



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA MECATRÓNICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN MECATRÓNICA**

**TEMA: INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN
SISTEMA DE REALIDAD AUMENTADA CON ASISTENTE
ROBÓTICO PARA EL MEJORAMIENTO DEL APRENDIZAJE,
CREATIVIDAD Y ENTRETENIMIENTO PARA NIÑOS DE
EDUCACIÓN PRIMARIA**

**AUTORES: CRISTIAN DAVID CHIMBO TAMAMI
VÍCTOR HUGO JIMÉNEZ CAIZA**

DIRECTORA: ING. PATRICIA CONSTANTE P.

LATACUNGA

2017



**DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA MECATRONICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, ***“INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE REALIDAD AUMENTADA CON ASISTENTE ROBÓTICO PARA EL MEJORAMIENTO DEL APRENDIZAJE, CREATIVIDAD Y ENTRETENIMIENTO PARA NIÑOS DE EDUCACIÓN PRIMARIA”*** realizado por los señores ***CRISTIAN DAVID CHIMBO TAMAMI y VÍCTOR HUGO JIMÉNEZ CAIZA***, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores ***CRISTIAN DAVID CHIMBO TAMAMI y VÍCTOR HUGO JIMÉNEZ CAIZA*** para que lo sustenten públicamente.

Latacunga, 22 de agosto del 2017

Ing. Patricia Nataly Constante Prócel

DIRECTORA



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRONICA

AUTORÍA DE RESPONSABILIDAD

Nosotros, **CRISTIAN DAVID CHIMBO TAMAMI y VÍCTOR HUGO JIMÉNEZ CAIZA**, con cédula de ciudadanía N° 0202348868 y N° 0603578576 respectivamente, declaramos que este trabajo de titulación "**INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE REALIDAD AUMENTADA CON ASISTENTE ROBÓTICO PARA EL MEJORAMIENTO DEL APRENDIZAJE, CREATIVIDAD Y ENTRETENIMIENTO PARA NIÑOS DE EDUCACIÓN PRIMARIA**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este trabajo es de nuestra autoría, en virtud de ello nos responsabilizamos del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 22 de agosto del 2017

Cristian David Chimbo Tamami

C.C.: 0202348868

Víctor Hugo Jiménez Caiza

C.C.:0603578576



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRONICA

AUTORIZACIÓN

Nosotros, **CRISTIAN DAVID CHIMBO TAMAMI** y **VÍCTOR HUGO JIMÉNEZ CAIZA**, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar en el repositorio institucional el trabajo de titulación “**INVESTIGACIÓN, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE REALIDAD AUMENTADA CON ASISTENTE ROBÓTICO PARA EL MEJORAMIENTO DEL APRENDIZAJE, CREATIVIDAD Y ENTRETENIMIENTO PARA NIÑOS DE EDUCACIÓN PRIMARIA**” cuyo contenido, ideas y criterios son de nuestra autoría y responsabilidad.

Latacunga, 22 de agosto del 2017

Cristian David Chimbo Tamami

C.C.: 0202348868

Víctor Hugo Jiménez Caiza

C.C.:0603578576

DEDICATORIA

Dedico este trabajo a mi familia.

A mis padres, Luis Chimbo y Elvira Tatami, quienes han estado junto a mi durante todo este proceso de estudio universitario, brindándome su cariño y apoyo incondicional.

A mis hermanos, sus palabras de aliento en los momentos más difíciles, me han dado fuerza para continuar y no perder el rumbo de mis metas trazadas.

Cristian Chimbo

A Dios, por ser la luz y guía de mi camino
Quien me ha hecho conocer muchos amigos
y amigas que me han apoyado en momentos
alegres y tristes.

A mis padres Elsa Caiza y Alfredo Jiménez por
ser partícipes e incentivadores de ánimo día a día
esfuerzo, dedicación invertidos en mi persona
para ser un profesional de éxito.

A mis abuelitos María Lara y Ángel Caiza que
me han brindado su respaldo incondicional
en los momentos más duros.

Víctor Jiménez

AGRADECIMIENTOS

Expresamos nuestros sinceros agradecimientos a Dios por darnos perseverancia y paciencia para culminar este proyecto.

Agradecemos infinitamente a la Ingeniera Patricia Constante por habernos guiado y ayudado en el desarrollo de nuestro proyecto de tesis.

A la docente Nereida Caiza por habernos colaborado en el desarrollo de las pruebas del proyecto así también en sugerirnos ideas de juegos creativos para el aprendizaje de los niños y niñas de educación básica

A nuestros amigos, que sería de la vida sin amigos, gracias por todos los momentos de alegrías, tristezas, euforia, momentos que siempre lo llevaremos en nuestros pensamiento y corazones.

A nuestros docentes de la Universidad de las Fuerzas Armadas Espe sede Latacunga por sus conocimientos brindados a lo largo de la carrera

Gracias por todo

Cristian Chimbo - Víctor Jiménez

ÍNDICE DE CONTENIDO

CARÁTULA.....	i
CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL).....	iv
DEDICATORIA.....	v
AGRADECIMIENTOS.....	vi
ÍNDICE DE CONTENIDO.....	vii
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS.....	xviii
RESUMEN.....	xx
ABSTRACT.....	xxi

CAPÍTULO I **1****FUNDAMENTO TEÓRICO** **1**

1.1. Educación en el Ecuador.....	1
1.1.1. Estándares de Calidad Educativa.....	2
1.1.2. Estado Actual de la instrucción primaria en el Ecuador.....	2
1.2. Tecnologías implementadas para la mejora del aprendizaje.....	3
1.2.1. Pizarras digitales.....	3
1.2.2. Tablets en las aulas.....	4
1.2.3. Robots.....	5
1.2.4. Visión Artificial en la educación.....	5
1.3. Visión artificial.....	6
1.3.1. Introducción.....	6
1.3.2. Definición de Visión Artificial.....	6

1.3.3.	Aplicaciones de la visión artificial.....	7
1.3.4.	Procesamiento digital de imágenes.....	7
1.3.5.	Librerías de Visión Artificial.....	8
1.3.6.	Reconocimientos de Objetos.....	9
1.3.7.	Entrenamiento de los clasificadores.....	13
1.4.	Realidad Aumentada.....	14
1.4.1.	Características.....	14
1.4.2.	Diferencia entre realidad aumentada y realidad virtual.....	15
1.4.3.	Elementos de la Realidad Aumentada.....	15
1.4.4.	Realidad aumentada aplicada en la educación de los estudiantes.....	16
1.4.5.	Plataformas para desarrollo de realidad aumentada.....	18
1.5.	Robots interactivos aplicados a la educación.....	20
1.5.1.	Robots Actualmente implementados en la educación.....	20
1.5.2.	Beneficios de la robótica en la educación de los niños y niñas.....	22
1.6.	Sistemas de comunicación para transmisión de datos.....	23

CAPÍTULO II **25**

ANÁLISIS Y SELECCIÓN DE ALTERNATIVAS **25**

2.1.	Introducción.....	25
2.2.	Sistemas y componentes del proyecto.....	26
2.3.	Necesidades y especificaciones técnicas.....	27
2.3.1.	Especificaciones técnicas.....	28
2.3.2.	Método de evaluación.....	29
2.4.	Alternativas de Realidad Aumentada.....	29
2.4.1.	PC / Procesador.....	31
2.4.2.	Monitor.....	34

2.4.3.	Cámara.....	35
2.4.4.	Software	37
2.5.	Robot Interactivo	40
2.5.1.	Aspecto del Robot.	40
2.5.2.	Tarjeta de control.....	41
2.5.3.	Motores.....	43
2.5.4.	Módulo de control de motores	45
2.5.5.	Mecanismo de transmisión.....	46
2.5.6.	Módulo de matriz leds	48
2.5.7.	Módulo de audio.....	50
2.6.	Sistema de transmisión de datos	52
2.7.	Componentes seleccionados	54

CAPÍTULO III **56**

DISEÑO Y CONSTRUCCIÓN DEL ROBOT DIDÁCTICO **56**

3.1.	Introducción	56
3.2.	Diseño Mecánico	56
3.2.1.	Diseño de la estructura del robot	56
3.2.2.	Análisis estático de la estructura del robot	63
3.2.3.	Análisis de Impacto de la carcasa del robot.....	75
3.2.4.	Análisis de espesor de la carcasa del robot.....	77
3.2.5.	Prototipado rápido de las piezas diseñadas.....	79
3.3.	Diseño Eléctrico y Electrónico	83
3.3.1.	Configuración-programación de los módulos y la tarjeta Arduino.	83
3.3.2.	Consumo de la energía eléctrica del robot.....	94

CAPÍTULO IV	96
REALIDAD AUMENTADA Y TRANSMISIÓN DE DATOS	96
4.1. Introducción	96
4.2. Interfaz de juegos de realidad aumentada	98
4.2.1. Softwares necesarios	98
4.2.2. Reconocimiento de la mano cerrada	99
4.2.3. Diseño de los algoritmos	108
4.2.4. Creación de pistas de Audio	116
4.3. Sistema de Comunicación	118
4.3.1. Comunicación entre Sistema RA y Arduino Maestro	118
4.3.2. Comunicación entre Arduino Maestro y Esclavo.	119
CAPÍTULO V	127
ENSAMBLE, IMPLEMENTACION Y PRUEBAS	127
5.1. Ensamble del Robot	127
5.1.1 Ensamble de las piernas.	127
5.1.2. Ensamble de los Brazos	129
5.1.3. Ensamble del Torso.....	130
5.1.4. Ensamble de la cabeza.....	133
5.2. Implementación del sistema de Realidad Aumentada	135
5.3. Especificaciones de los sistemas implementados.....	136
5.4. Pruebas y Análisis de Resultados	137
5.3.1. Pruebas de etapa de Preprocesamiento y Reconocimiento.....	137
5.3.2. Pruebas de funcionamiento del Robot	140
5.3.4. Pruebas de funcionamiento	146
5.3.5. Pruebas de usabilidad.....	159

5.5.	Validación de la hipótesis	162
5.6.	Análisis Técnico – Económico.....	169
CAPÍTULO VI		173
CONCLUSIONES Y RECOMENDACIONES		173
6.1.	Conclusiones	173
6.2.	Recomendaciones.....	175
REFERENCIAS BIBLIOGRÁFICAS		177
ANEXOS		182
ANEXO A: Instrumentos de evaluación y resultados		
ANEXO B: Casa de la calidad		
ANEXO C: Planos de piezas modeladas		
ANEXO D: Diagrama de circuitos		
ANEXO E: Programas principales		
ANEXO F: Instalación de Softwares		
ANEXO G: Lista de juegos RA desarrollados		
ANEXO H: Listado de canciones		

ÍNDICE DE FIGURAS

Figura 1 Estructura de la norma educativa ecuatoriana.....	1
Figura 2 Pizarras digitales	4
Figura 3 Uso de Tablets.....	4
Figura 4 Kit Lego Mindstorms	5
Figura 5 Aplicaciones de Visión Artificial	5
Figura 6 Ojo Artificial.....	6
Figura 7 Traductor de texto en tiempo real	7
Figura 8 Etapas del procesamiento de imágenes	7
Figura 9 Característica Haar	10
Figura 10 Funcionamiento del Operador LBP.....	11
Figura 11 Estructura del Clasificador Weak.....	12
Figura 12 Estructura de un clasificador Strong	12
Figura 13 Funcionamiento del clasificador cascada	13
Figura 14 Realidad Aumentada	14
Figura 15 Taxonomía de Realidad Mixta	15
Figura 16 Elementos de la Realidad Aumentada.....	16
Figura 17 Realidad Aumentada en la Educación	16
Figura 18 Juegos Didácticos con Realidad Aumentada.....	17
Figura 19 Matemáticas con Realidad Aumentada	17
Figura 20 ARcrowd Planetas	18
Figura 21 LearnAR aparato digestivo	18
Figura 22 Probadores virtuales	19
Figura 23 Juegos RA con Pygame	19
Figura 24 Teebot robot educativo	20
Figura 25 Nao educación.....	21
Figura 26 Interacción con Zowi	21
Figura 27 Rapiro Robot.....	22
Figura 28 Robot en la educación	22
Figura 29 Red Bluetooth	23

Figura 30 Red Wifi	24
Figura 31 Red ZigBee.....	24
Figura 32 Proceso de Diseño.....	25
Figura 33 Sistemas y componentes.....	26
Figura 34 Componentes seleccionados.....	55
Figura 35 Estructura del robot Nao	57
Figura 36 Cabeza del robot didáctico	57
Figura 37 Oreja del robot didáctico	58
Figura 38 Cuello del robot didáctico.....	58
Figura 39 Espalda del robot didáctico.....	59
Figura 40 Pecho del robot didáctico.....	59
Figura 41 Columna del robot didáctico	60
Figura 42 Torso 1 del robot didáctico.....	60
Figura 43 Torso 2 del robot didáctico.....	61
Figura 44 Articulación brazo del robot didáctico	61
Figura 45 Soporte pierna del robot didáctico	62
Figura 46 Rueda de motor del robot didáctico	62
Figura 47 Esqueleto de soporte.....	64
Figura 48 Geometría Fija en la columna.....	65
Figura 49 Fuerza aplicada en la cara inferior de la columna	66
Figura 50 Mallado de la columna del robot.....	66
Figura 51 Tensión de Von Mises en la columna	67
Figura 52 Factor de seguridad en la columna.....	69
Figura 53 Geometría Fija en la columna.....	69
Figura 54 Fuerza aplicada en la cara inferior de la columna	70
Figura 55 Mallado del soporte de pierna.....	70
Figura 56 Tensión de Von Mises en el soporte de pierna.....	71
Figura 57 Factor de seguridad en el soporte de pierna	72
Figura 58 Geometría Fija en el cuello	72
Figura 59 Fuerza aplicada en la cara superior del cuello.....	73
Figura 60 Torsión aplicada en la cara del cuello.....	73

Figura 61	Mallado del cuello	74
Figura 62	Tensión de Von Mises en el cuello	74
Figura 63	Factor de seguridad en el cuello	75
Figura 64	Análisis de impacto en el pecho	76
Figura 65	Mallado del pecho.....	76
Figura 66	Tensión de Von Mises en la pieza “pecho”	77
Figura 67	Análisis de espesor de la cabeza posterior.....	78
Figura 68	Análisis de espesor de la cabeza frontal.....	78
Figura 69	Diseño Final del Robot Didáctico	79
Figura 70	Impresión 3D del pecho	80
Figura 71	Impresión 3D de la espalda	80
Figura 72	Pulido de piezas impresas	81
Figura 73	Piezas de la carcasa del robot.....	81
Figura 74	Proceso de pintado de las piezas	82
Figura 75	Piezas de la carcasa pintadas	82
Figura 76	Conexión Arduino Mega y DFPlayer mini mp3	83
Figura 77	Proceso para reproducir un audio.....	84
Figura 78	Varios comandos básicos	85
Figura 79	Volumen y selección de canción.....	86
Figura 80	Conexión Arduino Mega y Matriz led	86
Figura 81	Proceso imprimir gesto en la matriz de led	87
Figura 82	Librerías y pines de salida Matriz de leds	88
Figura 83	Conexión Arduino Mega y Matriz led	88
Figura 84	Conexión Arduino Mega y Matriz led	89
Figura 85	Conexión Arduino Mega y el módulo de motores	90
Figura 86	Proceso para controlar motores.....	91
Figura 87	Librerías del módulo de motores	92
Figura 88	Comando que determina la velocidad de los motores	92
Figura 89	Comandos para mover el motor	92
Figura 90	Diagrama de la comunicación maestro - esclavo	93
Figura 91	Pilas recargables doble A 4700mah	94

Figura 92 Descripción de los sistemas	96
Figura 93 Diagrama de Flujo del Sistema Completo.....	97
Figura 94 Entorno de programación de Python	99
Figura 95 Reconocimiento de objetos.....	100
Figura 96 Estructura de directorio.....	101
Figura 97 Adquisición de imágenes	101
Figura 98 Índice de imágenes negativas	102
Figura 99 Índice de imágenes positivas.....	102
Figura 100 Obtención de coordenadas de la mano mediante Objectmarker ..	103
Figura 101 Línea de comando para creación de muestras.....	103
Figura 102 Creación de muestras.....	104
Figura 103 Línea de comando para entrenamiento del clasificador.....	104
Figura 104 Entrenamiento del clasificador.....	105
Figura 105 Conversión de imagen a escala de grises	106
Figura 106 Filtrado de imagen para eliminación de ruido	106
Figura 107 Funcionamiento del detector.....	107
Figura 108 Proceso General de los juegos RA.....	108
Figura 109 Importación de módulos.....	110
Figura 110 Definición de variables.....	110
Figura 111 Inicialización de cámara de video	111
Figura 112 Procesamiento de imagen	111
Figura 113 Detección de la mano cerrada	111
Figura 114 Extracción de posición y tamaño de la mano.....	112
Figura 115 Conversión imagen cv2 a imagen de superficie (pygame)	112
Figura 116 Creación del marcador.....	113
Figura 117 Clase para creación de botones	113
Figura 118 Algoritmo clase escena.....	114
Figura 119 Función para agregar texto.....	114
Figura 120 Función para crear sprites	115
Figura 121 Juego "Números Naturales" desarrollado	115
Figura 122 Grabador de voz	116

Figura 123 Convertir audios a mp3.....	116
Figura 124 Nivelación de volumen de los audios mp3.....	117
Figura 125 Audios mp3 almacenado es la tarjeta micro SD	117
Figura 126 Estructura de Comunicación entre los Juegos RA y el Robot.....	118
Figura 127 Abrir el puerto serial en Phytion.....	119
Figura 128 Escribir información en el puerto serial.....	119
Figura 129 Conexión Arduino Mega y Módulo Bluetooth HC-06	120
Figura 130 Selección tarjeta Arduino Mega 2650	121
Figura 131 Selección Puerto Serial.....	121
Figura 132 Configuración del nombre y contraseña módulo HC-06	122
Figura 133 Configuración del módulo HC-06.....	122
Figura 134 MAC del módulo bluetooth esclavo	123
Figura 135 Conexión Arduino Mega y Módulo Bluetooth HC-05	123
Figura 136 Ejemplo Software Serial.....	124
Figura 137 Definición de los pines de transmisión de datos	124
Figura 138 Velocidad y comunicación del puerto serial	124
Figura 139 Comando AT.....	125
Figura 140 Comando configurar función del dispositivo	125
Figura 141 Comando para conectar a una MAC específica	126
Figura 142 Comando enlazar el módulo maestro con el esclavo.....	126
Figura 143 Ensamble del Pie	127
Figura 144 Ensamble de la pierna	128
Figura 145 Ensamble final de la pierna derecha.....	128
Figura 146 Ensamble del brazo	129
Figura 147 Implementación del interruptor de encendido-apagado.....	130
Figura 148 Implementación de los mecanismos de soporte	131
Figura 149 Ensamble del cuerpo del robot	131
Figura 150 Implementación de la tarjeta de control	132
Figura 151 Ensamble Torso parte electrónica	133
Figura 152 Ensamble Cabeza.....	134
Figura 153 Ensamble final del Robot.....	135

Figura 154 Componentes del Sistema RA.....	135
Figura 155 Transformación a escala de grises	137
Figura 156 Filtro de Convolución	138
Figura 157 Algoritmo del Clasificador	138
Figura 158 Corrección del código de recepción y reenvío de datos	145
Figura 159 Pruebas de funcionamiento en alumna.....	147
Figura 160 Menú de Categorías de Juegos RA.....	147
Figura 161 Niño interactuando con los juegos RA	148
Figura 162 Parlantes de 8Ω 0,5w	149
Figura 163 Código de programación mejorado para interacción de juegos....	150
Figura 164 Detección mejorada del marcador	151
Figura 165 Implementación del segundo marcador	151
Figura 166 Parámetro del tiempo de activación de botones	152
Figura 167 Parámetro para control de iteraciones	152
Figura 168 Pruebas aplicando el método tradicional	154
Figura 169 Pruebas aplicando herramienta didáctica RA y Robot.....	154
Figura 170 Pruebas en el sector de Belisario Quevedo.....	155
Figura 171 Análisis de la tasa de mejora en Matemática.....	157
Figura 172 Análisis de la tasa de mejora en Lenguaje	157
Figura 173 Análisis de la tasa de mejora en Ciencias Naturales	158
Figura 174 Análisis de la tasa de mejora en Robótica.....	159
Figura 175 Resultados obtenidos del Cuestionario de usabilidad	161
Figura 176 Análisis de distribución de t de dos colas	167

ÍNDICE DE TABLAS

Tabla 1 Librerías de Visión Artificial.....	9
Tabla 2 Especificaciones técnicas del proyecto.....	28
Tabla 3 Escala de calificación.....	29
Tabla 4 Comparación entre Sistema RA Fijo y Móvil.....	30
Tabla 5 Evaluación de alternativas del Sistema RA.....	31
Tabla 6 Comparación de alternativas de PC/Procesador	32
Tabla 7 Evaluación de alternativas del PC/Procesador	33
Tabla 8 Comparación entre monitor LCD y LED.....	34
Tabla 9 Comparación entre las alternativas de la cámara	36
Tabla 10 Evaluación de alternativas para la cámara	37
Tabla 11 Comparación de alternativas de Software	38
Tabla 12 Evaluación de alternativas del software	39
Tabla 13 Comparación de robots según su arquitectura	40
Tabla 14 Comparación de tarjetas de control	41
Tabla 15 Evaluación de alternativas de la tarjeta de control.....	42
Tabla 16 Comparación de los motores	43
Tabla 17 Evaluación de alternativas de los motores.....	44
Tabla 18 Comparación de la tarjeta de control de motores	45
Tabla 19 Evaluación de alternativas de la tarjeta control de motores.....	46
Tabla 20 Comparación de mecanismos de transmisión	47
Tabla 21 Evaluación de alternativas del mecanismo de transmisión	48
Tabla 22 Comparación de la matriz de led	49
Tabla 23 Evaluación de alternativas de la matriz de leds	50
Tabla 24 Comparación del módulo de audio	51
Tabla 25 Evaluación de alternativas del módulo de audio	52
Tabla 26 Comparación del sistema de transmisión de datos.....	53
Tabla 27 Evaluación de alternativas del sistema de transmisión de datos	54
Tabla 28 Propiedades físicas del PLA	63
Tabla 29 Lista de peso de cada componente del robot	64

Tabla 30 Lista de comandos del módulo DFPlayer mini Mp3	84
Tabla 31 Lista de comandos matriz led mBot	88
Tabla 32 Lista de comandos del módulo Shield L293D	91
Tabla 33 Ventajas y desventajas de utilizar pilas recargables	94
Tabla 34 Consumo de cada componente electrónico	95
Tabla 35 Principales algoritmos para desarrollo de juegos RA.....	109
Tabla 36 Comandos para la comunicación entre Python-Pygame y Arduino .	119
Tabla 37 Lista de comandos AT	120
Tabla 38 Especificaciones finales de los sistemas	139
Tabla 39 Determinación de los parámetros del clasificador.....	140
Tabla 40 Primera prueba de movimiento en el robot	141
Tabla 41 Corrección del código de control de los motores	143
Tabla 42 Corrección de la primera prueba de movimiento en el robot.....	143
Tabla 43 Gestos o expresiones del robot	144
Tabla 44 Pruebas preliminares de transmisión de datos	146
Tabla 45 Pruebas finales de Transmisión de datos	156
Tabla 46 Resultados del cuestionario de Aprendizaje	160
Tabla 47 Promedio del cuestionario de usabilidad.....	163
Tabla 48 Taza de comparación entre los dos métodos	164
Tabla 49 Datos del promedio utilizando el método tradicional y experimental	164
Tabla 50 Resultados del test Shapiro Wilk en las pruebas finales.....	165
Tabla 51 Resumen de datos para el análisis del t estadístico	166
Tabla 52 Datos del promedio utilizando el método T student	168
Tabla 53 Resultado Análisis de Varianza para prueba de hipótesis	169
Tabla 54 Detalle de costos de materiales de la Realidad Aumentada	170
Tabla 55 Detalle de costos del Robot interactivo	170
Tabla 56 Detalle de costos de las impresiones 3D	171
Tabla 57 Detalle de costos de la comunicación inalámbrica.....	171
Tabla 58 Detalle de gastos varios.....	172
Tabla 59 Costo general del proyecto.....	173

RESUMEN

La educación constituye un medio para el desarrollo de una sociedad, es por eso la necesidad de contribuir a la mejora continua de las técnicas de enseñanza, principalmente en los primeros niveles de educación, para lo cual se propone el diseño e implementación de un sistema de Realidad Aumentada (RA) con asistente robótico. El proyecto se compone de tres subsistemas: el sistema de realidad aumentada el cual brinda una visión en tiempo real del entorno físico combinada con elementos virtuales, para esto se desarrolla un conjunto de juegos educativos basados en este tipo de tecnología, donde los estudiantes se visualizan en una pantalla a través de una cámara y pueden aprender resolviendo ejercicios de diferentes asignaturas, interactuando con los elementos virtuales utilizando su puño o un objeto de color; el robot asistente, es un robot didáctico de aspecto humanoide, cuya principal función es brindar las instrucciones de los juegos mediante sonidos de voz, puede también realizar movimientos básicos como caminar, mover los brazos y realizar gestos emocionales, además podrá ser comandado, es decir el usuario tendrá la opción de poder controlar los movimientos del robot mediante la interfaz de realidad aumentada, estos dos subsistemas se encuentra sincronizados a través de una sistema de comunicación vía bluetooth. La implementación de este proyecto permite introducir una nueva herramienta didáctica que sirve de apoyo para la excelencia en el aprendizaje de niños y niñas de seis a nueve años, que cursan preparatoria y básica elemental.

PALABRAS CLAVE:

- **EDUCACIÓN BÁSICA**
- **REALIDAD AUMENTADA**
- **ROBÓTICA**

ABSTRACT

Education is a mean for society development, that is why it is necessary to contribute to the continuous improvement of teaching techniques, especially in the first levels of education, so the design and implementation of a system of Augmented Reality (RA) with robotic assistant has been proposed. The project is made up of three subsystems: the augmented reality system which provides a real-time view of the physical environment combined with virtual elements. For this, a set of educational games based on this type of technology is developed, where students visualize themselves on a screen through a camera and can solve exercises of different subjects, interacting with the virtual elements using their fist or a colored object; The assistant robot, is a didactic robot with a humanoid aspect, and its main function is to provide the instructions of the games with voice sounds. It can also perform basic movements such as walking, moving the arms performing emotional gestures, and can be commanded. The user will have the option of being able to control the movements of the robot through the augmented reality interface, these two subsystems are synchronized through a communication system via bluetooth. The implementation of this project allows the introduction of a new didactic tool that supports excellence children is learning from six to nine years old, who are in elementary and high school.

