir su transformación para los valores de transformación del objeto de escena del modelo en 3D con el cual se quiere interactuar.

Se deshabilitó el objeto DefaultMesh en la jerarquía Interactable.Primary_Grab.Secondary swap object.

Duplicando el objeto de escena del modelo 3D deseado y se vuelve a pintar debajo del objeto Meshes en la jerarquía Interactable.Primary_Grab.Secondary_swap object y se ponga a cero los valores de transformación cuando sea necesario.

Se inhabilitó el objeto de escena original del modelo 3D deseado.

Si es necesario se ajusta la nueva transformación del modelo en 3D en los grados necesarios en el eje Y y se ajusta la transformación Interactable.Primary_Grab.Secondary swap object para ubicarse en el entorno 3D correctamente.

3.13.3. Interacciones manuales adicionales

Para dar una sensación de inmersión más fidedigna se integran los siguientes elementos, estos harán que la gravedad y la velocidad que es impuesta a un objeto se mantenga.

Se agregó el script OVRAnchorVelocityEstimator a los objetos de escena LeftHandAnchor, RightHandAnchor y CenterEyeAnchor.

En cada script OVRAnchorVelocityEstimator fue establecido el campo Tracked GameObject en el objeto padre del juego y el campo Relative To en el objeto de la escena TrackingSpace.

En el script LinkedAliasAssociationCollection adjunto al objeto de escena OVRCameraRig, establezca el campo Headset Velocity Tracker en el objeto de escena CenterEyeAnchor, el campo Left Controller Velocity Tracker en el objeto de escena LeftHandAnchor y el campo en el objeto de escena RightHandAnchor.

3.14. Integración de componentes del producto

Para el integración de producto, como el mismo nombre de esta fase lo menciona los productos antes realizados, los componentes multimedia y de software finalmente se unen dentro del sistema.

3.14.1. Generación de interacción con modelos 3D

Para poder importar los modelos del sistema digestivo del cuerpo humano estos son agregados dentro de la misma carpeta del proyecto y seleccionar nuevo Asset.

Posteriormente estos se agregan a la escena principal del sistema, dependiendo de cuales son los que se deseen integrar.

Duplicando el objeto de escena del modelo 3D deseado y se vuelve a pintar debajo del objeto Meshes en la jerarquía Interactable.Primary_Grab.Secondary_swap object y se ponga a cero los valores de transformación cuando sea necesario.

Se inhabilitó el objeto de escena original del modelo 3D deseado.

Si es necesario se ajusta la nueva transformación del modelo en 3D en los grados necesarios en el eje Y y se ajusta la transformación Interactable.Primary_Grab.Secondary swap object para ubicarse en el entorno 3D correctamente.

De esta manera se integran rápidamente los elementos multimedia diseñados anteriormente.

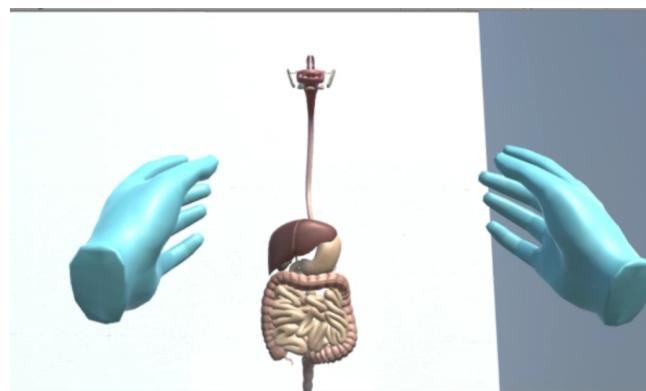


Figura 3.28: Integración de modelo del sistema digestivo 3D dentro del entorno 3D

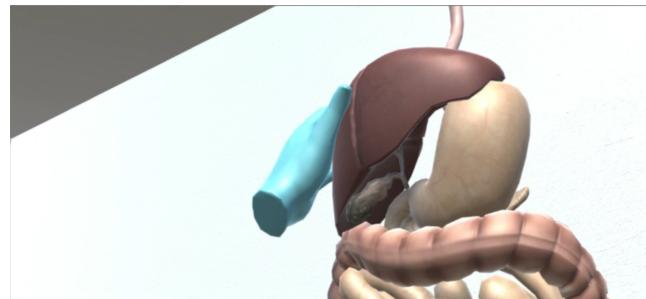


Figura 3.29: - Interacción con modelo del hígado del sistema digestivo 3D dentro del entorno 3D

3.15. Diagramas de Flujo del sistema.

Como es de esperarse la interacción de un usuario con un software de Realidad Virtual cambia en la percepción del uso de este, ya que a diferencia de un software en un ambiente tradicional, por ejemplo, una página web, un editor de textos o un sistema contable, este utiliza los periféricos y al usuario mismo para realizar los movimientos con su propio cuerpo para interactuar con los modelos.

La dificultad del modelado en diagramas de flujo recae en que el usuario puede tomar virtualmente decisiones infinitas solo siendo delimitado por las capacidades y opciones que el sistema provea al

usuario, esto prueba que el modelado con este tipo de diagramas no es la forma más ideal de demostrar el uso del sistema, pero sí muestra de las decisiones que se deben de tomar para llegar a el resultado de interacción esperado.

En este caso se describirán las acciones que se tienen que seguir para llegar a ejecutar una acción determinada, en sí, los elementos de los órganos y partes de los sistemas del cuerpo humano que serán incluidos en el sistema en este caso el sistema digestivo, además de las funcionalidades de locomoción y presencia e interacción de manos.

