

**HERRAMIENTA SOFTWARE DE VISUALIZACIÓN PARA ASISTIR LA
PERFORACIÓN DEL TÚNEL TIBIAL Y FEMORAL EN RECONSTRUCCIÓN DE
LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO REALIDAD
AUMENTADA**

DUVAN SNEYDER PINZÓN DÍAZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2017

**HERRAMIENTA SOFTWARE DE VISUALIZACIÓN PARA ASISTIR LA
PERFORACIÓN DEL TÚNEL TIBIAL Y FEMORAL EN RECONSTRUCCIÓN DE
LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO REALIDAD
AUMENTADA**

DUVAN SNEYDER PINZÓN DÍAZ

**Trabajo de Grado presentado como requisito para obtener el título de
Ingeniero de Sistemas**

DIRECTOR

SERGIO FERNANDO CASTILLO CASTELBLANCO

Ph.D. Ingeniería de Sistemas

CODIRECTOR

LUIS EDUARDO BAUTISTA ROJAS

MSc. Ingeniería de Sistemas e Informática

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2017

DEDICATORIA

En esta oportunidad quiero agradecer a mi familia, especialmente a mis padres *Ricaurte* y *Layla* que me forjaron desde pequeño con amor y me apoyaron durante todo el proceso académico y personal para lograr cumplir una de las tantas metas que me he planteado en mi vida. Este nuevo logro obtenido también es de ustedes, Muchas gracias por todo lo brindado.

AGRADECIMIENTOS

Gracias a mis padres que fueron los principales artífices de este nuevo triunfo obtenido.

Gracias al director de proyecto *Sergio Fernando Castillo Castelblanco* por su apoyo y confianza en la implementación de este proyecto.

Gracias al codirector de proyecto *Luis Eduardo Bautista Rojas* por sus enseñanzas a lo largo de este proyecto, éstas me serán de gran ayuda en mi vida personal y profesional.

Finalmente quiero agradecer a mi familia, compañeros, amigos y profesores los cuales hicieron parte directa o indirectamente de este logro y que me apoyaron durante todo el ciclo estudiantil.

CONTENIDO

INTRODUCCIÓN	16
1. OBJETIVOS	18
1.1 Objetivo general	18
1.2 Objetivos específicos	18
2. MARCO TEÓRICO	19
2.1 Generalidades	19
2.2 Herramientas.....	24
3. METODOLOGÍA.....	28
3.1 Investigación	29
3.2 Necesidades.....	33
3.3 Diseño	33
3.4 Verificación.....	36
3.5 Desarrollo	38
3.6 Pruebas	39
4. DESARROLLO E IMPLEMENTACIÓN	40
4.1 Herramientas de desarrollo:	40
4.2 Funcionalidades:	43
5. PRUEBAS	66
5.1 Pruebas técnicas de rendimiento.	66
5.2 Pruebas de precisión.....	68
5.3 Pruebas de usuario	73

6. ENTREGABLES.....	75
6.1 Visualización en realidad aumentada.....	75
6.2 Visualización en escritorio.....	76
7. CONCLUSIONES.....	77
8. RECOMENDACIONES.....	79
9. REFERENCIAS BIBLIOGRAFICAS.....	80
BILIOGRAFÍA.....	83
ANEXOS.....	85

ILUSTRACIONES

Ilustración 1 - Estructura general de NAVAR	21
Ilustración 2 - Ilustración proyecto NAVAR	21
Ilustración 3 - Flujo de trabajo de NAVAR	22
Ilustración 4 - Flujo de trabajo módulo de visualización	23
Ilustración 5 - Visualización en realidad aumentada - Prototipo	24
Ilustración 6 - Epson Moverio BT 300[11].	25
Ilustración 7 - Myo Armband[12].	25
Ilustración 8 - Rastreo de HMD y objetos rígidos usando sistema Optitrack[14] ...	26
Ilustración 9 - Componentes NatNet [16]	27
Ilustración 10 - Metodología prototipos evolutivos	28
Ilustración 11 - Arquitectura módulo de visualización	34
Ilustración 12 - Primer prototipo interfaz de usuario.....	34
Ilustración 13 - Perforación en primer prototipo	34
Ilustración 14 - Segundo prototipo Parte A – Información.....	35
Ilustración 15 - Segundo prototipo Parte B - Menú emergente	36
Ilustración 16 - Implementación básica en carga de modelo .obj	45
Ilustración 17 - Implementación básica en carga de archivo .txt.....	46
Ilustración 18 - Carga de modelo .obj (texture, mesh, shader, Collider) en Unity3D	46
Ilustración 19 - Proyección (punto azul) de la broca sobre el modelo.....	47
Ilustración 20 - Diferencia de tamaño entre el Collider y el modelo de la broca	49
Ilustración 21 - Representación broca (azul) y collider (verde)	49
Ilustración 22 - Representación puntos sobre el modelo del fémur	50
Ilustración 23 - Pseudocódigo que controla el color del túnel	51
Ilustración 24 - Pseudocódigo que controla el color de las esferas	51
Ilustración 25 – Retroalimentación puntos de perforación	52

Ilustración 26 - Sistema de orientación	53
Ilustración 27 - Ejemplo estado ubicación(arriba) - estado de perforación (abajo)	55
Ilustración 28 - Plano de calibración - Representación en Unity	56
Ilustración 29 - Ubicación de la broca con respecto al plano de calibración	56
Ilustración 30 - Antes y después de la calibración	57
Ilustración 31 - Puntos teóricos usados para el registro	58
Ilustración 32 - Flujo del registro 3D	58
Ilustración 33 - Pasos del registro de 3 puntos	59
Ilustración 34 - Configuración Streaming	60
Ilustración 35 - Sistema coordenadas (right handed - left handed).....	61
Ilustración 36 - Pseudocódigo matriz de transformación	62
Ilustración 37 - Tratamiento de matriz de rotación	63
Ilustración 38 - Uso de CPU	67
Ilustración 39 - Uso de memoria	68
Ilustración 40 - Ubicación de esferas sobre la entrada y salida del tunel perforado	69
Ilustración 41 - Imagen proyectada en gafas de realidad aumentada.....	75
Ilustración 42 - Imagen proyectada en pantalla	76

TABLAS

Tabla 1 - Análisis de literatura.....	30
Tabla 2 - Documentación técnica.....	31
Tabla 3 - Nivel de cumplimiento - Prototipo 1	37
Tabla 4 - Nivel de cumplimiento - Prototipo 2 (Cambios subrayados)	37
Tabla 5 - Funcionalidades sistema de orientación	53
Tabla 6 - Gestos ligados a acciones	65
Tabla 7 - Resultados - Punto de ingreso fémur.....	70
Tabla 8 - Resultados - Punto de salida fémur	71
Tabla 9 - Resultados - Punto de ingreso tibia	71
Tabla 10 - Resultados - Punto de salida tibia	72
Tabla 11 - Error de distancia medio en fémur usando gafas de realidad aumentada	72
Tabla 12 - Error de distancia medio en tibia usando gafas de realidad aumentada	73
Tabla 13 - Satisfacción uso de gafas RA	74

ANEXOS

Anexo A - Requerimientos del NAVAR-Q	85
Anexo B - Documento de especificación de requerimientos	87
Anexo C - Pruebas de interfaz gráfica realizadas por el equipo de diseño de NAVAR-Q a ortopedistas.....	99
Anexo D - Pruebas de funcionalidad realizadas por ortopedistas.....	101
Anexo E - Encuesta de satisfacción.....	103
Anexo F - Interfaz de usuario.....	105

RESUMEN

TÍTULO: HERRAMIENTA SOFTWARE DE VISUALIZACIÓN PARA ASISTIR LA PERFORACIÓN DEL TÚNEL TIBIAL Y FEMORAL EN RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO REALIDAD AUMENTADA*.

AUTOR: DUVAN SNEYDER PINZÓN DÍAZ**

PALABRAS CLAVE: Cirugía Asistida por Computador (CAS), Realidad Aumentada (RA), Captura de Movimiento (MOCAP), Ligamento Cruzado Anterior (LCA).

DESCRIPCIÓN:

La perforación de los túneles tibial y femoral en la Reconstrucción de Ligamento Cruzado Anterior (RLCA) es el proceso quirúrgico que consiste en perforar dos túneles que servirán para la ubicación y ajuste del injerto a colocar y que sustituirá el Ligamento Cruzado Anterior (LCA) dañado.

El desarrollo de la tecnología ha permitido realizar Cirugías Asistidas por Computadora (CAS), las cuales ofrecen más información al cirujano evitando procedimientos erráticos con el fin de obtener mejores resultados clínicos para el paciente y un mejor desarrollo del postoperatorio. La información ofrecida por estos sistemas de asistencia quirúrgica es de gran ayuda al momento de posicionar y perforar los túneles en la RLCA debido a la necesidad de replicar el movimiento natural de la rodilla evitando el fracaso de la cirugía y la necesidad de una nueva cirugía con más complicaciones.

En este proyecto se desarrolló un módulo de visualización quirúrgica con el fin de otorgar al cirujano ortopedista datos adicionales y de fácil acceso durante la RLCA usando Realidad Aumentada.

El módulo de visualización quirúrgica hace parte del proyecto de investigación interdisciplinario "SISTEMA DE ORIENTACIÓN USANDO REALIDAD AUMENTADA PARA LA PERFORACIÓN DE TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR", financiado por Colciencias y ejecutado por los grupos de investigación FICOMACO (Escuela de Física), GOTS (Escuela de Física), INTERFAZ (Escuela de Diseño Industrial) y GRICES (Escuela de Medicina).

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: PhD Sergio Fernando Castillo Castelblanco, Codirector: Mg Luis Eduardo Bautista Rojas

ABSTRACT

TITLE: VISUALIZATION SOFTWARE TOOL TO ASSIST TIBIAL AND FEMORAL TUNNEL DRILLING IN ANTERIOR CRUCIATE LIGAMENT RECONSTRUCTION (ACL RECONSTRUCTION) USING AUGMENTED REALITY*.

AUTHOR: DUVAN SNEYDER PINZÓN DÍAZ**

KEYWORDS: Cirugía Asistida por Computador (CAS), Realidad Aumentada (RA), Captura de Movimiento (MOCAP), Ligamento Cruzado Anterior (LCA).

DESCRIPTION:

Tibial and femoral tunnel drilling in Anterior Cruciate Ligament Reconstruction is the surgical process of drilling two tunnels that will serve to locate and adjust the graft to be placed and which will replace the damaged Anterior Cruciate Ligament(ACL).

The development of the technology has allowed to perform Computer Assisted Surgeries (CAS), which offer more information to the surgeon avoiding erratic procedures to obtain better clinical results for the patient and a better postoperative development. The information offered by these surgical assistance systems is of great help in positioning and drilling the tunnels in the ACL Reconstruction due to the need to replicate the natural movement of the knee avoiding the failure of the surgery and the need for a new surgery with more complications.

In this project was developed a surgical visualization module to give the orthopedic surgeon additional and easily accessible data during the ACL Reconstruction using Augmented Reality.

The surgical visualization module is part of the interdisciplinary research project "GUIDANCE SYSTEM USING AUGMENTED REALITY FOR TIBIAL AND FEMORAL TUNNEL DRILLING IN ANTERIOR CRUCIATE LIGAMENT RECONSTRUCTION ", financed by Colciencias and executed by the research groups FICOMACO (School of Physics), GOTS (School of Physics), INTERFAZ (School of Industrial Design) and GRICES (School of Medicine).

* Bachelor thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: PhD Sergio Fernando Castillo Castelblanco, Codirector: Mg Luis Eduardo Bautista Rojas

TABLA DE CUMPLIMIENTO DE OBJETIVOS

Objetivos específicos	
Objetivo	Cumplimiento
<ul style="list-style-type: none">Implementar el módulo de visualización de realidad aumentada para las gafas Epson Moverio BT300 usando el entorno de desarrollo Unity3D.	Capítulo “4. DESARROLLO E IMPLEMENTACIÓN”
<ul style="list-style-type: none">Diseñar el módulo de visualización teniendo en cuenta las necesidades expuestas en el documento de especificación de requerimientos.	Capítulo “3.3 Diseño”
<ul style="list-style-type: none">Llevar a cabo un plan de pruebas técnicas con el fin de verificar el cumplimiento de las necesidades expuestas en el documento de especificación de requerimientos.	Capítulo “5. PRUEBAS”

INTRODUCCIÓN

La cirugía asistida por ordenador (CAS, Computer Assisted Surgery, en inglés) representa una ayuda para el cirujano, permitiéndole realizar procedimientos más complejos, lo cual es fundamental al tratarse de cirugía mínimamente invasiva. La retroalimentación visual que ofrece la CAS consiste en la representación de detalles y zonas que se encuentran invisibles al ojo humano, esto se suma a la percepción táctil del cirujano y a su destreza motora, con el fin de tener toda la información estratégica, operativa y anatómica del paciente al momento de realizarse la intervención[1].

La RLCA es un procedimiento quirúrgico que se realiza cuando el Ligamento Cruzado Anterior (LCA) sufre una ruptura o desgarro comprometiendo la movilidad natural de la rodilla e impidiendo su correcto comportamiento dinámico. Se estima que un gran porcentaje de las RLCA son fallidas a causa de perforaciones imprecisas de los túneles tibial y femoral, por donde se ubicará el injerto[2]. En el proyecto de investigación “SISTEMA DE ORIENTACIÓN USANDO REALIDAD AUMENTADA PARA LA PERFORACIÓN DE TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR”, se desarrolló un sistema que asiste la realización de los túneles, usando realidad aumentada para proporcionar información relevante del paciente y del procedimiento, con el fin de mejorar la precisión de esta actividad. La Realidad Aumentada aplicada en el área de la medicina se centra en ofrecer datos visuales adicionales usando el mismo espacio físico que ocupan el cirujano y el paciente, esto implica diferentes ventajas al momento de realizar la cirugía como permitir al cirujano mantener una línea de visualización directa sobre el campo quirúrgico, facilidad al momento de usar el instrumental quirúrgico, visualización de las estructuras anatómicas en tiempo real, entre otras[3].

El proyecto de investigación se encarga de asistir al cirujano durante la perforación de túneles tibial y femoral en la Reconstrucción de Ligamento Cruzado Anterior proporcionándole la información necesaria del paciente y de la preplaneación quirúrgica, mostrando estructuras anatómicas ocultas a la vista del ortopedista, en tiempo real durante la intervención. Este proyecto requirió el diseño y desarrollo de una guía visual para el usuario cirujano usando realidad aumentada. La guía visual de realidad aumentada requerida se diseñó y desarrollo en dos partes: Primero, la interfaz de usuario creada en el trabajo de grado de diseño industrial titulado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA”, donde se diseñaron los aspectos visuales y de interacción para garantizar la utilidad de la guía. Segundo, Un módulo de visualización compatible con la arquitectura y flujo de trabajo del proyecto de investigación, este último es el trabajo descrito y desarrollado en el presente trabajo de grado.

En este proyecto, se desarrolló un módulo de visualización usando realidad aumentada en un Head Mounted Display (HMD) y una visualización de escritorio sobre una pantalla que sirve como sustituto al HMD en caso de que no esté disponible o el usuario no se sienta cómodo con su uso y que usa los mismos principios y funcionalidades del software de realidad aumentada. Este módulo permite cargar información de configuración por defecto y guiar la perforación de túneles por medio de una visualización en tiempo real de instrumental quirúrgico, estructuras anatómicas e información del paciente usando una interfaz de usuario.

El desarrollo de este módulo se realizó usando el entorno Unity3D y concluyó con pruebas de rendimiento, precisión y de usuario en donde se afirmó la estabilidad, robustez, satisfacción y correcta funcionalidad del software desarrollado en este trabajo de grado.

1. OBJETIVOS

1.1 Objetivo general

Realizar un módulo de visualización de apoyo al cirujano ortopedista que proporcione información necesaria para guiar la perforación de los túneles tibial y femoral en la intervención quirúrgica para la reconstrucción de ligamento cruzado anterior (RLCA) usando realidad aumentada.

1.2 Objetivos específicos

- Implementar el módulo de visualización de realidad aumentada para las gafas Epson Moverio BT300 usando el entorno de desarrollo Unity3D.
- Diseñar el módulo de visualización teniendo en cuenta las necesidades expuestas en el documento de especificación de requerimientos.
- Llevar a cabo un plan de pruebas técnicas con el fin de verificar el cumplimiento de las necesidades expuestas en el documento de especificación de requerimientos.

2. MARCO TEÓRICO

2.1 Generalidades

2.1.1 Cirugía ortopédica asistida por computadora. La cirugía asistida por ordenador (CAS, Computer Assisted Surgery) representa una ayuda para el cirujano, permitiéndole realizar procedimientos más complejos, lo cual es fundamental al tratarse de cirugía mínimamente invasiva[1]. El uso de la navegación por computadora en la cirugía ortopédica permite la retroalimentación intraoperatoria en tiempo real, lo que da como resultado una mayor precisión de los cortes óseos, una mejor alineación de los implantes y extremidades, menos radiación y la mejor documentación posible. No hay necesidad de visualización directa y repetida de muchos puntos de referencia anatómicos (método clásico) para tener una buena orientación intraoperatoria[4].

2.1.2 Ligamento Cruzado Anterior. Un ligamento es una banda de tejido conjuntivo fibroso muy sólido y elástico que une los huesos entre ellos en el seno de una articulación. El ligamento permite el movimiento, pero evita también mover los huesos de modo excesivo lo que previene las luxaciones en caso de movimientos forzados. El LCA es un ligamento intraarticular cuya principal función es dar estabilidad anteroposterior y rotacional, evitando el movimiento de la tibia hacia adelante del fémur[5].

2.1.3 Rotura de Ligamento Cruzado Anterior. La rotura del ligamento cruzado anterior (LCA) de la rodilla es una lesión ligamentosa frecuente, especialmente en la práctica deportiva, siendo el mecanismo más común de lesión el trauma indirecto,

donde habitualmente están involucradas fuerzas de desaceleración, hiperextensión y rotación. Este tipo de lesión se diagnóstica por diferentes vías como la resonancia magnética o pruebas diagnósticas físicas como el Pivot Shift o el Test de Lachman, entre otras[6].

2.1.4 Reconstrucción de Ligamento Cruzado Anterior. El objetivo principal de la cirugía es dar estabilidad anteroposterior y rotacional de la rodilla, disminuyendo complicaciones futuras como el desgaste temprano de la articulación. Además, los meniscos y el cartílago están generalmente dañados cuando el ligamento está roto, los cuales se deben reparar en el mismo acto quirúrgico. El ligamento NO debe reconstruirse en la fase aguda y se debe esperar entre 4-6 semanas a que la inflamación haya bajado, el dolor sea poco y la movilidad esté completa.

Consiste en reemplazar el ligamento nativo dañado por uno nuevo. Para reemplazarlo utilizamos tendones propios del paciente (autoinjertos) como el tendón patelar, los isquiotibiales o el tendón del cuádriceps, con o sin una pastilla o taco de hueso pegada el tendón. En algunos casos especiales podemos utilizar estos mismos tendones, pero provenientes del banco de huesos (aloinjertos).

Casi todas las técnicas utilizan el mismo principio que consiste en realizar un túnel en el fémur y en la tibia, a través de los cuales se pasa el injerto, fijándolo usualmente con tornillos o con anclajes a distancia[7].

2.1.5 Proyecto “SISTEMA DE ORIENTACIÓN USANDO REALIDAD AUMENTADA PARA LA PERFORACIÓN DE TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR” – NAVAR (Navegación quirúrgica con Realidad Aumentada). El sistema de realidad aumentada desarrollado en el proyecto tiene dos grandes componentes, como se muestra en la Ilustración 1 y la Ilustración 2: El primero, una herramienta digital de pre-planeación quirúrgica, conformado por un sistema de información que le

permitirá al médico ortopedista, realizar actividades de pre-planeación en un ambiente web que puede ser consultado vía PC o dispositivo móvil. El segundo componente, un sistema de asistencia quirúrgica en el quirófano, compuesto por apoyos digitales. El apoyo digital, se realiza con la integración de tres módulos de software: el módulo de visión y rastreo, el módulo de registro 3D y el módulo de visualización de RA. El módulo de visión y rastreo usa cámaras infrarrojas OptiTrack Prime13 para el seguimiento de: la extremidad del paciente, el instrumental quirúrgico y el dispositivo visualización de realidad aumentada. El módulo de registro que realiza la adecuación espacial bajo el mismo sistema de coordenadas, y el módulo de visualización de RA, objeto del presente trabajo.

Ilustración 1 - Estructura general de NAVAR

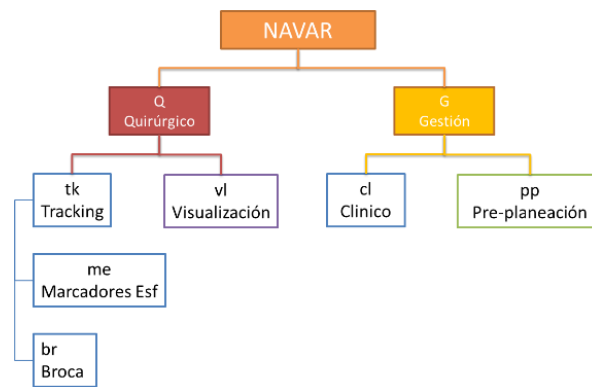
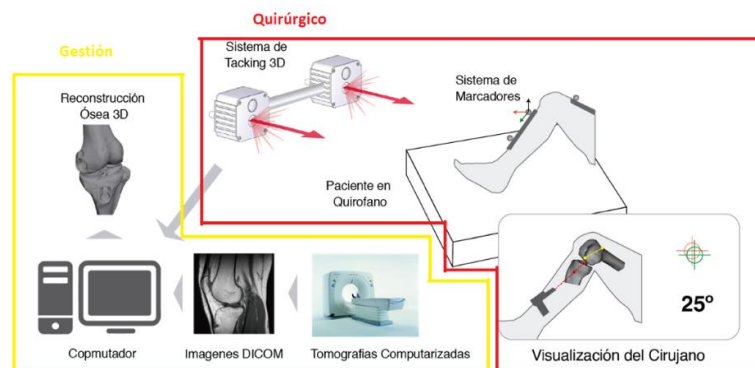


Ilustración 2 - Ilustración proyecto NAVAR



El desarrollo de este trabajo de grado se concentró en el desarrollo del módulo VL- Visualización del componente quirúrgico del proyecto NAVAR. Este módulo recibe datos de la preplaneación quirúrgica como: modelos 3D de la anatomía del paciente, la información necesaria que debe ser mostrada en el quirófano durante la cirugía, entre otras. Adicionalmente, este módulo está ligado al seguimiento realizado por las cámaras infrarrojas, las cuales por medio de un stream de datos enviarán la ubicación de los objetos rastreados en tiempo real hacia el HMD, y éste interpretará los datos para ofrecer una correcta visualización estereoscópica de alta precisión. En la Ilustración 3 y la Ilustración 4 se muestra el flujo de trabajo del proyecto de investigación y el del módulo de visualización respectivamente.