KEYWORDS:

- **PRIMARY EDUCATION**
- **AUGMENTED REALITY**
- **ROBOTICS**

CAPÍTULO I

FUNDAMENTO TEÓRICO

1.1. Educación en el Ecuador

La educación en el Ecuador se encuentra reglamentada por el Ministerio de Educación, organismo oficial, el cual garantiza que la educación pública gratuita es un derecho de todos los ciudadanos desde el nivel básico hasta el bachillerato o su equivalente (Cantos, 2015) . El Ministerio de Educación procura proporcionar herramientas y oportunidades que permitan alcanzar un óptimo desarrollo académico, cuyo objetivo es tener ciudadanos preparados, no únicamente en áreas técnicas, también en valores que consoliden una sociedad más inclusiva y participativa.

En la figura 1, se aprecia la como está estructurada la norma educativa ecuatoriana.

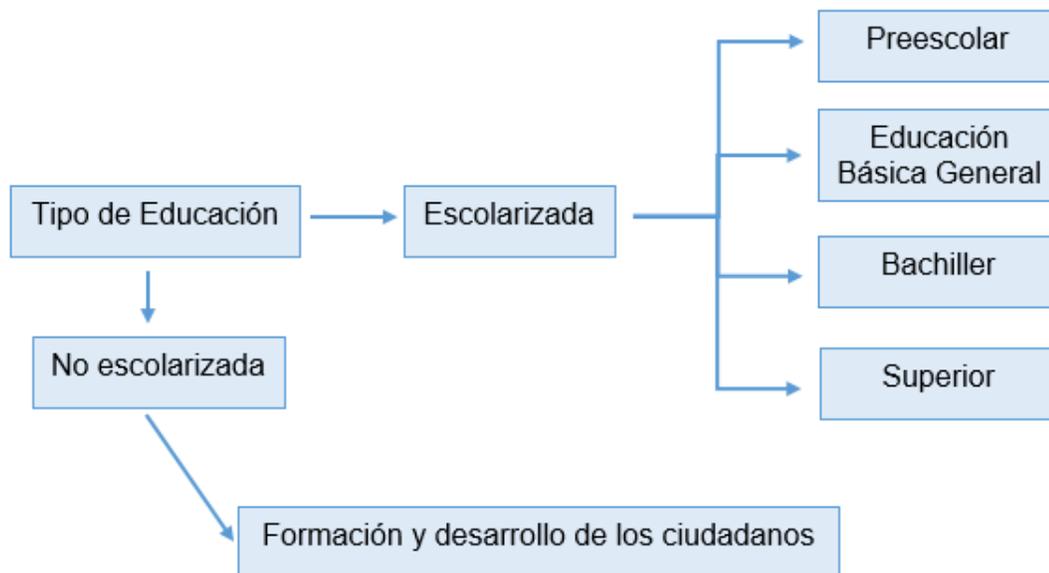


Figura 1 Estructura de la norma educativa ecuatoriana

Fuente: (Pineda, 2014)

La constitución política del país menciona en su artículo 26 que la educación es un derecho en el que todos los ciudadanos están exentos de participar, mientras que en el artículo 27 añade que la educación mejora el grado de educación, por lo cual el sistema educativo está basado en normas de calidad educativa. (Ministerio de Educación, 2016)

1.1.1. Estándares de Calidad Educativa.

Los estándares de calidad educativa son directrices de logros esperados de entidades públicas, cuyo principal objetivo es alcanzar una educación de nobleza. Estos estándares al aplicarse en los estudiantes, corresponde a las destrezas que los alumnos van adquiriendo en las diferentes actividades del área curricular mediante la aplicación de diversos métodos de estudio, cuando se aplican a las escuelas, corresponde a la organización y gestión de las actividades institucionales, destinadas a obtener resultados favorables que permitan mejorar la instrucción estudiantil. (Ministerio de Educación, 2016)

La mejora de las normas educativas, permite las siguientes características:

- Ser críticos, con facilidad de comprensión.
- Estar inspirados en ideales educativos.
- Fundamentarse en valores ecuatorianos y universales.
- Con equidad en estándares internacionales pero ajustable a la realidad ecuatoriana.
- Presentar un desafío para los actores e instituciones del sistema.

1.1.2. Estado Actual de la instrucción primaria en el Ecuador.

La Educación General Básica en el Ecuador está compuesta diez niveles de estudio, desde la preparatoria hasta la educación básica superior. Una vez culminado estos niveles los jóvenes están listos para continuar con el Bachillerato y por ende ser partícipes en la vida política y social y ejercer su rol como entes

ecuatorianos. (Cantos, 2015). Los alumnos que terminan sus estudios de Educación General Básica serán capaces de:

- Relacionarse e intervenir diligentemente en una sociedad con una diversidad de culturas y organizaciones internacionales.
- Resolver problemas de la vida cotidiana tomando en cuenta un criterio propio, razonativo y soñador.
- Defender y priorizar su salud sexual, física y psicológica, ante todo.
- Salvaguardar la naturaleza cuidándola y conservándola en un ambiente idóneo para el desarrollo de la vida.
- Aplicar tecnologías para dar solución a problemas de actividades académicas reforzadas con etapas de investigación.
- Conocer una instrucción básica extranjera que permita tener una comunicación asequible.
- Utilizar eficazmente el tiempo de ocio en actividades inherentes a actos culturales, deportivos, sociales con la finalidad de aumentar la convivencia social practicando valores éticos.

1.2. Tecnologías implementadas para la mejora del aprendizaje.

El desarrollo tecnológico ha dotado al personal docente de nuevas herramientas con las que puede realizar sus labores didácticas con el alumnado, esas oportunidades representan potenciar el desarrollo educativo y fomentar la atención en las aulas. (Colegio Juan de Valdés, 2015). Es imperioso en la actualidad adaptarse a los avances tecnológicos, es por ello que se menciona algunas tecnologías que se están aplicando en los salones de clase.

1.2.1. Pizarras digitales

Las pizarras digitales y sus correspondientes complementos, ordenador, proyector y sistema de altavoces, un ejemplo de esta tecnología se observa en la figura 2. (Colegio Juan de Valdés, 2015)



Figura 2 Pizarras digitales

Fuente: (Colegio Juan de Valdés, 2015)

Brindan beneficios al momento de aprender, como:

- Las pizarras y la conexión vía a internet son recursos multimedia que permiten reforzar los conocimientos en clases.
- Los alumnos se incentivan frente a estas técnicas, les causa curiosidad y gusto por ende entienden los contenidos de mejor forma.
- Las actividades del aula con la implementación de las TICs, permiten buscar información para la mejor comprensión de los trabajos.

1.2.2. Tablets en las aulas

La utilización de las Tablets dentro del aula como se observa en la figura 3, es una de las herramientas multimedia más novedosa para la innovación educativa. (Colegio Juan de Valdés, 2015)



Figura 3 Uso de Tablets

Fuente: (Colegio Juan de Valdés, 2015)

1.2.3. Robots

Los kits de robótica, son recursos didácticos para desarrollar un pensamiento lógico en la formación estudiantil de los niños y niñas, ya que desde temprana edad aprenden conceptos básicos de robótica (ver figura 4), despertando el interés de las actuales tecnologías. (Chuquimarca, 2016)



Figura 4 Kit Lego Mindstorms

Fuente: (Chuquimarca, 2016)

1.2.4. Visión Artificial en la educación

Las nuevas tecnológicas de la educación, apuntan al desarrollo de aplicaciones de visión artificial, como la realidad aumentada (ver figura 5), que es una nueva tendencia que está empezando a ser utilizada en escuela como herramientas de aprendizaje en diferentes asignaturas.



Figura 5 Aplicaciones de Visión Artificial

Fuente: (Cadillo, 2016)

1.3. Visión artificial

1.3.1. Introducción.

La visión constituye uno de los sentidos más importante de los seres humanos, ésta es empleada para obtener información visual del mundo físico, colores, formas etc., aproximadamente el 70% de las tareas que el cerebro realiza requiere de la información visual captada por los ojos. Actualmente las investigaciones tecnológicas realizadas sobre el sentido de la visión han permitido que se fabriquen dispositivos que simulan el comportamiento del ojo humano (ver figura 6), permitiendo obtener información visual en tiempo real del entorno físico, pero todavía no han logrado replicar exactamente la funcionalidad de este sentido de visión.



Figura 6 Ojo Artificial

Fuente: (El Estanquillo, 2013)

1.3.2. Definición de Visión Artificial.

La visión artificial toma imágenes reales mediante una computadora convirtiéndolas en imágenes digitales con la finalidad de obtener información de un entorno para su futuro estudio. (Garcia, 2013) .Por ende la visión por computador cumple la función de extraer información del mundo físico a partir de imágenes digitales haciendo uso de un ordenador.

1.3.3. Aplicaciones de la visión artificial.

La visión artificial esta inmiscuida en varios campos como: la medicina, análisis de video, detección de objetos, reconstrucción en 3D a partir de una imagen 2D, la industria, tal es el caso de la robótica industrial que facilita al robot guiarse en sitio desconocido. (Garcia, 2013). Actualmente se está trabajando mucho en aplicaciones educativas que se valen de la visión artificial para facilitar y mejorar el aprendizaje de los estudiantes en las aulas de clase (ver figura 7).



Figura 7 Traductor de texto en tiempo real

Fuente: (Heredia, 2016)

1.3.4. Procesamiento digital de imágenes.

El procesamiento digital de imágenes (PDI), sigue normalmente un proceso desde la adquisición de la imagen hasta la toma de decisiones, según la información procesada (ver figura 8).

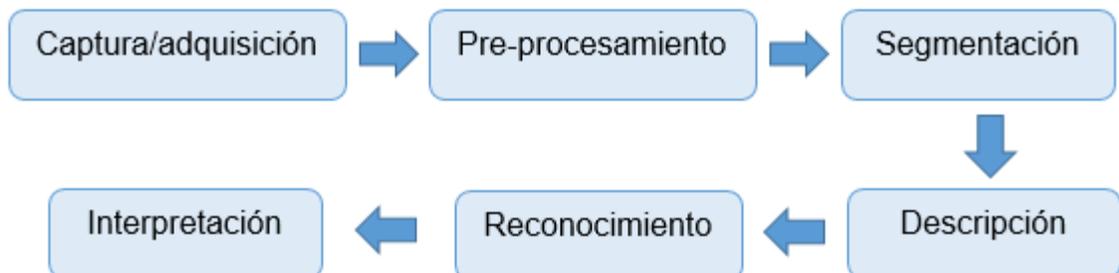


Figura 8 Etapas del procesamiento de imágenes

A continuación, se describe cada etapa del PDI. (Rodríguez R. , 2011)

- **La captura o adquisición.** Es un proceso que tiene como objetivo adquirir una imagen digital mediante la ayuda de una cámara digital, video cámara, etc.
- **El pre-procesamiento.** Proceso que reduce el entorno que no es de interés, incluye técnicas como: reducción del ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.
- **La segmentación.** Es un sistema que fragmenta una imagen en objetos que sean de nuestro interés de estudio.
- **La descripción.** Proceso que obtiene características idóneas y poder diferenciar tipos de elementos como: la forma, el tamaño, área, etc.
- **El reconocimiento.** Es un sistema que reconoce los objetos, por ejemplo: una mano, un rostro, automóvil, etc.
- **La interpretación.** Proceso que asocia un significado a un conjunto de objetos reconocidos y trata de emular la cognición.

1.3.5. Librerías de Visión Artificial.

Las librerías de visión artificial están compuestas de algoritmos enfocados a tareas básicas, que se utilizan para desarrollar aplicaciones en base a la exigencia del caso, estas librerías o bibliotecas contienen códigos y datos que proporcionan acceso a los programas independientes, permitiendo que se compartan y consigan cambiar su estructura modular.

Actualmente existen muchas librerías que se pueden emplear para desarrollar aplicaciones de visión artificial, algunas librerías son de pago, pero al igual existen librerías gratuitas que poseen las mismas características y posibilidad de desarrollar potentes aplicaciones, con la diferencia que su programación es en ocasiones más complicada. En la tabla 1 se muestra un resumen de las más importantes.

Tabla 1

Librerías de Visión Artificial

Librería de visión Artificial	Logo
<p>Torch3vision. Hace uso de C++ para el procesamiento básico de imágenes, algoritmos de extracción como por ejemplo la localización de caras empleando “Haar-like features”, es de uso alcanzable con licencia BDS.</p>	
<p>VLX. Escrita en C++, está compuesta en su la mayoría de los algoritmos habituales en visión artificial, existe la posibilidad de utilizar solamente las librerías de relevancia en razón que no existe dependencias.</p>	
<p>LTI-lib. Hace uso de más de 500 clases, entre las cuales las de mayor interés son: álgebra lineal, clasificación, procesamiento de imágenes y muchas otras características.</p>	
<p>OpenCV. Contiene más de 500 algoritmos entre los cuales se encuentran funciones básicas para el procesamiento de imágenes, descripciones geométricas, segmentación, seguimiento, etc. Permite emplear las destrezas de computación de las GPU, también el uso de las librerías de Intel (Integrated Performance Primitives, IPP) que incorpora una extensa lista de funciones optimizadas.</p>	

Fuente: (Blanco, 2013)

1.3.6. Reconocimientos de Objetos.

Los seres vivos somos capaces de reconocer objetos fácilmente, con cambios de escala, rotación, cambio de punto de vista, luminosidad y oclusiones parciales. El reconocimiento de objetos por lo tanto posee la tarea de buscar y reconocer un objeto en una imagen o cadena de imágenes (video).

a. CARACTERÍSTICAS

Una fase importante para este proceso es la extracción de las características de los objetos, en breve, se exponen dos de los métodos utilizados antes mencionados.

- **Características Haar.** Esta técnica consiste en identificar los rasgos que definen a un objeto en base a los niveles de intensidad que presentan sus píxeles en una imagen, esta información se extrae al aplicar un proceso de funciones llamadas características Haar, en la imagen del elemento. Una característica Haar se designa como una ventana de píxeles, de tamaño y orientación variables, dividida en regiones rectangulares que pueden ser de dos tipos positivas o negativas (ver figura 9). (Alonso V. , 2013)

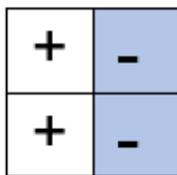


Figura 9 Característica Haar

Fuente: (Alonso V. , 2013)

- **Características LBP (Local Binary Patterns).** Trabajando en una imagen de 3x3 píxeles, tomando como referencia la intensidad del píxel central, esta se va comparando con la intensidad de los píxeles vecinos, si la intensidad del píxel vecino supera o es igual que la del píxel central toma una posición igual a "1", caso contrario llega a tomar un valor de "0", una vez culminada la comparación de todos los píxeles, se obtienen ceros y unos que vienen a representar un número binario. Para encontrar el valor de la característica LBP del píxel central, se debe multiplicar la posición de cada número binario con su correspondiente peso en decimal y sumar todo. (ver figura 10). (Alonso V. , 2013)

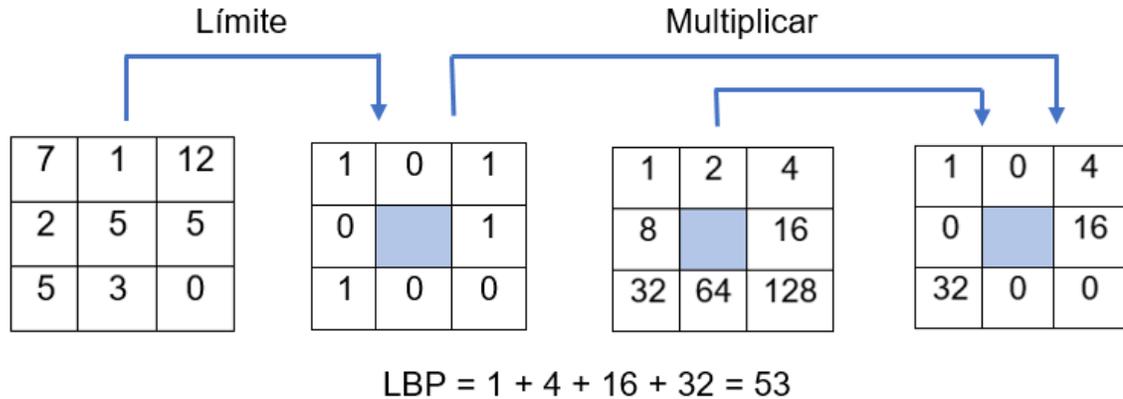


Figura 10 Funcionamiento del Operador LBP

Fuente: (Alonso V. , 2013)

b. CLASIFICADORES

Una vez extraídas las características de los objetos, el siguiente paso es clasificar aquellas que permitan identificar al objeto a detectar. Dentro de los más importantes se resumen.

- **Clasificadores Weak.** Este tipo de clasificador se asemeja al proceso de elegir un valor de umbral, de tal manera, que si el valor de la característica calculada está por encima de ese umbral se considerará que pertenece al objeto a detectar, o en caso contrario, no lo hará. Está definido por una función $h(x)$ que depende de: el valor de una característica f aplicada sobre una región x , un umbral t y una paridad p que representa el sentido de la disparidad (ver ecuación 1). El resultado del clasificador muestra un dato binario que representa si este se encuentra intermedio entre un mínimo valor. (Alonso V. , 2013)

$$h(x) = \begin{cases} 1, & pf(x) > pt \\ 0, & \text{si no} \end{cases} \quad (1)$$

En la figura 11, se ilustra un esquema referente a la ecuación anterior.

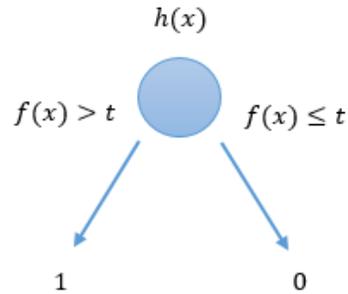


Figura 11 Estructura del Clasificador Weak

Fuente: (Alonso V. , 2013)

- **Clasificadores Strong.** Este clasificador agrupa varios clasificadores weak, reduciendo el error en la clasificación. Se la puede definir mediante las ecuaciones 2 y 3.

$$H(x) = \begin{cases} 1, & pF(x) > pT \\ 0, & \text{si no} \end{cases} \quad (2)$$

$$F(x) = \sum_{i=1}^N w_i h_i(x) \quad (3)$$

Donde: $h_i(x)$ representa a los clasificadores weak, w_i el peso de cada clasificador, N el número de clasificadores, T el umbral y p el sentido de la desigualdad. En la figura 12, se muestra un esquema de este tipo de clasificador.

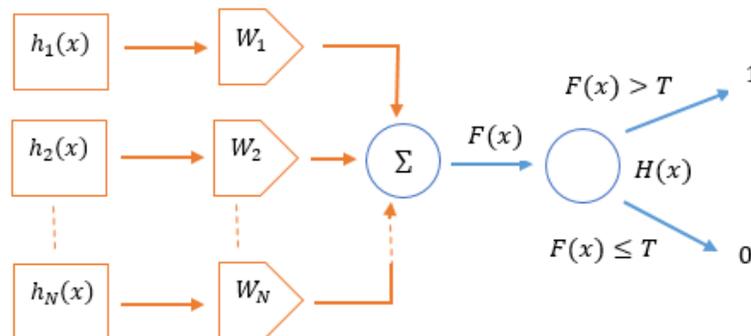


Figura 12 Estructura de un clasificador Strong

Fuente: (Alonso V. , 2013)

- **Clasificadores en Cascada.** Consiste en fusionar clasificadores strong en una estructura en forma de cascada o árbol jerárquico, éstas se encuentran uno detrás de otro formando etapas, es decir cada etapa es más dificultosa que la anterior (ver figura 13).

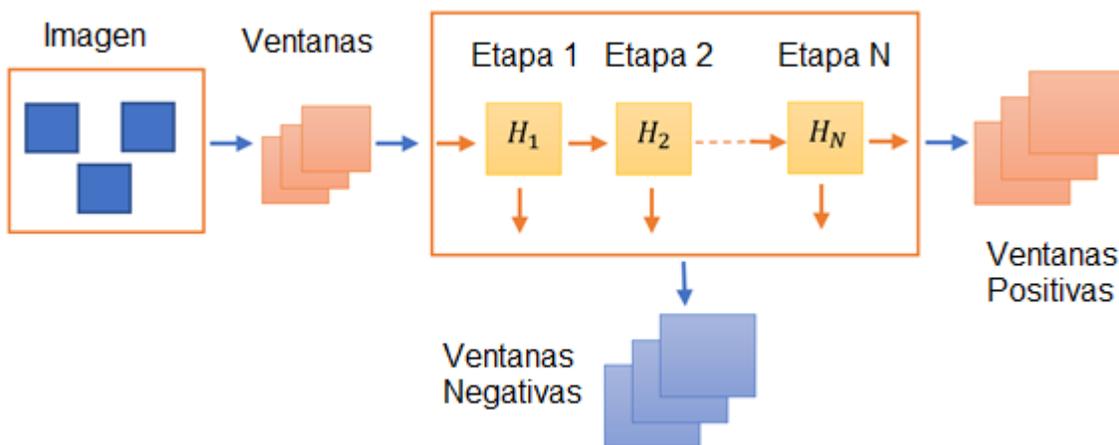


Figura 13 Funcionamiento del clasificador cascada

Fuente: (Alonso V. , 2013)

El clasificador cumple la función de ir comparando las distintas escalas de imagen una por una con la ventana de la imagen principal que posee un tamaño definido en las características del elemento clasificador. Si el resultado posee las particularidades halladas en la ventana que son las del elemento a buscar, entonces esta ventana se clasificará como positiva, caso contrario se clasificará como negativa; y así hasta evaluar todo el conjunto de ventanas de la imagen. (Alonso V. , 2013)

1.3.7. Entrenamiento de los clasificadores

Para que un clasificador se encuentre bien entrenado es necesario que este detecte un objeto en concreto, en el entrenamiento aprende a distinguir las características importantes que van a definir a ese objeto, para esto se hace procesar un conjunto amplio de muestras de dicho objeto (Alonso V. , 2013). Las librerías de visión artificial, poseen algoritmos de aprendizaje que permiten entrenar a los clasificadores.

1.4. Realidad Aumentada

La realidad aumentada consiste en acoplar un entorno real con el virtual, es decir tener una visualización en tiempo real del ambiente físico combinado con elementos virtuales mediante un proceso informático, este tipo de tecnología permite añadir información visual a la realidad (ver figura 14), para mejorar la experiencia visual y calidad de comunicación.



Figura 14 Realidad Aumentada

Fuente: (Matallana, 2011)

1.4.1. Características.

La realidad aumentada tiene las siguientes características: (Morcillo, Fernández, Jiménez, & Sanchez, 2014)

- Combina mundo real y virtual, introduciendo información a las imágenes captadas del entorno.
- Recíproco en tiempo real, es decir se da en ese instante.
- Introduce elementos virtuales en 2D o 3D, sincronizadas con lo que se percibe del entorno.

1.4.2. Diferencia entre realidad aumentada y realidad virtual.



Figura 15 Taxonomía de Realidad Mixta

Fuente: (Morcillo, Fernández, Jiménez, & Sanchez, 2014)

En la realidad virtual, el usuario interactúa en un mundo totalmente virtual, mientras que en la realidad aumentada el usuario visualiza en tiempo real, el entorno combinada con elementos virtuales con los cuales puede interactuar, esta diferencia se muestra gráficamente en la figura 15.

1.4.3. Elementos de la Realidad Aumentada

El sistema de realidad aumentada cuenta con los siguientes componentes (ver figura 16): (Vasquez, 2013)

- **Monitor del computador.** Es el componente donde se reflejará la imagen combinada (real-virtual).
- **Cámara.** Son las webcams de los ordenadores PC, la finalidad de este componente es capturar la información del mundo real.

- **Procesador y Software.** Estos dos elementos conjuntamente tienen la función de interpretar tanto, la información del mundo real que le llega a través de la cámara como, la información que debe añadirse.
- **El marcador.** Por medio de este componente se reproducirán los elementos virtuales en segunda o tercera dimensión.

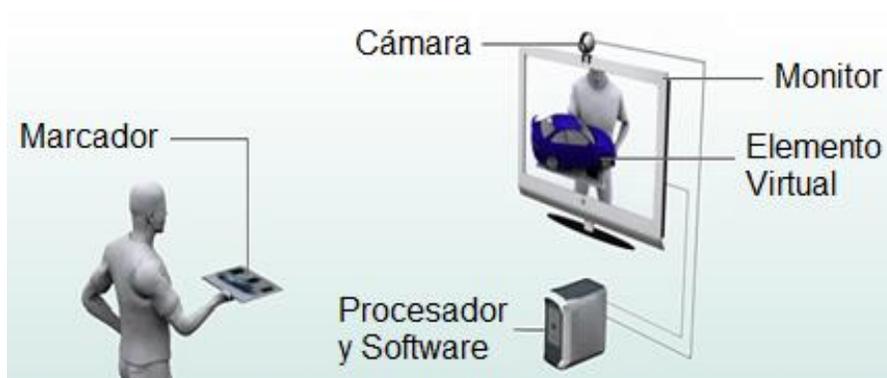


Figura 16 Elementos de la Realidad Aumentada

Fuente: (Suarez, 2015)

1.4.4. Realidad aumentada aplicada en la educación de los estudiantes.

En el sector educativo este tipo de tecnología puede ser utilizado para realizar juegos de entretenimiento y aprendizaje (ver figura 17).



(a) Juego de pensamientos

(b) Juego de actividades diarias

Figura 17 Realidad Aumentada en la Educación

Fuente: (Cadillo, 2016)

Los juegos de RA, pueden incluir actividades de (ver figura 18):

- Clasificación y seriación de objetos.
- Representación de la cardinalidad de un conjunto de objetos.
- Identificación de la relación entre los símbolos, mayor que “>”, menor que “<”, e igual que “=”
- Suma, resta de números.



