



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y

MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO

DE INGENIERO EN MECATRÓNICA

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA HÁPTICO

CON REALIDAD VIRTUAL, QUE PERMITA LA MANIPULACIÓN,

MONTAJE Y DESMONTAJE DE PIEZAS DE EQUIPOS

MECATRÓNICOS.

AUTOR: PILATASIG SARABIA, JUAN CARLOS

DIRECTOR: MSC. GORDÓN GARCÉS, ANDRÉS MARCELO

LATACUNGA

2019



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA CARRERA DE
INGENIERÍA MECATRÓNICA.**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA HÁPTICO CON REALIDAD VIRTUAL, QUE PERMITA LA MANIPULACIÓN, MONTAJE Y DESMONTAJE DE PIEZAS DE EQUIPOS MECATRÓNICOS**”, fue realizado por el señor **Pilatasig Sarabia, Juan Carlos** el mismo que fue revisado en su totalidad, analizado por la herramienta de verificación de similitud y contenido; por lo tanto, cumple con los requisitos teóricos, técnicos metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 5 de noviembre del 2019.

Firma:

Una firma manuscrita en tinta azul que dice "ANDRÉS GORDÓN" sobre una línea horizontal.

Ing. Andrés Gordón.

C.C.: 1803698800



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA CARRERA DE
INGENIERÍA MECATRÓNICA.**

AUTORÍA DE RESPONSABILIDAD

Yo, **Pilatasig Sarabia, Juan Carlos** declaro que el contenido, ideas y criterios del trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA HÁPTICO CON REALIDAD VIRTUAL, QUE PERMITA LA MANIPULACIÓN, MONTAJE Y DESMONTAJE DE PIEZAS DE EQUIPOS MECATRÓNICOS”**, es mi autoría y responsabilidad con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionado es veraz.

Latacunga, 5 de noviembre del 2019.

Firma:

Juan Carlos Pilatasig Sarabia.

C.C.:1727643601



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA CARRERA DE
INGENIERÍA MECATRÓNICA.**

AUTORIZACIÓN

Yo, **Pilatasig Sarabia, Juan Carlos** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA HÁPTICO CON REALIDAD VIRTUAL, QUE PERMITA LA MANIPULACIÓN, MONTAJE Y DESMONTAJE DE PIEZAS DE EQUIPOS MECATRÓNICOS”**, en el repositorio institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 5 de noviembre del 2019.

Firma:

Juan Carlos Pilatasig Sarabia.

C.C.:1727643601

DEDICATORIA

Dedico el siguiente trabajo de titulación a todos quienes formaron parte de mi formación profesional guiándome en cada paso de este gran proceso, en especial para mi padre José y mi madre Irma, que gracias a su paciencia y constante apoyo incondicional siempre supieron darme lo mejor. También dedico a mis hermanas y hermanos quienes fueron, son y serán mi fuente de inspiración y motivación al igual que mis profesores y amigos.

Juan Carlos

AGRADECIMIENTO

Agradezco de manera especial a mi madre Irma y a mi padre José por estar siempre pendientes en toda mi formación profesional, saciando todas mis necesidades tanto motivacionales como económicas, al igual que a toda mi familia de corazón que pese a las circunstancias siempre permanecemos unidos.

A todos mis profesores en especial a mi tutor de tesis el Ingeniero Andrés Gordón, quien me supo guiarme durante todo el desarrollo del presente proyecto.

Juan Carlos

ÍNDICE DE CONTENIDO**CARÁTULA**

CERTIFICACIÓN DEL DIRECTOR	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDO	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xvi
RESUMEN	xviii
ABSTRACT	xix

CAPÍTULO I**GENERALIDADES**

1.1. Antecedentes.	1
1.2. Planteamiento del problema.	2
1.3. Justificación.....	3
1.4. Objetivos.	5
1.5. Hipótesis.....	5

1.6. Marco teórico.	6
--------------------------	---

CAPÍTULO II

DISEÑO Y SELECCIÓN DE COMPONENTES

2.1. Metodología de selección de componentes.	16
2.2. Esquematización del Sistema Háptico con Realidad Virtual.	17
2.3. Selección del software.	18
2.4. Selección de hardware.	33
2.5. Sistema de percepción háptica.	45

CAPÍTULO III

IMPLEMENTACIÓN DEL SISTEMA HÁPTICO CON REALIDAD VIRTUAL

3.1. Construcción del Dispositivo Háptico.	63
3.2. Desarrollo de la interfaz.	66
3.3. Integración del dispositivo háptico con el entorno virtual.	113

CAPÍTULO IV

PRUEBAS Y FUNCIONAMIENTO

4.1. Pruebas de funcionamiento del dispositivo háptico.	116
4.2. Pruebas de funcionamiento del sistema del VR.	123
4.3. Pruebas de funcionamiento con estudiantes de ingeniería mecatrónica.	127

4.4. Validación de la hipótesis. 136

4.5. Impacto del Trabajo a corto plazo..... 141

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones. 147

5.2. Recomendaciones..... 149

REFERENCIA BIBLIOGRÁFICA..... 150

ANEXOS 158

ÍNDICE DE FIGURAS

Figura 1. Sistema de percepción háptica.....	6
Figura 2. Realidad Virtual de Wade Sisler.....	9
Figura 3. Control de gestos de mano para ordenador.....	11
Figura 4. Demostración de Técnica Box Modeling.	13
Figura 5. Demostración de Técnica NURBS.	14
Figura 6. Demostración de Técnica CAD Modeling.	14
Figura 7. Entorno virtual de Wrench.....	15
Figura 8. Arquitectura del Sistema Háptico con Realidad Virtual.....	18
Figura 9. Entorno de Desarrollo Blender 3D.	19
Figura 10. Entorno de Desarrollo Autodesk Maya.	21
Figura 11. Entorno de Desarrollo Cinema 4D.	23
Figura 12. Entorno de Desarrollo Unreal Engine4.....	27
Figura 13. Entorno de Desarrollo Unity 3D.....	29
Figura 14. Entorno de Desarrollo Godot Engine.	31
Figura 15. Dispositivo Leap Motion Controller.....	34
Figura 16. Dispositivo electrónico Kinect.	37
Figura 17. Placa Arduino Uno.	41
Figura 18. Placa Electrónica Arduino Nano.	41
Figura 19. Tarjeta de control Arduino Uno.....	43
Figura 20. Diseño de Guantes para el Sistema Háptico.....	47
Figura 21. Dimensionamiento del guante.	49
Figura 22. Diagrama de comunicación del sistema Háptico.....	50
Figura 23. Esquema eléctrico del sistema de control háptico.	50

Figura 24. Algoritmo de control del Sistema háptico.	51
Figura 25. Dimensionamiento para el encapsulado del Arduino.	52
Figura 26. Ensamble de la estructura para el encapsulado del Arduino.	53
Figura 27. Ubicación de la tarjeta de control del sistema háptico.	53
Figura 28. Elemento sujetador.	54
Figura 29. Posición de fuerza a elemento sujetador.....	55
Figura 30. Viga de análisis.....	56
Figura 31. Medida del peso en objeto de análisis mecánico.	57
Figura 32. propiedades físicas de la pieza a ser analizada.	57
Figura 33. Diagramas de fuerzas cortantes y momentos flectores con sus respectivos diagramas de esfuerzos.....	59
Figura 34. Circulo de Mohr para el punto 1.....	60
Figura 35. Circulo de Mohr para el punto 2.....	60
Figura 36. Ensamble total del Sistema de percepción Háptica.	62
Figura 37. Diseño final del par de guantes para el Sistema Háptico.....	63
Figura 38. Parámetros de configuración para Impresión 3D.	64
Figura 39. Disposición de los micro-motores de vibración en el guante.	65
Figura 40. Ensamble Total de Dispositivo Háptico.	65
Figura 41. Estructuración de escenas del Entorno Virtual.....	67
Figura 42. Definición de colores.....	68
Figura 43. División de la rueda de colores.....	69
Figura 44. Armonía de los colores.	70
Figura 45. Creación de paleta de colores.	71
Figura 46. Definición de modelos de color.....	71

Figura 47. Paleta de colores utilizadas en la Escena "Presentación del Proyecto".	72
Figura 48. Aplicación de colores en Unity.	72
Figura 49. Paleta de colores en Escena "Selección de Tipo de Juego".	73
Figura 50. Paleta de colores en Escenas "Montaje y Desmontaje".	73
Figura 51. Escena "Presentación del Proyecto".	74
Figura 52. Conectividad de dispositivos.	75
Figura 53. Código para determinar la conectividad del Leap Motion.	75
Figura 54. Código utilizado para determinar el dispositivo háptico.	76
Figura 55. Menú principal correspondiente al aplicativo.	77
Figura 56. Módulos necesarios para la integración de Leap Motion en Unity 3D.	77
Figura 57. Implementación de los módulos en Unity 3D.	78
Figura 58. Información adicional sobre cada botón del menú principal.	78
Figura 59. Pantalla de visualización de tips para el manejo dentro del aplicativo.	79
Figura 60. Función de control de tips para el aplicativo.	79
Figura 61. Segundo tips, correspondiente a la composición de la escena.	80
Figura 62. Función de cambio de pantalla a información de robot.	80
Figura 63. Pantalla de información de Robot.	81
Figura 64. Modelo 3D para el Botón Montaje.	81
Figura 65. Implementación de los botones en Unity 3D.	82
Figura 66. Restricción de movimiento a cada botón.	82
Figura 67. Configuración de parámetros para el componente InteractionButton.	83
Figura 68. Submenú correspondiente a los videos de demostración de la disposición de las piezas mecánicas del Robot.	83
Figura 69. Submenú de selección de tarea.	84

Figura 70. Escenas principales para las tareas de Montaje y Desmontaje.....	84
Figura 71. Esquema de dibujo de la estructura del brazo robótico.....	85
Figura 72. Modelado mecánico eje J1.....	88
Figura 73. Modelado mecánico eje J2 y J3.....	89
Figura 74. Modelado mecánico eje J4.....	89
Figura 75. Modelado mecánico eje J5.....	90
Figura 76. Modelado de Componentes de Mecanismo para eje J6.....	90
Figura 77. Inicio de modelado de la mano derecha en Blender.....	92
Figura 78. División de la pantalla en sus tres vistas seleccionadas.....	92
Figura 79. Modelado de mano por la metodología Box Modeling.....	93
Figura 80. Aplicación de Modificar " Subdivision Surface " al modelo de mano.....	94
Figura 81. Resultado final del modelado de mano.....	94
Figura 82. Creación de hueso.....	95
Figura 83. Componentes de un Hueso.....	95
Figura 84. Modos de aplicación para el hueso.....	96
Figura 85. Creación de más de un Hueso.....	96
Figura 86. Armadura de Mano.....	97
Figura 87. Asignación de pesos.....	97
Figura 88. Implementación del modelo 3D de antebrazo y mano hacia Unity 3D.....	98
Figura 89. Código utilizado para representar la posición y rotación de las manos del usuario dentro del entorno Unity.....	98
Figura 90. Menú para la tarea de Montaje.....	99
Figura 91. Composición de la escena de Montaje.....	100
Figura 92. Agarre de Objeto con la mano para Montar o Desmontar.....	101

Figura 93. Activación y Desactivación de Objetos para la Aplicación de Montaje y Desmontaje.....	102
Figura 94. Algoritmo para el control del montaje en el aplicativo.	103
Figura 95. Función de control de activación y desactivación de las secuencias de piezas.	104
Figura 96. Modularidad del algoritmo de control para Montaje y Desmontaje.	105
Figura 97. Ejecución del aplicativo montaje.....	105
Figura 98. Estructuración para el nivel 2 de la escena Montaje.....	106
Figura 99. Escena Montaje Parte 1.	106
Figura 100. Selección de Animación del funcionamiento del Robot.....	107
Figura 101. Tipos de Key frames en Blender.	108
Figura 102. Línea KeyFrame de Blender.	109
Figura 103. Entorno animado del funcionamiento del Robot Mitsubishi RV-2SDB.	109
Figura 104. Árbol de elementos hereditarios.	110
Figura 105. Ejecución de Animación del Robot.	111
Figura 106. Finalización de animación de clasificación por colores del Robot.....	111
Figura 107. Menú correspondiente a Montaje y Desmontaje.	112
Figura 108. Disposición de la Escena de Desmontaje.	113
Figura 109. Organización de piezas para el desmontaje.	113
Figura 110. Configuración de Compatibilidad del API para comunicar con el arduino a Unity.....	114
Figura 111. Integración de la clase IO.Ports.	114
Figura 112. Creación del objeto arduino de la clase SerialPort.	115
Figura 113. Definición de puerto y la velocidad de comunicación.....	115
Figura 114. Escritura hacia el arduino para el control del encendido de los motores.....	115

Figura 115. Prueba de funcionamiento del dispositivo Leap Motion Controller.....	116
Figura 116. Pruebas con la mano izquierda de cuatro personas.	117
Figura 117. Curvas de verificación de activación de los micro-motores de sistema háptico.	118
Figura 118. Código de verificación de conexión entre los micro-motores y el arduino.....	118
Figura 119. Entorno para la prueba de funcionamiento del Sistema háptico.....	119
Figura 120. Algoritmo de control para la prueba de funcionamiento del Sistema Háptico.....	120
Figura 121. Asignación del script “Interaction Behavior” al objeto esfera.	121
Figura 122. Inicio de prueba de funcionamiento del Sistema Háptico.	121
Figura 123. Interacción del usuario con el objeto esfera.	122
Figura 124. Interacción del usuario con el objeto esfera.	122
Figura 125. Pruebas de coherencia para el menú principal.....	123
Figura 126. Funcionamiento botón Videos.....	124
Figura 127. Funcionamiento botón Videos.....	124
Figura 128. Funcionamiento botón Salir.....	125
Figura 129. Búsqueda de la pieza a montar.	125
Figura 130. Funcionamiento del avance de la partida para el Montaje.	126
Figura 131. Inicio de partida para desmontaje.	126
Figura 132. Funcionamiento del avance de la partida Desmontaje.	127
Figura 133. Encuesta para los Usuarios del Aplicativo.	128
Figura 134. Matriz de toma de tiempo.	130
Figura 135. Resultados de ficha de evaluación del uso del aplicativo.....	130
Figura 136. Evaluación del Sistema de percepción háptica.	131
Figura 137. Resultados del usuario uno en modo Montaje.....	136
Figura 138. Resultados del usuario dos en modo Montaje.	137

Figura 139. Resultados del usuario tres en modo Montaje.	137
Figura 140. Resultados del usuario cuatro en modo Montaje.	138
Figura 141. Resultados del Usuario 1 en modo Desmontaje.	139
Figura 142. Resultados del Usuario 2 en modo Desmontaje.	139
Figura 143. Resultados del Usuario 3 en modo Desmontaje.	140
Figura 144. Resultados del Usuario 4 en modo Desmontaje.	141
Figura 145. Beneficios para la materia de Bases Fundamentales de la Ingeniería Mecatrónica.	143
Figura 146. Beneficios en la materia de Mecanismos.	143
Figura 147. Beneficios en la materia de Robótica Industrial.	144
Figura 148. Elementos básicos de un robot antropomórfico.	144
Figura 149. Mecanismo de transmisión de movimiento.	145

ÍNDICE DE TABLAS.

Tabla 1. <i>Tipos de Dispositivos de retroalimentación háptica.</i>	7
Tabla 2. <i>Matriz de selección por la Metodología por puntos.</i>	17
Tabla 3. <i>Requerimientos de Instalación de Blender 3D.</i>	20
Tabla 4. <i>Requerimientos para instalación de Autodesk Maya.</i>	22
Tabla 5. <i>Requerimientos para instalación de Cinema 4D.</i>	24
Tabla 6. <i>Matriz de Selección de Software para modelado 3D.</i>	25
Tabla 7. <i>Requerimientos de instalación para UnrealEngine4.</i>	27
Tabla 8. <i>Requerimientos de Instalación para Unity 3D.</i>	29
Tabla 9. <i>Requerimientos de instalación para Godot Engine.</i>	31
Tabla 10. <i>Matriz de selección para el software del motor de juegos.</i>	32
Tabla 11. <i>Requerimientos de instalación para Leap Motion.</i>	34
Tabla 12. <i>Características técnicas de Leap Motion.</i>	35
Tabla 13. <i>Requerimientos de instalación para Kinect.</i>	37
Tabla 14. <i>Características técnicas de Kinect.</i>	37
Tabla 15. <i>Matriz de selección para el Hardware controlador de gestos.</i>	39
Tabla 16. <i>Características Técnicas Arduino Nano.</i>	41
Tabla 17. <i>Características técnicas Arduino Uno.</i>	43
Tabla 18. <i>Matriz de selección de tarjeta de control electrónico.</i>	45
Tabla 19. <i>Nivel de complejidad que define a los dispositivos Hápticos.</i>	46
Tabla 20. <i>Resultados de las dimensiones de las manos en hombre y mujeres.</i>	48
Tabla 21. <i>Propiedades mecánicas del material PLA.</i>	54
Tabla 22. <i>Disposición y descripción de tornillos del Robot.</i>	87
Tabla 23. <i>Resultados del usuario uno en modo Montaje.</i>	132

Tabla 24. <i>Resultados del usuario dos en modo Montaje.</i>	133
Tabla 25. <i>Resultados del usuario tres en modo Montaje.</i>	133
Tabla 26. <i>Resultados del usuario cuatro en modo Montaje.</i>	133
Tabla 27. <i>Resultados del usuario uno en modo Desmontaje.</i>	134
Tabla 28. <i>Resultados del usuario dos en modo Desmontaje.</i>	134
Tabla 29. <i>Resultados del usuario tres en modo Desmontaje.</i>	135
Tabla 30. <i>Resultados del usuario cuatro en modo Desmontaje.</i>	135

RESUMEN

El presente proyecto de titulación fue desarrollado dentro del motor de video juegos Unity 3D en donde se implementó el dispositivo electrónico de reconocimiento gestual de los movimientos de las manos Leap Motion Controller, mismo que permite tener un control interactivo del usuario con la aplicación de Realidad Virtual, además de un dispositivo de percepción háptica que permita generar una retroalimentación al sistema táctil. El Sistema de Realidad Virtual da a conocer la parte de transmisión de movimiento de un Brazo Robótico de tipo antropomórfico permitiendo fortalecer el conocimiento a estudiantes de ingeniería mecatrónica en la materia de mecanismos y robótica industrial. Para empezar, se realizó una recopilación de información conceptual acerca de las tecnologías utilizadas dentro de entornos de Realidad Virtual, su alcance y limitaciones. Seguido a eso se realizó una selección de la parte del software y del hardware. Luego de ello se realizó el diseño y construcción del dispositivo de percepción háptica tomando como referencia la normativa ISO TC159/SC4/WG9. Se realizó la representación computacional tridimensional tanto del Robot como de la parte de la extremidad superior implementándolo dentro de Unity 3D que junto con la programación orientada a objetos en C# (C Sharp) y paquetes de Leap Motion permitieron el desarrollo del aplicativo. Finalmente se realizaron las pruebas de funcionamiento y se determinó el efecto que genera el sistema hacia los usuarios.

PALABRAS CLAVE:

- **CONTROLADOR LEAP MOTION.**
- **PERCEPCIÓN HÁPTICA.**
- **REALIDAD VIRTUAL.**
- **MODELADO 3D.**
- **ROBÓTICA.**

ABSTRACT

The present work project was developed within the Unity 3D game engine where the electronic gesture recognition device Leap Motion Controller was implemented, which allows the system to be intuitive, in addition to a haptic perception device that allows to generate a tactile feedback. The System was developed to strengthen learning within professional training in Mechatronics engineering, making known the mechanical movement transmission part of a Robotic Arm. To begin with, a collection of conceptual information about technologies used within Virtual Reality environments, their scope and limitations was made. Following that, a selection was made of both the software and hardware part. Then the design and construction of the haptic perception device was carried out taking as reference the ISO TC159 / SC4 / WG9. The three-dimensional computational representation of both the Robot and the part of the upper limb (hand and forearm) was implemented within the Unity 3D video game engine that together with the object-oriented programming in C # (C Sharp) and Leap Motion packages allowed the development of the application. Finally, the functional tests were carried out and the effect generated by the system on the users was determined.

KEYWORDS:

- **LEAP MOTION CONTROLLER.**
- **HAPTIC PERCEPTION.**
- **VIRTUAL REALITY.**
- **3D MODELING.**
- **ROBOTIC.**

CAPÍTULO I

GENERALIDADES

1.1. Antecedentes.

El sistema educativo está cambiando gracias al avance de la tecnología y la demanda de estudiantes que exigen una formación más flexible, multi-soporte y personalizada. La primera era del sistema educativo data hace 2000 A.C, en Sumeria, en Mesopotamia, en donde apareció por primera vez el término “Escuela”, enseñaban escritura y conocimientos básicos que ayudaban a solventar las necesidades del día a día, siendo esta la primera era educativa conocida como analógica que duró miles de años y aunque quedó obsoleta aún forma parte del sistema actual; la segunda era educativa fue la digital que no es otra cosa que la virtualización de la análoga. Finalmente aparece la educación 3.0 que tiene que ver con la educación interactiva valiéndose de sistemas inteligentes para ayudar a un aprendizaje más rápido, efectivo y relevante para las personas/empresas (González, 2016).

La realidad virtual dentro de la educación está siendo la tecnología de gran importancia, ya que permite estimular el proceso de aprendizaje en estudiantes y puede cambiar el modelo educativo actual. Varios fabricantes a nivel mundial están poniendo todo su esfuerzo en este campo, siendo uno de ellos Samsung con el Virtual School Suitcase, este es un kit para docente que incluye gafas dedicadas para la realidad virtual y una serie de contenido para cursos de secundaria. Por otro lado, la realidad virtual en la educación abre un mundo de posibilidades sin movernos del aula o laboratorio y poder utilizarlo en todas las materias existentes de la etapa educativa (Educación 3.0, 2016).

Un aporte de la realidad virtual en el Ecuador fue la que se realizó en el recinto del Habitat III, el Ministerio de Turismo ejecutó la actividad “Ecuador en 60 segundos”, que permite conocer las cuatro regiones del Ecuador por medio de la experiencia de la realidad virtual, donde los usuarios se colocan las gafas específicas para realidad virtual y audífonos; y automáticamente se ubica en un bosque lleno de árboles y cantos de aves (ELCOMERCIO, 2016).

La interacción de un individuo con un entorno de realidad virtual se puede lograr por medio de dispositivos hápticos y sensores de captura de movimiento corporal; permitiendo al individuo que reconozca esta realidad alternativa como válida y responda a ella como si fuera real (Giro Rodolfo, 2017). La háptica hace referencia al estudio del tacto en el ser humano. Los dispositivos hápticos son dispositivos electrónicos que permiten simular características como peso, fuerza, textura, etc, además estos dispositivos permiten trasladar dichas características al individuo permitiendo sentir, tocar y manipular objetos tridimensionales dentro de un entorno de realidad virtual (QUEVEDO, 2016). Los sensores de captura de movimiento ofrecen al individuo una mayor inmersión en aplicaciones de realidad virtual, ya que permite la interacción de forma directa con los objetos virtuales tridimensionales.

1.2.Planteamiento del problema.

Actualmente la tecnología es una pieza fundamental para nuestra vida diaria, y cada vez se están desarrollando herramientas más sofisticadas como simuladores hápticos de realidad virtual para manipulación de objetos que de forma directa fortalece el aprendizaje en estudiantes. Uno de los problemas que afronta el estudiante es la forma en cómo se prepara como futuro profesional, siendo esta intangible y abstracta, esto hace que no puedan responder de forma correcta a las exigencias de la nueva sociedad tecnológica.

El Ecuador es un país que está evolucionando rápidamente en la parte tecnológica, pero después de realizar una búsqueda en los principales repositorios de las Universidades del país se ha determinado que se han realizado pocos proyectos que involucren la realimentación háptica en el uso de la realidad virtual dentro de instituciones educativas como objeto de apoyo en el proceso de aprendizaje, siendo esta una herramienta muy potente para simular situaciones en diversos campos del mundo real con característica inmersiva.

La tecnología háptica proporciona varias ventajas al utilizarla junto con la realidad virtual, proporcionando al usuario una mayor inmersión, concentración y atención prestada que da como resultado una mejora en el proceso de aprendizaje de una actividad (Acosta, 2011). Por lo que, un sistema de realidad virtual sin retroalimentación háptica no permitirá al usuario manipular los objetos tridimensionales de forma natural como si formara parte de ese nuevo mundo.

En el Laboratorio de Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, existen varios equipos mecatrónicos como es el robot antropomórfico Mitsubishi, entre otros, destinados a la formación profesional, pudiendo ser dicho equipo manipulado pero no desmontado y montado para conocer y aprender su estructura interna ya que dicho equipo es de alto costo, por lo que la solución a este problema es la creación de un entorno virtual capaz de realizar dichas tareas antes mencionadas, al no existir un sistema háptico con realidad virtual que permita la manipulación, montaje y desmontaje de estos equipos mecatrónicos, los estudiantes no podrán experimentar mientras aprenden, no podrán conocer de forma casi real partes mecánicas de dicho equipo.

1.3. Justificación.

La demanda de estudiantes que exigen una formación universitaria más flexible, multisupoite y personalizada; da lugar al uso de entornos de realidad virtual para mejorar el sistema

educativo en el Ecuador. Un entorno virtual de enseñanza permitirá mejorar la toma de decisiones rápidas e inteligentes simuladas en tiempo real, esto ayudará al estudiante a evaluar su aprendizaje a partir de las consecuencias de sus acciones y reflexiones, además de mejorar en su formación profesional, capacitándoles con las necesidades reales de las empresas del siglo XXI.

La importancia de promover una herramienta háptica con realidad virtual, brindará a los estudiantes como futuros profesionales mayor confianza en el ámbito laboral, haciendo hincapié en conocer la ubicación de las partes de máquinas, su funcionalidad y la puesta en marcha.

En el Ecuador se han realizado proyectos universitarios hápticos con realidad virtual en el área médica en la parte de la rehabilitación de algún miembro, o en la parte de manipulación física de materiales peligroso, pero no existe simuladores hápticos con realidad virtual enfocados en el aprendizaje de piezas de máquinas y su ensamble, la mayoría de los centros de formación superior aún están en la era digital (educación 2.0), por lo que el desarrollo del presente proyecto se considera de suma importancia en el proceso de aprendizaje como una oportunidad para que las personas practiquen y se equivoquen sin necesidad de riesgo para sí o para otros, con la posibilidad de evaluar su desempeño y evolución (Giro Rodolfo, 2017).

Para el Laboratorio de Mecatrónica de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga, tener una herramienta háptica con realidad virtual para la manipulación, montaje y desmontaje de piezas de equipos mecatrónicos (robot antropomórfico Mitsubishi), ayudará a los estudiantes a poner en práctica su destreza; permitiendo desempeñarse mejor en la vida profesional, enfrentando eficientemente la toma de decisiones y mejorando los resultados en las industrias.

1.4.Objetivos.

1.4.1. Objetivo General.

Diseñar e implementar un sistema háptico con realidad virtual, que permita la manipulación, montaje y desmontaje de equipos mecatrónicos.

1.4.2. Objetivos Específicos.

- Investigar acerca de las técnicas de simulación y detección de objetos para visualizar el miembro superior (mano y antebrazo).
- Analizar y seleccionar componentes y materiales que cumplen con los requisitos para la finalidad del proyecto.
- Modelar las piezas de equipos de máquinas mecatrónicas y del miembro superior (mano y antebrazo) con el uso de software de modelado 3D.
- Diseñar el dispositivo háptico mediante el uso de un software de diseño asistido por computador.
- Construir el dispositivo háptico antes diseñado que permita colocar elementos adicionales utilizando materiales existentes en el mercado.
- Desarrollar algoritmos de programación para la interacción del usuario con el entorno virtual.
- Validar la hipótesis mediante el análisis de resultados de las pruebas de funcionamiento realizadas.

1.5.Hipótesis

¿El diseño del sistema háptico con realidad virtual, permitirá mejorar la habilidad de montar y desmontar piezas de equipos mecatrónicos?

1.6.Marco teórico.

1.6.1. Sistema de percepción háptica.

La terminología Háptica proviene del griego “Haptesthai” que significa “tocar”. Por lo tanto, la háptica es la ciencia que se encarga del estudio del sentido del tacto de manera alusiva, misma que proporciona información propioceptiva (Kinestésica), táctil y vestibular, cada una de ellas necesarias a ciertos casos (Raya, 2019). En la figura 1 muestra un aplicativo de la tecnología háptica utilizado junto a un entorno de Realidad Virtual.

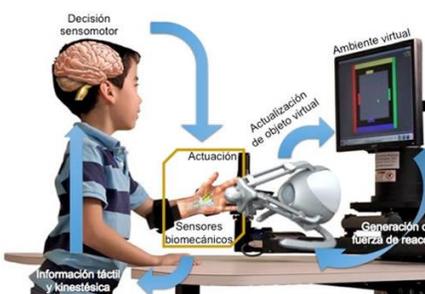


Figura 1. Sistema de percepción háptica.

Fuente: (Gidiño, 2019)

Sistema propioceptivo.

Sistema por el cual, el cerebro recibe la información de posición relativa de las partes corporales del cuerpo humano al igual que su movimiento (Miñambres, 2017).

Sistema Táctil.

Permite la interacción con las terminaciones nerviosas de la piel generando sensaciones por medio de calor, presión, vibración, dolor y textura. El sentido del tacto se encuentra en la piel, mismo que posee receptores nerviosos que permiten transmitir información susceptible de ser interpretados por el cerebro (Raya, 2019).

Sistema Vestibular.

Alusivo a la percepción de la posición de la cabeza su aceleración y deceleración (Raya, 2019).

Tipos de sistemas de percepción háptica.

Los sistemas de percepción háptica son dispositivo que permiten generar sensaciones hacia el usuario ya sea de fuerza o táctil, dependiendo si el usuario toca un tipo de superficie u otra, si colisiona un objeto tridimensional o si lo manipula, todo esto dentro de un entorno de Realidad Virtual. Definen su comportamiento como un pequeño robot que intercambian la energía mecánica con el usuario. Actualmente la tecnología y la robótica ofrecen a los usuarios multitud de dispositivos destinados para el ocio, entrenamiento, investigación y la industria, capaces de simular características casi reales de las sensaciones hápticas (Propioceptivas, táctiles y vestibulares) (Raya, 2019).

Tabla 1.

Tipos de Dispositivos de retroalimentación háptica.

Sistema Propioceptivo	Sistema táctil	Sistema Vestibular
Para proporcionar información de la posición relativa de las partes corporales y su respectiva orientación, existen dispositivos que poseen retroalimentación de fuerza que devuelve al usuario la	Para determinar información como temperatura presión o textura, existen dispositivos que poseen retroalimentación de fuerza y emplean normalmente vibradores electromecánicos para determinar datos de	Dispositivos de mayor complejidad, coste y tamaño. Se utilizan en simuladores de vuelo o automoción. Como ejemplo se tiene la plataforma <i>Stewart</i> , se basa en un manipulador paralelo de seis

CONTINÚA



sensación de colisionar o tocar objetos tridimensionales sean deformables o no. Como ejemplo tenemos a *Phantom Omni* de Sensable (Anónimo, Immersion, 2019).



El dispositivo ofrece tres grados de libertad pero seis grados de libertad en orientación.

textura o rugosidad. Son menos complejo su diseño, y baratos su construcción. Como ejemplo se tiene el Guante *CyberTouch Immersion CO* (CyberGlove, 2017).



El guante es un sistema de retroalimentación táctil, mismo que posee 6 pequeños vibro táctiles, en cada uno dedo y en la palma.

actuadores unidos a pares en tres posiciones a la placa base. También conocida como plataforma de seis grados de libertad (Anónimo, Wikipedia, 2019).



En realidad virtual, estos sistemas engañan al usuario jugando con la forma de percibir la aceleración o la inercia del sistema vestibular.

Fuente: (Raya, 2019)

1.6.2. Realidad virtual.

En términos técnicos la realidad virtual es usada para describir un ambiente tridimensional generado por un computador, en donde una persona puede explorar e interactuar con el entorno virtual como se aprecia en la figura 2 en donde muestra el uso de periféricos como son guantes para la captura de movimiento del usuario y gafas de realidad virtual, además que dicha persona

llega a formar parte de ese mundo virtual o estar inmersa, y mientras este allí, puede manipular objetos o realizar una serie de acciones (TheFreshUK, 2017).



Figura 2. Realidad Virtual de Wade Sisler.

Fuente: (Centro de Investigación Ames de la NASA, 2017)

Características de los sistemas de realidad virtual.

Existe una gran variedad de sistemas de realidad virtual, pero todos comparten las mismas características, a continuación, se describe las características más relevantes (TheFreshUK, 2017):

- Capacidad de permitir a personas ver imágenes tridimensionales.
- Las imágenes tridimensionales son de tamaño natural para la persona.
- Proporcionan un cambio en su campo de visión a medida que se mueve alrededor del entorno.
- Debe proporcionar las respuestas apropiadas, en tiempo real, a medida que explora el entorno.

Tipos de sistemas de realidad virtual.

La clasificación de los sistemas de realidad virtual se da según el nivel de inmersión, algunos de los cuales son (Covadonga Díez Sanmartín, 2018):

Inmersivo.

Este tipo de sistema se considera inmersión total cuando la persona se siente que accede a un mundo diferente.

Semi-inmersiva.

En este sistema virtual la persona se convierte parte del entorno, pero no está completamente sumergido.

Colaborativa.

En este tipo de sistema de realidad virtual cooperativo varias personas interactúan entre sí y puedan compartir ideas y experiencias.

Realidad mixta.

Es la combinación entre imágenes tomadas del entorno real con elementos virtuales o provenientes de otros contextos, es conocida también como realidad aumentada, a menudo es capaz de identificar ciertos aspectos de la realidad, como rostros, objetos 3D entre otros.

Aplicaciones de la realidad virtual.

Las aplicaciones de la realidad virtual actualmente en actividades cotidianas son muchas y diversas, a continuación, algunas de ellas:

- Educación.
- Formación o entrenamiento.
- Psicoterapia.
- Medicina.
- Entretenimiento.
- Videojuegos.
- Software.

1.6.3. Controlador de gestos de manos óptico.

Un controlador de gestos de manos óptico, es aquel dispositivo electrónico que permite realizar la tarea de seguimiento de la variación de movimiento de las manos y dedos en tiempo real como una secuencia de coordenadas cartesianas en el espacio tridimensional (Frank Weichert, 2013) tal como se puede apreciar en la figura 3 en donde dicha variación de movimiento puede ser representada en un pantalla.

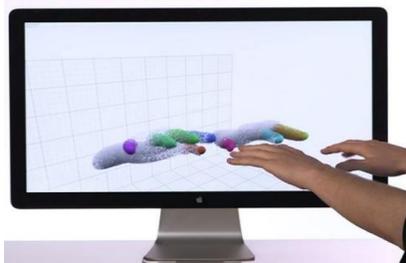


Figura 3. Control de gestos de mano para ordenador.

Fuente: (Infobae, 2018)

El principio de operativo de los sensores de captura de movimiento ópticos se divide en los siguientes mecanismos:

- Tiempo de vuelo.
- Visión Estereoscópica.

Captura de movimiento por tiempo de vuelo (Time of Flight TOF).

La captura de movimiento por tiempo de vuelo permite conocer la distancia entre el objeto y el sensor, el sensor posee varios leds infrarrojos, mismos que emiten luz infrarroja modulada, esta luz es reflejada por el objeto, la distancia es determinada por el tiempo en que la luz reflejada regrese al sensor (cámaras) (Ruiz-Sarmiento, 2011).

Captura de movimiento por Visión Estereoscópica.

Para la captura de movimiento por visión estereoscópica, se utiliza al menos dos cámaras de 2D, cuyos parámetros intrínsecos son conocidos como la distancia entre la dos cámaras y el

ángulo formado por los ejes focales de los mismos, cuando se enfoca al objeto destino (ICGEO, 2015).

Los sistemas de seguimiento de gestos y posición ópticos utilizan los datos como una nube de puntos de n – dimensiones, dicha nube de puntos es interpretada por medio de algoritmos matemáticos que los acondicionan para poder utilizarlo aplicaciones ya sea de Realidad Virtual o Aumentada (Frank Weichert, 2013).

1.6.4. Modelado 3d.

El modelado 3D, también conocida como computación gráfica, permite crear objetos tridimensionales dentro de un ordenador con la ayuda de software específico. Dentro de la Realidad Virtual el modelado 3D permite crear contenido cual imaginación tenga su desarrollador (Javir007, 2018).

Técnicas de modelado 3D.

Dentro del mundo 3D, existen varias técnicas que permiten la creación de modelos 3D, cada una de las cuales poseen características útiles que dependen de la finalidad del modelo. Una de las consideraciones a tener en cuenta es que estas técnicas no son estáticas, están en constante evolución, además, muchas de ellas desaparecen con el tiempo ya que simple y sencillamente están destinadas para un uso específico mientras que otras permanecen muy utilizadas ya que dan mayor solvencia al desarrollador (Javir007, 2018).

Modelado por caja.

Esta técnica permite la creación de modelo final a partir de modelos primitivos (Una caja, cilindro o una esfera), dando forma al modelo utilizando herramientas como extracción, escalado o por rotación de caras y aristas tal como se aprecia en la figura 4. Los detalles son añadidos

manualmente al modelo primitivo, ya sea agregando bucles de aristas o subdividiendo polígonos, este proceso se lo realiza de manera repetitiva hasta llegar al producto final. Una de las ventajas de esta técnica es la creación de modelos más eficientes y controlados (Javir007, 2018).

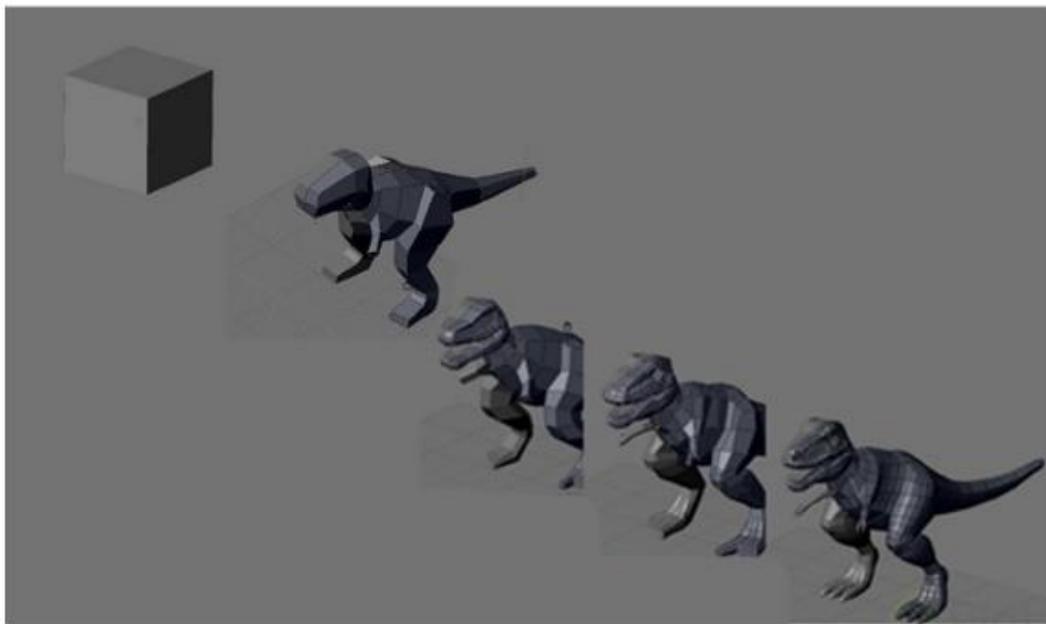


Figura 4. Demostración de Técnica Box Modeling.

