



**Diseño y construcción de un sistema de entrenamiento militar
enfocado a polígonos de tiro mediante realidad virtual
para mejorar la destreza del personal de las Fuerzas Armadas del Ecuador**

Alvaro Taipe, Edgar German y Guascal Vélez, Boris Julián

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniera en Mecatrónica

Ing. Constante Prócel, Patricia Nataly

Latacunga, 19 de marzo del 2021



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ENTRENAMIENTO MILITAR ENFOCADO A POLÍGONOS DE TIRO MEDIANTE REALIDAD VIRTUAL PARA MEJORAR LA DESTREZA DEL PERSONAL DE LAS FUERZAS ARMADAS DEL ECUADOR**” fue realizado por los señores **Guascal Vélez Boris Julián, Alvaro Taipe Edgar German** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 19 de marzo de 2021



Firmado electrónicamente por:
**PATRICIA NATALY
CONSTANTE PROCEL**

.....
Ing. Constante Prócel, Patricia Nataly MSc.

C. C. 0503354029



Document Information

Analyzed document	Alvaro-Guascal-PoligoRV.pdf (D98989914)
Submitted	3/20/2021 12:18:00 AM
Submitted by	
Submitter email	bjguascal@espe.edu.ec
Similarity	7%
Analysis address	pnconstante.espe@analysis.orkund.com

Sources included in the report

SA	Universidad de las Fuerzas Armadas ESPE / ESCRITO TESIS 3.pdf Document ESCRITO TESIS 3.pdf (D54401860) Submitted by: kepusda@espe.edu.ec Receiver: amgordon.espe@analysis.orkund.com	5
W	URL: https://repositorio.espe.edu.ec/bitstream/21000/20725/1/T-ESPEL-MEC-0192.pdf Fetched: 3/19/2021 9:14:09 AM	4
SA	AndradeYandunTesis-18-07-2019.docx Document AndradeYandunTesis-18-07-2019.docx (D54477804)	4
SA	Universidad de las Fuerzas Armadas ESPE / SistemaHapticoConRealidaVirtual_PilatasigJuan.pdf Document SistemaHapticoConRealidaVirtual_PilatasigJuan.pdf (D60312146) Submitted by: jcpilatasig@espe.edu.ec Receiver: amgordon.espe@analysis.orkund.com	8
W	URL: http://repositorio.esuelamilitar.edu.pe/bitstream/handle/EMCH/241/HERRERA%20-%20J... Fetched: 3/20/2021 12:34:00 AM	1
W	URL: https://conocimientolibre.mx/microcontrolador/ Fetched: 3/20/2021 12:34:00 AM	1
W	URL: https://www.placadedesarrollo.com/onion/ Fetched: 3/20/2021 12:34:00 AM	1
W	URL: https://tpwd.texas.gov/education/hunter-education/educacion-para-cazadores/capitulo-2 Fetched: 3/20/2021 12:34:00 AM	2
SA	MARCO TEORICO REV 18.pdf Document MARCO TEORICO REV 18.pdf (D42040715)	1
W	URL: https://www.oculus.com/rift/?locale=es_LA Fetched: 3/20/2021 12:34:00 AM	1
W	URL: https://teslabem.com/blog/arduinos-mas-pequenos/ Fetched: 3/20/2021 12:34:00 AM	1

1/45



Firmado electrónicamente por:
PATRICIA NATALY
CONSTANTE PROCEL

.....
Ing. Constante Prócel, Patricia Nataly MSc.

C. C. 0503354029

DIRECTORA DEL PROYECTO



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Guascal Vélez, Boris Julián**, con cédula de ciudadanía n° 1718075169, **Alvaro Taipe Edgar German** con cedula de ciudadanía n° 1725037160 declaramos que el contenido, ideas y criterios del trabajo de titulación: **“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ENTRENAMIENTO MILITAR ENFOCADO A POLÍGONOS DE TIRO MEDIANTE REALIDAD VIRTUAL PARA MEJORAR LA DESTREZA DEL PERSONAL DE LAS FUERZAS ARMADAS DEL ECUADOR”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 19 de marzo de 2021

Guascal Vélez, Boris Julián

C. C. 1718075169

Álvaro Taipe, Edgar German

C.C. 1725037160



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Guascal Vélez Boris Julián**, con cédula de ciudadanía n° 1718075169, **Alvaro Taipe Edgar German**, con cedula de ciudadanía n° 1725037160 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ENTRENAMIENTO MILITAR ENFOCADO A POLÍGONOS DE TIRO MEDIANTE REALIDAD VIRTUAL PARA MEJORAR LA DESTREZA DEL PERSONAL DE LAS FUERZAS ARMADAS DEL ECUADOR**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 19 de marzo de 2021

Guascal Vélez, Boris Julián
C. C. 1718075169

Álvaro Taipe, Edgar German
C.C. 1725037160

DEDICATORIA

A mis padres, Galo y Cristina que, a lo largo de mi vida, me han apoyado en cada paso, y decisión tomada, con su apoyo y amor incondicional, también a mi abuelita, que siempre estuvo dando palabras de aliento, a lo largo de mi carrera universitaria.

Guascal Vélez, Boris Julián

DEDICATORIA

Este trabajo va dedicado a mi familia, quienes siempre han sido un respaldo para mí, especialmente a mi padre, Víctor Hugo Alvaro Ramírez, por haberme ayudado a desarrollar mis habilidades y enseñarme a nunca rendirme. A mi madre, Lilian Genoveva Taipe Torres, por su lucha incansable por ayudarme a cumplir mis metas, por cuidar siempre de mí y ser incondicional conmigo. A mi hermano, Víctor Hugo, quien ha sido como un segundo padre y un pilar fundamental para alcanzar este objetivo. Gracias a mis hermanos Cristina, Mauricio, Darío, Jessica y Geovany, por el esfuerzo y apoyo brindado, a lo largo de estos años, hoy me siento muy feliz de poder cumplir esta meta.

Álvaro Taipe, Edgar German

AGRADECIMIENTO

A mis padres, que han sido el pilar fundamental de mi vida, a mi abuelita que siempre está para guiarme, a mi familia que siempre esta presentes para compartir mis triunfos y logros, a mis amigos que son la familia que uno escoge, y a todas las personas que formaron parte de esta travesía.

Guascal Vélez, Boris Julián

AGRADECIMIENTO

A mi familia, amigos, quienes me brindaron las facilidades para poder desarrollarme profesionalmente. Estoy muy agradecido, con mis docentes de control eléctrico industrial y automatización. por haberme ayudado a mejorar mis habilidades y dotarme de conocimiento para mi desempeño profesional y ayudarme a adquirir la pasión por la ingeniería.

A todos mis amigos de universidad, que hicieron gratos los años de mi vida como estudiante universitario, las experiencias conmigo me acompañaran por siempre. A todos quienes hicieron de mi la persona que hoy soy, gracias.

Álvaro Taipe, Edgar German

Tabla de contenido

Carátula.....	1
Certificación.....	2
Informe urkund.....	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento.....	8
Agradecimiento.....	9
Tabla de contenido.....	10
Índice de figuras.....	16
Índice de tablas.....	21
Resumen.....	23
Abstract.....	24
Introducción.....	25
Planteamiento del problema.....	25
Antecedentes.....	25
Justificación e importancia.....	27
Hipótesis.....	28
Objetivos.....	28
<i>Objetivo general</i>	28
<i>Objetivos específicos</i>	28
Marco teórico.....	30
Realidad virtual (RV).....	30

<i>Tipos de RV</i>	31
<i>RV inmersiva</i>	31
<i>RV no inmersiva</i>	32
Aplicaciones de la RV.....	33
Plataformas de desarrollo.....	34
<i>Unity</i>	34
<i>Unreal Engine</i>	35
<i>CryEngine</i>	37
<i>Modelado 3D</i>	38
<i>Blender</i>	39
<i>3ds Max</i>	41
Placas de desarrollo.....	42
<i>Modelos principales de placas de desarrollo</i>	43
Arduino.....	43
Raspberry Pi.....	44
Onion.....	44
Polígono de tiro.....	45
<i>Tipos de polígono de tiro</i>	45
Armas de bajo calibre.....	46
<i>Reglas principales para su uso seguro</i>	47
<i>Principios básicos de tiro</i>	47
<i>Armas utilizadas por el Ejército Ecuatoriano</i>	48
Entrenamiento militar.....	49
<i>Costo del entrenamiento militar</i>	50
<i>Costo del entrenamiento en polígonos de tiro</i>	51

Desarrollo del sistema mecánico y electrónico	52
<i>Descripción de necesidades.....</i>	<i>53</i>
<i>Arquitectura del diseño.....</i>	<i>53</i>
Metodología de selección de componentes	55
<i>Selección del software</i>	<i>55</i>
Software de desarrollo del entorno de RV.	55
Unity 3D.....	56
Unreal Engine.....	58
CryEngine 3.....	59
<i>Selección del software de desarrollo del entorno de RV</i>	<i>61</i>
Software de modelado 3D.	62
<i>Blender.....</i>	<i>63</i>
<i>Cinema 4D.....</i>	<i>64</i>
<i>Autodesk 3sd Max.....</i>	<i>66</i>
<i>Selección del software de modelado 3D.....</i>	<i>69</i>
<i>Selección del hardware.....</i>	<i>69</i>
Visores de RV para PC.	69
<i>HTC Vive Pro.....</i>	<i>70</i>
<i>HP Reverb.....</i>	<i>72</i>
<i>Oculus Rift.....</i>	<i>74</i>
<i>Selección de los visores de RV.....</i>	<i>76</i>
Tarjeta de control.....	77
<i>Arduino Nano.....</i>	<i>77</i>
<i>Arduino Pro Mini.....</i>	<i>79</i>
<i>Arduino Pro Micro.....</i>	<i>81</i>

<i>Selección de las tarjetas de control.</i>	82
Acelerómetro-giroscopio.	83
<i>MPU 6050.</i>	83
<i>MPU 9250.</i>	85
Accionamiento gatillo del mando.	87
<i>Micro Switch KW11-3Z.</i>	87
Accionamiento botón de recarga del mando.....	89
Motor vibrador.....	89
<i>Motor vibrador 9.000 RPM.</i>	90
<i>Motor vibrador 3.200 RPM.</i>	91
<i>Selección del motor vibrador.</i>	91
Diseño del mando.....	92
<i>Diseño electrónico</i>	92
Consumo MPU6050.	93
Consumo Arduino.....	94
Consumo motor vibrador.	94
<i>Diseño mecánico</i>	94
Técnicas de manufactura.	98
<i>Modelo por inyección de plástico.</i>	98
<i>Impresión 3D</i>	99
<i>Mecanizado</i>	100
Selección de la técnica de mecanizado.....	101
Selección del material para impresión 3D.....	102
<i>ABS.</i>	102
<i>PLA</i>	102

PETG	102
Fuerza de agarre de una persona	105
Análisis del pasador	109
Impresión 3D del mando	117
Desarrollo de la interfaz de realidad virtual	120
<i>Desarrollo del modelado 3D</i>	121
<i>Desarrollo de la interfaz en Unity</i>	128
<i>Integración de Oculus a Unity</i>	130
Desarrollo de las ventanas de inicio	131
<i>Desarrollo de los algoritmos de control del entorno de realidad virtual</i>	133
Pruebas y resultados	145
Orientación del prototipo de arma diseñado	145
Valoración del sistema de entrenamiento militar virtual	148
Evaluación de sujetos de prueba	149
<i>Crterios de idoneidad</i>	149
<i>Valoración de los participantes</i>	149
<i>Validación de la usabilidad del sistema de entrenamiento</i>	154
Validación de la hipótesis	156
<i>Planteamiento de la hipótesis</i>	156
<i>Variables de investigación</i>	157
<i>Validación de la hipótesis</i>	157
Conclusiones y recomendaciones	162
Conclusiones	162
Recomendaciones	163
Bibliografía	165

Anexos.....172

Índice de figuras

Figura 1 <i>Usuario de visor de RV</i>	31
Figura 2 <i>Percepción de RV inmersiva</i>	32
Figura 3 <i>Equipo requerido para RV no inmersiva</i>	33
Figura 4 <i>Logo de Unity</i>	35
Figura 5 <i>Logo de Unreal Engine</i>	36
Figura 6 <i>Gears of War 4</i>	36
Figura 7 <i>Logotipo de CryEngine</i>	37
Figura 8 <i>Far Cry 5</i>	38
Figura 9 <i>Software Vectary - modelado 3D</i>	39
Figura 10 <i>Logo de Blender</i>	40
Figura 11 <i>Modo objeto y modo edición de blender</i>	40
Figura 12 <i>Logo de 3ds Max</i>	41
Figura 13 <i>Renderizado con 3ds Max</i>	42
Figura 14 <i>Placa de desarrollo Arduino</i>	43
Figura 15 <i>Tarjeta de desarrollo Raspberry Pi A</i>	44
Figura 16 <i>Placa de desarrollo Onion</i>	45
Figura 17 <i>Arma bajo calibre Glock</i>	47
Figura 18 <i>Pietro Beretta 92 FS</i>	49
Figura 19 <i>Personal FF.AA. del Ecuador</i>	50
Figura 20 <i>Polígono de tiro de las FF.AA.</i>	51
Figura 21 <i>Arquitectura del diseño</i>	54
Figura 22 <i>Entorno de desarrollo Unity 3D</i>	56
Figura 23 <i>Entorno Unreal Engine 4</i>	58
Figura 24 <i>Entorno de desarrollo CryEngine 3</i>	60

Figura 25 <i>Entorno de desarrollo de Blender 3D</i>	63
Figura 26 <i>Entorno de desarrollo de Cinema 4D</i>	65
Figura 27 <i>Entorno de desarrollo de 3ds Max</i>	67
Figura 28 <i>Visores de RV HTC Vive Pro</i>	70
Figura 29 <i>Visores de RV Hp Reverb</i>	72
Figura 30 <i>Visores de RV Oculus Rift</i>	74
Figura 31 <i>Arduino Nano</i>	78
Figura 32 <i>Arduino Pro Mini</i>	79
Figura 33 <i>Arduino Pro Micro</i>	81
Figura 34 <i>MPU 6050</i>	84
Figura 35 <i>MPU 9250</i>	85
Figura 36 <i>Micro Switch KW11-3Z</i>	88
Figura 37 <i>Pulsador de dos pines</i>	89
Figura 38 <i>Motor vibrador 9.000 RPM</i>	90
Figura 39 <i>Motor vibrador 9.000 RPM</i>	91
Figura 40 <i>Diagrama de conexión</i>	92
Figura 41 <i>Pietro Beretta 92 FS</i>	95
Figura 42 <i>Mando diseñado, vista lateral derecha</i>	96
Figura 43 <i>Mando diseñado, vista lateral izquierda</i>	96
Figura 44 <i>Componentes del mando</i>	98
Figura 45 <i>Proceso de moldeo por inyección de plástico</i>	99
Figura 46 <i>Impresora 3D</i>	100
Figura 47 <i>Proceso de mecanizado</i>	100
Figura 48 <i>Deformación del mando</i>	105
Figura 49 <i>Esfuerzo sobre el mando</i>	106

Figura 50 <i>Factor de seguridad del mando</i>	107
Figura 51 <i>Deformación sobre el gatillo</i>	107
Figura 52 <i>Esfuerzo equivalente Von-mises sobre el gatillo</i>	107
Figura 53 <i>Factor de seguridad del gatillo</i>	108
Figura 54 <i>Pasador</i>	110
Figura 55 <i>Diagrama de cuerpo libre del pasador</i>	110
Figura 56 <i>Sección 1 del pasador</i>	112
Figura 57 <i>Sección 2 del pasador</i>	112
Figura 58 <i>Diagrama de fuerzas cortantes</i>	113
Figura 59 <i>Sección 1 momento flector</i>	113
Figura 60 <i>Sección 2 momento flector</i>	114
Figura 61 <i>Diagrama de momento flector</i>	115
Figura 62 <i>Deformación del pasador</i>	116
Figura 63 <i>Esfuerzo de Von Mises sobre el pasador</i>	116
Figura 64 <i>Factor de seguridad del pasador</i>	117
Figura 65 <i>Software cura</i>	118
Figura 66 <i>Impresión del mando</i>	118
Figura 67 <i>Diagrama de flujo del subsistema</i>	120
Figura 68 <i>Logotipo de blender</i>	121
Figura 69 <i>BoxModelling</i>	122
Figura 70 <i>Campo de tiro controlado</i>	123
Figura 71 <i>Modelado del carril de disparo</i>	124
Figura 72 <i>Mapa UV de carril de disparo</i>	125
Figura 73 <i>Logotipo de Krita</i>	125
Figura 74 <i>Textura de material caucho</i>	126

Figura 75 <i>Mapa de normales del fondo de pared</i>	127
Figura 76 <i>Nodos del material piso</i>	127
Figura 77 <i>Escenario optimizado a bajos polígonos</i>	128
Figura 78 <i>Logotipo de Unity</i>	129
Figura 79 <i>Escenario texturizado e iluminado</i>	129
Figura 80 <i>Box collider</i>	130
Figura 81 <i>Paquete Oculus Integration</i>	131
Figura 82 <i>Pantalla de inicio del proyecto</i>	132
Figura 83 <i>Ventana de acceso</i>	132
Figura 84 <i>Ventana de registro</i>	133
Figura 85 <i>Script menú principal</i>	134
Figura 86 <i>First person controller de Oculus Integration</i>	134
Figura 87 <i>Modelo de brazos y arma</i>	135
Figura 88 <i>Animación del modelo</i>	136
Figura 89 <i>Script vida</i>	137
Figura 90 <i>Script lógica jugador</i>	138
Figura 91 <i>Práctica concluida</i>	139
Figura 92 <i>Inicialización del script</i>	139
Figura 93 <i>Función update</i>	140
Figura 94 <i>Funciones de disparo</i>	141
Figura 95 <i>Función disparo directo</i>	142
Figura 96 <i>Función de animaciones</i>	142
Figura 97 <i>Funciones de recarga</i>	143
Figura 98 <i>Script de elementos visuales</i>	144
Figura 99 <i>Resultados posición del sensor</i>	146

Figura 100 <i>Datos de Movimiento del mando</i>	147
Figura 101 <i>Resultados de la práctica del progreso del participante 1</i>	152
Figura 102 <i>Progreso del participante 2</i>	154

Índice de tablas

Tabla 1 Descripción de necesidades.....	53
Tabla 2 Arquitectura modular.....	55
Tabla 3 Características necesarias para la instalación de Unity 3D.....	57
Tabla 4 Características necesarias para la instalación de Unreal Engine 4.....	58
Tabla 5 Características necesarias para la instalación de CryEngine 3.....	60
Tabla 6 Matriz de selección para el software de desarrollo.....	61
Tabla 7 Características necesarias para la instalación de Blender.....	63
Tabla 8 Características necesarias para la instalación de Cinema 4D.....	66
Tabla 9 Características necesarias para la instalación de 3ds Max.....	68
Tabla 10 Matriz de selección para el software de desarrollo.....	69
Tabla 11 Requisitos mínimos del computador para utilizar las HTC Vive Pro.....	71
Tabla 12 Requisitos mínimos del computador para utilizar las Hp Reverb.....	73
Tabla 13 Requisitos mínimos para utilizar las RV Oculus Rift.....	75
Tabla 14 Matriz de selección para los visores de RV.....	76
Tabla 15 Características técnicas Arduino Nano.....	78
Tabla 16 Características técnicas del Arduino pro Mini.....	80
Tabla 17 Características técnicas del Arduino Pro Micro.....	81
Tabla 18 Matriz de selección de las tarjetas de control.....	82
Tabla 19 Características técnicas del módulo MPU 6050.....	84
Tabla 20 Características técnicas del módulo MPU 9250.....	86
Tabla 21 Matriz de selección para el acelerómetro-giroscopio.....	87
Tabla 22 Características del micro switch KW 11-3z.....	88
Tabla 23 Características técnicas del motor vibrador de 9.000 RPM.....	90
Tabla 24 Características técnicas del motor vibrador de 3.200 RPM.....	91

Tabla 25 <i>Calificación, evaluación y selección para el accionamiento del gatillo</i>	92
Tabla 26 <i>Características técnicas de la Beretta 92-FS</i>	95
Tabla 27 <i>Matriz de selección para la técnica de manufactura</i>	101
Tabla 28 <i>Propiedades impresión de materiales</i>	103
Tabla 29 <i>Matriz de selección para el material de impresión</i>	104
Tabla 30 <i>Propiedades PLA</i>	104
Tabla 31 <i>Información de participantes</i>	150
Tabla 32 <i>Evolución del participante 1, tiempo 1</i>	150
Tabla 33 <i>Evolución del participante 1, tiempo 2</i>	151
Tabla 34 <i>Progreso del participante 2, tiempo 2</i>	153
Tabla 35 <i>Evolución del participante 2, tiempo 2</i>	153
Tabla 36 <i>Test SUS en participantes</i>	155
Tabla 37 <i>Ponderación SUS</i>	156
Tabla 38 <i>Datos de las pruebas de funcionamiento realizadas</i>	158
Tabla 39 <i>Frecuencias esperadas</i>	159
Tabla 40 <i>Datos calculados de chi-cuadrado</i>	160

RESUMEN

El presente proyecto se basa en la integración de la tecnología orientada al uso de la realidad virtual de tipo inmersivo enfocado al uso o entrenamiento del personal militar de la Fuerzas Armadas del Ecuador, principalmente enfocada en los polígonos de tiro, a través del diseño y la construcción de un simulador de realidad virtual, para lo cual se creó un prototipo similar a los utilizados en los entrenamientos reales, en el que se implementó los sensores y microcontroladores necesarios para poder llevar movimientos al ambiente virtual, y para poder tener una mejor experiencia se colocó un actuador que simula la sensación de disparo mediante un motor vibrador, en cuanto a la construcción del mando, el prototipo se diseñó mediante software CAD, se validó su diseño mediante cálculos analíticos y también mediante software de simulación CAE, este prototipo fabricó mediante impresión 3D, ya que cumplió con los requerimientos necesarios. El ambiente virtual fue creado mediante el motor de programación de videojuegos Unity, tomando en cuenta las especificaciones técnicas que deben tener los polígonos de tiro disponibles para el personal de las Fuerzas Armadas, en este software también se lleva el registro de la puntuación obtenida por cada usuario en cada sesión de entrenamiento para poder validar la eficiencia del mismo.

Palabras clave:

- **REALIDAD VIRTUAL**
- **POLÍGONOS DE TIRO**
- **ENTRENAMIENTO MILITAR**

ABSTRACT

This project is based on the integration of technology oriented to the use of immersive virtual reality focused on the use or training of military personnel of the Ecuadorian Armed Forces, mainly focused on shooting polygons, through the design and construction of a virtual reality simulator, for which a prototype similar to those used in real training was created , in which the sensors and microcontrollers needed to be able to carry movements to the virtual environment was implemented, and in order to have a better experience an actuator was placed that simulates the feeling of firing by means of a vibrating motor, in terms of the construction of the control, the prototype was designed by CAD software, its design was validated by analytical calculations and also by CAE simulation software , this prototype manufactured by 3d printing, as it met the necessary requirements. The virtual environment was created using the Unity video game programming engine, taking into account the technical specifications that the shooting polygons available to Armed Forces personnel must have, this software also keeps track of the score obtained by each user in each training session in order to validate the efficiency of the same.

Key words:

- **VIRTUAL REALITY**
- **SHOOTING POLYGONS**
- **MILITARY TRAINING**

Capítulo I

1. Introducción

1.1 Planteamiento del problema

El personal de las Fuerzas Armadas (FF.AA.) del Ecuador tiene como misión fundamental la defensa de la soberanía y la integridad territorial. Como consecuencia, surge la necesidad de un exhaustivo entrenamiento militar para que puedan ejecutar sus funciones de una manera adecuada, este entrenamiento es considerado altamente peligroso porque se realiza con munición y armamento real. Este equipo de entrenamiento puede resultar muy costoso ya que es necesario dotar de armamento a cada uno de los cadetes.

A nivel nacional e internacional se han implementado sistemas de entrenamiento virtual utilizando entornos netamente virtuales, logrando así reducir el riesgo a la integridad física del personal militar y los altos costos elevados, obteniendo resultados similares a un entrenamiento real en cuanto a mejora de destrezas y satisfacción. Por lo mencionado, se pretende implementar un sistema de entrenamiento virtual inmersivo similar a un terreno real de entrenamiento en un espacio reducido. Este sistema de entrenamiento se presenta como requisito previo al uso de munición y armamento real.

1.2 Antecedentes

Uno de los requisitos indispensables para el entrenamiento del personal de las FF.AA. del Ecuador es que ellos tengan la facultad y destreza para la utilización del equipamiento militar, y según la constitución del Ecuador en su Art. 158 dice entre otras cosas “Las FF.AA. tienen como misión fundamental la defensa de la soberanía y la

integridad territorial” (EC, Art.- 168, 2008). Llevando a cabo múltiples investigaciones relacionadas a la implementación virtual de polígonos de disparo a nivel nacional.

Existen diferentes investigaciones llevadas a cabo en universidades del país, basadas en la simulación de polígonos de disparo enfocados en la puntuación, seguimiento y registro de puntajes implementados mediante interfaces virtuales, donde cabe mencionar que el simulador implementado en la Universidad de las FF.AA. en el año 2018 funciona como un polígono de tiro virtual, pero este consta del tipo de realidad virtual no inmersiva, ya que este funciona mediante un infocus, cámara, y puntero laser. (Castro, 2018).

También se han hecho estudios sobre los diferentes tipos de simuladores de polígonos de tiro, así como las principales ventajas de cada uno tanto a nivel funcional, legal y de costos. El principal aspecto tomado en cuenta de esta investigación es el económico, y que, según las investigaciones, los costos de munición de entrenamiento por cada persona anualmente bordean los USD \$ 42.000 dólares (Villalva, 2015).

Las FF.AA. entonces necesitan de herramientas que vayan a la par del avance de la tecnología, permitiendo así la creación de sistemas de simulación más complejos como es el caso de la realidad virtual del tipo inmersiva, pero más eficientes, que permitan el entrenamiento de polígonos de tiro para a sus soldados sin tener que gastar en munición real y sin necesidad de la adecuación de espacios especializados en este tipo de entrenamientos.

1.3 Justificación e importancia

El Art. 15 del Reglamento a la ley sobre armas, municiones, explosivos y accesorios” en su capítulo I emitido en 2015, menciona:

Son armas de guerra y por tanto de uso privativo de las FF.AA., aquellas utilizadas con el objeto de defender la soberanía nacional, mantener la integridad territorial y el orden constitucional, tales como:

- a) Pistolas calibre 9 mm o superiores;
- b) Fusiles y armas automáticas, sin importar calibres;

Uno de los métodos de entrenamiento de armas de fuego son las pruebas en polígonos de tiro, los cuales constan de un terreno aislado en el que se tienen distintos tipos de objetivos, pero sobre todo estos terrenos deben tener la seguridad adecuada para no poner en riesgo la integridad física tanto de los practicantes como de terceras personas, por lo que este tipo de terrenos no son tan fáciles de ubicar en los cuales están presentes estos tipos de armas de uso exclusivo de las Armadas (Ecuador, Reglamento a la ley sobre armas, municiones, explosivos y accesorios, Decreto Ejecutivo 169, Registro Oficial 32, 15-jun., 2015).

El efecto de estos entrenamientos por el personal de las FF.AA. anualmente bordea los USD \$ 83.720 dólares por escuela, dicho valor representa una fuerte inversión económica para las FF.AA., es por este motivo que se plantea el uso de tecnologías para el desarrollo de un simulador con aplicación con gafas de realidad virtual y un prototipo de arma de bajo calibre el cual pueda proporcionar a los usuarios

una experiencia lo más real posible. La factibilidad del proyecto radica en el hecho de que se puede conseguir resultados similares al de un entrenamiento implementado con realidad virtual a comparación de un entrenamiento real, con los beneficios de la reducción de costos y sin la necesidad de poner en peligro la integridad física del personal militar.