(a) Juego de Figuras

(b) Juego de Clasificación

Figura 18 Juegos Didácticos con Realidad Aumentada

Fuente: (Cadillo, 2016)

Los niños y niñas pueden aprender matemáticas mediante juegos de números (ver figura 19) donde, además de razonar, interactúan moviendo sus articulaciones haciendo de este juego mucho más entretenido.



Figura 19 Matemáticas con Realidad Aumentada

Fuente: (Cadillo, 2016)

1.4.5. Plataformas para desarrollo de realidad aumentada.

Existen algunas plataformas, entre la cuales las principales son:

- **ARcrowd.** Es una plataforma gratuita que no requiere de instalación de software ya que funciona mediante una aplicación online que permite crear y ver contenido de RA (ver figura 20). (Educación 3.0, 2016)



Figura 20 ARcrowd Planetas

Fuente: (Hernández, 2014)

- **LearnAr.** Es una herramienta didáctica que emplea una cámara web, el paquete contiene actividades de: matemáticas, ciencias, anatomía (Ver figura 21), física, geometría, educación física e idiomas. (Alonso M. , 2013)

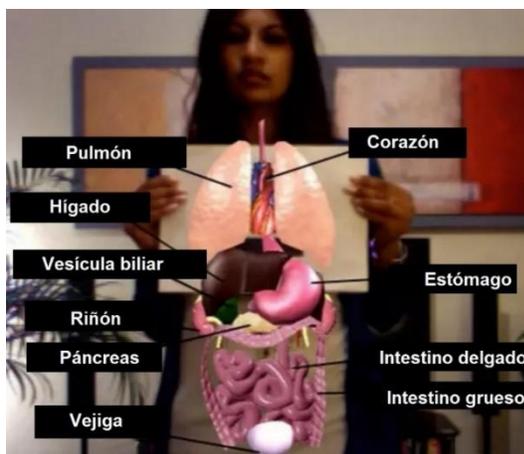


Figura 21 LearnAR aparato digestivo

Fuente: (Alonso M. , 2013)

- **Total Immersion.** Es una de las plataformas de RA más dinámicas del mercado ya que permite la visualización de muebles virtuales, venta de ropa de mujeres con tecnología RA (ver figura 22). (Mocholí, 2014)



Figura 22 Probadores virtuales

Fuente: (Urrutia, 2013)

- **Pygame.** Es un programa multiplataforma de módulos de lenguaje Python (código libre) que se especializa en la creación de videojuegos en dos dimensiones, permitiendo la creación de programas multimedia o interfaces de usuario. (Keerthi, 2015). Este tipo de plataforma, al poder emplear librerías de visión artificial debido a que se programa en Python, es una buena alternativa de bajo presupuesto para crear juegos que incorporen RA (ver figura 23).



Figura 23 Juegos RA con Pygame

Fuente: (Rodríguez J. , 2014)

1.5. Robots interactivos aplicados a la educación.

La Robótica Educativa, es la disciplina que busca potencializar el uso de robots o kits de robótica en las aulas de clase, esto permite al estudiante desde temprana edad conocer el funcionamiento, características principales de los mecanismos y electrónica de los cuales está compuesto el robot, logrando así la organización del pensamiento por intermedio del desarrollo lógico y programativo de robots. (Barrientos, 2013). Razón por la cual muchos países optan por utilizar este tipo de tecnología en las instituciones educativas.

1.5.1. Robots Actualmente implementados en la educación.

- **Teebot.** Es el primer robot ecuatoriano, compuesto por un kit de robótica adicional al robot, que permite tanto niños como niñas aprender conceptos de programación, electrónica, tecnología y ciencia, mediante el trabajo en equipo y el uso de su creatividad (ver figura 24). (Mosquera, 2015)



Figura 24 Teebot robot educativo

Fuente: (Mosquera, 2015)

- **NAO.** Es uno de los humanoides más avanzados del mundo, este robot está especialmente indicado para la enseñanza (ver figura 25), ya que por su gran versatilidad en la forma de programarlo se lo considera como una herramienta pedagógica idónea tanto en educación secundaria como en universidades. (Nuñez, 2015)



Figura 25 Nao educación

Fuente: (Rivera, 2015)

- **Zowi.** Compuesto por un Arduino como tarjeta principal, además es totalmente programable con un lenguaje de bloques tipo Scratch, las pruebas realizadas en varios establecimientos educativos (ver figura 26), arrojan resultados magníficos ya que los alumnos se sienten atraídos por la robótica. (Sánchez, 2015)



Figura 26 Interacción con Zowi

Fuente: (Sánchez, 2015)

- **Rapiro.** Es de tipo humanoide, emplea una Raspberry Pi como tarjeta de control, una cámara y doce servos los cuales permiten tener movimientos programados muy precisos. Su diseño es agradable y atractivo principalmente para los estudiantes más pequeños (ver figura 27).



Figura 27 Rapiro Robot

Fuente: (Adafruit, 2014)

1.5.2. Beneficios de la robótica en la educación de los niños y niñas.

Entre los muchos beneficios en la educación para los niños y niñas, están:

- Se involucran activamente en su propio proceso de aprendizaje, desarrollando la intuición científica y de ingeniería (ver figura 28)
- Potencian sus habilidades de investigación, resolución de problemas, centrándose en el razonamiento lógico, analítico, y pensamiento articulista.
- Transformarse en un autodidacta activo, siendo capaz de adquirir conocimiento por si solo.
- Capacita a los niños y niñas para desenvolverse en el mundo competitivo, especialmente en las ciencias tecnológicas, que son las de mayor demanda en la actualidad. (Mendez, 2015)



Figura 28 Robot en la educación

Fuente: (Robotrónica, 2015)

1.6. Sistemas de comunicación para transmisión de datos.

La comunicación inalámbrica utiliza la modulación de ondas de baja potencia como medio o canal para la transmisión de datos entre el emisor y el receptor, a continuación, se menciona algunas tecnologías de este tipo.

- **Bluetooth.** Se basa en el modo de operación maestro/esclavo (ver figura 29), dispone de una transmisión de datos de manera inalámbrica con una frecuencia de 2,4 Ghz, siendo la más usada para la sincronización de datos punto a punto, con diferentes dispositivos como celulares, tablets, etc. (Olea, 2012)

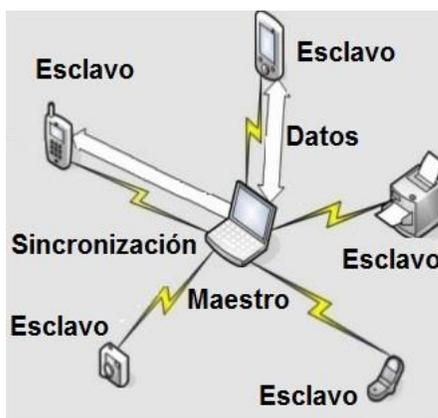


Figura 29 Red Bluetooth

Fuente: (Olea, 2012)

- **Wifi.** La transmisión de datos es diez veces más rápida que el Bluetooth y tiene un alcance de 100 metros en espacios totalmente cerrados, permite manejar grandes cantidades de datos hasta a una velocidad de transmisión de 600 Mbps. En la actualidad casi todos los dispositivos nuevos, como teléfonos inteligentes, tablets, pc (ver figura 30) utilizan esta tecnología. (Olea, 2012)



Figura 30 Red Wifi

Fuente: (Softigal, 2014)

- **ZigBee.** Es una tecnología inalámbrica que opera a 2,4 GHZ en aplicaciones industriales y domótica que requieren bajo envío de datos, pero a la vez necesitan máxima seguridad y robustez, posee un alcance de 100 metros (ver figura 31). (RS Components, 2016)



Figura 31 Red ZigBee

Fuente: (RS Components, 2016)

En el siguiente capítulo, se establece las necesidades del usuario, las especificaciones técnicas de los elementos de cada sistema, se realiza el análisis y selección de las alternativas de los componentes, que se requieren para el desarrollo e implementación del proyecto.

CAPÍTULO II

ANÁLISIS Y SELECCIÓN DE ALTERNATIVAS

2.1. Introducción

El diseño del sistema general se rige a un proceso de diseño concurrente, con la finalidad de integrar simultáneamente todos los aspectos que intervienen para el diseño de los sistemas, partiendo de la planeación y necesidades que se identifican en las primeras etapas. En la figura 32, se muestra la secuencia de diseño para todos los sub-sistemas que componen el proyecto.

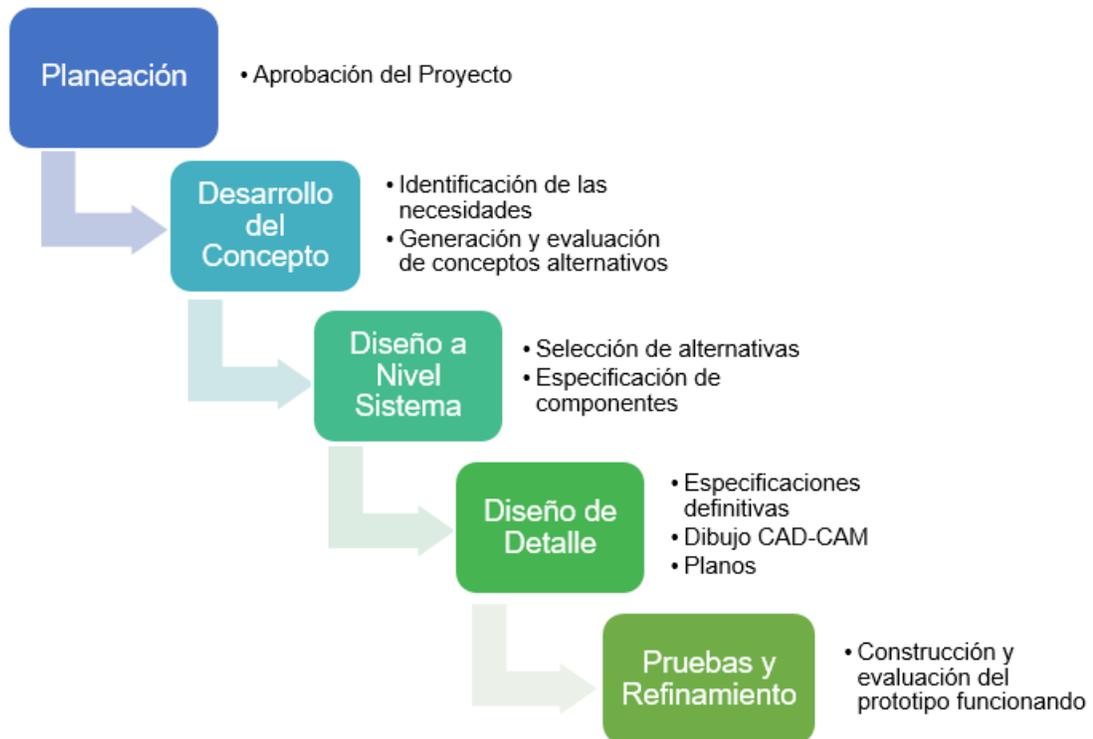


Figura 32 Proceso de Diseño

Fuente: (Ulrich & Eppinger, 2013)

El trabajo simultáneo en las etapas de diseño, exige mayor tiempo y por lo tanto mayor conceptualización detallada de los sistemas.

2.2. Sistemas y componentes del proyecto

Este proyecto tiene como meta el diseño y la ejecución de un sistema de Realidad Aumentada con Asistente Robótico mediante visión artificial y comunicación inalámbrica, que sirva como herramienta didáctica de apoyo para el mejoramiento del aprendizaje de los niños y niñas de educación básica.

El proyecto se compone de tres sistemas (ver figura 33):

- Sistema de realidad aumentada (RA)
- Sistema de transmisión de datos
- El robot asistente

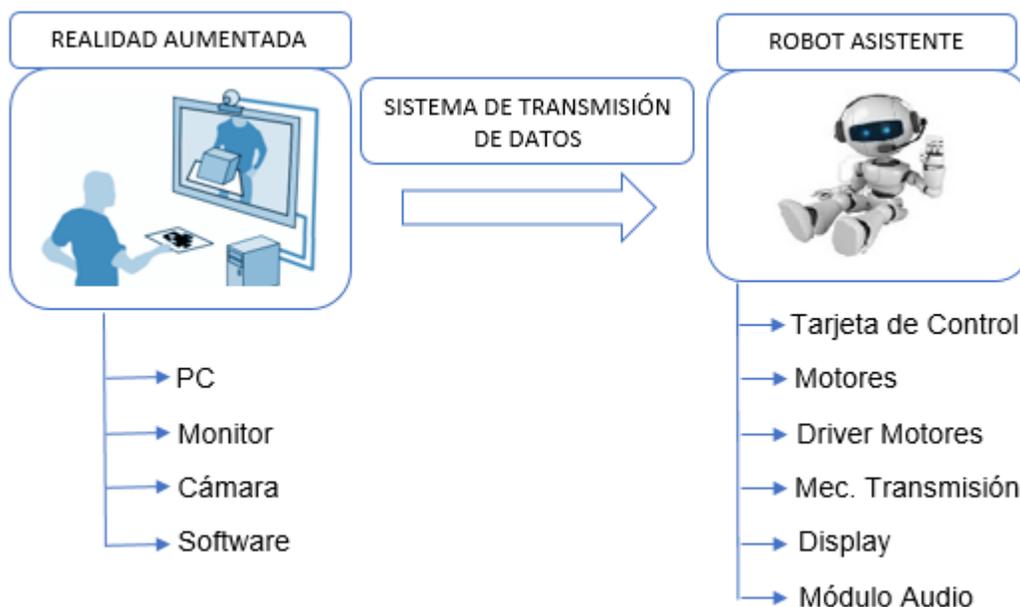


Figura 33 Sistemas y componentes

El sistema RA brinda una visión en tiempo real del entorno físico combinada con elementos virtuales, en este caso son juegos educativos. El sistema de transmisión de datos cumple con la función de establecer la conexión entre el procesador y el robot asistente, permitiendo la transmisión de información. El robot asistente es un robot de aspecto humanoide, tiene dos funciones la primera es recibir la información y reproducirlas mediante voz

brindando instrucciones del juego, la segunda es ser comandado a distancia mediante la interfaz de Realidad Aumentada.

2.3. Necesidades y especificaciones técnicas

Para establecer los requerimientos del proyecto se realiza una recopilación de datos de las necesidades de los usuarios, para lo cual se realiza encuestas a docentes de educación básica, en el anexo A se muestra el formato de encuesta utilizada y los resultados. Las prioridades se enlistan en forma jerárquica.

- El costo de implementación sea cómoda
- Sea estética
- Evite el cansancio de ojos, se puede configurar brillo y otros parámetros
- Las Imágenes sean nítidas
- Sea de fácil manejo
- Se pueda instalar en varias computadoras
- Las conexiones eléctricas sean seguras
- Las imágenes sean fluidas
- El robot realice varios movimientos
- El robot simule gestos
- Buena calidad de audio
- Sea fácil de transportar
- La batería sea duradera
- Mantiene fluidez de imágenes, respuesta es rápida
- Buena comunicación inalámbrica

En base a las necesidades de los usuarios se elabora la casa de la calidad (ver anexo B), con la finalidad de conocer como el proyecto satisface los requerimientos de los beneficiarios.

2.3.1. Especificaciones técnicas

De acuerdo con las necesidades establecidas y el análisis de las métricas de la casa de la calidad, se logra obtener las especificaciones técnicas de los componentes (ver tabla 2).

Tabla 2
Especificaciones técnicas del proyecto

Núm.	Métrica	Unidad	Valor
1	Velocidad del procesador	GHz	> 1,7
2	Memoria RAM	GB	> 2
3	Peso PC	Kg	< 2
4	Almacenamiento interno	GB	> 32
5	Resolución de la cámara	pixeles	1280x720 1920x1080
6	Campo de visión horizontal y vertical de la cámara	grados	> 40
7	Licencia del software	Libre GLP	
8	Tamaño de Monitor	pulgadas	5 - 32
9	Altura del Robot	cm	20 - 50
10	Número de E/S de la tarjeta de control		> 20
11	Voltaje de trabajo de motores	V	3 - 12
12	Canales del controlador de motores		3 - 6
13	Tamaño de matriz de leds	m x n	5 x 7 8 x 8 8 x 16
14	Formato de Audio	MP3 WAV WMA	
15	Alcance inalámbrico	m	> 2
16	Velocidad de transmisión de datos	bps	1000

2.3.2. Método de evaluación

Para la implementación del proyecto se realiza un análisis y selección de los componentes que se requiere en cada uno de los sistemas, la selección de la mejor alternativa se realiza acorde a un proceso de calificación de parámetros mediante escalas de ponderación de acuerdo al nivel de satisfacción del componente (ver tabla 3).

Tabla 3

Escala de calificación

Calificación	Porcentaje (%)	Desempeño Relativo
1	< 20	Pésimo
2	20 - 40	Peor
3	40 - 60	Igual
4	60 - 80	Mejor
5	80 -100	Óptimo

2.4. Alternativas de Realidad Aumentada

El sistema de RA cumple la función de crear una visualización de realidad mixta, entre el entorno real y su combinación con imágenes o elementos virtuales. Para el sistema RA según la configuración de su estructura se analizan dos alternativas.

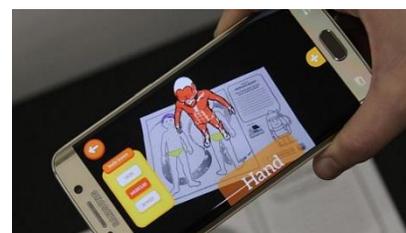
- Sistema RA Fijo
- Sistema RA Móvil

En la tabla 4 se realiza una comparación entre los sistemas de realidad aumentada fija y móvil con la finalidad de describir las principales características del sistema RA.

Tabla 4
Comparación entre Sistema RA Fijo y Móvil



Sistema RA Fijo



Sistema RA Móvil

Espacio Requerido	Se requiere un espacio grande para su instalación.	Compacto, tamaño cómodo de fácil transporte.
Modularidad	Sistema modular, cuando alguno de sus componentes sufre algún daño, se puede cambiar de manera fácil y cómoda.	Sistema no modular, si un componente sufre algún daño, se requiere del cambio de todo el sistema.
Capacidad de Usuarios	Múltiples usuarios, proceso de visualización enfocado a varios usuarios a la vez.	Se limita el uso a uno o máximo dos usuarios a la vez.
Escala de Visualización	Visualización del entorno a escala real, los elementos son visualizados a tamaño casi real.	Visualización del entorno a escala reducida.
Consumo de energía	Alto consumo de energía eléctrica.	Bajo consumo de energía eléctrica.

Selección del tipo de Sistema de Realidad Aumentada.

La selección del Sistema RA adecuado, se realiza según los siguientes requerimientos:

- Capacidad de usuarios
- Escala de visualización
- Modularidad
- Costo de implementación
- Espacio para instalación

En la tabla 5, se realiza una evaluación de las alternativas de Sistemas RA en base a los requerimientos establecidos.

Tabla 5
Evaluación de alternativas del sistema RA

Criterios de selección	Peso	Alternativas			
		A		B	
		Sistema RA Fijo		Sistema RA Móvil	
		Calificación	Evaluación Ponderada	Calificación	Evaluación Ponderada
Capacidad de Usuarios	30%	4	1.2	2	0.6
Escala de visualización	25%	4	1.0	3	0.75
Modularidad	25%	4	1.0	2	0.5
Costo de implementación	15%	3	0.45	2	0.3
Espacio para instalación	5%	3	0.15	5	0.25
Total	100%		3.8		2.4
	Lugar		1		2
	¿Continuar?		Desarrollar		No

La alternativa A (Sistema RA Fijo), es la opción más óptima de acuerdo con la tabla 5, por ser un sistema que permite ser utilizado por varios usuarios a la vez, con una escala de visualización acorde a lo real, además por su modularidad que facilita la extracción y cambio de cualquiera de sus componentes si existiera algún desperfecto o si se desea mejorar el sistema.

2.4.1. PC / Procesador

El procesador conjuntamente con el software debe interpretar la información que llega a través de la cámara, por lo cual debe poseer características técnicas que satisfagan los requerimientos mínimos para el desarrollo de este tipo de aplicaciones, se analizan tres alternativas de mini pc más comerciales.

- LattePanda
- MintBox
- Gigabyte Intel

La tabla 6 muestra una comparación de características entre las alternativas de PC / Procesador.

Tabla 6
Comparación de alternativas de PC/Procesador

			
	LattePanda	MintBox	Gigabyte Intel
Descripción	Mini Ordenador con Arduino	Mini PC con sistema operativo LinuxMint	PC ultra compacto, versátil y potente
Procesador	1,8 GHz Intel Cherry Trail Z8300 Quad Core	AMD A Series	Intel Core i5-4200U 2,6 GHz
Coprocesador	ATmega32u4	Procesador Grafico Radeon	NO
Memoria RAM	2 / 4GB DDR3L	4 GB SO-DIMM DDR3	Max. 16 GB
Compatibilidad	Windows 10 y Ubuntu	Linux MInt	Windows 10
Almacenamiento	32 / 64GB	64 GB	Max 128 GB
Inalámbrico	Wifi y Bluetooth 4.0	Wifi WLAN	Wifi – pre instalado
Salida de Video	HDMI	2 x HDMI	HDMI, micro HDMI
Salida de Audio	Jack audio 3,5 mm	Jack audio 3,5 mm	Jack audio 3,5 mm
Puertos USB	1 x USB 3.0, 2 x USB 2.0	3x USB 2.0, 2 x USB 3.0	4 x USB 3.0
Pines de control	2 GPIO de chips Intel, 20 GPIO para Arduino	NO	NO
Tarjeta gráfica	NO	NO	Intel HD 4400
Dimensión (largo, ancho, alto)	8,8 x 7 x 1,5 cm	10,6 x 8,4 x 2,3 cm	11,4 x 10,7 x 2,9 cm
Voltaje	5V / 2V	12V	100 – 240 V AC

Selección del PC/Procesador.

La selección de la mini PC conjuntamente con el procesador, se realiza según los siguientes parámetros:

- Procesador
- Memoria RAM
- Capacidad de almacenamiento
- Salida de Video
- Tarjeta Gráfica
- Puertos USB
- Voltaje
- Tamaño

En la tabla 7, se realiza una evaluación de las alternativas de Mini PC en base a los parámetros establecidos.

Tabla 7

Evaluación de alternativas del PC/Procesador

Criterios	Peso	Alternativas					
		A LattePanda		B MintBox		C Gigabyte Intel	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Procesador	20%	4	0.8	3	0.6	5	1.0
Memoria RAM	20%	3	0.6	2	0.4	4	0.8
Tarjeta Gráfica	15%	2	0.3	2	0.3	5	0.75
Almacenamiento	10%	5	0.5	5	0.5	5	0.5
Salida de Video	10%	3	0.3	4	0.4	5	0.5
Voltaje	10%	5	0.5	4	0.4	3	0.3
Costo	5%	4	0.4	3	0.3	3	0.3
Puertos USB	5%	3	0.15	5	0.25	4	0.2
Tamaño	5%	5	0.25	4	0.2	3	0.15
Total	100%		3.8		3.35		4.5
	Lugar						
¿Continuar?		No		No		Desarrollar	

De acuerdo con la tabla 7, la alternativa C (Gigabyte Intel), es la mini PC más conveniente debido su rapidez del procesador, memoria RAM aceptable y una tarjeta gráfica Intel 4400, por lo cual permite un óptimo procesamiento de la información (imágenes) del entorno captada a través de la cámara.

2.4.2. Monitor

El monitor permite la visualización de la realidad mixta entre las imágenes del entorno y los elementos virtuales, para lo cual se analizan dos alternativas de monitores.

- Monitor LCD
- Monitor LED

Los monitores LED y LCD son equipos tecnológicos que facilitan la visualización de imágenes, aunque la diferencia radica en el tipo de iluminación, los monitores LCD están compuestos de lámparas fluorescentes de cátodo frío para la iluminación de fondo, a diferencia los monitores LED usan diodos emisores de luz, siendo esta la principal diferencia entre los dos componentes. (Informatica, 2016). En la tabla 8, se efectúa un análisis entre estos dos tipos de tecnología para visualización.

Tabla 8
Comparación entre monitor LCD y LED

LCD	LED
	
Más gruesas y pesadas que las LED	Más delgadas y livianas que las LCD
Gasto de energía mayor	Gasto de hasta un 40% menos de energía que un monitor LCD
La calidad de imagen es buena, y aunque su gama de colores no es tan amplia, se ven brillantes.	La calidad de imágenes y color es mejor que la LCD, pues su gama de colores es un 46% más amplia.
Más gruesas y pesadas que las LED	Más delgadas y livianas que las LCD

CONTINÚA 

Emiten más calor	Emiten menos calor
No son tan “ecológicas”, ya que están hechas con un material (mercurio) por el cual se produce la imagen.	Son ecológicas, por la ausencia de mercurio en su composición.
Eventualmente, se pueden “congelar” las imágenes.	No se “congelan” las imágenes
El negro no se ve tan negro, sino como gris oscuro.	El negro se ve realmente negro
Son más baratos que un LED del mismo tamaño.	Son más caros comparando con el LCD.
Deben tener instalada una lámpara trasera para que la imagen se pueda ver más clara.	Contiene diodos emisores de luz instalados de manera interna.
Menor vida útil de la pantalla	Mayor vida útil de la pantalla

Selección del Monitor.

Mediante la comparación de la tabla 7, la mejor opción es el monitor LED, ya que es una tecnología mejorada del LCD, cumple a cabalidad los requerimientos actuales sobre cuidado del medio ambiente, menor consumo eléctrico, mejor calidad de imagen y mayor vida útil.

2.4.3. Cámara

La cámara tiene la función de obtener información real del entorno mediante captura de imágenes o fotografías, de este dispositivo depende la extensión de métodos y algoritmos de pre-procesamiento en el PDI (Procesamiento Digital de Imágenes) puesto que, si la cámara obtiene imágenes de baja calidad, requiere de bastantes métodos para la eliminación de ruido y otros factores que alteran la calidad de la imagen, caso contrario se requiere de pocos algoritmos de pre-procesamiento. Por lo tanto, la cámara cumple un rol importante en el sistema RA, para ello se analizan dos alternativas de cámaras, siendo estos los más comunes en el mercado (ver tabla 9).

