



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



ESCOM

Trabajo Terminal

**“ENTRENAMIENTO VIRTUAL DE INTERVENCIONES
QUIRÚRGICAS, UTILIZANDO SENSORES KINECT, CON
PERCEPCIÓN DE SENSACIONES FÍSICAS”**

2012-A011

Que para cumplir con la opción de titulación curricular en la carrera de:

“Ingeniería en Sistemas Computacionales”

Presentan

**Villada Quezada Marco Antonio
Damián Sandoval Sergio
Sánchez García Gabriel
López Flores Iván
Carreto Terán José Francisco**

Directores

M. en C. Dorantes Gonzáles Marco Antonio

M. en C. Cordero López Martha Rosa

14 de mayo de 2013



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO



SUBDIRECCIÓN ACADÉMICA

No. de Registro: 2012-A011

Serie: amarilla

14 de mayo de 2013

Documento técnico

ENTRENAMIENTO VIRTUAL DE INTERVENCIONES QUIRÚRGICAS, UTILIZANDO SENSORES KINECT, CON PERCEPCIÓN DE SENSACIONES FÍSICAS

Presentan

Villada Quezada Marco Antonio¹
Damián Sandoval Sergio²
Sánchez García Gabriel³
López Flores Iván⁴
Carreto Terán José Francisco⁵

Directores

M. en C. Dorantes Gonzáles Marco Antonio

M. en C. Cordero López Martha Rosa

RESUMEN

Este documento presenta la forma en que se ensamblan diferentes tecnologías para producir un sólo sistema de simulación de operaciones quirúrgicas. Las tecnologías a ensamblar son los sensores del Kinect de Microsoft para el reconocimiento de movimientos, el motor de gráficos Unity para la física y manejo de gráficos, CUDA para el procesamiento en paralelo de tareas monótonas de renderizado 3D y estimulación física para generar sensaciones hápticas. Estas tecnologías fueron seleccionadas por la amplia gama de usos que han tenido en los últimos años y debido a la compatibilidad que algunas de ellas presentan entre sí. Como caso de estudio y ejemplo del uso de este sistema se presenta una simulación de la operación de revascularización coronaria, en la cual se presentan diferentes eventos controlados que puedan ocurrir en una operación de este tipo.

PALABRAS CLAVE: Revascularización coronaria, realidad virtual, circulación extracorpórea, arteria, injerto, percepciones hápticas.

¹ E-Mail: curusniaga@gmail.com

² E-Mail: sergio_cloudff7_ps3@hotmail.com

³ E-Mail: gabito_cane@gmail.com

⁴ E-Mail: loboestepario91@gmail.com

⁵ E-Mail: paco333@gmail.com



ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE FORMACIÓN INTEGRAL E INSTITUCIONAL



COMISIÓN ACADÉMICA DE TRABAJO TERMINAL

México D.F a 14 de Mayo de 2013

ING APOLINAR FCO. CRUZ LÁZARO
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
PRESENTE

Por medio del presente, informamos que los alumnos que integran el TRABAJO TERMINAL 2012-A011 titulado "Entrenamiento Virtual de Intervenciones Quirúrgicas, Utilizando Sensores Kinect Con Percepción de Sensaciones Físicas", concluyeron satisfactoriamente su trabajo.

El empastado del Reporte Técnico y el Disco Compacto (CD) fueron revisados ampliamente por sus servidores y corregidos, cubriendo el alcance y el objetivo planeados en el protocolo original y de acuerdo a los requisitos establecidos por la comisión que Usted preside.

Atentamente

M. en C. Marco Antonio Dorantes González

M. en C. Martha Rosa Cordero López

Directores

Advertencia

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

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen. Información adicional sobre este reporte técnico podrá obtenerse en:

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

Índice general

I	Introducción	1
1.	Introducción	2
1.1.	Planteamiento del problema	2
1.2.	Objetivo general	2
1.3.	Objetivos particulares	3
1.4.	Justificación	3
1.5.	Originalidad del trabajo	3
1.6.	Complejidad	3
2.	Estado del arte	5
2.1.	Comparativa entre motores gráficos	6
II	Marco Teórico	7
3.	Marco teórico del proyecto	8
3.1.	Cirugía de revascularización coronaria	8
3.2.	Escáner de luz estructurada	10
3.3.	Hemodinámica básica	11
3.4.	Triangulación poligonal	12
3.5.	Reconocimiento del esqueleto humano en tiempo real utilizando una imagen de profundidad	14
3.6.	Háptica	14
III	Análisis	16
4.	Análisis del Sistema	17
4.1.	Análisis De Requerimientos	17
4.1.1.	Módulo kinect	17
4.1.2.	Módulo guante	18
4.1.3.	Módulo sistema médico	18
4.1.4.	Módulo herramientas	18
4.1.5.	Módulo sistema virtual	19
4.1.6.	Módulo perfil paciente	19
4.1.7.	Módulo cirujano virtual	19

4.2. Herramientas	20
4.2.1. Software	20
4.2.2. Hardware	20
4.3. Análisis de factibilidad	21
4.3.1. Factibilidad operativa	21
4.3.2. Factibilidad técnica	21
4.3.3. Factibilidad económica	21
4.3.4. Análisis de costos (COCOMO)	21
4.3.5. Factores de ajuste de valor	23
4.3.6. Puntos de función	24
4.4. Análisis de Riesgos	26
IV Diseño del Sistema	30
5. Diseño del Sistema	31
5.1. Diagramas de casos de uso	31
5.1.1. Diagrama general de casos de uso	31
5.1.2. Caso de uso: comando por voz	33
5.1.3. Caso de uso: obtener instrumento	33
5.1.4. Caso de uso: realizar comando por gesto	34
5.1.5. Caso de uso: manipular instrumento	35
5.1.6. Caso de uso: terminar operación	36
5.1.7. Caso de uso: seleccionar perfil paciente	36
5.1.8. Caso de uso: comenzar cirugía	37
5.1.9. Caso de uso: operar	38
5.1.10. Caso de uso: seleccionar perfil usuario	39
5.1.11. Caso de uso: modificar datos usuario	40
5.1.12. Caso de uso: mostrar pantalla de operación	41
5.2. Diagramas de secuencia	42
5.2.1. Diagrama de secuencia 1: representación general del sistema	42
5.2.2. Diagrama de secuencia 2: envío de alarmas y sensaciones físicas al usuario	43
5.2.3. Diagrama de secuencia 3: presentación de interfaces de operación y reporte final	43
5.2.4. Diagrama de secuencia 4: solicitud de instrumento de operación	45
5.2.5. Diagrama de secuencia 5: actualizado de pantalla	45
5.2.6. Diagrama de secuencia 6: interacción con el cliente	45
5.3. Diagrama de Clases	48
5.3.1. Módulo Instrumental	48
5.3.2. Módulo Guante	48
5.3.3. Módulo Sistema Virtual	49
5.3.4. Módulo Sistema Medico	49
5.3.5. Módulo Renderizado	49
5.3.6. Módulo Paciente	49

V	Desarrollo y pruebas del sistema	54
6.	Desarrollo	55
6.1.	Modelos Tridimensionales	55
6.1.1.	Modelos cardiovasculares	55
6.1.2.	Modelos quirúrgicos	56
6.2.	Háptica	59
6.2.1.	Material	59
6.2.2.	Actuadores	59
6.2.3.	Características	60
6.2.4.	Interfaz	60
6.3.	Comando de Voz	63
6.3.1.	Librerías	63
6.4.	Simulación de la Cirugía	64
6.4.1.	Guión de Cirugía	64
6.4.2.	Puntuación de la cirugía	64
6.4.3.	Animación de corte	68
6.4.4.	Modelado Matemático	69
6.5.	Procedimiento de modelado de manos	70
6.5.1.	Reflexión de puntos	70
6.5.2.	Discriminación de puntos	71
6.5.3.	Clustering de puntos	73
6.5.4.	Triangulación	75
6.6.	GUI	77
6.6.1.	Pantalla principal	77
6.6.2.	Creación de perfil	77
6.6.3.	Menú de perfil	77
6.6.4.	Menú de estadísticas	77
6.6.5.	Almacenamiento de la información	78
7.	Pruebas	80
7.1.	Modelos tridimensionales	80
7.1.1.	Sistema cardiovascular	80
7.1.2.	Quirófano	82
7.2.	Háptica	83
7.2.1.	Primer Prototipo	83
7.2.2.	Segundo Prototipo	83
7.2.3.	Tercer Prototipo	83
7.3.	Comando de Voz	85
7.4.	Procedimiento de Modelado de Manos	86
7.4.1.	Reflexión	86
7.4.2.	Discriminación de Puntos	87
7.4.3.	Clusterización	88

<i>ÍNDICE GENERAL</i>	IV
VI Trabajo a futuro	90
VII Conclusiones	92
VIII Anexos	95
7.5. Glosario	96

Índice de figuras

3.1. Bloqueo de una arteria coronaria. Cortesía de NIH	9
3.2. Diagrama del proceso del Escáner de luz estructurada	10
3.3. Diagrama de ejemplo de la Hemodinámica	11
3.4. División de la presión en las venas	11
3.5. Triangulación de puntos en polígonos	13
3.6. Diferentes polígonos pueden ser formados por puntos iguales	13
3.7. En la figura se muestra cómo se clasifican los pixeles por colores dependiendo de la parte del cuerpo en donde se encuentren para después proponer una o más posiciones en 3D de las articulaciones de este.	14
3.8. Data Gloves	15
5.1. Diagrama general de casos de uso	32
5.2. Diagrama de secuencias: diagrama de secuencia de la Representación general del sistema	42
5.3. Diagrama de secuencias: envío de alarmas y sensaciones físicas al usuario	43
5.4. Diagrama de secuencias: presentación de interfaces de operación y reporte final	44
5.5. Diagrama de secuencias: solicitud de Instrumento de operación	45
5.6. Diagrama de secuencias: actualizado de pantalla	46
5.7. Diagrama de secuencias: interacción con el cliente	47
5.8. Diagrama General de Clases	48
5.9. Diagrama de Módulo Instrumental	49
5.10. Diagrama del Módulo Guante y Módulo Sistema Virtual	50
5.11. Diagrama de Módulo Sistema Médico	51
5.12. Diagrama del Módulo Renderizado	52
5.13. Diagrama del Módulo Paciente	53
6.1. Modelos ofrecidos por la página Life Science DB	56
6.2. Modelado de pinzas para injertos	57
6.3. Tijeras de Diethrich	57
6.4. Tubo bifurcado para la bomba extracorpórea	58
6.5. Tubo para la bomba extracorpórea	58
6.6. Shaftless-vibration-motor	59
6.7. Algoritmo de comunicación entre pc/guante	61
6.8. Prototipo final	62
6.9. Vista de la animación de corte en Unity	68
6.10. Iluminación en cono	70

6.11. Reflexión del haz de luz	71
6.12. Algoritmo PIP para polígonos complejos	71
6.13. Ejemplo del Algoritmo DBScan	73
6.14. DBScan	74
6.15. CLARANS	74
6.16. Ejemplo del funcionamiento de DBScan	74
6.17. Triangularización	76
6.18. Pantalla Principal	77
6.19. Creación del Perfil	77
6.20. Menú perfil	78
6.21. Menú de estadísticas	79
7.1. Tomografía Axial Computarizada	81
7.2. Modelos ofrecidos por la empresa ZYGOTE	81
7.3. Software BodyParts3D basado en web.	82
7.4. Tomografía Axial Computarizada	83
7.5. Prototipo final	84
7.6. Modelado de bisturí	85
7.7. Modelo de fórceps	85
7.8. Nube de puntos original	86
7.9. Eliminación de puntos	87
7.10. Clusterización de puntos	88
7.11. Primer prototipo de algoritmo de triangularización	89

Índice de tablas

2.1. Tabla de estado del arte.	5
2.2. Tabla comparativa entre engines.	6
3.1. Algoritmos de triangulación	13
4.1. Tabla de costos del proyecto	22
4.2. Tabla general de riesgos	26
4.3. Tabla de riesgo 1.	26
4.4. Tabla de riesgo 2.	27
4.5. Tabla de riesgo 3.	27
4.6. Tabla de riesgo 4.	27
4.7. Tabla de riesgo 5.	28
4.8. Tabla de riesgo 6.	28
4.9. Tabla de riesgo 6A.	28
4.10. Tabla de riesgo 7.	29
4.11. Tabla de riesgo 8.	29
4.12. Tabla de riesgo 9.	29
6.1. Variables del objeto Collission	68

Parte I

Introducción

Capítulo 1

Introducción

En la historia de la medicina, los cirujanos han tenido que aprender su trabajo, únicamente con experiencia *In situ*, ya sea asistiendo o realizando las cirugías.

Para lograr un entrenamiento lo más cercano a la realidad para todas estas actividades se han utilizado diversas tecnologías como la realidad virtual, el uso de periféricos sensoriales y maniqués enfocados a simular características del cuerpo humano para lograr un nivel de inmersión adecuado.

En la actualidad contamos con diferentes tecnologías las cuales nos pueden ayudar a aumentar este tipo de inmersión, como es el caso de Kinect de Microsoft, que cuenta con un conjunto de sensores que nos ayudaran a brindarle a los cirujanos un medio de entrenamiento diferente y mucho más moderno al que han tenido a través de los años.

1.1. Planteamiento del problema

Hoy en día existen diferentes maneras con las cuales un médico cirujano puede desarrollar tanto su destreza como sus habilidades básicas de cirugía como en distintos y específicos procedimientos comunes en su profesión. Desde maniqués de alta tecnología, cadáveres reales de humanos o animales, personas con vida, hasta ambientes virtuales.

Dependiendo del medio en donde los aspirantes a cirujanos decidan desarrollarse, pueden o no tener algunos si no es que ninguno de estos recursos a su alcance. Incluso para todos ellos, podría ser ventajoso contar con algún método de enseñanza diferente a los libros en etapas tempranas de su formación.

Es por ello que el desarrollo de un sistema virtual que además genere sensaciones hápticas para el usuario podría llegar a ser la base de un cambio en el paradigma de enseñanza en la medicina, tanto en las diferentes etapas de la formación de un médico, como en las diferentes especialidades que existen.

1.2. Objetivo general

Desarrollar un sistema de simulación de intervenciones quirúrgicas que brindará una alternativa diferente y mas económica en la enseñanza de este tipo de intervenciones médicas.

1.3. Objetivos particulares

- Realizar la simulación de un caso en particular: revascularización coronaria.
- Interpretar resultados de manera que puedan ser manipulados y evaluados por el cirujano virtual (emitir alarmas).
- Generar sensaciones hápticas en la mano del cirujano.

1.4. Justificación

Las cirugías actuales y en caso particular la de revascularización coronaria requieren de un gran entrenamiento y de una amplia experiencia y conocimientos por parte del cirujano. Además de que para poder entrenar y perfeccionar las habilidades y técnicas para este tipo de cirugía actualmente se requieren de maniqués robotizados.

Esto nos motiva a realizar esta propuesta para que los alumnos de medicina o cualquier persona en general que tenga interés y la motivación en observar y experimentar una cirugía de revascularización coronaria, puedan realizarla con utilizando un simulador de dicha intervención a bajo costo.

En base a experiencias e información por parte de especialistas en Angiología y Revascularización modelaremos situaciones de la vida real en el tema en cuestión. Esta clase de simulación se implementara con la ayuda de un motor de videojuegos, ya que nos proporciona un punto intermedio entre la teoría y la práctica acentuando las cuestiones teóricas y preparando al usuario para situaciones reales

El sistema ofrecerá problemas imprevistos como suelen ocurrir en la vida real, de tal manera que se ponga a prueba la capacidad para mantener la concentración, la atención y la calma. Claro está que no hay mejor practica que la real; sin embargo, la utilización de este tipo de software mostrará otros aspectos que posiblemente sean inalcanzables en este momento por cierto personal de dichas prácticas.

1.5. Originalidad del trabajo

Actualmente existe una limitada cantidad de sistemas en el mercado o en desarrollo, parecidos al nuestro, lo cual aunado a la implementación de las nuevas tecnologías, brindará una nueva experiencia en la práctica y entrenamiento en la realización de cirugías.

1.6. Complejidad

Debido a la incorporación de tecnologías hápticas y basándonos en los requisitos aparecidos en [1], nos encontramos con los siguientes problemas:

La complejidad de los modelos matemáticos. Nos enfrentamos a un problema intrínseco del uso de kinect: la resolución de la cámara de rayos infrarrojos y la sensibilidad de detección, según los manuales técnicos tienen un margen de error de centímetros, para la cirugía que nosotros nos planteamos, debemos reducir el margen de error (y por tanto la resolución de nuestra cámara) a cuando mucho, un centímetro.

Los modelos 3D y su fidelidad anatómica. Para construir una herramienta realmente útil, debemos tener modelos que correspondan con las medidas promedio usadas en el ámbito médico profesional.

La resolución e independencia del Guante Haptico. Un sistema de simulación real, debe tener la capacidad de entregar al usuario mensajes de situaciones reales y en tiempo real. Además, debemos poder dar una alternativa real a otros sistemas de retroalimentación háptica como los que venden en [22].

Tiempo de Ejecución. El tiempo de ejecución y respuesta debe ser imperceptible para el usuario, por lo que debemos ser cuidadosos y poner especial énfasis en la programación de la simulación, envío y recepción de datos.

Área Médica. Ya que el proyecto involucra a una rama de la ciencia que es ligeramente ajena a las ciencias de la computación, debemos tener cuidado con la interacción que tenemos con profesionales del área y asumir que un posible desapego en las comunicación entre ambas áreas es un riesgo latente en el proyecto.

Integración. La integración de las tecnologías que proponemos es lo suficientemente innovadora como para considerarla un factor de complejidad. Preveemos enfrentarnos a problemas de compatibilidad de drivers, formatos de archivos, etc.

Capítulo 2

Estado del arte

En el estado del arte, podemos apreciar productos similares que se han desarrollado e implementado como simuladores quirúrgicos o métodos de percepción háptica.

Proyecto	Descripción	Ventajas	Desventajas
Simulador de cirugías de próstata.	Ambiente virtual con un modelo en tercera dimensión de la próstata, Aparenta deformaciones y re-secciones de tejido blando con un resectoscopio virtual de interfaz electromecánica.	Modelado de órganos muy parecidos a sus contrapartes. Efectos de iluminación que asemejan complicaciones visuales reales	Aspectos de la simulación son poco realistas debido al la tarjeta de gráficos que utilizan.
LapSim	Sistema de entrenamiento virtual que lleva a los doctores desde habilidades básicas hasta algunos procedimientos comunes como la apendectomía.	Gama extensa de procedimientos básicos. Diferentes escenarios reales para cada uno de los procedimientos.	Bajo rendimiento gráfico. No provee sensaciones físicas.
Dedos inteligentes	Circuitos con pistas de oro, instalados sobre un polímero que a su vez es acoplado a un tubo de silicona con la forma de un dedo para transmitir sensaciones.		En desarrollo
Simuladores médicos de alta tecnología	Maniqués de alta tecnología con diferentes fines para el adiestramiento médico.	Tecnología utilizada actualmente	

Tabla 2.1: Tabla de estado del arte.

2.1. Comparativa entre motores gráficos

Debido a la complejidad del proyecto, y el tiempo para su elaboración, el desarrollar un engine propio, para detectar todas las colisiones, es un reto ya no alcanzable, es por ello que optamos, por adoptar un engine, que nos reduzca el tiempo de programación y que a la vez nos dé una mejor calidad de entrega en el proyecto.

Después de descartar posibles opciones, como engine a utilizar escogimos el motor de videojuegos Unity justificando esta decisión con ayuda de la tabla comparativa mostrada a continuación.

Características	Unity	Unreal Engine
Mayor integración, de herramientas para su fácil uso.	X	
Mayor documentación	X	
Graficas más realistas		X
Menor curva de Aprendizaje	X	
Integración con DLLs externas.	X	X
Tipo de Licencia	Propietaria, Libre para uso no comercial	Propietaria, Libre para uso no comercial
Lenguajes de programación soportados	Java Script, C#, Boo	UnrealScript (similar a C++)
Soporte multijugador	Hasta más de 1000 usuarios al mismo tiempo	Solo 64 usuarios al mismo tiempo
Mejor integración con CUDA	X	

Tabla 2.2: Tabla comparativa entre engines.

Parte II

Marco Teórico

Capítulo 3

Marco teórico del proyecto

3.1. Cirugía de revascularización coronaria

Esta cirugía consiste en crear una nueva ruta, llamada derivación (bypass), para que la sangre y el oxígeno lleguen al corazón, pues las arterias que comunican al corazón se encuentran bloqueadas.

La ejecución de esta cirugía es la siguiente:

1. Aplicar Anestesia general.
2. Realizar incisión en el tórax.
3. Canulación de la arteria para la conexión a bomba extracorpórea ¹.
4. Obtención del injerto que derivará la sangre alrededor del área bloqueada en la arteria. Este injerto puede obtenerse de tres formas:
 - El médico puede usar una vena llamada safena, que está en la pierna. Para llegar a esta vena, se hará una incisión quirúrgica en la parte interna de la pierna, entre el tobillo y la ingle. Un extremo del injerto se suturará a la arteria coronaria y el otro se suturará a una abertura hecha en la aorta.
 - También se puede emplear como injerto un vaso sanguíneo del tórax llamado arteria mamaria interna (AMI). Un extremo de esta arteria ya está conectado a la aorta y el otro extremo va pegado a la arteria coronaria.
 - También se pueden emplear otras arterias para injertos en la cirugía de revascularización; la más común es la arteria radial en la muñeca.
5. Se ocluye la aorta y se administra solución cardiopléjica en la raíz de la arteria.
6. Se incide la arteria coronaria, y la vena, la arteria mamaria interna u otro vaso que se vaya a usar como injerto puente se anastomosa con la arteriotomía coronaria.
7. Se desclampea la aorta.
8. Los injertos venosos y los injertos libres se anastomosan con la aorta ascendente. Se pueden anastomosar mientras el clampeo cruzado está aplicado.

¹Según [2], [4],[5] y [6], es la forma más segura y la forma más común de realizar esta cirugía, por lo que en este trabajo terminal decidimos utilizar esta situación como un modelo base

9. Se retira la circulación extracorpórea y se realiza la descanulación.
10. Se colocan los electrodos del marcapasos y los tubos de drenaje.
11. Se cierra la herida.

Esta cirugía puede demorar de 4 a 6 horas. Después de la intervención, el paciente es llevado a la sala de cuidados intensivos.

Después de la operación, el paciente permanece de 3 a 7 días en el hospital y pasará la primera noche en una unidad de cuidados intensivos y probablemente lo pasen a una sala regular o de cuidados transitorios al cabo de 24 horas.

Durante dos o tres días, el paciente tendrá conectadas dos o tres sondas en el pecho para drenar el líquido del área del corazón. Puede que también tenga un catéter en la vejiga para drenar la orina.

Igualmente, puede tener vías intravenosas para los líquidos. Así mismo, el paciente estará conectado a máquinas que monitorean su pulso, temperatura y respiración.

Al cabo de 4 a 6 semanas, el paciente empezará a sentirse mejor y podrá salir del hospital reanudando algunas de las rutinas que requieran de poco esfuerzo físico.

Los riesgos de esta cirugía son:

- Coágulos de sangre en las piernas que pueden viajar a los pulmones.
- Problemas respiratorios.
- Infección, incluyendo los pulmones, las vías urinarias y el tórax.
- Pérdida de sangre.
- Infección de la herida en el pecho, que es más probable que suceda si el paciente es obeso, tiene diabetes o ya le han realizado esta cirugía antes.
- Ataque cardíaco o accidente cerebrovascular.
- Problemas del ritmo cardíaco.
- Insuficiencia pulmonar o renal.
- Febrícula y dolor torácico, juntos denominados síndrome pospericardiotomía, los cuales pueden durar hasta 6 meses.
- Pérdida de memoria, pérdida de claridad mental o "pensamiento confuso".

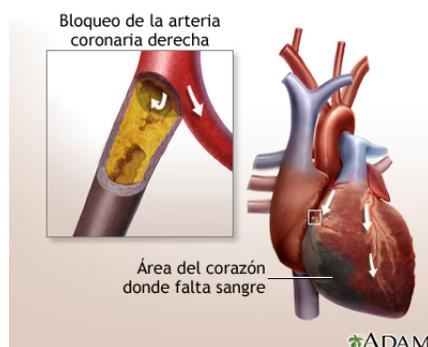


Figura 3.1: Bloqueo de una arteria coronaria. Cortesía de NIH

3.2. Escáner de luz estructurada

El escáner de luz estructurada es un dispositivo capaz de capturar la forma y características de un objeto mediante la proyección de un patrón de luz y su registro en un sistema de adquisición.

El escáner 3D consta de una fuente de luz que proyecta el haz y una cámara que captará los puntos o líneas de las superficies separados entre sí. Para escanear el objeto se define un sistema de coordenadas esféricas para determinar cada punto del espacio tridimensional que se está capturando.

En su forma básica (punto único) se proyecta un haz de luz sobre la superficie del objeto que genera un punto de iluminación en el sensor CCD (cámara) del sistema. Esta técnica sin embargo, requiere la exploración de los dos ejes por separado (con el añadido del retardo). El patrón de iluminación capturado variará según el haz de luz que se utilice: single dot, rendija, patrón de rallas o reja. Para la mayoría de las situaciones un sólo escaneo no producirá un modelo completo del objeto. Generalmente se necesitarán múltiples escáneres, desde diferentes direcciones, para obtener la información tridimensional completa (barrido).