Adicionalmente este módulo se apoya del proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA” a cargo de las estudiantes SARA MARITZA GUTIÉRREZ RONDÓN y ANDREA PAOLA PINZÓN BARBOSA del grupo de investigación INTERFAZ vinculado a la escuela de Diseño Industrial, en el cual se diseñaron las funcionalidades e interfaz gráfica de usuario a usar en el módulo de visualización.

Ilustración 3 - Flujo de trabajo de NAVAR

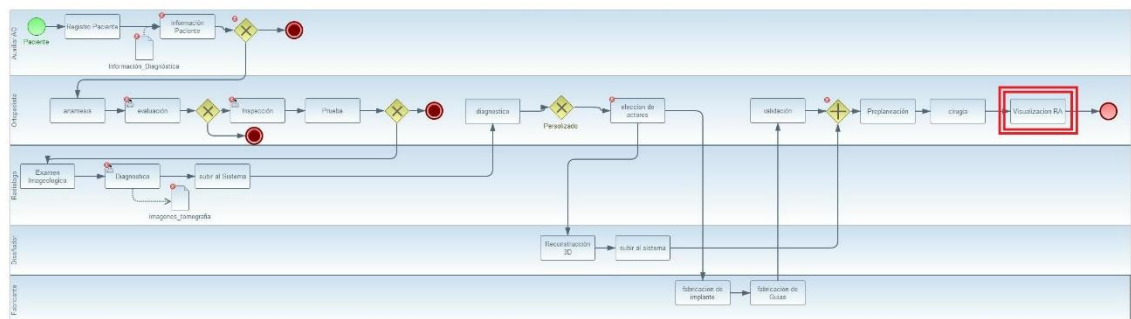
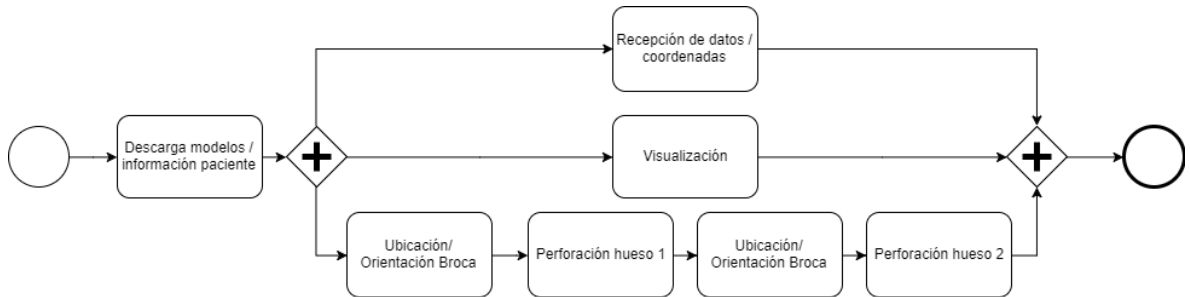


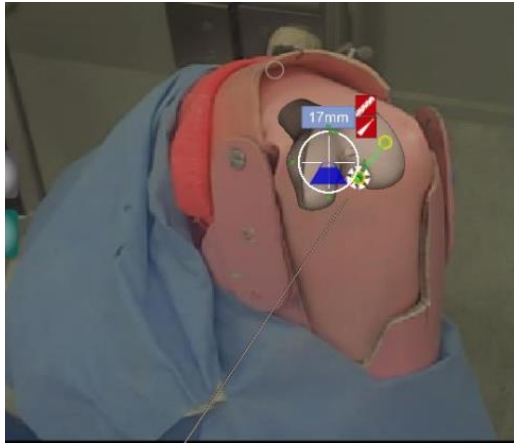
Ilustración 4 - Flujo de trabajo módulo de visualización



2.1.6 Realidad Aumentada. En el término de realidad aumentada se agrupan aquellas tecnologías que permiten la superposición, en tiempo real, de imágenes, marcadores o información generados virtualmente sobre imágenes del mundo real. Se crea de esta manera un entorno en el que la información y los objetos virtuales se fusionan con los objetos reales, ofreciendo una experiencia tal para el usuario, que puede llegar a pensar que forma parte de su realidad cotidiana, olvidando incluso la tecnología que le da soporte[8].

2.1.7 Visualización en Realidad Aumentada. Dentro de los sistemas de realidad aumentada, la visualización de la escena real con la información de aumento (Ilustración 5) es de los procesos más importantes sin este proceso la realidad aumentada no tendría razón de ser. Debido a las mejoras en las prestaciones del hardware que ofrecen los dispositivos, generan la posibilidad de usar modelos en realidad aumentada de mayor calidad visual con el fin de asemejar más su forma y textura al entorno real, haciendo así que los dos escenarios se mezclen de una forma más limpia; estos avances en hardware han permitido incluso la interacción con los objetos virtuales por medio de objetos reales, como las manos del usuario[9].

Ilustración 5 - Visualización en realidad aumentada - Prototipo



2.1.8 Rastreo y seguimiento (Captura de movimiento). El rastreo de objetos rígidos se realiza principalmente por medio de marcadores reflectivos, cámaras infrarrojas y un software como Motive[10] desarrollado por NaturalPoint que permite la creación de objetos digitales que representarán los objetos reales rastreados por medio de un algoritmo de rastreo normalmente basado en la distancia entre los marcadores reflectivos.

2.2 Herramientas

2.2.1 Epson ® Moverio BT300. Las gafas Epson ® Moverio BT300 (Ilustración 6) son unas gafas de realidad aumentada (HMD – Head Mounted Display), las cuales funcionan con un display ubicado sobre un lente transparente, dando la posibilidad de agregar objetos virtuales a la vista real, adicionalmente vienen dotadas de giroscopio, acelerómetro y una unidad de procesamiento dedicada con sistema operativo Android[11].

Ilustración 6 - Epson Moverio BT 300[11].



2.2.2 Myo Armband. Myo es una banda inalámbrica (Ilustración 7) que detecta la actividad eléctrica de los músculos del antebrazo y así ofrece control sobre dispositivos electrónicos como computadores, tablets, celulares, entre otros[12].

La banda Myo contiene un SDK que permite la asignación de gestos específicos a acciones o funciones. Es compatible con los sistemas operativos Windows, Mac, IOS y Android, adicionalmente viene con un paquete de desarrollo para usarse en el entorno de desarrollo Unity3D.

Ilustración 7 - Myo Armband[12].



2.2.3 Sistema Optitrack. El sistema Optitrack es un conjunto de cámaras y marcadores reflectivos, los cuales son rastreados por las cámaras infrarrojas, teniendo en cuenta la posición del centroide de estos marcadores y las distancias

entre ellos se puede generar un modelo o esqueleto virtual (objeto rígido) el cual puede ser animado en tiempo real por los marcadores a medida que el objeto físico se mueve.

Este sistema tiene diversas aplicaciones dentro del campo académico y del entretenimiento debido a que este rastreo es muy usado para animar personajes en videojuegos o películas como se muestra en la Ilustración 8. [13].

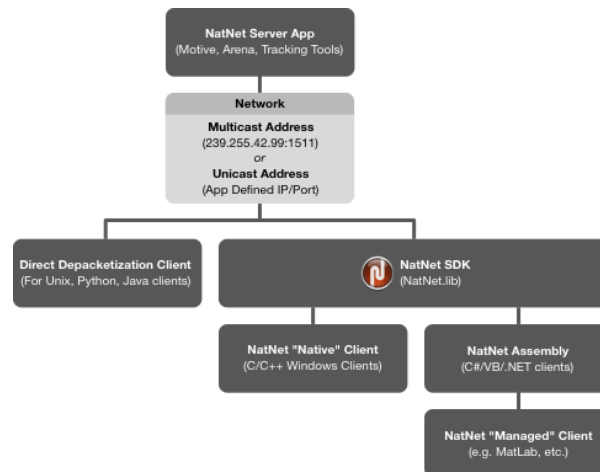
Ilustración 8 - Rastreo de HMD y objetos rígidos usando sistema Optitrack[14]



2.2.4 NatNet. NatNet es un SDK que contiene librerías para enviar y recibir (cliente/servidor) información de NaturalPoint® a través de la red. NatNet usa el protocolo UDP en conjunto con Point-To-Point Unicast o IP Multicast para enviar información. Los servidores y clientes de NatNet pueden estar en la misma o en diferentes máquinas[15].

NatNet empaqueta los datos de los objetos rígidos en formato XML y debido a su sencillez las librerías pueden ser fácilmente utilizadas en cualquier lenguaje de programación. En la Ilustración 9 se detallan los componentes de NatNet.

Ilustración 9 - Componentes NatNet [16]



2.2.5 Unity3D. Es un motor 3D con versión Personal (Free) y Professional (Premium) principalmente usado en el desarrollo de videojuegos debido a su facilidad de Scripting y manejo de modelos 3D. Adicionalmente Unity es el motor más usado al momento de desarrollar realidad aumentada debido a sus comodidades de renderizado y permitir compilar proyectos en diferentes plataformas como WebGL, Android, Windows, XBOX ONE, entre otras.

3. METODOLOGÍA

Para el desarrollo del módulo de visualización se realizó de manera conjunta y paralela con el proyecto que diseñó la interfaz de usuario y se siguió una metodología conjunta basada en el prototipado, con el fin de llevar a cabo una rápida ejecución del proyecto y un seguimiento frecuente de los requerimientos del sistema, generando así un resultado más sólido y adaptado a las necesidades generales del proyecto de investigación.

En la Ilustración 10, se detallan las fases que se llevaron a cabo durante la ejecución del proyecto, donde adicionalmente se aprecia la iteración que surge en la fase de “Verificación”, esto es debido a que se realizaron diferentes pruebas de usuario en donde se validaron las necesidades y el diseño planteado en el prototipo con el fin de obtener la configuración y las herramientas más adecuadas para el usuario final antes de empezar el desarrollo.

Ilustración 10 - Metodología prototipos evolutivos



A continuación, se detalla cada una de las fases y los entregables que surgieron al final de cada fase.

3.1 Investigación

Se realizó un análisis detallado de la literatura correspondiente a los conceptos generales tales como anatomía, realidad aumentada, visualización, entre otros. Esta investigación se basó principalmente en artículos científicos consultados en las librerías digitales de ASSOCIATION FOR COMPUTING MACHINERY (ACM) e INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE) con el fin de obtener información de investigaciones ya desarrolladas usando estos conceptos y adicionalmente tener claro el estado del arte de proyectos usando CAS y Realidad aumentada.

La investigación se realizó en base a las siguientes palabras clave:

- Augmented Reality
- Motion Capture
- Visualization
- Simulation
- Head Mounted Display
- 3D
- Rendering
- Computer Assisted
- Surgery
- Architecture

Teniendo en cuenta las palabras clave, se estableció la siguiente ecuación de búsqueda:

((Augmented reality or motion capture or computer Assisted or Head Mounted Display) and (visualization or simulation or 3D or rendering or architecture) and Surgery)

Con los resultados arrojados en la búsqueda, se seleccionó los artículos cuyo nombre y resumen presentaban información relevante relacionada con el proyecto a desarrollar. Una vez seleccionada la información, se analizaron las herramientas software usadas y su implementación en el campo quirúrgico, con el fin de extraer posibles herramientas y arquitecturas a usar durante el desarrollo.

Los artículos más influyentes y la información extraída de ellos se listan en la Tabla 1:

Tabla 1 - Análisis de literatura

Título	Autores	Información extraída
Understanding Motion Capture for Computer Animation	Menache, Alberto	<ul style="list-style-type: none"> • Usos de la captura de movimiento. • Funcionamiento de la captura de movimiento
Augmented reality in laparoscopic surgical oncology	Nicolau, Stephane Soler, Luc Mutter, Didier Marescaux, Jacques	<ul style="list-style-type: none"> • Estado del arte de la realidad aumentada en el quirófano. • Visión estereoscópica. • Realidad aumentada junto a captura de movimiento.
3-D augmented reality for MRI-guided surgery using integral videography autostereoscopic image overlay	Liao, Hongen Inomata, Takashi Sakuma, Ichiro Dohi, Takeyoshi	<ul style="list-style-type: none"> • Visualización de estructuras anatómicas usando realidad aumentada. • Realidad aumentada junto a captura de movimiento.
Computer assisted orthopaedic surgery -- CAOS.	Kanlić, Enes M. Delarosa, Fabian Pirela-Cruz, Miguel	<ul style="list-style-type: none"> • Información necesaria durante cirugías ortopédicas
Advanced medical displays: A literature review of augmented reality	Sielhorst, Tobias Feuerstein, Marco Navab, Nassir	<ul style="list-style-type: none"> • Uso de Head Mounted Displays para visualización anatómica en cirugías
A-me: Augmented Memories	Puig, Jordi Perkis, Andrew Hoel, Aud Sissel Cassinelli, Alvaro	<ul style="list-style-type: none"> • Visualización de estructuras anatómicas usando realidad aumentada
Augmented Reality Visualization for Laparoscopic Surgery	Fuchs, Henry Livingston, Mark a Raskar, Ramesh Colucci, D State, Andrei Crawford, Jessica R Rademacher, Paul Drake, Samuel H Meyer, Anthony a	<ul style="list-style-type: none"> • Configuración del hardware usando Head Mounted Displays • conceptos básicos de Head Mounted Displays • Arquitectura de Hardware usando realidad aumentada
Enhancing Collaboration on a Cloud-Based CVE for Supporting Surgical Education	Paiva, Paulo V F Machado, Liliane Dos S Valenca, Ana Maria G De Moraes, Ronei M. Batista, Thiago V V	<ul style="list-style-type: none"> • Arquitectura de software dentro de entorno quirúrgico • Entorno virtual en entorno quirúrgico

A Virtual Environment for Training and Assessment of Surgical Teams	Paiva, Paulo Vinicius De Farias Machado, Liliane S. Valenca, Ana Maria Gondim	<ul style="list-style-type: none"> • Manejo de la arquitectura de software orientado a realidad virtual
Input System Interface for Image-guided Surgery based on Augmented Reality	Martins, Silvino Vairinhos, Mário Eliseu, Sérgio Borgerson, Janet	<ul style="list-style-type: none"> • Principal referencia de arquitectura para el módulo de visualización • Uso de realidad aumentada y HMD en quirófano

Durante esta fase también se hizo una revisión de la documentación de desarrollo de las diferentes herramientas a usar tales como: el SDK de Android [17], SDK Epson ® BT300 [11][18], Unity3D [19], NatNet [16], entre otras; Esto con el fin de ligar cada concepto con una posible funcionalidad ofrecida por las herramientas a usar durante el proyecto. La información se investigó principalmente en la web de desarrolladores de cada herramienta y se consignó en la Tabla 2, adicionalmente se realizaron consultas contantes a lo largo del proyecto directamente en las páginas de los proveedores

Tabla 2 - Documentación técnica

Dispositivo/Software	Información extraída
Epson Moverio BT300 [11], [18], [20]	Especificaciones: <ul style="list-style-type: none"> • Resolución: 1280x720 pixeles • Aspecto: 16:9 • FOV: 23° Aproximadamente • Frecuencia de actualización: 30Hz • Sistema Operativo: Android 5.1 • Wifi: IEEE802.11 – Wifi Direct – Wifi MiraCast • Bluetooth: Bluetooth Smart Ready • USB: Micro USB 2.0 • CPU: Intel® Atom™ x5, 1.44GHz Quad Core • RAM: 2GB • Memoria interna: 16GB • Memoria externa: Máximo 32 GB • Sensores: GPS, sensor geomagnético, acelerómetro, giroscopio, sensor de iluminación. • Funciones: Bloqueo teclas, Cambio de luminosidad, 2D/3D. • Batería: 6 Horas de autonomía

Myo Armband [12]	<p>Especificaciones:</p> <ul style="list-style-type: none"> • Sensores: sensor EMG, giroscopio, acelerómetro, sensor geomagnético. • CPU: ARM Cortex M4 Processor • Retroalimentación: Vibración, luz LED • Bluetooth: Bluetooth® Smart Wireless Technology <p>Compatibilidad:</p> <ul style="list-style-type: none"> • Android, Windows, Mac, iOS <p>SDK:</p> <ul style="list-style-type: none"> • Attach() • Pose() • Vibrate()
Optitrack Prime 13 [21]	<p>Especificaciones:</p> <ul style="list-style-type: none"> • Resolución: 1280x1024 • FPS: 240 • FOV: 42°, 56° • Latencia: 4.2 ms
NatNet [16]	<p>Funcionalidades:</p> <ul style="list-style-type: none"> • Compatibilidad con diferentes softwares y lenguajes • Arquitectura Cliente-Servidor • Proyectos básicos de ejemplo y traducibles a cualquier lenguaje • Desempaquetado de archivos (XML) transmitidos por Streaming
Motive[10]	<p>Funcionalidades:</p> <ul style="list-style-type: none"> • Creación de objetos rígidos • Rastreo de objetos rígidos por medio de esferas reflectivas (marcadores) • Identificación de objetos rígidos • Stream de datos a servidor • Calibración • Compatibilidad con librería NatNet • Manejo de cámaras infrarrojas de captura de movimiento
Unity3D [22]	<p>Funcionalidades</p> <ul style="list-style-type: none"> • Compilación a multiplataforma (Windows, Android, WebGL, entre otras) • Compatibilidad Windows y Mac • Posibilidad de Scripting • Scripting por medio de C# o JavaScript • Compatibilidad con librería NatNet • Colliders, Textures, Shaders, Canvas, entre otros.

3.2 Necesidades

En esta fase se identificaron las necesidades del módulo de visualización teniendo en cuenta la información extraída en la fase de investigación como el flujo de trabajo de los usuarios, las necesidades del ambiente quirúrgico, el estado del arte, entre otras. Adicionalmente se usaron como referencia las necesidades ya extraídas en el proyecto de investigación “SISTEMA DE ORIENTACIÓN USANDO REALIDAD AUMENTADA PARA LA PERFORACIÓN DE TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR” (Anexo A), los requerimientos de la interfaz de usuario extraídos del proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA” y los objetivos específicos plasmados en el plan de proyecto.

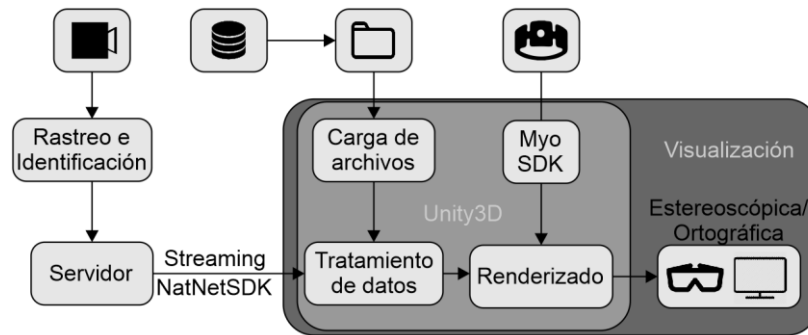
Teniendo en cuenta estas necesidades y enfocándolas al módulo de visualización se realizó el levantamiento de requerimientos del módulo (Anexo B), donde se especifican los requerimientos a cumplir al finalizar el proyecto, los actores y las funcionalidades del módulo.

3.3 Diseño

En esta fase se diseñó la arquitectura de software teniendo en cuenta el documento de especificación de requerimientos y la arquitectura establecida en el módulo general “NAVAR-Q”, con el fin de lograr una integración completa y óptima con el proyecto de investigación general que comprende la gestión, preplaneación y asistencia quirúrgica.

Esta arquitectura se plasmó en el documento de especificación de requerimientos (Anexo B) y se muestra en la Ilustración 11.

Ilustración 11 - Arquitectura módulo de visualización



El prototipo de la interfaz de usuario fue desarrollado en el proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA” donde a lo largo de las iteraciones se desarrollaron y evaluaron con usuarios dos prototipos de interfaz que se explican a continuación:

Ilustración 12 - Primer prototipo interfaz de usuario

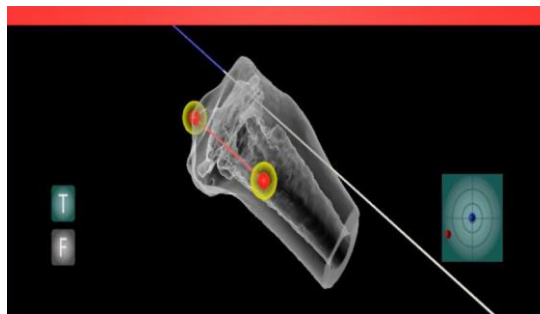
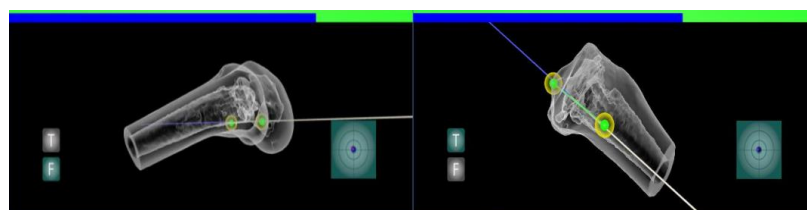


Ilustración 13 - Perforación en primer prototipo



En el primer prototipo (Ilustración 12 e Ilustración 13) se observa la base de la información que debe estar presente durante la perforación, que incluye:

- Retroalimentación visual de la correcta posición de la broca por medio de colores.
- Zona de seguridad alrededor del punto de perforación.
- Botones de cambio de visualización entre tibia y fémur.
- Orientación tipo radar para la correcta angulación hacia el punto de salida
- Avance de la broca durante la perforación por medio de una barra.
- Transparencia en los modelos para la correcta visualización del túnel
- Guía visual de trayectoria de la broca

Este primer prototipo fue mejorado y evaluado con nuevas funcionalidades y actualizaciones a las ya implementadas, como se explicará en el capítulo “4. DESARROLLO E IMPLEMENTACIÓN”. En la Ilustración 14 y la Ilustración 15 se muestra el segundo y último prototipo diseñado.