- KinectCam
- Logitech C525 Webcam

Tabla 9
Comparación entre alternativas de la cámara

	 KinectCam	 Logitech C525 Webcam
Descripción	Dispositivo capaz de capturar imágenes del esqueleto humano, reconocerlo y posicionarlo en el plano.	Pequeña cámara digital la cual puede capturar y transmitir las imágenes en vivo.
Resolución	a) 640 x 480 pixeles a 30 FPS b) 1280 x 1024 pixeles a 15 FPS.	Resolución HD 720p (1280 x 720 pixeles)
Enfoque	Utiliza un emisor y un sensor infrarrojos para enfocar	Enfoque automático
Megapíxeles	NA	Hasta 8 megapíxeles
Campo de Visión	Horizontal = 57° Vertical = 43° Inclinación = $\pm 27^\circ$	Horizontal = 360° Vertical = 180°
Compatibilidad	Xbox, Windows	Cualquier Sistema Operativo
Consumo de energía	Alto consumo de energía	Bajo consumo de energía
Requisitos Mínimos	Procesador: Dual Core RAM: 2GB	Procesador: Core 2 Duo RAM: 2GB
Instalación	Requiere librerías adicionales SDK.	Plug and play, no requiere de software para utilizarlo.

Selección de la cámara.

La selección de la cámara se realiza de acuerdo a los siguientes parámetros:

- Resolución
- Enfoque
- Instalación
- Diseño
- Costo

En la tabla 10, se realiza una evaluación de las opciones de cámaras en referencia a los requerimientos definidos.

Tabla 10
Evaluación de alternativas para la cámara

		Alternativas			
Criterios	Peso	A KinectCam		B Logitech C525 WebCam	
		Calificación	Evaluación Ponderada	Calificación	Evaluación Ponderada
Resolución	30%	4	1.2	4	1.2
Enfoque	25%	3	0.75	4	1.0
Instalación	20%	3	0.6	5	1.0
Diseño	15%	4	0.6	5	0.75
Costo	10%	2	0.2	3	0.3
Total	100%		3.35		4.25
	Lugar		2		1
¿Continuar?		No		Desarrollar	

La alternativa B (Logitech C525 Webcam), es la opción más recomendada de acuerdo con la tabla 10, por ser una cámara que tiene una buena relación calidad-precio, captura imágenes de buena calidad, fácil instalación y su diseño compacto que permite adherirlo a casi cualquier superficie.

2.4.4. Software

Para la creación de los juegos, se analizan tres plataformas de software para desarrollo de Realidad Aumentada (ver tabla 11).

- ArtoolKit
- Pygame + Opencv
- Unity3D + Vuforia

Tabla 11

Comparación de alternativas de Software

Software	Características
<p data-bbox="418 359 548 386">ArtoolKit</p> 	<ul data-bbox="732 359 1425 667" style="list-style-type: none"> • Permite desarrollar aplicaciones de RA en tiempo real, mediante la inserción de un dispositivo fotográfico. • Utiliza rectángulos, cuadrados negros o códigos QR llamados marcadores • Permite la posibilidad de usar distintos patrones en los marcadores • Compatibilidad con distintas plataformas
<p data-bbox="342 789 626 825">Pygame + OpenCV</p> 	<ul data-bbox="732 726 1425 1272" style="list-style-type: none"> • Permite procesar imágenes de video en tiempo real y utilizar sus características como datos o marcadores para el desarrollo de juegos. • Lenguaje de programación sencilla y de alto nivel • Se puede utilizar cualquier objeto como marcador, debido a su capacidad de reconocimiento de objetos. • Soporte solo para 2D, para integrar elementos en 3D requiere de herramientas adicionales como OpenGL, ARUCO. • Compatibles con sistemas Unix y Windows • Bajo costo computacional • Fácil instalación • Es de licencia libre
<p data-bbox="350 1346 618 1381">Unity3D + Vuforia</p> 	<ul data-bbox="732 1335 1425 1755" style="list-style-type: none"> • Interfaz de fácil uso • Múltiples plataformas móviles y PC • Permite realizar juegos y aplicaciones con elementos virtuales en 3D • Lenguaje de programación Javascript y C# • Gracias a la fácil interfaz se reduce el tiempo de desarrollo de los juegos • Alto costo computacional • No es de licencia libre • Para poder utilizar la cámara, se debe utilizar la versión de pago

Selección del software.

Las alternativas del software se analizan de acuerdo a los siguientes criterios:

- Interfaz gráfica
- Reconocimiento de objetos (R. de objetos)
- Coste computacional (C. computacional)
- Programación
- Instalación
- Licencia
- Multiplataforma

En la tabla 12, se realiza una evaluación de softwares para el desarrollo las interfaces de realidad aumentada, en base a los requerimientos presentados.

Tabla 12

Evaluación de alternativas del software

Criterios	Peso	Alternativas					
		A		B		C	
		ArtoolKit		Pygame+OpenCV		Unity3D+Vuforia	
		Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
Interfaz gráfica	15%	3	0.45	4	0.6	5	0.75
R. de objetos	20%	4	0.8	5	1.0	3	0.6
C. computacional	20%	4	0.8	4	0.8	2	0.4
Programación	10%	3	0.3	3	0.3	4	0.4
Instalación	10%	3	0.3	4	0.4	4	0.4
Licencia	15%	5	0.75	5	0.75	2	0.3
Multiplataforma	10%	5	0.5	5	0.5	3	0.3
Total	100%		3.9		4.35		3.15
	Lugar		2		1		3
¿Continuar?		No		Desarrollar		No	

La alternativa B (Pygame + OpenCV), es la opción más recomendada de acuerdo con la tabla 12, por ser un conjunto de software libre, orientado al desarrollo de juegos enfocados a la realidad aumentada, da la opción de usar objetos de cualquier característica como marcadores debido a la posibilidad de reconocer objetos, además de su instalación y programación fácil.

2.5. Robot Interactivo

El Robot didáctico cumple la función de asistente, entre la Realidad Aumentada y el usuario, proporcionando instrucciones del juego mediando audio, se empieza seleccionando el aspecto que tendrá el robot asistente.

2.5.1. Aspecto del Robot.

Puesto a que el robot está destinado para ser implementado en un ambiente rodeado de niños y niñas, el aspecto del robot juega un papel importante, se analiza tres tipos de robot según su arquitectura.

- Móviles
- Humanoides
- Zoomórficos

La tabla 13, muestra una comparación de características de cada tipo de robot.

Tabla 13

Comparación de robots según su arquitectura

		
Móviles	Humanoides	Zoomórficos
Gran capacidad de desplazamiento	Cinematía y forma parecida al ser humano	Cinematía y forma parecida al de los animales
Se utilizan para transportes de carga, generalmente se emplean en casi cualquier tipo de terreno.	Se emplea en el campo educativo para la interaprendizaje de los discentes.	Se utiliza como medios de entretenimiento para estudiantes, simulando el comportamiento de tiernas mascotas.
Las dimensiones dependen del tipo de aplicación la que este destinada	Son robots pequeños de menos de un metro de altura	Robots de pequeños, por lo general no mayor a 50 [cm] de alto.

Selección del robot según su arquitectura.

Una vez analizado las principales características se opta por diseñar un robot de arquitectura humana, ya que estos se caracterizan por interactuar y a la vez comunicarse con los discentes estableciendo normas con características sociales. (Nuñez, 2015)

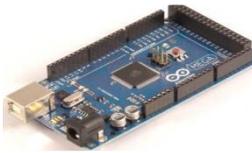
2.5.2. Tarjeta de control

Esta tarjeta tiene la función de controlar mediante programación los dispositivos digitales y analógicos como sensores, motores respectivamente, se analizan tres alternativas de tarjetas comerciales (ver tabla 14).

- Raspberry Pi 3
- Arduino Mega
- ATmega8L

Tabla 14

Comparación de tarjetas de control

			
	Raspberry Pi3	Arduino Mega	ATmega 8L
Descripción	Pequeño ordenador con pines para control	Plataforma física computacional	Circuito integrado de alto rendimiento
Costo	\$35	\$24	\$3
Tamaño	8,6 x 5,4 x 1,7 cm	108 x 53 x 15 cm	34,7 x 7,4 x 4,06 cm
Controlador	ARMv8	ATMEGA 2560	8 bits
Pines	40 GPIO	54 digitales E/S 15 salidas PWM	23 digitales E/S
Memoria	512 MB	256 KB	8KB
Voltaje	5 V	7-12V	4,5-5 V
Puertos	USB: Cuatro MicroUSB: Uno Ethernet: Uno	Serial: Uno	No
Peso	98 gramos	37 gramos	3 gramos
Programación	Scratch, IDLE, Linux	Arduino	Código C, AVR

Selección de la tarjeta de control

Para seleccionar la mejor opción se hace referencia a los siguientes criterios:

- Número de E/S digitales
- Controlador
- Memoria
- Dimensión
- Compatibilidad
- Programación
- Costo
- Peso

La tabla 15, muestra la evaluación realizada acorde a los criterios establecidos.

Tabla 15

Evaluación de alternativas de la tarjeta de control

Criterios	Peso	Alternativas					
		A		B		C	
		Raspberry Pi3		Arduino Mega		ATmega 8L	
		Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
E/S digitales	25%	4	1	5	1.25	2	0.5
Controlador	20%	4	0.8	4	0.8	3	0.6
Memoria	15%	5	0.75	4	0.6	1	0.15
Dimensión	10%	4	0.4	3	0.3	5	0.5
Compatibilidad	10%	3	0.3	5	0.5	1	0.1
Programación	10%	4	0.4	5	0.5	5	0.5
Costo	5%	2	0.1	4	0.2	5	0.25
Peso	5%	4	0.2	3	0.15	5	0.25
Total	100%		3.95		4.3		2.85
	Lugar		2		1		3
¿Continuar?			No		Desarrollar		No

De la tabla 15, resulta que la Alternativa B (Arduino Mega) es el componente idóneo, no solo por el número de pines E/S digitales requeridos sino también por la adaptabilidad con la mayoría de Shields de Arduino y su fácil programación.

2.5.3. Motores

Los motores cumplen la función de convertir la energía eléctrica en mecánica, provocando un movimiento rotatorio, que cause movimientos de desplazamiento en las articulaciones mecánicas del robot, se analizan tres alternativas de motores comerciales (ver tabla 16).

- Motores DC
- Micro servo
- Motor paso a paso

Tabla 16
Comparación de los motores

			
	Motor DC	Micro servo	Motor a pasos
Descripción	Convierte energía eléctrica en mecánica provocando movimiento rotatorio	Capacidad de ubicarse en cualquier posición dentro de su rango de operación	Convierte impulsos eléctricos en desplazamientos angulares discretos
Costo	\$2	\$6	\$7
Tamaño	20 x 15 cm	22,2 x 11,8 cm	35 x 15 cm
Voltaje	1,25 - 6V	4,7 - 2V	9 - 12V
Rotación	360 grados	180 grados	7,5 grados (paso)
Torque	bajo	1,8 Kg-cm	100 g / cm
Compatibilidad	Arduino	MCU - Arduino	Arduino
Pines	2	4	5
Velocidad	6800 RPM	0,02 RPM a 4,8 V	48 pasos (vuelta)
Peso	34 gramos	9 gramos	61 gramos
Eje de Transmisión	2 mm (Φ)	2 mm (Φ)	3 mm (Φ)

Selección de motores

Para seleccionar la mejor opción del motor se realiza una evaluación (tabla 17) en referencia a los siguientes criterios:

- Fácil control, precisión
- Velocidad, torque
- Ruido, dimensión
- Peso, costo

Tabla 17

Evaluación de alternativas de los motores

Criterios	Peso	Alternativas					
		A		B		C	
		Motor DC		Micro servo Sg90		Motor a pasos	
		Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
Fácil control	20%	5	1	5	1	5	1
Ruido	20%	5	1	3	0.6	2	0.4
Precisión	15%	4	0.6	4	0.6	4	0.6
Velocidad	10%	5	0.5	4	0.4	4	0.6
Torque	10%	2	0.2	4	0.4	4	0.4
Dimensión	10%	5	0.5	5	0.5	4	0.4
Costo	10%	5	0.5	3	0.3	3	0.3
Peso	5%	3	0.15	4	0.2	4	0.2
Total	100%		4.45		4		3.9
	Lugar		1		2		3
¿Continuar?		Desarrollar		No		No	

De la tabla 17, la Alternativa A (Motores DC) es el componente idóneo, no solo por el fácil control de velocidad y sentido de giro, sino también por las dimensiones del motor, por ser un motor silencioso, económico y liviano. Para solucionar el inconveniente de la alta velocidad y bajo torque, se puede adaptar cualquier sistema de mecanismos, por ejemplo, una caja de engranes que permite reducir la velocidad y aumentar el torque, por lo cual son idóneos para robots didácticos pequeños no mayores a 2 kilogramos de peso. (Bravo, 2014)

2.5.4. Módulo de control de motores

Este módulo se encarga de controlar la posición y sentido de giro de los motores para realizar los movimientos del robot, se analizan tres alternativas de módulos de control de motores comerciales (ver tabla 18).

- Módulo Shield con L293D
- Módulo L298N
- Módulo controlador A4988

Tabla 18

Comparación de la tarjeta de control de motores

			
	Módulo Shield	Módulo L298N	Módulo A4988
Controla	- Motores DC - Selenoides - Relés - Motor paso a paso	- Motores DC - Motores paso a paso bipolar	- Motor paso a paso bipolar
Costo	\$13	\$6	\$8
Dimensión	6,8 x 5,3 x 2 cm	4,3 x 2,39 x 4,3 cm	2,3 x 1,52 cm
Controlador	L293D	L298N	A4988
Voltaje	5V	5V	3,3 - 5V
PWM	Si	Si	No
Canales	4	2	1
Compatibilidad	Arduino	Arduino	Arduino

Selección del módulo control de motores

Para seleccionar la mejor opción del módulo de control de motores se realiza una evaluación (ver tabla 19) en base a los siguientes criterios:

- Canales de control
- Compatibilidad

- Voltaje
- Programación
- Costo
- Dimensiones

Tabla 19**Evaluación de alternativas de la tarjeta de control de motores**

		Alternativas					
		A		B		C	
		Módulo Shield		Módulo L298N		Módulo A4988	
Críterios	Peso	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
Canales	30%	5	1.5	1	0.3	1	0.3
Compatibilidad	20%	5	1	3	0.2	2	0.4
Voltaje	25%	3	0.75	3	0.75	3	0.75
Programación	10%	4	0.4	3	0.3	3	0.3
Dimensiones	10%	3	0.3	3	0.3	4	0.4
Costo	5%	3	0.15	4	0.2	4	0.2
Total	100%		4.1		2.05		2.35
	Lugar		1		3		2
¿Continuar?		Desarrollar		No		No	

De la tabla 19, la Alternativa A (Módulo Shield con L293D) es el componente idóneo, no solo por el controlar la velocidad y sentido de dirección de los motores, sino también por su voltaje compatible con la tarjeta de control, adaptable con el arduino Mega, posee una programación amigable y además dispone 3 canales para el manejo de motores que es lo que se requiere.

2.5.5. Mecanismo de transmisión

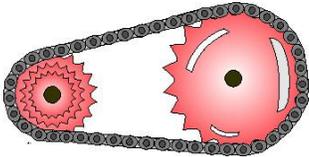
Una vez seleccionado el tipo de motor se procede a la selección del mecanismo que es un dispositivo que transforma el movimiento producido por un componente motriz en un tipo movimiento esperado a la salida. Se analizan tres alternativas de mecanismo de transmisión que serán encargados de transmitir movimiento de una parte del sistema a otra.

- Sistema de cadenas y piñones
- Sistema tornillo sinfín y rueda dentada
- Sistema de engranes

La tabla 20 muestra una comparación de características entre las alternativas de mecanismos.

Tabla 20

Comparación de mecanismos de transmisión

		
Sistema de cadenas y piñones	Sistema tornillo sinfín y rueda dentada	Sistema de engranes
Transmite potencias altas entre dos ejes	Transmitir fuerza entre ejes perpendiculares	Transmisión de movimiento rotatorio
Relación de transmisión constante	Relación de transmisión altas	Relación de transmisión estable
Sistema irreversible	Sistema irreversible	Sistema reversible
Sistema costoso y ruidoso	Sistema costoso	Sistema costoso
Requieren espacios amplios	Ocupan espacios reducidos	Ocupan espacios reducidos

Selección del mecanismo de transmisión.

La selección del mecanismo de transmisión que va a controlar la velocidad de los motores de corriente continua se basa en los siguientes parámetros:

- Tipo de transmisión
- Relación de transmisión
- Sentido de giro
- Dimensión
- Costo

En la tabla 21, se realiza una evaluación de las alternativas de mecanismos de transmisión en base a los parámetros establecidos.

Tabla 21

Evaluación de alternativas del mecanismo de transmisión

Criterios	Peso	Alternativas					
		A		B		C	
		Cadenas y piñones		Tornillo sinfín y rueda dentada		Sistema de engranes	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Tipo de transmisión	30%	5	1.5	5	1.5	5	1.5
Relación de transmisión	30%	4	1.2	4	1.2	5	1.5
Sentido de giro	15%	2	0.3	2	0.3	5	0.75
Dimensión	15%	3	0.45	5	0.75	5	0.75
Costo	10%	3	0.3	3	0.3	3	0.5
Total	100%		3.75		4.05		5
	Lugar		3		2		1
¿Continuar?			No		No		Desarrollar

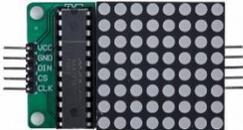
De acuerdo con la tabla 21, la alternativa C (Sistema de engranes), es el mecanismo de transmisión más conveniente debido al movimiento rotatorio, transmisión estable, sistema reversible y además que ocupa un espacio reducido para su instalación.

2.5.6. Módulo de matriz leds

El módulo de matriz leds se utiliza para simular y mostrar los gestos emocionales, que van acorde a las instrucciones que el robot brinda, se analizan tres alternativas de módulos de matriz leds comerciales (ver tabla 22).

- Matriz 5x7
- Matriz 8x8 con MAX7219
- Matriz 8x16 mBot

Tabla 22
Comparación de la matriz de led

			
	Matriz 5x7	Matriz 8x8	Matriz 8x16 mBot
Costo	\$5	\$10	\$9
Integrado	NA	MAX7219	NA
Tamaño	5,3 x 3,7 cm	4,3 x 2,3 cm	7,3 x 3,2 x 2 cm
Pines	14	Entrada 5 Salida 5	4
Leds	35	64	129
Color leds	Rojo	Rojo	Azul
Voltaje	5V	5V	5V
Compatibilidad	Arduino - PIC	Arduino	Arduino - mBot
Peso	40 gramos	100 gramos	35 gramos

Selección del módulo de matriz de leds

Para la selección de la matriz de leds se realiza una evaluación (ver tabla 23) de acuerdo a los siguientes criterios:

- Número de leds
- Número de pines de salida
- Compatibilidad
- Programación
- Dimensión
- Peso
- Costo

En la tabla 23 se realiza la evaluación y selección de la mejor alternativa de matriz de leds, particularizando los atributos más relevantes y característicos de los dispositivos.

Tabla 23**Evaluación de alternativas de la matriz de leds**

Criterios	Peso	Alternativas					
		A		B		C	
		Matriz 8x8		Matriz 5x7		Matriz 8x16 mBot	
		Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
Número de leds	25%	2	0.5	1	0.25	5	1.25
Pines de salida	20%	2	0.4	2	0.4	5	1
Compatibilidad	15%	3	0.45	3	0.45	5	0.75
Programación	10%	3	0.3	3	0.3	3	0.3
Dimensión	10%	4	0.4	3	0.3	5	0.5
Peso	10%	2	0.2	2	0.2	3	0.3
Costo	10%	2	0.2	4	0.4	3	0.3
Total	100%		2.45		2.3		4.4
	Lugar		2		3		1
¿Continuar?		No		No		Desarrollar	

De la tabla 23, la Alternativa C (Matriz de led 8x16 mBot) es el componente idóneo, no solo por el número de pines de salida a controlar, sino también por el número de leds para simular y mostrar gestos y emociones, es totalmente compatible con la programación del arduino Mega, y su dimensión y peso son ideales para instalarlo en cualquier espacio.

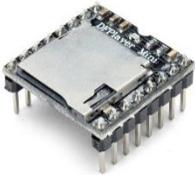
2.5.7. Módulo de audio

El módulo permite gestionar y reproducir formatos de audio que se encuentran sincronizadas con los procesos de la realidad aumentada para brindar instrucciones de audio a los usuarios, se analizan tres alternativas de módulos de audio comerciales.

- Módulo WTV020-SD-16P Mini
- Módulo DFPlayer Mini MP3
- Módulo ISD1820 grabador y reproductor de voz

En la tabla 24, se realiza la comparación entre los módulos de audio, con el propósito de detallar la singularidad técnica de cada módulo.

Tabla 24
Comparación del módulo de audio

			
	WTV020-SD-16P	DFPlayer Mini	Módulo ISD1820
Costo	\$10	\$12,50	\$11
Tamaño	1,74 x 2,06 cm	2 x 2 cm	5,4 x 3,8 cm
Altavoces	Externo	Externo	Interno
Micrófono	No	No	Si
Formato audio	AD4, WAV	MP3, WAV, WMA	Todos
Micro SD	1GB	32GB	No
Voltaje	3,3 - 5V	5V	2,4 - 5V
Compatibilidad	PIC - Arduino	Arduino	Arduino, STM32
Peso	5 gramos	4 gramos	17 gramos

Selección del módulo de audio

Para seleccionar la mejor opción del módulo de audio se realiza una evaluación ponderada (ver tabla 25) en referencia a los siguientes criterios:

- Formato de audio
- Voltaje
- Compatibilidad
- Tarjeta micro SD
- Dimensión
- Programación
- Costo

Tabla 25**Evaluación de alternativas del módulo de audio**

Criterios	Peso	Alternativas					
		A		B		C	
		WTV020-SD-16P		DFPlayer Mini		Módulo ISD1820	
		Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada	Califi- cación	Evaluación ponderada
Formato audio	25%	2	0.5	5	1.25	1	0.25
Voltaje	20%	3	0.6	3	0.6	3	0.6
Compatibilidad	15%	3	0.45	3	0.45	3	0.45
Micro SD	15%	2	0.3	5	0.75	1	0.15
Programación	10%	3	0.3	3	0.3	3	0.3
Dimensión	10%	3	0.3	5	0.5	3	0.3
Costo	5%	4	0.2	3	0.15	3	0.15
Total	100%		2.65		4		2.2
	Lugar		2		1		3
¿Continuar?		No		Desarrollar		No	

De la tabla 25, la Alternativa B (Módulo DFPlayer Mini MP3) es el componente más idóneo, por permitir reproducir formatos de audio MP3, voltaje ideal para la tarjeta de control, compatible con la programación del arduino Mega, y soporta una tarjeta micro SD hasta 32 GB.

2.6. Sistema de transmisión de datos

El sistema de transmisión de datos cumple la función de comunicar el sistema de Realidad Aumentada con el Robot Interactivo mediante una interfaz inalámbrica, se analizan tres alternativas de sistema de comunicación más usadas (ver tabla 26).

- Módulo RF APC220
- Módulo Bluetooth HC-05
- Módulo XBee

Tabla 26

Comparación del sistema de transmisión de datos

			
	RF APC220	Módulo Bluetooth	Módulo XBee
Costo	\$40	\$9	\$18
Tamaño	3,7 x 1,7 x 0,6 cm	4,4 x 1,6 X 0,7cm	5,4 x 3,8 cm
Alcance	800 - 100 metros	5 - 10 metros	40 - 120 metros
Protocolo	USART/TTL	USART5V	Zigbee
Frecuencia	418 – 455 MHz	2,4 GHz	900 MHz
Velocidad	1200 -19200bps	1200-115200 bps	256 kbps
Voltaje	3,3 - 5V	5V	3,3V
Compatibilidad	Arduino	MCU-Arduino	Arduino
Peso	30 gramos	30 gramos	4 gramos

Selección de sistema de transmisión de datos

La seleccionar de la mejor alternativa del módulo para transmisión de datos se realiza en base a una evaluación ponderada (ver tabla 27) acorde a los siguientes criterios.

- Velocidad de transmisión
- Alcance de transmisión
- Programación
- Adaptable
- Dimensión
- Peso
- Costo

Tabla 27**Evaluación de alternativas del sistema de transmisión de datos**

Criterios	Peso	Alternativas					
		A		B		C	
		RF APC220		Módulo Bluetooth		Módulo XBee	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Velocidad	25%	3	0.75	5	1.25	2	0.5
Alcance	20%	4	0.8	3	0.6	4	0.8
Programación	15%	3	0.45	3	0.45	3	0.45
Adaptable	15%	3	0.45	3	0.45	3	0.45
Dimensión	10%	3	0.3	5	0.5	2	0.20
Peso	10%	3	0.3	3	0.3	4	0.4
Costo	5%	2	0.1	3	0.15	2	0.1
Total	100%		3.15		3.7		2.9
	Lugar		2		1		3
¿Continuar?		No		Desarrollar		No	

De la tabla 27, la Alternativa B (Módulo Bluetooth HC-05) es el componente más recomendable para ser usado como medio transmisión de datos, por ser adaptable con la tarjeta Arduino, alta velocidad de transmisión, tener un rango de alcance bastante bueno, asequible de programar y costo muy económico.

2.7. Componentes seleccionados

Los componentes seleccionados cumplen los requerimientos necesarios para efectuar un sistema de RA con asistente robótico, estos componentes son de amplia gama y disponibilidad en tiendas electrónicas.

En la figura 34 se resumen todos los componentes que han sido seleccionados acorde a las especificaciones técnicas requeridas.



Figura 34 Componentes seleccionados

Una vez seleccionado los elementos principales, el siguiente paso es el diseño mecatrónico del robot didáctico donde se analizará los componentes mecánicos y electrónicos.

CAPÍTULO III

DISEÑO Y CONSTRUCCIÓN DEL ROBOT DIDÁCTICO

3.1. Introducción

En el presente capítulo se realiza el diseño mecatrónico del sistema de robot asistente, que comprende el diseño de detalle mecánico, electrónico y software del robot, en el capítulo 2 (sección 2.4.1. Aspecto del robot) se realizó la selección del tipo de robot a diseñar, en este caso el robot asistente tendrá un aspecto humanoide, parecido a los robots Nao.