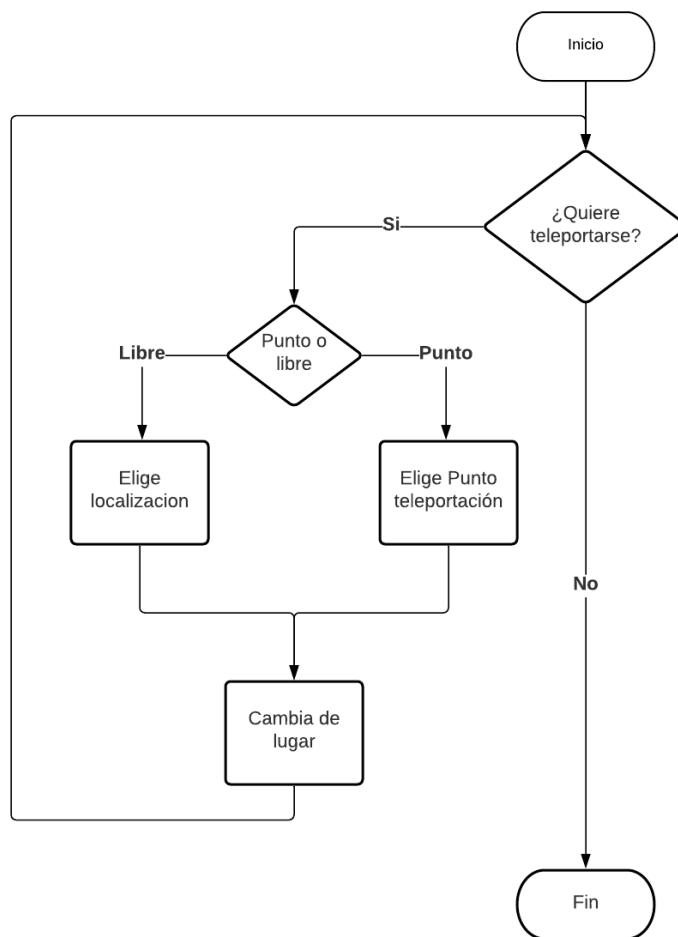


Figura 3.30: Diagrama de flujo de acción de teleportación

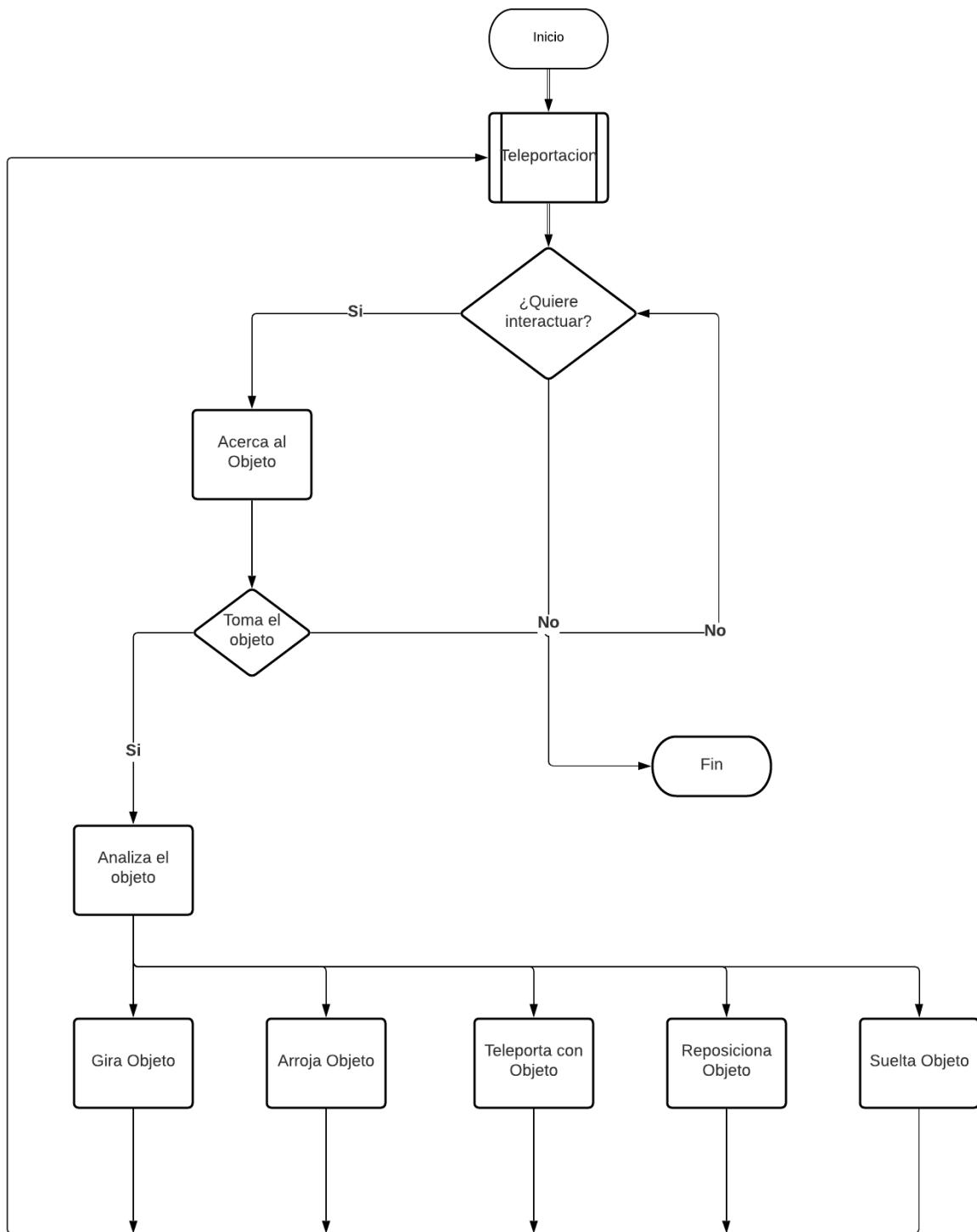


Figura 3.31: Diagrama de flujo de acción de interacción con objetos 3D

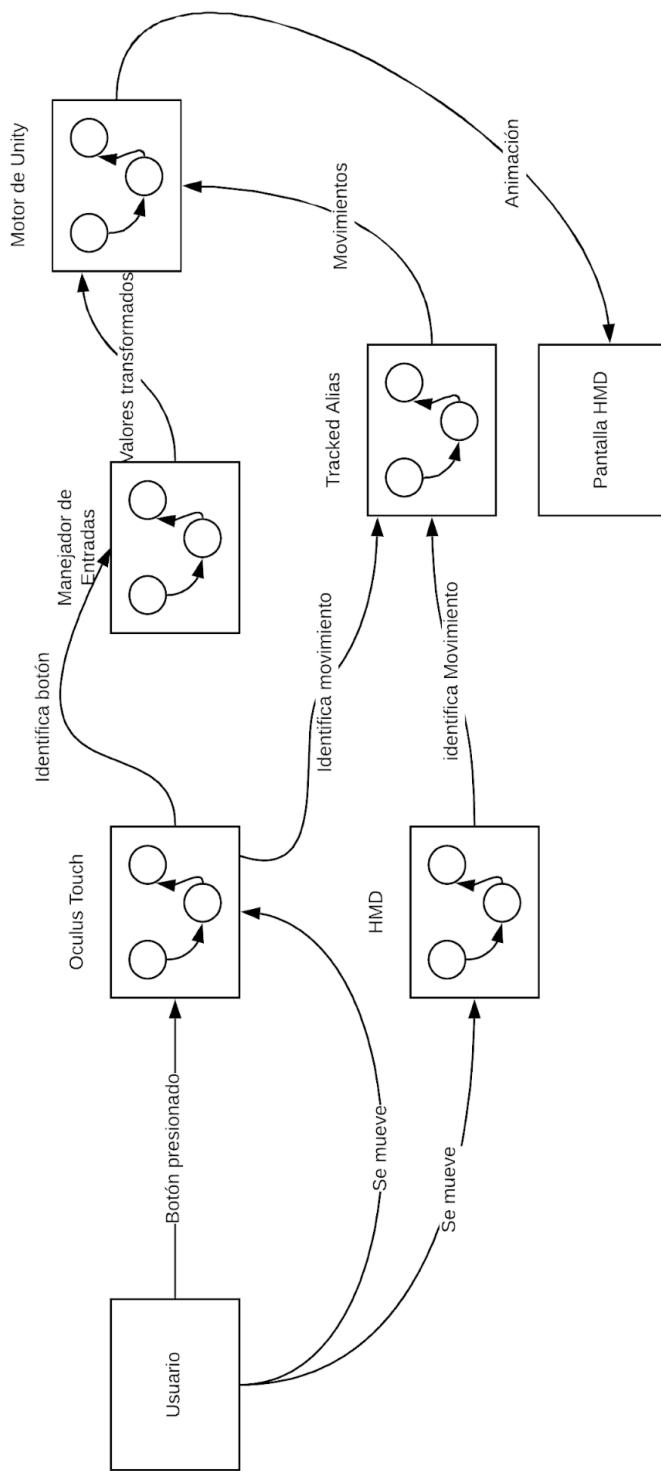


Figura 3.32: Diagrama de flujo de datos del análisis estructurado moderno[?]