1.4 Hipótesis

El sistema de entrenamiento del personal militar mejora las destrezas de manipulación y utilización de armamento de fuego, sin la necesidad de exponer al elemento humano y la reducción de costos de capacitación.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar y construir un sistema de entrenamiento militar enfocado a polígonos de tiro mediante realidad virtual para mejorar la destreza del personal de las FF.AA.

1.5.2 Objetivos específicos

- Investigar sobre las características de los polígonos de tiro utilizadas en los entrenamientos militares, en donde se podrá disponer de polígonos de tiro virtuales para armas de calibre menor, para el entrenamiento de los miembros de las FF.AA.
- Seleccionar los sensores y actuadores necesarios para llevar a cabo una correcta interacción entre el usuario y la interfaz virtual, para los movimientos del usuario.

- Diseñar y construir un prototipo de mando similar a una real utilizada en entrenamientos en polígonos de tiro, para incrementar la capacidad de maniobra con las misma.
- Diseñar un entorno virtual interactivo e intuitivo fácil de utilizar, que permita el desarrollo de las habilidades y destrezas necesarias para efectuar el entrenamiento de disparo.
- Desarrollar los algoritmos de programación que permitan la correcta conexión e interpretación de datos obtenidos por los sensores de posicionamiento del arma y el entorno de programación.
- Incrementar el grado de destreza que debe tener un miembro de las FF.AA. mediante el uso del simulador, para el cumplimiento de las operaciones de defensa del territorio nacional.
- Realizar pruebas simulador para verificar su eficacia.

Capítulo II

2. Marco teórico

2.1 Realidad virtual (RV)

En 1987 Jaron Lanier propuso el término realidad virtual (RV), junto con Tom Zimmerman se convirtieron en las principales figuras quienes aportaron en el desarrollo de esta tecnología (Time Graphics). El término RV tomó fuerza a principios de los 90 siendo este un tema muy popular dentro de las superproducciones de Hollywood. En la actualidad la RV se ha desarrollado tanto que es muy difícil poder diferenciar entre gráficos 3D y RV (EVE, Museos e Innovación, s.f.).

Se puede definir a la RV como la tecnología que a través de un visor transporta al usuario a escenarios virtuales, pero que tienen la característica de ser escenarios reales. En un principio se llamó RV inmersiva a la tecnología que para su ejecución requería de dispositivos electrónicos externos, en contraste la tecnología no inmersiva era aquella que se desarrollaba en entornos 3D, pero únicamente la interacción con el usuario se la realizaba a través de los periféricos del ordenador (Aznar, Romero, & Rodríguez, 2018).

Figura 1

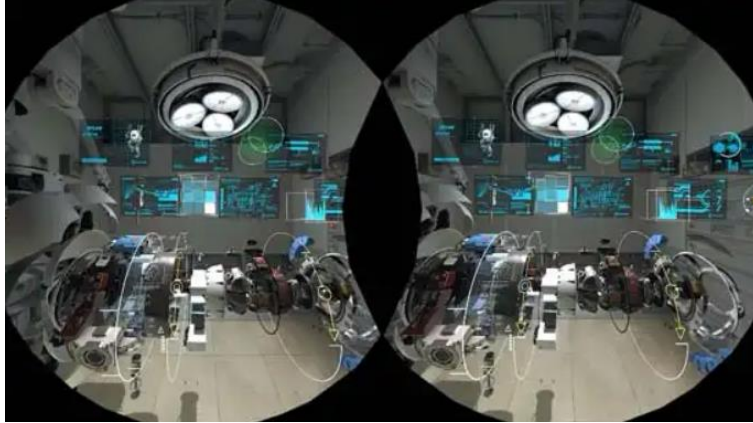
Usuario de visor de RV



Nota. La imagen mostrada muestra un usuario utilizando equipo de RV. Tomado de (Rodríguez, s.f.)

2.1.1 Tipos de RV**2.1.1.1 RV inmersiva.**

Este tipo de RV permite al usuario percibirse dentro del ambiente virtual tridimensional creado por ordenador, a través del uso de visores de RV y distintos accesorios que permiten la captura de los movimientos del usuario (Venturini, 2020).

Figura 2*Percepción de RV inmersiva*

Nota. La imagen mostrada es una representación de lo que percibe visualmente el usuario de visor de RV. Tomado de (Venturini, 2020)

2.1.1.2 RV no inmersiva.

Este tipo de RV únicamente necesita de un ordenador para poder navegar dentro del sistema, el sistema puede percibirse a través de la pantalla e interactuar a través de los periféricos del ordenador, debido a esta particularidad es el tipo de visión artificial más económica y de fácil acceso, haciéndolo el más popular y utilizado en video juegos (Venturini, 2020).

Figura 3

Equipo requerido para RV no inmersiva



Nota. La imagen mostrada es del equipo necesario para la RV no inmersiva. Tomado de (Venturini, 2020).

2.1.2 Aplicaciones de la RV

El uso de la RV cada vez es más aceptado, siendo el área terapéutica una de las más beneficiadas de esta tecnología. Dispositivos como los *Head Mounted Display (HDM)* han sido utilizados en varios campos como educación, salud e incluso entretenimiento, sin embargo, esta tecnología se convierte de difícil acceso para personas de edad avanzada, debido a la brecha tecnológica (Campo, 2021).

Un caso específico a la rama terapéutica es el mostrado en (Ivars & Suso, 2020) el cual indica que la RV puede ser un método efectivo ya que este ayuda a disminuir la ansiedad y dolor en los pacientes con quemaduras ya que esta tecnología trabaja sobre los aspectos cognitivos-atencionales, como respaldo a esto se tiene la hipótesis de que para que el paciente sienta dolor, este debe estar consciente o debe poner atención al mismo.

Otro de los enfoques de gran éxito es la educación su gran aporte a esta rama es indudable, para ello se toma como ejemplo el proyecto de la universidad de Murcia llamado *Cartago Nova*, este proyecto ha sido presentado en forma de video juego el mismo que fue ambientado en el Teatro Romano de Cartagena desarrollado para *Oculus Rift*. cuyas primeras pruebas han denotado un gran aporte para la educación e historia (Arias, Egea, & Alfonso, 2018).

El sector inmobiliario ha hecho uso de esta tecnología como estrategia de marketing y comercialización de bienes inmuebles, en esta modalidad se presenta la idea de tener al cliente como diseñador de su propio espacio, ya sea cambiando el color de las paredes, el tipo de piso que quiere para su vivienda, dándole al cliente la capacidad de modificar partes específicas. El objetivo de este tipo de marketing es influir en los cambios cognitivos, afectivos y conductuales del cliente, y puedan tener un vínculo con lo que están comprando (Vargas, 2019).

2.2 Plataformas de desarrollo

2.2.1 Unity

Unity se ha posicionado como uno de los principales motores de videojuegos en la actualidad, esta herramienta fue desarrollada por la empresa Unity Technologies para poder desarrollar videojuegos para diversas plataformas y además se lo ha utilizado en RV, su enfoque hacia el desarrollo de videojuegos lo hacen ideal para poder procesar sonidos, animaciones, inteligencia artificial, así como renderizar gráficos 2D y 3D junto con sus propiedades físicas (Asensio, 2019).

Figura 4

Logo de Unity



Nota. La imagen muestra el logotipo del motor de desarrollo de videojuegos Unity. Tomado de (Unity Technologies, s.f.).

La primera versión del popular motor de videojuegos apareció en el año 2005, sin embargo, la empresa *Unity Technologies fue fundada en el 2004* por David Helgason, Nicholas Francis y Joachim Ante. El enfoque principal de este grupo establecido en Copenhague, Dinamarca, era crear una plataforma lo más accesible a cualquier persona en el mundo. Unity tiene la característica de ser compatible con plataformas como Blender, 3ds Max, Z Brush además de esto el flujo de trabajo con estas herramientas es increíble ya que los cambios realizados desde estas herramientas se ven reflejados automáticamente dentro de Unity sin la necesidad de actualizar o volver a cargar la información (Román, Ortega, Gutierrez, & Nahuel, 2017).

2.2.2 Unreal Engine

La primera versión de *Unreal Engine* fue lanzado en el año 1998, el poderoso motor de videojuegos cuenta con cuatro versiones a la actualidad, pero no fue sino hasta el año 2005 donde se pudo usar la plataforma públicamente y de forma gratuita, al igual que *Unity*, Unreal tiene herramientas bien desarrolladas de sonido, edición de video y renderizado de animaciones.

Figura 5

Logo de Unreal Engine



Nota. La imagen muestra el logotipo del motor de desarrollo de videojuegos Unreal Engine. Tomado de (Epic-Games, s.f.)

Unreal Engine es una plataforma 100% gratuita y cualquier persona puede acceder a ella para desarrollar sus propios proyectos, sin embargo, si el proyecto se pretendió comercializar se deberá pagar el valor del 5% de las ganancias de los primeros 3000\$ a la empresa *Epic Games* quien es la que desarrollo *Unreal Engine*.

Figura 6

Gears of War 4



Nota. La imagen muestra una vista previa del popular videojuego Gears of War 4, desarrollado en Unreal Engine 4. Tomado de (Epic-Games, s.f.).

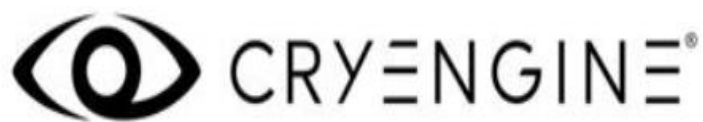
Entre los mejores exponentes de Unreal Engine es el icónico juego llamado Gears of War 4, la calidad del juego es una muestra de lo poderoso que puede ser este motor de videojuegos.

2.2.3 CryEngine

CryEngine de la empresa *Crytek*, fue lanzado en el año 2002, tiene una de las más recientes aportaciones para el mundo de los videojuegos, su más reciente versión es la 5.6, su mejor exponente es el famoso videojuego *Far Cry*, desarrollado por la misma empresa. A diferencia de Unity o Unreal, *Far Cry* es totalmente gratuito el aporte hacia los desarrolladores es voluntario y se lo realiza a través de donaciones (Rubi3D, 2018).

Figura 7

Logotipo de CryEngine



Nota. La imagen muestra el logotipo del motor de desarrollo de videojuegos CryEngine. Tomado (CryEngine, s.f.)

CryEngine alcanzó el éxito en poco tiempo, por lo cual la empresa *Ubisoft*, adquirió los derechos del reconocido motor de videojuegos, tiempo después lanzarían otro éxito mundial conocido como *Crysis*, que fue lanzado para las plataformas de PS3 y Xbox 360 con las versiones posteriores del CryEngine (CryEngine, s.f.).

Figura 8*Far Cry 5*

Nota. La imagen muestra una vista previa de Far Cry 5, el cual fue desarrollado en CryEngine. Tomado de (Play Station, s.f.)

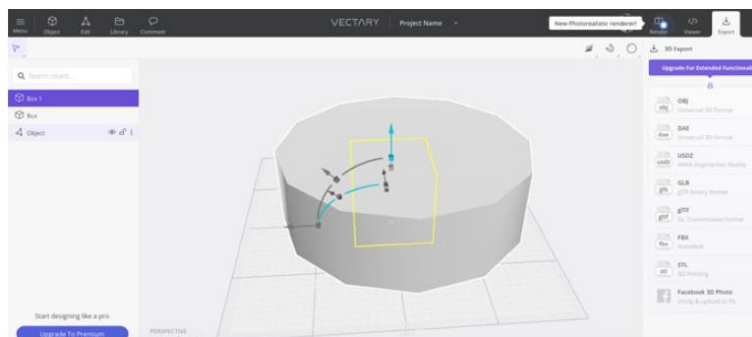
2.2.4 Modelado 3D

El modelado 3D tiene su origen en los años 70 cuando apareció la primera animación, aunque tuvo que desarrollarse mucho para llegar a su máxima expresión, el modelado 3D ha dado grandes saltos basado en rasgos tecnológicos que han alcanzado el éxito de tal forma que creó expectativa. En la actualidad el modelado 3D es un mundo virtual lleno de posibilidades en cuanto a la capacidad de generar imágenes y poder modificar en función de la necesidad del creador llevándolo a otro nivel (ESIC Business & Marketing School, s.f.).

El modelado 3D era utilizado en sus comienzos para el análisis y la ciencia que con el pasar del tiempo se convirtió en un vehículo para la creatividad aplicado a un sinnúmero de tecnologías y un primer largometraje de animación mediante computación gráfica (ESIC Business & Marketing School, s.f.).

Figura 9

Software Vectary - modelado 3D



Nota. La imagen de la figura muestra el modelado 3D de una pieza en el software Vectary. Tomado de (3D Natives, s.f.).

En un inicio las técnicas de modelado 3D aparentaban ser únicamente privilegio de ingenieros, en la actualidad las herramientas para modelado 3D se encuentran al alcance de cualquier persona ya que no se requiere de conocimiento teórico para desarrollarse, existen modeladores y artistas 3D especializados en diferentes campos como la medicina, arquitectura, ciencia ficción, videojuegos y muchas áreas más que presentan este tipo de requerimientos para un fin específico. En la actualidad existen muchos tipos de herramientas optimizadas para algún proceso (ESIC Business & Marketing School, s.f.).

2.2.5 Blender

Blender es una herramienta de modelado de código abierto y gratuita desarrollada por The Blender Foundation, además de modelado Blender tiene funciones de escultura, animación, simulación, renderizado y composición. Actualmente su última versión es la 2.92 disponible tanto para Windows como para Linux y Mac (The Blender Foundation, s.f.).

Figura 10

Logo de Blender

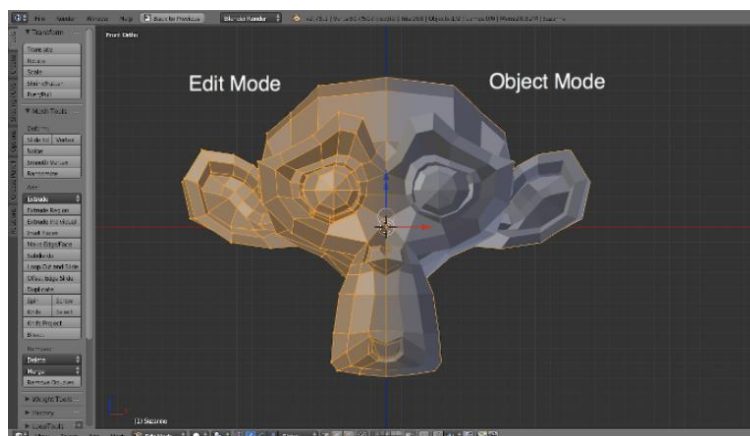


Nota. La imagen muestra el logotipo del software de modelado 3D, Blender. Tomado de (The Blender Foundation, s.f.).

Blender fue lanzado por primera vez en el año 1995, su gran acogida por parte de los usuarios principalmente es por el soporte y actualizaciones constantes gracias a que es de código abierto, blender está basado en el modelado poligonal, la *suite* completa de *Blender* es capaz de desarrollar proyectos completos en un mismo sitio. El flujo de trabajo y su uso son bastantes sencillos al poder trabajar en diferentes modos a la vez (The Blender Foundation, s.f.).

Figura 11

Modo objeto y modo edición de blender



Nota. La imagen muestra los modos de diseño que posee blender, en la parte izquierda está el modo edición donde se puede apreciar la cantidad de polígonos ocupados mientras que en la parte derecha está el modo objeto y se puede apreciar el resultado actual del modelo. Tomado de (3D Natives, s.f.).

2.2.6 3ds Max

El 3ds Max es un software de modelado y renderización para visualización de diseños desarrollado por la empresa Autodesk, el 3ds Max es un software dirigido hacia un público bastante amplio como modeladores, artistas 3D, decoradores y arquitectos, este software cuenta con un renderizador interno conocido como *Arnold* al igual que Blender, 3ds Max cuenta con una suite completa para el área de modelado, sin embargo, para su uso es necesario adquirir una licencia o suscripción anual.

Figura 12

Logo de 3ds Max



Nota. La imagen muestra el logotipo del software de modelado 3ds Max, Blender. Tomado de (Autodesk, s.f.).

3ds Max es uno de los *softwares* más reconocidos por los artistas 3D, su gran variedad de herramientas lo hacen el *software* perfecto para desarrolladores de videojuegos, esta aceptación por los usuarios proviene del nivel de detalle que se puede conseguir a través del *software* además de poder simular propiedades físicas de elementos, brindando la sensación de realismo buscado por los profesionales del área.

Figura 13

Renderizado con 3ds Max



Nota. La imagen muestra una escena y el nivel de detalle que puede llegar a tener 3ds Max junto con su potente renderizador (3D Natives, s.f.).

2.3 Placas de desarrollo

Una placa de desarrollo está compuesta por un dispositivo que cuenta con un microcontrolador, este tipo de placas se volvieron comerciales en los años setenta, con el microcontrolador Intel 4004 de cuatro bits, este fue el segundo CPU completo de un solo chip en la historia, a mediados del mismo siglo el microcontrolador de la empresa *Microchip Technology* se convirtió en la pionera en este mercado durante varios años, su éxito radica principalmente en su bajo costo, disponibilidad y alcance de herramientas de programación de *software* libre (Conocimiento libre, s.f.).

Estas placas tienen la característica de ser reprogramables, por lo que pueden ejecutar instrucciones específicas, estas cuentan con entradas y salidas de tipo digital y analógico, lo que le permite la interacción con sensores externos. El uso de las placas de desarrollo ha revolucionado la tecnología y el cómo se interactúa con el mundo, ya que con ellas se pueden realizar infinidad de proyectos como por ejemplo prototipos, robótica, automatización, video juegos entre otros. No existe un lenguaje específico

para poder dar las instrucciones a estas placas de desarrollo, se puede elegir entre los lenguajes de programación de preferencia por el usuario, teniendo opciones como C, C++, Arduino, *JavaScript*, *Python* entre otros. Algunas placas cuentan con un sistema operativo completo, generalmente que se basa en el núcleo de *Linux*, y sobre este pueden ejecutarse diversas aplicaciones, generalmente orientados a la electrónica.

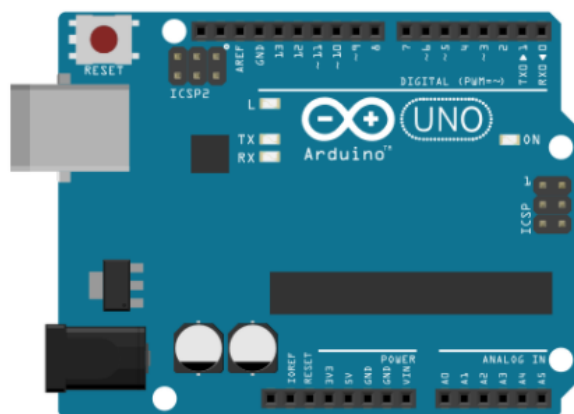
2.3.1 Modelos principales de placas de desarrollo

2.3.1.1 Arduino.

Esta es una placa que incorpora un microcontrolador de la familia AVR de la marca Atmel, así como también una serie de pines hembra que permiten la conexión con diferentes tipos de sensores y actuadores fácilmente (Arduino, s.f.). En la figura 14 se observa una de las placas de desarrollo de la marca Arduino.

Figura 14

Placa de desarrollo Arduino



Nota. La imagen muestra una tarjeta de desarrollo Arduino Uno, tomado de (Arduino, s.f.).

2.3.1.2 Raspberry Pi.

Básicamente esta tarjeta se trata de un ordenador de tamaño reducido y bajo costo, esta se puede conectar al monitor de un PC o una TV, acompañado de un ratón y teclado estándar, esta tarjeta tiene como sistema operativo a Linux, por lo que permite hacer tareas básicas de un computador, como es reproducir contenido multimedia, editar documentos de ofimática, entre otros. Además de esto, la Raspberry Pi tiene la característica de interactuar con distintos tipos de sensores, porque puede ser usada para gran cantidad de aplicaciones de electrónica (Raspberry, s.f.). En la figura 15 se observa la Raspberry Pi en su modelo A.

Figura 15

Tarjeta de desarrollo Raspberry Pi A



Nota: La imagen muestra una tarjeta de desarrollo Raspberry Pi (Raspberry, s.f.).

2.3.1.3 Onion.

La empresa Onion es fabricante de placas de desarrollo que se enfoca principalmente al internet de las cosas, donde su propósito es crear dispositivos que puedan interconectarse y así reducir el tiempo de desarrollo de prototipos y soluciones finales. Actualmente la empresa cuenta con un amplio catálogo de modelos, así como también extensiones para sus placas, donde se busca priorizar que tamaño reducido y

tengan un procesador de gran potencia capaces de correr sistemas operativos completos (Onion, s.f.). En la Figura 16 se observa la placa Onion Omega 2.

Figura 16

Placa de desarrollo Onion



Nota. La imagen muestra una tarjeta de desarrollo Onion, tomado de: (Onion, s.f.)

2.4 Polígono de tiro

Un polígono de tira es considerado como tal cuando reúne las características de tener los espacios, características y normas de seguridad que estén regulados por los entes pertinentes de cada país. Estos espacios pueden ser adecuados en terrenos que se ubican en las afueras de zonas urbanas debido a su peligrosidad, por lo que solo las personas que reúnan ciertas características psicológicas puedan acceder a ellos.

Por lo general los polígonos de tiro son usados por los cuerpos policial y militar, ya que ellos tienen la responsabilidad de brindar seguridad a la ciudadanía. Para efectuar las prácticas en los polígonos de tiro el personal debe llevar a cabo cursos de capacitación sobre seguridad del uso de armas (APPA, 2015).

2.4.1 Tipos de polígono de tiro

- **Real:** Consiste en un terreno generalmente descubierto donde se pueden realizar las practicas, donde existen tiros de corta, mediana y larga distancia.

- **Reducido:** en estos tipos de polígonos se pueden realizar prácticas de tiro con municiones reales o con municiones de salva, al ser un lugar pequeño la practica solo consiste en tiros de corta distancia.
- **Polígono de tiro láser:** su principal característica es que los tiros son simulados por un rayo láser, estos son considerados como polígonos de realidad virtual, constituyéndose como un simulador.
- **Interiores:** generalmente se los implementa en construcciones, aquí se implementan dianas, siluetas y trampas, además de protecciones para los usuarios como iluminación y aislamiento acústico.
- **Exteriores:** por lo general son implementados en bastos terrenos de hasta mil metros, aquí se permite el uso de armas de largo alcance

2.5 Armas de bajo calibre

Un arma de fuego, es un instrumento mecánico que envía un proyectil propulsado por una explosión de pólvora a alta velocidad a través de un tubo (Texas Parks & Wildlife, s.f.).

Figura 17*Arma bajo calibre Glock*

Nota. La figura muestra un arma de bajo calibre del fabricante Glock.

Son aquellas que están destinadas al uso personal, por su tamaño pueden ser disparadas con una sola mano, estas por lo general tienen poco peso, y se adaptan al tamaño de una mano, su calibre está determinado a un tubo de cañón menor a los veinte milímetros.

2.5.1 Reglas principales para su uso seguro

- Apuntar siempre a una dirección segura
- Toda arma de fuego debe ser manejada siempre como si estuviera cargada
- Mantener el dedo fuera del gatillo hasta que esté listo para disparar
- Asegurar y observar el blanco, así como también lo que está al frente y detrás de este.

2.5.2 Principios básicos de tiro

- **Estado psicológico:** está basado principalmente en la confianza que tiene en sí mismo el tirador.

- **Concentración mental:** esta debe estar enfocada totalmente en la actividad que se está realizando.
- **Postura:** para disparar cualquier arma es muy importante tener una buena postura, principalmente influye la firmeza y sujeción del arma.
- **Relajamiento:** aquí se debe tener en cuenta que los músculos no estén tensionados
- **Empuñadura:** existen reglas para cada tipo de arma, pero durante el aprendizaje es conveniente que el soldado acostumbre a tomar una posición correcta.
- **Alineación de miras:** básicamente consiste en centrar la mira con el orificio del alza
- **Respiración:** es importante que durante la ejecución del disparo el soldado aprenda a controlar su respiración, ya que el movimiento del pecho puede afectar la dirección del disparo, éste se debe efectuar con los pulmones libres de aire.

2.5.3 Armas utilizadas por el Ejército Ecuatoriano

Existe gran variedad de marcas de fabricantes de armas que utilizan las FF.AA. del Ecuador, pero las principales son las siguientes:

- Fusil H&K
- Sub ametralladora UZI
- Pistola H&K USP
- Pistola Pietro Beretta

Figura 18*Pietro Beretta 92 FS*

Nota. La figura muestra un arma Pietro Beretta 92 FS

La Pietro Beretta es un arma legendaria, se la ha catalogado como la mejor pistola de servicio que ha sido utilizada por las fuerzas militares y policiales de varios países del mundo, una de sus características es que tiene gran capacidad de munición, además de un diseño ergonómico, se la puede observar en la Figura 18

2.6 Entrenamiento militar

Este tipo de entrenamiento significa en lograr adquirir ciertas habilidades físicas para poder intervenir en combate, además de adquirir destrezas necesarias para sobrevivir en el marco de conflictos bélicos, por lo general las capacidades que se deben desarrollar es en uso de armas y supervivencia (RAE, s.f.).

Actualmente las FF.AA. del Ecuador disponen a su personal en diferentes tipos de cursos de perfeccionamiento en estrategia de tipo terrestre, administración, planificación, educación, entre otras que ayudan a mejorar las competencias establecidas en su perfil profesional (Andrade & Yandún, 2020).

Figura 19

Personal FF.AA. del Ecuador



Nota. La imagen muestra una foto del informe DIMT del Ejército Ecuatoriano.

Los principales puntos de enfoque de las FF.AA. del Ecuador son:

- Ejercer el control del territorio nacional
- Apoyar a las instituciones en la protección de la población
- Fortalecer las capacidades estratégicas de las FF.AA.
- Contribuir a cooperación internacional
- Contribuir al desarrollo de la nación

2.6.1 Costo del entrenamiento militar

Para un país el gasto militar se lo puede considerar como una póliza de seguro que estaría a disposición de proteger la integridad territorial contra riesgos externos, pero este seguro lleva consigo un costo que se afronta a contra peso con las necesidades de los ciudadanos (Scheetz, 2011).

Cada año el país gasta alrededor de 2.37% de PIB en presupuesto militar, siendo que para el año 2018 esta cifra llegó a los 2.5 mil millones de dólares, empleando aproximadamente 41.000 armas de fuego, se estima que por cada 409 ciudadanos existe un efectivo militar (Revista Gestión, s.f.).

2.6.2 Costo del entrenamiento en polígonos de tiro

Según los datos de la Escuela Superior Militar (ESMIL) Eloy Alfaro al personal se lo somete a dos prácticas de tiro anuales, donde se detalla que en cada práctica se ponen a disposición cien municiones por cadete, donde el costo de cada munición es de setenta centavos de dólar, dando un total de setenta dólares por cadete, y para toda la escuela el valor asciende a USD \$ 83.720 dólares (Encalada, 2011).

Figura 20

Polígono de tiro de las FF.AA.



Nota. La figura muestra un polígono de tiro utilizado por las FF.AA. del Ecuador.

Capítulo III

Desarrollo del sistema mecánico y electrónico

En el presente capítulo se detalla la fase correspondiente a la selección y evaluación de componentes tanto a nivel físico como de software necesarios para llevar a cabo la construcción del sistema de entrenamiento, enfocado a los polígonos de tiro mediante RV. Para esto es de suma importancia conocer las características técnicas necesarias para realizar el entrenamiento, así como las necesidades de los usuarios, por lo que se plantearán diferentes propuestas que serán comparadas y evaluadas con el fin de seleccionar los elementos adecuados, el diseño será definido y finalmente se corregirán errores de ser necesario, todo esto basado en la metodología del diseño concurrente (Riba Romeva, 2006).