El robot didáctico humanoide se componen de una parte mecánica conformada por las piezas que en conjunto forman la estructura de soporte, los mecanismos de transmisión de movimiento y los elementos que generan movimiento rotatorio, se componen también de una parte electrónica que alberga todos los sistemas electrónicos de control y alimentación eléctrica, a todo esto se complementa la parte de software donde mediante un controlador y por medio de programación se realiza la sincronización de los sub-sistemas del robot didáctico.

3.2. Diseño Mecánico

En este apartado se describe la funcionalidad y se realiza el modelado 3D de cada una de las partes del robot, para posteriormente realizar el análisis estático de aquellas piezas que se encuentran sometidas a mayor esfuerzo, esto con la finalidad de verificar si el diseño es seguro y cumple con los requerimientos para ser materializado en un modelo 3D físico y funcional.

3.2.1. Diseño de la estructura del robot

Se comienza con el diseño de las piezas que conforman la estructura y la carcasa, para la carcasa de la cabeza y las extremidades se realiza un diseño similar al modelo del robot Nao (ver figura 35), el torso será diseñado en base a

la forma y tamaño del controlador Arduino Mega, para que éste pueda ser colocado y retirado de una manera fácil; las piezas de soporte internas serán diseñadas de tal manera que soporten el peso de los componentes y la carcasa.

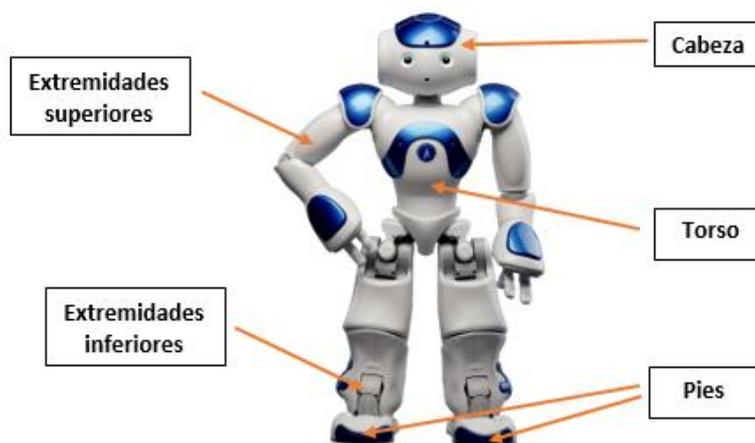


Figura 35 Estructura del robot Nao

Fuente: (Robotlab, 2017)

Las piezas se diseñan en un software de modelado 3D, los planos y ensambles del robot se incluyen en el Anexo C. Las partes principales se describen a continuación.

- **Cabeza.** En la parte inferior de la cabeza frontal va estar alojado la matriz de leds 8x16 que simulará los gestos y expresiones. Mientras que en la parte central estará el módulo esclavo HC-06, ver la figura 36.

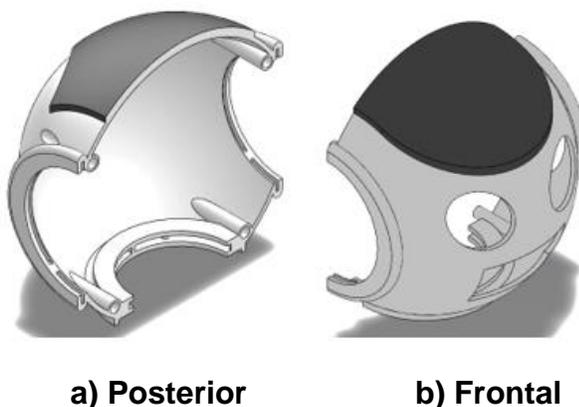


Figura 36 Cabeza del robot didáctico

- **Oreja.** En las partes laterales de la cabeza se localizan las orejas, en donde se van alojar parlantes pequeños ver figura 37, los cuales emitirán sonidos para que el niño y niña pueda ejecutar las instrucciones.

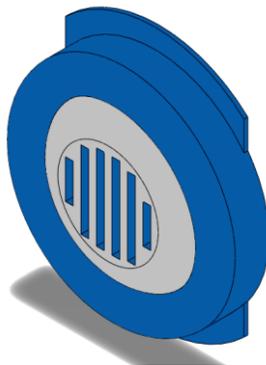


Figura 37 Oreja del robot didáctico

- **Cuello.** Este componente se encuentra ubicado en la parte inferior de la cabeza, es el elemento de unión entre la cabeza y el cuerpo, se caracteriza por que permite el movimiento de 120 grados (ver figura 38).

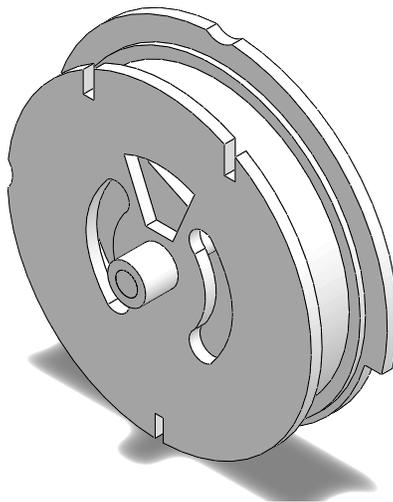


Figura 38 Cuello del robot didáctico

- **Espalda.** En la espalda se van a ubicar los componentes como: columna, torso, clavícula, articulaciones del brazo, eje del brazo. Este componente sirve de soporte a la cabeza como se puede ver la figura 39.

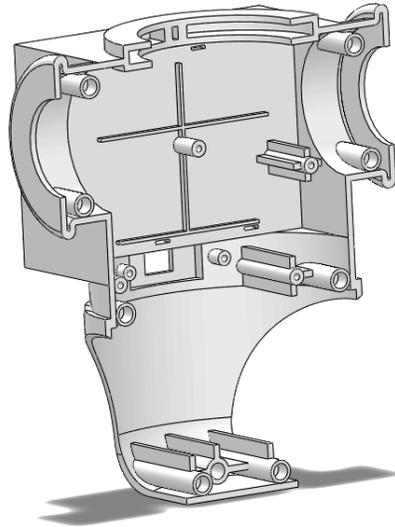


Figura 39 Espalda del robot didáctico

- **Pecho.** En la parte posterior central del componente va estar ubicado el arduino Mega, que cumple la función de tarjeta principal tanto para la comunicación y el control de motores, como se ve en la figura 40.

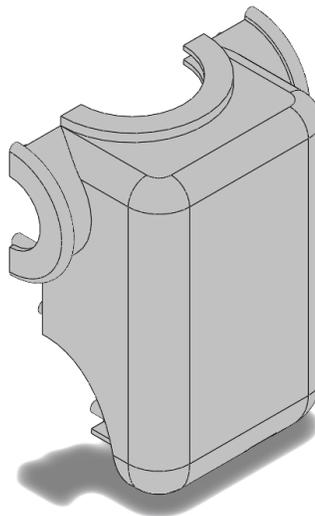


Figura 40 Pecho del robot didáctico

- **Columna.** Este componente sirve como soporte de la cabeza y brazos. En las partes inferiores laterales del componente se encuentran 4 soportes en las cuales van a ir las piernas como se visualiza en la figura 41.

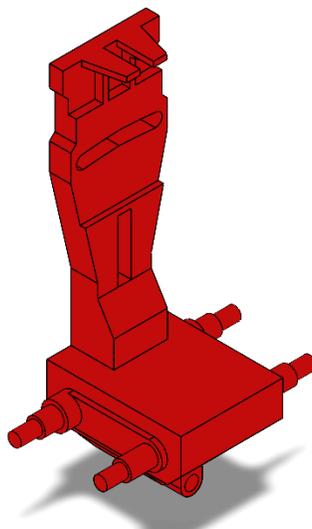


Figura 41 Columna del robot didáctico

- **Torso 1.** Este componente como se puede ver en la figura 42 en conjunto con el torso 2 va unido con la columna. En la parte inferior va ir colocado un motor el cual servirá para dar movimiento a los brazos del robot.

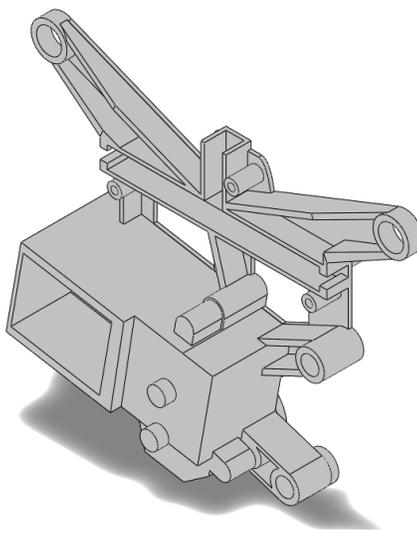


Figura 42 Torso 1 del robot didáctico

- **Torso 2.** Como se ve en la figura 43 esta pieza va unida a la parte posterior del torso 1.

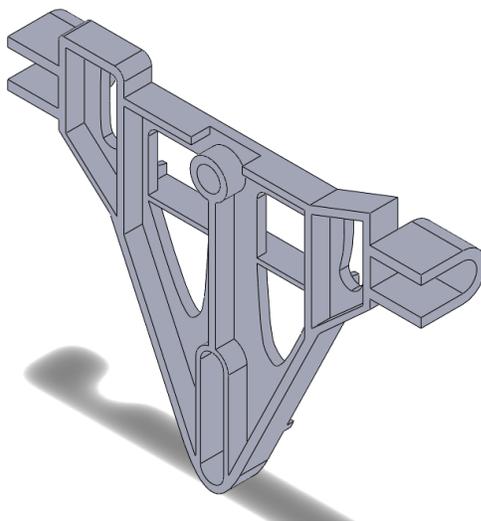


Figura 43 Torso 2 del robot didáctico

- **Articulación del brazo.** Es el elemento de unión entre el brazo y el cuerpo, sirve también como guía para el movimiento de los brazos del robot como se puede ver en la figura 44.

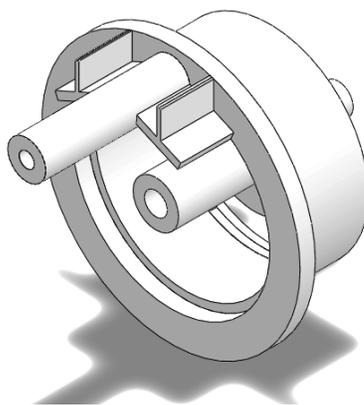


Figura 44 Articulación brazo del robot didáctico

- **Soporte pierna.** Junto con la columna, el soporte de pierna va a resistir todos los pesos de los componentes (ver figura 45).

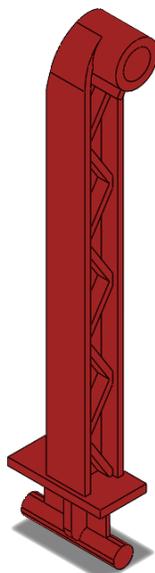


Figura 45 Soporte pierna del robot didáctico

- **Rueda de motor.** Este componente se encuentra ubicado en el interior de cada pie como se puede ver en la figura 46, este posee en su interior una caja de engranes, lo que permite al motor aumentar el torque.

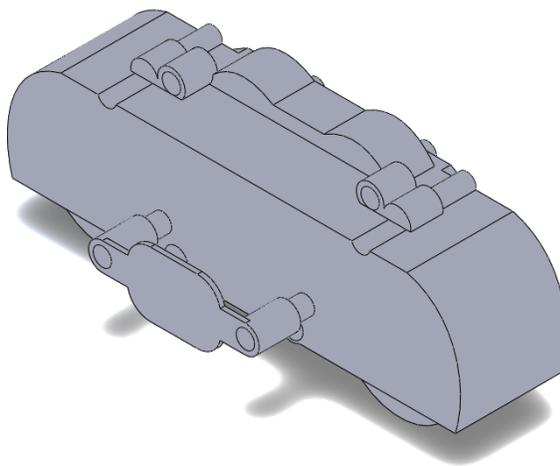


Figura 46 Rueda de motor del robot didáctico

3.2.2. Análisis estático de la estructura del robot

El análisis estático ayudará a comprobar si el diseño de la misma es robusto ya que este va soportar peso de todos componentes incluido motores y componentes electrónicos.

Para el diseño de toda la estructura se emplea un material Acido Poli láctico (PLA) que se obtiene a partir del almidón de maíz, es biodegradable, no tóxico y utilizado ampliamente en la impresión 3D. Además de ser un material resistente muy utilizado para la fabricación de juguetes. En la tabla 28 se puede identificar algunas propiedades mecánicas y físicas del PLA.

Tabla 28
Propiedades físicas del PLA

PLA (Ácido Poli ácido láctico)	
Densidad	<i>1.25 g/cm³</i>
Módulo de corte	<i>2400 MPa</i>
Módulo de flexión	<i>4000 MPa</i>
Resistencia a la flexión	<i>80 MPa</i>
Módulo de elasticidad	<i>3500 MPa</i>
Limite elástico	<i>60 MPa</i>
Límite de tracción	<i>50 MPa</i>
Temperatura de transición vítrea	<i>65°C</i>
Comienzo de fusión	<i>160°C</i>
Coeficiente de Poisson	<i>0.29</i>

Fuente: (Narváez, 2012)

Primeramente, se empieza analizando las partes del esqueleto, debido a que estas sirven de soporte para las demás piezas y están sometidas a las fuerzas que ejercen los demás componentes (ver figura 47).

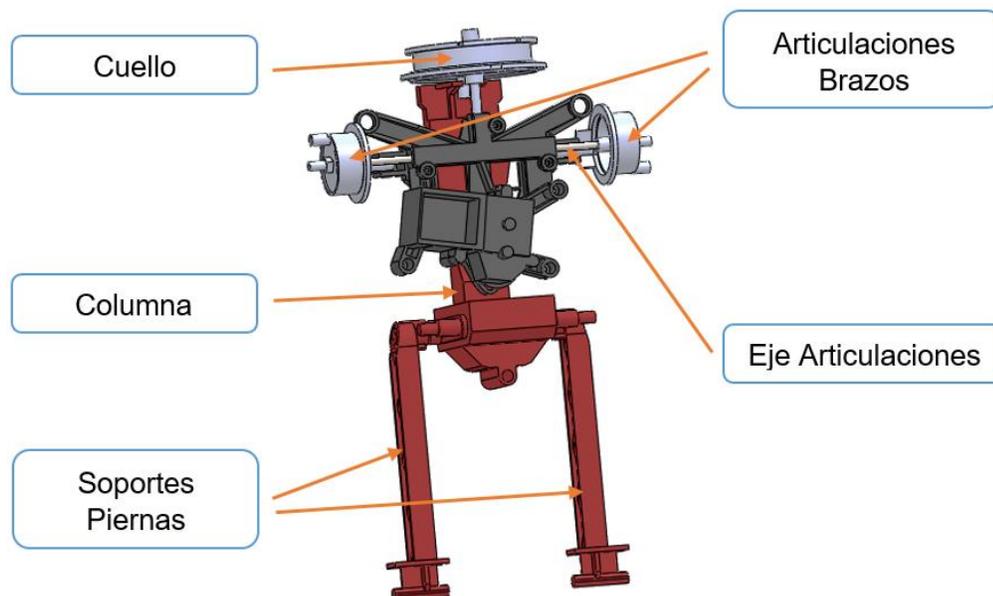


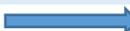
Figura 47 Esqueleto de soporte

En la tabla 29 se indica la cantidad y el peso de cada componente, cabe destacar que estos datos se han tomado de las propiedades físicas del programa de modelado 3D.

Tabla 29

Lista de peso de cada componente del robot

Elemento	Cantidad	Peso [gr]	Peso [N]	Peso Total[N]
Cabeza	1	133.22	1.3	1.3
Articulación brazo	2	8.36	0.04	0.08
Antifaz de ojos	1	7.30	0.07	0.07
Clavícula	1	1.83	0.017	0.017
Columna	1	84.75	0.83	0.83
Cuello	1	17.20	0.16	0.16
Oreja	2	77.76	0.38	0.76
Switch	1	2.81	0.027	0.027
Torso 1	1	42.73	0.41	0.41

CONTINÚA 

Torso 2	1	8.81	0.086	0.086
Espalda	1	103.71	1.01	1.01
Pecho	1	96.10	0.94	0.94
Brazo	2	116.3	0.57	1.14
Pierna	2	90.9	0.44	0.89
Soporte pierna	2	17.34	0.084	0.17
Rueda Motor	2	154.42	0.75	1.5
Pie	2	37.24	0.73	1.46
TOTAL	24	1000.76		10.85

Una vez conocido los pesos de cada elemento se procede a crear un nuevo estudio estático. Primero se coloca una fuerza perpendicular al piso vector de gravedad (9.8 m/s^2), para luego crear una sujeción de geometría fija como se observa en la figura 48.

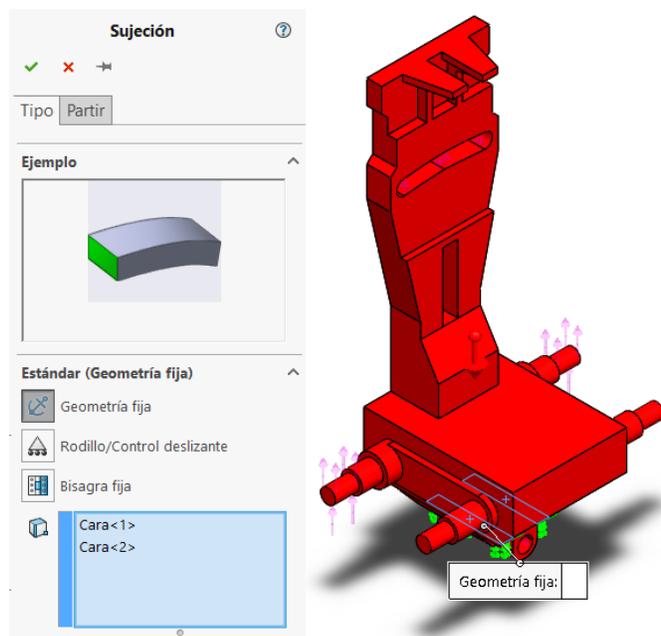


Figura 48 Geometría Fija en la columna

En el siguiente paso se selecciona la opción **Fuerza**, luego se escoge las caras y la dirección de la fuerza contraria al vector gravedad como se puede ver en la figura 49.

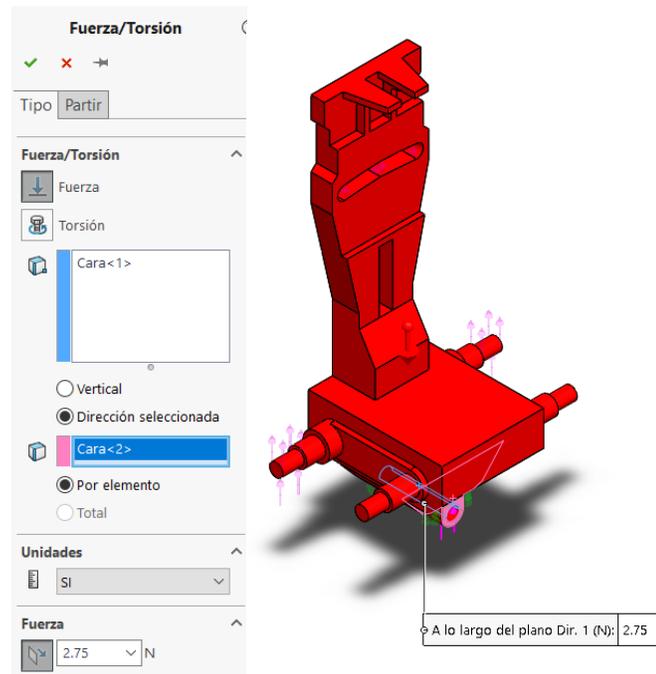


Figura 49 Fuerza aplicada en la cara inferior de la columna

Se procede a crear una malla fina de la pieza anterior con un espesor de 3 a 5 milímetros como máximo (figura 50).

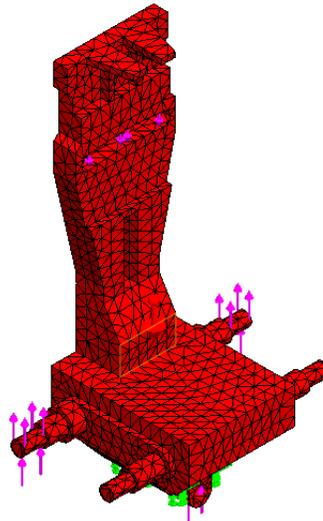


Figura 50 Mallado de la columna del robot

La tensión de Von Mises expone que al aplicar una fuerza a los materiales dúctiles estos tienden a deformarse sin romperse, es decir que la tensión de Von Mises debe ser menor a la tensión del límite elástico ver ecuación 4.

$$\sigma_{VonMises} < \sigma_{Limite\ Elástico} \quad (4)$$

Se observa en la figura 51 un valor máximo de la tensión de Von Mises del $0.801[MPa]$, al conocer este dato inmediatamente se reemplaza en la ecuación 4 y compara con la tensión del límite elástico del material PLA de la tabla 25. Con certeza se puede decir que no existirá fallo del material al aplicar esta fuerza.

$$\sigma_{VonMises} < \sigma_{Limite\ Elástico}$$

$$0.801[MPa] < 60[MPa]$$

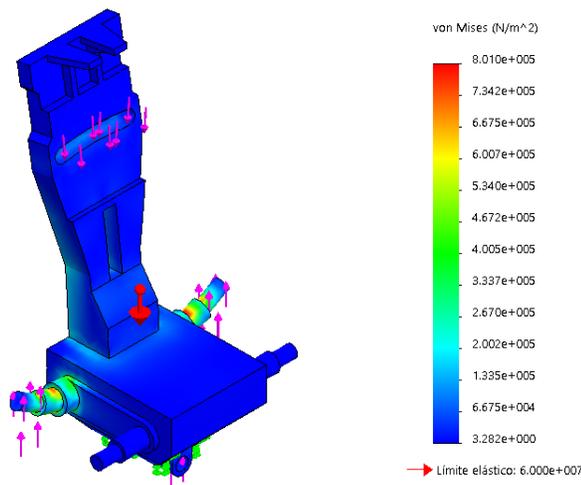


Figura 51 Tensión de Von Mises en la columna

Con la ecuación 5 se considera que el esfuerzo de diseño tiene que ser mayor al esfuerzo máximo que soporta la estructura, para que no provoque ningún fallo.

$$\sigma_d \geq \sigma' \quad (5)$$

$$\sigma_d = \frac{S_y}{N} \quad (6)$$

Donde:

S_y : Resistencia a la flexión.

N : Factor de seguridad

$$\frac{S_y}{N} = \sigma_d$$

$$\frac{80 \text{ [MPa]}}{1.5} = 53.33 \text{ [MPa]}$$

Una vez reemplazada los datos de la ecuación 6 con un factor de seguridad del 1.5 valor que se ha puesto por seguridad se tiene que la tensión de diseño es 53.33 [MPa].

$$\sigma_d \geq \sigma'$$

$$53.33 \text{ [MPa]} \geq 0.801 \text{ [MPa]}$$

Al aplicar la ecuación 5 se puede decir que el esfuerzo de diseño es mayor o igual que el esfuerzo que soporta la estructura.

Como se puede ver en la figura 52 el factor de seguridad de la columna tiene un valor mínimo de 2.7 y un valor máximo 5 lo cual se deduce que es seguro para su construcción.

El diseño de la columna debe ser bien robusto, puesto a que es el elemento que más esfuerzo soporta, tanto por los acoples de las extremidades inferiores como las demás elementos que componen el torso.

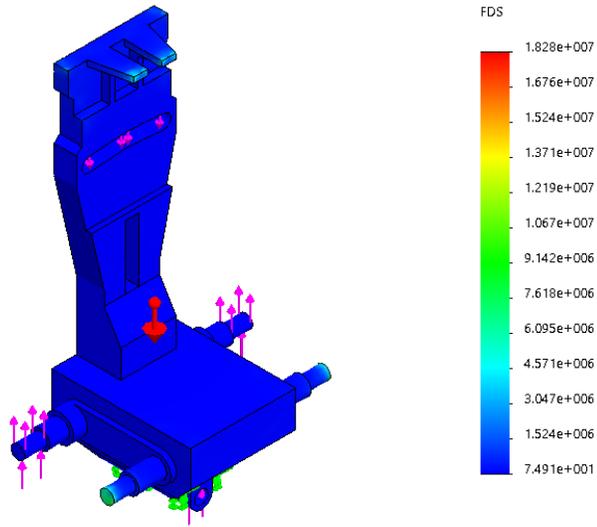


Figura 52 Factor de seguridad en la columna

Una vez realizado el análisis en la columna se procede a realizar el análisis en el soporte de la pierna del robot e igualmente primero se elige una sujeción fija en la parte inferior para realizar el análisis como se puede ver en la figura 53.

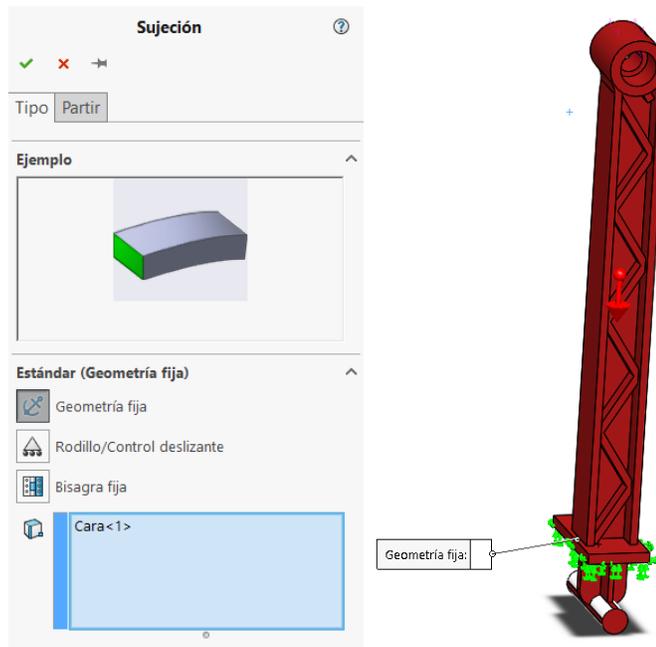


Figura 53 Geometría Fija en la columna

Se ingresa un valor en Newtons de la Fuerza que se va aplicar en la cara inferior, además de colocar la fuerza de gravedad (ver figura 54).