Capítulo 4

Verificación y Pruebas

4.1. Mecanismos de medición para la fase de pruebas

Para este trabajo se realizaron dos tipos de prueba: pruebas de experiencia de usuario mediante las cuales, se busca obtener retroalimentación por parte de potenciales usuarios, de manera que el desarrollador pueda identificar errores y/o posibles cambios que pudieran hacerse al producto.

Como mecanismo de medición para las pruebas de experiencia de usuario, se emplea el método de validación de la metodología de Design Sprint de Google ® Ventures[?]. El proceso de validación se compone de dos partes:

1. Pruebas de usuario.
2. Retroalimentación de los stakeholders.

Después de la finalización del prototipo se procede a realizar pruebas. Una simple prueba de usuario permite descubrir información valiosa rápidamente. Ayuda a responder preguntas como ¿Qué es lo que los usuarios disfrutan o no gustan del prototipo? ¿Qué les gustaría que mejorara?

Después se continúa con la validación de los interesados o accionistas (stakeholders). Estas personas son el director de la Escuela Superior de Medicina, profesores del área de morfología y los alumnos de la misma.

Para motivos de este trabajo y los tiempos que se viven, el stakeholder será el mismo estudiante que presenta este trabajo, ya que resulta imposible hacer pruebas con los usuarios potenciales debido a la situación de epidemia nacional por el virus SARS-COV-2.

De igual forma se tomará en cuenta la retroalimentación de los sinodales y directores del proyecto mismo. Así, la revisión y aprobación de estos es esencial para que la validación se considere exitosa.

4.2. Pruebas de Entorno 3D con headset de R.V.

A continuación, se mostrarán las pruebas no automatizadas realizadas a las características, cada prueba está relacionada a un feature o features específicos, los cuales fueron probados conforme fueron desarrollados bajo el flujo de trabajo Git Flow[?].

Se detalla que es lo que se espera conseguir y bajo qué condiciones se consigue este objetivo, así como se indica si la característica ha pasado la prueba o la ha fallado. Así mismo se incluyen capturas de pantallas que muestran el funcionamiento de la característica y observaciones particulares a la prueba, indicando según lo requiera detalles de implementación o configuración.

4.2.1. Configuración de entradas

- id de Develop Realizados
 - 1
- Descripción de la Prueba

- Configurar la entrada de movimiento de los controles Oculus Touch ®.

▪ Condiciones de la Prueba

- El usuario, desde una posición inicial, realiza el movimiento del control izquierdo y derecho, uno a la vez, en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z. Recibiendo el movimiento de los mismos y este será mostrado dentro del HMD, demostrado por el placeholder de ambos controles.

▪ Observaciones

- A pesar de que aún no se utilizan todos los entradas que ofrece el control Oculus Touch ® el rastreo de los controles en el espacio funciona correctamente, esto nos sirve para desarrollar a futuro las siguientes características adicionales.

A continuación se muestra la exemplificación de el movimiento que se realiza y el cual es mostrado posteriormente dentro del sistema.

▪ Estado

- Aprobado

▪ Capturas de Pantalla

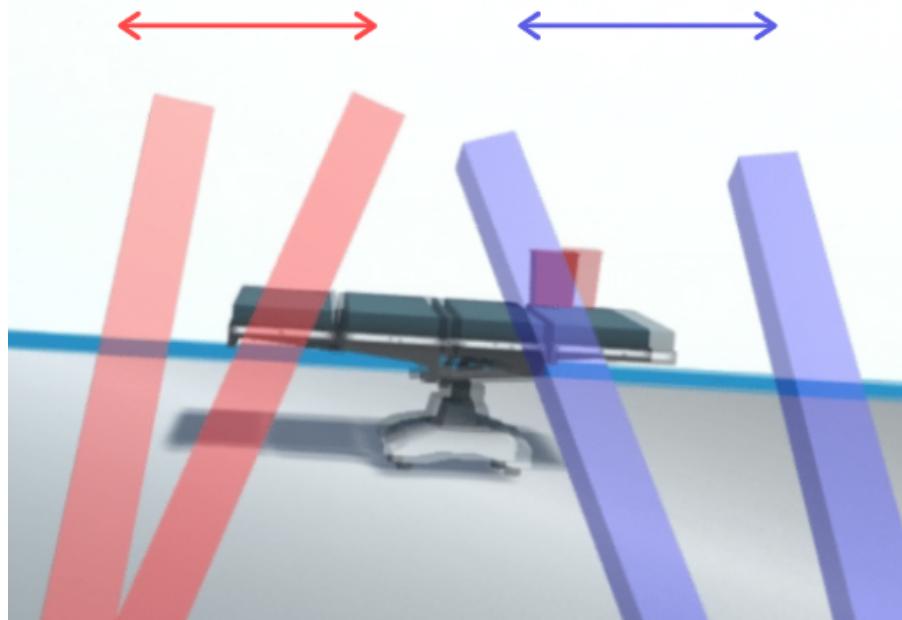


Figura 4.1: Placeholders con forma de prismas rectangulares representando y mostrando el movimiento de las manos con los controles Oculus Touch.

4.2.2. Teleportación

▪ id de Develop Realizados

- 2

■ Descripción de la Prueba

- Prueba de en la cual el sistema es capaz de emitir una curva indicadora que emana desde el placeholder del usuario hacia un destino requerido por el mismo.

■ Condiciones de la Prueba

- El usuario, desde una posición inicial, toca el stick de cualquier control y realiza el movimiento del control izquierdo o derecho, uno a la vez, dirigiendo el arco de indicación de localización de teleportación en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z, moviendo este hacia el lugar deseado dentro del entorno 3D y al dejar de tocarlo este deberá de desaparecer inmediatamente. Este será mostrado por el HMD, demostrando la capacidad de realizar la llamada de estos elementos.

■ Observaciones

- Al tocar el stick de cualquiera de los controles se emite una curva para determinar una próxima función de teleportación, el arco creado es claro y útil para determinar la localización y este se mueve dependiendo de la distancia y ángulo en el cual sea invocado. Puede que sea poco necesario que el usuario pueda utilizar ambos controles para la misma función.

■ Estado

- Aprobado

■ Capturas de Pantalla

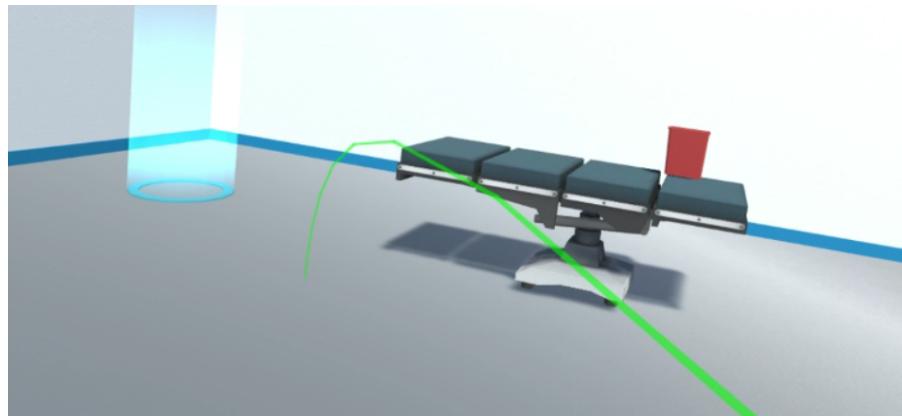


Figura 4.2: Línea verde indicadora desde el placeholder hacia un punto en el entorno virtual(Simulando una trayectoria)

■ id de Develop Realizados

- 3

■ Descripción de la Prueba

- Prueba de la capacidad de que el sistema permite el movimiento dentro del entorno virtual mediante teleportación.