En primer lugar se realizará la selección del software destinado a la creación del entorno de RV, ya que de este se derivarán los diferentes medios necesarios para llevar a cabo el entrenamiento en sus diferentes etapas, como son selección de visores RV, compatibilidad a modo de desarrollador, soporte brindado por parte del fabricante, características del hardware necesarias para poder ejecutar el programa sin problemas, evaluando así sus ventajas y desventajas, para mediante la metodología de ponderación se seleccionará la mejor opción.

En segundo lugar, se realizará la selección de los componentes del hardware, siendo estos sensores, actuadores, y visores de RV los mismos que serán los encargados de llevar a cabo la lectura y visualización de movimiento del mando en el entorno de RV, de igual manera evaluando sus ventajas y desventajas mediante la metodología de ponderación para seleccionar la mejor opción. Finalmente se realizará la selección de la técnica de manufactura más adecuada para la fabricación del mando.

3.1 Descripción de necesidades

La descripción de necesidades para el proceso de desarrollo del producto es parte fundamental en la elaboración del mismo, ya que aquí es donde se deben identificar las necesidades y características que éste debe poseer para llevar a cabo el propósito del proyecto, que es el entrenamiento mediante RV enfocado a los polígonos de tiro. Las necesidades que se tomaron en consideración se las obtuvo mediante una charla con personal militar, en donde se pudo conocer las principales características y necesidades que debe constituir un polígono de tiro, con esta información se estructuró la Tabla 1 Descripción de necesidades.

Tabla 1

Descripción de necesidades

Concepto	No.	Necesidad
Función	1	Similitud a polígono de tiro real
	2	Uso intuitivo
	3	Control del puntaje obtenido
Construcción	4	Modelado del entorno RV
	5	Llevar los movimientos al entorno RV
	6	Base de datos
	7	Mando similar al usado en entrenamientos
Control	8	Llevar los movimientos al entorno virtual
Costo	9	Bajo costo

3.2 Arquitectura del diseño

Los elementos por lo que está constituido el sistema de entrenamiento militar enfocado a los polígonos de tiro para mejorar la destreza del personal de Las FF.AA. del Ecuador se muestran en la Figura 21. En donde el funcionamiento del sistema está

dado en primera instancia por que el usuario se registre en el sistema, posteriormente toma el mando, el cual tendrá los componentes electrónicos necesarios para leer, procesar y enviar los datos hacia el computador, el cual procesara la información y la llevara al entorno virtual que es mostrado a través de las gafas de RV que deberán estar conectadas al PC, y finalmente el sistema es retroalimentado al seguir los movimientos del usuario, dicha tarea es llevada a cabo gracias a los sensores incorporados en las gafas de RV así como también los movimientos que registre el mando.

Figura 21

Arquitectura del diseño



Establecida la arquitectura del diseño se puede crear una arquitectura modular del sistema, para permitir que durante la etapa de desarrollo del proyecto se pueda evaluar y modificar de ser necesario los componentes en sus diferentes etapas, en la Tabla 2 se describen los módulos por los que estará constituido el sistema.

Tabla 2*Arquitectura modular*

#	Módulo	Funciones
1	Entorno virtual	Encargado de simular el entrenamiento en un polígono de tiro
2	Estructura mecánica	Mando similar al utilizado en entrenamientos
3	Sensores y actuadores	Adquieren las señales para el procesamiento de movimientos llevador al entorno RV.
4	Control	Sistema de adquisición y procesamiento de señales
5	Suministro de energía	Alimenta los componentes electrónicos del sistema

3.3 Metodología de selección de componentes

Con el fin de seleccionar la alternativa más adecuada para la creación del sistema de entrenamiento militar, se utilizará la metodología del sistema de puntuación, que consiste en ponderar los factores más relevantes. Esta herramienta permitirá comparar y ordenar en forma cuantitativa las alternativas, para de esta forma tomar la mejor decisión de selección (Urbina, 2013).

3.3.1 Selección del software**3.3.1.1 Software de desarrollo del entorno de RV.**

Para determinar el software que preste las mejores características para el diseño del entorno de RV se tomará en cuenta las siguientes características:

- Licencia
- Compatibilidad con RV
- Manuales

- Interfaz grafica
- Lenguaje de programación
- Coste computacional

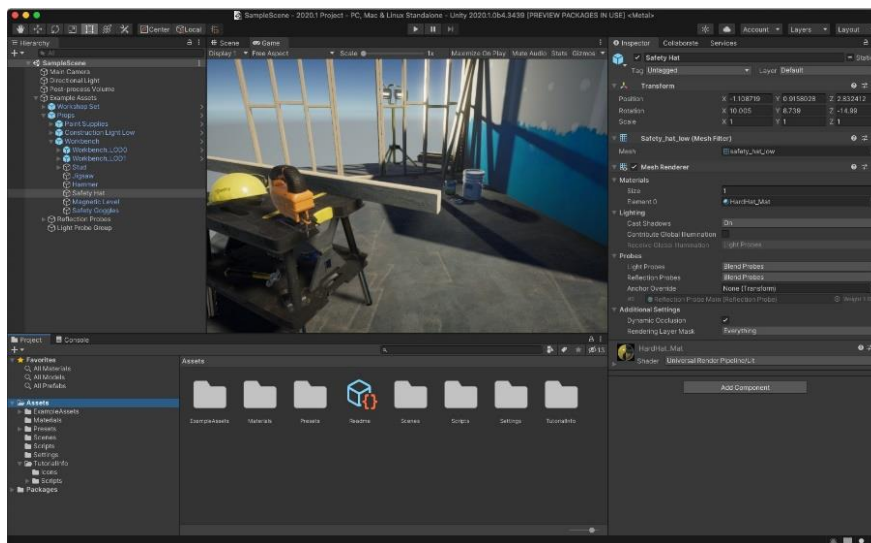
3.3.1.2 Unity 3D.

Unity 3D es un motor de videojuegos para Pc y MAC utilizado como una herramienta para crear videojuegos, aplicaciones interactivas, visualizaciones y animaciones tanto 2D como 3D y tiempo real (Unity, 2020).

En la Figura 22 se observa el entorno del motor de desarrollo Unity 3D en si versión 2020.

Figura 22

Entorno de desarrollo Unity 3D



Nota. Tomado de (Unity, 2020).

El editor de Unity ofrece una completa gama visual para la creación de aplicaciones, las cuales se van construyendo desde el editor, que se programan mediante lenguaje de scripts, lo que quiere decir que los desarrolladores

necesitan conceptos básicos de programación en C++. El juego en Unity (Unity, 2020).

En la tabla 3 se describen las características necesarias para la instalación de Unity 3D.

Tabla 3

Características necesarias para la instalación de Unity 3D

Requisitos mínimos	
Sistema operativo	Windows 7 SP1+, 8,10
CPU	Arquitectura X64 con soporte de conjunto de instrucciones SSE2
Memoria RAM	8 GB
Espacio en disco	100 MB de espacio en disco
Tarjeta gráfica	Tarjeta gráfica con capacidades DX10

Nota. Tomado de (Unity 3d, 2020).

Ventajas y desventajas que presenta el motor de desarrollo Unity 3d.

Ventajas

- Documentación disponible, manuales y tutoriales.
- Curva de aprendizaje rápida.
- Gran variedad de plataformas de publicación.
- Manejo de lenguaje de programación basado en C++.
- Disponibilidad de tienda de *assets*.

Desventajas

- Uso excesivo de memoria del disco duro.

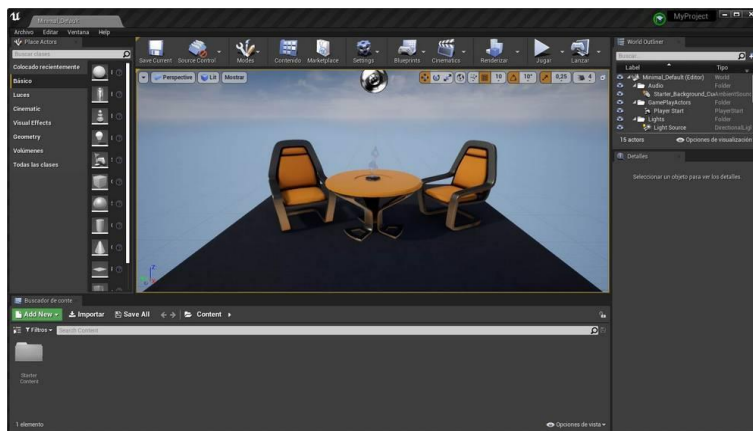
- Perdida del rendimiento en proyectos complejos.
- Mala gestión de versiones.

3.3.1.3 Unreal Engine.

Unreal Engine es uno de los motores de creación de videojuegos basado en código C++, este software incluye todas las herramientas necesarias para la creación de aplicaciones, como son, editor de video, estudio de sonido, renderización de animaciones, entre otras (UnrealEngine, 2020). En la Figura 23 se observa el entorno de Unreal Engine en su versión 4.

Figura 23

Entorno Unreal Engine 4



Nota. Tomado de (UnrealEngine, 2020)

En la tabla 4 se muestran los requisitos mínimos para la instalación de Unreal Engine 4.

Tabla 4

Características necesarias para la instalación de Unreal Engine 4

Requisitos mínimos	
Sistema operativo	Windows 7 SP1+, 8,10

Requisitos mínimos	
CPU	Intel o AMD 2.5 GHz 4 núcleos o superior 64 Bits
Memoria RAM	8 GB
Espacio en disco	15GB de espacio en disco
Tarjeta gráfica	NVIDIA GeForce GTX 470 o AMD Radeon 6870 HD series

Nota. Tomado de (UnrealEngine, 2020).

Ventajas y desventajas que presenta el motor de desarrollo Unreal Engine 4.

Ventajas:

- Licencia gratuita
- Lenguaje de programación C++
- Variedad de plataformas de publicación
- Ambiente grafico en 3D
- Completo kit de desarrollo

Desventajas

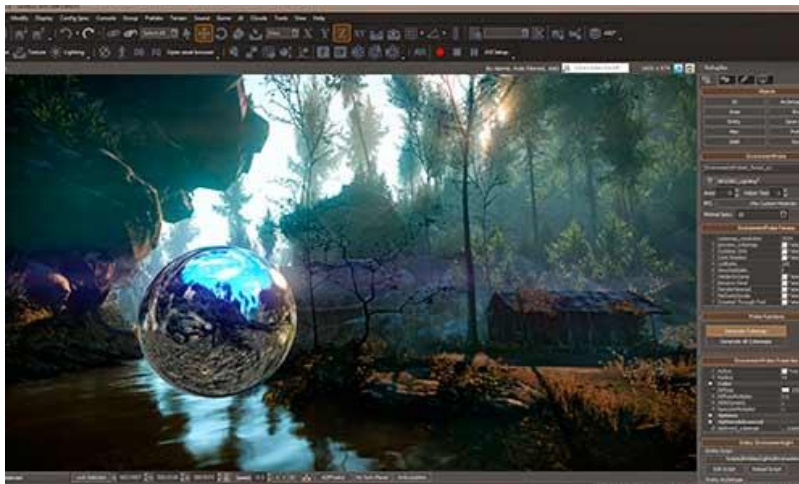
- Peso del proyecto.
- Poca documentación.
- Difícil aprendizaje.
- Comunidad de desarrolladores reducida.

3.3.1.4 CryEngine 3.

CryEngine 3 es un entorno de desarrollo de videojuegos que soporta plataformas como *Windows*, *Xbox 360* y *Play Station 3*, una de sus principales características es la calidad de gráficos y renderizado de gran calidad que este puede generar. En la Figura 24 se puede observar el entorno de desarrollo de CryEngine 3.

Figura 24

Entorno de desarrollo CryEngine 3



Nota. Tomado de (CryEngine, s.f.)

En la tabla 5 se muestran los requisitos mínimos para la instalación de CryEngine 3.

Tabla 5

Características necesarias para la instalación de CryEngine 3

Requisitos mínimos	
Sistema operativo	Windows 7 SP1+, 8,10
CPU	Intel Quad-Core (i5 2300) o AMD Octa-Core (FX 8150)
Memoria RAM	8 GB
Espacio en disco	8 GB de espacio en disco
Tarjeta gráfica	NVIDIA GeForce 660Ti o superior, AMD Radeon HD 7950 o superior

Nota. Tomado de (CryEngine, 2020)

Ventajas y desventajas que presenta el motor de desarrollo CryEngine 3.

Ventajas:

- Licencia gratuita

- Tienda disponible gratuita
- Gran calidad en gráficos 3D
- Comunidad de desarrolladores activa
- Buena optimización de espacio en disco

Desventajas

- Interfaz y herramientas muy complejas
- Poca documentación.
- Difícil aprendizaje.
- Pocas plataformas soportadas

3.3.2 Selección del software de desarrollo del entorno de RV

En la tabla 6 se detalla la calificación y evaluación de las alternativas de selección para el software de desarrollo del entorno de RV.

Tabla 6

Matriz de selección para el software de desarrollo

Criterio de selección	P%	Unity 3D		Unreal Engine		CryEngine	
		Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Licencia	10	5	0.5	5	0.5	5	0.5
Compatible con RV	50	5	2.5	5	2.5	5	2.5
Manuales	10	4	0.5	2	0.2	1	0.1
Interfaz grafica	10	4	0.4	3	0.3	2	0.2
Lenguaje de programación	10	3	0.3	4	0.4	2	0.2

		Unity 3D	Unreal Engine	CryEngine			
Coste computacional	10	2	0.2	2	0.2	3	0.3
Total	100		4.3		4.1		3.8
Resultado			Si		No		No

Según los resultados de la matriz de comparación para la selección del software de desarrollo del entorno virtual, la mejor opción es utilizar Unity 3D.

3.3.2.1 Software de modelado 3D.

Para determinar el software que preste las mejores características para el modelado 3D se tomará en cuenta las siguientes características:

- Licencia
- Instalación
- Manuales
- Interfaz grafica
- Tutoriales
- Coste computacional

Parte fundamental del polígono de tiro implementado con RV es el aspecto gráfico, ya que este dará la impresión de realismo dentro del entorno, por lo que se consideraron varias propuestas de programas para su uso, de los cuales se han considerado los siguientes:

3.3.2.1.1 Blender.

Blender es un programa de creación 3D gratuita y de código abierto, que está enfocado al modelado, iluminación, renderizado, animación de gráficos 3D. Este es un programa multiplataforma que cuenta con numerosas herramientas que ayudan a obtener resultados gráficos de gran calidad (Blender, s.f.). En la Figura 25 se puede observar el entorno de desarrollo de Blender.

Figura 25

Entorno de desarrollo de Blender 3D



Nota. Tomado de (Blender, s.f.)

Los requerimientos mínimos necesarios para la instalación de Blender se presentan en la tabla 7 a continuación.

Tabla 7

Características necesarias para la instalación de Blender

Requisitos mínimos	
Sistema operativo	Windows 7 SP1+, 8, 10
CPU	CPU de 2 Ghz de doble núcleo de 64 bits con soporte SSE2
Memoria RAM	4 GB
Pantalla	Resolución 1280 x 768
Tarjeta gráfica	Tarjeta gráfica con 1 GB de RAM, OpenGL 3.3

Nota. Tomado de (Blender, s.f.)

Ventajas y desventajas que presenta el software de modelado Blender.

Ventajas:

- Licencia gratuita
- Es multiplataforma
- Basta información en tutoriales
- Consume pocos recursos del computador
- Se puede programar con Python
- Posee varias herramientas de modelado

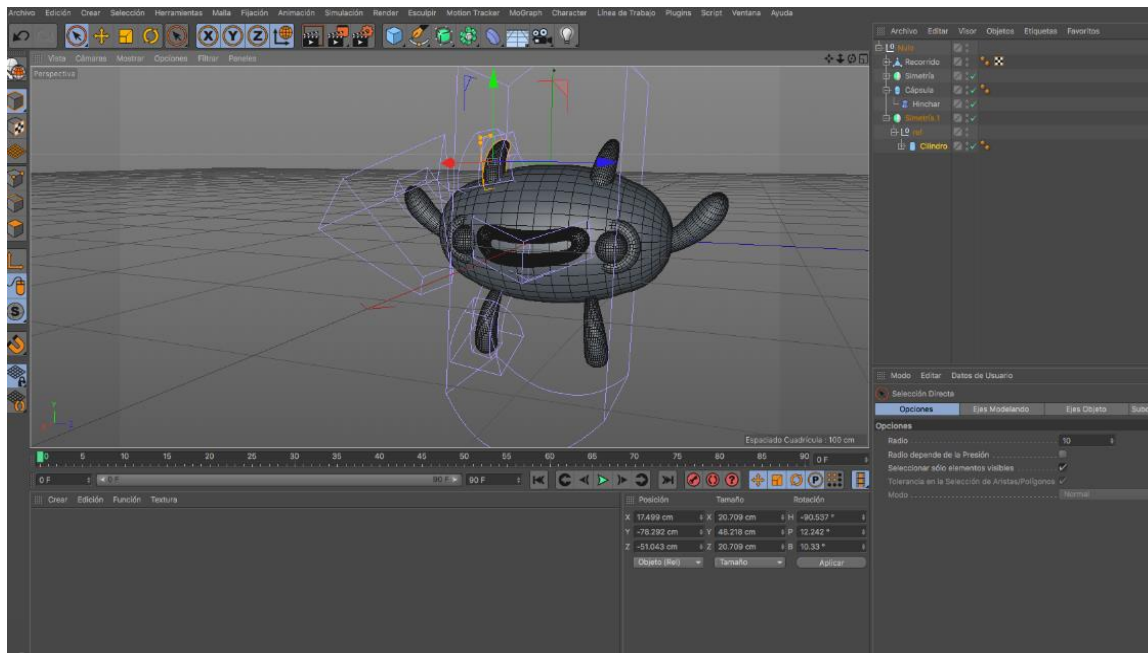
Desventajas

- No utiliza todos los procesadores del computador
- Defectos de modelado en polígonos mayores a cuatro lados
- Interfaz un poco confusa para principiantes
- Poca fuente de ingresos para contratar desarrolladores

3.3.2.1.2 Cinema 4D.

Cinema 4D es un software de modelado 3d, renderizado y animación, que tiene gran aceptación en la industria del diseño gráfico y animaciones gracias a sus características de *motion graphics*. En la Figura 26 se puede observar el entorno de desarrollo de Cinema 4D.

Figura 26

Entorno de desarrollo de Cinema 4D

Nota. Tomado de (Maxon, s.f.)

Este software tiene a su disposición un potente modelador y una gran gama de herramientas y efectos especiales para la representación de proyectos tridimensionales que pueden ser aplicados a las ramas de la ingeniería y arquitectura, incluso sus simulaciones se pueden utilizar para experimentos científicos, desarrollo de entornos virtuales y hasta efectos para el cine y la televisión (EcuRed, Ecured, s.f.).

Los requerimientos mínimos necesarios para la instalación de Cinema 4D se indican en la tabla 8.

Tabla 8

Características necesarias para la instalación de Cinema 4D

Requisitos mínimos	
Sistema operativo	Windows 10, MacOS: 10.14.6 o superior
CPU	CPU de 2 Ghz de doble núcleo de 64 bits con soporte SSE2
Memoria RAM	4 GB
Pantalla	Resolución 1280 x 768
Tarjeta gráfica	Tarjeta gráfica OpenGL compatible con OpenGL 4.1 o posterior (se recomienda GPU dedicada)

Nota. Tomado de (Maxon, s.f.)

Ventajas y desventajas que presenta el software de modelado Cinema 4D.

Ventajas

- Fácil aprendizaje
- Interfaz intuitiva
- Manual de uso incluido en el programa
- Constantes actualizaciones
- Gran velocidad de renderizado
- Múltiples herramientas de modelado

Desventajas

- Licencia de pago

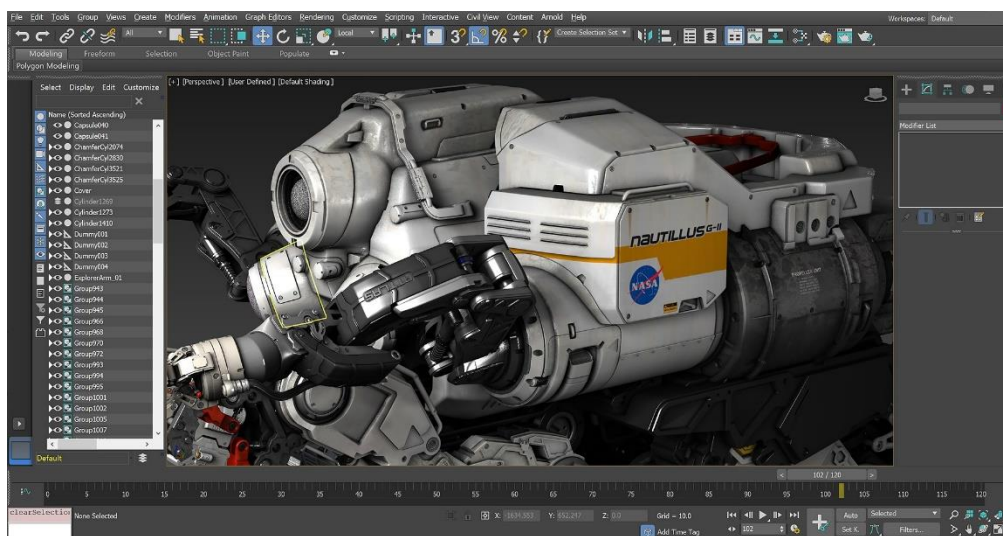
3.3.2.1.3 Autodesk 3ds Max.

Autodesk 3ds Max es un software de modelado, renderización y animación de calidad profesional, este proporciona varias herramientas para la creación de modelos

3d en el menor tiempo posible (Autodesk, s.f.). En la figura 27 se muestra el entorno de desarrollo del software 3ds Max.

Figura 27

Entorno de desarrollo de 3ds Max



Nota. Tomado de (Autodesk, s.f.)

Este software de uno de los más utilizados en cuanto a la animación 3D se refiere, ya que tiene a su disposición una gran capacidad de edición, su mayor campo de aplicación es por los desarrolladores de videojuegos, siendo también utilizado en proyectos de animación como películas o comerciales televisivos (EcuRed, s.f.).

Los requerimientos mínimos necesarios para la instalación de 3ds Max se indican en la tabla 9.

Tabla 9

Características necesarias para la instalación de 3ds Max

Requisitos mínimos	
Sistema operativo	Windows® 7 (SP1), 8, 8.1 y Windows 10 Professional
CPU	Procesador de varios núcleos de 64 bits Intel® o AMD® con el conjunto de instrucciones SSE4.2
Memoria RAM	4 GB
Espacio en disco	9 GB
Tarjeta gráfica	Tarjeta gráfica Nvidia quadro K1200 o superior.

Nota. Tomado de (Maxon, s.f.)

Ventajas y desventajas que presenta el software de modelado 3ds Max.

Ventajas

- Gran cantidad de herramientas de edición
- Variedad de información disponible
- Gran cantidad de *plugins* disponibles
- Permite la integración con Adobe Photoshop
- Compatible con múltiples formatos

Desventajas

- Licencia de pago
- Difícil de usar
- Solo funciona en Windows

3.3.3 Selección del software de modelado 3D

En la tabla 10 se detalla la calificación y evaluación de las alternativas de selección para el software de modelado 3D.

Tabla 10

Matriz de selección para el software de desarrollo

Criterio de selección	Peso	Blender 3D		Cinema 4D		3ds Max	
		Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Licencia	10%	5	0.5	2	0.2	2	0.2
Instalación	10%	5	0.5	3	0.3	4	0.4
Manuales	20%	4	0.8	4	0.8	3	0.6
Interfaz grafica	30%	4	1.2	4	1.2	3	0.9
Tutoriales	10%	4	0.4	4	0.4	3	0.3
Coste computacional	20%	4	0.8	4	0.8	3	0.8
Total	100%		4.2		3.7		3.2
Resultado			Si		No		No

Según los resultados de la matriz de comparación para la selección del software de modelado 3d, la mejor opción es utilizar Blender 3D.

3.3.4 Selección del hardware

3.3.4.1 Visores de RV para PC.

Para seleccionar los visores de RV se tomará en cuenta las siguientes características:

- Costo
- Instalación
- Manuales
- Compatibilidad a modo de desarrollador
- Compatibilidad de nuevos mandos
- Tutoriales
- Coste computacional

3.3.4.1.1 HTC Vive Pro.

Los visores de RV HTC Vive Pro son un dispositivo que se diseñó para ser utilizado en un espacio físico y sumergirse en el mundo virtual, ya que estas permiten caminar y usar controladores de interacción con objetos virtuales, brindando una experiencia fiel a la realidad (Vive, s.f.).

En la figura 28 se observan las gafas de RV HTC Vive Pro.

Figura 28

Visores de RV HTC Vive Pro



Nota. Tomado de (Vive, s.f.)

Los visores de RV HTC Vive Pro cuentan con dos pantallas AMOLED con una resolución de 2.880 x 1.600 píxeles a una tasa de refresco de 90 Hz permitiendo una calidad de imagen con gran definición, además cuenta con altavoces integrados, que junto a su diseño ergonómico los hacen muy cómodos de usar (Vive, s.f.).

En la tabla 11 se muestran los requisitos mínimos necesarios del computador para ser utilizados

Tabla 11

Requisitos mínimos del computador para utilizar las HTC Vive Pro

Requisitos mínimos	
Sistema operativo	Windows® 7, 8.1 y Windows 10
CPU	Intel® Core™ i5-4590 o AMD FX™ 8350
Memoria RAM	4 GB
Puertos USB	1x USB 3.0
Salida de video	DisplayPort 1.2
Tarjeta gráfica	NVIDIA® GeForce® GTX 970 o AMD Radeon™ R9 290

Nota. Tomado de (Maxon, s.f.)

Ventajas y desventajas que presentan las HTC Vive Pro

Ventajas:

- Gran resolución de pantalla
- Alta tasa de refresco
- Cuenta con giroscopio
- SDK (kit de desarrollo de software) compatible con Unity
- Diseño ergonómico

- Se puede controlar remotamente

Desventajas

- Alto costo
- Coste computacional alto
- Compatibilidad solo con mandos del fabricante

3.3.4.1.2 HP Reverb.

Estos visores fueron diseñados para ofrecer una excelente ergonomía, ya que a partir de su comodidad dependerá la experiencia de uso, estos tienen un peso de 500 gramos, ofrece una resolución muy alta e incluye auriculares y un doble micrófono compatible con asistentes inteligentes, con el fin de brindar una experiencia totalmente inmersiva (McPro, s.f.).

En la Figura 29 se observan los visores de RV Hp Reverb.

Figura 29

Visores de RV Hp Reverb



Nota. Tomado de (McPro, s.f.)

Estos visores cuentan con mandos emparejados de fábrica, y además es compatible con Windows Mixed Reality, lo que hace que sean un dispositivo plug and play, la resolución de pantalla es de 2160x2160 por cada ojo mediante un panel LCD, lo que hace tener una imagen muy nítida, además de contar con un ángulo de visión de 114 grados asegurando una experiencia envolvente e inmersiva (Hp, s.f.). En la tabla 12 se muestran los requisitos mínimos necesarios del computador para ser utilizados.

Tabla 12

Requisitos mínimos del computador para utilizar las Hp Reverb

Requisitos mínimos	
Sistema operativo	Windows 10
CPU	Core i5 (7 ^a generación), con 2 núcleos físicos con tecnología Intel Hyper-Threading
Memoria RAM	8 GB
Puertos USB	1x USB 3.0
Salida de video	HDMI 2.0
Tarjeta gráfica	Intel HD Graphics 620 integrada o GPU integrada superior con DX12.