Ilustración 14 - Segundo prototipo Parte A – Información

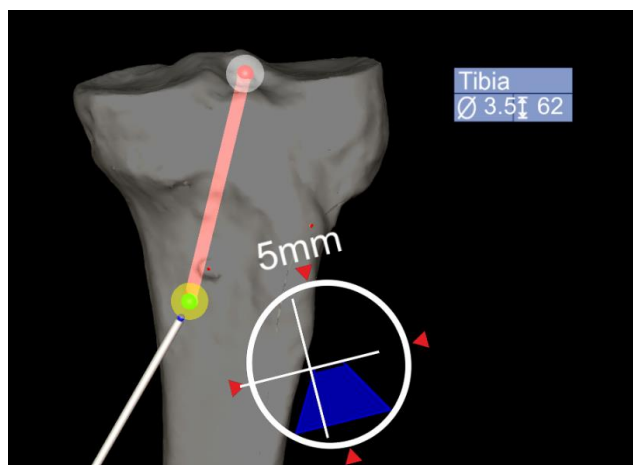
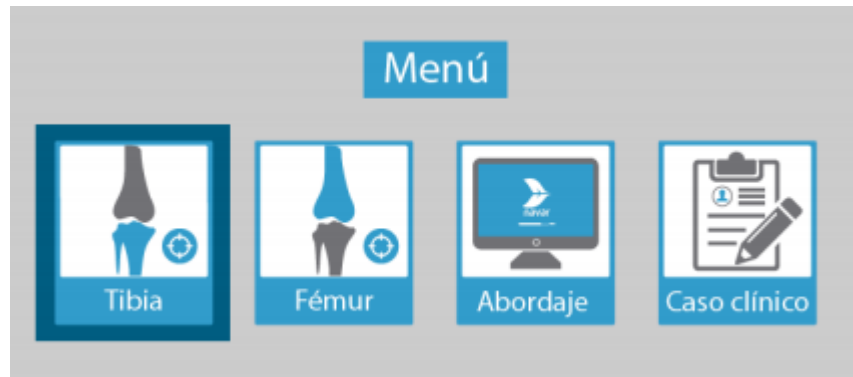


Ilustración 15 - Segundo prototipo Parte B - Menú emergente



3.4 Verificación

La fase de verificación tiene como objetivo verificar el cumplimiento de los requisitos con el diseño planteado en la Fase 2, durante esta fase se realizaron pruebas de integración del sistema haciendo uso del Streaming de datos desde la interfaz de administración hacia el módulo de visualización usando la librería NatNet SDK, Adicionalmente se realizaron pruebas de interacción con usuarios guiados por el equipo de diseño de "NAVAR-Q" con el fin de encontrar puntos débiles en la interacción del usuario con el sistema y reestructurar o añadir información necesaria para el cirujano al momento de realizar una Reconstrucción de Ligamento Cruzado Anterior(Anexo C).

Con la información obtenida por estas pruebas y el listado de requerimientos cumplidos (Tabla 3) se evidenció que el primer prototipo cumplía algunos requerimientos con un bajo nivel de cumplimiento, por lo tanto, se realizó una segunda iteración con un nuevo prototipo y los cambios se evidenciaron en la Tabla 4.

Tabla 3 - Nivel de cumplimiento - Prototipo 1

Requerimiento	Descripción	Cumplimiento
FSR1	El módulo debe seguir un flujo de trabajo adaptado al procedimiento de RLCA.	Alto
FSR2	El módulo debe ofrecer una ubicación registrada en 3D de los modelos sobre la anatomía del paciente	Alto
FSR3	El módulo debe permitir la visualización de herramientas e información que intervengan durante el procedimiento.	Medio
FSR4	El módulo debe mostrar el estado en que el sistema está trabajando.	Bajo
FSR5	El módulo debe mostrar los puntos de perforación sobre la anatomía del paciente.	Alto
FSR6	El módulo debe mostrar el avance en la perforación.	Bajo
FSR7	El módulo debe ofrecer al usuario guías visuales al momento de ubicar el pin guía sobre la rodilla del paciente.	Bajo
FSR8	El módulo debe permitir la visualización de posibles desvíos o inconsistencias que se generen durante la perforación.	Medio
FSR9	El módulo debe servir de cliente en la recepción de coordenadas e información enviada vía Streaming por el módulo de Administración.	Alto
FSR10	El módulo debe estar en capacidad de consultar archivos e información de configuración descargados de una base de datos.	Alto
FSR11	El módulo debe ofrecer al usuario la capacidad de seleccionar la información a mostrar.	Medio
FSR12	El módulo debe ofrecer al usuario la posibilidad de escoger cual hueso va a perforar primero.	Alto
FSR13	El módulo debe poder ser visualizado a través del dispositivo de realidad aumentada o una aplicación de escritorio.	Medio

Tabla 4 - Nivel de cumplimiento - Prototipo 2 (Cambios subrayados)

Requerimiento	Descripción	Cumplimiento
FSR1	El módulo debe seguir un flujo de trabajo adaptado al procedimiento de RLCA.	Alto
FSR2	El módulo debe ofrecer una ubicación registrada en 3D de los modelos sobre la anatomía del paciente	Alto
FSR3	El módulo debe permitir la visualización de herramientas e información que intervengan durante el procedimiento.	<u>Alto</u>

FSR4	El módulo debe mostrar el estado en que el sistema está trabajando.	<u>Alto</u>
FSR5	El módulo debe mostrar los puntos de perforación sobre la anatomía del paciente.	Alto
FSR6	El módulo debe mostrar el avance en la perforación.	<u>Alto</u>
FSR7	El módulo debe ofrecer al usuario guías visuales al momento de ubicar el pin guía sobre la rodilla del paciente.	<u>Alto</u>
FSR8	El módulo debe permitir la visualización de posibles desvíos o inconsistencias que se generen durante la perforación.	<u>Alto</u>
FSR9	El módulo debe servir de cliente en la recepción de coordenadas e información enviada vía Streaming por el módulo de Administración.	Alto
FSR10	El módulo debe estar en capacidad de consultar archivos e información de configuración descargados de una base de datos.	Alto
FSR11	El módulo debe ofrecer al usuario la capacidad de seleccionar la información a mostrar.	<u>Alto</u>
FSR12	El módulo debe ofrecer al usuario la posibilidad de escoger cual hueso va a perforar primero.	Alto
FSR13	El módulo debe poder ser visualizado a través del dispositivo de realidad aumentada o una aplicación de escritorio.	<u>Alto</u>

Se realizaron dos iteraciones durante el desarrollo del módulo. Esto permitió mejorar principalmente la precisión del prototipo y el alto grado de cumplimiento de los objetivos de la herramienta software.

3.5 Desarrollo

Durante esta fase se desarrolló el módulo de visualización teniendo en cuenta el documento de especificación de requerimientos y el prototipo de interfaz validado en la fase de verificación. El proceso de desarrollo se explica con mayor detalle en el apartado “4. DESARROLLO E IMPLEMENTACIÓN”.

3.6 Pruebas

En esta fase se desarrollaron pruebas técnicas, de precisión y de usuario, comprobando la robustez del módulo implementado en el proyecto. Las pruebas se realizaron en el Laboratorio de Simulación de la Facultad de Salud de la Universidad Industrial de Santander, donde cirujanos ortopedistas realizaron el flujo de trabajo completo desde la preplaneación quirúrgica y finalizando con la visualización, en donde se comprobó la capacidad del módulo y sus mejoras para próximas versiones del prototipo.

Las pruebas realizadas y sus resultados son descritos detalladamente en el apartado “5. PRUEBAS”

4. DESARROLLO E IMPLEMENTACIÓN

4.1 Herramientas de desarrollo:

Anteriormente se realizó una investigación (Capítulo 3.1 Investigación) de las posibles herramientas a usar durante el desarrollo del proyecto, estas herramientas se eligieron teniendo en cuenta las necesidades a cumplir, su arquitectura y posible integración con el proyecto de investigación “NAVAR” desarrollado usando C (Lenguaje) y a su vez la sincronía que estas herramientas tenían con el hardware a usar, que incluye el dispositivo de realidad aumentada (Epson® Moverio BT300), las cámaras infrarrojas de captura de movimiento (Optitrack Prime 13) y la banda electromiográfica (Myo Armband) que se encargó de la interacción del usuario con el sistema.

A continuación, se describe las herramientas de desarrollo usadas, sus características, beneficios y la implementación dentro del módulo de visualización en realidad aumentada.

4.1.1 Unity3D. Unity3D es un motor gráfico principalmente enfocado en el desarrollo de videojuegos, sin embargo, es un software compatible y de gran uso al momento de desarrollar aplicaciones en realidad aumentada, ya que tiene distintas características que ayudan al momento de desarrollar una aplicación, algunas de estas características se listan a continuación:

- Diferentes opciones de renderizado 3D.
- Amplia variedad de plataformas destino para la aplicación desarrollada.
- Documentación web completa y de fácil acceso.
- Gran comunidad disponible en foros.

- Facilidad de creación de scripts y librerías de reconocimiento de marcadores, manejo de posición, rotación y escala de objetos.

Además de sus características, la compatibilidad con el Hardware a usar y los requerimientos del sistema era casi completa, por lo tanto, no hubo necesidad de crear librerías extra para desarrollar el módulo y se usaron sus características de compatibilidad para unir las diferentes tecnologías y dispositivos.

Los scripts en Unity3D se pueden escribir usando dos lenguajes conocidos: JavaScript y C#. No existe una diferencia sustancial entre usar uno u otro, las mayores diferencias son del orden de tiempo de compilación (dependiendo de la plataforma destino), librerías (C# puede incluir nativamente bibliotecas de enlace dinámico DLL, sin embargo, éstas también se pueden incluir fácilmente usando JavaScript) y documentación (C# es más usado al momento de desarrollar en Unity3D y contiene documentación disponible por Microsoft ®). Por las librerías nativas, la documentación disponible y teniendo en cuenta que será una aplicación de escritorio ejecutada sobre Windows y Android se eligió C# como lenguaje de programación.

Los scripts desarrollados contienen funcionalidades incluidas en el API de Unity3D, como la detección de colisiones, shaders y renderers.

4.1.2 NatNet SDK. NatNet es un SDK enfocado a la transmisión de datos generados por software y hardware de NaturalPoint (empresa creadora del sistema de cámaras de captura de movimiento Optitrack) a través de la red.

Este SDK trae ejemplos de implementación en diferentes lenguajes entre los que se encuentra C, Python, Matlab entre otros, adicionalmente tiene un ejemplo de cliente y servidor sencillo, esto con el fin de poder traducir a otros lenguajes que NatNet no soporta directamente como Java. La implementación en estos lenguajes es sencilla ya que comprende la importación de las librerías y el uso de las funciones ya establecidas en ellas[15].

El stream de datos depende de un Servidor que se puede crear localmente en el PC anfitrión y se puede enviar información a través de una red local hacia un Cliente en la misma red usando Unicast o Multicast. Adicionalmente se puede recibir información del servidor usando la dirección de loopback (127.0.0.1) con el fin de realizar un stream de datos desde una aplicación a otra estando en un solo computador.

La implementación de NatNet es requerida para el stream de datos de los objetos rígidos capturados por el sistema de captura de movimiento implementado en el software de administración "NAVAR-Q". Este stream de datos se requirió en las primeras fases del proyecto donde se validaban funcionalidades y se corregían aspectos de acoplamiento de objetos rígidos directamente en Unity3D y usando dos computadores conectados a una red local por comodidad a la hora de depurar.

Además de la facilidad de depuración que ofrecía esta herramienta, también fue necesaria implementarla para enviar el stream de datos a la aplicación de visualización alojada en el dispositivo de realidad aumentada (Epson® Moverio BT 300) debido a que es un hardware inalámbrico que contiene su propia unidad de procesamiento que corre un sistema Android 5.1[11] y que sólo tiene acceso a información externa a través de redes inalámbricas como Wifi o Bluetooth.

4.1.3 Myo SDK. La Banda electromiográfica (EMG) Myo trae consigo un SDK que proporciona opciones de conexión, orientación, retroalimentación y traducción de información EMG a gestos predefinidos. Estos gestos permiten principalmente la navegación por la interfaz de usuario y la ejecución de funciones automáticas como el registro 3D y la calibración de la broca. A continuación, se muestran los principales métodos que trae el SDK de Myo:

Conexión:

Attach to adjacent: Esta función permite conectarse al Myo acercando los dos dispositivos mutuamente, normalmente se llama en la función start(), de esta forma se emparejan los dos dispositivos al iniciar la aplicación.

Attach using MAC address: Esta función de conexión escanea en busca de un Myo con una MAC predefinida y de esta forma se emparejan.

Scan: Esta función escanea los dispositivos bluetooth de los alrededores y permite elegir la banda Myo a la que se desee conectar, sin embargo, es necesario el desarrollo de una pantalla o actividad extra para la selección del Myo.

Retroalimentación:

Vibrate: la función Vibrate() permite que la banda Myo vibre por 3 tiempos definidos como SHORT, MEDIUM y LONG, lo cual sirve como una retroalimentación no visual.

Orientación:

Quaternion: debido al IMU que incorpora Myo es posible obtener la rotación del dispositivo en cuaternios y aplicársela a un objeto dentro de la aplicación.

Gestos:

La funcionalidad principal del Myo es la de traducir información EMG y asignársela a objetos de tipo Pose, los gestos incorporados con el SDK son Fist, FingerSpread, WaveOut, WaveIn, DoubleTap.

4.2 Funcionalidades:

4.2.1 Carga de información. Debido a que la escalabilidad y flexibilidad del software es pieza importante en el módulo de visualización, es necesario cargar información dinámicamente de cada paciente, además de archivos básicos de configuración por defecto.

Los archivos necesarios se encuentran localmente, debido a que la interfaz de administración del módulo “NAVAR-Q” hace previamente una descarga total de la información que se requiere durante todo el proceso, la cual es subida a una base de datos por el módulo de preplaneación “NAVAR-G”; por lo tanto, el módulo de visualización debe acceder a archivos locales, ubicados en un directorio separado por casos (pacientes).

Los archivos necesarios que el módulo carga son los siguientes:

- Modelo de fémur reconstruido a partir de tomografía del paciente (*.obj*) (NAVAR-G).
- Modelo de tibia reconstruido a partir de tomografía del paciente (*.obj*) (NAVAR-G).
- Archivo de texto de distancias y diámetros de las perforaciones tibial y femoral (*.txt*) (NAVAR-G).
- Matriz de transformación del modelo de la tibia y el fémur, generados por el registro (*.txt*) (NAVAR-Q).
- Coordenadas de puntos de entrada y salida de las perforaciones tibial y femoral (*.txt*) (NAVAR-G).
- Matriz de IDs de los objetos rígidos(*.txt*) (NAVAR-Q)

Para la carga de archivos se usó un script dedicado a ello, el cual carga la información ubicada en cierto directorio en la función *awake()*, esta función es la primera que se llama al ejecutar una aplicación desarrollada en Unity3D y normalmente se usa para inicialización de variables junto con *start()*.

Con los archivos de texto (*.txt*) no se encontró mayor dificultad, debido a que el lenguaje usado para desarrollar los scripts (C#) contiene la clase *StreamReader* heredada de *System.IO*. Esta clase implementa un *TextReader* que lee los caracteres que se encuentran en un archivo de texto plano como *.txt* o *.csv*[23]. En la Ilustración 17 se muestra la implementación de esta clase.

Para la carga de modelos .obj fue necesario un Script de uso libre desarrollado por un tercero (Hammm) [24] y publicado bajo licencia del MIT(Massachusetts Institute of Technology), el cual carga modelos 3D de formato .obj durante el tiempo de ejecución como lo muestra la Ilustración 16. La lógica base de este script se centra en descomprimir un modelo .obj en polígonos y generar con estos un nuevo UnityEngine.Mesh (maya). Este script fue modificado con el fin de cargar modelos y asignarles una textura predefinida, usar el mesh generado para crear un *Collider* y aplicarle una transparencia al modelo cargado usando un *Shader* para permitir una mejor visualización de los túneles y los puntos de entrada y salida (Ilustración 18) [24].

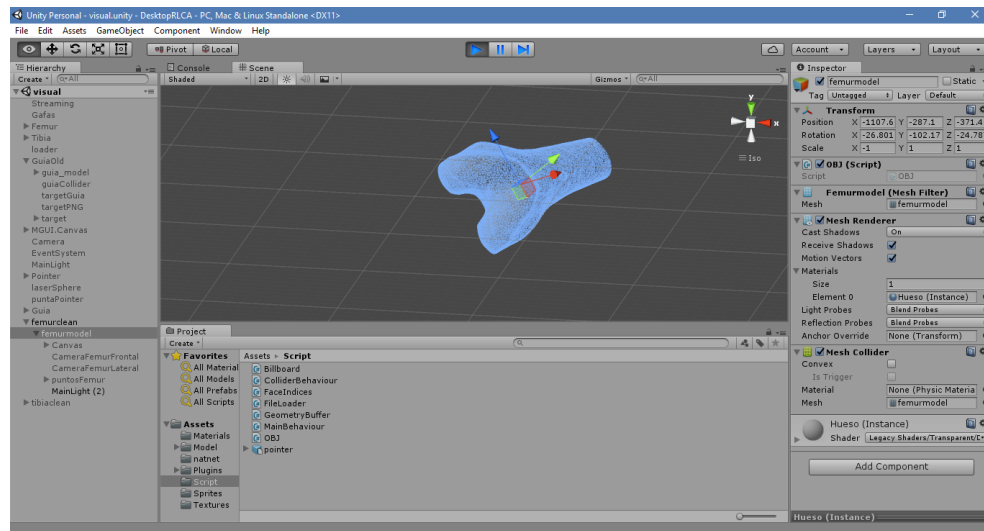
Ilustración 16 - Implementación básica en carga de modelo .obj

```
public class FileLoader : MonoBehaviour {  
  
    OBJ ObjScript1;  
    OBJ ObjScript2;  
  
    // Use this for initialization  
    void Awake()  
    {  
        tibiaModel = GameObject.Find("tibiamodel");  
        femurModel = GameObject.Find("femurmodel");  
        ObjScript1 = GameObject.Find("tibiamodel").GetComponent<OBJ>();  
        ObjScript2 = GameObject.Find("femurmodel").GetComponent<OBJ>();  
        StartCoroutine(ObjScript1.Load("file:/// " + rutaExe + "tibiaFinal.obj"));  
        StartCoroutine(ObjScript2.Load("file:/// " + rutaExe + "femurFinal.obj"));  
    }  
}
```

Ilustración 17 - Implementación básica en carga de archivo .txt

```
public class FileLoader : MonoBehaviour {  
  
    public string[][] matriz = new string[6][];  
  
    // Use this for initialization  
    void Awake()  
    {  
        matriz = readMatriz(rutaExe + "matriz.txt");  
    }  
  
    public string[][] readMatriz(string ruta)  
    {  
        contador = 0;  
        StreamReader stm = new StreamReader(ruta);  
        while (!stm.EndOfStream)  
        {  
            matriz[contador] = stm.ReadLine().Split(',');  
            contador++;  
        }  
  
        stm.Close();  
        return matriz;  
    }  
}
```

Ilustración 18 - Carga de modelo .obj (texture, mesh, shader, Collider) en Unity3D



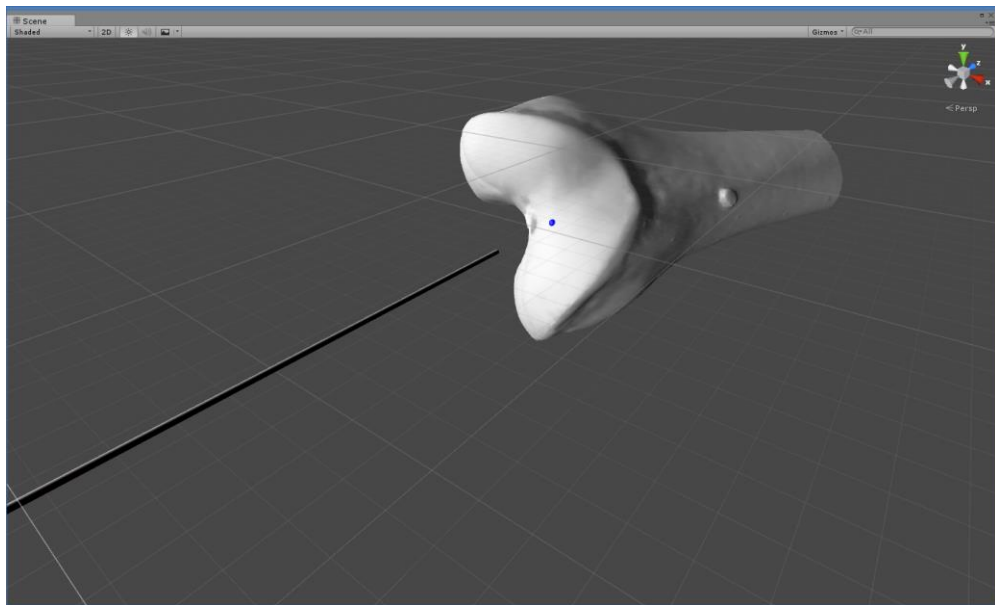
4.2.2 Perforación. El proceso de perforación es la principal actividad en el módulo de visualización y comprende un flujo de trabajo sencillo, sin embargo, fue necesario implementar distintas funcionalidades para brindar al usuario la mayor asistencia posible al momento de perforar los túneles tibial y femoral.

Las funcionalidades implementadas se listan y explican a continuación:

4.2.2.1 Proyección broca. Debido a que la cirugía de Reconstrucción de Ligamento Cruzado Anterior es una intervención quirúrgica mínimamente invasiva, el hueso se encuentra cubierto principalmente por tejido blando, ocasionando dificultad al momento de posicionar la broca guía sobre el hueso correctamente; Por esta razón se implementó una proyección en la punta de la broca que funciona como lo haría un haz de luz láser (Ilustración 19), de esta forma se identifica el punto sobre el hueso donde la broca está apuntando a lo largo de su eje longitudinal.

Esta funcionalidad facilita la percepción de ubicación de la broca guía en el espacio de trabajo quirúrgico, y ayuda en la percepción de profundidad ofrecida por el dispositivo de realidad aumentada.