Fuente: (Javir007, 2018)

Modelado por NURBS.

B-Splines racionales no uniformes o NURBS (Non Uniform Rational B-Spline), es una técnica muy utilizada en la computación gráfica ya que permite generar y representar curvas y superficies con la ayuda de un modelo matemático. La representación de una superficie dentro de un espacio tridimensional con NURBS como se observa en la figura 5, es a través de funciones de dos parámetros de asignación y la forma de la misma está determinada por puntos de control. Estos puntos de control van conectados de forma directamente a la curva o actúan como lo estuviese (no interpola) (EcuRed, 2019).

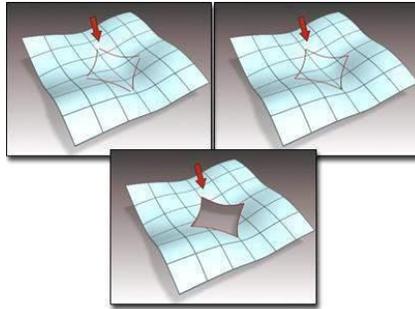


Figura 5. Demostración de Técnica NURBS.

Fuente: (EcuRed, 2019)

Modelado por CAD.

Dibujo Asistido por Computador o CAD (Computer Aided Desing), es la técnica utilizada para la representación gráfica bidimensional o tridimensional de objetos físicos. Este tipo de técnica es utilizado generalmente en la ingeniería, industria y la arquitectura tal como se aprecia en la figura 6 como objeto de la generación de prototipos y productos finales, partiendo desde su diseño conceptual y estructural, su correspondiente análisis de ensamblaje hasta la propuesta del método de análisis de manufactura. Además, el CAD permite crear variantes con el objetivo de obtener un producto de calidad, reduciendo costes en el desarrollo, ganando velocidad y mejorando la productividad (Siemens, 2019).



Figura 6. Demostración de Técnica CAD Modeling.

Fuente: (Siemens, 2019)

1.6.5. Educación 3.0.

La incorporación de dispositivos digitales dentro del sistema educativo, permite mejorar la educación de forma tan radical. En (Infobae, 2018) describe que para aprovechar la gran potencialidad que posee la tecnología, es de suma importancia que tanto los alumnos como los profesores adquieran habilidades totalmente nuevas e innovadoras de aprender y ya no solo de forma orgánica. La Realidad Virtual permite disfrutar mundos infinitos creados desde cero, adaptados a un entorno real o siendo totalmente ficticio. Además, una de las claves más importantes de la Realidad Virtual está en la educación, ya que nos permite viajar a mundos diferentes sin la necesidad de salir del salón, pero cabe recalcar que solo es una herramienta complementaria pero no sustitutivo al maestro quien es el pilar fundamental de la educación (Elesapiens, 2017).

1.6.6. Estado del arte.

Sistema de realidad virtual para la manipulación y ensamble de máquinas.

Wrench es un juego de realidad virtual, este tipo de sistema permite realizar la tarea de ensamble de piezas de vehículos de competencia como se aprecia en la figura 7. Cada jugador es responsable de verificar el estado de los vehículos que se lo fueron encargados, puede realizar tareas simples como el cambio de fluidos o complejas como el reemplazo de cojinetes principales del motor.



Figura 7. Entorno virtual de Wrench.

Fuente: (Alec Moody, 2018)

CAPÍTULO II

DISEÑO Y SELECCIÓN DE COMPONENTES.

Par el desarrollo del Sistema de Percepción Háptica con Realidad Virtual destinado para realizar tareas de Manipulación, Montaje y Desmontaje de Piezas de Equipos Mecatrónicos, es necesario la selección de componentes tanto la parte de software como la de hardware de manera adecuada, tomando en consideración varias alternativas con el objetivo de optimizar el sistema a desarrollar.

En primera instancia se realiza la selección del software destinado para el modelado 3D, a cada alternativa se le dará su respectiva caracterización correspondiente a su instalación, ventajas y desventajas; se propondrá criterios relevantes de selección los cuales será ponderados utilizando la metodología por puntos para determinar la mejor alternativa de forma objetiva. Luego se realiza la selección del software destinado para el desarrollo de Entorno de Realidad Virtual, que de la misma manera se le dará su respectiva caracterización correspondiente a su instalación, ventajas y desventajas; se utilizó la metodología por puntos para la selección. Finalmente se realiza la selección del dispositivo controlador de gestos de las manos y la tarjeta electrónica de control para el dispositivo de percepción háptica que de la misma manera se determinó su respectiva caracterización correspondiente a su instalación, ventajas y desventajas; y con la ayuda de la metodología por puntos se determinó la mejor alternativa.

2.1. Metodología de selección de componentes.

Para determinar la mejor alternativa, se hizo uso de la metodología cualitativa por puntos, que consiste en ponderar factores de selección que son relevantes para el desarrollo del proyecto en sí. Esta herramienta permite comparar y jerarquizar de forma cuantitativa las alternativas existentes y así poder tomar la mejor decisión de selección (Urbina, 2013). La metodología consiste en:

- Determinar una lista de factores relevantes.
- Asignar un peso a cada factor para indicar su importancia relativa (los pesos deben sumar 1 o 100%), y el peso asignado dependerá exclusivamente del criterio del investigador.
- Asignar una escala común a cada factor (para este caso de 0 a 5) y elegir cualquier mínimo.
- Calificar a cada sitio potencial de acuerdo con la escala designada y multiplicar la calificación por el peso.
- Sumar la puntuación de cada sitio y elegir el de máxima puntuación.

En la tabla 2 se puede observar un ejemplo de una matriz que se utilizada en esta metodología, para selección de componentes de entre varias alternativas de manera objetiva cuantitativa.

Tabla 2.
Matriz de selección por la Metodología por puntos.

	Alternativa 1		Alternativa 2		
Factor	Peso(%)	Calificación	Calificación	Calificación	Calificación
Relevante			ponderada		ponderada
Factor 1.	50%	5	2.5	5	2.5
Factor 2.	50%	5	2.5	4	2
Total	100%		5		4.5

Fuente: (Urbina, 2013)

2.2.Esquemización del Sistema Háptico con Realidad Virtual.

En la figura 8 se muestra el esquema que define al Sistema Háptico con Realidad Virtual.



Figura 8. Arquitectura del Sistema Háptico con Realidad Virtual.

El sistema permite el reconocimiento de movimiento y gestos de las manos por parte de algún dispositivo electrónico, la información tomada se enviará hacia el computador, dentro del computador con la ayuda de un software que permite el desarrollo de entorno virtuales se realizará la correspondiente representación gráfica digital de las manos. Dentro del entorno de realidad virtual el usuario podrá interactuar con los objetos, realizar tareas de montaje y desmontaje; además, existirá retroalimentación táctil hacia el usuario cuando este toque algún objeto con la ayuda del dispositivo de percepción háptica.

2.3. Selección del software

2.3.1. Software de modelado.

Alternativas de software de Modelado 3D.

Para el desarrollo de la aplicación de Realidad Virtual para el Montaje y Desmontaje de Piezas de Equipos Mecatrónicos, es necesario la creación de modelos en 3D mismo que serán destinados a la visualización y manipulación dentro del entorno virtual por parte del usuario, por lo que dentro de la Red de Internet se puede encontrar varios programas destinados al modelado 3D, pero se han considerado los siguientes:

a. Blender.

Blender es un software que está destinado a crear visualizaciones en 3D, como imágenes fijas, animaciones 3D, toma de VFX y edición de video. Es un programa multiplataforma y adecuado para pequeños proyectos que se benefician de su flujo de trabajo unificado y su activo proceso de desarrollo (Blender, 2019). En la figura 9 se puede apreciar una vista previa dentro del entorno de desarrollo de Blender 3D.



Figura 9. Entorno de Desarrollo Blender 3D.

Fuente: (Blender, 2019)

Blender 3D tiene una gran variedad de herramientas que permiten desarrollar casi cualquier tipo de producción de medios. Personas de todo el mundo lo utilizan como objeto de hobby, comerciales y película (Blender, 2019).

Características clave

- Es una suite de creación de contenido 3D totalmente integrada, que ofrece una amplia gama de herramientas esenciales, incluyendo Modelado, Renderizado, Animación y Rigging, Edición de Video, VFX, Composición, Texturizado, y algunos tipos de Simulaciones (Blender, 2019).

- Es multiplataforma, con una OpenGL GUI que es uniforme en todas las plataformas principales (y personalizable con scripts de Python) (Blender, 2019).
- Tiene una arquitectura 3D de alta calidad, lo que permite un flujo de trabajo de creación rápida y eficiente (Blender, 2019).

Los requerimientos de instalación de blender 3D se muestra a continuación en la tabla 3.

Tabla 3.

Requerimientos de Instalación de Blender 3D.

Requerimientos Recomendados	
CPU	2 Ghz
RAM	8 Ghz
Tipo de Sistema	64 bits quad core CPU
Tarjeta Gráfica	Gráficos compatibles con OpenGL 3.2 con 2 GB de RAM.

Fuente: (BlenderES, 2017)

Ventajas y Desventajas de la utilización de Blender 3D.

Ventajas.

- Software libre y de código abierto.
- Multiplataforma, compatible con Linux, Windows, Mac OS X, Solaris, IRIX y FreeBSD.
- Gran cantidad de tutoriales y una comunidad dedicada en línea.

Desventajas.

- No utiliza todos los procesadores del computador por defecto.
- Interfaz complicada debido a que es un software muy potente.

- No funciona correctamente con polígonos con más de 4 lados.

b. Autodesk Maya.

Autodesk Maya es un potente software que está destinado al desarrollo de gráficos 3D, efectos especiales y animación (Anónimo, EcuRed, 2019). En la figura 10 se puede apreciar una vista previa del entorno de desarrollo de Autodesk Maya.



Figura 10. Entorno de Desarrollo Autodesk Maya.

Fuente: (Anónimo, EcuRed, 2019)

Autodesk Maya también conocido como Maya 3D, es un programa para modelar, animar y renderizar objetos tridimensionales muy completos. Se caracteriza por su potencia y su posibilidad de expansión y personalización de su interfaz gráfica y las avanzadas herramientas de diseño y edición, esto lo convierte en uno de los programas más utilizados a nivel profesional tanto para la producción de contenido televisivo, cine y videojuegos (Anónimo, EcuRed, 2019).

Los requerimientos de instalación de Autodesk Maya se muestran a continuación en la siguiente tabla 4.

Tabla 4.
Requerimientos para instalación de Autodesk Maya.

Requerimientos Necesarios	
CPU	Procesador de varios núcleos de 64 bits Intel® o AMD® con el conjunto de instrucciones SSE4.2.
RAM	Mínimo 8GB (16GB recomendada)
Espacio en el Disco	4GB de espacio disponible
Tarjeta Grafica	Tarjeta gráfica de 128 MB con soporte Direct3D 9, OpenGL o Direct3D 10.

Fuente: (Anónimo, EcuRed, 2019).

Ventajas y Desventajas de la utilización de Autodesk Maya.

Ventajas.

- Licencia de paga y estudiantil.
- Sistema operativo Microsoft Windows, GNU/Linux, IRIX (descontinuado por Alias), y Mac OS X.
- Interfaz intuitiva

Desventajas.

- Controladores más complejos pero completos.
- No hay entendimiento con las versiones anteriores (tal vez podría suponer un problema).
- Tiempo de aprendizaje lento.

c. Cinema 4D.

Cinema 4D, es un software destinado para la creación de modelos tridimensionales y animación. Fue desarrollada por la compañía Alemania Maxon. En la figura 11 se puede apreciar una vista previa del entorno de desarrollo de Cinema 4D.

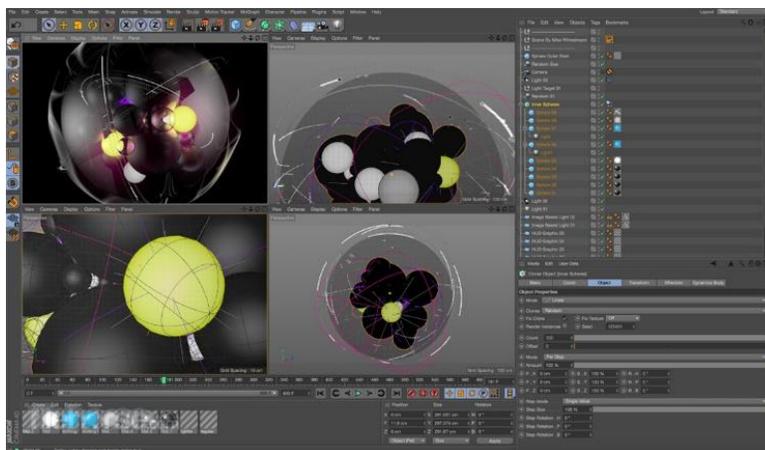


Figura 11. Entorno de Desarrollo Cinema 4D.

Fuente: (Anónimo, EcuRed, 2019).

El software dispone de un potente modelador de objetos tridimensionales al igual que una gran gama de funciones y efectos especiales, esto con el objetivo de ayudar en la representación de proyectos arquitectónicos e ingenieriles. Es una herramienta muy intuitiva debido a que posee una velocidad de renderizado incomparable, lo que denota ser un software de alto rendimiento, permitiendo crear un diseño 3D a cualquier nivel, animaciones fotorrealistas, entornos virtuales y efectos especiales para el cine y la televisión (Anónimo, EcuRed, 2019).

Los requerimientos de instalación de Cinema 4D se muestra a continuación en la siguiente tabla 5.

Tabla 5.
Requerimientos para instalación de Cinema 4D.

Requerimientos Necesarios	
CPU	Procesador de 64 bits a partir de la versión r14 Intel o AMD con soporte para SSE3.
RAM	Mínimo 4GB (8GB recomendada)
Espacio en el Disco	7GB de espacio disponible
Tarjeta Grafica	Tarjeta gráfica NVIDIA or AMD para Windows o AMD en MacOS, soporta OpenCL 1.2. Mínimo 4 GB VRAM para GPU rendering

Fuente: (Anónimo, EcuRed, 2019).

Ventajas y desventajas que presenta Cinema 4D en su utilización.

Ventajas.

- Interfaz intuitiva.
- Sistema operativo Windows y Macintosh (OS 9 y OS X).
- Tiempo de aprendizaje fácil.

Desventajas.

- Licencia de paga.
- Manual de usuario disponible en versiones de pago.

Selección de software para modelado 3D.

Ahora, para determinar el software adecuado para el desarrollo de diseño en 3D se tiene los siguientes criterios:

- Instalación.
- Multiplataforma.
- Licencia.
- Coste computacional.
- Interfaz gráfica.
- Manual de Usuario.
- Tutoriales en línea.

En la tabla 6 se muestra la calificación y evaluación de las alternativas para la selección del software para el modelado 3D.

Tabla 6.
Matriz de Selección de Software para modelado 3D.

Criterios	Blender 3D.			Maya.			Cinema 4D.		
	de	Peso	Calificació	Evaluación	Calificació	Evaluació	Calificació	Evaluació	
Selección.			n		n	n	n	n	
Instalación		5%	5	0.25	3	0.15	3	0.15	
Multiplataform		5%	5	0.25	5	0.25	4	0.2	
a									
Licencia		10%	5	0.5	2	0.2	2	0.2	
Coste		20%	4	0.8	3	0.6	3	0.6	
Computacional.									
Interfaz gráfica.		30%	2	0.6	2	0.6	4	1.2	

CONTINÚA 

Manual de Usuario.	de	15%	4	0.6	4	0.6	3	0.45
Tutoriales en línea.	en	15%	5	0.75	5	0.75	5	0.75
Total		100		3.75		3.15		3.55
		%						
Resultado				Desarrolla		NO		NO
				r				

Con los requerimientos de instalación y las correspondientes ventajas y desventajas de cada software, se determinó que la alternativa con mejores características para el uso en el proyecto corresponde a Blender 3D.

2.3.2. Motor de desarrollo para creación de Entorno Virtual.

Alternativas de selección de Software para el Entorno Virtual.

a. Unreal Engine 4.

El motor de juegos Unreal Engine 4 está estructurado de varios componentes que nos ayuda en el desarrollo de cualquier aplicación. Su extensa librería de herramientas y editores permite organizar todos los activos y manipularlos para crear el juego (Lee, 2015). En la figura 12 se puede apreciar una vista previa correspondiente al motor de videojuegos Unreal Engine 4.

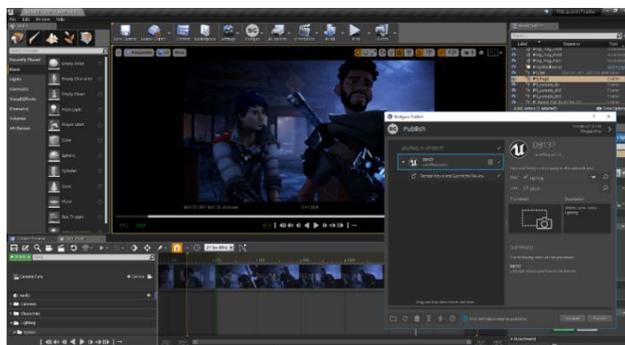


Figura 12. Entorno de Desarrollo Unreal Engine4.

Fuente: (Mayeda, 2018)

Unreal Engine 4, proporciona una serie de componentes en donde incluyen un motor de sonido, motor de física, gráficos, entrada y marco de juegos, y un módulo en línea. Estos componentes son diseñados específicamente para cualquier persona que trabaje con tecnología en tiempo real, desde aplicaciones empresariales, experiencias cinematográficas hasta juegos de alta calidad para ordenador, consolas, dispositivos móviles, realidad virtual y realidad aumentada (UnrealEngine, 2019). En la tabla 7 se muestra los requerimientos para su instalación.

Tabla 7.
Requerimientos de instalación para UnrealEngine4.

Requisitos necesarios	
CPU	Procesador Quad-core Intel o AMD, 2.5 GHz o mayor.
RAM	8GB de RAM en el sistema.
Espacio en Disco	15 GB de espacio libre en el Disco.
Tarjeta Grafica	Video compatible con DirectX10

Fuente: (UnrealEngine, 2019).

Ventajas y desventajas correspondiente a la utilización de Unreal Engine.

Ventajas.

- Calidad gráfica excelente.
- Permite crear aplicaciones de gran dimensión y mucho más compleja.
- Licencia Gratuita.
- Resultados de gráficos increíbles y de excelente rendimiento
- Lenguaje de programación C++ y por medio de blueprints.

Desventajas.

- Curva de aprendizaje complicada.
- Posee poca optimización en proyectos orientados a dispositivos móviles.
- Reducida comunidad.

b. Unity 3d.

El motor de juego Unity 3d, permite el desarrollo de tecnología 2D como el de la 3D que presenta prestaciones y funcionalidades para las necesidades específicas del desarrollador en los diversos géneros (Unity3D, 2019). En la figura 13 se puede apreciar una vista previa del Unity 3D en su versión paga.



Figura 13. Entorno de Desarrollo Unity 3D.

Fuente: (Unity3D, 2019)

Dentro del editor de Unity 3D presenta multitud de herramientas que permite una edición e iteración de manera rápida dentro del ciclo de desarrollo destinadas a implementar la lógica del juego y un juego del alto rendimiento, además, incluye el modo Play permitiendo obtener vistas previas de forma rápida del trabajo desarrollado en tiempo real (Unity3D, 2019).

En la tabla 8 se describe los requerimientos necesarios para instalación de Unity 3D.

Tabla 8.

Requerimientos de Instalación para Unity 3D.

Requisitos necesarios	
CPU	Core 2 Duo ó superior , Soporte para el conjunto de instrucciones SSE2.
RAM	8GB de RAM en el sistema.
Espacio en Disco	100MB de espacio libre en el disco.
Tarjeta Grafica	Tarjeta de video con capacidad para DX10 (shader modelo 4.0).

Fuente: (Unity3D, 2019).

Ventajas y desventajas correspondiente a la utilización del motor de juegos Unity 3D.

Ventajas.

- Interfaz intuitiva, fácil de utilizar.
- Manejo de dos lenguajes de programación JavaScript y Csharp.
- Multiplataforma: Windows y Mac, o iOS y Android.
- Asset Store completa.
- Manual de usuario disponible gratuita.
- Licencia de pago y gratuita.
- Resultados de gráficos increíbles y de excelente rendimiento

Desventajas.

- Mala gestión de memoria.
- Mala gestión de versiones específicamente de 4 a 5.
- Espacio ocupado por los proyectos al uso del Assets muy complejos.

c. Godot Engine.

Godot Engine es motor de video juegos multiplataforma que permite crear complejos aplicativos de 2D y 3D desde una interfaz unificada. Godot proporciona a los desarrolladores un gran conjunto integral de herramientas comunes como objeto de que el usuario tenga la capacidad de concentrarse en desarrollar el juego sin la necesidad de reinventarse la rueda (GodotEngine, 2019). En la figura 14 se puede apreciar una vista previa del entorno de desarrollo de GodotEngine.

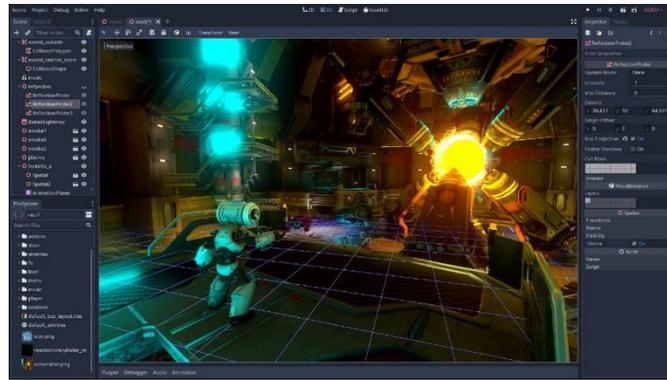


Figura 14. Entorno de Desarrollo Godot Engine.

Fuente: (GodotEngine, 2019)

En la tabla 9 se muestra los requerimientos correspondientes para su instalación.

Tabla 9.

Requerimientos de instalación para Godot Engine.

Requisitos necesarios	
CPU	Intel Core 2 Duo E8400.
RAM	4GB de RAM en el Sistema.
Espacio en Disco	Desconocido.
Tarjeta Grafica	NVIDIA GeForce 8200M G <i>CONTINÚA</i> 

Fuente: (GodotEngine, 2019)

Ventajas y desventajas correspondiente a la utilización del motor de juegos Godot Engine.

Ventajas.

- Licencia MIT, libre y completamente open source sin ninguna restricción.
- Sistema operativo Windows, macOS, X11 (Linux, *BSD).
- Depuración del aplicativo en el dispositivo.
- Lenguaje de programación GDScript.

Desventajas.

- Curva de aprendizaje complicada.
- No existe mucha documentación.

Selección de software para desarrollo del Entorno Virtual.

Ahora, para determinar el software adecuado para el desarrollo de diseño en 3D se tiene los siguientes criterios:

- Licencia.
- Coste computacional.
- Interfaz gráfica.
- Curva de aprendizaje.
- Calidad de Resultados Gráficos.
- Lenguaje de programación.

En la tabla 10 se muestra la calificación y evaluación de las alternativas para la selección del software de desarrollo del Entorno de Realidad Virtual.

Tabla 10.

Matriz de selección para el software del motor de juegos.

Criterios	Unreal Engine.		Unity 3D.		Godot Engine.			
	de	Peso	Calificació	Evaluació	Calificació	Evaluación	Calificació	Evaluació
Selección.			n	n	n		n	n
Licencia		10%	5	0.5	4	0.4	5	0.5
Coste Computacional		10%	4	0.4	4	0.4	4	0.4

Interfaz gráfica	20%	4	0.8	4	0.8	4	0.8
Curva de Aprendizaje.	30%	3	0.9	5	1.5	3	0.9
Calidad de resultados gráficos.	20%	5	1	5	1	5	1
Lenguaje de programación.	10%	4	0.4	5	0.5	3	0.3
Total	100%		4		4.6		3
Resultado			NO		Desarrolla		NO
					r		

Con los requerimientos de instalación y las correspondientes ventajas y desventajas de cada software, se determinó que la alternativa con mejores características para el uso en el proyecto corresponde a Unity 3D.

2.4. Selección de hardware.

2.4.1. Dispositivo controlador de Gestos.

Leap Motion Controller.

Es un dispositivo electrónico que permite el reconocimiento y seguimiento de los movimientos de las manos y dedos con gran precisión. El principio de funcionamiento del dispositivo se basa en la captura de movimiento por tiempo de vuelo, posee dos cámaras monocromáticas y tres LEDs infrarrojos, por lo que gracias a esta tecnología se puede controlar varias interfaces sin la necesidad de tocar las pantallas, simplemente con movimiento de las manos en el aire (BELDA, 2015). En la figura 15 se puede apreciar el dispositivo Leap Motion Controller.



Figura 15. Dispositivo Leap Motion Controller.

Fuente: (BELDA, 2015)

El dispositivo Leap Motion en comparación con otros controladores de gestos existentes en el mercado tiene su dimensionamiento reducido, específicamente de 75 mm de largo, 25 mm de ancho y 11 mm de alto. La dirección de los sensores cuando está en su posición de funcionamiento estándar es a lo largo del eje Y. El trabajo de seguimiento y la detección del controlador funciona mejor cuando el controlador posee una vista clara y un alto contraste referente al objeto, además que cuando el mismo funciona en condiciones extremas el software combina los datos del sensor con un modelo interno de la mano humana con el objetivo de proporcionar un seguimiento predictivo (LeapMotion, Leap Motion Controller, 2018).

En la tabla 11 se muestra los requerimientos de instalación para Leap Motion Controller.

Tabla 11.

Requerimientos de instalación para Leap Motion.

Requerimientos de Instalación.	
Espacio para la instalación.	Espacio de libre en el disco para la instalación de 400 MB.
Procesador.	Procesador AMD Phenom™ II or Intel® Core™ i3/i5/i7.
Plataforma.	Sistema Operativo Windows® 7+ or Mac® OS X 10.7+

Alimentación y transmisión de Controlador USB 2.0 o 3.0.

datos.

RAM RAM en el Sistema 8GB

GPU Tarjeta Gráfica NVIDIA GTX 970 / AMD R9 290
equivalente o mayor.

Fuente: (LeapMotion, Leap Motion Controller, 2018)

En la tabla 12 se muestra las características técnicas de Leap Motion Controller.

Tabla

12.

Características técnicas de Leap Motion.

Características Técnicas

Dimensionamiento 75mm de largo, 25mm de ancho y 11mm de alto.

Cámara

- Cámaras con sensores monocromáticas.
- Sensible a la luz infrarroja, con una longitud de onda de 850nm.
- Trabaja a una velocidad de hasta 200 fps.

Luz infrarroja

- Trabajan a un espectro ^{CONTINÚA} de luz infrarroja a una longitud de onda a 850nm, misma a la de los sensores óptimos.
- Consumo de energía variante dependiendo de la luz en el entorno.

Zona de Cobertura Zona de cobertura semiesférica de 61cm ^{Continúa} de radio. 

Integración con motores de Unity 3D, Unreal Engine, Godot Engine
juegos

Lenguajes de programacion C++, C#, Java, JavaScript, Python y Objective-C.

Fuente: (BELDA, 2015)

Ventajas y desventajas correspondiente a la utilización del dispositivo Leap Motion Controller.

Ventajas.

- Precio reducido.
- Tamaño reducido y compacto.
- Admite únicamente detección de manos y dedos.
- Alta precisión para escritorio físico hacia arriba, también para montar en un casco de Realidad Virtual.
- Correcto funcionamiento al exponer a condiciones de luz escasa y extrema.

Desventajas.

- No incorpora detección de voz.
- Espacio de detección reducido.
- Su uso excesivo provoca recalentamiento.

Microsoft Kinect.

Dispositivo electrónico que permite el reconocimiento y seguimiento de los movimientos del cuerpo humano. Fue desarrollado por Microsoft que inicialmente fue desarrollado como un accesorio de juego Xbox 360 y One como periférico al estilo de webcam, dando el control de la consola por parte del usuario sin la necesidad de tocar el mismo, únicamente con los gestos y comandos hablados. Luego gracias al auge que tuvo en el mercado fue impulsado la línea de usos a funciones más avanzadas como sensor de bajo coste (Greene, 2012). En la figura 16 se aprecia una vista previa del dispositivo de reconocimiento de movimientos del cuerpo Kinect.



Figura 16. Dispositivo electrónico Kinect.

Fuente: (Greene, 2012)

En la tabla 13 podemos encontrar los requerimientos necesarios para la instalación de Kinect.

Tabla 13.

Requerimientos de instalación para Kinect.

Requerimientos de instalación

Procesador físico dual-core de 3.1 GHz (2 núcleos lógicos por sistema físico) o procesador más rápido de 64Bits.

Controlador USB 3.0 dedicado al sensor Kinect para Windows v2 o al Adaptador Kinect para Windows para usar con el sensor Kinect para Xbox One.

4GB de RAM en el Sistema

CONTINÚA 

Tarjeta gráfica que admita DirectX 11.

Sistema Operativo Windows 8 u 8.1, Windows Embedded 8 o Windows 10

Fuente: (Microsoft, 2019)

En la tabla 14 se muestra las características técnicas de Kinect.

Tabla 14.

Características técnicas de Kinect.

Características técnicas

Sensor

Lentes de color y sensación de profundidad

Campo de vision	<ul style="list-style-type: none">• Horizontal 57 grados.• Vertical 43 grados.• Rango de inclinación física ± 27 grados.• Rango de profundidad del sensor 1.2 – 3.5 metros
Flujo de datos	<ul style="list-style-type: none">• 320 x 240 a 16 bits de profundidad a 30 fps.• 640 x 480 a 32 bits de color a 30 fps.
Sistema de Seguimiento	<ul style="list-style-type: none">• Rastrea hasta 6 personas, incluyendo 2 jugadores activos.• Rastrea 20 articulaciones por jugador activo.
Sistema de Audio	<ul style="list-style-type: none">• Reconocimiento de voz múltiple.
Integración con motores de juegos	Unity 3D y Unreal Engine.
Lenguajes de programación	de C Sharp.

Fuente: (Microsoft, 2019)

Ventajas y Desventajas correspondiente la utilización de Kinect.

Ventajas.

- Capacidad de reconocer y captar movimiento aun en la oscuridad.
- Reconocimiento de movimiento del cuerpo completo.
- Incorpora detección de voz.
- Amplio rango de detección.

Desventajas.

- Soporte solo para plataforma Windows.
- Único lenguaje de programación CSharp.
- Requerimiento de espacio físico amplio.
- Tasa de flujo de datos baja.

Selección de hardware para el reconocimiento de gestos y movimiento de las manos.

Para determinar el dispositivo controlador de gestos adecuado para el proyecto, se tomó en consideración los siguientes criterios.

- Tasa de flujo de datos.
- Capacidad de reconocimiento a diferentes cantidades de luz.
- Rango de detección amplio.
- Reconocimiento de manos y dedos.
- Integración con motores de videojuegos.
- Manejo de varios lenguajes de programación.

En la tabla 15 se muestra la calificación y evaluación de las alternativas para la selección de hardware para el reconocimiento de gestos y movimiento de las manos.

Tabla 15.

Matriz de selección para el Hardware controlador de gestos.

Criterios de Selección.	Leap Motion.			Kinect.	
	Peso	Calificación	Evaluación	Calificación	Evaluación
Tasa de flujo de datos	30%	5	1.5	3	0.9

Capacidad de reconocimiento a diferentes cantidades de luz.	10%	4	0.4	4	0.4
Rango de detección amplio	10%	3	0.3	5	0.5
Reconocimiento de manos y dedos.	20%	5	1	3	0.6
Integración con motores de videojuegos	20%	5	1	2	0.4
Lenguajes de programación	10%	5	0.5	2	0.2
Total	100%		4.7		3
Resultado			Desarrollar		NO

Con los requerimientos de instalación y las correspondientes ventajas y desventajas de cada hardware controlador de gesto, se determinó que la alternativa con mejores características para el uso en el proyecto corresponde a Leap Motion Controller.

2.4.2. Tarjeta electrónica de control.

Arduino

El proyecto Arduino es una plataforma de creación electrónica de código abierto, por lo tanto, la misión de arduino es hacer que la tecnología sea accesible para todos, basado en hardware y software libre, flexible y fácil de utilizar para creadores y desarrolladores. El proyecto nació en el año 2003 como objetivo de facilitar el acceso y uso de la electrónica y programación; y como resultado se tuvo una placa con todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador siendo una alternativa económica para ese entonces (FM, 2018). En la figura 17 se aprecia la placa arduino Uno

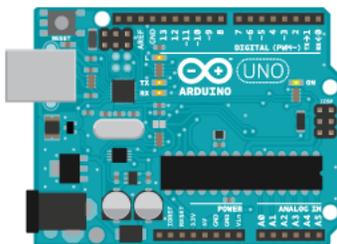


Figura 17. Placa Arduino Uno.

Fuente: (Arduino, Arduino, 2019)

En la actualidad arduino posee varias formas, tamaños y colores para cualquier necesidad. Para este proyecto se toma en consideración dos alternativas: Arduino Nano y Arduino Uno.

Arduino Nano.

Arduino nano es una pequeña placa electrónica compacta y completa basada en el microcontrolador ATmega328P en su versión 3.x o ATmega168 en su versión 2.x, compatible con protoboards. No posee conector para alimentación externa, funciona con un cable USB mini-B (Aduino, 2019). En la figura 18 se aprecia una vista superior correspondiente al Arduino nano.

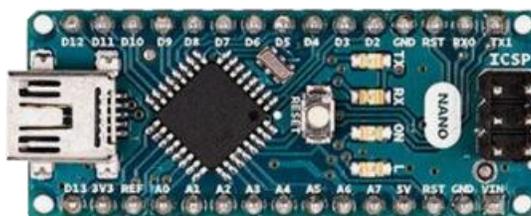


Figura 18. Placa Electrónica Arduino Nano.

Fuente: (Arduino, Arduino, 2019)

En la tabla 16 se muestra las características técnicas para la tarjeta Arduino nano.

Tabla 16.

Características Técnicas Arduino Nano.

Características técnicas

Microcontrolador	ATmega 328 versión 2.x, y ATmega 328P versión 3.x.
Tensión de operación	5V
Tensión de alimentación	7 – 12 Recomendada
Puertos de Entrada/Salida de 0 a 13	14 de las cuales 6 son PWM
Corriente de salida	40 mA
Frecuencia de trabajo	16MHz

Fuente: (Aduino, 2019)

Ventajas y desventajas de la utilización de la placa electrónica Arduino nano.

Ventajas.

- Tamaño reducido (18,5mm x 43,2mm).
- Ideal para trabajos en PCBs.
- Selección automática de fuente de alimentación.
- Facilidad de comunicación con la computadora, otro Arduino u otro microcontrolador.
- Precio económico.

Desventajas.

- Numero de pines de entrada y salida digitales 14.
- Máxima corriente de entrega de 40mA.
- No posee conector para alimentación externa

Arduino Uno.

Arduino uno es una placa de microcontrolador basada en ATmega328P similar al arduino nano en su versión 3.x, una de las mejores placas de control electrónica para iniciar con el aprendizaje de la electrónica y código, se caracteriza por ser robusta. Esta placa fue elegida para el lanzamiento del software Arduino IDE 1.0 (Arduino, ARDUINO UNO REV3, 2019). En la figura 19 se aprecia la vista superior correspondiente al Arduino Uno.



Figura 19. Tarjeta de control Arduino Uno.

Fuente: (JADIAZ, 2016)

En la tabla 17 se muestran las características técnicas correspondientes al Arduino Uno.

Tabla 17.

Características técnicas Arduino Uno.

Características técnicas	
Microcontrolador	ATmega 328P
Tensión de funcionamiento	5V
Voltaje de entrada	7 – 12 (Recomendada), 6 – 20 (Límites)
Pines de Entrada y Salida Digitales	14 pines de los cuales 6 son PWM
Entradas Analógicas	6
Corriente de salida por Pin	40 mA
Corriente de salida a 3.3 V	50mA
Frecuencia de trabajo	16Mhz

CONTINÚA



Fuente: (JADIAZ, 2016)

Ventajas y desventajas correspondiente a la utilización de la placa electrónica Arduino Uno.

Ventajas.

- Ideal para iniciar proyectos, placa electrónica robusta.
- Soporte para comunicar entre la computadora, otro Arduino u otro microcontrolador.
- Posee conector para la alimentación.

Desventajas.

- No apto para trabajos con PCBs.
- Tamaño de 68,6 mm de largo y 53,4 mm de ancho, mayor tamaño en comparación con el Arduino nano.
- Precio mayor en comparación con el Arduino nano.

Selección de hardware para la tarjeta de control electrónico.

Se considera los siguientes parámetros de selección para la tarjeta de control Arduino y se aplica la matriz de selección de componentes.

- Soporte para comunicación entre PC – Arduino.
- Implementación en PCBs.
- Tamaño reducido.
- Precio económico.
- Numero de pines.

En la tabla 18 se muestra la calificación y evaluación de las alternativas para la selección de hardware para la tarjeta de control electrónico.

Tabla 18.
Matriz de selección de tarjeta de control electrónico.

Criterios de Selección.	Arduino Nano			Arduino Uno	
	Peso	Calificación	Evaluación	Calificación	Evaluación
Soporte para comunicación entre PC – Arduino.	40%	5	2	5	2
Implementación en PCBs.	20%	5	1	3	0.6
Tamaño reducido	20%	5	1	3	0.6
Precio económico	10%	4	0.4	3	0.3
Numero de pines	10%	5	0.5	5	0.5
Total	100%		4.9		4
Resultado			Desarrollar		NO

Dada los criterios de selección correspondiente a la tarjeta de control se determinó que la alternativa con mejores características técnicas y sus ventajas de uso corresponde a Arduino Nano.

2.5. Sistema de percepción háptica.

Para el desarrollo del sistema de percepción háptica que será utilizado dentro del entorno de Realidad Virtual se tiene como referencia la normativa ISO TC159/SC4/WG9, en donde define la parte ergonómica de los sistemas de interacción desde el nivel de complejidad del mismo en relación con el tipo de retroalimentación proporcionado hacia el usuario. Esta normativa focaliza la caracterización de los dispositivos Hápticos en definición de medidas físicas. En (Erp, Kyung, Kassner, & Carter, 2010) determina la clasificación de la definición de medida física de los dispositivos Hápticos en tres niveles de complejidad que se muestran en la tabla 19.

Tabla 19.
Nivel de complejidad que define a los dispositivos Hápticos.

Nivel de complejidad	Medida Física	Ejemplo Aplicativo
Medida Elemental	Fuerza, tiempo y distancia	Umbrales de percepción, capacidades máximas/mínimas de usuarios y dispositivos, resoluciones, etc.
Medida Compuesta	Velocidad	Velocidad máxima de la mano de un aparato
	Frecuencia	La percepción de la vibración depende de la frecuencia
	Conductividad térmica	Material constante
Medida Derivada	Impedancia	Comportamiento mecánico de un usuario
	Transparencia	Transmisión de la transparencia
	Rugosidad	Varias medidas como Ra , Rz , Rq y Rsk

Fuente: (Erp, Kyung, Kassner, & Carter, 2010)

Se define que el diseño para el Sistema de percepción háptica se toma como referencia el nivel de complejidad elemental influyente al sistema táctil, en donde se utilizara pequeños motores de vibración ubicados en las yemas de los dedos.