Nota. Tomado de (Hp, s.f.)

Ventajas y desventajas que presentan las HTC Vive Pro

Ventajas

- Gran resolución de pantalla
- Cámaras de seguimiento de movimientos
- Compatible con diferentes plataformas
- SDK (kit de desarrollo de software) compatible con Unity
- Diseño ergonómico

- Alta tasa de refresco

Desventajas

- Alto costo
- Coste computacional alto
- Compatibilidad solo con mandos del fabricante

3.3.4.1.3 Oculus Rift.

Oculus Rift son unos visores de RV que tienen un diseño suave y cómodo, que proporcionan un amplio campo de visión diseñados para recrear diferentes perspectivas en función de los movimientos de la cabeza, pudiendo así ofrecer un seguimiento de 360 grados y una experiencia inmersiva (Oculus, s.f.).

En la Figura 30 se observan los visores de RV Oculus Rift.

Figura 30

Visores de RV Oculus Rift



Nota. Tomado de (Oculus, s.f.)

Estos visores cuentan con una pantalla Oled de resolución 24.160 x 1.200 pixeles con un ángulo de visión de 110 grados a una tasa de refresco de noventa Hz, además de sensores de movimiento como giroscopio, acelerómetro, magnetómetro y sistema posicional de 360 grados (Oculus, s.f.). En la tabla 13 se muestran los requisitos mínimos necesarios del computador para ser utilizados.

Tabla 13

Requisitos mínimos para utilizar las RV Oculus Rift

Requisitos mínimos	
Sistema operativo	Windows 7 SP1 64 bits ó superior
CPU	Intel i5-4590 o superior
Memoria RAM	8 GB
Puertos USB	3 puertos USB 3.0 más 1 puerto USB 2.0
Salida de video	HDMI 2.0
Tarjeta gráfica	NVIDIA GTX 970 / AMD R9 290

Nota. Tomado de (Oculus, s.f.)

Ventajas y desventajas que presentan las HTC Vive Pro

Ventajas

- Gran resolución de pantalla
- Sensores de seguimiento de movimientos
- Compatibilidad de mandos
- SDK (kit de desarrollo de software) compatible con Unity
- Diseño ergonómico
- Alta tasa de refresco
- Documentación para desarrolladores
- Existen varias versiones comerciales

- Bajo costo

Desventajas

- Limitadas unidades en el país
- Venta de mandos por separado

3.3.4.1.4 Selección de los visores de RV.

En la tabla 14 se detalla la calificación y evaluación de las alternativas de selección para los visores de RV.

Tabla 14

Matriz de selección para los visores de RV

Criterio de selección	HTC Vive Pro			Hp Reverb		Oculus Rift	
	Peso	Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Costo	20%	2	0.4	2	0.4	5	1
Instalación	10%	3	0.3	3	0.3	4	0.4
Manuales	10%	2	0.2	3	0.3	4	0.4
Compatible modo desarrollador	30%	4	1.2	4	1.2	5	1.5
Compatible con mandos	10%	2	0.2	2	0.2	5	0.5
Tutoriales	10%	3	0.3	2	0.2	4	0.4
Coste computacional	10%	4	0.4	4	0.4	2	0.2
Total	100%		2.0		3.0		4.4
Resultado			No		No		Si

Según los resultados de la matriz de comparación para la selección de los visores de RV son las Oculus Rift.

3.3.4.2 Tarjeta de control.

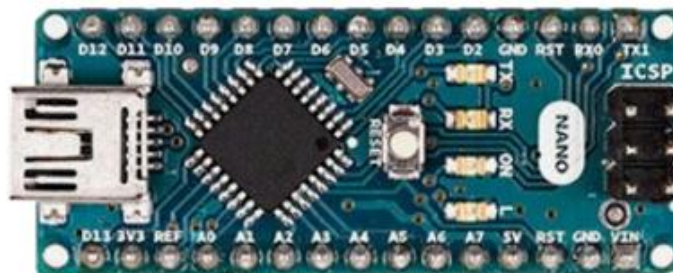
Para seleccionar la tarjeta de control se tomará en cuenta las siguientes características:

- Comunicación USB
- Costo
- Dimensiones
- Numero de pines
- Puerto de programación
- Tutoriales

3.3.4.2.1 Arduino Nano.

Arduino Nano es una tarjeta de desarrollo de tamaño reducido, pero completa, ya que posee las misma capacidad que el Arduino Uno, está basada en el microcontrolador Atmega 328p, este no dispone de pines para alimentación externa, su programación y alimentación de tensión se la realiza mediante puerto Mini-USB (Arduino, s.f.).

En la Figura 31 se observa la tarjeta Arduino Nano.

Figura 31*Arduino Nano*

Nota. Tomado de (Arduino, s.f.).

En la tabla 15 se muestran las características de la tarjeta Arduino Nano.

Tabla 15*Características técnicas Arduino Nano*

Características técnicas	
Microcontrolador	ATmega328p
Voltaje de operación	5 V
Voltaje de entrada	7-12 V
Corriente E/S	40 mA
Velocidad de reloj	16MHz
Pines E/S analógicos	8
Pines E/S digitales	22
Dimensiones	18x45 mm

Nota. Tomado de (Arduino, s.f.)

Ventajas y desventajas que presenta el Arduino Nano.

Ventajas

- Tamaño reducido
- Ideal para PCBs
- Gran cantidad de pines E/S

- Facilidad de programación
- Bajo precio
- Disponible en el mercado nacional

Desventajas

- Puerto de comunicación en desuso
- No dispone de pines de alimentación externa
- No posee comunicación USB, solo serial.

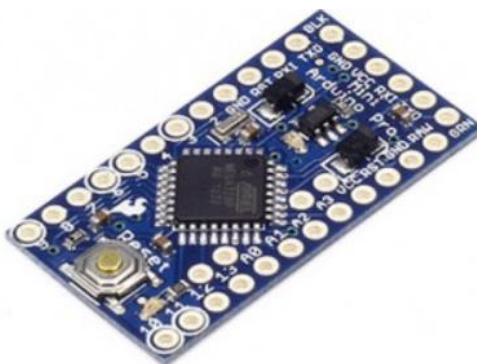
3.3.4.2.2 Arduino Pro Mini.

El Arduino Pro Mini es una pequeña placa que posee el microcontrolador ATmega328, diseñada para usarse en placas de prueba y cuando el espacio es reducido, este tiene 14 pines E/S digital y 8 analógicos, además de un cristal de 16 MHz, ya que la placa no incorpora puerto de conexión se necesita usar adaptadores serie USB o RS232 a TTL (Arduino, Arduino, s.f.).

En la figura 12 se observa la placa Arduino Pro mini.

Figura 32

Arduino Pro Mini



Nota. Tomado de (Arduino, s.f.)

En la tabla 16 se muestran las características de la tarjeta Arduino Pro Mini.

Tabla 16

Características técnicas del Arduino pro Mini

Características técnicas	
Microcontrolador	ATmega328
Voltaje de operación	3.3-5 V
Voltaje de entrada	3.35-12 V
Corriente E/S	40 mA
Velocidad de reloj	8 MHz (versiones de 3,3 V) o 16 MHz (versiones de 5 V)
Pines E/S analógicos	6
Pines E/S digitales	14
Dimensiones	18x33 mm

Nota. Tomado de (Arduino, s.f.)

Ventajas y desventajas que presenta el Arduino Pro Mini.

Ventajas

- Tamaño reducido
- Ideal para PCBs
- Uso de varios tipos de adaptadores de conexión
- Facilidad de programación
- Bajo precio

Desventajas

- No posee comunicación USB, solo serial.

3.3.4.2.3 *Arduino Pro Micro.*

El Arduino Pro Micro es considerado como la versión miniatura de la tarjeta Arduino Leonardo, ya que este incorpora la comunicación USB, lo que permite prescindir de procesadores secundarios, esto quiere decir que al ser conectado al computador este es detectado y reconocido como hardware (TeslaBem, s.f.).

En la figura 33 se observa la placa Arduino Pro micro.

Figura 33

Arduino Pro Micro



Nota. Tomado de (Arduino, s.f.)

En la tabla 17 se muestran las características de la tarjeta Arduino Pro Micro.

Tabla 17

Características técnicas del Arduino Pro Micro

Características técnicas	
Microcontrolador	ATmega32U4
Voltaje de operación	3.3-5 V
Voltaje de entrada	3.35-12 V
Corriente E/S	40 mA
Velocidad de reloj	16 MHz
Pines E/S analógicos	9
Pines E/S digitales	18
Dimensiones	18x33 mm

Nota. Tomado de (Arduino, s.f.)

Ventajas y desventajas que presenta el Arduino Pro Mini.

Ventajas

- Tamaño reducido
- Ideal para PCBs
- Posee comunicación USB
- Facilidad de programación
- Puerto micro USB
- Bajo precio

Desventajas

- Ralentización con el uso múltiples librerías.

3.3.4.2.4 Selección de las tarjetas de control.

En la tabla 14 se detalla la calificación y evaluación de las alternativas de selección para la tarjeta de control.

Tabla 18

Matriz de selección de las tarjetas de control

Criterio de selección	Peso	Arduino Nano		Arduino Pro mini		Arduino Pro Micro	
		Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Comunicación USB	30%	0	0	0	0	5	1.5
Costo	10%	5	0.5	5	0.5	5	0.5
Dimensiones	10%	4	0.4	4	0.4	4	0.4
Numero de pines	30%	4	1.2	4	1.2	5	1.5

		Arduino Nano	Arduino Pro mini	Arduino Pro Micro			
Puerto de programación	10%	4	0.4	3	0.3	5	0.5
Tutoriales	10%	5	0.5	5	0.5	5	0.5
Total	100%		3		2.9		4.9
Resultado			No		No		SI

Según los resultados de la matriz de comparación para la selección de la tarjeta de control es el Arduino Pro micro.

3.3.4.3 Acelerómetro-giroscopio.

Este sensor será necesario para obtener los movimientos del mando, y enviarlos hacia el entorno de RV. Para seleccionar el módulo adecuado se tomará en cuenta las siguientes características:

- Costo
- Disponibilidad en el mercado
- Grados de libertad
- Librerías

3.3.4.3.1 MPU 6050.

El módulo MPU 6050 contiene los elementos necesarios para medir movimiento en seis grados de libertad, esto lo logra combinando un giroscopio de tres ejes y un acelerómetro de tres ejes en un mismo chip, este módulo se comunica a través de interfaz I2C por medio de una librería de código abierto (NayLamMechatronics, s.f.). En la Figura 34 se observa el módulo MPU 6050

Figura 34*MPU 6050*

Nota. Tomado de (NayLamMechatronics, s.f.).

En la tabla 18 se muestran las características del módulo MPU 6050.

Tabla 19*Características técnicas del módulo MPU 6050*

Características técnicas	
Interfaz	I2C
Voltaje de operación	(3.3 – 5) V DC
Grados de libertad	6
Rango acelerómetro	$\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$
Rango giroscopio	± 250 , ± 500 , ± 1000 , ± 2000 grad/seg
Convertor AD	16bits
Dimensiones	10x16x03 mm

Nota. Tomado de (NayLamMechatronics, s.f.)

Ventajas y desventajas que presenta el Arduino Nano.

Ventajas:

- Tamaño reducido
- Suficientes grados de libertad
- Comunicación I2C

- Facilidad de programación
- Bajo precio
- Disponibilidad en el mercado

Desventajas

- Tiempo de calibración

3.3.4.3.2 MPU 9250.

El módulo MPU-9250 es un dispositivo capaz de capturar movimiento de 9 ejes, combina un giroscopio de 3 ejes, un acelerómetro de 3 ejes, un magnetómetro de 3 ejes y un procesador digital de movimiento, su comunicación la realiza a través de interfaz I2C, este módulo también está diseñado para interactuar con múltiples sensores digitales no inerciales (DualTronica, s.f.).

En la figura 35 se observa el módulo MPU 9250

Figura 35

MPU 9250



Nota. Tomado de (DualTronica, s.f.)

En la tabla 20 se muestran las características del módulo MPU 9250.

Tabla 20*Características técnicas del módulo MPU 9250*

Características técnicas	
Interfaz	I2C
Voltaje de operación	(3.3 – 5) V DC
Grados de libertad	9
Rango acelerómetro	±2g, ±4g, ±8g, ±16g
Rango giroscopio	±250, ±500, ±1000, ±2000 grad/seg
Rango magnetómetro	±4800µT
Conversor AD	16bits
Dimensiones	20x16x03 mm

Nota. Tomado de (DualTronica, s.f.)

Ventajas y desventajas que presenta el módulo MPU 9250

Ventajas:

- Tamaño reducido
- Suficientes grados de libertad
- Comunicación I2C
- Facilidad de programación
- Fácil calibración

Desventajas

- Poca disponibilidad en el mercado

Selección del acelerómetro-giroscopio

En la tabla 20 se detalla la calificación y evaluación de las alternativas de selección para la tarjeta de control.

Tabla 21*Matriz de selección para el acelerómetro-giroscopio*

Criterio de selección	Peso	MPU 6050		MPU 9250	
		Calificación	Evaluación	Calificación	Evaluación
Costo	30%	5	1.5	4	1.2
Disponibilidad en el mercado	40%	5	2	2	0.8
Grados de libertad	15%	4	0.6	5	0.75
Librerías	15%	5	0.75	5	0.75
Total	100%		4.85		4.1
Resultado			SI		No

Según los resultados de la matriz de comparación para la selección de la del acelerómetro-giroscopio la mejor elección es el módulo MPU 6050.

3.3.4.4 Accionamiento gatillo del mando.

Este sensor será necesario para obtener la señal de accionamiento del mando, por lo que deberá presentar robustez en su diseño. Debido a las limitadas opciones comerciales, se optó por utilizar el siguiente micro switch.

3.3.4.4.1 Micro Switch KW11-3Z.

Este accionador tiene la finalidad de enviar señales que puedan modificar el comportamiento de un circuito eléctrico, está formado internamente por dos tipos de contactos, normalmente cerrado y normalmente abierto que están conmutados (Electronics, s.f.).

En la Figura 36 se observa el micro switch KW11-3Z.

Figura 36

Micro Switch KW11-3Z



Nota. Tomado de (Electronics, s.f.).

En la tabla 21 se muestran las principales características del micro switch KW11-3Z.

Tabla 22

Características del micro switch KW 11-3z

Características técnicas	
Rango de corriente	3 A
Rango de voltaje	250V
Resistencia	$\leq 50\text{M}\Omega$
Resistencia de aislamiento	$\geq 100\text{M}\Omega$
Vida útil	≥ 10000 ciclos

Nota. Tomado de (DualTronica, s.f.)

Ventajas y desventajas que presenta el micro switch KW11-3Z

Ventajas:

- Dimensiones 20 x 6 x 4.4 mm

- Larga vida útil
- Facilidad de acople

Desventajas

- Velocidad de detección de accionamiento.

3.3.4.5 Accionamiento botón de recarga del mando.

Debido al tamaño reducido que debe ocupar este elemento en el mando, se optó por la única opción adecuada en el mercado, que es el pulsador de dos pines de tamaño 6 x 6 mm, el mismo que se observa en la Figura 37.

Figura 37

Pulsador de dos pines



Nota. Tomado de (DualTronica, s.f.).

3.3.4.6 Motor vibrador.

Este será el encargado de dar la sensación del accionamiento del gatillo del mando por medio de la sensación de vibración. Para seleccionar el accionador adecuado se tomará en cuenta las siguientes características:

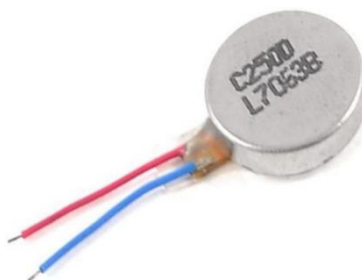
- Dimensiones
- Facilidad de acople

3.3.4.6.1 Motor vibrador 9.000 RPM.

Este actuador es ideal para trabajar directamente con tarjetas microcontroladoras, ya que presenta un tamaño reducido y un bajo consumo de corriente. En la Figura 38 se observa el motor vibrador de nueve mil RPM.

Figura 38

Motor vibrador 9.000 RPM



Nota. Tomado de (NayLamMechatronics, s.f.)

En la tabla 22 se muestran sus principales características.

Tabla 23

Características técnicas del motor vibrador de 9.000 RPM

Características técnicas	
Tamaño	12x3 mm
Alimentación	3 VDC
Velocidad de giro	9000 RPM

Nota. Tomado de (NayLamMechatronics, s.f.).

3.3.4.6.2 Motor vibrador 3.200 RPM.

Este actuador es utilizado generalmente en mandos de consolas y juguetes para niños, presentando un peso y dimensiones adecuadas para su fácil colocación.

Figura 39

Motor vibrador 9.000 RPM



Nota. Tomado de (NayLamMechatronics, s.f.).

En la tabla 23 se muestran sus principales características.

Tabla 24

Características técnicas del motor vibrador de 3.200 RPM

Características técnicas	
Tamaño	15x12x10 mm
Alimentación	5-9 VDC
Velocidad de giro	3200 RPM

Nota. Tomado de (DualTronica, s.f.)

3.3.4.6.3 Selección del motor vibrador.

En la tabla 24 se detalla la calificación y evaluación de las alternativas de selección para el accionamiento del gatillo.

Tabla 25

Calificación, evaluación y selección para el accionamiento del gatillo

Criterio de selección	Peso	Motor vibrador 3200		Motor vibrador 900	
		Calificación	Evaluación	Calificación	Evaluación
Dimensiones	50%	5	2.5	4	2
Facilidad de acople	50%	5	2.5	2	1
Total	100%		5		3
Resultado			SI		No

Según los resultados de la matriz de comparación para la selección del motor vibrador el ideal es el de 3.200 RPM ya que presenta mayor facilidad de acople.

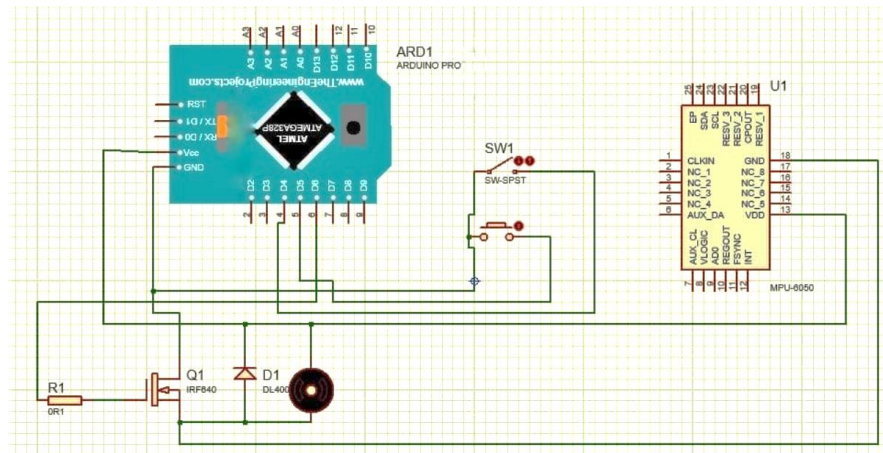
3.4 Diseño del mando

3.4.1 Diseño electrónico

El diseño electrónico del mando está constituido por la tarjeta Arduino pro micro, encargada de la adquisición y procesamiento de las señales, un sensor IMU 6050 que detectará la posición del mando, motor vibrador que dará la sensación de pulsar el gatillo, micro switch y un pulsador utilizado para el gatillo y para recarga de munición, cabe recalcar que el sistema electrónico será alimentado por medio de cable USB conectado al computador ya que de esta forma también se comunica el mando con el entorno virtual, por lo no se requiere de una fuente externa.

Figura 40

Diagrama de conexión



Nota. En la figura se observa el diagrama de conexión del subsistema electrónico.

Para calcular el consumo del subsistema electrónico se asumirá que todos los elementos están activos, como los elementos se encuentran en paralelo, el consumo será la suma total de la potencia de cada uno. La fórmula para calcular potencia es:

$$W = V * I \quad (1)$$

W: potencia eléctrica

V: voltaje

I: corriente

3.4.1.1 Consumo MPU6050.

Este módulo funciona a 3.3 voltios a una corriente de 20 mA.

$$W = 3.3v * 20mA$$

$$W = 66 mW$$

3.4.1.2 Consumo Arduino.

Al ser alimentado mediante cable USB conectado al computador la fuente de voltaje es de 5 voltios llegando hasta consumir 200 mA.

$$W = 5v * 400mA$$

$$W = 1000 mW$$

3.4.1.3 Consumo motor vibrador.

Este motor funciona a cinco voltios a una corriente de 120 mA.

$$W = 5v * 120mA$$

$$W = 600 mW$$

$$W_{total} = (66 + 1000 + 600) mW$$

$$W_{total} = 1666 mW$$

El consumo total del subsistema electrónico es de 1.666 mW.

3.4.2 Diseño mecánico

El prototipo de mando que será diseñado se basará en las armas utilizadas en los entrenamientos reales del personal de las FF.AA. del Ecuador, como el polígono de tiro implementado con RV está enfocado para armas de calibre menor, se utilizará como guía la pistola Pietro Beretta 92 FS, que se observa en la Figura 41.

Figura 41*Pietro Beretta 92 FS*

Nota. Tomado de (Beretta, s.f.).

La Pietro Beretta es un arma legendaria, se la ha catalogado como la mejor pistola de servicio que ha sido utilizada por las fuerzas militares y policiales de varios países del mundo, una de sus características es que tiene gran capacidad de munición, además de un diseño ergonómico.

Tabla 26*Características técnicas de la Beretta 92-FS*

Características técnicas	
Tipo	Semiautomática
Lugar de origen	Italia
Masa	970 g
Largo	217 mm
Longitud del cañón	125 mm
Ancho	38 mm
Altura	137 mm

Nota. Tomado de (Beretta, s.f.).

Según las especificaciones dadas, mediante el uso de software CAD, se procedió al diseño, teniendo en cuenta las modificaciones que este debe tener para poder colocar la electrónica necesaria en su interior, en la figura 42 se observa una vista lateral del prototipo creado.

Figura 42

Mando diseñado, vista lateral derecha



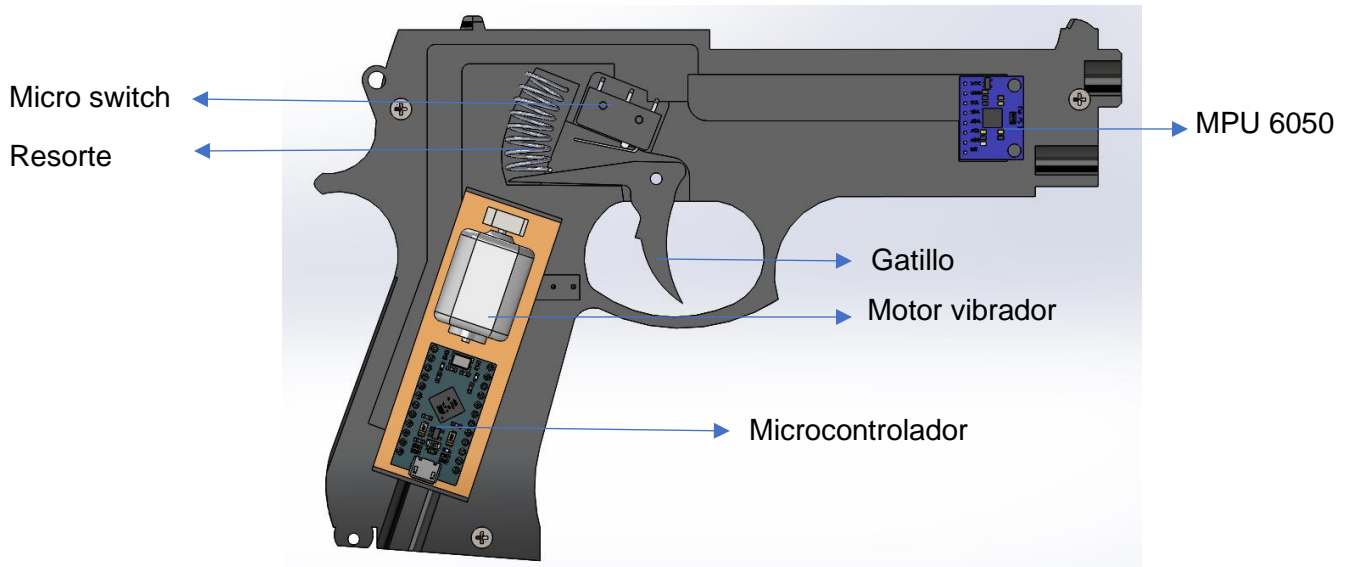
El prototipo está compuesto de diferentes partes, para su facilidad de ensamble y manufactura la estructura principal se la dividió en dos partes, en la Figura 43 se muestra la vista lateral izquierda.

Figura 43

Mando diseñado, vista lateral izquierda



En la parte izquierda interior aloja el gatillo, su respectivo eje de acople y resorte para permitir el desplazamiento del mismo, tiene espacio para el microcontrolador, sensor de movimiento, motor vibrador, micro switch y pulsador, como se observa en la figura 44.

Figura 44*Componentes del mando*

Para realizar el análisis de esfuerzos del sistema se necesita realizar la selección del proceso de manufactura, para según esto obtener las características de los materiales utilizados y realizar los cálculos correspondientes.

3.4.2.1 Técnicas de manufactura.

Las principales técnicas de manufactura requeridas para la elaboración del mando se detallan a continuación, las mismas que deberán cumplir con especificaciones como costo, tiempo de producción, facilidad de manufactura, y resistencia de materiales.

3.4.2.1.1 Modelo por inyección de plástico.

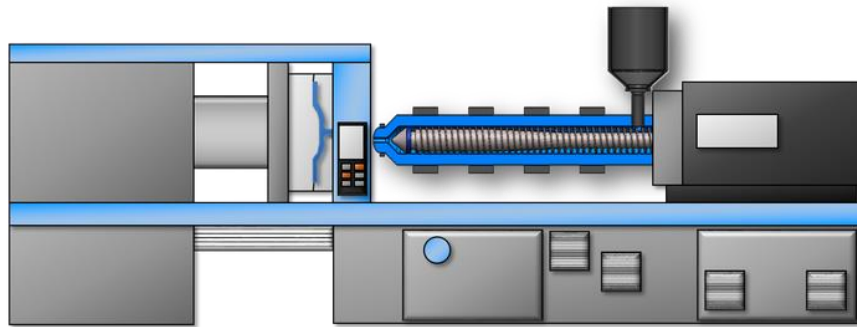
Este proceso permite el modelado de piezas por medio de un molde al cual se le inyecta a presión un polímero que se suministra a través de gránulos, esto se realiza en 4 etapas, que son: cierre del molde, inyección, enfriamiento y apertura

y expulsión de la pieza, este proceso requiere una gran precisión (Plasticsl, s.f.).

Mediante esta técnica se pueden manufacturar piezas complejas de forma rápida y eficiente, es por esto que se la utiliza en la industria, principalmente en la producción en masa, en la figura X se observa una inyectora de plásticos.

Figura 45

Proceso de moldeo por inyección de plástico



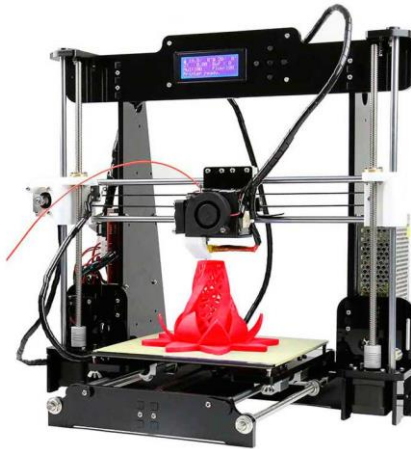
Nota. Tomado de (Plasticsl, s.f.).