■ Condiciones de la Prueba

- El usuario, desde una posición inicial, toca el stick de cualquier control y realiza el movimiento del control izquierdo o derecho, uno a la vez, dirigiendo el arco de indicación de localización de teleportación en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z, indicando la localización hacia la cual desea y presionando el stick respectivo para realizar la acción de teleportación. Recibiendo el movimiento de los mismos y este será mostrado dentro del HMD, demostrado por la nueva ubicación en el entorno virtual.

■ Observaciones

- Ya se utilizan más entradas del control Oculus Touch ® estas pueden llegar a no ser tan intuitivas y presentar dificultades de ejecución para el usuario. La curva indicando la dirección y la posterior reubicación de del usuario en la ubicación previamente elegida por el mismo.

■ Estado

- Aprobado

■ Capturas de Pantalla

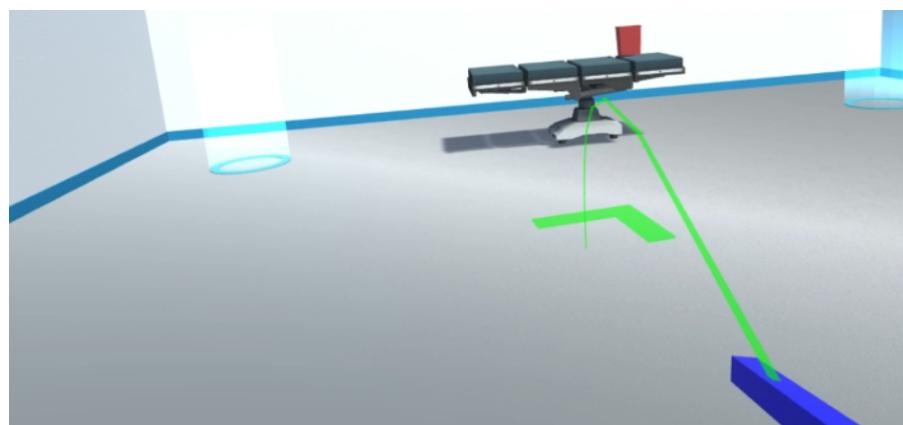


Figura 4.3: Uso de la curva indicadora para especificar un lugar y una orientación hacia donde se teleportará el usuario.

■ id de Develop Realizados

- 3

■ Descripción de la Prueba

- Prueba de la capacidad de que el sistema permite el movimiento dentro del entorno virtual mediante teleportación.

■ Condiciones de la Prueba

- El usuario, desde una posición inicial, toca el stick de cualquier control y realiza el movimiento del control izquierdo o derecho, uno a la vez, dirigiendo el arco de indicación de

localización de teleportación en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z, indicando la localización hacia la cual desea y presionando el stick respectivo para realizar la acción de teleportación. Recibiendo el movimiento de los mismos y este será mostrado dentro del HMD, demostrado por la nueva ubicación en el entorno virtual.

■ Observaciones

- Ya se utilizan más entradas del control Oculus Touch ® estas pueden llegar a no ser tan intuitivas y presentar dificultades de ejecución para el usuario. La curva indicando la dirección y la posterior reubicación de del usuario en la ubicación previamente elegida por el mismo.

■ Estado

- Aprobado

■ Capturas de Pantalla

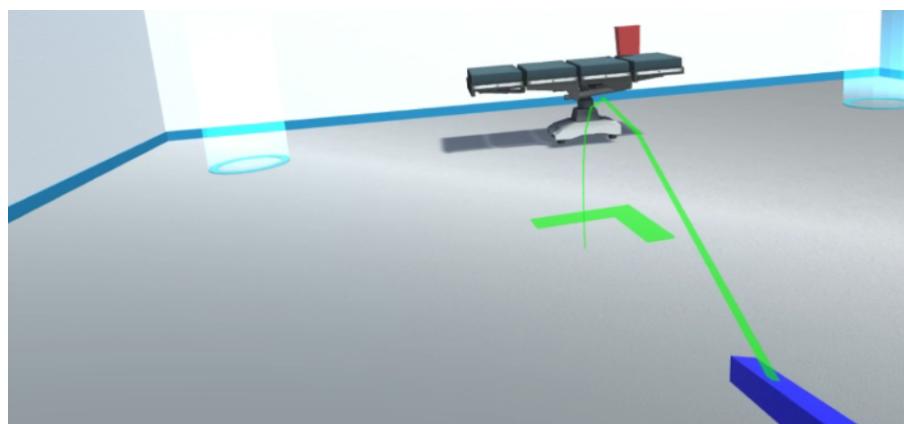


Figura 4.4: Uso de la curva indicadora para especificar un lugar y una orientación hacia donde se teleportará el usuario.

■ id de Develop Realizados

- 4

■ Descripción de la Prueba

- El usuario puede elegir la dirección en la cual desea reaparecer después invocar la teleportación, esto será indicado mediante una flecha la cual hará referencia a la dirección la cual el usuario habrá determinado anteriormente.

■ Condiciones de la Prueba

- El usuario, desde una posición inicial, toca el stick de cualquier control y realiza el movimiento del control izquierdo o derecho, uno a la vez, dirigiendo el arco de indicación de localización de teleportación en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z, indicando la localización hacia la cual desea, posteriormente moviendo el stick en los ejes X y Z para elegir la dirección en la cual quiere mirar al determinar el proceso

de teleportación, posteriormente soltando el stick respectivo para realizar la acción de teleportación. Recibiendo el movimiento de los mismos y este será mostrado dentro del HMD, demostrado por la nueva ubicación y dirección en el entorno virtual.

- **Observaciones**

- Se ha actualizado la manera de interactuar con el proceso de teleportación, así este se pretende que sea más intuitivo para el usuario reduciendo la capacidad de movimientos posibles y agregando controles que ayudan a la inmersión en el entorno virtual.

- **Estado**

- Aprobado

- **Capturas de Pantalla**

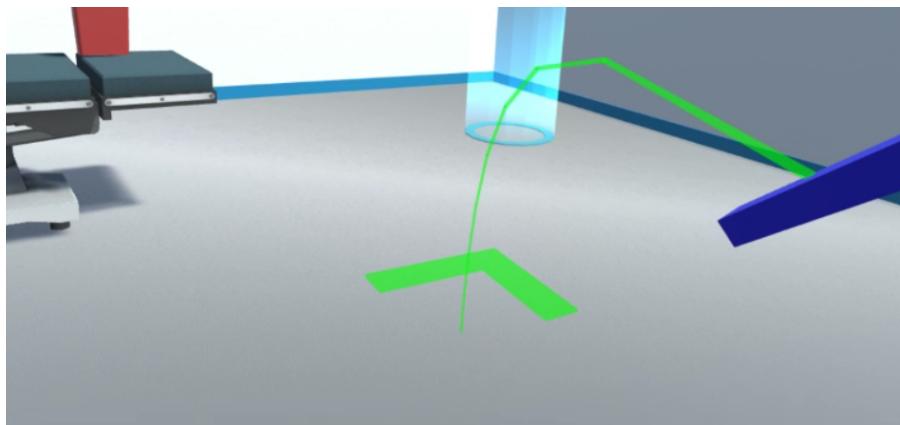


Figura 4.5: Flecha verde en el punto de teleportación, indicadora de la dirección hacia donde estará orientado el usuario una vez teletransportado.

