



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Sistema de realidad aumentada para fetoteca

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A N

Rodrigo Terpán Arenas

Saúl Rodríguez García

DIRECTOR DE TESIS

M. en A. Luis Yair Bautista Blanco



Ciudad Universitaria, Cd. Mx., 2017



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Saúl Rodríguez García

“Una vez hayas probado el vuelo siempre caminarás por la Tierra con la vista mirando al Cielo, porque ya has estado allí y allí siempre desearás volver”

– Leonardo da Vinci.

Es una noche fría *sin lluvia* de Noviembre, irónicamente acompañado del tema “November Rain” de Guns N’ Roses para que surja en mí, la melancolía para escribir esta dedicatoria...

A dos días de cumplir veintitrés primaveras, siendo una vida de carencias y dificultades, pero también de grandes experiencias y éxitos. De las cuales sigo agradeciendo a mi mamá por todos los sacrificios que ha hecho por darme la oportunidad de prepararme profesionalmente y que no viva las mismas carencias que ella tuvo al venir a la Ciudad de México a la edad de nueve años para salir adelante. Estoy más que satisfecho, que ella vea mis resultados al obtener los títulos de: Programador en Sistemas e Ing. Mecatrónico. Y puedo decir que, de ahora en adelante, me toca a mí, mejorar nuestro estilo de vida.

Siempre había pensado que mis éxitos eran gracias a mi esfuerzo y dedicación y que solamente este crédito correspondía exclusivamente a mí y de nadie más, pero luego vas madurando y ves la vida de una manera totalmente diferente. Es por esta razón que también estoy eternamente agradecido a esta máxima casa de estudios, por ser una gran institución en todo el país y que te ofrece muchas cosas, pero está en uno mismo aprovecharlas. Por mencionar algunas cosas: experiencias, espacios culturales y recreativas, pero sobre todo madurez.

Agradezco tanto a la Fundación Telmex y a la misma UNAM por otorgarme becas a largo de mi trayectoria estudiantil, que sin ese apoyo hubiera sido más complicado culminar este camino. Además, mi *alma mater* me dio la gran oportunidad de conocer a personas que son amigos para toda la vida: Alejandro, que será un gran médico y fue una inspiración para mí; Cynthia,

una mujer inteligente y que será una excelente divulgadora científica como Física; Jesús, aún con su condición física no se rinde por terminar la carrera (realmente admiro eso). Ricardo y Claudia, siendo que los conozco por poco tiempo veo en ustedes a quién le puedo decir: amigo.

Finalmente, esta pequeña nota es para el Saúl del futuro, cuando tenga por lo menos treinta años. Para recordarte que cuando escribías esto, estabas lleno muchas ambiciones y metas. Y en caso de que estés en la plenitud profesional, no olvides por todos los sacrificios que hiciste, para llegar hasta ahí.

Rodrigo Terpán Arenas

Mi abuelo solía decir que uno no sabía a dónde lo iba a llevar la vida. En un principio, pensé que esto era porque las condiciones de su vida lo habían llevado a tener diversas profesiones. Ahora me doy cuenta que esto aplica a todos nosotros. Basta con ver mi trayectoria escolar para darse cuenta de esto. Me tomó un tiempo considerable de mi carrera para encontrar los temas que me apasionaban.

El que esté escribiendo esto es el resultado de un viaje que ha durado veinticinco años. El trayecto ha implicado sacrificios, tanto míos como de mi familia, pero también ha traído consigo beneficios a futuro y muchos buenos momentos que recordar.

Esta etapa de mi vida está llegando a su fin, y la siguiente se avecina con rapidez. Pero, aprovechando esta pequeña pausa, quiero mirar atrás y agradecer a todos los que me han acompañado y me han hecho la persona que soy en este momento.

A mi madre, Silvia, por enseñarme el valor del trabajo duro, inculcarme conocimiento y cultura y por ponerme en el camino correcto cuando me desviaba. Eres una de las personas más fuertes que conozco y eres un ejemplo para mí.

A mi padre, Leobardo, por siempre hablar conmigo y por hacerme entender el valor de los libros y de la música. Hemos tenido muchas conversaciones cortas y largas. Espero que en el futuro podamos seguir teniéndolas.

A mi hermana, Natalia, por estar ahí desde el inicio, compartiendo mis gustos, apoyándome y molestándome al mismo tiempo. Como hermanos, siempre estaremos juntos sin importar la distancia. Espero que nunca se nos olvide.

A mi familia en México DF, Morelos y Yucatán por enseñarme el valor de mantenernos unidos y apoyarnos el uno al otro. Siempre están presentes en mi mente y en mi corazón sin importar la distancia y el tiempo que nos separen.

A mis amigos más cercanos: Adrian, Julieta, Jorge, Isaac, Mariana, Jenny, Jaime, Jhovan, Pablo, Said y Misael; por estar allí para apoyarme, hacerme reír, soportar mis ataques nerds y servir como psicólogos ocasionales.

A los profesores que han marcado mi vida académica: Norma García Vasconcelos, Lili Mara Raquel de Gortari Rabiela, Erick Castañeda, Rolando

Peralta y Jesús Manuel Dorador. No sólo me han formado cómo profesional, sino también como persona.

A mi maestro Luis Yair Bautista Blanco, por ser mi profesor, coach, voz de la razón en asuntos escolares y en especial, por ser mí amigo durante la gran mayoría de mi trayectoria universitaria. Y también gracias por todas las pláticas amenas sobre comics y videojuegos, que espero continúen a futuro.

Y finalmente, gracias a mis casas de estudio, el Colegio Madrid y la UNAM, por formarme como un adulto profesional y responsable y por inculcarme buenos hábitos de estudio y de pensamiento. También gracias por permitirme formar parte de esta comunidad y otorgarme las herramientas para poder avanzar en mi vida.

Muchas gracias a todos.

“Todos nos contamos una historia sobre nosotros mismos. Siempre. Continuamente. Esa historia es lo que nos convierte en lo que somos. Nos construimos a nosotros mismos a partir de esa historia.”

-Patrick Rothfuss

“Hay muchos tiempos difíciles en el futuro, pero debes conservar tu sentido del humor, superar las dificultades y divertirte en el camino”

-Shigesato Itoi (Earthbound)

“Nadie nace siendo “fuerte”. Una persona se vuelve fuerte poco a poco, aprendiendo a plantar cara a las dificultades en lugar de huir de ellas, por mucho que le asuste hacerlo...”

...Nadie sabe lo que está por venir, pero una cosa es segura: todos aquellos que creen en ti estarán allí para ayudarte.

Confía en ellos”.

-Shigesato Itoi (Mother 3)

“The world ends with you. If you want to enjoy life, expand your world. You gotta push your horizons out as far as they'll go.”

-Sanae Hanekoma (The world ends with you)

“No es importante saber cuánto tiempo queda, sino saber que haces con el tiempo que se te concede...”

-Gandalf el gris (El señor de los anillos: La comunidad del anillo)

De parte de ambos

Muchas gracias a los departamentos de Embriología e Informática Biomédica de la facultad de medicina de la UNAM, así como al personal que forma parte de esos organismos, por otorgarnos la oportunidad, el material, el tiempo y el espacio para realizar este trabajo.

Índice

Índice	I
Lista de figuras	III
Lista de tablas	V
Introducción	1
1. Objetivos	2
1.1 Objetivo general	2
1.2 Objetivos específicos	2
2. Justificación	2
3. Marco teórico	8
3.1 Realidad Aumentada	8
3.1.1 Definición	9
3.1.2 Comparación con Realidad Virtual y Realidad Mixta	10
3.1.3 Usos	14
3.2 Desarrollo Embrionario	17
3.2.1 Preservación de material orgánico	20
4. Modelado	22
4.1 Captura y generación	23
4.1.1 Escalamiento	24
4.1.2 Técnicas de generación	29
4.2 Procesamiento y filtrado de modelos	40
4.2.1 Descripción del problema	40
4.2.2 Software utilizado	41
4.2.3 Proceso de sanación	42
4.2.4 Resultados	46
5. Desarrollo de la aplicación	47
5.1 Diseño Conceptual	47
5.1.1. Necesidades	48
5.1.2. Requerimientos	48
5.1.3. Especificaciones	49
5.1.4. Concepto	51
5.1.5. Propuestas	51
5.1.6. Elección.	54
5.2 Diseño de Detalle	55
5.2.1. Programas y complementos	55
5.2.2. Importación de modelos	57
5.2.3. Interfaz de usuario	61
5.2.4. Rotación de modelos	64
5.2.5. Marcadores de medición	70

5.2.6. Pantalla táctil	73
5.2.7. Localización simultánea y mapeo del entorno real	76
6. Pruebas de implementación	81
6.1 Protocolo de pruebas	81
6.2 Resultados	84
6.3 Discusión	86
7. Conclusiones	88
8. Trabajo a futuro	90
9. Referencias	91
10. Anexos	98

Lista de figuras

Figura 1 Dos ejemplos de uso de la realidad aumentada utilizando diferentes dispositivos	3
Figura 2 Número de artículos sobre la RA en entornos educativos según la IEEE	4
Figura 3 Inversión desde 2011 en miles de millones de USD en las industrias de RV y RA	5
Figura 4 Inversiones por sector en el mercado de RV/RA en el periodo comprendido entre mediados del 2016 a mediados del 2017	6
Figura 5 Cantidad de usuarios de tecnologías de RA móvil proyectado hacia 2021	7
Figura 6 Proyecciones de ganancias en las industrias de RA y RV proyectadas a 2021	7
Figura 7 Sistema SixthSense	9
Figura 8 Realidad aumentada en un teléfono móvil.	9
Figura 9 Diagrama general de la tecnología RA	11
Figura 10 Reconocimiento de superficies con realidad mixta	13
Figura 11 Superficie con elementos aleatorios	13
Figura 12 Demostración de la tienda virtual	14
Figura 13 Presentación de modelos 3D en edificios	15
Figura 14 Aplicación móvil de Niantic	16
Figura 15 Modelo tridimensional de un corazón humano	16
Figura 16 Demostración de la realidad mixta	17
Figura 17 Comparación de los embriones de un humano y un pollo	17
Figura 18 Diagrama del periodo de desarrollo de un feto humano. Incluyendo los períodos con mayor probabilidad de anomalías.	19
Figura 19 Causas de 2.68 millones de muertes neonatales en 2015 a nivel mundial de acuerdo con la OMS (WHO) (Imagen traducida al español)	20
Figura 20 Modelo de pecho femenino de la página bio-digital	22
Figura 21 Ejemplo de figuras geométricas sencillas modeladas utilizando polígonos.	23
Figura 22 Superficie generada por NURBS (izq) y por polígonos (der).	24
Figura 23 (izq) Conjunto de modelos de diferentes sitios con escalas distintas (der) El mismo conjunto de modelos re-escalados mediante un algoritmo.	25
Figura 24 Opciones de escalamiento al importar un modelo a Unity	26
Figura 25 (izq) Ejemplo de un modelo con malla dañada. (der) El mismo modelo reparado	30
Figura 26 (Izq) Textura con baja resolución. (Der) Textura con alta resolución	31
Figura 27 (Izq) Sensores presentes en el Kinect (Der) Proyección de puntos infra rojos por el emisor IR del Kinect	32
Figura 28 Modelo sin textura generado por Kinect	33
Figura 29 Escáner profesional GoScan3D de Creaform	34
Figura 30 Diagrama del proceso de fotogrametría	34
Figura 31 Nube de puntos creada a partir de fotografías	35
Figura 32 Preparación del modelo sobre una superficie de color contrastante.	38
Figura 33 Toma de fotografías del modelo.	38
Figura 34 Selección de imágenes a usar para la creación del modelo	39
Figura 35 Modelo resultante de una captura de imágenes exitosa.	39
Figura 36 Acercamiento a los pies del modelo. Se observa una unión no existente en los talones del modelo, así como una falta de detalles en los dedos.	41
Figura 37 Vista inferior del modelo. Se observa que la superficie inferior presenta un espacio vacío debido a que no se puede mover el modelo durante el proceso de captura.	41
Figura 38 Pantalla inicial de Autodesk Meshmixer	42

Figura 39 Selección de partes innecesarias del modelo para la operación de corte.	43
Figura 40 Resultado de la operación de corte.	44
Figura 41 Herramienta de sanación de vacíos del modelo.	44
Figura 42 Herramientas para la modificación física del modelo.	45
Figura 43 Herramienta de transformación del modelo.	46
Figura 44 Resultado de modelo con imperfecciones arregladas.	46
Figura 45 Segundo modelo con imperfecciones arregladas.	47
Figura 46. Visor Samsung Gear VR	52
Figura 47 Guante háptico desarrollado en España	53
Figura 48 Visor HoloLens desarrollado por Microsoft	53
Figura 49. De izquierda a derecha: Código QR y Marcador.	56
Figura 50. Programa de animación 3D. Autodesk 3D's Max.	58
Figura 51. Parámetros de configuración para Unity.	58
Figura 52. Textura del modelo del feto.	59
Figura 53. Configuración de exportación en 3D's Max.	59
Figura 54. Importación del modelo en Unity 3D.	60
Figura 55. Configuración de textura.	60
Figura 56. Textura y ejes acoplados al entorno de Unity 3D.	60
Figura 57. Diagrama de transición de estados de ventanas.	61
Figura 58. Inicio: Presentación de la aplicación.	62
Figura 59. Menú principal del programa.	62
Figura 60. Ventana para definir el número de puntos de medición.	63
Figura 61. Interfaz de Realidad Aumentada.	63
Figura 62. Ventana de créditos de colaboración en el proyecto.	64
Figura 63. Rotaciones particulares de los Ángulos de Euler.	65
Figura 64. Matrices B, C y D.	65
Figura 65. Rotación en el origen del plano complejo.	68
Figura 66. Botones para aplicar la transformación de rotación.	70
Figura 67. Rotación utilizando los cuaterniones en Unity.	70
Figura 68. Marcador esférico.	71
Figura 69. Estructura de las instancias en C#.	72
Figura 70. Distancia entre dos puntos.	72
Figura 71. Segmento dirigido en código.	72
Figura 72. Nueve marcadores siendo usados para medir el diámetro de la cabeza del feto.	73
Figura 73. Simulación del movimiento del mouse en la pantalla táctil.	74
Figura 74. Coordenadas del espacio con base en las coordenadas de la pantalla táctil.	75
Figura 75. Constructor del Raycasting.	75
Figura 76. Actualización de la nueva posición en el espacio.	75
Figura 77. Alineación de los marcadores en la columna del feto.	75
Figura 78. Aplicación del SLAM para la ubicación de un robot.	77
Figura 79. Declaración de variable.	79
Figura 80. Información del IMU del teléfono.	79
Figura 81. Seguimiento sin el uso de marcadores.	79
Figura 82. Ubicación fija del modelo en el entorno.	80
Figura 83. Giro de la cámara sin afectar la ubicación fija del modelo.	80
Figura 84. Prueba de la aplicación con el cliente.	84

Lista de tablas

<i>Tabla 1</i>	<i>Factor de escala al importar modelos a Unity.</i>	<i>26</i>
<i>Tabla 2</i>	<i>Modelos 3D, Archivos genéricos y de propietario</i>	<i>27</i>
<i>Tabla 3</i>	<i>Programas de modelado 3D y las escalas que utilizan</i>	<i>28</i>
<i>Tabla 4</i>	<i>Matriz de decisión del método de captura</i>	<i>36</i>
<i>Tabla 5</i>	<i>Requerimientos de la aplicación</i>	<i>48</i>
<i>Tabla 6</i>	<i>Especificaciones de la aplicación</i>	<i>49</i>
<i>Tabla 7</i>	<i>Valores numéricos de las especificaciones</i>	<i>50</i>
<i>Tabla 8</i>	<i>Tabla de evaluación de las propuestas.</i>	<i>54</i>
<i>Tabla 9</i>	<i>Tabla de Cayley</i>	<i>66</i>
<i>Tabla 10</i>	<i>Encuesta de evaluación</i>	<i>83</i>
<i>Tabla 11</i>	<i>Resultados de la encuesta</i>	<i>85</i>

Introducción

En el trabajo presentado a continuación se desarrolla el proceso de generación de un software para dispositivos móviles, cuyo objetivo es presentar al usuario un compendio de modelos de fetos humanos, los cuales serán visualizados por medio de la tecnología de realidad aumentada.

En el capítulo 1 se listan los objetivos principales y su subdivisión en objetivos secundarios que se pretenden alcanzar en el desarrollo del trabajo.

El capítulo 2 desarrolla los argumentos que justifican la importancia del proyecto. Listando argumentos de índole económica y social.

En el capítulo 3 se hace una recopilación de información necesaria para dar contexto a los capítulos posteriores. Esto incluye temas de realidad aumentada, su relación con tecnologías similares y sus usos. Posteriormente, se hace una breve descripción del desarrollo embrionario para recalcar la importancia de generar el compendio digital al finalizar el trabajo.

El capítulo 4 se divide en dos partes principales referentes a la captura y sanación de modelos, respectivamente. La primera parte comienza con una introducción al tema de escalamiento para posteriormente describir el proceso de diseño en la captura de modelos digitales. La segunda parte del capítulo describe el proceso de reparación y adecuación de los modelos por medio de programas de computadora.

Una vez se tienen los modelos, se procede a integrar la aplicación, este proceso es el enfoque del capítulo 5, el cual comienza describiendo el proceso de diseño y elección de la plataforma y modo de visualización adecuados. La segunda parte del capítulo describe la creación de la aplicación incluyendo la implementación de la realidad aumentada.

Con la aplicación terminada se procede a realizar una prueba con el cliente, esta prueba, junto con sus resultados y una discusión sobre los mismos son el tema del capítulo 6.

El capítulo 7 presenta las conclusiones que se obtuvieron con base en los resultados de la prueba y el proceso de diseño. Analizando si se alcanzaron los objetivos planteados. Finalmente el capítulo 8 lista el trabajo que se debe realizar a futuro.

1. Objetivos

1.1 Objetivo general

El objetivo general de este trabajo es el desarrollo de un compendio de modelos de fetos humanos, el cual podrá ser consultado en una aplicación de dispositivos móviles haciendo uso de técnicas de realidad aumentada (RA) adicionalmente, el desarrollo del trabajo presente tiene la posibilidad de derivar en herramientas para los estudiantes de la universidad.

1.2 Objetivos específicos

Para la realización de este objetivo general, es necesario el cumplimiento de varios objetivos específicos los cuales surgen de la división del problema en subtarear.

- Evaluar, seleccionar y utilizar métodos de generación de modelos en tercera dimensión a partir de objetos reales.
- Acondicionar los modelos generados para poder ser utilizarlos en entornos virtuales.
- Diseñar la interfaz gráfica del compendio.
- Unir modelos y elementos interactivos mediante programación en la interfaz gráfica.

2. Justificación

De acuerdo con la UNESCO el “desarrollo sostenible” satisface las necesidades de la generación presente, sin comprometer las necesidades de las generaciones futuras. Y en la cual existen cuatro dimensiones en el proceso: la sociedad, el medio ambiente, la cultura y la economía. Por lo cual el desarrollo sostenible de un país implica una serie de procesos y medios para lograrlo, por ejemplo: la investigación y tecnología, la educación y la capacitación, entre otros. [1]

Por esta razón, es la tarea de cada país llevar a la vanguardia en el desarrollo educativo, enfatizando en dar uso a las nuevas tecnologías que se empiezan a emplear en importantes instituciones educativas y empresas en el mundo.

“La educación médica ha puesto el énfasis en estudiar la enseñanza en los ambientes formales y escolarizados. Empero, la vida profesional de los

médicos está llena de acciones de aprendizaje informales, entrelazadas con el ejercicio cotidiano de la clínica...”[2]

Así comienza un artículo que cuestiona cambios en los paradigmas educativos en el campo de la medicina, concluyendo en la necesidad de crear nuevos paradigmas y técnicas.

La información sobre el funcionamiento y desarrollo de las distintas partes del cuerpo usualmente proviene de dos fuentes principales, libros de texto especializados y análisis en autopsias. Estas fuentes proporcionan información a los estudiantes, pero en ocasiones no son suficientes por falta de modelos físicos con los cuales trabajar, dificultando el aprendizaje de procesos biológicos más abstractos.

Se tuvo un acercamiento al área de informática biomédica en la facultad de medicina de la UNAM, durante este primer contacto, se detectó esta necesidad en los alumnos de la sección de embriología. Se informó que los alumnos presentaban dificultad al estudiar el desarrollo embrionario, el cual consiste de procesos biológicos complejos y presenta varias fases clave en su desarrollo.

El problema de los alumnos, radicaba en que no contaban con modelos para estudiar estas fases en un entorno tridimensional, limitándose a fotografía y diagramas provenientes de libros de texto. Estos recursos no dan una idea completa de la forma del feto en desarrollo, por lo cual era necesaria la implementación de una herramienta o recurso que ayudase a los alumnos a visualizar de una mejor manera estas estructuras.

Diversas tecnologías han sido empleadas y estudiadas como complemento educativo. Una de estas tecnologías es la Realidad Aumentada (RA), a veces denominada también como Realidad Mixta (RM). La cual permite observar imágenes digitales superpuestas sobre un entorno real mediante un dispositivo móvil (celulares, tabletas digitales, etc.).



Figura 1 Dos ejemplos de uso de la realidad aumentada utilizando diferentes dispositivos [3]

El uso de la RA como herramienta educativa ha ido en aumento desde la década de los noventa donde se usaba para entrenar pilotos [4], generando numerosos prototipos y modelos los cuales se han estudiado. En la última década, ha aumentado la relevancia de esta tecnología, siendo cada vez más frecuente en contextos didácticos y en establecimientos como museos u monumentos.

Al ser una tecnología emergente, es necesario tener un panorama claro con respecto al estado del arte y la utilización de esta tecnología para justificar su utilización.

En la universidad normal de Beijing China, se realizó un estudio, el cual tuvo como propósito estudiar diversos artículos sobre el uso de la RA en educación, así como los resultados obtenidos en los casos. Se estudiaron 55 casos diferentes entre 2011 y 2016, se descubrió no sólo que el número de artículos ha ido en aumento a lo largo de los años, sino que también descubrieron que un 40.5% de los textos usaban la RA para enseñar conceptos de ciencia seguido por ciencias sociales con un 16.3%. El estudio, también subrayó que la mayoría de los artículos indicaban un mayor aprendizaje al utilizar la RA ya que se podían mostrar conceptos abstractos para mejorar la comprensión. [4]

Por otro lado, la IEEE realizó un estudio en el cual se analizaron 87 artículos científicos sobre pruebas con RA y su efectividad en entornos educativos. Observando la frecuencia de publicación de estos artículos en un periodo de 10 años y observando los resultados obtenidos en cada uno. Utilizando los resultados de los experimentos se llegó a un porcentaje de efecto del 56%, es decir, el uso de la RA tenía un efecto moderado sobre el proceso de aprendizaje. [3]

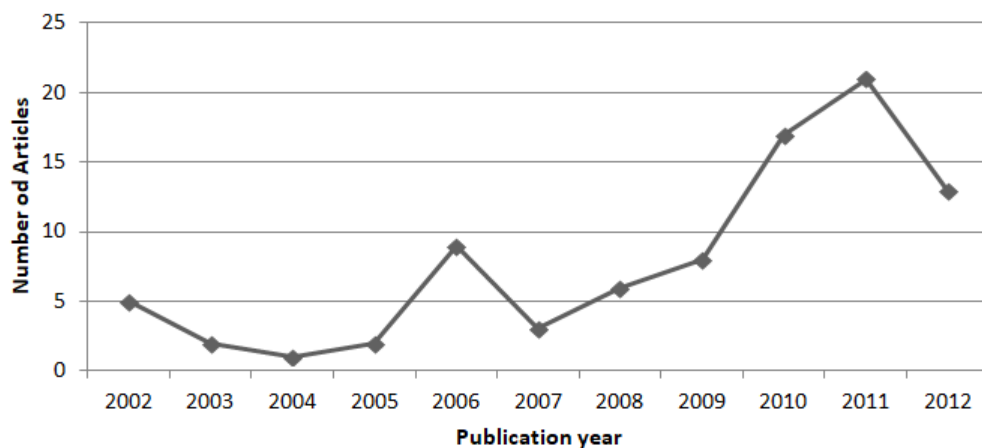


Figura 2 Número de artículos sobre la RA en entornos educativos según la IEEE [3]

Los estudios mencionados concluyen recalcando que sus resultados son variables ya que estos dependen de factores externos a la tecnología. Sin embargo, todos recalcan que el uso de la RA aumenta la efectividad del aprendizaje en los alumnos. Por lo que se alienta a la creación de nuevos usos de la RA, para generar una base de resultados más concreta.

Estos estudios dan el panorama en el cual la RA es una herramienta bastante útil como complemento educativo en varios niveles de escolarización, incluyendo el nivel universitario, es por lo anterior que se identifica un primer elemento para justificar la realización del presente trabajo.

La segunda justificación del proyecto se presenta en el aspecto económico de la industria y su tendencia en los últimos años.

La realidad aumentada ha estado en crecimiento desde los últimos 10 años. Con empresas como Google, Microsoft, Apple, Sony o MagicLeap liderando el mercado con sus avances tecnológicos. Al haber una gran cantidad de empresas dedicadas a distintas facetas de esta tecnología, existe, por ende, una gran competencia. La cual obliga a las empresas competidoras a mejorar su tecnología y servicios. Esto implica que nuevos dispositivos, sensores, tecnologías de rastreo entre otras tecnologías tienen que ser constantemente desarrolladas y lanzadas al mercado, resultando en un gran flujo de dinero desde y hacia estas empresas.

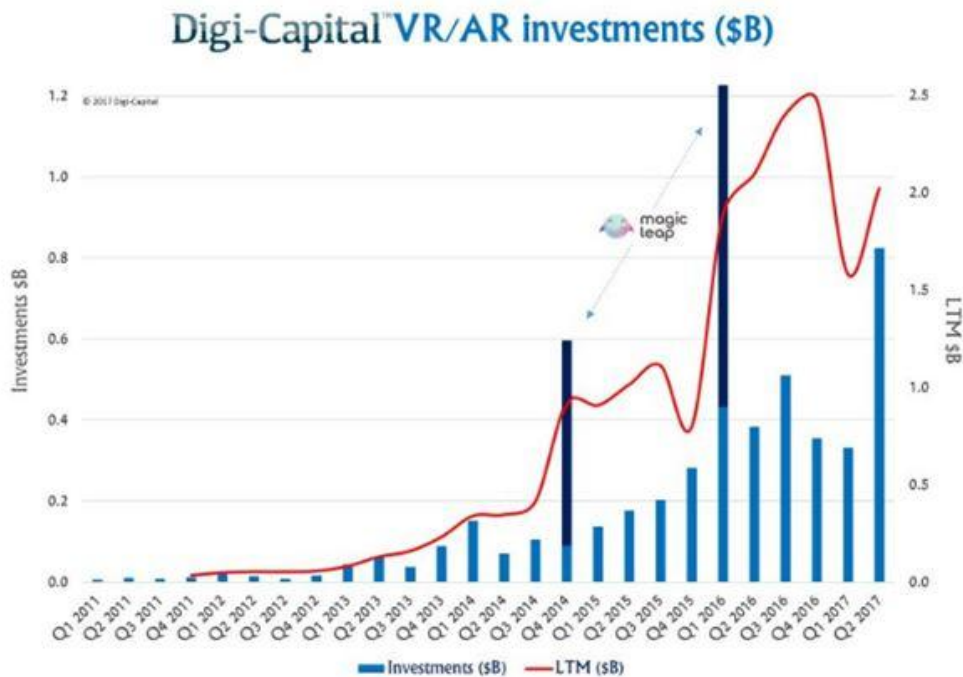


Figura 3 Inversión desde 2011 en miles de millones de USD en las industrias de RV y RA [5]

Según el sitio digi-capital, el cual hace análisis económicos alrededor de esta tecnología, la cantidad de inversiones dirigidas hacia el mercado de la RA y la realidad virtual (RV) ha aumentado en los últimos años. La inversión en el período comprendido entre Julio 2016 a Julio 2017 fue de 2 mil millones de dólares. Con 800 millones de dólares en el segundo cuarto de 2016,[5] siendo éste uno de los mayores períodos de inversión económica en los últimos años.[5] En la figura 3 se puede observar esta tendencia, siendo el primer cuarto de 2016 el período con mayor inversión económica.