3.4.2.1.2 Impresión 3D.

Este proceso consiste en la creación de objetos a través de adición de material por capas que corresponden a las sucesivas secciones transversales de un modelo 3D, los materiales más utilizados para este proceso son los plásticos, mediante esta técnica se pueden crear piezas de geometrías únicas y complejas de una forma rápida, sus aplicaciones son principalmente en el prototipado, educación e incluso en la industria (Autodesk, s.f.). En la figura 46 se observa una impresora 3D comercial.

Figura 46

Impresora 3D



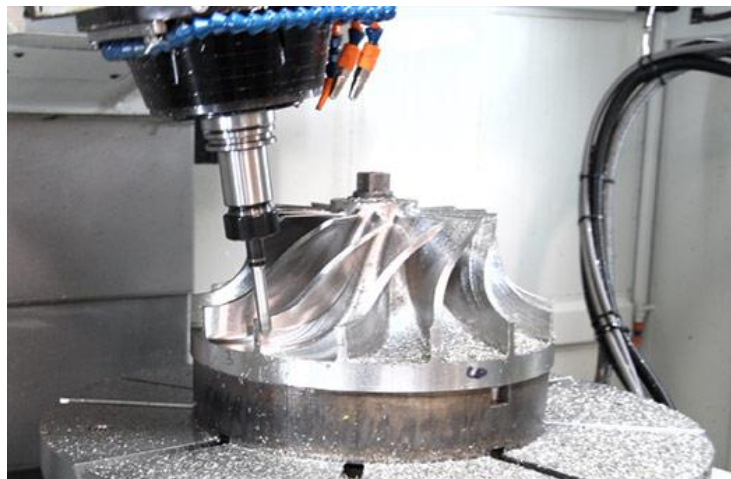
Nota. Tomado de (Global, s.f.)

3.4.2.1.3 Mecanizado.

Dentro de los procesos de fabricación de piezas, el mecanizado es uno de los más utilizados, generalmente se crean piezas metálicas, pero también se pueden utilizar otros materiales como plásticos o materiales compuestos (IngMecánica, s.f.). En la figura 47 se observa un ejemplo de proceso de mecanizado.

Figura 47

Proceso de mecanizado



Nota. Tomado de (Global, s.f.).

3.4.2.2 Selección de la técnica de mecanizado.

Para seleccionar la mejor técnica de manufactura para la construcción del mando se tomará en cuenta las siguientes características:

- Accesibilidad
- Costo
- Robustez

En la tabla 25 se detalla la calificación y evaluación de las alternativas de selección para la técnica de manufactura.

Tabla 27

Matriz de selección para la técnica de manufactura

Criterio de selección	%	Inyección de plástico		Impresión 3D		Mecanizado	
		Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Costo	30%	2	0.6	5	1.5	3	0.9
accesibilidad	35%	3	1.05	5	1.75	3	1.05
Robustez	35%	4	1.4	3	1.05	5	1.75
Total	100%		3.05		4.3		3.7
Resultado			No		Si		No

Según la tabla 25, la mejor opción para realizar el proceso de manufactura es mediante impresión 3D, ya que debido a sus prestaciones es más rápida, menos costosa y de fácil acceso.

3.4.2.3 Selección del material para impresión 3D.

La selección del material para impresión 3D tendrá como principales características facilidad de impresión, accesibilidad y costo.

3.4.2.3.1 ABS.

El filamento ABS por sus siglas acrilonitrilo butadieno estireno, es una termoplástico de uso industrial, ya que entre sus principales características destaca su buena resistencia a las temperaturas bajas y poco peso. Para ser utilizado en impresión 3D se recomienda que el extrusor tenga una temperatura entre 230-260 °C y la cama alrededor de los 130°C, al ser un material que se contrae al contacto con el aire es recomendable que la impresora esté totalmente cerrada para evitar desprendimientos o deformaciones en la impresión (Natives, s.f.).

3.4.2.3.2 PLA.

Sus siglas significan ácido poliláctico, este es un material termoplástico de origen natural, por lo que por medio de composta es biodegradable en un tiempo aproximado de tres meses, característica que lo hace amigable con el medio ambiente. Este material es muy fácil de imprimir ya que la temperatura de extrusión debe estar entre los 185-205 °C y se lo puede imprimir sin cama caliente (Impresoras3D, s.f.).

3.4.2.3.3 PETG.

Este material está compuesto de tereftalato de polietileno (PET) mejorado con glicol (G), aditivo que da un resultado estético más claro, menos frágil y más fácil

de imprimir, una característica importante de este material es que su uso es seguro para tener contacto con alimentos. Este material necesita una temperatura mínima de 260° para poder fundirse y realizar la impresión 3D (All3D, s.f.).

En la tabla 26 se muestra una comparativa de las propiedades de los materiales para impresión 3D.

Tabla 28

Propiedades impresión de materiales

Propiedad	ABS	PLA	PETG
Temperatura fusión	220-240 °C	200-215 °C	230-260°C
Temperatura cama	90-110 °C	0-60 °C	60-90 °C
Ventilador de capa	0%	100%	30-50 %
Facilidad de impresión	Muy baja	Muy alta	Alta
Adhesión entre capas	Baja	Media	Alta
Olor	Toxico	Ninguno	Poco

Nota. Tomado de (Of3lia, s.f.).

Para seleccionar el mejor material para la impresión del mando se tomará en cuenta las siguientes características:

- Accesibilidad
- Costo
- Facilidad de impresión

En la tabla 27 se detalla la calificación y evaluación de las alternativas de selección para el material de impresión.

Tabla 29*Matriz de selección para el material de impresión*

Criterio de selección	ABS		PLA		PETG		
	Peso	Calificación	Evaluación	Calificación	Evaluación	Calificación	Evaluación
Accesibilidad	35%	5	1.4	5	1.4	4	1.4
Costo	25%	4	1	4	1	4	1
Facilidad de impresión	40%	2	0.8	5	2	4	1.6
Total	100%		3.2		4.4		4
Resultado			NO		SI		NO

Según los resultados de la matriz de selección el material escogido para la impresión 3D del mando es el PLA, debido a sus características.

Análisis y construcción del mando.

Ya que se seleccionó como proceso de manufactura la impresión 3D y como material PLA, en la Tabla 28 se detallan las características técnicas del material.

Tabla 30*Propiedades PLA*

Propiedades	Valor
Densidad	1.25 g/cm ³
Módulo de elasticidad	3.5 GPa
Elongación a rotura	6%
Módulo de flexión	4 GPa
Resistencia a la flexión	80 MPa
Temperatura de transición	60°C
Temperatura de flexión al calor	65°C

Propiedades	Valor
Comienzo de fusión	160°C
Módulo de corte	2.4 GPa
Resistencia la tracción	50 MPa

Nota. Tomado de (Hxx, s.f.).

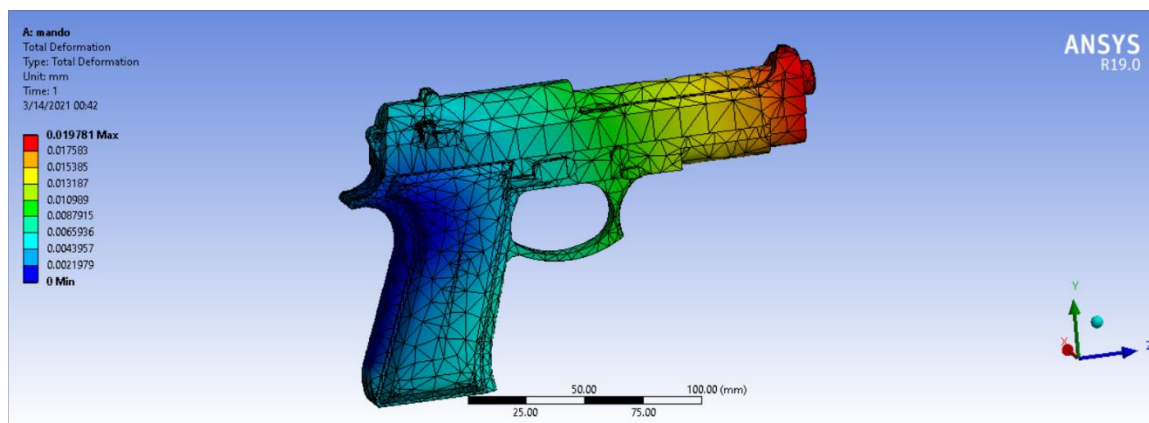
Sabiendo las principales características, se utilizará software de simulación CAE, para determinar el esfuerzo, deformación y factor de seguridad de los diferentes componentes por los que está compuesto el mando.

3.4.2.4 Fuerza de agarre de una persona.

Para el análisis de esfuerzos que se generarán sobre el mando, es necesario saber cuál es la fuerza que una persona puede generar con sus manos, entonces según la Revista Iberoamericana de la cirugía de la mano, esta tiene un valor promedio de 27 Kg (Lázaro, 2008).

Figura 48

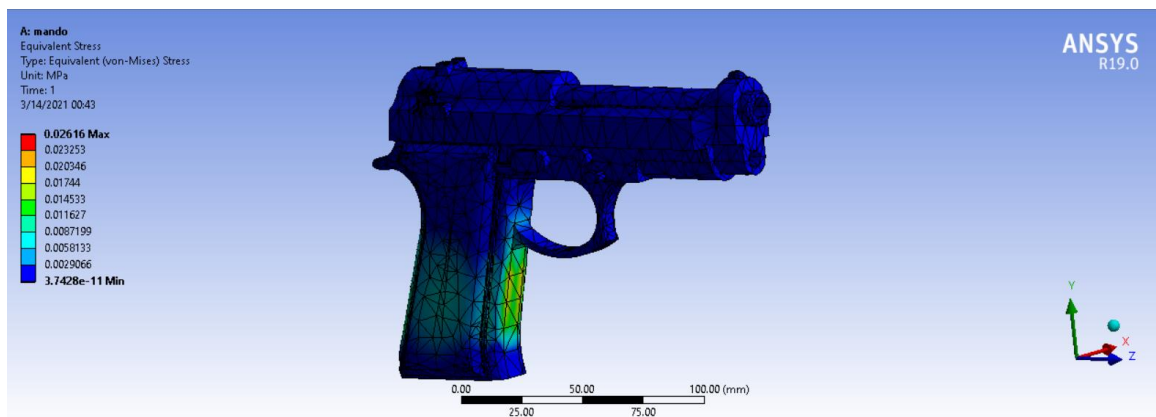
Deformación del mando



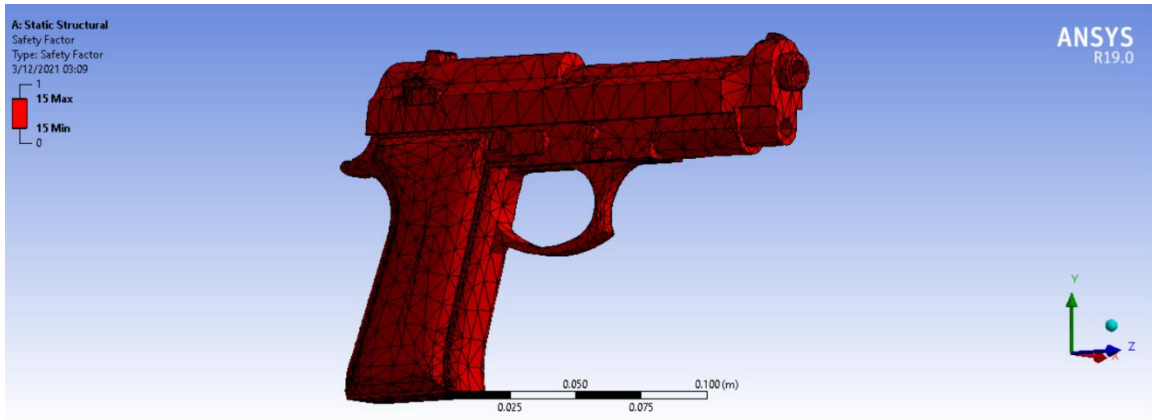
En la figura 48 se observa que al aplicar la fuerza de 27Kg sobre el mango del mando se obtiene un valor máximo de deformación casi despreciable, ya que es igual a 0.019 mm.

Figura 49

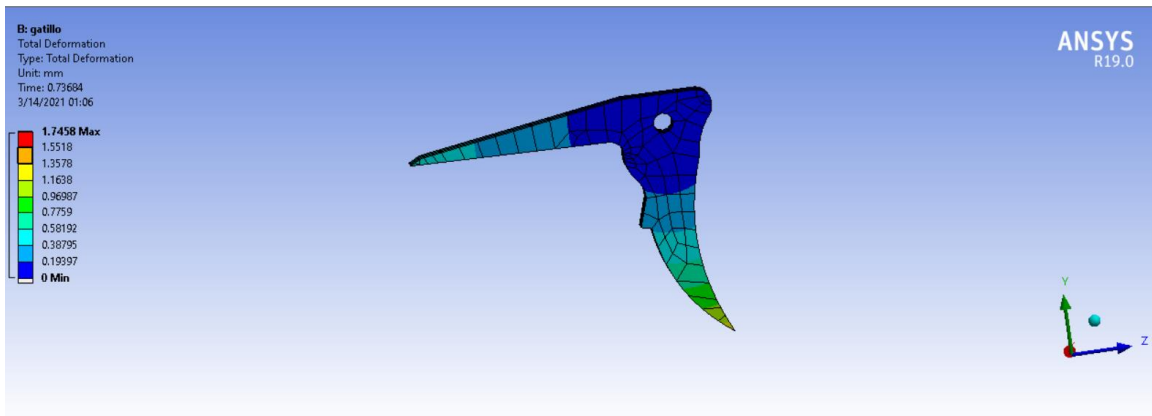
Esfuerzo sobre el mando



En la figura 49 se observa que mediante la simulación realizada el esfuerzo máximo que actúa sobre el mando es de 0.026 MPa, valore relativamente bajo con respecto a la resistencia del material a la compresión, que, según la Tabla de propiedades del PLA es igual a 50 Gpa, por lo que su selección va acorde a las necesidades del diseño.

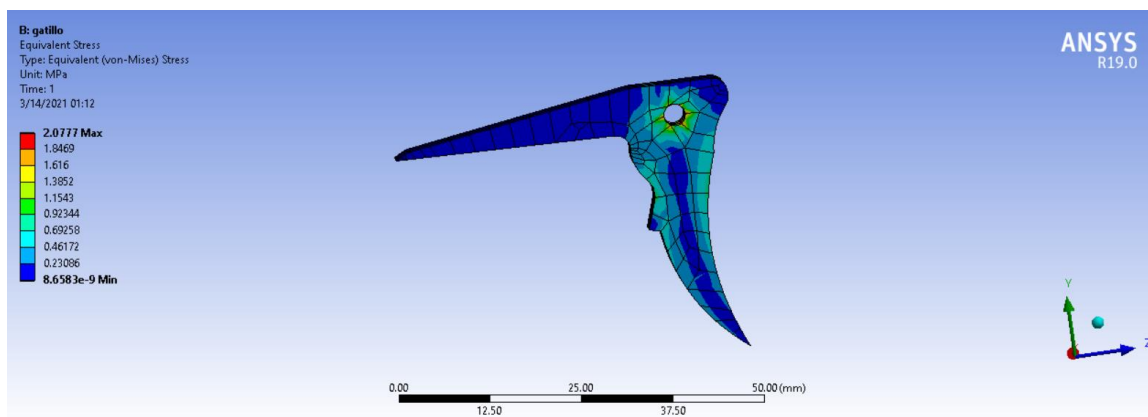
Figura 50*Factor de seguridad del mando*

En la figura 50 se observa que el factor de seguridad del mando es quince, dado que los valores normales de diseño deben ser mayores a dos, se concluye que el diseño cumple con los requerimientos en cuanto a esfuerzos.

Figura 51*Deformación sobre el gatillo*

Como se observa en la figura 51 se obtiene una deformación de 1.74 mm al aplicar la fuerza de cinco Kg. sobre el gatillo.

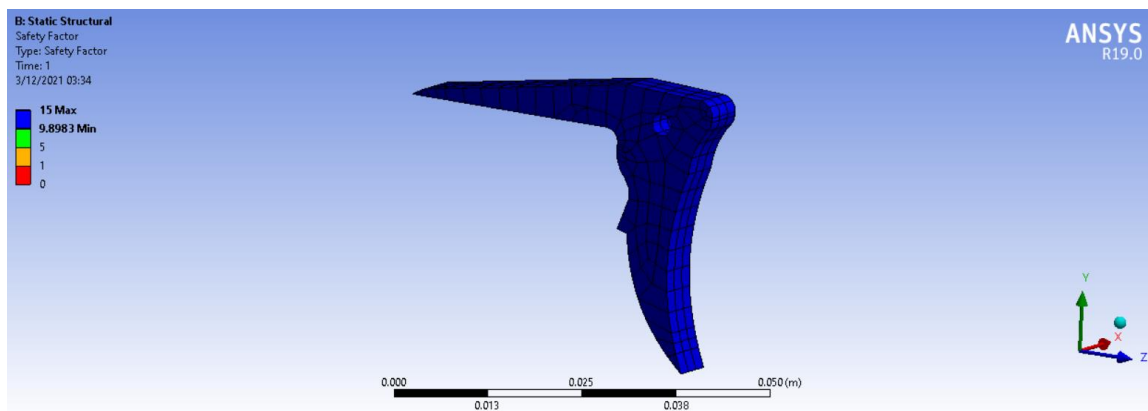
Figura 52*Esfuerzo equivalente Von-mises sobre el gatillo*



En la figura 52 se observa que el esfuerzo máximo que actúa sobre el gatillo es de 2.077 MPa, valor que al comparar con la resistencia del material es mucho menor.

Figura 53

Factor de seguridad del gatillo



Finalmente se obtiene el factor de seguridad del gatillo, el cual como se puede observar en la figura 53 es quince, por lo que el diseño es seguro. El mecanismo de retroceso del gatillo se lo llevó a cabo mediante un resorte, el cual se mandó a fabricar bajo la característica que debe tener 25cm de longitud, se pueda contraer 15 mm y que soporte una fuerza aproximada de 5 Kg. transmitida por el gatillo. Entonces se debe calcular la constante de elasticidad del resorte en función de las condiciones dadas, para su fabricación.

$$F_e = k * x \quad (1)$$

$$\frac{F_e}{x} = k$$

$$\frac{50N}{0.1m} = k$$

$$k = 500N/m$$

Donde:

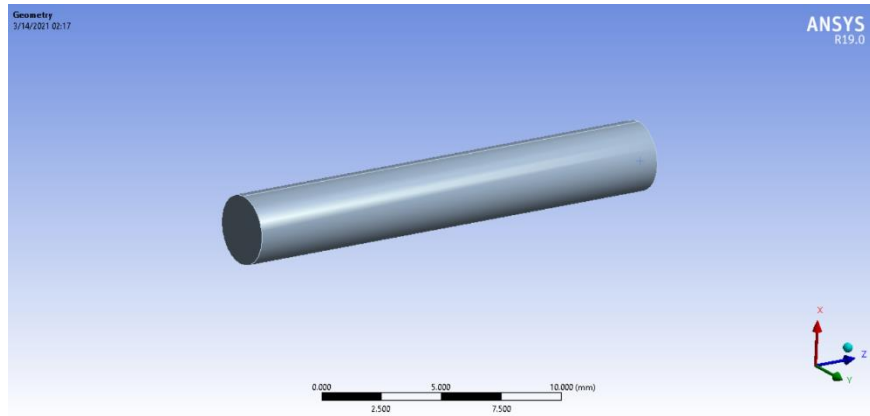
F_e : Fuerza elastica

k: Constante del resorte

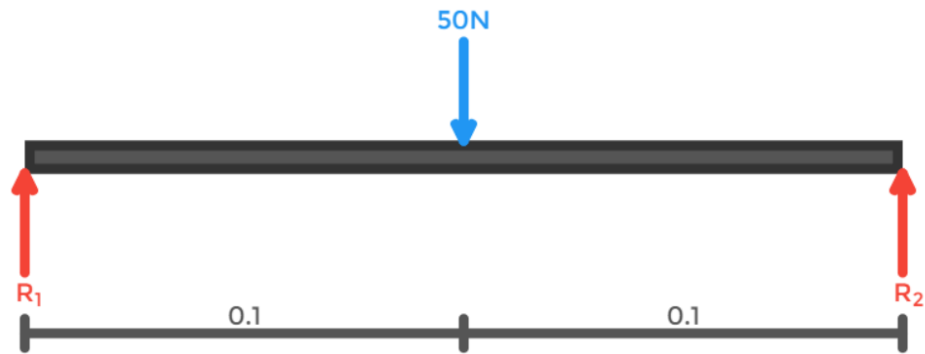
x: elongación del resorte

3.4.2.5 Análisis del pasador.

Este pasador tiene una longitud de 20mm con un diámetro de 3mm, el cual está apoyado en sus extremos y la fuerza del gatillo es transmitida en su parte central, entonces se procede a crear los diagramas de fuerza cortante y momento flector.

Figura 54*Pasador*

El primer paso es hacer su diagrama de cuerpo libre.

Figura 55*Diagrama de cuerpo libre del pasador*

Como el sistema está en equilibrio la sumatoria de las fuerzas que actúan sobre él es igual a cero.

$$\sum Fy = 0$$

$$F_1 - R_1 - R_2 = 0 \quad (1)$$

$$R_1 + R_2 = 50N \quad (2)$$

Para encontrar el valor de las reacciones se hace una sumatoria de momentos en el primer apoyo.

$$\sum Fm = 0$$

$$R_2(0.2m) = 50N * 0.1m$$

$$0.2 * R_2 = 5N \quad (3)$$

Resolviendo el sistema de ecuaciones del sistema se obtiene

$$R_1 + R_2 = 50N$$

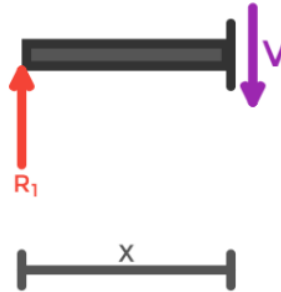
$$0.2 * R_2 = 5N$$

$$R_1 = 25N$$

$$R_2 = 25N$$

Obtenidos los valores de las reacciones, ahora se calcula el diagrama de fuerza cortante, para lo que se secciona el elemento.

Sección 1 ($0 \leq x \leq 0.1$)

Figura 56*Sección 1 del pasador*

$$\sum F_Y = V(x)$$

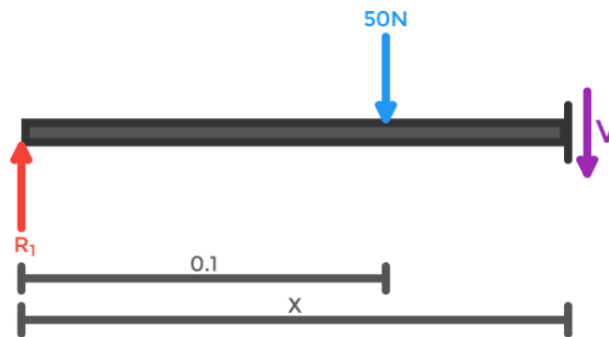
$$-R_1 + V(x) = 0$$

$$R_1 = 25N$$

$$V(x) = 25N$$

(4)

Posteriormente se analiza la sección 2 ($0.1 \leq x \leq 0.2$)

Figura 57*Sección 2 del pasador*

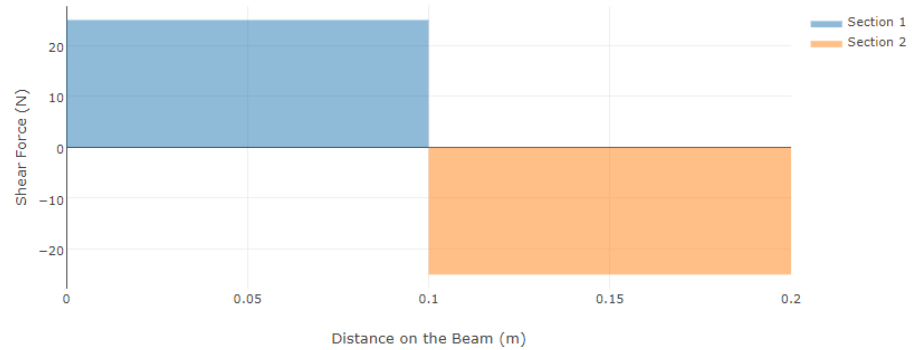
$$F_1 - R_1 + V(x) = 0$$

$$V(x) = -25N$$

Con los resultados obtenidos se dibuja el diagrama de fuerzas cortantes.

Figura 58

Diagrama de fuerzas cortantes

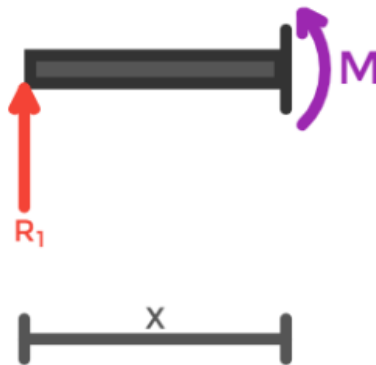


El siguiente paso es el cálculo del momento flector.

Sección 1 ($0 \leq x \leq 0.1$)

Figura 59

Sección 1 momento flector



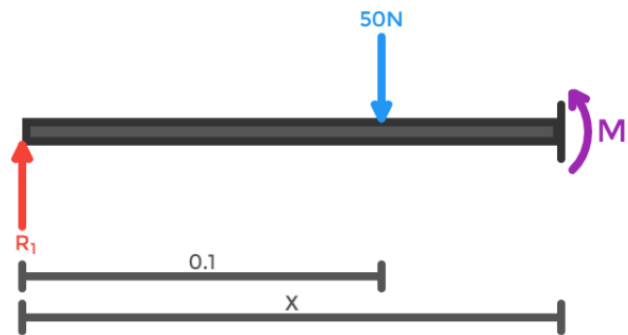
$$-R_1(x) + M(x) = 0$$

$$M(x) = 25x \tag{5}$$

Sección 2 ($0.1 \leq x \leq 0.2$)

Figura 60

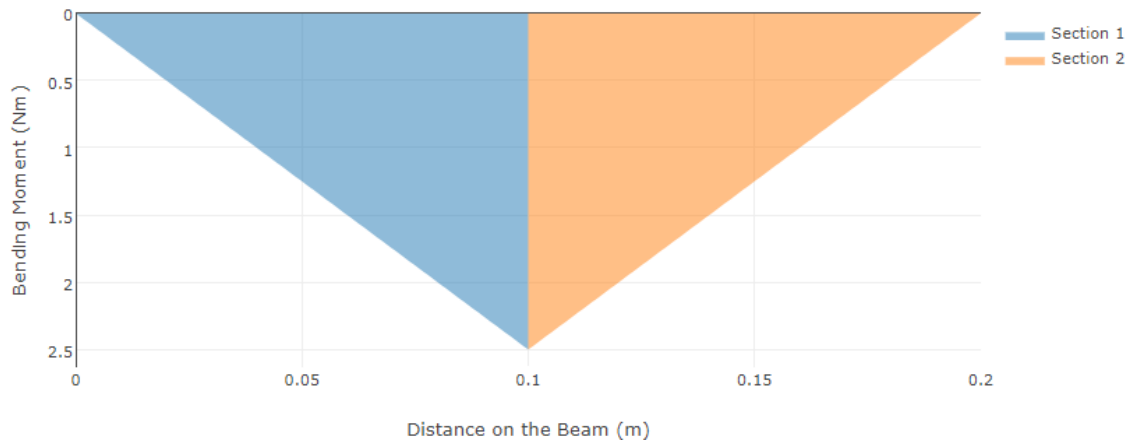
Sección 2 momento flector



$$F_1(x - 0.1) - R_1(x) + M(x) = 0$$

$$M(x) = -25x + 5 \quad (6)$$

Obtenida la ecuación del momento flector, se reemplazan los valores de distancia y se dibuja el diagrama.