- **id de Develop Realizados**

- 5

- **Descripción de la Prueba**

- Funcionamiento correcto de puntos de teleportación.

- **Condiciones de la Prueba**

- El usuario, desde una posición inicial, toca el stick de cualquier control y realiza el movimiento del control izquierdo o derecho, uno a la vez, dirigiendo el arco de indicación de localización de teleportación en el espacio en tres dimensiones moviéndose en los tres ejes X, Y y Z, indicando la localización hacia la cual punto de teleportación desea, posteriormente moviendo el stick en los ejes X y Z para elegir la dirección en la cual quiere mirar al determinar el proceso de teleportación, posteriormente presionando el stick respectivo para realizar la acción de teleportación. Recibiendo el movimiento de los mismos y este será mostrado dentro del HMD, demostrado por la nueva ubicación y dirección en el entorno virtual.

■ Observaciones

- Se ha actualizado la manera de interactuar con el proceso de teleportación, así este se pretende que sea más intuitivo para el usuario reduciendo la capacidad de movimientos posibles y agregando controles que ayudan a la inmersión en el entorno virtual. La utilidad de los puntos de teleportación y su uso depende del diseño que haya disponible en el entorno virtual, estos pueden resultar muy beneficioso en el momento de resaltar un lugar el cual se quiere que el usuario se acerque para que realice una situación específica en el sitio de interés.

■ Estado

- Aprobado

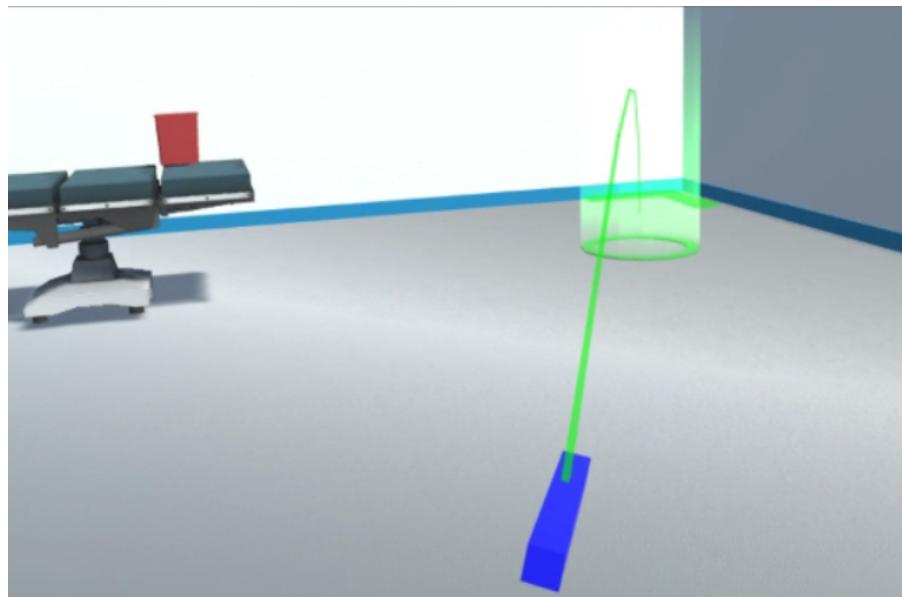
■ Capturas de Pantalla

Figura 4.6: Punto de teleportación marcando un lugar específico y preestablecido hacia donde se teleportará.

■ id de Develop Realizados

- 6

■ Descripción de la Prueba

- El usuario puede cambiar de dirección de orientación sin la necesidad de realizar una teleportación.

■ Condiciones de la Prueba

- El usuario, desde una posición inicial, mueve el stick del control izquierdo en los ejes X y Z recibiendo el movimiento del mismo y este será mostrado dentro del HMD generando una pequeña alteración de orientación en los ejes antes mencionados.

- **Observaciones**

-

- **Estado**

- Aprobado

- **Capturas de Pantalla**

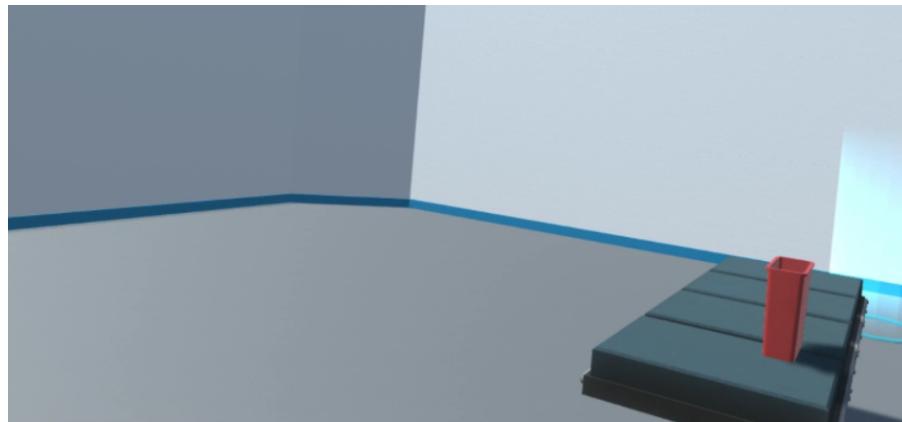


Figura 4.7: Vista descriptiva hacia una orientación en específico.

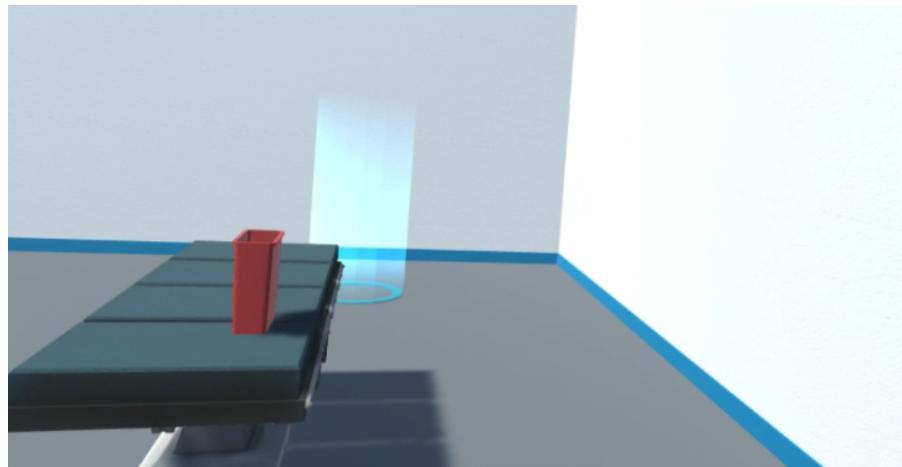


Figura 4.8: Vista descriptiva de un cambio de orientación respecto a la imagen anterior. Sin necesidad de teleportarse.

- **id de Develop Realizados**

- 7

- **Descripción de la Prueba**

- El usuario no es capaz de introducirse en el entorno virtual en lugares no admitidos por el programa, así como permite.

■ Condiciones de la Prueba

- El usuario, después de haberse desplazado de una posición inicial, con los comandos anteriormente mencionados intenta explorar fuera del entorno virtual “exponiendo” su cabeza fuera de los límites del entorno designado, lo cual causará una oclusión del entorno para evitar que este pueda salir del área designada. Asimismo con objetos en los cuales queremos que pueda ver a través de ellos quedan exentos de las limitaciones anteriores.