Digi-Capital™ VR/AR investments (LTM to Q2 2017)

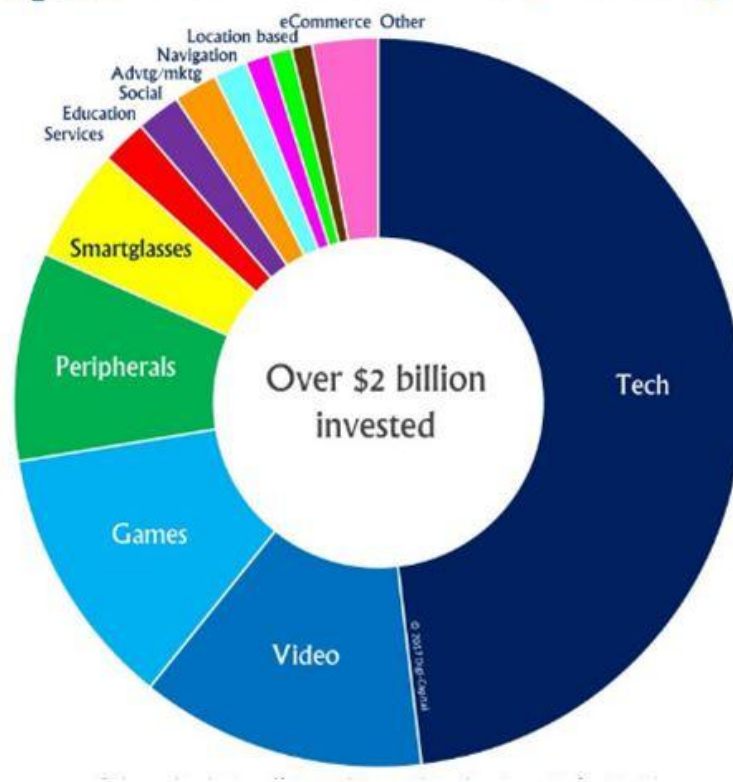


Figura 4 Inversiones por sector en el mercado de RV/RA en el periodo comprendido entre mediados del 2016 a mediados del 2017 [5]

Este crecimiento económico se debe en parte a la nueva tecnología que se está generando, pero también se debe al crecimiento en la cantidad de usuarios disponibles. Proyecciones indican que el mercado global de la RA va a ser de aproximadamente 60 mil millones de USD para el año 2021, con un estimado de casi dos mil millones de usuarios para ese mismo año. [6]

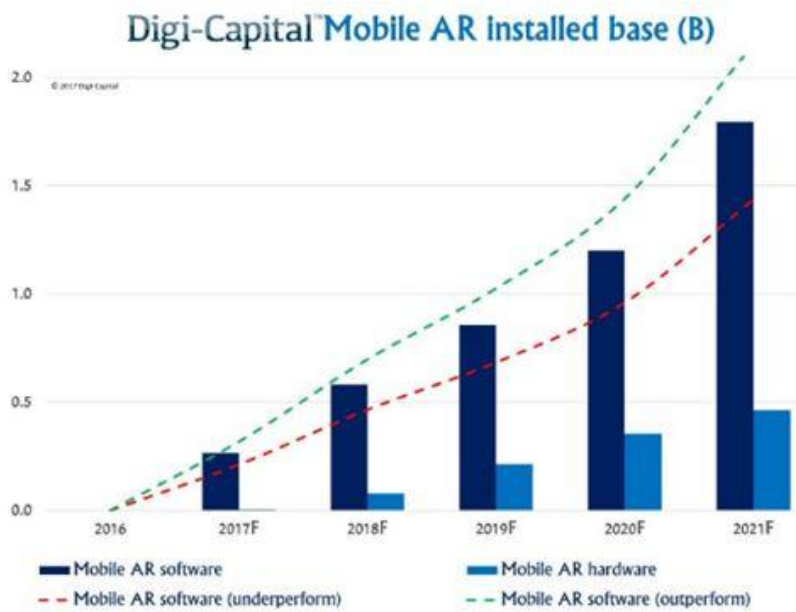


Figura 5 Cantidad de usuarios de tecnologías de RA móvil proyectado hacia 2021 [6].

Con la cantidad de dinero invertido y la creciente cantidad de usuarios, es de esperar que tanto el software como el hardware mejoren constantemente, se espera que para el 2021, la RA móvil dirija un mercado de aproximadamente 108 mil millones de dólares (mmUSD) dividido entre RV (25 mmUSD) y RA(83 mmUSD) [7] El crecimiento de las ganancias se puede observar en la figura 6.

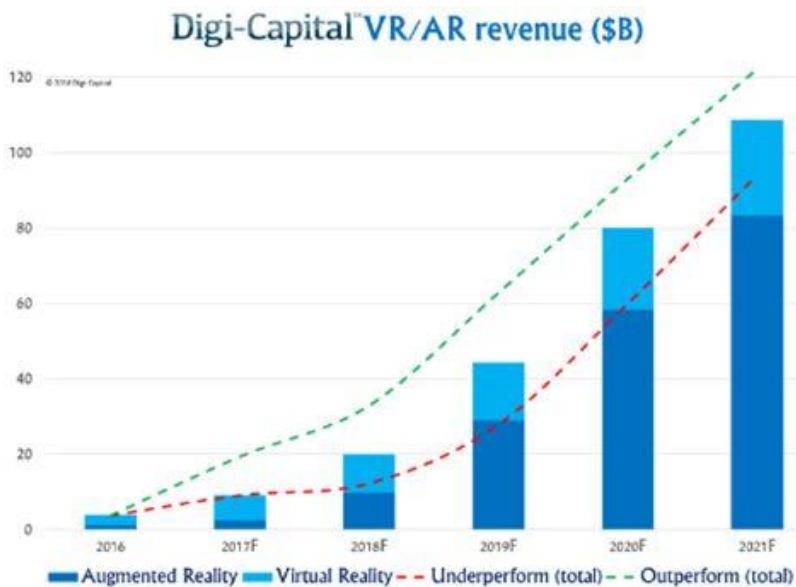


Figura 6 Proyecciones de ganancias en las industrias de RA y RV proyectadas a 2021 [7]

El aprovechar el gran crecimiento económico a nivel mundial de esta tecnología puede llegar a brindar muchas oportunidades de desarrollo tecnológico a futuro.

3. Marco teórico

3.1 Realidad Aumentada

Cada vez la sociedad manipula mayores niveles de información, la ciencia evoluciona rápidamente y se observa cómo la generación del conocimiento avanza con el paso de los años, a tal grado que difícilmente puede ser absorbido con suficiente rapidez por la población. Desde esta perspectiva la innovación tecnológica, plantea una renovación y ampliación de productos y servicios, requiriendo así, un incremento en la investigación enfocada al desarrollo de nuevas tecnologías. Este tipo de innovación es un elemento clave para la competitividad de un país y depende de la capacidad de su industria para innovar y mejorar. Es por esto que se ha denominado “tecnologías emergentes” a las innovaciones científicas que pueden crear una nueva industria o transformar una existente y ofrecen una rica gama de oportunidades en la sociedad.

Una de estas tecnologías emergentes es la Realidad Aumentada (AR por sus siglas en inglés), la cual ha permitido generar el interés de innovar la forma tradicional en la enseñanza, estudio, desarrollo y presentación profesional de proyectos educativos. Teniendo en cuenta el potencial que tiene esta tecnología, su correcta difusión y su uso concreto en distintos campos de estudio investigación o trabajo, depende del impulso que le otorguen las distintas instituciones educativas, privadas y gubernamentales, consiguiendo colocarse dentro de una competitividad por la innovación.

Como es el caso de los estudiantes del Instituto Tecnológico de Massachusetts (MIT por sus siglas en inglés) de Media Lab desarrollaron una tecnología corporal llamada “SixthSense” mostrado en la Figura 7, que convierte cualquier superficie en una pantalla interactiva. El dispositivo puede acceder a información digital, proyectar un teclado en la mano o realizar una llamada, por dar ejemplos. Esta investigación fue dirigida con el fin de crear un nuevo sexto sentido digital para los seres humanos. [8]

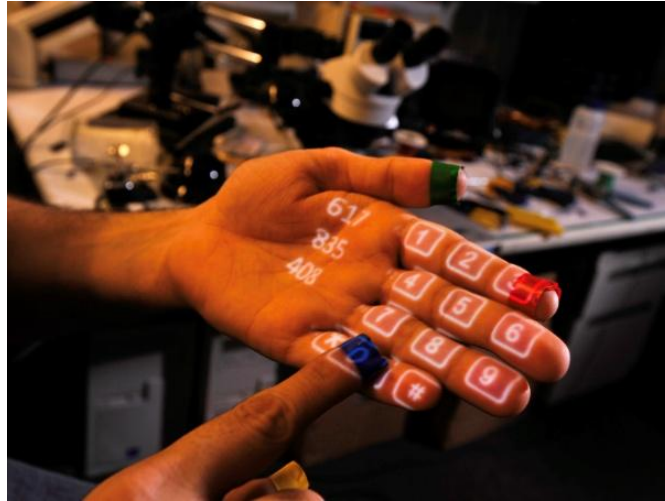


Figura 7 Sistema SixthSense[8]

Los usuarios podrán ver, tocar y escuchar gracias a la investigación AR y otras tecnologías de próxima generación, que ayudarán a enfrentar los obstáculos actuales y futuros frente a la adopción generalizada de AR.

3.1.1 Definición

La tecnología de la Realidad Aumentada consiste en superponer objetos, imágenes y animaciones generadas por una computadora en la visión del usuario en el mundo real, sobreponiendo estos objetos irreales sobre un terreno, paisaje o superficie que desee la persona en tiempo real. Mediante el uso de esta tecnología se pueden observar los elementos de una realidad virtual en el entorno real y permite al usuario obtener cierta información, de tal manera que complemente su percepción e interacción con el mundo real. [9]

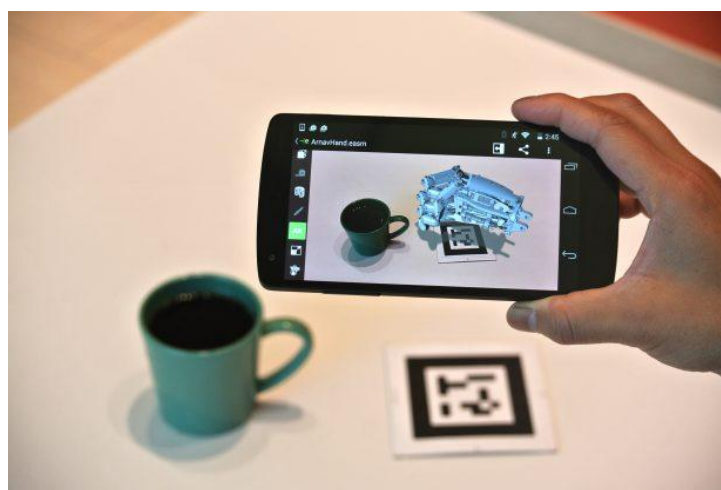


Figura 8 Realidad aumentada en un teléfono móvil. [10]

Por ejemplo, un espejo de maquillaje con AR que permita a los consumidores probar diferentes tipos de sombra de ojos sin tocar sus caras. Aunque, a menudo se relaciona junto con la Realidad Virtual (VR por sus siglas en inglés), la Realidad Aumentada es una experiencia totalmente diferente.

3.1.2 Comparación con Realidad Virtual y Realidad Mixta

En la literatura se pueden encontrar muchas definiciones de lo que es la realidad virtual, pero una de las más precisas es la que propuso A. Rowell:

“La Realidad Virtual es una simulación interactiva por computador desde el punto de vista del participante, en el cual se sustituye o se aumenta la información sensorial que recibe”. [11]–[13]

Cabe destacar que a partir de esta definición se pueden apreciar los elementos necesarios y básicos que están presentes en todo sistema de realidad virtual y que se diferencian en el modo de interacción que tienen los usuarios con los sistemas de realidad aumentada, los cuales se discuten a continuación:

- Simulación interactiva:

Un sistema de realidad virtual es una simulación en el sentido de que recrea un mundo virtual que sólo existe como una representación digital en la memoria del ordenador. El usuario puede escoger libremente su movimiento del mundo virtual y, por tanto, sus acciones afectan de forma directa a lo que verá en su entorno, pues el sistema responderá en tiempo real a sus acciones.

- Interacción implícita:

Estos sistemas se caracterizan por capturar las acciones del usuario implícitas en sus movimientos naturales. Un ejemplo de esto es el control de la cámara virtual, si el usuario quiere ver la parte del mundo virtual que tiene detrás, simplemente debe hacer el gesto natural de girar la cabeza. La revolución de la realidad virtual en la interacción hombre-máquina radica en que el usuario

deja de percibir los dispositivos (incluso del mismo ordenador) para pasar a interactuar directamente con los objetos de la escena.

- **Inmersión sensorial:**

El tercer elemento fundamental para la realidad virtual es la propia inmersión sensorial que permite a los usuarios la desconexión de los sentidos del mundo real y la conexión al mundo virtual. De los sentidos de la persona, la vista es el que nos proporciona una mayor cantidad de información y a la vez es la que genera la mayor sensación de presencia por consecuencia, el usuario deja de percibir el entorno que le rodea y pasa a estar inmerso dentro del mundo virtual que se recrea por el ordenador en tiempo real. [14]

Como ya se dijo, la tecnología AR es un sistema que establece la unión entre un contenido originado digitalmente con la realidad. A diferencia de la tecnología VR, la cual priva al usuario de su contexto real para adentrarlo por completo a un mundo virtual. Otra diferencia de la tecnología AR, es que sobrepone el contenido virtual en un contexto real, dando así una experiencia completamente diferente y al mismo tiempo poder interactuar con esa información virtual superpuesta. Con esto se logra la creación de nuevos mundos coherentes, complementados y enriquecidos al coexistir objetos del mundo real y objetos del mundo virtual.

A continuación, se describirá brevemente la interacción de la realidad aumentada con el usuario en la Figura 9:

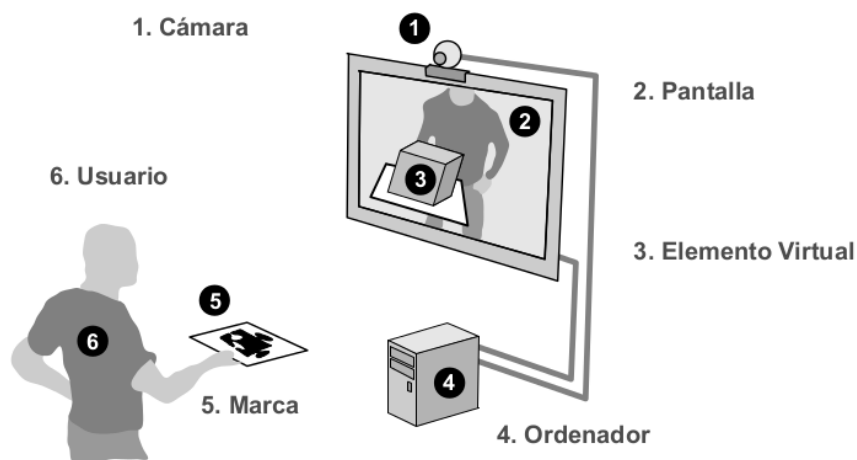


Figura 9 Diagrama general de la tecnología RA [15]

En la figura anterior, se observa la secuencia para visualizar los objetos virtuales. Haciendo énfasis en los principales componentes que se utilizan. Primero es necesario tener una cámara para que a través de ella se dé seguimiento al marcador objetivo y sea posible medir la orientación de los objetos. También se requiere de una pantalla para ver los objetos interactivos así como el diseño del software que permita la interacción de estos elementos. Por último, el papel o el marcador que se muestra en el esquema contiene la información codificada necesaria para que el software reconozca a qué paquete de datos se refiere, una vez identificado esto, el software permite la visualización en la pantalla de un objeto tridimensional interactivo. Este objeto interactivo permite al usuario observar desde imágenes, videos, modelos 3D, información textual, interfaces gráficas, etc.

Realidad mixta:

La tecnología de realidad mixta permite a los usuarios una de las mejores experiencias en los mundos tanto físicos y como digitales a través de una visión en tiempo real del entorno combinado con la superposición de objetos virtuales inteligentes, en la cual se permiten nuevas interacciones a través del gesto y entradas basadas en voz. Otra ventaja de los dispositivos de realidad mixta es proporcionar a los usuarios la realización de una tarea con sus manos libres con un controlador virtual, con el fin de operar una maquinaria y usar herramientas. Esto es especialmente útil para trabajos manuales en lugares remotos o peligrosos. [16]

Aunque los componentes de software y hardware existían previamente para la realidad virtual y aumentada, el avance tecnológico permitió elevar el grado de innovación suficiente para crear la experiencia completa de la realidad mixta. Los principales componentes de esta tecnología son:

- Reconocimiento de imagen:

La habilidad de un sistema de computadora de analizar de manera rápida y precisa una imagen e identificar las características de la misma. El ejemplo sencillo son las aplicaciones de realidad aumentada que usan códigos QR para reconocer objetos específicos en superficies para la interacción.

La realidad mixta, sin embargo, necesita usar más el reconocimiento de imagen avanzado para realizar el reconocimiento de objetos con el objetivo de identificar paredes, pisos, mesas y otros objetos dinámicos con los que se pueda interactuar con él, como se muestra en la Figura 10 a continuación.



Figura 10 Reconocimiento de superficies con realidad mixta [17]

- Localización simultánea y mapeo (SLAM por sus siglas en inglés):

Son un conjunto de tecnologías orientadas a localizar a una persona y mapear al mismo tiempo el entorno. SLAM es la clave para aplicaciones de realidad mixta. Para funcionar en entornos nuevos, desconocidos o cambiantes, esta tecnología necesita tanto crear continuamente un mapa de esos entornos como localizar y rastrear el movimiento de una persona dentro de ellos, como en la Figura 11. Los algoritmos SLAM aplicados a la realidad mixta hacen uso del reconocimiento de imágenes y los datos del sensor de profundidad, con el fin de calcular dónde está el usuario en el mundo físico. [18]

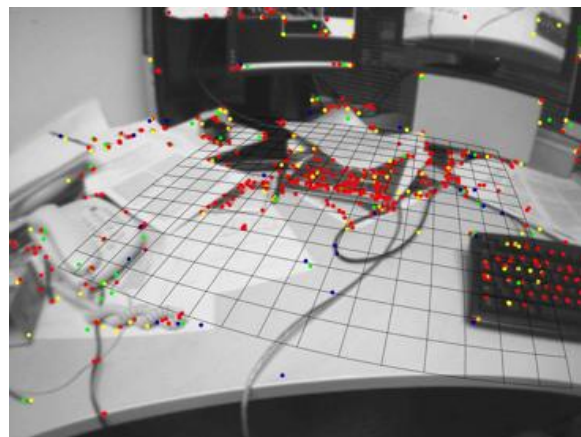


Figura 11 Superficie con elementos aleatorios [19]

3.1.3 Usos

La tecnología AR ofrece una infinidad de nuevas posibilidades de interacción en el mundo virtual y real, lo que permite tenerlo presente en varios ámbitos, como lo son la arquitectura, entretenimiento y ocio, la educación, medicina, entre otros. A continuación, se dará una breve descripción de algunas de sus aplicaciones:

- **Publicidad:**

Una de las mayores aplicaciones en el mercado es en el campo de la publicidad, en el cual se genera un factor sorpresa en los clientes, favoreciendo la exclusividad de la marca, producto o servicio. Para las empresas, esto aporta una gran ventaja sobre la competencia que no utilice este tipo de tecnologías ya que, al estar incluidos en el campo de visión del usuario, es más difícil ignorar estos anuncios. Adicionalmente, los costos son relativamente reducidos, lo que facilita su uso e implementación en empresas privadas y en organismos públicos. [20]

Un ejemplo de esto se observa con la tienda de zapatos Goertz, la cual instaló probadores virtuales de sus modelos en estaciones de tren alemanas, de forma que las personas que transiten en el lugar puedan disfrutar de un momento de compras. Tras escoger el modelo, el comprador recibía un código QR para finalizar el procedimiento de pago y envío. También se vinculaba con las redes sociales para pedir opiniones y sugerencias de su apariencia con sus nuevos zapatos. [21]



Figura 12 Demostración de la tienda virtual [21]

- **Arquitectura:**

Ofrece una presentación profesional de los proyectos arquitectónicos que están en la etapa de diseño y prototipos. La capacidad de realizar presentaciones con esta tecnología ofrece ventajas respecto a una presentación con un modelo convencional por ofrecer un mejor entendimiento a los clientes, desde ofrecerles distintas opciones de diseño, contener mayor detalle e información, en el modelo, generar variables de costo dependiendo de las decisiones tomadas incluso en el momento de la presentación, poder realizar cambios en los materiales antes de la construcción, o mostrar detalles específicos del proyecto. Todo esto es una pequeña muestra del potencial que ofrece esta herramienta dentro de este campo. [22]



Figura 13 Presentación de modelos 3D en edificios [22]

- **Ocio y Entretenimiento:**

El uso de la tecnología AR en este campo se ha centrado en juegos que salen de la pantalla para ofrecer experiencias únicas en consolas, móviles y tabletas. Se estima que el mercado de videojuegos en Estados Unidos alcance un valor de \$41 billones al final del año 2017. Seguido de otros mercados importantes como Japón. Por lo cual es comprensible que se esté apostando mucho por la realidad aumentada en este campo pues ésta puede aportar muchas nuevas posibilidades de entretenimiento. [23]

Un ejemplo del impacto generado en este mercado, es la compañía Niantic que logró recaudar 800 millones de dólares en sus primeros 110 días. Gracias a la aplicación conocida como “Pokemon Go”, el interés por la realidad

aumentada ha crecido, logrando captar la atención de empresas como Apple y Google con el objetivo de conseguir nuevas innovaciones tecnológicas. [24]



Figura 14 Aplicación móvil de Niantic [24]

- **Educación y Medicina:**

Estos dos campos están relacionados por las nuevas formas de aprender, estudiar y conocer las áreas del conocimiento. Para el caso de Medicina se estudia y analiza el detalle del cuerpo humano, es decir, sus órganos tanto internos como externos y de cómo afectan los medicamentos en simulaciones como se muestra en la Figura 15.

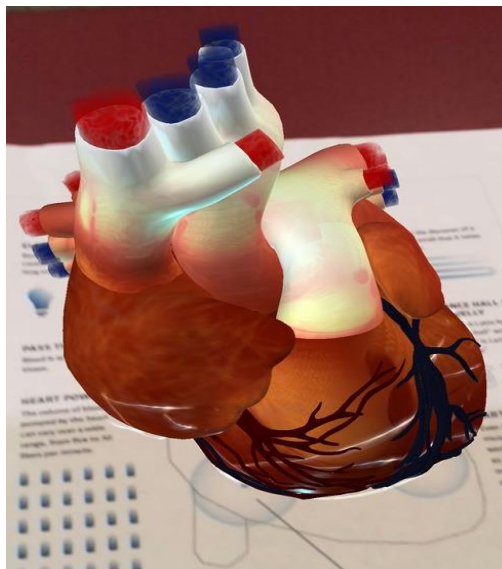


Figura 15 Modelo tridimensional de un corazón humano [25]

La unión entre la educación con la tecnología de AR se puede encontrar en la aplicación VR Dentist, utilizando realidad mixta para fines educativos para

dentistas en formación. La aplicación permite visualizar modelos y prótesis dentales en 360 grados, esto se puede observar en la Figura 16.



Figura 16 Demostración de la realidad mixta [26]

3.2 Desarrollo Embrionario

Debido a que el desarrollo del presente trabajo está relacionado con los temas de desarrollo embrionario de forma temática, se considera apropiado hacer una breve introducción al tema. Haciendo énfasis en los temas que justifican el desarrollo del trabajo. Los cuales son el estudio del desarrollo del embrión y la conservación de los mismos.

La ontogenia se define como “*Desarrollo del individuo, referido en especial al desarrollo embrionario*” [27]. Es la ciencia que estudia el desarrollo físico del cuerpo, incluyendo los procesos químicos y biológicos que intervienen en el mismo. El estudio morfológico de los embriones humanos permite identificar características similares en el desarrollo de otras especies, como, por ejemplo, peces. [28]

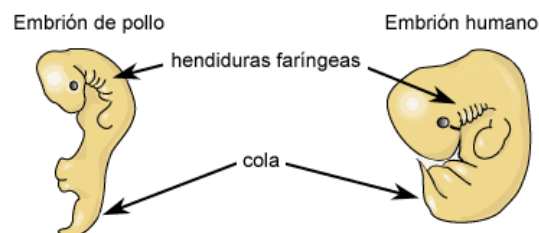


Figura 17 Comparación de los embriones de un humano y un pollo [28]

Se reconocen diferentes etapas del desarrollo inicial del cuerpo humano, diferenciadas por el nivel de complejidad del organismo.

- Fecundación, etapa en la cual el óvulo materno es fecundado por el espermatozoide, en el cual se fusionan los núcleos de las dos células resultando en un cigoto con 46 cromosomas.
- Durante los días siguientes, el cigoto se divide, llegando a las etapas de mórula y blástula, en estas etapas comienza la diferenciación celular lo que lleva en semanas posteriores al proceso de organogénesis. En esta etapa, la blástula se instala en la pared uterina.
- A las dos semanas de vida, la blástula ya es considerada un embrión. Las células se han diferenciado lo suficiente para generar el embrión, el cordón umbilical y la placenta.
- Entre la segunda y la sexta semana, las células del embrión se diferencian, generando tres capas de células de las cuales se generarán todos los órganos: El endodermo (aparatos digestivo y respiratorio), el mesodermo (aparatos circulatorio y locomotor) y el ectodermo (piel y tejido nervioso).
- Entre la séptima y la décima semana, continua el proceso de organogénesis, generándose órganos como las orejas, los ojos, los riñones, los aparatos circulatorio y digestivo y los brazos. Al finalizar este período, el embrión mide entre 3 y 4 cm y pesa de 2 a 4 g.
- A partir del tercer mes, el embrión ya tiene una forma característicamente humana y se considera como un feto. En este mes comienza a latir el corazón y comienza a haber movimiento. Muchos de los rasgos del cuerpo ya pueden ser distinguidos. Al finalizar este mes, el feto mide de 9 a 10 cm y pesa entre 15 y 20 g.
- Durante el cuarto mes de desarrollo, se desarrollan los órganos de la parte inferior del cuerpo. Es en esta etapa cuando ya se pueden apreciar los órganos genitales. Al finalizar este mes, el feto mide unos 16 cm y pesa alrededor de 150 g.
- A partir del quinto mes, no se producen demasiados cambios y el feto simplemente crece de forma constante. En el séptimo mes el feto aminora su crecimiento y gira 180° en la cavidad intrauterina. A partir de esta etapa pueden darse nacimientos prematuros.
- En el octavo mes se completa la maduración del feto y durante el noveno mes de gestación se terminan de definir algunos aspectos del feto como las características anatómicas de los genitales. El feto mide usualmente unos 50 cm y pesa entre 3 y 5.5 kg. [29][30]

En total, el desarrollo prenatal de un ser humano dura aproximadamente 266 días (entre 37 y 41 semanas). [31] Los infantes nacidos antes de este período de tiempo tiene una mayor probabilidad de perecer o desarrollar una discapacidad que perjudica su calidad de vida. (con un límite usual de 23 semanas, aunque hay casos de nacimientos previos en donde el infante sobrevive gracias a tratamientos), de acuerdo a la OMS, al año nacen en promedio unos 15 millones de infantes prematuros [32][33]

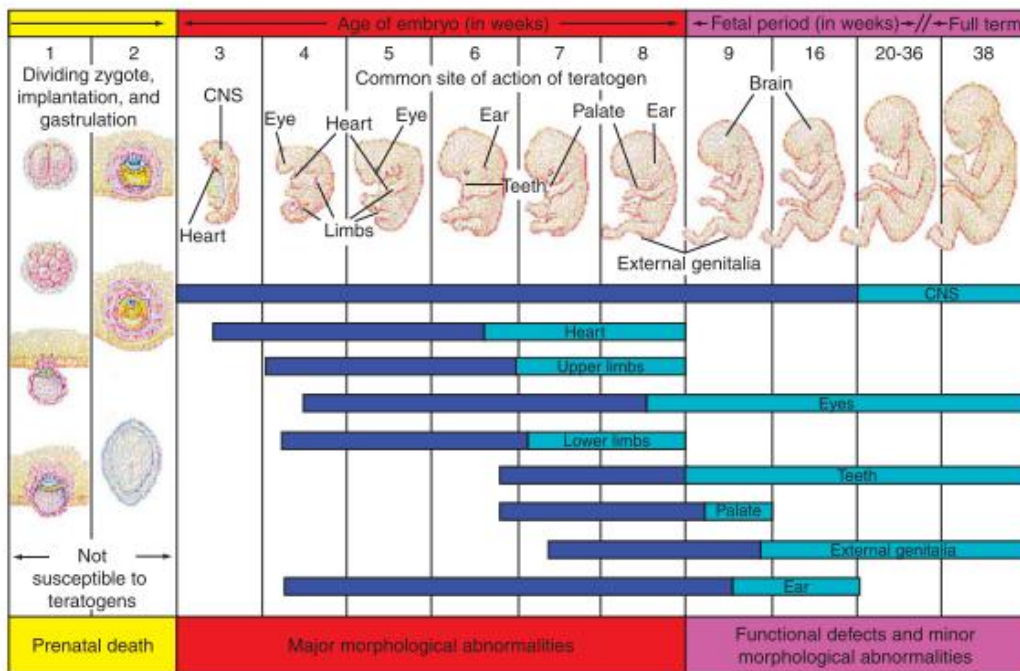


Figura 18 Diagrama del periodo de desarrollo de un feto humano. Incluyendo los períodos con mayor probabilidad de anomalías. [31]

Otra causa de muerte prematura son los defectos congénitos o malformaciones congénitas. Estos se definen como “anormalidades estructurales o funcionales, incluyendo desórdenes en el metabolismo que están presentes desde el nacimiento”. [34] Estos defectos son tratables si se cuenta con conocimiento previo sobre las causas y su efecto en el embrión en desarrollo

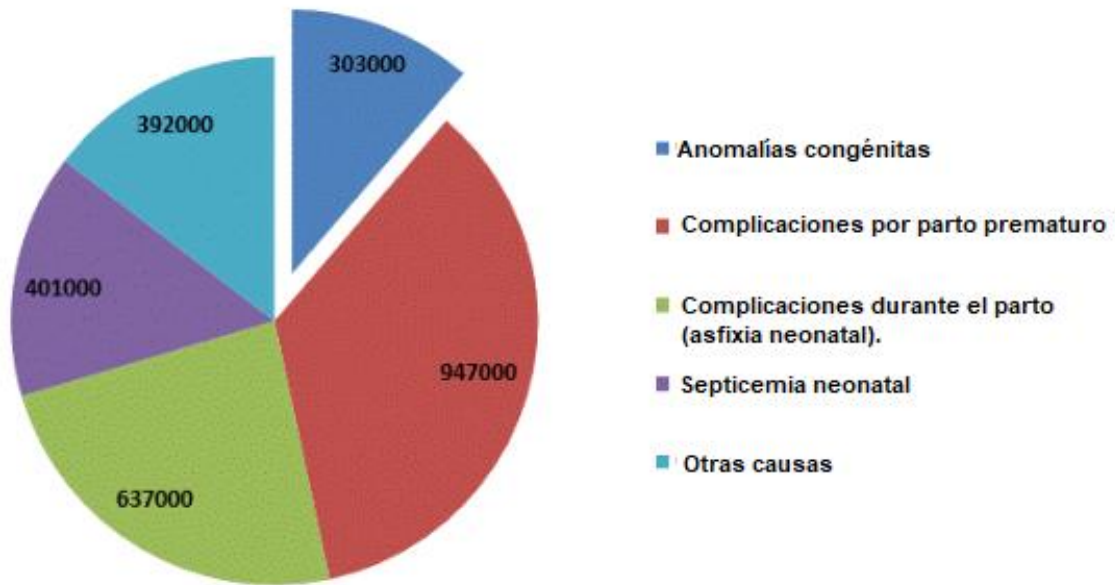


Figura 19 Causas de 2.68 millones de muertes neonatales en 2015 a nivel mundial de acuerdo con la OMS (WHO) (Imagen traducida al español) [35]

3.2.1 Preservación de material orgánico

Los modelos anatómicos son una buena fuente de información para el estudio de la medicina ya que permiten observar las estructuras y procesos biológicos en tercera dimensión, haciendo más sencilla la visualización del conocimiento teórico adquirido. [36]

Para poder asegurar la correcta visualización de este material biológico y evitar que queden inutilizables para el alumno y el personal docente, es necesario someterlo a un proceso de conservación para retardar o detener completamente el proceso de descomposición de la muestra. Estos procesos por lo general conservan la muestra en un entorno adverso a los microorganismos responsables de la descomposición o sustituyen tejidos del cuerpo con sustancias químicas. Entre los diversos procesos de preservación se encuentran:

- **Conservación en líquido:**

Para esta forma de conservación se sumerge el ejemplar en un líquido especial (puede ser alcohol 70° o formol al 10%) y el recipiente en el que se encuentra se mantiene cerrado. Estos químicos tienen el efecto secundario de aclarar un poco el color del ejemplar. Además, el alcohol 70° encoje ligeramente al ejemplar y el formol lo endurece un poco.