Ilustración 19 - Proyección (punto azul) de la broca sobre el modelo



4.2.2.2 Retroalimentación puntos de perforación. La RLCA se basa en perforar un túnel tibial y femoral, para ayudar a esto el equipo de desarrollo de “NAVAR-G” realizó una aplicación web que permite marcar los puntos de entrada y de salida en cada uno de los huesos usando una técnica conocida por los ortopedistas. Estos puntos son almacenados en una base de datos para luego ser descargados por el módulo de administración de “NAVAR-Q” y así ser usados por el módulo de visualización.

El propósito de este proyecto es asistir en la perforación de estos túneles teniendo en cuenta los puntos de entrada marcados previamente por el ortopedista, para esto se desarrolló un completo sistema de retroalimentación con el fin de ayudar al cirujano a ubicar la broca guía que perforará el fémur y la tibia.

El sistema de retroalimentación fue desarrollado teniendo en cuenta los resultados de una prueba de funcionalidad realizada por el equipo de diseño de “NAVAR-Q” Y consiste en un código de colores que le indican al cirujano qué tan bien posicionada y orientada se encuentra la broca sobre el modelo, estos colores cambian a medida que un *collider* 2 veces más largo que la longitud de la broca (como se muestra en la Ilustración 20 y la Ilustración 21) y que está ubicado longitudinalmente colisiona con los puntos de entrada y salida representados por esferas, estas esferas tienen un diámetro de 5 milímetros y un área de seguridad representado por una esfera semitransparente de 10 milímetros de diámetro. El punto de entrada se distingue fácilmente del punto de salida por unos triángulos ubicados radialmente a lo largo de la circunferencia del punto de entrada, como se muestra en la Ilustración 22.

Ilustración 20 - Diferencia de tamaño entre el Collider y el modelo de la broca

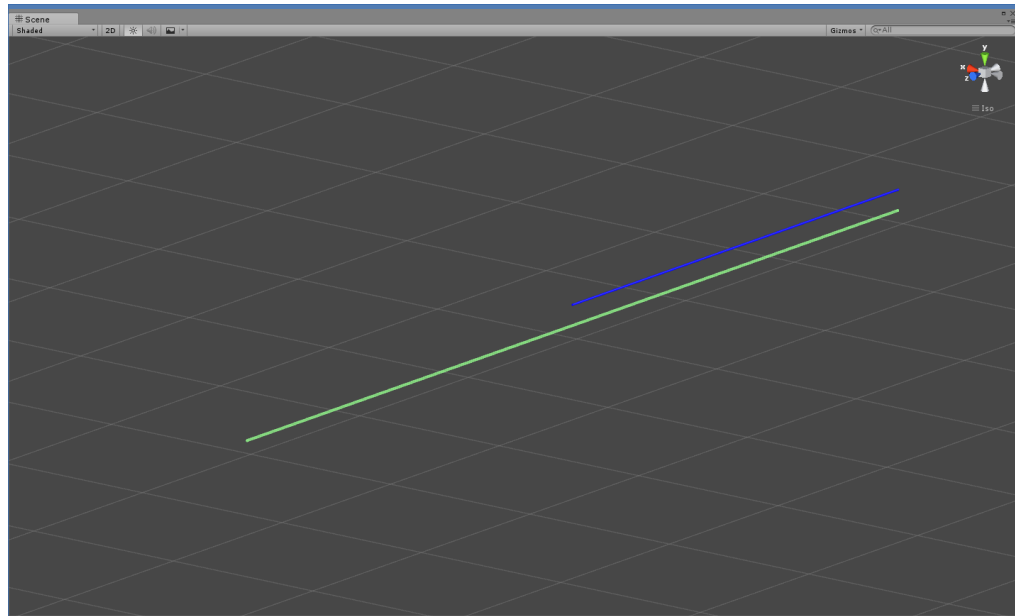


Ilustración 21 - Representación broca (azul) y collider (verde)

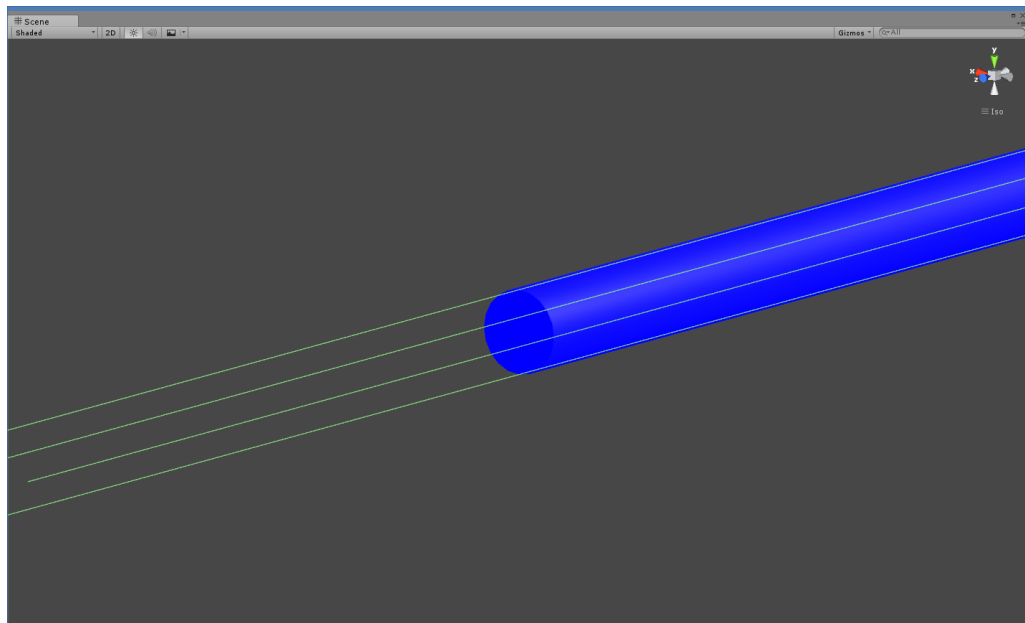
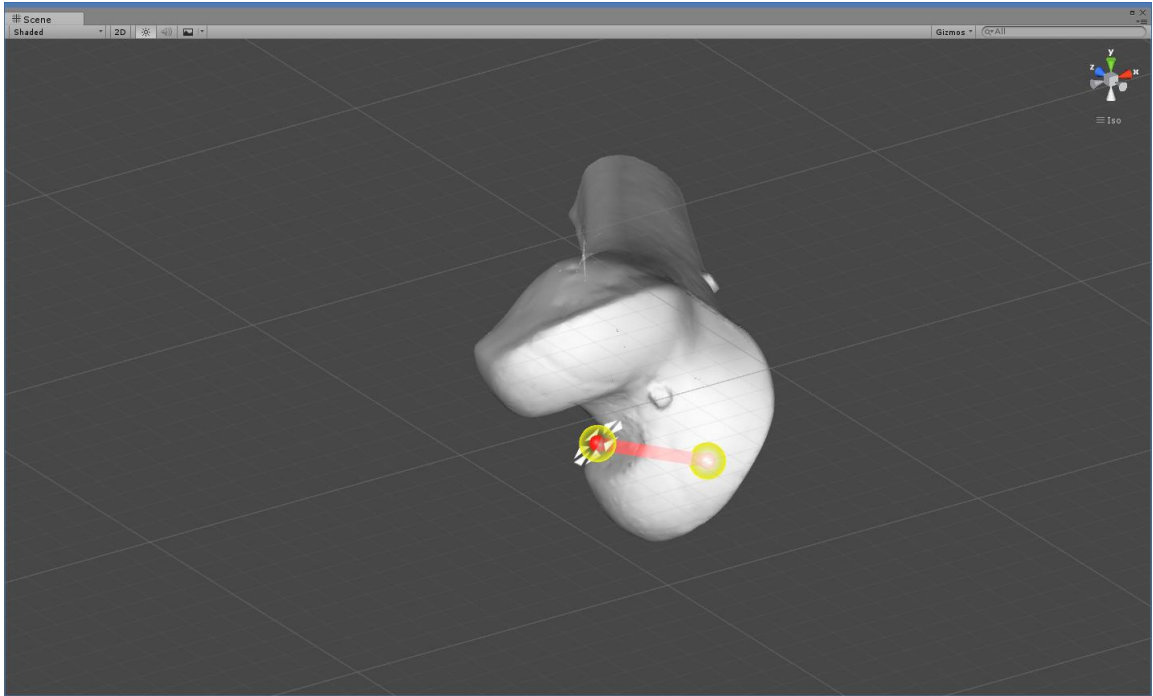


Ilustración 22 - Representación puntos sobre el modelo del fémur



La Ilustración 22 muestra el estado inicial de los puntos y el túnel, los cuales inician en color rojo, e irán cambiando de color a medida que el *collider* de la broca empieza a colisionar con las esferas, los cambios de color son diferentes entre los puntos y el túnel, y son explicados a continuación:

- Cuando el *collider* de la broca está tocando el punto de entrada (esfera pequeña) éste cambia a color verde.
- Cuando el *collider* de la broca está tocando con un punto de salida (esfera pequeña) éste cambia a color verde.
- Cuando el *collider* de la broca está tocando el punto de entrada y salida (los dos puntos están en color verde) el túnel cambia a color verde.
- Cuando el *collider* de la broca está tocando las dos zonas de seguridad (esferas amarillas semitransparentes) el túnel cambia a color amarillo.

Estas interacciones se explican mejor y con más detalles en los siguientes pseudocódigos (Ilustración 23 e Ilustración 24):

Ilustración 23 - Pseudocódigo que controla el color del túnel

```

countR = 2; //contador de rojos
countA = 0; //contador de amarillos
countV = 0; //contador de verdes

void Update ()
{ // color testigos
  if (countV == 2)
  {
    tunel.color =verde;
  }
  else if (countA==2)
  {
    tunel.color = amarillo;
  }
  else if (countR == 2)
  {
    tunel.color = verde;
  }
}

```

Ilustración 24 - Pseudocódigo que controla el color de las esferas

```

void OnCollisionEnter(Collider objeto)
{
  if (objeto=="zonaSeguridadEntrada")
  {
    countA++;
    countR--;
  }
  else if (objeto=="puntoEntrada")
  {
    countV++;
    puntoEntrada.color=verde;
  }

  if (objeto=="zonaSeguridadA")
  {
    countA++;
    countR--;
  }
  else if (objeto=="puntoSalida")
  {
    countV++;
    puntoSalida.color = verde;
  }
}

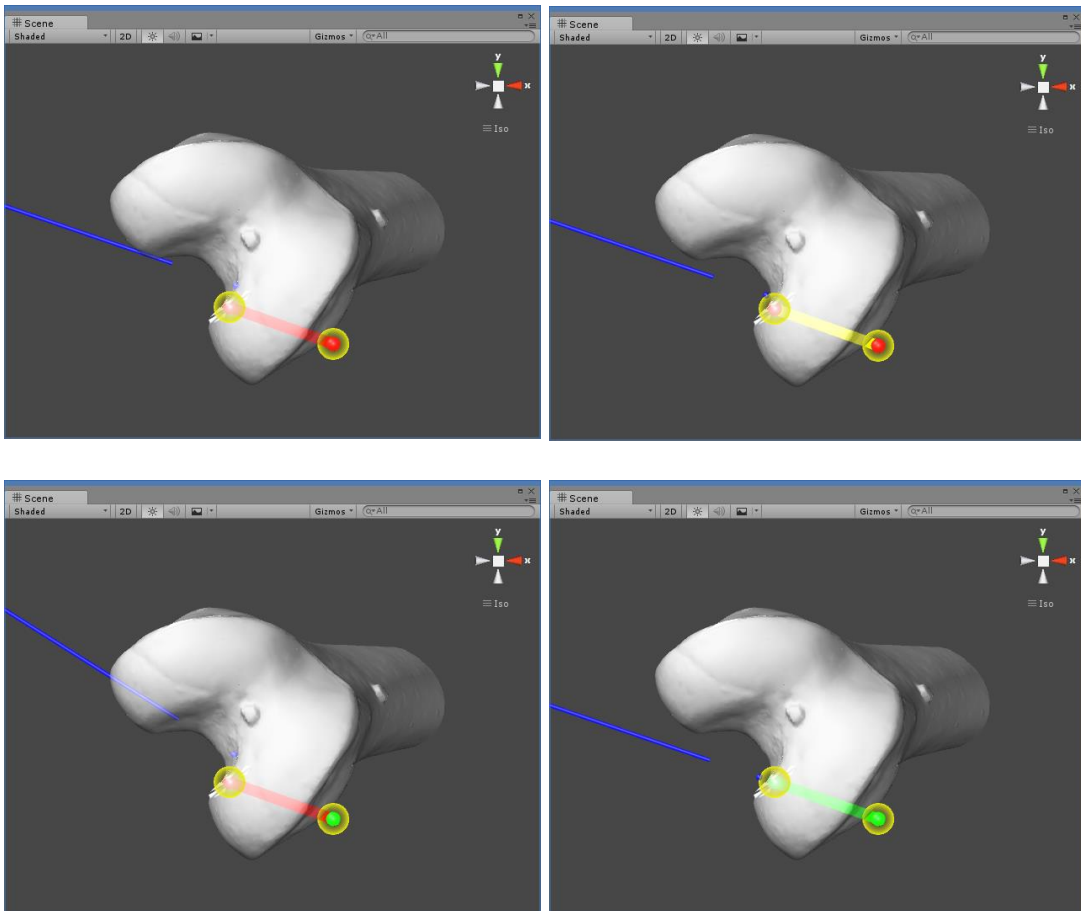
void OnCollisionExit(Collider objeto)
{
  if (objeto=="zonaSeguridadEntrada")
  {
    countA--;
    countR++;
  }
  else if (objeto=="puntoEntrada")
  {
    countV--;
    puntoEntrada.color=rojo;
  }

  if (objeto=="zonaSeguridadA")
  {
    countA--;
    countR++;
  }
  else if (objeto=="puntoSalida")
  {
    countV--;
    puntoSalida.color = rojo;
  }
}

```

En la Ilustración 25, se muestra el cambio de colores de los puntos de perforación y el túnel dependiendo de la posición de la broca:

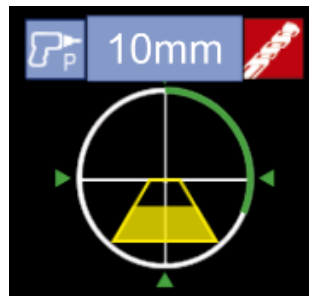
Ilustración 25 – Retroalimentación puntos de perforación



Con la información proporcionada por la retroalimentación desarrollada, se logra establecer, cuando está en una posición correcta para empezar a perforar el tunel, sin embargo, fue necesario desarrollar otra funcionalidad, que se explicara en el siguiente apartado.

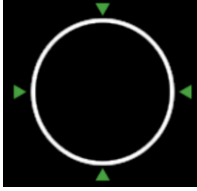

4.2.2.3 Sistema de orientación. El sistema de orientación (Ilustración 26) fue creado para asistir la ubicación y orientación de la broca con el fin de encontrar fácilmente los puntos de entrada y salida de la perforación, adicionalmente contiene información acerca de la perforación. Este sistema se ubica sobre la punta de la broca y así el usuario tendrá toda la información necesaria de la perforación al alcance visual mientras se ubica correctamente la broca y se perfora.

Ilustración 26 - Sistema de orientación



La información suministrada por este sistema se explica en con detalles en la Tabla 5:

Tabla 5 - Funcionalidades sistema de orientación

	<p>El borde del <i>target</i> sirve de plantilla para alojar algunas de las otras funcionalidades, adicionalmente sus triángulos cambian de color de la misma forma que lo hace el túnel.</p>
	<p>El trapecio simula gráficamente la distancia que hay desde la punta de la broca al punto de entrada, donde 70 milímetros es completamente vacío y 0 milímetros es completamente lleno. Una vez se empieza el proceso de perforación, este trapecio se oculta.</p>


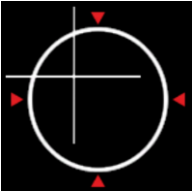



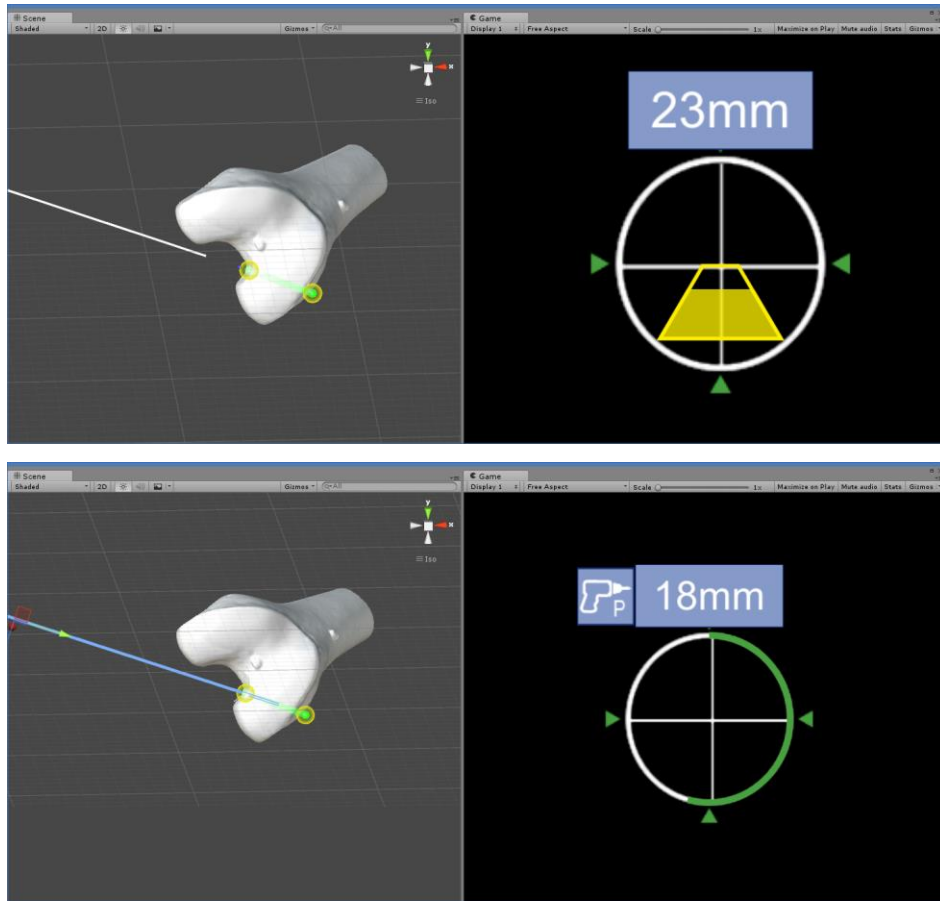
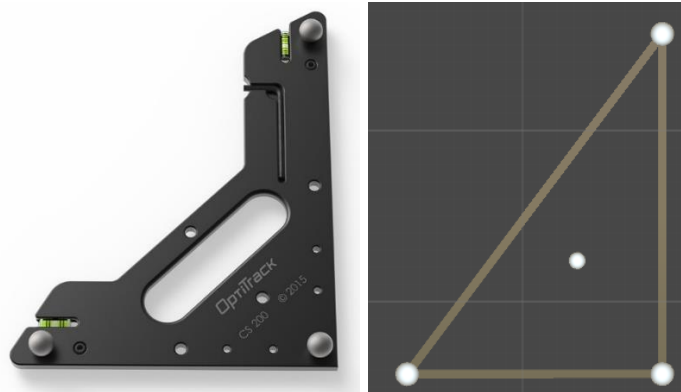
	<p>Este texto es un texto dinámico que cambia en tiempo real y representa dos cosas diferentes dependiendo del estado (ubicación o perforación).</p> <ul style="list-style-type: none"> • En el estado de ubicación representa la distancia desde la punta de la broca hasta el punto de entrada (similar al trapecio, pero en texto). • En el estado de perforación representa el avance de la perforación (distancia recorrida desde el punto de entrada hasta la punta de la broca).
	<p>Una vez ubicado el punto de entrada, es necesario angular la broca con el fin de alinearla con el punto de salida.</p> <p>Esta funcionalidad permite asistir en la angulación mostrando una cruz la cual se irá moviendo a través del <i>target</i>, cuando la cruz se encuentre en el medio, los triangulos cambiarán a color verde y la broca estara correctamente angulada y lista para perforar.</p>
	<p>Este icono funciona como un testigo booleano, cuando se entra en estado de perforación aparecerá indicando que la distancia mostrada en el cuadro de texto corresponde al avance (Ilustración 27).</p>
	<p>Durante la perforación, el avance se mostrará en el cuadro de texto, pero adicionalmente se irá llenado un scroll radial de color verde que indicará el porcentaje (de 0% a 100%) teniendo en cuenta la distancia del tunel.</p>
	<p>Este ícono funciona como un testigo booleano y se activa cuando no se está detectando la broca en el sistema de captura de movimiento.</p>

Ilustración 27 - Ejemplo estado ubicación(arriba) - estado de perforación (abajo)



4.2.2.4 Calibración broca. La calibración de la broca es un proceso importante donde se ajusta la orientación del modelo de la broca dentro de la aplicación, esto se necesario debido a que el error en la posición y rotación que maneja el sistema de captura de movimiento es pequeño en el centroide del objeto rígido, sin embargo, a medida que se acerca a la punta de la broca éste se va a incremento haciendo necesaria su calibración. En la **¡Error! No se encuentra el origen de la referencia.** se muestra el plano de calibración que se usa en este proceso.

Ilustración 28 - Plano de calibración - Representación en Unity



La calibración consiste en centrar físicamente la broca apuntando a una esfera del plano de calibración (Ilustración 29) y por medio de la interfaz y un gesto predefinido del Myo, se reflejará esta posición automáticamente dentro de la aplicación (Ilustración 30). Esta calibración sólo ajusta la rotación de la broca, ya que la posición sólo se calibra la primera vez que se crea el objeto rígido y no es necesario su ajuste a menos que se cambie la posición de las esferas o su distribución.