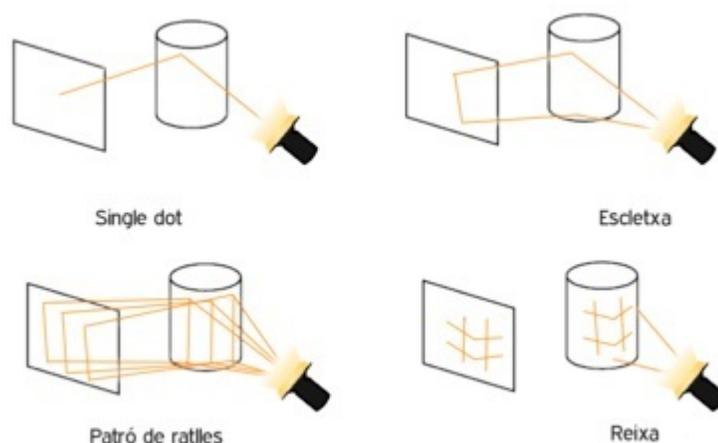


Figura 3.2: Diagrama del proceso del Escáner de luz estructurada

La técnica más rápida en cuanto a tiempo de captura, es la proyección de patrones bidimensionales como el patrón de rallas o de reja. La primera (y la más utilizada) consiste en la proyección de multitud de rallas correlativas entre sí o distribuidas arbitrariamente. La segunda proyecta una reja de $m \times n$ puntos que iluminan parte del objeto de una sola tirada. El tiempo se reduce debido a que de esta forma se toman diversas muestras de una sola vez.

Existen varios métodos para proyectar la luz necesaria para capturar en 3D al objeto. El más utilizado hoy en día y el más pertinente para esta investigación es el método basado en luz estructurada infrarroja, el cual por el tipo de luz que utiliza no se altera por el color o la iluminación de la escena.

Para reducir la distorsión ocasionada por la cámara que se utiliza, se utilizan modelos matemáticos los cuales toman en cuenta tanto aberraciones ópticas como distorsiones geométricas que puedan causar el proyector y sus lentes. Incluso se le pueden realizar ajustes al modelo aplicándole fotogrametría a la imagen.

3.3. Hemodinámica básica

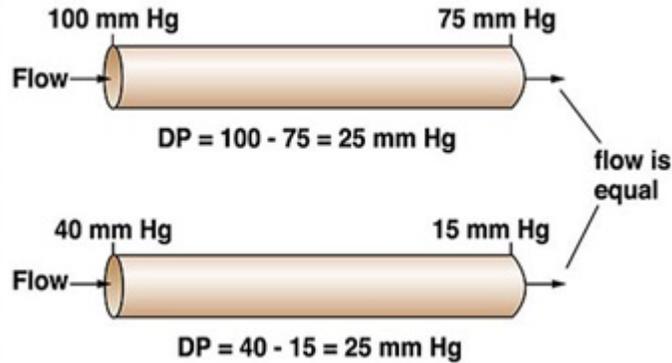


Figura 3.3: Diagrama de ejemplo de la Hemodinámica

Un fluido se desplaza en el interior de un tubo cuando la presión en el inicio es superior a la existente al final del tubo, moviéndose desde una zona de mayor presión a una de menor presión.

El flujo depende directamente de la diferencia de presión entre esos dos puntos e inversamente de la resistencia.

La resistencia depende de las dimensiones del tubo y de la naturaleza del fluido, y mide las fuerzas de rozamiento o fricción entre las propias moléculas del fluido y entre éstas y las moléculas de la pared del tubo.

La velocidad con la que circula la sangre en el interior de un tubo es directamente proporcional al flujo e inversamente proporcional al área transversal del tubo.

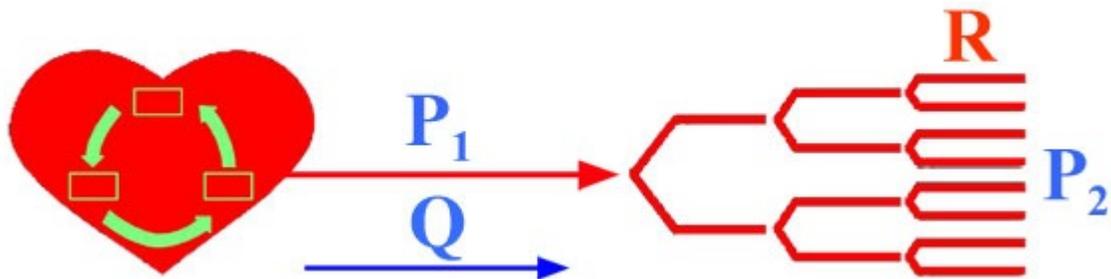


Figura 3.4: División de la presión en las venas

$$Q(\text{flujo}) = \frac{\Delta P(P_1 - P_2)}{R(\text{resistencia})}$$

El flujo (*volumen/minuto*) se define también como el volumen circulante por un segmento transversal del circuito en la unidad de tiempo:

$$v = \frac{Q}{\pi r^2}$$

La resistencia no puede medirse directamente por ser una magnitud compuesta, pudiendo obtenerse de la ecuación inicial al establecer un gradiente de presión entre dos puntos y medir el flujo que se establece:

$$Flujo = \frac{\Delta P}{Flujo}$$

(*mmHg.min/ml, URP* → unidad de resistencia periférica hemodinámica)

Su magnitud depende de las dimensiones del tubo por donde circula el fluido, de su viscosidad y del tipo de flujo o corriente que se realice.

En flujos laminares que se desarrollan en tubos cilíndricos, se pueden deducir las relaciones entre la intensidad del flujo, el gradiente de presión y la resistencia o fuerzas de fricción que actúan sobre las capas de envoltura.

La Ley de Poiseuille (o de Hagen-Poiseuille) es una ecuación hemodinámica fundamental en la que se establece:

$$Flujo = \frac{\Delta P \pi r^4}{8 \eta l}$$

8 es el factor que resulta de la integración del perfil de la velocidad.

Debido a que la longitud de los vasos y la viscosidad son relativamente constantes, el flujo viene determinado básicamente por el gradiente de presión y por el radio. De la ecuación representada, destaca el hecho de que el radio al estar elevado a la cuarta potencia, se constituye como el factor más importante. Si suponemos un vaso con un flujo de *1ml/seg* al aumentar el diámetro dos veces el flujo pasa a ser de *16ml/seg*, y si el diámetro aumenta cuatro veces el flujo pasará a ser *256ml/seg*. Por esta relación se puede justificar el papel preponderante que los cambios en el radio del conducto juegan en la regulación del flujo sanguíneo.

La ecuación de Poiseuille está formulada para flujos laminares de fluidos homogéneos con viscosidad constante, sin embargo, en los vasos sanguíneos estas condiciones no siempre se cumplen; si la velocidad del flujo es alta o si el gradiente de presión es elevado, se pueden generar remolinos o turbulencias que modifican el patrón del flujo. Al producirse turbulencias se necesitarán gradientes de presión mayores para mantener el mismo flujo.

3.4. Triangulación poligonal

Triangulación es el problema general de subdividir un dominio espacial en símplexes (o simplíces), lo que en el plano significa triángulos.

Para triangular un polígono simple (cuyos lados no se intersectan) P es necesario dibujar diagonales entre pares de sus vértices. Una diagonal es un segmento de línea que conecta dos vértices de P y cae en el interior de dicho polígono.

De manera más formal la triangulación de un polígono es la descomposición de éste en triángulos utilizando para ello el conjunto máximo de diagonales que no se intersectan. Se requiere que el conjunto de diagonales que no se intersectan sea máximo para garantizar que ningún triángulo tenga un vértice del polígono en el interior de una de sus aristas. Esto podría suceder si el polígono tiene tres vértices colineales consecutivos.

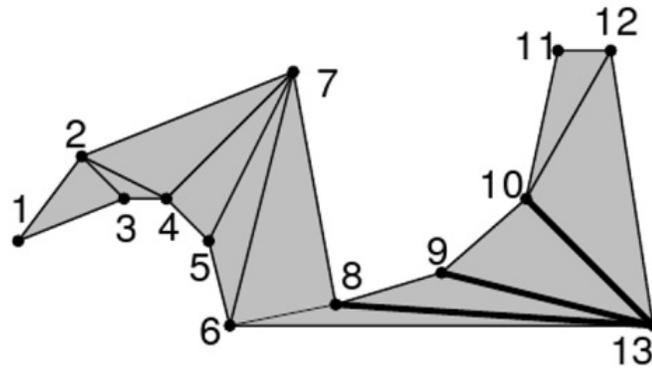


Figura 3.5: Triangulación de puntos en polígonos

Usualmente la triangulación de un polígono no es única:

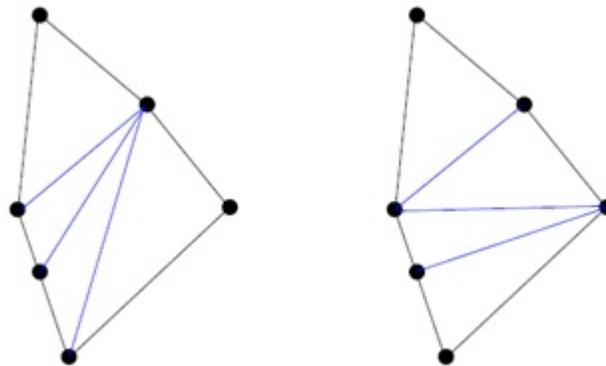


Figura 3.6: Diferentes polígonos pueden ser formados por puntos iguales

Todo polígono simple admite una triangulación, y toda triangulación de un polígono simple con n vértices consiste exactamente de $n - 2$ triángulos

Computacionalmente existen varios algoritmos para resolver este problema utilizando diferentes propiedades:

Propiedad	Algoritmo	Orden
Delaunay	Varios	$O(n \bullet \log n)$
Minimiza máximo ángulo	Inserción rápida de aristas	$O(n^2 \bullet \log n)$
Minimiza máxima pendiente	Inserción de aristas	$O(n^3)$
Minimiza longitud total	Algoritmos de aproximación	$O(n \bullet \log n)$

Tabla 3.1: Algoritmos de triangulación

3.5. Reconocimiento del esqueleto humano en tiempo real utilizando una imagen de profundidad

Este método, utilizado por el Kinect de Microsoft, propone el predecir las posiciones en 3D de las articulaciones de un cuerpo con una sola imagen de profundidad de este, es decir, sin utilizar ningún tipo de información temporal. El acercamiento que se le da al problema es el de reconocimiento de objetos, diseñando una representación de las partes del cuerpo intermedia la cual convierte el difícil problema de estimación de postura a un problema de clasificación pixel por pixel mucho más simple. El conjunto de datos (los cuales son tanto reales como sintéticos) que se utilizó para entrenar al clasificador es bastante vasto y variado que le permite estimar las partes del cuerpo sin importar su posición, la forma del cuerpo, la ropa etc.

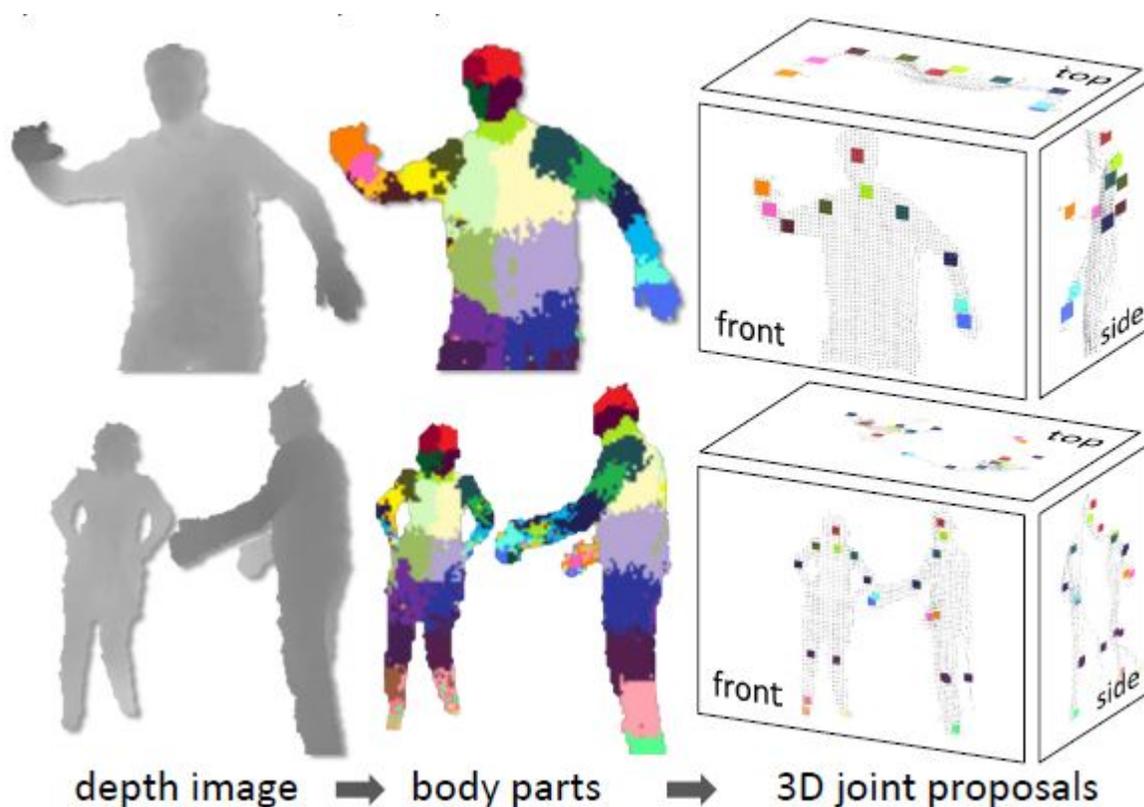


Figura 3.7: En la figura se muestra cómo se clasifican los píxeles por colores dependiendo de la parte del cuerpo en donde se encuentren para después proponer una o más posiciones en 3D de las articulaciones de este.

3.6. Háptica

Se refiere al conjunto de tecnologías que interactúan con las personas mediante el sentido del tacto; que con el uso de dispositivos de entrada/salida especiales (guantes de datos u otros dispositivos), los usuarios pueden recibir información de las aplicaciones informáticas en forma de

sensaciones sentidas en la mano u otras partes del cuerpo.

Se trata de una tecnología emergente desde aproximadamente los años 90 y que ha estado evolucionando prometedoramente, dando lugar a grandes avances en el campo de la realidad aumentada.

Estamos acostumbrados a interactuar con creaciones tecnológicas a través de interfaces que son táctiles en el sentido hombre-máquina, pero visuales o auditivas en el sentido máquina-hombre. Tomemos como ejemplo un ordenador: podemos introducir datos mediante toda una colección de dispositivos basados en el tacto, tales como teclados, ratones, pantallas touchscreen. Pero la obtención de información desde el ordenador se produce mediante interfaces basadas casi exclusivamente en la vista (pantallas LED, documentos impresos) o en el oído (sonidos de alarma, banda sonora en juegos, etc.). Es por eso que muchas compañías han empezado a apostar por esta tecnología,



Figura 3.8: Data Gloves

logrando que actualmente exista toda una gama de dispositivos que se caracterizan por permitir la interacción puntual con el objeto virtual a través de un terminal, materializado como un lápiz, dedo virtual o un joystick. Sin embargo, con este tipo de dispositivos no permite la manipulación en múltiples puntos de contacto sintiendo toda su superficie, y su uso se limita a sistemas de realidad virtual donde el usuario interactúa con el entorno a través de un punto.

Cuando los requisitos del sistema virtual requieren que el usuario interactúe con la mano entera para poder tocar, manipular y coger los objetos virtuales, existen los llamados "Data Gloves". Sin embargo, la mayoría de estos guantes están diseñados para controlar el entorno virtual de forma un poco más natural, más no para hacer percibir al usuario algún tipo de sensación. Es por eso que existen otro tipo de guantes (Tactile Gloves), los cuales son más ligeros que los Data Gloves y emplean normalmente vibradores electromecánicos para proporcionar datos de textura y rugosidad. La colocación de los actuadores es más simple que en los Data Gloves, lo que facilita su diseño, reduciendo a su vez el costo de fabricación.

Parte III

Análisis

Capítulo 4

Análisis del Sistema

El siguiente listado de requerimientos funcionales está basado en una visita al área de cardiología del hospital Juárez en donde el Dr. Carlos Viveros Contreras nos recibió y nos hizo favor de mostrarnos el quirófano y presentarnos con un grupo de expertos los cuales nos mostraron a grandes rasgos lo que se hacía en esa área. Además, nos platicaron acerca de todo el proceso que se realizaba para una operación al corazón, las herramientas que se utilizaban, el personal que se requería, el equipo tecnológico con el que trabajan y muchos otros detalles acerca del procedimiento.

También nos basamos en libros los cuales los mismos expertos nos recomendaron para poder sacar detalles más específicos acerca de la participación de cada uno de los participantes en la cirugía y de las herramientas.

4.1. Análisis De Requerimientos

4.1.1. Módulo kinect

Requerimientos funcionales

RF1: El usuario deberá ser capaz de obtener "herramientas virtuales" por parte del sistema a partir de comandos de voz, las cuales son:

- Vena safena
- pinzas
- bisturí
- tijeras
- porta agujas
- sujeta arterias
- catéter

RF2: El usuario deberá ser capaz manipular el sistema a partir de comandos de gestos para la interacción con el sistema.

RF3: El sistema deberá reconocer las manos del usuario en el entorno físico

Requerimientos no funcionales

RNF1: El sistema requerirá un espacio más reducido en comparación con el que se requiere en un ambiente real.

4.1.2. Módulo guante**Requisitos funcionales**

RF1: El usuario deberá percibir sensaciones físicas con ayuda del guante, de acuerdo a la interacción que se tenga con el sistema.

RF2: El sistema deberá enviar alarmas al guante de acuerdo a los acontecimientos en la cirugía.

Requisitos no funcionales

RNF1: El guante deberá ser ligero para no afectar físicamente al usuario.

RNF2: El sistema reconocerá la comunicación del guante mediante USB.

4.1.3. Módulo sistema médico**Requisitos funcionales**

RF1: El sistema deberá presentar ritmo cardíaco, presión, tiempo de operación, en un apartado de monitoreo que podrá visualizar el usuario.

RF2: El sistema alertará mediante sonido cuando alguno de los signos vitales entre en un punto crítico.

Requisitos no funcionales

RF1: El sistema presentará los resultados de la operación en base a las acciones realizadas durante la cirugía.

RF2: El sistema deberá simular eventos médicos aleatorios que sean inherentes a la operación.

4.1.4. Módulo herramientas**Requisitos funcionales**

RF1: El sistema deberá cargar la animación adecuada de la "herramienta" de acuerdo al evento que el usuario realice con éste.

RF2: El sistema deberá responder con una alarma por medio del guante o sonido adecuado a las acciones que el usuario realice con éste.

Requisitos no funcionales

RNF1: Se deberá desarrollar los modelos de cada uno de los siguientes instrumentos de operación:

RNF2: El sistema deberá simular el contacto de las "herramientas virtuales" con el modelo de paciente.

4.1.5. Módulo sistema virtual

Requisitos funcionales

RF1: El sistema deberá reconocer el movimiento del usuario para la realización de un corte al paciente y se deberá responder de acuerdo a este evento simulando una operación real.

RF2: El sistema deberá contar con alta sensibilidad al movimiento para hacer la simulación lo más cercano posible a la realidad para los cortes.

RF3: El sistema deberá de crear una renderización del ambiente real, en especial de las manos del usuario.

Requisitos no funcionales

RNF1: El sistema sólo podrá utilizarse para sistemas operativos Windows Vista y 7.

RNF2: El sistema contará con efectos de sonido para generar un ambiente más realista.

RNF3: El sistema contará con una funcionalidad de zoom para poder visualizar mejor zonas delicadas.

RNF4: El sistema contará con una vista donde se observe la simulación gráfica de la operación y otra con el monitor de signos vitales.

4.1.6. Módulo perfil paciente

Requisitos funcionales

RF1: El usuario podrá seleccionar diferentes tipos de perfiles de pacientes previamente establecidos.

4.1.7. Módulo cirujano virtual

Requisitos funcionales

RF1: El usuario podrá llenar un perfil para conocer sus estadísticas y avances en el sistema.

RF2: El sistema podrá ser capaz de darle al usuario una evaluación de desempeño.

Requisitos no funcionales

RNF1: El sistema será capaz de guardar la operación realizada por el usuario en un vídeo para computadora.

4.2. Herramientas

El motor de juegos Unity se utilizó debido a que cuenta con un tipo de licencia libre para uso no propietario, lo cual se amolda perfectamente a las necesidades del proyecto. Además también se decidió utilizar este motor de juegos ya que cuenta con abundante documentación, integración con DLLs externas, integración con CUDA y soporte para el lenguaje de programación con C#.

Debido a que se tenían que hacer gran cantidad de cálculos y muchos de estos repetidos para el renderizado de las manos, se decidió utilizar CUDA la cual es una arquitectura de cálculo en paralelo de NVIDIA que aprovecha la gran potencia de la GPU para proporcionar un incremento extraordinario del rendimiento del sistema.

Se optó por utilizar la placa de entrenamiento Arduino, debido a que cuenta con las siguientes características: Precio accesible, multiplataforma, entorno de programación simple y directo, software y hardware ampliable y de código abierto.

Puesto que la arquitectura CUDA es propietaria de NVIDIA, no había otra opción más que utilizar una tarjeta gráfica de esta compañía.

En vez de usar el SDK de Kinect se utilizó OpenNI ya que este es multiplataforma, open source y existe extensa documentación.

4.2.1. Software

Como herramientas de software se eligieron las siguientes:

- Unity como engine para programar los gráficos y la simulación.
Por su amplia documentación y características de compatibilidad con librerías externas, así como la familiarización que ya se posee con este.
- OpenNI para drivers de Kinect.
Por permitirnos mayor grado de libertad y, al igual que el anterior, por su documentación.
- CUDA como librería de procesos en paralelo.
Por ser una tecnología accesible para todo el público y su buen comportamiento con el procesamiento de gráficos y procesos concurrentes.
- Los drivers y librerías Arduino.

4.2.2. Hardware

- Kinect como dispositivo sensorial.
Lo elegimos por la gama de aplicaciones e investigación que se ha realizado con este producto.
- Arduino UNO como tarjeta de interfaz Computadora-Guante.
La facilidad de uso y el bajo costo que supone esta tarjeta, así como sus capacidades de almacenamiento nos hizo inclinarnos hacia su uso.
- Tarjetas gráficas NVIDIA para el procesamiento gráfico.
Al utilizar CUDA como tecnología de procesamiento en paralelo, nos vemos forzados a utilizar estas tarjetas.

4.3. Análisis de factibilidad

4.3.1. Factibilidad operativa

Contamos con un equipo de 5 personas, estudiantes de ingeniería en sistemas computacionales plan 2009, uno con carrera técnica en sistemas digitales, uno con carrera técnica en telecomunicaciones, dos con carrera técnica en computación y uno con estudios de preparatoria bilingüe. Todos con al menos el 60 % de la carrera.

4.3.2. Factibilidad técnica

Nuestro sistema utilizará como base de funcionamiento sensores de profundidad. Este tipo de tecnología está a nuestro alcance gracias al Kinect de Microsoft, el cual puede comunicarse con la computadora gracias al puerto USB con el que cuenta. Además cuenta con 3 micrófonos los cuales utiliza para distinguir el usuario que envía órdenes por medio de la voz.

Para procesar todos los datos que nos proporcionen los sensores de profundidad y los micrófonos, utilizaremos los drivers y las bibliotecas proporcionadas por la empresa OpenNI las cuales nos dan el control total de la información de los sensores.

Para utilizar todos estos datos y para el ambiente virtual se utilizará el motor de gráficos Unity el cual está disponible para descargar.

Otra de las piezas clave para cumplir con la funcionalidad de nuestro sistema, es el guante, debido a que será el que generará las sensaciones hápticas. Debido al amplio rango de sensaciones, las cuales se pueden desarrollar de diferentes maneras, haciendo uso de diferentes componentes electrónicos; y debido a que muchos de estos dispositivos no están disponibles debido a que se encuentran en desarrollo, se creará un guante propio con las sensaciones hápticas necesarias para nuestro sistema.

Al menos uno de los integrantes cuenta con experiencia en programación con el motor gráfico Unity y las bibliotecas de OpenNI.

Otro integrante cuenta con amplia experiencia y habilidad en circuitos electrónicos.

Todos cuentan con experiencia en programación orientada a objetos la cual ayudará a complementar los conocimientos que se requieren para programar con el motor gráfico y las bibliotecas para los sensores.

4.3.3. Factibilidad económica

El motor gráfico y las librerías para manejar la información de los sensores se encuentran disponibles bajo la licencia GNU. Los recursos tales como internet, luz, agua y mobiliario serán proporcionados por la escuela y por los mismos miembros del equipo. Los equipos de cómputo necesarios para el desarrollo del sistema son propiedad de los miembros del equipo. Se tiene que adquirir únicamente el Kinect de Microsoft y los componentes utilizados para el guante háptico, cuyos costos estimados se ven reflejados en la tabla de costos.