2.5.1. Conceptualización.

Para el sistema de percepción háptica se utilizará dos micro-motores de vibración que irán ubicados en las yemas de los dedos índice. Para que al usuario se le haga fácil el uso del mismo se confeccionará un par de guantes que se tomó como referencia a la chaqueta de escaneo de (Limited, 2019), para este guante se utilizó materiales Sintéticos como el poliéster acolchado y el cuero ecológico. Luego, para la parte de control el sistema tendrá una tarjeta arduino nano que se

comunicará con el entorno virtual a través de un puerto USB por comunicación serial, mismo que estará dispuesto dentro de una caja modelada en 3D.

2.5.2. Diseño del par de Guantes.

El par de guante estará dividido en cuatro capas, dos de ellas van ubicadas en la parte de la yema de del dedo índice y del dedo menique entre la cuales van dispuesto los micro-motores de vibración y las otras dos van en la parte dorsal de la mano, como se muestra en la figura 20.

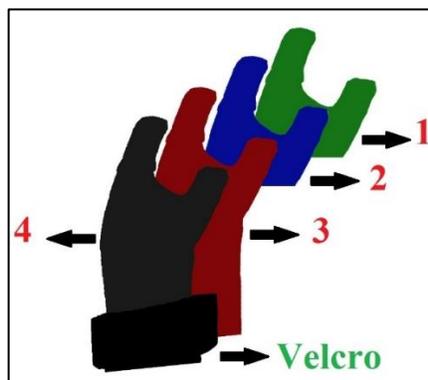


Figura 20. Diseño de Guantes para el Sistema Háptico.

Como se describió anteriormente el objetivo de colocar dos capas en la parte de las yemas de los dedos índice es para evitar el contacto directo con el usuario y los micro-motores, permitiendo reducir la vibración en toda la estructura del guante y centrándola únicamente en las yemas de los dedos índice. Las dos primeras capas corresponden al material sintético poliéster, material utilizado generalmente en ropa deportiva. La segunda capa corresponde al material poliéster, pero acolchado para proporcionar una mejor percepción visual del mismo en cuestión de robustez y estética, la última capa corresponde al material sintético cuero ecológico que tiene por objetivo proporcionar una mejor presentación del mismo.

Aspectos biométricos de las manos.

Para el desarrollo de cada uno de los guantes que permitan la disposición de los motores de vibración utilizados dentro del sistema de percepción háptica se tiene en primera instancia determinar algunos aspectos biométricos de las manos, tales como el ancho y largo tanto de la mano como de las falanges de cada dedo. En (Binvignat, 2012) nos indica el registro realizado en su estudio para determinar los aspectos biométricos de la mano de un individuo de entre 20 y 24 años de edad, en donde se tiene un promedio de 166 mm en mujeres y 186 mm en hombres tomadas desde el pliegue distal de la muñeca hasta el extremo distal del dedo medio. El ancho de 78mm en mujeres y 91mm en hombre, con respecto a las longitudes de los dedos se considera la sumatoria de cada sector de las falanges teniendo como resultados la siguiente tabla número 20 en donde también se considera su respectivo ancho.

Tabla 20.

Resultados de las dimensiones de las manos en hombre y mujeres.

Longitud mujeres				
Dedo pulgar	Dedo indicador	Dedo medio	Dedo anular	Dedo menique
63,5 mm	91 mm	100,4 mm	94,4 mm	75,8 mm
Longitud Hombres				
72,5 mm	105 mm	115 mm	109 mm	88,5 mm

Fuente: (Binvignat, 2012))

De la tabla 20 se toma un promedio entre las longitudes del dedo índice entre el hombre y la mujer con un valor de 98 mm y 82,15 mm respectivamente, al igual que los valores del ancho y largo de la mano de 84,5mm y 176 mm, tal como se muestra en la figura 21.

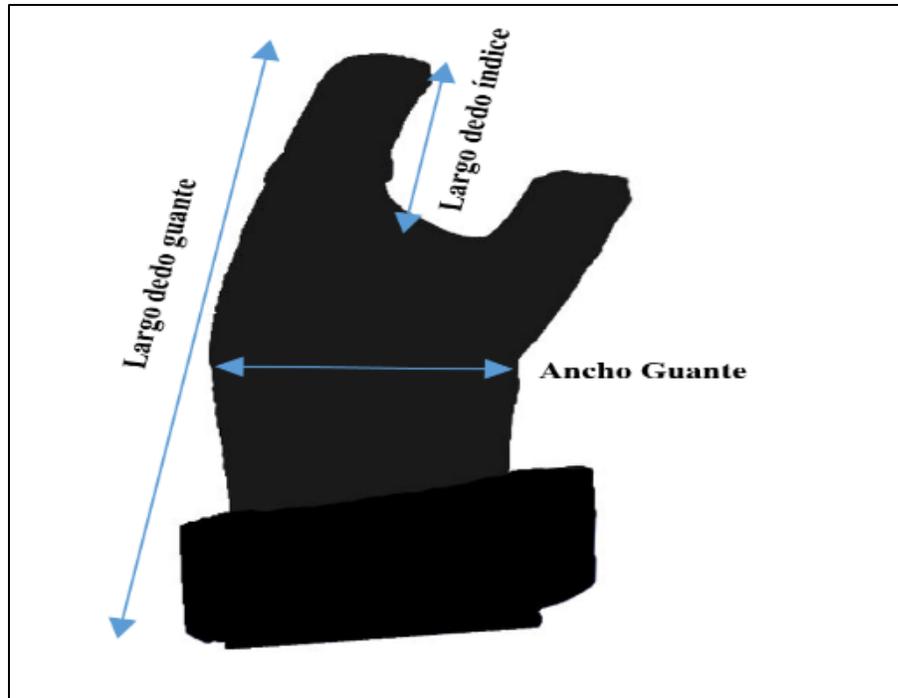


Figura 21. Dimensionamiento del guante.

2.5.3. Diseño de electrónico del sistema háptico.

En la presenta sección se describe la configuración de la conexión electrónica y del control del sistema háptico, siendo el arduino nano la tarjeta de control principal con sus respectivos micro-motores de vibración.

2.5.3.1. Diagrama de conexión entre el Computador, el arduino y los micro motores de vibración.

La conexión entre la tarjeta de control arduino nano y los micro-motores de vibración es directa, como se muestra en la figura 22.

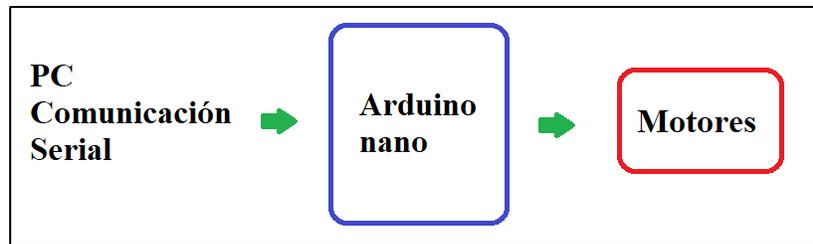


Figura 22. Diagrama de comunicación del sistema Háptico.

2.5.3.2. Esquematación del diagrama de conexión entre el Arduino y Micro-motores.

Como se ha descrito anteriormente, la parte electrónica del sistema de percepción háptica está compuesta por un arduino nano, dos micro-motores de vibración. Por lo tanto, el esquema electrónico de conexión se muestra en la figura 23.

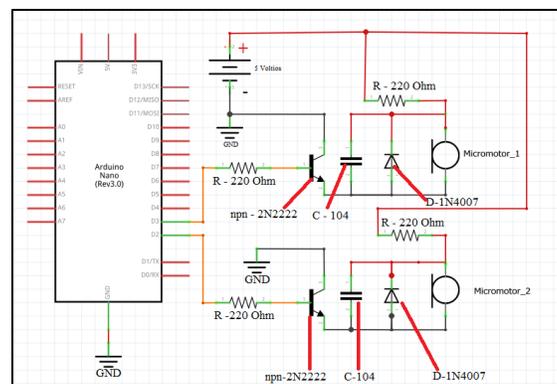


Figura 23. Esquema eléctrico del sistema de control háptico.

2.5.3.3. Algoritmo de control para la activación de los micro-motores de vibración.

La programación para el control de activación de los micro-motores de vibración se realizó dentro de la IDE de arduino en donde el arduino recibe un dato por medio del puerto USB a través de la comunicación serial, y dependiendo del caso se activa o desactiva los micro-motores, en la figura 24 se muestra el algoritmo de control para el sistema háptico conformado por dos motores de vibración.

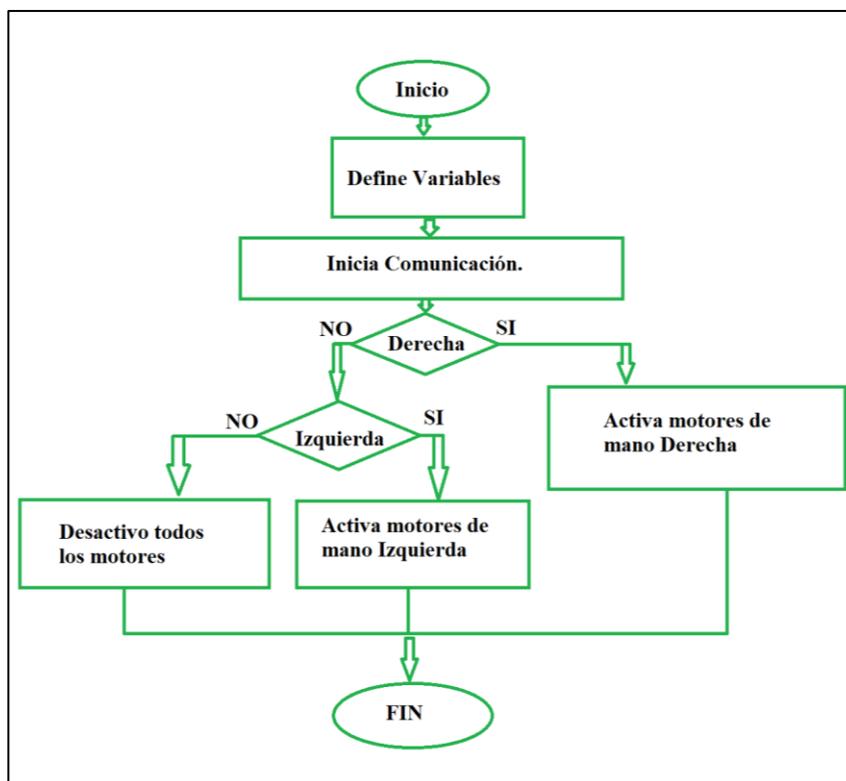


Figura 24. Algoritmo de control del Sistema háptico.

Como se observa en la figura anterior dentro del apartado b), el arduino recibe dato de tipo entero, este dato lo compara con tres casos diferentes, para el caso de valor cero (valor de entrada 0) el arduino da la orden de señal en bajo para los pines digitales de los motores de ambas manos, para el caso de valor uno (valor de entrada 1) el arduino da la orden de señal en alto para el pin digital del motor de la mano derecha y señal en bajo para el de la mano izquierda, y para el caso de valor dos el arduino da la orden de señal alto para el pin digital del motor de la mano izquierda y señal en bajo para el de la mano derecha.

2.5.4. Diseño mecánico del sistema háptico.

En esta sección se muestra en diseño de la parte mecánica del dispositivo háptico, mismo que tiene por objetivo encapsular la parte de la tarjeta de control arduino. Además, permitirá ser

colgado en cualquier lugar sobre el usuario para evitar interferencia con el reconocimiento de las manos por parte del dispositivo LeapMotion. Por lo tanto, se diseñará una caja rectangular que contenga tanto al arduino nano su respectiva tarjeta expansiva y los cables de conexión, mismo que será impreso en 3D en el material ácido poli-láctico más conocido como PLA.

2.5.4.1. Diseño CAD de la estructura base del arduino nano.

Para facilitar la conexión entre los motores de vibración y el arduino nano se utilizó una tarjeta expansiva prefabricada diseñada exclusiva para el arduino nano, mismo que tiene de dimensiones 5.72 cm x 5.35 cm de largo y ancho respectivamente. Además, otro dato a tener en consideración para el modelado de caja base es la altura del conjunto: Arduino nano, shield de expansión y los cables de conexión con un valor de 26mm. Por lo tanto, la base tendrá por dimensiones 55 mm de ancho y 61mm de largo con una altura de 30 mm, como se muestra en la figura 25.

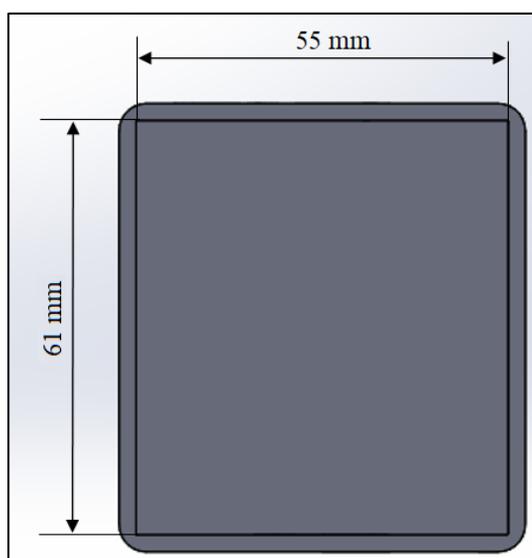


Figura 25. Dimensionamiento para el encapsulado del Arduino.

En la figura 26 se muestra el ensamble de la estructura, misma que está dividida por:

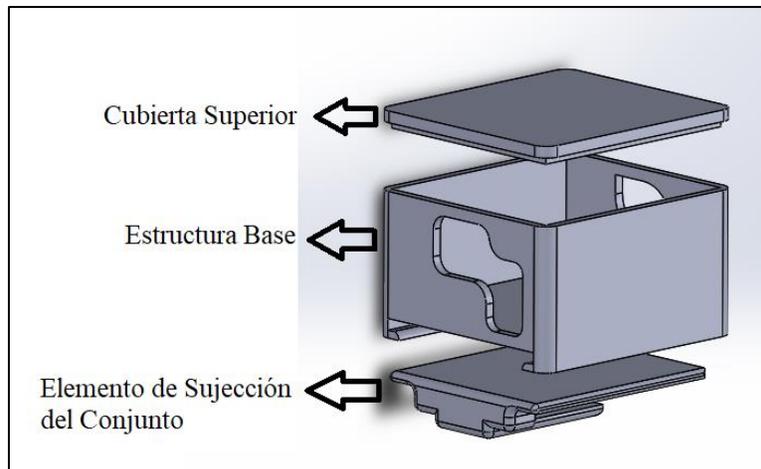


Figura 26. Ensamble de la estructura para el encapsulado del Arduino.

2.5.4.2. Análisis de resistencia para el elemento de sujeción.

En la figura 27 se determina la forma en donde va ubicada la estructura base de la tarjeta de control de sistema de háptico para una mayor comodidad por parte del usuario del aplicativo, por lo tanto, el elemento sujetador va a estar sujeto a esfuerzos de flexión por lo que es necesario un análisis mecánico para determinar su resistencia.

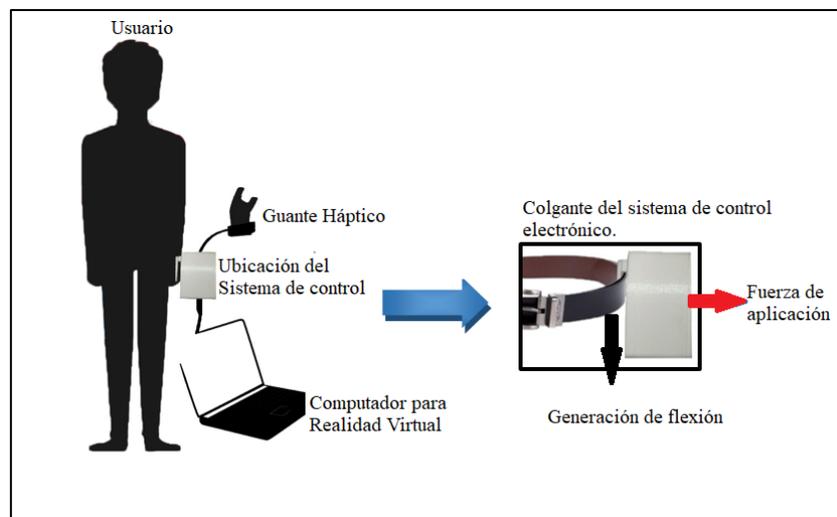


Figura 27. Ubicación de la tarjeta de control del sistema háptico.

Para determinar la fuerza de aplicación se tomó de referencia el estudio realizado en (DMena, 2017), en donde determina las fuerzas ejercidas por la mano, dicha fuerza fue determinada utilizando galgas extenso-métricas como captador de señal. En dicho estudio se realizó pruebas en tres posiciones, para este caso se toma como referencia la posición B correspondiente a: Pinza término-lateral del pulgar y el segundo dedo con apoyo del resto de dedos trifalángicos. El valor máximo resultante es equivalente a 10,5 Kilogramos-Fuerza (103 Newtons). Para el análisis de resistencia mecánica se tomó al elemento sujetador mostrado en la figura 28.

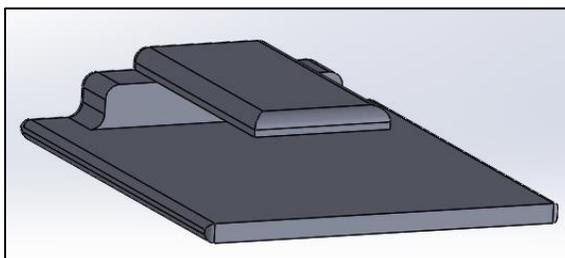


Figura 28. Elemento sujetador.

Como material de construcción para todo el conjunto se utilizó PLA (Acido Poli Láctico) determinado a través de la polimerización por condensación directamente de su ácido láctico que se deriva de la fermentación de la glucosa de fuentes como el maíz, papa caña de azúcar y trigo, mismo que es un material biodegradable, no toxico y utilizado en la tecnología de impresión 3D que por lo general es la tecnología más rápida, barata y fácil de utilizar en comparación con otras tecnologías de fabricación. En la tabla 21 se muestra algunas de las propiedades mecánicas del material PLA de marca Ultimaker.

Tabla 21.
Propiedades mecánicas del material PLA.

Propiedad	Valor	Unidad	Método de Prueba
------------------	--------------	---------------	-------------------------

CONTINÚA 

Densidad	1.26	g/cm ³	
Módulo de tensión	2,346.5	MPa	ISO 527 (1mm/min)
Resistencia a la fluencia	71.7	MPa	ISO 527 (50mm/min)
Resistencia ultima a la fluencia	70.32	MPa	ISO 527 (50mm/min)
Resistencia a la flexión	110.31	MPa	ISO 178
Módulo de Young	4495.38	MPa	ISO178
Limite elástico	60	MPa	

Fuente: (Ultimaker, 2018)

Los elementos impresos en 3D tendrá 0% de material de relleno por lo que será netamente sólido y la dirección de la impresión será de 45 grados y -45 grados.

Como parte del análisis al elemento sujetador, se determina la dirección de la fuerza de aplicación. En la figura 29 se muestra la dirección de la fuerza aplicada con un valor de 113 Newtons.

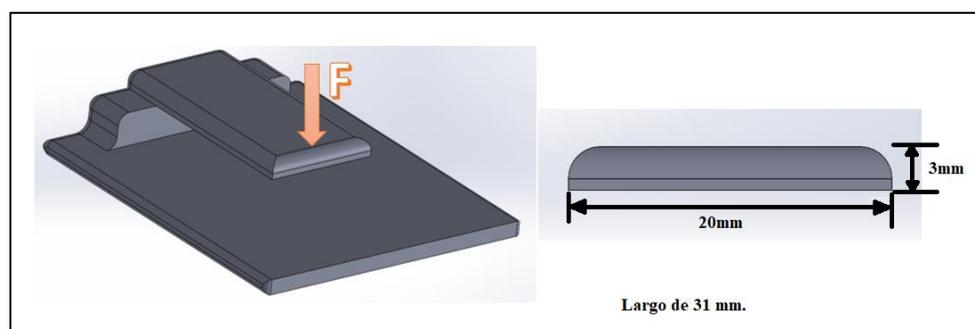


Figura 29. Posición de fuerza a elemento sujetador.

El elemento sujetador se analiza como una viga en voladizo como se muestra en la figura 30, con sección transversal equivalente a 20 mm de ancho y 3 mm de alto. La viga estará sometida tanto a esfuerzo normal (por flexión) como a cortantes como se observa en la figura 30.

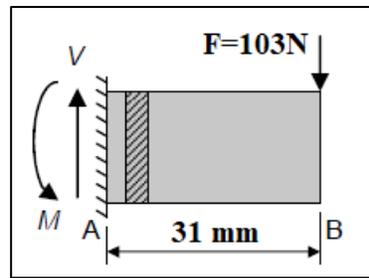


Figura 30. Viga de análisis.

Para determinar los puntos críticos en primera instancia se determina las reacciones en el empotramiento. Con las ecuaciones de equilibrio estático (Ecuación 1 y 2) se determina las reacciones en el empotramiento.

$$\sum F = 0; \quad \text{Ec.1}$$

$$V - 103 [N] = 0$$

$$V = 103 [N]$$

$$+\text{SentidoHorario} \sum M_A = 0; \quad \text{Ec.2}$$

$$(103 [N])(0.031 [m]) - M = 0;$$

$$M = 3.193 [Nm];$$

2.5.4.2. Determinación de la densidad del material impreso.

Una de las características a tener en consideración es el cambio que genera su densidad. Como es un material que varía. En la figura 31 se determina la masa para el material impreso en 3D con el objetivo de realizar el respectivo análisis de falla.



Figura 31. Medida del peso en objeto de análisis mecánico.

Como se observa en la figura 31 la masa es equivalente a 10.1 gramos, luego se procede a medir el volumen. En la figura 32 se muestra las propiedades físicas del elemento en donde se determina el volumen total de la pieza de $12547,16 \text{ mm}^3$.

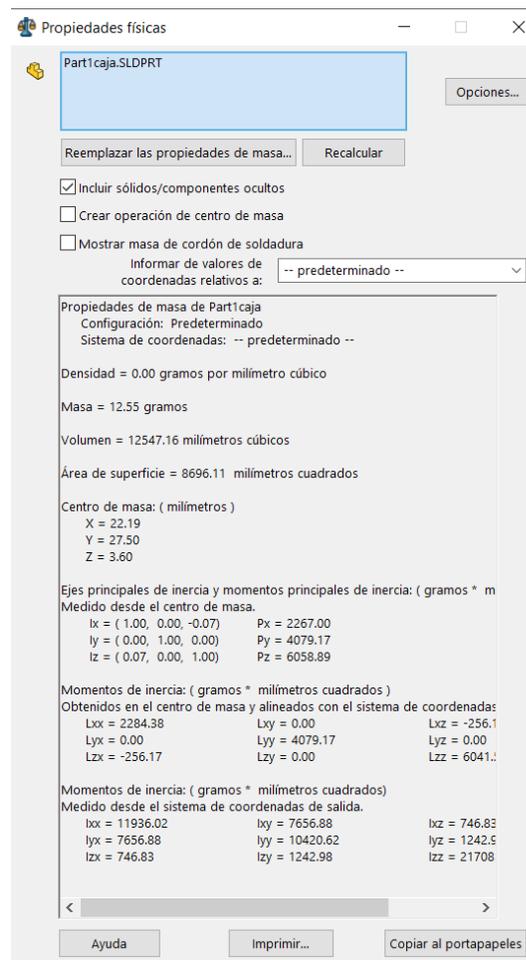


Figura 32. propiedades físicas de la pieza a ser analizada.

Luego se determina su correspondiente densidad con la ecuación 3.

$$\rho = \frac{m}{v}; \quad \text{Ec.3}$$

Donde:

m es la masa del elemento de 10.1 gramos.

v el volumen del elemento de $12547,16 \text{ mm}^3$.

ρ es la densidad.

$$\rho = \frac{m}{v} = \frac{10.1 \text{ g}}{12547,16 \text{ mm}^3} = \frac{8.0496 \times 10^{-4} \text{ g}}{\text{mm}^3} = 0.80496 \text{ g/cm}^3;$$

Se compara esta densidad calculada con la densidad que proporciona las propiedades mecánicas del material PLA utilizado para la impresión en 3D.

$$\rho = 0.80496 \frac{\text{g}}{\text{cm}^3} \text{ calculado} < 1.26 \frac{\text{g}}{\text{cm}^3} \text{ densidad del material}$$

Se considera un material compuesto cuando existe un cambio significativo en su densidad por lo tanto el análisis mecánico se realiza para un material compuesto.

En la siguiente figura 33 se muestra los correspondientes diagramas de fuerzas cortantes y momentos flectores con sus respectivos diagramas de esfuerzos. Los puntos más críticos de la viga son (1), (2) y (3). (1) y (2) son los puntos más alejados del eje neutro, soportando los mayores esfuerzos normales a tracción y compresión respectivamente, y (3) es el esfuerzo que soporta mayor esfuerzo cortante. La teoría de falla para materiales compuestos que se aplica para determinar el buen diseño es la *teoría de falla de la máxima tensión*, que dice que “*El material falla cuando el mismo supera una cierta tensión limite en una zona determinada*”. (Molinier, 2018)

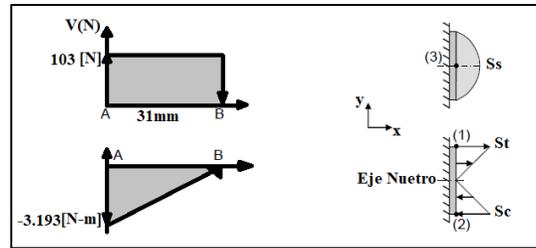


Figura 33. Diagramas de fuerzas cortantes y momentos flectores con sus respectivos diagramas de esfuerzos.

Para el elemento sometido a flexión, el esfuerzo se calcula a partir de la ecuación 4.

$$S_{xx} = \frac{\pm Mc}{I}; \quad \text{Ec.4}$$

Donde:

$$M = 3.193 [N - m];$$

$$c = 1.5mm;$$

La inercia de la sección transversal se calcula a partir de la ecuación 5.

$$I = \frac{1}{12} (b * h^3); \quad \text{Ec.5}$$

$$I = 4.5 \times 10^{-11} [m^4];$$

$$S_{xx} = \pm 106.43 [MPa];$$

Entonces:

$$S_t = 106.43 [MPa];$$

$$S_c = -106.43 [MPa];$$

Esfuerzos principales para el punto 1 como se observa en el círculo de Mohr de la figura 34.

$$\sigma_1 = \sigma_A = 106.43 \text{ [MPa]}, \quad \sigma_2 = \sigma_3 = 0; \quad \tau_{max} = 53.215 \text{ [MPa]}.$$

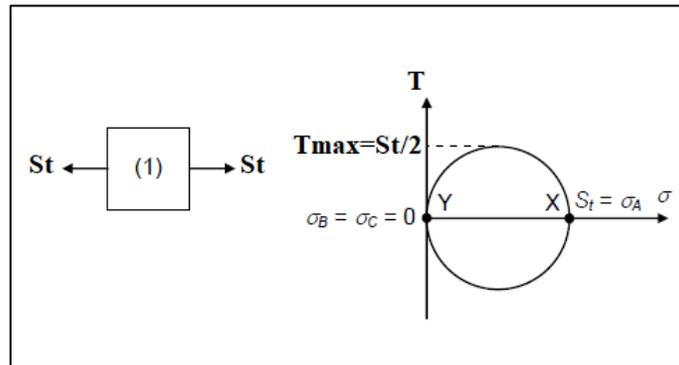


Figura 34. Circulo de Mohr para el punto 1.

Esfuerzos principales para el punto 2 como se observa en el círculo de Mohr de la figura 35.

$$\sigma_1 = \sigma_2 = 0, \quad \sigma_3 = \sigma_B = -106.43 \text{ [MPa]}; \quad \tau_{max} = 53.215 \text{ [MPa]}.$$

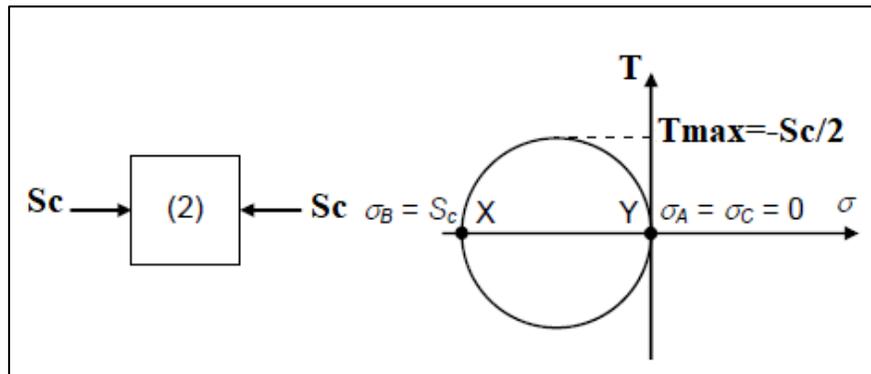


Figura 35. Circulo de Mohr para el punto 2.

Para validar el diseño se procede a determinar el facto de seguridad de acuerdo al esfuerzo por corte con la ecuación 6.

$$FS = \frac{S_{ut}}{\tau_{max}}; \quad \text{Ec.6}$$

$$FS = \frac{S_{ut}}{\tau_{max}};$$

$$Fs = \frac{70.32 [MPa]}{53.215 [MPa]};$$

$$Fs = 1.32 \therefore \text{Si cumple}$$

Como el valor es mayor a 1 teniendo en cuenta que la fuerza es máxima promedio el diseño es correcto. Además, se determina la máxima deflexión que generara con la fuerza aplicada, en donde se determina con la ecuación 7.

$$y_{max} = \frac{-P x^3}{3EI}; \quad \text{Ec.7}$$

Donde x es la distancia entre el empotramiento hasta la fuerza de aplicación equivalente a 31 mm, la inercia con un valor de $I = 4.5 \times 10^{-11} [m^4]$, el módulo de Young de 3150 [MPa]. Se determina la deflexión.

$$y_{max} = \frac{-P x^3}{3EI} = \frac{-(103)(.031)^3}{3(4495.38 \times 10^6)(4.5 \times 10^{-11})} = \frac{N(m^3)}{\left(\frac{N}{m^2}\right)(m^4)};$$

$$y_{max} = -5.0562428 \times 10^{-3} m = -5.0562428 \text{ mm};$$

2.5.4.3. Ensamble total del sistema de percepción háptica.

El ensamble total correspondiente al sistema de percepción háptica basado en la retroalimentación del sistema táctil por medio de un sistema vibro-táctil se presenta en la figura 36 con sus respectivas etiquetas referente a sus componentes.

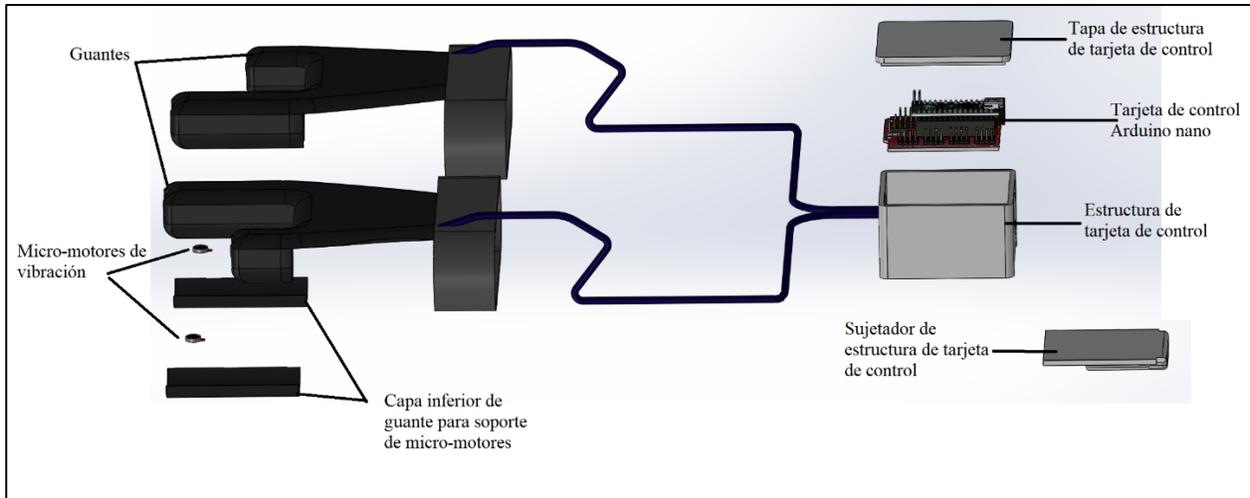


Figura 36. Ensamble total del Sistema de percepción Háptica.

CAPÍTULO III

IMPLEMENTACIÓN DEL SISTEMA HÁPTICO CON REALIDAD VIRTUAL.

3.1. Construcción del Dispositivo Háptico.

Al tener diseño correspondiente del Sistema de percepción háptica se procede a la manufactura del mismo, en la figura 37 se puede apreciar la confección del par de guantes.



Figura 37. Diseño final del par de guantes para el Sistema Háptico.

3.1.1. Impresión en 3D de la estructura mecánica.

Al tener el modelo en 3D de la estructura del encapsulado de la tarjeta de control Arduino nano validado por el software CAE Workbench Student, se procede a su manufactura utilizando la tecnología de Impresión 3D con material PLA (Acido Poli Láctico). Para preparar la impresión se utiliza el software Ultimaker Cura. Dentro de los parámetros a tener en consideración para la impresión son:

- **Calidad:** Altura de capa, para este caso 0,2mm.
- **Temperatura del extrusor:** 210 grados Centígrados.
- **Temperatura de la cama:** 60 grados Centígrados.
- **Velocidad de impresión:** 40mm/s.

Cada uno de los parámetros anteriormente mencionados influyen directamente en el resultado final de la impresión, cada uno de estos datos fueron tomados en (Ultimaker, 2018) data sheet del PLA utilizado e implementados en el software de preparación de impresión 3d Cura Ultimaker, como se muestra en la figura 38.

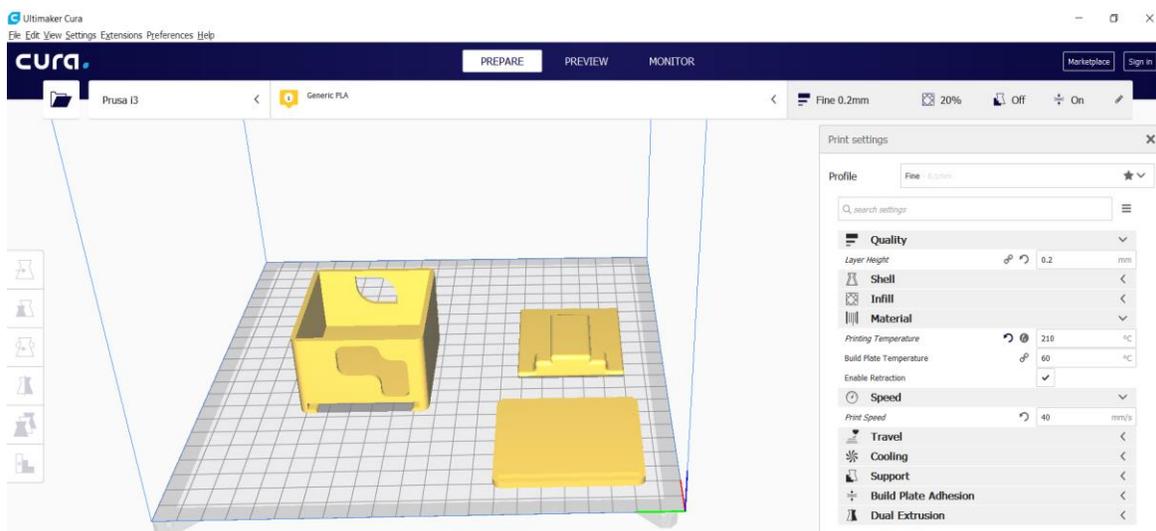


Figura 38. Parámetros de configuración para Impresión 3D.

3.1.2. Ensamble Total del dispositivo Háptico.

Los micro-motores de vibración van ubicados en la parte de las yemas de los dedos índice de las manos. Como se especificó dentro del diseño del par de guantes, en cada dedo tiene un espacio para los micro-motores, los cuales se los coloca tal como se muestra en la figura 39.



Figura 39. Disposición de los micro-motores de vibración en el guante.

La estructura que encapsula al arduino debe tener cierta distancia con respecto a los guantes con el objetivo de evitar interferencia en el reconocimiento de las manos y para que el usuario se sienta más cómodo con su uso, por lo que se dio una distancia de 1 metro. Para la alimentación de los micro-motores de vibración, se utilizó cable de alimentación USB mismo que se adecuo para su uso dentro del sistema de percepción háptica. El ensamble total se muestra en la figura 40.



Figura 40. Ensamble Total de Dispositivo Háptico.

3.2. Desarrollo de la interfaz.

3.2.1. Conceptualización.

Una de las características antes de empezar con el desarrollo del aplicativo es que Unity 3D se basa en el desarrollo de aplicaciones por escenas, en cada escena el desarrollador puede implementar una tarea en específico. Por lo tanto, para el desarrollo del Entorno de Realidad Virtual creado dentro del motor de juegos Unity 3D se tendrá al inicio de la aplicación una presentación sobre el proyecto en donde se muestre el tema de proyecto y su respectivo nombre de autor y director, en la misma escena se tendrá un apartado en donde determine la conexión tanto del dispositivo Leap Motion como del dispositivo Háptico, luego pasará a una segunda escena llamada menú principal en donde se tendrá la parte de inicio de una partida, videos de la disposición de las piezas mecánicas del robot y su salida del aplicativo, al dar inicio el aplicativo se le indicara al usuario a modo de ayuda unos Tips para poder manejarse dentro de ella de manera adecuada cuando realice la tarea de montaje o desmontaje. Cuando el usuario ingrese a una nueva partida podrá visualizar otro apartado llamado submenú en donde podrá seleccionar seis modos de juegos cada uno correspondiente a sus partes principales del robot y una general; cada parte se divide en tareas de Montaje y Desmontaje que depende de la elección del usuario, al finalizar alguna de las dos tareas el usuario se tendrá un apartado en donde puede observar el funcionamiento del robot con un aplicativo de clasificación de objetos por colores.