Ilustración 29 - Ubicación de la broca con respecto al plano de calibración

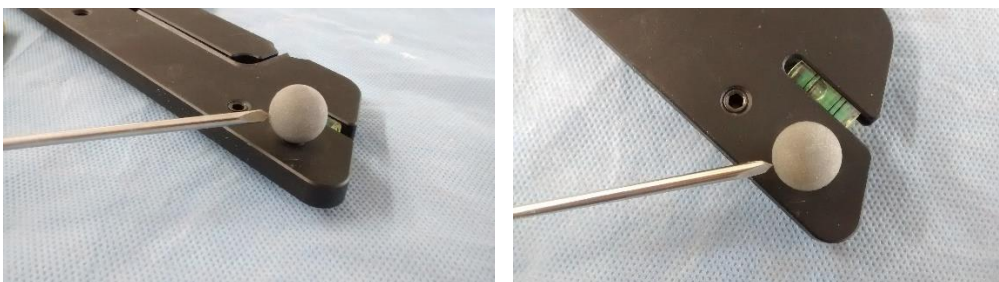
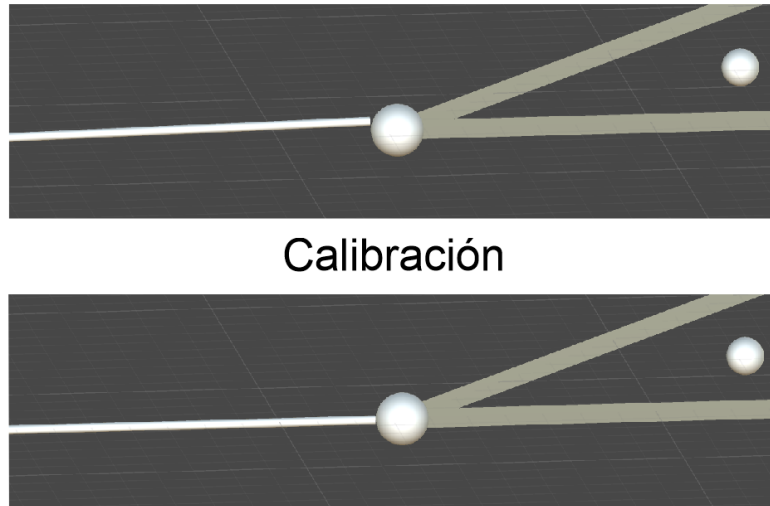


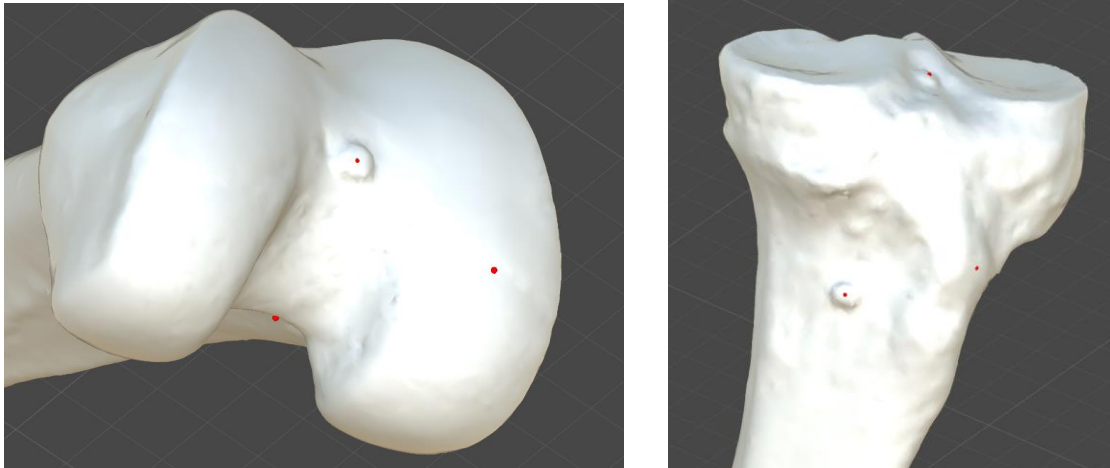
Ilustración 30 - Antes y después de la calibración



4.2.2.5 Registro 3D. El registro se encarga de empalmar el modelo virtual en la posición adecuada del hueso real en el sistema de coordenadas y en la visualización. Para el proyecto de investigación se planteó el registro haciendo uso de dos nubes de puntos y el algoritmo ICP (Iterative Closest Point) el cual empalma las dos nubes de puntos y devuelve una matriz de rotación que corresponde a la transformación que realizó una nube de puntos para ubicarse en la posición y rotación de la otra. El módulo de visualización puede cargar esta matriz de rotación por medio de un archivo y hacer el debido tratamiento de datos para que se ligen correctamente los objetos rígidos y los modelos reconstruidos.

Sin embargo, se implementó adicionalmente una opción de registro con sólo 3 puntos durante la visualización haciendo uso del Myo . Este registro hace uso de 3 puntos teóricos marcados previamente sobre el modelo (Ilustración 31) y haciendo uso de la broca se marcan 3 puntos experimentales; La idea de este procedimiento es que los puntos teóricos se empalmen con los experimentales, y en ese mismo instante con el objeto rígido.

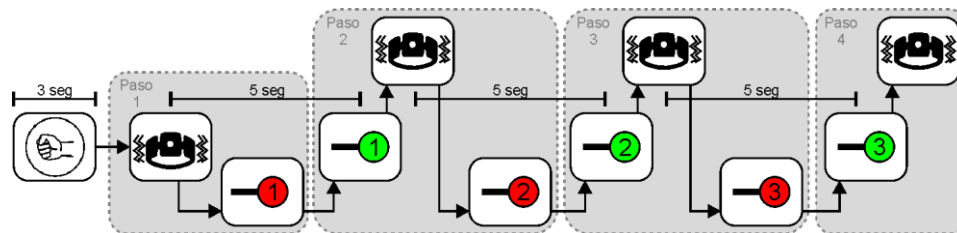
Ilustración 31 - Puntos teóricos usados para el registro



La ubicación de los puntos teóricos se elige teniendo en cuenta la posición habitual de los puntos de entrada y salida en una RLCA, lo ideal es que los puntos teóricos se encuentren cerca de estos puntos de perforación, así se reduce el error en esa área y se obtiene una perforación más fiel a los puntos marcados por el ortopedista en la preplaneación.

A continuación se explica el proceso de registro haciendo uso de los puntos teóricos (t1, t2, t3) y experimentales (e1, e2, e3) (Ilustración 32 e Ilustración 33):

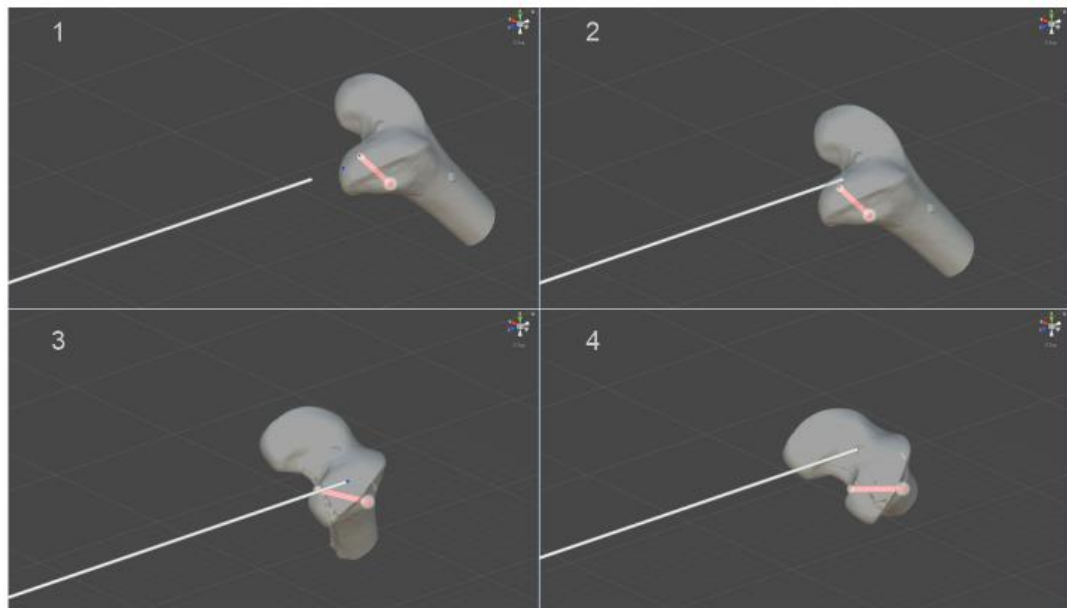
Ilustración 32 - Flujo del registro 3D



1. La banda Myo vibra para indicar al usuario que tiene 5 segundos para ubicar la broca en el punto t1.

2. La banda vibra para indicar al usuario que se capturó el punto e1. En este momento el punto t1 se mueve a la posición del punto e1. Adicionalmente informa al usuario que tiene 5 segundos para ubicar la broca en el punto t2.
3. La banda vibra para indicar al usuario que se capturó el punto e2. En este momento el modelo hace uso unicamente de rotaciones (teniendo como pivote el punto e1) para acercar lo más posible el punto t2 a el punto e2. Adicionalmente informa al usuario que tiene 5 segundos para ubicar la broca en el punto t3.
4. La banda vibra para indicar al usuario que se capturó el punto e3. En este momento el modelo rota usando como pivote el eje generado entre e1 y e2 para acercar lo más posible el punto t3 a el punto e3. Adicionalmente liga el modelo al objeto rígido que está siendo rastreado por el sistema.

Ilustración 33 - Pasos del registro de 3 puntos



4.2.3 Streaming. El Streaming del módulo de administración hacia el módulo de visualización se realiza a través de la librería NatNetSDK, la cual facilita la

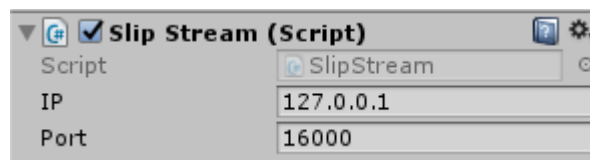
transmisión de datos. Sin embargo, es necesario hacer un tratamiento de datos entre un módulo y otro debido a diferentes factores que involucran el sistema de coordenadas, la elaboración de los marcadores reflectivos y la forma de calcular la matriz de transformación de cada objeto rígido. En total se rastrean y se envían por medio del stream 6 objetos rígidos (Pointer, Broca, Clavo pequeño, Clavo Grande, Gafas y Plano de calibración).

A continuación, se explicarán las principales funcionalidades desarrolladas para sistema de Streaming.

4.2.3.1 Funcionamiento. NatNet contiene dentro de su SDK ejemplos implementados en diferentes lenguajes que involucran básicamente la creación y funcionamiento de un servidor y un cliente, en este proyecto, el módulo de visualización será el cliente, ya que es el que va a recibir los datos de los objetos rígidos proporcionados por el sistema de rastreo y reconocimiento del módulo de administración “NAVAR-Q” (servidor) a través de Ethernet o Wifi (para el caso de los dispositivos inalámbricos con unidad de procesamiento propia, como las gafas Epson ® Moverio BT300).

Al momento de configurar el cliente las configuraciones son pocas, sólo ese necesario ajustar la IP del servidor y el puerto a usado para el Streaming de datos, el cual por defecto se usa el 16000. Estos ajustes se realizan sobre un script base de ejemplo que contiene la librería. En la Ilustración 34 se muestran las configuraciones necesarias.

Ilustración 34 - Configuración Streaming

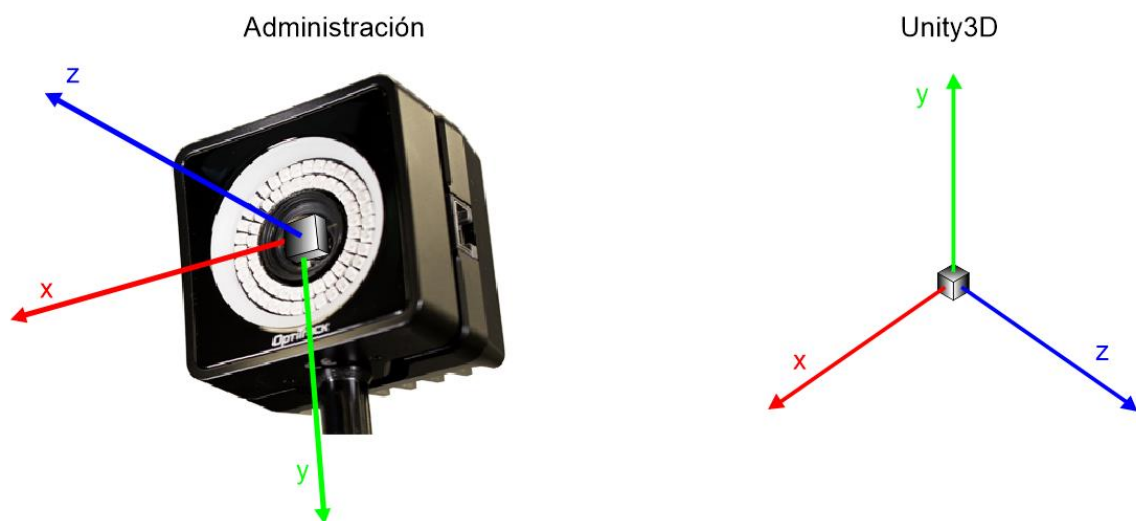


El script mencionado anteriormente es el encargado de recibir el archivo XML enviado a través de protocolo UDP. Sin embargo, también es necesario usar otro script, el cual se encarga de desempaquetar y asignar los valores desempaquetados a un objeto nuevo o predefinido.

Para el módulo de visualización se usan 3 datos principales del objeto rígido: ID, posición XYZ y rotación en cuaternios. Sin embargo, es necesario realizar un tratamiento específico a estos datos, lo cual se explicará en el siguiente apartado.

4.2.3.2 Tratamiento de datos. Debido a diferentes causas, es necesario realizar un tratamiento a la información de los objetos rígidos proporcionada por el módulo de administración. Las principales causas de esto es la diferencia en el sistema de coordenadas (entre el módulo de administración y Unity3D), el posicionamiento de los marcadores reflectivos sobre los objetos, y el cálculo de la matriz de transformación.

Ilustración 35 - Sistema coordenadas (right handed - left handed)



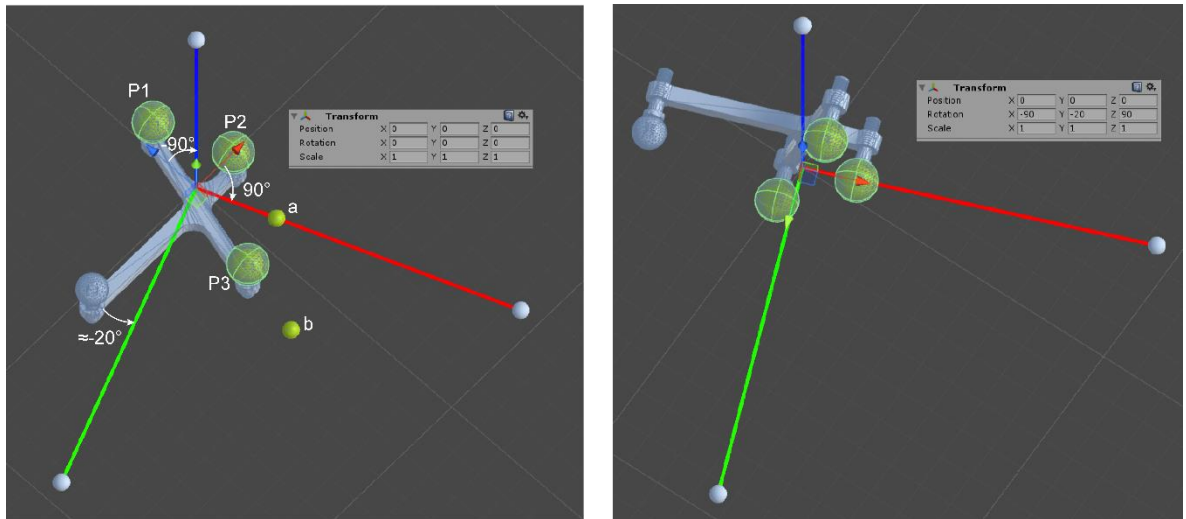
Como se muestra en la Ilustración 35, el módulo de administración usa un sistema de coordenadas de mano derecha y Unity3D uno de mano izquierda, por esta razón es necesario realizar un tratamiento a la matriz de transformación de los objetos. Para solucionar esto, se realiza una inversión de posición y rotación en el eje Y ($y = -y$, $qy = -qy$). Este cambio es necesario realizarlo a todos los objetos rígidos recibidos por el Streaming, ya que, si no se realiza, las rotaciones y traslaciones en tiempo real se moverán en ejes incorrectos.

Otro tratamiento que se realizó consiste en una pre-rotación a los modelos correspondientes a marcadores y objetos rastreables encontrados en el espacio de trabajo real, esto es debido a la forma como se crearon estos modelos físicamente (posición, organización y enumeración de las esferas reflectivas), del modelado del objeto virtual (sistema de coordenadas local del modelo) y del algoritmo que genera la matriz de transformación (Ilustración 36). Este cambio consiste en una rotación predefinida a los modelos dependiendo de su geometría.

Ilustración 36 - Pseudocódigo matriz de transformación

```
getRotationMatrix (p1, p2, p3)
{
    a = p2 - p1;
    b = p3 - p1;
    Z = Normalize(Cross(a, b));
    Y = Normalize(Cross(Z, a));
    X = Normalize(a);
}
```

Ilustración 37 - Tratamiento de matriz de rotación



Como se observa en la Ilustración 37, se realiza una transformación al modelo 3D teniendo en cuenta la forma como se genera la matriz de rotación, de esta forma el sistema de coordenadas local del modelo se liga al generado por la matriz de rotación para reproducir fielmente las rotaciones enviadas a través del Streaming.

El último tratamiento hecho a los datos recibidos por el Streaming es al objeto rígido de la broca, esto es debido a su composición, ya que, a diferencia de los otros objetos rígidos, esta sólo cuenta con 2 esferas reflectivas con el fin de rastrearla óptimamente incluso si está girando a causa de un taladro en funcionamiento durante la perforación. Debido a esto se reciben dos posiciones en vez de una posición y una rotación como se viene haciendo con los otros objetos rígidos, y es necesario usar las dos posiciones enviadas y hacer que éstas conformen la broca. El API de Unity contiene la función *lookAt()* que permite a un objeto orientarse a lo largo de su eje Z, en dirección a otro objeto, de esta forma se obtiene una reconstrucción fiel de la orientación en el eje X y Y. Debido a que el modelo de la broca es un cilindro perfecto, no importa la rotación longitudinal a lo largo del eje Z.

4.2.4 Visión estereoscópica. Debido a que las gafas de realidad aumentada usan dos displays (uno para cada ojo) es necesario generar una visión estereoscópica de tal forma que la imagen renderizada en cada una de las pantallas sea proyectada desde un ángulo y una posición diferente (simulando la posición de cada ojo). Para realizar este efecto se usaron dos cámaras separadas 6 centímetros entre ellas suponiendo que esta sea la distancia interpupilar del usuario. Esta distancia se eligió por defecto ya que es cercana al promedio mundial y ofreció buenos resultados a los usuarios de prueba.

Adicionalmente fue necesario estirar verticalmente al doble de su tamaño la imagen renderizada en cada ojo, ya que al dividir una pantalla en dos se pierde la proporción de los objetos dentro de la escena.

4.2.5 Interfaz de Usuario. El diseño de la interfaz de usuario estuvo a cargo del grupo de diseño de NAVAR-Q, el cual desarrollo un proyecto de grado titulado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA” y se enfoca en ajustes de interacción entre el usuario y el sistema al momento de realizar la reconstrucción de ligamento cruzado anterior.

La interfaz se implementó nativamente en Unity3D y consiste en una ventana (*Canvas*) con botones que cumplen dos tipos de funciones:






1. Dan acceso a información específica del paciente (caso clínico, diagnóstico, abordaje) durante cualquier punto de la cirugía.
2. Activan y desactivan la información visual proporcionada por los modelos 3D ubicados en la visualización.

Esta interfaz, hace uso del dispositivo Myo, el cual por medio de sensores mioeléctricos detectan gestos realizados por el mano del usuario, permitiéndole así interactuar de manera cómoda, inalámbrica y precisa con el sistema sin necesidad

de tocar pantallas, teclados o controles. La interacción del Myo evita que el cirujano ortopedista navegue sin soltar el instrumental quirúrgico o abandonar el espacio de trabajo, además evita perder la esterilización durante la cirugía.

A continuación, se muestran los gestos ligados a acciones configuradas en el Myo:

Tabla 6 - Gestos ligados a acciones

 <p>DOUBLE TAP</p>	<p>Este gesto contiene dos acciones:</p> <ol style="list-style-type: none"> 1. Bloquea y desbloquea el dispositivo para evitar acciones inconscientes. 2. Abre y cierra el menú
 <p>WAVE LEFT</p>	<p>Mueve el cuadro de selección a la izquierda.</p> <p>Sostenido por 3 segundos: Calibrar broca.</p>
 <p>WAVE RIGHT</p>	<p>Mueve el cuadro de selección a la derecha.</p>
 <p>MAKE FIST</p>	<p>Seleccionar.</p> <p>Sostenido por 3 segundos: Registro</p>
 <p>SPREAD FINGERS</p>	<p>Volver atrás.</p>

La interfaz completa se encuentra en el Anexo F.

5. PRUEBAS

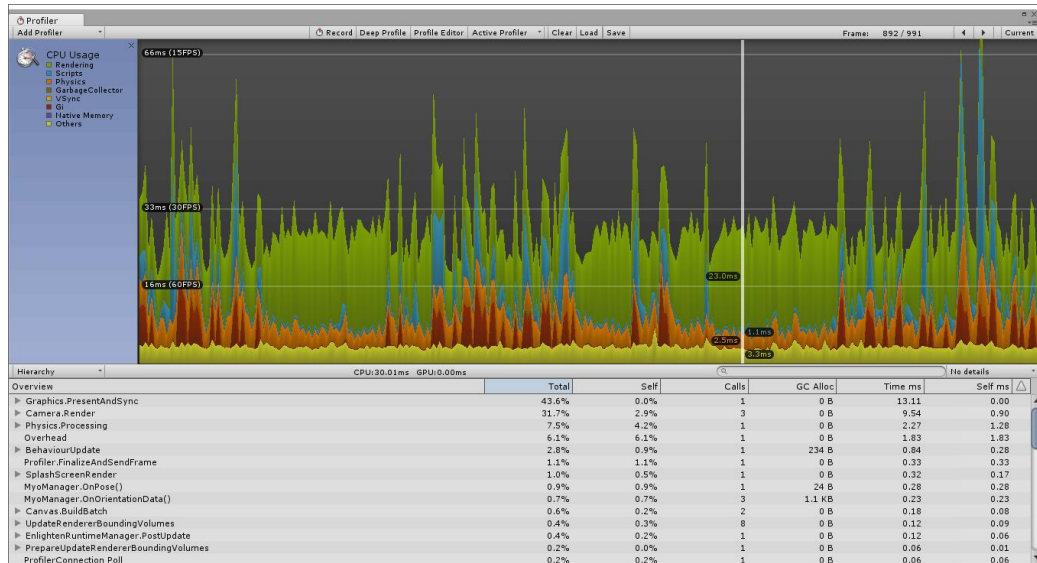
5.1 Pruebas técnicas de rendimiento.