La estimación del costo por el desarrollo del software se desglosa a continuación.

4.3.4. Análisis de costos (COCOMO)

Debido a que nuestro proyecto es de naturaleza híbrida y la metodología de desarrollo es en Espiral, se decidió utilizar COCOMO como modelo de estimación de costos para el proyecto. Esto debido a dos razones:

	Costo Unitario	Unidades	Total
Guante			
Guantes Latex	\$20.00	5 piezas	\$100.00
Cable magnético	\$3.00	10 metros	\$30.00
Tarjeta recepción de datos	\$500.00	4 piezas	\$2,000.00
Equipo de Detección			
Kinect	\$2,500.00	2 equipos	\$5,000.00
Computadora	\$4,000.00	1 equipos	\$4,000.00
Monitor	\$2,200.00	1 equipos	\$2,200.00
Software			
Sistema Software	\$90,916.89	1	\$90,916.89
Total			\$104,246.89

Tabla 4.1: Tabla de costos del proyecto

1. La difusión que tiene en la industria de los proyectos informáticos.
2. Su gran eficiencia en la estimación de costos para proyectos híbridos.

Pese a que COCOMO no está diseñado para soportar metodologías orientadas a objetos, existen varios reportes técnicos que nos brindan promedios de puntos por función para lenguajes orientados a objetos.

Entrada externa

Mapa de profundidad	4
Mapa de textura	4
Audio	6
Mapa de Sensaciones (Guante)	4
Modelo Renderizado Procesado	6
Modelo del instrumental de operación	6
Base de Datos	3
Total	33

Salida externa

Señales de alerta al guante	5
Reporte de resultado de la operación	7
Monitor ritmo cardiaco	5
Monitor presión	5
Monitor nivel oxigeno	5
Modelo virtualizado de las manos	7
Base de Datos	4
Total	38

Consultas externas: 4

Vídeo de la operación realizada	4
Total	4

Archivos lógicos internos: 15

Vídeo de resultado de la operación realizada	15
Total	15

Archivos de interfaz externos: 10

Bases de datos	10
Total	10

Conteo Total: 100**4.3.5. Factores de ajuste de valor**

1. ¿El sistema requiere respaldo y recuperación confiables?
Valor: **3**
Debido a que nuestro sistema requiere que guarde el vídeo de la operación, así como el puntaje del Doctor que está haciendo la operación virtual.
2. ¿Se requiere comunicaciones de datos especializadas para transferir información a la aplicación u obtenerla de ella?
Valor: **5**
Debido a la comunicación que se tiene entre la aplicación y el guante de sensaciones físicas.
3. ¿Hay funciones distribuidas de procesamiento?
Valor: **0**
Debido a que todo se hace sobre la misma máquina y no requiere estar distribuido.
4. ¿El desempeño es crítico?
Valor: **5**
Debido a los riesgos humanos involucrados.
5. ¿El sistema se ejecutará en un entorno existente que tiene un uso pesado de operaciones?
Valor: **5**
El sistema es en tiempo real, por lo que las operaciones realizadas son bastantes.
6. ¿EL sistema requiere entrada de datos en línea?
Valor: **0**
Debido a que todo es en la misma computadora, y no es necesario subirlo a ningún servidor.

7. ¿La entrada de datos en línea requiere que la transacción de entrada se construya en varias pantallas u operaciones?
Valor: **0**
No hay entrada de datos en línea.
8. ¿Los archivos lógicos internos se actualizan en línea?
Valor: **0**
Todo se realiza en la misma maquina.
9. ¿Las entradas, las salidas, los archivos o las consultas son complejos?
Valor: **5**
Debido al Kinect, la entrada del mapa de profundidad, y la salida para el guante para las sensaciones físicas.
10. ¿Es complejo el procesamiento interno?
Valor: **5**
Debido a los volúmenes de información que se manejan, y guardan.
11. ¿El código utilizado será reutilizable?
Valor: **5**
Debido a que estamos incursionando en la innovación y se requerirá de otro tipo de operaciones en el futuro lejano.
12. ¿Se incluye la conversión e instalación en el diseño?
Valor: **3**
13. ¿Está diseñado el sistema para instalaciones múltiples, en diferentes organizaciones?
Valor: **5**
Debido a que no está sujeto a un sólo entorno.
14. ¿La aplicación está diseñada para facilitar el cambio y para que el usuario lo use fácilmente?
Valor: **5**
Aunque solamente para una operación en específico.

Total de Factores de Ajuste: 46

4.3.6. Puntos de función

Para obtener los puntos de función, se usaron las medidas estándar que el proceso nos indica:

$$PF = \text{conteo total} * [0,65 + 0,01 * \text{Factores de Ajuste}]$$

$$PF = 100 * [0,65 + 0,46] = 100 * 1,11$$

$$PF = 1,11$$

$$LCD = 1,11 * 59$$

$$LCD = 6,549$$

Las constantes son las estándares para COCOMO:

<i>Proyecto de Software</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2,40	1,05	2,50	0,38
Semiacoplado	3,00	1,12	2,50	0,35
Empotrado	3,60	1,20	2,50	0,32

$$E = aKLCD^b$$
$$D = cE^d$$

E es el esfuerzo aplicado en persona – mes.

D es el tiempo de desarrollo en meses cronológicos y $KLDC$ es el número estimado de líneas de códigos distribuidas (miles) para el proyecto.

El tipo de proyecto a utilizar será de tipo empotrado, sustituyendo las variables, las formulas quedan:

$$E = 3,6 * 6,549^{1,2}$$
$$E = 34,33 \text{ personas-mes}$$
$$D = 2,5 * E^{0,32}$$
$$D = 2,5 * 34,33^{0,32}$$
$$D = 7,6779 \text{ meses}$$

El valor de la duración del proyecto permite al planificador recomendar un número de N personas para el proyecto.

$$N = E/D$$
$$N = 34,33/7,6779$$
$$N = 4,4712 \text{ personas}$$

En nuestro equipo contamos con 5 personas y 11 meses,

4.4. Análisis de Riesgos

Analizando el proyecto y las diferentes dificultades que se pueden presentar, se propusieron los siguientes riesgos:

Riesgo	Tipo
No terminar en tiempo y forma el proyecto establecido.	Proyecto
Uno o más de los integrantes del equipo fallen con el trabajo.	Proyecto
Falta de recursos para la construcción del hardware.	Producto
Falla de algún componente del Hardware.	Producto
Los procesos utilizados hagan lento el sistema y se aprecie la diferencia de sincronía entre el mundo virtual y el real.	Proyecto
Falta de sincronía en la red distribuida propuesto como plan de contingencia.	Proyecto
No conseguir la información requerida para la elaboración de casos de usos, a la institución de salud que se le solicito ayuda.	Proyecto
Falta de realismo en el ambiente virtual.	Proyecto
Fallo de transmisión de datos desde el guante al Kinect.	Proyecto

Tabla 4.2: Tabla general de riesgos

Esta tabla se desglosa a continuación:

ID de Riesgo: 1	Fecha: 6 / 08 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	No terminar en tiempo y forma el proyecto establecido.		
Contexto:	No cumplir las actividades en los tiempos establecidos, debido a factores externos, debido a que se contempla colaboración de personas externas.		
Reducción:	Tener tiempos de holgura que reduzcan el impacto de retraso, así como para contemplar planes alternativos.		
Plan de Contingencia:	Contemplar todas las posibles alternativas, para poder terminar a tiempo lo establecido.		

Tabla 4.3: Tabla de riesgo 1.

ID de Riesgo: 2	Fecha: 6 / 08 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	No terminar en tiempo y forma el proyecto establecido.		
Contexto:	No cumplir las actividades en los tiempos establecidos, debido a factores externos, debido a que se contempla colaboración de personas externas.		
Reducción:	Tener tiempos de holgura que reduzcan el impacto de retraso, así como para contemplar planes alternativos.		
Plan de Contingencia:	Contemplar todas las posibles alternativas, para poder terminar a tiempo lo establecido.		

Tabla 4.4: Tabla de riesgo 2.

ID de Riesgo: 3	Fecha: 6 / 08 / 2012	Probabilidad: Bajo	Impacto: Medio
Descripción:	Uno o más de los integrantes del equipo fallen con el trabajo.		
Contexto:	Debido a que un integrante, está en pendiente su trámite de TT, ya que no cumplía con los créditos, y hasta el ETS, se conocería su situación final.		
Reducción:	Apoyar al compañero en todo lo posible para que pueda lograr obtener sus créditos a tiempo.		
Plan de Contingencia:	Reducir el alcance del proyecto justificando la pérdida de un miembro del equipo, sin perder de vista el objetivo planteado.		

Tabla 4.5: Tabla de riesgo 3.

ID de Riesgo: 4	Fecha: 6 / 08 / 2012	Probabilidad: Bajo	Impacto: Catastrófico
Descripción:	Falta de recursos para la construcción del hardware.		
Contexto:	La construcción del prototipo final requiere de precisión para tener éxito en la construcción de la simulación, y alguno de los materiales involucrados podría ser difícil de conseguir.		
Reducción:	Investigar alternativas en tecnologías y recursos utilizados, para tener un rango amplio de alternativas en caso de no conseguir un componente.		
Plan de Contingencia:	Usar alternativas dada su funcionalidad, y de no ser posible encontrar un reemplazo comprobar su existencia y buscar la manera de conseguirla mediante otros medios.		

Tabla 4.6: Tabla de riesgo 4.

ID de Riesgo: 5	Fecha: 6 / 08 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	Falla de algún componente del Hardware.		
Contexto:	Algún componente deje de funcionar o bien no funcione adecuadamente, previo o durante alguna de las presentaciones, o bien en simulaciones para su funcionamiento.		
Reducción:	Depender lo menos posible de los componentes físicos para reducir su impacto de fallo o mal funcionamiento.		
Plan de Contingencia:	Depender lo menos posible de los componentes físicos para reducir su impacto de fallo o mal funcionamiento.		

Tabla 4.7: Tabla de riesgo 5.

ID de Riesgo: 6	Fecha: 6 / 08 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	Los procesos utilizados hagan lento el sistema y se aprecie la diferencia de sincronía entre el mundo virtual y el real.		
Contexto:	Debido a la cantidad de información que se maneja por el Kinect y los procesos utilizados para pasar el mundo real al virtual atasquen el sistema y se haga lento.		
Reducción:	Optimizar lo más posible los procesos utilizados y hacer lo menos posibles de cálculos para evitar estos posibles atascos.		
Plan de Contingencia:	Hacer el sistema distribuido, para que una maquina se encargue exclusivamente de hacer cierto tipo de cálculos.		

Tabla 4.8: Tabla de riesgo 6.

ID de Riesgo: 6A	Fecha: 6 / 08 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	Falta de sincronía en la red distribuida propuesto como plan de contingencia.		
Contexto:	Debido a los medios físicos en este caso los cables de red, para el envío de la información ya procesada, para su utilización de reprocesamiento en otro modulo.		
Reducción:	Enviar la menor cantidad de información posible, y evitar bloquear la red por mucho tiempo.		
Plan de Contingencia:	Utilizar una computadora con mayor poder de cómputo y colocar dentro de esta la mayor cantidad de procesos para así reducir el tiempo de utilización de red, si es que se sigue requiriendo.		

Tabla 4.9: Tabla de riesgo 6A.

ID de Riesgo: 7	Fecha: 27 / 09 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	No conseguir la información requerida para la elaboración de casos de usos, a la institución de salud que se le solicitó ayuda.		
Contexto:	Debido a que no tienen ninguna obligación para mostrar los procedimientos utilizados en un cirugía, o bien porque no lo desean, o no tienen tiempo		
Reducción:	Atender en tiempo y forma las citas establecidas, y de la manera más cordial para tener su confianza y apoyo en la elaboración del proyecto.		
Plan de Contingencia:	Buscar otra institución de salud, para conseguir la información requerida.		

Tabla 4.10: Tabla de riesgo 7.

ID de Riesgo: 8	Fecha: 27 / 09 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	Falta de realismo en el ambiente virtual.		
Contexto:	Los detalles en cualquier cirugía hace la diferencia para un doctor experimentado, el objetivo de este simulador es crear experiencia para poder manejar usuarios reales.		
Reducción:	Observar la mayor cantidad de detalles, para hacerlo lo más acercado a la realidad.		
Plan de Contingencia:	Establecer una base estándar de realismo de acuerdo con profesionales en el área.		

Tabla 4.11: Tabla de riesgo 8.

ID de Riesgo: 9	Fecha: 27 / 09 / 2012	Probabilidad: Medio	Impacto: Catastrófico
Descripción:	Fallo de transmisión de datos del Kinect al guante.		
Contexto:	Al ser un sistema que aún no se ha demostrado su compatibilidad, no se tiene la certeza de un correcto envío de información.		
Reducción:	Consultar e investigar los posibles métodos que se pueden utilizar para la solución al problema (En base al estado del arte).		
Plan de Contingencia:	Cambiar la forma de interacción que tendrá el usuario con el sistema.		

Tabla 4.12: Tabla de riesgo 9.

Parte IV

Diseño del Sistema

Capítulo 5

Diseño del Sistema

El objetivo general del diseño de nuestro sistema es definir una estructura que tenga un comportamiento específico y que satisfaga ciertas restricciones de costo y prestaciones. Así, el proceso de diseño es descompuesto en varias tareas de diseño independientes y especializadas. Estas tareas son distinguibles por los tipos de componentes con que tratan y los niveles de complejidad de éstos.

5.1. Diagramas de casos de uso

Los diagramas de casos de uso fueron generados a partir del análisis de requisitos funcionales de la sección 4. A continuación se muestra un diagrama general de nuestro sistema.

5.1.1. Diagrama general de casos de uso

En este diagrama sólo se muestran las interacciones que tiene el usuario (médico) con el sistema, los cuales son comandos por gestos y comandos por voz. Los comandos por voz nos permiten solicitar la instrumentación que se necesita para la operación, así como también ejecutar órdenes que otros médicos harían en una operación real. Los comandos por gestos se utilizan para interactuar con los instrumentos y el corazón del paciente, los cuales serían movimientos de las manos capturados mediante el sensor.

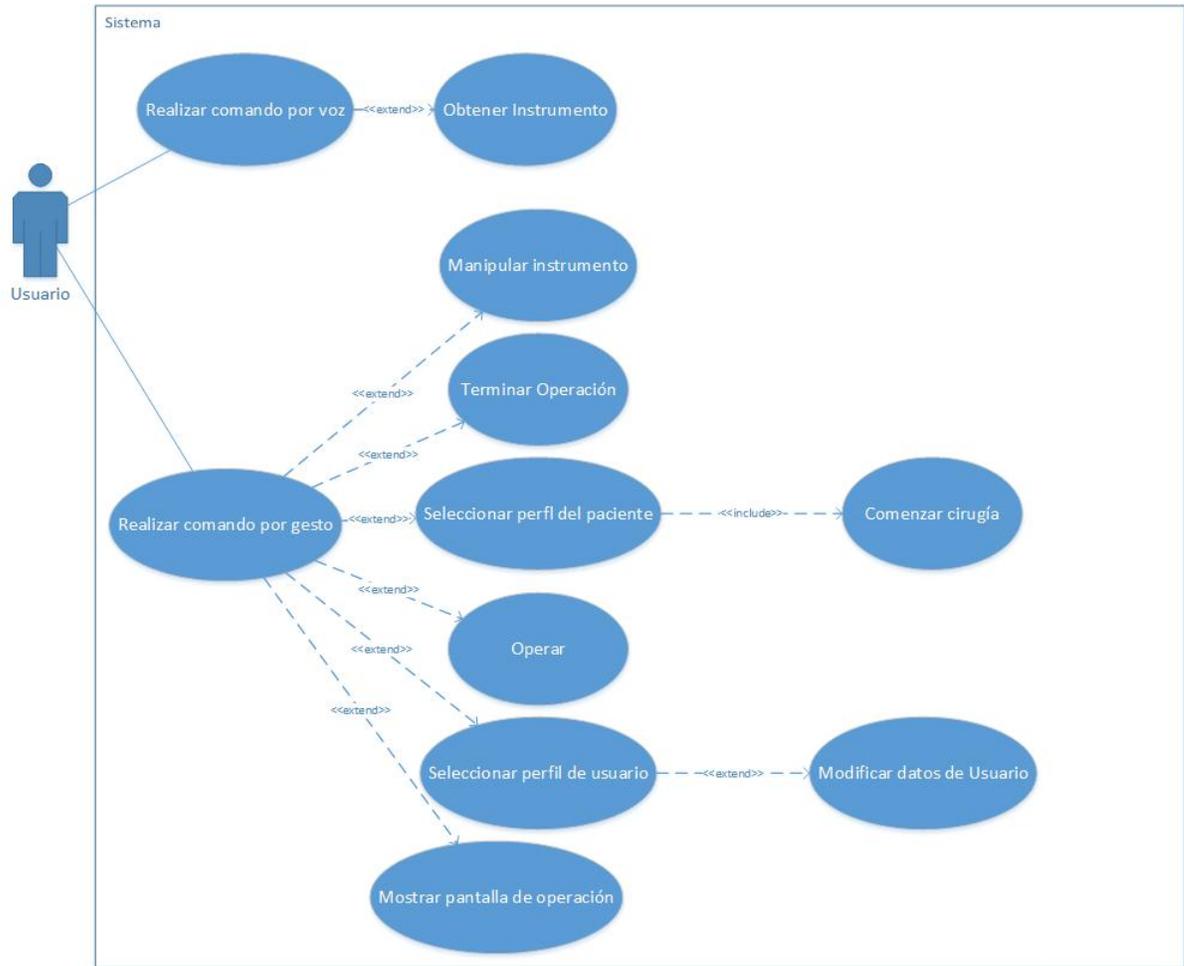


Figura 5.1: Diagrama general de casos de uso

Descripción del diagrama de casos de Uso con ID	DCU001
Nombre:	Diagrama General
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Diagrama general de casos de uso para los requisitos funcionales del sistema.
Actores:	Médico.
Precondiciones:	Iniciar sistema
Flujo normal:	T1.1 Iniciar sistema
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. Error de conexión entre componentes de sistema.

5.1.2. Caso de uso: comando por voz

Los comandos por voz son leídos desde los micrófonos de Kinect

Descripción del diagrama de casos de Uso con ID	DCU002
Nombre:	Realizar Comando por Voz
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Caso de uso para comandos por voz.
Actores:	Médico.
Precondiciones:	PRE1.-Iniciar sistema PRE2.-Iniciar cirugía
Flujo normal:	T1.1 Iniciar sistema T1.2 Seleccionar Paciente T1.3 Iniciar simulación T1.4 Leer gestos
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. Gesto no reconocido.

5.1.3. Caso de uso: obtener instrumento

Se cuenta con un catálogo limitado de instrumentos que puede utilizar el médico, siendo el nombre de estos el método de acceso para que puedan ser renderizados.

Descripción del diagrama de casos de Uso con ID	DCU003
Nombre:	Obtener instrumento
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Al reconocer un comando de voz, se debe cargar en pantalla un instrumento.
Actores:	Médico.
Precondiciones:	PRE1.-Iniciar sistema PRE2.-Iniciar cirugía PRE3.- Reconocer gesto PRE4.- El instrumento pedido se encuentra en el sistema
Flujo normal:	T1.1 Iniciar sistema T1.2 Seleccionar paciente T1.3 Iniciar simulación T1.4 Leer gestos T1.5 Reconocer gesto T1.6 Validar instrumento
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. Gesto no reconocido. 3. El instrumento solicitado no se encuentra en el sistema. 4. Error al renderizar instrumento.

5.1.4. Caso de uso: realizar comando por gesto

Los gestos son todos aquellos movimientos que realizan las manos del doctor para interactuar con el sistema.

Descripción del diagrama de casos de Uso con ID	DCU004
Nombre:	Realizar Comando Por Gesto
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Al detectar las manos del usuario con el dispositivo multisensorial, el usuario podrá realizar gestos que el sistema debe reconocer.
Actores:	Médico.
Precondiciones:	PRE1.-Iniciar sistema PRE2.-Reconocer las manos del usuario
Flujo normal:	T1.1 Iniciar sistema T1.2 Reconocer las manos del usuario
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario.

5.1.5. Caso de uso: manipular instrumento

Los instrumentos se podrán manipular directamente con las manos del médico.

Descripción del diagrama de casos de Uso con ID	DCU005
Nombre:	Manipular Instrumento
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Cuando el usuario tiene en mano un instrumento, puede moverse para manipularlo.
Actores:	Médico.
Precondiciones:	PRE1.-Reconocer las manos del usuario PRE2.- Iniciar cirugía PRE3.-Obtener instrumento
Flujo normal:	T1.1 Reconocer las manos del usuario T1.2 Reconocer gesto de manipulación de instrumento Realizar actividad pertinente.
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. Se realiza un gesto con el instrumento que no tenga un controlador.

5.1.6. Caso de uso: terminar operación

El fin de la operación se realizará únicamente cuando el médico haya terminado una operación.

Descripción del diagrama de casos de Uso con ID	DCU006
Nombre:	Terminar Operación
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Al terminar de estabilizar y suturar al paciente, el médico realiza un gesto para indicar que ha concluido el procedimiento.
Actores:	Médico.
Precondiciones:	PRE1.-Reconocer las manos del usuario PRE2.- Iniciar cirugía PRE3.-Estabilizar al paciente PRE4.-Soturar al paciente PRE5.- Reconocer comando de finalizar cirugía
Flujo normal:	T1.1 Reconocer las manos del usuario T1.2 Realizar operaciones quirúrgicas T1.3 Estabilizar signos vitales del paciente. T1.2 Realizar gesto de fin.
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. Se realiza un gesto con el instrumento que no tenga un controlador.

5.1.7. Caso de uso: seleccionar perfil paciente

Para poder comenzar una operación, se debe seleccionar un perfil prediseñado de un caso de paciente.

Descripción del diagrama de casos de Uso con ID	DCU007
Nombre:	Seleccionar Perfil Paciente
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Después de seleccionar un perfil para el médico, el usuario deberá seleccionar un perfil para el paciente que desea operar. Este perfil describirá varias características del paciente.
Actores:	Médico.
Precondiciones:	PRE1.- Seleccionar Perfil Médico
Flujo normal:	T1.1 Reconocer las manos del usuario T1.2 Seleccionar Perfil Médico T1.3 Escoger un perfil de paciente que sea del interés del médico
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario.

5.1.8. Caso de uso: comenzar cirugía

Inmediatamente después de seleccionar un perfil de paciente, se dará inicio con la cirugía desde un punto previamente definido

Descripción del diagrama de casos de Uso con ID	DCU008
Nombre:	Comenzar Cirugía
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Al seleccionar un perfil de paciente, este perfil se cargará al módulo correspondiente y dará inicio con la cirugía. De esta forma se habilitarán otras funciones del sistema.
Actores:	Médico.
Precondiciones:	PRE1.- Reconocer manos PRE2.- Seleccionar perfil del paciente PRE3.- Renderizar cirugía
Flujo normal:	T1.1 Seleccionar perfil del paciente T1.2 Renderizar al paciente
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. Error al cargar el paciente. 4. Error de renderizado.

5.1.9. Caso de uso: operar

Mientras se encuentra en cirugía, se pueden tocar órganos con o sin instrumentos.

Descripción del diagrama de casos de Uso con ID	DCU009
Nombre:	Operar
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Operar involucra el contacto del médico con el paciente en el mundo virtual.
Actores:	Médico.
Precondiciones:	PRE1.- Reconocer manos PRE2.- Comenzar Cirugía
Flujo normal:	T1.1 Reconocer manos T1.2 Detectar colisiones entre las manos y el paciente, los instrumentos y el paciente o los instrumentos y las manos.
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. Perdida virtual de las manos físicas. 4. Procesos zombies en procesos paralelos.

5.1.10. Caso de uso: seleccionar perfil usuario

Para iniciar las cirugías, el médico debe seleccionar su perfil.

Descripción del diagrama de casos de Uso con ID	DCU010
Nombre:	Seleccionar Perfil Usuario
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	El médico debe seleccionar su perfil para poder guardar registros de actividades
Actores:	Médico.
Precondiciones:	PRE1.- Reconocer manos PRE2.- Contar con un perfil en el sistema
Flujo normal:	T1.1 Reconocer manos T1.2 Seleccionar perfil en el sistema.
Flujo alternativo:	Ninguno.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. No contar con un perfil en el sistema.

5.1.11. Caso de uso: modificar datos usuario

Para hacer cambios o crear perfiles se accede a una pantalla adecuada para esto.

Descripción del diagrama de casos de Uso con ID	DCU011
Nombre:	Modificar datos Usuario
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	El médico puede modificar su perfil o crearlo en caso de no existir.
Actores:	Médico.
Precondiciones:	PRE1.- Reconocer manos PRE2.- Seleccionar perfil o crear perfil.
Flujo normal:	T1.1 Reconocer manos T1.2 Escoger perfil de usuario. T1.3 Editar perfil seleccionado
Flujo alternativo:	TA1.1 Reconocer manos TA1.2 Seleccionar Crear perfil.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario. 3. No contar con un perfil en el sistema.

5.1.12. Caso de uso: mostrar pantalla de operación

Mientras se encuentra en cirugía, el médico puede cambiar la pantalla para visualizar los signos vitales del paciente.