Figura 61*Diagrama de momento flector*

El momento flector máximo es 2.5 Nm, que servirá para calcular el esfuerzo normal.

$$\sigma = M * \frac{c}{I} \quad (7)$$

M: momento flector

c: distancia desde el centroide a la fibra más externa

I: momento de inercia

Esta fórmula se puede reemplazar por utilizando el módulo de sección, como se observa en la ecuación X.

$$\sigma = \frac{M}{S} \quad (8)$$

S: módulo de sección

$$S = \frac{\pi * D^3}{32} \quad (9)$$

Donde el diámetro del eje es de tres mm.

$$S = \frac{\pi \cdot (3\text{mm})^3}{32}$$

$$S = 2.65 \text{ mm}^3$$

$$\sigma = \frac{2.5 \cdot 10^{-3} \text{ N/mm}}{2.65 \text{ mm}^3}$$

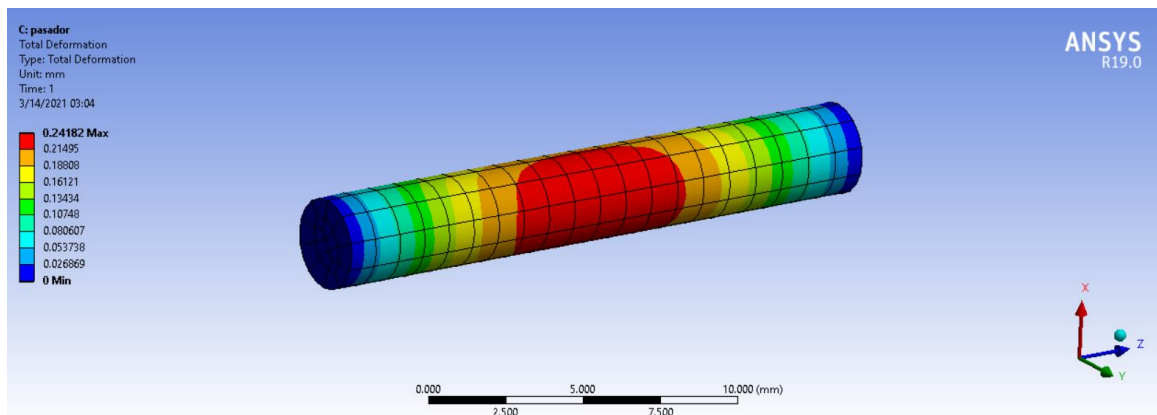
$$\sigma = 0.000943 \text{ N/mm}^2$$

$$\sigma = 943 \text{ Pa}$$

Es esfuerzo calculado será validado mediante software, el resultado será aproximado, ya que el software realiza un análisis por elementos finitos, en donde intervienen varios factores, como es el mallado y el método de resolución.

Figura 62

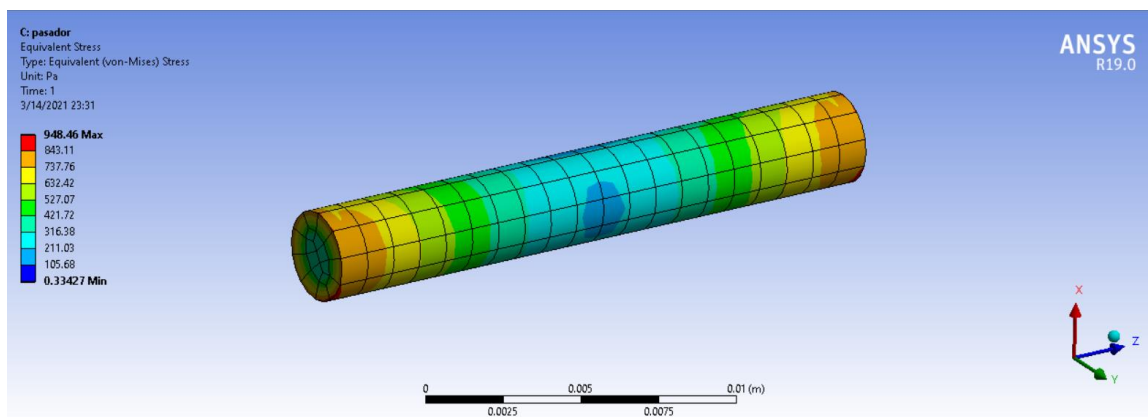
Deformación del pasador



Según la simulación realizada, al aplicar la fuerza de cincuenta N sobre el pasador, este se deforma 0.24 mm valor aceptable para este tipo de aplicación.

Figura 63

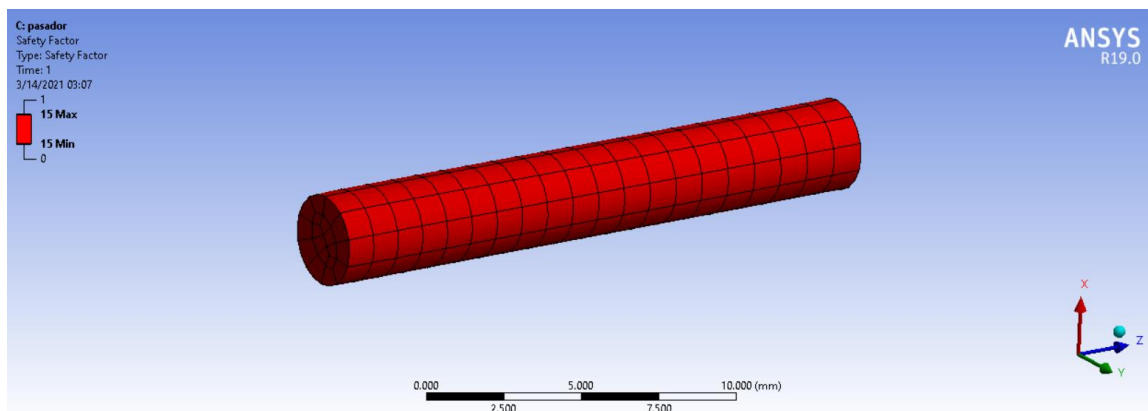
Esfuerzo de Von Mises sobre el pasador



El esfuerzo producido sobre el pasador es de 948 Pa, que al compararlo con la resistencia del material este es mucho menor, por lo que el material no fallará, dicho valor es similar al obtenido analíticamente.

Figura 64

Factor de seguridad del pasador



Como se observa en la figura, el factor de seguridad obtenido es de 15, por lo que el diseño es seguro.

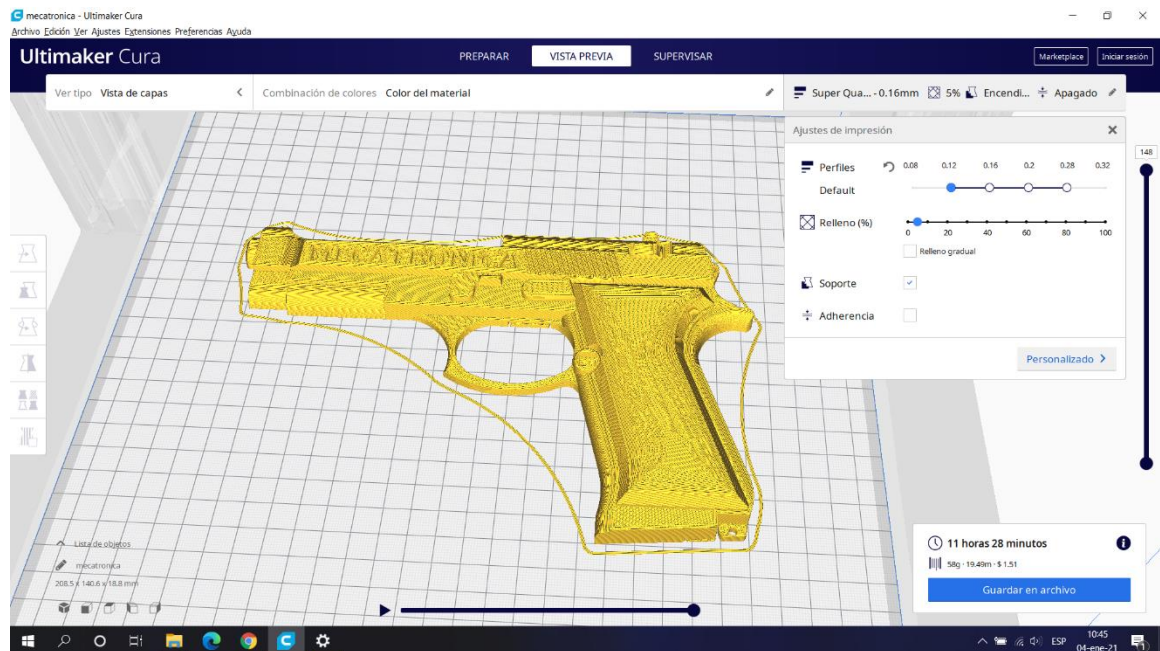
3.4.2.6 Impresión 3D del mando.

Para poder realizar las impresiones del mando, se procede a guardar los archivos del software CAD en formato. stl para que puedan ser procesador por el

software Ultimaker Cura, este programa es el encargado de crear el código G para que la impresora lo pueda interpretar.

Figura 65

Software cura



En la figura 65 se puede observar una de las partes del mando en el software cura una vez que fue laminada, las configuraciones utilizadas para el material PLA fue de 205°C para el extrusor, 60°C cama, velocidad de impresión de 55mm/s obteniendo un tiempo estimado de 11 horas y 28 minutos de impresión.

Figura 66

Impresión del mando



En la figura 66 se observa como una parte izquierda del mando ha sido impresa, usando la misma configuración del programa Cura, se procedió a la impresión de las demás piezas para su posterior ensamble.

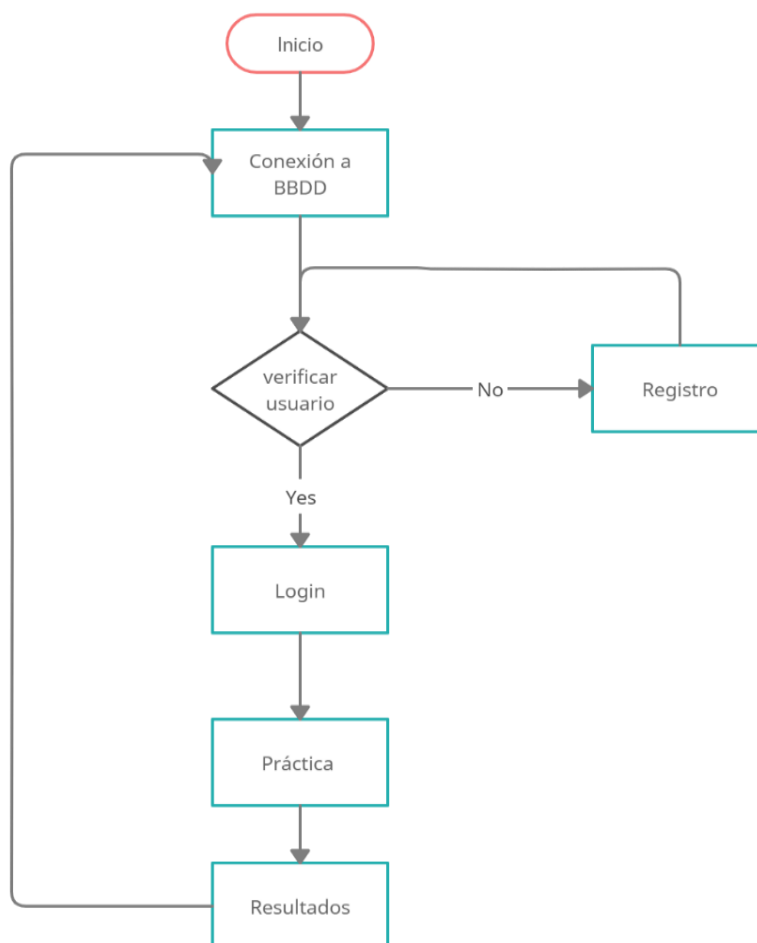
Capítulo IV

Desarrollo de la interfaz de realidad virtual

Este capítulo se detallará el proceso llevado a cabo para crear el entorno de RV para el entrenamiento militar enfocado a los polígonos de tiro, se partirá por el modelado 3D, programación de la interfaz y primeras pruebas realizadas.

Figura 67

Diagrama de flujo del subsistema



4.1 Desarrollo del modelado 3D

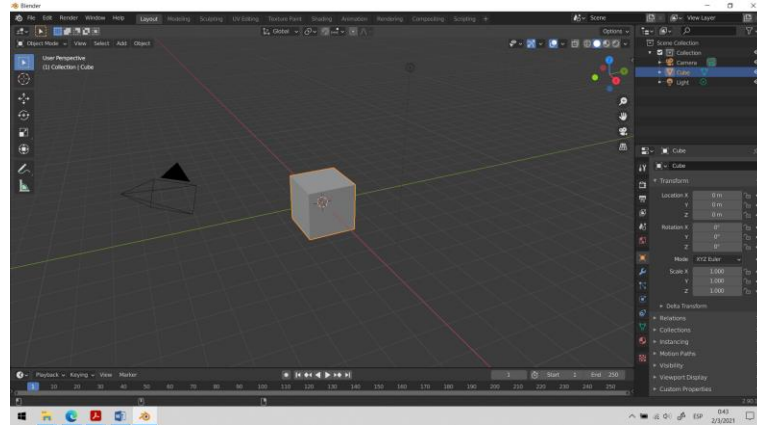
Para el desarrollo del prototipo de entrenamiento de disparo en el polígono de tiro virtual, es necesario el modelado del ambiente o escenario virtual. Uno de los factores importantes a considerar en este modelado es el número de polígonos con el que se genera los objetos contenidos dentro del escenario, ya que esto afectará directamente a la ejecución del programa. Se debe tratar de reducir el número de polígonos al máximo para que la aplicación pueda ser ejecutado en cualquier PC. Es por esto que se ha decidido utilizar blender versión 2.9 (figura 1), ya que es un programa de licencia gratuita el cual ha sido desarrollado específicamente para este propósito.

Figura 68

Logotipo de blender



El modelado del polígono de tiro virtual se lo realizó utilizando el método de modelado por caja, para esto se ha considerado que los objetos que conforman la escena, tienen la característica de ser geométricas y se las puede modelar utilizando un cubo como punto de partida como se muestra en la Figura 69.

Figura 69*BoxModelling*

El escenario que se modelo a través de blender está basado en un campo de tiro controlado (Figura 69), utilizado para entrenamientos de disparo en lugares cerrados. Las dimensiones del campo de tiro son de diez m de ancho y 35 m de largo, sin embargo, los carriles de disparo se encuentran a 25 m del blanco para poder realizar el ejercicio de disparo conocido como “precisión con tiempo”.

Figura 70

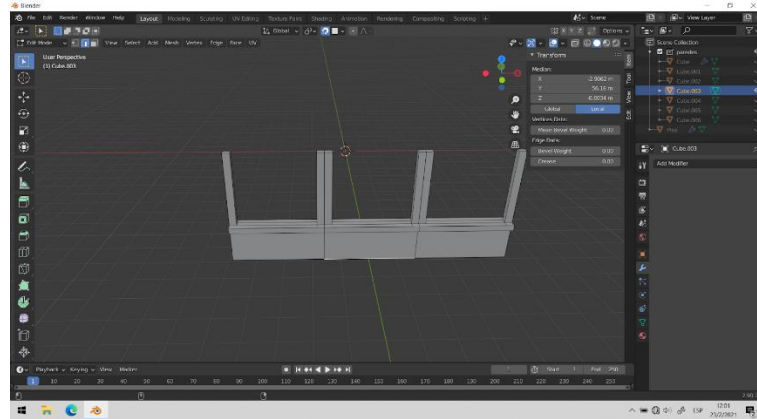
Campo de tiro controlado



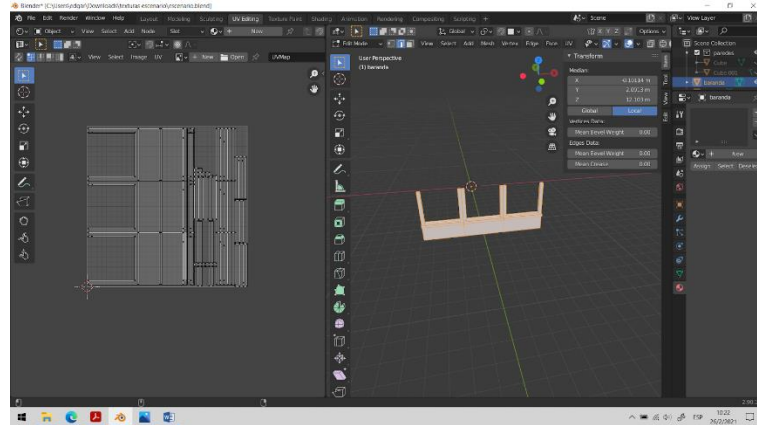
El escenario está compuesto por el piso que es de madera, el carril de disparo (figura 70) también de madera, las paredes laterales de bloque y el fondo es de goma para que las balas no produzcan rebote alguno. Cada uno ha sido modelado por separado para poder realizar un correcto mapeado UV y una posterior texturizado, sin embargo, todo pertenece a una misma colección "Scene collection" para poder exportarlo como un solo objeto y mantener las proporciones dentro del motor de videojuegos.

Figura 71

Modelado del carril de disparo



Para poder aplicar el mapeado UV de los objetos del escenario, es necesario que cada componente se trabaje individualmente, ya que cada objeto ocupa un material distinto, tomando en consideración que los objetos del escenario no son formas complejas se utilizó la herramienta “Smart UV Project” para generar el mapeado obteniendo resultados satisfactorios como se puede observar en el mapa UV del carril de disparo de la figura 71.

Figura 72*Mapa UV de carril de disparo*

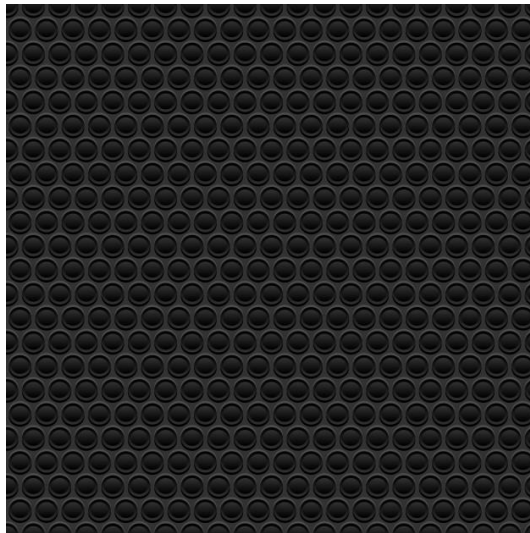
El objetivo de este prototipo virtual es brindar al usuario una sensación natural o realista dentro del mismo. Blender cuenta con las herramientas necesarias para poder generar la pintura y texturizado de los objetos, sin embargo, la curva de aprendizaje de esta herramienta es muy compleja. Por esta razón se utilizó un software adicional de licencia gratuita conocido como “Krita” versión 4.4.2 (Figura 72) conocido por su simplicidad para poder aplicar las texturas y poder generar el mapa de normales que ayudaran con los objetivos del proyecto.

Figura 73*Logotipo de Krita*

Para conseguir el aspecto realista que se buscaba se utilizaron texturas reales de los diferentes materiales, un ejemplo de esto es el material caucho que se utilizó en la pared del fondo del polígono (figura 73). Este material se encuentra en la red y es de uso libre.

Figura 74

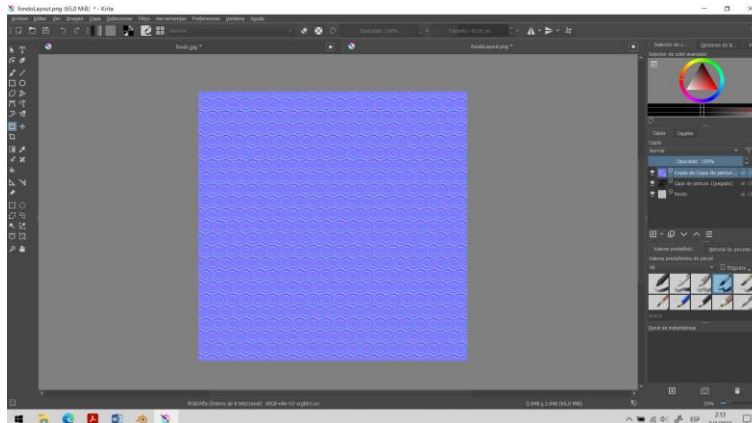
Textura de material caucho



Para obtener el resultado realista que se buscaba, se generó el mapa de normales que brinda la sensación de relieve en las texturas (figura 74).

Figura 75

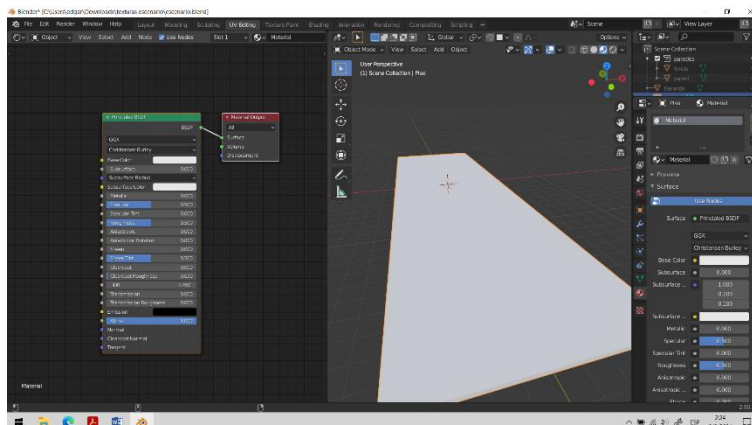
Mapa de normales del fondo de pared



Teniendo listo el mapa de normales y el color base, se procedió a crear el material en blender 2.9 (Figura 75), este material contendrá ambos archivos creados en krita 4.4.2. Cabe señalar que blender utiliza un sistema de generación de texturas que utiliza nodos como se puede apreciar en la parte izquierda de la Figura 11.

Figura 76

Nodos del material piso

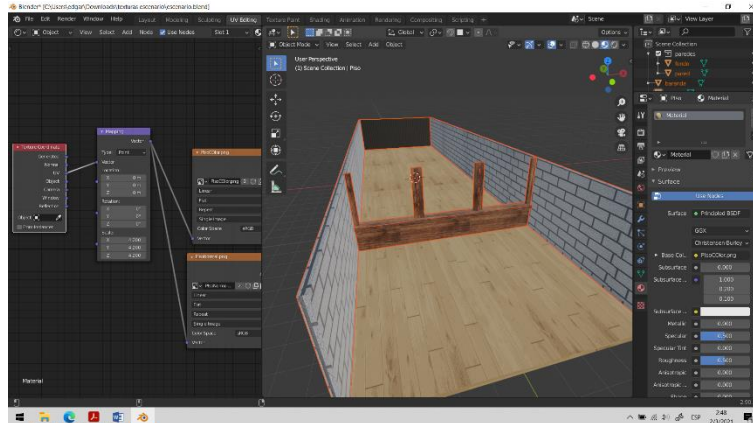


Como ya se mencionó en este apartado del proyecto, uno de los objetivos al crear el entorno es generar el menor número de polígonos posibles en la malla del escenario, en este caso se logró crear el escenario con un total de 224 polígonos, haciéndolo súper ligero para ser ejecutado dentro del motor de videojuegos.

Un resultado similar se hubiera logra haciendo el escenario bloque por bloque o tabla por tabla en el caso del piso o a su vez utilizando herramientas más complejas de Blender 2.9 como es el esculpido digital, sin embargo, esto haría una malla mucho más saturada de polígonos haciendo que el motor de videojuegos exija más recursos para poder ejecutar la aplicación, limitando el número de equipos en los que se puede correr este proyecto.

Figura 77

Escenario optimizado a bajos polígonos



4.2 Desarrollo de la interfaz en Unity

Para esto se utilizará el escenario creado en Blender 2.9 y en el motor de videojuegos Unity 2019.4.15 es necesario exportar en el formato fbx para que puedan exportarse las texturas y configuraciones. Para el desarrollo del entorno virtual se debe

utilizar un motor de videojuegos que tenga soporte para RV específicamente en sobre la plataforma de Oculus, por esta razón se ha decidido usar Unity en su versión 2019.4.15 (figura15) siendo esta una versión estable del software de licencia gratuita.

Figura 78

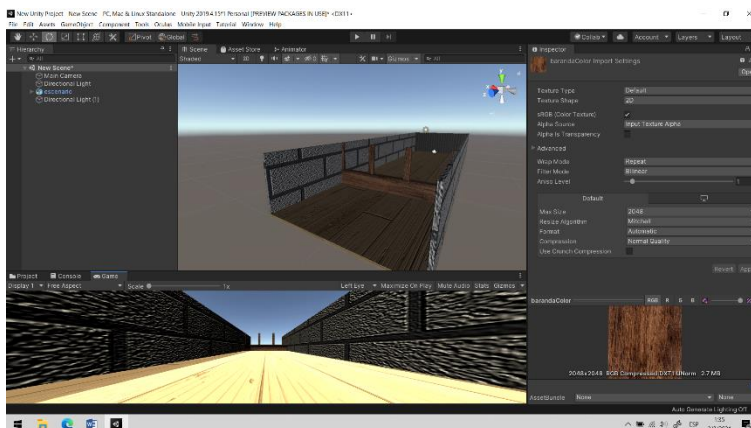
Logotipo de Unity



Una de las ventajas de Unity es la compatibilidad con casi cualquier tipo de extensión 3D. de esta manera fue posible el importar el escenario creado en blender, cargando cada una de sus texturas y mapas de normales, añadiendo la iluminación adecuada como se indica en la figura 16.

Figura 79

Escenario texturizado e iluminado

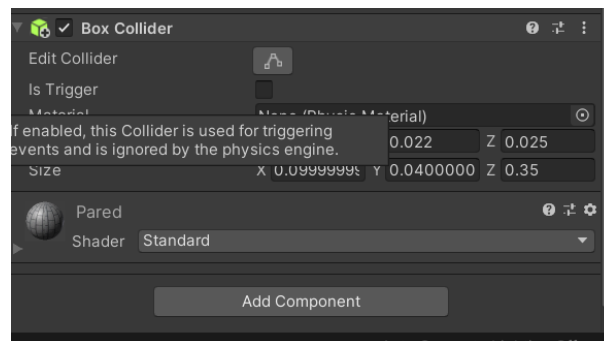


Para que el escenario sea de utilidad y pueda servir como el entorno virtual que se requiere se deben agregar propiedades físicas, ya que, si se agrega un *player* en el escenario, al momento de ejecutar el *player* no se quedaría sobre el escenario, este caería por efecto de la gravedad atravesando el piso y demás objetos.

Para solucionar este inconveniente, a cada uno de los objetos se agrega un *collider*, en este caso como todas las figuras parten de un cubo y mantienen esa forma, se consideró ideal elegir el tipo *box collider*. (Figura 22) que envuelve automáticamente a los objetos dentro del escenario, logrando de esta manera convertirlos en elementos con los que se pueden interactuar.