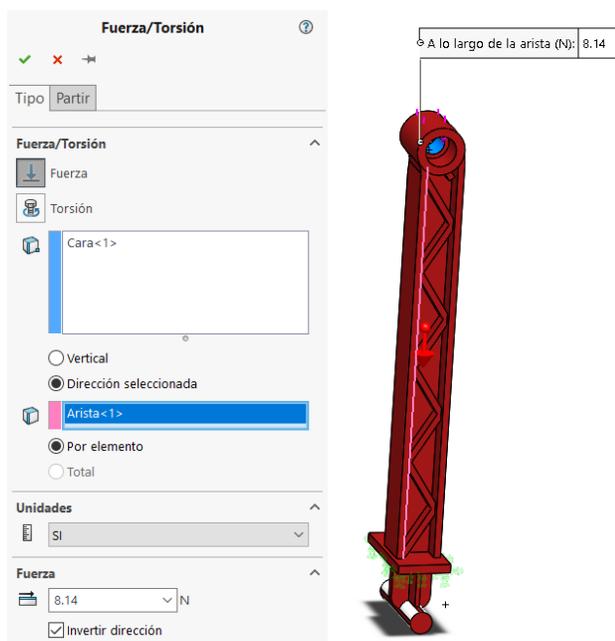


Figura 54 Fuerza aplicada en la cara inferior de la columna

Se procede a crear una malla fina de la pieza anterior con un espesor de 3 a 5 milímetros como máximo como se puede ver en la figura 55.

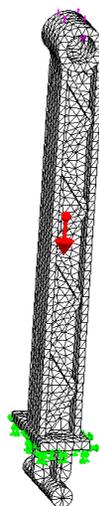


Figura 55 Mallado del soporte de pierna

En la figura 56 se observa un valor máximo de la tensión de Von Mises del 0.72 MPa. Con lo que se puede aplicar directamente la ecuación 4 que nos indica que no ocurrirá ninguna falla y soporta la carga establecida.

$$\sigma_{VonMises} < \sigma_{Limite\ Elástico}$$

$$0.72[MPa] < 60[MPa]$$

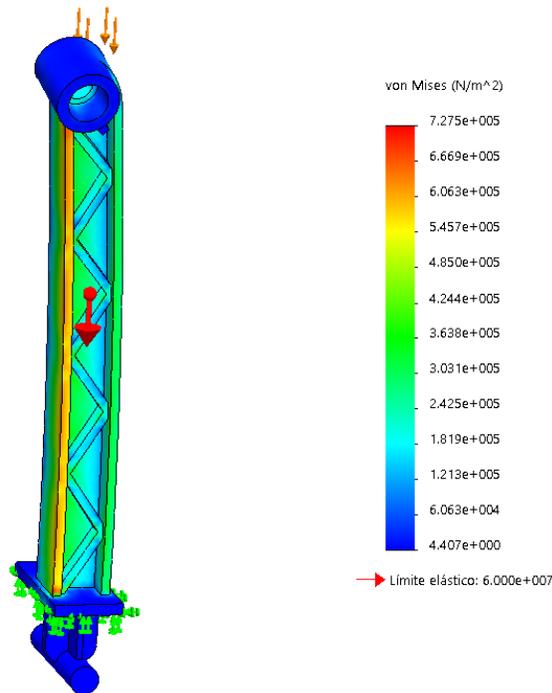


Figura 56 Tensión de Von Mises en el soporte de pierna

Se ha aplica nuevamente la ecuación 5 para comprobar si el coeficiente de diseño es mayor o igual que el esfuerzo que soporta la estructura con lo que se puede observar que cumple con lo deseado.

$$\sigma_d \geq \sigma'$$

$$53.33 [MPa] \geq 0.72[MPa]$$

El factor de seguridad es 64 lo cual se deduce que el material es seguro para la construcción como se puede ver en la figura 57.

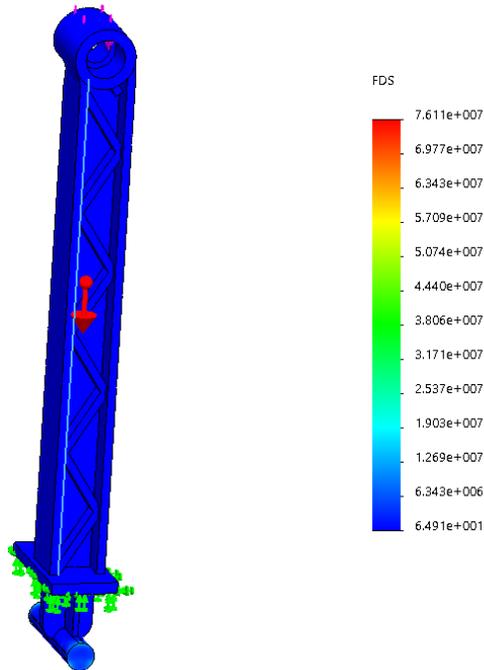


Figura 57 Factor de seguridad en el soporte de pierna

Se procede a realizar el análisis del cuello, para ello se crea una sujeción de geometría fija en la cara inferior (figura 58).

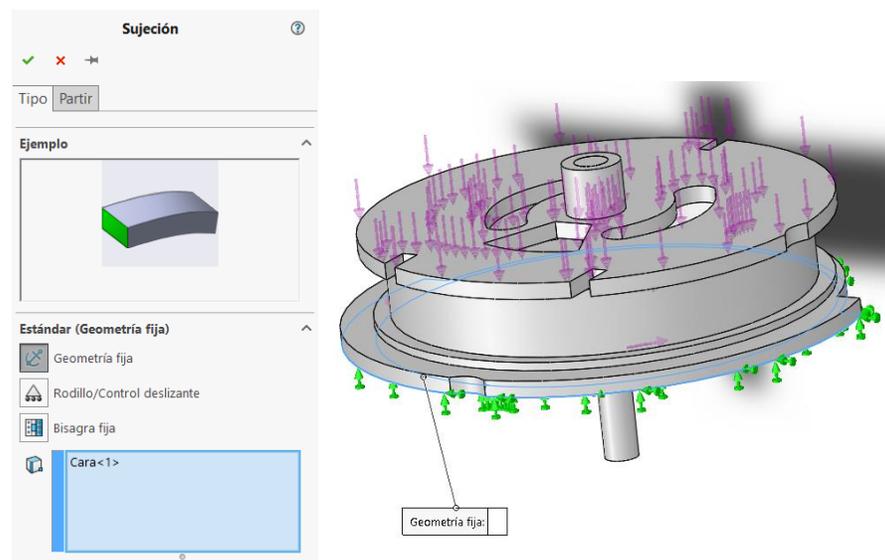


Figura 58 Geometría Fija en el cuello

En el siguiente paso se escoge la opción **Fuerza**, y se selecciona la cara superior en conjunto con la dirección que va a tener la fuerza en este caso la misma que la gravedad (ver figura 59).

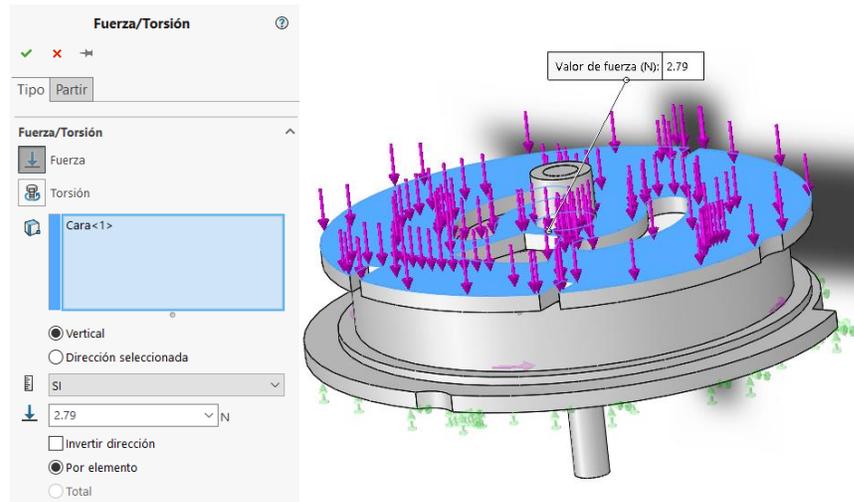


Figura 59 Fuerza aplicada en la cara superior del cuello

En el siguiente paso se escoge la opción **Torsión**, se selecciona la cara y el plano de la dirección que va a tener la torsión (ver figura 60).

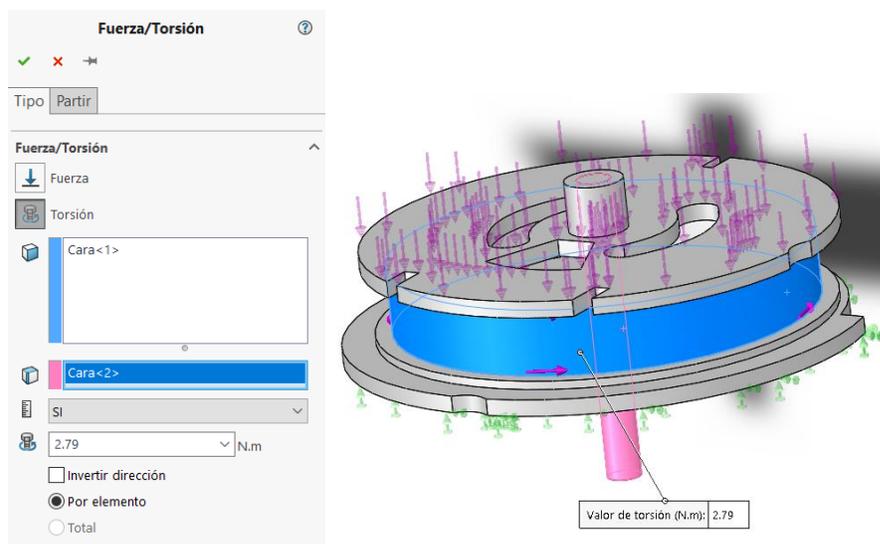


Figura 60 Torsión aplicada en la cara del cuello

Se procede a crear una malla fina de la pieza anterior con un espesor de 3 a 5 milímetros como máximo como se puede ver en la figura 61.

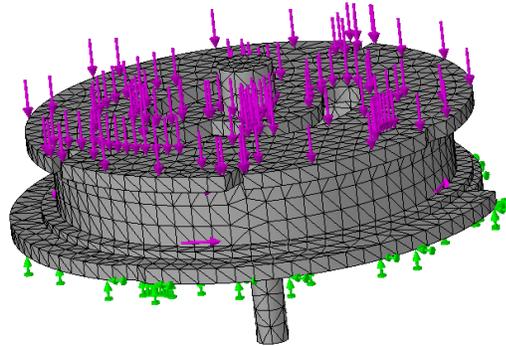


Figura 61 Mallado del cuello

Se observa en la figura 62 un valor máximo de la tensión de Von Mises del $1.246[MPa]$, se procede a reemplazar en la ecuación 4 y comparar con la tensión del límite elástico. Como resultado se tiene que no existirá fallo del material al aplicar esta fuerza.

$$\sigma_{VonMises} < \sigma_{Limite\ Elástico}$$

$$1.246[MPa] < 60[MPa]$$

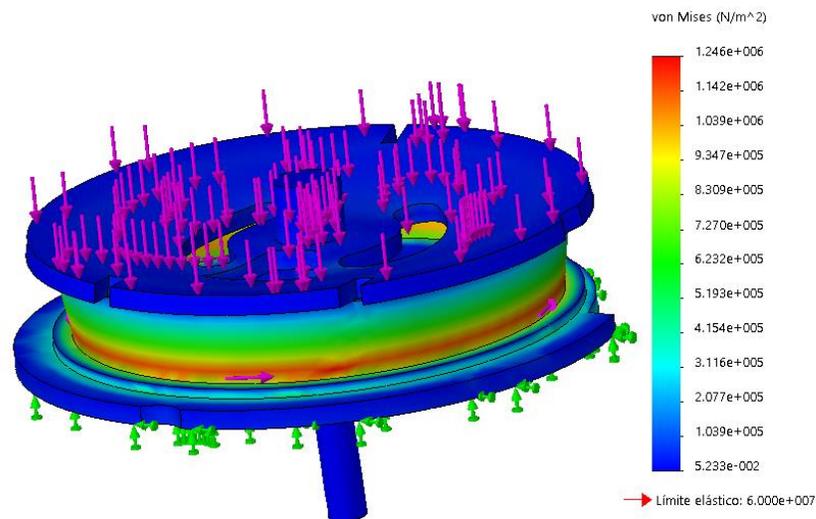


Figura 62 Tensión de Von Mises en el cuello

Una vez reemplazado los resultados en la ecuación 5 se comprueba que el coeficiente de diseño es mayor o igual que el esfuerzo que soporta la estructura.

$$\sigma_d \geq \sigma'$$

$$53.33 [MPa] \geq 1.246[MPa]$$

Como se puede ver en la figura 63 el factor de seguridad de la columna tiene un valor mínimo de 48, lo cual se deduce que el diseño de la pieza es seguro para su construcción.

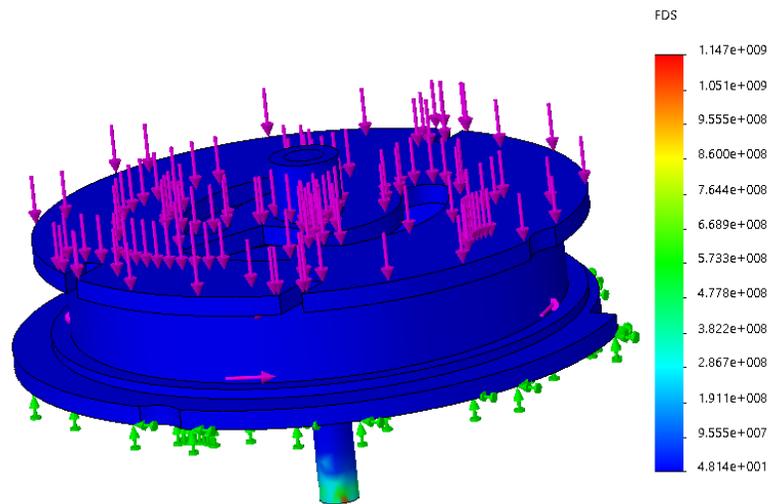


Figura 63 Factor de seguridad en el cuello

3.2.3. Análisis de Impacto de la carcasa del robot

El análisis de impacto de la estructura ayuda a comprobar si la carcasa del robot puede soportar las caídas. Se procede a crear un nuevo estudio se selecciona **Caída** y se establece la altura de un metro y la gravedad en este caso (9.8 m/s^2), tal cual se observa en la figura 64.

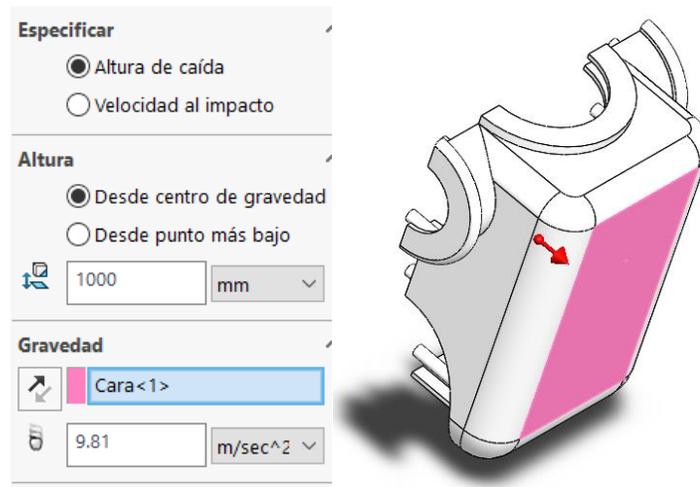


Figura 64 Análisis de impacto en el pecho

Se procede a realizar el mallado de la pieza “pecho” como se puede ver en la figura 65.

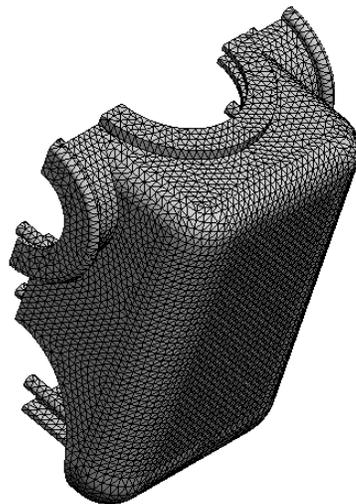


Figura 65 Mallado del pecho

Se observa en la figura 66, un valor máximo de la tensión de Von Mises del $51.07[MPa]$, se procede a reemplazar en la ecuación 4 y comparar con la tensión del límite elástico. Como resultado se tiene que no existirá fallo del material al aplicar esta fuerza.

$$\sigma_{VonMises} < \sigma_{Limite\ Elástico}$$

$$51.07[MPa] < 60[MPa]$$

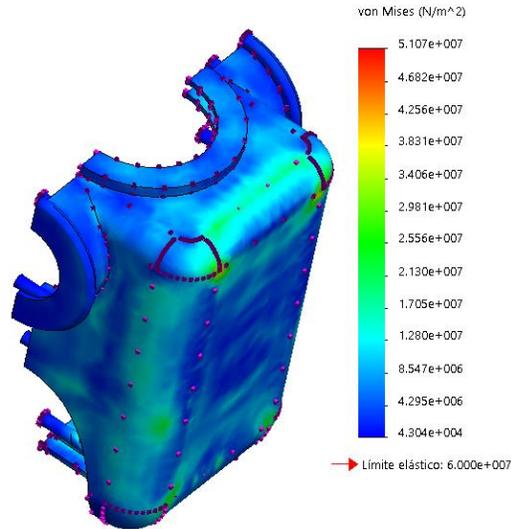


Figura 66 Tensión de Von Mises en la pieza “pecho”

Una vez reemplazado los resultados en la ecuación 5 se comprueba que el coeficiente de diseño es mayor o igual que el esfuerzo que soporta la estructura.

$$\sigma_d \geq \sigma'$$

$$53.33 [MPa] \geq 51.07[MPa]$$

3.2.4. Análisis de espesor de la carcasa del robot

Para el análisis se define un espesor de 2 mm como mínimo para toda la carcasa del robot interactivo y se procede a realizar el análisis de espesor del cabeza en la que se muestra que la mayoría del elemento cumple con el espesor deseado ver figura 67.

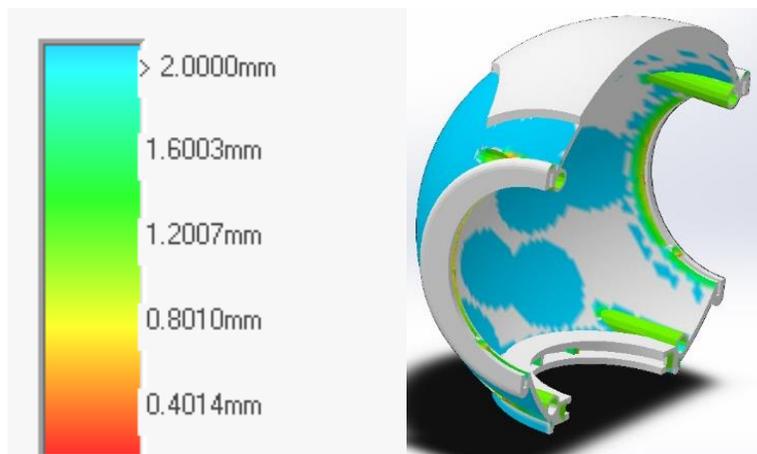


Figura 67 Análisis de espesor de la cabeza posterior

De igual manera se procede a analizar la cabeza frontal con lo que se puede concluir que el elemento no presenta ningún espesor crítico para su construcción, revisar figura 68.



Figura 68 Análisis de espesor de la cabeza frontal

Los análisis estáticos realizados permitieron verificar la robustez del diseño de cada una de las piezas y corregir los parámetros que sean necesarios para garantizar la robustez del diseño.

En la figura 69 se observa el modelado 3D y diseño final del prototipo del robot, con todas las piezas ensambladas.

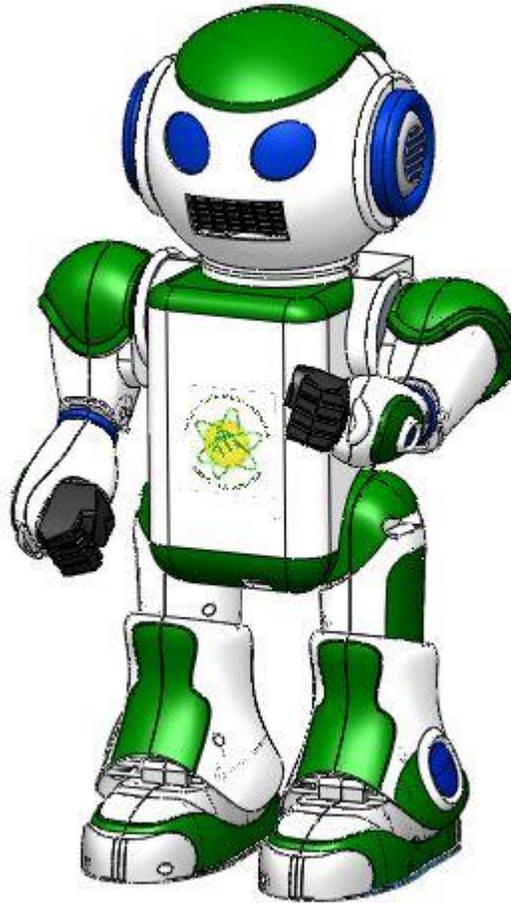


Figura 69 Diseño Final del Robot Didáctico

3.2.5. Prototipado rápido de las piezas diseñadas

Una vez verificado la robustez de las piezas, se procede a replicarlas en piezas físicas funcionales, para esto se recurre a una de las técnicas de prototipado rápido, la impresión 3D que permite crear modelos tridimensionales mediante adición de capas de un determinado material, en este caso el material utilizado es el Ácido Poli láctico (PLA), un plástico biodegradable, las propiedades físicas se muestran en la tabla 28 (sección 3.2.2).

En la figura 70 se observa la impresión de una de las piezas más grandes, en este caso se trata de la carcasa que cubre la parte delantera el “pecho”.

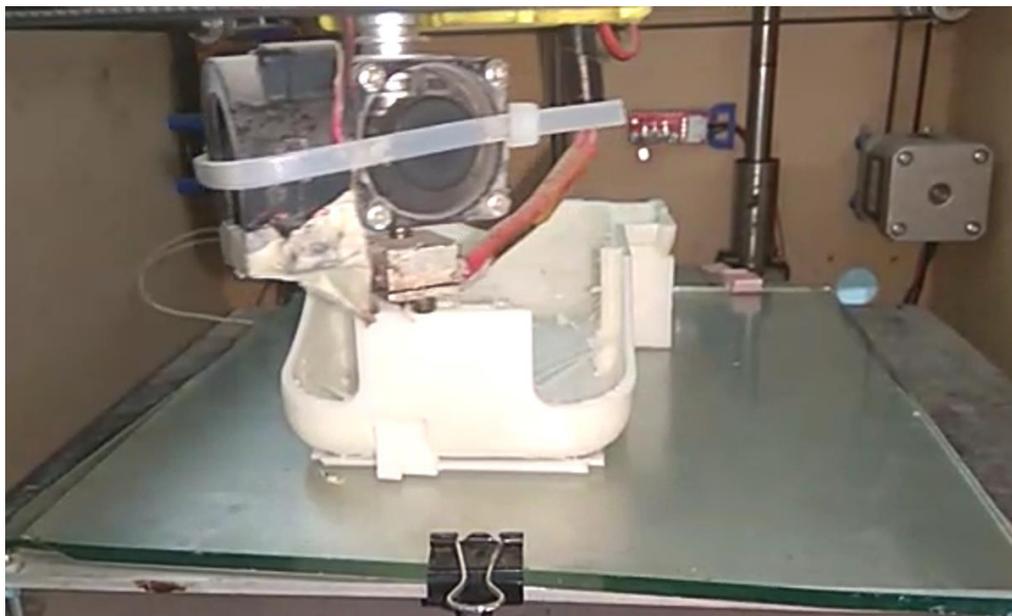


Figura 70 Impresión 3D del pecho

En la figura 71, se observa la pieza “espalda” otra pieza de las más grandes, donde conjuntamente con el “pecho” albergan la mayor parte de los componentes tanto mecánicos y electrónicos.

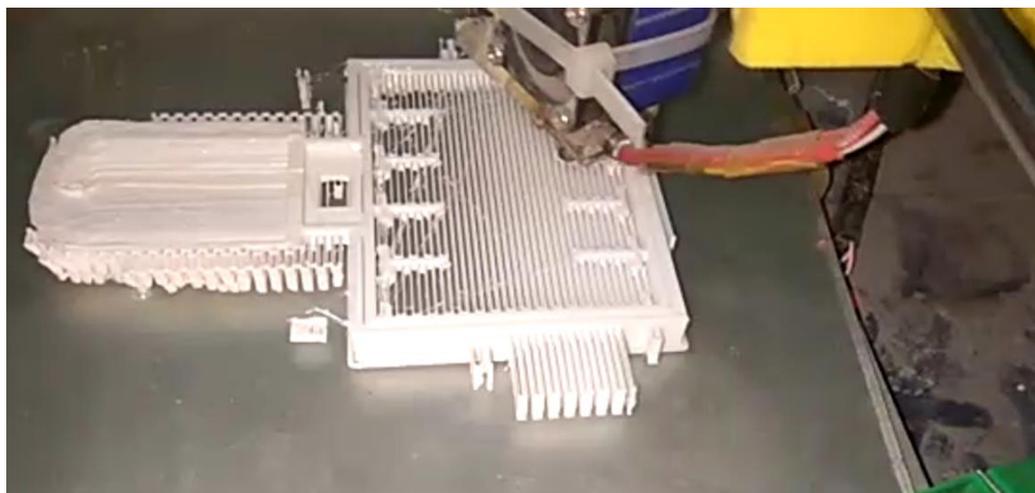


Figura 71 Impresión 3D de la espalda

Al final todas las piezas impresas en 3D, se pulen para lograr un acabado liso, ver figura 72.



(a) Pulido del Pecho (b) Pulido de la espalda (c) Pulido pieza de la pierna

Figura 72 Pulido de piezas impresas

En la figura 73, se observa algunas de las piezas ya terminadas y listas para ser sometida a un proceso de pintado.