Descripción del diagrama de casos de Uso con ID	DCU012
Nombre:	Mostrar pantalla de operación
Autor:	Damián Sandoval Sergio Arturo y López Flores Iván
Fecha:	5-Nov-2012
Descripción:	Mientras se encuentra en operación, el médico realiza un gesto para cambiar la pantalla de la operación por una pantalla donde se muestren los signos vitales del paciente. También se incluye el uso de ese mismo gesto para volver al monitor de operación desde el monitor médico
Actores:	Médico.
Precondiciones:	PRE1.- Reconocer manos PRE2.- Estar en operación. PRE3.- Reconocer gesto
Flujo normal:	T1.1 Reconocer manos T1.2 Encontrarse en Operación. T1.3 Realizar un gesto pre ajustado T1.4.-Validar el gesto T1.5.-Renderizar el monitor de paciente.
Flujo alternativo:	TA1.1 Reconocer manos TA1.2 Encontrarse en Operación. TA1.3 Realizar un gesto pre ajustado TA1.4.-Validar el gesto TA1.5.-Renderizar la operación.
Excepciones:	<ol style="list-style-type: none"> 1. Falta de recursos en el sistema. 2. No reconocer las manos del usuario.

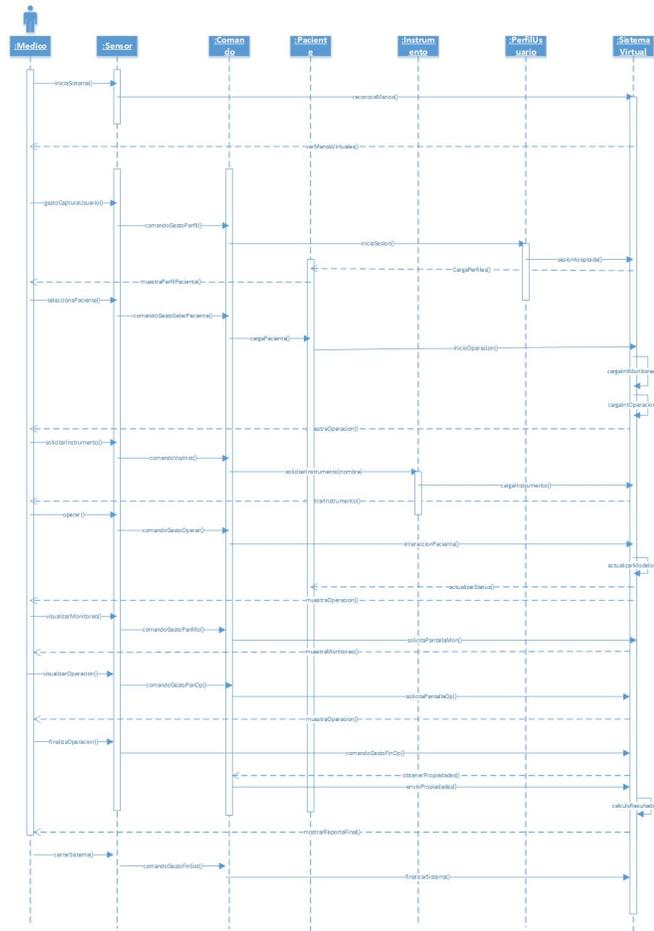


Figura 5.2: Diagrama de secuencias: diagrama de secuencia de la Representación general del sistema

5.2. Diagramas de secuencia

5.2.1. Diagrama de secuencia 1: representación general del sistema

En el diagrama 5.2 encontramos un caso específico posible que se puede presentar en nuestro sistema, donde se realiza el reconocimiento de manos y el inicio de sesión del usuario para que posteriormente se seleccione un perfil de un paciente que se encuentre dentro de los predeterminados en el sistema. Es entonces cuando se realiza el inicio de la simulación, donde podemos apreciar que el usuario recibe las interfaces correspondientes. Posteriormente, el usuario solicita un instrumento de operación al sistema mediante un comando de voz y le es enviado el modelo correspondiente para su manipulación. Después el usuario realiza un movimiento de gestos con las manos para continuar con la operación, solicita la pantalla de monitoreo, después cambia a la pantalla de operación y decide concluir la operación. El sistema entonces, elabora un reporte final de la operación y finalmente se realiza el cierre del sistema.

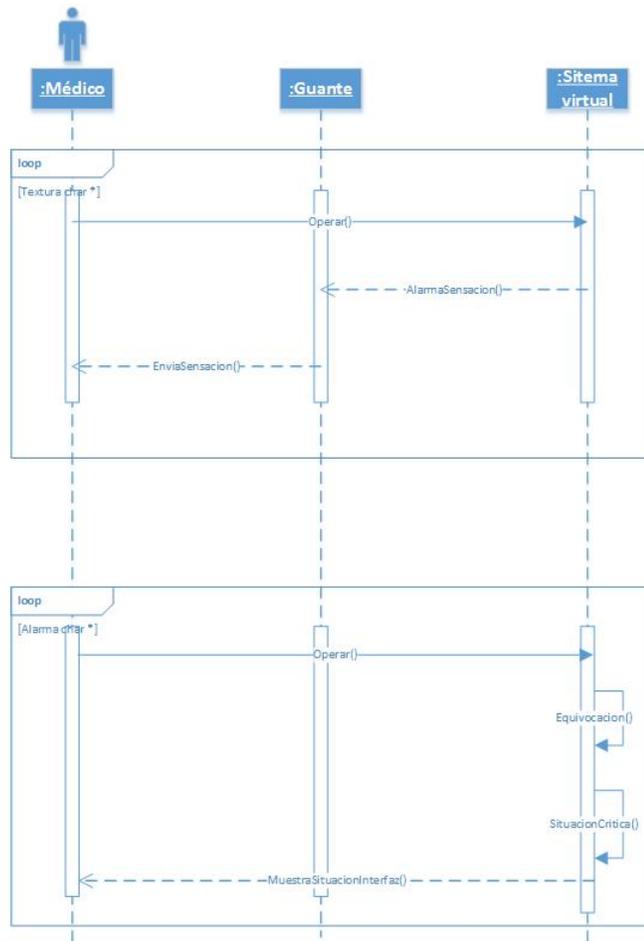


Figura 5.3: Diagrama de secuencias: envío de alarmas y sensaciones físicas al usuario

5.2.2. Diagrama de secuencia 2: envío de alarmas y sensaciones físicas al usuario

En diagrama 5.3 podemos apreciar que estos dos casos se presentan en todo momento durante el desarrollo de la simulación de la operación en el sistema. El primero de ellos es enviar las sensaciones físicas, donde se realiza la operación, se envía la alarma hacia el guante y posteriormente el guante presenta al usuario la sensación correspondiente. El segundo de ellos presenta un envío de un procedimiento de operación equivocado por parte del usuario y el sistema detecta y presenta una situación de riesgo en la operación, donde los síntomas del paciente cambiarían y el usuario tiene que resolver esos problemas.

5.2.3. Diagrama de secuencia 3: presentación de interfaces de operación y reporte final

En el diagrama 5.4 tenemos tres casos de interacción, donde el usuario solicita la interfaz de monitoreo al sistema mediante un comando por gesto, y se presenta la pantalla de monitoreo del paciente en la operación. Lo mismo se trata para la pantalla de operación, donde podemos apreciar

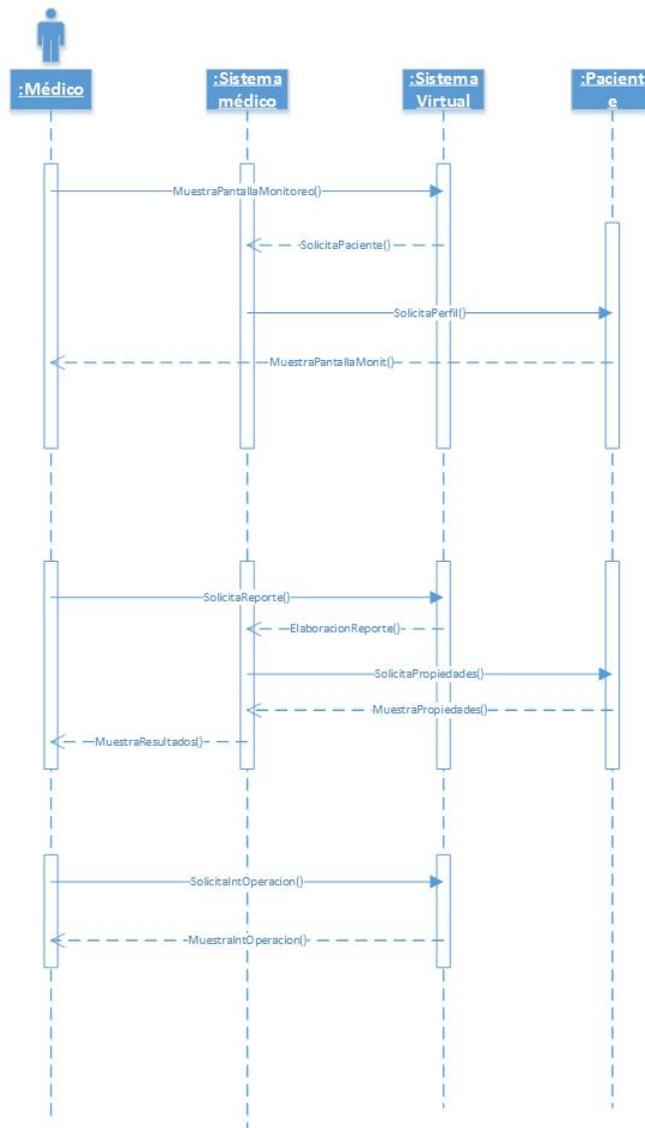


Figura 5.4: Diagrama de secuencias: presentación de interfaces de operación y reporte final

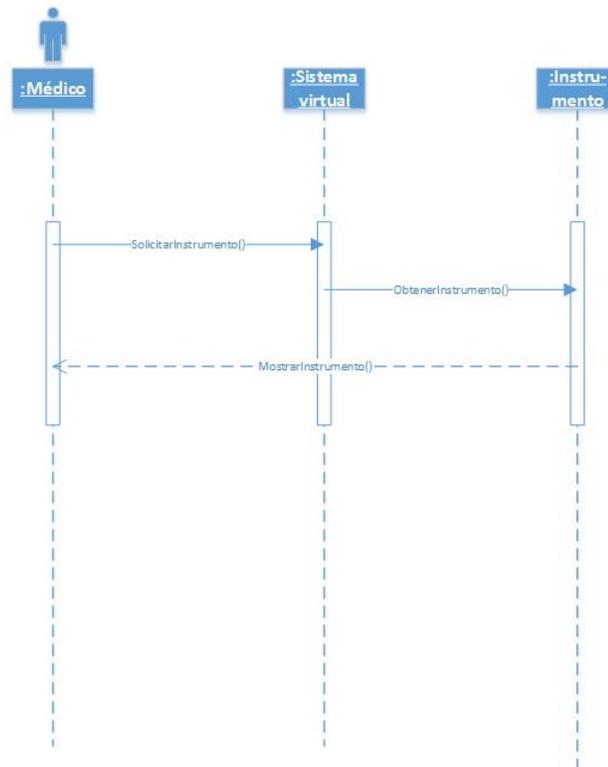


Figura 5.5: Diagrama de secuencias: solicitud de Instrumento de operación

el modelo del paciente y donde se puede ejecutar la operación virtualmente.

5.2.4. Diagrama de secuencia 4: solicitud de instrumento de operación

El diagrama 5.5 presenta cómo se obtiene un instrumento de operación en el sistema. El usuario solicita el instrumento mediante un comando de voz que es recibido por el dispositivo multisensorial y el sistema virtual determina cuál es el instrumento solicitado, mandando llamar dentro de nuestros instrumentos, el solicitado.

5.2.5. Diagrama de secuencia 5: actualizado de pantalla

El diagrama 5.6 presenta el procedimiento para actualizar las pantallas gráficamente en tiempo real, mediante un objeto que tenga funciones de renderización. Se presenta en todo momento en la simulación de la operación.

5.2.6. Diagrama de secuencia 6: interacción con el cliente

El diagrama 5.7 presenta la solicitud y elección de un paciente determinado para visualizar sus características. También podemos apreciar una solicitud de un historial médico para un paciente que ya haya sido operado y poder apreciar los resultados correspondientes.

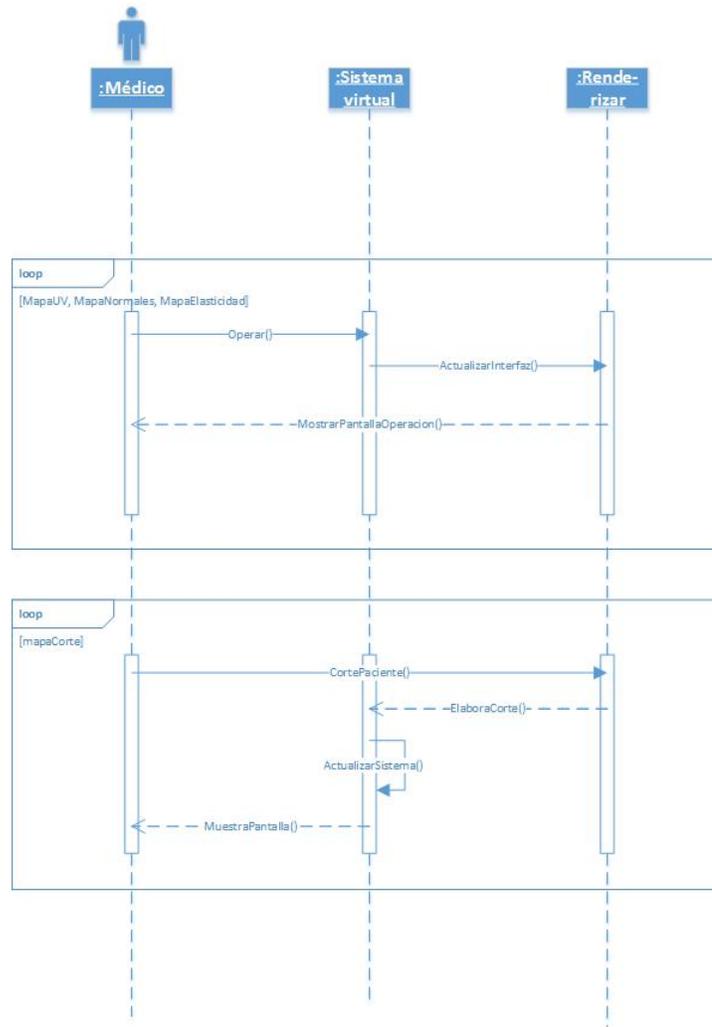


Figura 5.6: Diagrama de secuencias: actualizado de pantalla

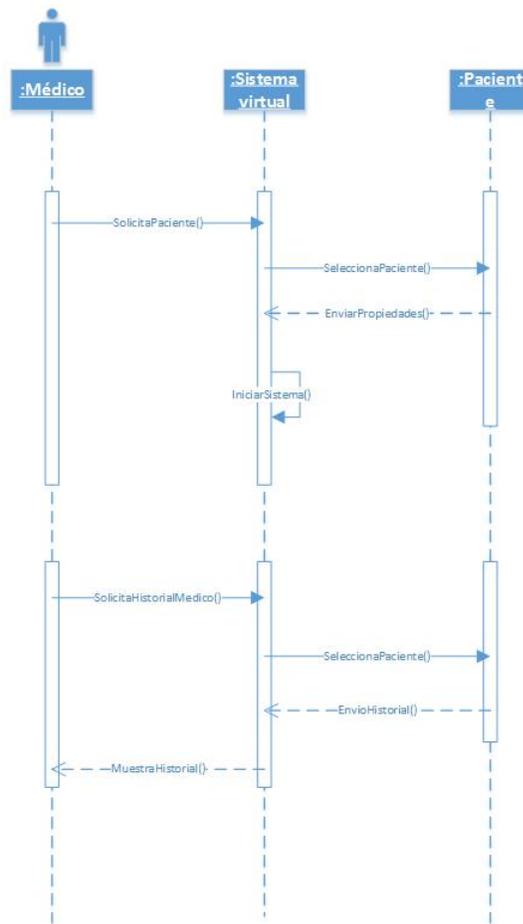


Figura 5.7: Diagrama de secuencias: interacción con el cliente

5.3. Diagrama de Clases

Los diagramas de clases representan la estructura estática del sistema.



Figura 5.8: Diagrama General de Clases

La forma en que estos módulos se relacionan están organizados para que se pueda escalar, de manera que no sólo se pueda enfocar a la regularización coronaria, o bien, no sólo para intervenciones quirúrgicas, ya que modelar al mundo real, y percibirlo, de otra forma gracias al guante, abre nuevas posibilidades de estudio.

La manera en que se separaron las clases, fue el sistema virtual, el sistema médico, la parte de conectividad con los dispositivos como son el Kinect; guante de percepción física, y la parte en que se modela la realidad en el sistema virtual, apoyándose en el Módulo Renderizado.

5.3.1. Módulo Instrumental

En este módulo se definen los tipos de instrumento que se utilizarán en la cirugía virtual, a través de una enumeración, esta necesidad de utilizar una enumeración, surge porque no toda la parte del instrumento es cortante, flexible, y/o perforante, esta enumeración ayudará a crear un mapa, definiendo sobre el modelo, que sección es cortante, otra es el mango, como si fuera una textura, de funcionalidad.

Este módulo tiene interacción con el Módulo Renderizado, para modificar las características del modelo.

5.3.2. Módulo Guante

Este módulo tiene la finalidad de interactuar con los drivers Arduino, mandando las señales pertinentes de alarma, así como los puntos en los que nuestro cirujano real debe de percibir como

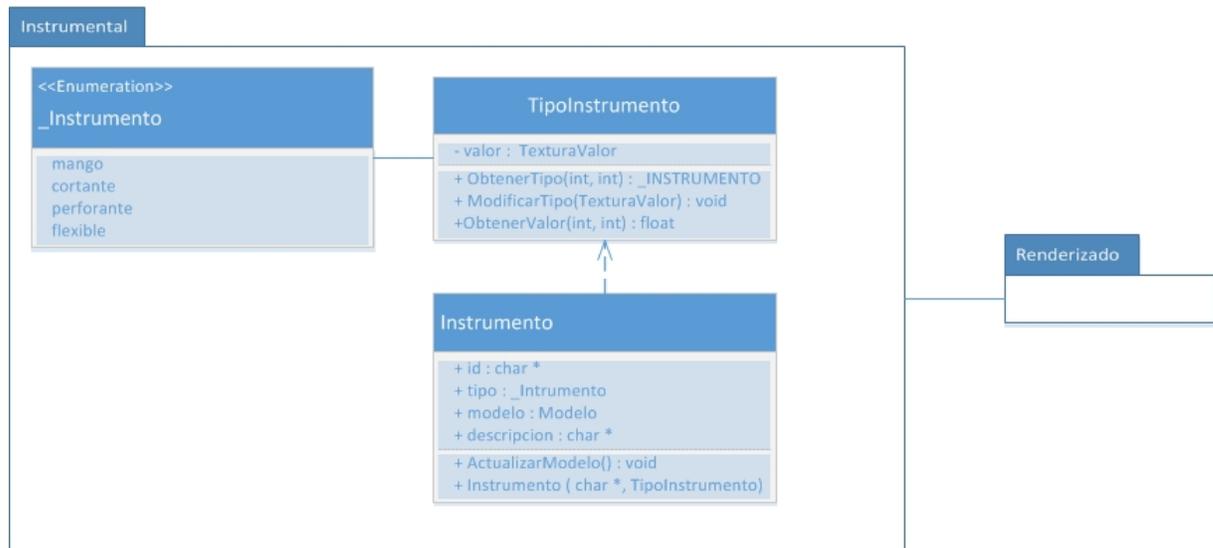


Figura 5.9: Diagrama de Módulo Instrumental

lo hace el cirujano virtual.

5.3.3. Módulo Sistema Virtual

Este módulo es el encargado de recibir, los modelos, y colocarlos y esta involucrado con el módulo de sistema virtual, guante, y el dispositivo multisensorial, y después del módulo de sistema Medico, este módulo es la piedra angular del proyecto, ya que este módulo de manera independiente podría utilizarse, como núcleo para proyectos que requieran de un dispositivo para sensaciones físicas, y reciban modelos, en tiempo real.

5.3.4. Módulo Sistema Medico

Este módulo es el encargado del monitoreo del paciente real, de revisar los gestos, que hace el medico, ya sean de voz o movimiento, así mismo, tiene conexión con el módulo de Paciente, Renderizado, y el dispositivo multisensorial, para realizar sus tareas.

5.3.5. Módulo Renderizado

En este módulo se procesan los modelos, para hacer estos los mas apegados a la realidad, apoyándose en mapas de valor, como son los mapas de elasticidad, flexibilidad, corte, asignando estas características a los triángulos, del mesh.

Este módulo tiene el alcance de expandirse e incluir otras características del mundo físico, para tener una mejor inmersión en la simulación.

5.3.6. Módulo Paciente

Este módulo es el encargado de los datos del modelo de la simulación, los signos del paciente, las complicaciones que podría o no tener durante la simulación, así mismo este se encarga de las

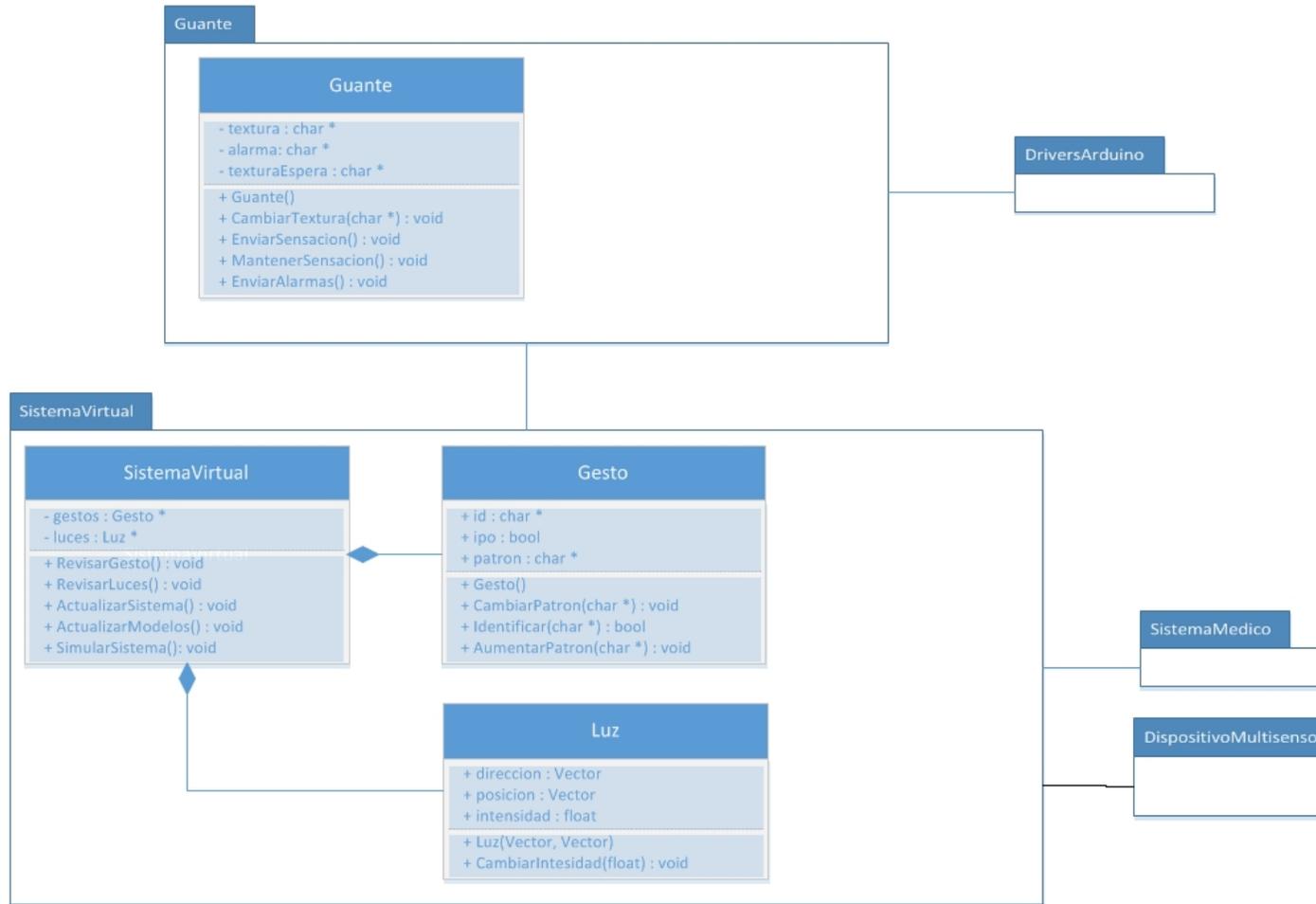


Figura 5.10: Diagrama del Módulo Guante y Módulo Sistema Virtual

propiedades del paciente, sangre, presión, oxígeno, datos que serán mostrados en los monitores virtuales