■ Observaciones

-

■ Estado

- Aprobado

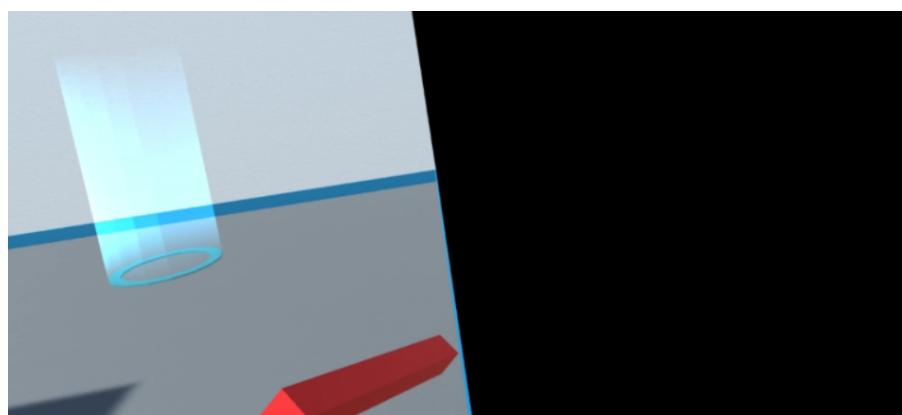
■ Capturas de Pantalla

Figura 4.9: Vista descriptiva del impedimento hacia afuera del entorno virtual admitido.

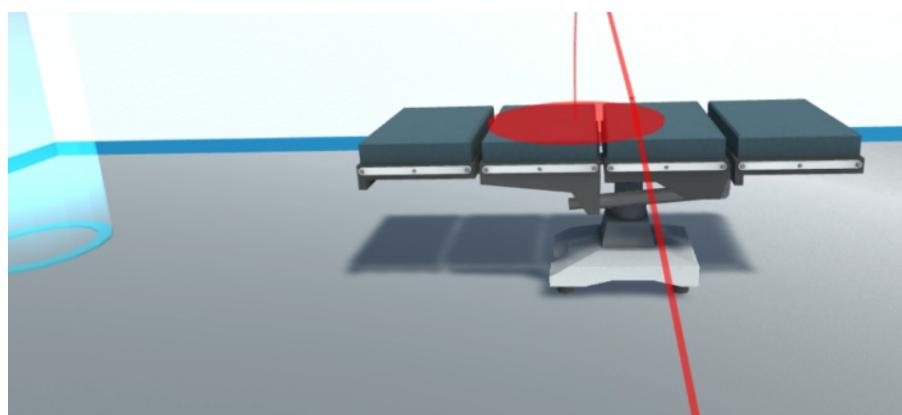


Figura 4.10: Punto rojo sobre una superficie no permitida en el entorno virtual, impidiendo la tele-portación.



Figura 4.11: Vista del usuario antes de intentar introducirse a un entorno virtual no permitido.

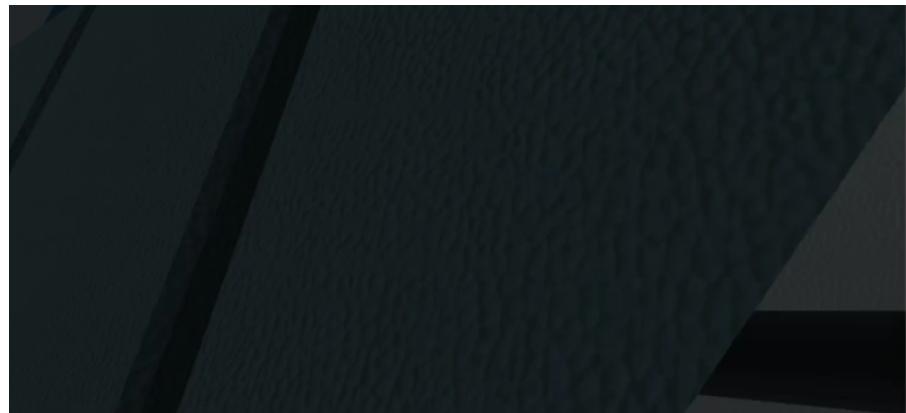


Figura 4.12: Vista del usuario intentando introducirse a un entorno virtual no permitido. La vista se empieza a oscurecer.



Figura 4.13: Vista del usuario introduciéndose en un entorno virtual no permitido, la vista está completamente oscura

4.3. Presencia e interacción de manos

- **id de Develop Realizados**

- 8

- **Descripción de la Prueba**

- El usuario es capaz de realizar los movimientos permitidos de las manos haciendo uso de los botones y sensores de los controles.

- **Condiciones de la Prueba**

- Los modelos virtuales de las manos imitan los movimientos que son realizados al tomar los controles e interactuar con los sensores y botones de los mismos, estos son limitados pero las acciones asemeja el movimiento de la mano de manera consistente e interactiva. Acciones como señalar, tomar, saludar, levantar dedos están incluidas dentro de este y son apreciadas por el usuario en el HMD en cuanto estas acciones son realizadas.

- **Observaciones**

- Los movimientos se encuentran limitados a las capacidades del control mismo, aunque comparados con interacciones "antiguas" de realidad virtual, estas proveen una experiencia de inmersión más real dentro del sistema de realidad virtual.

- **Estado**

- Aprobado

- **Capturas de Pantalla**

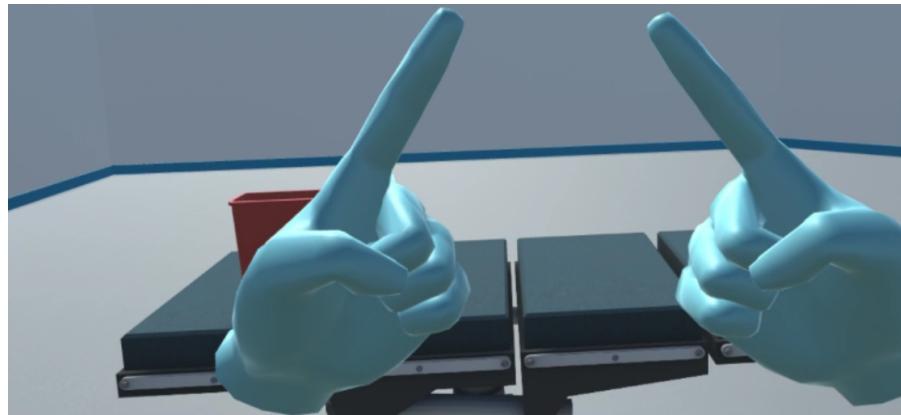


Figura 4.14: Modelo de manos, mostrando cómo se alza el dedo índice cuando el usuario lo hace a través de los sensores y botones del control.

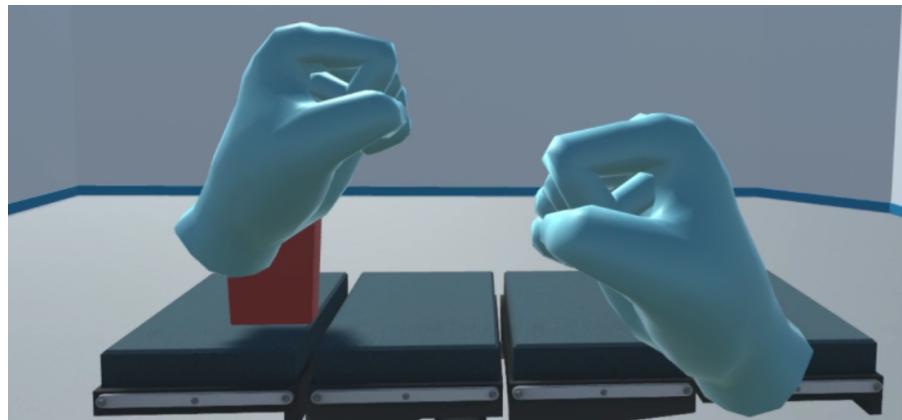


Figura 4.15: Modelo de manos, mostrando la mano cerrada cuando el usuario lo hace a través de los sensores y botones del control.

