CIS1310TK02

# VRLS: **V**irtual **R**eality **L**aparoscopic **S**imulator

Pablo Enrique Quiñones Garcia

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**FACULTAD DE INGENIERÍA**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**BOGOTÁ, D.C.**

**2013**

CIS1310TK02

# VRLS: Virtual Reality Laparoscopic Simulator

**Autor:**

Pablo Enrique Quiñones García

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS REQUISITOS PARA OPTAR AL TITULO DE INGENIERO DE SISTEMAS

**Director**

Leonardo Flórez Valencia

**Jurados del Trabajo de Grado**

César Julio Bustacara Medina

Daniel Ricardo Suarez Vanegas

**Página web del Trabajo de Grado**

http://pegasus.javeriana.edu.co/~CIS1310TK02

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS

BOGOTÁ, D.C.

Mayo, 2013

**PONTIFICIA UNIVERSIDAD JAVERIANA**

**FACULTAD DE INGENIERIA**

**CARRERA DE INGENIERIA DE SISTEMAS**

**Rector Magnífico**

Joaquín Emilio Sánchez García S.J.

**Decano Académico Facultad de Ingeniería**

Ingeniero Jorge Luis Sánchez Téllez

**Decano del Medio Universitario Facultad de Ingeniería**

Padre Sergio Bernal Restrepo S.J.

**Director de la Carrera de Ingeniería de Sistemas**

Ingeniero Germán Alberto Chavarro Flórez

**Director Departamento de Ingeniería de Sistemas**

Ingeniero Rafael Andrés González Rivera

**Artículo 23 de la Resolución No. 1 de Junio de 1946**

*"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"*

## AGRADECIMIENTOS

En esta oportunidad quiero agradecer inicialmente a mi hermano, quien ha sido un gran apoyo y un gran ejemplo a seguir y que me ha enseñado que los objetivos que uno se propone se pueden lograr. Quiero agradecerle a mis padres por haberme dado la oportunidad de estudiar en una excelente universidad y por siempre guiarme y apoyarme en las metas que me he propuesto a lo largo de mi vida.

Quiero agradecer de manera muy especial a mi director del trabajo de grado, Leonardo Flórez, por la confianza brindada para lograr concluir con este gran proyecto, por sus consejos y por todas las enseñanzas a lo largo del proceso.

Además quiero agradecer a el Dr. Fernando Alvarado, jefe de residentes del Hospital Universitario Sán Ignacio por atender la petición de validar el proyecto y los trabajos futuros en los que este podría ser incluido. También al residente de tercer año, Oscar David Rubio, quien muy amablemente verifió el simulador y comentó desde su experiencia como estudiante, el valioso aporte que esta herramienta podría ofrecer.

Finalmente quiero agradecer a la persona que me há acompañado durante mi proceso academico, quien comparte mis alegrías y mis triunfos y quien fue de gran ayuda durante este proyecto, mi novia, Monica Espinosa.

# Contenido


**I DESCRIPCIÓN GENERAL** **15**

**1. Problemática Identificada** **16**

1.1. Descripción de Oportunidad . . . . . 16
1.2. Pregunta de Investigación . . . . . 16
1.3. Justificación . . . . . 16
1.4. Impacto Esperado del Proyecto . . . . . 17

**2. Propuesta del Proyecto** **18**

2.1. Objetivo General . . . . . 18
2.2. Fases Metodológicas y Objetivos Específicos . . . . . 18
2.3. Proceso . . . . . 19
2.3.1. Fase Metodológica 1 . . . . . 19
2.3.1.1. Metodología . . . . . 19
2.3.1.2. Actividades . . . . . 19
2.3.2. Fase Metodológica 2 . . . . . 20
2.3.2.1. Metodología . . . . . 20
2.3.2.2. Actividades . . . . . 20
2.3.3. Fase Metodológica 3 . . . . . 21
2.3.3.1. Metodología . . . . . 21
2.3.3.2. Actividades . . . . . 21

**II ESTADO DEL ARTE** **22**

**3. Marco Contextual** **23**

3.1. Cirugías Mínima Invasión . . . . . 23
3.1.1. Descripción Temática . . . . . 23
3.1.2. Procedimientos . . . . . 24
3.2. Apendicectomía Laparoscópica . . . . . 25
3.2.1. Instrumentos . . . . . 25
3.2.2. Procedimiento [1] . . . . . 26

**4. Marco Teórico** **27**
4.1. Simuladores Médicos . . . 27
4.2. Modelo del Simulador . . . 27
4.2.1. Modelo . . . 28
4.2.1.1. Modelo de Datos . . . 28
4.2.1.2. ITK . . . 28
4.2.1.3. Procesamiento ITK . . . 29
4.2.1.4. Representación de los Datos . . . 29
4.2.1.5. Manejo de Lectura de Mallas . . . 30
4.2.1.6. ParaView . . . 30
4.2.2. Módulo Visualización . . . 30
4.2.2.1. Ogre3D . . . 31
4.2.2.2. Arquitectura General de Ogre3D . . . 31
4.2.2.3. Componentes de la Arquitectura de Ogre3D [2] . . . 32
4.2.2.4. Ambientes en Ogre3D [2] . . . 33
4.2.3. Módulo Interacción . . . 33
4.2.3.1. VRPN . . . 33
4.2.3.2. OpenHaptics . . . 35
4.2.4. Dispositivos de retroalimentación háptica . . . 40
4.2.4.1. Phantom Omni . . . 40

**III DESARROLLO DEL SIMULADOR VRLS** **41**

**5. Desarrollo del Simulador de Realidad Virtual** **42**
5.1. Fase I - VISUALIZACIÓN . . . 43
5.1.1. Requerimientos . . . 43
5.1.1.1. Investigación . . . 43
5.1.1.2. Modelo de Transformación [3] . . . 44
5.1.2. Prototipos . . . 46
5.1.2.1. Prototipo Inicial . . . 46
5.1.2.2. Segundo Prototipo . . . 47
5.1.2.3. Tercer Prototipo . . . 50
5.1.2.4. Cuarto Prototipo . . . 54
5.2. Fase II - INTERACCIÓN . . . 56
5.2.1. Investigación . . . 56
5.2.2. QuickHaptics Micro API . . . 56
5.2.3. Investigación II . . . 56
5.2.4. Haptic Device API . . . 57
5.2.5. Conclusión . . . 58
5.3. Fase III - INTEGRACIÓN . . . 58
5.3.1. Intergración HDAPI . . . 58

5.3.1.1. Manejo de eventos del dispositivo . . . . . 59

**6. Validación del Simulador** **63**
6.1. Proceso de Validación . . . . . 63
6.2. Comentarios de los expertos . . . . . 63
6.2.1. Médico Cirujano . . . . . 63
6.2.2. Residente HUSI . . . . . 64
6.2.3. Jefe de Residentes HUSI . . . . . 64
6.2.4. Análisis de la Validación . . . . . 65
6.3. Requerimientos de Validación . . . . . 65

**IV VRLS: VERSIÓN FINAL** **66**

**7. Versión Final del Prototipo** **67**
7.1. Modelo . . . . . 68
7.2. Módulo de Visualización . . . . . 68
7.3. Módulo de Interacción . . . . . 68

**8. Resultados del Proyecto** **69**
8.1. Cumplimiento de los Objetivos del Proyecto . . . . . 69
8.1.1. Objetivo General . . . . . 69
8.1.2. Objetivos Especificos . . . . . 70
8.2. Aporte del Proyecto . . . . . 70
8.3. Impacto Potencial del Simulador . . . . . 71

**9. Conclusiones, Recomendaciones y Trabajos Futuros** **72**
9.1. Conclusiones . . . . . 72
9.1.1. Respuesta a la Pregunta Generadora . . . . . 72
9.1.2. Cumplimiento de los Objetivos . . . . . 72
9.1.3. Aporte a la solución de la problemática . . . . . 72
9.1.4. Análisis Final . . . . . 72
9.2. Recomendaciones . . . . . 73
9.2.1. Para la Carrera . . . . . 73
9.2.2. Para la Universidad . . . . . 73
9.3. Trabajos Futuros . . . . . 73
9.3.1. Incluir nuevas funcionalidades . . . . . 73
9.3.2. Mejorar los algoritmos . . . . . 73
9.3.3. Incluir herramientas . . . . . 74
9.3.4. Un nuevo simulador . . . . . 74
9.3.5. Incluir nuevos dispositivos . . . . . 74

**V REFERENCIAS Y BIBLIOGRAFÍA** **75**

**VI ANEXOS** **78**

**10. Glosario** **79**

**11. Análisis Post-Mortem del Proyecto** **80**

11.1. Metodología Aplicada Vs. Metodología Propuesta . . . . . 80

11.2. Actividades Propuestas Vs. Actividades Realizadas . . . . . 80

11.3. Efectidad en la estimación de tiempos del proyecto . . . . . 80

11.4. Efectividad en la estimación y mitigación de los riesgos del proyecto . . . . 81

# Índice de Ilustraciones


4.1. Modelo Arquitectura VRLS . . . . . 28
4.2. Ejemplo de malla del simulador . . . . . 28
4.3. Procesamiento Imágenes ITK . . . . . 29
4.4. ITK VTKPolyDataReader Architecture . . . . . 30
4.5. Arquitectura Ogre3D . . . . . 31
4.6. Arquitectura General de OpenHaptics 3.0 . . . . . 35
4.7. Arquitectura QuickHaptics Micro API . . . . . 36
4.8. QuickHaptics Cursor . . . . . 38
4.9. QuickHaptics Application . . . . . 39
4.10. HLAPI Application & HDAPI Application . . . . . 40

5.1. Mallas de Ogre3D . . . . . 44
5.2. Transformación ITK a Ogre3D. . . . . 44
5.3. Fórmula Filtro Normales ITK . . . . . 45
5.4. Estómago: vista de superficie y vista de geometría . . . . . 46
5.5. Torso: vista de superficie y vista de geometría . . . . . 47
5.6. Transformación de Obj a Mesh . . . . . 48
5.7. Vista aérea . . . . . 49
5.8. Vista lateral . . . . . 49
5.9. Vista interna de los órganos . . . . . 50
5.10. RayCasting a nivel del polígono . . . . . 51
5.11. Diagrama Arquitectura . . . . . 51
5.12. Grasper ( Pinzas ) . . . . . 52
5.13. Scissors ( Tijeras ) . . . . . 53
5.14. Cámara Principal . . . . . 55
5.15. Cámara Panoramica . . . . . 55
5.16. Ejemplo I QuickHaptics MIcro API . . . . . 57
5.17. Datos de Interacción . . . . . 58
5.18. Modelo Integración . . . . . 59
5.19. Manejador de Eventos del Dispositivo . . . . . 60

7.1. Diagrama de componentes del simulador . . . . . 67

# Índice de Tablas

8.1. Tabla Análisis Objetivos Específicos . . . . . 70

11.1. Tabla de riegos del proyecto . . . . . 81

## ABSTRACT

*In medicine, the minimal invasive surgery procedures had become one of the most important techniques. Thousands of patients had chosen this type or surgery over others because of its advantages. The training process on this type of surgery is very difficult for medicine students, so this work suggests a technological tool created to help in this process. This tool is a VR-based simulator that lets the doctor practice over a virtual body in order to help improving its abilities in a laparoscopic procedure.*



## RESUMEN

*En medicina, los procedimientos de cirugía de mínima invasión han adquirido gran importancia. Son procedimientos que buscan disminuir los riesgos presentes en las cirugías y que mejoran el proceso de recuperación. Para realizar este tipo de cirugía, es fundamental realizar un proceso de entrenamiento, el cual, es bastante complejo. VRLS es una herramienta informática pensada para apoyar ese proceso y para ayudar a mejorar las habilidades de los médicos cirujanos. El simulador recrea una escena de cirugía laparoscópica y permite la interacción a través de un dispositivo manual háptico que genera la sensación de tacto sobre los objetos presentes en la escena.*

## RESUMEN EJECUTIVO

El concepto de Cirugías de Mínima Invasión nace de la necesidad de mejorar las técnicas con las que se llevan a cabo los procesos quirúrgicos. Se há enfocado en buscar formas de mínimizar riesgos y de mejorar el estado final de los pacientes luego de someterse a una cirugía.

A partir de esto, se presenta un problema al momento de lograr llevar a cabo este tipo de cirugías, ya que esta limitado por la complejidad de este tipo de procedimientos. El conocimiento requerido para operar sin tener un acceso completo a la zona afectada es bastante complejo y arduo de adquirir, por lo que ha sido fundamental encontrar soluciones a este problema.

Una de las principales soluciones que há sido formulada, es lo que se conoce como los Simuladores Médicos, herramientas que recrean escenarios médicos para apoyar el entrenamiento de los cirujanos. Existen varios tipos de simuladores, en los que se destacan los de Realidad Virtual, los cuales por medio de un dispositivo y creando una escena virtual, generan la sensación de estar en un ambiente real.

VRLS es una herramienta informática que pertenece a esta categoria y que esta pensada para apoyar el proceso de aprendizaje de los residentes de cirugía. Lo que busca, es dar una primera idea de lo que es realizar una cirugía de este tipo y mejorar las habilidades de quienes lo utilicen. Incluye un dispositivo háptico, que genera retroalimentación de fuerzas sobre la mano del usuario permitiendole sentir lo que esta "tocando" en la escena virtual.

Al concluir el proyecto dió como resultado un prototipo funcional del simulador que le permite al usuario navegar en la escena virtual e interactuar con los órganos presentes a través del dispositivo háptico. Esta herramienta paso ppor un proceso de validación por parte de un especialista en la materia, que dió su perspectiva sobre esta herramienta y su futuro a mediano plazo.

# INTRODUCCIÓN

En la medicina, el tema de las cirugías ha sido muy importante a lo largo de los años, ya que a través de ellas se tratan y se atacan muchas de las enfermedades descubiertas que afectan al ser humano. Al ser estas de vital importancia para nuestras vidas, es aún más importante que las personas que tienen la responsabilidad de realizarlas tengas las capacidades suficientes para lograr excelentes resultados y para minimizar los errores comentidos durante las mismas.

Hace unos años se han venido haciendo estudios de como mejorar los procedimientos quirirgicos, dada la ubicación de la enfermedad y la manera en que el médico accede a la zona afectada a la que en la mayoria de los casos, se accesede a través de incisiones de gran tamaño. Esto trae consigo unas implicaciones, entre las cuales estan, la posibilidad de contraer infecciones al momento de la operación, por lo que los médicos deben ser bastante cuidadosos. Adicional a esto, después de realizadas este tipo de intervenciones, el daño requerido para poder acceder al sitio de la enfermedad, repercute en un tiempo de recuperación durante el cual es frecuente que se presenten complicaciones en el paciente.

La investigación ha llevado a que se empiecen a practicar las cirugías de mínima invasión, las cuales logran tener el mismo resultado que las cirugías convencionales, solo que no requieren que el médico tenga acceso a la zona afectada diretamente, sino que realizando pequeñas incisiones sobre el paciente y a través de herramientas especializadas, se pueden manipular los órganos y con una pequeña cámara se pueden visualizar. Una de las limitaciones que tiene este tipo de cirugía es que las habilidades que este procedimiento requiere son muy altas, por lo que la preparación debe ser muy completa y rigurosa. Sumado a esto, los mecanismos para adquirir este conocimiento son limitados, ya que los estudiantes, al comienzo, realizan prátcias sobre cadáveres, lo que tiene también cierto problema ético.

Es por todo esto que nace este proyecto, en el cual se busca dar una mano creando una herramienta tecnológica que es el campo que nos compete y aportar una ayuda para ese proceso de adquirir habilidades en este tipo de cirugías. En el contexto de la universidad, no se había llevado a cabo un proyecto de estan naturaleza, ya que anteriormente se han venido realizando proyectos de investigación sobre diferentes enfermedades y de el mejoramiento en el proceso de visualización de imágenes médicas.

Durante el desarrollo del documento se mostrará el objetivo que se planteó, el proceso que se llevó a cabo y los resultados que el mismo arrojó.

# Parte I

# DESCRIPCIÓN GENERAL

# Capítulo 1

# Problemática Identificada

## 1.1. Descripción de Oportunidad

En el campo de la medicina se buscan oportunidades que puedan mejorar los procedimientos que en la actualidad se practican para mejorar la seguridad de las personas. En esa constante busqueda, aparece un campo de la medicina en el que se quizo basar este proyecto, este es el campo de las cirugías de mínima invasión.

Lo que la medicina quiere lograr con las cirugias de mínima invasión es disminuir el número de complicaciónes y el riesgo que trae consigo una cirugía abierta convencional.

En el marco de este proyecto aún no se tratado este tema y por eso surge como un tema relevante para el grupo de investigación y lo que se busca con esto es apoyar de diferentes formas la labor que realizan los médicos a la hora de operar a sus pacientes.

## 1.2. Pregunta de Investigación

Con este trabajo de grado se propone hacer un aporte con una herramienta tecnológica al aprendizaje del proceso de cirugía de mínima invasión.