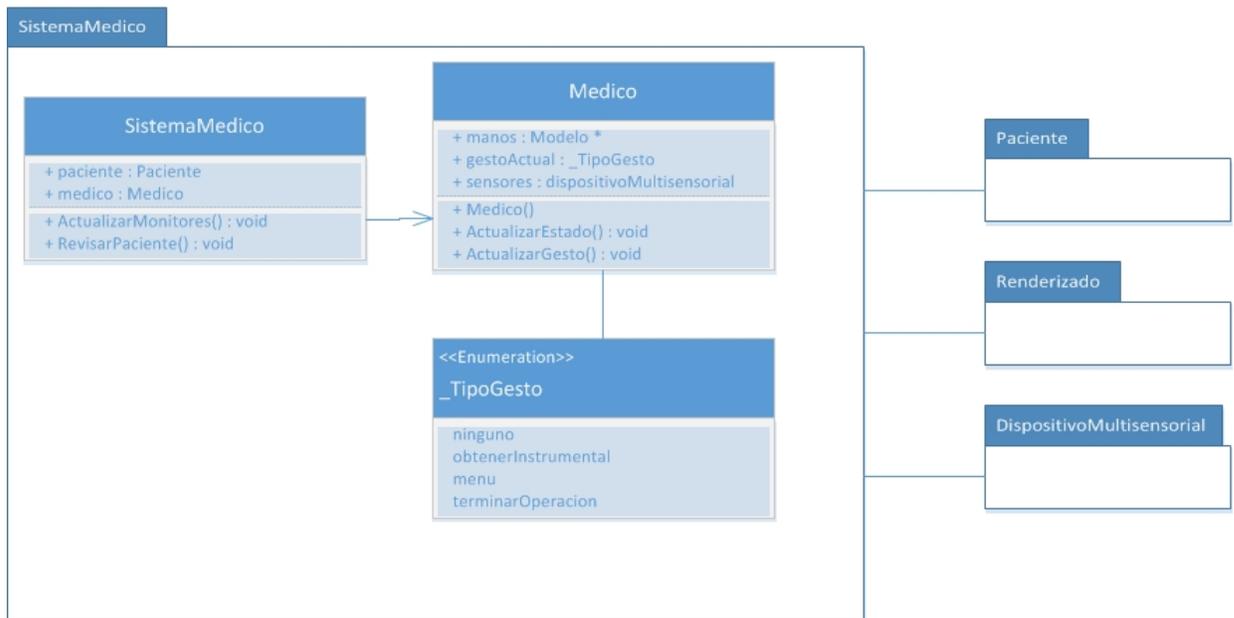


Figura 5.11: Diagrama de Módulo Sistema Médico

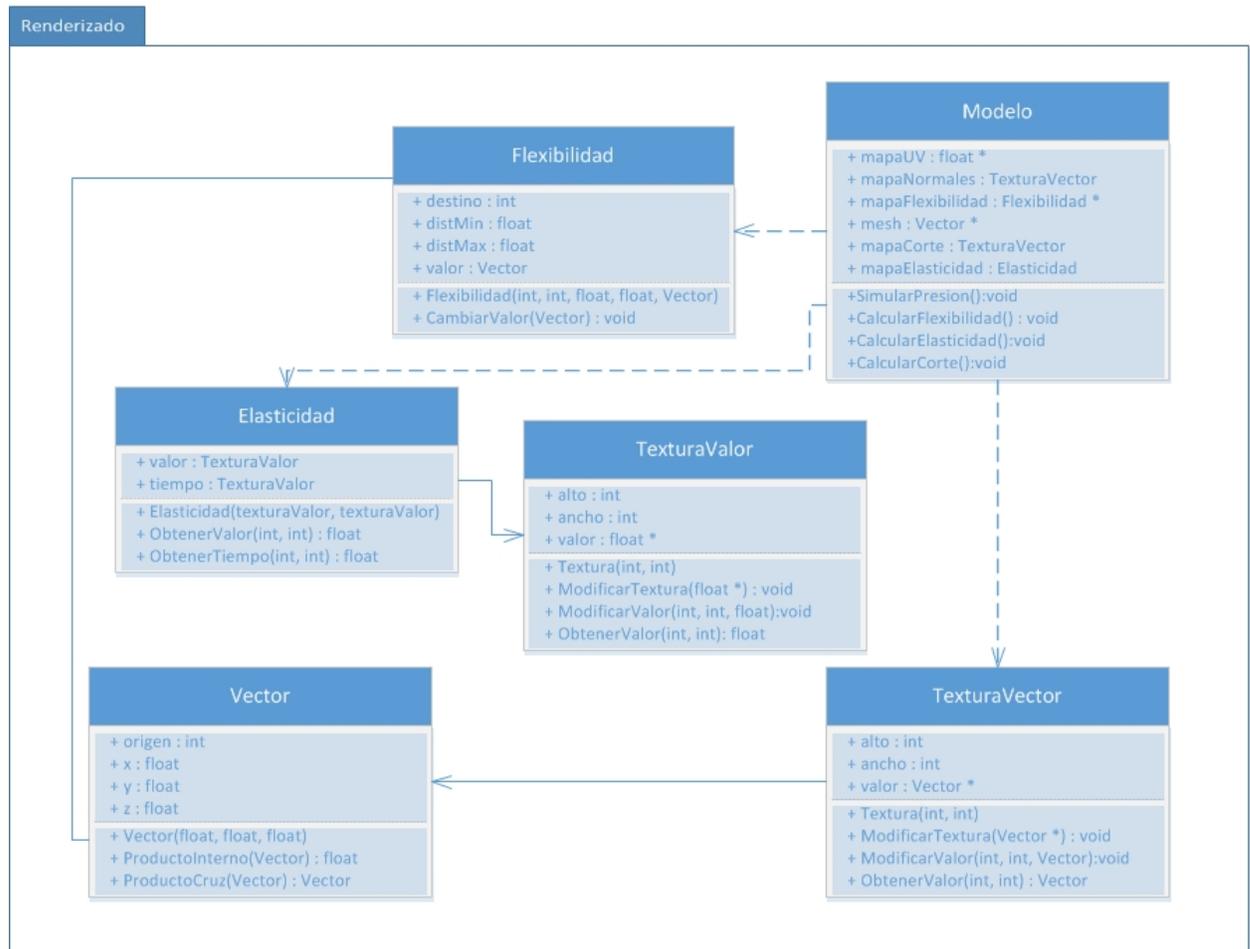


Figura 5.12: Diagrama del Módulo Renderizado

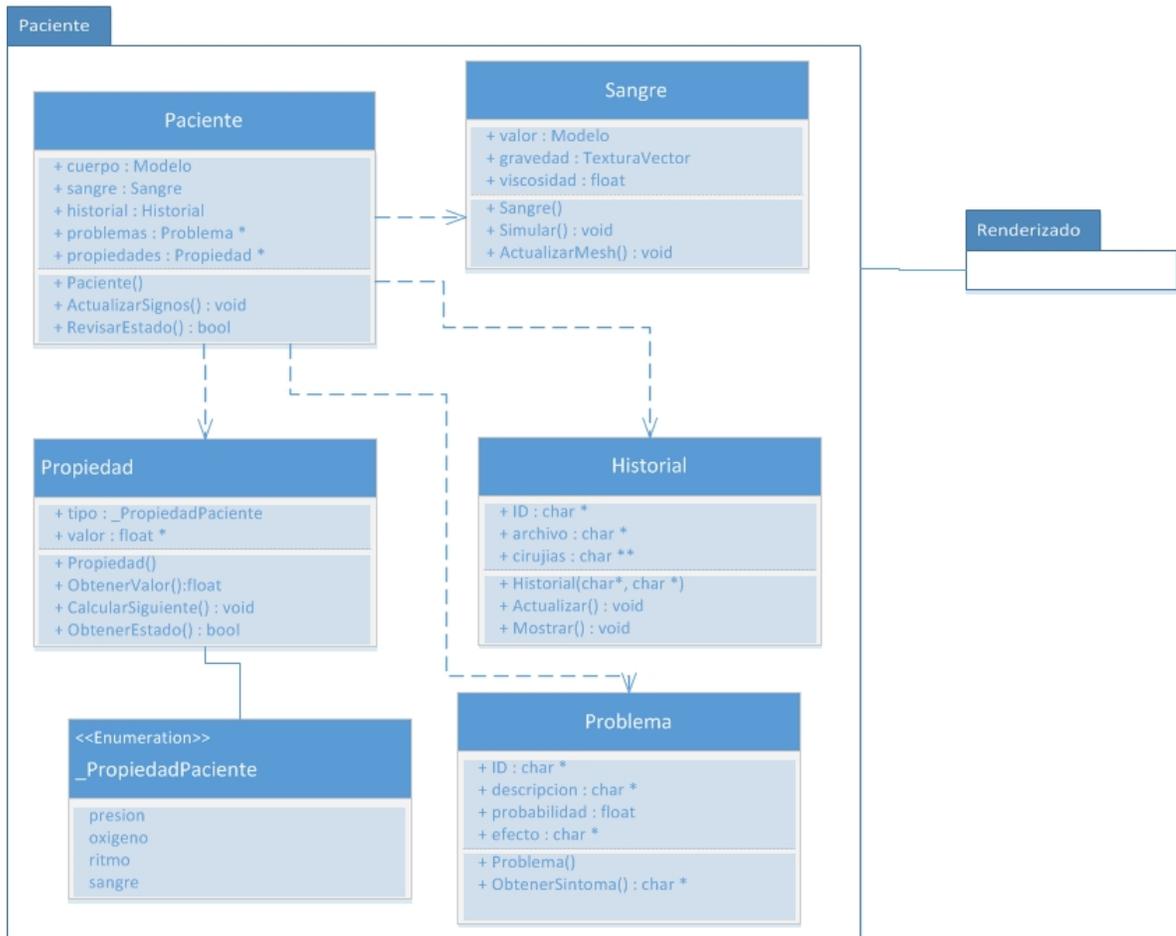


Figura 5.13: Diagrama del Módulo Paciente

Parte V

Desarrollo y pruebas del sistema

Capítulo 6

Desarrollo

En este capítulo se aborda la implementación del sistema. Esta fue desarrollada principalmente entre los meses de Febrero y Abril de 2013. Se muestra el desarrollo de cada módulo del sistema por separado y su integración en el ambiente de desarrollo Unity 4.1.2 f1, que utiliza JavaScript y Mono-C# y el IDE Mono para el scripting y las dinámicas del sistema. Así mismo presentamos el desarrollo de las librerías dinámicas de captura, procesamiento y optimización de datos que comunican Kinect con el resto del sistema. También incluimos el desarrollo y las pruebas del guante de retroalimentación háptica.

Los elementos que conforman esta sección es:

- Modelos
- Módulos Hápticos
- Módulo de comandos de voz
- Simulación de cirugía

6.1. Modelos Tridimensionales

6.1.1. Modelos cardiovasculares

Los modelos que se utilizarán, serán los encontrados en una base de datos japonesa, los cuales son parte de un proyecto para facilitar a los doctores crear modelos ejemplo en tercera dimensión de órganos internos. Cabe mencionar que los modelos son anatómicamente correctos y se pueden utilizar bajo la licencia Creative Commons Attribution-Share Alike 2.1 Japan.

Como podemos observar en la imagen, la página ofrece un sin fin de modelos. Incluye huesos, músculos, órganos, venas y arterias. Para este trabajo terminal hemos decidido únicamente usar los modelos que pertenecen al sistema cardiovascular principal.

En la base de datos no se proporcionan ningún tipo de texturas reales para los modelos, es por eso que utilizamos texturas básicas de color en ellos ya que no se encontraron ningún tipo de texturas que se pudieran usar de manera gratuita.

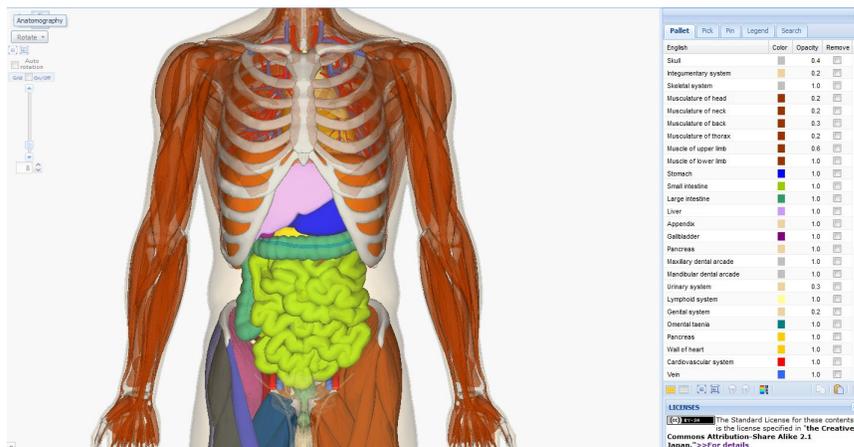


Figura 6.1: Modelos ofrecidos por la página Life Science DB

6.1.2. Modelos quirúrgicos

Para la interacción del usuario con el sistema, se necesitaron modelar 23 instrumentos quirúrgicos especializados en la operación de revascularización coronaria. En primer lugar decidimos realizar una búsqueda en la web para encontrar los modelos y agilizar el trabajo, pero por ser especializados, sólo unos cuantos fueron localizados aunque no contaban con la calidad necesaria para el proyecto. Después de cubrir la curva de aprendizaje para diseñar modelos, texturas e incluso animación nos dimos a la tarea de elaborarlos en Blender versión 2.6. El nombre de cada uno de ellos es:

- Bisturí,
- Fórceps,
- 2 Pinzas,
- 5 Tijeras Diethrich,
- Porta agujas,
- 7 cánulas para diferentes venas y arterias,
- 5 cabezas de bisturí,

Éstos son llamados a partir del módulo de comandos de voz, y el usuario puede interactuar con ellos cuando son llamados.

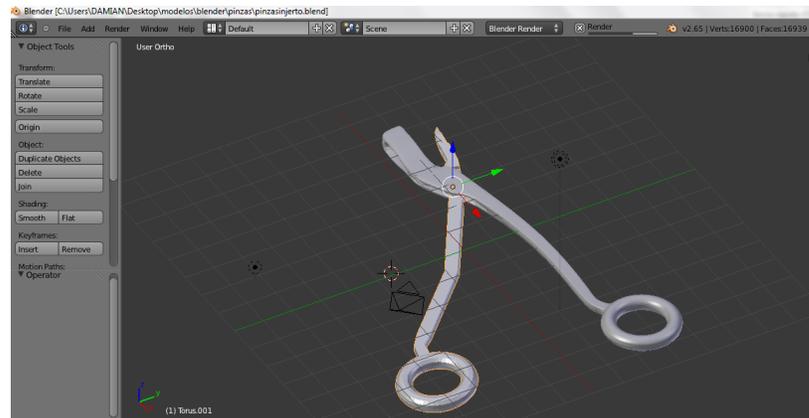


Figura 6.2: Modelado de pinzas para injertos

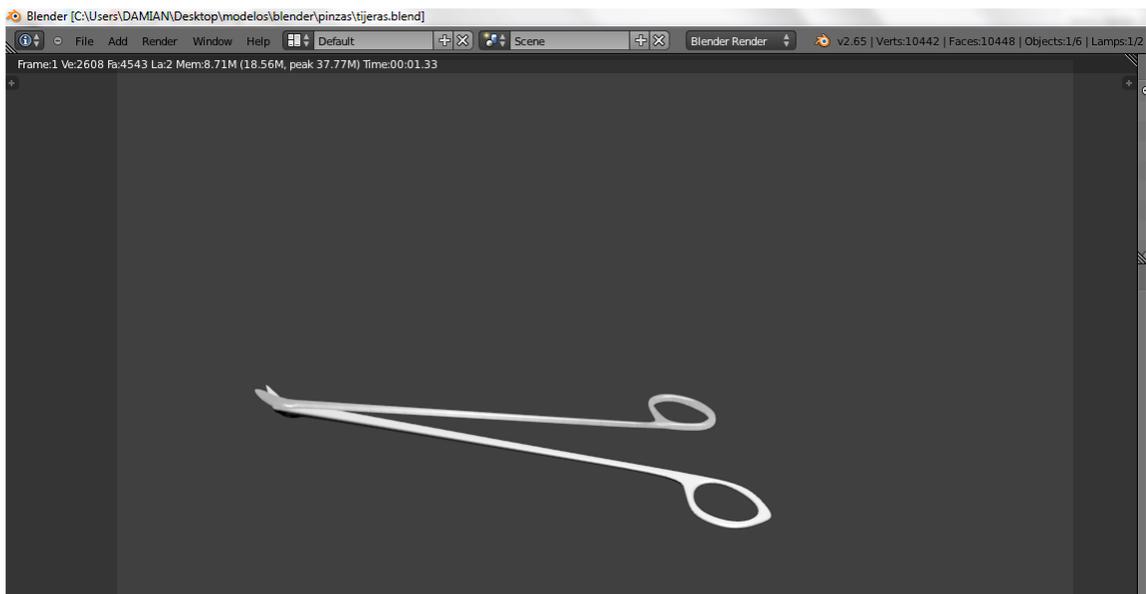


Figura 6.3: Tijeras de Diethrich

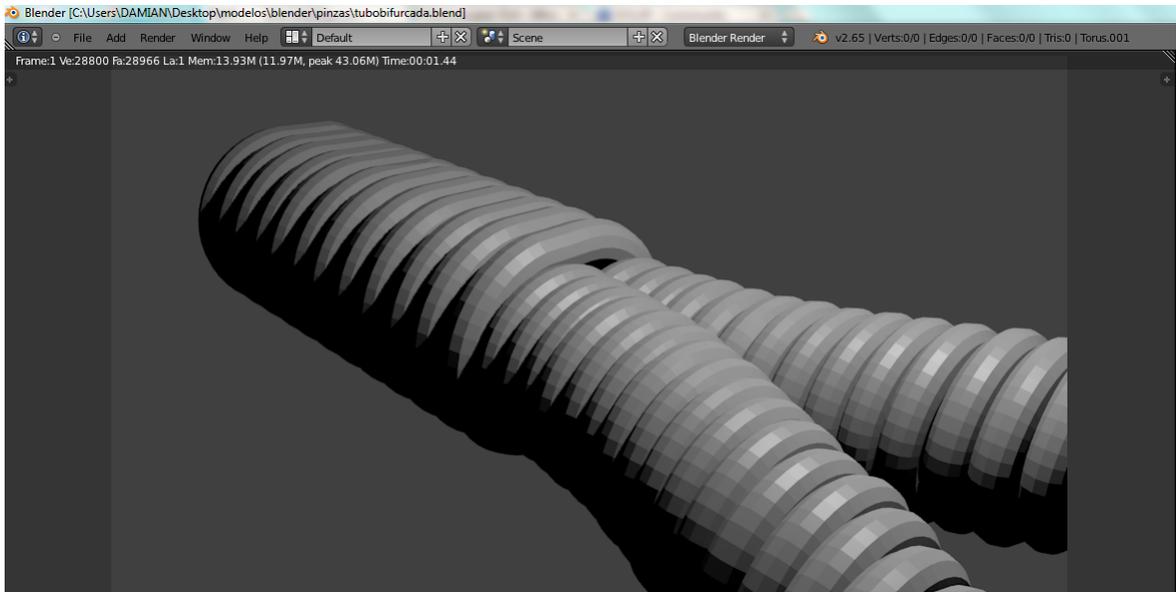


Figura 6.4: Tubo bifurcado para la bomba extracorpórea

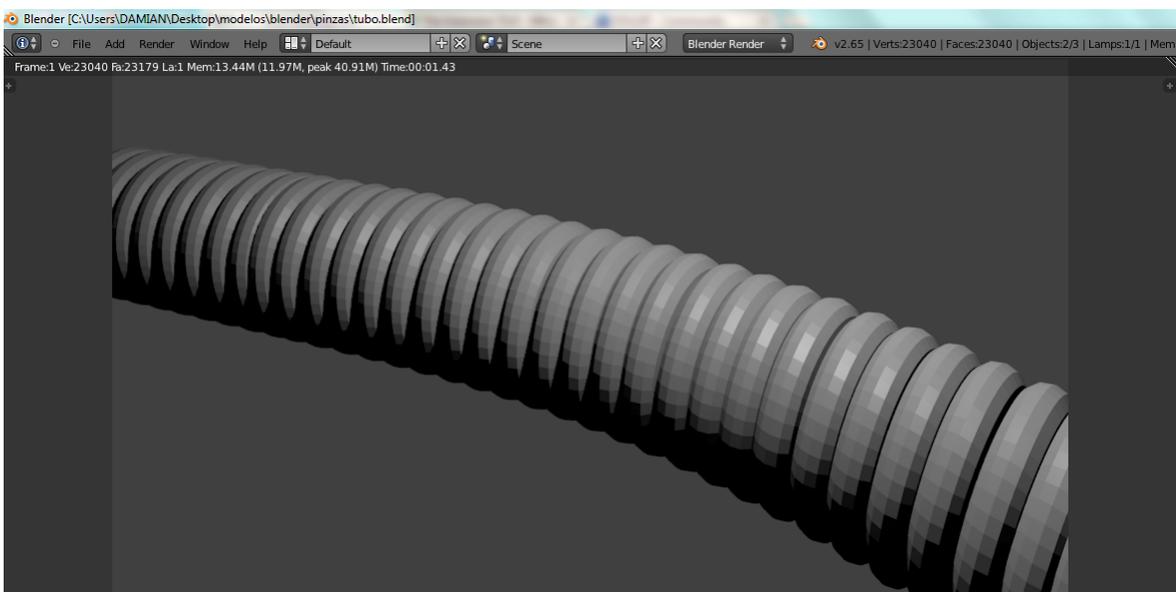


Figura 6.5: Tubo para la bomba extracorpórea

6.2. Háptica

6.2.1. Material

- Arduino UNO.
- Motores vibratorios 10x2,0mm.
- Alambre
- Eliminador 5V – 2A
- Circuito integrado ULN2003A
- Guante de material cómodo

6.2.2. Actuadores

Un actuador es cualquier mecanismo hidráulico, neumático o eléctrico que ejecuta una acción como respuesta a una señal. Como actuador, nosotros utilizamos motores vibratorios sin eje.

Estos actuadores funcionan alternando la dirección del campo a través de las bobinas mediante un circuito de conmutación, y esto interactúa con los pares de polos N-S que se producen en el imán de Neodimio. El disco del motor gira debido a la masa descentrada que incorpora, haciendo que el motor vibre.

Este tipo de motores resultaron ideales para las tecnologías hápticas e incluso son recomendados por [1] para las simulaciones en cirugías, pues su tamaño es sumamente poco invasivo y resulta poco estorboso para los cirujanos.

Otras aplicaciones de este tipo de motores son: Teléfonos celulares, scanner RFID, instrumentos industriales, aplicaciones médicas, etc.

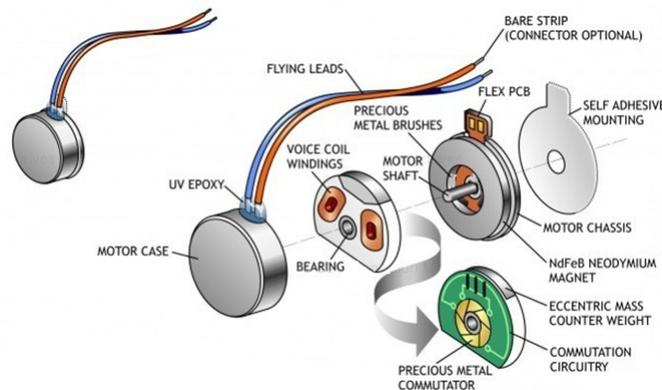


Figura 6.6: Shaftless-vibration-motor

6.2.3. Características

Tamaño	10 mm de diámetro, 2 mm de alto
Peso	0.8g
Voltaje de operación recomendado	2.5 3.5 V
Amplitud de vibración a 3V	0.75g
Velocidad a 3V	14500 RPM
Consumo de corriente	60 mA (80mA MAX)
Picos de corriente	120 mA MAX
Resistencia del motor	29 +- 6 Ω

6.2.4. Interfaz

La comunicación Computadora–Arduino se realiza en dos partes:

1. Multiplexación en la PC.
2. Demultiplexado en la placa

La multiplexación de los datos comienza cuando el sistema virtual detecta una colisión que involucra alguna mano del cirujano. Se calculan las zonas donde debería existir algún tipo de presión sobre la mano y se manda a llamar a la función Enviar Sensación (Sección 5.10). Dependiendo de la cantidad de triángulos que intersectan con el elemento que produce la colisión, se calcula un número de 0 a 55, donde 55 es el máximo impacto soportado por el guante y 0 es Sin Contacto.

Utilizamos una función de corrimiento de bits para aprovechar los 32 bits de un entero, usando el primer conjunto de 6 bits para definir el nivel de sensación háptica que deberá tener el primer dedo, el segundo conjunto de 6 bits para el segundo dedo, y así sucesivamente.

Una vez Arduino recibe el entero, descompone en los cinco valores correspondientes mediante el corrimiento de bits y la operación AND con una máscara de valor 63; posteriormente se mandan estos valores a los motores del guante mediante la técnica de PWM el cual para este caso en particular sólo acepta valor en un rango de 40 – 180 debido a que un valor menor no lograría hacer funcionar los motores y un valor mayor podría disminuir la vida útil de los motores.