3.2.2. Estructuración del Concepto.

En este apartado se crea el esquemático que defina la conceptualización del proyecto, mismo que definirá el comportamiento, reglas, acciones y limitaciones por el usuario, para que el mismo pueda manejarse de forma adecuada dentro de cada escena del Entorno Virtual. En la figura 41 se muestra la estructura del concepto del Entorno Virtual

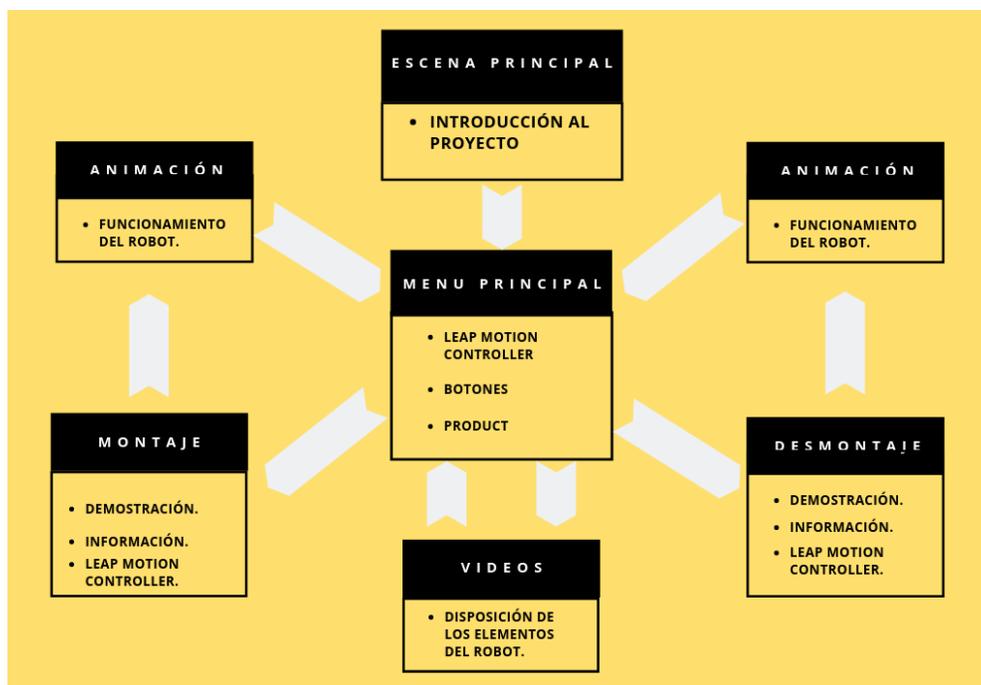


Figura 41. Estructuración de escenas del Entorno Virtual.

3.2.3. Proceso de implementación.

A. Psicología de color.

El uso del color dentro de cualquier aplicación es de suma importancia ya que afecta de manera muy compleja hacia el cerebro, causando sensaciones y respuestas por parte del usuario. Por lo tanto, el uso de una paleta de colores adecuada será una parte fundamental para el proyecto, a continuación, se muestra la base para seleccionar los colores adecuados para nuestra paleta de colores que se utilizará para el diseño de las escenas de Sistema de Realidad Virtual (García, 2018).

Bloques de Colores.

Colores cálidos.

Estos colores son aquellos que van desde el rojo hasta el amarillo, estos colores representan la pasión, felicidad, entusiasmo, fuerza y energía.

Colores fríos.

Estos colores son aquellos que van desde el azul al verde, y representan la calma, la estabilidad, la profesionalidad y la honradez. En la figura 42 se muestra la rueda de colores.

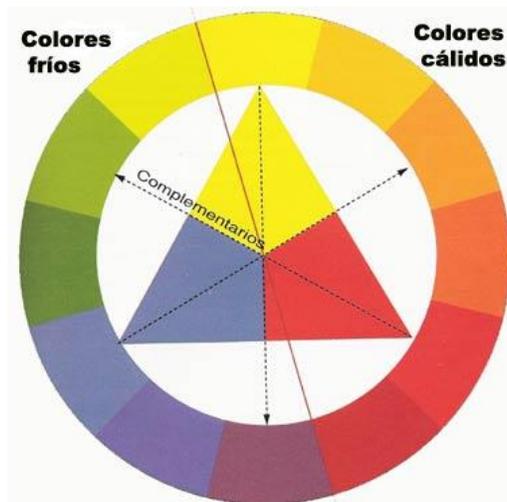


Figura 42. Definición de colores.

Fuente: (García, 2018).

Un color cálido y un frío se complementan, mismos que se debe tener en cuenta para transmitir el mensaje, esta parte de la psicología del color ayuda a crear la paleta de colores

Rueda de colores.

La rueda de colores es una distribución de 12 colores alrededor de un círculo, mismos que se encuentran divididos en tres grupos:

1. **Primarios:** Rojo, Amarillo y Azul, estos no se pueden crear mezclando otros colores.
2. **Secundarios:** Creados a partir de la combinación de los tres anteriores; Verde, Violeta y Naranja.
3. **Terciarios:** Creados a partir de la combinación de los primarios y secundarios.

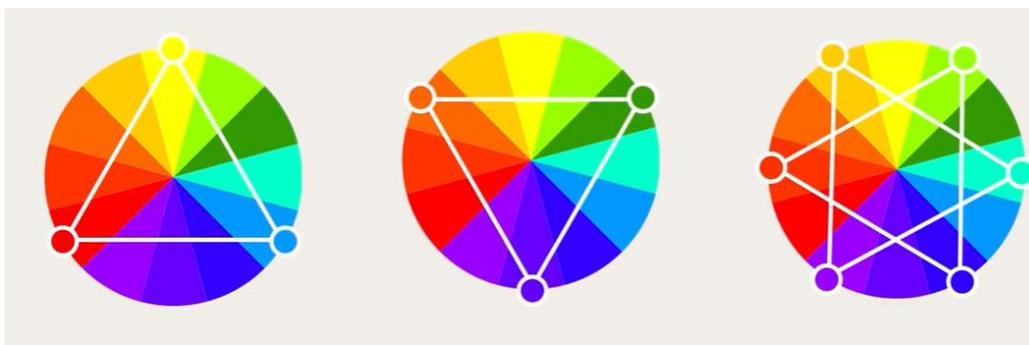


Figura 43. División de la rueda de colores.

Armonías y reglas del color.

La armonía de colores permite combinar de una manera fácil para crear una paleta de colores. Dentro de las reglas de armonía se tiene las siguientes:

Colores Complementarios.

Son aquellos que se encuentran ubicados en puntos opuestos en la rueda de colores. Estos colores se refuerzan mutuamente, de manera que un mismo color parece más vibrante e intenso cuando se halla asociado a su complementario. Estos contrastes son, pues, idóneos para llamar la atención y para proyectos donde se quiere un fuerte impacto a través del color (ADesign, 2011).

Colores Análogos.

Los colores análogos ocupan posiciones inmediatamente próximas en la rueda de colores. En razón de su parecido, armonizan bien entre sí. Este tipo de combinaciones es frecuente en la naturaleza (ADesign, 2011).

Colores en Triadas.

Son colores que equidistan dentro de la rueda de colores situadas a 120 grados uno del otro. Versiones más complejas incluyen grupos de cuatro o de cinco colores, igualmente equidistantes

entre sí (situados en los vértices de un cuadrado o de un pentágono inscrito en el círculo.) (ADesign, 2011)

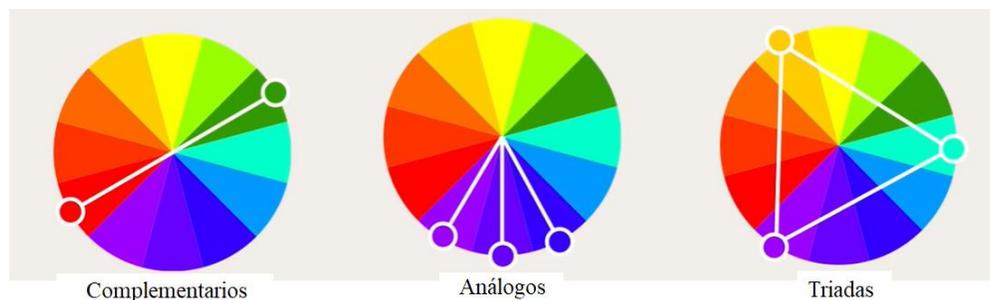


Figura 44. Armonía de los colores.

Fuente: (ADesign, 2011)

Anatomía del color.

Cada color tiene una serie de parámetros que los diferencia de uno a otro. Al añadir blanco o negro se tiene la posibilidad de conseguir nuevos colores. Dentro de los parámetros a tener en consideración para la definición de los colores tenemos:

- **Tono:** Se define como la propia cualidad de un color, siendo todos aquellos que pertenecen a la rueda de colores como son los primarios, secundarios y terciarios.
- **Saturación:** Se define como el grado de pureza de un color. Un color está en su máxima saturación cuando no contiene blanco, negro o su color complementario.
- **Luminosidad:** Se define a la cantidad de blanco o negro contenido en un tono, a mayor cantidad de blanco mayor luminosidad.

Creación de la paleta de colores.

Con la teoría del color antes descrita, se procede a la creación de las paletas de colores que se utilizarán dentro de todo el sistema de Realidad Virtual. Para lo cual se hará uso de la herramienta

Online Adobe Color CC, misma que nos ayuda a generar impresionantes paletas de colores o armonía de cualquier color base (Dixon, 2019).

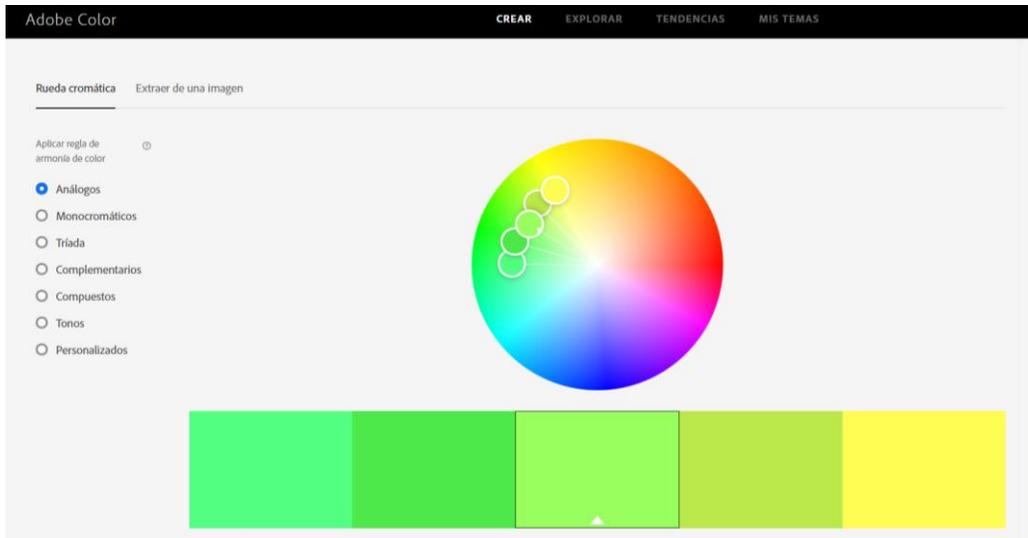


Figura 45. Creación de paleta de colores.

Una de las consideraciones a tener en cuenta es el modelo de color. En Adobe Color CC da la posibilidad de crear paletas de color es variable se puede encontrar modelos en RGB, CMYK y HSV. Unity 3D maneja el modelo de color en RGB, por lo tanto, se crea la paleta en ese modelo. En Blender 3D el modelo de color es RGB unitario, es decir, varía de 0 a 1.

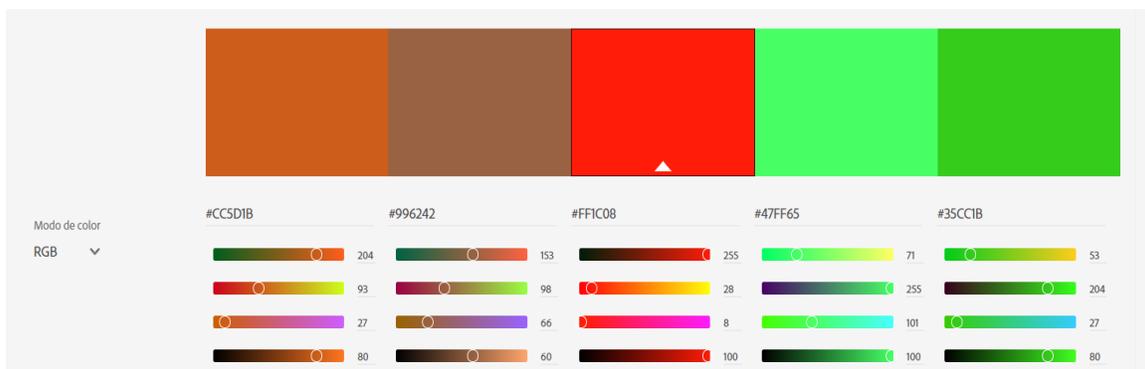


Figura 46. Definición de modelos de color.

Escena “Presentación del Proyecto”.

Para la primera escena denominada “Presentación del Proyecto”, se utiliza la siguiente paleta de colores mostrados en la figura 47. Se utiliza estos colores ya que el gris/negro representa la formalidad, esto ayudara a la presentación del proyecto para llamar la atención del usuario de la aplicación.

RGB 128, 144, 166	RGB 33, 37, 38	RGB 55, 61, 64	RGB 176, 191, 191	RGB 152, 166, 166
HSV 214, 23, 65	HSV 197, 15, 15	HSV 197, 14, 25	HSV 180, 8, 75	HSV 180, 8, 65
CMYK 23, 13, 0, 35	CMYK 15, 4, 0, 85	CMYK 14, 4, 0, 75	CMYK 8, 0, 0, 25	CMYK 8, 0, 0, 35
LAB 59, -1, -13	LAB 14, -1, -2	LAB 25, -2, -2	LAB 76, -5, -2	LAB 67, -5, -2

Figura 47. Paleta de colores utilizadas en la Escena "Presentación del Proyecto".

Para la implementar la paleta de colores dentro de Unity 3D en primera instancia se debe crear dentro de la carpeta de Assets un material dándole un nombre cualquiera, luego se procede a colocar los valores en RGB del color seleccionado dentro de la paleta de colores, finalmente se procede a arrastrar el material hacia el objeto. En la figura 48 se muestra el color utilizado como fondo de la escena.

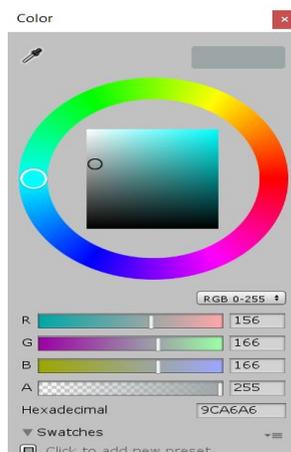


Figura 48. Aplicación de colores en Unity.

Escenas “Menú Principal”, “Submenú Montaje”, y “Submenú Desmontaje”.

Para esta escena se utilizó colores más puros, se utilizó la paleta de color que fue utilizada en CloseUpUVPportrait. Se utiliza el rojo ya que representa la aventura, el verde el equilibrio y azul la inteligencia; colores mayormente utilizados en esta escena junto con el negro. En la figura 49 muestra la paleta correspondiente a estos colores con sus respectivos valores en RGB.

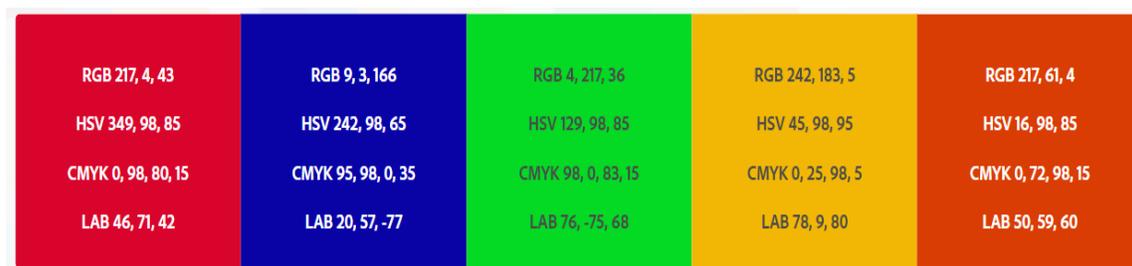


Figura 49. Paleta de colores en Escena "Selección de Tipo de Juego".

Escenas “Montaje” y “Desmontaje”.

Para estas dos escenas se utilizó los siguientes colores, sin tener en consideración las piezas del robot. Estos colores son utilizados dentro de habitaciones para trabajar, ya que calma y tranquilidad además de aumenta la productividad (VALENZUELA, 2019). En la figura 50 muestra la paleta correspondiente a estos colores con sus respectivos valores en RGB.

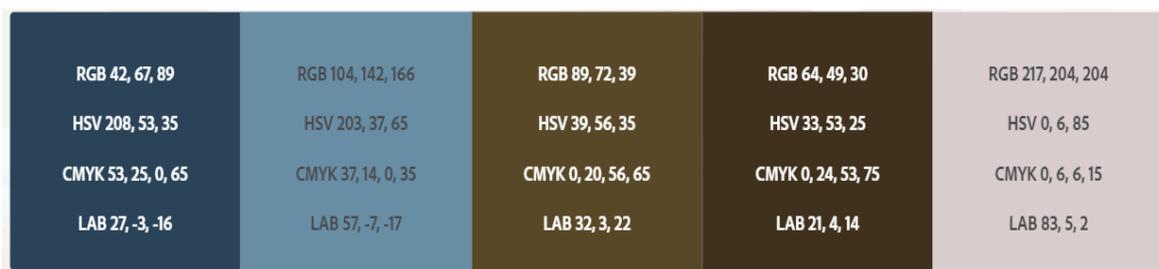


Figura 50. Paleta de colores en Escenas "Montaje y Desmontaje".

B. Escena “Presentación del Proyecto”

En esta escena llamada *Presentación del Proyecto*, es la escena más básica de todas, misma que cuenta con un fondo de imagen (Background), que denota el modelo 3D del Equipo Mecatrónico, siendo dicho fondo de imagen un Objeto gráficos en 2D que es conocido dentro de la concepción de Unity como “Sprite”, siendo un Sprite texturas que permiten ser agregados a un GameObject como una superficie donde se dibujara dicha imagen. Otro componente dentro de esta escena es el CANVAS, en donde podemos encontrar el logotipo tanto de la carrera como del Establecimiento; además del tema del Proyecto, el autor y tutor; como se muestra en la figura 51. Esta escena tiene un script llamado “EscenaPrincipal”, mismo que tiene por objetivo proporcionar un tiempo de visualización, cargando la escena de menús llamado “SelecciónTipoJuego” al llegar a los 9 segundos.

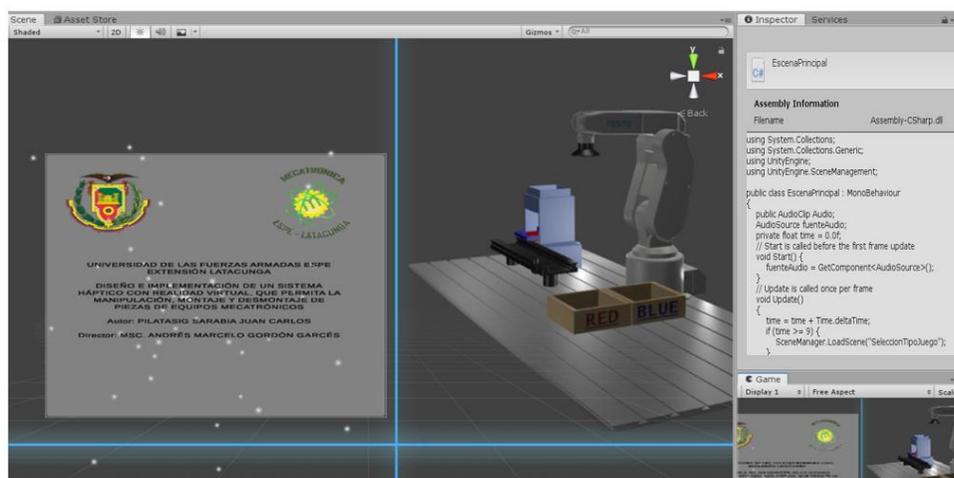


Figura 51. Escena "Presentación del Proyecto".

Dentro de esta escena se encuentra un apartado en donde se determina la conexión del dispositivo Háptico y del controlador Leap Motion. El sistema permitirá el uso o no del dispositivo háptico dentro del Entorno de Realidad Virtual, eso dependerá del usuario si desea usar el programa

con o sin retroalimentación háptica, a diferencia del dispositivo Leap Motion, este si es necesario tener su conexión. En la figura 52 se muestra las pantallas correspondientes a la no conectividad y la conectividad de los dispositivos e información sobre el posicionamiento de las manos sobre el dispositivo Leap Motion, al estar conectados aparecerá un botón de prueba para poderlo manipular dando 10 golpes, al terminar con la tarea aparece dos botones en donde permite ingresar a la aplicación o salir.



Figura 52. Conectividad de dispositivos.

En la figura 53 se muestra el código utilizado para determinar la conectividad correspondiente al dispositivo háptico. Se puede apreciar el uso de la librería Leap con *using Leap*, la creación del objeto *controller* correspondiente a la clase *Controller*, y su respectiva sentencia *If Else* que determina la activación o desactivación de los gráficos en la pantalla cuando se ha detectado la conectividad del dispositivo Leap Motion.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Leap;
using UnityEngine.UI;
using System.IO.Ports;

public class estadoDispositivos : MonoBehaviour
{
    // Start is called before the first frame update
    Controller controller = new Controller();

    //LeapMotion
    if (controller.IsConnected)
    {
        textoConectLeap.SetActive(true);
        textoDescoLeap.SetActive(false);
        DescoLeapImg.SetActive(false);
        ConectLeapImg.SetActive(true);
        botonPrueba.SetActive(true);
        panelManos.SetActive(true);
    }
    else
    {
        textoConectLeap.SetActive(false);
        textoDescoLeap.SetActive(true);
        DescoLeapImg.SetActive(true);
        ConectLeapImg.SetActive(false);
        botonPrueba.SetActive(false);
        panelManos.SetActive(false);
    }
}

```

Figura 53. Código para determinar la conectividad del Leap Motion.

En la figura 54 se determina el código utilizado para determinar la conectividad del dispositivo Háptico. Se hace uso de la librería *System.IO.Ports* para el uso de puerto serie dentro del programa. Para determinar si se ha conectado algún dispositivo al puerto USB reconociéndole como puerto COM se hace uso de la función *SerialPort.GetPortNames()* que dentro de un *String* se guarda los nombres de ello, luego se compara si se ha conectado al puerto COM4 y se activa o desactiva los gráficos en la pantalla.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Leap;
using UnityEngine.UI;
using System.IO.Ports;

string[] ports = SerialPort.GetPortNames();
if (ports.Length > 0)
{
    if (ports[0] == "COM4")
    {
        texPuertoComConect.SetActive(true);
        texPuertoComDesconect.SetActive(false);
        ImgPuertoCom4conect.SetActive(true);
        ImgPuertoCom4Noconect.SetActive(false);
        texPuertoCom4Noconect.SetActive(false);
        scoreBoton.SetActive(true);
    }
    else
    {
        texPuertoComConect.SetActive(false);
        texPuertoComDesconect.SetActive(false);
        texPuertoCom4Noconect.SetActive(true);
        ImgPuertoCom4conect.SetActive(false);
        ImgPuertoCom4Noconect.SetActive(true);
        scoreBoton.SetActive(false);
    }
}
```

Figura 54. Código utilizado para determinar el dispositivo háptico.

C. Escena “Inicio de partida”.

En esta escena llamada *Inicio de Partida* es la parte correspondiente al menú principal del aplicativo, en donde el usuario de la aplicación puede iniciar una partida, visualizar videos demostrativos de la disposición de las piezas mecánicas del sistema de transmisión de movimiento del equipo mecatrónico y la opción de salir del aplicativo, tal como se muestra en la figura 55.



Figura 55. Menú principal correspondiente al aplicativo.

Para la interacción del usuario con el aplicativo dentro de esta escena se hace uso del dispositivo LeapMotion Controller. Por lo tanto, para poder hacer uso del mismo dentro del Unity 3D agregar los siguientes módulos dentro de la escena: *LeapMotion Core* y *LeapMotion Interaction Engine*, estos módulos son importados como un nuevo Assets. Dentro de cada uno de los módulos se encuentran los algunos componentes que permiten la interacción dentro de la escena.

Componentes básicos de la interacción.

- **Hand Model Manager & Hand Models**, modelo 3D de las manos.
- **Interaction Manager**, recibe el FixedUpdate de Unity, este componente maneja la lógica de la interacción entre usuario y escena, misma que es indispensable para que funciones.
- **Interaction Objects**, son GameObjects adjunto un InteractionBehaviour, mismo requiere un Rigidbody y un Collider para que funciones, para esta escena son los botones.

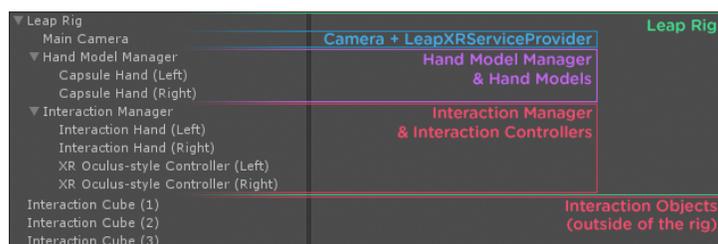


Figura 56. Módulos necesarios para la integración de Leap Motion en Unity 3D.

En la figura 57 se muestra los componentes agregados dentro de esta escena, todos los componentes son hijos del Objeto Cámara.



Figura 57. Implementación de los módulos en Unity 3D.

A cada uno de los botones de la figura 58 se le asigna información sobre que hace cada uno de ellos, la información se mostrara cuando el usuario este suspendido sobre cada botón de forma independiente, en la figura 58 se muestra como se observa la información correspondiente a cada botón. Esta función se logra gracias a los eventos que proporciona el script de *Interaction Button* utilizado en cada botón.



Figura 58. Información adicional sobre cada botón del menú principal.

Al presionar Nueva Partida el programa nos dirige hacia el apartado de ayuda, en este apartado aparecerá datos que permitirá ser de ayuda para el usuario del aplicativo cuando esté realizando ya sea la tarea de Montaje o Desmontaje. Por lo tanto, para la primera parte se tiene unas ventanas que describen como el usuario debe de interactuar con la aplicación como se muestra en la figura 59. Además, posee un botón que permite ir al segundo Tips.



Figura 59. Pantalla de visualización de tips para el manejo dentro del aplicativo.

En la figura 60 se puede observar la función con nombre de *segundoTips()*, del script *ControlPantallas* mismo que permite desactivar el *GameObject primerTipsText* y activar el *GameObject segundoTips*.

```
public GameObject primerTipsText;
public GameObject segundoTipsText;
public GameObject informacionRobot;
public GameObject tips;
public GameObject entorno;
public GameObject parte1;
public GameObject BotonesDemo;
public GameObject CanvasBotonesDemo;
public void segundoTips() {
    primerTipsText.SetActive(false);
    segundoTipsText.SetActive(true);
}
```

Figura 60. Función de control de tips para el aplicativo.

A continuación, se tiene el segundo tips, en donde se describe la parte de la composición de la escena, en la figura 61 se muestra la pantalla del aplicativo. De la misma manera posee un botón de aceptar que se controla por medio del script *ControlPantallas* con la función *informacionRobot()*.

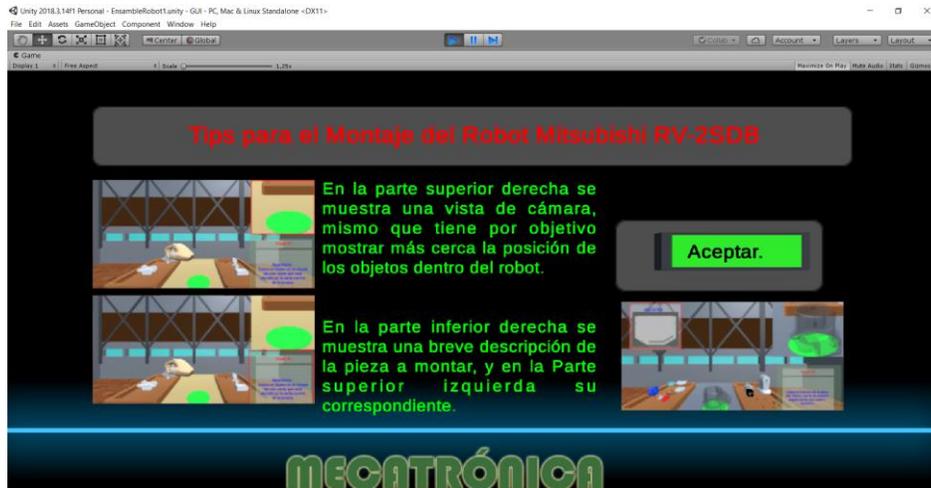


Figura 61. Segundo tips, correspondiente a la composición de la escena.

En la figura 62 se muestra la función utilizada en el evento del botón correspondiente al segundo tips en la escena demostración de montaje.

```
public void iforacionRobot()
{
    tips.SetActive(false);
    informacionRobot.SetActive(true);
}
```

Figura 62. Función de cambio de pantalla a información de robot.

Para la parte de la información del robot como se observa en la figura 61 se muestra características específicas que describen de forma ambigua al robot Mitsubishi RV-2SDB, tal información es la cantidad de grados de libertad, máxima carga, velocidad, repetitividad entre otras, además de una imagen del robot y botones de control. El primer botón permite iniciar el menú de montaje el siguiente botón permite regresar a la escena *Tipo Juego* correspondiente al menú principal del aplicativo.



Figura 63. Pantalla de información de Robot.

Diseño e implementación de botones.

Dentro de Blender 3D se modela cada uno de los botones. Mismo que fue desarrollado a partir de un cubo como base al método de modelado 3D *BOX modeling*, además se agregó un Text como parte de la descripción del mismo, tal como se muestra en la figura 64.

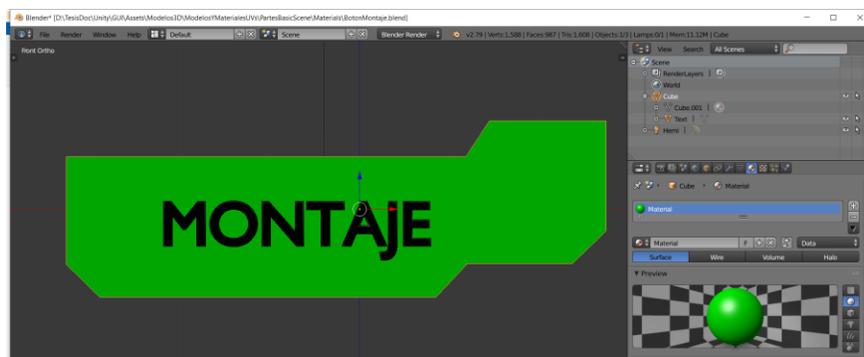


Figura 64. Modelo 3D para el Botón Montaje.

A cada uno de los botones se los guardó dentro de la carpeta Assets del proyecto, para su posterior uso dentro de Unity 3D. Dentro de Unity3D se agrega un GameObject vacío de nombre “Botones”, mismo tendrá contenido los botones de la escena como objeto padre, y los objetos 3d de los botones como hijos tal como se muestra en la figura 65.



Figura 65. Implementación de los botones en Unity 3D.

Para que se dé la interacción entre usuario y los botones dentro del aplicativo, es indispensable y necesario agregar a cada botón de forma independiente un Rigidbody, Collider y un InteractionButton. Dentro de Rigidbody en Constrains se restringe la parte de rotación como se muestra en la figura 66.

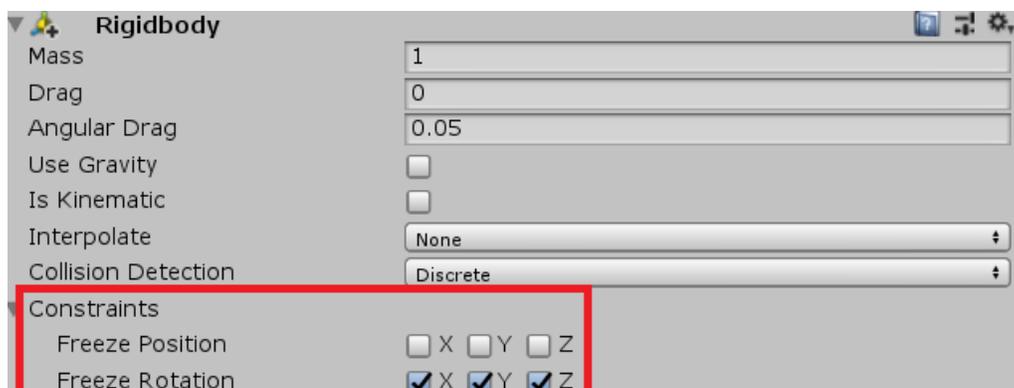


Figura 66. Restricción de movimiento a cada botón.

En “*InteractionButton*”, para que el objeto actúe como botón se configura la parte de *InteractionOverrides*, el cual se debe colocar el visto en *IngnoreGrasping*, la parte de *ContactSetting* en *ContactForceMode* se coloca en UI y finalmente se coloca dentro del evento *On Unpress* la función que controla la acción de cada botón. Cada uno de estos parámetros configurados se muestran en la figura 67.

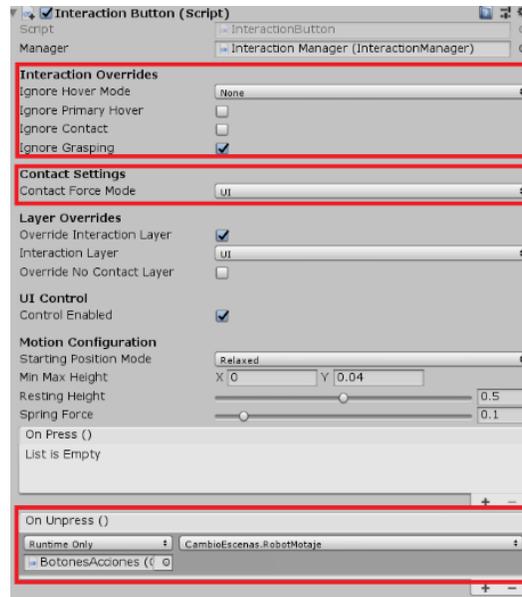


Figura 67. Configuración de parámetros para el componente InteractionButton.

Lo anterior descrito se aplica para todos los botones implementados dentro del proyecto. Además, dentro de la misma escena está la parte de un submenú de videos demostrativos correspondiente a la posición de las piezas mecánicas del robot, tal como se muestra en la figura 68.



Figura 68. Submenú correspondiente a los videos de demostración de la disposición de las piezas mecánicas del Robot.

D. Escenas “Montaje” y “Desmontaje”.

Para que las escenas correspondientes a las tareas de Montaje y Desmontaje se muestre, se realizó un apartado en donde se muestra un submenú correspondiente a las seis partes del robot, en la figura 69 se muestra en submenú correspondiente a la selección de las partes, y su respectiva selección de tarea de Montaje/Desmontaje.



Figura 69. Submenú de selección de tarea.

“Montaje” y “Desmontaje”, son escenas diferentes, pero comparten una misma composición de objetos, como son la parte de la cámara secundaria, el botón de salir de la aplicación, un texto descriptivo de la pieza destinada al montaje o desmontaje y su correspondiente imagen tal como se muestra en la figura 7.

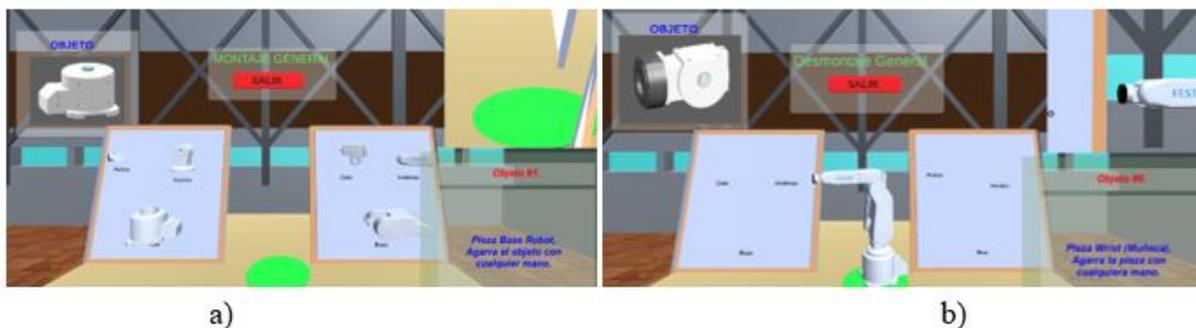


Figura 70. Escenas principales para las tareas de Montaje y Desmontaje.

a) Escena "Montaje", b) Escena Desmontaje.

Previo al desarrollo de las escenas “Montaje” y “Desmontaje”, es necesario el modelado de las partes mecánicas del Robot Mitsubishi RV-2SDB, así mismo el modelado del miembro superior (antebrazo y mano), mismos serán implementados dentro de cada una de las escenas antes mencionadas. Para los cual, se tiene el siguiente apartado.

Modelado Mecánico del Robot Mitsubishi RV-2SDB.

Estructura del Brazo Robótico Mitsubishi RV-2SDB.

En este apartado se muestra un esquema de dibujo de la estructura del brazo robótico figura 71, como se puede apreciar, el esquema proporciona la información de la parte de transmisión mecánica del Robot dirigida hacia cada uno de los ejes.

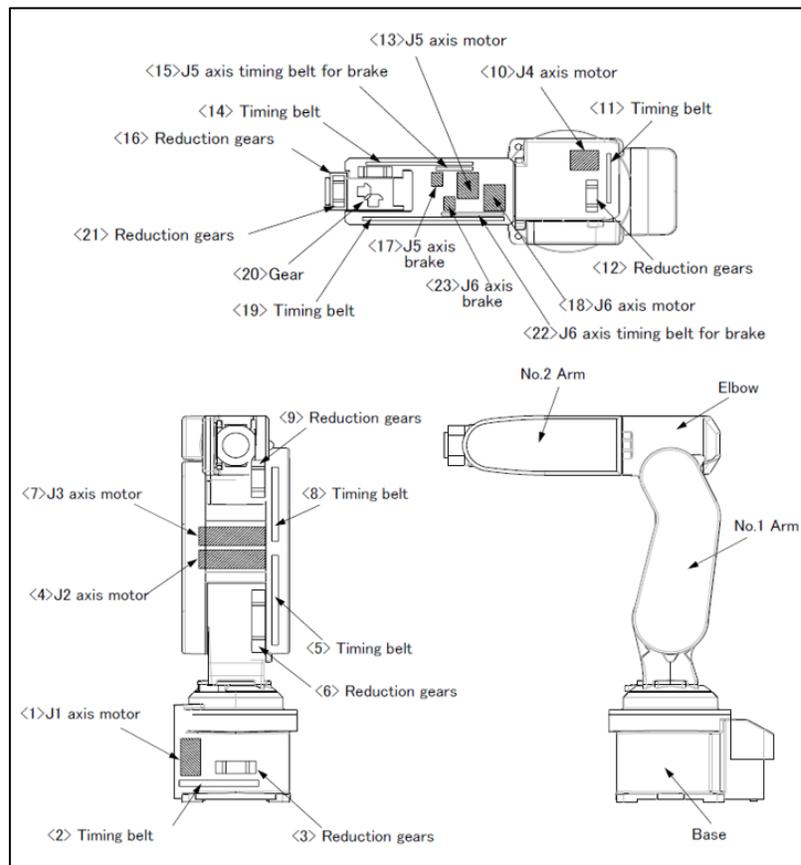


Figura 71. Esquema de dibujo de la estructura del brazo robótico

Fuente: (MelfaRobots, 2010)

A continuación, una breve descripción de cada uno de los ejes correspondientes a Robot Mitsubishi RV-2SDB.