4.3.1. Interactuando con entorno

- **id de Develop Realizados**

- 9

- **Descripción de la Prueba**

- El usuario es capaz de tomar objetivos específicos designados como pruebas de entorno con las manos.

- **Condiciones de la Prueba**

- Al acercarse a un objeto de prueba específico el usuario es capaz de tomarlos e inspeccionarlos. Al tomarlo debe de ser capaz de dejar el objeto en un lugar diferente del cual fue tomado.

- **Observaciones**

- Las características de un objeto, entiéndase, que este pueda ser interactuado con las manos virtuales pueden ser transferidas a más modelos 3D lo cual nos da la libertad de interacción con una mayor cantidad de objetos. La prueba fue realizada exitosamente.

- **Estado**

- Aprobado

- **Capturas de Pantalla**

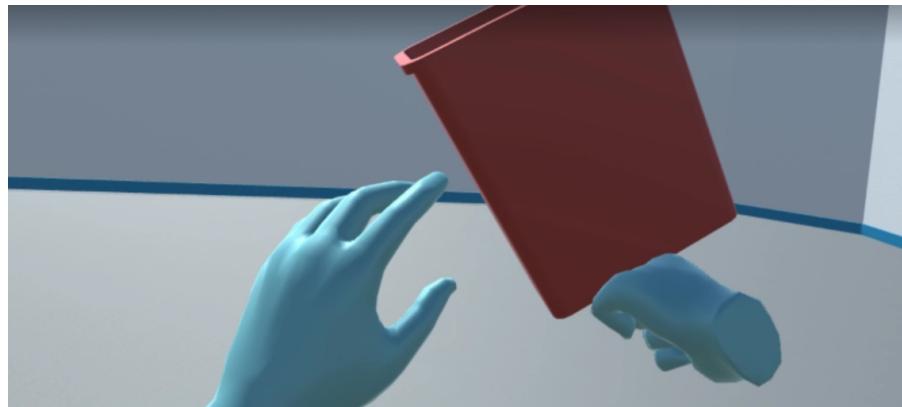


Figura 4.16: Previo a la toma de un objeto

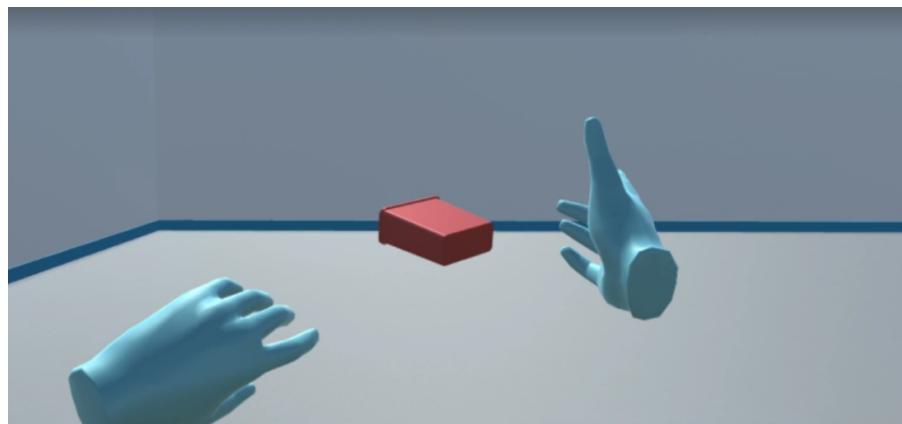


Figura 4.17: Toma de un objeto antes y después de lanzarlo

4.3.2. Interacciones manuales adicionales

- **id de Develop Realizados**

- 10

- **Descripción de la Prueba**

- El usuario es capaz de cambiar de manos el objeto que se tiene tomado y de arrojar el mismo para proveer una experiencia más interactiva dentro del entorno virtual.

- **Condiciones de la Prueba**

- El usuario debe de tomar un objeto interactivo con una mano y después de esto cambiará de mano el objeto, posteriormente dejará el objeto y lo tomará de nuevo con la mano y lo arrojará frente a él.

- **Observaciones**

- Las características de un objeto, entiéndase, que este pueda ser interactuado con las manos virtuales pueden ser transferidas a más modelos 3D lo cual nos da la libertad de interacción con una mayor cantidad de objetos.