- **Mezcla Glicerina-Etanol:**

Una alternativa a la conservación por medio de formol, esta tiene la capacidad de conservar los tejidos en un buen estado y por períodos de tiempo superiores al mes. Además, presentan menos efectos nocivos a la salud en comparación con el formol.

- **Método de Walter Thiel:**

Método alternativo al uso del formol que se divide en tres procesos de fijación, desinfección y conservación. Entre los productos que ocupa se encuentran, 4-cloro-3- metilfenol junto con varias sales, ácido bórico y etilenglicol.

- **Plastinación:**

Proceso que permite la conservación de cuerpos o partes de los mismos mediante la eliminación de agua y lípidos del cuerpo y su posterior reemplazo con polímeros. El proceso puede llevar más de 2 meses entre los procesos de fijación con formol, deshidratación mediante acetona e impregnación con poliéster.

- **Diafanización:**

Proceso de conservación cuyo objetivo es tornar transparentes varias partes del ejemplar para poder apreciar estructuras y procesos internos. Esto se hace mediante una serie de procesos de fijación, deshidratación y transparentado (mediante una solución de KOH). Este proceso dura aproximadamente 20 días.

- **Corrosión:**

Conjunto de técnicas destinadas a la conservación y observación de conductos vasculares.

El uso de estas técnicas implica una inversión en los productos químicos necesarios y en el material biológico. De igual forma se requiere una inversión de tiempo ya que estos procesos requieren de semanas para poder preservar el ejemplar biológico. Adicionalmente, estos ejemplares conservados

requieren de cuidados posteriores para conservar su calidad, por lo que la capacidad de manipularlos es limitada. [37]–[39]

La digitalización de este tipo de recursos es una opción adecuada para aquellas instituciones que no cuentan con el espacio o los recursos para adquirir y conservar los modelos físicos. Un ejemplo de esto es la página de internet www.biodigital.com, la cual permite visualizar un cuerpo humano y sus componentes de manera digital.[40]

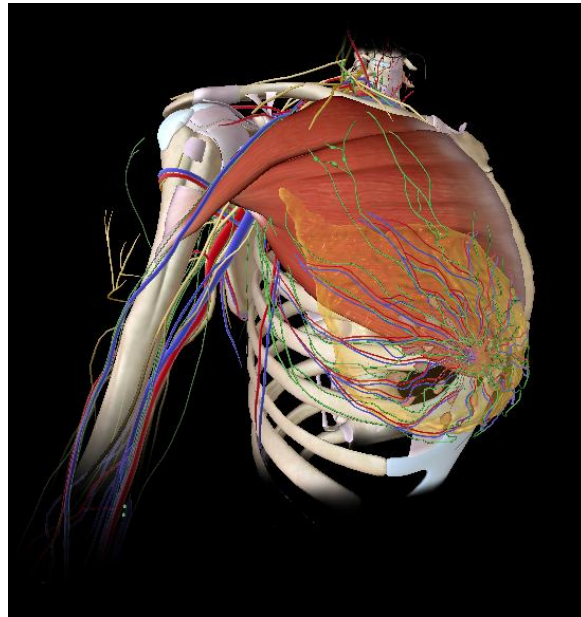


Figura 20 Modelo de pecho femenino de la página bio-digital[40]

4. Modelado

Esta sección comprende la captura y creación de los modelos necesarios para la aplicación que se ha descrito en capítulos anteriores, esto incluye una descripción de los métodos de captura, así como del tratamiento de los modelos para poder ser utilizados en el entorno gráfico.

En la primera parte del capítulo se hace la descripción del escalamiento en modelos generados por computadora y las dificultades que pueden surgir de ignorar las diferencias de escala entre modelos de distintos programas. Posteriormente se habla sobre las técnicas de captura de modelos, describiendo cual se eligió, así como los criterios tomados en consideración.

Posteriormente se enuncian los conceptos clave para la sección de procesamiento y filtrado de modelos, describiendo a la par los diversos problemas que puede tener un modelo generado por computadora. Finalmente se describe a detalle el proceso de tratamiento de los modelos incluyendo el software y técnicas utilizadas para adecuarlo al uso que se le dará en la fetoteca.

4.1 Captura y generación

Muchos de los modelos generados por computadora surgen de programas especializados en la creación y modificación de gráficos 3D. El modelado de objetos en 3D parte de la generación de formas geométricas simples que en conjunto dan como resultado una figura más compleja.

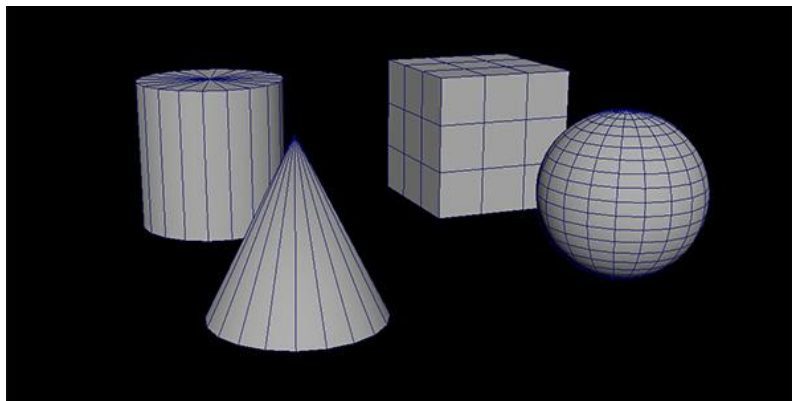


Figura 21 Ejemplo de figuras geométricas sencillas modeladas utilizando polígonos. [41]

Existen varios métodos de modelado que varían en torno de la creación de estas formas simples. Los tres principales son:

- Modelado por polígonos: Figuras generadas por unidades llamadas polígonos. Permiten un mayor detalle en los modelos generados.
- Modelado NURBS y de curvas: También conocidos como B-splines racionales no uniformes, estos son generados mediante fórmulas matemáticas. Son ideales para modelar superficies curvas.
- Modelado de subdivisiones (Mixto): Sus unidades poseen características de los polígonos y NURBS. Permitiendo tener características de los dos anteriores. [42]

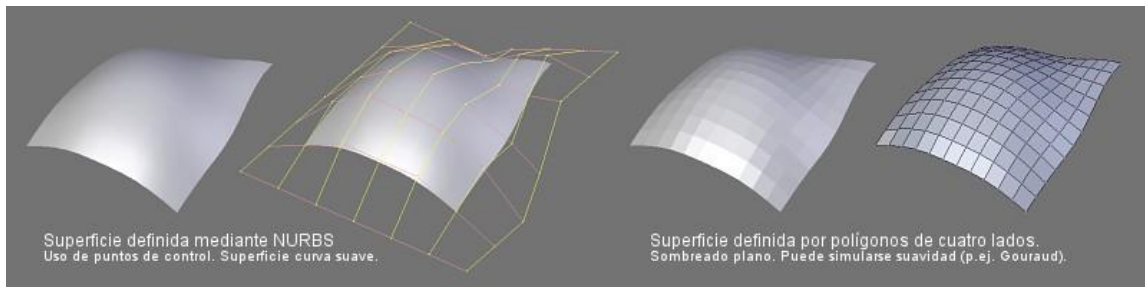


Figura 22 Superficie generada por NURBS (izq) y por polígonos (der).[42]

El método más popular de modelado 3D es el modelado por polígonos. Un polígono se define como *“Formas con lados rectos, definidos por puntos tridimensionales (vértices) y líneas rectas que los conectan (bordes). Generalmente, toman forma triangular o cuadrada”*. [41]

La amplia utilización de los polígonos se debe a la velocidad de procesamiento y a la exactitud de definición que permite. Sin embargo, para figuras más complejas es necesario tomar en cuenta que la cantidad de polígonos en un modelo afecta al rendimiento del equipo computacional al representar a este modelo. Mientras más polígonos tenga un modelo, mayor será la cantidad de memoria que este va a ocupar. Es por esto que es importante definir el nivel de complejidad que un modelo va a tener con base en el fin que se le va a dar al mismo.[42]

La complejidad de los modelos está ligada a la complejidad de las formas que lo componen. Por lo general, formas curvas o formas con un alto nivel de detalle requieren un nivel muy alto de polígonos, por lo que es recomendable generar estas formas utilizando otro método de generación de modelos.

4.1.1 Escalamiento

Según el diccionario de la lengua Oxford, se define escala como:

“Proporción que existe entre las dimensiones de un dibujo, mapa, plano, etc., y las de la realidad que representa”. [43]

El concepto de escala es dependiente del contexto que se le da. Por lo general, la escala se da en torno a un estándar con el cual se comparan las demás cosas. Un ejemplo de esto es el metro, el cual se usa como el estándar internacional para medir la distancia.

En el caso de los modelos generados por computadora, el concepto de escala es relevante sólo cuando se tienen dos modelos o más en un mismo entorno. Por sí, los modelos no tienen un tamaño dado, a menos de que se especifique en el programa en donde se está visualizando.

Al existir una gran cantidad de programas de generación y modificación de modelos generados por computadora, existen por consecuencia una gran cantidad de modelos con escalas distintas, lo cual dificulta la integración de un conjunto de estos a los programas o aplicaciones en los que se utilizan.

Un ejemplo de la discrepancia entre tamaños de modelos se aprecia en la figura 23, en la cual se aprecian modelos de objetos con escalas incorrectas. Esta imagen procede de un proyecto de recolección, inventariado y escalamiento de modelos de diversas fuentes. En la imagen de la izquierda se puede apreciar un grupo de modelos antes del re-escalamiento, mientras que en la imagen derecha se observa el resultado del algoritmo empleado para la operación. [44]

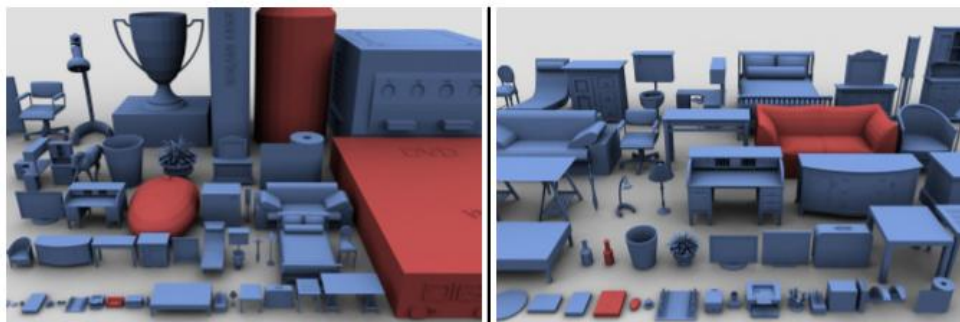


Figura 23 (izq) Conjunto de modelos de diferentes sitios con escalas distintas (der) El mismo conjunto de modelos re-escalados mediante un algoritmo. [44]

Esta discrepancia entre los tamaños aparentes entre modelos provenientes de distintos programas puede llevar a otros problemas. Un ejemplo de esto es el proceso de volver a escalar los modelos para colocarlos en un entorno virtual. Al no tener datos sobre el tamaño real de los objetos, el cambiar la escala de estos se vuelve un proceso tardado con resultados imprecisos. Lo cual es un problema para aplicaciones que necesitan escalas precisas.

Entornos virtuales y factores de escala

Para poder utilizar los modelos es necesario contar con un entorno gráfico en donde estos se puedan visualizar. Para este proyecto, se eligió Unity como el entorno gráfico a utilizar. Unity es un programa de generación de aplicaciones el cual toma elementos de otros programas (modelos, audio,

animaciones, etc.), y los puede unir en su interfaz gráfica, además de poder programar diversos comportamientos en Java o C#. Estos elementos hacen que Unity sea fácil de usar, lo cual agiliza el proceso de creación de programas.

Un ejemplo de la facilidad de usar Unity radica en la importación de modelos. Nativamente, tiene la capacidad de escalar los modelos importados dentro de la aplicación una vez han sido incluidos en el entorno gráfico.

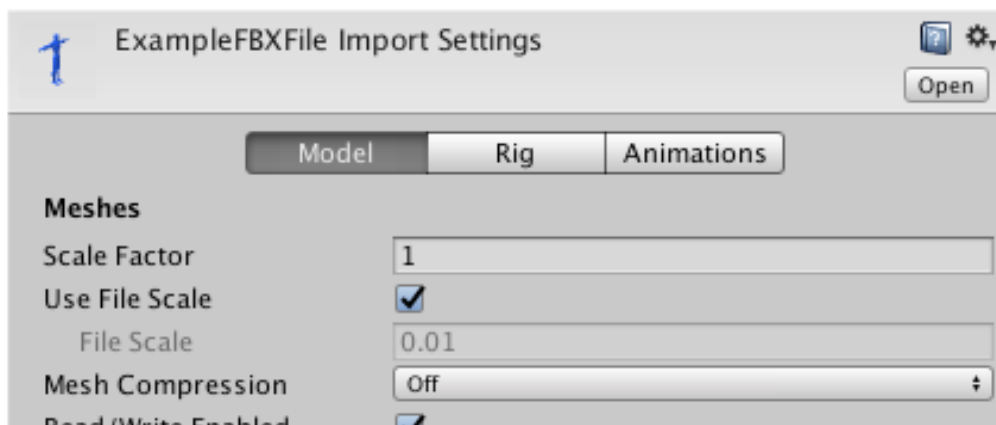


Figura 24 Opciones de escalamiento al importar un modelo a Unity[45]

Si bien, Unity permite la opción de re-escalar un modelo al ser importado, es más conveniente mantener una escala uniforme en los modelos importados para facilitar el desarrollo de cualquier aplicación. En la tabla 1 se listan los factores de escala por defecto que maneja Unity con distintas extensiones de archivo.

Tabla 1 Factor de escala al importar modelos a Unity.

Terminación	Escala (equivalente a una unidad en Unity)
. fbx, .max, .jas, .cd4, .obj	0.01
.mb, .ma, .lxo, .dfx, .blend, .dae	1
.3ds	0.1

[45]

Cabe mencionar, que no todas las extensiones son aceptadas directamente por Unity, algunas extensiones requieren tener el programa de origen específico instalado para poder importarlo exitosamente. En la tabla 2 se listan las extensiones por método de importación en Unity.

Tabla 2 Modelos 3D, Archivos genéricos y de propietario

	Archivos 3D exportados (genéricos)	Archivos 3D de software propietario
Tipo de archivo	.obj, .fbx .dae .3ds .dfx, .skp	.ma .mb .max .cd4 .blend
Ventajas	<ul style="list-style-type: none"> • Se pueden importar partes del modelo en lugar del modelo completo. • Usualmente, los archivos son más pequeños que los archivos de software propietario. • Permite un acercamiento modular al manejo de modelos. • Capacidad de importar modelos de software no compatible con Unity. • Capacidad de re- exportar modelos a software de modelado 3D. 	<ul style="list-style-type: none"> • Cambios realizados en el modelo original se reflejan en Unity
Desventajas	<ul style="list-style-type: none"> • Los modelos deben de ser importados nuevamente a Unity en caso de que se modifiquen en el software original. • Se requiere vigilar que no existan discrepancias entre el archivo fuente y el archivo exportado a Unity. 	<ul style="list-style-type: none"> • Se requiere de una copia del software propietario en todas las máquinas usando el proyecto de Unity. • Se requiere que todas las máquinas usando el proyecto de Unity tengan la misma versión de los programas. • Archivos pueden volverse grandes por información innecesaria. • Ralentización de los procesos de la computadora al tener el software propietario abierto durante la importación de modelos. • Conversión automática a .fbx, lo que impide encontrar errores.
Programas que generan estos archivos	Cualquier software de modelado 3D	Max, Maya, Blender, Cinema4D, Modo, Lightwave, Cheetah3D

[46]

Finalmente, para terminar esta sección, se listan varios programas de generación y modificación de modelos 3D generados por computadora, así como algunas de sus características:

Tabla 3 Programas de modelado 3D y las escalas que utilizan

Programa	Descripción	Escala utilizada por unidad
3DS Max	Aplicación de modelado 3D exclusiva para Windows. Ofrece muchas herramientas para crear y modificar los modelos. Precio alto.	1 cm ó 1 pulgada
TinkerCAD	Software gratuito en navegador que permite la rápida creación y exportación de modelos 3D sencillos. Estos modelos pueden ser llevados posteriormente a impresión 3D o a corte.	Ajustable de 0.1mm a 5 mm
Maya	Aplicación de Autodesk dedicada al modelado 3D. Ofrece una gran gama de herramientas. Precio moderado.	1 cm
Blender	Software gratuito de plataforma libre, dedicado al modelado 3D.	Variable. Puede sustituirse por unidades estándar o unidades imperiales.
AutoCAD	Software de diseño en 2D y 3D. Permite el diseño a detalle de los modelos. Muy usado en la industria y en aplicaciones industriales.	Varias unidades métricas e imperiales.
Skaneect pro	Software que permite la creación de modelos 3D por medio de un scanner. El programa también permite la modificación y exportación de los modelos en diferentes extensiones como .stl, .obj, .ply o .vrmf	Pulgadas, milímetros, centímetros y metros.
Solid Works	Software de creación de modelos 3D, diseñado para usos industriales, con opción de probar características físicas de los modelos.	Pulgadas, milímetros, centímetros y metros.
ReMake	Software de Autodesk que permite la creación y modificación de modelos por medio de fotometría. La creación de modelos se puede hacer localmente o en internet.	Varias unidades métricas e imperiales.

[47], [48], [49], [60], [61], [62], [63]

4.1.2 Técnicas de generación

La búsqueda de soluciones para la captura de los modelos abarca varios métodos ya que es necesario determinar varias características importantes que debe cumplir el modelo para poder ser utilizado en la aplicación.

El número de soluciones existentes para este problema es grande, pero depende en gran medida de las tecnologías que se encuentren disponibles. Cada una de estas soluciones difiere en características técnicas o monetarias. Por lo que es necesario probar diversos métodos de captura, para poder hacer una elección de la mejor solución para el problema.

Necesidades

Antes de mencionar los puntos a evaluar, es necesario establecer las necesidades del problema de generación de modelos. De acuerdo a lo solicitado por la sección de informática biomédica de la facultad de medicina de la UNAM, los modelos deben:

- Estar completos y con morfología correcta.
- Tener un tamaño adecuado para poder apreciarse correctamente.
- Tener una escala adecuada para poder medirse de forma correcta.
- Poder apreciarse con suficiente detalle para distinguir características físicas (cara, dedos, genitales).
- No requerir de mucha memoria para evitar crear archivos demasiado pesados.

Características a evaluar:

Escalamiento del modelo:

El concepto de escalamiento descrito en la sección anterior, comprende uno de los puntos importantes a considerar al elegir un método de captura de modelos.

Ya que en el trabajo se utilizan fetos humanos, es necesario que todos los modelos tengan una escala adecuada entre sí. Esto es debido a que una de

las funciones del programa es poder medir los fetos dentro de la aplicación, lo cual requiere medidas exactas para garantizar un buen aprendizaje.

Claro está que la escala de modelos es relativa dependiendo del software utilizado y el cuidado que se tenga en manejar debidamente el modelo durante su captura y procesamiento. Por lo tanto, para la evaluación de esta característica se considerarán las opciones de escalamiento del software vinculado con el método de captura.

Estructura del modelo:

Se entiende como estructura del modelo al conjunto de polígonos que componen al mismo. Un modelo con una buena estructura es aquel que se asemeja al objeto en el cual se basó.

En algunos casos, la información de los modelos puede corromperse, puede estar incompleta, o puede ser errónea por alguna falla en el método de captura. Esto causa que el modelo presente características no deseadas en su estructura como espacios vacíos, protuberancias o polígonos flotando por fuera del objeto. Estas imperfecciones se pueden arreglar con software de modelado. El tiempo requerido para reparar los modelos depende de cuántas imperfecciones estén presentes y de su magnitud.

Para la evaluación de este punto se tomarán en cuenta las imperfecciones generadas por cada método de captura de modelos. Idealmente, los modelos generados deberían presentar el menor número de imperfecciones posibles.

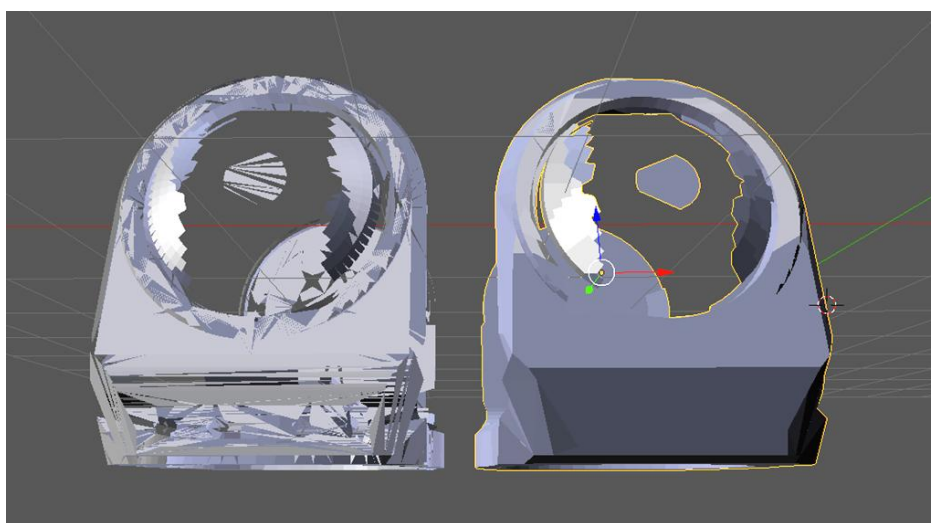


Figura 25 (izq) Ejemplo de un modelo con malla dañada. (der) El mismo modelo reparado [54]

Textura del modelo:

Una textura es una imagen digitalizada que se utiliza para cubrir la superficie de un objeto digital 3D o un objeto 2D. La textura le da al objeto la apariencia, colores y en ocasiones el sombreado del objeto.

Al ser un mapa de bits, las imágenes que se utilizan para la textura pueden tener varias calidades, dependiendo de la cantidad de información que posean. Las bajas calidades ocupan menos memoria, a consecuencia de perder calidad en los detalles de la imagen.

Ya que la aplicación que se está desarrollando requiere que los modelos de fetos sean detallados, es necesario que la calidad de las texturas sea alta para conservar la mayor calidad de detalles posible. También es importante que la mayor parte del modelo se conserve para evitar rellenar las texturas.



Figura 26 (Izq) Textura con baja resolución. (Der) Textura con alta resolución[55]

Tiempo de captura requerido:

Cada método de captura tiene un tiempo de captura distinto. Esto depende de varias cosas como el tiempo requerido de preparación previa del modelo, el tiempo que se requiere para preparar el equipo de captura y el tiempo que se tarda en generar el modelo en la computadora.

El tiempo requerido varía por caso. Para la evaluación de este punto se tomará en cuenta el tiempo en minutos para generar el modelo en la computadora, incluyendo la preparación del equipo necesario. Siendo considerados positivamente aquellos que requieran de un menor tiempo.

La combinación de estas características determina si un modelo es apto para ser usado. Dependiendo del uso que se requiera para el modelo va a ser la característica o características a las que se les dé prioridad, tomando en cuenta que las fallas en algunos de estos puntos pueden ser corregidas con software especializado de modelado 3D.

Métodos de captura

Propuesta 1: Kinect

Originalmente desarrollado para permitir controlar videojuegos mediante movimientos de partes del cuerpo, la ingeniería inversa permitió que los datos obtenidos por este dispositivo pudieran ser usados en otras aplicaciones. [56]

La tecnología del Kinect permite detectar la ubicación de partes importantes del cuerpo como la cabeza o los brazos mediante el uso de sus sensores. El dispositivo cuenta con una cámara RGB, la cual da información sobre los colores de la imagen captada y un emisor y receptor de rayos infrarrojos que permiten medir la profundidad del entorno.

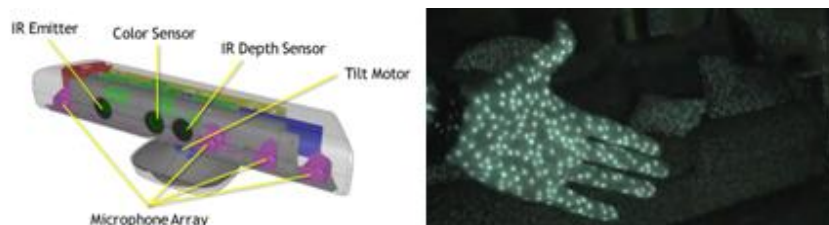


Figura 27 (Izq) Sensores presentes en el Kinect [57] (Der) Proyección de puntos infrarrojos por el emisor IR del Kinect [58]

El campo de visión es de 60 grados de forma vertical y 70 grados de forma horizontal. Adicional a esto, la distancia de operación óptima se encuentra entre ochenta centímetros y 4 metros.[58]

Debido a la distancia óptima necesaria para obtener imágenes y la forma en la que trabaja el sensor, es difícil obtener detalles pequeños en las imágenes capturadas. Un estudio realizado por el grupo de fotometría y geomática de Francia determinó que el error en las mediciones variaba entre 0.5 y 2.7 centímetros dependiendo de la distancia.[59]

Se probó esta tecnología para la captura de modelos utilizando el programa Skanect pro, el cual permite utilizar los sensores del dispositivo para generar una imagen en 3D.

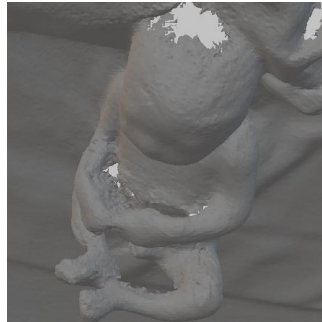


Figura 28 Modelo sin textura generado por Kinect

Como se puede observar en la figura 28, el modelo generado presentó una buena morfología, con pocas imperfecciones y el tiempo en que fue generado el modelo fue de menos de una hora. El software cuenta con opción de escalamiento lo cual asegura una escala constante entre modelos. Sin embargo, debido a los errores de medición de distancia del dispositivo, la textura presentaba muchos errores y un detallado pobre.

Propuesta 2: Escáner

Los escáneres digitalizadores especializados son dispositivos diseñados exclusivamente para obtener información de la forma de su entorno y traducirla en un modelo digital. Hay diversos tipos de escáneres, siendo la distinción más grande si son por contacto (Necesitan un elemento palpador el cual es un elemento invasivo) o sin contacto.

Los escáneres sin contacto funcionan de una manera muy similar al Kinect en función de que cuentan con un emisor y un receptor. Por lo general, el emisor envía un láser sobre el objeto y el receptor captan en tiempo real las diferencias de distancia entre el láser y el objetivo. Esta información es procesada por un software especializado que genera el modelo en la computadora.[60]

La característica principal de esta tecnología con respecto a las demás es la precisión que se maneja al capturar objetos. Algunos escáneres de grado profesional como el GoScan3D manejan resoluciones de 0.2 mm y pueden generar una textura a color detallada.[61]



Figura 29 Escáner profesional GoScan3D de Creafom [62]

Propuesta 3: Fotogrametría

La fotogrametría es definida por la Sociedad Americana de Fotogrametría y Teledetección como *“La técnica cuyo objeto es estudiar y definir con precisión la forma, dimensiones y posición en el espacio de un objeto cualquiera, utilizando esencialmente medidas hechas sobre una o varias fotografías de ese objeto”*. [63]

Es decir, la fotogrametría es la ciencia que permite obtener las medidas de un objeto mediante varias imágenes superpuestas del mismo.

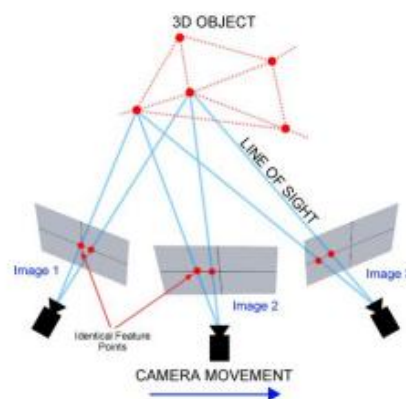


Figura 30 Diagrama del proceso de fotogrametría [62]

La técnica ha evolucionado con el paso de los años, haciendo uso de la evolución de la tecnología como las fotografías aéreas, las fotografías a color y las fotografías digitales. Es con esta última categoría con la cual se ha tenido un avance significativo, ya que el uso de las computadoras ha hecho más

rápido y sencillo el uso de esta tecnología. Hoy en día existen un gran número de aplicaciones y programas de fácil acceso como PhotoScan Pro o Autodesk ReMake dedicados a crear modelos y entornos digitales mediante el uso de esta técnica.