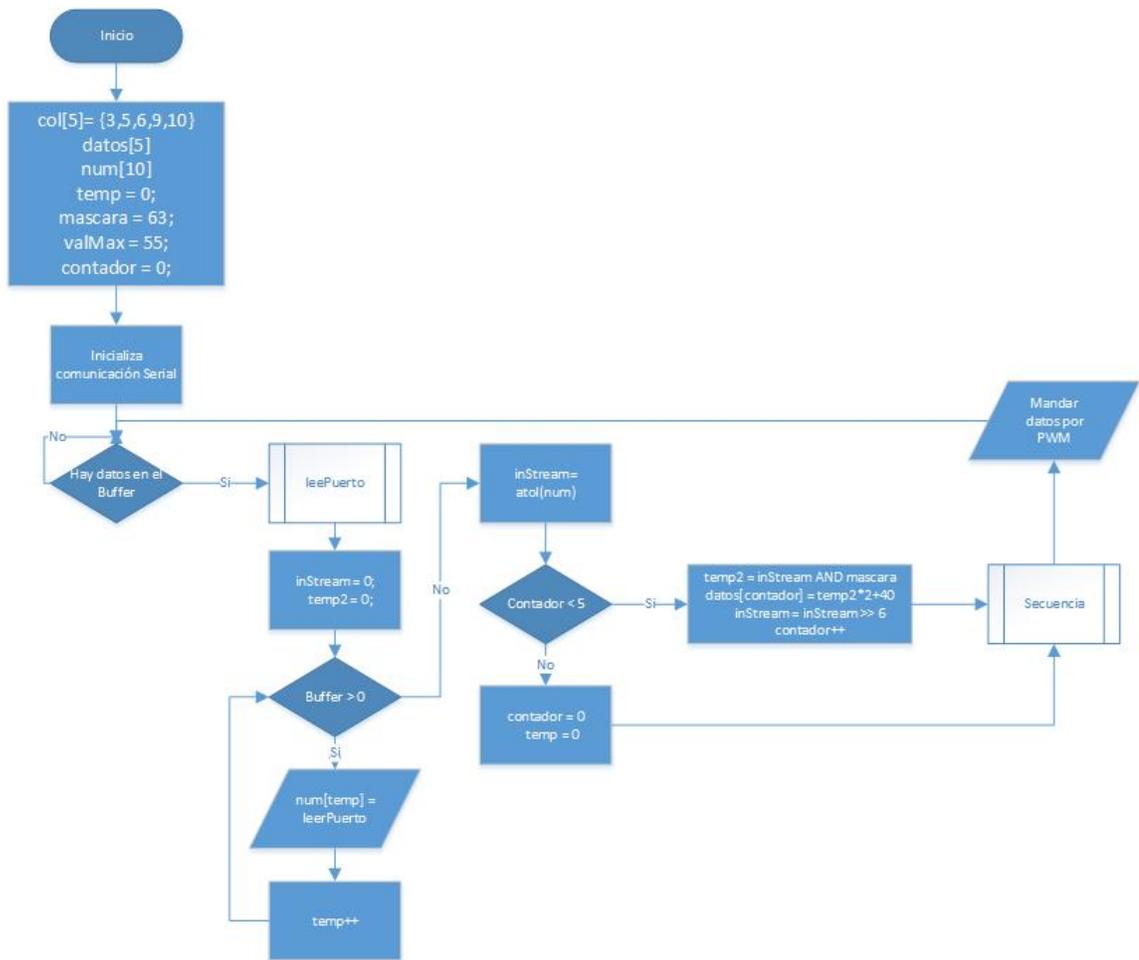


Figura 6.7: Algoritmo de comunicación entre pc/guante

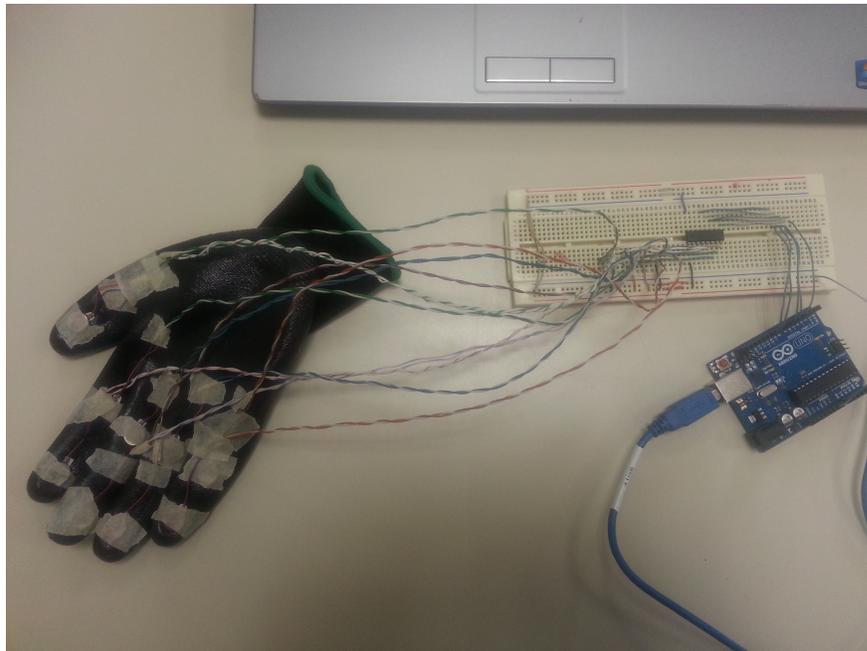


Figura 6.8: Prototipo final

6.3. Comando de Voz

6.3.1. Librerías

Para la detección de la voz del usuario y su posterior interacción, se utiliza una librería llamada *System.Speech*, la cual se instala junto con las demás librerías del lenguaje de programación *C#*. La utilidad de esta librería es que puede ser conectada con cualquier micrófono que el usuario posea, pero para nuestros intereses, utilizaremos el micrófono que se encuentra integrado en el Kinect.

El mismo SDK de Kinect posee una librería propia de reconocimiento de voz llamada *Microsoft.Speech*, pero desafortunadamente no es compatible con el motor gráfico que utilizamos para el desarrollo del sistema, de hecho, ninguna librería que utilice reconocimiento de voz es aceptada.

Ya que no es permitida en Unity, desarrollamos nuestra librería dinámica, la cual utiliza *System.Speech* y fue desarrollada en *C#* con el nombre de **SpeechDLL** para la identificación del micrófono y de las palabras que reconoce.

Mientras que **SpeechDLL** es llamada por una tercer librería, desarrollada en *C++*, llamada *SpeechRecognition.Checked* que nos permite finalmente la integración del módulo con el sistema.

El módulo presenta una fidelidad de reconocimiento de las palabras del setenta al ochenta por ciento.

6.4. Simulación de la Cirugía

6.4.1. Guión de Cirugía

Desarrollo del script

Para darle seguimiento a lo que el usuario estaba haciendo durante la cirugía lo que se propuso fue definir en un archivo los pasos que se tenían que completar durante la simulación para finalizarla de manera correcta. Estos pasos fueron definidos a partir de la investigación que se llevó a cabo acerca de la cirugía de revascularización coronaria y posteriormente, se tradujeron a acciones que el usuario puede hacer en Unity. Esto dio como resultado final, una máquina de estados.

Seguimiento del script

Ejemplo:

Paso de la cirugía: Realizar un corte en la vena cava superior.

Acciones en la simulación que se deben llevar a cabo:

1. Llamar o tener activa la herramienta Bisturí.
2. Realizar corte.

Manera en la que se pueden comprobar esas acciones por medio de código:

- Se reconoció la palabra Bisturí en el módulo de comandos de voz.
- Se realizó una colisión de las manos del cirujano con el modelo del bisturí
- Se realizó una colisión del Bisturí con el modelo de la vena cava superior.

6.4.2. Puntuación de la cirugía

La manera en la que se lleva a cabo la puntuación total es la siguiente:

- Se comienza con una puntuación de 100.
- Se descuentan puntos por cada paso que se omitió en la cirugía
- Se descuentan pocos puntos por cada colisión de las manos del cirujano con una parte del corazón que no debería de estar tocando.
- Se descuentan muchos puntos por cada corte del bisturí con una parte del corazón que no debería de cortarse.
- La cirugía termina si se llega a 0 en la puntuación

Script

1. Conectar a CEC.
 - a) Clampeado de Vena Cava Inferior.
 - 1) Sutura.
 - a'* Pedir Porta Agujas.
 - b'* Agregar puntas de hilo a "Tubo de extracción extracorpórea". Colocar una sutura en jareta sobre la aorta ascendente.
 - c'* Retirar Porta Agujas.
 - 2) Incisión.
 - a'* Pedir Bisturí.
 - b'* Localizar área para realizar incisión.
 - c'* Clampear con el dedo para controlar el sangrado.
 - d'* Realizar incisión. Incidir en la aorta y dilatar la incisión con un dilatador aórtico o con una pinza de Schmidt.
 - e'* Retirar Bisturí.
 - 3) Entubación
 - a'* Pedir tubo con sutura.
 - b'* Acoplar tubo a "Vena Cava Inferior".
 - 4) Sellar sutura.
 - a'* Pedir Porta Agujas Sin Aguja.
 - b'* Suturar de manera sencilla o solo amarrar.
 - c'* Retirar Porta Agujas.
 - 5) Clamp.
 - a'* Pedir Pinzas.
 - b'* Clampear debajo de la incisión.
 - b) Clampeado de Aurícula Derecha.
 - 1) Sutura.
 - a'* Pedir Porta Agujas.
 - b'* Agregar puntas de hilo a "Tubo de extracción extracorpórea".
 - c'* Retirar Porta Agujas.
 - 2) Incisión.
 - a'* Pedir Bisturí.
 - b'* Buscar un buen lugar para cortar.
 - c'* Clampear con el dedo.
 - d'* Realizar incisión.
 - e'* Retirar Bisturí.
 - 3) Entubación.
 - a'* Pedir tubo con sutura.
 - b'* Acoplar tubo a "Aurícula Derecha".

- 4) Sellar sutura.
 - a' Pedir Porta Agujas Sin Aguja.
 - b' Amarrar sutura.
 - c' Retirar Porta Agujas.
 - 5) Clamp.
 - a' Pedir Pinzas.
 - b' Clampear debajo de la incisión.
2. Pedir que prendan bomba extracorpórea.
3. Colocación del Bypass.
- a) Arteria.
 - 1) Incisión.
 - a' Pedir Bisturí.
 - b' Buscar un lugar para cortar.
 - c' Clampear con el dedo.
 - d' Realizar incisión.
 - e' Retirar Bisturí.
 - 2) Entubación.
 - a' Pedir injerto con sutura.
 - b' Acoplar injerto a "Arteria".
 - c' Desclampear.
 - 3) Cierre
 - a' Pedir Porta Agujas Sin Aguja.
 - b' Suturar injerto a "Arteria".
 - c' Retirar Porta Agujas.
 - b) Corazón.
 - 1) Incisión.
 - a' Pedir Bisturí.
 - b' Buscar un lugar para cortar.
 - c' Clampear con el dedo.
 - d' Realizar incisión.
 - e' Retirar Bisturí.
 - 2) Entubación.
 - a' Tomar extremo de injerto suturado.
 - b' Acoplar injerto al Corazón.
 - 3) Cierre.
 - a' Pedir Porta Agujas Sin Aguja.
 - b' Suturar.
 - c' Retirar Porta Agujas.

4. Retirar el CEC.

a) Declampeonato de Vena Cava Inferior.

- 1) Desclamp.
 - a'* Abrir Pinzas.
 - b'* Retirar Pinzas que sujetan la Vena Cava Inferior.
- 2) Abrir Sutura.
 - a'* Pedir Tijeras.
 - b'* Cortar suturas.
 - c'* Retirar Tijeras.
 - d'* Pedir Pinzas.
 - e'* Retirar suturas con Pinzas.
 - f'* Retirar Pinzas.
- 3) Retirar Tubo.
 - a'* Sujetar Tubo con la mano.
 - b'* Desacoplar Tubo.
 - c'* Retirar Tubo.
- 4) Suturar.
 - a'* Pedir Porta Aguja.
 - b'* Suturar herida.
 - c'* Retirar Porta Aguja.

b) Declampeonato de Aurícula Derecha.

- 1) Desclamp.
 - a'* Abrir Pinzas que clamplean la Aurícula Derecha.
 - b'* Retirar Pinzas que sujetan la Aurícula Derecha.
- 2) Abrir Sutura.
 - a'* Pedir Tijeras.
 - b'* Cortar suturas.
 - c'* Retirar Tijeras.
 - d'* Pedir Pinzas.
 - e'* Retirar Suturas con Pinzas.
 - f'* Retirar Pinzas.
- 3) Retirar Tubo.
 - a'* Sujetar Tubo con La Mano.
 - b'* Desacoplar Tubo.
 - c'* Retirar Tubo.
- 4) Suturar.
 - a'* Pedir Porta Aguja.
 - b'* Suturar herida.
 - c'* Retirar Porta Aguja.

5. Pedir que se coloque el marca pasos.

6.4.3. Animación de corte

Unity es un entorno de desarrollo orientado a operaciones gráficas de alto nivel, en el que ya están implementadas la detección de colisiones, la simulación de la gravedad, por mencionar algunas. Esto significa que el acceso a controles de bajo nivel resulta problemático.

El efecto se lleva a cabo de la siguiente manera. Usando la información de los puntos de contacto, los cuales se guardan en el objeto Collision a la hora de la colisión, se determina cuál es el vértice del mesh que se encuentra más cercano al punto de colisión calculando la distancia de ese punto con todos los vértices del mesh. Se hace de esta forma ya que no hay manera de determinar qué triángulo es el que se toca o qué vértice es el más cercano de otra manera. Únicamente se sabe el mesh con el que se está haciendo colisión.

Variable	Descripción
relativeVelocity	La velocidad lineal relativa de los dos objetos que colisionan (Sólo lectura).
rigidbody	El componente Rigidbody del objeto con el que se choca (Sólo lectura). Es null si se colisiona con un objeto que no contiene un componente Rigidbody
collider	El componente Collider del objeto con el que se choca (Sólo lectura)
transform	El componente Transform del objeto con el que se choca (Sólo lectura).
gameObject	El objeto completo con el que se choca.
contacts	Los puntos de contacto generados por el motor de física.

Tabla 6.1: Variables del objeto Collision

Una vez que se tiene el vértice más cercano al punto de colisión, lo que se hace es mover ese vértice en sentido contrario de su normal para que siempre se hunda y no se eleve.

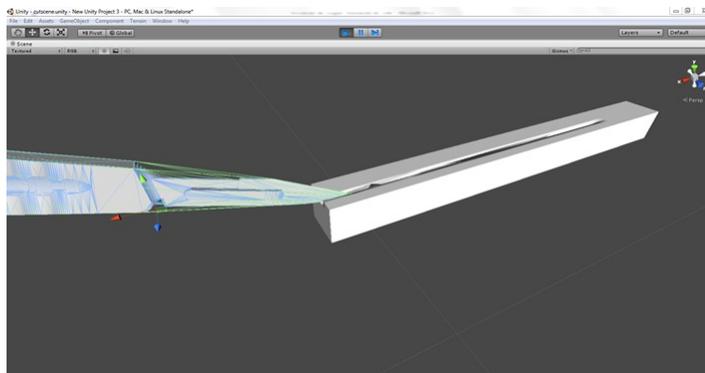


Figura 6.9: Vista de la animación de corte en Unity

6.4.4. Modelado Matemático

Para poder modelar el mecanismo de circulación sanguínea, se siguieron los pasos recomendados en [1], [8] y [16], utilizamos la fórmula de Poiseuille para calcular la presión de un fluido en un tubo laminar.

$$\Delta P = \frac{8\mu LQ}{\pi r^4} \quad (6.1)$$

Donde: ΔP es la diferencia de presión

L es el largo del tubo

μ es la viscosidad dinámica

Q es la tasa de flujo volumétrico y

r es el radio del tubo.

Todos estos datos los obtuvimos de las fuentes antes mencionadas y de la base de datos del NIH.

6.5. Procedimiento de modelado de manos

6.5.1. Reflexión de puntos

La forma en que el dispositivo kinect detecta los objetos que se encuentran frente a él, es por medio de rayos infrarrojos que parten de un emisor y son lanzados de forma cónica. Estos rebotan sobre el objeto y son detectados por un receptor, que se encuentra junto al emisor; estos rayos son imperceptibles por los humanos, y no dependen de la iluminación de la habitación, por lo que no se requiere de un entorno controlado. Esto tiene una enorme ventaja sobre el análisis de imágenes, que sólo detecta objetos por medio de imágenes en entornos controlados.

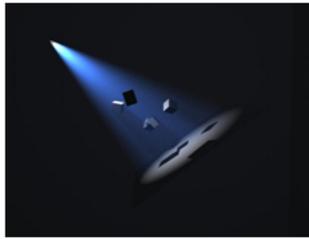


Figura 6.10: Iluminación en cono

Una forma de entender como Kinect funciona es viéndolo como si fuera una lámpara, los haces de luz que son emitidos sólo se pueden apreciar sobre los objetos frente a él, sin embargo sólo es iluminado un lado, dejando a "oscuras" la parte posterior, si nuestro objeto a iluminar es medianamente pequeño, se desperdicia una gran parte de los haces de luz, la manera en que se pueden "reutilizar" estos haces, y específicamente los rayos infrarrojos que son emitidos por el Kinect, es utilizando espejos, redireccionando los haces a la parte posterior que no es iluminada, al hacer esto podemos tener una descripción más completa de un objeto que se encuentre dentro del rango de reflexión.

Lo que recibimos del kinect es un arreglo de 640×480 valores, donde cada celda contiene la distancia a la que se encuentra un punto de un objeto, para poder calcular la nueva posición de un punto es por medio de la siguiente fórmula:

$$\vec{r} = \vec{d} - \frac{(\vec{d} \cdot \vec{n})\vec{n}}{\|\vec{n}\|^2} \quad (6.2)$$

donde:

\vec{d} es el vector del haz de luz incidente

\vec{n} es la normal de la superficie reflejante

\vec{r} es el vector del haz de luz reflejado

Para que se pueda calcular el nuevo punto reflejado debe cumplir que se encuentre dentro del área que delimitamos para los espejos y así mismo que la distancia sea mayor a donde se encuentra la superficie reflejante, de lo contrario el haz de luz no se debe reflejar, ya que aunque se encuentra en la trayectoria del espejo no toca su superficie.

Debido a que se deben procesar 307200 Vectores (x, y, z) cada frame, el tiempo de ejecución se apreciaría muy lento, por lo que todo el módulo de reflexión es procesado por la tarjeta gráfica y su mayor tiempo dependerá del tiempo en que se envían y reciben los vectores dependiendo de las



Figura 6.11: Reflexión del haz de luz

especificaciones de cada tarjeta gráfica. En nuestro caso particular, mandamos llamar un bloque con 1024 threads, donde cada thread realiza el trabajo de 300 Vectores por lo que el tiempo de ejecución dentro de la tarjeta gráfica es despreciable, sin mencionar que es un dispositivo dedicado y optimizado a realizar cálculos matemáticos, por lo que su fiabilidad en los resultados es superior a los resultados procesados por el CPU.

Para la tarjeta gráfica 640M en la que se realizaron pruebas, el tiempo calculado de la tasa de envío de datos es de $0,0715ms$. mientras que si estos cálculos se realizarán en el CPU, el tiempo calculado es de $2,17ms$, en un procesador i7 de tercera generación a $2,10Ghz$. esto justifica el uso del lenguaje CUDA para poder optimizar los cálculos para este trabajo terminal.

6.5.2. Discriminación de puntos

Una vez que se tienen calculados las posiciones de los puntos que han pasado sobre una superficie reflejante, es necesario eliminar todos los puntos que no aportan información para el cálculo de las manos. Estos puntos pueden encontrarse detrás de los espejos o detrás de Kinect (debido a la reflexión de los puntos), o aquellas regiones en las que el cirujano virtual jamás se encontrará, debido a que se está operando sobre una región delimitada por el cuerpo del paciente virtual.

La técnica utilizada para determinar si un punto está dentro de un mesh, es la utilizada para determinar si un punto está dentro de un polígono trasladado a tres dimensiones.[\[19\]](#)

Esta técnica funciona así: en 2D se proyecta un rayo desde el punto evaluado a cualquier dirección, y se debe contar cuántos segmentos de recta interseca, estos segmentos de recta son los que forman el polígono, si el número de intersecciones es par podemos asegurar que el punto se encuentra fuera del polígono, si el número es impar el punto se encuentra dentro del polígono.

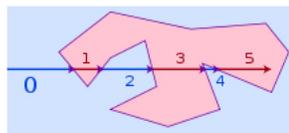
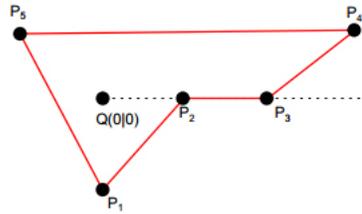
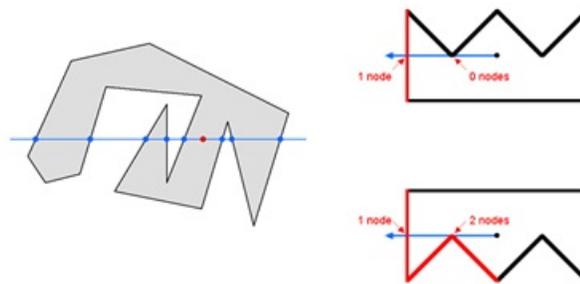


Figura 6.12: Algoritmo PIP para polígonos complejos

Sin embargo existen casos especiales en los que no cumplen con esta afirmación y esto se debe a que toca en los bordes de los segmentos, es decir, los vértices de los segmentos de recta.



Una posible solución a este problema en 2D, es calcular la proyección sobre el par de segmentos que se interseccionan calculando su producto punto. Si ambas magnitudes son positivas, se cuentan como dos intersecciones, o para efectos prácticos, como ninguna intersección. Si no cumple con esto se considera como una intersección ordinaria.



Al trasladar esta técnica a tres dimensiones cambia la manera en que se comparan las intersecciones, ahora se realizan con los triángulos que conforman el mesh, y quedan tres casos:

- Cuando el vector de intersección toca una cara formada por los triángulos que componen el mesh.

Este es el caso base, ya que para tres dimensiones ya no se comparan segmentos de recta, se comparan los triángulos que componen el mesh y solo se calcula la intersección de un vector con un plano. Este plano está compuesto por los vértices del triángulo. Este caso, siempre se considerará como una intersección simple.

- Cuando el vector de intersección toca una arista de los triángulos que componen el mesh. Para este caso se debe calcular la proyección del vector sobre los dos planos con los que se tiene intersección. Al obtener dos vectores nuevos, se hace una proyección sobre estos dos vectores y si la magnitud de ambos vectores es positiva, no se considera la intersección, de lo contrario es sólo otra intersección más.
- Cuando el vector de intersección toca un vértice del mesh. En este caso se deben de ver las caras como si formaran una pirámide donde la punta de la pirámide es el vértice con el que interceptamos. Si el vector de intersección, .^{entra.} a la

pirámide, no se debe de considerar esta intersección. Para los otros casos en los que sale, pasa solo por la punta, o está sobre una de las caras de la pirámide, sí se debe de considerar su intersección.

Estos algoritmos al igual que con el módulo de reflexión, son calculados en su mayor parte en la tarjeta gráfica.

6.5.3. Clustering de puntos

Utilizamos el clustering de puntos para poder reducir los puntos sobre los que se van a trabajar en la triangulación y eliminar los puntos agrupados, que no cumplan con un tamaño mínimo.

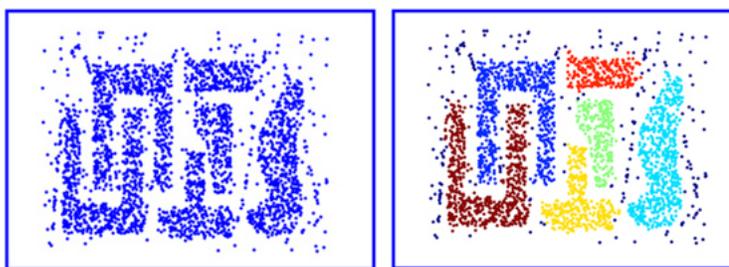


Figura 6.13: Ejemplo del Algoritmo DBSCAN

El algoritmo usado para el clustering de puntos es DBSCAN el cual requiere que la densidad de puntos sea muy grande. Esto se logra con la ayuda de la reflexión de los espejos ya que no se necesita saber cuántos clusters existen en la nube de puntos a procesar.

Ventajas

- Pueden existir clusters que estén "zigzageando" entre sí, y esta técnica pueda separarlos, siempre y cuando estén separados al menos un epsilon de distancia.
- Reduce el ruido debido a que los puntos dispersos no son agrupados. Este ruido se espera, debido a que el Kinect no es un dispositivo especializado para la creación de una nube de puntos de un objeto físico. Aun cuando se tenga la entrada de un dispositivo especializado, siempre va a existir ruido.
- Es una técnica ampliamente documentada.

Desventajas

- Al usar un epsilon muy pequeño, se deben de hacer más comparaciones contra todos los puntos aún no visitados, lo que puede llegar a hacer el algoritmo muy lento.
- No es recomendado para una gran cantidad de puntos por todas la distancias que se calculan. Al menos no en el algoritmo tradicional.