Figura 73 Piezas de la carcasa del robot

Posteriormente las piezas son sometidas a un proceso de pintado, el cual permite obtener un aspecto y acabado vivo, ver figura 74.



Figura 74 Proceso de pintado de las piezas

Para el proceso de pintado de las piezas se utiliza pintura en spray no toxico, en la figura 75 se observa algunas de las piezas pintadas y listas para su ensamble.



Figura 75 Piezas de la carcasa pintadas

3.3. Diseño Eléctrico y Electrónico

En este apartado se describen las configuraciones de conexión de los diferentes sistemas electrónico y de control de la tarjeta principal Arduino mega, al igual el consumo eléctrico de todo el sistema en general.

3.3.1. Configuración-programación de los módulos y la tarjeta Arduino.

A. Diagrama de conexión entre Arduino y módulo de audio

El módulo DFPlayer Mini MP3 recibe señales a través de sus pines RX y TX (entradas de transmisión de datos) de los pines digitales de la tarjeta Arduino Mega (ver figura 76). Se emplea este módulo DFPlayer tanto para reproducir formatos de audio y brindar instrucciones a los niños y niñas ya que se encuentra sincronizado con la realidad aumentada.

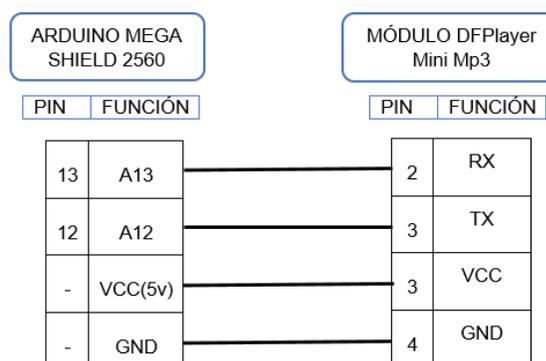


Figura 76 Conexión Arduino Mega y DFPlayer mini mp3

Se utiliza el software Eagle con el propósito de realizar el diseño del circuito electrónico, en el anexo D, se aprecia el diseño de la placa y el diagrama del circuito electrónico del DFPlayer mini.

La programación entre el Arduino Mega y el módulo de audio para la reproducción de sonidos se basa en una secuencia de procesos, donde el dato recibido corresponde al audio que se debe reproducir, en la figura 77 se muestra el diagrama de proceso para esta etapa.

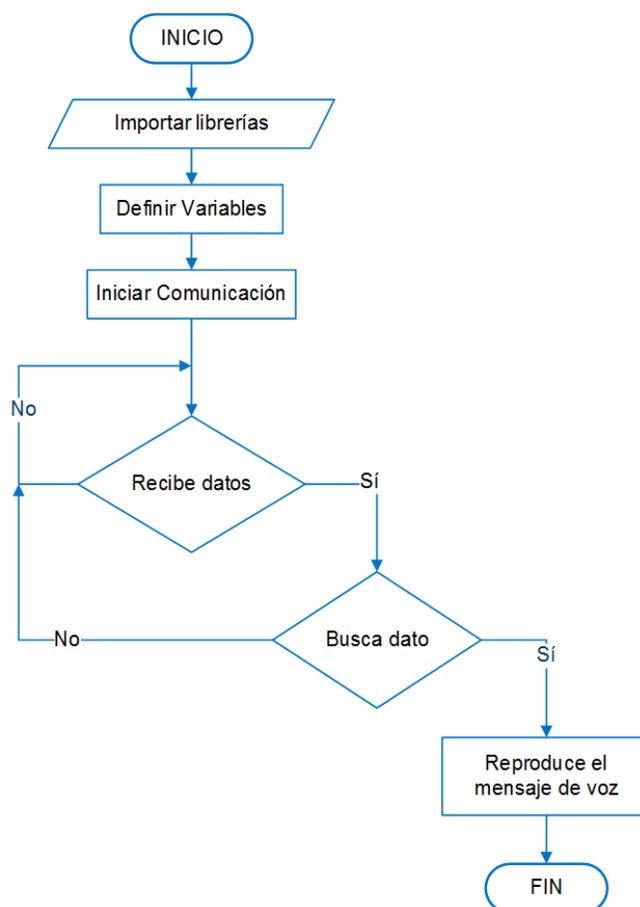


Figura 77 Proceso para reproducir un audio

En la tabla 30 se muestra los comandos más utilizados para programar el módulo de audio DFPlayer mini Mp3.

Tabla 30
Lista de comandos del módulo DFPlayer mini Mp3

COMANDO	FUNCIÓN
0x03	Selecciona una canción de la carpeta (0-2999)
0x06	Especifica el volumen (0-30)
0x0E	Pause
0x0D	Play
0x01	Siguiente
0x02	Anterior

En la figura 78, las funciones `void pause()` se encarga de pausar el audio que se está reproduciendo, `void play()` reproduce nuevamente el audio pausado, `void playNext()` reproduce el audio posterior y `void playPrevious()` cumple la tarea de reproducir el audio anterior.

```

void pause()
{
  execute_CMD(0x0E,0,0);
  delay(500);
}

void play()
{
  execute_CMD(0x0D,0,1);
  delay(500);
}

void playNext()
{
  execute_CMD(0x01,0,1);
  delay(500);
}

void playPrevious()
{
  execute_CMD(0x02,0,1);
  delay(500);
}

```

Figura 78 Varios comandos básicos

En el anexo E-1, se muestra el programa completo en arduino, tanto de la configuración de las variables como de las funciones de control de selección, cambio y de volumen, para la reproducción de los sonidos.

La función `void SetVolume()` cumple la tarea de configurar el volumen de las canciones cuyo rango va de 0-30 (mínimo - alto), mientras que la función `void playsing1()` se encarga de seleccionar una sola canción de la carpeta para posteriormente reproducirla, cabe destacar que la carpeta puede contener de 1 hasta 255 canciones para escogerlas. (ver figura 79).

```

void setVolume(int volume)
{
  execute_CMD(0x06, 0, volume);
  delay(2000);
}

void playsingl ()
{
  execute_CMD(0x03,0,1);
  delay(500);
}

```

Figura 79 Volumen y selección de canción

B. Diagrama de conexión entre Arduino y matriz de leds

La matriz de leds 8x16 MBot recibe señales del Arduino Mega a sus parámetros de entrada SCK (Pin de módulo de control de línea de reloj) y DIN (Pin de módulo de control de línea de datos) mediante un programa en Arduino con el fin de poder mostrar gestos o emociones a través del encendido de sus diodos led. A continuación, se muestra la conexión Arduino Mega y matriz led (ver figura 80).

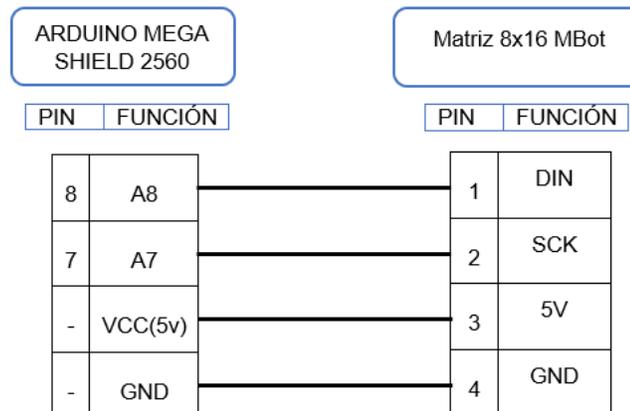


Figura 80 Conexión Arduino Mega y Matriz led

En el anexo D-5 se puede observar el circuito electrónico de la conexión entre Arduino y la matriz de leds 8x16.

El algoritmo simple para mostrar un gesto y visualizar en la matriz de leds se detalla mediante un diagrama de flujos en la figura 81, los gestos de las emociones se encuentran definidos acorde a las expresiones como: felicidad, tristeza, impresión, indiferencia, alegría y sonriente, estos gestos se van actualizando acorde a los datos que recibe de la RA.

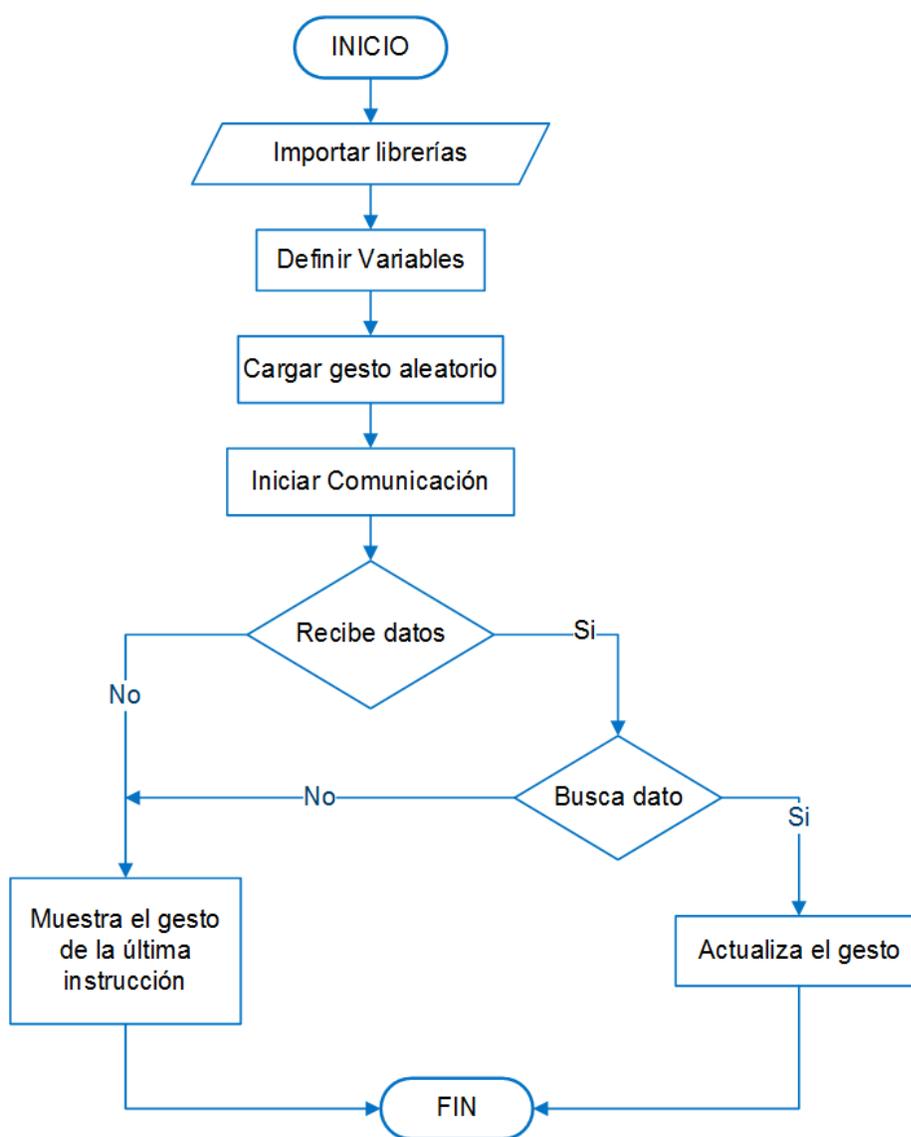


Figura 81 Proceso imprimir gesto en la matriz de led

En la tabla 31 se muestra los comandos más utilizados para programar la matriz de leds 8x16 mBot,

Tabla 31
Lista de comandos matriz led mBot

COMANDO	FUNCIÓN
clearDisplay	Limpiar la pantalla de la matriz
setRow	Establecer fila

En la figura 82 para configurar la matriz de leds se define las librerías y los pines de salida de la matriz de leds 7 y 8 (data - clock) para conectarlos físicamente en la tarjeta Arduino Mega.

```
#include <TM1638.h>
#include <TM1640.h>
TM1640 module(7, 8);
```

Figura 82 Librerías y pines de salida Matriz de leds

La variable `boolean` cumple la función de mantener apagado en un inicio todos los leds de la matriz para luego encenderlos mediante el comando `flip`, además la variable entera `speed` regula la velocidad de encendido de los diodos led (ver figura 83).

```
void draw(boolean flip= false);
int speed = 1000;
```

Figura 83 Conexión Arduino Mega y Matriz led

En la figura 84, la función `void draw()` se encarga de encender y apagar ciertos leds con el fin de visualizar gestos o emociones que se hayan programado. Para ello se dispone de dos variables tipo `byte color` que se encargan de controlar todos los leds de la matriz. El comando `module.ClearDisplay()` limpia el código anterior para visualizar el nuevo código. Mientras que el comando `module.setRow()` cumple la función de encender los diodos leds cada vez que sean verdaderos y apagar los leds cada

vez que sean falsos, teniendo en cuenta que se debe utilizar la variable color 1 y color 2 para controlar todos los leds de la matriz. En este caso como se dispone de una matriz de 8x16 se coloca por cada columna o dígitos ya sean “0” leds apagados o “1” leds encendidos.

```
void draw(boolean flip)
{
  byte color1 = 1;
  byte color2 = 2;

  module.clearDisplay();
  module.setRow(color1,0,0b00000000);
  module.setRow(color1,1,0b00000000);
  module.setRow(color1,2,0b00000000);
  module.setRow(color1,3,0b00000010);
  module.setRow(color1,4,0b00000110);
  module.setRow(color1,5,0b00001010);
  module.setRow(color1,6,0b00010010);
  module.setRow(color1,7,0b00010010);

  module.setRow(color2,0,0b00010010);
  module.setRow(color2,1,0b00010010);
  module.setRow(color2,2,0b00001010);
  module.setRow(color2,3,0b00000110);
  module.setRow(color2,4,0b00000010);
  module.setRow(color2,5,0b00000000);
  module.setRow(color2,6,0b00000000);
  module.setRow(color2,7,0b00000000);
}
```

Figura 84 Conexión Arduino Mega y Matriz led

En el anexo E-2, se presenta el programa completo en arduino del control de la matriz de leds para la simulación de los gestos emocionales.

C. Diagrama de conexión entre Arduino y módulo de motores

En el módulo Shield L293D, los cuatro pines digitales activan o desactivan las entradas del control y la velocidad de los motores de corriente continua. De acuerdo a las señales enviadas por la programación de la tarjeta Arduino principal. A continuación, se muestra la conexión entre Arduino Mega y el módulo Shield L293D (ver figura 85).

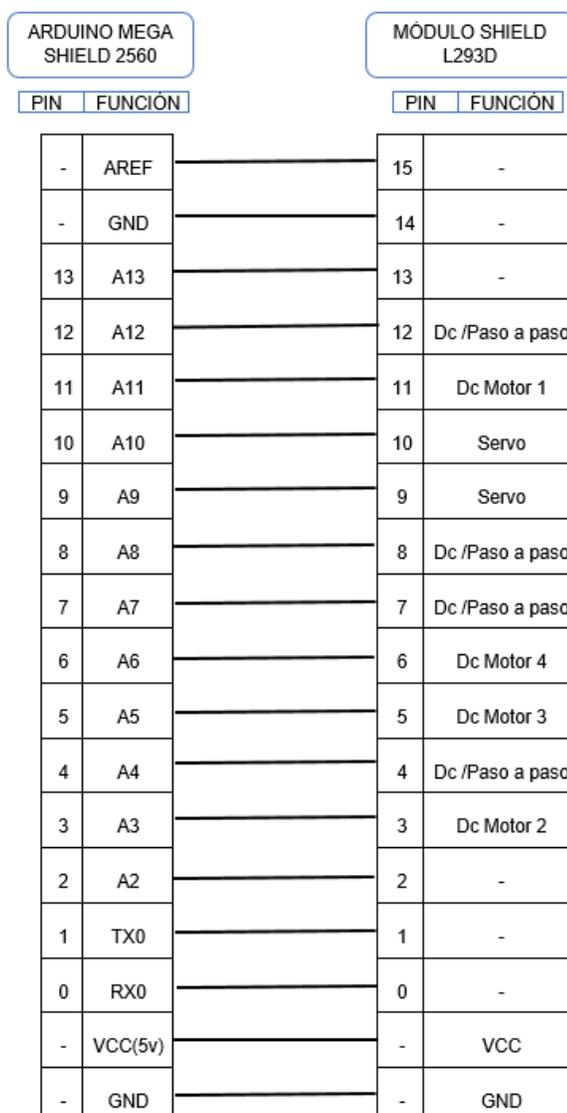


Figura 85 Conexión Arduino Mega y el módulo de motores

En el anexo D-6, se puede observar el circuito electrónico de la conexión entre Arduino y el módulo de motores Shield L293D.

En cuanto a la programación del módulo, en la figura 86 se muestra un diagrama de flujo que describe el proceso de control de los motores, este proceso se basa en recibir un dato el cual indica la secuencia de movimiento que debe realizar el robot mediante la sincronización de los motores, por lo tanto, si el dato

que recibe no se encuentra en la base de datos no se sincroniza ningún motor y continúa recibiendo los datos.

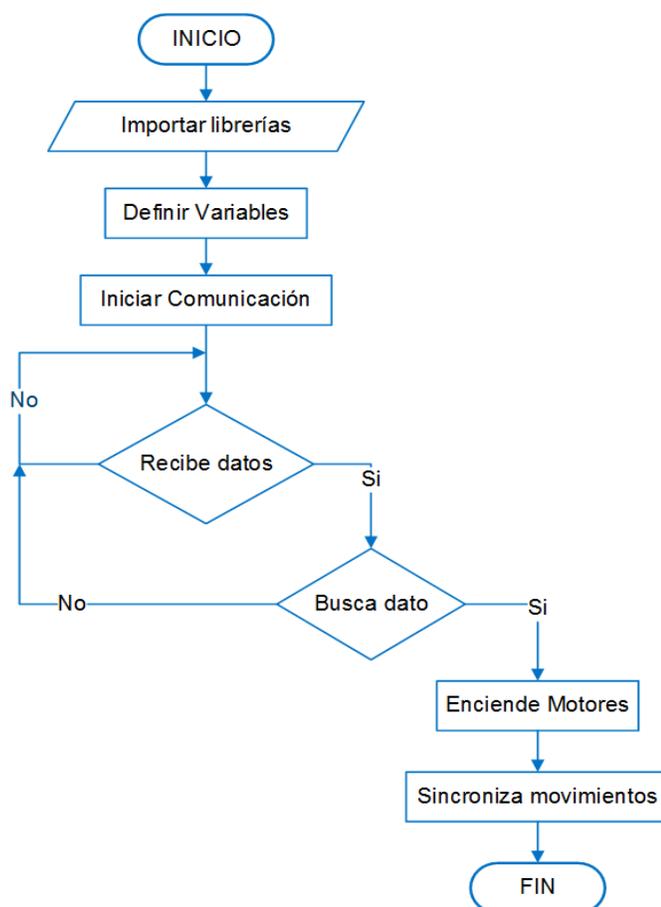


Figura 86 Proceso para controlar motores

En la tabla 32, se muestra los comandos más utilizados para la programación del módulo Shield L293D.

Tabla 32
Lista de comandos del módulo Shield L293D

COMANDO	FUNCIÓN
setSpeed	Determina la velocidad del motor 0-255
run(FORWARD)	Motor hacia adelante
run(BACKWARFD)	Motor hacia atrás
run(RELEASE)	Motor detenido

En la figura 87, para configurar el módulo de motores se debe cargar la librería `AFMotor.h`, seguido se da nombres a las variables que se va a utilizar anteponiendo el código `AF_DCMotor`. Se debe de tener en cuenta que la tarjeta Shield dispone de 4 canales para el control de motores.

```
#include <AFMotor.h>
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
```

Figura 87 Librerías del módulo de motores

El comando `motor.setSpeed()`, determina la velocidad inicial en rpm de los motores (ver figura 88).

```
motor1.setSpeed(250);
motor2.setSpeed(250);
```

Figura 88 Comando que determina la velocidad de los motores

En la figura 89, la función `void motor()` está compuesta por varios comandos como lo es el `run(FORWARD)` que sirve para hacer girar el motor hacia adelante, y el `setSpeed()` que determina la velocidad en rpm del motor, por último, se tiene el comando `delay()` que se encarga de dar una pausa en milisegundos a la función luego de haber ejecutado los comandos del paréntesis.

```
void motor() {
    motor1.run(FORWARD);
    motor1.setSpeed(150);
    motor2.run(FORWARD);
    motor2.setSpeed(150);
    delay(5000);
}
```

Figura 89 Comandos para mover el motor

En el anexo E-3, se muestra el programa completo para el control de la posición y velocidad de los motores DC.

D. Conexión general maestro-esclavo

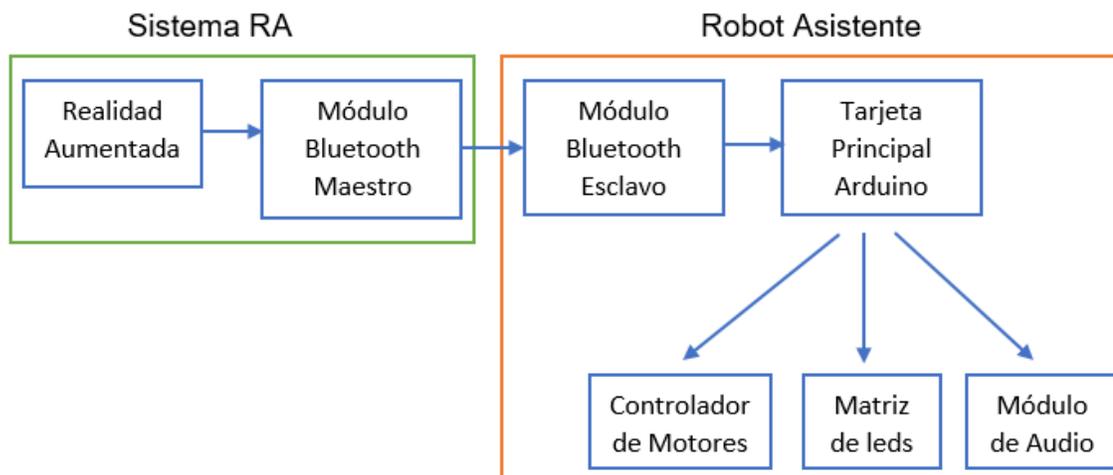


Figura 90 Diagrama de la comunicación maestro - esclavo

En el diagrama de la figura 90, se muestra más a detalle como da la comunicación entre el sistema de realidad aumentada con la tarjeta principal del robot. El módulo de bluetooth maestro es parte del sistema de Realidad aumentada el cual se encarga recibir los datos que envía el sistema RA, estos datos son verificados para saber si no son datos falsos, una vez verificados los datos son enviados de manera inalámbrica al módulo bluetooth esclavo que se encuentra instalado en la tarjeta Arduino del robot, esta información es procesada para determinar si el dato recibido corresponde a una instrucción, si el dato corresponde a una de las instrucciones se ejecuta la acción pertinente, en este caso las acciones que puede realizar el robot son: sincronización de motores para ejecución de movimientos, impresión de gestos en la pantalla matriz leds o reproducir un sonido de audio.

En el anexo E-4 se puede encontrar el código completo del módulo maestro, así como también el código de programación del módulo bluetooth esclavo.

3.3.2. Consumo de la energía eléctrica del robot.

Para la alimentación de todo el sistema del robot se utilizan un paquete de 6 pilas doble A recargables de 1,2 V y 4700 mah cada una, conectadas en serie dan un valor de 7,2 V y 4700 mah.



Figura 91 Pilas recargables doble A 4700mah

En la tabla 33, se muestra las ventajas de desventajas de utilizar pilas recargables como fuente de alimentación.

Tabla 33

Ventajas y desventajas de utilizar pilas recargables.

Ventajas	Desventajas
Ahorro económico a largo plazo	Inversión inicial mayor.
Portables, adaptables a casi cualquier espacio de trabajo	Requieren un conocimiento de uso
Alta duración, las pilas actuales de níquel-metal posee una capacidad de carga mayor.	Son sensibles a elevadas temperaturas.
Amigable con el medio ambiente	Se debe seguir un procedimiento adecuado para reciclar estas pilas.
Se puede obtener los voltajes deseados adicionándole o quitándole pilas a la fuente en serie.	Individualmente poseen, menor tensión que una pila batería convencional, tienen 1,2 V.

En la tabla 34, se muestra el consumo eléctrico de cada uno de los componentes electrónicos que se utilizan en el proyecto.

Tabla 34
Consumo de cada componente electrónico

Componentes	mA
Motor de 6 v x3	750 mA
Arduino Mega	93 mA
Módulo Audio	20 mA
Matriz led	5 mA
Parlantes	0.8 mA
Luces led	20 mA
Módulo Maestro HC-05	50 mA
Módulo Esclavo HC-06	50 mA
Módulo motor Shield	65 mA
TOTAL	1053.8 mA

En base a los datos anteriores se realiza el siguiente cálculo para determinar cuánto tiempo van a durar las pilas.

$$\frac{4700 \text{ mA/H}}{1053.8 \text{ mA}} = 4.46 \text{ H}$$

Una vez realizado el cálculo se puede deducir, que empleando unas pilas doble A de 4700 miliamperios se va a tener 4 horas de funcionamiento óptimo, tiempo más que suficiente, puesto a que el promedio de uso diario según los resultados de la encuesta (revisar anexo A-2) es de 3 horas diarias, por lo tanto, las baterías suplen a cabalidad este requerimiento energético.

En el siguiente capítulo, se realiza el diseño de las interfaces de los juegos de realidad aumentada, y se establece la comunicación inalámbrica para la transmisión de datos, entre el sistema de realidad aumentada y el robot.

CAPÍTULO IV

REALIDAD AUMENTADA Y TRANSMISIÓN DE DATOS

4.1. Introducción

En este capítulo se realiza el diseño de la interfaz de los juegos de realidad aumentada y el sistema de transmisión de datos para la comunicación entre la interfaz y el robot didáctico, la interfaz de realidad aumentada es el sistema que se ejecuta en primera instancia y en base al cual los demás sistemas se activan dependiendo de los procesos que se ejecuten en los juegos, estos juegos están orientados a la educación y el entretenimiento de los niños y niñas de educación básica, por lo cual el diseño estético es uno de los parámetros importantes puesto que a ellos comúnmente les llama la atención imágenes que posean una combinación de colores vivos, alegres, y por supuesto poseer también contenido educativo que permitan nutrir el conocimiento y estimular el aprendizaje.