La pregunta que dió lugar a este trabajo de grado fue la siguiente:

*¿Cómo un simulador basado en un ambiente de realidad virtual de una laparoscopia digestiva puede contribuir al aprendizaje de un medico cirujano?*

## 1.3. Justificación

La propuesta es una solución informática que ayude y contribuya a ese aprendizaje de los médicos cirujanos de mínima invasión. Esa solución sería un prototipo que permita la simulación virtual de una laparoscopia digestiva y que permita su interacción a través de un

dispositivo llamado phantom, del cual se especificara mas adelante en el documento, que simularía las herramientas que son introducidas en el paciente.

Este simulador a través de una combinación de retroalimentación háptica y de una interfaz visual en 3D permite a los médicos estudiantes y a los residentes a realizar procedimientos complejos antes de tener que enfrentarse a situaciones similares en la vida real. Además estas simulaciones al poder repetirse y ejecutarse constantemente permite que los médicos estudiantes y/o los residentes puedan tener herramientas para preparar un mejor diagnóstico y planear como actuar en diferentes situaciones que se podrían presentar. Todo esto contribuye a reducir el número de errores humanos cometidos durante estas intervenciones quirúrgicas. **[4]**

Adicionalmente, realizar el entrenamiento medico a través de un ambiente de realidad virtual da la posibilidad de realizar medición sobre las habilidades del practicante. Los métodos tradicionales de medición de las habilidades dependen de la apreciación de un experto que observa el procedimiento completo y decide de acuerdo a su experiencia. El uso de un simulador permite generar mediciones objetivas como el tiempo que se toma en realizar una actividad del procedimiento, la cantidad de errores cometidos durante el proceso y hasta la eficiencia con que se realizaron las actividades de la simulación. **[5]**

## 1.4. Impacto Esperado del Proyecto

La idea principal de este proyecto es lograr tener una herramienta que permita realizar una laparoscopia digestiva, donde el ambiente virtual es generado de imágenes reales y con un modelo de interacción con retroalimentación háptica, el cual permita realizar actividades sencillas sobre los órganos internos del cuerpo humano.

Adicional a esto, el objetivo principal de este proyecto es que se cree una herramienta que ayude a los médicos a mejorar sus habilidades y que los pacientes que requieren de una intervención de este tipo, se vean beneficiados con todo lo que esta herramienta puede aportar.

Este tema, en la ultima década ha adquirido gran relevancia y la idea es que este proyecto pudiera servir como un aporte a la comunidad médica que esta trabajando e investigando en este tema.

# Capítulo 2

# Propuesta del Proyecto

## 2.1. Objetivo General

Desarrollar una prototipo funcional basado en un ambiente de realidad virtual para simular un procedimiento de una laparoscopia digestiva que sirva como apoyo al proceso de aprendizaje y pueda aportar para mejorar las habilidades de los estudiantes de medicina y/o residentes.

## 2.2. Fases Metodológicas y Objetivos Específicos

**1. Fase de Investigación**

1.1 Investigación del procedimiento de laparoscopias digestiva.

1.2 Estudio del toolkit OpenHaptics.

1.3 Investigación librerías para generación de ambientes de realidad virtual.

1.4 Investigación de algoritmos para generación de ambientes de realidad virtual.

1.5 Estudio del dispositivo Phantom Omni.

**2. Fase de Diseño y Desarrollo**

2.1. Diseño del modelo del simulador virtual.

2.2. Diseño del modelo de interacción.

2.3. Desarrollo del ambiente de realidad virtual.

2.4. Desarrollo del modelo de Interacción.

**3. Fase de Validación y Pruebas**

3.1. Validar el simulador en su etapa final con el experto.

## 2.3. Proceso

### 2.3.1. Fase Metodológica 1

**Fase de Investigación**

- Investigación del procedimiento de laparoscopias digestiva.
- Estudiar la documentación de la librería OpenHaptics.
- Investigación librerías para generación de ambientes de realidad virtual.
- Investigación de algoritmos para generación de ambientes de realidad virtual.
- Estudio del dispositivo Phantom Omni.

#### 2.3.1.1. Metodología

La metodología que se aplicará para cumplir con los objetivos de esta fase es la de **Revisión Bibliográfica**, con la cual se buscarán y se seleccionarán las fuentes valiosas para este Trabajo de Grado. Lo que se busca en esta fase es realizar una investigación de material bibliográfico base, del cual se pueda hacer un análisis comparativo de los diferentes algoritmos y librerías existentes en la actualidad para el desarrollo de ambientes de realidad virtual y que a partir de esto pueda tener la información suficiente para escoger los más apropiados para el desarrollo del simulador propuesto.

#### 2.3.1.2. Actividades

La lista de actividades que se llevarán a cabo y se programarán para cumplir con esta primera fase metodológica es la siguiente:

- Recopilación y clasificación de material bibliográfico médico.
- Recopilación y clasificación de material bibliográfico sobre ambientes de realidad virtual.
- Lectura del material bibliográfico sobre laparoscopias digestivas.
- Lectura del material bibliográfico sobre generación de ambientes de realidad virtual.
- Lectura documentación de la librería OpenHaptics.
- Análisis de ejemplos de la librería OpenHaptics.

- Análisis comparativo de las librerías encontradas para desarrollo de entornos virtuales.
- Análisis comparativo de los algoritmos encontrados para desarrollo de entornos virtuales.
- Pruebas con el dispositivo Phantom Omni y los algoritmos de captura de movimiento.

### 2.3.2. Fase Metodológica 2

**Fase de Diseño y Desarrollo**

- Organización general del desarrollo del proyecto.
- Diseño del modelo del simulador virtual.
- Diseño del modelo de interacción.
- Desarrollo del ambiente de realidad virtual.
- Desarrollo del modelo de Interacción.

#### 2.3.2.1. Metodología

La metodología que se aplicara para cumplir con los objetivos de esta fase es: **Programación Extrema** [1], lo que permite que a medida que transcurra el proyecto se podrán incluir requerimientos que se hayan identificado, además no se le dedicara tiempo a la creación de documentos formales sino que las iteraciones estarán dedicadas a ir incluyendo funcionalidades necesarias del proyecto.

#### 2.3.2.2. Actividades

La lista de actividades que se llevarán a cabo y se programarán para cumplir con esta segunda fase metodológica es la siguiente:

- Definir el ambiente de pruebas del código.
- Definir un diseño inicial del ambiente de realidad virtual y del modelo de interacción que irá evolucionando a lo largo del proyecto.
- Definir un sistema de entregas parciales para ir analizando la evolución del proyecto.
- Definir el estándar de programación y documentación que se mantendrá durante todo el desarrollo.
- Definir el lenguaje de programación con el que se desarrollará el simulador.
- Desarrollar el modelo de interacción con la librería OpenHaptics.

[1] Ver *Extreme Programming*, Disponible en: *http://www.extremeprogramming.org/*

- Desarrollar el ambiente virtual del simulador. Hacer la integración entre el modelo de interacción y el ambiente virtual.

### 2.3.3. Fase Metodológica 3

**Fase de Validación y Pruebas**

- Validar el simulador en su etapa final con el experto.

#### 2.3.3.1. Metodología

La metodología que se aplicara para cumplir con los objetivos de esta fase es: **TAM (Modelo de Aceptación Tecnológica)**. Para esto, existen dos determinantes que sirven para tener una medida, estas son: la Utilidad Percibida y la Facilidad de Uso. Para este proyecto, estas dos unidades de medición van a ser muy validas, ya que el principal interés es que la herramienta en un futuro, sea escogida por los mismos médicos estudiantes y por los residentes para capacitarse y entrenarse en este tipo de procedimientos quirúrgicos.**[6]**

#### 2.3.3.2. Actividades

La lista de actividades que se llevarán a cabo y se programarán para cumplir con esta tercera fase metodológica es la siguiente:

- Ultimar pruebas sobre la versión final del simulador y su integración con el modelo de interacción desarrollado.
- Realizar una sesión final con el experto para realización de una prueba completa del simulador desarrollado con los dispositivos de interacción utilizados.

# Parte II

# ESTADO DEL ARTE

# Capítulo 3

# Marco Contextual

## 3.1. Cirugías Mínima Invasión

### 3.1.1. Descripción Temática

Las cirugías médicas son, en la actualidad, procedimientos de vital importancia para tratar los problemas de salud más complejos descubiertos. Para la realización de cirugías médicas, es fundamental que el médico poséa las habilidades requeridas para dicho procedimiento, esto hace que el proceso de capacitación en esta modalidad, sea vital en el desarrollo de un médico cirujano.

Antes de un médico poder comenzar a realizar cirugías sobre pacientes con problemas reales, debe pasar por un proceso de entrenamiento intensivo y debe realizar unas pruebas de evaluación que aseguren que el médico es apto para realizar los diferentes procedimientos quirúrgicos. [7]

En el campo de la medicina, existen múltiples técnicas para realizar cirugías, la más común y conocida es la técnica de cirugías abiertas, en las que es necesario realizar una incisión sobre el cuerpo del paciente para que el médico pueda manipular los órganos internos, estos quedan expuestos al ambiente y esto tráe consigo algunas consecuencias, entre las que se destacan la cicatrización y el riesgo de contraer una infección, el cual es mucho mayor que en otras intervenciones.

Otra técnica importante que existe y que es por la cual se da la oportunidad para este trabajo de grado es la de Cirugías de Mínima Invasión ó MIS, por sus siglas en ingles. Esta técnica se caracteriza por el uso de herramientas especiales, a diferencia de la anterior, no requiere de una gran incisión sobre el cuerpo del paciente, en esta, lo que se hacen son pequeñas incisiones sobre la zona del cuerpo que es necesario intervenir. Otra característica importante es, que el acceso a los órganos internos del paciente es muy limitado. Esta técnica tiene unas desventajas frente a las demás en cuanto a la dificultad, ya que las imágenes se muestran

sobre una pantalla, donde es difícil percibir la profundidad y no se visualiza el cuerpo entero. Es una limitante de este tipo de cirugías que las habilidades mínimas requeridas para practicarlas deben ser mucho mayores que las de un cirujano convencional. **[8]**

La laparoscopia es un tipo de cirugía de mínima invasión, la cual se caracteriza por el uso del laparoscopio, un instrumento médico que permite la exploración de los conductos y de los órganos. A través de este, se introduce la cámara que permite la visualización de la cavidad toráxica del paciente. En la laparoscopía también se requiere del uso de trocares, que es un instrumento médico para realizar una punción sobre el paciente, y este permite la inyección de dioxido de carbono en el abdomen para expandir el área donde se practicará la cirugía. Las herramientas que se utilizan solo permiten cinco grados de libertad, lo que restringe sus movimientos. Las pantallas donde se proyecta la imagen del paciente esta a cierta altura que le da una mayor dificultad a la cirugía. A su vez, la práctica de laparoscopias también tiene una variedad de ventajas que la hacen importante. Una de ellas es que al no ser una cirugía agresiva, el proceso post-operatorio no es tan doloroso como en una cirugía abierta, la recuperación de los pacientes toma menos tiempo y el riesgo de infección se reduce notablemente. [1]

Todo esto lleva a la conclusión de que la responsabilidad que se tiene al momento de hacer una cirugía de este tipo es bastante alta, por lo que el entrenamiento de los médicos debe ser riguroso para evitar que se presenten inconvenientes que puedan llevar a consecuencias graves como lo es la muerte de un paciente. Es aquí donde se centra el enfoque de este proyecto, ya que con este lo que se busca, es que a través del desarrollo de una ayuda tecnológica, los estudiantes de medicina y los residentes puedan mejorar sus habilidades y apoyar su diagnostico por medio de la interacción con el producto.

### 3.1.2. Procedimientos

En la actulidad hay varios procedimientos que están aceptados como adecuados para la cirugía laparoscópica, entre estos se encuentran la Colecistectomía, Bypass gástrico(Desviación gástrica), Esplenectomía y otras más. El procedimiento que se escogió para llevar a cabo en el proyecto es la **Apendicectomía Laparoscópica.**

"*Más del 90% de las apendicectomías que se practican en la actualidad, se realizan por apendicitis aguda, es decir, como un procedimiento urgente. Siempre que el equipo quirúrgico de urgencias esté habituado a la laparoscopia, es recomendable que se haga por esta vía*". [2]

El aporte que da este tipo de cirugía esta llevando a que se analicen más procedimientos para que sean aceptados como adecuados para la cirugía laparoscopica. Uno de los procesos que

[1] Ver *General and Digestive System*, Disponible en *http://www.surgeryandholidays.com/?page_id=48*

[2] Ver *¿Que es la Cirugía Mínimamente Invasiva?*, Disponible en: *http://cmiyo.com/index.php?id_categoria=11&id=9*

esta teniendo buenos resultados es la Nefrectomía en el donante (Extracción de riñon de un donante vivo para transplantarlo a un enfermo), ya que se reduce la morbilidad y el periodo de recuperación. **[9]**

## 3.2. Apendicectomía Laparoscópica

"*La apendicitis aguda es una de las patalogías más frecuentes en la cirugía de abdomen.*"[3] En respuesta a esto, se ha venido aplicando la apendicectomía laparoscópica como el método de atender estos casos. Consiste en la extirpación del apendice cecal por medio de pequeñas incisiones sobre el abdomen y el ombligo. Hay diferentes autores que han evualado la técnica y que han dado a conocer varias ventajas que éste ofrece. Las que más resaltan son: mayor exploración de la cavidad abdominal, mejor y más pronta recuperación postoperatoria, menor riesgo de complicaciones postoperatorias y pronto retorno a la actividad normal de los pacientes. Es importante mencionar que este procedimiento requiere que se suministre anestecia general sobre el paciente, por lo que es importante conocer de antemano si el paciente podría presentar algún riesgo, pero se ha analizado que en cualquier persona podría, en caso de ser necesario, ser realizado a través de este método.

### 3.2.1. Instrumentos

Para la realización de este procedimiento se requiere de una serie de instrumentos que permite la visualización, manipulación, extirpación y limpieza del órgano del paciente. Los elementos necesarios se dividen en dos(2) grupos principales, el circuito de video con óptica y el material para apendicectomía laparoscópica.

Para el circuito de vídeo con óptica puede ser de 30 o 45° y requiere de una cámara, un insuflador, un procesador de vídeo, una fuente de luz fría, un sistema de fotografía y un monitor.

Para el material del procedimiento se requiere de la óptica de 30 o 45°, un trocar de 10 mm, un trocar y mandril romo de 5 mm, una pinza bipolar coagulante de 5 mm, una pinza atraumática de 5 mm, unas tijeras de 5 mm, un empujanudos, un nudo prefabricado, un portaagujas, un palpador romo y una bolsa de recuperación para tejidos blandos. **[11]**

[3] Ver [10]

### 3.2.2. Procedimiento [1]

Los pasos básicos de una apendicectomía laparoscopica son: identificación completa del órgano, interrupción de su irrigación, ligadura de su base, revisión y eventual drenaje.

Para el paso de identificación es muy importante asegurar un buen acceso laparoscópico, el cual requiere introducir un trocar por la cicatriz umbilical y se inspecciona completamente la cavidad peritoneal. Con esto se busca descartar yatrogenias, confirmar el diagnóstico, descartar patologías asociadas de la esfera ginecológica y comprobar la factibilidad de llevar a cabo al operación. Se debe colocar un trocar en lo que se llama la fosa ilíaca derecha ( FID ) por el cual se introduce una pinza que va a manipular el apendice y se encargara de realizar su extracción.

Hay que realizar un control de hemostasia, o lo que sería un sangrado, antes de realizar el corte, se debe decidir el método por el que se va a hacer, se puede realizar un nudo con un elemento llamado endoloop que permite amarrar la base para evitar que haya un sangrado, se puede hacer una quemadura laser sobre la base para que la piel selle y se pueda extraer.

Una vez asegurada la base, se pasa a realizar la ligadura de la base, o lo que sería el corte. Finalmente se prepara para la extracción. Se puede realizar de multiples maneras, una de ellas es a través del uso de una bolsa de polietileno, donde se introduce el apendice y se extrae. Otra forma es directamente por el trocar por el que se introdujo la pinza para manipular el apendice, teniendo cuidado de no tener contacto con la pared abdominal.

Después de todo esto se hace una revisión de la zona donde se realizó la intervención, verificando que no se hayan causado lesiones y se pasa a hacer una limpieza de la zona donde se encontraba el apéndice. Si se considera necesario, se puede realizar un drenaje exteriorizado.

Al momento de retirar los trócares hay que prestar atención que no se haya generado alguna lesión sobre la pared abdominal que pudieran sangrar. Se deben suturar los orificios más grandes manteniendo la vigilancia laparoscópica y se colocan puntos de seda en los demás.