Las pruebas técnicas de rendimiento se realizaron sobre el dispositivo de realidad aumentada haciendo uso del profiler incorporado en Unity3D, el cual ofrece información valiosa en cuanto a uso de CPU y memoria. A continuación, se presenta una captura de aproximadamente 1000 frames que nos muestra información a través de un gráfico de las principales cargas de información y rendimiento del Hardware.

La captura se realizó cuando el software se ejecutaba en el sistema operativo Android 5.1 de las gafas Epson Moverio BT300 mientras se realizaba el Streaming de datos y se observaba directamente a un hueso y a la broca, adicionalmente la broca se movía de tal forma que su *collider* colisionaba de manera intermitente con los puntos de entrada y salida del modelo.

5.1.1 Uso de CPU. El uso de la CPU se mantiene en un promedio del 50% en el dispositivo que usa un Intel® Atom™ x5, 1.44GHz Quad Core, este uso se distribuye de acuerdo con la Ilustración 38.

Ilustración 38 - Uso de CPU

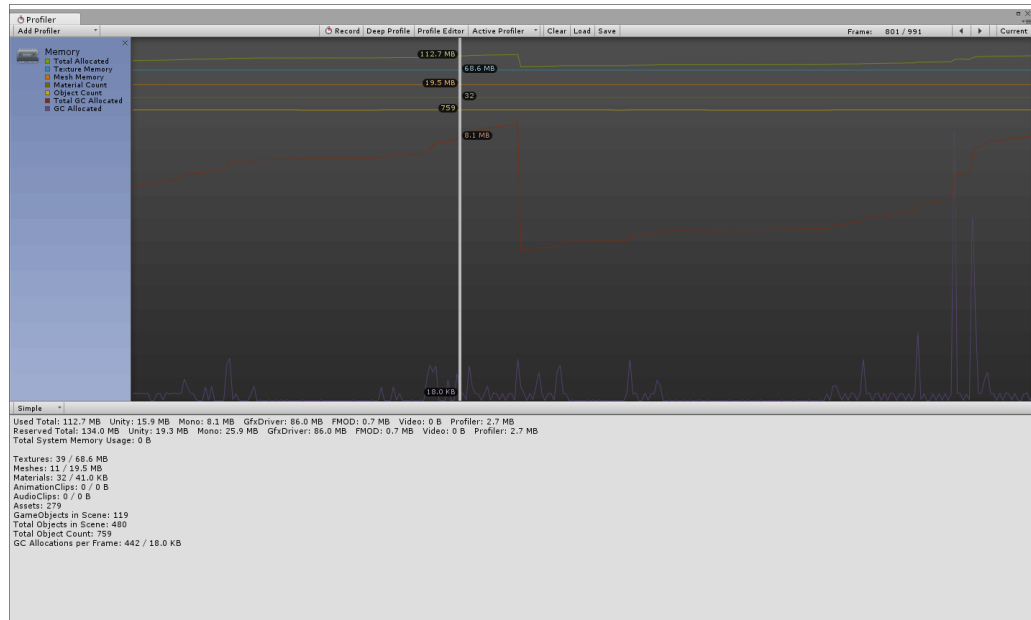


En la Ilustración 38 se observa que el uso de la CPU se distribuye principalmente en el renderizado (verde), las físicas (naranja), los scripts (azul), la iluminación (rojo) y otros (amarillo) que corresponde principalmente a la captura de frames para el profiler y al procesamiento interno de la escena. Adicionalmente se observa que la frecuencia de actualización de la imagen oscila principalmente entre los 30 y 40 FPS lo cual garantiza una buena fluidez durante la perforación.

En la ventana inferior se muestra el porcentaje de uso de cada uno de los servicios necesarios durante la ejecución de la aplicación.

5.1.2 Memoria. Durante la perforación se usa aproximadamente 110 MB de los 2 GB que tiene el dispositivo como se muestra en la Ilustración 39.

Ilustración 39 - Uso de memoria



5.2 Pruebas de precisión

Para probar la precisión del sistema se realizaron 24 perforaciones (12 tibias y 12 fémures) sobre unos huesos construidos a partir de un modelo 3D (reconstruidos desde una tomografía en formato DICOM) y ubicados sobre la representación física de una rodilla.

El objetivo de estas pruebas se centra en estimar el error absoluto entre los puntos planeados por los cirujanos ortopedistas y los puntos perforados usando la visualización y su sistema de orientación.

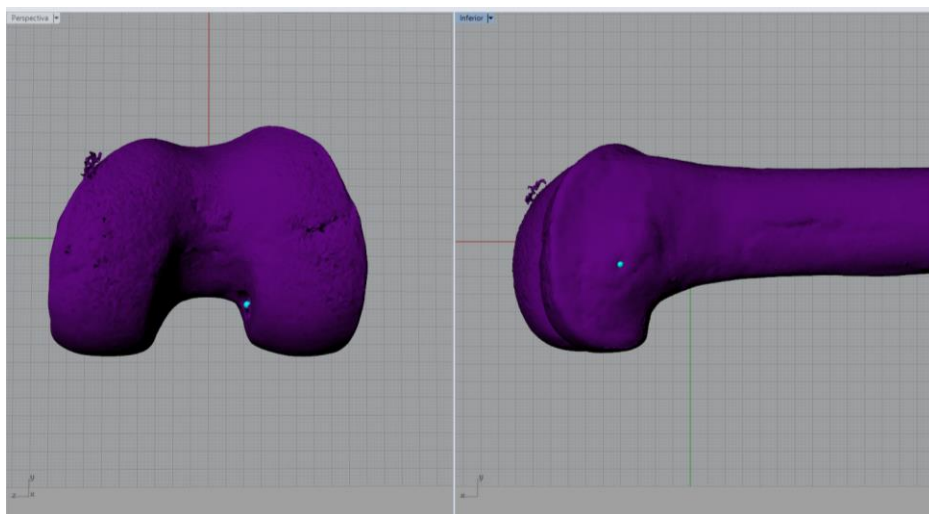
5.2.1 Metodología. Para estas pruebas se usaron los puntos de perforación teóricos marcados por 4 cirujanos ortopedistas en el software de preplaneación. Usando cada uno de estos puntos se realizaron 3 perforaciones tibiales y 3 femorales.

Una vez perforados, se escanearon los huesos usando un escáner 3D con el fin de obtener las coordenadas de los puntos experimentales perforados y así compararlos con los puntos teóricos. Con esta comparación se obtuvo el error medio absoluto.

5.2.2 Perforación. Las perforaciones se realizaron en el Laboratorio de Simulación quirúrgica de la Facultad de Salud de la UIS, en donde se llevó a cabo la preplaneación, calibración, registro y perforación de los túneles tibial y femoral usando la herramienta de visualización y su guía de orientación.

5.2.3 Recopilación de datos. Una vez perforados los túneles se digitalizaron usando un escáner 3D y por medio del software MeshLab®, Cloud Compare® y Rhinoceros® se obtuvieron las coordenadas de los puntos de ingreso y de salida perforados como se muestra en la Ilustración 40; estos puntos se registraron en la base de datos.

Ilustración 40 - Ubicación de esferas sobre la entrada y salida del tunel perforado



5.2.4 Resultados. Para obtener la precisión del sistema se calculó el error absoluto el cual representa la distancia de error entre los puntos marcados y los perforados. El error se calculó para cada uno de los túneles perforados sobre la tibia y el fémur.

$$\epsilon_a = X - X_i$$

Inicialmente se calculó el error en cada uno de los ejes y luego se obtuvo en términos de la distancia entre el punto teórico y el experimental usando la siguiente fórmula.

$$d = \sqrt{(X - X_i)^2 + (Y - Y_i)^2 + (Z - Z_i)^2}$$

A continuación, muestran los resultados del error entre coordenadas (X con X, Y con Y y Z con Z) para los puntos de entrada y salida del fémur y la tibia usando el dispositivo de realidad aumentada.

Tabla 7 - Resultados - Punto de ingreso fémur

PUNTO INGRESO FEMUR																
USUARIO	MUESTRA	COORDENADA PLANEADA			COORDENADA PERFORADA			ERROR ABSOLUTO -Ea (mm)			MEDIA ϵ_a			IMPRECISION ABSOLUTA - Ea		
		X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
E1	F1	25,5	-22,2	-10,0	25,4	-17,6	-8,5	0,1	-4,6	-1,5	0,0	-1,9	-0,7	1,1	1,5	1,0
E2	F1	24,0	-22,6	-9,9	25,8	-18,7	-9,1	-1,8	-3,9	-0,8						
E3	F1	22,7	-21,9	-9,1	24,8	-18,2	-4,9	-2,1	-3,7	-4,2						
E4	F1	26,7	-22,6	-10,2	28,7	-19,6	-9,8	-2,0	-3,0	-0,4						
E1	F2	24,3	-20,7	-9,0	22,4	-19,5	-8,1	1,9	-1,2	-0,9						
E2	F2	25,1	-19,4	-8,2	25,6	-19,2	-9,2	-0,5	-0,2	1,0						
E3	F2	22,7	-21,9	-9,1	21,6	-20,5	-9,1	1,1	-1,4	0,0						
E4	F2	24,1	-19,2	-9,0	23,6	-18,8	-8,9	0,5	-0,4	-0,1						
E1	F3	25,5	-22,2	-10,0	24,7	-19,7	-8,4	0,8	-2,5	-1,6						
E2	F3	23,9	-22,3	-8,4	23,6	-24,0	-10,2	0,3	1,7	1,8						
E3	F3	22,7	-21,9	-9,1	21,2	-18,9	-8,4	1,5	-3,0	-0,7						
E4	F3	24,1	-19,2	-9,0	24,0	-18,2	-8,2	0,1	-1,0	-0,8						

Tabla 8 - Resultados - Punto de salida fémur

PUNTO SALIDA FEMUR																
		COORDENADA PLANEADA			COORDENADA PERFORADA			ERROR ABSOLUTO -Ea (mm)			MEDIA ea			IMPRECISION ABSOLUTA - Ea		
USUARIO	MUESTRA	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
E1	F1	12,0	-4,5	-33,7	6,5	-11,4	-28,0	5,5	6,9	-5,7	1,8	0,4	-1,4	1,6	2,3	1,6
E2	F1	19,9	-9,8	-36,8	16,6	-5,6	-34,5	3,3	-4,2	-2,3						
E3	F1	9,6	0,7	-33,2	7,7	2,6	-29,8	1,9	-1,9	-3,4						
E4	F1	17,2	-1,4	-36,8	19,9	-4,5	-36,8	-2,7	3,1	0,0						
E1	F2	14,5	3,6	-30,4	11,7	-0,2	-32,8	2,8	3,8	2,4						
E2	F2	19,1	-8,9	-35,6	18,1	-7,4	-36,4	1,0	-1,5	0,8						
E3	F2	9,6	0,7	-33,2	6,8	0,9	-31,2	2,8	-0,2	-2,0						
E4	F2	20,3	0,2	-35,3	20,7	0,3	-34,8	-0,4	-0,1	-0,5						
E1	F3	12,0	-4,5	-33,7	12,0	-6,4	-33,3	0,0	1,9	-0,4						
E2	F3	15,1	-7,0	-33,8	12,4	-4,5	-32,3	2,7	-2,5	-1,5						
E3	F3	9,6	0,7	-33,2	7,1	1,2	-30,5	2,5	-0,5	-2,7						
E4	F3	20,3	0,2	-35,3	17,7	0,0	-34,4	2,6	0,2	-0,9						

Tabla 9 - Resultados - Punto de ingreso tibia

PUNTO INGRESO TIBIA																
		COORDENADA PLANEADA			COORDENADA PERFORADA			ERROR ABSOLUTO -Ea (mm)			MEDIA ea			IMPRECISION ABSOLUTA - Ea		
USUARIO	MUESTRA	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
E1	T1	-8,6	-7,3	-17,6	-7,4	-9,3	-16,1	-1,2	2,0	-1,5	-1,2	0,3	-0,6	0,8	1,3	1,4
E2	T1	-5,7	-20,4	-16,4	-6,2	-23,2	-14,1	0,5	2,8	-2,3						
E3	T1	-15,1	-1,9	-13,4	-12,8	0,6	-13,6	-2,3	-2,5	0,2						
E4	T1	-13,1	-18,7	-10,4	-11,4	-18,3	-11,2	-1,7	-0,4	0,8						
E1	T2	-8,6	-7,3	-17,6	-6,7	-6,7	-18,1	-1,9	-0,6	0,5						
E2	T2	-5,7	-20,4	-16,4	-6,2	-23,2	-14,1	0,5	2,8	-2,3						
E3	T2	-15,1	-1,9	-13,4	-12,2	-1,9	-14,6	-2,9	0,0	1,2						
E4	T2	-13,1	-18,7	-10,4	-12,8	-18,5	-7,8	-0,3	-0,2	-2,6						
E1	T3	-8,6	-7,3	-17,6	-7,1	-8,6	-16,5	-1,5	1,3	-1,1						
E2	T3	-5,7	-20,4	-16,4	-4,1	-20,5	-16,1	-1,6	0,1	-0,3						
E3	T3	-15,1	-1,9	-13,4	-13,4	0,6	-15,3	-1,7	-2,5	1,9						
E4	T3	-13,1	-18,7	-10,4	-12,3	-18,9	-8,3	-0,8	0,2	-2,1						

Tabla 10 - Resultados - Punto de salida tibia

PUNTO SALIDA TIBIA																
		COORDENADA PLANEADA			COORDENADA PERFORADA			ERROR ABSOLUTO -Ea (mm)			MEDIA ea			PRECISION ABSOLUTA -		
USUARIO	MUESTRA	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
E1	T1	-3,0	42,4	0,7	-5,4	40,3	-2,3	2,4	2,1	3,0	0,4	0,9	1,6	1,6	1,6	1,1
E2	T1	-10,4	39,9	-0,6	-12,8	38,6	-3,5	2,4	1,3	2,9						
E3	T1	-6,2	41,5	-4,5	-4,3	43,4	-2,8	-1,9	-1,9	-1,7						
E4	T1	-11,2	39,6	-1,7	-8,8	41,6	-2,2	-2,4	-2,0	0,5						
E1	T2	-3,0	42,4	-0,7	-3,7	42,8	-2,1	0,7	-0,4	1,4						
E2	T2	-10,4	39,9	-0,6	-11,8	36,6	-3,7	1,4	3,3	3,1						
E3	T2	-6,2	41,5	-4,5	-8,8	38,6	-5,7	2,6	2,9	1,2						
E4	T2	-11,2	39,6	-1,7	-11,9	38,0	-3,2	0,7	1,6	1,5						
E1	T3	-3,0	42,4	-0,7	-4,1	39,4	-3,2	1,1	3,0	2,5						
E2	T3	-10,4	39,9	-0,6	-8,4	40,3	-2,1	-2,0	-0,4	1,5						
E3	T3	-6,2	41,5	-4,5	-7,6	39,8	-4,2	1,4	1,7	-0,3						
E4	T3	-11,2	39,6	-1,7	-9,7	39,6	-4,9	-1,5	0,0	3,2						

Con los datos anteriores se calculó el error en la distancia usando las gafas de realidad aumentada, el cual se muestra a continuación:

Tabla 11 - Error de distancia medio en fémur usando gafas de realidad aumentada

DISTANCIA GAFAS R.A. - TUNEL FEMORAL										
MUESTRA	DISTANCIA EN ENTRADA	MEDIA POR MUESTRA	DESV. STPOR MUESTRA	MEDIA TOTAL	DESV. ST TOTAL	DISTANCIA EN SALIDA	MEDIA POR MUESTRA	DESV. STPOR MUESTRA	MEDIA TOTAL	DESV. ST TOTAL
F1	4,84	4,70	0,99	2,92	1,61	10,50	6,19	2,97	4,04	2,50
	4,37					5,82				
	5,98					4,33				
	3,63					4,11				
F2	2,42	1,50	0,77							
	1,14					5,30	2,84	2,00		
	1,78					1,97				
	0,65					3,45				
F3	3,07	2,57	0,94							
	2,49					1,94	3,10	0,93		
	3,43					3,97				
	1,28					3,71				
	2,76									

Tabla 12 - Error de distancia medio en tibia usando gafas de realidad aumentada

DISTANCIA GAFAS R.A. - TUNEL TIBIAL																		
MUESTRA	DISTANCIA EN ENTRADA	MEDIA POR MUESTRA	DESV. ST POR MUESTRA	MEDIA TOTAL	DESV. ST TOTAL	DISTANCIA EN SALIDA	MEDIA POR MUESTRA	DESV. ST POR MUESTRA	MEDIA TOTAL	DESV. ST TOTAL								
T1	2,77	2,94	0,77	2,75	0,73	4,38	3,68	0,60	3,32	0,98								
	3,66					3,98												
	3,40					3,18												
	1,92					3,16												
T2	2,05	2,87	0,69			2,75	0,73	1,62			3,18	1,47	3,32	0,98				
	3,66							4,74										
	3,14							4,08										
	2,62							2,30										
T3	2,27	2,43	0,82					2,75			0,73	4,06			3,09	0,86	3,32	0,98
	1,63											2,53						
	3,57											2,22						
	2,26											3,53						

En las tablas anteriores se puede observar la mejora que hubo entre la primera perforación (T1 y F1) y las dos siguientes (T2, F2, T3 y F3), esto es debido a que la primera perforación se realizó usando el primer prototipo, este prototipo fue mejorado principalmente con un registro más preciso y la inclusión de la calibración de la broca. Este mismo procedimiento se realizó con la aplicación de pantalla y se obtuvieron resultados similares, debido a que el cambio de dispositivo de visualización no interfiere en la precisión.

5.3 Pruebas de usuario

Las pruebas de usuario fueron realizadas en el Laboratorio de Simulación Quirúrgica de la Facultad de Salud en donde 4 cirujanos ortopedistas y personas del área de la salud realizaron el flujo de trabajo completo desde la preplaneación hasta la visualización. Durante estas pruebas los usuarios validaron el correcto funcionamiento de la herramienta y el nivel de satisfacción (Tabla 13) por medio de

una encuesta realizada en el proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA” (Anexo D y Anexo E)

Tabla 13 - Satisfacción uso de gafas RA

Usuario	Edad	Gafas	Complicaciones visuales	Evaluación
1	35	Lentes de contacto	Miopía	10
2	36	Sí	Astigmatismo	10
3	41	Sí	Miopía/Astigmatismo	8
4	28	No	-	8
			Promedio	9

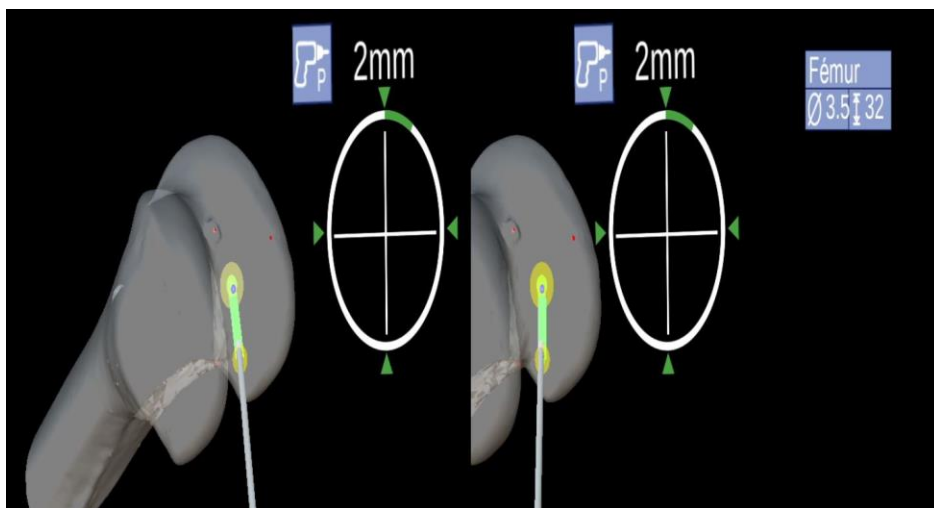
6. ENTREGABLES

Durante este proyecto se desarrollaron 2 aplicaciones que hacen uso del mismo principio de visualización y del mismo Stream de datos proveniente del módulo de administración "NAVAR-Q". Estas aplicaciones se describen brevemente a continuación:

6.1 Visualización en realidad aumentada

Esta aplicación es la principal del módulo de visualización, y consiste en una aplicación de realidad aumentada usando las gafas Epson ® Moverio BT300, sus principales funcionalidades son la carga de información preoperatoria, la asistencia en la perforación a través de un sistema de orientación e información extra, y un Streaming de datos en tiempo real. En la Ilustración 41 se muestra la visualización proyectada en las gafas.