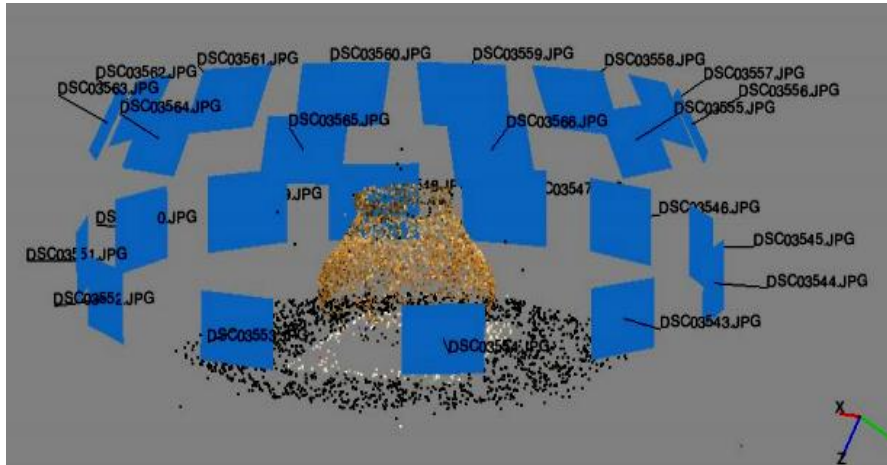


Figura 31 Nube de puntos creada a partir de fotografías [62]

Propuesta 4: Modelado por computadora

El diseñar modelos en 3D directamente desde la computadora requiere de software especializado y conocimiento en manejo del mismo, así como en creación y aplicación de texturas. Esta solución al problema brinda la capacidad de tener modelos muy detallados. Sin embargo, el hacer uso de esta solución requiere la inversión de tiempo para la adquisición de estas habilidades o requiere la contratación o reclutamiento de un individuo con conocimientos sobre la técnica.

Selección del método de captura

En la tabla 4 se agrupan las propuestas, las características a evaluar y cómo se evaluó cada propuesta.

Las casillas con la palabra “Cumple” indican que la propuesta cumplió con la especificación. Las casillas con la frase “No cumple” indican lo opuesto. Las casillas con la frase “Cumple con consideraciones” indican que la propuesta cumple parcialmente con la especificación y es necesaria la adición de elementos y trabajo adicional.

Tabla 4 Matriz de decisión del método de captura

Característica	Importancia	Propuesta 1 Kinect	Propuesta 2 Escáner	Propuesta 3 Fotogrametría	Propuesta 4 Modelado por computadora
Escalamiento	4	Cumple	Cumple	Cumple	Cumple
Calidad del modelo	2	Cumple	Cumple	Cumple con consideraciones	Cumple
Textura del modelo	1	No cumple	Cumple	Cumple con consideraciones	Cumple
Tiempo de captura	3	Corto	Corto	Medio	Largo
Disponibilidad	3	Cumple	No cumple	Cumple	-

Como ya se mencionó anteriormente, la elección del método de captura de modelos es dependiente de cómo cumpla con las especificaciones requeridas de la calidad de los modelos, del tiempo requerido y de la disponibilidad de la tecnología.

Se puede observar que la propuesta 1, satisface la gran mayoría de las características. Ya que con ésta es posible generar modelos digitales de forma rápida. Sin embargo, debido al rango efectivo de la tecnología, esta no puede captar detalles en el rango de milímetros, por lo que no es apta para este trabajo sin modificaciones severas.

La propuesta 2, satisface todas las características. Este método de captura puede generar modelos 3D digitales con un gran detalle y sin tener errores morfológicos con un tiempo de captura razonable. Sin embargo, no se pudo hacer uso de esta tecnología para este trabajo debido a la poca disponibilidad de la misma.

La propuesta 4, es descartada a pesar de cumplir con las especificaciones ya que llevaría un tiempo muy largo la elaboración de estos modelos.

La propuesta 3, cumple con los requerimientos de calidad de modelo con ciertas consideraciones. El método puede llegar a generar modelos con imperfecciones, pero por lo general, estas pueden ser corregidas fácilmente en software de edición. La textura de los modelos generada es aceptable para el uso que se le va a dar a los modelos y el tiempo de captura no es mayor a una hora por modelo. Otro punto importante es la disponibilidad de esta tecnología, ya que, de las cuatro propuestas, ésta es la más fácilmente accesible al existir aplicaciones y programas dedicados a esta técnica.

Analizando las distintas propuestas, se eligió la propuesta 3: Fotogrametría, como la tecnología para realizar la captura de modelos.

Proceso y consideraciones

Al escoger el uso de la fotogrametría para poder generar los modelos 3D digitales, es necesario preparar tanto el método de captura como el modelo físico. Para esto es necesario contar con un entorno en donde el modelo pueda ser colocado, una cámara digital y una computadora con un software especializado de fotogrametría.

Para la correcta toma de fotografías, es necesario tomar en cuenta las siguientes consideraciones:

- El espacio en donde se encuentra el modelo debe tener buena iluminación.
- Es preferente que las paredes y objetos cercanos al modelo no presenten colores uniformes.
- Las superficies color blanco o superficies que reflejen luz son perjudiciales para la creación del modelo en los programas de fotografía, por lo que es importante no colocar el modelo cerca de este tipo de superficies.
- De la misma forma, es importante que la superficie en donde se coloque el modelo contraste con este.
- La presencia de agua u otro líquido cerca o en el modelo no es recomendada ya que tiene el mismo efecto que las superficies reflejantes.
- Para una mejor captura del modelo, se debe de colocar alrededor de este, objetos que ayuden a identificar puntos importantes de posición. Es recomendable que estos marcadores contrasten con la superficie en donde se encuentra el modelo.
- Es importante que no se mueva el modelo durante el proceso de toma de fotografías. Esto se debe a que la técnica de fotogrametría depende del reconocimiento de objetos similares en las diversas fotografías, las cuales son superpuestas con base en esos objetos. Si el modelo se mueve, se pierde la referencia que se tenía en las fotografías.



Figura 32 Preparación del modelo sobre una superficie de color contrastante.

Una vez tomadas en cuenta estas consideraciones se procedió a la toma de fotografías. Se tomaron un aproximado de 50 fotografías rodeando al modelo en dos ángulos distintos para poder captar la mayor cantidad de detalles posible.



Figura 33 Toma de fotografías del modelo.

Con las fotografías en formato digital, el siguiente paso fue utilizar el programa de fotogrametría. En este caso se optó por usar Autodesk ReCap[64] ya que el programa permite generar de forma gratuita modelos por fotogrametría de forma digital en la nube.

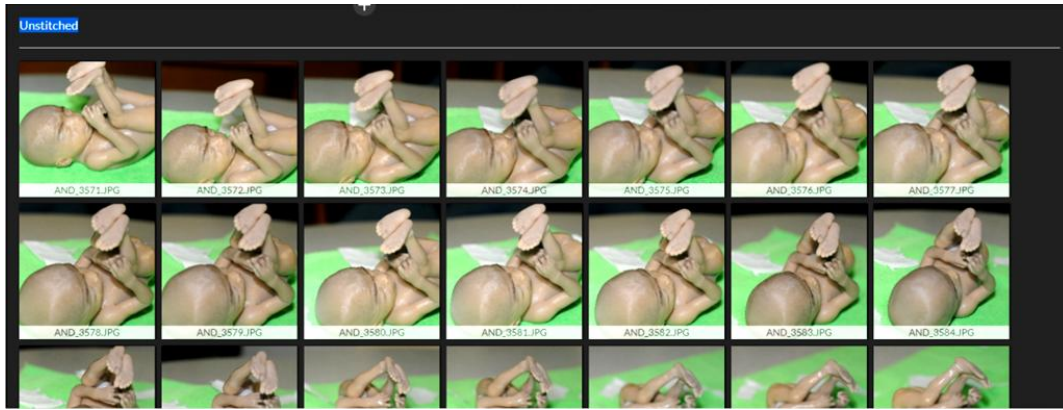


Figura 34 Selección de imágenes a usar para la creación del modelo

Una vez seleccionadas las imágenes a utilizarse, estas son usadas por el software para generar una nube de puntos correspondientes a las coincidencias entre las imágenes. Estos puntos son unidos mediante polígonos, con lo cual se genera el modelo.

En la figura 35 se puede observar el modelo obtenido por medio de la solución elegida.

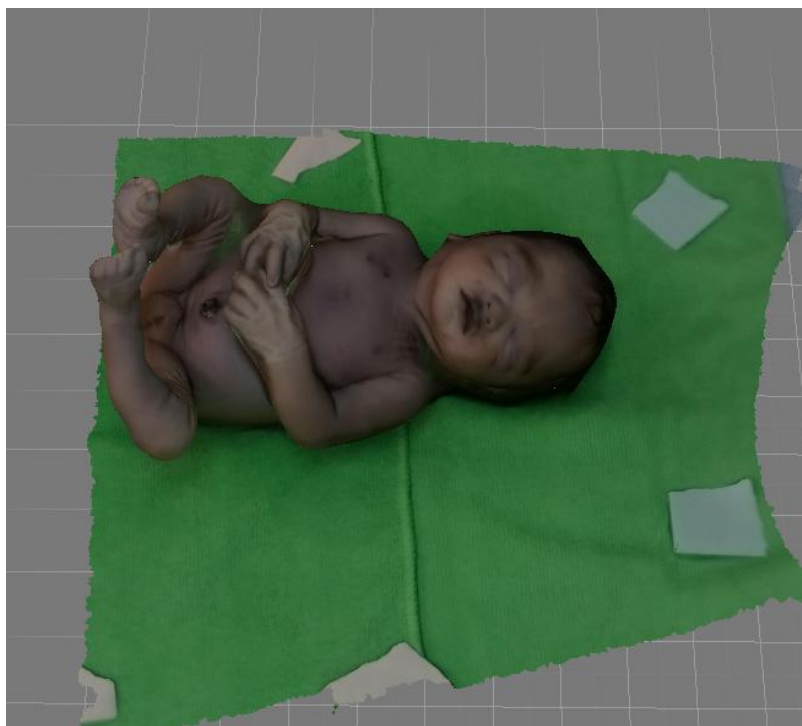


Figura 35 Modelo resultante de una captura de imágenes exitosa.

4.2 Procesamiento y filtrado de modelos

4.2.1 Descripción del problema

Debido al proceso de selección del método de captura, no se puede asegurar que los modelos generados por fotogrametría sean representativos del objeto real en su totalidad. Diversos factores como iluminación, superficies circundantes y el mismo proceso de creación de los modelos pueden generar diversos errores en la morfología y detallado de los mismos.

Entre los errores más comunes que pueden presentar los modelos se encuentran:

- Espacios vacíos de diferentes tamaños en ciertas partes del modelo. En el caso de la fotogrametría, siempre existe un espacio vacío en la parte inferior del modelo, ya que el modelo no se puede mover durante el proceso de captura.
- Partes del modelo que presentan uniones no existentes y uniones con la superficie en donde se encuentra el modelo.
- Errores de coloración en la textura del modelo debido a errores en el proceso de creación.
- Fallo de generación de geometría detallada. En el caso de este trabajo esto se ve reflejado sobre todo en los dedos de los fetos, los cuales no son generados por separado si no que se generan como una masa singular.

Dependiendo de la magnitud de estas imperfecciones, se decide si el modelo se desecha o se conserva para tratamiento. En caso de que este se conserve, es necesario eliminar las imperfecciones mediante el uso de programas de modelado 3D, los cuales poseen herramientas que permiten la preparación y modificación de modelos 3D digitales.

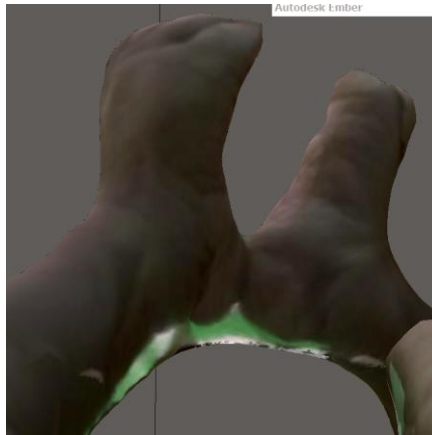


Figura 36 Acercamiento a los pies del modelo. Se observa una unión no existente en los talones del modelo, así como una falta de detalles en los dedos.

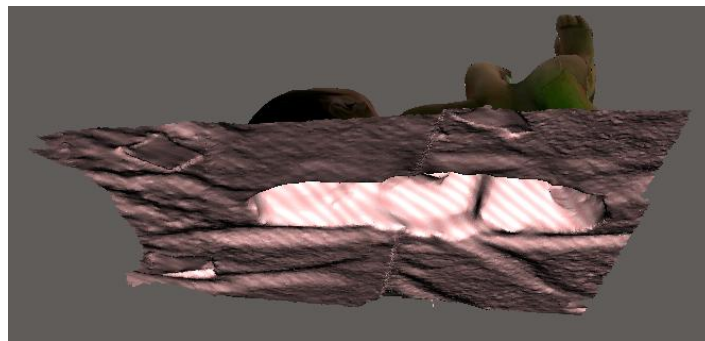


Figura 37 Vista inferior del modelo. Se observa que la superficie inferior presenta un espacio vacío debido a que no se puede mover el modelo durante el proceso de captura.

4.2.2 Software utilizado

Para el proceso de sanación de los modelos se necesita un software que permita importar y manipular los modelos para poder seleccionar y eliminar las imperfecciones en estos.

Para esto se optó por usar el software Autodesk Meshmixer, software de modelado virtual gratuito que permite modificar los parámetros físicos de modelos 3D compatibles. Entre los formatos compatibles se encuentra el formato OBJ, el cual es el que se usó para la creación de los modelos.

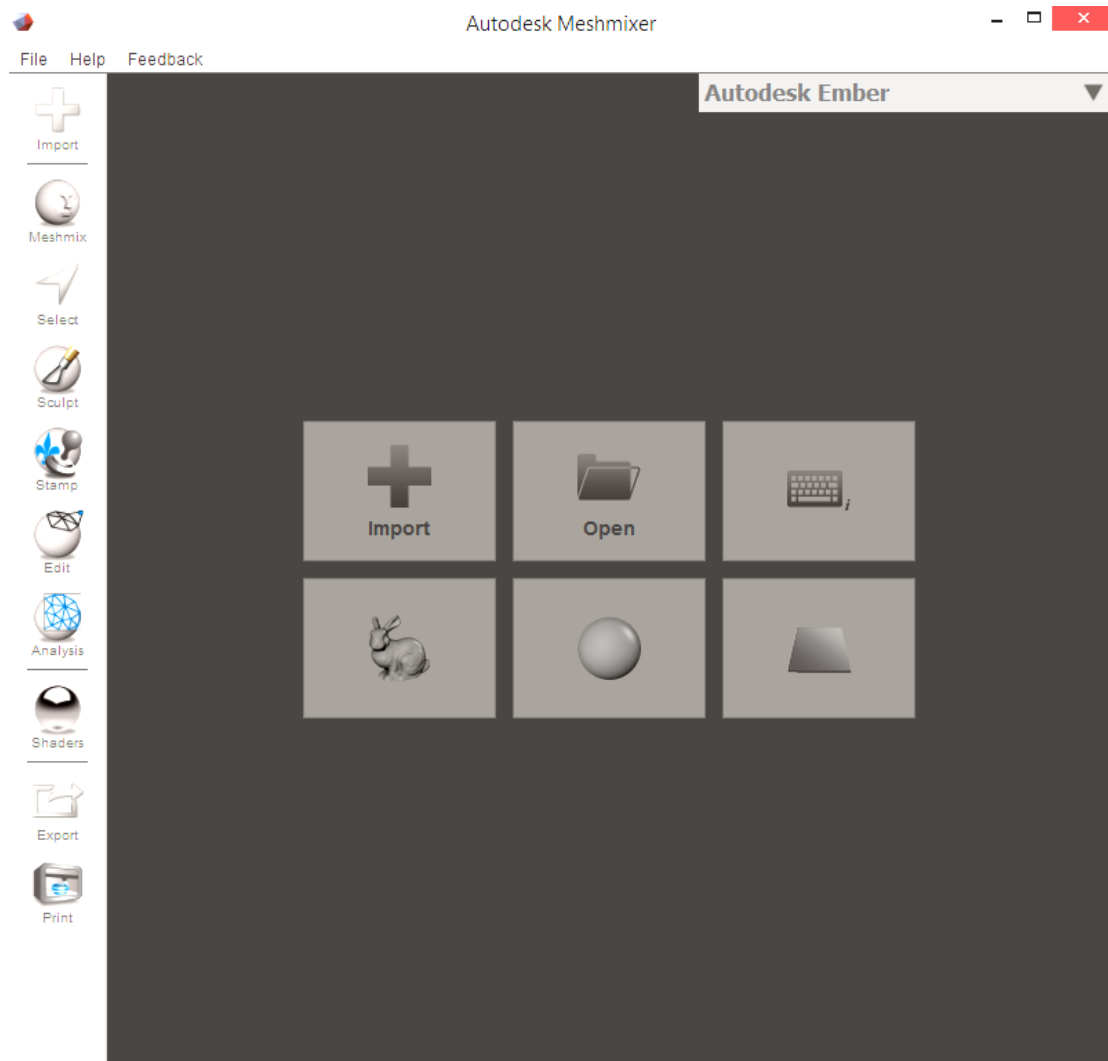


Figura 38 Pantalla inicial de Autodesk Meshmixer

4.2.3 Proceso de sanación

El primer paso en el proceso es la importación del modelo al entorno gráfico de Meshmixer. Para eso se elige la opción de Importar proyecto en la pantalla principal y se elige el archivo del modelo.

Ya con el modelo, se procede a cortar todas las partes innecesarias del mismo, empezando por la superficie que rodea al modelo del feto. Esto se hace mediante una selección manual de estas secciones no deseadas, las cuales son resaltadas en color naranja como muestra la figura 39.

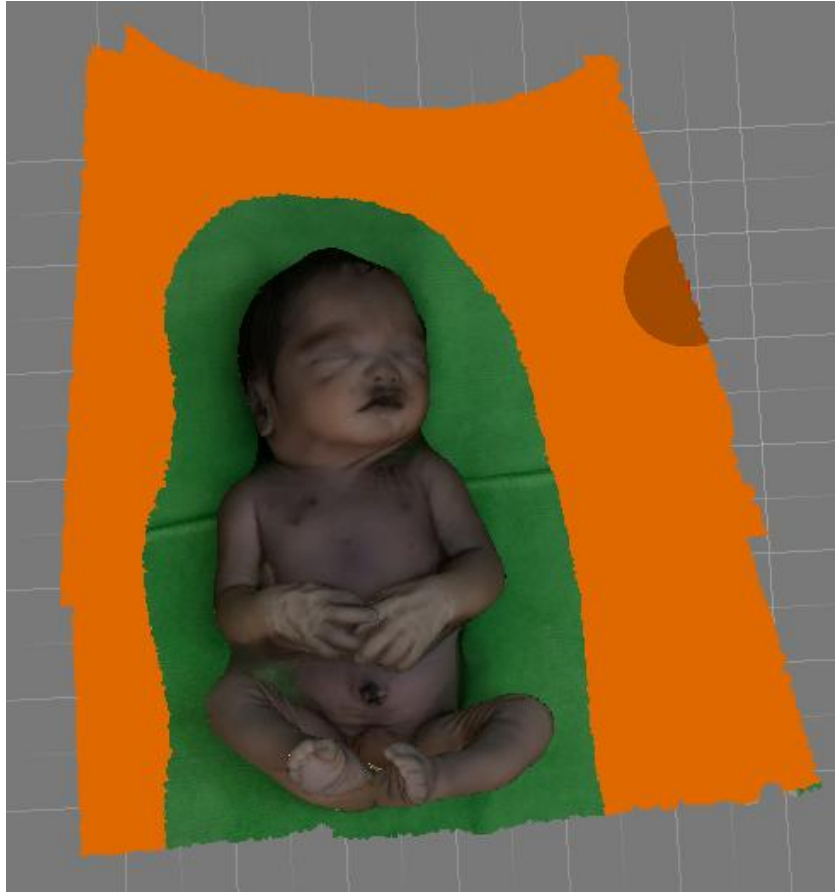


Figura 39 Selección de partes innecesarias del modelo para la operación de corte.

Al tener seleccionada una parte del modelo, aparece un menú que permite la edición de la selección. Esto incluye operaciones de modificación como suavizado, separación, unión, duplicación, extrusión, y eliminación. La opción de eliminación descarta la selección del espacio de trabajo.

Esta operación de selección y eliminación de zonas redundantes es iterada varias veces hasta haber eliminado toda la superficie circundante al modelo del feto. Esto incluye eliminar polígonos de unión entre las dos superficies mediante la modificación del tamaño de la herramienta de selección lo cual permite escoger polígonos específicos de esta parte del modelo. El resultado de esta operación se puede observar en la figura 40.

La unión entre los dos pies del modelo (observado previamente en la figura 36) se corrige mediante el mismo proceso de selección y eliminación de polígonos individuales.



Figura 40 Resultado de la operación de corte

Una vez eliminados los elementos no deseados, se procede al siguiente paso que consiste en corregir los espacios vacíos presentes en el modelo mediante operaciones de reparación.

Para acceder a esta opción se elige la opción análisis en el menú principal y del menú desplegado se selecciona la opción Inspector. Inspector es una herramienta que arregla de forma automática todos los vacíos en un modelo. El menú de inspector permite modificar la forma de la malla con la que se rellena el vacío y el tamaño mínimo de los elementos que se usan para crear esta malla. Una vez seleccionadas estas opciones, se procede a elegir el vacío al cual se le aplicará la operación.

En el caso del modelo utilizado, se utilizó la función “Smooth fill” con un límite mínimo de 0.005 milímetros.

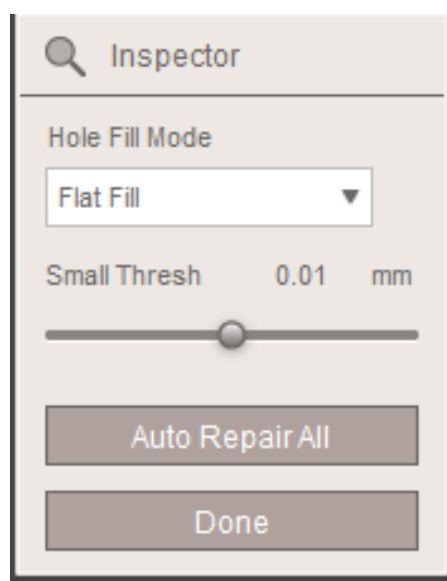


Figura 41 Herramienta de sanación de vacíos del modelo.

El sólido resultante de esta operación cubre el espacio vacío de la parte trasera de forma exitosa, pero carece de características físicas correspondientes al modelo.

Para poder solucionar esto, se hace uso de las herramientas de esculpir presentes en el programa, estas se dividen en herramientas de modificación 3D (para modificar el volumen) y modificación 2D; para poder adecuar la superficie de relleno se utilizó la herramienta Draw, la cual permite elevar o sumir las partes del modelo seleccionadas con el ratón con respecto al resto del modelo.

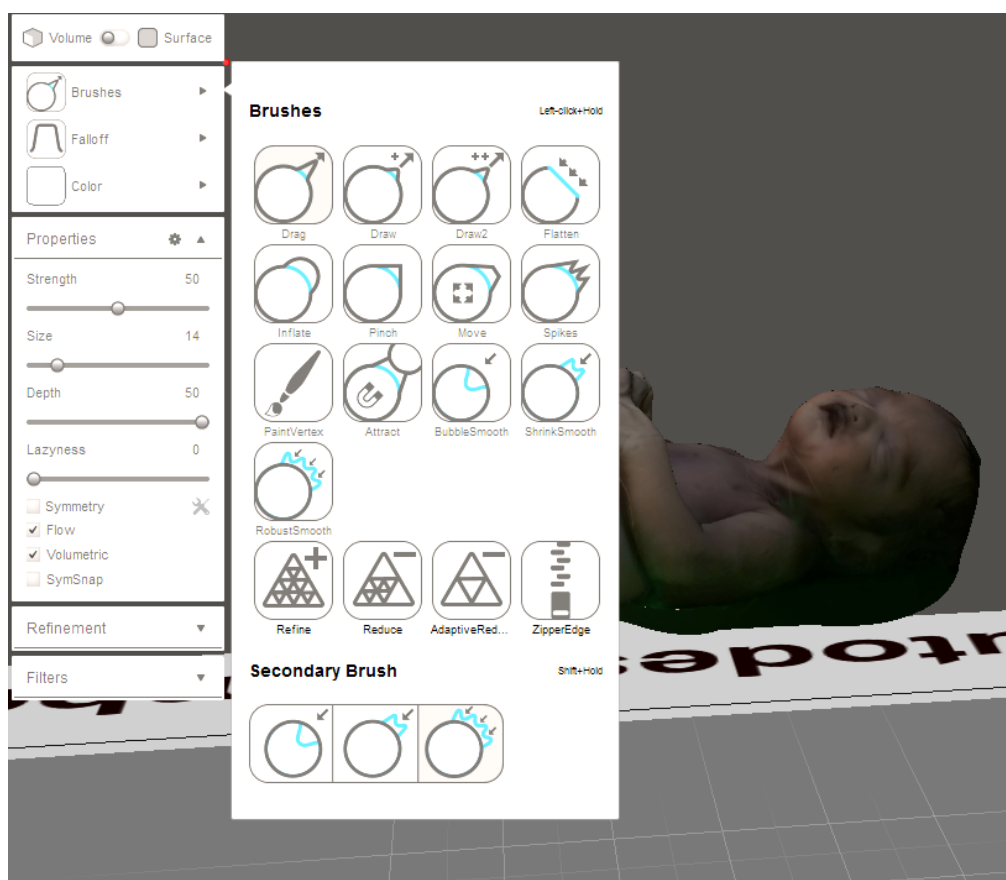


Figura 42 Herramientas para la modificación física del modelo.

Finalmente, se procede a ajustar el tamaño del modelo en la interfaz del programa. Para esto se utiliza la herramienta de transformación que se encuentra en la sección "editar" en la pantalla principal. La herramienta permite posicionar el modelo y escalarlo. Al escalar el modelo, el programa da la opción de escalar en los tres ejes de manera uniforme o por separado y permite modificar numéricamente el tamaño del modelo en el eje especificado.

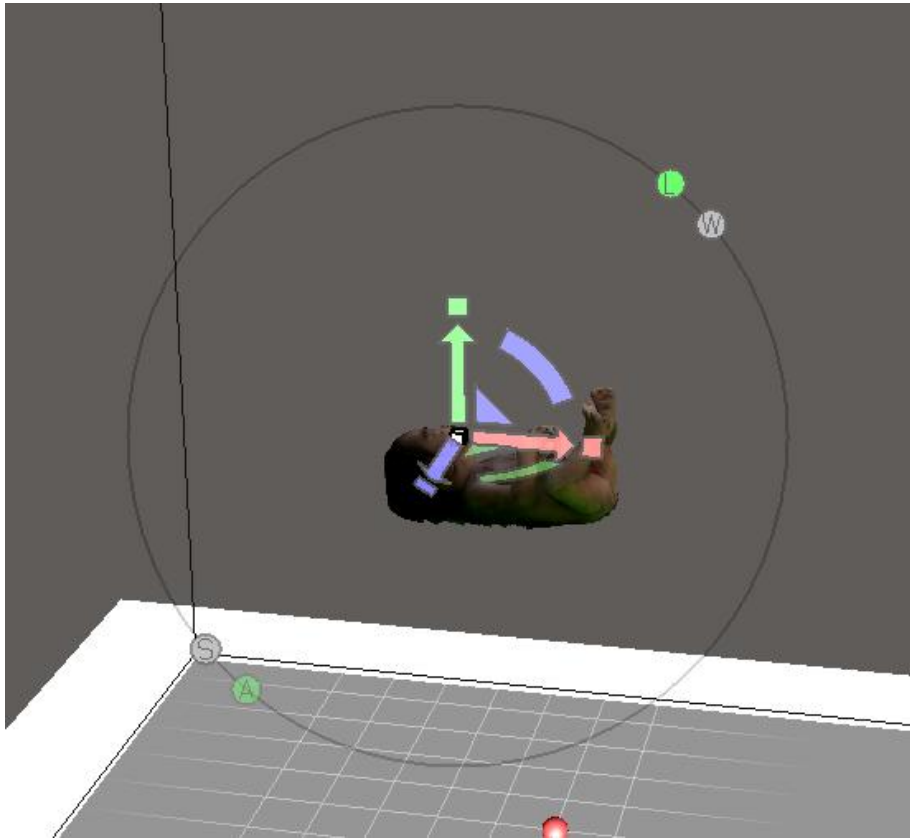


Figura 43 Herramienta de transformación del modelo.

4.2.4 Resultados

Gracias a las herramientas otorgadas por el programa Meshmixer, fue posible corregir la mayor parte de las imperfecciones de los modelos capturados. Las uniones innecesarias fueron eliminadas en su mayoría y los espacios vacíos fueron llenados.



Figura 44 Resultado de modelo con imperfecciones arregladas.

Los modelos terminados fueron posteriormente exportados en formato OBJ para poder ser utilizados en la plataforma de desarrollo Unity.



Figura 45 Segundo modelo con imperfecciones arregladas.

5. Desarrollo de la aplicación

En este apartado del documento abarca en un inicio, el diseño conceptual de la aplicación para dispositivos móviles, desde las necesidades y requerimientos hasta las propuestas y soluciones realizadas. Seguido del diseño a detalle que contiene la estructura y cuerpo de la aplicación.

Posteriormente, se detallan los componentes interactivos para el usuario, dada su importancia por ser un canal que nos permite la comunicación entre el usuario y la aplicación.

Finalmente, se hace mención de la utilización de la realidad aumentada con el entorno real sin la utilización de marcadores externos, siendo una mejora proactiva. Debido a que, la primera versión desarrollada y presentada tanto en ExpoDIMEI y en el Palacio de Escuela de Medicina, utilizaba marcadores.

5.1 Diseño Conceptual

Una vez obtenido el modelado tridimensional de los fetos reales que se detalló en el capítulo 4, se procede al desarrollo de la aplicación. Para esta etapa se cuenta con el apoyo de la información brindada por los Departamentos de Embriología e Informática Biomédica, ambos de la Facultad de Medicina.