# Capítulo 4

# Marco Teórico

## 4.1. Simuladores Médicos

La laparoscopia es uno de los principales procedimientos médicos de la actualidad y por las bondades que ofrece, es de vital importancia adquirir los conocimientos necesarios para su ejecución. Es aquí, donde los simuladores han tomado una gran relevancia, ya que estos permiten realizar prácticas que buscan mejorar las ténicas de quienes lo realizan y minimizar los errores que se comenten por su alta complejidad. Los simuladores se pueden dividirse en tres(3) grupos: mécanicos, híbridos o de realidad virtual. En los mecánicos, se utiliza una caja que simula el abdomen en donde se introducirán los instrumentos quirúrjicos, que son similares a los que se usan en el procedimiento real. En los híbridos, se incluyen tejidos orgánicos en los modelos mecánicos, por lo que las estructuras se basan en tejido animal. Los simuladores de realidad virtual, que son los que usaremos para este proyecto, se definen como la interfaz entre el humano y el computador que simula un ambiente digital en tercera dimensión que permite la interacción con el especialista. **[12]**

En el caso de los simuladores de realidad virtual, existen herramientas diseñadas para la realización de gráficos 3D por computador, procesamiento de imágenes y visualización. Estas herramientas permiten generar los cuerpos virtuales que son sometidos a las cirugías laparoscópicas que servirán de entrenamiento para los médicos. (Amira, 3DSlicer, OSG)

## 4.2. Modelo del Simulador

La primera parte del proyecto era diseñar la arquitectura del simulador que sería desarrollado posteriormente. A partir de las características propuestas que deberia contener el simulador y a través de la investigación de herramientas que permitieran la construcción del simulador, se logró definir el siguiente modelo:

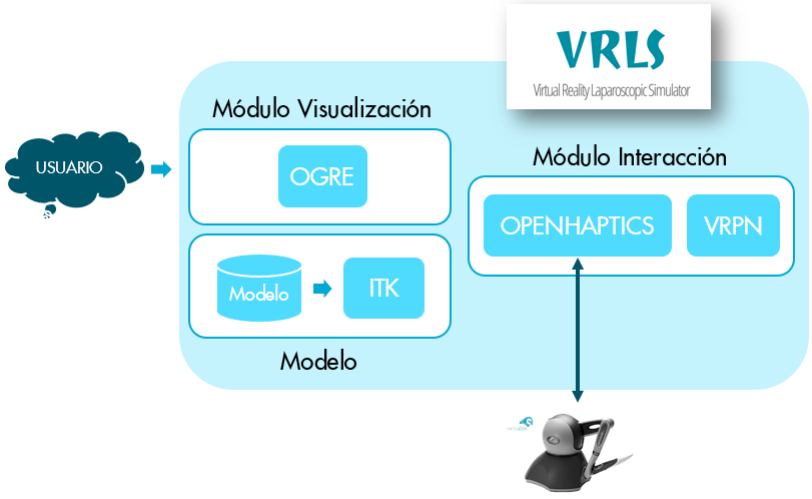


Figura 4.1: Modelo Arquitectura VRLS

### 4.2.1. Modelo

#### 4.2.1.1. Modelo de Datos

El componente principal del modelo de datos son las mallas, estructura de datos que respresenta un objeto. Se caracterizan por ser un conjunto de puntos que tienen bordes y colores. Estas mallas van a ser los datos de entrada al componente ITK para la creación de los órganos del simulador. Las mallas fueron tomadas de un repositorio de imágenes médicas pertenecientes a un laboratorio ubicado en Lyon,Francia[1].

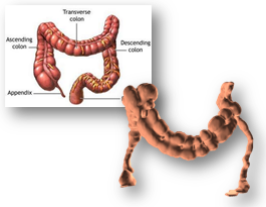

Figura 4.2: Ejemplo de malla del simulador

#### 4.2.1.2. ITK

El **kit de herramientas de registro y segmentación (ITK, por sus siglas en ingles),** es una herramienta de software libre que provee a los desarrolladores de una suite para el desarrollo de programas de segmentación y registro de imágenes. Generalmente esas imágenes son

[1] Ver: *http://www.websurg.com/softwares/vr-render/*

generadas por escaneos médicos como los TAC cerebrales o las resonacias magnéticas. El proyecto de ITK se desarrollo impulsado por la Biblioteca Nacional de Medicina de los Estados Unidos (NLM), y nació como un recurso libre de algoritmos diseñados para analizar las imágenes de un proyecto denominado "El Hombre Visible".

Este kit de herramientas esta diseñado en lenguaje C++ y al ser un proyecto de software libre, es una herramienta mantenida por la comunidad en general, que hace aportes para su continuo mejoramiento. Es una herramienta multiplataforma, lo que la hace compatible con sistemas operativos como Linux, Windows, Mac, entre otros. [2]

#### 4.2.1.3. Procesamiento ITK

ITK utiliza un pipeline para el procesamiento de los datos. La imagen es la variable de entrada que pasa por uno o más filtros que la procesan y generan una imagén resultado.

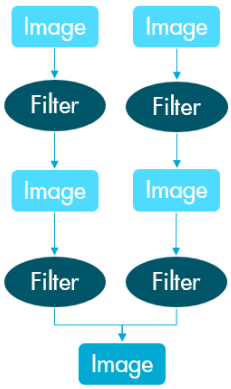


**Figura 4.3: Procesamiento Imágenes ITK**

#### 4.2.1.4. Representación de los Datos

Para la representación de los datos, ITK utiliza una clase principal llamada **"DataObject"**. Entre los tipos de datos que maneja ITK se encuentran:

- **Imagenes:** Grillas rectilineares de n- dimensiones que contiene la información de la imagen.
- **Meshes:** Colecciones de puntos de n-dimensiones unidos en células( e.g Triángulos ).

Para el procesamiento de imágenes de gran tamaño, ITK realiza un proceso de separación por regiones. Los tipos de regiones que define son los siguientes:

- **BufferedRegion:** Es la región de la imagen que esta siendo procesada.
- **RequestedRegion:** Es la región de la imagen que vá a ser procesada.
- **LargestPossibleRegion:** Es el conjuto de datos de la imagen.

[2] Ver *ITK* Disponible en: *www.itk.org*

#### 4.2.1.5. Manejo de Lectura de Mallas

Para leer mallas 3D, ITK proveé la clase: **itk::VTKPolyDataReader.** Una de las restricciones que tiene es que no puede leer archivos vtk binarios y que deben ser mallas triangulares, ya que utiliza un componente llamado vtkTriangleFilter que realiza el proceso

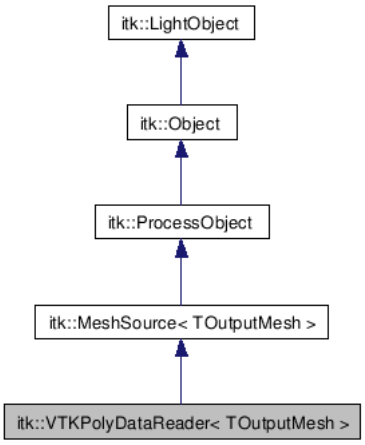


**Figura 4.4: ITK VTKPolyDataReader Architecture**

#### 4.2.1.6. ParaView

Paraview es una aplicación open-source que permite analizar y visualizar datos. Paraview fue diseñado para analizar grandes conjuntos de datos usando recursos de memoria distribuidos pero también, permite utilizarlo en computadores con imágenes de menor escala. [3]

Esta herrmienta se utilizó para generar las mallas de los órganos en formato VTK ASCII, ya que este era el formato soportado por ITK. **[13]**

### 4.2.2. Módulo Visualización

En la actualidad, existen varios motores de realidad virtual en el mercado, entre ellos se encuentra: Unity, Unreal Engine, Ogre3D, entre otros. Estos motores, no son mas que herramientas que permiten el diseño de ambientes de realidad virtual, esto quiere decir, que permiten recrear escenarios en los que los usuarios interactuan con el entorno a traves de un conjunto de dispositivos haciendo que perciba como si fuese real.

[3] Ver *Paraview*, Disponible en: *http://www.paraview.org/*

De todas las opciones que hay para desarrollar este tipo de aplicaciones, fue escogido Ogre3D, que además de ser una herramienta de software libre, permite hacer integración con otras librerias que serán explicadas más adelante en este documento.

#### 4.2.2.1. Ogre3D

**OGRE (Object-Oriented Graphics Rendering Engine),** es un motor de realidad virtual en 3D, orientado a escenas y flexible, que esta desarrollado en C++ y esta diseñado para producir aplicaciones que requieran gráficos 3D acelerados por hardware. Es una herramienta que ofrece una interfaz bastante comoda para adecuarla con otras herramientas y asi lograr crear aplicaciónes de todo tipo, desde juegos de video, aplicaciones de negocio, simuladores, entre otras. Esta bajo una licencia MIT que permite su uso y que no obliga a entregar los cambios hechos a la herramienta o las aplicaciones desarrolladas por medio del uso de la misma. [4]

La simulación de la apendicectomía laparoscópica va a ser realizada con mallas de Ogre3D, lo que va a permitir tener una integración con interfaces de interacción para el manejo de las herramientas quirúrjicas y de la cámara. Permite el manejo de luces en la escena, multiples vistas y navegación por el entorno.

#### 4.2.2.2. Arquitectura General de Ogre3D

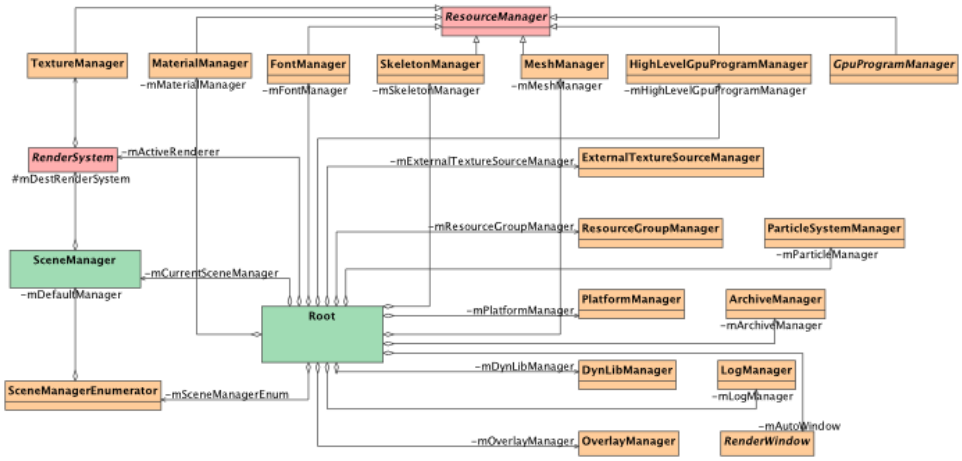


**Figura 4.5: Arquitectura Ogre3D**

[4] Ver *Ogre3D*, Disponible en: *www.ogre3D.org*

#### 4.2.2.3. Componentes de la Arquitectura de Ogre3D [2]

- **Root:** Este es el punto de partida del sistema de Ogre. Este elemento debe ser el primero en ser creado al momento de desarrollar una escena y debe ser el último que se destruya. Es quien permite toda la configuración del sistema. Es el director principal, por lo que es el encargado de dirigir a los principales manejadores: SceneManager( Manejador de Escena ), RenderSystem y el ResourceGroupManager( Manejador de Recursos ).

- **ResourceGroupManager:** Cualquier cosa que Ogre necesita para renderizar en una aplicación se le da el nombre de recurso, por lo que el Manejador de Recursos es el encargado de localizar e inicializar los recursos definidos en la aplicación.

  Estos son los recursos que permite Ogre:

  - **Mesh:** Son las mallas o geometrias, Ogre solo permite formato binario singular y deben tener la extensión .mesh para ser reconocidos por el manejador de recursos.
  - **Entity:** Instancia del mesh, que se vincula a los Nodos de Escena.
  - **Skeleton:** Representan una jerarquia de huesos, generalmente vienen unidos a un mesh y son usados para realizar animaciones. Deben estar definidos con la extensión .skeleton para ser reconocidos por el manejador de recursos.
  - **Material:** Representan el estado de las geometrias, su apariencia. Pueden estar incluidas en un mesh o pueden generarse a través de un script que debe tener la extensión .material para ser reconocidos por el manejador de recursos.

  Cada uno de estos recursos tiene un responsable o manejador correspondiente el cual se comunica directamente con el Manejador de Recursos.

- **SceneManager:** El manejo de escena diseñado para Ogre esta basado en un *Grafo de Escena* el cual esta controlado por este componente. A través de este diseño se permiten tener múltiples Manejadores de Escena para controlar la aplicación.

  Con este Manejador de Escena se tiene acceso a un tipo de dato especial llamado **SceneNode( Nodo de Escena ),** el cual esta encargado de manejar todas las transformaciones realizadas sobre las entidades que esten vinculadas a dicho nodo. Entre esas tranformaciones aparecen rotaciones, translaciones, escala, etc.

- **RenderSystem:** Es la interfaz de integracción entre Ogre y el API de hardware utilizado para la renderización( OpenGL ó Direct3D ). Es el encargado de definir la Render Window o ventana donde se visualizará la escena de la aplicación.

#### 4.2.2.4. Ambientes en Ogre3D [2]

**Creación de Entidades**

Al momento de iniciar la creación de una aplicación en Ogre, es importante conocer como se hace la inclusión de las mallas que se van a utilizar en el ambiente virtual. Para esto, primero se debe definir una variable de tipo Entity a la cual hay que asignarle **un nombre único** y definirle el archivo que contiene la malla, para Ogre, el .mesh. Esto se hace por medio del SceneManager que es quien permite la creación de entidades.

**Puesta en Escena**

Para poder ubicar la entidad creada en la escena se debe crear un nodo de escena, al cual se le debe asignar **un nombre único.** El SceneManager es el encargado de la creación de los nodos de escena. Adicional a esto, se puede escoger entre vincular el nodo de escena al nodo raiz de la aplicación o a otro nodo de escena, lo que permite generar la jerarquia en el Grafo de Escena. Finalmente se debe vincular la entidad creada con el nodo de escena.

Cuando un nodo de escena se vincula a otro, genera una dependencia en donde cualquier transformación que se realice sobre el nodo padre afectara de la misma forma al nodo hijo.

### 4.2.3. Módulo Interacción

Por su definición formal, *"una interfaz es una conexión física y funcional entre dos aparatos o sistemas independientes."* [5]. Cuando se trabaja con ambientes de realidad virtual, una de las características más importantes y en la que se debe enfocar quien lo diseña, es en la interfaz de interacción entre el ambiente generado y el usuario, quien será finalmente quien interactue con la aplicación. Para este proyecto, utilicé una librería de software libre que se llama VRPN, que al integrarla con Ogre3D me permite realizar interacción entre dispositivos externos( mouse, teclado, tracker, etc.) y la aplicación.

Para este proyecto se quizo hacer un aporte adicional, en el cual esta involucrado el dispositivo Phantom Omni. Para poder hacer la integración de este dispositivo y la aplicación fue necesaria la inclusión de una librería adicional llamada OpenHaptics.

#### 4.2.3.1. VRPN

**VRPN (Virtual-Reality Peripheral Network)** es un conjunto de clases que forman una librería con un conjunto de servidores que implementan una interfaz transparente a conexión e independiente de dispositivo entre las aplicaciones y el grupo de dispositivos

[5]Definición: *Interfaz* de *www.rae.es*

físicos utilizados para la interacción con entornos de realidad virtual, entre los que se destacan: dispositivos con botones, trackers, dispositivos análogos y dispositivos con retroalimentación de fuerzas.

Con esta libreria se busca ofrecer un camino de comunicación con el fin de lograr que los usuarios puedan relacionarse directamente con los entornos de realidad virtual y que esa información pueda ser interpretada para definir un comportamiento sin tener que preocuparse por el manejo de las conexiones de los dispositivos; interesante de esta librería es, que ofrece servidores que permiten el manejo de varios dispositivos en simultaneo actuando sobre la escena y también provee de una interfaz genérica para el control de eventos de los diferentes tipos de dispositivos. [6]

Para este proyecto, se penso en el uso de esta librería para desarrollar la interacción del usuario, en este caso, el médico en entrenamiento, con la simulación de la apendisectomía laparoscópica. El dispositivo principal que se piensa utilizar para dicha interacción es el Phantom Omni, que será descrito más adelante en este documento.

**Tipos de Dispositivos**

Una de las principales ventajas que ofrece VRPN es que no ofrece drivers para un conjunto de dispositivos, sino que proveé de interfaces para un conjunto de funciones. Define un grupo de tipos canonicos de dispositivos y además permite la creación de nuevos tipos de dispositivos. Los tipos de dispositivos que maneja son:

- **Analog:** Reporta uno o más canales análogos del dispositivo.
- **Button:** Reporta los eventos cuando se oprime y se suelta el botón o los botones.
- **Tracker:** Reporta poses ( posición y orientación ), su velocidad y/o su aceleración, dependiendo del dispositivo.
- **ForceDevice:** Permite al usuario especificar superficies y campos de fuerza en 3 dimensiones.

**Arquitectura Cliente-Servidor**

VRPN utilizar un servidor para atender las conexiones de los dispositivos y un cliente para hacer uso de ellos.

Para la configuración del servidor es importante revisar el archivo de configuración que trae ( vrpn.cfg ), donde se define que tipos de dispositivos se van a conectar al servidor. VRPN incluye una aplicación para correr el servidor ( vrpn_server ) y para realizar la conexión de un dispositivo, basta con colocar el id del dispositivo que se definio en el archivo de configuración y la ip donde se encuentra corriendo el servidor ( e.g WiiMote1@localhost ).