1. La rotación del motor del eje J1 (1) dispuesto en la base es transmitida a la caja de engranes reductora (3) a través de una correa de distribución (2) para rotar el eje J1. Dispone de un freno electromagnético montado en el motor del eje J1 (MelfaRobots, 2010).
2. La rotación del motor del eje J2 (4) dispuesto en el brazo número 1 (No. 1 Arm) es transmitida hacia la caja de engranes (6) a través de una correa de distribución (5) para rotar el eje J2. Dispone de un freno electromagnético montado en el motor del eje J2 (MelfaRobots, 2010).
3. La rotación del motor del eje J3 (7) dispuesto en el brazo número 1 (No. 1 Arm) es transmitida hacia la caja de engranes (9) a través de una correa de distribución (8) para rotar el eje J3. Dispone de un freno electromagnético montado en el motor del eje J2 (MelfaRobots, 2010).
4. La rotación del motor del eje J4 (10) dispuesto en el codo del brazo robótico es transmitida hacia la caja de engranes (12) a través de una correa de distribución (11) para rotar el eje J4. Dispone de un freno electromagnético montado en el motor del eje J2 (MelfaRobots, 2010).
5. La rotación del motor del eje J5 (13) dispuesto en el brazo número 2 (No. 2 Arm) es transmitida transmite hacia la caja de engranes (16) a través de una correa de distribución (14) para rotar el eje J5. Dispone de un freno electromagnético (17) montado vía una correa de transmisión (14) en el motor del eje J5 (MelfaRobots, 2010).
6. La rotación del motor del eje J6 (18) dispuesto en el brazo número 2 (No. 2 Arm) es transmitida hacia la caja de engranes (21) a través de una correa de distribución (19) y una

engrane cónico perpendicular (20) para rotar el eje J6. Dispone de un freno electromagnético (23) montado vía una correa de transmisión (22) en el motor del eje J6 (MelfaRobots, 2010).

Disposición y descripción de tornillos de la estructura del Robot.

En la tabla 22 muestra una descripción de los pernos que se utilizan para cada una de las cubiertas del Robot.

Tabla 22.

Disposición y descripción de tornillos del Robot.

Ítem	Nombre de Cubierta	Tornillos de Cubiertas
1	Cubierta para la batería (Battery Cover)	4 tornillos de presión M3
2	Cubierta de motor J1 (J1 Motor Cover)	4 tornillos de presión M3
3	Cubierta derecha de Brazo (No. 1 Arm Cover R)	4 tornillos de presión M3
4	Cubierta izquierda de Brazo(No. 1 Arm Cover L)	4 tornillos cabeza pequeña M3
5	Cubierta trasera Codo (Elbow Cover B)	2 tornillos de presión M3
6	Cubierta derecha Codo (Elbow Cover R)	5 tornillos de presión M3
7	Cubierta Izquierda Antebrazo(No. 2 Arm Cover L)	4 tornillos de presión M3
8	Cubierta derecha Antebrazo (No. 2 Arm Cover R)	4 tornillos de presión M3
9	Pato inferior (Botton Plate)	4 tornillos cabeza pequeña M3

Fuente: (MelfaRobots, 2010)

Diseño CAD del Sistema mecánico.

Con la información anteriormente proporcionada se procede al modelado 3D de las partes del sistema mecánico del Robot Mitsubishi. RV-2SDB, mismo que está dividido por ejes, como se muestra a continuación.

Sistema de transmisión Eje J1.

En la figura 72 se muestra la descripción gráfica y su respectivo modelado 3D de los componentes correspondiente al sistema de transmisión del eje J1, estos ubicados en la parte de la Base del Robot.

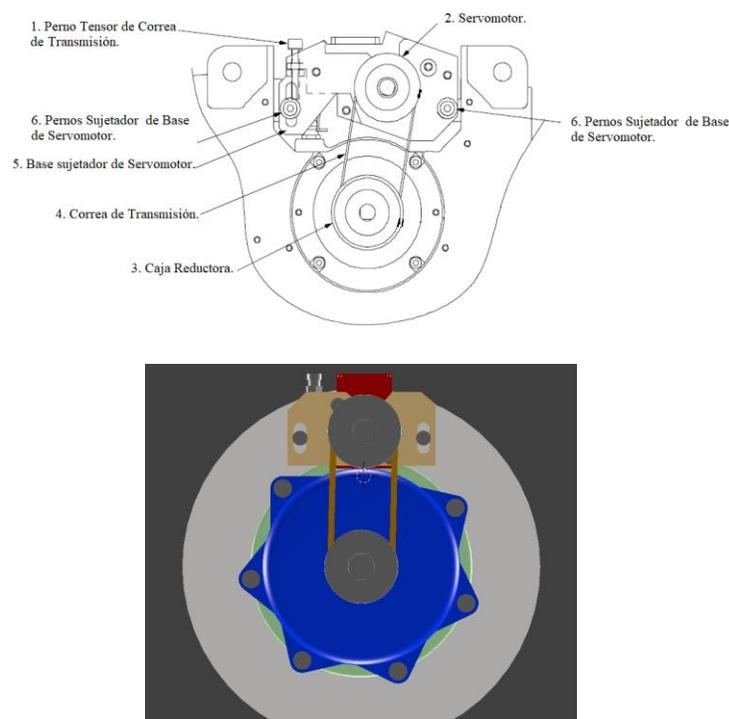


Figura 72. Modelado mecánico eje J1.

Sistema de transmisión Eje J2.

En la figura 73 se muestra la descripción gráfica y su respectivo modelado 3D de los componentes correspondiente al sistema de transmisión del eje J2 y J3, estos ubicados en la parte del Brazo del Robot.

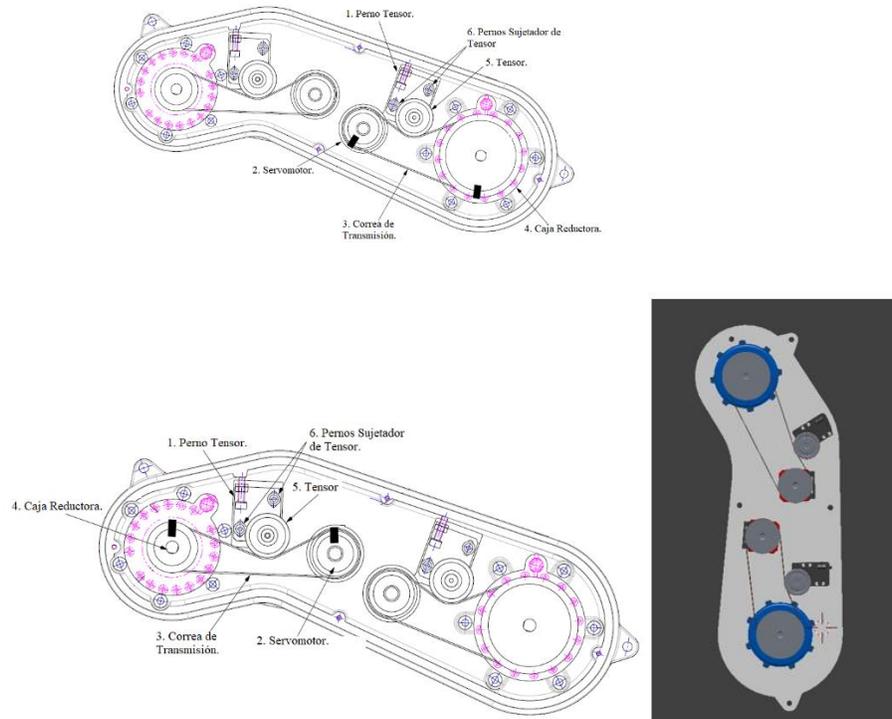


Figura 73. Modelado mecánico eje J2 y J3.

Sistema de transmisión Eje J4.

En la figura 74 se muestra la descripción gráfica y su respectivo modelado 3D de los componentes correspondiente al sistema de transmisión del eje J4, estos ubicados en la parte del Codo del Robot.

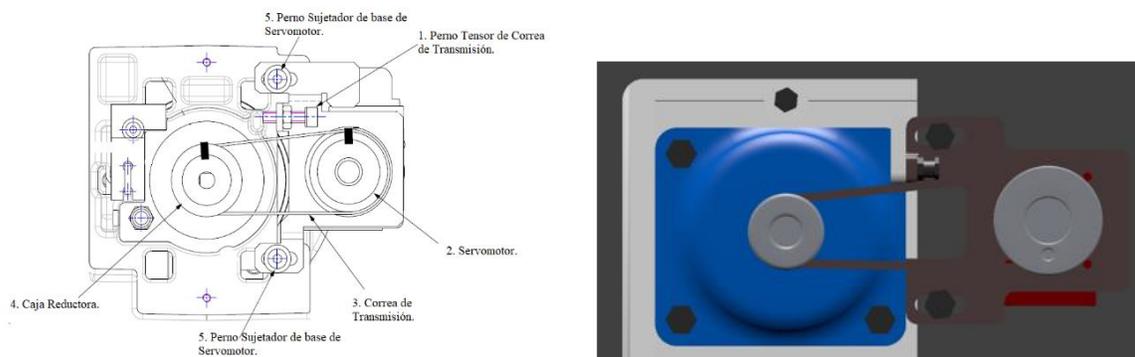


Figura 74. Modelado mecánico eje J4.

Sistema de transmisión Eje J5.

En la figura 75 se muestra la descripción gráfica y su respectivo modelado 3D de los componentes correspondiente al sistema de transmisión del eje J5, estos ubicados en la parte derecha del Antebrazo del Robot.

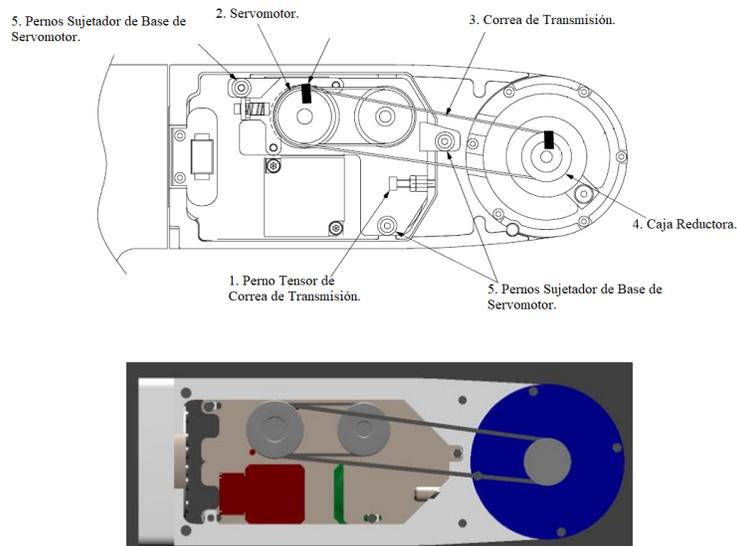


Figura 75. Modelado mecánico eje J5.

Sistema de transmisión Eje J6.

En la figura 76 se muestra una descripción gráfica y su respectivo modelado 3D de los componentes correspondiente al sistema de transmisión del eje J6, estos ubicados en la parte izquierda del Antebrazo del Robot.

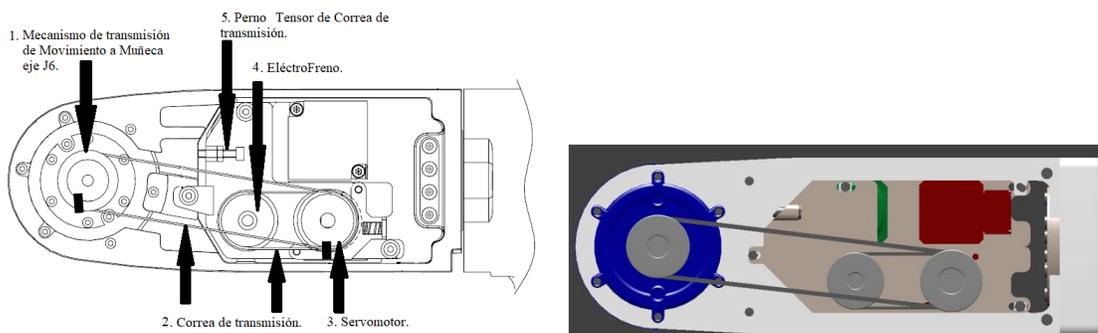


Figura 76. Modelado de Componentes de Mecanismo para eje J6.

Modelado de antebrazo y mano y su implementación.

Para las escenas correspondientes a “Montaje” y “Desmontaje” del aplicativo, es necesario el modelado del miembro superior (antebrazo y mano), por lo tanto, para realizar el modelado tridimensional gráficamente efectiva de manera fácil y rápida se utilizó la técnica disponible en Blender llamada “BoxModelling”. Una de las características que hay que tener en consideración al trabajar un modelo 3D en Blender son sus imágenes de referencia y es que el uso de ellas para el desarrollo del modelado está pensado para trabajar desde puntos de vista concretos (frente, perfil o planta) u ortogonales, por lo tanto, es necesario que se desactive la vista ortográfica presionando la tecla “5” y luego activando cualquiera vista ortogonal antes mencionadas con las teclas “1, 3 y 7” correspondiente a frente, perfil y planta respectivamente.

Para incorporar el boceto que es la imagen de referencia dentro del proyecto en Blender 3D se sigue el siguiente procedimiento:

1. Con la tecla “N o n” se activa la ventana de “Transform”, en esa ventana buscar el apartado “BackGround Images”, este apartado debe ser activado para poder visualizarse los bocetos que se utilizaran.
2. Dentro del apartado “BackGround Images”, buscar la imagen de referencia a utilizar, como se dijo anteriormente el uso de imágenes de referencia permite trabajar en vistas concretas.
3. Para este caso se selecciona la vista frontal con la ayuda del teclado numérico “1”, además dentro de “BackGround Images”, existe un submenú, buscar “Axis” y seleccionar “Front”, con esto se limita que la imagen aparezca únicamente en la vista Fontal como se muestra en la figura 77. Así mismo se realiza con la vista lateral derecha y trasera.



Figura 77. Inicio de modelado de la mano derecha en Blender.

Para tener una mejor idea del modelo 3D a realizar, fue necesario incluir varios bocetos en diferentes vistas, para este caso se hizo uso de la vista frontal, lateral derecha y trasera. Otras de las características que ofrece Blender es la división de ventanas, esto nos permite tener un control de lo que estamos haciendo. A cada una de las ventanas se les asigna una vista diferente, en la figura 78 se muestra la división de la pantalla en sus tres vistas seleccionadas.

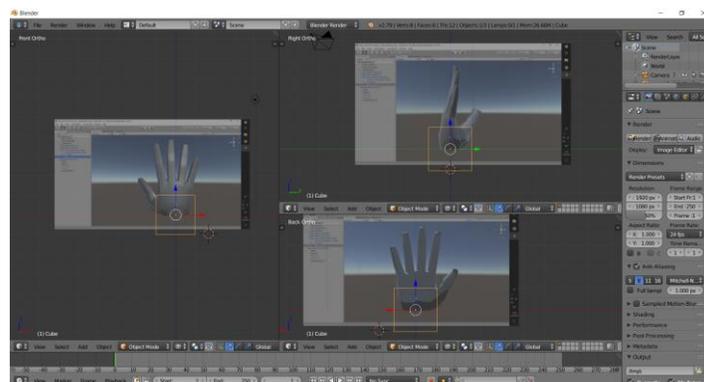


Figura 78. División de la pantalla en sus tres vistas seleccionadas.

Para el desarrollo del modelo de antebrazo y mano se utiliza un cubo como modelo base, luego para ganar forma y detalle se va agregando vértices y caras, para lo cual en modo de edición se utiliza la herramienta “Loop Cut and Slide” con la tecla de acceso rápido “Ctrl + R”, esta

herramienta permite dividir un bucle de caras insertando un nuevo bucle de borde que intersecta el borde seleccionado, al llegar a la parte de los dedos se utiliza la herramienta “Extrude Region” con la tecla de acceso rápido “E” y con la selección de caras, esta herramienta permite generar nuevos vértices manteniendo la nueva geometría conectada a los vértices originales, la figura 79 muestra el resultado final de la base del modelo 3D de la mano derecha.

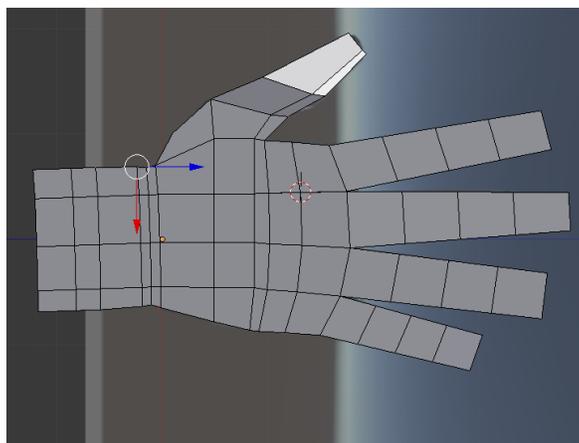


Figura 79. Modelado de mano por la metodología Box Modeling.

A partir del modelo básico, se prosigue a la utilización de modificadores tal como lo hace en (Guevara Bryan, 2018). Los modificadores son operaciones no destructivas que transforman a un objeto en forma y/o topología, que de forma automática permite general efectos que de otra manera serian tediosas de realizar de manera manual (Guevara Bryan, 2018). Uno de los modificadores más utilizados es el denominado “Subdivision Surface”, mismo que permite suavizar superficies poliedras utilizando el algoritmo “Catmull–Clark” dividiendo la superficie del modelo. Para poder utilizar este modificador hay que seleccionar el objeto y desde el menú de “Modifiers”, seleccionar “Add modifiers” y luego seleccionar “Subdivision Surface” tal como se muestra en la figura 80.

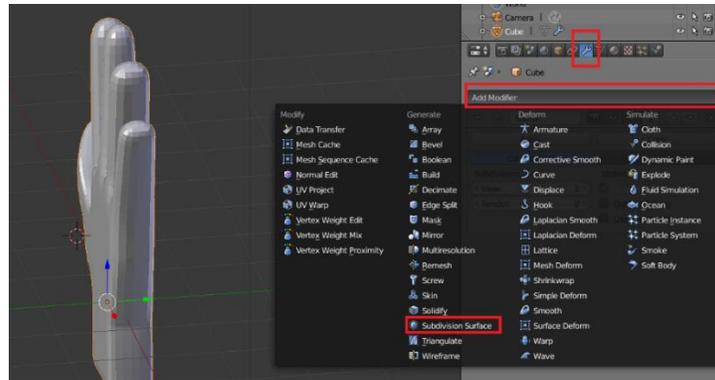


Figura 80. Aplicación de Modificar " Subdivision Surface " al modelo de mano.

Es necesario ir modificando el modelo de forma manual y sin aplicar el modificador para obtener el resultado esperado, hay que tener mayor énfasis en el detalle de la parte de los dedos, nudillos, uñas entre otros detalles; y finalmente se procede a aplicar el modificador, al aplicarlo se generará de forma automática una nueva geometría reemplazando la original, incrementado de forma considerable el número de vértices y superficies del modelo. Una de las características a tener en consideración es la parte de la vista del modelo y el render final, lo que se recomienda es que la parte de visualización sea menor frente a la parte de render para cuestión de rendimiento computacional. En la figura 81 se aprecia el resultado final del modelo de mano mayormente detallado.

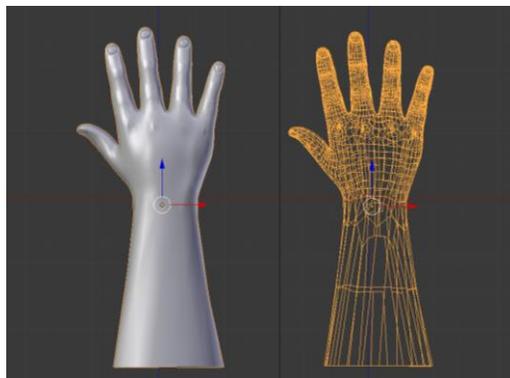


Figura 81. Resultado final del modelado de mano.

Creación de Armadura para el modelo de antebrazo y mano.

La armadura dentro de un modelo tridimensional es considerada como un conjunto de huesos, este puede estar integrada por varias cadenas independientes dentro del modelo generando así una dependencia nula entre una y otra al realizar un movimiento que es el objetivo de una armadura.

Para agregar un hueso dentro de un modelo tridimensional se presiona las teclas *Shift + A* luego se selecciona *Esqueleto* y finalmente *Hueso Individual* como se muestra en la figura 82.

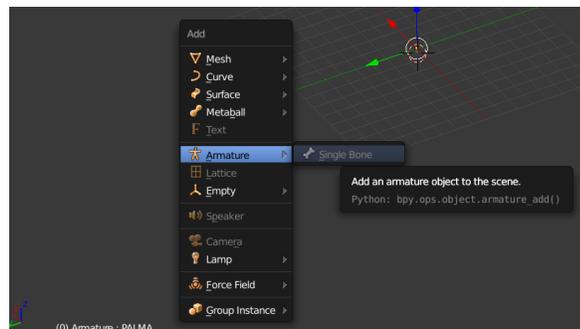


Figura 82. Creación de hueso.

Al realizar la acción anterior aparece un hueso cuyas partes son: Cabeza, Cuerpo y Cola como se muestra en la figura 83, en donde la cabeza es el punto de rotación fija, la cola como indica el final y en varios casos el inicio de otro hueso formando así un esqueleto.

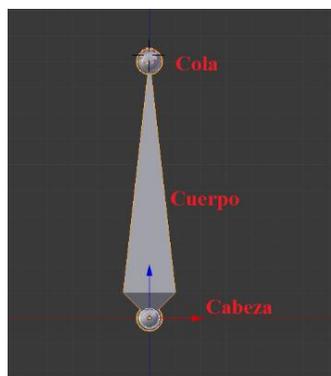


Figura 83. Componentes de un Hueso.

El hueso tiene tres modos: Modo Objeto, Modo Pose y Modo Edición como se muestra en la figura 83; el modo edición podemos agregar más huesos para nuestro modelo y colocándolo en la posición que corresponda para nuestro modelo, en modo pose ya una vez agregada los pesos correspondientes podemos proceder a crear una animación dando los movimientos necesarios.

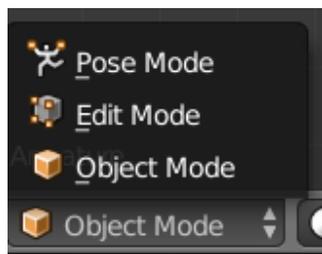


Figura 84. Modos de aplicación para el hueso.

Para agregar más huesos se debe estar en modo *Edit*, luego posicionar en la cola con clic derecho sobre el mismo, y finalmente presionando la tecla de extrusión “E” o sobre cualquier parte de nuestro modelo con las teclas *Shift + A*, como se muestra en la figura 85.

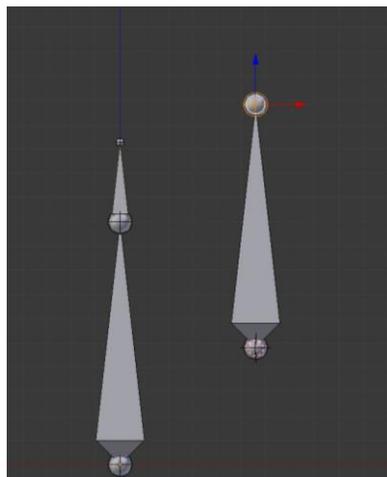


Figura 85. Creación de más de un Hueso.

Para emparejar un hueso que tiene una cierta distancia hay que seleccionar primero el hueso que va a ser hijo y luego el hueso padre, luego se presiona las teclas *Ctrl + P* y seleccionar

KeepOffset que finalmente quedaran unidos dichos huesos. Ahora para el modelo de la mano se tiene lo siguiente mostrado en la figura 86, esta cadena ayudara a simular la apertura y cierre de la mano para utilizarlo dentro del Entorno Virtual creado en el motor de video juegos Unity 3D.

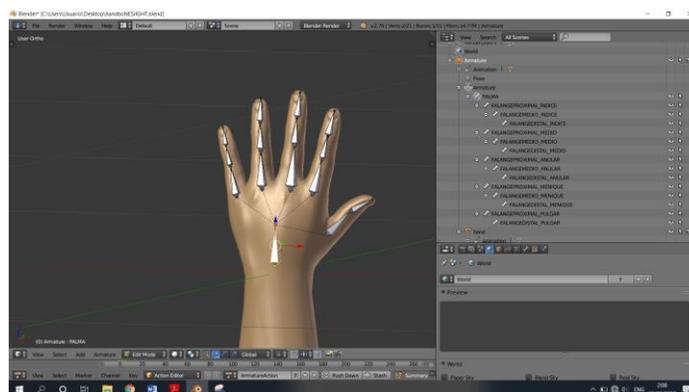


Figura 86. Armadura de Mano.

En modo Objeto se procede a agregar los pesos, para lo cual primero se selecciona la Modelo y luego el Armadura y con las teclas *Ctrl + P* seleccionar en deformación de armadura con Pesos Automáticos como se muestra en la figura 87.

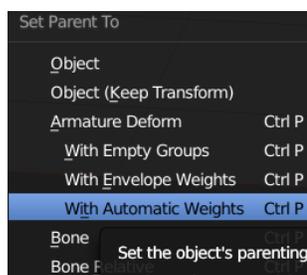


Figura 87. Asignación de pesos.

Para agregar dentro del entorno de desarrollo Unity el modelo 3D del antebrazo y mano tanto derecho como izquierdo, hay que guardar el modelo dentro de la ubicación del proyecto en la carpeta *Assets*. Luego solo se agrega el modelo dentro de la escena que será utilizada, tal como se muestra en la figura 88. Además, se utiliza un script de nombre *HandControl* que permite

determinar la posición y rotación de la palma de la mano del usuario con la ayuda del dispositivo LeapMotion.

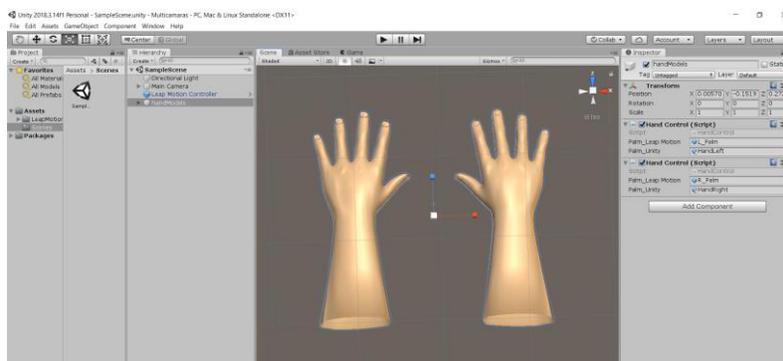


Figura 88. Implementación del modelo 3D de antebrazo y mano hacia Unity 3D.

Luego se crea un script con el nombre de *HandControl*, mismo que permite tomar los valores de posición y rotación del objeto *RigidRoundHand_L*, objeto que pertenece al módulo núcleo de *LeapMotion*, que es agregado a la escena. El objeto *RigidRoundHand_L* no tiene definida su forma física dentro de cualquier escena, y se puede aprovechar para tomar la posición y rotación del mismo con el objetivo de pasar al modelo 3D antes desarrollado, a continuación, en la figura 89 se muestra en código utilizado para representar la posición y rotación de las manos del usuario dentro del entorno Unity.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class HandControl : MonoBehaviour
{
    public GameObject Palm_LeapMotion;
    public GameObject Palm_Unity;
    // Update is called once per frame
    void Update()
    {
        Palm_Unity.transform.position = Palm_LeapMotion.transform.position;
        Palm_Unity.transform.rotation = Palm_LeapMotion.transform.rotation;
    }
}

```

Figura 89. Código utilizado para representar la posición y rotación de las manos del usuario dentro del entorno Unity.

Desarrollo del apartado “Montaje”.

Dentro del menú principal como se explicó en la sección de menús existe un submenú en donde el usuario del aplicativo tiene la capacidad de seleccionar entre realizar la tarea de Montaje o Desmontaje ya sea sobre las piezas en forma general o las otras cinco corresponde a sus partes de forma individual (Base, Brazo, Codo, Antebrazo, y Muñeca). Por lo tanto, se tiene lo siguiente.

1. Modo del aplicativo “Montaje”

Al iniciar cualquier parte de las seis en el menú principal aparece de forma inmediata el apartado de selección entre Montaje y Desmontaje. En la figura 90 se muestra la pantalla de del submenú.



Figura 90. Menú para la tarea de Montaje.

Dentro de cada parte en el modo Montaje se tiene la siguiente distribución mostrados en la figura 91, dicha composición corresponde al apartado de *General*, para las otras no existen las flechas indicadoras. A continuación, se lista los componentes:

1. Imagen de objeto que está en el orden a ser ensamblar.
2. Botón de salida de partida hacia el menú principal.
3. Vista previa de la posición del objeto a ensamblar, visor de cámara pequeña que se renderiza dentro la cámara principal.
4. Información del objeto a ensamblar.
5. Parte a Ensamblar.
6. Piezas principales a ensamblar.

Los objetos 1 y 4 corresponden a un solo Canvas, mientras que el objeto 2 es un Canvas diferente que permite mover en la dirección frontal de la cámara y colocar los botones correspondientes a Saltar y Salir.

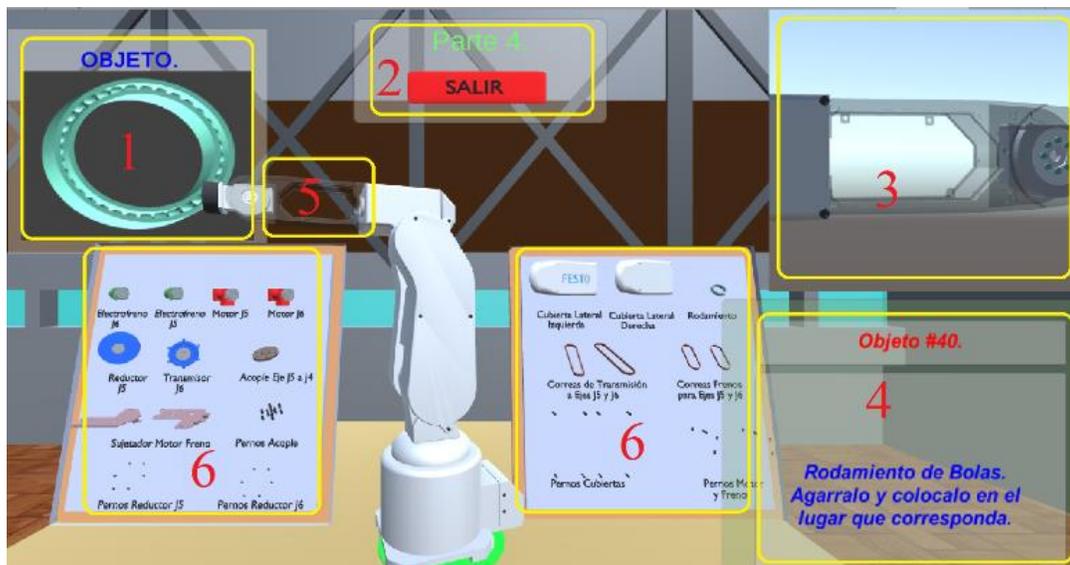


Figura 91. Composición de la escena de Montaje.

El algoritmo de control del ensamble permite al usuario tener una retroalimentación hacia el sentido del tacto cuando esta interactuando con el objeto, encendiendo los micro-motores de vibración ubicados en las yemas de los dedos, tal como se describe en la sección 3.2. El programa mostrado en la figura 92 se describe de la siguiente manera.

El usuario tiene la capacidad de mover sus manos con total libertad dentro de este apartado de la escena. El usuario debe buscar el objeto correspondiente al orden que muestra en la parte superior izquierda con cualquier mano, al estar a una distancia menor a 0,15 (valor determinado experimentalmente) entre la punta del cilindro de la mano y el objeto, el color del objeto cambia el color del render a rojo e inmediatamente se coloca en la punta del cilindro de color verde de cualquier mano ya sea de la mano derecha o izquierda. Los micro-motores de vibración se encienden inmediatamente proporcionando al usuario la retroalimentación de agarrar y estar tocando el objeto, dicho control se lo realiza por medio del puerto COM hacia la tarjeta de arduino enviando el valor de 0, 1, 2 tal como se explicó en la sección 3.2.3.3. En la figura 93 se muestra la secuencia de acción para el agarre de las piezas con la mano.

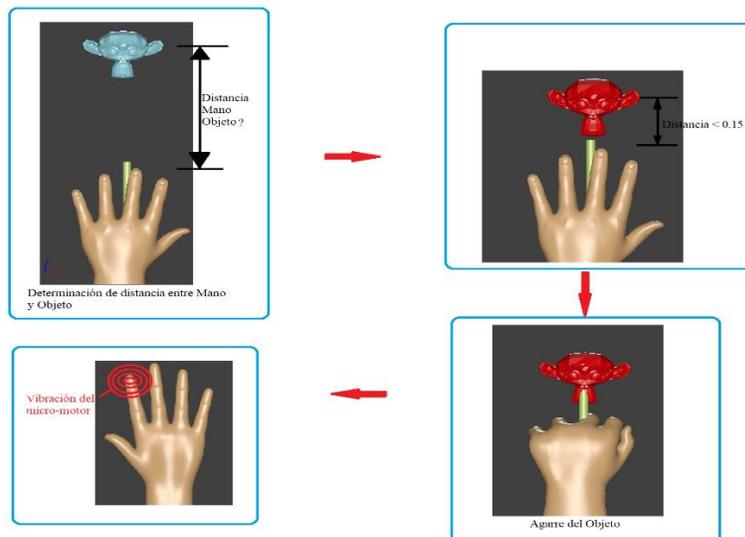


Figura 92. Agarre de Objeto con la mano para Montar o Desmontar

Luego, el usuario debe mover su mano hasta donde se encuentra un doble del objeto que está agarrando mismo que se encuentra desactivado. Luego se compara la distancia entre el objeto activo y el desactivo y al estar a una distancia 0.3 el objeto desactivado se activa y se pone en color rojo. Al estar a una distancia 0.1 entre los dos objetos (El activo y el desactivo) los dos objetos cambian su estado, de estar activo a desactivo y viceversa generando la acción de ensamble, todo eso hasta que termine con todos los objetos que se encuentran dentro de este apartado. La figura 93 mostrada a continuación indica la secuencia de acción que sigue para que se genera las tareas sea de Montaje o Desmontaje dentro del aplicativo.

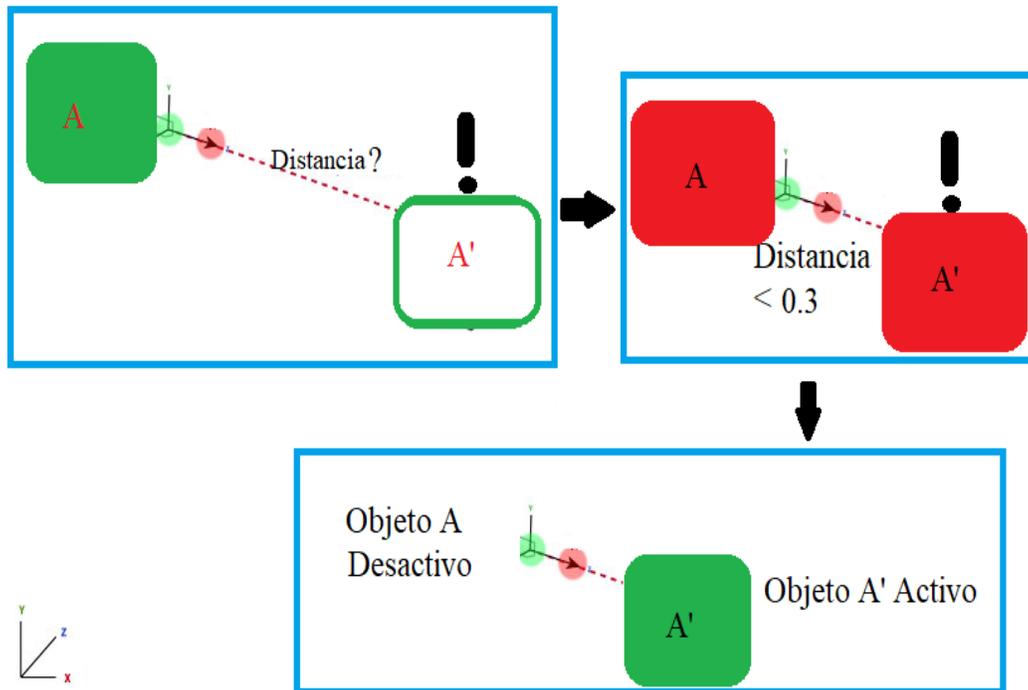


Figura 93. Activación y Desactivación de Objetos para la Aplicación de Montaje y Desmontaje.

En la figura 94 se muestra el código utilizado para la parte del ensamble, destinado a la comparación entre el objeto y los dedos índices de las manos.

```

if (contador < cantidadObj)
{
    distancia = Vector3.Distance(ObjetosA[contador].transform.position, ObjetosA1[contador].transform.position);
    distanciaManoL = Vector3.Distance(ObjetosA1[contador].transform.position, ManoL.transform.position);
    distanciaManoR = Vector3.Distance(ObjetosA1[contador].transform.position, ManoR.transform.position);
    Renderer rendA = ObjetosA[contador].GetComponent<Renderer>();
    Renderer rendA1 = ObjetosA1[contador].GetComponent<Renderer>();
    if (distanciaManoL < 0.04 || distanciaManoR < 0.04)
    {
        rendA1.material = materialTransparente;
        if (distanciaManoL < 0.03)
        {
            if (cierreHL > 0.7)
            {
                rendA1.material = materiales[contador];
                ObjetosA1[contador].transform.position = ManoL.transform.position;
                if (distanciaManoL < 0.025)
                {
                    TextosPasos1[contador].SetActive(false);
                    TextosPasos2[contador].SetActive(true);
                }
                if (controlArduino == true)
                {
                    if (arduino.IsOpen)
                    {
                        arduino.Write("2");
                    }
                }
            }
        }
        if (distanciaManoR < 0.03)
        {
            if (cierreHR > 0.7)
            {
                rendA1.material = materiales[contador];
                ObjetosA1[contador].transform.position = ManoR.transform.position;
                if (distanciaManoR < 0.025)
                {
                    TextosPasos1[contador].SetActive(false);
                    TextosPasos2[contador].SetActive(true);
                }
                if (controlArduino == true)
                {
                    if (arduino.IsOpen)
                    {
                        arduino.Write("1");
                    }
                }
            }
        }
    }
}
else
{
    rendA1.material = materiales[contador];
    TextosPasos1[contador].SetActive(true);
    TextosPasos2[contador].SetActive(false);
    if (controlArduino == true)
    {
        if (arduino.IsOpen)
        {
            arduino.Write("0");
        }
    }
}
}
}
}

```

Figura 94. Algoritmo para el control del montaje en el aplicativo.

Para la parte de la activación del objeto duplicado se tiene la siguiente parte del código que se muestra en la figura 95, en donde, se utiliza una función llamada *Ensamble()*, en dicha función

se activa el objeto doble y desactiva el objeto que el usuario está agarrando, así mismo, activa y desactiva la parte de la información del objeto y su respectiva imagen de referencia.