5.1.1. Necesidades

Como ya se mencionó en el apartado de la Justificación del presente documento, en el acercamiento con la Facultad de Medicina a través de los dos departamentos de la misma, se detectaron las siguientes necesidades:

- Tener un compendio de fetos virtuales para la Fetoteca Digital.
- El nivel de detalle de los fetos virtuales debe ser cercano a la realidad.
- El desarrollo de la aplicación debe ser compatible para dispositivos Android e iOS.
- Debe brindar la información de las medidas en milímetros en cualquier parte del cuerpo del feto.
- Debe ser interactivo e intuitivo la acción de medir el cuerpo del feto.

5.1.2. Requerimientos

Una vez identificadas las necesidades, se procede a analizar esta información de tal manera que sea posible identificar los puntos importantes de las necesidades del cliente. Una vez identificados estos puntos importantes, se procede a escribir los requerimientos que cubren las necesidades.

Tabla 5 Requerimientos de la aplicación

Necesidades	Requerimientos
Tener un compendio de fetos virtuales para la Fetoteca Digital.	Contar con modelos tridimensionales Con base en los fetos que disponen en su almacén.
El nivel de detalle de los fetos virtuales debe ser cercano a la realidad.	Definir el método adecuado para la generación tridimensional de los modelos.
El desarrollo de la aplicación debe ser compatible para dispositivos Android y iOS.	Definir la plataforma de desarrollo para ambos sistemas operativos.

Debe brindar la información de las medidas en milímetros en cualquier parte del cuerpo del feto.	Proyección de la medición en la pantalla del dispositivo móvil.
Debe ser interactivo e intuitivo la acción de medir el cuerpo del feto.	Contar con elementos interactivos en el sistema, con el propósito de extraer la información necesaria del usuario para que pueda utilizar la aplicación de forma adecuada.

5.1.3. Especificaciones

Para satisfacer las necesidades es importante contar con parámetros o especificaciones, las cuales se obtienen al interpretar los requerimientos anteriores y que son las siguientes:

Tabla 6 Especificaciones de la aplicación

Requerimientos	Especificaciones
Contar con modelos tridimensionales con base en los fetos que disponen en su almacén.	Elección de los fetos reales.
Definir el método adecuado para la generación tridimensional de los modelos.	Porcentaje de exactitud en cada método de la fotogrametría.
Definir la plataforma de desarrollo para ambos sistemas operativos.	Capacidad del programa de desarrollo de cambiar de plataforma.
Proyección de la medición en la pantalla del dispositivo móvil.	Capacidad del programa de mostrar medidas en [mm].

Contar con elementos interactivos en el sistema, con el propósito de extraer la información necesaria del usuario para que pueda utilizar la aplicación de forma adecuada.	Interfaz de usuario sencilla.
--	-------------------------------

Siendo en cada especificación la asignación de un rango de valores con su respectiva justificación, que a lo largo del desarrollo de la aplicación deben cumplirse, como se muestra en la siguiente tabla:

Tabla 7 Valores numéricos de las especificaciones

Especificaciones	Valor	Justificación
Elección de los fetos reales.	2 fetos reales.	La técnica de la fotogrametría es muy laboriosa en su preparación. Es necesario realizar diversas fotografías en todos los ángulos posibles por cada feto.
Porcentaje de exactitud en cada método de la fotogrametría.	>70%	El nivel de detalle es importante para que el usuario observe características físicas en el cuerpo del feto.
Capacidad del programa de desarrollo de cambiar de plataforma.	Programa.	La estructura del código debe ser compatible para ambas plataformas.

Capacidad del programa de mostrar medidas en [mm].	Largo < 150 [mm] Ancho < 100 [mm] Profundidad < 100 [mm]	Con base en las medidas realizadas en el cuerpo del feto real.
Interfaz de usuario sencilla.	5 escenas de pantalla. < 15 botones interactivos. [4 a 12] marcadores de medición.	Se limita a estos valores para no producir fastidio en el usuario.

5.1.4. Concepto

Después de lograr analizar las necesidades de los usuarios a través de los requerimientos y especificaciones, se puede definir el concepto para el desarrollo de la aplicación, resultando lo siguiente:

Programa desarrollado y compatible en plataformas Android e iOS, que contenga un compendio de modelos de fetos humanos de manera virtual, y también permita observar a detalle las características del cuerpo del feto. Asimismo, tomar medidas con marcadores de manera intuitiva y que no sea fastidioso para el usuario.

5.1.5. Propuestas

El uso adecuado de la tecnología para dispositivos móviles es fundamental en el desarrollo de la aplicación, aunque no está explícito en las necesidades de los usuarios, la elaboración de este proyecto está encaminada a no depender de elementos externos, es decir, que sea exclusivamente para el uso de los teléfonos inteligentes con el objetivo de que los usuarios puedan utilizar dicha aplicación en cualquier lugar donde se encuentren.

Debido a que existen muchas maneras de abordar este trabajo con las diferentes características que nos ofrecen cada una de las tecnologías para realizar la Fetoteca Digital, la solución para este proyecto se basará sobre la

elección de una tecnología que se acople más a las necesidades de los usuarios y que no dependa de elementos externos.

Con base en el capítulo 3 se pueden resumir las siguientes propuestas:

PROPUESTA 1: Realidad Aumentada

Como ya se mencionó en el capítulo 3, la tecnología de Realidad Aumentada permite colocar objetos tridimensionales en la visión del usuario en el mundo real, es decir, se pueden observar estos elementos en una realidad virtual en el entorno real. El uso de este tipo de tecnologías se ha extendido en plataformas tanto Android e iOS por lo que la primera parte conceptual de la aplicación se estaría cumpliendo. Además, de no depender de otros elementos externos para presentar los elementos virtuales al usuario. Para lograr la medición de los fetos es necesario desarrollar una interfaz intuitiva y sencilla, dentro del propio programa y utilizando la pantalla del teléfono inteligente.

PROPUESTA 2: Realidad Virtual

La contraparte de la Realidad Aumentada, es la Realidad Virtual y también permite presentar elementos virtuales en el campo de visión del usuario, logrando así que la información sensorial que recibe se complemente con su entorno. El uso de esta tecnología permitirá presentar los modelos digitales de los fetos de una manera más atractiva para las personas. El desarrollo de esta aplicación también se ha extendido para ambas plataformas móviles (Android e IOS). El único detalle de este tipo de tecnología es la dependencia de visores para su adecuada visualización en los teléfonos inteligentes.



Figura 46. Visor Samsung Gear VR [65]

De utilizar esta tecnología es necesario desarrollar el hardware para realizar las respectivas mediciones en los fetos. Actualmente es posible con el uso de guantes hápticos para realidad virtual.



Figura 47 Guante háptico desarrollado en España [66]

PROPUESTA 3: Realidad Mixta

Como ya se mencionó esta tecnología surgió de la combinación entre Realidad Aumentada y Virtual, por lo que se obtienen todas las ventajas presentes en ambas. La principal ventaja es que ofrece la detección de la posición y orientación de las manos de los usuarios mediante sensores especializados. Esto podría ser usado para realizar las mediciones de los objetos virtuales de manera muy sencilla y amena. Pero de la misma manera como sucede con la Realidad Virtual depende del uso de un visor especial diseñado por Microsoft. Además de eliminar el uso de los dispositivos móviles, debido a que la proyección de los objetos se hace a través de los propios visores.

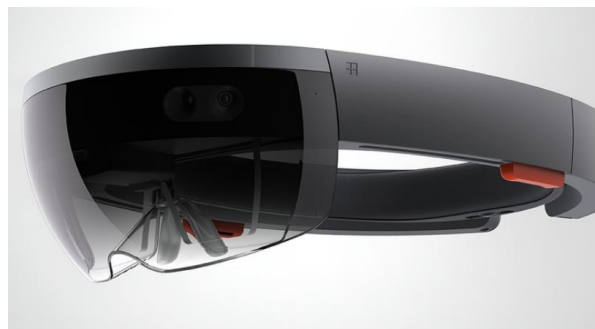


Figura 48 Visor HoloLens desarrollado por Microsoft[67]

5.1.6. Elección.

En la siguiente Tabla 8 se enlistan las propuestas y con las especificaciones pertinentes para seleccionar la tecnología.

Tabla 8 Tabla de evaluación de las propuestas.

Especificaciones	Prop 1: R.A.	Prop 2: R.V.	Prop 3: R.M.
Flexibilidad de funcionamiento en diferentes plataformas	Cumple	Cumple	No cumple
Interfaz de usuario sencilla.	Desarrollar	Desarrollar	Cumple
Capacidad del programa de mostrar medidas en [mm].	Desarrollar	Desarrollar	Desarrollar

En esta tabla, si una casilla tiene escrito la palabra “Cumple” quiere decir que la propuesta cumple con la especificación de manera satisfactoria. Si una casilla tiene escrito “No cumple” quiere decir que no se cumple con la especificación. Si en la casilla está escrito “Desarrollar” quiere decir que la propuesta no cumple completamente con la especificación, pero puede cumplirse mediante trabajo adicional en la propuesta.

Como se puede observar en la Tabla 8, si se toma la tercera propuesta, sólo sería necesario desarrollar la capacidad de mostrar las medidas con las manos del usuario, pero la primera parte del diseño conceptual se especifica que es necesario que se trabaje en plataformas de dispositivos móviles por lo que esta opción queda descartada.

Con base en esto, las opciones disponibles son las tecnologías de realidad virtual y realidad aumentada. Ambas cuentan con las características necesarias para satisfacer las necesidades de acuerdo con la Tabla 8. Sin embargo, como ya se mencionó al inicio de este apartado, para el desarrollo del proyecto es importante no depender de objetos externos como lo son los visores y guantes hápticos.

Dicho lo anterior y de analizar las distintas se elige la Realidad Aumentada siendo la primera propuesta, como la tecnología sobre la cual se desarrollará la solución.

Cabe mencionar que el desarrollo de esta aplicación será inicialmente en Android debido a que no se cuenta con una disponibilidad inmediata de las licencias de desarrollo en iOS para proseguir con la elaboración de esta plataforma.

5.2 Diseño de Detalle

5.2.1. Programas y complementos

Unity 3D

Unity es un reconocido motor de desarrollo de videojuegos. Entre sus características más destacadas está la facilidad de desarrollo e implementación de software, así como la gran variedad de plataformas para las cuales se puede desarrollar, desde consolas de videojuegos, PC y dispositivos móviles con sistemas operativos Android e IOS.

Adicionalmente, Unity es una plataforma de desarrollo muy accesible ya que es posible descargarla y utilizarla de forma gratuita. Cobrando sólo si se quiere descargar la versión profesional, la cual cuenta con más herramientas o si se quiere publicar un software de forma remunerada, en cuyo caso se le debe pagar una comisión a la empresa.

Unity 3D provee al usuario con un editor visual fácil de utilizar. Este editor permite importar archivos como modelos 3D, texturas, sonidos, entre otros. La importación permite que el usuario use sus propias creaciones o compre archivos en la tienda virtual de la compañía.

Unity incluye las herramientas de desarrollo **MonoDevelop** y **Visual Studio** con las que es posible crear scripts en JavaScript, C#, y en una estructura similar a la de Python llamada Boo. Estos códigos son los elementos que permiten la interacción entre los demás tipos de archivo, creando entornos interactivos. La flexibilidad y versatilidad ofrecida por la interfaz de usuario y los diversos lenguajes de programación, hacen de Unity 3D una gran herramienta de desarrollo en diversas aplicaciones en la industria.

Android Studio

Para desarrollar aplicaciones en Android es necesario utilizar la plataforma de desarrollo Android Studio, la cual complementa a Unity 3D con las herramientas necesarias para la edición, depuración, pruebas y perfilamiento de código. Cabe destacar que esta IDE permite crear apps para Android de forma nativa, es decir con el lenguaje de programación propio de estas plataformas: Java.

Unity 3D + Vuforia = Realidad Aumentada

Un complemento de Unity 3D es Vuforia, el cual es un kit de desarrollo que permite crear aplicaciones que utilizan realidad aumentada. Utiliza la tecnología de visión por computadora para reconocer y dar seguimiento a imágenes planas en tiempo real, reconociéndolas como “Imágenes objetivo” o marcadores.

Esta capacidad de reconocimiento de superficies objetivo, permite a los desarrolladores posicionar y orientar objetos virtuales en 3D, en relación con imágenes en el mundo real cuando estos son visualizados a través de la cámara de un dispositivo móvil, al reconocer cierto patrón específico que eligió el desarrollador.

Una vez identificada a través de la cámara del dispositivo, el software rastrea la posición y orientación del patrón. Normalmente se utilizan imágenes que sean fáciles de identificar con el entorno real como los códigos QR y marcadores, como se muestran en la Figura 49:



Figura 49. De izquierda a derecha: Código QR y Marcador. [68]

Es importante mencionar que existen varios niveles de Realidad Aumentada:

Nivel 0: Hiperenlaces en el mundo físico. Los activadores en este nivel son los códigos QR que sirven como un indicador para activar una función, ya sea desplegar información, gráficas o un sitio web.

Nivel 1: Basada en marcadores. En este nivel los activadores son marcadores, figuras que cuando son escaneadas, generan un modelo 3D que se superpone en la imagen real.

Nivel 2: Sin marcadores. Los activadores son imágenes, objetos, superficies planas o bien localizaciones GPS.

Nivel 3: Visión aumentada. Realidad Aumentada incorporada en gafas o visores. [68]

Al inicio de este capítulo se mencionó que la primera versión de la aplicación que se desarrolló estaba en el nivel 1, es decir, utilizando marcadores. En esta primera iteración era necesario imprimir el marcador en una hoja de papel y apuntar con la cámara del dispositivo móvil al marcador, la combinación de esta acción junto con la medición del modelo mediante el uso de la pantalla táctil resultaba poco eficiente en ejecución.

Es por esta razón que se realizó una mejora a la aplicación con el fin de que los usuarios no dependan del uso de marcadores. Se sustituyó el uso de marcadores por el nivel 2 de realidad aumentada, cuyo funcionamiento se desarrollará posteriormente.

5.2.2. Importación de modelos

Como ya se mencionó en la última parte del capítulo cuatro, los modelos generados son exportados en el formato OBJ, siendo uno de los dos formatos que acepta Unity 3D (el otro formato soportado es el FBX). Debido a que la orientación de los ejes principales (XYZ) que genera el programa Autodesk Meshmixer, presentan una configuración diferente a la que utiliza Unity 3D es necesario corregir dichos ejes, además de extraer la textura de los modelos en un formato de imágenes para visualizarlos correctamente. Para resolver estos detalles, se utiliza Autodesk 3D's Max, un programa de creación de gráficos y animación 3D, el cual cuenta con las herramientas y funciones necesarias para realizar dichas modificaciones.

Orientación de ejes principales

Para adecuar la orientación de los ejes X, Y, Z de cada uno de los modelos generados con base en la configuración de ejes que maneja Unity, basta con indicar en los parámetros de los ángulos, el cambio que se desea generar.

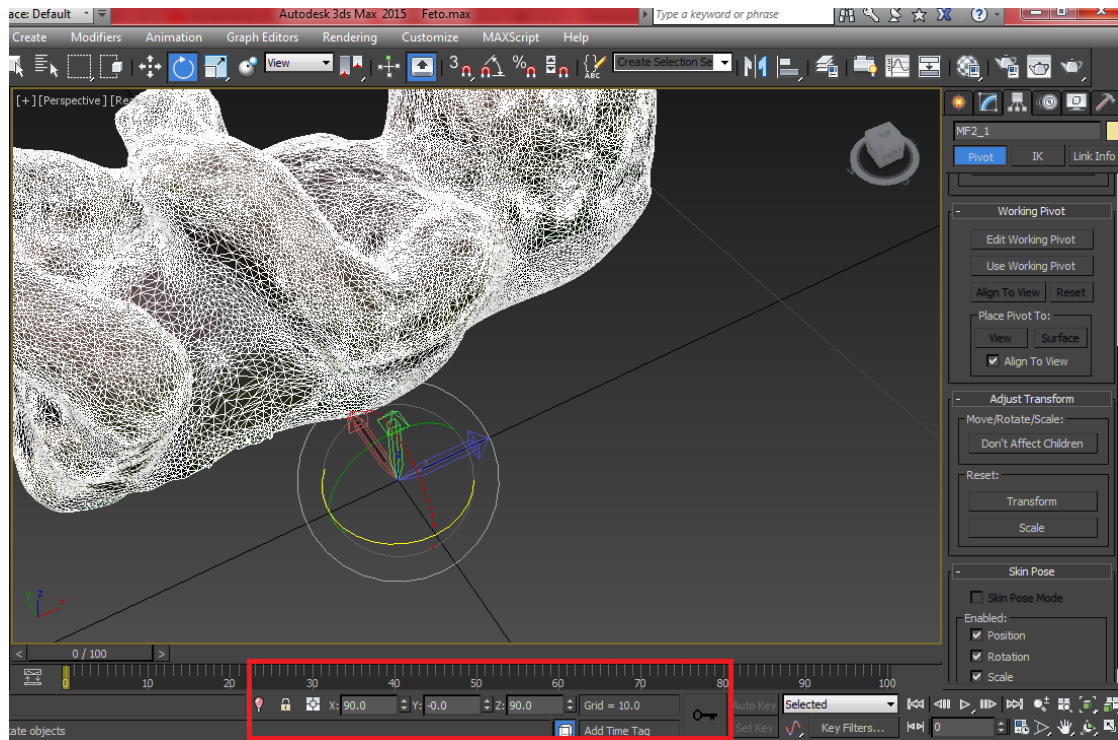


Figura 50. Programa de animación 3D. Autodesk 3D's Max.

En la figura 51 se observan los parámetros introducidos para igualar los ejes del modelo con los ejes de Unity:

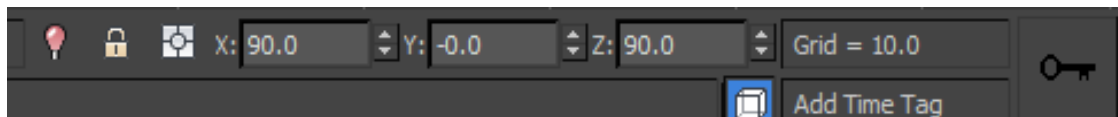


Figura 51. Parámetros de configuración para Unity.

Textura de modelos

Otra de las herramientas disponibles en este programa es la obtención de la textura o material de los modelos tridimensionales, como se puede observar en la siguiente figura, el programa realiza un mapeo de bits de la textura del modelo tridimensional, en una imagen bidimensional. Siendo esta imagen las características físicas del feto.

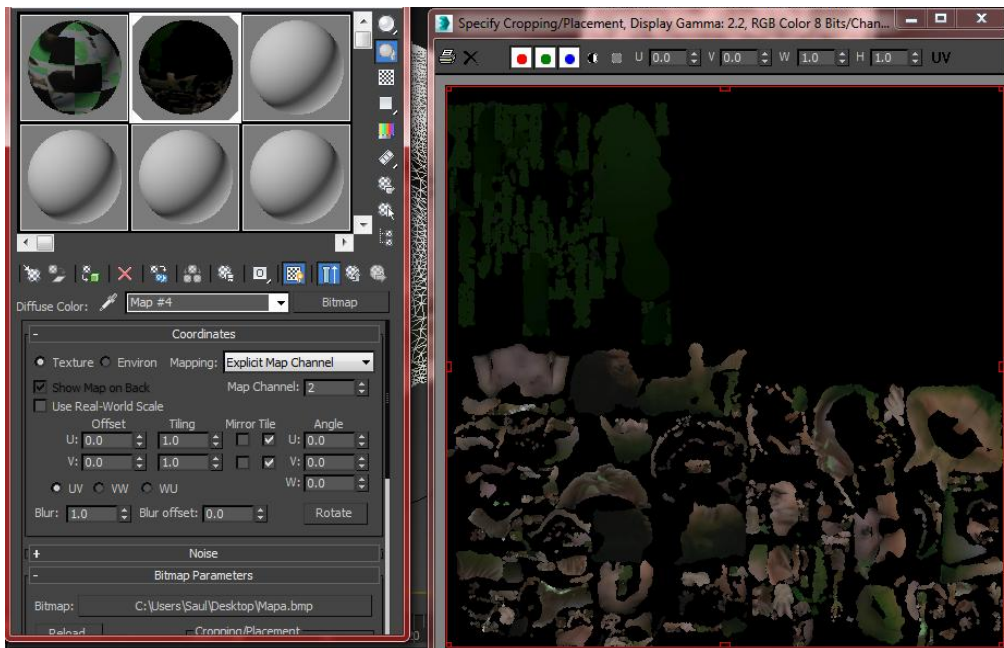


Figura 52. Textura del modelo del feto.

Una vez configurado el modelo, se procede a exportarlo en el formato OBJ desde 3D's Max con los nuevos ejes principales y la imagen de textura que se utilizará posteriormente en Unity, como se muestra en la Figura 53.

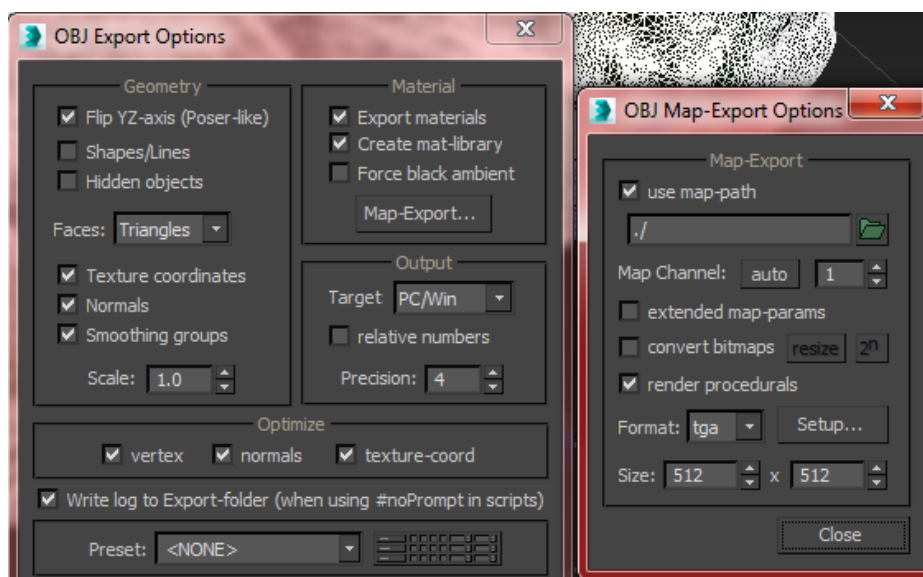


Figura 53. Configuración de exportación en 3D's Max.

Configuración de modelos desde Unity 3D

Como se puede observar en la Figura 54, los modelos importados presentan una textura uniforme, por lo cual es necesario importar la imagen de textura generada en 3D's Max. Cabe destacar que, se observa cómo los ejes del modelo coinciden con los del entorno de Unity. Por lo cual la corrección de ejes fue exitosa.

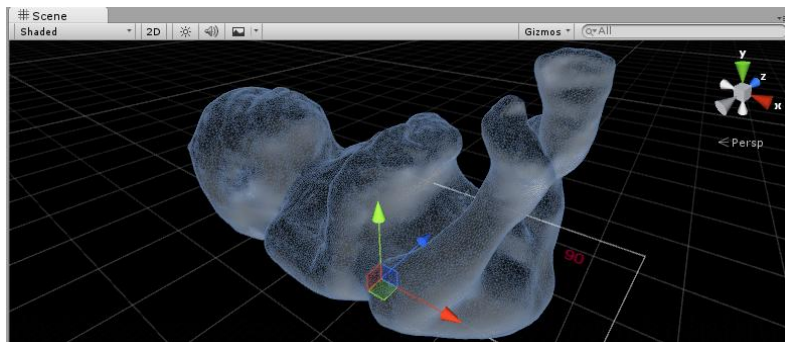


Figura 54. Importación del modelo en Unity 3D.

Con la ayuda de los paneles de Unity, se puede utilizar la información generada en la imagen de bits para implementar la textura de cada modelo.

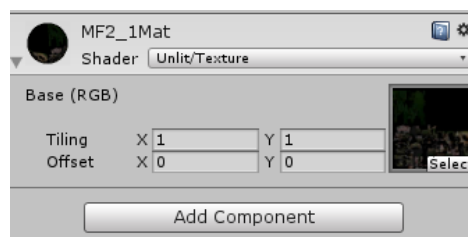


Figura 55. Configuración de textura.

En la Figura 56 se observa el resultado final de esta primera etapa del desarrollo de la aplicación:

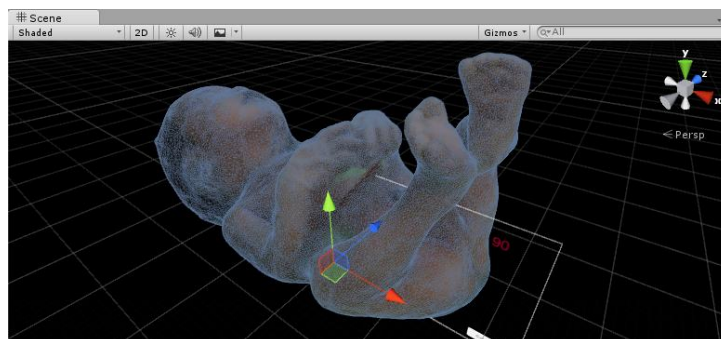


Figura 56. Textura y ejes acoplados al entorno de Unity 3D.

5.2.3. Interfaz de usuario

De acuerdo a las consideraciones realizadas en el apartado del diseño conceptual, se propone el siguiente diagrama de estados para la transición de ventanas. El cual muestra el orden en el cual el usuario navegará dentro de la aplicación como se muestra en la Figura 57.

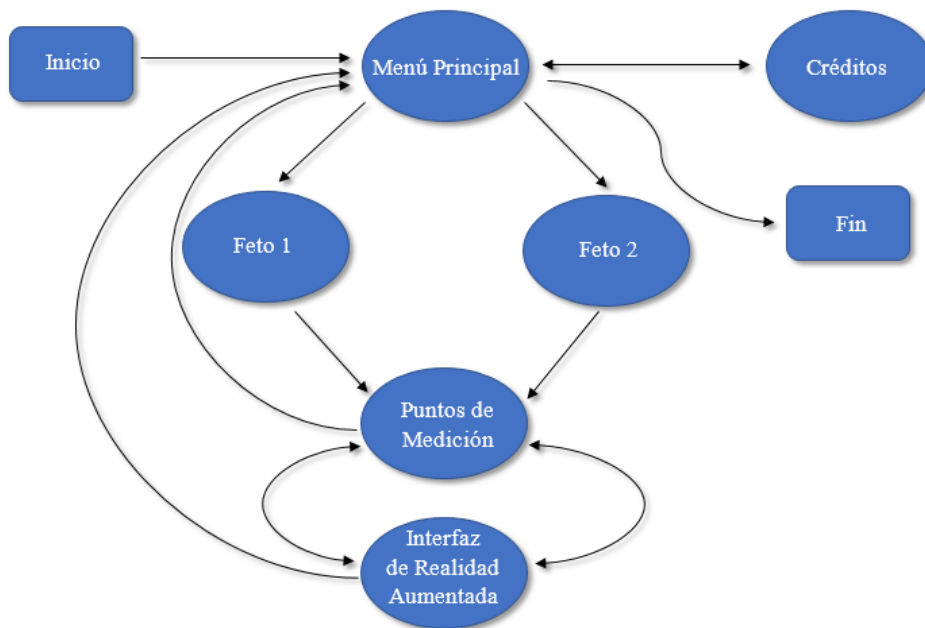


Figura 57. Diagrama de transición de estados de ventanas.

A continuación, se presentarán las ventanas para la interfaz de usuario, hechas con base en el diagrama de transición propuesto. Como se podrá observar se respetaron las especificaciones del diseño conceptual, tanto por número de ventanas, botones y demás elementos interactivos, siendo una interfaz sencilla.



Figura 58. Inicio: Presentación de la aplicación.

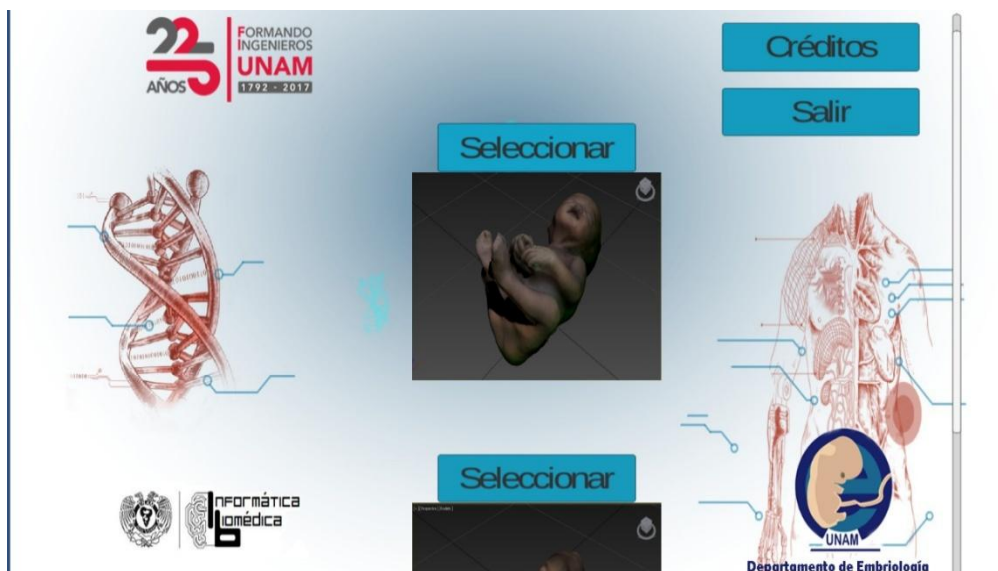


Figura 59. Menú principal del programa.