[6] Ver *VRPN*, Disponible en: *http://www.cs.unc.edu/Research/vrpn/*

#### 4.2.3.2. OpenHaptics

**OpenHaptics Toolkit** es un kit de herramientas que se preocupa por la interacción *háptica.* La háptica es la ciencia que incorpora el sentido del tacto y control en aplicaciones de cómputo a través de la retroalimentación táctil o de fuerzas (cinestésica). Por medio del uso de dispositivos especiales de entreda/salida, llamados ***dispositivos hápticos,*** y con una aplicación que permita interacción háptica, los usuarios pueden sentir y manipular objetos virtuales en tercera dimensión(3D). En la actualidad, la háptica tiene muchos campos de acción y uno de los más importantes es en el campo de la medicina a través de los simuladores de cirugías y entrenamiento médico. **[14]** El toolkit de OpenHaptics esta compuesto de: *QuickHaptics Micro API*, que contiene las clases desarrolladas en lenguaje c++ para el desarrollo de aplicaciones hápticas, *Haptic Device API ( HDAPI )* y *Haptic Library API( HLAPI ).*

**Arquitectura General OpenHaptics**

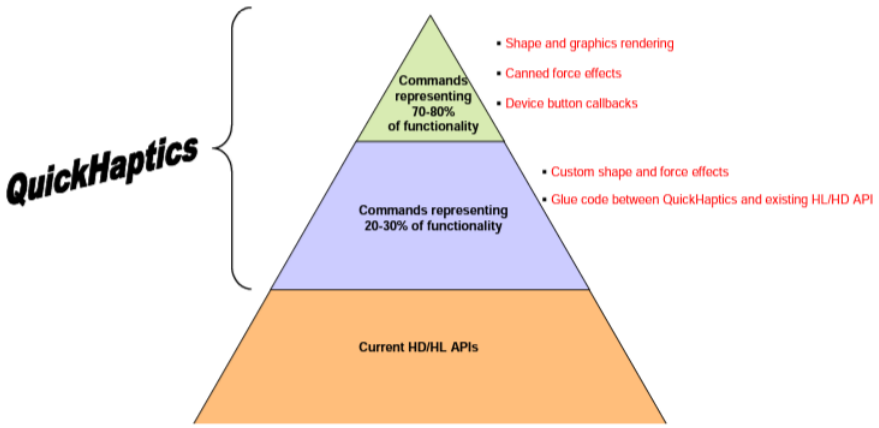


Figura 4.6: Arquitectura General de OpenHaptics 3.0

**Arquitectura QuickHaptics Micro API**

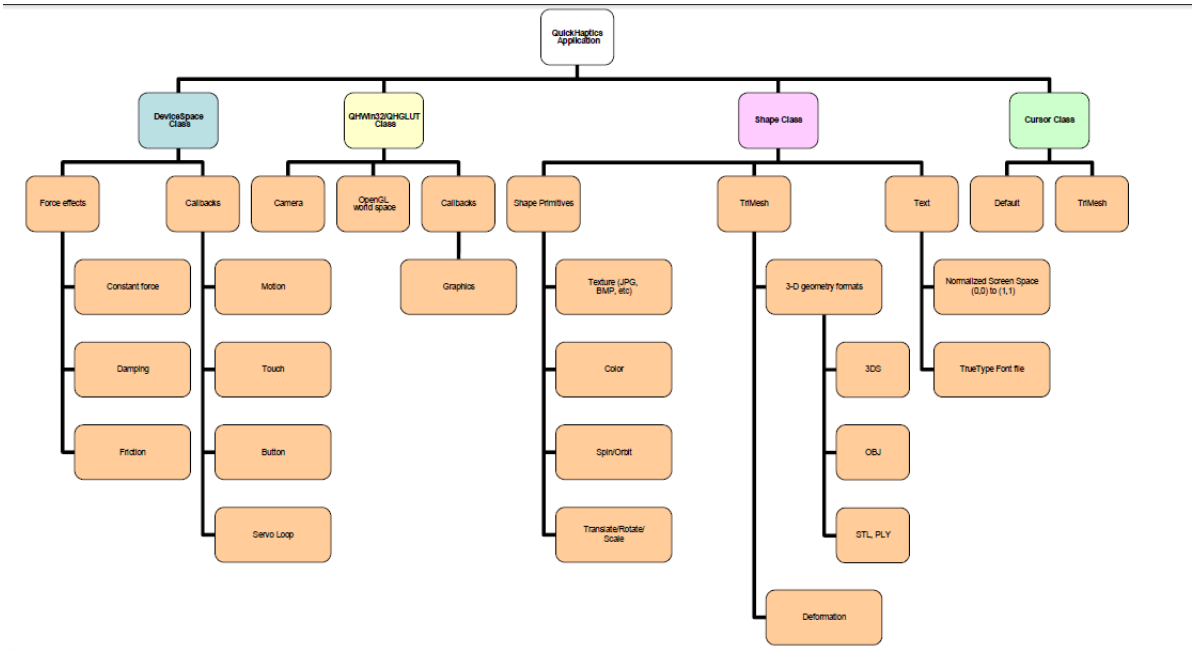


Figura 4.7: Arquitectura QuickHaptics Micro API

Este componente fue integrado en la última versión de OpenHaptics para facilitar la programación a quienes no conocen de la programación háptica y facilita la interacción entre los dispositivos y las mallas 3D.

**Componentes QuickHaptics Micro API** QuickHaptics esta dirigido por 4 clases principales que son:

- **DeviceSpace:** Representa el espacio de trabajo por el cual el dispositivo se puede mover.
- **QHRenderer:** Clase básica de QHWin32 ( Aplicaciones de Windows ) y QHGLUT ( Aplicaciones con GLUT ), genera la ventana que renderiza las formas con una camara de escena que permite al usuario sentir esas formas que se visualizan en la pantalla a través del dispositivo háptico.
- **Shape:** Clase base para una o mas figuras geométricas que se pueden renderizar, ya sea gráficamente o hápticamente.
- **Cursor:** Representación gráfica del extremo del dispositivo PHANTOM. El extremo del dispositivo algunas veces los denominan como el punto de interfaz háptica( HIP ).

**Propiedades de los componentes de QuickHaptics Micro API**

**DeviceSpace Class**

Define las propiedades de fuerza del dispositivo PHANTOM y las de interacción del usuario, las cuales son:

- **Force Effects:** Involucra la fricción, amortiguación y la fuerza constante.
- **User Callbacks:** Llamados de funciones que ocurren como resultado de un evento; movimiento, toque, presión de un botón, son algunos de los eventos del dispositivo.

**QHRenderer Class ( QHWin32 o QHGLUT )**

Define un espacio OpenGL para las transformaciones generadas con el dispositivo PHANTOM. Incluye modelos para iluminación y camaras en la escena.

**Shape Class**

Define la clase para todas las primitivas geometricas que pueden ser incluidas en la escena, entre las que se encuentran:

- Line ( Linea )
- Plane ( Plano )
- Cone ( Cono )
- Cylinder ( Cilindro )
- Sphere ( Esfera )
- Box ( Caja )
- TriMesh ( Malla )
- Text ( Texto )

Estas primitivas contienen propiedades que pueden ser modificadas para personalizar la escena; textura, color, posición, rotación, entre otras.

### Cursor Class

Define el punto de interfaz háptico. Interactua con todas las otras clases involucradas en la escena. Necesita conocer la posición del dispositivo, necesita conocer las figuras que hay en la escena para saber como interactuar con ellas y debe saber como se renderiza la aplicación para poder ser visualizado en la pantalla.

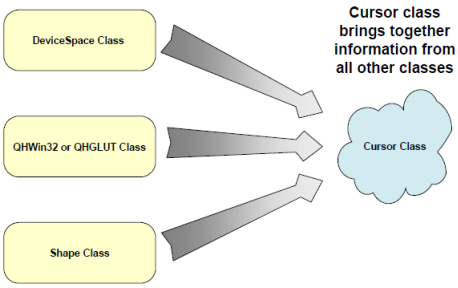


**Figura 4.8: QuickHaptics Cursor**

### Diseño de una aplicación con QuickHaptics

Lo importante en el proceso del diseño de una aplicación es que el paso de creación de la ventana, sea para una aplicaión de Windows o de GLUT, se ejecute al comienzo de la aplicación, de lo contrario generará problemas. Los demas pasos no son estrictos pero se recomienda seguir el orden propuesto.

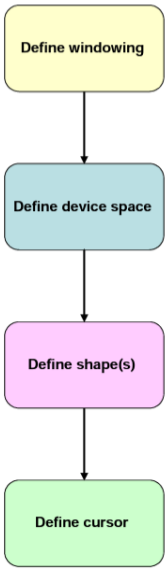


**Figura 4.9: QuickHaptics Application**

### HLAPI & HDAPI [14]

El HLAPI proveé renderización háptica de escenas de alto nivel, lo que quiere decir, que se programa más a nivel de visualización y comportamiento del dispositivo sobre la escena. Se pueden utilizar ambos para crear aplicaciones y tener control tanto de la visualización como del manejo de atributos del dispositivo.

El HDAPI proveé acceso a bajo nivel al dispositivo háptico, permite a los programadores trabajar directamente con las fuerzas y esta dirigido a desarrolladores que este familiarizados con los paradigmas de la programación háptica.

Para este proyecto se enfocó más el desarrollo con el HDAPI, ya que como la visualización ya estaba desarrollada con Ogre3D, solo se necesitaba conocer el manejo del dispositivo para a partir de eso poder modificar los componentes de la escena con las interacciones con el dispositivo Phantom Omni.

**Diseño de una aplicación con HLAPI & HDAPI**

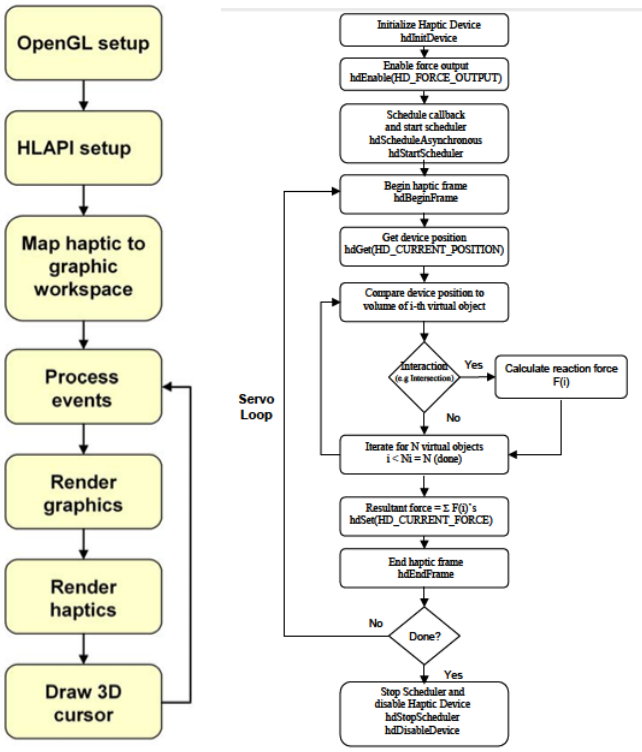


**Figura 4.10: HLAPI Application & HDAPI Application**

### 4.2.4. Dispositivos de retroalimentación háptica

#### 4.2.4.1. Phantom Omni

El Phantom Omni es un dispositivo de retroalimentación háptica diseñado por Geomagic's Sensable Group y es considerado el dispositivo de este tipo con mayor despliegue de la industria. Es utilizado en diferentes investigaciones, proyectos que involucran modelado 3D, entre otros tipos de aplicaciones. Es un dispositivo motorizado que aplica retroalimentación de fuerzas sobre la mano del usuario, permitiendo la sensación del objeto virtual sobre el cual se esta interactuando.

Los dispositivos hápticos son considerados como parte vital de muchos entrenamientos médicos. Por medio del uso del Phantom Omni, los médicos cirujanos que lo utilizan pueden planear y ensayar por medio de cirugías sobre pacientes virtuales pero adicionando la reacción de sensación frente a los tejidos sobre los cuales se realiza el procedimiento. Una de las principales características que ofrece este dispositivo son los 6 grados de libertad con los que cuenta para su manipulación. [7]

[7] Ver *Phantom Omni*, Disponible en: *http://www.geomagic.com/en/products/phantom-omni/overview*

# Parte III

# DESARROLLO DEL SIMULADOR VRLS

# Capítulo 5

# Desarrollo del Simulador de Realidad Virtual

Para el desarrollo del simulador se propuso la metodología rápida Extreme Programming( XP ) en la que el director del Trabajo de Grado, dueño del producto, era quien iba definiendo las necesidades que debieron ser desarrolladas a lo largo del proceso.

Desde el inicio del proyecto se definieron tres( 3 ) fases principales que cubrían todas los requerimientos del proyecto. Estas fueron :

- Fase I ( Desarrollo de la parte visual del simulador )
- Fase II ( Desarrollo del modelo de interacción con el Phantom Omni )
- Fase III ( Inclusión del modelo de interacción en el simulador )

Para cada una de las fases se definieron unos requerimientos, se hizo un pequeño diagrama de componentes y se realizó el prototipo que cumplía con dichos requisitos.

## 5.1. Fase I - VISUALIZACIÓN

La idea principal de esta fase era lograr visualizar los órganos que íban a estar involucrados en el procedimiento médico propuesto, para esto era necesario averiguar como se realiza la integración de Ogre3D con ITK para la generación de las mallas en 3D.

### 5.1.1. Requerimientos

Los requerimientos definidos por el dueño del producto fueron:

1. Investigar integración entre Ogre3D e ITK.
2. Algoritmo de lectura de mallas .vtk con la librería ITK.
3. Algoritmo de renderización de las mallas generadas con ITK en una escena de Ogre3D.
4. Diseño e inclusión de mallas de herramientas de operación al simulador.

#### 5.1.1.1. Investigación

Teniendo en cuenta el objetivo principal de esta fase del simulador, se inició por leer y entender la documentación de la libreria ITK y de Ogre3D, para saber como es el manejo de las mallas 3D en cada uno de ellos.

Inicialmente se investigó el manejo de mallas en Ogre. Una malla en Ogre3D esta compuesta de:

- Vértices
- Normales de los vértices
- Caras ( geometría definida a partir de los vértices )
- Color de los vértices.

Para lograr almacenar esa información y que se pueda visualizar, Ogre3D utiliza *buffers de memoria.* En Ogre3D, el manejo de buffers de memoria se da a través de la clase "Ogre::HardwareBuffer". Un buffer de hardware es cualquier área de memoria que se encuentra en la memoria ram de video, sin embargo esta clase puede ser usada para otras áreas de memoria del sistema. [1]

[1] Ver *HardwareBuffer,* Disponible en: *www.ogre3d.org/documentation*

Estos son unos ejemplos de visualización de mallas en Ogre3D.

**Figura 5.1: Mallas de Ogre3D**

#### 5.1.1.2. Modelo de Transformación [3]

Una vez hecha la investigación sobre las mallas de Ogre3D, se analizó la documentación de ITK para saber cual era el proceso para transformar una imagen en formato .vtk a una malla de Ogre3D.

El modelo de transformación es presentado a continuación:

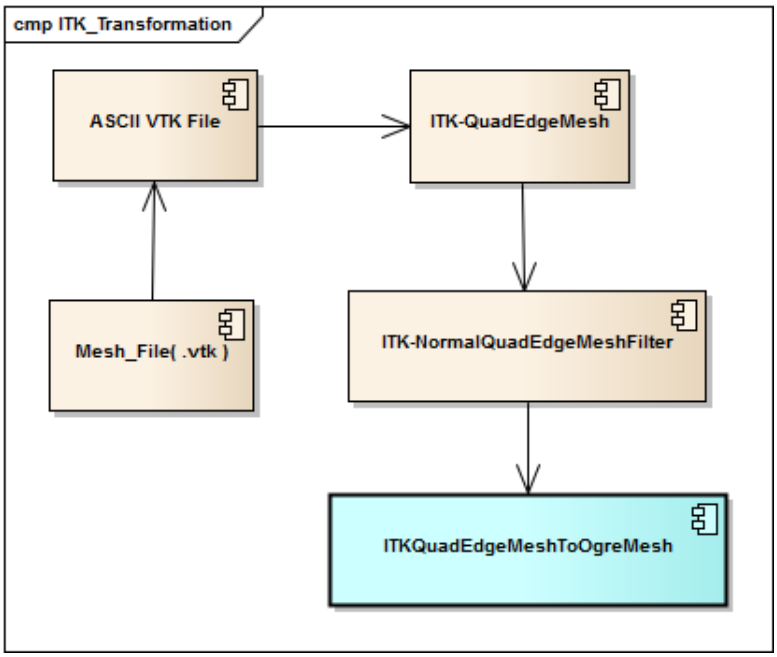


**Figura 5.2: Transformación ITK a Ogre3D.**

**Mesh File**

Componente que representa el archivo .vtk de la malla 3D del órgano que va a ser transformardo para su renderización en formato Binary.

**ASCII VTK File**

Componente que representa la transformación que se realizó con la herramienta Paraview, en la que se paso de una malla VTK Binaria a una malla VTK ASCII. Este proceso es fundamental, ya que este es el formato al que pertmite ITK aplicarle la transformación.

**ITK-QuadEdgeMesh**

Componente que representa el proceso de lectura del archivo .vtk con la libreria ITK. Para más detalles, lea la sección Manejo de lectura y escritura de mallas de ITK. El resultado de ese proceso genera un elemento de tipo QuadEdgeMesh, el cual representa la malla cargada en memoria de ITK.