Figura 80

Box collider



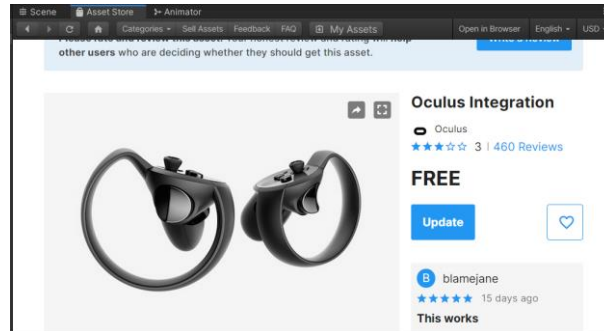
4.2.1 Integración de Oculus a Unity

Teniendo el escenario en dentro del motor de videojuegos y verificando que no existan errores al momento de ejecutar se procede a cargar las librerías y complementos necesarios para el desarrollo del proyecto. Este proyecto ocupa la tecnología de RV de Oculus para su desarrollo, por lo tanto, se debe descargar la librería de *Oculus integration* para poder ocupar las funcionalidades de Oculus, esta librería se las puede descargar del *asset store* de *Unity* de forma gratuita como se

muestra en la Figura 80. Una vez descargado se procede a dar *click* sobre importar para poder agregar las librerías al proyecto que se está desarrollando.

Figura 81

Paquete Oculus Integration



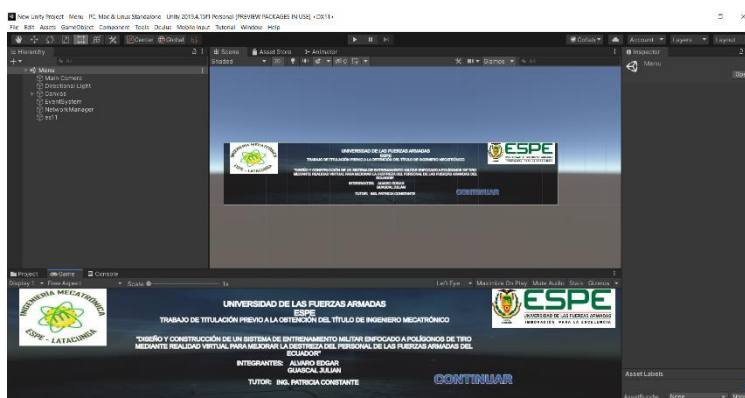
Otra de las librerías importantes para este proyecto es la librería de standard assets, cabe destacar que en esta versión de Unity ya no incluyen de manera predeterminada, ni se los encuentra en la asset store, para ello hay que dirigirse a la página oficial de Unity y descargar el paquete manualmente. Además, de lo anterior el proyecto ocupa el paquete de FPS *weapon package*, este paquete contiene el tipo de arma y animaciones que se utilizarán dentro de este proyecto.

4.3 Desarrollo de las ventanas de inicio

Antes de realizar los algoritmos que determinarán el comportamiento de la plataforma virtual, se procede a agregar las ventanas que aparecerán al iniciar la aplicación, para esto se creó un escenario vacío que contiene el lienzo o Canva que servirán de guía e información del proyecto y su propósito como se muestra en la figura 82.

Figura 82

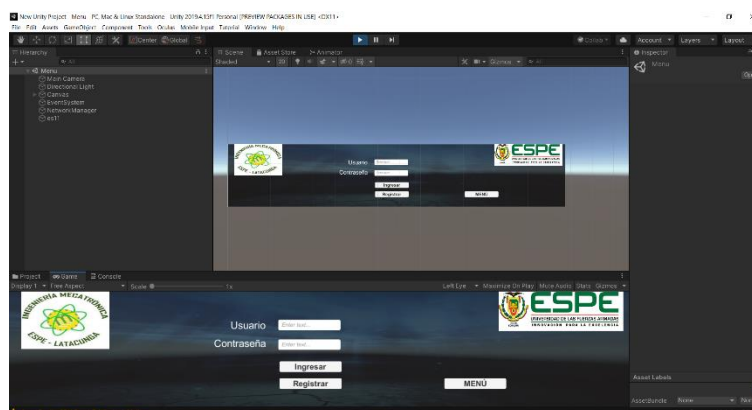
Pantalla de inicio del proyecto



Para que el usuario pueda acceder a su registro es necesario cargar sus datos en la ventana de la (figura 83), en el caso de que no exista el usuario se deberá crear el usuario, cuyos datos serán almacenados en una base de datos local.

Figura 83

Ventana de acceso

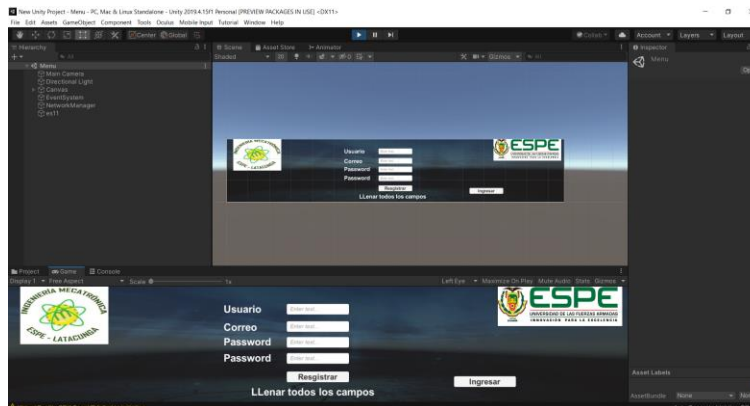


Los datos de cada usuario van a quedar almacenados dentro de la base de datos local, cada usuario deberá llenar los datos necesarios de la (figura 84) para poder

acceder al sistema, este registro se debe realizar una sola vez ya que el sistema no admite el registro del usuario en más de una ocasión.

Figura 84

Ventana de registro



4.3.1 Desarrollo de los algoritmos de control del entorno de realidad virtual

Los algoritmos de control del escenario son escritos en C# como lenguaje de programación, para ello se utiliza el IDE de Visual Studio. Para poder dar funciones a los botones de la pantalla principal en la carpeta menú se crea un script con el mismo nombre que será el encargado de manejar las diferentes ventanas del canva.

En la figura 84 se muestra el algoritmo utilizado para poder navegar entre las diferentes ventanas, la escena del menú y el entorno virtual. Unity 2019.4.15 tiene una librería dedicada para poder desplazarse entre las escenas llamada *Scene Management*. Dentro de la carpeta Menú se crea el script llamado “Menú Principal” El código usado es el encargado de controlar los botones de iniciar y salir de la figura 26, para esto se utiliza dos funciones “Iniciar” y “Cerrar”.

Figura 85

Script menú principal

```
using UnityEngine;
using UnityEngine.SceneManagement;

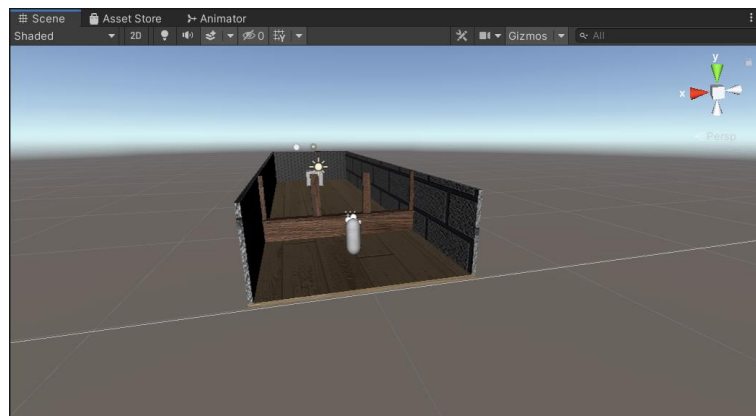
public class MenuPrincipal : MonoBehaviour
{
    public void Iniciar()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void Cerrar()
    {
        Application.Quit();
        Debug.Log("Salir");
    }
}
```

Para continuar con el desarrollo del entorno virtual, se agregó el *Player controller* quien es el encargado de transmitir las acciones asociadas del usuario al entorno virtual. Es de vital importancia usar el *first player controller* que se encuentra en el paquete de Oculus Integration como se muestra en la figura 30.

Figura 86

First person controller de Oculus Integration



Para brindar la sensación de estar presente en el entorno virtual se procedió a agregar brazos al personaje con sus respectivas animaciones, para esto se utilizó el paquete FPS weapon package importado anteriormente. En la carpeta *assets* se encuentra la carpeta *Resource* donde se encuentran los modelos disponibles de armas, para este proyecto se utilizó el modelo "Police 9mm" como indica la figura 87.

Figura 87

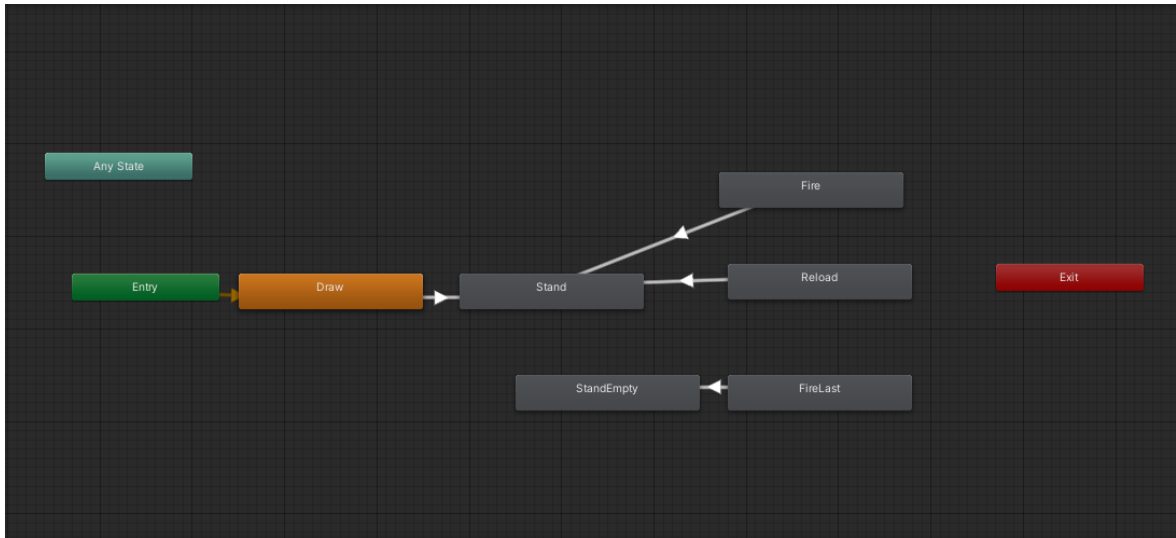
Modelo de brazos y arma



El modelo de arma seleccionada tiene sus propias animaciones, para esto se creó la carpeta animaciones y se creó el archivo de tipo *Animation* y se lo nombro CA Police 9 mm, en este archivo se establece la secuencia de animación que debe seguir el modelo seleccionado como muestra la figura 88.

Figura 88

Animación del modelo



El player prefabricado que se agregó cuenta con funciones como desplazamiento, salto y visualización además de esto tiene agregado las propiedades físicas necesarias para poder interactuar dentro del escenario, sin embargo, estas funciones básicas no son suficientes para el desarrollo de este proyecto.

El concepto del juego se basa en ejercicios cronometrados, por lo que se tomó la decisión de crear un script llamado vida, cuyo código se muestra en la figura 89, este código consiste en un temporizador de cuenta regresiva utilizando la función delta time de Unity para poder contar segundos exactos, este temporizador utiliza una variable publica llamada tiempo para poder ser llamado desde otros scripts.

Figura 89*Script vida*

```
using UnityEngine.UI;
using UnityEngine;

public class vida : MonoBehaviour
{
    // Start is called before the first frame update
    public Text tiempoText;
    public float tiempo = 0.0f;
    // Update is called once per frame
    void Update()
    {
        tiempo = tiempo - 1 * Time.deltaTime;

        if (tiempo < 0)
        {
            tiempo = 0;
        }
        tiempoText.text = "" + tiempo.ToString("f0");
    }
}
```

El código que utiliza el script vida es el llamado “LogicaJugador”, este script contiene los parámetros de ejecución del entorno, es decir, es el encargado de monitorear el status “vida” creado anteriormente, para esto el script crea una variable del mismo nombre vida indicada en la figura 90.

Figura 90*Script lógica jugador*

```
void Start()
{
    vida = GetComponent<vida>();
}

void Update()
{
    RevisarVida();
}

void RevisarVida()
{
    if (vida0) return;
    if(vida.tiempo <= 0)
    {
        vida0 = true;
        gameOverText.gameObject.SetActive(true);
        restButton.gameObject.SetActive(true);
        returnButon.gameObject.SetActive(true);
    }
}

public void ReiniciarJuego()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}

public void Regresar(string sceneName)
{
    SceneManager.LoadScene(sceneName);
}
```

A través de su función “RevisarVida” el script “Lógica jugador” de la figura 90 compara en cada *frame* si la práctica continua, en el caso de agotarse el tiempo este activa la notificación de finalización y habilita los botones reiniciar y atrás mostrados en la figura 90 para poder navegar en el software después de concluida la práctica.

Figura 91

Práctica concluida



El script considerado el más importante o la raíz de este proyecto es el que se encarga de realizar todos los eventos y animaciones del software. En la figura 92 se puede observar cómo al momento de iniciar la aplicación este obtiene las componentes de animación y sonido, además de esto asigna los valores iniciales requeridos en las municiones.

Figura 92

Inicialización del script

```
// Use this for initialization
void Start()
{
    AudioSource = GetComponent<AudioSource>();
    animator = GetComponent<Animator>();

    balasEnCartucho = tamañoDeCartucho;
    balasRestantes = maximoDeBalas;

    Invoke("HabilitarArmar", 0.5F);
}
```

En la figura 93 a través de la función predeterminada *update*, el algoritmo revisa en cada frame, si el usuario está disparando o a su vez si está recargando municiones, estas dos acciones se logran a través de las funciones “Revisar Disparo” y “Revisar Recargar” que fueron creadas para este propósito.

Figura 93

Función update

```
void Update()
{
    if (modoDeDisparo == ModoDeDisparo.FullAuto && Input.GetButton("Fire1"))
    {
        RevisarDisparo();
    }
    else if (modoDeDisparo == ModoDeDisparo.SemiAuto && Input.GetButtonDown("Fire1"))
    {
        RevisarDisparo();
    }

    if (Input.GetButtonDown("Reload"))
    {
        RevisarRecargar();
    }
}
```

La función Revisar Disparo de la figura 94 se encarga de revisar si existen las condiciones necesarias para ejecutarse el disparo, además de esto en la misma figura 38 se muestra la función Disparar. Esta función es de vital importancia y está compuesta por varias funciones, esta función es la encargada de ejecutar los sonidos y animaciones del disparo, así como disminuir el número de balas en el cartucho.

Figura 94

Funciones de disparo

```
void RevisarDisparo()
{
    if (!puedeDisparar) return;
    if (tiempoNoDisparo) return;
    if (recargando) return;
    if (balasEnCartucho > 0)
    {
        Disparar();
    }
    else
    {
        SinBalas();
    }
}

void Disparar()
{
    audioSource.PlayOneShot(SonDisparo);
    tiempoNoDisparo = true;
    fuegoDeArma.Stop();
    fuegoDeArma.Play();
    ReproducirAnimacionDisparo();
    balasEnCartucho--;
    StartCoroutine(ReiniciarTiempoNoDisparo());
    DisparoDirecto();
}
```

Otra de las funciones importantes es la Disparo Directo mostrada en la figura 95, esta función contiene el *raycast*. Si bien es cierto la función se resume en pocas líneas de código, sin embargo, es la encargada de ejecutar el disparo hacia el objetivo utilizando una comparación de tags.

Figura 95

Función disparo directo

```
void DisparoDirecto()
{
    RaycastHit hit;
    if(Physics.Raycast(puntoDeDisparo.position, puntoDeDisparo.forward, out hit))
    {
        if (hit.transform.CompareTag("objetivo"))
        {
            puntuacion++;
        }
    }
}
```

Una de las funciones utilizadas en la función Disparar es la de Reproducción de animaciones de la figura 96, estas cortas líneas de código no son de vital importancia para el desarrollo correcto del algoritmo, sin embargo, es el encargado de brindar una experiencia mucho más confortable a través de la reproducción de animaciones.

Figura 96

Función de animaciones

```
public virtual void ReproducirAnimacionDisparo()
{
    if (gameObject.name == "Police9mm")
    {
        if (balasEnCartucho > 1)
        {
            animator.CrossFadeInFixedTime("Fire", 0.1f);
        }
        else
        {
            animator.CrossFadeInFixedTime("FireLast", 0.1f);
        }
    }
    else
    {
        animator.CrossFadeInFixedTime("Fire", 0.1f);
    }
}
```

Para el caso que el usuario tenga la necesidad de recargar el cartucho, en la figura 97 se han implementado los scripts necesarios para realizar este proceso, por

una parte, la función recargar estará monitoreando si se presiona la tecla para la recarga, mientras que la función recargar munición llevara la cuenta de las balas que tiene el cartucho, así como las balas disponibles para recarga. Para que estas dos funciones se puedan ejecutar desencadenando la acción de recarga antes deben superar las condiciones que solicita la función Revisar Recargar.

Figura 97

Funciones de recarga

```

void RevisarRecargar()
{
    if (balasRestantes > 0 && balasEnCartucho < tamañoDeCartucho)
    {
        Recargar();
    }
}

void Recargar()
{
    if (recargando) return;
    recargando = true;
    animator.CrossFadeInFixedTime("Reload", 0.1f);
}

void RecargarMuniciones()
{
    int balasParaRecargar = tamañoDeCartucho - balasEnCartucho;
    int restarBalas = (balasRestantes >= balasParaRecargar) ? balasParaRecargar : balasRestantes;

    balasRestantes -= restarBalas;
    balasEnCartucho += balasParaRecargar;
}

```

Para poder mostrar en tiempo real el valor de las variables, puntuación. Tiempo y cartuchos, son necesarias las líneas de código de la figura 98, que son las encargadas de almacenar estos valores y relacionarlos con cuadros de texto dentro de la escena.

Figura 98*Script de elementos visuales*

```
using UnityEngine;
using UnityEngine.UI;

public class PantallaBalas : MonoBehaviour
{
    public Text texto;
    public Text texto1;
    public logicaArma logicaArma;

    // Update is called once per frame
    void Update()
    {
        texto.text = logicaArma.balasEnCartucho + "/" + logicaArma.tamañoDeCartucho + "\n" + logicaArma.balasRestantes;
        texto1.text = logicaArma.puntuacion;
    }
}
```

Además de los scripts detallados con anterioridad se necesitan scripts ligados a plataformas externas, para este caso se ocupó una base de datos utilizada para almacenar a los usuarios y sus respectivas puntuaciones.

Capítulo V

Pruebas y resultados

Este capítulo contiene el detalle de las pruebas y funcionamiento del sistema de entrenamiento militar enfocado a polígonos de tiro mediante RV. Las pruebas realizadas se planificaron a desarrollarse dentro de destacamentos militares a varios sujetos de pruebas, sin embargo, debido a la presente situación emergente producida por el Covid-19 han surgido varios inconvenientes para tener acercamiento con este recurso humano, debido a esto se ha visto la necesidad de utilizar sujetos de prueba distintos a los de la planificación.

Los sujetos expuestos a estas pruebas serán los encargados de validar la usabilidad, así como el funcionamiento correcto y eficiencia del sistema de entrenamiento virtual. Pese a los inconvenientes presentados durante la ejecución de este proyecto, los autores tuvieron acercamiento con instructores de las pruebas de disparo, los mismos quienes asistieron al equipo de trabajo en las pruebas y realización del sistema de entrenamiento virtual.

Una prueba fundamental es la de orientación del prototipo de arma diseñado, ya que esta es parte fundamental del sistema, puesto que, mediante la electrónica que por la que está constituida enviará la información correspondiente a la posición en la que se encuentra, el sensor encargado de esto es el módulo MPU 6050, del cual se leerán los datos de sus ejes.

5.1 Orientación del prototipo de arma diseñado

Para la orientación del prototipo de mando diseñado, se utilizó el módulo MPU 6050, que enviará las señales de las mediciones inerciales leídas hacia el

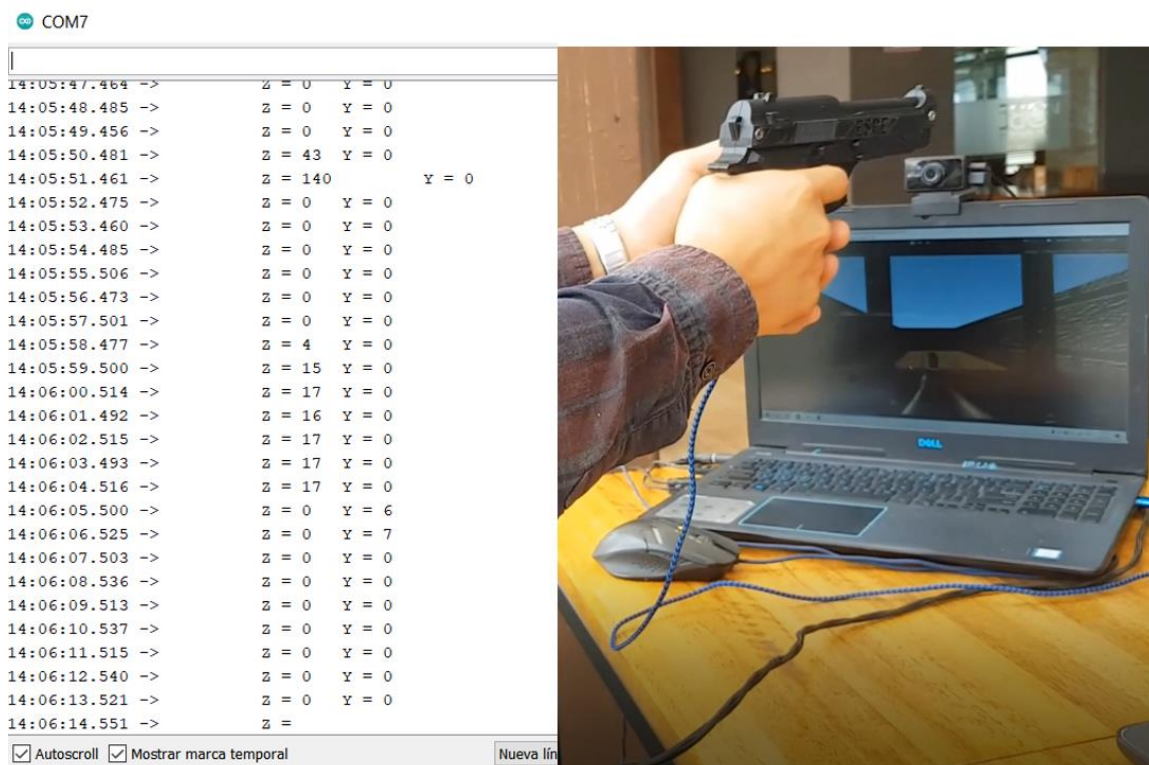
microcontrolador, para posteriormente ser procesadas por el computador y ser transportadas al entorno de realidad virtual, entonces esta prueba consiste en la lectura de los datos correspondientes a los ejes Y y Z, se utilizaron estos dos ejes debido a la ubicación del sensor en el diseño mecánico del mismo.

Debido a que el microcontrolador utilizado correspondiente al Arduino Pro Micro es el Atmega34U4 para la realización de esta prueba se tuvo que iniciar la comunicación serial, cabe recalcar que este microcontrolador posee comunicación USB, razón por la que fue seleccionado, ya que esta permite la conexión directa con Unity.

En la figura 99 se observan los datos correspondientes a la posición de los ejes Y y Z, cuando es prototipo de arma está estática en posición de disparo.

Figura 99

Resultados posición del sensor



Como se observa en la figura 99 los datos de posición se encuentran bordeado el valor de cero, ya que el mando no está en movimiento, estos valores son llevados hacia el entorno de Unity, donde el usuario en dicha posición apunta hacia el centro de la pantalla.

Figura 100

Datos de Movimiento del mando



En la figura 100 se puede observar que al empezar el movimiento del prototipo de arma diseñado los valores empiezan a variar, esto recae en que son llevados al entorno de realidad virtual de forma satisfactoria.

5.2 Velocidad de respuesta de disparo

Esta prueba es realizada de manera netamente experimental, por lo que se realizo un análisis de mediante las grabaciones de las pruebas realizadas por los

autores, en donde se cronometró el tiempo en el que se observa que la persona presiona el gatillo y el tiempo que tarda en producirse el disparo en el entorno de realidad virtual, dichos resultados se observan en la tabla.

Tabla 31

Tiempo de respuesta del gatillo

Numero de prueba	Tiempo (segundos)
1	0.7
2	0.6
3	0.5
4	0.6
5	0.4
6	0.7
7	0.6
8	0.5
9	0.7
10	0.6
Promedio	0.59

Como se observa en la tabla 31 el tiempo de respuesta del gatillo en promedio es de 0.59 segundos, por lo que se concluye que el tiempo de respuesta esta en un rango aceptable para esta aplicación, ya que la velocidad de reacción en promedio de una persona es de un segundo.

5.3 Valoración del sistema de entrenamiento militar virtual

El presente proyecto está enfocado en el área militar ya que busca ayudar al personal en entrenamiento a la adaptación y familiarización con entrenamientos de disparo desarrollo de sus habilidades sin la necesidad de recurrir a equipamiento de alto costo. Para poder tener una idea clara y poder brindar una experiencia satisfactoria a

los usuarios del sistema de entrenamiento militar, se ha consultado con varios profesionales del área militar, así como de varias fuentes bibliográficas de polígonos de tiro deportivo, motivo por el cual se ha decidido replicar un campo de tiro controlado.

5.4 Evaluación de sujetos de prueba

Las pruebas requeridas fueron realizadas por dos voluntarios considerados como idóneos por parte de los autores para desarrollar el testeo. Los indicadores considerados se presentan a continuación.

5.4.1 Criterios de idoneidad

- Hombres y mujeres sin problemas de visión.
- Hombres y mujeres que no padezcan de migrañas ni mareos.
- Hombres y mujeres dispuestos a colaborar de forma honesta.

5.4.2 Valoración de los participantes

Los participantes seleccionados cumplen con los criterios de idoneidad establecidos por los autores haciéndolos aptos para el testeo voluntario del sistema de entrenamiento militar. Los datos requeridos de los participantes se detallan en la tabla uno. El participante uno es miembro de las FF.AA. en servicio activo y ha sido capacitado en varios cursos de entrenamientos de disparo. Los autores han elegido al participante considerando su experiencia y poder tener un mejor criterio de uso del sistema de entrenamiento virtual.

Tabla 32*Información de participantes*

Nombre	Edad	Participante #
Identidad reservada	36	1
Jhonny Vaca	26	2

Nota. Tomado de encuesta realizada in situ, por Guascal, J. & Alvaro, T. 2021.

Los participantes realizaron las prácticas en 8 ocasiones, las pruebas fueron aplicadas dos veces por día con diferentes tiempos, cada participante tuvo un día de descanso entre cada práctica.

Ejercicio realizado – Tiro de precisión con tiempo

Este ejercicio tiene como fin desarrollar la puntería, tiene un total de 10 cartuchos los cuales deberán ser completados en un tiempo determinado, el objetivo de la práctica es determinar el funcionamiento correcto del sistema. En la tabla 33 se muestra la evolución con respecto al sistema del participante 1 en el tiempo 1.

Tabla 33*Evolución del participante 1, tiempo 1*

Días	Repetición	Tiempo	Aciertos	Promedio de aciertos
1	1	10	4	5
	2	10	6	
2	1	10	7	6
	2	10	5	
3	1	10	6	6
	2	10	6	
4	1	10	7	6.5
	2	10	6	

Se ha establecido un tiempo de veinte segundos para cada ejercicio, por lo tanto, el participante debió ejecutar un disparo cada 2 segundo para poder completar la práctica. En la tabla 3 se muestra los resultados del participante 1 al contar con un tiempo de quince segundos, es decir, 25 segundos por disparo.