```

        if (distancia < 0.1)
        {
            rendA.material = materialTransparente;
            rendA.enabled = true;
            if (distancia <= 0.044)
            {
                EnsambleRobot();
            }
        }
        else
        {
            rendA.enabled = false;
        }
    }
}
public void EnsambleRobot()
{
    if (control == false)
    {
        Renderer rendA = ObjetosA[contador].GetComponent<Renderer>();
        rendA.material = materiales[contador];
        rendA.enabled = true;
        ObjetosA1[contador].SetActive(false);
        informacion[contador].SetActive(false);
        Camaras[contador].SetActive(false);
        imagenObjetos[contador].SetActive(false);
        contador += 1;
        fuente.clip = sonidoOk;
        fuente.Play();
        if (contador < cantidadObj)
        {
            ObjetosA[contador].SetActive(true);
            informacion[contador].SetActive(true);
            Camaras[contador].SetActive(true);
            imagenObjetos[contador].SetActive(true);
        }
        else
        {
            control = true;
            if (controlArduino == true)
            {
                arduino.Write("0");
                arduino.DiscardOutBuffer();
                arduino.Close();
            }
            CanvasBotones.SetActive(false);
            Botones.SetActive(false);
            completado.SetActive(true);
            entorno.SetActive(false);
        }
    }
}

```

Figura 95. Función de control de activación y desactivación de las secuencias de piezas.

El sistema presente de Realidad Virtual es modular de tipo bus ya que se puede utilizar el algoritmo mostrado en la figura 94 y 95 las veces que sea necesario en otras escenas, esto es gracias a que dentro de la declaración de las variables se crean siendo arreglos de *GameObjects* en donde

el desarrollador del proyecto puede crear otro entorno e ir colocando los objetos que conforman la escena e ir escalando con el objetivo de mejorar el aplicativo y convertir en una herramienta de ayuda en aprendizaje para equipos mecatrónicos completa. En la figura 96 se muestra la declaracion de variables como arreglos para Unity.

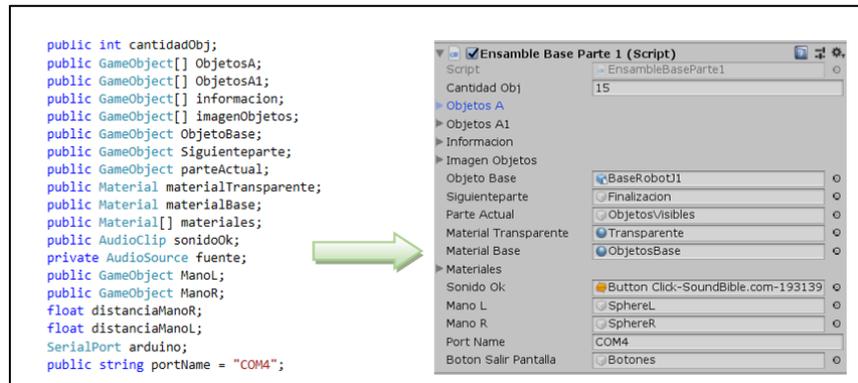


Figura 96. Modularidad del algoritmo de control para Montaje y Desmontaje.

Como se aprecia en la figura 97, el usuario al ir avanzando con el ensamble, la imagen de referencia y su respectivo panel de informacion de la pieza a ensamblar van cambiando de acuerdo al orden establecido del aplicativo.

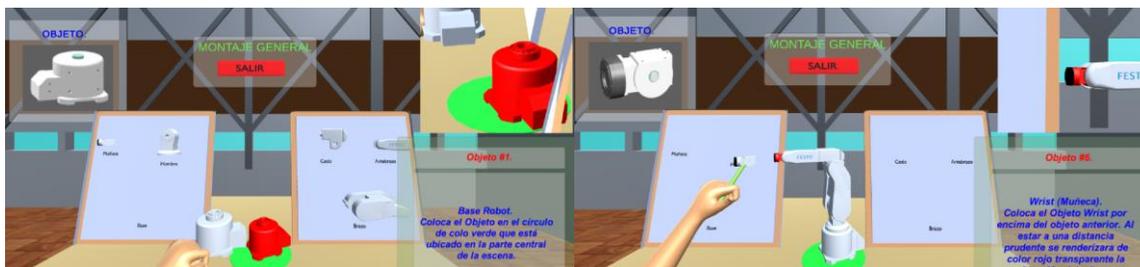


Figura 97. Ejecución del aplicativo montaje.

2. Modo Montaje por partes.

Ahora para la parte del ensamble con las piezas específicas relacionadas al sistema de transmisión del robot se dividirá en cinco partes. En donde, la primera parte corresponde a las

piezas mecánicas de la base y el hombro de robot, la segunda parte corresponde al brazo, la tercera parte al codo, la cuarta parte al antebrazo y la quinta parte la muñeca, como se muestra en la figura 98.

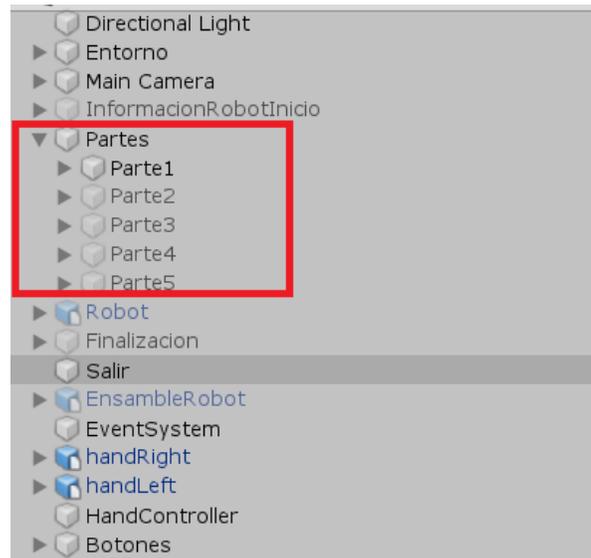


Figura 98. Estructuración para el nivel 2 de la escena Montaje.

A cada una de las partes antes descritas, se le asigna el código del script utilizado en la parte del ensamble general. La escena parte 1 se puede apreciar en la figura 99.

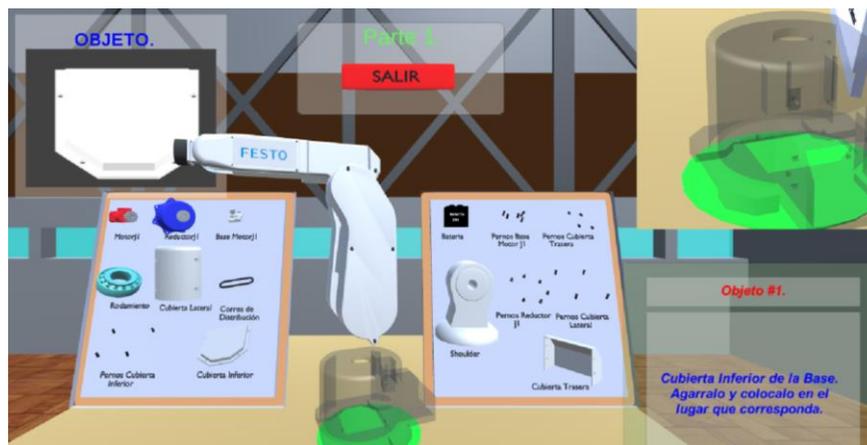


Figura 99. Escena Montaje Parte 1.

Al final de cada parte cuando el usuario haya completado la tarea aparecerá un apartado en donde el usuario puede elegir entre visualizar la animación del funcionamiento del robot o salir hacia el menú principal, tal como se muestra en la figura 100.



Figura 100. Selección de Animación del funcionamiento del Robot.

Animación del funcionamiento de Robot Mitsubishi RV-2SDB.

La animación 3D, es un proceso de creación moviendo o cambio de forma de un objeto sobre una línea de tiempo (Blender, 2019). La animación 3D puede ser desarrollada por diferentes maneras, tales son:

- Transformaciones básicas en sus tres ejes (xyz): cambio de su position, orientación y forma.
- Deformación: Animación de sus vértices o control de puntos.
- Animación inherente: Movimiento del objeto causado por el movimiento de otro objeto, ya sea por su armadura, su padre etc.

Para este caso se utilizó el método de transformación básica en sus tres ejes, tanto para la parte de posición y orientación.

KeyFrames.

Otro concepto dentro de la animación son los Keyframes. Un keyframe es simplemente un punto de tiempo en donde almacena valores de una propiedad. Por ejemplo, un keyframe define que la posición horizontal de un objeto está a 4 metros en el frame 1. Luego el propósito de un Keyframe es permitir la interpolación dentro de la animación, lo que significa que se puede agregar otro punto en el frame 10, especificando que la posición horizontal del cubo está a 20 metros y Blender automáticamente determina la posición correcta para todos los frames entre 1 y 10, claro que depende del método de interpolación antes seleccionada ya sea lineal, cuadrática entre otras (Blender, 2019).

Agregar KeyFrames.

Para poder agregar un keyframe es necesario que este en Modo Objeto. Luego en la barra de menú dirigir a Objeto luego a Animación y dentro de animación encontramos el tipo de keyframe que deseamos agregar. Otra manera más rápida de agregar es por medio del atajo “I” del teclado. Como se observa en la figura 101, existe varios tipos de keyframe, para este caso se utilizó el de tipo *Location and Rotation*, en este tipo de keyframe almacenará las propiedades de cambio tanto de posición y orientación del objeto robot.

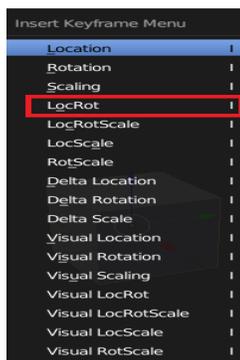


Figura 101. Tipos de Key frames en Blender.

Otro de las características a tener en consideración durante el proceso de animación 3D, es la parte de la línea Keyframe. Esta parte ayuda a visualizar los Keyframes, mismos que están pintados de color amarillo, así mismo permite determinar el número de frames de la animación que para este caso son 590 a una velocidad de 24 fps y de resolución de 1920 pixeles en X y 1080 pixeles en Y, como se observa en la figura 102.

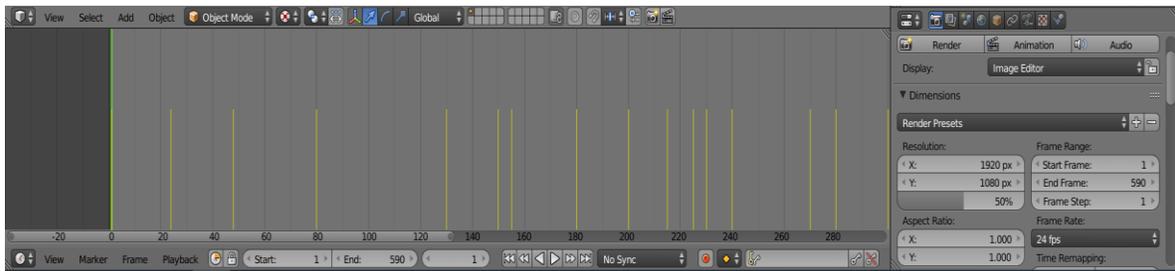


Figura 102. Línea KeyFrame de Blender.

En la figura 103 se muestra al Robot Mitsubishi RV-2SDB con efector final correspondiente a una Ventosa, así mismo, para la animación se agregó otros elementos tridimensionales como, una banda transportadora, dos contenedores, dos objetos cilíndricos (de color rojo y azul) y una maquina dispensadora.

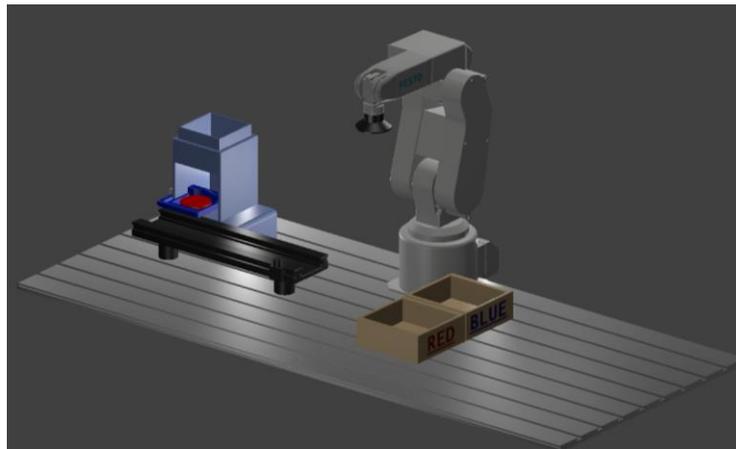


Figura 103. Entorno animado del funcionamiento del Robot Mitsubishi RV-2SDB.

Para que el movimiento dado en el eje J1 de robot correspondiente a rotación afecte a todas las partes del mismo se dispuso dentro del árbol de elementos la herencia como se muestra en la figura 104, siendo como objeto padre la parte de la base del robot, luego el objeto Shoulder objeto hijo de Base pasa a ser objeto padre de No1Arm, así sucesivamente hasta llegar al efector final que es objeto hijo de la parte de la muñeca. Así, al rotar o mover cualquier parte del robot, este afecte a sus correspondientes objetos hijos.



Figura 104. Árbol de elementos hereditarios.

Como base de la animación se tiene al robot con la capacidad de clasificar los objetos cilíndricos, mismos que viajan por medio de una banda transportadora hasta la posición de alcance del robot como se observa en la figura 105. El robot dependiendo del color (rojo o azul) del objeto, lo posiciona en el contenedor adecuado.

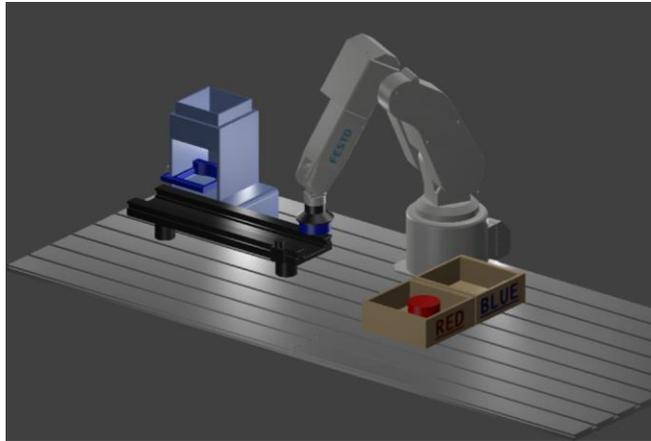


Figura 105. Ejecución de Animación del Robot.

Como resultado final de animación correspondiente a un proceso de clasificación con dependencia a colores se tiene la figura 106, en donde se aprecia que los cilindros se encuentran en el contenedor al que corresponde a su color.

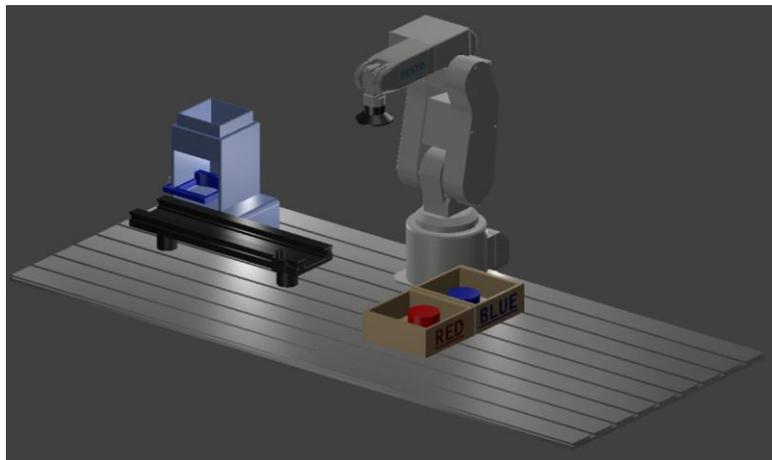


Figura 106. Finalización de animación de clasificación por colores del Robot.

Desarrollo del apartado “Desmontaje”.

Como se describe en el apartado de Desarrollo de Escenas en modo Montaje, la parte de la tarea de Desmontaje se divide en seis apartados, en donde se tiene una parte general y otras cinco referentes a sus partes principales. Por lo tanto, se tiene los siguiente:

1. Modo del aplicativo “Desmontaje”.

Al iniciar cualquier parte de las seis en el menú principal aparece de forma inmediata el apartado de selección entre Montaje y Desmontaje. En la figura 107 se muestra la pantalla de del submenú.

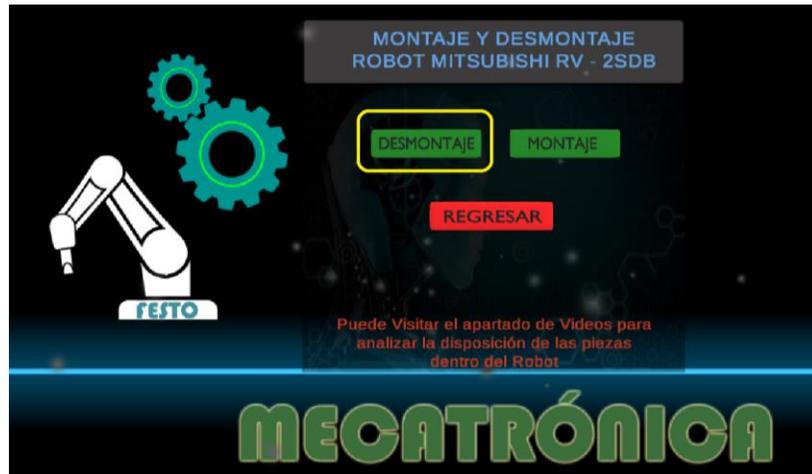


Figura 107. Menú correspondiente a Montaje y Desmontaje.

2. Modo Desmontaje por Partes.

Dentro de la parte del apartado de desmontaje se tiene la siguiente ventana, en donde se tiene los siguientes elementos mostrados en la figura 108, dicha composición la hereda los apartados por parte específicas con la excepción de las flechas indicadoras:

1. Imagen de objeto que está en el orden a ser desmontado.
2. Botón de salida de partida hacia el menú principal.
3. Vista previa de la posición del objeto a desmontar, visor de cámara pequeña que se renderiza dentro la cámara principal.
4. Información del objeto a desmontar.
5. Robot Mitsubishi RV-2SDB a ser desmontado.

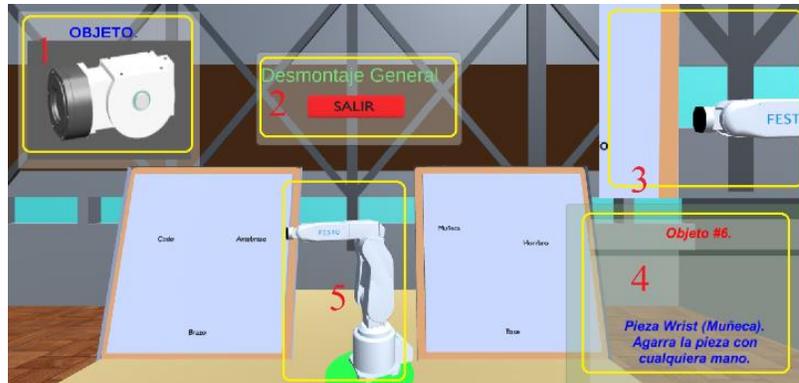


Figura 108. Disposición de la Escena de Desmontaje.

El algoritmo de control de desmontaje es el mismo utilizado en la parte de montaje, la variación es el orden del vector de objetos, como se observa en la figura 109, el vector *ObjetosA*, tiene por *elemento cero* a la muñeca del robot y como *elemento cinco* la base del robot.

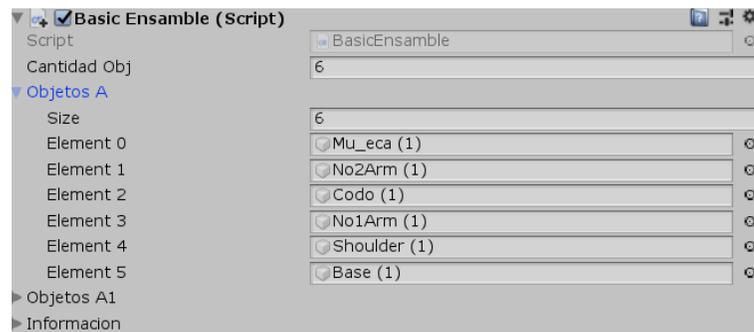


Figura 109. Organización de piezas para el desmontaje.

Así mismo, al finalizar con la tarea de desmontaje, aparece la siguiente ventana, en donde el usuario puede seleccionar entre visualizar la animación del funcionamiento del robot o salir hacia el menú principal.

3.3. Integración del dispositivo háptico con el entorno virtual.

Para entablar comunicación entre el dispositivo háptico y el entorno virtual, se hizo uso de comunicación serial a través del puerto COM del computador. Cuando conectamos el Arduino a

un Ordenador por medio de un cable USB, el ordenador instala un puerto serie virtual (COM viene de “Comunicación”), desde el cual podemos acceder al puerto del Arduino. Para la comunicación entre el ordenador y el Arduino se hace uso de la clase `SerialPort`, dicha clase viene incluida en la namespace `System.IO.Port` destinada para controlar los puertos serie. Para este caso la clase se encarga de mediar lo que se envía desde el entorno hacia el Arduino. Por lo tanto, para hacer uso del puerto serie, es necesario configurar el Nivel de compatibilidad de la API en Unity, que contenga todas las funciones del framework de Microsoft .Net, para lo cual dentro de Unity en *File*, *BuiltSettings*, dentro del apartado de *PlayerSettings* se podrá configurar dicho Nivel de Compatibilidad de la API, que para este caso se seleccione el .NET 4 (Zucconi, 2015).

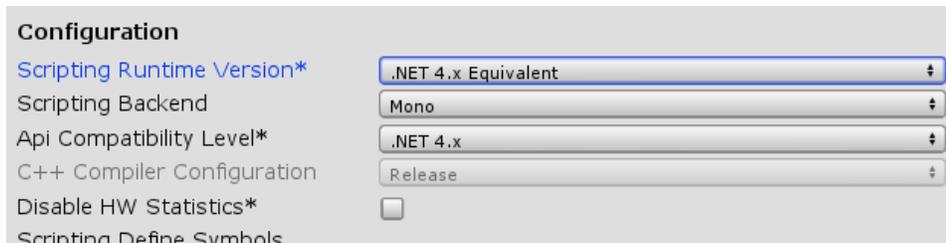


Figura 110. Configuración de Compatibilidad del API para comunicar con el arduino a Unity.

Dentro del script utilizado para montaje y desmontaje se incluye la clase `System.IO.Ports`, tal como se muestra en la figura 111.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.IO.Ports;
```

Figura 111. Integración de la clase `IO.Ports`.

Luego se crea un objeto de la clase `SerialPort` con el nombre de arduino, así mismo, el puerto COM, que para este caso se utiliza en numero 4 como se muestra en la figura 112.

```

SerialPort arduino;
public string portName = "COM4";
void Start()

```

Figura 112. Creación del objeto arduino de la clase SerialPort.

Dentro de la función Start, se define al objeto arduino con su respectivo puerto de comunicación y velocidad de datos de 9600 baudios como se muestra en la figura 113.

```

void Start()
{
    fuente = GetComponent<AudioSource>();
    arduino = new SerialPort(portName, 9600);
    arduino.Open();
}

```

Figura 113. Definición de puerto y la velocidad de comunicación.

Y para el envío de datos que para este caso la activación de los micro-motores de vibración se utiliza la función Write del objeto arduino, en donde se envía un dato entero hacia el arduino como se muestra en la figura 114.

```

{
    arduino.Write("2");
}

```

Figura 114. Escritura hacia el arduino para el control del encendido de los motores.

CAPÍTULO IV

PRUEBAS Y FUNCIONAMIENTO

Para este capítulo se presenta las pruebas de funcionamiento del Sistema Háptico con Realidad Virtual para la manipulación, montaje y desmontaje de piezas de equipos mecatrónicos, como parte de la validación de la hipótesis del proyecto realizaron pruebas tanto para el dispositivo háptico y el sistema de realidad virtual de forma independiente. Las pruebas de funcionamiento de todo el Sistema se realizaron a cuatro estudiantes de diferentes niveles de Ingeniería Mecatrónica

4.1. Pruebas de funcionamiento del dispositivo háptico.

Como parte del Sistema de Percepción háptica el correcto funcionamiento del dispositivo LeapMotion Controller es de suma importancia ya que permite el reconocimiento de gestos de las manos y su representación dentro del entorno Unity. Como parte de la instalación de la SDK de LeapMotion Orion, esta provee la característica de un visualizador para determinar el correcto funcionamiento, entonces se hizo uso del mismo haciendo uso de los guantes. Como se puede observar en la figura 115, LeapMotion Controller reconoce de forma correcta las manos a 120 fps.



Figura 115. Prueba de funcionamiento del dispositivo Leap Motion Controller.

Además, se realizaron pruebas de reconocimiento de múltiples manos colocando sobre el dispositivo electrónico Leap Motion la mano izquierda de cuatro personas de edades entre 24, 10, 5 y 28 años como se observa en la figura 116, dando como resultado la detección de una sola mano izquierda siendo esta la que mejor se visualiza dentro de las cámaras del dispositivo y la primera en ser capturada por el dispositivo.

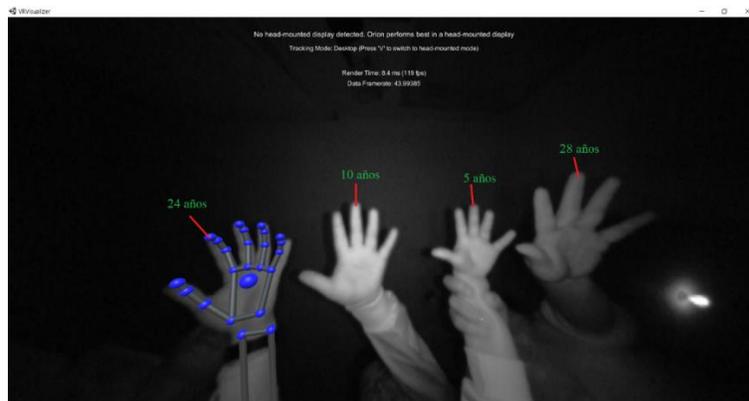


Figura 116. Pruebas con la mano izquierda de cuatro personas.

4.1.1. Funcionamiento de dispositivo Háptico.

Dentro de la IDE de Arduino se crea un pequeño programa en donde se controla de forma manual la activación de los micro-motores de vibración ubicados en la yema del dedo índice en cada mano. En la figura 117 se muestra dos diferentes curvas la una de color roja que corresponde a la activación del micro-motor de la mano izquierda y la otra de color azul correspondiente a la activación del micro-motor de la mano derecha. Al iniciar la comunicación serial con el arduino y la computadora una variable de control de tipo entero va incrementándose, cuando el valor de control este hasta 100 ciclos los micro-motores permanecen apagados, de 100 ciclos a 200 ciclos se activa el micro-motor derecho luego se desactiva de 200 ciclos a 300 ciclos, luego de 300 ciclos a 400 ciclos se activa el micro-motor izquierdo y finalmente se desactiva ambos micro-motores.



Figura 117. Curvas de verificación de activación de los micro-motores de sistema háptico.

En la figura 118 se muestra el programa utilizado para la generación de curvas dentro de la IDE de arduino para determinar el correcto funcionamiento de la conexión de los micro-motores a la tarjeta de control Arduino nano.

```

void loop() {
    control = control+1;
    if(control<100) {
        Serial.print(0);
        Serial.print(",");
        Serial.println(0);
        digitalWrite(pin11,LOW);
        digitalWrite(pin12,LOW);
        digitalWrite(pin13,LOW);
        digitalWrite(pin21,LOW);
        digitalWrite(pin22,LOW);
        digitalWrite(pin23,LOW);
    }
    if(control>100 && control<200) {
        Serial.print(100);
        Serial.print(",");
        Serial.println(0);
        digitalWrite(pin11,HIGH);
        digitalWrite(pin12,HIGH);
        digitalWrite(pin13,HIGH);
        digitalWrite(pin21,LOW);
        digitalWrite(pin22,LOW);
        digitalWrite(pin23,LOW);
    }
    if(control>200 && control<300) {
        Serial.print(0);
        Serial.print(",");
        Serial.println(0);
        digitalWrite(pin11,LOW);
        digitalWrite(pin12,LOW);
        digitalWrite(pin13,LOW);
        digitalWrite(pin21,LOW);
        digitalWrite(pin22,LOW);
        digitalWrite(pin23,LOW);
    }
    if(control>300 && control<400) {
        Serial.print(0);
        Serial.print(",");
        Serial.println(100);
        digitalWrite(pin11,LOW);
        digitalWrite(pin12,LOW);
        digitalWrite(pin13,LOW);
        digitalWrite(pin21,HIGH);
        digitalWrite(pin22,HIGH);
        digitalWrite(pin23,HIGH);
    }
    if(control>400) {
        Serial.print(0);
        Serial.print(",");
        Serial.println(0);
        digitalWrite(pin11,LOW);
        digitalWrite(pin12,LOW);
        digitalWrite(pin13,LOW);
        digitalWrite(pin21,LOW);
        digitalWrite(pin22,LOW);
        digitalWrite(pin23,LOW);
    }
}

```

Figura 118. Código de verificación de conexión entre los micro-motores y el arduino.

Para la evaluación del correcto funcionamiento del dispositivo háptico se realizó un aplicativo sencillo dentro del motor de juego Unity, en donde se crea un objeto 3D de tipo esfera, un plano y se agrega los modulo necesarios para el reconocimiento de las manos. A la esfera se agrega el componente script “Interaction behavior”, un rigidbody 3D y un box collider para la

interacción de con el usuario dentro del aplicativo. Luego se crea un script que contiene una función en donde evalúa si la mano está o no tocando al objeto esfera, misma que se pondrá en color roja si lo está agarrando con la mano derecha, azul si lo está agarrando con la mano izquierda o verde si no existe interacción por parte del usuario. Así mismo para determinar cuál es la mano con la que está tocando el objeto esfera se utiliza el método “Distance”, este método devuelve la distancia entre dos Objetos de tipo flotante, dicho valor se compara con un valor obtenido por experimentación equivalente a 0.15 adimensional. En la figura 119 se muestra los elementos antes descritos utilizados para la verificación del correcto funcionamiento del sistema de percepción háptico.

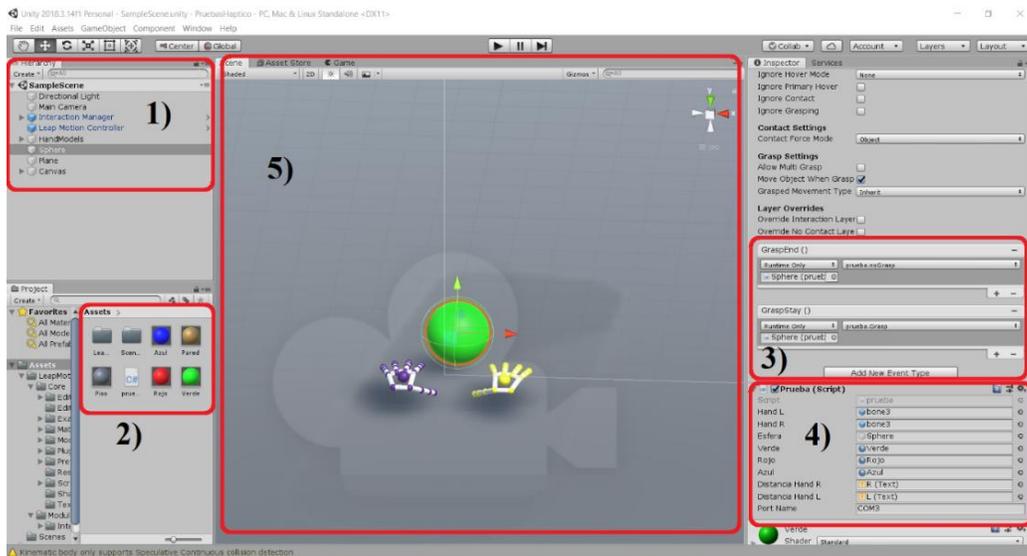
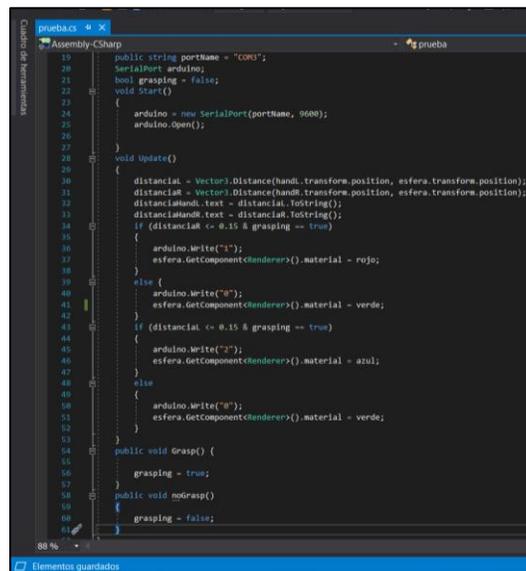


Figura 119. Entorno para la prueba de funcionamiento del Sistema háptico.

En la figura 120 se observa el script “prueba”, utilizado para la verificación del correcto funcionamiento del sistema de percepción háptico, dentro del mismo se puede observar que dentro del a función “void Update”, se controla el encendido y apagado de los micro motores de las dos manos, teniendo las siguientes variables de control:

- **Grasping:** Valor booleano, ayuda con el control de los eventos de GraspStay y GraspEnd, estando en true al estar agarrando el objeto y false al terminar dicha acción, mismo que se encuentra ubicado en dos funciones Grasp y noGrasp respectivamente.
- **DistanciaL:** Correspondiente al valor de distancia entre el objeto esfera y la mano izquierda.
- **DistanciaR:** Correspondiente al valor de distancia entre el objeto esfera y la mano derecha.

Estas variables tienen dependencia “and”, mismas que al cumplir con dicha condición dentro del “if”, envía el valor de activación hacia el arduino, ya sea de encendido de mano derecha o izquierda o de apagado.



```

Assembly-CSharp
public string portName = "COM3";
SerialPort arduino;
bool grasping = false;
void Start()
{
    arduino = new SerialPort(portName, 9600);
    arduino.Open();
}
void Update()
{
    distanciaL = Vector.Distance(handL.transform.position, esfera.transform.position);
    distanciaR = Vector.Distance(handR.transform.position, esfera.transform.position);
    distanciaHandL.text = distanciaL.ToString();
    distanciaHandR.text = distanciaR.ToString();
    if (distanciaL <= 0.15 & grasping == true)
    {
        arduino.Write("1");
        esfera.GetComponent<Chanderer>().material = rojo;
    }
    else {
        arduino.Write("0");
        esfera.GetComponent<Chanderer>().material = verde;
    }
    if (distanciaR <= 0.15 & grasping == true)
    {
        arduino.Write("2");
        esfera.GetComponent<Chanderer>().material = azul;
    }
    else {
        arduino.Write("0");
        esfera.GetComponent<Chanderer>().material = verde;
    }
}
public void Grasp() {
    grasping = true;
}
public void noGrasp()
{
    grasping = false;
}

```

Figura 120. Algoritmo de control para la prueba de funcionamiento del Sistema Háptico.

En la figura 121 apartado a), se observa que el script “Interaction Behavior” se agrega los eventos de GraspStay y GraspEnd, y luego se agrega los métodos creados dentro del script prueba llamados Grasp y noGrasp respectivamente.

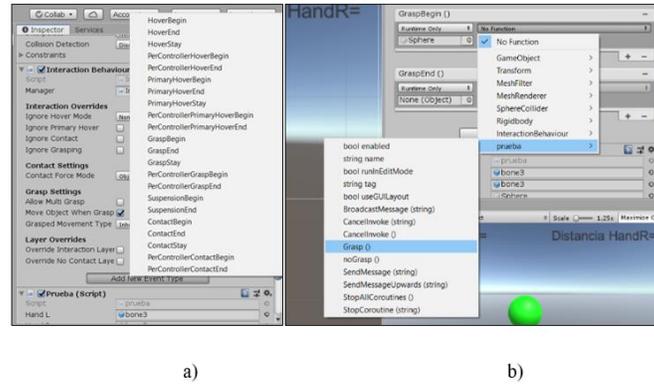


Figura 121. Asignación del script “Interaction Behavior” al objeto esfera.

a) Agregar los Eventos de Grasp. b) Funciones de activación de motores.

Como se puede apreciar en la figura 122, el objeto esfera dentro de Unity se encuentra en color verde cuando no existe interacción por parte del usuario, además envía el valor de 0 hacia el arduino, cuyo valor corresponde al estado apagado de los micro motores de las dos manos.



Figura 122. Inicio de prueba de funcionamiento del Sistema Háptico.

En la figura 123 apartado a) se observa que la distancia entre la mano derecha y la esfera es menor a 0,15, además que el usuario tiene la interacción de Agarre (GrpasStay), como se explicó anteriormente, cuando la distancia es menor a 0,15 y está en interacción con el objeto esfera, el objeto esfera cambia de color, para este caso siendo la mano derecha, la esfera se pone en rojo y envía hacia el arduino el valor de 1, cuyo valor corresponde al micro motor de la mano derecha.

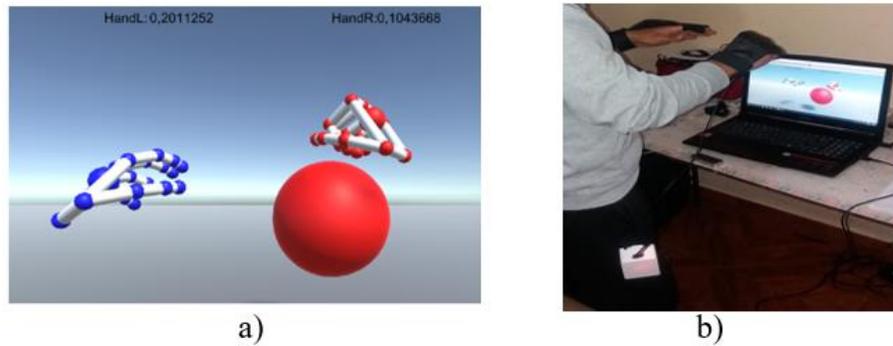


Figura 123. Interacción del usuario con el objeto esfera.

a) Agarre del objeto con la mano Derecha y el cambio de color del objeto esfera. b) Espacio físico.

Para el caso de la mano izquierda como se observa en la figura 124 apartado a) se muestra que la distancia es menor a 0,15 y está en interacción con el objeto esfera, el objeto esfera se pone en color azul y envía el valor de 2 hacia el arduino, cuyo valor corresponde al micro motor vibrador de la mano izquierda.