**ITK-NormalQuadEdgeMeshFilter**

Componente que representa el proceso de cálculo de las nomales a las superficies y a los vectores de la malla de ITK. Este filtro esta dado por la formula:

$$n_v = \frac{\sum_{i=0}^{N_f} \omega_i \cdot n_i}{\| \sum_{k=0}^{N_f} \omega_x \cdot n_k \|}$$

Figura 5.3: Fórmula Filtro Normales ITK

La diferencia entre cada uno de los métodos esta dado por lo que ellos denominan el "*peso*" ( weight ), el cual puede ser:

- GOURAUD: $\omega_i = 1$
- THURMER: $\omega_i =$ Ángulo del triangulo en el vertice señalado
- AREA: $\omega_i = Area(t_i)$

Para las mallas que se utilizaron, se aplico peso de **Gouraud**. **[15]**

**ITKQuadEdgeMeshToOgreMesh**

Componente que representa el paso final de la transformación, en donde se toma el QuadEdgeMesh generado y se recorren sus vertices y sus normales, se definen las superficies o caras y se cargan en el buffer para generar el mesh de Ogre3D.

Ya con esto, desde el componente de Ogre que se encarga de la creación de la escena, se genera el nodo de escena, la entidad y se le asigna un material a cada uno de los órganos que se van a renderizar en el simulador.

Es importante mencionar, que para que Ogre renderice las normales positivas y negativas, hay que desabilitar el componente de CULLING y así, permitirá la visualización, tanto del interior como del exterior.

### 5.1.2. Prototipos

#### 5.1.2.1. Prototipo Inicial

El objetivo principal de la primera aproximación del prototipo consistía en lograr renderizar una malla de ITK en una ambiente básico de Ogre3D, el cual contaba simplemente con una cámara para poder visualizar la malla y una luz que mostrará la textura de la malla.

El prototipo se realizó pensando en el patron MVC, donde el modelo íba a representar todos los algoritmos que realizaran el proceso de transformación de mallas ya anteriormente descrito, la vista sería la visualización del ambiente de realidad virtual y el controlador sería el modelo de interacción que sería desarrollado en una fase posterior.

El primer resultado que se obtuvo fue la renderización del estómago, órgano que se escogió para realizar la pruebas iniciales y esto fue lo que arrojó:

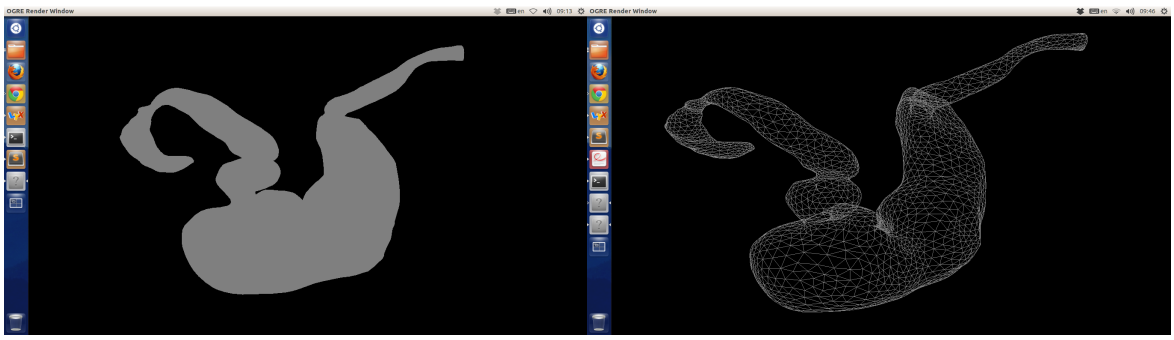

**Figura 5.4: Estómago: vista de superficie y vista de geometría**

Ya habiendo logrado visualizar uno de los órganos aplicando el modelo de transformación, se diseño un algoritmo genérico que aplicara el mismo proceso a todas la mallas de órganos que se íban a utilizar para el simulador virtual, dando como resultado la siguiente imagen:

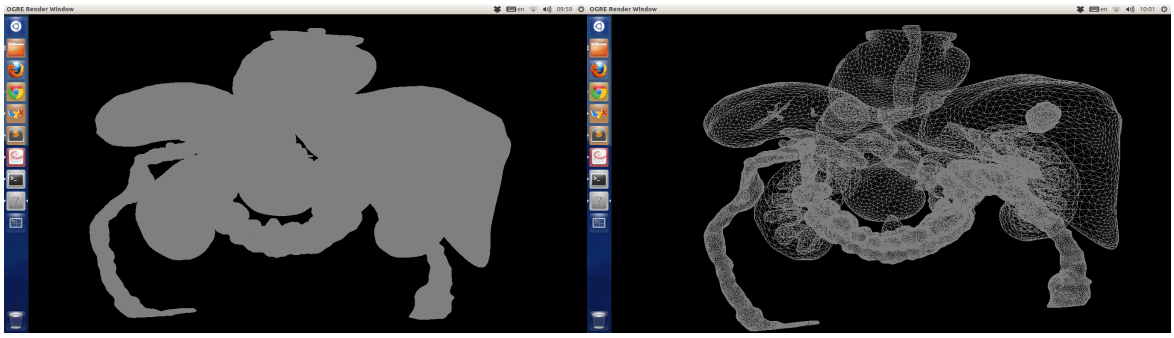

**Figura 5.5: Torso: vista de superficie y vista de geometría**

**Conclusión**

Los resultados que se habían generado en esta etapa del proyecto eran satisfactorios ya que se había logrado aplicar la transformación de manera correcta a las mallas suministradas para el proyecto y ya era posible visibilizar lo que sería el comienzo del simulador. Una de las principales ventajas que se pudo observar era que la renderización y aplicación de la transformación sobre las diferentes mallas no tomaba más de 1 minuto, lo que daba la oportunidad de presentarlo en tiempo real.

#### 5.1.2.2. Segundo Prototipo

Después de tener las mallas dentro de un ambiente de Ogre3D, el objetivo que se propuso era complementar la escena, adecuar las texturas de las mallas, incluir más luces a la escena y mejorar el simulador.

Para el manejo de las texturas en Ogre3D se definió en un archivo de extensión *.material*, en el cual se definieron los colores de cada una de las mallas que compondrían la escena.

Se incluyeron cuatro( 4 ) luces que mejorarian la visibilidad y el efecto de sombras de las mallas y se incluyó un piso y una mesa de operaciónes, que le darían un toque de realismo al simulador.

**Blender: Transformación de OBJ a MESH**

Uno de los requerimientos de Ogre3D es que las mallas que vayan a ser incluidas en la escena deben estar en formato *.mesh,* por lo que para lograr incluir mallas adicionales a la escena era necesario investigar alguna herramienta que permitiera ese proceso.

Blender[2] fue la herramienta escogida, ya que existe un plugin que al ser instalado, permite exportar cualquier malla visualizada en extensión *.mesh* de Ogre3D.

Siguiendo esta idea, se decidió buscar en repositorios de imágenes 3D y de los que se encontraron, se escogieron el repositorio de Google 3D [3] y BlendSwap.[4] De allí, fueron tomadas las mallas que ambientarían el simulador de realidad virtual.

Finalmente, en Blender se renderiza la malla en cualquiera de los formatos soportados, para este caso se utilizaron *.obj*, se adecuaron y se exportaron en *.mesh* de Ogre3D.

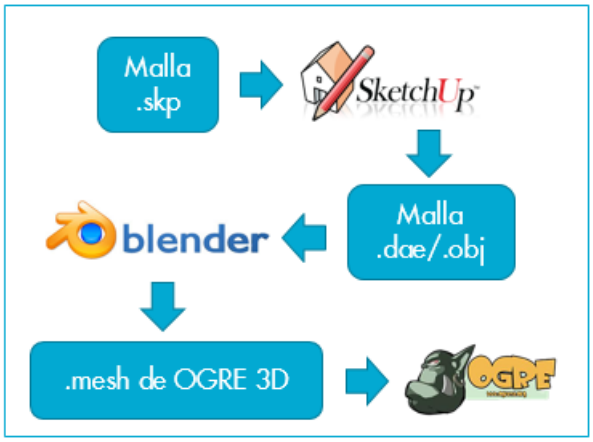


Figura 5.6: Transformación de Obj a Mesh

### Visualización

Cuando se logro aplicar la trasformación de las mallas adicionales por medio de Blender, dió como resultado una segunda versión del prototipo que ya cumplia con los primeros requerimientos definidos para esta fase del proyecto.

[2] Ver *Blender*, Disponible en: *www.blender.org*
[3] Ver *Google 3D Warehouse*, Disponible en: *http://sketchup.google.com/3dwarehouse/*
[4] Ver *Blendswap*, Disponible en: *http://www.blendswap.com*

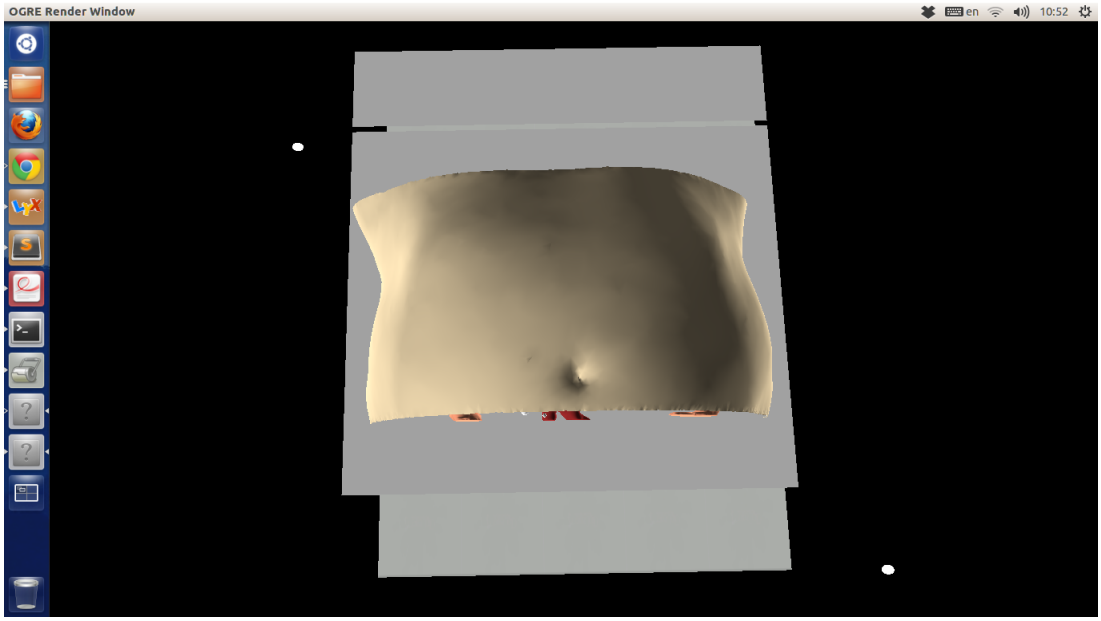


Figura 5.7: Vista aérea

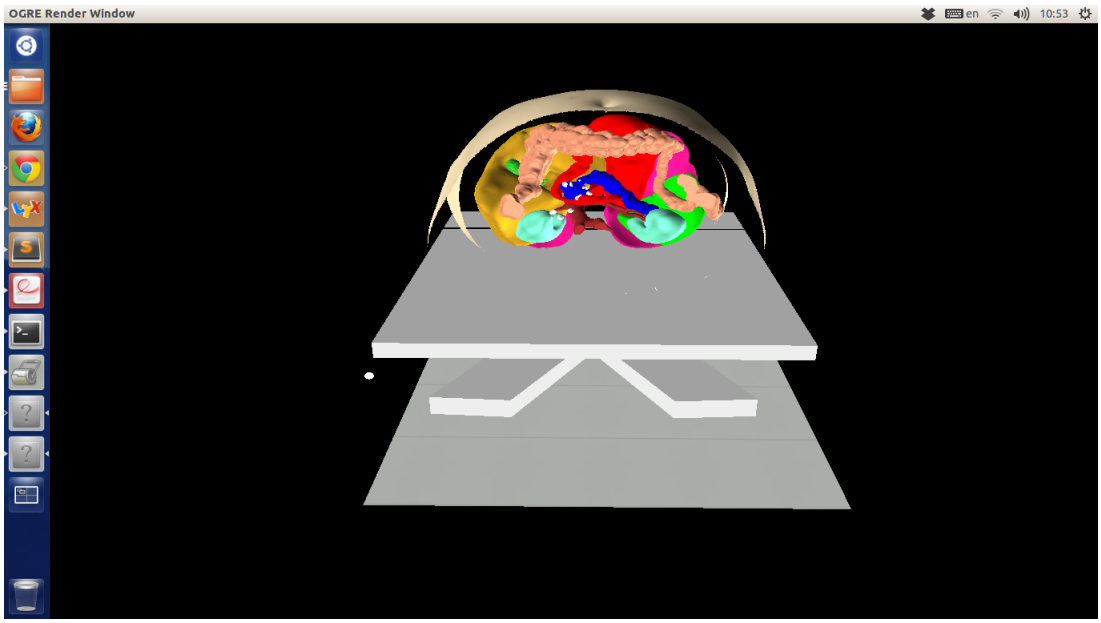


Figura 5.8: Vista lateral

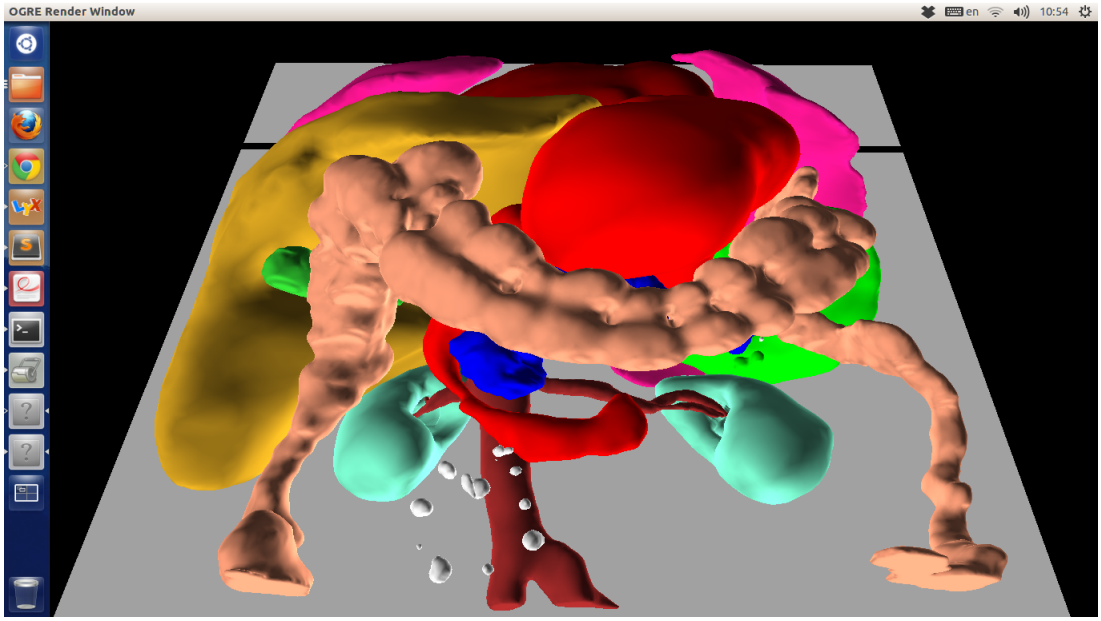


Figura 5.9: Vista interna de los órganos

**Conclusión**

Los resultados que se estaban viendo eran muy satisfactorios, ya en esta etapa el simulador se podía apreciar mucho más de lo que se esperaba lograr. El proceso de aprendisaje con la herramienta Blender fue un poco complejo, pero con la ayuda de tutoriales y videos se pudo ajustar a las necesidades del proyecto.

#### 5.1.2.3. Tercer Prototipo

En esta parte de la fase ya se habían cumplido 3 de los requerimientos planteados, faltaba la inclusión de las herramientas de operación. Para cumplir con este requisito, se utilizo el proceso aprendido el paso anterior, la diferencia fue que esas mallas no se encontraron ya listas para transformarlas e incluirlas, estas pasaron por un proceso adicional de diseño, en el que tocó aprender a editar y generar mallas 3D en Blender y así finalmente poder incluirlas en el simulador.