Tabla 34

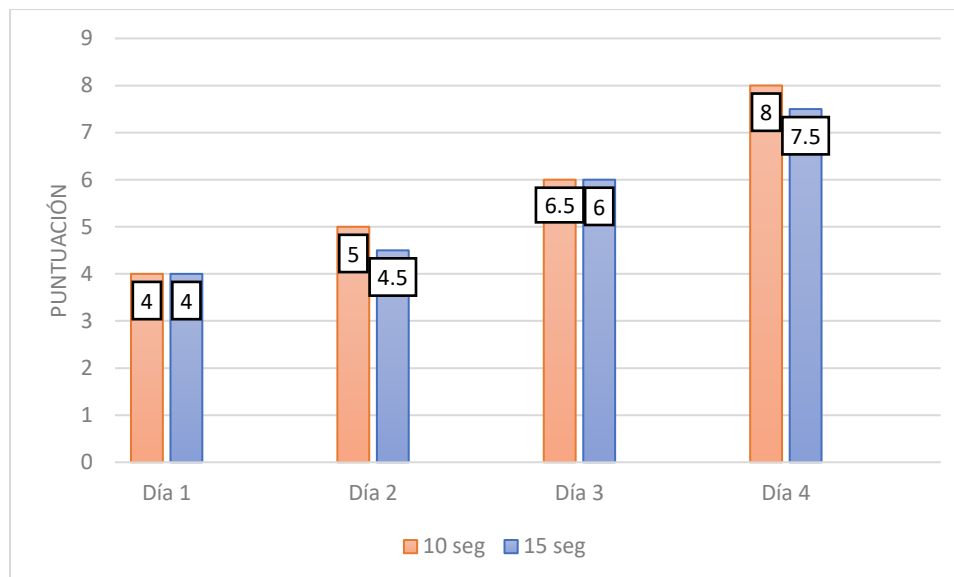
Evolución del participante 1, tiempo 2.

Días	Repetición	Tiempo	Aciertos	Promedio de aciertos
1	1	15	8	7
	2	15	6	
2	1	15	7	7.5
	2	15	8	
3	1	15	6	7
	2	15	8	
4	1	15	9	7.5
	2	15	6	

En la tabla 33 se puede apreciar la evolución del participante realizando el mismo ejercicio con diferente tiempo, en el segundo caso se puede determinar que el sistema al contar con mayor tiempo de ejecución presenta una precisión más alta.

Figura 101

Resultados de la práctica del progreso del participante 1



Nota. Tomado de encuesta realizada in situ, por Guascal, J. & Alvaro, T. 2021.

En la figura 101 se muestra la notable mejoría en cuanto a aciertos del participante, en el primer caso a 20 segundos el puntaje máximo alcanzado es de 6.5, en el segundo caso a 25 segundos el promedio de aciertos es de 7.5 siendo esto un indicador para afirmar que el tiempo es proporcional a los resultados. Considerando que los ensayos a mayor tiempo se hicieron en todas las ocasiones después de los ejercicios a veinte segundos, se puede afirmar que el usuario se adapta al uso del sistema y obtiene mejores resultados.

En la tabla 35 se muestra la evolución con respecto al sistema del participante dos en el tiempo uno. Ambos participantes fueron sometidos a la misma práctica con las mismas condiciones.

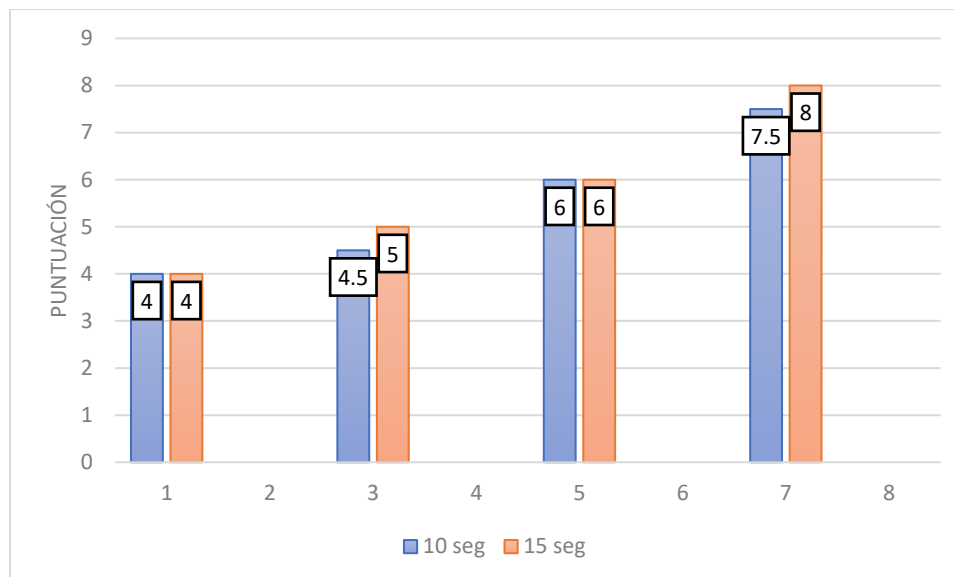
Tabla 35*Progreso del participante 2, tiempo 2*

Días	Repetición	Tiempo	Aciertos	Promedio de aciertos
1	1	10	3	4
	2	10	5	
2	1	10	4	4.5
	2	10	5	
3	1	10	5	6
	2	10	7	
4	1	10	7	7.5
	2	10	8	

En la tabla 36 se muestra los resultados del participante dos al contar con un tiempo de quince segundos, es decir, 1.5 segundos por disparo.

Tabla 36*Evolución del participante 2, tiempo 2*

Días	Repetición	Tiempo	Aciertos	Promedio de aciertos
1	1	15	4	4
	2	15	4	
2	1	15	6	5
	2	15	4	
3	1	15	6	6
	2	15	6	
4	1	15	9	8
	2	15	7	

Figura 102*Progreso del participante 2*

En la figura 102 se puede evidenciar una adaptación al sistema con una variación mínima entre cada participante,

5.4.3 Validación de la usabilidad del sistema de entrenamiento

Uno de los objetivos primordiales de este proyecto es brindar un sistema intuitivo, fácil de usar por cualquier usuario que requiera el apoyo de la herramienta, además de esto el sistema no debe ser perjudicial para los usuarios, proporcionando todas las garantías necesarias.

La validación de este proyecto se realiza a través del test conocido como SUS (Escala de usabilidad del sistema) para poder determinar la usabilidad del mismo debido a su sencillez y resultados efectivos. El test SUS está conformado por 10 interrogantes las cuales tienen un rango de uno a cinco, donde uno es total desacuerdo y cinco, total acuerdo.

Los resultados de este test se obtienen a partir de la suma de los resultados promedios de los cuestionarios realizados a los participantes. Las preguntas impares toman el valor indicado por el participante y se le restará un punto, las preguntas pares cinco y se restará el valor asignado por los participantes. El resultado final se obtiene del producto de la sumatoria por 2.5, el resultado final estará ponderado entre cero y cien por ciento, siendo el cien siendo el último el máximo teórico.

Tabla 37

Test SUS en participantes

	Participante 1	Participante 2
Pregunta 1	4	4
Ponderación 1	3	3
Pregunta 2	2	1
Ponderación 2	3	4
Pregunta 3	5	4
Ponderación 3	4	3
Pregunta 4	2	2
Ponderación 4	3	3
Pregunta 5	3	3
Ponderación 5	2	2
Pregunta 6	2	1
Ponderación 6	3	4
Pregunta 7	5	5
Ponderación 7	4	4
Pregunta 8	1	1
Ponderación 8	4	4
Pregunta 9	3	3
Ponderación 9	2	2
Pregunta 10	1	2
Ponderación 10	4	3
Sumatoria individual	32	32
Promedio	32	
Puntaje SUS	80	

Para poder tener una referencia más clara del puntaje obtenido, se propuso un conjunto de equivalencias mostradas en la tabla 38.

Tabla 38*Ponderación SUS*

Puntaje	Equivalencia
00 – 20	Muy deficiente
21 – 40	Deficiente
41 – 60	Buena
61 – 80	Excelente
81 – 100	Muy excelente

El puntaje sus de la tabla 36 fue de 80, utilizando la equivalencia propuesta de la tabla 36, la usabilidad del sistema es considerada como excelente, considerándolo como eficiente y es capaz de cumplir sus objetivos.

5.5 Validación de la hipótesis

5.5.1 Planteamiento de la hipótesis

Debido a la situación emergente que atraviesa el país expuesto en este proyecto, los autores se ven limitados a realizar las pruebas necesarias para poder comprobar o descartar la hipótesis inicial planteada ya que no se cuenta con el recurso humano necesario. Por lo tanto, los autores únicamente se ven forzados a la comprobación del funcionamiento y usabilidad del sistema con los datos recolectados en las practicas realizadas por los voluntarios.

¿El sistema de entrenamiento del personal militar mejora las destrezas de manipulación y utilización de armamento de fuego, sin la necesidad de exponer al elemento humano y la reducción de costos de capacitación?

La hipótesis será comprobada en la medida de lo posible con los datos obtenidos a través de las pruebas a las que fueron sometidos cada uno de los voluntarios.

5.5.2 Variables de investigación

Variable independiente: Simulador de tiro implementado mediante RV

Variable dependiente: Entrenamiento del personal de las FF.AA.

5.5.3 Validación de la hipótesis

La hipótesis del sistema de entrenamiento virtual fue comprobada por el método del Chi-cuadrado para ello se estableció la hipótesis alternativa y la hipótesis nula.

- **Hipótesis alternativa (H_i):** El sistema de entrenamiento del personal militar mejora las destrezas de manipulación y utilización de armamento de fuego, sin la necesidad de exponer al elemento humano y la reducción de costos de capacitación.
- **Hipótesis nula (H_o):** El sistema de entrenamiento del personal militar no mejora las destrezas de manipulación y utilización de armamento de fuego, sin la necesidad de exponer al elemento humano y la reducción de costos de capacitación.

La ecuación descrita en el libro de Mario Triola permite el cálculo del Chi-cuadrado.

$$\lambda^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

Donde:

λ^2 : *chi cuadrado*

O_i : frecuencia absoluta observada

E_i : frecuencia esperada

Nivel de confianza = 90%.

Los datos experimentales necesarios para calcular el Chi-cuadrado se presentan en la tabla 39. Estos datos fueron obtenidos durante las pruebas realizadas a los participantes. Se tomó en consideración la cantidad de impactos recibidos por el objetivo.

Tabla 39

Datos de las pruebas de funcionamiento realizadas

Participante	Funciona	No funciona	Referencia
Participante 1	4	6	10
	6	4	10
	8	2	10
	7	3	10
Participante 2	6	4	10
	8	2	10
	10	0	10
	9	1	10
Total	58	22	80

Utilizando la ecuación se determina la frecuencia esperada, utilizando los valores de frecuencia observada. Donde:

$O_i =$ frecuencia observada

E_{ij} = frecuencia esperada

O = total

Se obtiene:

$$E_{ij} = \frac{10 * 58}{80} = 7.25$$

$$E_{ij} = \frac{10 * 22}{80} = 2.75$$

Tabla 40

Frecuencias esperadas

Participantes	Funciona	No funciona	Referencia
Participante 1	7.25	2.75	10
	7.25	2.75	10
	7.25	2.75	10
	7.25	2.75	10
Participante 2	7.25	2.75	10
	7.25	2.75	10
	7.25	2.75	10
	7.25	2.75	10

Con los valores de la tabla 40 y la tabla 41, se calcula el Chi-cuadrado con la ecuación.

$$\lambda^2 = \frac{O_{ij} - E_{ij}}{O}$$

Donde:

i = número de filas.

$j = \text{número de columnas}$

Tabla 41

Datos calculados de chi-cuadrado

Participantes	Funciona	No funciona
Participante 1	1.4568	3.8409
	0.2155	0.5681
	0.0775	0.2045
	0.0086	0.2227
Participante 2	0.2155	0.5681
	0.0775	0.2045
	1.0431	2.75
	0.4224	1.1136
Total	3.5169	9.4724
Sumatoria total	12.9893	

El grado de libertad se calcula a través de la ecuación.

$$n = (i - 1) * (j - 1)$$

$$n = (8 - 1) * (2 - 1) = 7$$

Para obtener los valores para la distribución de Chi-cuadrado se compara con la tabla del Anexo B. Considerando un 90% de confiabilidad, se obtiene:

$$\lambda^2 \text{ de tabla} = 12.017$$

Se dice que la hipótesis alternativa es verdadera si se cumple que, el Chi-cuadrado calculado es mayor que el Chi-cuadrado de tabla.

$$\lambda^2 \text{ de tabla} \geq \lambda^2 \text{ calculado}$$

12.9893 \geq 12.017

La condición de aceptación de hipótesis fue cumplida, por lo tanto, El sistema de entrenamiento del personal militar mejora las destrezas de manipulación y utilización de armamento de fuego, sin la necesidad de exponer al elemento humano y la reducción de costos de capacitación, cabe recalcar que esta hipótesis fue validada con participantes voluntarios no incluidos originalmente en la planificación de pruebas.

Capítulo VI

Conclusiones y recomendaciones

6.1 Conclusiones

- El sistema de entrenamiento virtual permite simular el ejercicio de disparo conocido como *tiro precisión* mediante percepción visual con las Oculus y táctil a través del mando. la prueba de usabilidad SUS califico este proceso con 80 puntos, es decir, el sistema tiene un desempeño excelente bajo esta ponderación.
- El polígono de tiro virtual desarrollado es una réplica de un campo de tiro controlado, tiene un área de 10x25m subdividido en tres carriles de disparo, de los cuales se utiliza uno, obteniéndose un entorno adecuado para que el usuario tenga una mejor experiencia al usar el sistema.
- El cuanto, al desarrollo del mando, este se basó en armamento utilizado por el personal militar, este mando se diseñó mediante software CAD, se validó su resistencia mecánica mediante software CAE, en su interior se realizaron modificaciones para acoplar la electrónica necesaria para trasladar los movimientos que el usuario realice al entorno virtual, por lo que se utilizó un sensor MPU 6050 que es el encargado de sensar los movimientos del mando, esta información se procesó mediante una tarjeta de desarrollo Arduino Pro Micro, también se incluyeron dos accionamientos más, uno para la función de gatillo y otro para la función recargar el arma en el entorno de RV, además de contar con un motor vibrador para dar la sensación de disparo.

- El mando de disparo es una réplica de arma de bajo calibre llamada Pietro Beretta FS 92 utilizada en campos de tiro militares, así como deportivos, el usuario pudo familiarizarse con el uso y manipulación de la misma.
- El test SUS en su numeral 3 indica el nivel de dificultad del uso del sistema, la puntuación alcanzada es de 4.5/5 lo cual indica que el sistema es intuitivo y puede ser usado por cualquier usuario.
- Se desarrolló han desarrollado algoritmos capaces de recibir y procesar la información proveniente del prototipo de mando diseñado, así como también de las gafas de RV, con el fin de brindar al usuario una experiencia de RV de tipo inmersiva, ya que los movimientos del mando acompañados del movimiento de la cabeza de los usuarios permitieron simular estar en un polígono de tiro real.
- El sistema de entrenamiento virtual, al concluir satisfactoriamente las pruebas propuestas brinda al participante la capacidad de reconocimiento del dispositivo de disparo como lo es el mando similar a la Pietro *Beretta*, el uso de texturas realistas junto con la RV inmersiva permite al usuario la adaptación al polígono de disparo virtual.
- El test SUS aplicado a los participantes determinó que el sistema es completamente funcional ya que sus resultados indican una puntuación de 80 puntos sobre 100, además se validan estos resultados con la respuesta de Chi-cuadrado el cual confirmó la hipótesis planteada.

6.2 Recomendaciones

- El uso de las gafas de realidad virtual prolongado puede presentar complicaciones como mareo, se recomienda utilizar el sistema en periodos

cortos, a la vez el sistema debe ser ocupado con prudencia por personas con afecciones como migraña o propensas a mareos.

- El sistema fue desarrollado en la versión más estable de Unity 2019. En el caso que se requiera modificar el sistema, no se debe actualizar las librerías de realidad virtual XR ya que ocasiona problemas de compatibilidad en sus últimas actualizaciones.
- El sistema de entrenamiento virtual debe ser utilizado como apoyo y familiarización para el entrenamiento del ejercicio de tiro de precisión controlado por tiempo, en ningún caso el sistema servirá como reemplazo a dicha actividad.
- Para el uso del sistema se recomienda tener un espacio adecuado, ya que el usuario al estar realizando la práctica de tiro realiza movimientos con la cabeza y manos, lo que puede provocar golpes al usuario o al sistema, pudiendo así provocar lesiones o daños.

Bibliografía

- 3D Natives. (s.f.). Recuperado el 15 de Julio de 2021, de <https://www.3dnatives.com/>
- All3D. (s.f.). *All3D*. Recuperado el 21 de Febrero de 2021, de <https://all3dp.com/es/1/filamento-petg-impresion-3d/>
- Andrade, J., & Yandún, D. (2020). *Prototipo de un sistema de tiro para el entrenamiento del personal de las Fuerzas Armadas del Ecuador, usando realidad virtual y técnicas de inteligencia artificial*. Sangolquí: ESPE.
- Arduino. (s.f.). Recuperado el 21 de Marzo de 2021, de <https://arduino.cl/que-es-arduino/>
- Arduino. (s.f.). *Arduino*. Recuperado el 20 de Marzo de 2021, de <https://arduino.cl/arduino-nano/>
- Arduino. (s.f.). *Arduino*. Recuperado el 20 de Enero de 2021, de <https://store.arduino.cc/usa/arduino-pro-mini>
- Arias, L., Egea, A., & Alfonso, G. (2018). Aprender historia a través del juego de realidad virtual inmersiva "Carthago Nova". Propuesta de integración de un serious game en el proceso de enseñanza-aprendizaje. *CLÍO History and History teaching*.
- Asensio, I. (8 de Noviembre de 2019). *Master.D*. Recuperado el 18 de Marzo de 2021, de <https://www.masterd.es/>
- Autodesk. (s.f.). *Autodesk*. Obtenido de <https://www.autodesk.com/>

Autodesk. (s.f.). *Autodesk*. Recuperado el 03 de Febrero de 2021, de
<https://latinoamerica.autodesk.com/solutions/3d-printing>

Autodesk. (s.f.). *Autodesk*. Recuperado el 06 de Enero de 2021, de
<https://latinoamerica.autodesk.com/products/3ds-max/features>

Aznar, I., Romero, J., & Rodríguez, A. (2018). La tecnología móvil de Realidad Virtual en educación: una revisión del estado . *edmetic*, 4-5.

Barcelona, F. d. (25 de Febrero de 2021). *Facultad de informática de Barcelona*.

Obtenido de

file:///C:/Users/julia/Desktop/TESIS/escrito/tesis%20guia/Tesis_t1692ec.pdf

Beretta. (s.f.). *Beretta*. Recuperado el 29 de Enero de 2021, de
<https://www.beretta.com/es/92-fs/>

Blender. (s.f.). *Blender*. Recuperado el 15 de Diciembre de 2020, de
<https://www.blender.org/>

Campo, P. (2021). *Realidad Virtual Inmersiva en personas mayores*. España:
Universidad de Vigo.

Castro, C. (2018). Desarrollo de un sistema de calificación para un polígono virtual de tiro basado en visión por computador. Sangolquí, Ecuador: Universidad de las Fuerzas Armadas ESPE.

Conocimiento libre. (s.f.). *ConocimientoLibre*. Recuperado el 16 de Febrero de 2021, de
<https://conocimientolibre.mx/microcontrolador/>

CryEngine. (s.f.). *CryEngine*. Recuperado el 7 de Marzo de 2021, de

<https://www.cryengine.com/>

CryEngine. (s.f.). *CryEngine*. Recuperado el 10 de Enero de 2021, de Cryengine:

<https://www.cryengine.com/>

DualTronica. (s.f.). *DualTronica*. Recuperado el 21 de Enero de 2021, de

<https://dualtronica.com/modulos/644-acelerometro-giroscopio-magnetometro-mpu-9250-gy-9250-9-ejes.html>

EC, Art.- 168. (1 de octubre de 2008). *Constitución de la República del Ecuador*.

Montecristi, Ecuador. Obtenido de Asamblea Constituyente:

[https://educacion.gob.ec/wp-](https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/Constitucion.pdf)

[content/uploads/downloads/2012/08/Constitucion.pdf](https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/Constitucion.pdf)

Ecuador, Reglamento a la ley sobre armas, municiones, explosivos y accesorios,

Decreto Ejecutivo 169, Registro Oficial 32, 15-jun. (2015).

EcuRed. (s.f.). *Ecured*. Recuperado el 06 de Enero de 2021, de

https://www.ecured.cu/Cinema_4D

EcuRed. (s.f.). *Ecured*. Recuperado el 07 de Enero de 2021, de

https://www.ecured.cu/Autodesk_3D_Studio_Max

Electronics, A. (s.f.). *AV Electronics*. Recuperado el 23 de Enero de 2021, de

<https://avelectronics.cc/producto/fin-de-carrera-kw11-3z-5a-250v/>

Encalada, J. (2011). *Incidencia de la utilización del polígono de tiro virtual y el mejoramiento en la eficiencia del tiro real en la Escuela Superior Militar Eloy Alfaro*. Sangolquí: Espe.

Epic-Games. (s.f.). *Unreal Engine*. Recuperado el 06 de Marzo de 2021, de <https://www.unrealengine.com/>

ESIC Business & Marketing School. (s.f.). *ESIC*. Recuperado el 15 de Febrero de 2018, de <https://www.esic.edu/>

EVE, Museos e Innovación. (s.f.). *evemuseografia*. Recuperado el 30 de Marzo de 2018, de <https://evemuseografia.com/>

Global, G. (s.f.). Recuperado el 10 de Febrero de 2021, de <https://edu.gcfglobal.org/es/cultura-tecnologica/que-es-una-impresora-3d/1/>

Hp. (s.f.). *Hp*. Recuperado el 08 de Enero de 2021, de <https://www8.hp.com/us/en/vr/reverb-g2-vr-headset.html>

Hxx. (s.f.). *Ingeniería de impresión 3D*. Recuperado el 20 de Febrero de 2021, de <http://hxx.es/2015/03/12/materiales-de-impresion-3d-i-pla-acido-polilactico/>

Impresoras3D. (s.f.). Recuperado el 16 de Febrero de 2021, de <https://www.impresoras3d.com/filamento-pla/>

IngMecánica. (s.f.). Recuperado el 05 de Febrero de 2021, de <https://ingmecanica.com/tutorialsemanal/tutorialn38.html>

Ivars, M., & Suso, L. (2020). EFECTIVIDAD DE LA REALIDAD VIRTUAL INMERSIVA EN. *Journal of move and therapeutic science*, 4-8.

Lázaro, M. (2008). Nuevas tablas de fuerza de la mano para población adulta de Teruel. *SciELO*, 5-6.

Maxon. (s.f.). *Maxon a Nemetschek Company*. Recuperado el 04 de Enero de 2021, de <https://www.maxon.net/es/cinema-4d>

McPro. (s.f.). *McPro*. Recuperado el 15 de Enero de 2021, de Transformacion Digital: <https://www.muycomputerpro.com/2019/03/21/hp-reverb-vr-realidad-virtual#:~:text=A%20nivel%20de%20especificaciones%20el,experiencia%20envolvente%20y%20altamente%20inmersiva.>

Natives, 3. (s.f.). Recuperado el 10 de Febrero de 2021, de <https://www.3dnatives.com/es/filamento-de-abs-impresion-3d-06062019/#!>

NayLamMechatronics. (s.f.). *NayLamMechatronics*. Recuperado el 10 de Enero de 2021, de <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/33-modulo-mpu6050-acelerometro-giroscopio-i2c.html>

Oculus. (s.f.). *Oculus*. Recuperado el 10 de Enero de 2021, de https://www.oculus.com/rift/?locale=es_LA

Of3lia. (s.f.). Recuperado el 18 de Febrero de 2021, de <https://of3lia.com/pla-vs-abs-vs-petg-comparativa/>

Onion. (s.f.). Recuperado el 05 de Marzo de 2021, de <https://www.placasdedesarrollo.com/onion/>

Plasticsl. (s.f.). *Plasticsl*. Recuperado el 01 de Febrero de 2021, de <https://plasticsl.com/inyeccion-plastico-que-es-para-que-sirve/>

Play Station. (s.f.). *Play Station*. Recuperado el 07 de Marzo de 2021, de

<https://www.playstation.com/>

RAE. (s.f.). *RAE*. Recuperado el 05 de Febrero de 2021, de

<https://definicion.de/entrenamiento/>

Raspberry. (s.f.). Recuperado el 02 de Marzo de 2021, de [https://raspberrypi.cl/que-es-](https://raspberrypi.cl/que-es-raspberry/)

[raspberry/](https://raspberrypi.cl/que-es-raspberry/)

Revista Gestión. (s.f.). Recuperado el 17 de Marzo de 2021, de

<https://www.revistagestion.ec/economia-y-finanzas-analisis/el-gasto-militar-aun-pesa-sobre-el-pib-y-el-presupuesto-nacional>

Riba Romeva, C. &. (2006). *Ingeniería concurrente: una metodología integradora*.

Castellon de la Plana: Edicions UPC.

Rodríguez, E. (s.f.). *Xataka*. Recuperado el 09 de Noviembre de 2020, de

<https://www.xataka.com/>

Román, E., Ortega, A., Gutierrez, J., & Nahuel, A. (2017). *WikiGraphic*. Recuperado el

16 de Febrero de 2021, de <https://sites.google.com/>

Rubi3D. (17 de Enero de 2018). *Rubi3D*. Recuperado el 07 de Marzo de 2021, de

<https://rubi3d.com/>

Scheetz, T. (2011). *Teoría de la gestión económica de las Fuerzas Armadas*. Buenos

Aires: Escuela de defensa nacional.

TeslaBem. (s.f.). *TeslaBem*. Recuperado el 22 de Enero de 2021, de

<https://teslabem.com/blog/arduinios-mas-pequenos/>

- Texas Parks & Wildlife. (s.f.). Obtenido de <https://tpwd.texas.gov/education/hunter-education/educacion-para-cazadores/capitulo-2>
- The Blender Foundation. (s.f.). *Blender*. Recuperado el 21 de Marzo de 2021, de <https://www.blender.org/>
- Time Graphics. (s.f.). *Time Graphics*. Recuperado el Marzo de 2021, de <https://time.graphics/event>
- Unity. (10 de Octubre de 2020). *Unity*. Recuperado el 17 de Marzo de 2021, de Unity: <https://unity.com/es>
- Unity Technologies. (s.f.). *Unity*. Recuperado el 05 de Marzo de 2021, de <https://unity.com/>
- UnrealEngine. (20 de Octubre de 2020). *Unreal Engine*. Recuperado el 04 de Marzo de 2021, de <https://www.unrealengine.com/>
- Urbina. (2013). *Evaluación de Proyectos*. México: McGRAW-HILL Editores S.A.
- Vargas, C. (2019). *REALIDAD VIRTUAL INMERSIVA APLICADA*. Valencia: Universidad Politecnica de Valencia.
- Venturini, G. (16 de Julio de 2020). *Tecnología + Informática*. Recuperado el 17 de Febrero de 2021, de <https://www.tecnologia-informatica.com>
- Villalva, P. (2015). Gerencia de tecnologías de simuladores de polígonos de tiro. Quito, Ecuador: Pontificia Universidad Católica del Ecuador.

Vive. (s.f.). *Vive*. Recuperado el 10 de Enero de 2021, de

<https://www.vive.com/mx/product/vive-pro/>

ANEXOS