- **Estado**

- Aprobado

- **Capturas de Pantalla**

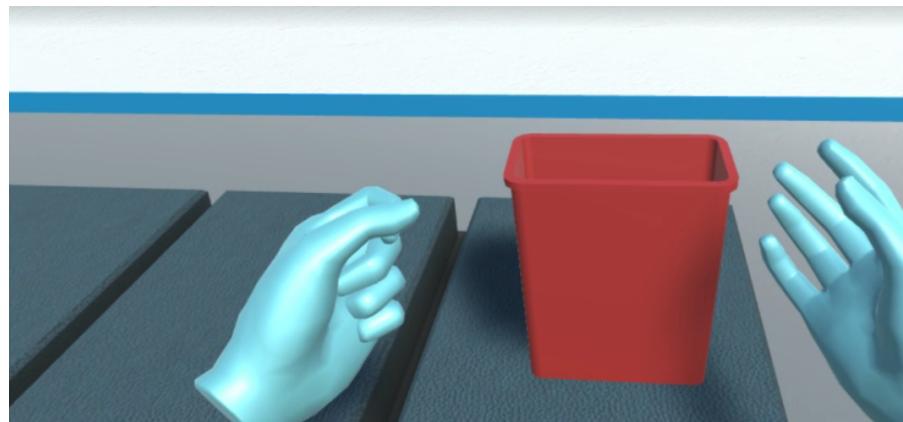


Figura 4.18: Muestra las manos y un objeto antes de ser tomado por las manos

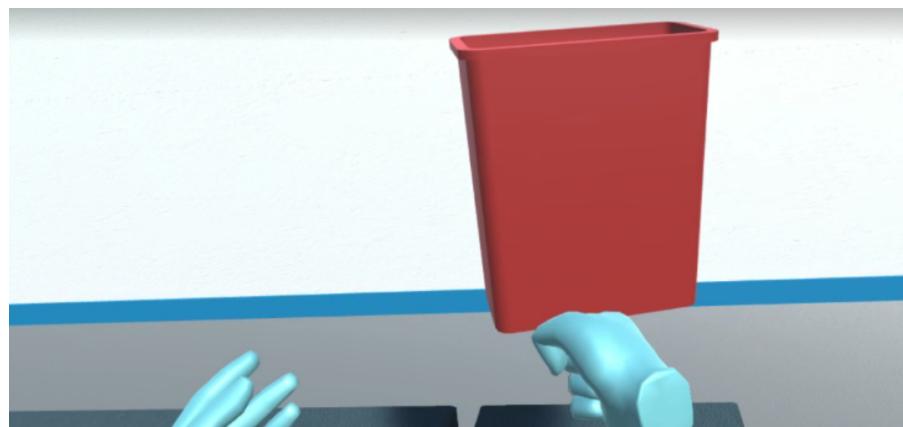


Figura 4.19: Una mano tomando y levantando el objeto

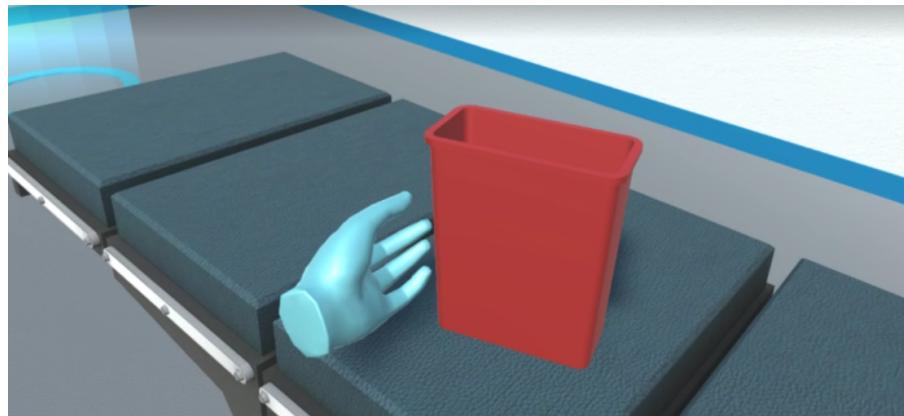


Figura 4.20: Después de pasar el objeto de una mano a otra se vuelve a soltar en un lugar distinto en el que estaba.

4.4. Retroalimentación de los stakeholders y usuarios potenciales

Debido a la situación global que se ha vivido durante el año 2020 comenzando en el mes de marzo la posibilidad de realizar pruebas presenciales con los stakeholders y usuarios potenciales se vio mermada de gran manera, lo cual desembocó en poca retroalimentación por parte de los usuarios finales, profesor de seguimiento, directores y sinodales.

Los problemas que surgieron son inherentes a la naturaleza del proyecto mismo, ya que al ser un sistema de realidad virtual, para ser probado se necesita la presencia física del usuario para que éste evalúe y experimente el sistema con todas sus características, así como sus capacidades y áreas de oportunidad en las cuales se debería de invertir más tiempo o cambios que se requieran realizar para mejorar el resultado final del proyecto, esto dependiendo completamente de la experiencia de uso que el stakeholder obtenga al realizar las pruebas con el sistema mismo.

Por lo tanto toda la retroalimentación del proyecto proviene únicamente del desarrollador, afectando el resultado final y dejando una versión menos refinada del mismo.

Capítulo 5

Conclusión

5.1. Conclusión

Como sucede con la mayoría de los sistemas de software, quizás al seguir siendo un campo de desarrollo tan reciente, sea esa la razón de porqué la gente suele infravalorar la cantidad de tiempo, recursos y esfuerzo que supone la puesta en obra. A diferencia de otros trabajos terminales, este ha sido una incursión fuera del software “tradicional” que se propone frecuentemente en la Escuela Superior de Cómputo, en este caso, el tiempo invertido en el desarrollo, investigación de un área completamente ajena a mi formación y tiempo invertido en la experiencia de usuario, áreas en las cuales puedo decir que es poco común que se centren las unidades académicas de la escuela, aunado a esto dadas las actividades que se requieren para un proyecto de esta naturaleza, el tiempo y los recursos así como el esfuerzo necesario implicaría quizás como en muchos desarrollos de este tipo un equipo de desarrollo completo para llegar a el estado “final” del mismo.

Podría concluir que la necesidad de crear un sistema innovador y poco recurrido para los trabajos terminales es importante a la hora de desarrollar cualquier cosa.

En general las habilidades que no forman parte explícitamente el desarrollo de software no suelen ser tomados en cuenta y frecuentemente son tratadas como no necesarias para el estudiante y profesores. Quiero mencionar que, sobre todo, y como en los sistemas de software tradicionales, la planeación y diseño es fundamental, y que no se debe de dejar de lado, incurrir en un sistema con tecnología tan nueva y utilizada en un menor grado para Trabajos Terminales complica el desarrollo del mismo.

La abstracción de un sistema de realidad virtual, el cual tiene una línea muy delgada para ser catalogado un videojuego y muchas veces es catalogado como un serious game, el cual al momento de su desarrollo se pueden definir elementos concisos y enlistarlos para su planeación, pero cada feature debe nacer de un concepto previo, debe tener un propósito y consideración. Es importante tener enfoque en el crecimiento del proyecto. En el desarrollo de experiencias de realidad virtual, la composición de elementos es una forma natural para la construcción de dicho sistema, esto facilita un crecimiento del proyecto, pero es un arma de doble filo ya que al permitir esta libertad da paso a vicios y concentraciones, de ahí que todo venga de una planeación dividiendo los elementos de software y multimedias y se pensado a la posible expansión.

El proyecto, si bien es pequeño comparado a la mayoría de producciones comerciales que se encuentran en el mercado, hace ver claramente que el camino a seguir para el desarrollo de sistemas basados en realidad virtual son un campo que definitivamente requiere de una amplia gama de personas hábiles tanto en diseño, tecnología, investigación y planeación. Es un campo que requiere muchísimo desarrollo en cuanto a sus metodologías y formas de documentación para sistemas no “tradicionales”, aunque tengo la certeza de mientras más avance la tecnología y se creen más sistemas de esta índole el desarrollo irá creciendo, haciéndose cada vez más profesional y disponible para todos.

Capítulo 6

Referencias y Glosario

6.1. Glosario

- **VR:** Virtual Reality, en español, Realidad Virtual
- **RV:** Realidad Virtual
- **ICs:** Tecnologías de la Información y la Comunicación
- **WIMP:** del inglés window-icon-menu-pointing device
- **MRI:** Imagen por resonancia magnética del inglés Magnetic Resonance Imaging
- **CT:** Computed Tomography del inglés Tomografía axial computarizada
- **PET:** Tomografía por emisión de positrones del inglés Positron Emission Tomography
- **DICOM:** Imagen digital y comunicación sobre medicina. Digital Imaging and Communication On Medicine
- **HMD:** Casco de realidad virtual del inglés Head Mounted Display
- **UML:** Lenguaje de modelado unificado del inglés Unified Modeling Language
- **3D:** Tres Dimensiones del inglés 3 Dimensions
- **I/O:** Entrada y salida del inglés Input/Output
- **UX:** UX (por sus siglas en inglés User eXperience) o en español Experiencia de Usuario, es aquello que una persona percibe al interactuar con un producto o servicio. Logramos una buena UX al enfocarnos en diseñar productos útiles, usables y deseables, lo cual influye en que el usuario se sienta satisfecho, feliz y encantado.
- **VOD:** Del inglés Video On Demand, El vídeo bajo demanda, o televisión a la carta, es un servicio OTT de televisión. Esta modalidad de difusión de contenidos multimedia, permite al usuario acceder a un contenido concreto, en el momento que lo solicita, visualizandolo en línea en su dispositivo.
- **SDK:** Un kit de desarrollo de software (en inglés, software development kit o SDK) es generalmente un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto, por ejemplo ciertos paquetes de software, entornos de trabajo
- **RUP:** El Proceso Unificado de Rational o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software desarrollado por la empresa Rational Software, actualmente propiedad de IBM.