En esta etapa de la fase, el dueño del producto definió una nuevo requerimiento que consistia en diseñar una arquitectura para el manejo de las herramientas que al final, lo que permitiaría era incluir esas herramientas en el modelo de interacción. Adicional a esto, se incluyó el requisito de que las herramientas tuvieran un control de colisión, esto se refiere, a que la herramienta al momento de ingresar en el torso debía reconocer las mallas que habían en el interior y no atravesarlas. Para esto se implementó un algoritmo de RayCasting a nivel del

polígono. [5]

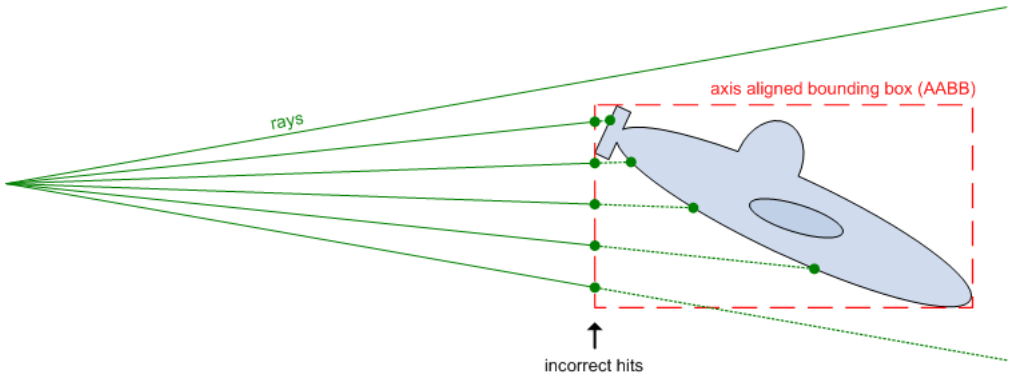


**Figura 5.10: RayCasting a nivel del polígono**

### Arquitectura Tool

El diseño de la arquitectura de las herramientas consta de una clase principal y unas subclases que heredan de ella. Una de las necesidades principales para el manejo de estas herramientas era que para poder hacer la verificación de colisiones requeria de conocer la posición actual de la punta de la herramienta, para esto simplemente se aplicó una estrategia, en la que se ubicó un nodo de escena que depende de la posición de la herramienta lo que le permite mantener su posición y asi garantizar que se podía realizar la validación. El diagrama del diseño ( Figura 6.10 ).

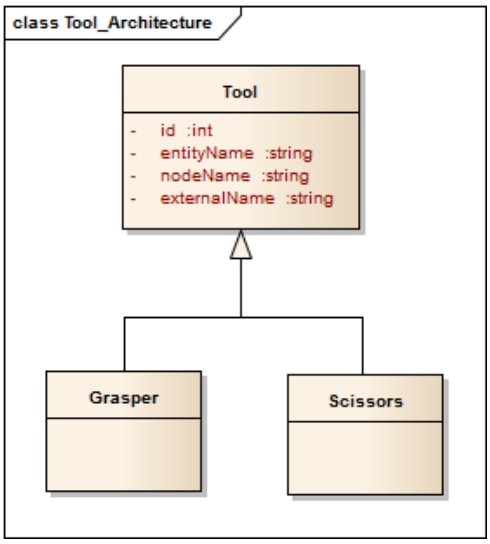


**Figura 5.11: Diagrama Arquitectura**

[5] Ver *Raycasting Polygon Level*, Disponible en: *http://www.ogre3d.org/tikiwiki/Raycasting+to+the+polygon+level*

**Tool**

Clase principal, representa una herramienta genérica la cual tiene el nombre de la entidad que se le asigno en el ambiente de Ogre3D, el nombre del nodo de escena, el nombre del material y el nombre del nodo de escena ubicado en el extremo de la herramienta ( punta ).

**Scissors**

Subclase, representa una especificación de una herramienta, particularmente las tijeras. Hereda todas las características de la clase Tool.

**Grasper**

Subclase, representa una especificación de una herramienta, particularmente las pinzas. Hereda todas las características de la clase Tool.

El diseño de esta arquitectura permite que si se necesitan incluir más herramientas en el simulador, lo único que se debe hacer es crear la clase con el nombre de la herramienta e implementar los métodos que hereda de la clase principal.

Teniendo ya la arquitectura definida, se aplicó en la escena de Ogre3D, donde se diseñó una colección de herramientas y cada vez que se incluye la malla con la herramienta nueva se invoca el método ***CreateTool*,** el cual recibe la información necesaria y crea la nueva herramienta. Finalmente para el manejo y selección de las herramientas, se les asigna un **id** basado en su posición en la colección de herramientas y se definió una variable que controla cual es la herramienta seleccionada para usarse.

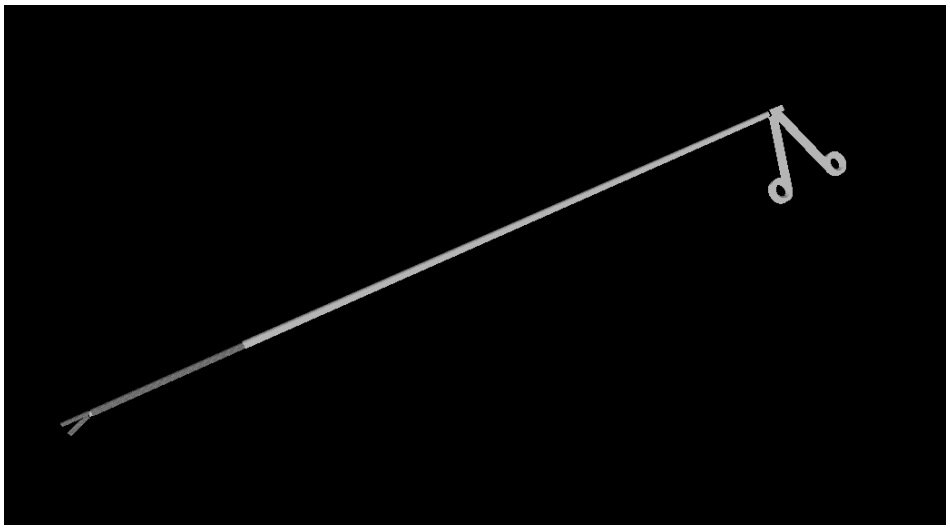

Figura 5.12: Grasper ( Pinzas )

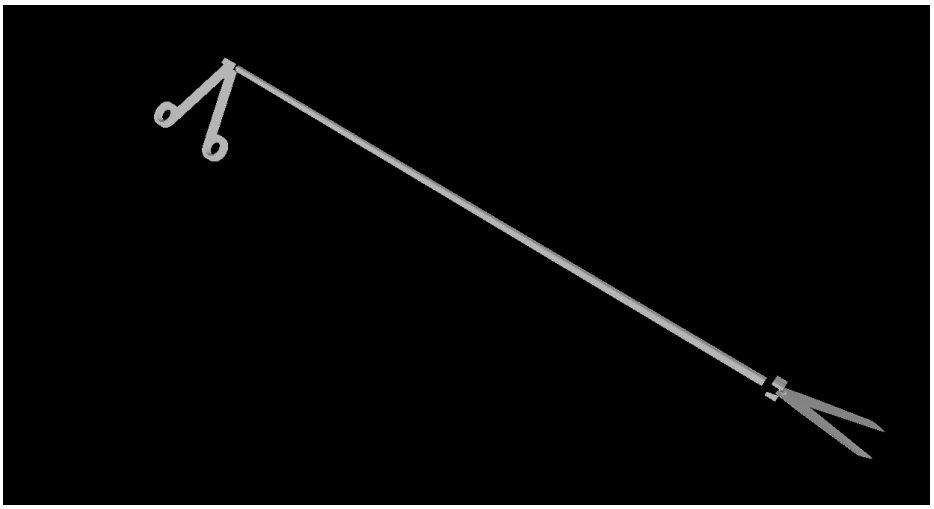

Figura 5.13: Scissors ( Tijeras )

**Conclusión**

Esta etapa del proyecto fue una de las más complicadas, ya que al comienzo no se tenía mucho conocimiento del manejo de colisiones, y cuando se empezó a implementar el algoritmo no se sabía que Ogre3D realizaba la colisión con lo que se denomina el BoundingBox de la malla, por lo que la colisión no quedaba bien. Después de una investigación un poco mayor fue que se encontro que existía la detección de colisiones a nivel del polígono y fue cuando se intento acoplar al simulador. Sin embargo, después de las dificultades, fue interesante el resultado que se logró. Las herramientas logran detectar que hay una malla en su trayectoria y no continuan avanzando.

#### 5.1.2.4. Cuarto Prototipo

En esta cuarta etapa del prototipo de la fase de visualización se definieron los dos últimos requerimientos por parte del dueño del producto: Aplicar el control de colisiones en la cámara de visualización del cateter y permitir que el usuario de la aplicación tuviera una vista panorámica del simulador, donde pudiera ver todo el entorno.

Inicialmente se trato de completar el primer requisito, se aplicó el mismo algoritmo de RayCasting a nivel del polígono pero generado desde la posición de la cámara y hacia donde esta se dirigía. Adicional a esto y en vista de que en el procedimiento real la cámara que es introducida en el abdomen para la operación va ajustada dentro de un catater( tubo ), se decidió modelar esa herramienta, que lo que permite es darle un toque de realismo al simulador.

Seguidamente se empezó a investigar como es el manejo de multiples cámaras en una escena de Ogre3D para así, poder tener las diferentes perspectivas. Lo primero que se tuvo que investigar fue el componente Viewport de Ogre3D.

**Ogre3D Viewport**

En Ogre3D, una *viewport* es un rectangulo en el que el manejador de la escena renderiza sus componentes visibles, desde la perspectiva de una simple cámara. La cámara definida para visualizar la viewport puede ser cambiada en cualquier momento de la aplicación. La viewport es el único punto de interacción entre lo que denominan el *render target*, componentes de la escena a ser renderizados, y la cámara. La viewport representa la ventana en la escena. A través del uso de multiples viewports se pueden realizar efectos como el de los espejos retrovisores, donde en una viewport adicional se podría visualizar la parte trasera del vehiculo. **[2]**

### Segunda Vista

Al conocer las propiedades de las viewport de Ogre3D se logró que el simulador tuviera dos vistas diferentes y que a través de un comando se pueda hacer el intercambio entre las dos posiciones. Esto es una funcionalidad interesante porque la vista principal es la que el médico que esta operando necesita y la segunda vista sería la que las personas externas verían.

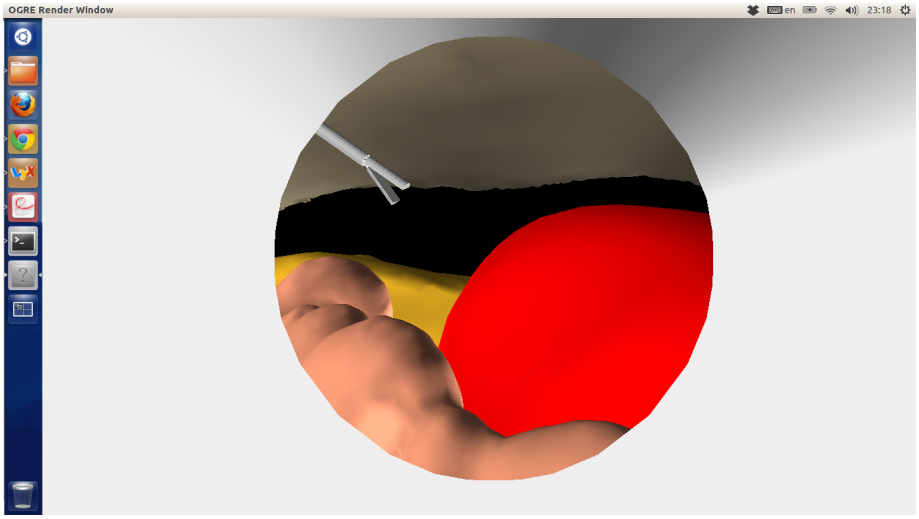


Figura 5.14: Cámara Principal

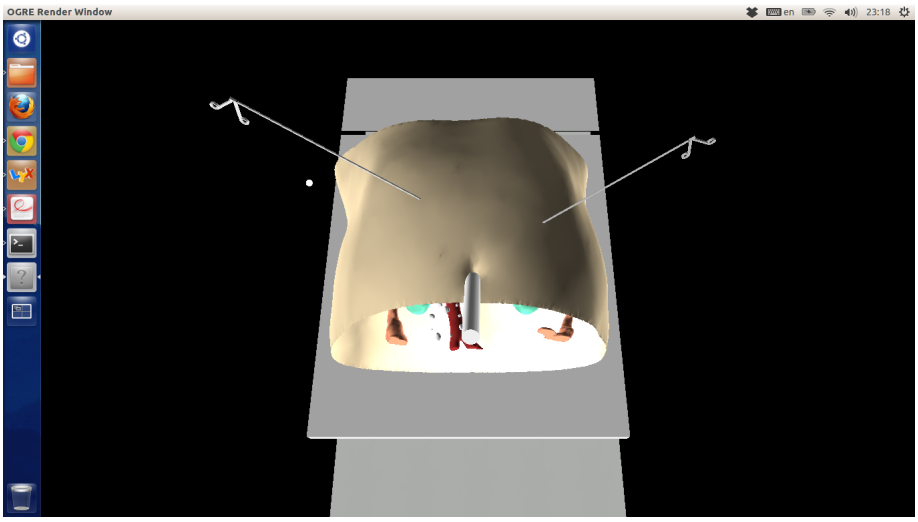


Figura 5.15: Cámara Panoramica

**Conclusión**

En esta fase los resultados eran muy positivos, se había logrado cumplir con los requerimientos exigidos y las expectativas de dueño del producto. Un detalle muy importante es que este prototipo contaba con un modelo de interacción básico manejado por medio del teclado y el ratón, pero que nos permitió tener una primera aproximación a lo que sería la siguiente fase del proyecto.

## 5.2. Fase II - INTERACCIÓN

En esta fase, el objetivo principal era lograr generar el código con el que se controlaran los eventos del Phantom Omni para modificar la escena y así poder realizar la interacción.

### 5.2.1. Investigación

Teniendo como idea principal aprender a utilizar la librería OpenHaptics, se inició por leer y entender los tres componentes que lo conforman. ( Ver 4.2.3.2 ) Para entender como funcionaban los algoritmos para el manejo de eventos se recurrió a los ejemplos de código que tráe OpenHaptics incluidos.

### 5.2.2. QuickHaptics Micro API

En primera instancia se intentaron utilizar los ejemplos del componente QuickHaptics Micro API porque después de haber leído la documentación, este es el componente de más alto nivel, el cual ya integraba clases que permitían el manejo del dispositivo y su interacción con mallas 3D. En ese primer ejercicio comenzaron a aparecer los problemas, ya que el componente tenía una serie de restricciones en cuanto al encadenamiento de las dependencias y particularmente porque requería de una variable de entorno específica que no pudo ser definida en el sistema operativo Linux.

Sin embargo, se puedo correr uno de los ejemplos y se puedo observar y manipular un cursor sobre la escena que dependia de la posición del aparato. Este ya era un primer acercamiento, pero para lograr la integración futura íba a traer unas complicaciones al proyecto, por lo tanto, se desistio rápidamente de la idea de usar este componente.

### 5.2.3. Investigación II

A partir del problema que se tuvo, el dueño del producto decidió que era importante realizar una investigación mayor sobre los demás componentes de OpenHaptics para intentar lograr el objetivo planteado.

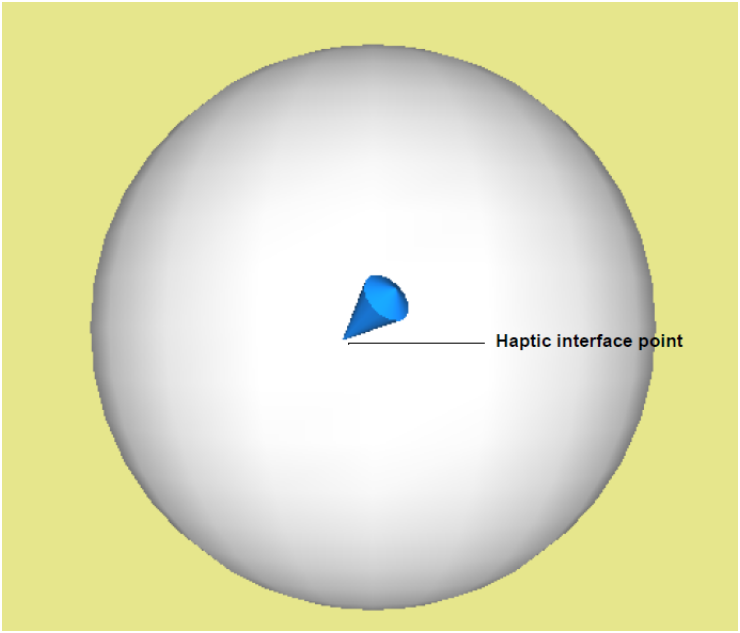


Figura 5.16: Ejemplo I QuickHaptics MIcro API

Verificando los componentes de más bajo nivel, se decidió escojer HDAPI, ya que este componente permite trabajar directamente con el dispositivo aunque aumentaba su complejidad. Esto tenía una gran ventaja, y era que no tenía ninguna dependencia con alguna librería de visualización 3D.

### 5.2.4. Haptic Device API

El primer paso que se desarrollo fue analizar las clases que pertenecen a este componente y como interactuan con el dispositivo.

Al intentar correr los ejemplos que traía consigo, fue de grata sorpresa que corrieron sin problemas, teniendo en cuenta el problema presentado con el componente anterior, y se tomo el primer ejemplo para entender el desarrollo de una aplicación háptica con esta librería. El ejemplo consistia en que se definía un centro y dependiendo de la posición del dispositivo se sentía la fuerza generada, entre más cerca al punto mucho más fuerza era generada. En el código de este ejemplo se explicaba paso a paso las funciones para que se utilizaban y concordaba con el proceso de creación de aplicaciones HDAPI presentado anteriormente. Con este ejemplo se pudo entender como se realizaba la conexión con el dispositivo, como se obtenían valores del dispositivo, como su posición o su fuerza y como se programa lo que denominan el *planificador,* quien permite el manejo de eventos del dispositivo.