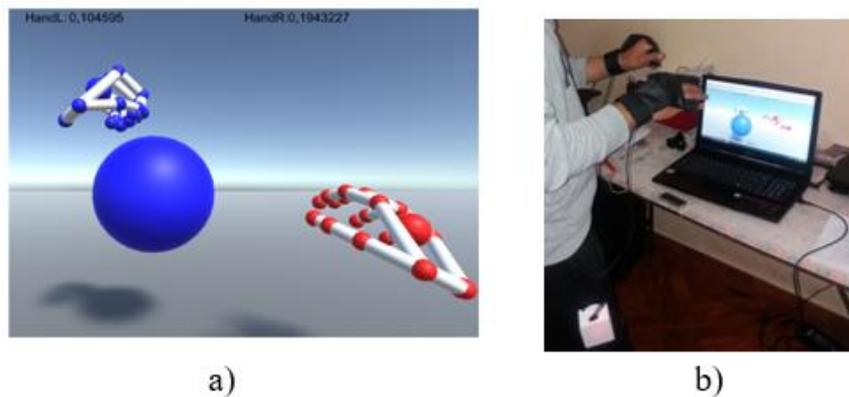


Figura 124. Interacción del usuario con el objeto esfera.

a) Agarre del objeto con la mano Izquierda y el cambio de color del objeto esfera. b) Espacio físico.

4.2. Pruebas de funcionamiento del sistema del VR.

4.2.1. Funcionamiento de la coherencia de los Botones del menú principal.

Para determinar el correcto funcionamiento del Sistema de Realidad Virtual, como primera instancia se realiza las pruebas de coherencia de los botones correspondientes al menú principal. En la figura 125 se muestra los cuatro botones del menú principal, el primer botón conduce hacia las escenas de la aplicación de Sistema para el Montaje y Desmontaje de las piezas Mecánicas correspondientes al Robot Mitsubishi RV-2SDB respectivamente, el segundo botón conduce hacia Videos de Montaje y Desmontaje como parte de ayuda para el usuario, y el tercer botón finaliza la ejecución de la aplicación.

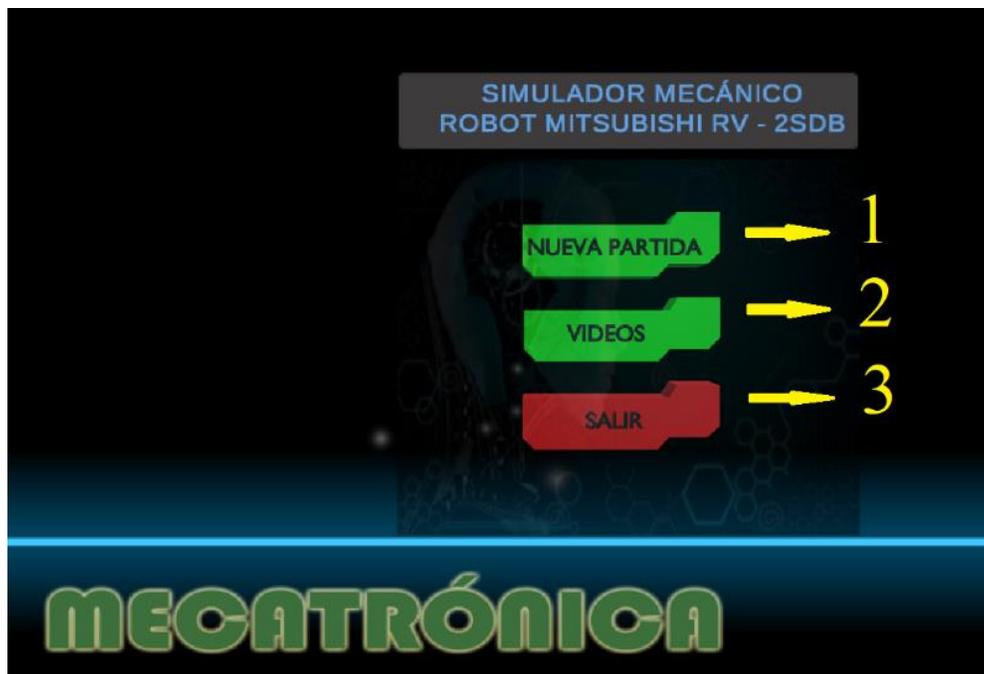


Figura 125. Pruebas de coherencia para el menú principal.

Por lo tanto, para determinar el correcto funcionamiento se hizo uso del aplicativo. Como primera parte se tiene el botón de Montaje y Demontaje, mismo que ser presionado, inmediatamente conduce hacia la parte de los tips como se muestra en la figura 126.

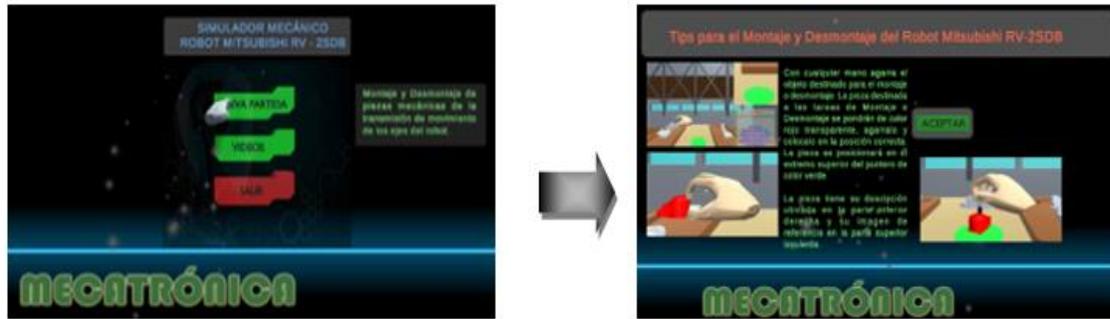


Figura 126. Funcionamiento botón Videos.

Para el segundo botón, correspondiente al botón de Videos, de la misma manera al ser presionado, de manera inmediata conduce hacia un submenú para visualizar una demostración de la posición de las piezas mecánicas del robot, mismo que ayudara al usuario dentro de las escenas de Montaje y Desmontaje. En la figura 127 se muestra el cambio de escena.

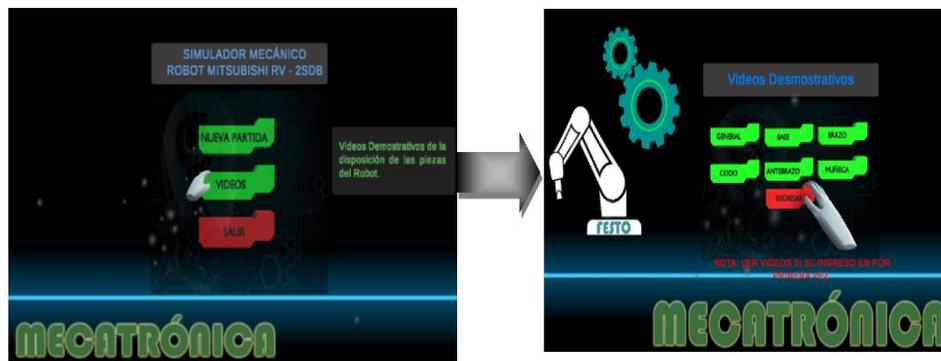


Figura 127. Funcionamiento botón Videos.

Para el tercer botón, correspondiente al botón de Salir, de la misma manera al ser presionado de manera inmediata finaliza la sesión de la aplicación, como se muestra en la figura 128.

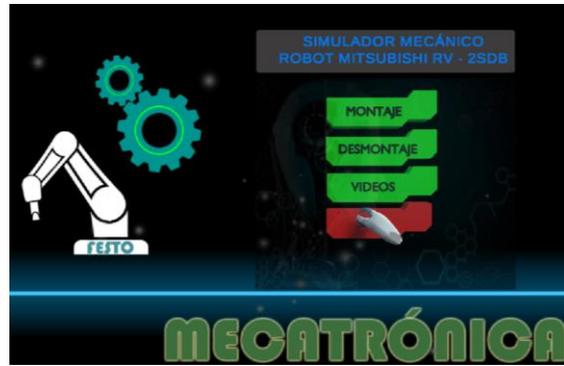


Figura 128. Funcionamiento botón Salir.

4.2.2. Funcionamiento del algoritmo para las tareas de Montaje y Desmontaje del Robot.

Otro de los puntos importantes a evaluar el correcto funcionamiento del aplicativo es la evaluación del funcionamiento de los modos Montaje y Desmontaje. Para lo cual, de la misma manera se hizo uso de la aplicación.

Dentro de la primera parte se tiene el modo de Montaje, en donde el usuario una vez aceptado todos Tips de ayuda y ha iniciado una de las partes dentro del menú de montaje, se procede al realizar la tare seleccionada, para lo cual, el usuario debe buscar la pieza a ensamblar, con el nombre descrito en la parte inferior derecha, que para este caso es la Base Robot, misma que pondrá de color rojo, como se muestra en la figura 129, por lo tanto, el algoritmo está funcionando correctamente.

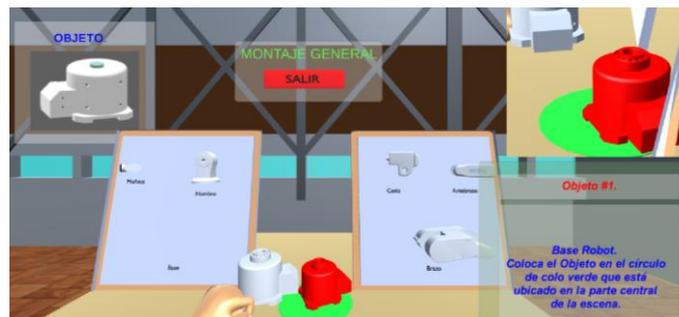


Figura 129. Búsqueda de la pieza a montar.

Luego, se agarra el objeto y se posiciona donde se especifica la flecha de color roja transparente, y de forma inmediata existe cambios tanto la parte de la descripción del objeto ubicada en la parte inferior derecha, su imagen representativa ubicada en la parte superior izquierda y la vista de cámara secundaria ubicada en la parte superior derecha, como se muestra en la figura 130.

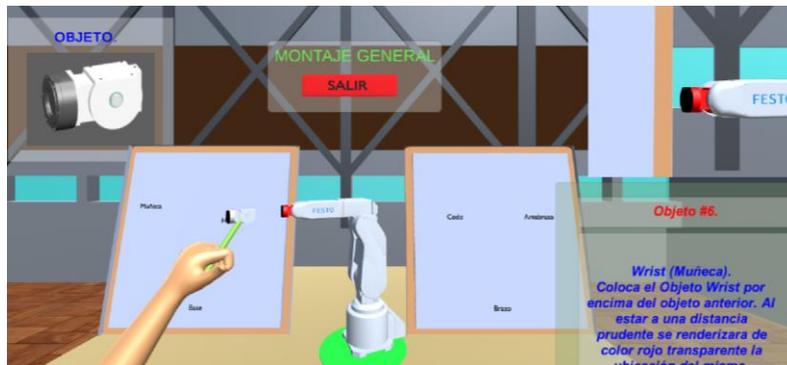


Figura 130. Funcionamiento del avance de la partida para el Montaje.

La otra parte corresponde al modo de Desmontaje, cuyo algoritmo es el mismo que se utilizo en la parte de Montaje, por lo tanto, el usuario debe determinar el objeto a Desmontar, mismo que se pondra de color rojo, el usuario lo agarra y lo posiciona donde la flecha lo indica, como se muestra en la figura 131.

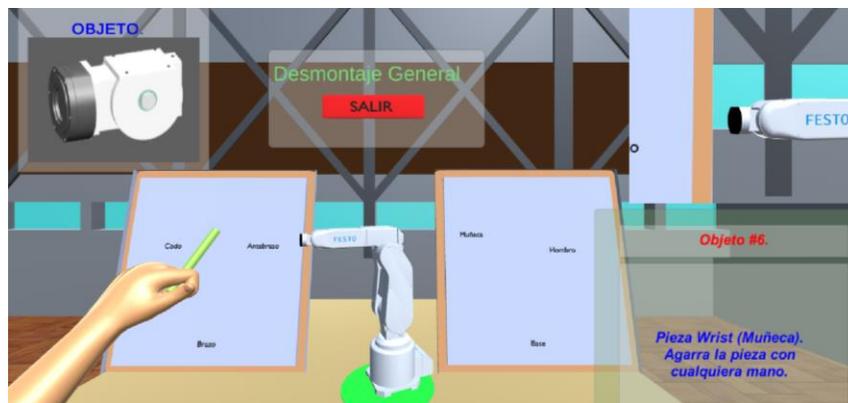


Figura 131. Inicio de partida para desmontaje.

De la misma manera, al continuar con el proceso, dentro de la escena existe cambios en las pantallas de ayuda, tanto de la parte de la imagen de referencia de la pieza, la vista de la cámara secundaria y la descripción de la misma, como se muestra en la figura 132.

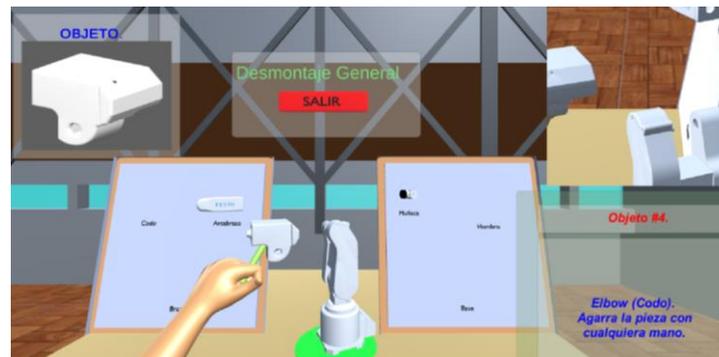


Figura 132. Funcionamiento del avance de la partida Desmontaje.

Por lo tanto, se puede concluir que el funcionamiento del aplicativo de Realidad Virtual desarrollado dentro del motor de juegos Unity es correcto. Cabe recalcar que el algoritmo utilizado es el mismo tanto en la parte general del Montaje y Desmontaje como en las demás partes en donde denota las piezas del robot de forma más específica.

4.3. Pruebas de funcionamiento con estudiantes de ingeniería mecatrónica.

Para la evaluación del funcionamiento del Sistema Háptico con Realidad Virtual para la Manipulación, Montaje y Desmontaje de piezas de Equipos Mecatrónicos se realizó las pruebas a cuatro estudiantes de niveles inferiores a sexto nivel de Ingeniería Mecatrónica, tomando como métrica el tiempo que le toma en cada una de las etapas tanto en la parte del montaje como del desmontaje para validar la hipótesis. Además, a cada uno de los estudiantes que hayan hecho uso del aplicativo, se le aplicó una ficha de evaluación como objeto de determinar el impacto y la usabilidad que genera el aplicativo al usuario. Para el desarrollo de la ficha se tomó en consideración las métricas de prueba de usabilidad basadas en la norma ISO/IEC 9126-4 (ISO

25000) (Anexo3) para evaluación de software y el cuestionario SUS (System Usability Scale), para lo cual se tiene en primera instancia el cuestionario que consta de 5 preguntas como se muestra en la figura 133.

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE SEDE LATACUNGA

Ficha de evaluación para el Sistema Háptico con Realidad Virtual para la Manipulación, Montaje y Desmontaje de Piezas de Equipos Mecatrónicos.

Ítem	Pregunta	Observaciones
1	¿El entorno Virtual es intuitivo y amigable con el usuario?	
2	¿Considera usted que la interactividad del usuario con la aplicación es adecuada para la realización de la tarea de montaje y desmontaje?	
3	¿Considera que la aplicación proporciona conocimiento de los componentes mecánicos de un Robot.?	
4	¿Cree usted que está capacitado como para desarrollar el trabajo montaje y desmontaje al robot Mitsubishi RV-2SDB?	

Calificación del Sistema háptico de 0 al 5, siendo 0 totalmente insatisfecho y 5 totalmente satisfecho.

¿Considera que el sistema de percepción háptico dentro del entorno virtual proporciona una mayor inmersión?

Calificación:

Figura 133. Encuesta para los Usuarios del Aplicativo.

Luego con la ayuda de la normativa ISO/IEC 9126-4 (ISO 25000) (Anexo 3) se podrá evaluar y validar la hipótesis del proyecto mismo que se encuentra en el anexo 3. La definición de la hipótesis es la siguiente:

¿El diseño del sistema háptico con realidad virtual, permitirá mejorar la habilidad de montar y desmontar piezas de equipos mecatrónicos?

Se establece la hipótesis alternativa y la hipótesis nula:

- **Hipótesis alternativa:** El diseño del sistema háptico con realidad virtual, permite mejorar la habilidad de montar y desmontar piezas de equipos mecatrónicos.

- **Hipótesis nula:** El diseño del sistema háptico con realidad virtual, no permite mejorar la habilidad de montar y desmontar piezas de equipos mecatrónicos.

Los criterios de evaluación del sistema son:

- Facilidad de uso por parte del usuario.
- Niveles de satisfacción.
- Información legible y entendible para el usuario.
- Métricas de productividad.

La métrica de productividad nos permite evaluar los recursos que los usuarios más consumen según la normativa ISO/IEC 9126 (ISO 25000) (Anexo 3) apartado 4, para este caso se toma el tiempo para completar las tareas de Montaje y Desmontaje en sus correspondientes variantes. Las métricas a evaluar son:

- Tiempo de la tarea.
- La eficiencia en la tarea.

Características de toma de datos y su validación:

1. Variable: tiempo de respuesta.
2. Formula de aplicación: $X = \text{tiempo}$.
3. Interpretación: Entre más corto mejor.

En la misma ficha se tiene un apartado en donde se mide el tiempo de cada usuario en las diferentes partes del robot tanto para las tareas de montaje y desmontaje. En la figura 134 se muestra dicho apartado en forma de matriz.

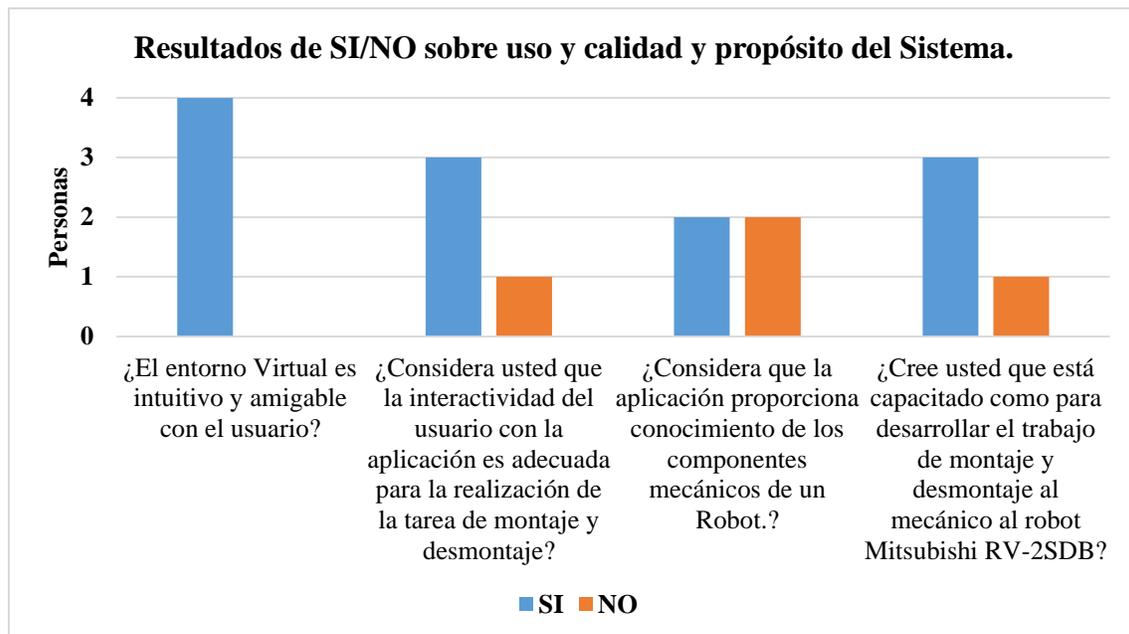
Tiempo utilizado en cada tarea.

Día 1	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Montaje					
Desmontaje					
Día 2	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Montaje					
Desmontaje					
Día 3	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Montaje					
Desmontaje					
Día 4	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Montaje					
Desmontaje					

Figura 134. Matriz de toma de tiempo.

4.3.1. Análisis de Resultados.

Dentro de la ficha de evaluación (Figura 133) correspondiente al aplicativo, se tiene cuatro preguntas que tienen por objeto, estimar la aceptación del Sistema por parte de los usuarios, así como la calidad de su interfaz y el propósito de sus tareas (Montaje y desmontaje). En la figura 135 se muestra los resultados obtenidos por los cuatro usuarios en las cuatro preguntas cerradas de SI y NO.

*Figura 135.* Resultados de ficha de evaluación del uso del aplicativo.

Con los datos proporcionados en la figura se puede indicar que el sistema proporciona cierto grado de aceptación ya que a la mayoría de las preguntas dan un resultado sobresaliente del Si en relación al NO. La pregunta 1 correspondiente a la pregunta de **¿El entorno Virtual es intuitivo y amigable con el usuario?**, se tiene que de los cuatro usuarios los cuatro tiene la subjetividad de responder que les ha parecido intuitivo el sistema en general, pero la pregunta 3 correspondiente a **¿Considera que la aplicación proporciona conocimiento de los componentes mecánicos de un Robot?**, se tiene que el 50 por ciento dice Si y el otro No, por lo tanto el sistema tiende a desarrollar mejoras en este aspecto de información para el usuario. Y dentro de las preguntas 2 y 4 sobresale el SI, lo que ayuda a determinar que la interactividad del usuario con el aplicativo es adecuada.

Para evaluar el Sistema de Percepción Háptica se desarrolló una ponderación de satisfacción de usabilidad como se muestra en la figura, para lo cual se hizo una evaluación que va de 0 al 5 siendo 0 valor que determina la insatisfacción total y 5 la satisfacción total. En la figura 136 se muestra los resultados obtenidos por parte de los cuatro usuarios.

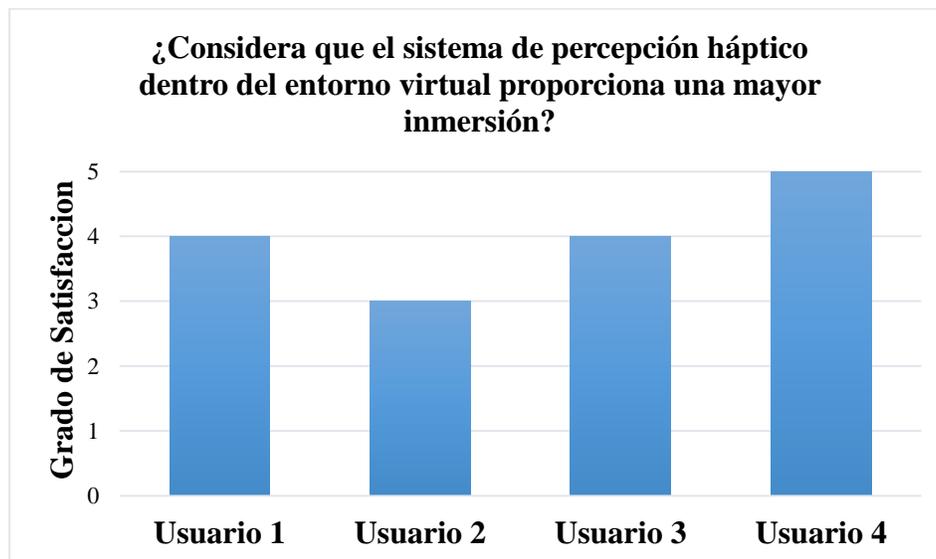


Figura 136. Evaluación del Sistema de percepción háptica.

En la figura 136 se muestra los resultados de la pregunta de evaluación correspondiente al sistema de percepción háptica en donde indica que el valor mínimo de satisfacción es 3 y el mayor 5, así mismo se tiene una media igual a 4, valor considerado un nivel de satisfacción adecuado para el aplicativo de Realidad Virtual.

Para determinar la métrica de productividad se realizaron pruebas de uso del sistema tomando como referencia el tiempo en cada una de las partes tanto para cada modo de juego (Montaje y Desmontaje) por cuatro ocasiones al mismo usuario.

Modo aplicativo Montaje.

La tabla 23 muestra los resultados de tiempo medido al usuario uno, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 23.
Resultados del usuario uno en modo Montaje.

Usuario 1					
Montaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	1,34	1,5	1,1	2,21	0,57
Día 2	1,25	1,4	1,1	2,21	0,57
Día 3	1,15	1,5	1,1	2,21	0,45
Día 4	1	1,5	0,58	2	0,57

La tabla 24 muestra los resultados de tiempo medido al usuario dos, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 24.
Resultados del usuario dos en modo Montaje.

Usuario 2					
Montaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	1,56	2,41	1,58	3,56	1,21
Día 2	1,45	2,35	1,47	3,54	1,2
Día 3	1,49	2,41	1,54	3,45	1,28
Día 4	1,38	2,36	1,42	3,45	1,12

La tabla 25 muestra los resultados de tiempo medido al usuario tres, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 25.
Resultados del usuario tres en modo Montaje.

Usuario 3					
Montaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	2,12	2,56	1,45	3,25	1,24
Día 2	2,08	2,45	1,56	3,36	1,3
Día 3	1,58	2,36	1,39	3,21	1,21
Día 4	1,47	2,5	1,4	3,18	1,15

La tabla 26 muestra los resultados de tiempo medido al usuario cuatro, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 26.
Resultados del usuario cuatro en modo Montaje.

Usuario 4					
Montaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5

CONTINÚA 

Día 1	2,23	3,56	2,12	4,45	2,25
Día 2	2,15	3,45	2,15	5,2	2,15
Día 3	2,2	3,47	2,2	4,56	2,2
Día 4	2,04	3,36	2,03	4,4	2,05

Modo aplicativo Desmontaje.

La tabla 27 muestra los resultados de tiempo medido al usuario uno, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 27.
Resultados del usuario uno en modo Desmontaje.

Usuario 1					
Desmontaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	0,58	2,42	1,27	2,34	1,51
Día 2	0,47	2,36	1,21	2,21	1,42
Día 3	1,1	2,43	1,18	2,21	1,35
Día 4	0,56	2,34	1,21	2,12	1,3

La tabla 28 muestra los resultados de tiempo medido al usuario dos, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 28.
Resultados del usuario dos en modo Desmontaje.

Usuario 2					
Desmontaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	1,34	2,45	1,62	2,12	1,14
Día 2	1,25	2,36	1,32	2,21	1,25

CONTINÚA 

Día 3	1,15	2,42	1,36	2,1	1,12
Día 4	1,1	2,3	1,25	2	1,04

La tabla 29 muestra los resultados de tiempo medido al usuario tres, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 29.
Resultados del usuario tres en modo Desmontaje.

Usuario 3					
Desmontaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	1,25	2,6	1,25	2,45	1,3
Día 2	1,34	2,4	1,14	2,2	1,21
Día 3	1,05	2,68	1,1	2,25	1,32
Día 4	1	2,5	1,14	2,15	1,15

La tabla 30 muestra los resultados de tiempo medido al usuario cuatro, en donde indica los tiempos correspondientes a cada día y de cada parte del robot.

Tabla 30.
Resultados del usuario cuatro en modo Desmontaje.

Usuario 4					
Desmontaje	Parte 1	Parte 2	Parte 3	Parte 4	Parte 5
Día 1	1,34	3,45	2,25	3,24	2,12
Día 2	1,45	3,25	2,34	3,15	2,15
Día 3	1,15	3,2	2,15	2,54	2,19
Día 4	1,05	3,25	2,1	3	2,14

4.4. Validación de la hipótesis.

Para determinar que los usuarios del aplicativo han mejorado su habilidad en los procesos de Montaje y Desmontaje del equipo mecatrónico se tiene los datos de medida de tiempo a cada usuario.

Montaje.

En la figura 137 muestra las curvas de los resultados obtenidos de la tabla 22 del usuario uno, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

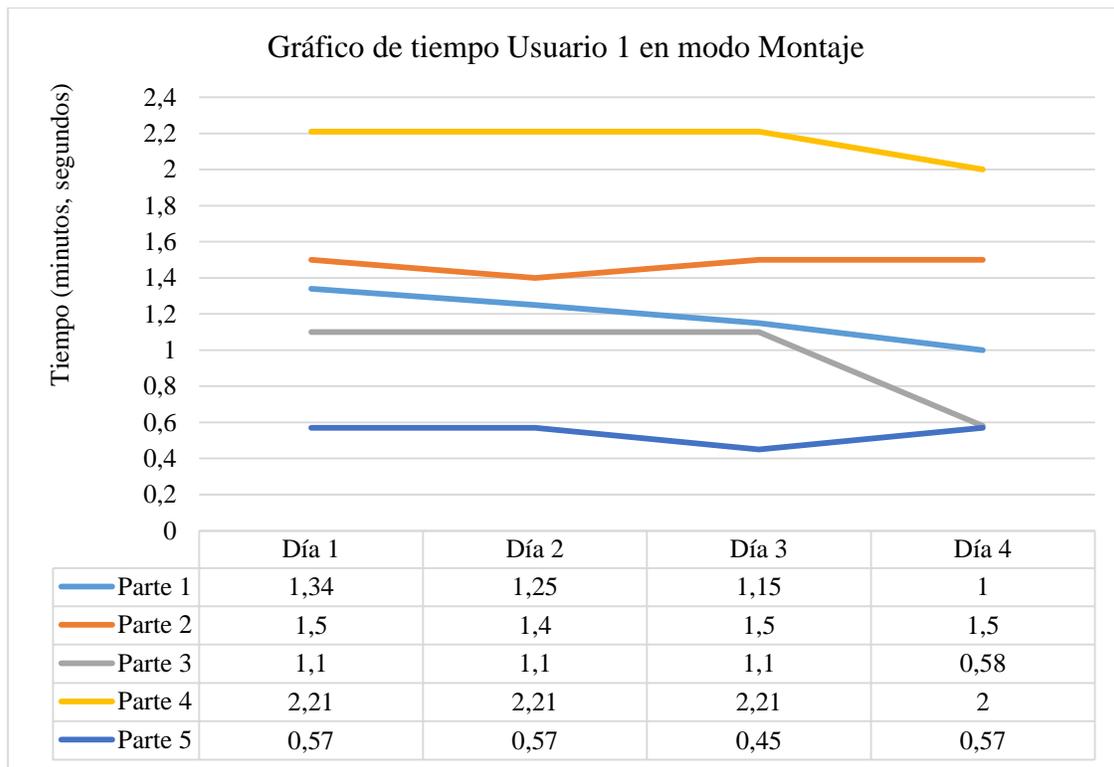


Figura 137. Resultados del usuario uno en modo Montaje.

En la figura 138 se muestra las curvas de los resultados obtenidos de la tabla 23 del usuario dos, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

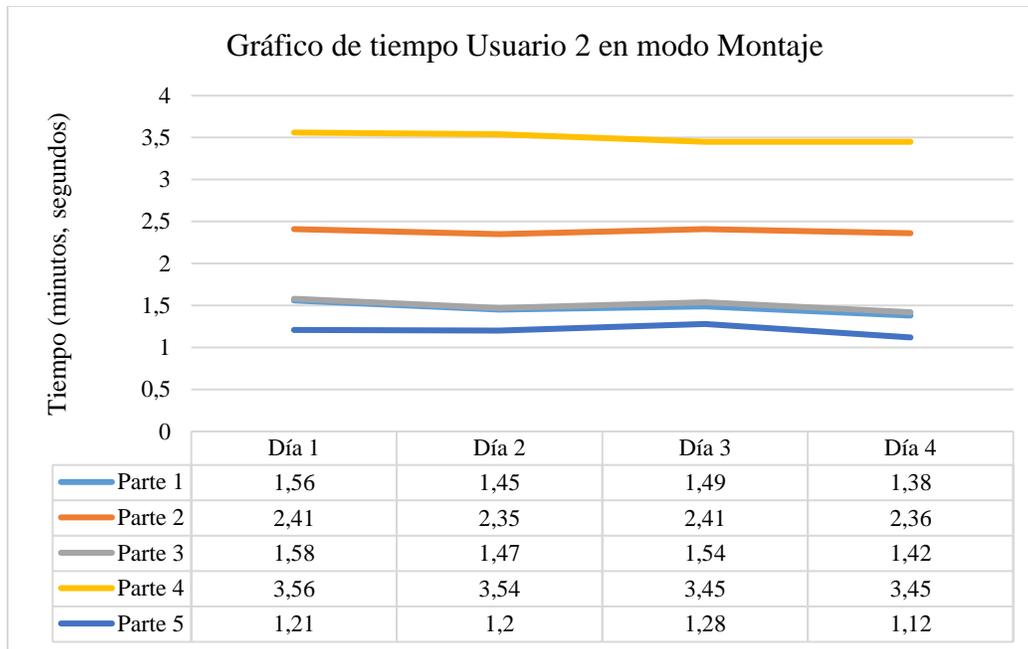


Figura 138. Resultados del usuario dos en modo Montaje.

En la figura 139 se muestra las curvas de los resultados obtenidos de la tabla 24 del usuario tres, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

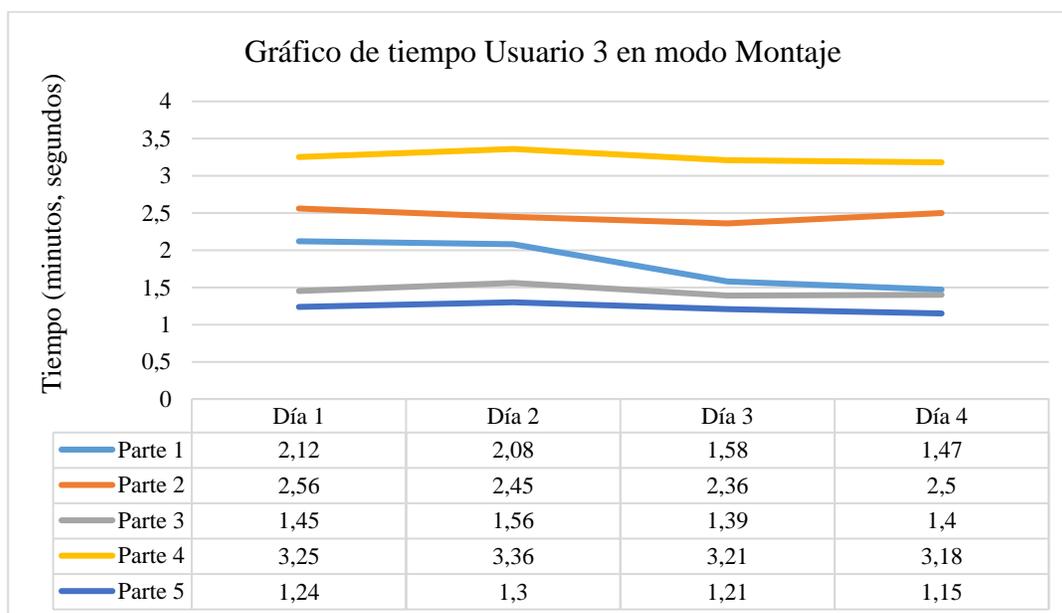


Figura 139. Resultados del usuario tres en modo Montaje.

En la figura 140 se muestra las curvas de los resultados obtenidos de la tabla 25 del usuario cuatro, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

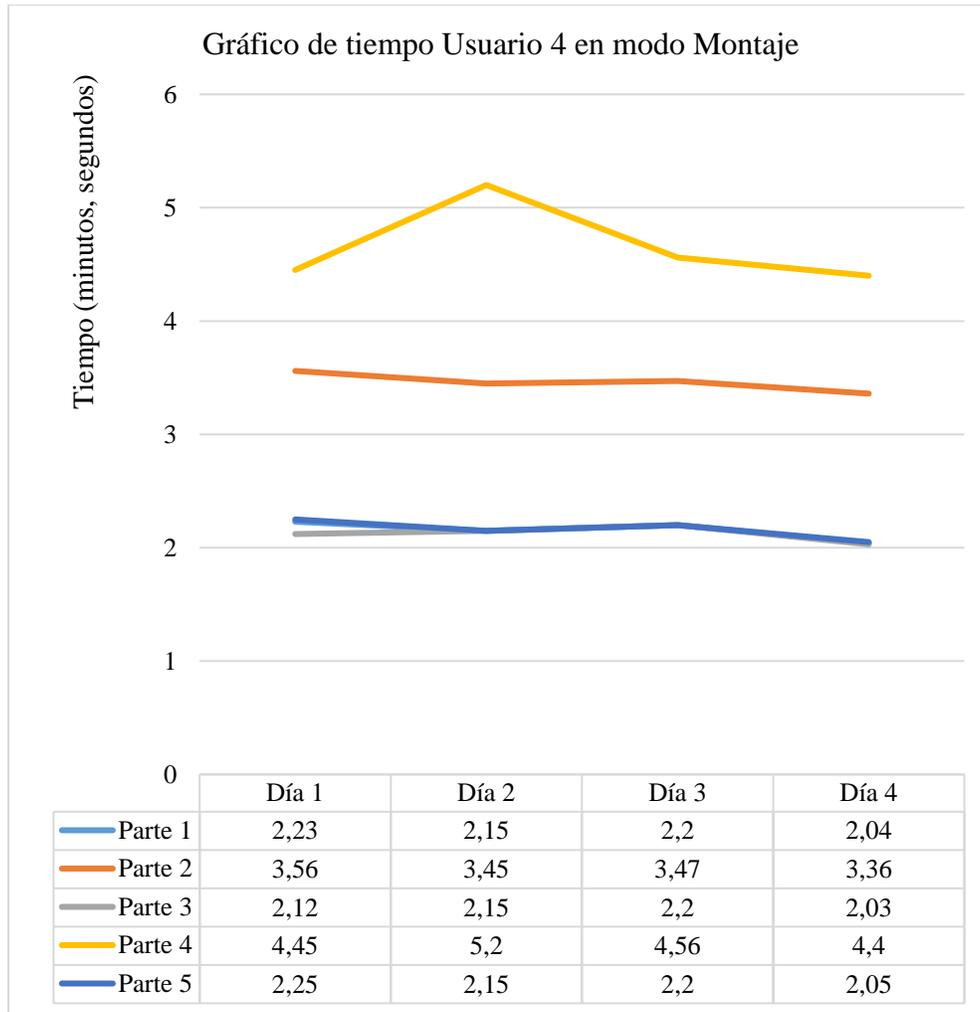


Figura 140. Resultados del usuario cuatro en modo Montaje.

Desmontaje.