Este algoritmo de clustering, aunque es muy eficiente y es lo que se requiere para tener un menor número de datos para procesar en la triangulación y detección de huesos, el tiempo que requiere

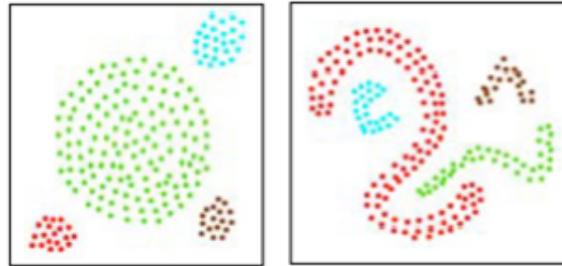


Figura 6.14: DBScan

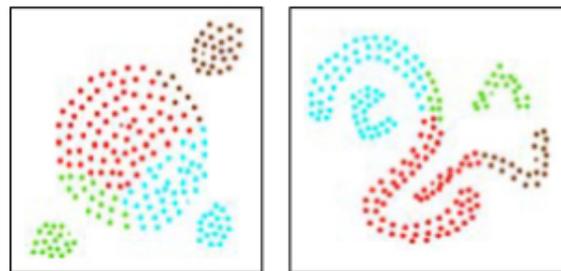


Figura 6.15: CLARANS

para realizarlo, es alarmante para este trabajo terminal ya que se requiere en tiempo real y este clustering de puntos se debe de realizar cada frame.

Una solución con la cual se atacó el problema del tiempo requerido es por medio de la agrupación de puntos por sectores, esto lo realizamos con la ayuda de los límites dados por el mesh de discriminación de puntos. Esta región, la cual trataremos como un cubo que tiene inscrito el mesh de discriminación, se divide en bloques más pequeños. Dentro de estos bloques, guardamos los puntos de la nube a clusterizar, con esto tenemos un menor número de datos a revisar y la revisión de puntos será más rápida y eficiente.

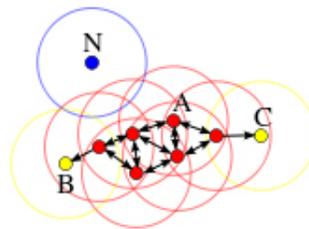


Figura 6.16: Ejemplo del funcionamiento de DBScan

De acuerdo con el algoritmo DBSCAN, para poder considerar la expansión de un cluster, un grupo de puntos que están a un epsilon de distancia deben ser de al menos un tamaño establecido. Para comprobar esto, debemos revisar en los bloques si existen al menos esa cantidad de puntos en

dicha región, de lo contrario se revisa el siguiente bloque.

Para poder expandir el cluster, el algoritmo nos indica que para cada punto no evaluado del grupo inicial, se deben de buscar los puntos más cercanos a un ϵ de distancia y agregarlos al grupo inicial, hasta que ya no exista dentro de nuestro grupo ningún punto cercano no evaluado.

Al tener la nube de puntos en bloques, sólo se requiere buscar en seis direcciones para poder expandir el grupo, esto hace más eficiente el poder expandir un nuevo cluster, lo que decrementa su tiempo de ejecución.

Para asegurarnos que cualquier punto dentro de un bloque se encuentra a no más de un ϵ de distancia de otro punto contenido dentro de un bloque contiguo, el tamaño del cubo del bloque deberá de ser:

$$L = \frac{\epsilon}{\sqrt{6}} \quad (6.3)$$

Esto es debido a que si un punto se encuentra en la esquina de uno de los bloques, el punto más alejado del bloque contiguo se encontrará en la esquina opuesta. Esta máxima distancia debe de medir ϵ , de aquí se demuestra el tamaño del L calculado.

Puede existir un punto que se encuentre cerca del borde del cubo y otro a dos bloques contiguos cerca de la cara más contigua. Estos estarán a una distancia menor que ϵ , sin embargo no se conectarán debido a que no hay un punto contiguo inmediato que los conecte. Este caso es muy raro, sin embargo puede llegar a ocurrir, es por eso que no se revisan las diagonales ya que aumentarían las posibilidades de que esto ocurra, sin embargo si los puntos no están muy dispersos, estos se agruparán en menos bloques, lo que reducirá la comparación de puntos. El mejor de los casos es que todos los puntos a evaluar se encuentren dentro de un sólo bloque y no se revise ningún punto. El peor de los caso está dado por el máximo ancho, alto y profundo, dividido entre el $L = \frac{\epsilon}{\sqrt{6}}$.

Una vez que se tiene el cluster calculado, se pueden eliminar clusters que no cumplan con un tamaño mínimo. Esta cantidad de puntos debe ser lo suficientemente grande para que después se puedan calcular los huesos, o dicho de otra forma, el centro del mesh.

El artículo [20], reporta que para 12512 puntos tarda un estimado de 41,7 segundos para un procesador estándar de 1996 (aproximadamente 250MHz). Para la computadora de pruebas, a 2,10Ghz se calcula que tardará para 307200 puntos 2 minutos, esto hace que con la forma tradicional esta técnica no sea viable con la implementación de bloques y guardando estos en un árbol AVL el tiempo de ejecución es de /*Colocar tiempo, es del orden de milisegundos*/.

Este módulo, al requerir un mayor manejo de apuntadores que los otros módulos, toda su implementación está en CPU.

6.5.4. Triangulación

Una vez que se han reflejado los puntos, se han eliminado los puntos que no se encuentran dentro de una región y estos puntos resultantes han sido agrupados, se procede a triangular para cada una de estas regiones. La técnica utilizada está basada en el artículo [21], en la que sólo existen dos procesos a seguir:

1. Expandir un segmento de la superficie.
2. Juntar dos bordes de triángulos para formar uno nuevo.

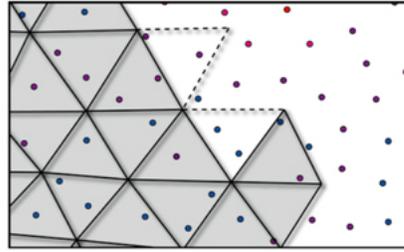


Figura 6.17: Triangularización

Esta técnica contempla y espera que la nube de puntos a triangularizar tenga ruido, lo cual es una ventaja ya que se acopla a las necesidades de este trabajo terminal. El aporte que se realiza a esta técnica es la agrupación de triángulos que tienen una normal muy parecida, o difieren respecto a un rango establecido. Este polígono no necesariamente será convexo; una vez que a que polígono esté terminado se deberán calcular los triángulos dentro de este polígono.

Ventajas

- Debido a que forma una nueva superficie, los vértices del mesh final, no necesariamente corresponderá con los que contiene la nube de puntos, y la cantidad de puntos finales se asegura que siempre será menor.

Desventajas

- Se deben de calcular los triángulos de cada polígono por medio de la técnica de Delaunay.
- Al buscar el siguiente punto para expandir el segmento se debe de hacer sobre toda la región agrupada, ya que de lo contrario, se deberán de eliminar los puntos que están abarcando los triángulos.

Para este módulo se utiliza el GPU para calcular los puntos de expansión y el CPU para la ordenación, intersección y búsqueda de los puntos que se encuentran en la región.

6.6. GUI

6.6.1. Pantalla principal

Para la GUI de creación y selección de perfiles se creó una interfaz pensando en que se manejarían muchos perfiles. La interfaz incluye pestañas por letra las cuales hacen el acceso a los perfiles mucho más sencillo.

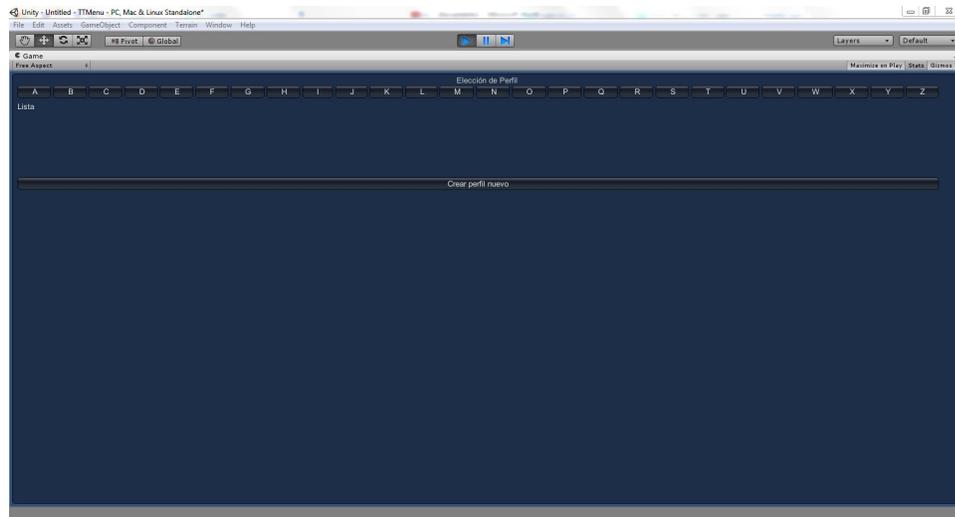


Figura 6.18: Pantalla Principal

6.6.2. Creación de perfil

Para crear un perfil, únicamente se solicita el nombre del doctor.

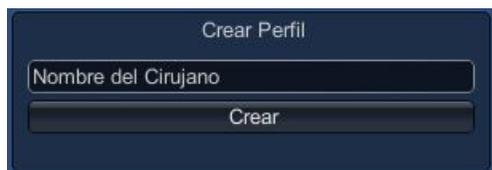


Figura 6.19: Creación del Perfil

6.6.3. Menú de perfil

Para crear un perfil, únicamente se solicita el nombre del doctor.

6.6.4. Menú de estadísticas

Las estadísticas se crean a partir de la información de todas las cirugías que el doctor ha realizado con ese perfil. Incluyen tiempo total de cirugía, intentos fallidos y exitosos, un promedio



Figura 6.20: Menú perfil

de todas las puntuaciones que ha obtenido y su ranking en comparación con los demás perfiles.

6.6.5. Almacenamiento de la información

Para almacenar toda la información de los perfiles se pensó en usar una base de datos, pero debido a que esto le agregaría más requerimientos al sistema final, lo que se usó fue el software SQLite. Este software tiene las ventajas de una base de datos y de un archivo como se explica a continuación.

Ventajas

1. El software es del dominio público
2. No requiere de soporte alguno de librerías externas y requiere de poco soporte por parte del sistema operativo
3. No se necesita de un servidor para guardar la información
4. No requiere de instalación alguna
5. La base de datos que se crea es completamente transaccional

Este software se aprovecha al 100

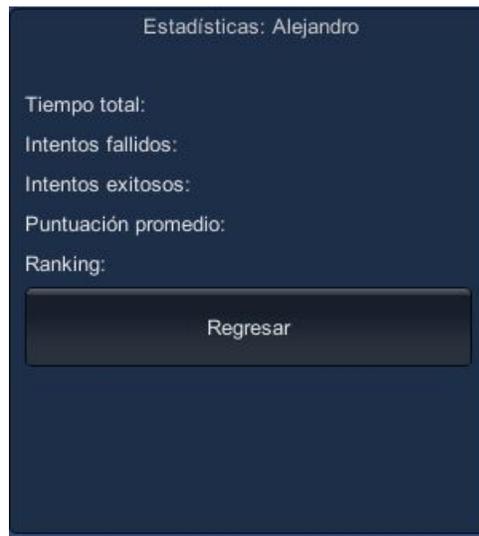


Figura 6.21: Menú de estadísticas

Capítulo 7

Pruebas

7.1. Modelos tridimensionales

Los modelos en tercera dimensión los dividimos en dos categorías:

Modelos de sistema cardiovascular: Son los modelos que constituyen al cuerpo y la sección a operar.

Instrumentos de quirófano: Son los modelos que integran el equipo médico que es requerido para la operación.

Por fortuna, encontramos el modelo de cada pieza del sistema cardiovascular en [7] donde pudimos descargar cada una de las partes del sistema.

En el caso de los modelos de los instrumentos de quirófano, hayamos varios paquetes de modelos que contenían todos los que necesitábamos, sin embargo todos eran de paga, por lo que decidimos hacer nosotros los instrumentos que hacían falta.

7.1.1. Sistema cardiovascular

Para realzar el realismo de la simulación, se requiere que los modelos del sistema cardiovascular sean lo mas realistas posibles. Así que realizamos una lista con las características requeridas, tanto técnicas como estéticas:

- Medidas anatómicamente correctas.
- Formato compatible con Unity.
- Texturas realistas.
- Qué contarán con esqueletos para animación y manipulación del mesh.

En base a estos criterios, consideramos utilizar OsiriX[17], que utiliza como entrada un archivo CAT Scan (tomografía axial computarizada) que permite que los modelos 3D se puedan exportar a diferentes formatos, una de ellas el formato .obj, el cual es compatible con Unity.

Se contaban con las tomografías necesarias para ingresar en el programa, pero los requerimientos del programa nos impidieron seguir con esta posibilidad.

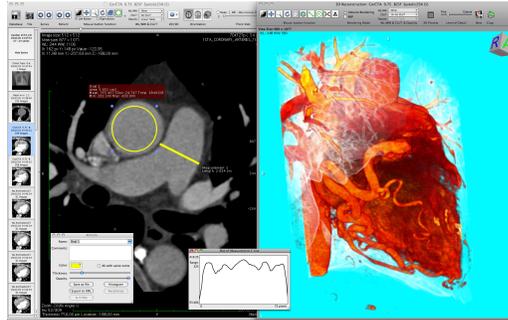


Figura 7.1: Tomografía Axial Computarizada

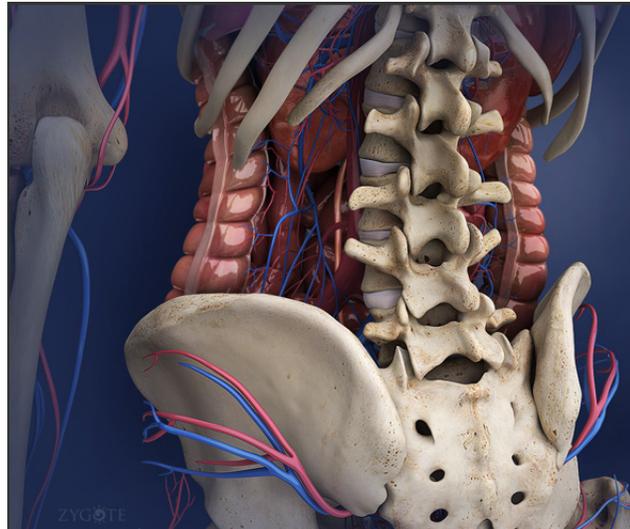


Figura 7.2: Modelos ofrecidos por la empresa ZYGOTE

Después, se propuso encontrar modelos 3D los cuales pudiéramos simplemente comprar para usarlos en la simulación. Se encontraron modelos muy buenos como los ofrecidos por la empresa ZYGOTE[18]. Sin embargo el precio de estos modelos se salía completamente de nuestro presupuesto (desde \$1000 dólares hasta \$18,000 dólares) por lo que reservamos esa opción como último recurso.

Al final se optó por utilizar unos modelos 3D que se encontraron en una base de datos japonesa la cual es parte de un proyecto para facilitar ejemplos de medicina a través de la web en forma de atlas anatómico. Sus modelos estaban disponibles para descarga y uso bajo la licencia Creative Commons Attribution-Share Alike 2.1 Japan.

Usamos estos criterios para seleccionar los modelos a usar. Finalmente utilizamos los que proveen en [7] pues cumplían con las primeras dos características deseadas.



Figura 7.3: Software BodyParts3D basado en web.

7.1.2. Quirófano

Para los modelos del quirófano, debido a que se requería darle ambientación a la operación se decidió buscar los modelos en internet, ya que existen muchas páginas web en donde se pueden encontrar este tipo de contenidos de manera gratuita. Los modelos que se encontraron tenían texturas demasiado complicadas para el motor Unity y este las convertía a texturas default completamente grises, por lo que se tuvo que dar texturas más simples después de exportar el modelo.

7.2. Háptica

7.2.1. Primer Prototipo

Se hizo un primer prototipo con una matriz de LEDs donde pudiéramos probar la etapa de control y el correcto funcionamiento de la etapa de multiplexado y envío de datos.

La etapa de control y multiplexado consisten en tratar de manejar 15 LEDs acomodados en forma de una matriz de 3x5 mediante 8 salidas de la placa Arduino; tres de estas salidas están configuradas como salidas analógicas, ya que por ellas se implementa la técnica de PWM para poder variar la intensidad de encendido de los LEDs. Las cinco salidas restantes están configuradas como salidas digitales, ya que su función es la de activar cada una de las columnas.

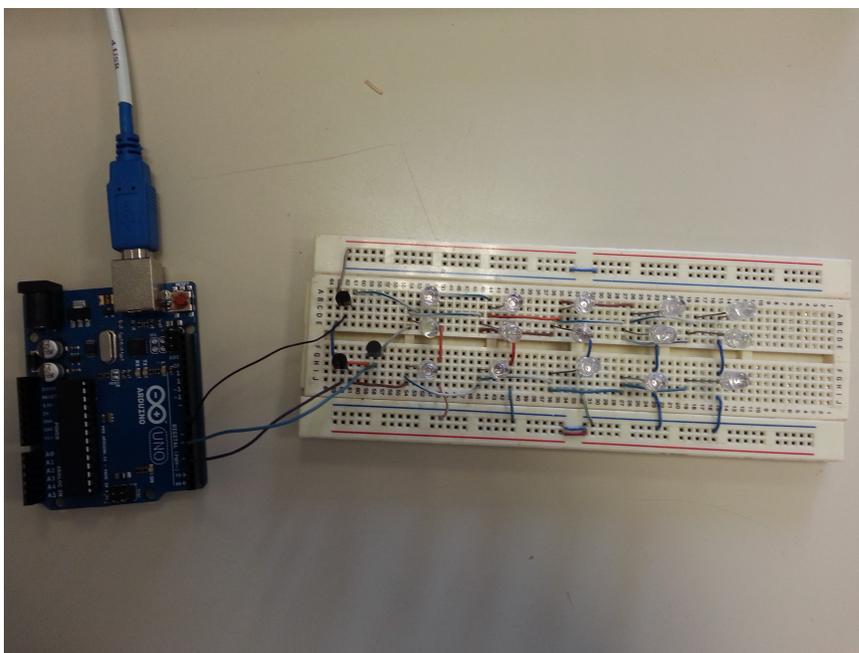


Figura 7.4: Tomografía Axial Computarizada

7.2.2. Segundo Prototipo

Una vez implementadas y probadas la primer etapa, se elaboró un segundo prototipo el cual utilizaba como dispositivos de salida para el guante unas pequeñas bocinas, con las cuales se pretendía generar un cierto tipo de onda mecánica que fuese capaz de hacer sentir al usuario algún tipo de sensación física en las manos.

Sin embargo, la baja potencia de la señal mecánica y el tamaño de la bocina terminaron por eliminar la posibilidad de utilizar bocinas como actuadores hápticos.

7.2.3. Tercer Prototipo

Como tercer prototipo se optó por utilizar unos pequeños motores vibratorios, los cuales mediante las señales PWM se puede hacer variar la intensidad con la que vibran.

Sin embargo, tuvimos los siguientes problemas a la hora de realizar las pruebas correspondientes:

El primer problema fue la etapa de multiplexado que ya no funcionaba como debía, ya que los motores se polarizan en ambos sentidos por lo que al activar una columna, los motores se polarizaban en un sentido y al desactivar esa misma columna, se polarizaban en sentido inverso; lo que provocaba que todo el tiempo estuvieran en funcionamiento.

El segundo problema encontrado fue el switcheado que se estaba implementando tampoco funcionaba; pues funcionaba a una frecuencia muy por encima del rango de respuesta de los motores, lo que hacía que estos no llegasen a girar como debían o incluso a veces no llegaban a girar.

En sí, la idea principal era que en cada dedo se ubicaran tres motores y que cada uno de ellos fuese independiente en cuanto al control se refiere pero cuando encontramos los problemas mencionados, se tuvo que replantear la forma de controlarlos quedando de la siguiente manera:

En lugar de ocupar tres motores por dedo, ahora solo son dos y en lugar de que cada motor se controle de manera independiente, ahora cada dedo es independiente y los motores que incorpora cada dedo vibran exactamente a la misma intensidad.

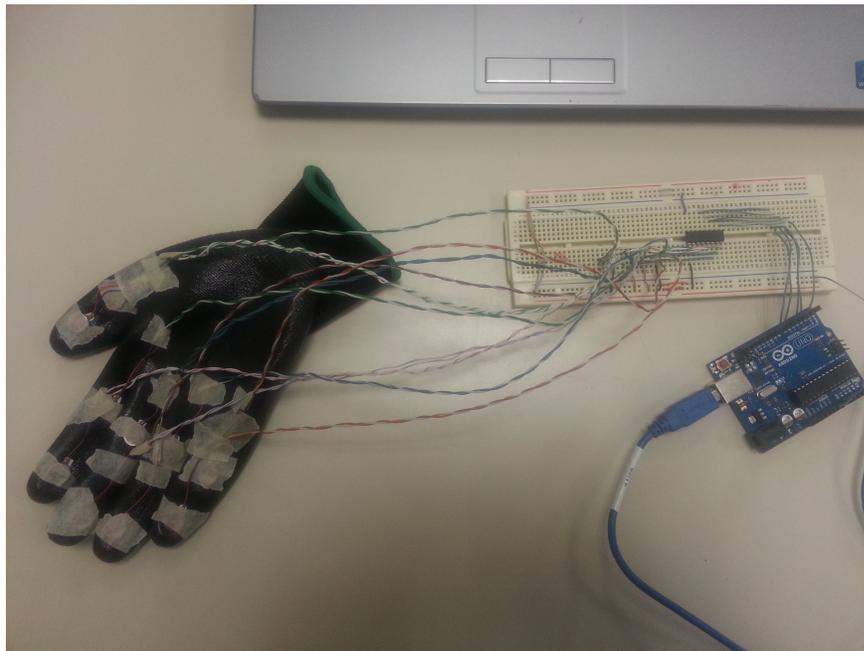


Figura 7.5: Prototipo final

7.3. Comando de Voz

Se presentaron problemas al momento de integrar este módulo con el sistema principal, ya que no reconocía la librería que desarrollamos. Finalmente, el problema fue resuelto añadiendo todas las librerías requeridas en el sistema. El reconocimiento de voz es sensible, por lo que puede presentar errores si existe demasiado ruido en el ambiente, principalmente por la voz de otras personas que pueden encontrarse cercanas al micrófono. El módulo presenta los modelos de los instrumentos en pantalla para que el usuario pueda interactuar con ellos.

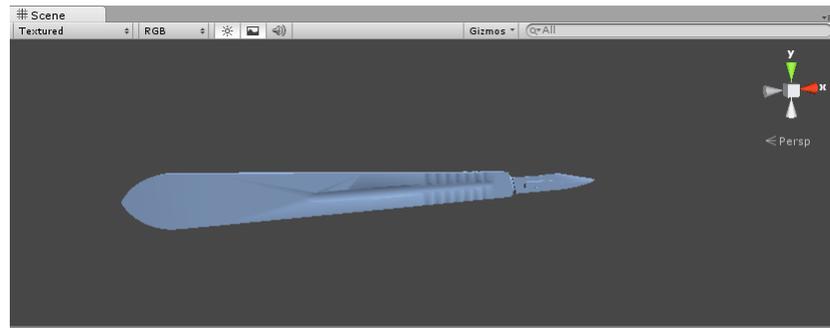


Figura 7.6: Modelado de bisturí

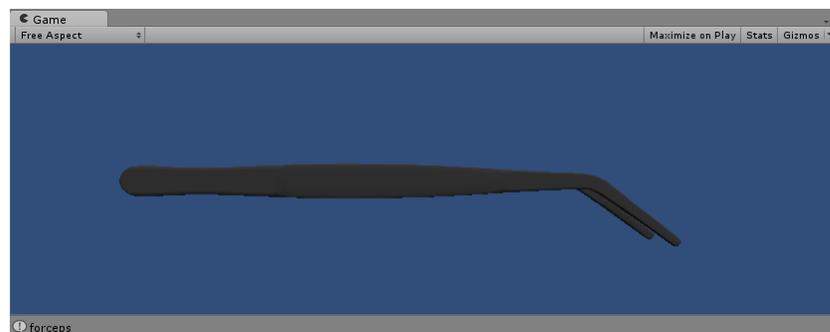


Figura 7.7: Modelo de fórceps

7.4. Procedimiento de Modelado de Manos

7.4.1. Reflexión

La utilización de espejos como herramienta, para poder obtener un mayor número de datos de la superficie de un objeto, fue la propuesta que se tuvo para poder tener un mayor grado de precisión en el movimiento de las manos, así mismo nos permite tener una redundancia de datos de la superficie esto con el objetivo de reducir el ruido generado por el dispositivo de captura, sólo en proporción de datos reales, respecto al ruido generado.

El primer desafío que se presentó para realizar este trabajo terminar fue, el mejorar la forma en que se manejan los datos recibidos del dispositivo Kinect, proporcionados por el Wrapper de OpenNI (el intermediario entre el Kinect y el motor de físicas Unity), lo que se recibe es un apuntador no "interpretado" por Unity, lo que se realiza hasta el momento es pasar esa localidad de memoria a una arreglo instanciado desde Unity, este volcado de memoria reduce el tiempo de ejecución ya que se requiere mover 307200 datos cada frame, para que pueda ser utilizado, la "mejora" que se realizó fue pasar directamente este apuntador a la DLL, con sus respectivos desafíos, evitando un volcado de memoria innecesario.