A partir de esto y con el dueño del producto se concluyó que ya se podía pensar en realizar la integración del dispositivo con el prototipo del simulador, basandose en los ejemplos.


Figura 5.17: Datos de Interacción

### 5.2.5. Conclusión

La primera conclusión fue que gracias al trabajo en conjunto con el dueño del producto se pudo identificar rápidamente el problema y se solucionaron, de forma que permitió que el proyecto continuara con su desarrollo de forma natural. Algo que vale la pena resaltar es que a pesar de que el componente de más alto nivel tiene un mayor desarrollo, para nuestro caso particular no era el adecuado.

## 5.3. Fase III - INTEGRACIÓN

En esta fase final del proyecto, el objetivo principal era lograr integrar el prototipo de visualización generado en la Fase I con el dispositivo Phantom Omni por medio del componente HDAPI de la librería OpenHaptics.

### 5.3.1. Intergración HDAPI

Luego del proceso que se realizó en la fase anterior, se propuso integrar el componente HDAPI en el prototipo que ya se tenía hecho. Para esto se tenían que realizar varias cosas. Primero, era fundamental diseñar el manejo de ticks o frames para que se pudieran controlar los eventos del dispositivo en el transcurso del tiempo. Esto permitiría que a medida que se ejecutaba el simulador, paralelamente el planificador del dispositivo íba generando llamadas asíncronas sobre el dispositivo para poder obtener los datos del mismo en cada instante de tiempo.

Para lograr esto, fue necesario la creación de una clase llamada "OpenHapticsOmni", la cual tenía los servicios de conexión con el dispositivo ( connect ) , actualización para obtener el estado del dispositivo en cada momento ( update ) y todas las funcionalidades que se requirieron para interactuar con el Phatom Omni.

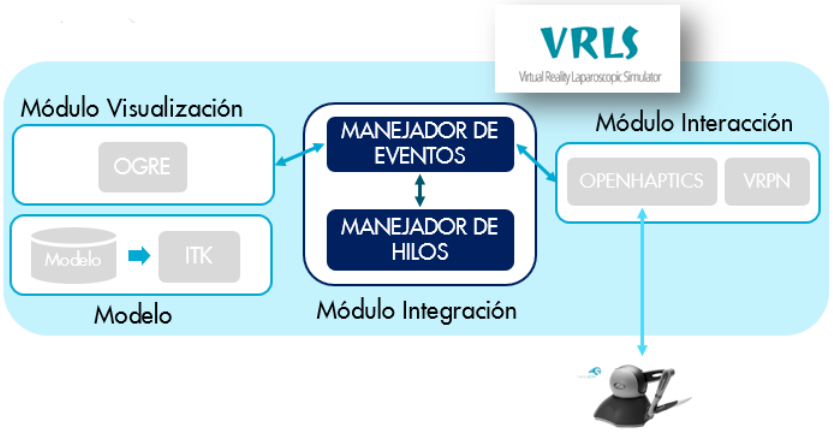


Figura 5.18: Modelo Integración

**Primer Resultado**

Después del proceso anteriormente explicado, se probo que el dispositivo fuera reconocido correctamente cuando iniciaba el simulador y mostrar un mensaje que se mostrara en cada tick del planificador del dispositivo, así se aseguraba que se podría obtener cualquier dato del dispositivo en cada instante de tiempo.

#### 5.3.1.1. Manejo de eventos del dispositivo

Ya habiendo integrado el dispositivo al simulador, se pasó a diseñar e implementar los algoritmos que íban a obtener la información del dispositivo. Estas funcionalidades se denominan *callbacks* del planificador.

Para este proyecto se definieron cuatro eventos que deberían se controlados por el componente de HDAPI, entre los que se destacan, la posición, la orientación, la fuerza y la selección por medio de los botones del dispositivo. Para lograr esto, se requeria de implementar un callback que obtuviera los datos del dispositivo requeridos.


Figura 5.19: Manejador de Eventos del Dispositivo

**HDAPI Callback**

Para definir un callback, el HDAPI proveé un prototipo el cual se muestra acontinuación:

HDCallbackCode HDCALLBACK Callback_Name( void *pUserData );

En HDAPI, los callbacks tienen diferentes estados, un callback puede terminarse o continuarse ejecutando, para estos estados se definen las constantes:

HD_CALLBACK_DONE y HD_CALLBACK_CONTINUE

El HDCallbackCode es el tipo de dato definido para manejar el estado del callback.

Para implementar un callback en el HDAPI, es importante conocer la estructura de manejo de llamadas asíncronas del planificador. Las llamadas asíncronas permiten que se obtenga retroalimentación del dispositivo y que se continue ejecutando la aplicación. Para realizar una llamada asíncrona del planificador, se utiliza la función:

HDSchedulerHandle handle = **hdScheduleAsAsynchronous**( Callback_Name, 0, Callback_Priority );

Donde:

- *HDSchedulerHandle* es quien sabe el estado del callback.
- *Callback_Name* es el nombre del callback que se quiere ejecutar.

- *Callback_Priority* es la constante que define la prioridad del callback. En HDAPI, los callbacks tienen una prioridad para ejecutarse, aunque todos los callbacks planificados se ejecutan en cada tick. Los callbacks de mayor prioridad se ejecutan antes que los de menor prioridad. Los elementos con prioridades igual se ejecutan en un orden arbitrario definido por el planificador. Las constantes para definir la prioridad son: HD_DEFAULT_SCHEDULER_PRIORITY, HD_MIN_SCHEDULER_PRIORITY y HD_MAX_SCHEDULER_PRIORITY.

Para saber cuando un callback ha finalizado, HDAPI proveé la función:

bool ended = **hdWaitForCompletion**( HDSchedulerHandle_Name,
HD_WAIT_CHECK_STATUS );

Donde:

- *HDSchedulerHandle_Name* es la variable a la que se asignó la llamada asíncrona planificada.
- *HD_WAIT_CHECK_STATUS* es una constante que conocé si el callback aún sigue planificado o ya termino.

### Callback de Estado

La posición del dispositivo en HDAPI esta definida por el componente *hduVector3Dd.* HDAPI cuenta con unas constantes que permiter interactuar con el dispositivo, en el caso de la posición, la constante es *HD_CURRENT_POSITION.*

HDAPI cuenta también con unos métodos **Get** que son, por medio de los cuales se obtienen los valores de las constantes. Entonces, dicho todo esto, la forma en que se obtiene la posición actual del dispositivo es:

hdGetDoublev( HD_CURRENT_POSITION, position );

Donde *position* es una variable de tipo hduVector3Dd, en la cual quedará almacena la posición actual del dispositivo.

La fuerza del dispositivo en HDAPI esta definida por el componente *hduVector3Dd.* La constante definida en HDAPI que permite la interacción con la fuerza del dispositivo es *HD_CURRENT_FORCE.*

HDAPI además de contar con los métodos Get para obtener información del dispositivo, tambíen cuenta con métodos **Set** para alterar el estado del dispositivo. Entonces, dicho esto, la forma de definir la fuerza que debe ejercer el dispositivo es:

hdSetDoublev( HD_CURRENT_FORCE, force );

Donde *force* es una variable de tipo hduVector3Dd que tiene el valor de la fuerza que se quiere percibir del dispositivo.

**Resultado Obtenido**

Aplicando todo lo explicado anteriormente, se probo que cuando se arranca el simulador se concectara el dispositivo y se planificara el callback de estado para que mostrara la posición y la fuerza del dispositivo, inicialmente. De manera grata, se pudó observar que en cada instante de tiempo, se mostraban ambos datos del dispositivo, lo que significaba que ya era posible utilizar esos datos para modificar las herramientas del simulador.

# Capítulo 6

# Validación del Simulador

Para esta fase del proyecto, se propuso realizar una validación con un experto en el área de cirugía laparoscópica digestiva.

## 6.1. Proceso de Validación

Lo primero que se hizo para lograr realizar la validación fue empezar a buscar un experto en la materia que puediera sacar un poco de su tiempo para utilizar el simulador y así dar su valoración en cuanto a la Utiliadad del Producto y la Facilidad de Uso del mismo.

En este proceso de validación fue complicado que alguno de los expertos consultados puediera darnos una estricta validación, por lo que se recurrió a simplemente pedir su opinion frente a la herramienta, que diera sugerencias para el producto y que comentara si era posible implementar una herramienta como estas en el ambiente de los residentes del Hospital Universitario San Ignacio ( HUSI ).

## 6.2. Comentarios de los expertos

### 6.2.1. Médico Cirujano

El primer experto que validó el proyecto fue el Dr. Guillermo García Quiñones, médico cirujano. Él comento de su experiencia en laparoscopias, había asistido a varias de ellas y examino el simulador. Para él, fue muy interesante la idea desarrollada, nos comento que desde su experiencia, el veía que la medicina está en la busqueda de lograr que los procedimientos que requieran de acceder al cuerpo de un paciente, puedan lograrse por medio de los huecos naturales del cuerpo humano, que no se requiera de ninguna incisión, por lo que mencionó, que en estos momentos la laparoscopia es el procedimiento que más se cercano esta a ese futuro. Otra de las ventajas que menciono y que por las cuales le pareció interesante ayudar este proceso de aprendizaje, fue que desde su experiencia, que son más de 30 años, las complicaciones más recurrentes de los pacientes intervenidos se presentan durante el periodo post-operatorio. Entonces, como la laparoscopia disminuye

ese tiempo, al causar un trauma menor sobre los tejidos del cuerpo, se disminuirían las complicaciónes durante ese tiempo. También comento de un caso particular que vivió en una de las laparoscopias a las que asistió, en la que se cometió un error, se perforó el diafragma del paciente y eso le trajo severas complicaciones, por lo que citó que era una herramienta con posible aplicación real.

### 6.2.2. Residente HUSI

El segundo experto que validó el proyecto fue el Dr. Oscar David Rubio, residente de tercer año de cirugía general del HUSI. Su validación fue muy valiosa, ya que él, en la actualidad, se encuentra en el proceso de aprendizaje y nos comentó, que en el hospital se hacen algunos simuladores muy básicos con "palitos, aritos y una cámara", como nos él mismo se refirió. Nos contó de simuladores que existen en el mercado en la actualidad, pero que su alto costo los hace inasequibles por parte del HUSI, por lo que dijo que él creía que el simulador podría ser de ayuda en el coxtexto del hospital. Una de las cosas más valiosas de su validación fueron las sugerencias que hizó al simulador. Comentó que sería importante llevar el proyecto a una fase adicional donde ya se pudiera hacer la deformación de mallas para poder llevar a cabo el procedimiento completo. Otra de las cosas que mencionó es la importancia de los colores utilizados en los órganos presentes en el simulador. Habló de la importancia de la sensación de la resistencia que ofrece el dispositivo y comentó de algunas mejoras sobre los órganos que están en el simulador. Finalmente dijo que le parecía muy interesante el proyecto.

### 6.2.3. Jefe de Residentes HUSI

El último experto que validó el proyecto fue el Dr. Felipe Alvarado, jefe de residentes del HUSI. Después de haber evaluado el simulador, el doctor mostró gran interés en el proyecto, tanto que propuso vincularlo a un proyecto de investigación que él vá a liderar, que trata el tema de estrategías para el entrenamiento de cirugía laparoscópica.

Comentó que el trabajo realizado le parecía muy bien logrado, que le gustaría mucho que el proyecto tuviera una fase posterior en la que se pudiera incluir funcionalidades para la alteración de las mallas al momento de ejecutar el simulador. Le llamó mucho la atención la idea de que a través de un dispositivo se pudiera sentir la fuerza al momento de hacer contacto con los objetos.

También ofreció material médico para mejorar el simulador y propuso invitar al desarrollador del proyecto a un procedimiento laparoscópico real para que conociera de primera mano el proceso.

Una de las ideas más importantes que propuso, fue la de hacer una prueba en la que sometieran a los residentes de primer año a entrenamiento con el simulador y comparar-

los con los de tercer año que no tuvieron contacto con el simulador para poder medir el efecto del proyecto en la curva de aprendizaje de los residentes.

### 6.2.4. Análisis de la Validación

A partir de estas intervenciones, se pueden rescatar muchos detalles que sirven para la formulación de trabajos futuros. Es interesante ver la aceptación por parte de los médicos a este tipo de herramientas para el proceso de aprendizaje y de conocer la forma en que tratan de atacer la problematica planteada en la actualidad.

Algó que vale la pena rescatar es que en el ámbito real se concibe la idea de usar un simulador para entrenamiento, sea de cualquiera de los tres tipos mencionados en este documento y que da una idea de la importancia que puede llegar a tener una herramienta como VRLS.

A partir de las propuestas y de la reacción de los especialistas que validaron el simulador, genera una gran espectativa los resultados que se pueden obtener a partir de esta herramienta tecnológica de apoyo y de realmente creer que su aplicación en un ambiente real es factible para el contexto actual del grupo de investigación.

## 6.3. Requerimientos de Validación

Para el proceso de validación es importante tener en cuenta ciertas características que determinan la usabilidad y utilidad del producto. Es importante que esta validación no solo sea de forma cualitativa sino que debe ser de forma cuantitativa. Las características que se requieren para realizar una validación formal son las siguientes:

- Tiempo de respuesta del simulador frente a un estímulo realizado por el usuario a través del dispositivo.
- Magnitud de la fuerza que se genera al momento de realizar una colisión en el simulador.
- Velocidad del movimiento de las herramientas con respecto al procedimiento real.

Todo esto requiere de que el experto, utilice la herramienta y analice cada uno de los componentes anteriores para poder definir la usabilidad de la aplicación.

# Parte IV

# VRLS: VERSIÓN FINAL

# Capítulo 7

# Versión Final del Prototipo

La versión final de VRLS es un prototipo funcional que símula una laparoscopía digestiva y permite el manejo de las herramientas por medio de un dispositivo de retoralimentación háptica que esta compuesto principalmente por 3 módulos:

- Módulo de Visualización
- Modelo ( Procesamiento de Datos )
- Módulo de Interacción

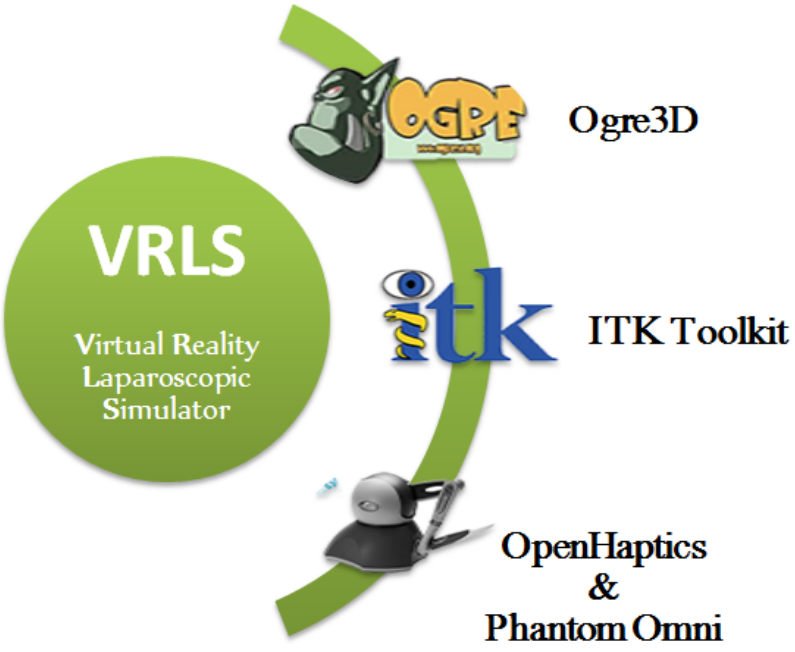


Figura 7.1: Diagrama de componentes del simulador

## 7.1. Modelo

El modelo esta compuesto de las mallas que estan presentes en el simulador y el componente ITK, la herramienta responsable de la generación de los órganos que requería este tipo de simulador. Los órganos pasaron por un modelo de transformación para poder ser visualizados ( Ver 5.1.1.2 ) y manipulados en el simulador.

## 7.2. Módulo de Visualización

El módulo de visualización compone toda la interfaz gráfica del simulador. Este módulo fue dasarrollado con Ogre3D, por lo que permite la creación de una escena en 3D, la cual cuenta con una cámara de visualización, un manejador de eventos a través de la librería OIS y dos vistas diferentes sobre la escena.

Se le incluyeron objetos adicionales a la escena para darle un aspecto más real al simulador y se diseñaron e integraron las herramientas quirúrgicas requeridas en este procedimiento y la herramienta que hizo esto posible fue Blender.

Se desarrollo un modelo de control de colisiones en el simulador que mejoraría la interacción del las herramientas con el ambiente y se crearon dos diferentes perspectivas del simulador, una para el médico que opera, y otra para ver lo que esta sucediendo en la escena.