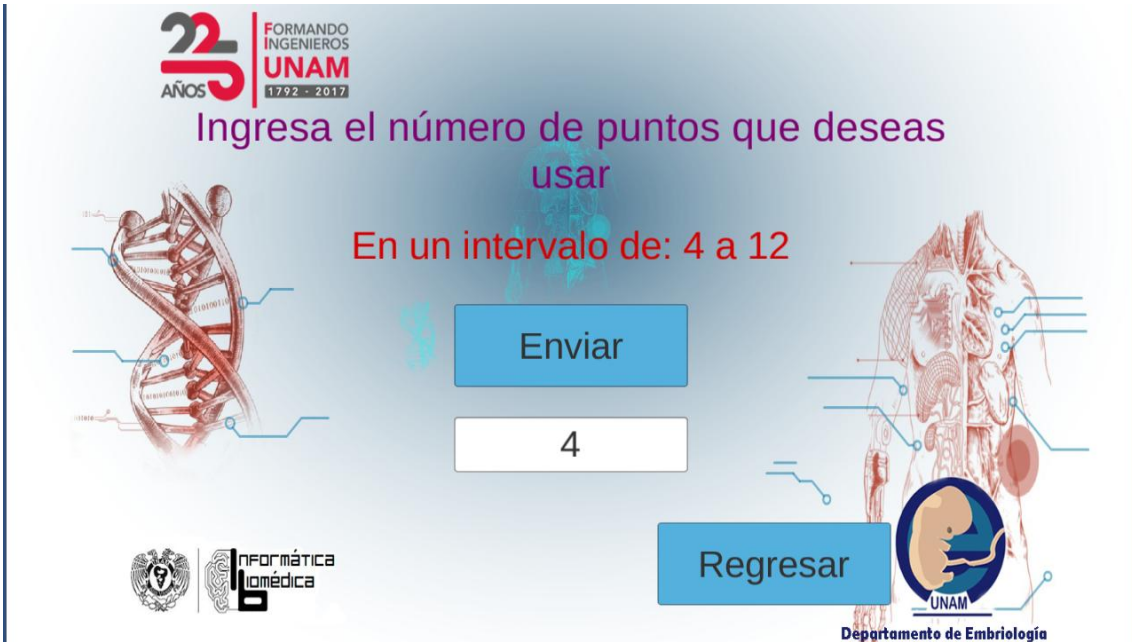


Figura 60. Ventana para definir el número de puntos de medición.



Figura 61. Interfaz de Realidad Aumentada.



Figura 62. Ventana de créditos de colaboración en el proyecto.

5.2.4. Rotación de modelos

Para permitir la manipulación por parte del usuario de los fetos en el entorno de Realidad Aumentada es necesario permitir ciertos grados de libertad en la manipulación de los modelos. Existen dos formas para resolver este problema: Ángulos de Euler y Cuaterniones.

Ángulos de Euler

Cualquier rotación de un cuerpo rígido puede describirse por medio del sistema de coordenadas fijo: X, Y, Z. Estas rotaciones pueden escribirse en términos de una matriz de rotación (A), siendo de manera general:

$$A = B * C * D$$

Las matrices B, C y D describen las rotaciones mostradas en la Figura 63. Los ángulos de rotación mostrados en la figura se representan con las tres matrices de rotación llamadas: Ángulos de Euler.

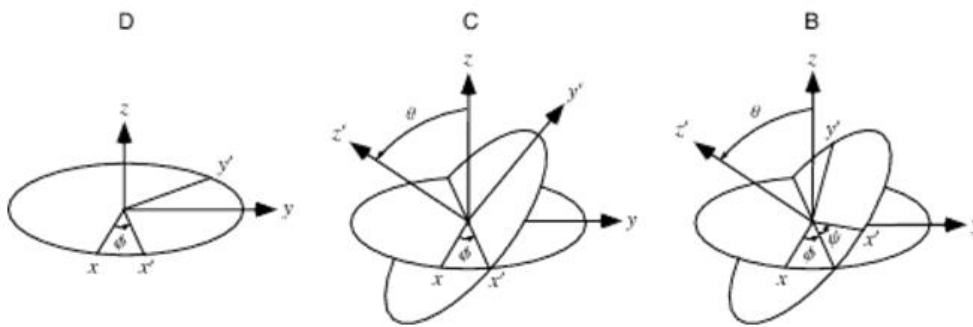


Figura 63. Rotaciones particulares de los Ángulos de Euler. [69]

Existen varias convenciones para estos ángulos, según los ejes sobre los que se realizan las rotaciones, siendo la convención más común y utilizada: “x-convención”, y se describe de la siguiente manera:

- 1) La primera rotación es con el ángulo ϕ , alrededor del eje Z usando la matriz D .
- 2) La segunda rotación es el ángulo θ que pertenece a un intervalo $[0, \pi]$ alrededor del nuevo eje X formado (ahora x'), usando la matriz C .
- 3) La tercera rotación es con el ángulo ψ , alrededor del nuevo eje Z formado (ahora z'), usando la matriz B .

Por lo tanto, las matrices descritas son:

$$\begin{aligned}
 D &\equiv \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 C &\equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \\
 B &\equiv \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix},
 \end{aligned}$$

Figura 64. Matrices B, C y D.[69]

Se debe recordar que para representar una secuencia finita de rotaciones se multiplican matrices de rotación entre sí, cuidando la secuencia de estas a la hora de realizar las rotaciones, debido a que la operación no es conmutativa. Cada una de estas matrices expresa una transformación rotacional respecto del sistema de coordenadas fijo (X, Y, Z) o de los ejes de sistema de

coordenadas en movimiento (del propio objeto en rotación). Debido a esto, el orden depende de:

- o Si el sistema de coordenadas en movimiento, se gira respecto a uno de los ejes del sistema fijo, entonces se debe pre-multiplicar la matriz de rotación. Por esta razón, la definición de la matriz **A** que se presentó al inicio, siguiendo el orden de la “x-convención” se representa como:

$$A = B * C * D$$

- o Si el sistema de coordenadas en movimiento se gira respecto de alguno de sus ejes, entonces se debe post-multiplicar la matriz de rotación. [70]

Cuaterniones

Se entiende a los cuaterniones como una extensión de los números reales, análoga a los números complejos. Mientras que los números complejos son una extensión de los reales con la adición de la unidad imaginaria *i*, tal que $i^2 = -1$, los cuaterniones son una extensión con la adición de las unidades imaginarias: *i*, *j*, *k* a los números reales de tal manera que:

$$i^2 = j^2 = k^2 = i * j * k = -1$$

Además, con la ayuda de la Tabla de Cayley, se observan otras propiedades al multiplicar los componentes de un cuaternión.

Tabla 9 Tabla de Cayley [71]

	<i>1</i>	<i>i</i>	<i>j</i>	<i>k</i>
<i>1</i>	1	<i>i</i>	<i>j</i>	<i>k</i>
<i>i</i>	<i>i</i>	-1	<i>k</i>	- <i>j</i>
<i>j</i>	<i>j</i>	- <i>k</i>	-1	<i>i</i>
<i>k</i>	<i>k</i>	<i>j</i>	- <i>i</i>	-1

A partir de esta tabla es posible obtener otras propiedades importantes:

$$i * j = k$$

$$j * k = i$$

$$k * i = j$$

De esta manera se definen a los cuaterniones como números hipercomplejos de la forma: $a + bi + cj + dk$, donde a, b, c, d son números reales adicional a las tres unidades imaginarias i, j, k .

Fueron introducidos por W. R. Hamilton en 1843 y gracias a esta convención se adoptó en el formalismo vectorial en espacio R^3 con los componentes i, j, k .

Esto lleva a la pregunta: ¿Cómo se relacionan con la representación de la rotación de los cuerpos? Para responder a esto es necesario retomar la literatura de los cursos de cálculo y de variable compleja: La función exponencial.

La función exponencial tiene la propiedad de transformar sumas en productos:

$$e^{x+y} = e^x * e^y$$

Siendo esta representación en los números reales. Sin embargo, también es posible considerar la función exponencial en un número complejo. Bajo este contexto general, Leonhard Euler probó la relación entre el seno, el coseno y la exponencial obteniendo la expresión conocida como fórmula de Euler:

$$e^{i\theta} = \cos(\theta) + i\text{sen}(\theta)$$

Ayudando a la expresión de un número complejo de la siguiente manera:

$$z = r\cos(\theta) + ir\text{sen}(\theta) = re^{i\theta}$$

De esta expresión se obtiene su conjugado:

$$z' = r'e^{i\theta'}$$

A partir de lo anterior, se encuentra que:

$$z * z' = re^{i\theta} * r'e^{i\theta'} = r * r' * e^{i(\theta+\theta')}$$

Por lo tanto, de esta última expresión se concluye que al multiplicar el número complejo z' por otro z con coordenadas polares (r, θ) , se rota a z' por un ángulo θ . Entonces, la fórmula de Euler representa una rotación de θ radianes alrededor del origen [72]:

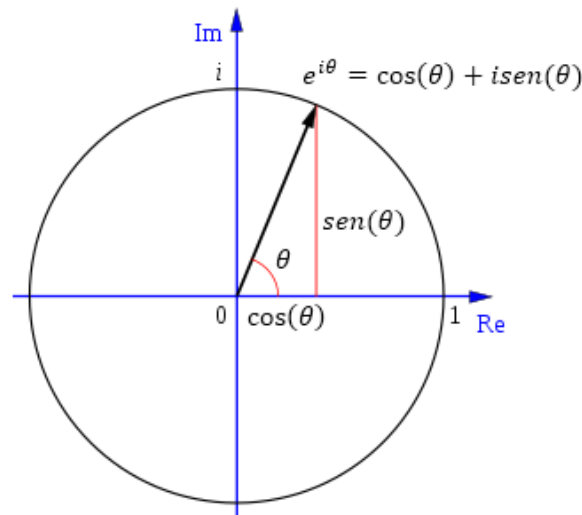


Figura 65. Rotación en el origen del plano complejo. [72]

Y esto puede ser extendido para el álgebra de los cuaterniones, siendo estos números hipercomplejos. Para observar este comportamiento en los cuaterniones, es necesario usar el caso más sencillo donde solamente se cuenta con la parte real con una dimensión en la parte compleja, es decir:

$$Q = q_0 + iq_1 + j0 + k0$$

Este cuaternión se comporta exactamente como el que se observó al inicio de la sección: *una rotación en el plano*:

$$Q = \cos(\theta) + i\text{sen}(\theta) + j0 + k0$$

Redefiniéndolo en dos dimensiones, se obtiene que:

$$\text{Plano } YZ: Q_x = \cos(\theta) + i\text{sen}(\theta) + j0 + k0$$

$$\text{Plano } XZ: Q_y = \cos(\theta) + i0 + j\text{sen}(\theta) + k0$$

$$\text{Plano } XY: Q_z = \cos(\theta) + i0 + j0 + k\text{sen}(\theta)$$

De la misma manera, el álgebra de los cuaterniones permite la rotación entre vectores de la misma forma como sucede con los números complejos: basta con multiplicarlos. Con esta analogía para rotar un punto mediante un cuaternión la operación necesaria es pre-multiplicar el punto (expresado también en cuaternión) por el cuaternión de la rotación, y post-multiplicarlo por el conjugado del cuaternión de la rotación [73].

$$v' = Q * v * Q^*$$

¿Ángulos de Euler o Cuaterniones?

Usualmente en las aplicaciones de rotaciones en 3D se usan estas dos representaciones, cada una con sus ventajas y desventajas las cuales se presentan a continuación:

Ángulo de Euler

- Esta representación es más intuitiva, al representarse en el formato de los tres ejes principales, realizando rotaciones acordes a los ejes que uno desee.
- Estas rotaciones pueden representar una orientación con un giro de más de 180 grados.
- Limitación: Estos ángulos sufren el efecto “Gimbal Lock”, que consiste en la pérdida de un grado de libertad. Cuando dos ejes principales rotan y se colocan en paralelo, da como resultado que el tercer eje esté en la misma dirección que los ejes anteriores, por lo tanto, la última rotación no se puede aplicar, siendo este un punto de singularidad.

Cuaterniones

- No sufre el efecto de “Gimbal Lock”
- Unity 3D internamente maneja los cuaterniones por su rapidez computacional.
- Limitación:
 - 1) Su representación no es intuitiva, ni fácil de entender en comparación con los Ángulos de Euler.
 - 2) No es posible representar rotaciones que excedan a más de 180 grados en cualquier dirección.

A pesar de las dos limitaciones que tienen los cuaterniones, el entorno de Unity facilita su uso en la programación, por lo cual se usarán los cuaterniones para evitar el “Gibmal Lock”, además permite expresar los ángulos en Euler para después hacer la transformación en cuaterniones [74].

Para que el usuario pueda manejar los fetos se colocaron en la interfaz de realidad aumentada, 8 botones que le permitan 4 grados de libertad mediante rotaciones. De esta manera, se logra la rotación de los modelos tridimensionales en el espacio y entorno real de la persona.

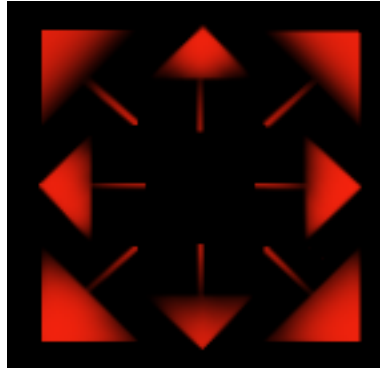


Figura 66. Botones para aplicar la transformación de rotación.

Tomando en cuenta que el modelo se puede considerar un conjunto finito de vectores, para realizar la rotación en el espacio absoluto del entorno virtual es necesario realizar una multiplicación entre cuaterniones.

```
void RotationUpButton()  
{  
    transform.rotation = Quaternion.Euler(rotationSpeed, 0, 0) * transform.rotation;  
}
```

Figura 67. Rotación utilizando los cuaterniones en Unity.

Como se puede observar en la Figura 67, se hizo uso del método **Quaternion** que permite utilizar los cuaterniones con Ángulos de Euler, por su facilidad de implementación. Esta rotación se hace en el eje X local, es decir en el eje propio del modelo, si se logra que rote en su propio eje, su visualización con el entorno real de la persona no es la deseada. Por lo tanto, es necesario referirlo al sistema absoluto del entorno de desarrollo, para ello es importante recordar las reglas de la multiplicación de matrices de rotación que se mencionó en este subtema con los Ángulos de Euler:

- Si el sistema de coordenadas en movimiento, se gira respecto a uno de los ejes del sistema fijo, entonces se debe pre-multiplicar la matriz de rotación.

Es decir, `Quaternion.Euler ()` funge como la matriz de rotación pre-multiplicada.

5.2.5. Marcadores de medición

Otra de las especificaciones en el proyecto concierne a tener la capacidad en el programa de mostrar las medidas en milímetros en cualquier parte del feto virtual, para ello se propone una serie de puntos en el espacio tridimensional

con el propósito de que el usuario los utilice y deslice los puntos al lugar en donde desee tomar las medidas. Estos puntos estarán limitados en un intervalo de 4 a 12 puntos para evitar colapsar el programa y esta información se enviará desde la ventana que se mostró en la Figura 60, del subtítulo: 5.2.3 Interfaz de usuario.

Recordando en el capítulo 4 de este documento en el apartado: 4.1.1 Escalamiento, se mostró la Tabla 2 donde podemos observar los diferentes factores de escala que utiliza Unity al importar cada formato de extensión de un modelo tridimensional. Basándonos en esta información, todos los elementos en Unity se escalaron de tal manera que cada unidad del entorno de Unity, sea equivalente a una unidad de milímetro real.

Los marcadores previamente mencionados que se utilizarán para las funciones de medición son objetos esféricos en el espacio como se muestra en la Figura 68.

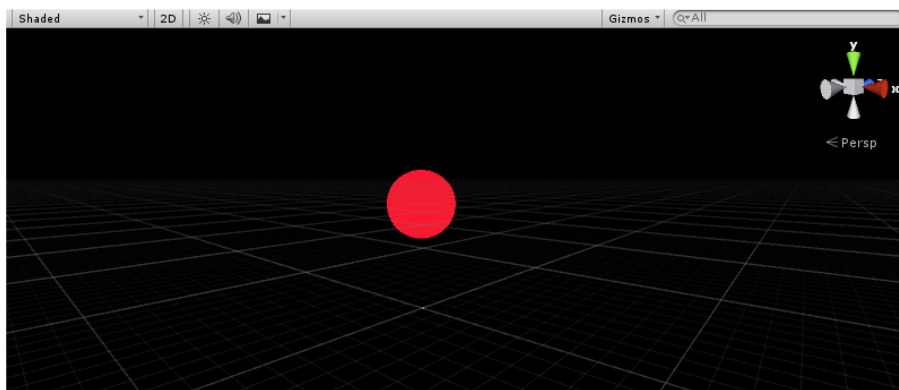


Figura 68. Marcador esférico.

Pero como la cantidad de ellos depende de la información que el usuario introduzca, es necesario recurrir a la literatura de programación orientada a objetos (POO por sus siglas en inglés). En la POO, es importante distinguir dos conceptos:

- Clase: Es un prototipo o molde que indica qué características van a tener y cómo se van a comportar los elementos creados a partir de dicha clase.
- Objeto: Son elementos creados a partir de las clases antes mencionadas. En algunos contextos o lenguajes se suele usar también el término instancia.

Con estas dos definiciones se puede concluir que a partir de una clase se pueden crear infinitos objetos o instancias [75]. El lenguaje de programación

utilizado en Unity es C#, gracias al cual fue posible crear “instancias” de los marcadores esféricos de la siguiente manera:

```
go = Instantiate (ball, new Vector3(-50,j*50,0),Quaternion.identity) as GameObject;
```

Figura 69. Estructura de las instancias en C#.

Esta línea de código permite “instanciar” o crear nuevos objetos con una nueva posición diferente en el espacio de tal manera que estén separados a cierta distancia y referenciados al origen de los ejes principales de Unity con `Quaternion.identity`.

La estructura anterior, está dentro de un ciclo en el programa. Este bucle está condicionado a terminar hasta que se hayan creado todos los marcadores que definió el usuario.

Una vez creados los objetos esféricos, se procede con la obtención de la distancia de cada uno de estos puntos. El resultado final de esa suma otorga la distancia total de esta serie de puntos. Para ello sólo es necesario obtener la posición tridimensional de un punto y el punto consecutivo:

```
dist = Vector3.Distance (values [numball].transform.position, values [numball + 1].transform.position);
```

Figura 70. Distancia entre dos puntos.

En el fondo, lo que hace Unity es usar la siguiente expresión de geometría espacial:

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2}$$

Además, se dibuja una línea entre cada punto para distinguir la trayectoria y dirección que sigue cada marcador, esto a través de un segmento dirigido:

```
lineRenderer.SetPosition (nummeasure, values [numball].transform.position);
```

Figura 71. Segmento dirigido en código.

Siendo la expresión matemática detrás de la programación:

$$\overrightarrow{AB} = (b_1 - a_1, b_2 - a_2, b_3 - a_3)$$

Finalmente se suma la distancia entre cada punto en un ciclo finito para obtener la distancia total, esta información es presentada al usuario como se observa en la Figura 72



Figura 72. Nueve marcadores siendo usados para medir el diámetro de la cabeza del feto.

5.2.6. Pantalla táctil

Para que el usuario pueda mover libremente estos marcadores en el espacio es necesario que tenga la libertad de moverlos con la pantalla de su dispositivo móvil. Por lo cual es importante obtener la información de las coordenadas donde la persona coloque su dedo.

En la fabricación de los celulares existen dos tipos de pantallas táctiles: Capacitivas y Resistivas, diferenciándose de la siguiente forma:

Capacitivas: Funcionan con una pantalla que detecta la corriente producida por el toque de una parte conductora del cuerpo, cuando esto sucede, la corriente es atraída y el dispositivo reconoce la posición del toque.

Ventajas:

- Bajo tiempo de respuesta.
- Resistentes al rayado.
- Fáciles de limpiar.
- Usualmente son multi-táctiles.

Desventajas:

- Sólo funcionan con partes conductoras del cuerpo.
- Se ensucian con facilidad.
- Poco resistentes al polvo, agua y huellas.

Resistivas: Funcionan con varias capas de resistencia, por lo que la presión en un punto hace que las capas entren en contacto y se cree la corriente para reconocer la posición.

Ventajas:

- Más económicas.
- Más precisas.
- Más resistentes al agua, el polvo y huellas.
- Funcionan con objetos como lápices o la uña.

Desventajas:

- Requieren presión para funcionar.
- No muy resistentes a rayados.
- Poca visibilidad al tener varias capas.
- Pierden su calibración con el uso.

Dependiendo de los usos de cada pantalla, no se puede decir que una sea mejor que la otra, ya que tienen características únicas. Lo importante de esto es que ambas detectan las pulsaciones a los cambios en la corriente eléctrica y así obtienen las coordenadas de la pantalla [76].

Otra de las herramientas que ofrece Unity es obtener estas pulsaciones con la entrada de información a la pantalla táctil. Puede rastrear hasta cinco dedos tocando la pantalla al mismo tiempo con sólo acceder a una matriz que guarda estos cambios. Para el proyecto, solamente es necesario simular el movimiento de un “mouse o ratón” por lo que sólo se debe detectar una pulsación a la vez con un solo toque de dedo [77]. Esta simulación de mouse en los dispositivos móviles proporciona movimiento por lo que es muy útil para desplazar los puntos de los marcadores de medición.

```
Vector3 mousePosFar = new Vector3 (Input.mousePosition.x, Input.mousePosition.y,  
Camera.main.farClipPlane);
```

Figura 73. Simulación del movimiento del mouse en la pantalla táctil.

La figura 73 muestra cómo se guardan en un vector las coordenadas XY de la pantalla. Esta información es utilizada posteriormente para mover los puntos.

Para poder desplazar estos marcadores es necesario detectarlos en el espacio con las coordenadas obtenidas de la pantalla táctil. Para esto se hizo

uso de un evento muy utilizado en los videojuegos: **Raycasting**, que es el proceso de disparar un “rayo” invisible desde un punto, en una dirección específica para detectar si hay colisiones con algún objeto que esté en el camino del rayo [78].

Para lograrlo, se realizó la conversión de las coordenadas de la pantalla táctil a las coordenadas del espacio virtual:

```
Vector3 mousePosN = Camera.main.ScreenToWorldPoint (mousePosNear);
```

Figura 74. Coordenadas del espacio con base en las coordenadas de la pantalla táctil.

Luego se creó el evento del Raycasting:

```
Ray mr = new Ray (mousePosN);
```

Figura 75. Constructor del Raycasting.

Una vez detectado el objeto, que en este caso sería un punto de medición creado en el espacio virtual y con la condición de que el usuario esté desplazando su dedo en la pantalla táctil, basta con actualizar la nueva posición de este marcador.

```
g_Obj.transform.position = mRay.GetPoint (rayDistance);
```

Figura 76. Actualización de la nueva posición en el espacio.

Con esto se consigue que el usuario mueva libremente cualquier punto que desee en el espacio virtual para colocarlo en un punto específico en el cuerpo del feto tridimensional.

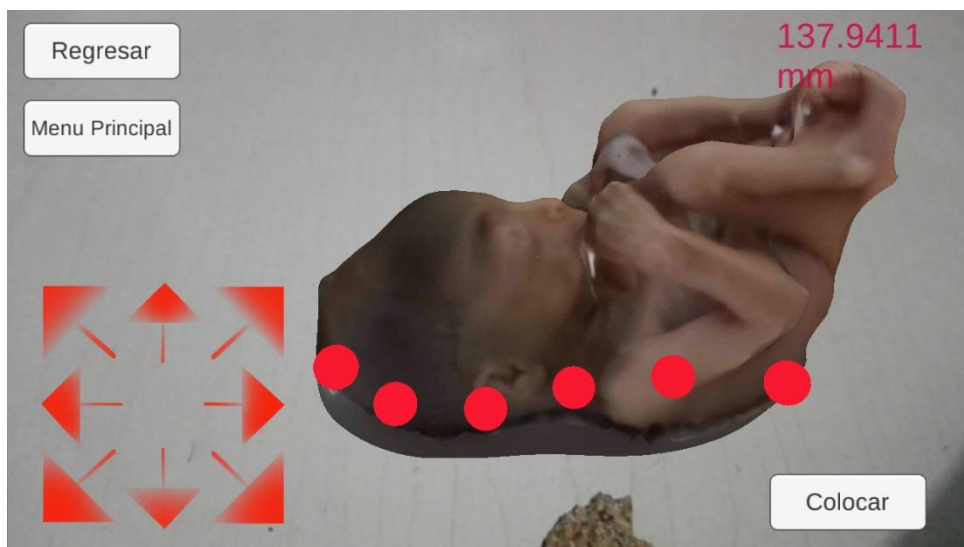


Figura 77. Alineación de los marcadores en la columna del feto.

5.2.7. Localización simultánea y mapeo del entorno real

Este último apartado está dedicado a la implementación de la Realidad Aumentada como tal, en la aplicación móvil. Como se mencionó al inicio de este capítulo y como se abarcó en el Diseño Conceptual, se realizaron dos versiones en este proyecto:

- ❖ Versión 1: Esta primera versión se presentó en el concurso de diseño e innovación que se realiza cada semestre en el Centro de Ingeniería Avanzada (CIA) de la Facultad de Ingeniería, llamado ExpoDIMEI, obteniendo el primer lugar en la categoría de sistemas biomédicos. Además, esta versión fue tema de parte de una conferencia para presentar el proyecto ante estudiantes y profesores de la Facultad de Medicina en el Palacio de Escuela de Medicina del Centro Histórico de la CDMX.

Esta primera versión utilizaba el nivel 1 de realidad aumentada basada en marcadores. Resolvía todas las especificaciones del proyecto, aunque presentaba ciertas dificultades a la hora de detectar la impresión del marcador debido a que los algoritmos de visión artificial están sujetos a ruidos en el ambiente, por ejemplo: luminosidad. Por lo que la cámara del teléfono tardaba en detectar el marcador. Debido a esto se decidió, después de estas dos presentaciones ante el público, seguir trabajando en la segunda versión.

- ❖ Versión 2: Siendo la última versión que se presentó ante Informática Biomédica con el propósito de desarrollar el capítulo 6, de pruebas e implementación. Esta versión se llevó al nivel 2 de realidad aumentada sin el uso de marcadores. Este tipo de tecnología como se mencionó en el capítulo 3 es conocida por sus siglas en inglés como SLAM. A continuación, se describirá su funcionamiento

La localización simultánea y mapeo del entorno real (SLAM) históricamente fue relacionada con la navegación robótica para adquirir un mapa de su entorno y la ubicación en relación con el mapa generado, adquiriendo de manera síncrona las mediciones para retroalimentar a su ley de control.

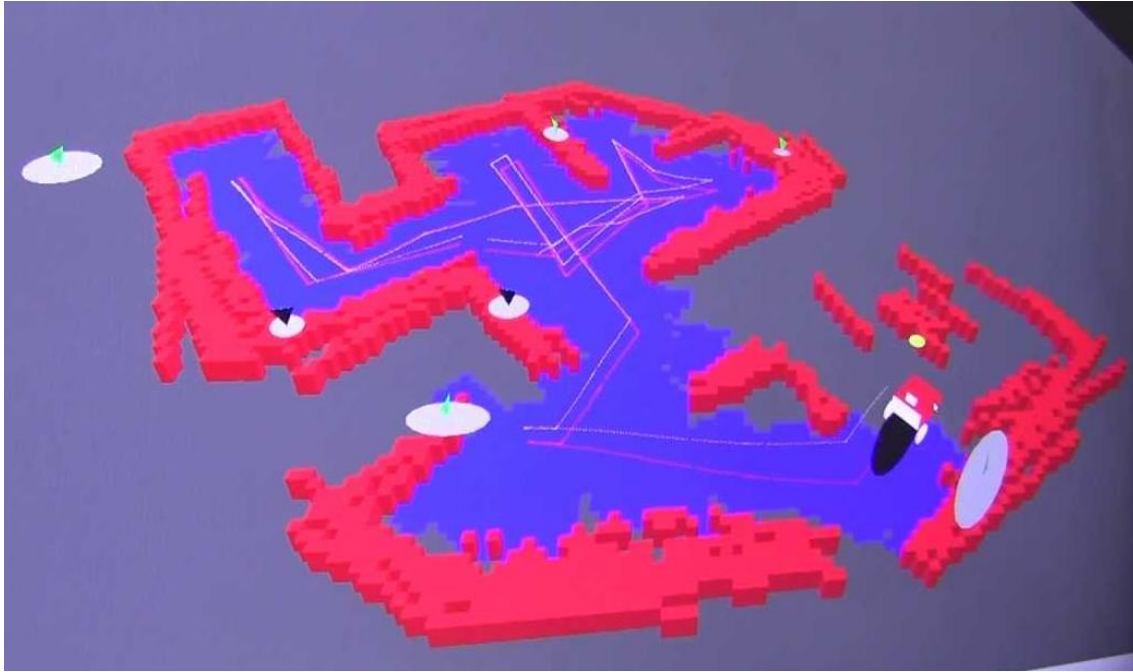


Figura 78. Aplicación del SLAM para la ubicación de un robot. [79]

Las mediciones más típicas eran en combinación de un número determinado de sensores como lo es el sistema de localización global (GPS), localizador láser, radares, sensores inerciales. La localización es un sub-problema de la determinación de la posición relativa en el mapa, por lo tanto, está incluida en el problema del SLAM. Esta localización y rastreo del movimiento se basan en el Filtro de Kalman también conocido como: EKF-Localisation. Este tipo de filtros es usado en entornos de aire libre como lo son los GPS.

Para hacer uso de las técnicas que se presentan dentro del SLAM y dar seguimiento visual-inercial, se han desarrollado tres estrategias que representan modelos en el espacio de estados (de la teoría de control), siendo estos modelos: giroscopio, gravedad, aceleración.

Modelo del giroscopio:

El vector de estados X está compuesto:

$$x^T = [s^T \ \dot{s}^T \ q_\theta^T \ \omega^T \ b_\theta^T]$$

Donde:

s = posición

\dot{s} = velocidad

q_θ = orientación en cuaternion

ω = velocidad angular

$$b_{\theta} = \text{medida del giroscopio (rotaciones)}$$

Para realizar simplificaciones en la expresión, la posición y orientación de la IMU con respecto al marco absoluto se estiman en el vector de estado, esto significa, que utilizando un observador o estimador de estado, se predicen estos dos estados con base en las mediciones de entrada y salida del sistema.