Una vez obtenido este apuntador, se puede manipular para realizar la reflexión de los puntos sobre la superficie de los espejos.

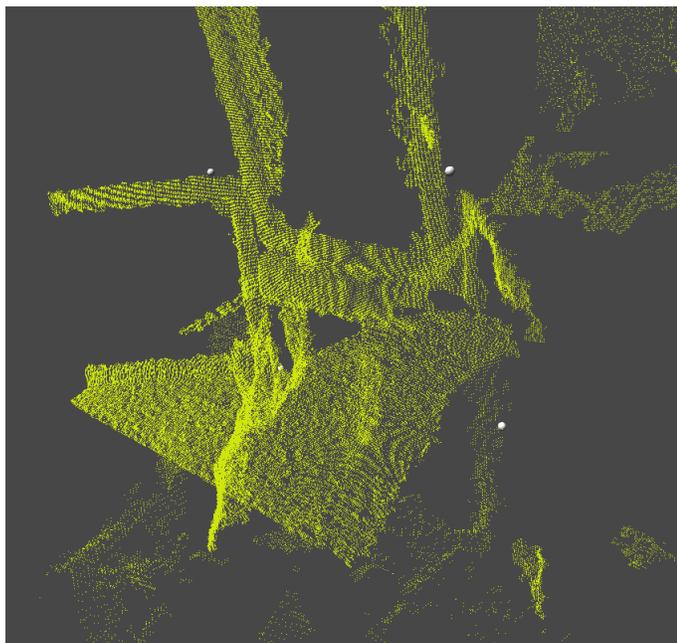


Figura 7.8: Nube de puntos original

La manera en que se calibran los espejos, es por medio de un contraste, en este caso una cinta de color azul, estos puntos formarán el plano, y delimitan la localización de los espejos; Para el prototipo entregado, estos puntos, se calibran de manera manual solamente sobre el eje X Y, la profundidad es calculada respecto al grupo de puntos que se encuentran sobre este medio de contraste, para trabajo a futuro, estos puntos se localizarán y agruparán de manera automática

evitando al usuario lo molestia de calibrar los espejos.

Para la reflexión de los puntos primero se estableció el tiempo base a superar, es decir los cálculos que se realizan solamente en el CPU, arrojando un tiempo de ejecución de 0.38 seg, esto para los 640 x 480 puntos manejados, mientras que haciendo los mismos cálculos sobre la tarjeta gráfica, devuelve un resultado en un tiempo de 0.0002 seg, sólo de tiempo de procesamiento, al pasar los datos y regresar los datos del GPU, el tiempo total de ejecución es de 0.0014 segundos, este tiempo será posible reducir según la tarjeta con la que se esté trabajando, ya que dependerá de las especificaciones de esta.

Esta reducción dramática de tiempo, justifica nuestra elección del uso de CUDA, para la implementación de ciertos procesos sobre la tarjeta gráfica.

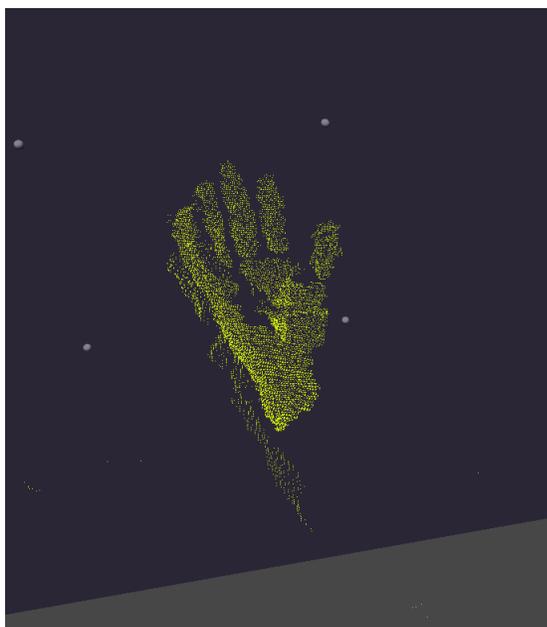


Figura 7.9: Eliminación de puntos

7.4.2. Discriminación de Puntos

Existen muchas y diversas técnicas para determinar si un punto se encuentra dentro de un Mesh, cada una de ellas con sus cualidades y desventajas, la técnica final que se implementó es una variación de la determinación de un punto dentro de un polígono, trasladado a tres dimensiones, esto previamente explicado.

Estos cálculos, se realizaron sobre la tarjeta gráfica mostrando un desempeño adecuado para las especificaciones de este trabajo terminal, sin embargo al probarlos en diferentes máquinas, con diferentes tarjetas gráficas, queda al descubierto la precisión con la que se realizan las intercepciones con las caras del Mesh, en modelos más recientes no se aprecia tanta fuga de puntos localizados entre las caras, y los vértices que forman el Mesh, estos puntos ya no forman parte de ningún objeto que tenga utilidad para la representación de las manos, lo que para este punto son considerados como basura, estos puntos serán eliminados, al clusterizarlos.

De esto cabe destacar que la implementación realizada, tiene la teoría de poder calcular si un punto se encuentra dentro de un Mesh, sin importar si este es totalmente convexo o cóncavo.

Para este trabajo terminar, la superficie de trabajo, está conformado por los espejos y un cuadro, lo que forma una pirámide trunca, esta región es convexa, la mejor de todas las regiones para determinar puntos interiores, sin embargo esto no ayuda ni perjudica a la técnica utilizada.

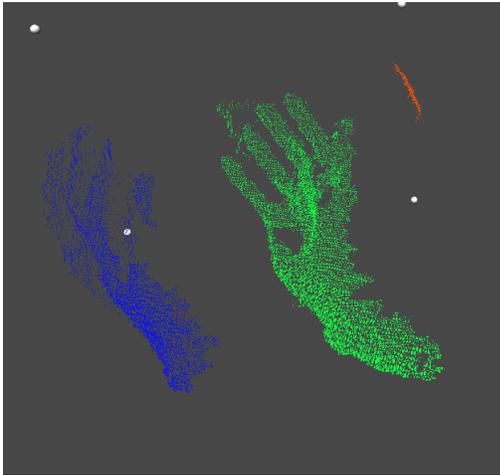


Figura 7.10: Clusterización de puntos

7.4.3. Clusterización

La agrupación de puntos nos servirá para poder calcular solo la superficie de regiones que específicas, al mismo tiempo el grupo resultante tendrá un tamaño específico, este tamaño podrá ser criterio de discriminación, para esta imagen mostrada, se aprecia una tercer región que no aportará datos útiles, esta región será eliminada.

La implementación y mejora de la técnica DBSCAN, es uno de los puntos críticos para la implementación de este trabajo terminal.

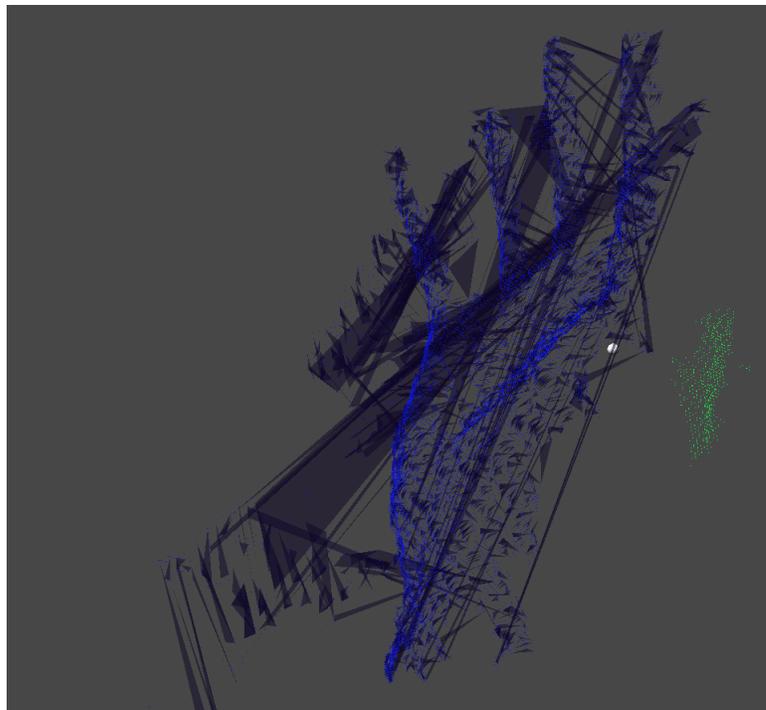


Figura 7.11: Primer prototipo de algoritmo de triangularización

Parte VI

Trabajo a futuro

Como trabajo a futuro se pretende encontrar una manera más eficiente de controlar los actuadores que se incorporan en el guante háptico, forma que estos funcionen de forma más rápida y precisa; además de que puedan ser manipulados de una manera más independiente para poder brindar una mejor experiencia de inmersión para el usuario.

Hemos contemplado utilizar nuevas tecnologías, como es el caso de "Leap Motion" el cual nos permitiría un mejor control y precisión en la detección y manejo de las manos del cirujano dentro de la simulación, al menos según las especificaciones que prometen, pues es un producto aún en desarrollo.

Para los modelos, se propone mejorar los modelos actuales o incluso quizá trabajar en mejores texturas para que la parte visual influya aún más en la inmersión del usuario en el ambiente del quirófano. También se trabajarán las animaciones de los modelos, tanto de herramientas quirúrgicas como de los órganos internos.

Como extra, pensamos en mejorar el ambiente circundante del quirófano, aunque esto sólo sea como parte de una introducción a la operación.

En cuanto a la animación del corte de los órganos, si bien no podemos complicarlo demasiado con un modelo físico para agregarle realismo, podríamos perfeccionar la animación de corte prescindiendo de los métodos de el motor gráfico nos da, y desarrollar métodos propios que estén optimizados para nuestros propósitos.

Para mejorar la simulación médica, hemos contemplado avanzar en el modelado matemático de la presión en arterias y venas del corazón y superar las dificultades de obtención de datos anatómicamente correctos que correspondan con perfiles de pacientes reales.

Actualmente los comandos de voz tienen un 70 % de eficiencia en casos sin ruido, disminuyendo drásticamente su eficiencia en situaciones con ruido, por lo que debemos buscar una manera de reducir la forma en que el ruido afecta al reconocimiento de palabras y aumentar su eficiencia con respecto a los comandos. También consideramos como una buena probabilidad de implementar a futuro, si el motor gráfico Unity es capaz de soportar las librerías de reconocimiento de voz sin ningún problema, utilizar la librería "*Microsoft.Speech*" para ya cuenta con una gran fiabilidad.

Además, se debe ampliar la gamma de comandos de voz para que pueda adecuarse a las necesidades de las operaciones y tal vez, ciertos gentilicios o expresiones usadas dentro del quirófano.

Parte VII

Conclusiones

Conclusión de Trabajo Terminal 1

Como conclusión de TT1, dadas las condiciones tecnológicas del mundo actual y la cantidad de integrantes de nuestro equipo, así como el tiempo y los recursos económicos que tenemos disponible, se estima que este proyecto es factible puede llevarse a cabo.

Aunque nos encontramos en una etapa temprana de análisis del problema y diseño de la solución, al dividir el sistema en partes lo suficientemente atómicas, gran parte de estas partes han sido solucionadas como respuesta a problemas ajenos a nuestro sistema, pero no han sido utilizados en conjunto.

Un ejemplo de esto es el uso de las librerías OpenNI y CUDA, que sólo han sido usadas en conjunto una sola vez por estudiantes graduados de Inglaterra.

Sin embargo, en otros hábitos como lo son los guantes hápticos, estos estudios ni siquiera han llegado a las aplicaciones en ingeniería, quedándose en meras pruebas de ciencia básica. Por lo que debemos insidir en este hábito como desarrolladores de tecnologías novedosas.

Para TT2 presentaremos una mayor integración de estos componentes, que contendrá el sistema de reconocimiento y seguimiento de las manos de médico, los modelos matemáticos y en tres dimensiones del paciente así como de los instrumentos y el guante de sensaciones hápticas que se activará cuando el sistema virtual detecte colisiones entre objetos virtuales.

También mostraremos la simulación de la cirugía de revascularización coronaria como ejemplo del funcionamiento de nuestro sistema, probado por estudiantes y practicantes de esta profesión.

Conclusión de Trabajo Terminal 2

Durante el desarrollo de este trabajo terminal se creó un sistema de simulación de cirugías utilizando un conjunto de tecnologías diferentes que consiguen efectos inmersivos al usuario, gracias a tecnologías hápticas, comandos de voz y el uso de sus manos del usuario dentro de la cirugía.

Integrar estas tecnologías no fue tarea fácil, pues algunas de ellas tenían poca documentación. Por lo que nos dimos a la tarea investigar acerca de este tipo de tecnologías, además de la inminente necesidad de investigar acerca de la cirugía del caso práctico y la familiarización con los términos médicos.

Aun con la complejidad del presente trabajo, finalmente desarrollamos un guante que gracias a pequeños motores, envía sensaciones hápticas al usuario de manera simultánea a su propia interacción con el mundo virtual.

Una de las cosas que nos facilitó mucho esta comunicación fue el mismo motor gráfico Unity, que contaba con muy buena documentación acerca de la comunicación con hardware externo, como la tarjeta Arduino, que fue la que finalmente usamos para el control del circuito háptico, actualizando datos dada una colisión dentro de nuestra aplicación.

Para el desarrollo del caso práctico de la cirugía de revascularización coronaria, tuvimos que familiarizarnos no sólo con el procedimiento que se lleva a cabo para realizarla, sino con todo el entorno que rodea a una cirugía. Esto nos llevó a buscar modelos de herramientas que fueran útiles en esta y otras cirugías, pero encontramos demasiadas dificultades, tanto financieras como de requisitos que debían cubrir estos modelos. Por lo que finalmente creamos nuestras propias herramientas virtuales con el software de modelado Blender del cual teníamos muy poco conocimiento.

Ya que la funcionalidad del sistema era darle inmersión al usuario, el cual está cumpliendo el rol del cirujano, nos enfocamos en las tareas que sólo el cirujano principal de la operación lleva a

cabo durante la cirugía y no en las demás, las cuales realiza el resto del equipo de trabajo (son ayudantes, enfermeros y anestesiólogos).

Para puntuar el desempeño del usuario durante la cirugía, desarrollamos una máquina de estados bastante práctica, la cuál nos marca en qué fase de la operación se está, las acciones erróneas que el cirujano puede cometer y los requisitos de inicio y termino de cada fase. A estos desaciertos les dimos un peso y dependiendo de este se va restando del total. Para la realización de esta máquina de estados se desarrolló un script de las acciones que el cirujano lleva a cabo, las cuales a su vez, sacamos de la investigación de la cirugía de revascularización coronaria.

Muchos de los procedimientos como lo son el de corte y sutura, se tuvieron que mantener sencillos dejando un poco de lado su realismo ya que al agregarle complejidad, la tasa de frames por segundo de toda la aplicación disminuía al realizar estos procedimientos, lo cuál volvía que la aplicación no tuviera una respuesta en tiempo real que era un objetivo mucho más importante para el proyecto.

Finalmente, durante el desarrollo de este sistema aprendimos muchas cosas y recordamos algunas más. Por ejemplo, tuvimos que recordar el manejo y sincronización de procesos para el uso de CUDA, algebra vectorial para la de algunos procesos gráficos, el uso de circuitos análogos para el tratamiento de señales para el diseño del guante háptico, etc.

Por otro lado, tuvimos que aprender bastante sobre los motores gráficos, la experiencia de usuario, la situación actual de la enseñanza medica asistida por realidad virtual, sobre medicina, hidráulica, etc.

Parte VIII

Anexos

7.5. Glosario

Glosario Técnico

Arduino Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.[23]

CUDA Siglas de Compute Unified Device Architecture (Arquitectura Unificada de Dispositivos de Cómputo) es una arquitectura de cálculo paralelo de NVIDIA que aprovecha la gran potencia de la GPU (unidad de procesamiento gráfico) para proporcionar un incremento extraordinario del rendimiento del sistema.[24]

DLL Siglas de Dynamic Link Library (biblioteca de enlace dinámico), es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

Fotogrametría La fotogrametría es una técnica para determinar las propiedades geométricas de los objetos y las situaciones espaciales a partir de imágenes fotográficas.

Háptica Conjunto de tecnologías que interactúan con las personas mediante el sentido del tacto

IDE Siglas de Integrated Development Environment (Entorno de Desarrollo Integrado), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

Kinect Es un controlador de juegos el cual por medio de una cámara infrarroja y un micrófono, permite al usuario controlar un juego con su propio cuerpo y voz.

OpenNI Es un SDK de código abierto usado para el desarrollo de aplicaciones y librerías middleware de sensado 3D.

Polígono Simple Es un polígono cuyos lados no adyacentes no se interceptan.

Realidad Virtual Es una ciencia basada en el empleo de computadoras y otros dispositivos, los cuales producen una apariencia de realidad que permite al usuario tener la sensación de estar presente en ella.

Renderizar Es un término informático para referirse al proceso de generar una imagen o vídeo mediante el cálculo de iluminación global partiendo de un modelo en 3D.

Sensor Es un dispositivo capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas.

Siglas de Charge-Coupled Device es un circuito integrado que contiene un número determinado de condensadores acoplados.

Simulación Es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos - para el funcionamiento del sistema.[25]

USB Siglas de Universal Serial Bus (bus universal en serie), es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.

Unity Es un ecosistema de desarrollo de juegos: un potente motor de renderizado totalmente integrado con juego completo de herramientas intuitivas y flujos de trabajo rápidos para crear contenido 3D interactivo.[26]

Pixel es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

Glosario Médico

Anastomosis Es la reconstrucción de unos elementos anatómicos habitualmente similares de manera quirúrgica

Angiología Es la especialidad médica que se encarga del estudio de los vasos del sistema circulatorio y del sistema linfático; incluyendo la anatomía de los vasos sanguíneos (como arterias, venas, capilares) y la de los linfáticos, además de sus enfermedades.

Arteria Mamaria Interna Arteria par que nace en las primeras porciones de la arteria subclavia e irriga los músculos pectorales, las mamas, el pericardio y los músculos abdominales.

Arteria Radial Es una arteria del antebrazo que se origina como rama de bifurcación externa de la arteria humeral.

Arteriotomía Coronaria Incisión que se practica en una arteria del corazón con el fin de introducir un catéter para extraer un trombo que oblitere su luz.

Bomba Extracorpórea Es una máquina la cual consta de una bomba y un oxigenador los cuales cumplen con las funciones del corazón y de los pulmones respectivamente.

Clampear Maniobra quirúrgica que consiste en comprimir un conducto vascular con una pinza.

Canular Maniobra quirúrgica que consiste en introducir un catéter en un conducto generalmente vascular, arteria o vena, para la infusión de líquidos, la extracción de sangre o para pruebas diagnósticas.

Catéter Es un dispositivo de forma tubular que puede ser introducido dentro de un tejido o vena.

Circulación extracorpórea Se refiere a la circulación producida por la bomba extracorpórea.

Febrícula Es una fiebre ligera especialmente la de larga duración y origen no bien conocido.

In Situ expresión latina que significa "en el sitio".

Injerto Venoso Tejido venoso autógeno u homogéneo que se emplea para la sustitución de una pérdida de sustancia en una vena o una arteria.

Marcapasos Dispositivo electrónico diseñado para generar y enviar al impulsos eléctricos al corazón..

Solución Cardiopléjica Sustancia empleada en Cirugía Cardíaca (circulación extracorpórea), para detener la actividad eléctrica y mecánica del corazón

Sonda Véase catéter.

Tejido Blando Todos los tejidos corporales no óseos, como los músculos, la grasa, el tejido fibroso, los vasos sanguíneos o cualquier otro tejido conjuntivo del cuerpo.

Aorta Es la principal arteria del cuerpo humano.

[27]

Bibliografía

- [1] Medical Devices and Systems. The Biomedical Engineering Handbook. Tercera Edición. CRC Press. 2006
- [2] Shabir Bhimji MD, Biblioteca Nacional de Medicina de EE.UU Instituto Nacional de Medicina (NIH), Cirugía de revascularización coronaria Disponible:
www.nlm.nih.gov/medlineplus/spanish/ency/article/002946.htm
- [3] Shabir Bhimji MD, Biblioteca Nacional de Medicina de EE.UU Instituto Nacional de Medicina (NIH), Cirugía a corazón abierto. Disponible:
<http://www.nlm.nih.gov/medlineplus/spanish/ency/article/002950.htm>
- [4] Dr. Roberto González L., Enrique Seguel , Aleck Stockins, Rodrigo Campos M., Luis Neira, Emilio Alarcón Cirugía Coronaria: Revascularización miocárdica sin circulación extracorpórea, Rev. Chilena de Cirugía. Vol 61 - No. 6, Diciembre 2009; pág. 578-581. Disponible:
http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-40262009000600016
- [5] Fuster, R. G., Montero, J. A., Gil, Ó., Hornero, F., Cánovas, S. J., Dalmau, M. J., & Bueno, M. (2002). Ventajas de la revascularización miocárdica sin circulación extracorpórea en pacientes de riesgo. Rev Esp Cardiol, 55(4), 383-90.
- [6] Palma-Ruiz, M., García de Dueñas, L., Rodríguez-González, A., & Sarría-Santamera, A. (2003). Análisis de la mortalidad intrahospitalaria de la cirugía de revascularización coronaria. Rev Esp Cardiol, 56(7), 687-94.
Disponible: https://portal.uah.es/portal/page/portal/epd2_profesores/prof152579/publicaciones/2003_re
- [7] Revisado el 21 de Enero de 2013. National by Science Data Base Center. Licencia CC. Disponible <http://lifesciencedb.jp/bp3d/>
- [8] López-Palop, R., Moreu, J., Fernández-Vázquez, F., & Hernández, R. (2004). Registro español de hemodinámica y cardiología intervencionista. XIII Informe Oficial de la Sección de Hemodinámica y Cardiología Intervencionista de la Sociedad Española de Cardiología (1990-2003). Rev Esp Cardiol, 57(11), 1076-89. Disponible:
<http://www.revespcardiol.org/es/pdf/13068169/S300/>
- [9] Ian Sommerville. Ingeniería del Software. Madrid: Pearson Educación, S.A., 2005
- [10] Stephen Cawood and Pat McGee. Microsoft XNA Game Studio Creator's Guide. United States: McGraw-Hill, 2009

- [11] Eduardo Otero. Riesgos y complicaciones en cirugías cardíacas. Editorial Medica Panamericana, 2005
- [12] Jarret Webb and James Ashley. Beginning Kinect Programming with the Microsoft Kinect SDK. United States: Apress, 2012
- [13] Sean Kean, Jonathan, and Phoenix Perry. Meet the Kinect: An Introduction to Programming Natural User Interfaces. United States: Technology in Action, 2011
- [14] Dr. Eduardo A. Rodriguez Tello, Triangularización de Polígonos, Disponible:
<http://www.tamps.cinvestav.mx/ertello/gc/sesion11.pdf>
- [15] Helena Curtis, N. Sue Barnes: Biología. Editorial Médica Panamericana, 6ta edición en Español, 2009. Disponible:
<http://www.cobach-elr.com/academias/quimicas/biologia/biologia/curtis/libro/c42f.htm>
- [16] Borge, M. J. N. Tema 5. Hemodinámica o física del flujo sanguíneo. Disponible:
<http://ocw.unican.es/ciencias-de-la-salud/fisiologia-humana-2011-g367/material-de-clase/bloque-tematico-1.-fisiologia-del-aparato/>
- [17] Navigating the Fifth Dimension: Innovative Interface for Multidimensional Multimodality Image Navigation. Radiographics. 2006 Jan-Feb;26(1):299-308. Disponible en:
<http://www.ncbi.nlm.nih.gov/pubmed/16418259?dopt=Abstract>
- [18] Zygote Media Group, Inc. Disponible:
<http://www.zygote.com/>
- [19] A correct even-odd algorithm for the point-in-polygon (PIP) problem for complex polygons. Michael Galetzka, Patrick O. Glauner. Disponible:
<http://arxiv.org/pdf/1207.3502v1.pdf>
- [20] A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu. Institute for Computer Science, University of Munich
- [21] Triangulating Point Set Surfaces with Bounded Error, Carlos E. Scheidegger Shachar Fleishman Cláudio T. Silva. University of Utah
- [22] Virtual Realities. Página visitada el 12-Diciembre-2012
<http://www.vrealities.com/glove.html>
- [23] Página oficial de Arduino. Disponible:
<http://arduino.cc/es/>
- [24] Sitio Oficial de CUDA-NVIDIA. http://www.nvidia.es/object/what_is_cuda_new_es.html
- [25] Shannon, Robert; Johannes, James D. (1976). «Systems simulation: the art and science». IEEE Transactions on Systems, Man and Cybernetics 6(10). pp. 723-724.
- [26] Sitio Oficial de Unity. <http://spanish.unity3d.com/unity/>
- [27] Diccionario Mosby - Medicina, Enfermería y Ciencias de la Salud, Ediciones Hancourt, S.A. 1999.