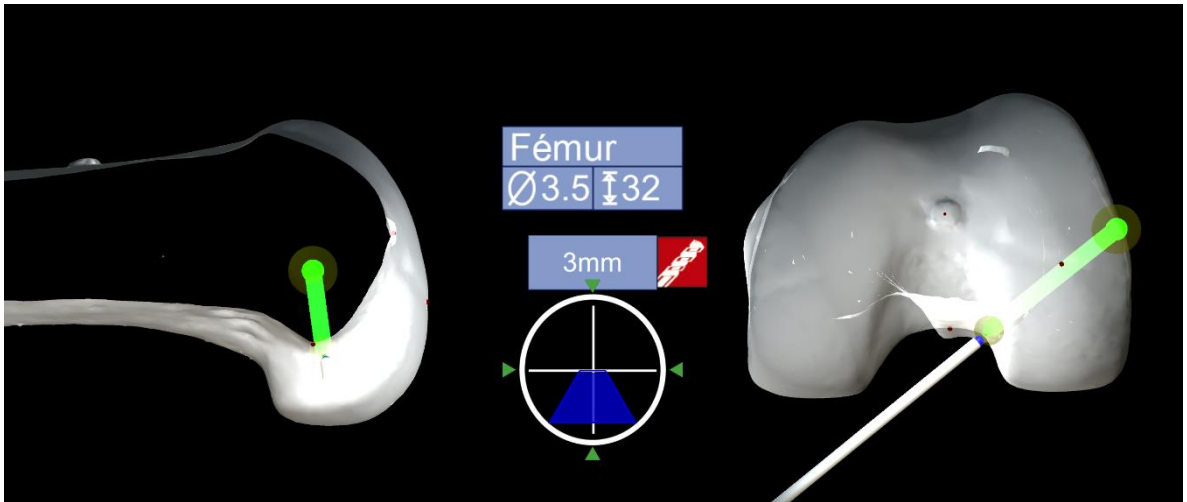
Ilustración 41 - Imagen proyectada en gafas de realidad aumentada



6.2 Visualización en escritorio

Esta aplicación se basa en la aplicación de realidad aumentada y contiene las mismas funcionalidades, sin embargo, está diseñada para ser visualizada a través de una pantalla. En la Ilustración 42 se muestra la visualización proyectada en la pantalla.

Ilustración 42 - Imagen proyectada en pantalla



7. CONCLUSIONES

1. Se desarrolló e implementó el módulo de visualización de realidad aumentada para las gafas Epson Moverio BT300 usando el entorno de desarrollo Unity3D como se expone en el capítulo de desarrollo.
2. Se diseñó el módulo de visualización usando como referencia el documento de especificación de requerimientos haciendo uso de la metodología de prototipado.
3. Se desarrollaron pruebas técnicas, de precisión y de usuario con el fin de dar culminado satisfactoriamente el desarrollo de la herramienta software y del presente trabajo de grado.
4. La calibración de la broca y el algoritmo de registro de 3 puntos implementado significó una mejoría considerable en el error absoluto final de la perforación y así se obtuvo una herramienta más fiable, robusta y precisa que servirá de base para futuros proyectos de investigación asociados como se muestra en el capítulo de pruebas.
5. Los usuarios presentaron un alto nivel de entendimiento y aprobación de la herramienta durante las pruebas y la introducción al software, esto es debido a la correcta funcionalidad del software y su similitud al flujo de trabajo de una reconstrucción de ligamento cruzado anterior. Lo anterior se evidencia en la encuesta de satisfacción realizada.
6. El uso de un dispositivo inalámbrico de realidad aumentada proporcionó ventajas en la comodidad, visualización, y sensación de profundidad durante el procedimiento quirúrgico en comparación con el usado en el primer prototipo (Gafas Meta).
7. El software de pantalla presentó una aprobación positiva como remplazo del dispositivo de realidad aumentada durante alguna situación que impida su

uso como la descarga de la batería del dispositivo o incomodidad por parte del usuario al usarlo.

8. El uso de la metodología de prototipado permitió un seguimiento constante de los requerimientos identificados y a su vez un grado alto de cumplimiento de estos al concluir el proyecto de investigación y el presente trabajo de grado.
9. El trabajo interdisciplinario realizado a lo largo del proyecto de investigación fue de gran ayuda ya que permitió abordar el problema expuesto desde diferentes perspectivas y a su vez ofreció un aprendizaje importante de las diferentes ramas de la ciencia junto a lo adquirido a través de la documentación de las tecnologías y herramientas usadas. Este aprendizaje será una ayuda importante a mi futuro profesional.

8. RECOMENDACIONES

1. Ofrecer un resumen al finalizar la cirugía teniendo en cuenta el error entre los puntos de perforación teóricos y prácticos.
2. Desarrollar Shaders y materiales que ofrezcan una mejor transparencia y visualización de los modelos y puntos de entrada.
3. Desarrollar una guía visual en 3D que permita facilitar el abordaje al punto de entrada, similar a la guía de orientación desarrollada para el punto de salida.
4. Adquirir, investigar y utilizar un dispositivo de realidad aumentada de tipo Head Mounted Display con mayor resolución y campo de visión que ofrezca una mejor visualización al usuario.
5. Realizar rastreo y visualización de la broca definitiva al momento de perforar el túnel femoral.
6. adquirir más cámaras de captura de movimiento y así obtener una mayor área de cobertura por las cámaras y a su vez un mejor rastreo de los objetos rígidos en la escena.

9. REFERENCIAS BIBLIOGRAFICAS

- [1] STIEHL, James B. "Navigation and MIS in orthopedic surgery". Springer, 2007.
- [2] SAMITIER, Gonzalo. MARCANO, Alejandro I. ALENTORN-GELI, Eduard. CUGAT, Ramon. FARMER, Kevin W. MOSER, Michael W. "Failure of Anterior Cruciate Ligament Reconstruction," *Arch Bone Jt Surg.*, vol. 3, no. 4, pp. 220–240, 2015.
- [3] SIELHORST, Tobias. FEUERSTEIN, Marco. NAVAB, Nassir. "Advanced medical displays: A literature review of augmented reality," *IEEE/OSA J. Disp. Technol.*, vol. 4, no. 4, pp. 451–467, 2008.
- [4] KANLIĆ, Enes M. DELAROSA, Fabian. PIRELA-CRUZ, Miguel. "Computer assisted orthopaedic surgery -- CAOS.," *Bosn. J. Basic Med. Sci.*, vol. 6, no. 1, pp. 7–13, 2006.
- [5] JD, Ayala-Mejías. GA, García-Estrada. PÉREZ-ESPAÑA, Alcocer. "Lesiones del ligamento cruzado anterior," *Acta Ortopédica Mex.*, vol. 28, no. 1, pp. 57–67, 2014.
- [6] RABAT, Carolina. DELGADO, Gonzalo. "Simple radiography Signs of a torn anterior Cruciate ligament," *Rev. Chil. Radiol. año*, vol. 14, no. 1, pp. 11–13, 2008.
- [7] LLINÁS, Paulo José. HERRERA, Gilberto. "Lesión de ligamento cruzado anterior," *Gundación Valle del Lili*, vol. 207. pp. 1–4, 2012.
- [8] TELEFONICA, Fundacion. "Realidad Aumentada : una nueva lente para ver el mundo," *Fund. Telef.*, vol. 1, pp. 1–97, 2011.
- [9] POMBO, H López. "Análisis y Desarrollo de Sistemas de Realidad Aumentada," p. 124, 2010.

- [10] “OptiTrack - Motive - Optical motion capture software.” [En línea]. [Revisado: 22 de Agosto de 2017]. Disponible en: <http://optitrack.com/products/motive/>.
- [11] “Gafas Moverio BT-300 - Epson.” [En Línea]. [Revisado: 12 de Julio de 2017]. Disponible en: <https://www.epson.es/products/see-through-mobile-viewer/gafas-moverio-bt-300>.
- [12] “Myo Gesture Control Armband | Wearable Technology by Thalmic Labs.” [En Línea]. [Revisado: 20 de Julio de 2017]. Disponible en: <https://www.myo.com/>.
- [13] “OptiTrack Expands Mocap Offerings - OptiTrack.” [En Línea]. [Revisado: 11 de Enero de 2017]. Disponible en: <https://www.optitrack.com/about/press/20150528.html>.
- [14] “OptiTrack’s Precise ‘Void’ Style Tracking Lets You Play Real Basketball in VR – Road to VR.” [En Línea]. [Revisado: 20 de Septiembre de 2017]. Disponible en: <https://www.roadtovr.com/optitracks-vr-precision-void-style-tracking-lets-you-play-real-basketball-in-vr/>.
- [15] OPTITRACK , “NatNet API User ’ s Guide,” no. October, pp. 1–24, 2015.
- [16] “NatNet SDK - Stream motion tracking data across networks - OptiTrack.” [En Línea]. [Revisado: 11 de Enero de 2017]. Disponible en: <http://optitrack.com/products/natnet-sdk/>.
- [17] “Desarrollar apps | Android Developers.” [En Línea]. [Revisado: 24 de Julio de 2017]. Disponible en: <https://developer.android.com/develop/index.html>.
- [18] “BT-300 - Technical Information - MOVERIO - Epson.” [En Línea]. [Revisado: 24 de Julio 2017]. Disponible en: <https://tech.moverio.epson.com/en/bt-300/index.html>.
- [19] “Unity - Manual: Manual de Unity.” [En Línea]. [Revisado: 24 de Julio de 2017]. Disponible en: <https://docs.unity3d.com/es/current/Manual/index.html>.
- [20] CORPORATION, Epson, “User's Guide Welcome to the BT-200”, 2014.

- [21] "OptiTrack - Prime 13." [En Línea]. [Revisado: 22 de Agosto de 2017]. Disponible en: <https://optitrack.com/products/prime-13/>.
- [22] "Unity - Game Engine." [En Línea]. [Revisado: 13 de Enero de 2017]. Disponible en: <https://unity3d.com/es>.
- [23] Microsoft, "Clase StreamReader (System.IO)." [En Línea]. [Revisado: 15 de Julio de 2017]. Disponible en: [https://msdn.microsoft.com/es-es/library/system.io.streamreader\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.io.streamreader(v=vs.110).aspx).
- [24] Hammmm, "GitHub - hammmm/unity-obj-loader: runtime .OBJ file loader for Unity3D." [En Línea]. [Revisado: 15 de Julio de 2017]. Disponible en: <https://github.com/hammmm/unity-obj-loader>.

BILIOGRAFÍA

Hammmm, "GitHub - hammmm/unity-obj-loader: runtime .OBJ file loader for Unity3D." [En Línea]. [Revisado: 15 de Julio de 2017]. Disponible en: <https://github.com/hammmm/unity-obj-loader>.

JD, Ayala-Mejías. GA, García-Estrada. PÉREZ-ESPAÑA, Alcocer. "Lesiones del ligamento cruzado anterior," *Acta Ortopédica Mex.*, vol. 28, no. 1, pp. 57–67, 2014.

KANLIĆ, Enes M. DELAROSA, Fabian. PIRELA-CRUZ, Miguel. "Computer assisted orthopaedic surgery -- CAOS.," *Bosn. J. Basic Med. Sci.*, vol. 6, no. 1, pp. 7–13, 2006.

LLINÁS, Paulo José. HERRERA, Gilberto. "Lesión de ligamento cruzado anterior," *Gundación Valle del Lili*, vol. 207. pp. 1–4, 2012.

Microsoft, "Clase StreamReader (System.IO)." [En Línea]. [Revisado: 15 de Julio de 2017]. Disponible en: [https://msdn.microsoft.com/es-es/library/system.io.streamreader\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/system.io.streamreader(v=vs.110).aspx).

POMBO, H López. "Análisis y Desarrollo de Sistemas de Realidad Aumentada," p. 124, 2010.

RABAT, Carolina. DELGADO, Gonzalo. "Simple radiography Signs of a torn anterior Cruciate ligament," *Rev. Chil. Radiol.* año, vol. 14, no. 1, pp. 11–13, 2008.

SAMITIER, Gonzalo. MARCANO, Alejandro I. ALENTORN-GELI, Eduard. CUGAT, Ramon. FARMER, Kevin W. MOSER, Michael W. "Failure of Anterior Cruciate Ligament Reconstruction," *Arch Bone Jt Surg.*, vol. 3, no. 4, pp. 220–240, 2015.

SIELHORST, Tobias. FEUERSTEIN, Marco. NAVAB, Nassir. "Advanced medical displays: A literature review of augmented reality," *IEEE/OSA J. Disp. Technol.*, vol. 4, no. 4, pp. 451–467, 2008.

STIEHL, James B. "Navigation and MIS in orthopedic surgery". Springer, 2007.

TELEFONICA, Fundacion. "Realidad Aumentada : una nueva lente para ver el mundo," Fund. Telef., vol. 1, pp. 1–97, 2011.

Unity - Manual: Manual de Unity. [En Línea]. [Revisado: 24 de Julio de 2017].
Disponible en: <https://docs.unity3d.com/es/current/Manual/index.html>.

ANEXOS

Anexo A - Requerimientos del NAVAR-Q

FSR1	El Software debe poder crear pacientes
FSR2	El software debe poder asignar varias operaciones a un paciente
FSR3	El software debe manejar la historia clínica del paciente
FSR4	el software debe asignar una cita al paciente
FSR5	El software debe poder buscar un paciente y el caso clínico asignado a la cita
FSR6	el software debe poder registrar la anamnesis del paciente
FSR7	el software debe poder registrar las Exámenes y Prueba
FSR8	el software debe poder asignar los exámenes tomográficos
FSR9	el software debe poder subir archivos tomográficos al sistema asignados al caso
FSR10	el software debe poder asignarle un diagnostico al caso del paciente
FSR11	El software debe poder crear una cirugía asignada a un caso
FSR12	el software debe poder elegir los actores necesarios para la planeación de la cirugía
FSR13	el software debe poder crear el caso asignado al fabricante

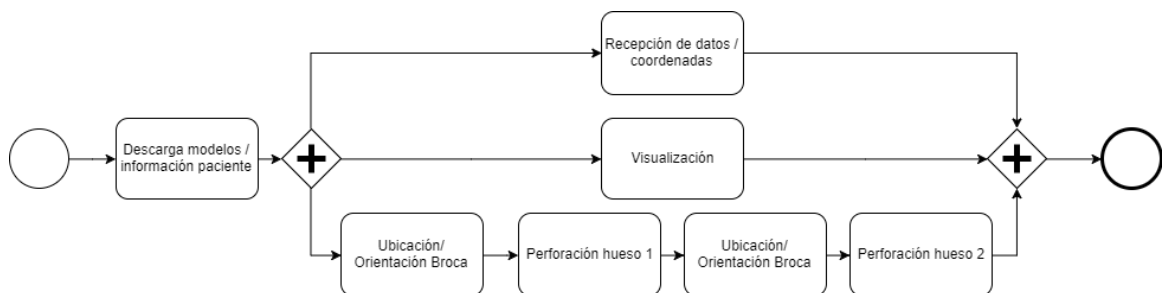
FSR14	el software debe poder cargar los archivos del fabricante que entrega al ortopedista
FSR15	el software debe poder registrar la validación del ortopedista al fabricante
FSR16	el software debe poder subir el modelo 3d
FSR17	el software debe poder visualizar el modelo 3d
FSR18	el software debe poder registrar puntos de cirugía
FSR19	el software debe poder visualizar en RA el modelo 3d
FSR20	el software debe poder cerrar la operación
FSR21	el software debe poder cerrar el caso clínico

PREFACIO

Este documento describe los requerimientos de software del Módulo de visualización para asistir la perforación del Túnel Tibial y Femoral en Reconstrucción del Ligamento Cruzado Anterior (RLCA) usando Realidad Aumentada.

Este documento de requerimientos de software es la base del desarrollo de software del proyecto. Describe los siguientes tópicos requerimientos de software, requerimientos de calidad, requerimientos de ambiente, restricciones y arquitectura básica del software. Este documento no describe el diseño detallado ni la implementación de los requerimientos.

1. Flujo del proceso



2. Requerimientos Software

2.1 Requerimientos Funcionales

Los requerimientos funcionales describen lo que el sistema debe hacer, los servicios que debe proporcionar, de la manera en que éste debe reaccionar a entradas.

FSR1	El módulo debe seguir un flujo de trabajo adaptado al procedimiento de RLCA.
FSR2	El módulo debe ofrecer una ubicación registrada en 3D de los modelos sobre la anatomía del paciente
FSR3	El módulo debe permitir la visualización de herramientas e información que intervengan durante el procedimiento.
FSR4	El módulo debe mostrar el estado en que el sistema está trabajando.
FSR5	El módulo debe mostrar los puntos de perforación sobre la anatomía del paciente.
FSR6	El módulo debe mostrar el avance en la perforación.
FSR7	El módulo debe ofrecer al usuario guías visuales al momento de ubicar el pin guía sobre la rodilla del paciente.
FSR8	El módulo debe permitir la visualización de posibles desvíos o inconsistencias que se generen durante la perforación.
FSR9	El módulo debe servir de cliente en la recepción de coordenadas e información enviada vía Streaming por el módulo de Administración.
FSR10	El módulo debe estar en capacidad de consultar archivos e información de configuración descargados de una base de datos.
FSR11	El módulo debe ofrecer al usuario la capacidad de seleccionar la información a mostrar.
FSR12	El módulo debe ofrecer al usuario la posibilidad de escoger cual hueso va a perforar primero.

FSR13	El módulo debe poder ser visualizado a través del dispositivo de realidad aumentada o una aplicación de escritorio.
-------	---

3. Requerimientos de Calidad

Los requerimientos no funcionales operacionales definidos para este sistema son:

3.1 Escalabilidad

Por escalabilidad se hace referencia a la capacidad del sistema de crecer sin desmejorar la calidad.

Intercambio de usuarios

El módulo debe estar en la capacidad de intercambiar entre usuarios y pacientes de manera rápida y sin pérdida de información en el caso en que hubiese dos cirugías seguidas.

Adaptabilidad

El módulo debe estar desarrollado de tal forma que permita adaptarse fácilmente a diferentes tipos de estándares de desarrollo y software.

3.2 Integración

Por Integración se hace referencia al nivel de acople con el sistema general.

Integración completa

El módulo debe integrarse completamente con el proyecto de investigación "SISTEMA DE ORIENTACIÓN USANDO REALIDAD AUMENTADA PARA LA

PERFORACIÓN DE TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR”

3.3 Rendimiento

EL rendimiento hace referencia a los tiempos de respuesta que ofrece el software.

Tiempo real

Debido al entorno quirúrgico, el módulo debe responder con poca latencia asemejándose a una respuesta en tiempo real.

4. Requerimientos de Ambiente

4.1 Requerimientos de Ambiente de Desarrollo

4.1.1 Hardware de Desarrollo

Computador con procesador i5 o superior, 4GB memoria RAM, Puerto HDMI, y 1GB de video

4.1.2 Desarrollo de Software

- Unity 5.4 o Superior
- Android SDK, mínimo para Android 5.1
- Sistema operativo Windows 7 o superior.

4.2 Requerimientos de Ambiente de Testing

Entorno quirúrgico simulando la ubicación de dispositivos, espacio e iluminación dentro de un quirófano.

5. Restricciones

5.1 Restricciones Hardware

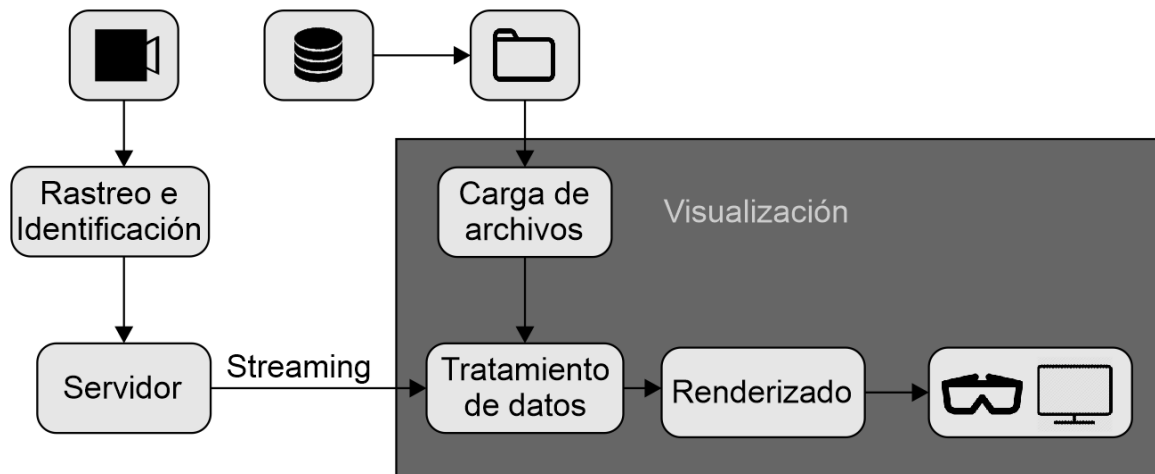
- Capacidad de renderizado y resolución del dispositivo de realidad aumentada.
- Capacidad de procesamiento del dispositivo anfitrión del software
- Latencia generada por el Streaming de datos a través de protocolo UDP, Ethernet y Wifi.
- Cableado del dispositivo de realidad aumentada que impide el libre desplazamiento del usuario durante la ejecución del software.

5.2 Restricciones Software

- La sensibilidad de calibración de los marcadores para el sistema de Realidad Aumentada.
- Restricciones de entorno y lenguaje del SDK del dispositivo de realidad aumentada

6. Arquitectura del Software

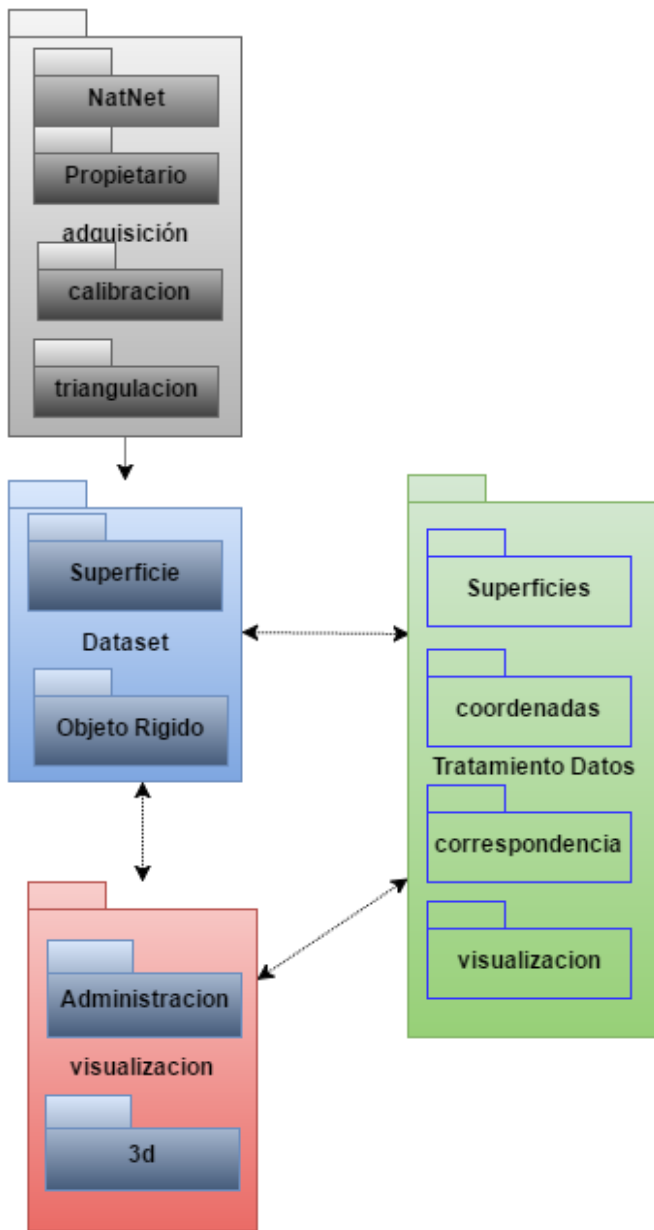
Para el módulo de visualización se piensa en una arquitectura de escritorio la cual se estará constituido por periféricos (cámaras) los cuales reconocerán los objetos del mundo real y se asociarán con los reconstruidos, permitiendo servir de apoyo al ortopedista en la cirugía realizada.