La estimación de la orientación se compara con el valor real, es decir, de la medida del giroscopio para compensar el error en futuras mediciones. Para el caso de la velocidad, se obtiene a partir del cambio de posición entre una diferencia del tiempo. Las rotaciones se representan en cuaterniones debido a que los ángulos en esta representación quedan libres de singularidades. [79]

Nota: Una unidad de medición inercial o IMU denota a un dispositivo electrónico cuyo objetivo es obtener mediciones de velocidad angular, rotación y fuerzas gravitacionales, usando en combinación de acelerómetros, giroscopios y a veces magnetómetros [80].

Modelo de gravedad:

Con base en el modelo del giroscopio, el vector de estados X está compuesto:

$$x^T = [s^T \ \dot{s}^T \ q_{\theta}^T \ \omega^T \ b_{\alpha}^T \ b_{\theta}^T]$$

Donde:

$$b_{\alpha} = \text{medida del acelerómetro}$$

En este modelo de gravedad, se considera la medición del acelerómetro de la IMU para modelarlo como un inclinómetro, que es simplemente una estimación de la rotación asumiendo que la aceleración del cuerpo es despreciable.

Modelo de aceleración:

Con base en el modelo de gravedad, el vector de estados X está compuesto:

$$x^T = [s^T \ \dot{s}^T \ \ddot{s}^T \ q_{\theta}^T \ \omega^T \ b_{\alpha}^T \ b_{\theta}^T]$$

Donde:

$$\dot{s}^T = \text{aceleración}$$

En este modelo considera la estimación de la aceleración del cuerpo en el vector de estados [79].

De estos modelos presentados el desarrollo de la aplicación está inspirado en el modelo del giroscopio, considerando la posición y orientación en el modelo de estados, siendo que la mayoría de los teléfonos inteligentes tienen integrada una IMU. A través de Unity es posible adquirir estas mediciones de la siguiente manera:

```
position = new Vector3();  
orientation = new Quaternion();
```

Figura 79. Declaración de variable.

Como se puede observar en la Figura 79, primero se declaran el vector de posición y el cuaternión de orientación. Después se obtiene la posición y orientación de la IMU del teléfono:

```
position.x = floorPosition.Get<float>("x");  
position.y = floorPosition.Get<float>("y");  
position.z = floorPosition.Get<float>("z");  
  
orientation.x = floorOrientation.Get<float>("x");  
orientation.y = floorOrientation.Get<float>("y");  
orientation.z = floorOrientation.Get<float>("z");
```

Figura 80. Información del IMU del teléfono.

Posteriormente se orienta el entorno del espacio virtual creado por Unity para superponerlo con el espacio real:

```
_kudanTracker.ArbiTrackStart (floorPosition, floorOrientation);
```

Figura 81. Seguimiento sin el uso de marcadores.

Y finalmente, en la ejecución de la aplicación con realidad aumentada se puede visualizar en las siguientes figuras:

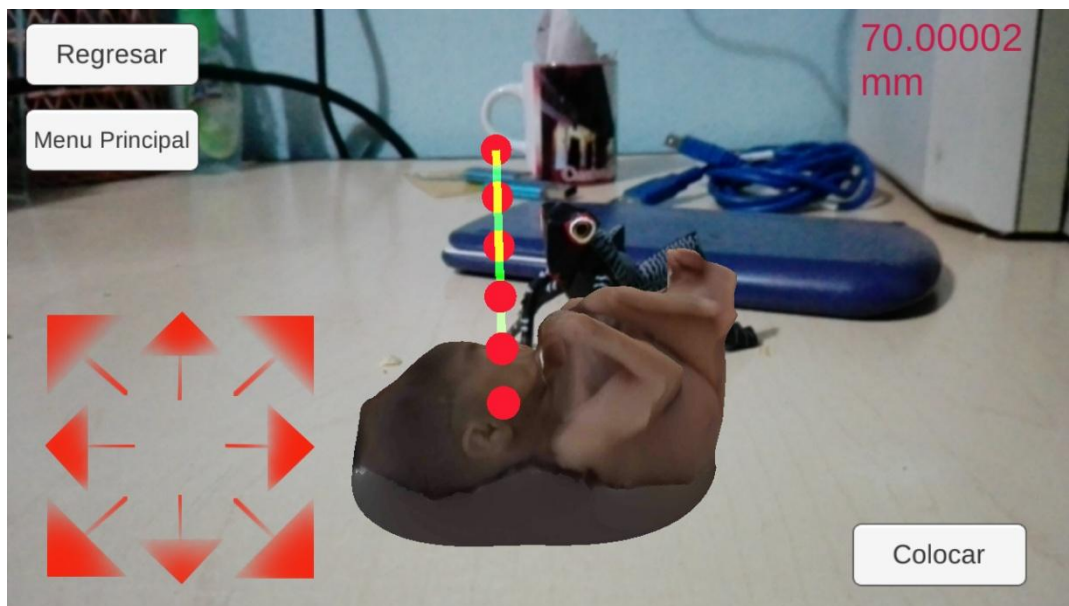


Figura 82. Ubicación fija del modelo en el entorno.



Figura 83. Giro de la cámara sin afectar la ubicación fija del modelo.

De esta manera se consigue la ilusión visual de profundidad en los modelos tridimensionales, es decir, estos modelos están en el espacio real a través de la cámara del teléfono. Una vez colocado el modelo en una superficie, éste se mantendrá superpuesto en una posición fija debido a que ya se posicionó en el espacio virtual que coincide con el espacio real. Esto gracias a la inspiración conceptual de lo que consiste el SLAM.

6. Pruebas de implementación

6.1 Protocolo de pruebas

Con el software generado y accesible en un dispositivo electrónico, el siguiente paso es comprobar que la aplicación sea funcional y cumpla con los objetivos planteados.

El objetivo general del trabajo, mencionado en el capítulo 1 es la creación del compendio de modelos virtuales de fetos humanos, los cuales pueden ser consultados por medio de la aplicación para dispositivos móviles. Los objetivos secundarios planteados se enfocaron en subsistemas del proyecto.

El cumplimiento de estos objetivos secundarios permitirá valorar si se cumplió el objetivo principal del proyecto. Los objetivos secundarios se listan a continuación:

- Evaluar, seleccionar y utilizar métodos de generación de modelos en tercera dimensión a partir de objetos reales.
- Acondicionar los modelos generados para poder ser utilizarlos en entornos virtuales.
- Diseñar la interfaz gráfica del compendio.
- Unir modelos y elementos interactivos mediante programación en la interfaz gráfica.

Para la evaluación se desarrollará un protocolo de pruebas. Este consistirá en una prueba en vivo por parte del cliente seguido de una serie de preguntas sobre ciertos aspectos de la aplicación como la facilidad de manejo de la misma y la calidad de aspectos de la interfaz gráfica como letra y uso de colores.

El protocolo será aplicado al cliente que solicitó el software para verificar que cumpla con los puntos especificados. Esto se hace con la suposición de que el cliente ya tiene estudiado al usuario objetivo, por lo que ya tiene identificadas las necesidades que debe cumplir el software. De esta forma será posible evaluar el proyecto desde un punto técnico.

El primer punto por considerar será que el cliente no recibirá instrucciones durante el uso del programa, a menos que el cliente no pueda utilizar una función básica durante un período de tiempo prolongado ya que se planea eventualmente distribuirlo al público universitario, el cual no va a contar con apoyo directo del desarrollador. En caso de que el cliente necesite ayuda, esto servirá como un indicador para futuras mejoras.

Al otorgarle al cliente el programa se le instruirá usar sus funciones durante un período de tiempo de entre cinco a quince minutos para que el cliente pueda adecuarse a las funciones del software y usarlo de la manera que le parezca más conveniente.

Una vez transcurrido el tiempo de interacción con el programa, se retirará el dispositivo móvil y se le otorgará al cliente la encuesta en la cual se evaluarán los elementos estéticos y dinámicos del software mediante una serie de preguntas relacionadas a los diferentes elementos de la aplicación.

Después de usar la aplicación, marque la opción que mejor representa su opinión a las siguientes aseveraciones:

Tabla 10 Encuesta de evaluación

1	Resultó fácil manejar la aplicación.	Si	No	No sé
2	El texto en la aplicación fue fácil de leer.	Si	No	No sé
3	Los modelos usados en la aplicación son adecuados.	Si	No	No sé
4	El uso de la realidad aumentada es intuitivo.	Si	No	No sé
5	La interacción con los modelos es intuitiva y fácil.	Si	No	No sé
6	Las transiciones entre pantallas son rápidas.	Si	No	No sé
7	La elección de colores fue adecuada.	Si	No	No sé
8	La medición de los modelos es intuitiva.	Si	No	No sé
9	Fue difícil moverse a través de los menús de la aplicación.	Si	No	No sé
10	El dato de la medida es útil.	Si	No	No sé
11	El programa proporcionó instrucciones suficientes para su manejo.	Si	No	No sé
12	La realidad aumentada hace fácil el ver e interactuar con los modelos.	Si	No	No sé
13	La apariencia de la aplicación es atractiva.	Si	No	No sé
14	La elección de formas en la aplicación fue adecuada.	Si	No	No sé
15	Leer en la aplicación es fácil.	Si	No	No sé
16	El moverse entre las funciones de la aplicación es rápido.	Si	No	No sé
17	El sistema de medición dentro de la aplicación es fácil de usar.	Si	No	No sé
18	La aplicación es estéticamente agradable.	Si	No	No sé
19	La aplicación fue clara.	Si	No	No sé
20	Al iniciar la aplicación, es fácil saber cómo usar los comandos.	Si	No	No sé
21	Manejar los modelos en realidad aumentada es complicado.	Si	No	No sé
22	Los datos numéricos que proporciona la aplicación son útiles para su utilización	Si	No	No sé
23	La aplicación es rápida al usarse.	Si	No	No sé
24	Fue fácil entender el cómo usar la aplicación.	Si	No	No sé
25	Fue sencillo manejar los modelos en la aplicación.	Si	No	No sé
26	El dato de la medida es fácil de entender.	Si	No	No sé
27	La elección de texto en la aplicación fue adecuada.	Si	No	No sé

6.2 Resultados

La prueba se realizó de acuerdo a lo establecido en el protocolo de pruebas. Se acudió con la Dra. E. Mahuina Campus Castolo, quien fue el cliente que solicitó el software. El cliente usó la aplicación durante un período de 12 minutos antes de responder la encuesta.

El equipo utilizado para realizar la prueba fue un celular LG Magna H500, el cual cuenta con una cámara de 8.0 Mega-píxeles y un procesador de 1.3 Ghz. [81]. Siendo este un celular de gama media, la prueba con este equipo tiene el propósito de asegurar que una gran gama de dispositivos móviles pueda utilizar el software. El funcionamiento de la aplicación fue corroborado previo a la prueba para evitar fallos técnicos ajenos al software desarrollado.



Figura 84. Prueba de la aplicación con el cliente.

Durante el período de prueba, el cliente manifestó dificultades con la función de posicionamiento del modelo y con la interacción con los marcadores de medición. Sin embargo, con el uso, el cliente se fue acostumbrando a estas funciones. Relacionado a esto, el cliente manifestó comentarios sobre la necesidad de incluir más instrucciones para ciertas funciones. Estos fueron resumidos al final de la encuesta de forma escrita y se encuentran transcritos a continuación.

Después de usar la aplicación, marque la opción que mejor representa su opinión a las siguientes aseveraciones:

Tabla 11 Resultados de la encuesta

1	Resultó fácil manejar la aplicación.	<input checked="" type="radio"/>	No	No sé
2	El texto en la aplicación fue fácil de leer.	<input checked="" type="radio"/>	No	No sé
3	Los modelos usados en la aplicación son adecuados.	<input checked="" type="radio"/>	No	No sé
4	El uso de la realidad aumentada es intuitivo.	<input checked="" type="radio"/>	No	No sé
5	La interacción con los modelos es intuitiva y fácil.	<input checked="" type="radio"/>	No	No sé
6	Las transiciones entre pantallas son rápidas.	<input checked="" type="radio"/>	No	No sé
7	La elección de colores fue adecuada.	<input checked="" type="radio"/>	No	No sé
8	La medición de los modelos es intuitiva.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
9	Fue difícil moverse a través de los menús de la aplicación.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
10	El dato de la medida es útil.	<input checked="" type="radio"/>	No	No sé
11	El programa proporcionó instrucciones suficientes para su manejo.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
12	La realidad aumentada hace fácil el ver e interactuar con los modelos.	<input checked="" type="radio"/>	No	No sé
13	La apariencia de la aplicación es atractiva.	<input checked="" type="radio"/>	No	No sé
14	La elección de formas en la aplicación fue adecuada.	<input checked="" type="radio"/>	No	No sé
15	Leer en la aplicación es fácil.	<input checked="" type="radio"/>	No	No sé
16	El moverse entre las funciones de la aplicación es rápido.	<input checked="" type="radio"/>	No	No sé
17	El sistema de medición dentro de la aplicación es fácil de usar.	<input checked="" type="radio"/>	No	No sé
18	La aplicación es estéticamente agradable.	<input checked="" type="radio"/>	No	No sé
19	La aplicación fue clara.	<input checked="" type="radio"/>	No	No sé
20	Al iniciar la aplicación, es fácil saber cómo usar los comandos.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
21	Manejar los modelos en realidad aumentada es complicado.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
22	Los datos numéricos que proporciona la aplicación son útiles para su utilización	<input checked="" type="radio"/>	No	No sé
23	La aplicación es rápida al usarse.	<input checked="" type="radio"/>	No	No sé
24	Fue fácil entender el cómo usar la aplicación.	<input type="radio"/> Si	<input checked="" type="radio"/> No	No sé
25	Fue sencillo manejar los modelos en la aplicación.	<input checked="" type="radio"/>	No	No sé
26	El dato de la medida es fácil de entender.	<input checked="" type="radio"/>	No	No sé
27	La elección de texto en la aplicación fue adecuada.	<input checked="" type="radio"/>	No	No sé

Comentarios adicionales:

“Hace falta un instructivo o guía para ciertas funciones, como el significado del <<Número>> y <<enviar>> así como la opción <<colocar>>.”

El escalamiento puede ser más adaptado a la realidad

Los marcadores podrían aparecer fuera del modelo para que sea más sencillo colocarlos y que no <<se pierdan>> dentro del modelo”

6.3 Discusión

Con base en los resultados de la prueba y la encuesta, se pueden notar ciertos puntos importantes. El primero de estos fue que el cliente realizó la prueba dentro del tiempo establecido, lo cual indica que el proceso de aprendizaje de la aplicación se encuentra dentro de esa ventana de tiempo.

Sin embargo, es necesario recalcar algunos problemas: El cliente presentó dificultades durante el proceso de colocar el modelo y mover los marcadores de medición. Lo cual indica que estas funciones no son muy intuitivas para los usuarios.

Estos comentarios se ven reforzados por los resultados de la encuesta y sus subsecuentes comentarios:

- Las preguntas 1, 11, 19, 20 y 24 tienen que ver con que tan fácil es usar la aplicación y los diferentes comandos dentro de la misma. En esta respuesta se tiene el mayor número de respuestas negativas, ya que el cliente respondió que no a las preguntas 11, 20 y 24. Estas respuestas indican que, para el cliente, es necesario que la aplicación cuente con una serie de instrucciones para entender sus funciones. También hay que recalcar que las respuestas a las preguntas 1 (facilidad de manejo) y 19 (Claridad de la aplicación), contrastan con las demás respuestas. Esto indica que la aplicación en si no es clara y fácil de entender en un principio y que sería necesario indicar el funcionamiento de los distintos comandos para facilitar la experiencia de los usuarios.
- Las preguntas 2, 15 y 27 se formularon respecto a la elección y uso de texto en la aplicación, y las preguntas 7, 13, 14 y 18 tienen que ver con la apariencia estética de la aplicación. Todas fueron respondidas de forma positiva, por lo que se considera que el objetivo se cumplió de una manera funcional.

- Las preguntas 6, 9, 16 y 23 solicitaron al cliente que evaluara la velocidad de transición entre pantallas y funciones de la aplicación. De acuerdo con el cliente, la velocidad de la aplicación, tanto en sus transiciones como en su utilización fue satisfactoria, lo cual indica que las funciones de la aplicación no tuvieron problemas con respecto a la velocidad de procesamiento del hardware utilizado.
- Las preguntas 3, 5 y 25 se centran en el uso y calidad de los modelos de fetos humanos usados en la aplicación. De acuerdo a las respuestas del cliente, los modelos usados en la aplicación son adecuados para el uso que se les va a dar. Además, la interacción con los modelos es fácil e intuitiva. Esto indica que el nivel alcanzado en los procesos de captura, corrección e implementación de los modelos es satisfactorio para el cliente.
- Las preguntas 8, 10, 17, 22 y 26 se centran en el sistema de medición de los modelos. De acuerdo con las respuestas del cliente, este considera que el dato otorgado por la función es fácil de entender y es útil. Sin embargo, a pesar de que considera la función fácil de usar, el cliente recalca que esta función no es intuitiva en su utilización, lo cual se refleja en los comentarios en los cuales menciona que la medición puede adaptarse a la escala y que se puede mejorar la interacción con los puntos de medición.
- Finalmente, las preguntas 4, 12 y 21 evaluaron la integración de la tecnología de realidad aumentada en la aplicación. De acuerdo con las respuestas del cliente, esta tecnología es intuitiva, fácil de usar y permite interactuar con los modelos de forma fácil. Durante el proceso de prueba, el cliente presentó reacciones positivas con respecto a observar el uso que se le dio a la tecnología, considerándola como una buena herramienta para los usuarios.

Con estos datos, se cuenta con información suficiente para elaborar las conclusiones del proyecto.

7. Conclusiones

Después de todo el proceso de diseño conceptual y de detalle se cuenta con un software que se puede distribuir y ejecutar en dispositivos móviles de gama media que cuenten con una cámara y una pantalla táctil. El desarrollo de este trabajo implicó la integración de varias áreas del conocimiento del campo de la ingeniería como álgebra lineal, geometría, variable compleja, visión artificial, programación orientada a objetos, modelado digital y la aplicación del proceso de diseño.

Valorando los resultados obtenidos de la prueba con el cliente y el análisis que se realizó en la sección anterior, se puede decir que el software generado cumplió con los objetivos planteados. Desde un punto de vista funcional, el programa permite la visualización de más de un modelo digital de fetos humanos con la herramienta de realidad aumentada. Adicional a esto, la aplicación permite realizar mediciones de estos modelos de acuerdo con las especificaciones del cliente.

En términos estéticos, no hubo necesidad de implementar mejoras ya que el aspecto de la aplicación fue del agrado del cliente. Cabe recalcar, que el aspecto de la aplicación es meramente funcional y sería necesario contar con el apoyo de un profesional en diseño para crear un aspecto más atractivo para el usuario.

Durante el proceso inicial de captura de modelos, se realizaron un gran número de pruebas para conseguir modelos de calidad aceptable. Gracias a esto se llegó a la conclusión de que es necesario contar con un ambiente idóneo para la correcta captura. Esto incluye tener una iluminación adecuada y contar con superficies que no tengan colores uniformes o demasiado claros. En sí, el proceso de captura por fotogrametría lleva bastante tiempo y los modelos resultantes presentan muchas imperfecciones a menos de que se cuente con el equipo adecuado, por lo cual se considera conveniente adquirir un escáner digital en caso de ampliar la selección de modelos para la fetoteca. Esto con el propósito de que el proceso pueda agilizarse.

El tener un mejor método de captura también impactaría el proceso y la duración de la corrección de los modelos en software especializado. En caso de seguir con el proyecto, es necesario contar con la participación de un especialista en diseño gráfico para mejorar la calidad de los modelos generados.

Pasando al desarrollo de la aplicación en sí es necesario recalcar la evolución del mismo durante el proceso de diseño. Inicialmente, la funcionalidad de realidad aumentada se implementó mediante el uso de marcadores QR, siendo este el primer nivel de la realidad aumentada. Durante la segunda mitad del proyecto, se decidió dejar de depender de los marcadores y pasar al segundo nivel el cual coloca los modelos en superficies planas que fije el usuario. Esto con el propósito de cumplir con la visión del proyecto de generar una herramienta que no dependa de elementos externos.

Relacionado a lo anterior, es necesario considerar mejorar el sistema de medición de la aplicación. En primer lugar, se puede mejorar la interacción de los marcadores para que estos permanezcan en la superficie del modelo sin perderse dentro de la superficie de este. En caso de que el software pudiese ser adaptado a algún tipo de visor, podría considerarse la adición de sistemas hápticos para la interacción con los marcadores y el modelo.

Es necesario agregar instrucciones claras en futuras iteraciones del programa para que el aprendizaje de las distintas funciones del software sea más ameno para los usuarios. Esto se puede implementar fácilmente en el entorno de Unity por lo que se considera que esto puede ser una de las primeras modificaciones a futuro. Adicionalmente, se podrían incluir videos e imágenes demostrativas dentro de la aplicación como una herramienta adicional.

En cuanto a su objetivo a futuro, el cliente solicitó este proyecto para otorgarles una herramienta a los alumnos de la facultad de medicina. Esto con el propósito de que ellos puedan complementar su aprendizaje en temas de embriología sin largos procesos de adquisición del material necesario para sus prácticas. Este proyecto es en sí, un primer acercamiento a esta realidad utilizando el tema de embriología debido a la necesidad detectada. A futuro, sería posible expandir el compendio de modelos para cubrir una gran cantidad de temas de embriología y anatomía.

Cabe mencionar que el futuro de este proyecto no está limitado a temas relacionados con la medicina, ya que es posible adecuar el concepto a la enseñanza y divulgación de otras áreas del conocimiento como la ingeniería, arquitectura, biología entre otros temas. Ampliar la gama de aplicaciones que abarquen distintos temas permitiría extender esta herramienta a un número mayor de usuarios.

Adicionalmente, durante el desarrollo del proyecto se hizo evidente la importancia de la propiedad intelectual y el registro de la misma. Esto con el propósito de asegurar los créditos de desarrolladores del proyecto.

Como conclusión final del trabajo, es pertinente mencionar que este proceso, al igual que la ingeniería en muchos campos, es un proceso continuo e interdisciplinario. El aprendizaje de un profesional no termina al final de la carrera, sino que continúa durante el resto de su trayectoria. La continua generación de herramientas y tecnología abre la puerta a la generación de nuevos productos, ideas y métodos. De igual forma, un profesional no puede realizar un proyecto de gran magnitud por su cuenta, es necesaria la cooperación entre especialistas de diversos campos para generar un producto o servicio más completo enfocado en beneficio de la sociedad.

8. Trabajo a futuro

Gracias a las experiencias obtenidas durante la realización de este proyecto y con base en las conclusiones descritas en el capítulo anterior, se pueden desglosar los puntos en los cuales es necesario continuar el desarrollo o implementar nuevos sistemas para mejorar la experiencia del usuario con el proyecto.

- Para futuras capturas de modelos es necesario conseguir un espacio con las condiciones de iluminación adecuadas para evitar ruido causado por el ambiente.
- Para agilizar el proceso de captura, se sugiere la adquisición de un escáner digitalizador para evitar los problemas relacionados al proceso de fotogrametría.
- Considerar la colaboración con especialistas en diseño gráfico para mejorar la calidad de las texturas de los modelos y la apariencia de la aplicación.
- Considerar la integración de sistemas hápticos en conjunto con la tecnología de realidad mixta para incrementar la experiencia del usuario.
- Implementar instrucciones en la aplicación para las funciones que no quedaron claras para el cliente.
- Mejorar el sistema de medición para evitar que los indicadores se queden dentro del modelo.
- Considerar incluir un indicador de escala dentro de la escena de visualización y que se tenga la opción de cambiar la escala del modelo.
- Exportar el software a otras plataformas para ampliar la cantidad de usuarios que pueden acceder a él.
- Considerar la posibilidad de expandir la temática de la aplicación a otras áreas de la medicina o de la educación en general.

9. Referencias

- [1] UNESCO, “Desarrollo Sostenible | Educación | Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura”, 2017. [En línea]. Disponible en: <http://www.unesco.org/new/es/education/themes/leading-the-international-agenda/education-for-sustainable-development/sustainable-development/>. [Consultado: 28-ago-2017].
- [2] L. F. A. Hernandez y C. B. I. Castañeda, *La educación médica frente a los retos de la sociedad del conocimiento*, vol. 140, núm. 4. México: Academia Nacional de Medicina, 2004.
- [3] M. E. C. M. E. C. M. E. C. Santos, A. Chen, T. Taketomi, G. Yamamoto, J. Miyazaki, y H. Kato, “Augmented reality learning experiences: Survey of prototype design and evaluation”, *IEEE Trans. Learn. Technol.*, vol. 7, núm. 1, pp. 38–56, 2014.
- [4] P. Chen *et al.*, “A review of using Augmented Reality in Education from 2011 to 2016”, *Innov. Smart Learn.*, pp. 13–14, 2017.
- [5] Digi Capital, “AR/VR Dealmakers invested over \$800M in Q2 2017”, *Julio 1*, 2017. [En línea]. Disponible en: <http://www.digi-capital.com/news/2017/07/arvr-dealmakers-invested-over-800m-in-q2-2017/#.WXvLKmLhC03>. [Consultado: 27-jul-2017].
- [6] Digi Capital, “Apple and Facebook to drive mobile AR over 1B users and \$60B by 2021”, *Junio*, 2017. [En línea]. Disponible en: <http://www.digi-capital.com/news/2017/06/mobile-ar-to-top-a-billion-users-and-60-billion-by-2021/#.WXwSeWLhC03>. [Consultado: 27-jul-2017].
- [7] Digi Capital, “After a mixed year, mobile AR to drive \$108 billion VR/AR market by 2021”, *Enero*, 2017. [En línea]. Disponible en: <http://www.digi-capital.com/news/2017/01/after-mixed-year-mobile-ar-to-drive-108-billion-vrar-market-by-2021/#.WXwNs2LhC03>. [Consultado: 27-jul-2017].
- [8] MIT, “Augmented Reality Summit @ MIT Media Lab | MIT Sloan Fellows”, 2017. [En línea]. Disponible en: <http://mitsloan.mit.edu/fellows/leadership-blog/augmented-reality-summit-at-mit-media-lab-brings-the-worlds-ar-pioneers-together-on-the-ar-frontier/>. [Consultado: 28-ago-2017].
- [9] CeDiCyT, “Realidad Aumentada - Instituto Politécnico Nacional”, *CeDiCyT*, 2017. [En línea]. Disponible en: <http://www.cedicyt.ipn.mx/RevConversus/Paginas/RealidadAumentada.aspx>. [Consultado: 28-ago-2017].
- [10] CADAESHMEIP, “Realidad aumentada con eDrawings - CAD AVSHMEIP”, *Noticias Solidworks*, 2016. [En línea]. Disponible en: <https://www.cadavshmeip.com/realidad-aumentada-con-edrawings-mucho-antes-de-pokemon-go/>. [Consultado: 28-ago-2017].
- [11] J. Alvarez, “Posibilidades didácticas de la realidad virtual - Revista de Educación Virtual”, *Revista Educación Virtual*, 2016. [En línea]. Disponible en: <https://revistaeducacionvirtual.com/archives/2024>.

- [Consultado: 28-ago-2017].
- [12] AIR, "AIR: Artificial Intelligence and Rendering Research Group". [En línea]. Disponible en: <http://air.esi.uclm.es/linea2.php>. [Consultado: 28-ago-2017].
- [13] R. Jimenez Cano, "Microsoft impulsa la realidad mixta con Hololens | Tecnología | EL PAÍS", 2017. [En línea]. Disponible en: https://elpais.com/tecnologia/2017/05/11/actualidad/1494523600_396504.html. [Consultado: 28-ago-2017].
- [14] G. J. Kim, *Designing Virtual Reality Systems. The Structured Approach*. NJ, USA: Springer, 2005.
- [15] R. Enriquez Caro, "ADMINISTRACIÓN MODERNA: La realidad aumentada: Concepto que llegó al Perú", *Administración Moderna*, 2015. [En línea]. Disponible en: <http://www.administracionmoderna.com/2015/11/la-realidad-aumentada-concepto-que-llego-al-peru.html>. [Consultado: 28-ago-2017].
- [16] R. Technologies, "What is Mixed Reality (MR)? Ultimate Guide to Mixed Reality (MR) Technology", *Reality Technologies*, 2016. [En línea]. Disponible en: <http://www.realitytechnologies.com/mixed-reality>. [Consultado: 28-ago-2017].
- [17] T. Bishop, "HoloLens launches in 6 new countries in first overseas expansion for Microsoft's mixed reality headset – GeekWire", *GeekWire*, 2016. [En línea]. Disponible en: <https://www.geekwire.com/2016/hololens-launches-6-new-countries-first-overseas-expansion-microsofts-mixed-reality-headset/>. [Consultado: 28-ago-2017].
- [18] Accenture, "Mixed reality brings real benefits to enterprises - Accenture", 2016. [En línea]. Disponible en: https://www.accenture.com/t20160526T011326__w_/us-en/_acnmedia/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Industries_19/Accenture-Mixed-Reality-POV-Final.pdf. [Consultado: 28-ago-2017].
- [19] J. Engel, V. Koltun, y D. Cremers, "Direct Sparse Odometry", *Cornell University Library*, 09-jul-2016. [En línea]. Disponible en: <http://arxiv.org/abs/1607.02565>. [Consultado: 28-ago-2017].
- [20] A. Sagrado Strik, "La realidad aumentada, una nueva forma de publicidad", *Forbes*, 2017. [En línea]. Disponible en: <https://www.forbes.com.mx/realidad-aumentada-la-nueva-publicidad/>. [Consultado: 28-ago-2017].
- [21] G. Duncan, "Augmented Reality Shoe Shopping with Goertz and the Kinect | Coding4Fun Kinect Projects | Channel 9", *Channel 9*, 2012. [En línea]. Disponible en: <https://channel9.msdn.com/coding4fun/kinect/Augmented-Reality-Shoe-Shopping-with-Goertz-and-the-Kinect>. [Consultado: 28-ago-2017].
- [22] R. Espinosa de los Monteros Corona, "Una nueva herramienta para la Arquitectura", UNAM, 2015.
- [23] K. Ocampo, "El mercado de los videojuegos en Estados Unidos, para el 2017, será de 40.5 billones | Atomix", *Atomix*, 2013. [En línea].