## 7.3. Módulo de Interacción

El módulo de interacción compone todos los mecanismos que permiten la comunicación entre el dispositivo Phantom Omni y el simulador de realidad virtual. Involucra el uso de la librería OpenHaptics, que es la que proveé todas las funcionalidades para obtener los datos del dispositivo ( Ver 5.3.1.1 ).

Lo más importante del módulo de interacción es como a través del componente HDAPI se obtiene la información del dispositivo y se utiliza para modificar la escena del módulo de visualización.

Adicional a esto, tiene la característica que el usuario puede percibir cuando se presenta una colisión entre la herramienta que se esta manipulando y los órganos presentes en la escena.

# Capítulo 8

# Resultados del Proyecto

## 8.1. Cumplimiento de los Objetivos del Proyecto

### 8.1.1. Objetivo General

El objetivo general que se planteó para el proyecto fue el siguiente:

*"Desarrollar una prototipo funcional basado en un ambiente de realidad virtual para simular un procedimiento de una laparoscopia digestiva que sirva como apoyo al proceso de aprendizaje y pueda aportar para mejorar las habilidades de los estudiantes de medicina y/o residentes."*

El objetivo se cumplió de forma satisfactoria, ya que el producto final es un prototipo funcional que símula una laparoscopia digestiva y que permite la interacción del médico que va a ejecutar la simulación con el entorno que se presenta en el simulador.

Un detalle que es importante remarcar es que la validación que se alcanzó a obtener no pudo ser medible. Por el tiempo que se dispuso y por los problemas de disponibilidad de los especialistas a los que se recurrió, fue un poco complicado lograr medir la facilidad de uso percibida por el especialista y la utilidad percibida. Estas características se preguntaron y se obtuvieron comentarios, opiniones y sugerencias para una posible fase adicional de este proyecto.

### 8.1.2. Objetivos Especificos

La siguiente tabla muestra los objetivos específicos planteados en el proyecto y el resultado de cada uno:

| ID | Objetivo | Estado | Observaciones |
|---|---|---|---|
| 1.1 | Investigación del procedimiento de laparoscopias digestiva. | ***Cumplido*** | Se ejecuto para la creación del Marco Teórico. |
| 1.2 | Estudio del toolkit OpenHaptics. | ***Cumplido*** | Se ejecuto en la fase de Integración del desarrollo. |
| 1.3 | Investigación librerías para generación de ambientes de realidad virtual. | ***Cumplido*** | Se escogió Ogre3D para el desarrollo. |
| 1.4 | Investigación de algoritmos para generación de ambientes de realidad virtual. | ***Cumplido*** | Se estudió la documentación de Ogre3D y de ITK. |
| 1.5 | Estudio del dispositivo Phantom Omni. | ***Cumplido*** | Se llevo a cabo en la fase de Integración del desarrollo. |
| 2.1 | Diseño del modelo del simulador virtual. | ***Cumplido*** | Se llevo a cabo en la fase de visualización del desarrollo. |
| 2.2 | Diseño del modelo de interacción. | ***Cumplido*** | Se llevo a cabo en la fase de Integración del desarrollo. |
| 2.3 | Desarrollo del ambiente de realidad virtual. | ***Cumplido*** | Se desarrollaron cuatro prototipos para llegar a la versión final. |
| 2.4 | Desarrollo del modelo de Interacción. | ***Cumplido*** | Se llevo a cabo entre la Fase de Interacción e Integración. |
| 3.1 | Validar el simulador en su etapa final con el experto. | ***Cumplido*** | Se obtuvo validación empírica, no se tomaron datos medibles. |

Cuadro 8.1: Tabla Análisis Objetivos Específicos

## 8.2. Aporte del Proyecto

Es importante recalcar que en la universidad no se tenía un proyecto de este tipo. En el grupo de investigación, a nivel de pregrado, se venían planteando proyectos muy interesantes de investigación, en cuanto a visualización de imágenes médicas para diagnósticos de diferentes patologías. Este proyecto quizó mostrar que se podía proponer ir más alla y desarrollar una herramienta para apoyar al campo de la médicina de forma más practica.

Este proyecto incursiono en el tema de la programación háptica, aprovechando la inversión realizada por la universidad, en el Phantom Omni. Pocas universidades colombianas cuentan con la fortuna de tener un dispostivo de esta tecnología, y por su complejidad, no son muchos los proyectos que se encuentran en los que se há incluido.

De los proyectos que se encontraron, hay tres proyectos muy interesantes. El primero fue una apliación que permitía la interacción con los objetos del Museo del Oro por medio de este dispositivo **[16]**.

El segundo fue un trabajo de maestría de la Universidad Nacional, en el cual se compararon diferentes librerías de programación háptica y se aplicaron sobre un modelo de 3D de medicina **[17]**.

El último era un sistema de entrenamiento médico para una artroscopía de rodilla, en el cual generaban un modelo de realidad virtual y realizaban la interacción con un Phantom Omni. **[18].**

## 8.3. Impacto Potencial del Simulador

Inicialmente como herramienta de investigación, el proyecto busca abrir nuevas posibilidades para aprovechar las ventajas que ofrecen los motores de juegos con aplicaciones que buscan ayudar en el campo de la medicina.

Dada la validación del médico especialista, el simulador puede ser incluido en un proyecto de investigación con el HUSI para realizar estudios con los residentes de dicha institución y medir el impacto que la herramienta puede generar en el conocimiento de los médicos que practican este tipo de cirugías.

# Capítulo 9

# Conclusiones, Recomendaciones y Trabajos Futuros

## 9.1. Conclusiones

### 9.1.1. Respuesta a la Pregunta Generadora

Un simulador de entrenamiento puede contribuir con sesiones de entrenamiento, con la creación de situaciones que se podrían presentar en un ambiente real para minimizar la cantidad de errores cometidos y las consecuencias de los mismos. Con VRLS se puede visualizar e interactuar por medio de herramientas quirúrgicas con los órganos del paciente.

### 9.1.2. Cumplimiento de los Objetivos

Para un proyecto de la complejidad que un simulador médico representa fue un reto y un logro haber cumplido con todos los objetivos propuestos.

### 9.1.3. Aporte a la solución de la problemática

Con una herramienta como VRLS se puede contribuir al aprendizaje de los estudiantes de medicina en el campo de la cirugía laparoscópica. Un simulador virtual vá a ayudar a mejorar las habilidades y a conocer de antemano con lo que se va a tener que enfrentar cuando deba hacerlo sobre un paciente real.

### 9.1.4. Análisis Final

Cuando se propuso el proyecto se consideró de una complejidad muy alta para su desarrollo teniendo en cuenta la cantidad de participantes en el mismo y el tiempo con el que se contaba para terminarlo, sin embargo después de haber hecho el ejercició y de ver la herramienta que se construyó es muy interesante ver lo que se puede lograr en el marco de la universidad con un grupo de investigación.

## 9.2. Recomendaciones

### 9.2.1. Para la Carrera

Definitivamente, seguir apoyando los grupos de investigación. Incentivar a los estudiantes desde que inician la carrera a interesarse por los semilleros y los proyectos de investigación que se realizan. Esa es la forma en que la carrera puede mejorar y contribuir con grandes proyectos.

Sería interesante ver trabajos de investigación que involucren estudiantes de otras disciplinas, así podrían enriquecerse de conocimientos diferentes a los que competen directamente con la carrera.

### 9.2.2. Para la Universidad

Apostarle a la investigación. Desafortunadamente nuestro país no apoya mucho la investigación como si lo hacen en otros lados como Brasil, sin embargo es importante que la misma universidad y los diferentes grupos de investigación de cada uno de los despartamentos incentiven a los estudiantes a formar parte de ellos y de trabajar en proyectos que sirvan para mejorar lo que ya existe, o para proponer cosas nuevas que se puedan llevar a un ambiente real.

## 9.3. Trabajos Futuros

### 9.3.1. Incluir nuevas funcionalidades

Ya teniendo una version inicial de VRLS, sería importante que se incluyera la parte de deformación de mallas, lo cual, lo permite ITK y lograría que el proceso de laparoscopia puediera realizarse de manera más completa. Se podría pensar en permitir definir algunas condiciones iniciales al usuario, con el fin de recrear situaciones especificas en el simulador para ver como el estudiante reaccionaría ante las mismas.

### 9.3.2. Mejorar los algoritmos

Sería muy bueno mejorar los algoritmos del colisión para que se asemeje más a la realidad. También se podría mejorar la integración con OpenHaptics para que la retoralimentación de las fuerzas y el movimiento a través del dispositivo sea mejor.

Mejorar la asignación de texturas de los órganos presentes en el simulador para que sean más parecidos a la realidad.

### 9.3.3. Incluir herramientas

Se podría pensar en incluir más herramientas de las que se requieren en un procedimiento de este tipo, se podrían incluir animaciones sobre las herramientas para que el procedimiento cada vez sea lo más parecido a la realidad.

### 9.3.4. Un nuevo simulador

Sería interesante pensar en crear un simulador para otro tipo de patologías, como lo pueden ser las fobias o que no necesariamente fuera en el campo de la medicina.

### 9.3.5. Incluir nuevos dispositivos

Con el aporte que se hizo de la integración con VRPN adicional de OpenHaptics, al proyecto se le podrían incluir varios dispositivos, como cascos de realidad virtual, incluir vista estereoscopica, guantes de sensores, para el manejo y reconocimiento de gestos, entre otros. Esto serviriría para complementar funcionalidades requeridas en una laparoscopia real.

# Parte V

# REFERENCIAS Y BIBLIOGRAFÍA

# Referencias


[1] M. Ross Zelada, "Apendicectomía laparoscópica," *Cirugest*, 2007.

[2] G. Junker, *Pro OGRE 3D Programming*. Apress, Sept. 2006.

[3] N. A. Mejía Molina, *MIsT (Medical Interactive visualization Tool)*. Undergrad, Pontificia Universidad Javeriana, Bogotá, Colombia, 2012.

[4] C. Basdogan, M. Sedef, M. Harders, and S. Wesarg, "VR-Based simulators for training in minimally invasive surgery," *IEEE Computer Graphics and Applications*, vol. 27, pp. 54–66, Mar. 2007.

[5] S. Haque and S. Srinivasan, "A meta-analysis of the training effectiveness of virtual reality surgical simulators," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 51 –58, Jan. 2006.

[6] B. Hernández, J. Jiménez, and J. M. M, "Utilidad percibida y facilidad de uso en el comportamiento tecnológico organizacional,"

[7] J. García Murillo, M. Arias Correa, and E. Valencia Díaz, "Diseño de prototipo de simulador para entrenamiento en cirugía laparoscópica," *Revista Ingeniería Biomédica*, vol. 5, pp. 13–19, June 2011.

[8] F. Anderson, *Objective Surgical Skill Evaluation*. MSc thesis, University of Alberta, Edmonton, Alberta, 2010.

[9] I. Castillón Vela, R. Méndez, R. Méndez, and R. Vela Navarrete, "Trasplante renal con donante vivo: una revisión desde la experiencia," *Actas Urológicas Españolas*, vol. 25, pp. 150–151, 2001.

[10] M. Cevallos and J. C. Montoya, "Apendicectomia laparoscopica: Experiencia en 30 casos," *Revista de Gastroenterología del Perú*, vol. Volumen 17, no. $N^{\underline{o}}3$, 1997.

[11] G.-F. Begin, "Apendicectomia laparoscopica," *EMC - Técnicas Quirúrgicas - Aparato Digestivo*, vol. 22, pp. 1–9, Jan. 2006.

[12] F. Ramos-Salgado, J. Quintero-Becerra, and N. Hernández-Toriz, "Modelo para entrenamiento de cirugía laparoscópica urológica," *Revista Mexicana de Urología*, no. 1, pp. 31–35, 2010.

[13] *The VTK User's Guide*. Kitware Inc, $11^{th}$ ed., 2010.

[14] Sensable Technologies Inc., *OpenHaptics Toolkit - Programmer's Guide*. 3.0, 1999.

[15] H. Gouraud, "Continuous shading of curved surfaces," *IEEE Transactions on Computers*, vol. C-20, no. 6, pp. 623–629, 1971.

[16] P. Figueroa, M. Coral, P. Boulanger, J. Borda, E. Londoño, F. Vega, F. Prieto, and D. Restrepo, "Multi-modal exploration of small artifacts: an exhibition at the gold museum in bogota," in *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, (New York, NY, USA), pp. 67–74, ACM, 2009.

[17] M. L. Pinto Salamanca, *Análisis e implementación de una interfaz háptica en entornos virtuales*. Maestría, Universidad Nacional de Colombia, Bogotá, Colombia, 2009.

[18] I. O. Herrera, D. R. Espitia, D. G. Zayed, and I. P. Figueroa, "Desarrollo de un sistema de entrenamiento médico para artroscopia de rodilla," *Revista Colombiana de Ortopedia y Traumatología*, vol. 22, no. 3, pp. 172–177, 2008.

# Parte VI

# ANEXOS

# Capítulo 10

# Glosario

**ITK:** Plataforma de código abierto que proveé herramientas para el manejo de imágenes.

**Ogre3D:** Plataforma de código abierto que permite la creación de escenas 3D.

**Renderización:** Generación de imágenes a partir de un modelo.[1]

**OpenHaptics:** Plataforma que proveé herramientas para el desarrollo de aplicaciones hápticas.

**HLAPI:** Componente de OpenHaptics de más bajo nivel que permite trabajar con dispositivos de retroalimentación háptica.

**HDAPI:** Componente de OpenHaptics de segundo nivel, que permite integrar dispositivos de retroalimentación háptica con OpenGL.

**Raycasting:** Técnica de programación que transforma una forma con limites en una proyección 3D por medio de rayos desde un punto a la superficie.[2]

**Viewport:** Plano que renderiza los objetos visibles en una escena 3D.

**HUSI:** Hospital Universitario Sán Ignacio

[1] Ver: *http://www.alegsa.com.ar/Dic/renderizacion.php*
[2] Ver: *http://www.permadi.com/tutorial/raycast/rayc1.html*

# Capítulo 11

# Análisis Post-Mortem del Proyecto

## 11.1. Metodología Aplicada Vs. Metodología Propuesta

La metedología aplicada fue la misma que se pensó cuando se propuso el proyecto. Se adecuó perfectamente, ya que a medida que el dueño del producto íba verificando el estado del prototipo, se íban realizando las correcciones pertinentes para lograr con el objetivo final. Lo único que tal vez no se aplicó al pie de la letra de como se planificó fue el orden en que se plantearon las actividades de las diferentes fases.

## 11.2. Actividades Propuestas Vs. Actividades Realizadas

Las actividades propuestas se cumplieron a cabalidad, la única variación que se presentó en estas, fue el momento en que se ejecutaron, ya que debido a como se organizaron las fases de desarrollo ( Ver 5 ), se llevaron a cabo en un orden diferente para lograr cumplir con los objetivos especificos estipulados y con los requerimientos planteados por el dueño del producto en cada una de las fases de desarrollo del simulador.

## 11.3. Efectidad en la estimación de tiempos del proyecto

Para el desarrollo del proyecto se intento seguir con el cronograma que se había diseñado en la propuesta más sin embargo hubo actividades que en la practica se realizaron pero no el momento que se habían pensado.

Una ventaja fue que la fase de investigación tomó menos de lo propuesto, ya que se hizo uso de un curso de la universidad, especificamente de la maestría de la carrera, que coloboró para aprender de varias de las herramientas que se aplicaron en el proyecto. Este tiempo que no se utilizó en la fase de investigacón, se utilizó en la fase de desarrollo del prototipo.

Con el director del proyeto, el cuál llamamos dueño del producto a largo del documento, se tuvieron programadas reuniones todos los martes, las cuales se usaron para revisar los

requerimientos definidos en cada fase y para corregir o generar nuevos requerimientos si este los requería.

## 11.4. Efectividad en la estimación y mitigación de los riesgos del proyecto

Afortunadamente, durante el desarrollo del proyecto fueron pocos los riesgos definidos en la propuesta que realmente afectaron el proceso. Los riesgos que se presentaron durante el proyecto fueron los siguientes:

| ID | Riesgo | Impacto |
|---|---|---|
| 3 | Alta carga académica por parte del estudiante del Trabajo de Grado. | 2 |
| 9 | Problema de tiempo ó falta de colaboración por parte del especialista que validará el simulador. | 3 |
| 12 | Enfermedad del estudiante y/o del Director del Trabajo de Grado. | 3 |

Cuadro 11.1: Tabla de riegos del proyecto

A estos riesgos que se presentaron, se aplicó la estrategia pensada para efectos mayores y puedieron ser controlados de manera existosa.

En el caso de la carga academica, debido a una situación particular del estudiante del Trabajo de Grado fue imposible disminuir el número de materias cursadas, pero el tema se pudo manejar de manera correcta siendo un poco más riguroso con los avances presentados al director.

En cuanto a la enfermedad, no fue mayor cosa, cuando se presento la situación, el director no tuvo inconveniente con que el estudiante se fuera a su casa a recuperarse y luego repusiera el trabajo atrasado.

Finalmente con los especialista se pudieron concretar citas cortas para que pudieran conocer, probar y dar su opinión frente al simulador. Uno de las consecuencias de este problema fue que no se pudieron tomar medidas de esa validación, se tomo como validación empirica pero su aporte fue muy valioso para conocer el futuro de este proyecto.