En la representación se observa que la solución esta propuesta sobre un software de escritorio y sobre un dispositivo de realidad aumentada debido a que el proceso es el mismo y sólo se diferencia al momento del renderizado donde se hace intercambia de una visión estereoscópica a vistas ortográficas y viceversa. En el modelo se puede visualizar la adquisición de la información por medio de los periféricos(cámaras). Los archivos de recursos estarán locales y serán gestionados por medio de un módulo de la aplicación para poder cargarlos y representados en pantalla por medio de módulo de visualización

6.1 Vista de Módulos

Esta Diagrama nos representa la vista de desarrollo la cual nos muestra la distribución de trabajo como se relacionan los módulos dentro de la aplicación, esto nos facilitara la visión que debe tomar y el Reusó de código de parte del desarrollador.



Este proyecto se enfocará en el módulo de tratamiento de datos y el de visualización:

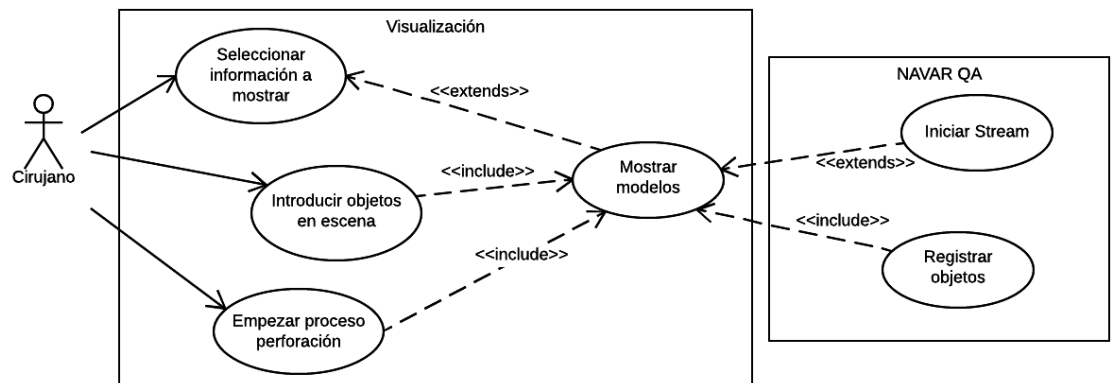
Módulo de Tratamiento de datos: modulo encargado de transformar los datos tales como superficies, coordenadas, visualizaciones, y uno adicional que es el modelo de correspondencia entre la parte real y la reconstrucción.

Módulo de Visualización: el módulo de visualización contar de dos partes la interfaz de administración en donde se deben llamar las funcionalidades que permiten poner a punto el software y la visualización de la operación en pantalla, esta debe contar con las facilidades y modos de vistas que presten todas las facilidades al cirujano

6.1.1 Casos de uso módulo de visualización

Los casos de uso representan la interacción entre los actores y el sistema.

Debido a que el módulo de visualización es un software de asistencia visual no contiene interacciones directas con el software más que navegar por el menú y seleccionar la información que desea mostrar, por esta razón los casos de uso son limitados en este proyecto.



6.1.2 Casos de uso detallados

Caso de uso #1	Seleccionar información a mostrar
Descripción	El usuario selecciona la información que requiera
Actores	Cirujano.

Condiciones previas	No se requiere condición previa	
Detalles	Acción del actor	Respuesta del sistema
Flujo	1. El usuario realiza el gesto de abrir el menú	
		2. El menú se despliega frente al cirujano
	3. El usuario navega por el menú a través de gestos	
		4. La sombra de selección se mueve con respecto a los gestos del cirujano
	5. El usuario selecciona la información que requiera	
		6. La información se muestra frente al cirujano

	7. El realiza el gesto de cerrar el menú	
		8. El menú se cierra

Caso de uso #2	Introducir objetos en escena	
Descripción	El usuario prepara el instrumental quirúrgico y lo introduce en la escena	
Actores	Cirujano	
Condiciones previas	El Streaming de datos debe estar enviando datos	
Detalles	Acción del actor	Respuesta del sistema
Flujo	1. El usuario introduce el objeto en escena	
		2. El modelo 3D del objeto aparece en la visualización

Caso de uso #3	Empezar proceso de perforación	
Descripción	El usuario comienza el proceso de perforación asistida por el módulo	
Actores	Cirujano	

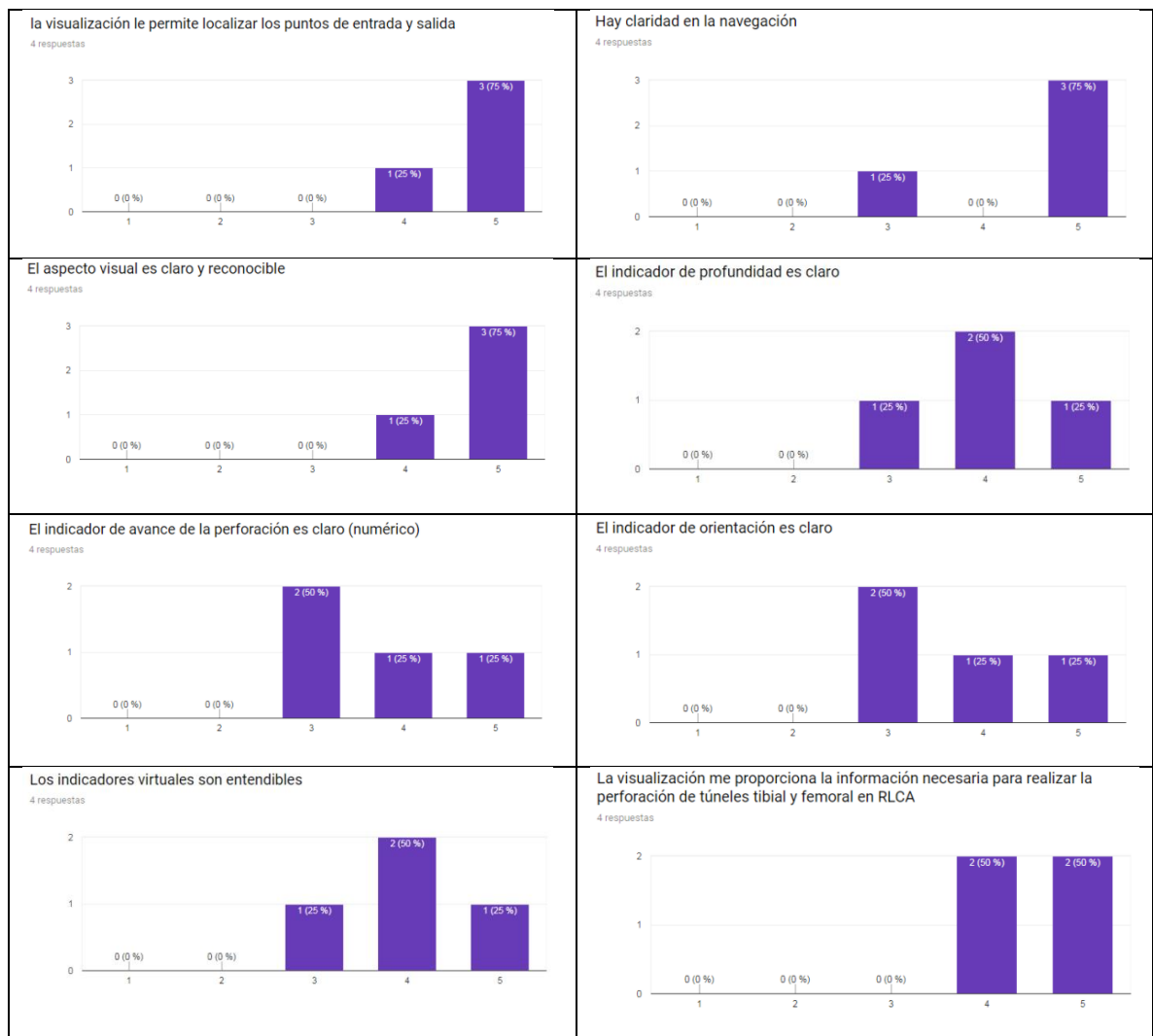
Condiciones previas	El Streaming de datos debe estar activo, se debe haber realizado el registro	
Detalles	Acción del actor	Respuesta del sistema
Flujo	1. el usuario coloca la broca guía en el punto de entrada guiado por la distancia (texto y scroll) que se muestra en pantalla	
		2. El punto de entrada cambia a color verde cuando la proyección del pin guía se orienta con éste
	3. El usuario alinea la broca con el punto de salida guiado por el <i>target</i> ubicado en la punta de la broca	
		4. El punto de salida cambia a color verde cuando la proyección del pin guía se orienta con éste

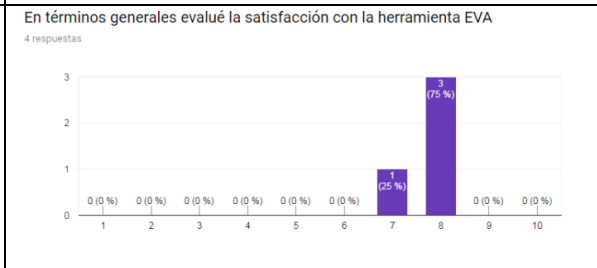
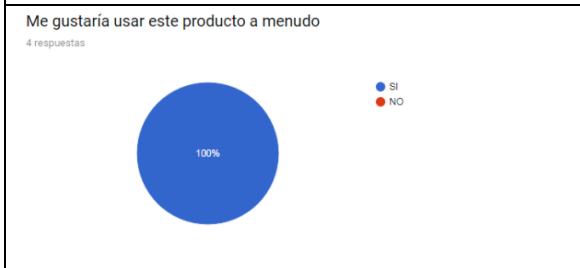
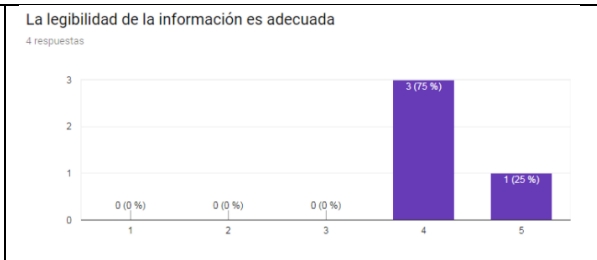
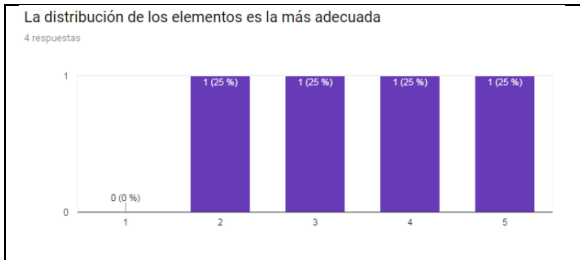
		<p>5. La representación del túnel cambia a color amarillo (zona de seguridad) o verde (correcta alineación) dependiendo de la orientación y posición de la broca</p>
	<p>6. El usuario comienza la perforación</p>	
		<p>7. Un ícono muestra que se entró en estado de perforación, se muestra en tiempo real la profundidad de la perforación (texto y scroll)</p>

Anexo C - Pruebas de interfaz gráfica realizadas por el equipo de diseño de NAVAR-Q a ortopedistas

(Tomado del proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA”).

Estas pruebas fueron realizadas a 4 ortopedistas y comprenden los aspectos de funcionalidad y legibilidad de la interfaz gráfica de usuario.

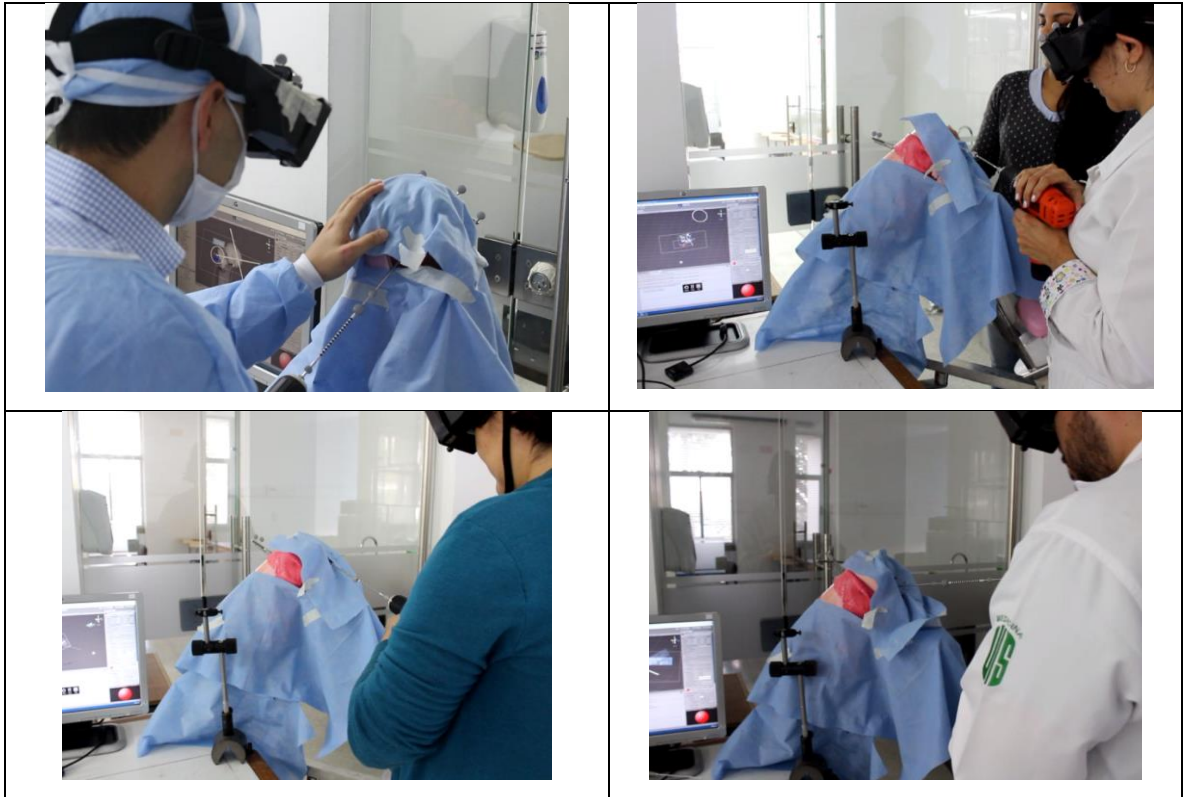




Anexo D - Pruebas de funcionalidad realizadas por ortopedistas.



Pruebas de usuario usando versión de escritorio.



Pruebas de usuario usando gafas de realidad aumentada.

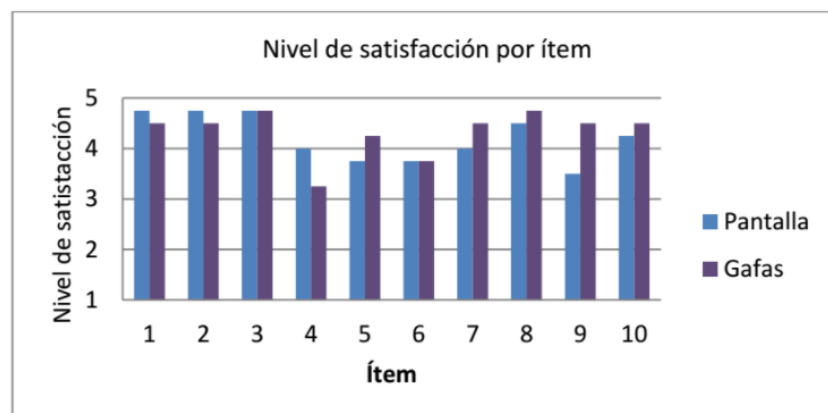
Anexo E - Encuesta de satisfacción

(Tomado del proyecto de grado “GUÍA VISUAL PARA LA UBICACIÓN DE LOS TÚNELES TIBIAL Y FEMORAL EN LA RECONSTRUCCIÓN DE LIGAMENTO CRUZADO ANTERIOR (RLCA) USANDO LA REALIDAD AUMENTADA”).

Items de la encuesta:

Número	Ítem
1	La visualización le permite localizar
2	los puntos de entrada y salida
3	El aspecto visual es claro y reconocible
4	El indicador de profundidad es claro
5	El indicador de avance de la perforación es claro (numérico)
6	El indicador de orientación es claro
7	Los indicadores virtuales son entendibles
8	La visualización me proporciona la información necesaria para realizar la: perforación de túneles tibial y femoral en RLCA
9	La distribución de los elementos es la más adecuada
10	La legibilidad de la información es adecuada

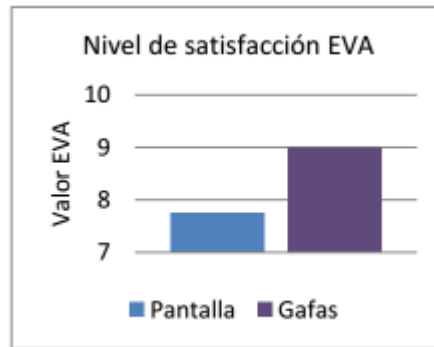
Nivel de satisfacción por ítem:



Estadísticos descriptivos:

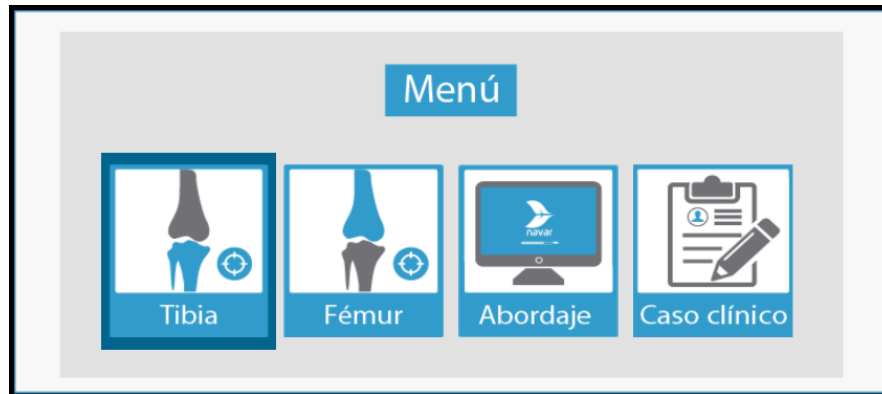
Variable	N	Media	Desv.Est	Suma	Mínimo	Máximo	Rango
Pantalla	4	7,75	0,5	31	7	8	1
Gafas	4	9	1,15	35	8	10	2

Nivel de satisfacción general:

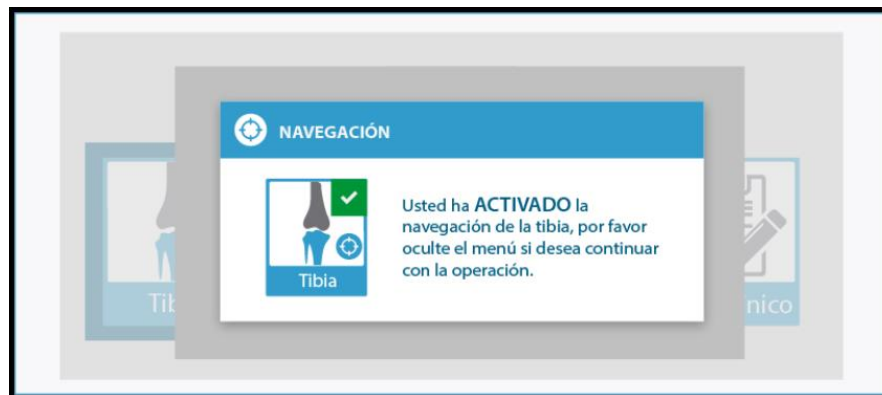


Anexo F - Interfaz de usuario

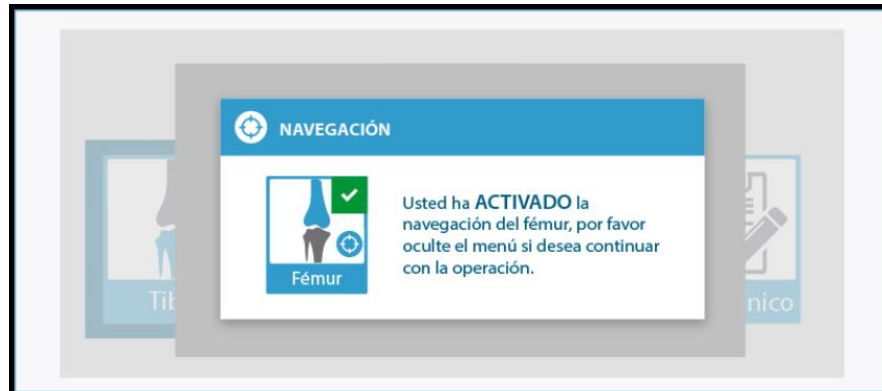
Menú principal



Tibia activada



Fémur activado



Abordaje fémur




Abordaje tibia



Caso clínico

Caso clínico



Manuel Fernando Jiménez contreras
68 Años

ENFERMEDAD ACTUAL

- *Fractura hundimiento traumática, de meseta externa de tibia izquierda con diagnóstico de lesión tumoral en la meseta
- *Traumatismo directo sobre la rodilla izquierda
- *Inflamación impotencia funcional en la rodilla

MEDICAMENTOS CONSUMIDOS

- *Loratadina