- Disponible en: <http://atomix.vg/el-mercado-de-los-videojuegos-en-estados-unidos-para-el-2017-sera-de-40-5-billones/>. [Consultado: 28-ago-2017].
- [24] González Alberto, “Pokémon GO generó más ingresos en 2016 que el film Batman v. Superman - Vandal”, *Vandal*, 2017. [En línea]. Disponible en: <http://www.vandal.net/noticia/1350695589/pokemon-go-genero-mas-ingresos-en-2016-que-el-film-batman-v-superman/>. [Consultado: 28-ago-2017].
- [25] “25 mejores imágenes de Augmented Reality en Pinterest”. [En línea]. Disponible en: <https://www.pinterest.com.mx/kagapios/augmented-reality/>. [Consultado: 28-ago-2017].
- [26] J. Rodrgick, “5 More Ways VR is Changing the World! - VR Fitness Insider”, 2016. [En línea]. Disponible en: <https://www.vrfitnessinsider.com/5-ways-vr-changing-world/>. [Consultado: 28-ago-2017].
- [27] R. A. Española, “Ontogenia”, *Diccionario de la real academia española (23.a ed.)*, 2014. [En línea]. Disponible en: <http://dle.rae.es/srv/search?m=30&w=ontogenia>. [Consultado: 05-ago-2017].
- [28] M. deVries, “Ontogenia y filogenia”, *Museo de paleontología de la universidad de california*, 2003. [En línea]. Disponible en: http://evolution.berkeley.edu/evolibrary/article/side_0_0/ontogeny_01_sp. [Consultado: 07-ago-2017].
- [29] G. sur Editorial, Ed., “La reproducción: el embarazo y el parto”, en *Enciclopedia integral para el estudiante*, Primera ed., España: MMII Ediciones credimar, 2002, pp. 550–556.
- [30] B. Planeta, Ed., “Feto”, *Gran enciclopedia hispánica tomo 8*. Bansa Planeta, España, p. 2564, 2006.
- [31] J. A. DiPietro, “Prenatal Development”, *Encycl. Infant Early Child. Dev.*, vol. 2, pp. 604–614, 2008.
- [32] OMS, “OMS | Nacimientos prematuros”, *WHO*, 2017. [En línea]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs363/es/>. [Consultado: 10-ago-2017].
- [33] M. A. Rysavy *et al.*, “Between-Hospital Variation in Treatment and Outcomes in Extremely Preterm Infants”, *N. Engl. J. Med.*, vol. 372, núm. 19, pp. 1801–1811, may 2015.
- [34] World Health Organization, B. Defects, G. Newborn, y C. Mortality, “Birth defects Report by the Secretariat”, en *World Health Assembly*, 2010, núm. May 2006, pp. 1–5.
- [35] OMS, “WHO | Congenital anomalies”, *WHO*, 2017. [En línea]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs370/en/>. [Consultado: 10-ago-2017].
- [36] E. Narváez, H. Blvd, E. Narváez-Hernández, y E. I. Murillo-Rábago, “Herramientas de estudio utilizadas por alumnos de ciencias de la salud en la materia de Anatomía”, *Inv Ed Med*, vol. 3, núm. 12, pp. 204–208, 2014.
- [37] F. Correa Alarcon, “Conservacion de piezas anatómicas en seco

- mediante el método de privas”, *REDVET. Rev. Electrónica Vet.*, vol. VI, pp. 1–7, 2005.
- [38] C. Alfonso, M. Gómez, y J. Alejandro, “Conservación y elaboración de piezas anatómicas con sustancias diferentes al formol en la Facultad de Ciencias Agropecuarias de la Universidad de La Salle **”, *Rev. Med. Vet. (Toulouse)*, vol. 29, núm. 2, pp. 531–536, 2011.
- [39] M. Liseth *et al.*, “A RTÍCULO DE REFLEXIÓN C omparación de técnicas de conservación morfológica y su posible aplicación para la enseñanza de la anatomía”, vol. 6, núm. 3, 2014.
- [40] Biodigital, “BioDigital: 3D Human Visualization Platform for Anatomy and Disease”, 2017. [En línea]. Disponible en: <https://www.biodigital.com/product>. [Consultado: 06-nov-2017].
- [41] Autodesk, “Software de modelado 3D | Aplicaciones y recursos gratuitos | Autodesk”, 2017. [En línea]. Disponible en: <https://www.autodesk.mx/solutions/3d-modeling-software>. [Consultado: 22-ago-2017].
- [42] C. González Morcillo y D. Vallejo Fernández, *Síntesis de fundamentos de imagen 3D un enfoque práctico a blender*, Primera ed. Castilla la Mancha: Universidad de Castilla-La mancha, Centro de excelencia de software libre, 2009.
- [43] O. university Press, “escala – definición de escala en español del Diccionarios Oxford”, *Oxford living dictionaries*, 2017. [En línea]. Disponible en: <https://es.oxforddictionaries.com/definicion/escala>. [Consultado: 22-ago-2017].
- [44] M. Savva, A. X. Chang, G. Bernstein, C. D. Manning, y P. Hanrahan, “On Being the Right Scale: Sizing Large Collections of 3D Models”, *SIGGRAPH Asia 2014 Indoor Scene Underst. Where Graph. Meets Vis.*, vol. 4, 2014.
- [45] Unity, “Unity - Manual: Models”, 2017. [En línea]. Disponible en: <https://docs.unity3d.com/Manual/FBXImporter-Model.html>. [Consultado: 23-ago-2017].
- [46] Unity, “Unity - Manual: 3D formats”, *Unity documentation*, 2017. [En línea]. Disponible en: <https://docs.unity3d.com/Manual/3D-formats.html>. [Consultado: 30-ago-2017].
- [47] M. Bousquet, “3D Modeling Standards: Real-World Scale - TurboSquid Blog”, 2010. [En línea]. Disponible en: <https://blog.turbosquid.com/2010/11/23/3d-modeling-real-world-scale/>. [Consultado: 03-sep-2017].
- [48] Rayek, “Units - BlenderWiki”, 2013. [En línea]. Disponible en: <https://wiki.blender.org/index.php/User:Rayek/Doc:2.6/Manual/Interface/Units>. [Consultado: 04-sep-2017].
- [49] B. Roskes, “Getting Started in Tinkercad Getting Started in Tinkercad Step 1: Have a Look Around”. [En línea]. Disponible en: <http://www.3dvinci.net/PDFs/GettingStartedInTinkercad.pdf>. [Consultado: 04-sep-2017].
- [50] Autodesk, “To Specify the Drawing Units | AutoCAD Architecture | Autodesk Knowledge Network”, 2017. [En línea]. Disponible en:

- <https://knowledge.autodesk.com/support/autocad-architecture/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/AutoCAD-Architecture/files/GUID-652DA67D-C364-43C6-A71D-FD1E676A0374-htm.html>. [Consultado: 04-sep-2017].
- [51] Solidworks, “2017 SOLIDWORKS Help - Units and Dimension Standard”, 2017. [En línea]. Disponible en: http://help.solidworks.com/2017/english/SolidWorks/sldworks/hidd_units_dim_std.htm. [Consultado: 04-sep-2017].
- [52] Autodesk, “AUTODESK® REMAKE GETTING STARTED GUIDE Welcome to Autodesk® ReMake!”, 2016. [En línea]. Disponible en: http://remake.autodesk.com/assets/Autodesk_ReMake_Guide_01.pdf. [Consultado: 04-sep-2017].
- [53] Skanect, “Skanect 3D Scanning Software By Occipital - The Easiest Way To 3D Scan With the Structure Sensor and Kinect-like 3D Sensors”, 2017. [En línea]. Disponible en: <http://skanect.occipital.com/>. [Consultado: 04-sep-2017].
- [54] H. Hasnain, “vertices - Mesh Repair, "Torn" Mesh with Holes, Shrinkwrap Modifier? - Blender Stack Exchange”. [En línea]. Disponible en: <https://blender.stackexchange.com/questions/24031/mesh-repair-torn-mesh-with-holes-shrinkwrap-modifier>. [Consultado: 26-sep-2017].
- [55] “have different quality for textures of game objects in one scene - Unity Answers”, 2015. [En línea]. Disponible en: <http://answers.unity3d.com/questions/1040694/have-different-quality-for-textures-of-game-object.html>. [Consultado: 29-sep-2017].
- [56] N. Simón, “Reconocimiento de objetos mediante sensor 3D Kinect”, Universidad Carlos III de Madrid., 2012.
- [57] Microsoft, “Kinect for Windows Sensor Components and Specifications”, *Kinekt for Windows 1.5, 11.6, 1.7, 1.8*, 2017. [En línea]. Disponible en: <https://msdn.microsoft.com/en-us/library/jj131033.aspx?f=255&MSPPErr=-2147217396>. [Consultado: 01-oct-2017].
- [58] Microsoft, “Skeletal Tracking”, *Kinekt for Windows 1.5, 11.6, 1.7, 1.8*, 2017. [En línea]. Disponible en: <https://msdn.microsoft.com/en-us/library/hh973074.aspx#ID4ENB>. [Consultado: 01-oct-2017].
- [59] E. Lachat, H. Macher, M.-A. Mittet, T. Landes, y P. Grussenmeyer, “FIRST EXPERIENCES WITH KINECT V2 SENSOR FOR CLOSE RANGE 3D MODELLING”, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XL, núm. 5/W4, pp. 93–100, 2015.
- [60] Joelfrax, “Escáner 3D: Tipos y Aplicaciones | Frax Design”, 2015. [En línea]. Disponible en: <http://frax3d.com/escaner-3d-tipos-y-aplicaciones>. [Consultado: 02-oct-2017].
- [61] Creaform, “PF: Preguntas frecuentes | Escáneres Go!SCAN 3D”, 2017. [En línea]. Disponible en: <https://www.goscan3d.com/es/preguntas-frecuentes>. [Consultado: 02-oct-2017].
- [62] A. Mason, “Making 3D Models with Photogrammetry Getting Started with Agisoft PhotoScan.” [En línea]. Disponible en: <https://thehaskinssociety.wildapricot.org/resources/Documents/Tutorials/>

- PhotogrammetrywithPhotoScanTutorial.pdf. [Consultado: 02-oct-2017].
- [63] J. A. Sánchez Sobrino, "Introducción a la fotogrametría", 2007. [En línea]. Disponible en: ftp://ftp.unsj.edu.ar/agrimensura/Fotogrametria/Unidad6/Introduccion_a_la_Fotogrametria.pdf. [Consultado: 02-oct-2017].
- [64] Autodesk, "ReCap | Reality Capture And 3D Scanning Software | Autodesk", 2017. [En línea]. Disponible en: <https://www.autodesk.com/products/recap/overview>. [Consultado: 13-oct-2017].
- [65] A. Chaun, "Quiero disfrutar de realidad virtual en México, ¿qué opciones tengo?", *Xataka*, 2016. [En línea]. Disponible en: <https://www.xataka.com.mx/accesorios/quiero-disfrutar-de-realidad-virtual-en-mexico-que-opciones-tengo>. [Consultado: 13-nov-2017].
- [66] "Guante háptico fabricado en España", *realovirtual*, 2014. [En línea]. Disponible en: <https://www.realovirtual.com/noticias/1077/guante-haptico-fabricado-espana>. [Consultado: 13-nov-2017].
- [67] G. Sama, "Microsoft HoloLens Previa - CNET en Español", 2017. [En línea]. Disponible en: <https://www.cnet.com/es/analisis/microsoft-hololens/primer-vistazo/>. [Consultado: 13-nov-2017].
- [68] M. López Moreno, "Realidad aumentada en la educación", 2014. [En línea]. Disponible en: <https://www.nubemia.com/realidad-aumentada-en-la-educacion/>. [Consultado: 13-nov-2017].
- [69] E. W. Weisstein, "Euler Angles", 2017. [En línea]. Disponible en: <http://mathworld.wolfram.com/EulerAngles.html>. [Consultado: 13-nov-2017].
- [70] A. Jaramillo Botero, "DESCRIPCIONES Y TRANSFORMACIONES ESPACIALES", 2005. [En línea]. Disponible en: http://www.wag.caltech.edu/home/ajaramil/libro_robotica/transformacion_es_espaciales.pdf. [Consultado: 13-nov-2017].
- [71] Academic, "¿Qué son los cuaterniones?", 2010. [En línea]. Disponible en: <http://www.esacademic.com/dic.nsf/eswiki/317441>.
- [72] G.F Torres del Castillo, "La representación de rotaciones mediante cuaterniones", 1999.
- [73] Mauricio López, "Otro tutorial de Quaternions – Rumbo Loxodromico", 2012. [En línea]. Disponible en: <https://rastergraphics.wordpress.com/2012/07/20/otro-tutorial-de-quaternions/>.
- [74] Unity, "Unity - Manual: Rotation and Orientation in Unity", *Unity documentation*, 2017. [En línea]. Disponible en: <https://docs.unity3d.com/Manual/QuaternionAndEulerRotationsInUnity.html>. [Consultado: 13-nov-2017].
- [75] "java - Que es una Instancia POO - Stack Overflow en español", 2016. [En línea]. Disponible en: <https://es.stackoverflow.com/questions/7576/que-es-una-instancia-poo>.
- [76] A. Hernández, "Resistivas vs Capacitivas: Conoce cuál pantalla táctil es la mejor para ti", 2012. [En línea]. Disponible en: <http://www.biobiochile.cl/noticias/2012/03/04/resistivas-vs-capacitivas->

- conoce-cual-pantalla-tactil-es-la-mejor-para-ti.shtml. [Consultado: 13-nov-2017].
- [77] Unity, “Unity - Manual: Input de Dispositivo Móvil”, *Unity documentation*, 2017. [En línea]. Disponible en: <https://docs.unity3d.com/es/current/Manual/MobileInput.html>. [Consultado: 13-nov-2017].
- [78] J. Glover, “Learn and Understand Raycasting in Unity3D – Zenva | GameDev Academy”, 2017. [En línea]. Disponible en: <https://gamedevacademy.org/learn-and-understand-raycasting-in-unity3d/>. [Consultado: 13-nov-2017].
- [79] G. Bleser, “Towards Visual-Inertial SLAM for Mobile Augmented Reality”, Technischen Universität Kaiserslautern, 2009.
- [80] Robdos Team underwater robotics, “¿Qué es una IMU?”, 2016. [En línea]. Disponible en: http://www.robdosteam.com/wp-content/uploads/2016/12/08.IMU_.pdf. [Consultado: 13-nov-2017].
- [81] LG, “Celulares Magna H500, Curvo, mayor ángulo para selfies, LG México”, 2017. [En línea]. Disponible en: <http://www.lg.com/mx/celulares/lg-H500>. [Consultado: 13-nov-2017].

Código Fuente

INTERFAZ DE INTERACCIÓN DE REALIDAD AUMENTADA

```
using UnityEngine;
using Kudan.AR.Samples;
using System.Collections;
using System.IO;
public class UserInterfaceButtons : MonoBehaviour
{
    public float rotationSpeed = 70.0f;
    bool repeatRotateLeft = false;
    bool repeatRotateRight = false;
    bool repeatRotateUp = false;
    bool repeatRotateDown = false;
    bool repeatRotate2RXYZ = false;
    bool repeatRotate2LXYZ = false;
    bool repeatRotate3RXYZ = false;
    bool repeatRotate4LXYZ = false;
    void Update ()
    {
        if (repeatRotateRight) {
            RotationRightButton();
        }
        if (repeatRotateLeft) {
            RotationLeftButton();
        }
        if (repeatRotateUp) {
            RotationUpButton();
        }
        if (repeatRotateDown)
        {
            RotationDownButton();
        }
        if (repeatRotate2RXYZ)
        {
            Rotation2RXYZButton();
        }
        if (repeatRotate2LXYZ)
        {
            Rotation2LXYZButton();
        }
        if (repeatRotate3RXYZ)
        {
            Rotation3RXYZButton();
        }
        if (repeatRotate4LXYZ)
        {
            Rotation4LXYZButton();
        }
    }

    public void RegresarMenu_1()
    {
        Interfaz.Escena = "Feto_1";
        if (SampleApp.Bandera)
        {
            GameObject.Find("SimpleUI").GetComponent<SampleApp>().StartClicked();
        }
    }
}
```

```

    }
    Application.LoadLevel("Escena_Base");
}
public void RegresarMenu_2()
{
    Interfaz.Escena = "Feto_2";
    if (SampleApp.Bandera)
    {
        GameObject.Find("SimpleUI").GetComponent<SampleApp>().StartClicked();
    }
    Application.LoadLevel("Escena_Base");
}
public void RegresarPrincipal()
{
    if (SampleApp.Bandera)
    {
        GameObject.Find("SimpleUI").GetComponent<SampleApp>().StartClicked();
    }
    Application.LoadLevel("Escena_0");
}
void RotationUpButton()
{
    GameObject.FindWithTag ("Model").transform.Rotate (0, 0, -rotationSpeed *
Time.deltaTime);
    GameObject.FindWithTag ("PlanoPuntos").transform.Rotate (0, 0, -rotationSpeed *
Time.deltaTime);
}

void RotationDownButton()
{
    GameObject.FindWithTag("Model").transform.Rotate(0, 0, rotationSpeed *
Time.deltaTime);
    GameObject.FindWithTag("PlanoPuntos").transform.Rotate(0, 0, rotationSpeed *
Time.deltaTime);
}

void Rotation2RXYZButton()
{
    GameObject.FindWithTag("Model").transform.Rotate(rotationSpeed * Time.deltaTime, -
rotationSpeed * Time.deltaTime, -rotationSpeed * Time.deltaTime, Space.World);
    GameObject.FindWithTag("PlanoPuntos").transform.Rotate(rotationSpeed * Time.deltaTime,
-rotationSpeed * Time.deltaTime, -rotationSpeed * Time.deltaTime, Space.World);
}
void Rotation2LXYZButton()
{
    GameObject.FindWithTag("Model").transform.Rotate(rotationSpeed * Time.deltaTime,
rotationSpeed * Time.deltaTime, rotationSpeed * Time.deltaTime, Space.World);
    GameObject.FindWithTag("PlanoPuntos").transform.Rotate(rotationSpeed * Time.deltaTime,
rotationSpeed * Time.deltaTime, rotationSpeed * Time.deltaTime, Space.World);
}
void Rotation3RXYZButton()
{
    GameObject.FindWithTag("Model").transform.Rotate(-rotationSpeed * Time.deltaTime, -
rotationSpeed * Time.deltaTime, -rotationSpeed * Time.deltaTime, Space.World);
    GameObject.FindWithTag("PlanoPuntos").transform.Rotate(-rotationSpeed *
Time.deltaTime, -rotationSpeed * Time.deltaTime, -rotationSpeed * Time.deltaTime,
Space.World);
}
void Rotation4LXYZButton()
{

```

```

        GameObject.FindWithTag("Model").transform.Rotate(-rotationSpeed * Time.deltaTime,
rotationSpeed * Time.deltaTime, rotationSpeed * Time.deltaTime, Space.World);
        GameObject.FindWithTag("PlanoPuntos").transform.Rotate(-rotationSpeed *
Time.deltaTime, rotationSpeed * Time.deltaTime, rotationSpeed * Time.deltaTime, Space.World);
    }
    void RotationRightButton ()
    {
        GameObject.FindWithTag ("Model").transform.Rotate (0, -rotationSpeed *
Time.deltaTime, 0,Space.World);
        GameObject.FindWithTag ("PlanoPuntos").transform.Rotate (0, -rotationSpeed *
Time.deltaTime, 0, Space.World);
    }
    void RotationLeftButton ()
    {
        GameObject.FindWithTag ("Model").transform.Rotate (0, rotationSpeed *
Time.deltaTime, 0, Space.World);
        GameObject.FindWithTag("PlanoPuntos").transform.Rotate(0, rotationSpeed *
Time.deltaTime, 0, Space.World);
    }
    public void RotationUpButtonRepeat ()
    {
        repeatRotateUp=true;
    }
    public void RotationDownButtonRepeat()
    {
        repeatRotateDown = true;
    }
    public void Rotation2RXYZButtonRepeat()
    {
        repeatRotate2RXYZ = true;
    }
    public void Rotation2LXYZButtonRepeat()
    {
        repeatRotate2LXYZ = true;
    }
    public void Rotation3RXYZButtonRepeat()
    {
        repeatRotate3RXYZ = true;
    }
    public void Rotation4LXYZButtonRepeat(),,
    {
        repeatRotate4LXYZ = true;
    }
    public void RotationRightButtonRepeat ()
    {
        repeatRotateRight=true;
    }
    public void RotationLeftButtonRepeat ()
    {
        repeatRotateLeft=true;
    }
    public void RotateLeftButtonOff ()
    {
        repeatRotateLeft = false;
        Debug.Log ("Off");
    }
    public void RotateRightButtonOff ()
    {
        repeatRotateRight = false;
        Debug.Log ("Off");
    }
}

```



```

public void RotateUpButtonOff ()
{
    repeatRotateUp = false;
    Debug.Log ("Off");
}
public void RotateDownButtonOff()
{
    repeatRotateDown = false;
    Debug.Log("Off");
}

public void Rotate2RXYZButtonOff()
{
    repeatRotate2RXYZ = false;
    Debug.Log("Off");
}
public void Rotate2LXYZButtonOff()
{
    repeatRotate2LXYZ = false;
    Debug.Log("Off");
}
public void Rotate3RXYZButtonOff()
{
    repeatRotate3RXYZ = false;
    Debug.Log("Off");
}

public void Rotate4LXYZButtonOff()
{
    repeatRotate4LXYZ = false;
    Debug.Log("Off");
}
}

```

SISTEMA DE MEDICIÓN TRIDIMENSIONAL

```

using UnityEngine;
using UnityEngine.UI;
using System.Collections;
public class Test : MonoBehaviour {
    public GameObject ball;
    public GameObject[] values;
    private Vector3[] position;
    private float distance;
    public static int NumPoints;
    //Dibujar líneas
    private float lineDrawSpeed = 10f;
    private LineRenderer lineRenderer;
    private float dist;
    private float counter;
    private int nummeasure;
    //Mostrar distancia
    public Text stringmeasure;
    void Start () {
        //Dibujar líneas
        lineRenderer = GetComponent<LineRenderer> ();
        lineRenderer.SetWidth (8f,8f);
        nummeasure = 0;
        lineRenderer.SetVertexCount (NumPoints);
    }
}

```

```

values = new GameObject[NumPoints];
position = new Vector3[NumPoints];
int j = 0;
for (int i = 0; i <= (NumPoints-1); i++) {
    GameObject go;
    if (i > 5) {
        go = Instantiate (ball, new Vector3(-
            50,j*50,0),Quaternion.identity) as GameObject;
        j++;
    } else {
        go = Instantiate (ball, new Vector3(-
            50,(int)i*50,50),Quaternion.identity) as GameObject;
    }
    go.transform.parent = GameObject.Find ("PlanoBolas").transform;
    go.transform.name = "P" + i;
    values[i] = go;
}
}
void Update () {
    for (int numball = 0; numball <= (NumPoints - 2); numball++) {
        dist = Vector3.Distance (values [numball].transform.position, values [numball +
            1].transform.position);
        distance += dist;
        lineRenderer.SetPosition (nummeasure, values [numball].transform.position);
        nummeasure++;
        while (counter < dist) {
            counter += 1f / lineDrawSpeed;
            float x = Mathf.Lerp (0, dist, counter);
            Vector3 pointA = values [numball].transform.position;
            Vector3 pointB = values [numball + 1].transform.position;
            Vector3 pointAlongLine = x * Vector3.Normalize (pointB - pointA) +
                pointA;
            lineRenderer.SetPosition (nummeasure, pointAlongLine);
        }
        counter = 0;
    }
    nummeasure = 0;
    stringmeasure.text = distance.ToString () + " mm";
    distance = 0;
}
}

```

MOVIMIENTO DE OBJETOS 3D CON LA PANTALLA TOUCH

```

using UnityEngine;
using System.Collections;
public class TouchManager : MonoBehaviour {
    GameObject g_Obj = null;
    Plane objPlane;
    Vector3 m0;
    Ray GenerateMouseRay()
    {
        Vector3 mousePosFar = new Vector3 (Input.mousePosition.x, Input.mousePosition.y,
            Camera.main.farClipPlane);
        Vector3 mousePosNear = new Vector3 (Input.mousePosition.x, Input.mousePosition.y,
            Camera.main.nearClipPlane);
        Vector3 mousePosF = Camera.main.ScreenToWorldPoint (mousePosFar);
        Vector3 mousePosN = Camera.main.ScreenToWorldPoint (mousePosNear);
        Ray mr = new Ray (mousePosN, mousePosF - mousePosN);
        return mr;
    }
}

```

```

}
void Update () {
    if (Input.GetMouseButtonDown (0)) {
        Ray mouseRay = GenerateMouseRay ();
        RaycastHit hit;
        if (Physics.Raycast (mouseRay.origin, mouseRay.direction, out hit)) {
            g_Obj = hit.transform.gameObject;
            objPlane = new Plane (Camera.main.transform.forward * -1,
            g_Obj.transform.position);
            //Offset
            Ray mRay = Camera.main.ScreenPointToRay (Input.mousePosition);
            float rayDistance;
            objPlane.Raycast (mRay, out rayDistance);
            m0 = g_Obj.transform.position - mRay.GetPoint (rayDistance);
        }
    }
    else if (Input.GetMouseButton (0) && g_Obj) {
        Ray mRay = Camera.main.ScreenPointToRay (Input.mousePosition);
        float rayDistance;
        if (objPlane.Raycast (mRay, out rayDistance))
            g_Obj.transform.position = mRay.GetPoint (rayDistance) + m0;
    }
    else if (Input.GetMouseButtonUp (0) && g_Obj) {
        g_Obj = null;
    }
}
}
}

```

MENU PRINCIPAL

```

using UnityEngine;
using System.Collections;
using System.IO;
using UnityEngine.UI;
public class Menu : MonoBehaviour
{
    public void Feto1()
    {
        Interfaz.Escena = "Feto_1";
        Application.LoadLevel("Escena_Base");
    }
    public void Feto2()
    {
        Interfaz.Escena = "Feto_2";
        Application.LoadLevel("Escena_Base");
    }
    public void Regresar()
    {
        Application.LoadLevel("Escena_0");
    }
    public void Credito()
    {
        Application.LoadLevel("Creditos");
    }
    public void CloseApp()
    {
        Application.Quit();
    }
}
}

```

SEGUIMIENTO DE PLANO REAL

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
namespace Kudan.AR.Samples
{
    public class SampleApp : MonoBehaviour
    {
        public KudanTracker _kudanTracker;
        public TrackingMethodMarker _markerTracking;
        public TrackingMethodMarkerless _markerlessTracking;
        public Text buttonText;
        bool SLAM = false;

        public static bool Bandera = false;

        public void MarkerClicked(){
            _kudanTracker.ChangeTrackingMethod(_markerTracking);}

        public void MarkerlessClicked(){
            _kudanTracker.ChangeTrackingMethod(_markerlessTracking);}

        public void StartClicked(){
            if (!SLAM)
            {
                _kudanTracker.ChangeTrackingMethod(_markerlessTracking);
                SLAM = true;
            }
            if (!_kudanTracker.ArbiTrackIsTracking ())
            {
                Vector3 floorPosition;
                Quaternion floorOrientation;
                _kudanTracker.FloorPlaceGetPose (out floorPosition, out
                floorOrientation);
                _kudanTracker.ArbiTrackStart (floorPosition, floorOrientation);
                Bandera = true;
            }
            else {
                Bandera = false;
                _kudanTracker.ArbiTrackStop ();}}
        void Update()
        {
            if (!_kudanTracker.ArbiTrackIsTracking ()) {
                buttonText.text = "Start Markerless Tracking";
            }
            else {
                buttonText.text = "Stop Markerless Tracking";
            }
        }
    }
}
```

REALIDAD AUMENTADA: LOCALIZACIÓN SIMULTÁNEA Y MAPEO (SLAM)

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;
namespace Kudan.AR
{
    [DisallowMultipleComponent]
    [AddComponentMenu("Kudan AR/Tracking Methods/Markerless Tracking")]
    public class TrackingMethodMarkerless : TrackingMethodBase
    {
        public override string Name
        {
            get { return "Markerless"; }
        }
        public override int TrackingMethodId
        {
            get { return 1; }
        }
        public MarkerUpdateEvent _updateMarkerEvent;
        public float _floorDepth = 200.0f;
        public override void ProcessFrame()
        {
            Vector3 position;
            Quaternion orientation;

            _kudanTracker.ArbiTrackGetPose(out position, out orientation);

            Trackable trackable = new Trackable();
            trackable.position = position;
            trackable.orientation = orientation;

            trackable.isDetected = _kudanTracker.ArbiTrackIsTracking();

            _updateMarkerEvent.Invoke(trackable);
        }
        public override void StartTracking()
        {
            _kudanTracker.SetArbiTrackFloorHeight(_floorDepth);

            base.StartTracking();
        }
        public override void StopTracking()
        {
            base.StopTracking();
            Trackable trackable = new Trackable();
            trackable.isDetected = false;

            _updateMarkerEvent.Invoke(trackable);
        }
    }
}

using UnityEngine;
using System.Text;
using System.Collections;

namespace Kudan.AR
```

```

{
    [AddComponentMenu("Kudan AR/Transform Drivers/Markerless Driver")]
    public class MarkerlessTransformDriver : TransformDriverBase
    {
        private TrackingMethodMarkerless _tracker;
        protected override void FindTracker()
        {
            _trackerBase = _tracker =
(TrackingMethodMarkerless)Object.FindObjectOfType(typeof(TrackingMethodMarkerless));
        }
        protected override void Register()
        {
            if (_tracker != null)
            {
                _tracker._updateMarkerEvent.AddListener(OnTrackingUpdate);
            }
            this.gameObject.SetActive(false);
        }
        protected override void Unregister()
        {
            if (_tracker != null)
            {
                _tracker._updateMarkerEvent.RemoveListener(OnTrackingUpdate);
            }
        }
        public void OnTrackingUpdate(Trackable trackable)
        {
            this.transform.localPosition = trackable.position;
            this.transform.localRotation = trackable.orientation;

            this.gameObject.SetActive(trackable.isDetected);
        }
    }
};

```