En la figura 141 muestra las curvas de los resultados obtenidos de la tabla 26 del usuario uno, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

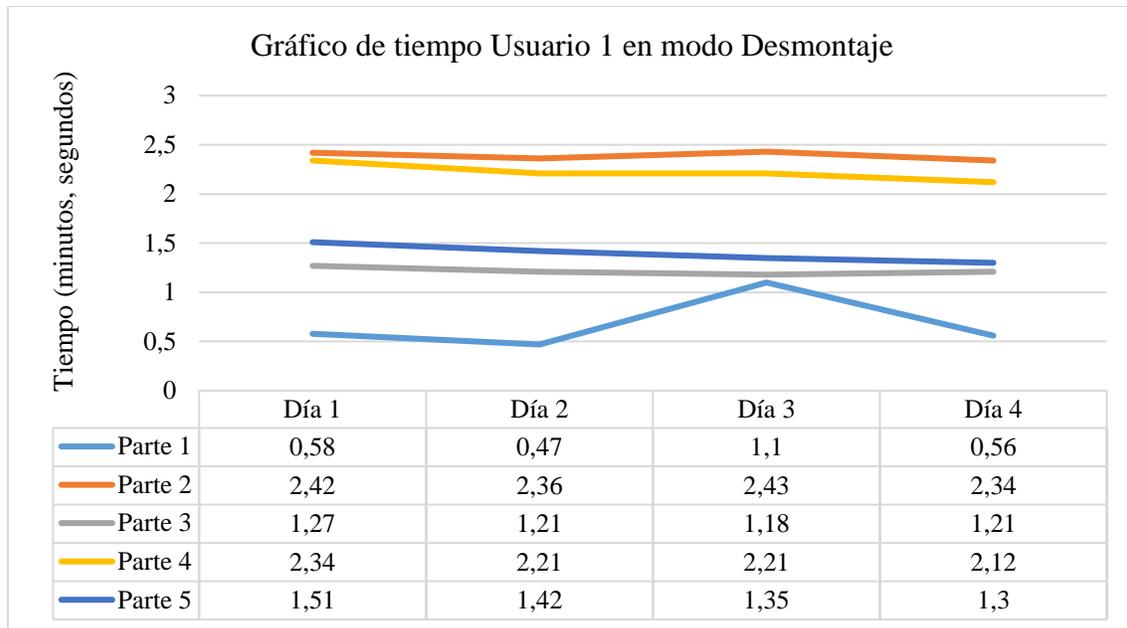


Figura 141. Resultados del Usuario 1 en modo Desmontaje.

En la figura 142 muestra las curvas de los resultados obtenidos de la tabla 27 del usuario uno, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de montaje.

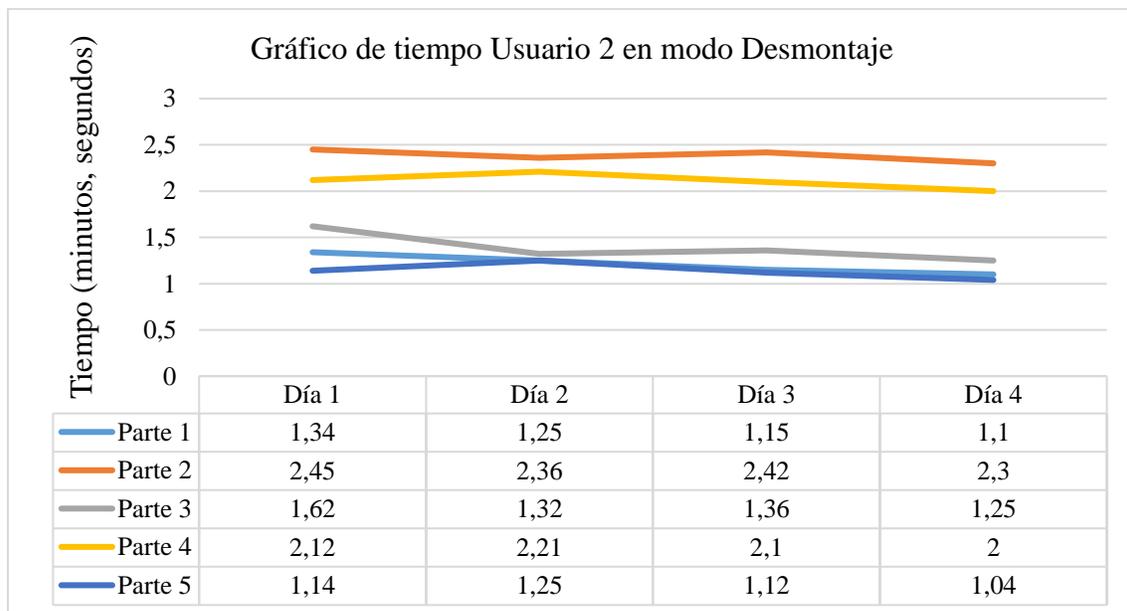


Figura 142. Resultados del Usuario 2 en modo Desmontaje.

En la figura 143 muestra las curvas de los resultados obtenidos de la tabla 28 del usuario uno, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de desmontaje.

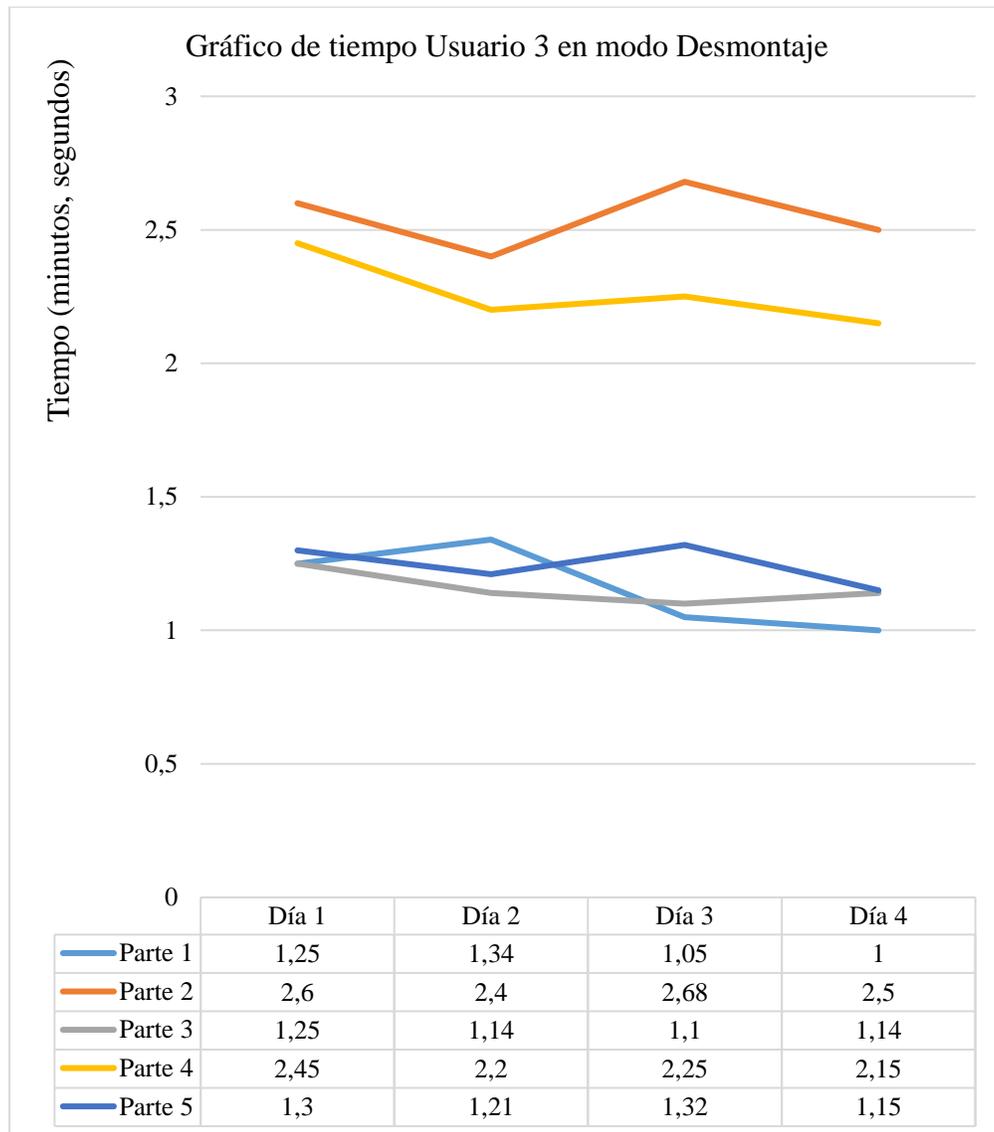


Figura 143. Resultados del Usuario 3 en modo Desmontaje.

En la figura 144 muestra las curvas de los resultados obtenidos de la tabla 29 del usuario uno, cada curva corresponde a un día que el usuario ha realizado la pruebas en las diferentes partes del proceso de desmontaje.

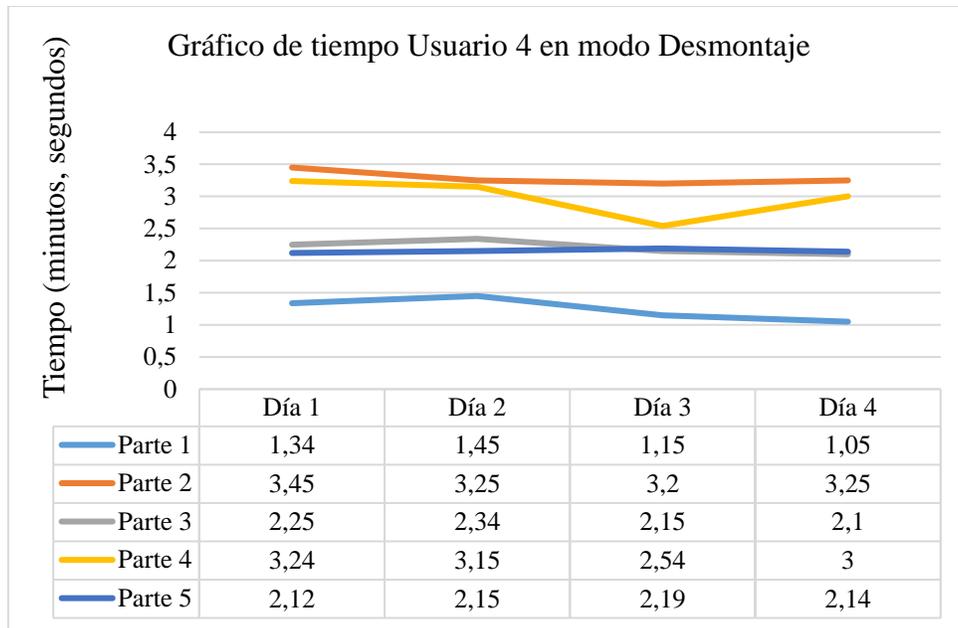


Figura 144. Resultados del Usuario 4 en modo Desmontaje.

Como se puede apreciar en cada una de las figuras (137 – 144) las curvas en cada parte del Sistema en sus dos variantes (Montaje y Desmontaje) tienen a decaer con el pasar los días, por lo que se puede decir que el usuario mejora la habilidad en las tareas de Montaje y Desmontaje. Por lo tanto, se puede decir que las métricas evaluadas en función del tiempo cumplen con lo descrito en el estándar, el tiempo correspondiente al último día de uso es menor al tomado en el primer día de uso y se puede verificar que la hipótesis según la normativa que determina la calidad de un software ISO/IEC 9126 – 4 (ISO 25000) (Anexo 3) *¿El diseño del sistema háptico con realidad virtual, permitirá mejorar la habilidad de montar y desmontar piezas de equipos mecatrónicos?* es verdadera.

4.5. Impacto del Trabajo a corto plazo.

El desarrollo de la Realidad Virtual como herramienta tecnológica dentro del Sistema Educativo aún no tiene un gran impacto en la formación profesional, ya que no se puede encontrar

aplicaciones totalmente funcionales. Como se observó en los resultados de la encuesta realizada a los estudiantes que hicieron uso del aplicativo el desarrollo de esta aplicación permite dar a conocer a los estudiantes de ingeniería mecatrónica las partes que conforman un robot y su respectivo sistema de transmisión de movimiento.

4.5.1. Aplicaciones y Beneficios.

En la actualidad la Realidad Virtual está siendo muy utilizada en el ámbito del entretenimiento de la misma manera, aunque no en gran medida en la parte de la educación, medicina y entrenamiento industrial y militar. La utilización de este tipo de tecnología en dichos campos frente a las metodologías tradicionales proporciona ventajas muy favorables. Una de las ventajas principales en el uso es el tipo de tecnología es la optimización del tiempo ya que siendo un sistema virtual el usuario puede hacer uso del mismo las veces que sea necesario y con la seguridad de estropear los equipos.

Dentro de la formación profesional en ingeniería mecatrónica se tiene las asignaturas de mecanismos y robótica industrial. Siendo óptimo el uso a estudiantes de niveles inferiores a sexto nivel, ya que el aplicativo proporciona información sobre las partes principales de un robot de tipo antropomórfico, los mecanismos de transmisión de movimiento que va desde la parte de un motor eléctrico hasta sus ejes principales a través de cajas reductoras tipo epicicloidal y correas dentadas.

Para la materia de Bases Fundamentales de la Ingeniería Mecatrónica misma que se encuentra en el primer nivel de la formación como ingeniero mecatrónico que corresponde a la nueva malla curricular tiene dentro de su contenido en la Unidad 2 de Estructuras de los Sistemas Mecatrónicos el apartado de comprender los Fundamentos de los Sistemas Mecánicos en donde el estudiante puede conocer los sistemas de transmisión de movimiento, con la ayuda del aplicativo el estudiante podrá conocer de forma más afectiva ya que se trata de un sistemas de realidad virtual

el sistema de transmisión de movimiento para un robot de tipo antropomórfico. En la figura 145 se puede observar la correspondiente asignatura de Bases Fundamentales de la Ingeniería Mecatrónica con su respectivo contenido en la segunda unidad.

<table border="1"> <tr> <td>EMEC MTU01</td> <td>FT</td> <td colspan="3">Bases Fundamentales de la Ingeniería Mecatrónica</td> </tr> <tr> <td>CD</td> <td>CP</td> <td>AA</td> <td>TOT.S</td> <td>TOT.A</td> </tr> <tr> <td>2</td> <td>0</td> <td>3</td> <td>5</td> <td>80</td> </tr> </table>					EMEC MTU01	FT	Bases Fundamentales de la Ingeniería Mecatrónica			CD	CP	AA	TOT.S	TOT.A	2	0	3	5	80	<p>Unidad 2</p> <p><u>Estructura de los Sistemas Mecatrónicos</u></p> <p>Fundamentos de los sistemas mecánicos</p> <p>Fundamentos de los sistemas electrónicos</p> <p>Fundamentos de los sistemas computacionales.</p> <p>Fundamentos de los sistemas integrales.</p> <p>Aplicaciones de sistemas, procesos y productos mecatrónicos.</p> <p>Fundamentos básicos de los proyectos de investigación.</p>	<p>Resultados de Aprendizaje de la Unidad 2</p> <p>Comprender la integración sinérgica de los conocimientos de Ingeniería Mecánica, Electrónica y Control Computacional para la creación, diseño, implementación y operación de equipos y maquinaria que busca mejorar la producción, innovación, tecnología, calidad y productividad.</p>
EMEC MTU01	FT	Bases Fundamentales de la Ingeniería Mecatrónica																			
CD	CP	AA	TOT.S	TOT.A																	
2	0	3	5	80																	

Figura 145. Beneficios para la materia de Bases Fundamentales de la Ingeniería Mecatrónica.

Para la materia de mecanismo misma que se encuentra en el nivel sexto de la formación como ingeniero mecatrónico tiene dentro de su contenido la clasificación de los mecanismos como parte fundamental de la ciencia de los mecanismos.

No.	UNIDADES DE ESTUDIO Y SUS CONTENIDOS	EVIDENCIA DEL APRENDIZAJE Y SISTEMA DE TAREAS
	Unidad 1: ANÁLISIS CINEMÁTICO Y DINÁMICO DE MECANISMOS	Producto de unidad: MEMORIA DE CÁLCULO DEL ANÁLISIS CINEMÁTICO DE UN MECANISMO SIMPLE (APLASTADOR DE LATAS, CIZALLA DE VARILLAS).
	1.1. FUNDAMENTOS DE LA CIENCIA DE LOS MECANISMOS. Análisis topológico de los mecanismos Clasificación de los mecanismos Criterio de clasificar un mecanismo Condición de Grashof Inversión cinemática y curvas de acoplador	Tarea principal 1.1: Analizar cinemáticamente un mecanismo simple Tarea principal 1.2: Modelar en Working Model 2D mecanismos simples Tarea principal 1.3: Prácticas de laboratorio de los temas planteados.
1	1.2. ANÁLISIS CINEMÁTICO DE MECANISMOS. Métodos gráficos y analíticos de Análisis del Movimiento, Velocidad, Aceleración, Ventaja Mecánica, Ángulos de Transmisión, Curvas de acoplador, Aceleración de Centros de Gravedad en: - Mecanismos manivela corredera - Mecanismo de cuatro barras - Cadenas cinemáticas en serie - Mecanismo de retorno rápido - Mecanismo de corredera invertida - Mecanismos de eslabonamientos y engranajes - Mecanismos de eslabonamientos y engranajes planetarios	

Figura 146. Beneficios en la materia de Mecanismos.

Para la materia de robótica industrial misma que se encuentra en el nivel noveno de la formación como ingeniero mecatrónico tiene dentro de su contenido la clasificación de los robots con la tarea asignada en sistema de transmisión de movimiento y reductores de velocidad, así como también los elementos y componentes de un robot.

EMEC 44050 ROBÓTICA INDUSTRIAL					
ELEE-30033	ELEE-30095	EMEC-33058		3	

<p>1.1 Antecedentes históricos</p> <p>1.1.1 Antecedentes históricos</p> <p>1.2 DEFINICIONES</p> <p>1.2.1 DEFINICIONES</p> <p>1.3 CLASIFICACIÓN DE LOS ROBOTS</p> <p>1.3.1 CLASIFICACIÓN DE LOS ROBOTS</p> <p><small>CÓDIGO: SGC.DI.321 VERSIÓN: 1.3 FECHA ÚLTIMA REVISIÓN: 23/09/14</small></p>	<p>Tarea 1 Resumen de evolución histórica del robot industrial</p> <p>Tarea 2 Clasificación del robot industrial según diferentes criterios: Evolución del sistema de control; Generación del robot</p>
Página 1 de 7	

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE <small>Vicerrectorado de Docencia</small>		
PROGRAMA DE ASIGNATURA - SÍLABO		
2. SISTEMA DE CONTENIDOS Y RESULTADOS DEL APRENDIZAJE		
<p>1.4 MORFOLOGÍA DEL ROBOT MANIPULADOR</p> <p>1.4.1 MORFOLOGÍA DEL ROBOT MANIPULADOR</p> <p>1.5 ELEMENTOS Y COMPONENTES DE UN ROBOT.</p> <p>1.5.1 ELEMENTOS Y COMPONENTES DE UN ROBOT.</p> <p>1.6 ORGANIZACIÓN FUNCIONAL DE UN ROBOT.</p> <p>1.6.1 ORGANIZACIÓN FUNCIONAL DE UN ROBOT.</p> <p>1.7 SUBSISTEMAS DE UN ROBOT.</p> <p>1.7.1 SUBSISTEMAS DE UN ROBOT.</p> <p>1.8 GRADOS DE LIBERTAD Y ESPACIO DE TRABAJO</p> <p>1.8.1 GRADOS DE LIBERTAD Y ESPACIO DE TRABAJO</p>	<p>Tarea 3 Sistemas de transmisión de movimiento y reductores de velocidad del robot industrial</p>	

Figura 147. Beneficios en la materia de Robótica Industrial.

En la figura 148 muestra un extracto de los videos de demostración en dentro del aplicativo en donde muestra las partes básicas del robot Mitsubishi RV-2SDB.

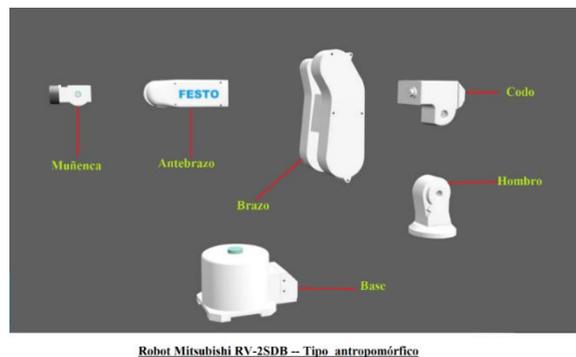


Figura 148. Elementos básicos de un robot antropomórfico.

De la misma manera en la figura 149 se muestra los componentes de un sistema de transmisión de movimiento de los ejes 2 y 3 que contienen sus respectivas cajas reductoras de velocidad.

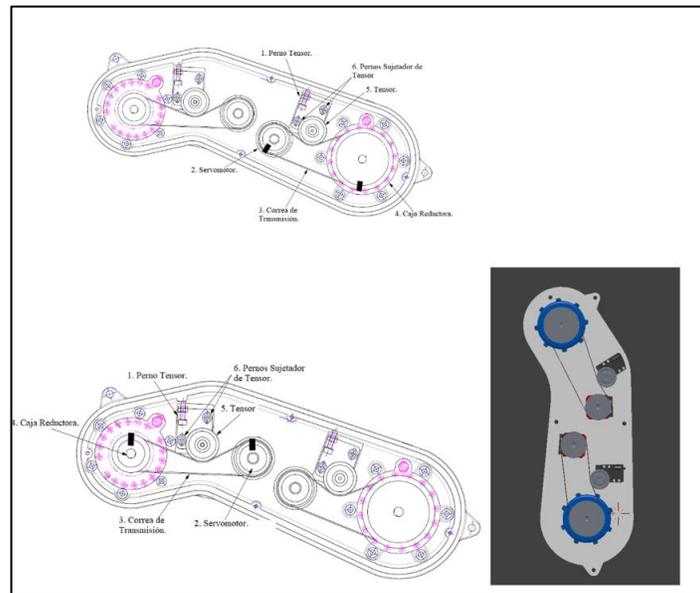


Figura 149. Mecanismo de transmisión de movimiento.

4.5.2. Futuras Mejoras del aplicativo.

Como todo software no siendo este la excepción tiene a existir mejoras tanto en la parte de software (Interfaz y Programación) como la de hardware (periféricos de interacción con el entorno virtual).

Para las mejoras del software se tiene lo siguiente:

- Como se tiene un programa modular utilizado tanto para la parte de Montaje y Desmontaje se puede crear más escenas de las existentes con las mismas tareas, pero con otras máquinas o agregar más componentes a las existentes tales como los finales de carrera entre otros elementos.

- El uso de herramientas adecuadas para el uso de montaje o desmontaje de las piezas del robot, como puede ser destornilladores o llaves.
- Un sistema de control que permita generar secuencias para el control del movimiento del robot dentro del sistema de realidad virtual con el objetivo dar a conocer la programación para el control de movimiento de un robot antropomórfico.

Para las mejoras del hardware se tiene lo siguiente:

- La utilización de gafas de Realidad Virtual como objetivo de incrementar el grado de inmersión por parte del usuario dentro del aplicativo.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES.

5.1. Conclusiones.

- El sistema háptico de realidad virtual que permite la manipulación, montaje y desmontaje de piezas de equipos mecatrónicos como herramienta educativa permite a los usuarios conocer la parte mecánica que conforma un Robot Mitsubishi RV-2SDB.
- La técnica de captura de gestos y movimiento que aplica el dispositivo Leap Motion Controller presenta mayor eficiencia dentro de aplicaciones de Realidad Virtual, teniendo como características principales su tasa de flujo de datos de un valor de 120 fps y su soporte para trabajar dentro los motores de video juegos 3D.
- Mediante el uso de la metodología cualitativa por puntos fue posible dar una selección tanto de software y hardware de manera objetiva, tomando en consideración las características técnicas y las respectivas ventajas y desventajas correspondiente a cada alternativa.
- Con la ayuda de la técnica de modelado 3D “Box Modeling” implementada en Blender 3D, se logró obtener el modelo tridimensional del miembro superior con excelentes características visuales para su implementación dentro del proyecto investigativo de Realidad Virtual.
- El dispositivo háptico dentro del Entorno de Realidad Virtual permite interactuar con los objetos pieza del Robot Mitsubishi RV-2SDB generando al usuario una retroalimentación vibro-táctil de manera efectiva dando como resultado a su correspondiente evaluación un promedio ponderado equivalente a 4 dentro del rango de satisfacción de usabilidad de 0 a 5 por parte de los cuatro usuarios.

- La estructura correspondiente al soporte de la tarjeta electrónica de control fue construida por medio de la tecnología de impresión 3D con el método modelado por deposición fundida (MDF) utilizando como material ácido poliláctico (PLA) dando como resultado una buena resistencia a esfuerzo de deflexión y corte con un valor en su factor de diseño de 1.3 mismo que es mayor a 1 y se considera que el diseño es correcto para la fuerza máxima aplicada.
- El aplicativo permite al usuario tener la capacidad de conocer las piezas correspondientes al sistema mecánico de transmisión de movimiento de un Robot, manipularlas y realizar las tareas de montaje y desmontaje de forma individual, proporcionando una ayuda a los estudiantes de ingeniería mecatrónica sobre el conocimiento básico de un robot antropomórfico
- La habilidad dentro del aplicativo para el proceso de montaje y desmontaje por parte de los usuarios van mejorando conforme mayor vez lo utilicen, dado como resultado la medida del tiempo correspondiente al último uso un valor mucho menor referente a su primera vez dentro del aplicativo.
- El sistema de realidad virtual cuenta seis partes, cada una de las partes corresponden a los componentes principales de un robot antropomórfico permitiendo así conocer de forma específica los componentes del sistema de transmisión de movimiento de cada eje.
- El algoritmo de control de secuencia tanto para la tarea de montaje como de desmontaje admite un número indefinido de piezas el cual está limitado por la cantidad de piezas que conforman una maquina en específico, siendo 7 piezas el número mínimo en el modo de montaje y desmontaje general y 23 el número máximo correspondiente a la parte del Antebrazo.

5.2. Recomendaciones.

- Para mejorar la habilidad en el proceso de montaje y desmontaje de piezas de equipos mecatrónicos se debe emplear varias sesiones del uso del aplicativo, manteniendo siempre un tiempo prudente entre cada sesión, esto con el objetivo de que el usuario no experimente cansancio muscular y visual.
- El aplicativo consume grandes recursos computacionales por lo que es necesario para su implementación disponer de un computador con características mínimas de: Tarjeta Gráfica NVIDIA GTX 970 / AMD R9 290 equivalente o mayor, memoria RAM de 8GB entre otras no muy relevantes.
- Para que el dispositivo de percepción háptica entre en funcionamiento con el aplicativo es necesario que el puerto USB que está conectado sea el puerto COM4, esto debido a que dentro de la programación se le asignó ese nombre al objeto de la clase Serial.IO como parte de la inicialización de la comunicación serial.

REFERENCIA BIBLIOGRÁFICA

ADesign. (2 de Abril de 2011). *Armonia de Colores*. Recuperado el 05 de Febrero de 2019, de

ADesign: <https://adesignperu.blog/2011/04/02/armonia-del-color/>

Aduino. (2019). *Arduino.CL*. Recuperado el 06 de Mayo de 2019, de Arduino Nano:

<https://arduino.cl/arduino-nano/>

Anónimo. (11 de Noviembre de 2019). *EcuRed*. Recuperado el 12 de Enero de 2019, de Autodesk

Maya: https://www.ecured.cu/Autodesk_Maya#Perspectiva_General

Anónimo. (11 de Noviembre de 2019). *EcuRed*. Recuperado el 25 de Noviembre de 2019, de

Cinema 4D: https://www.ecured.cu/Cinema_4D

Anónimo. (11 de Noviembre de 2019). *Immersion*. Recuperado el 25 de Agosto de 201, de Phantom

Omni: <https://www.immersion.fr/en/phantom-touch/>

Anónimo. (23 de Mayo de 2019). *Wikipedia*. Recuperado el 20 de Abril de 2019, de A Platform

with Six Degrees of Freedom: https://en.wikipedia.org/wiki/Stewart_platform

Arduino. (25 de Febreo de 2019). *Arduino*. Recuperado el 25 de Agosto de 2019, de Official Page:

<https://www.arduino.cc/>

Arduino. (1 de Febrero de 2019). *ARDUINO UNO REV3*. Recuperado el 12 de Septiembre de 2019,

de ARDUINO UNO REV3: <https://store.arduino.cc/usa/arduino-uno-rev3>

BELDA, J. (19 de Mayo de 2015). *Leap Motion Controller*. Recuperado el 14 de Septiembre de

2019, de SHOWLEAP TECHNOLOGIES: <http://blog.showleap.com/2015/05/leap-motion-api/>

- Binvignat, O. &. (Junio de 2012). *International Journal of Morphology*. Recuperado el 10 de Octubre de 2019, de Aspectos Biométricos de la Mano de Individuos Chilenos: https://www.researchgate.net/publication/262715372_Aspectos_Biometricos_de_la_Mano_de_Individuos_Chilenos
- Blender. (17 de Octubre de 2019). *Blender*. Recuperado el 02 de Noviembre de 2019, de Blender 2.80 Reference Manual: <https://docs.blender.org/manual/en/latest/index.html>
- BlenderES. (09 de Enero de 2017). *Requisitos del Sistema*. Recuperado el 12 de Julio de 2019, de BlenderES: <https://bldes.blogspot.com/p/requisitos-del-sistema.html>
- Covadonga Díez Sanmartín, M. A. (12 de Agosto de 2018). *Tecnologías multimedia e interacción/Realidad Virtual*. Recuperado el 06 de Septiembre de 2019, de Wikiuniversidad: https://es.wikiversity.org/wiki/Tecnolog%C3%ADas_multimedia_e_interacci%C3%B3n/Realidad_Virtual#Tipos_de_realidad_virtual
- CyberGlove. (1 de Enero de 2017). *CyberGloveSystems*. Recuperado el 15 de Enero de 2019, de CyberGloveSystems: <http://www.cyberglovesystems.com/cybertouch/>
- Dixon, M. (01 de Enero de 2019). *Mejor paleta de colores generadores*. Recuperado el 18 de Junio de 2019, de Paletas de Colores: <https://htmlcolorcodes.com/es/recursos/mejor-paleta-de-colores-generadores/>
- DMena, R. (30 de Junio de 2017). *Tesis Doctoral en Xarxa*. Recuperado el 25 de Septiembre de 2019, de Diseño y validación de un nuevo sistema para medir la función de la mano con un estabilizador de la postura de exploración y captadores de fuerza electrónicos: <https://www.tesisred.net/handle/10803/405908#page=8>

EcuRed. (19 de Agosto de 2019). *Nurbs*. Recuperado el 14 de Septiembre de 2019, de EcuRed:
<https://www.ecured.cu/NURBS>

Elesapiens. (19 de Mayo de 2017). Recuperado el 21 de Junio de 2019, de Realidad Virtual en educación. ¡El futuro ahora!: <https://www.elesapiens.com/blog/realidad-virtual-en-educacion-el-futuro-ahora/>

Erp, J., Kyung, K.-U., Kassner, S., & Carter, J. (14 de Septiembre de 2010). *EuroHaptics Part II LNCS*. Recuperado el 12 de Agosto de 2019, de Setting the Standards for Haptic and Tactile Interactions: ISO's Work:
https://www.researchgate.net/publication/229033513_Setting_the_Standards_for_Haptic_and_Tactile_Interactions_ISO's_Work

FM, Y. (21 de Julio de 2018). *Xataka*. Recuperado el 11 de Junio de 2019, de Qué es Arduino, cómo funciona y qué puedes hacer con uno: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Frank Weichert, D. B. (14 de Mayo de 2013). *Analysis of the Accuracy and Robustness of the Leap Motion Controller*. Recuperado el 14 de Enero de 2019, de National Institute of Health:
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3690061/#b6-sensors-13-06380>

García, J. (2018). *Psicología y Mente*. Recuperado el 12 de Agosto de 2019, de Psicología del color: significado y curiosidades de los colores:
<https://psicologiaymente.com/miscelanea/psicologia-color-significado>

Gidiño, J. (11 de Noviembre de 2019). *Universidad Autónoma del Estado de Hidalgo*. Recuperado el 28 de Febrero de 2019, de Las Tic Aplicadas A Un Dispositivo Háptico Novint Falcon:
<https://www.uaeh.edu.mx/scige/boletin/tlahuelilpan/n8/a1.html>

GodotEngine. (12 de Septiembre de 2019). *GodotDocs*. Recuperado el 14 de Noviembre de 2019, de Acerca de Godot Engine: <https://docs.godotengine.org/en/stable/about/introduction.html>

Greene, J. (1 de Febrero de 2012). *C/Net*. Recuperado el 12 de Octubre de 2019, de Microsoft presenta Kinect para Windows, SDK comercial: <https://www.cnet.com/news/microsoft-debuts-kinect-for-windows-commercial-sdk/>

Guevara Bryan, M. A. (13 de Julio de 2018). *Repositorio ESPEL*. Recuperado el 21 de Junio de 2019, de Diseño y desarrollo de un sistema inmersivo de reconocimiento y control de gestos, ostensible por medio de realidad virtual como método de ayuda en la rehabilitación de la capacidad motriz de las extremidades superiores en pacientes con accidente cerebro: <http://repositorio.espe.edu.ec/xmlui/handle/21000/14928>

ICGEO. (12 de Julio de 2015). *VISION ESTEREOSCOPICA*. Recuperado el 23 de Septiembre de 2019, de Proyectos y Construcciones: <http://www.icgeo.org/2018/01/la-vision-estereoscopica-es-la-base-de.html>

Infobae. (03 de Octubre de 2018). *Los gestos de la mano, el nuevo mouse*. Recuperado el 12 de Enero de 2019, de Infobae: <https://www.infobae.com/2012/05/22/1050861-los-gestos-la-mano-el-nuevo-mouse/>

JADIAZ. (21 de Enero de 2016). *Arduino*. Recuperado el 23 de Agosto de 2019, de PLACA ARDUINO UNO: <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>

Javir007. (31 de Octubre de 2018). *Platzi*. Recuperado el 12 de Febrero de 2019, de Tipos de Modelado 3d: <https://platzi.com/clases/1405-modelado-3d/14678-tipos-de-modelado-3d/>

LeapMotion. (10 de Julio de 2018). *Leap Motion Controller*. Recuperado el 12 de Enero de 2019, de Leap Motion Controller: https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html

LeapMotion. (19 de Julio de 2018). *Leap Motion Documentation*. Recuperado el 14 de Junio de 2019, de Leap Motion Documentation: https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Coordinate_Mapping.html

Lee, J. (9 de Diciembre de 2015). *Packt*. Recuperado el 23 de Enero de 201, de Una descripción general de Unreal Engine: <https://hub.packtpub.com/overview-unreal-engine/>

Limited, S. U. (2019). *Shenzhen Unique Electronic International Limited*. Recuperado el 24 de Febrero de 2019, de Shenzhen Unique Electronic International Limited: <https://posunitech.en.made-in-china.com/product/ZNmXkgPTvfcr/China-Bluetooth-2D-Barcode-Scanner-Wireless-Qr-Code-Reader.html>

Mayeda, R. (29 de Agosto de 2018). *UnrealEngine*. Recuperado el 13 de Septiembre de 2019, de Shotgun integration now available in Unreal Engine 4.20: <https://www.unrealengine.com/en-US/blog/shotgun-integration-now-available-in-unreal-engine-4-20>

MelfaRobots. (03 de Marzo de 2010). *MelfaRobots*. Recuperado el 23 de Febrero de 2019, de Intruction Manual RV-2SDB Robot Arm Setup & Maintenance: http://meltrade.hu/open_pdf.php?f=9641-rv-2sd-instruction-manual-

Microsoft. (2019). *Soporte técnico de Xbox*. Recuperado el 21 de Octubre de 2019, de Configuración de Kinect para Windows: <https://support.xbox.com/es-ES/xbox-on-windows/accessories/kinect-for-windows-setup>

Miñambres, D. (28 de Noviembre de 2017). *PremiumMadrid. Global Health Care*. Recuperado el 04 de Marzo de 2019, de ¿QUE ES EL SISTEMA PROPIOCEPTIVO EN FISIOTERAPIA Y DEPORTE?: <https://rehabilitacionpremiummadrid.com/blog/diego-minambres/que-es-el-sistema-propioceptivo-propiocepcion-en-fisioterapia-y-deporte/>

Molinier, M. (2 de Enero de 2018). *ANÁLISIS DE LOS CRITERIOS DE FALLA APLICADOS A LOS*. Recuperado el 12 de Noviembre de 2019, de ANÁLISIS DE LOS CRITERIOS DE FALLA APLICADOS A LOS: <http://materias.fi.uba.ar/mlc/monografias/MLC-MOLINIER.pdf>

Paletton. (28 de Agosto de 2019). *Color Scheme Designer*. Recuperado el 12 de Septiembre de 2019, de Paletton: <http://paletton.com/#uid=1000u0kl1llaFw0g0qFqFg0w0aF>

Raya, L. (11 de Noviembre de 2019). *Universidad Rey Juan Carlos*. Recuperado el 23 de Septiembre de 2019, de Visión Global Sobre Tecnología Táctil: https://www.acta.es/medios/articulos/ciencias_y_tecnologia/061115.pdf

Ruiz-Sarmiento, G. G.-J. (12 de Junio de 2011). *CÁMARAS BASADAS EN TIEMPO DE VUELO. USO EN LA MEJORA DE MÉTODOS DE DETECCIÓN DE CARAS*. Recuperado el 12 de Noviembre de 2019, de Universidad de Malaga: http://mapir.uma.es/papersrepo/2011/2011-raul-Jornadas_CEA-Camaras_basadas_en_Tiempo_de_Vuelo.pdf

Siemens. (19 de agosto de 2019). *Siemens*. Recuperado el 23 de Noviembre de 2019, de Diseño asistido por ordenador (CAD): <https://www.plm.automation.siemens.com/global/es/our-story/glossary/computer-aided-design-cad/12507>

TheFreshUK. (01 de Enero de 2017). *What is Virtual Reality?* Recuperado el 12 de Octubre de 2019, de Virtual Reality society: <https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html>

Ultimaker. (3 de Septiembre de 2018). *Utilamker 3D*. Recuperado el 23 de Septiembre de 2019, de Technical Data Sheet PLA: <https://ultimaker.com/download/74599/UM180821%20TDS%20PLA%20RB%20V10.pdf>

Unity3D. (2019). *Unity*. Recuperado el 12 de Octubre de 2019, de Unity 2019: Performance by Default, gráficos de alta fidelidad en tiempo real y herramientas para artistas: https://unity3d.com/es/unity?_ga=2.228523805.1436473692.1573568291-1403840706.1562887087

UnrealEngine. (2019). *UnrealEngine*. Recuperado el 12 de Octubre de 2019, de Features Unreal Engine: <https://www.unrealengine.com/en-US/features>

Urbina, B. (2013). *Evaluación de Proyectos*. México: McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V.

VALENZUELA, I. (01 de Enero de 2019). *El efecto de los colores en las habitaciones del hogar*. Recuperado el 25 de Septiembre de 2019, de VIX Inc: <https://www.vix.com/es/btg/curiosidades/4267/el-efecto-de-los-colores-en-las-habitaciones-del-hogar>

Wiki, G. (2019). *Geeetech Wiki*. Recuperado el 23 de Noviembre de 2019, de Arduino Nano: https://www.geeetech.com/wiki/index.php/Arduino_Nano

XSens. (16 de Septiembre de 2019). *XSens Mocap*. Recuperado el 23 de Octubre de 2019, de Motion Capture: <https://www.xsens.com/tags/motion-capture/>

Zucconi, A. (07 de Octubre de 2015). *How to integrate Arduino with Unity*. Recuperado el 25 de Septiembre de 2019, de How to integrate Arduino with Unity: <https://www.alanzucconi.com/2015/10/07/how-to-integrate-arduino-with-unity/>

ANEXOS



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor **Pilatasig Sarabia, Juan Carlos**.

En la ciudad de Latacunga, a los 8 días del mes de enero del 2020.

Aprobado por,

Ing. Andrés Gordón.

DIRECTOR DEL PROYECTO.

Ing. Darío Mendoza.

DIRECTOR DE CARRERA.

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO.