El proceso general del proyecto se ejecuta en base al sistema de realidad aumentada (ver figura 92), la interacción con los juegos genera datos a manera de instrucciones, estos datos son enviados mediante una comunicación inalámbrica hacia el robot didáctico, el cual recibe las instrucciones y las ejecuta, este posee dos modos de uso modo instrucción y modo control manual.

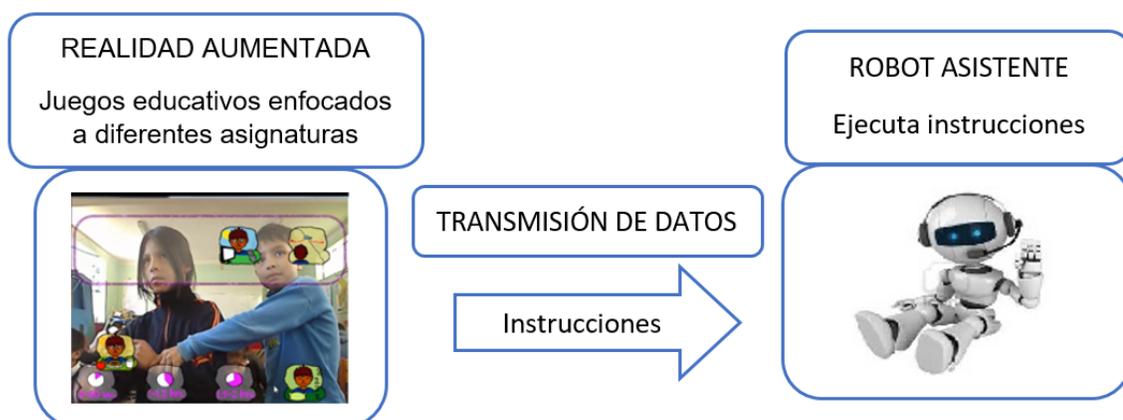


Figura 92 Descripción de los sistemas

En el modo instrucción, el robot sincroniza los sonidos de audio con gestos de boca que simula display, para generar instrucciones de voz, mientras que en el modo control manual los datos recibidos se sincronizan con movimientos de motores para generar desplazamiento y giros. En la figura 93 se muestra el diagrama de flujo que describe el proceso en general del proyecto.

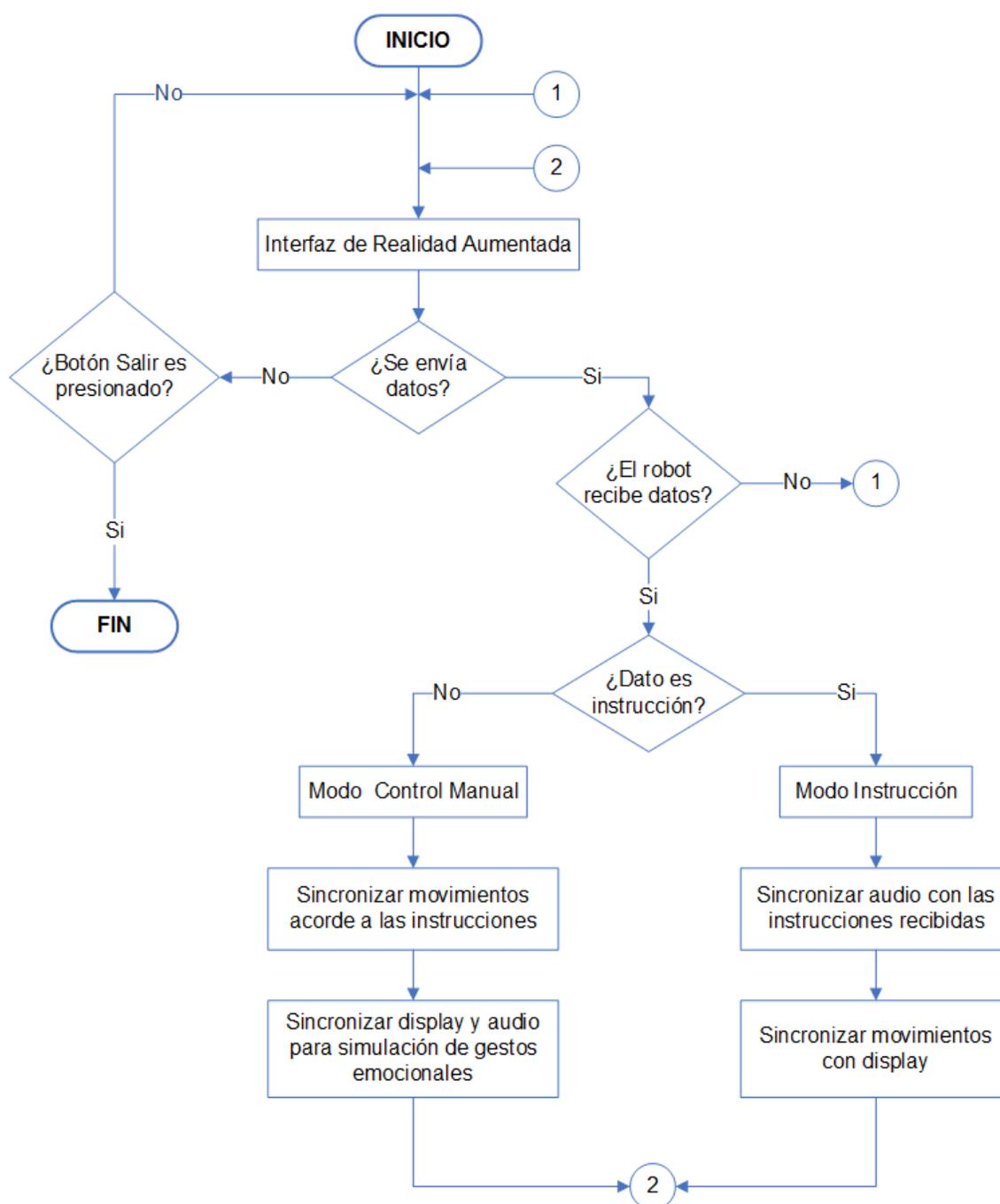


Figura 93 Diagrama de Flujo del Sistema Completo

4.2. Interfaz de juegos de realidad aumentada

Los juegos de realidad aumentada se van a orientar en 4 secciones:

- **Aprendizaje.** Estos juegos se orientan a reforzar el conocimiento, a partir de juegos de matemáticas, lenguaje, saberes ancestrales, entre otros.
- **Entretenimiento.** Tienen el objetivo de entretener y reducir el estrés que los niños y niñas enfrentan en las aulas de clase.
- **Razonamiento.** En esta sección entran juegos que permitan potenciar las habilidades mentales, es decir ejercitar la mente para el mejoramiento de la memoria, comprensión verbal entre otros beneficios.
- **Robótica.** Esta sección comprende juegos que contengan información sobre realidad aumentada y además contenido que permita introducir a los estudiantes al mundo de la robótica con juegos sobre partes principales del robot y las funciones que cumplen.

Cada uno de los juegos además de priorizar en el contenido educativo, debe resaltar con una interfaz atractiva y fácil de entender, para el desarrollo de los juegos se tomaron como referencia los temas mencionados en los libros de educación básica que proporciona el gobierno ecuatoriano.

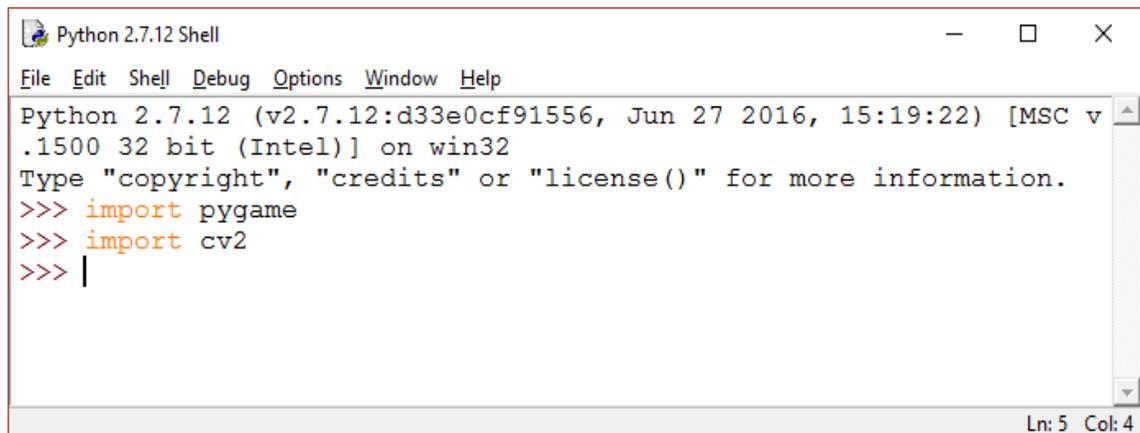
4.2.1. Softwares necesarios

El conjunto de softwares a utilizar son Pygame-OpenCV como librerías principales y como entorno de programación Python, estos softwares fueron seleccionados en el capítulo 2 (sección 2.3.4. software), por cumplir con todos los requerimientos exigentes para el desarrollo de los juegos, cada uno de estos softwares son de licencia libre y multiplataforma por lo tanto se pueden descargar de manera gratuita desde las páginas oficiales e instalar en cualquier sistema operativo Windows, Linux o Mac OS.

Para el desarrollo de los juegos se emplea Python 2.7.12, Pygame 1.9.1, OpenCV 2.4.9 por ser las librerías y software más estables, si se desea utilizar otras versiones se debe tener en cuenta que la versión de Pygame debe ser

compatible con la versión de Python para que no existan errores al momento de importar las librerías, como sistema operativo se utiliza Windows 10, debido a la compatibilidad del sistema operativo con el mini pc de control.

El proceso de instalación de los softwares varia con respecto al sistema operativo en el que se pretenda utilizar, para este caso existen tutoriales de instalación de cada uno de los softwares para los diferentes sistemas operativos, en el caso de Windows, la instalación resulta fácil, el proceso de instalación de los programas mencionados se puede consultar en el anexo F, una vez instalados los programas se debe abrir Python y verificar si los módulos principales se pueden importar, en la figura 94, se muestra la interfaz de trabajo de Python con los módulos de Pygame y Opencv importados.



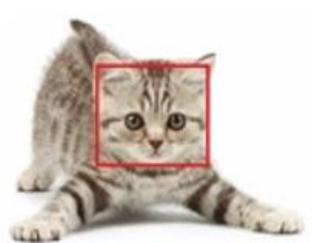
```
Python 2.7.12 Shell
File Edit Shell Debug Options Window Help
Python 2.7.12 (v2.7.12:d33e0cf91556, Jun 27 2016, 15:19:22) [MSC v
.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import pygame
>>> import cv2
>>> |
```

Ln: 5 Col: 4

Figura 94 Entorno de programación de Python

4.2.2. Reconocimiento de la mano cerrada

Antes de empezar a desarrollar los juegos se realiza el reconocimiento de la mano cerrada que va a representar el marcador, elemento que permite la interacción con los elementos virtuales. El software OpenCV cuenta con clasificadores en cascada entrenados, los cuales se encuentran en el directorio `opencv\sources\data\haarcascades`, estos clasificadores sirven para detectar rostros humanos, ojos, nariz, boca, placas de autos, rostros de gatos, entre otros, en la figura 95 se muestran algunos ejemplos de reconocimiento de objetos.



(a) Rostro de gato



(b) Rostro de persona



(c) Reconocimiento de auto

Figura 95 Reconocimiento de objetos

Fuente: (Marin, 2015)

Además, OpenCV cuenta con herramientas que facilitan el entrenamiento de clasificadores personalizados, en el caso que se requiera realizar el reconocimiento de otro tipo de objeto, en el directorio `opencv\build\x86\vc11\bin` se encuentran los archivos ejecutables “`opencv_createsamples.exe`” y “`opencv_traincascade.exe`” que sirven de ayuda para el trabajo de entrenamiento de manera sencilla.

El reconocimiento de la mano cerrada se va a realizar en dos etapas, la primera etapa consiste en generar un archivo xml mediante un proceso de entrenamiento de clasificador, este clasificador al final va a contener los rasgos característicos que definen a la mano cerrada, que permita reconocerlo y diferenciarlo de las demás partes del cuerpo humano, la segunda etapa consiste en la detección de la mano cerrada mediante un programa que permita utilizar este clasificador xml y verificar su eficacia.

a. Etapa 1: Entrenamiento del clasificador

Lo primero que se debe hacer es generar una estructura de directorio (ver figura 96) para el alojamiento de las imágenes recolectadas y los archivos resultantes que se obtienen en los subprocesos del entrenamiento del clasificador.



Figura 96 Estructura de directorio

El proceso de entrenamiento se realiza mediante 4 pasos:

- **Paso 1: Adquisición de imágenes positivas y negativas**

Este paso consiste en generar una base de datos de imágenes positivas y negativas, las cuales se colocan en las carpetas correspondientes del directorio, teniendo en cuenta que las imágenes negativas son cualquier imagen en la cual no aparece el objeto de interés, y las imágenes positivas son las que contienen el objeto que se trata de reconocer (ver figura 97).



(a) Imágenes Positivas

(b) Imágenes Negativas

Figura 97 Adquisición de imágenes

Es recomendable utilizar como mínimo 200 imágenes positivas y negativas para obtener resultados aceptables, para este proyecto se utilizan un total de 400 imágenes positivas, estas imágenes se obtienen mediante fotografías tomadas de la mano cerrada de niños y niñas.

- **Paso 2: Organizar las imágenes positivas y negativas**

Se debe crear un archivo (negativas.txt) en la carpeta entrenamiento, este archivo sirve de índice para las etapas posteriores (ver figura 98).

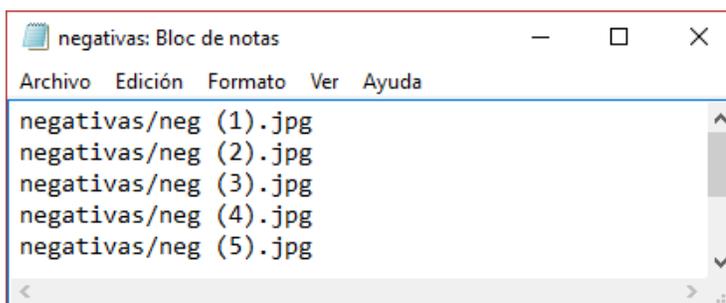


Figura 98 Índice de imágenes negativas

Para las imágenes positivas de igual manera se crea un archivo (positivas.txt), pero este archivo contiene algo más de información (ver figura 99). Este archivo contiene información específica de donde se encuentra el objeto dentro de la imagen positiva, los dos primeros números indican las coordenadas (x, y) del punto superior izquierdo de inicio del objeto, mientras que los dos siguientes el ancho y el largo

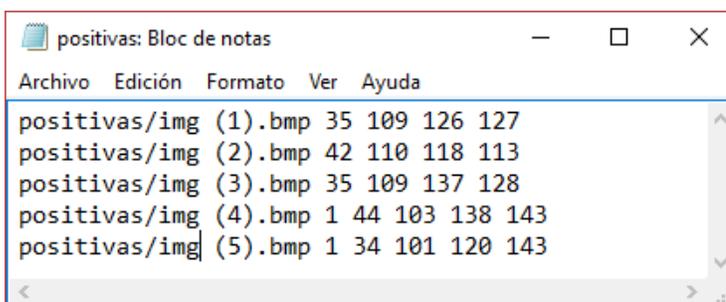


Figura 99 Índice de imágenes positivas

Para generar este archivo “positivas.txt”, OpenCV cuenta con una herramienta llamada “objectmarker”, que permite marcar en un cuadro la zona donde se encuentra el objeto a detectar, para obtener las coordenadas y el tamaño, en este caso se marca la zona de la mano cerrada dentro de la imagen, esto se realiza para cada una de las imágenes positivas (ver figura 100).

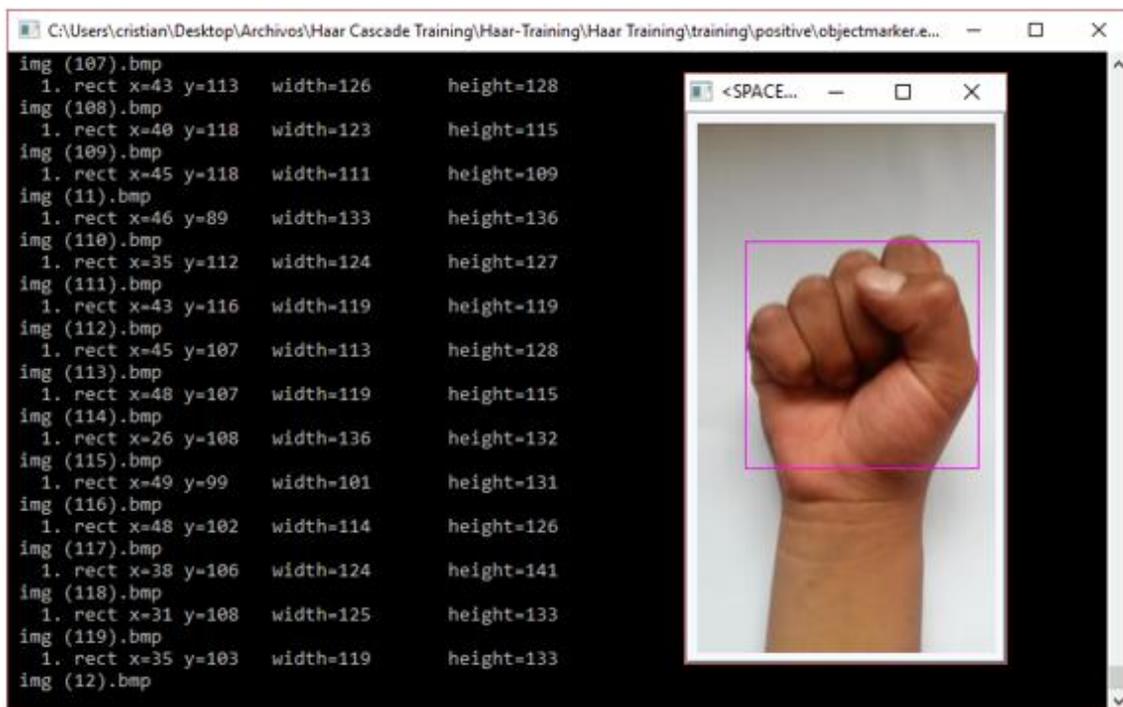


Figura 100 Obtención de coordenadas de la mano mediante Objectmarker

- **Paso 3: Crear muestra**

En este paso, es necesario la utilización de la herramienta llamada “createsamples” de OpenCV, para lo cual se debe ejecutar la línea de comando de la figura 101 en la ventana de comandos de Windows:

```
opencv_createsamples.exe -info positivas.txt -vec data/muestra.vec -num
400 -w 20 -h 20
```

Figura 101 Línea de comando para creación de muestras

donde:

- info**: es la ubicación del archivo con el índice las imágenes positivas.
- vec**: es el nombre del archivo de salida con la muestra generada.
- num**: cantidad de imágenes positivas.
- w**: ancho de la muestra de salida.
- h**: alto de la muestra de salida.

```
c:\CvFile2>opencv_createsamples.exe -vec pos.vec -info face_2429_info.txt -bg non_face_4548_info.txt -num 2429 -w 19 -h 19
Info file name: face_2429_info.txt
Img file name: <NULL>
Vec file name: pos.vec
BG file name: non_face_4548_info.txt
Num: 2429
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 19
Height: 19
Create training samples from images collection...
Done. Created 2429 samples
```

Figura 102 Creación de muestras

- **Paso 4: Entrenamiento**

La herramienta necesaria en este paso se llama “haartraining”, se requiere del archivo negativas.txt y del archivo muestra.vec generado anteriormente, otra vez desde la línea de comandos:

```
Opencv_haartraining.exe -data data/hand -vec data/muestra.vec -bg negativas.txt -nstages 30 -nsplit 2 -minhitrate 0.999 -maxfalsealarm 0.5 -npos 400 -nneg 400 -w 20 -h 20 -mem 1300 -mode ALL
```

Figura 103 Línea de comando para entrenamiento del clasificador

donde:

- data**: directorio de salida para la generación del entrenamiento, en este caso el archivo resultante es el clasificador “hand.xml”

- vec**: archivo con muestras generado en la etapa anterior.
- bg**: archivo índice con las imágenes negativas.
- nstages**: número de etapas de entrenamiento.
- nsplit**: debilidad del clasificador.
- minhitrate**: mínimo rango de cada etapa.
- maxfalsealarm**: rango máximo de falsas alarmas para cada etapa.
- npos**: número de imágenes positivas indexadas.
- nneg**: número de imágenes negativas indexadas.
- w**: ancho.
- h**: alto.
- mem**: memoria a utilizar en el proceso.

```

C:\windows\system32\cmd.exe - opencv_traincascade.exe -data mydat...
C:\>cd CvFile
C:\CvFile>opencv_traincascade.exe -data mydata -vec pos.vec -bg non_face_4548_in
fo.txt -numPos 2000 -numNeg 4000 -numStages 20 -featureType HAAR -w 19 -h 19 -mo
de ALL
PARAMETERS:
cascadeDirName: mydata
vecFileName: pos.vec
bgFileName: non_face_4548_info.txt
numPos: 2000
numNeg: 4000
numStages: 20
precalcValBufSize[Mb] : 256
precalcIdxBufSize[Mb] : 256
stageType: BOOST
featureType: HAAR
sampleWidth: 19
sampleHeight: 19
boostType: GAB
minHitRate: 0.995
maxFalseAlarmRate: 0.5
weightTrimRate: 0.95
maxDepth: 1
maxWeakCount: 100
mode: ALL

==== TRAINING 0-stage ====
<BEGIN
POS count : consumed 2000 : 2000
NEG count : acceptanceRatio 4000 : 1
Precalculation time: 8.718
+-----+
| N | HR | FA |
+-----+
| 1 | 1 | 1 |
+-----+
| 2 | 1 | 1 |
+-----+
| 3 | 0.998 | 0.374 |
+-----+
END>
Training until now has taken 0 days 0 hours 2 minutes 18 seconds.

```

Figura 104 Entrenamiento del clasificador

b. Etapa 2: Detección de la mano cerrada

Esta etapa de reconocimiento consiste en aplicar un detector sobre las imágenes de video, en este caso se utiliza el clasificador de la mano cerrada entrenada “hand.xml” para encontrar la posición donde se localiza la mano, esta etapa consiste de 2 pasos.

- **Pre-procesado de imágenes**

Una de las características del detector es que trabaja con imágenes en gris, por lo tanto, una de las primeras operaciones es convertir las imágenes a escala de grises (ver figura 105).



(a) Imagen original



(b) Imagen en escala de grises

Figura 105 Conversión de imagen a escala de grises

La siguiente operación es filtrar la imagen por medio de un filtro de convolución, que permite eliminar el ruido que puede contener las imágenes (ver figura 106).



(a) Imagen en escala de grises



(b) Imagen suavizada con filtro

Figura 106 Filtrado de imagen para eliminación de ruido

- **Detección de la mano**

En este paso se utiliza una función de OpenCV llamado `detectMultiScale()`, esta función es básicamente un algoritmo detector que trabaja en torno al clasificador entrenado, este detector empieza reduciendo la imagen a diferentes escalas de tamaño para tener más posibilidades de encontrar la mano, luego extrae ventanas de un tamaño determinado de cada una de las imágenes escaladas y las pasa al clasificador, el clasificador evalúa las características de cada ventana, aquellas ventanas que posean características similares a los rasgos de la mano cerrada se clasifican como ventanas positivas y se extrae la posición y el tamaño para ser almacenados en un vector, caso contrario las ventanas se clasifican como negativas y se desechan, en la figura 107 se muestra el funcionamiento del algoritmo detector.

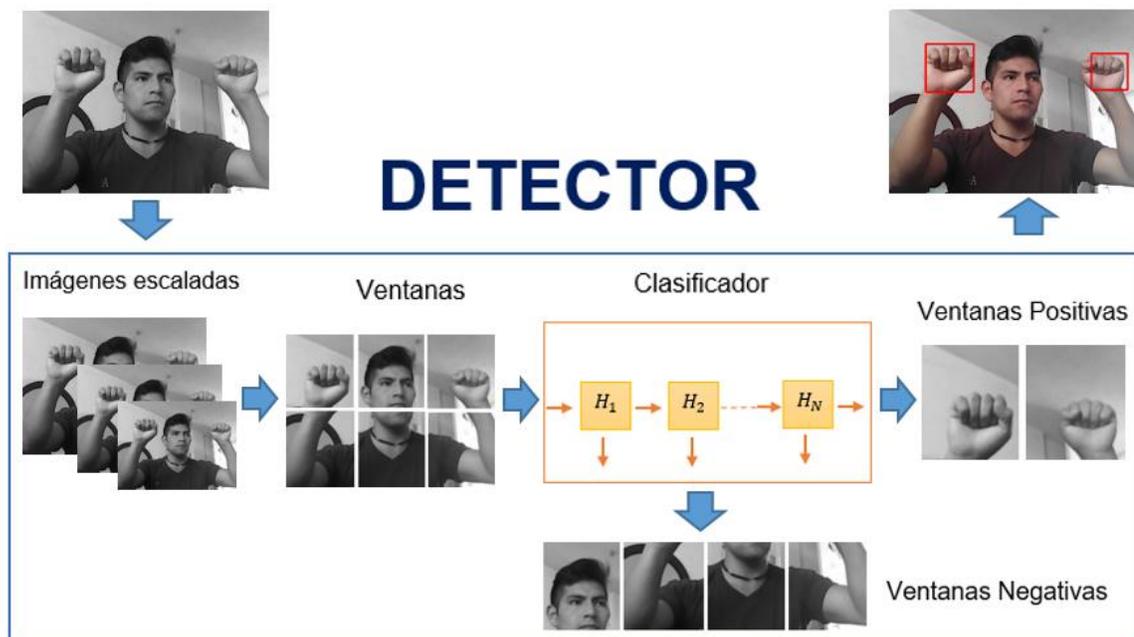


Figura 107 Funcionamiento del detector

El detector cumple las expectativas, encuentra las ventanas que contienen las características de las manos cerradas, el programa completo para la detección se llama "DeteccionMano.py", se puede revisar en el anexo E-5.

4.2.3. Diseño de los algoritmos

Los juegos de realidad aumentada se van a desarrollar en base a una secuencia de procesos, secuencia que se ilustra mediante un diagrama de flujo en la figura 108.

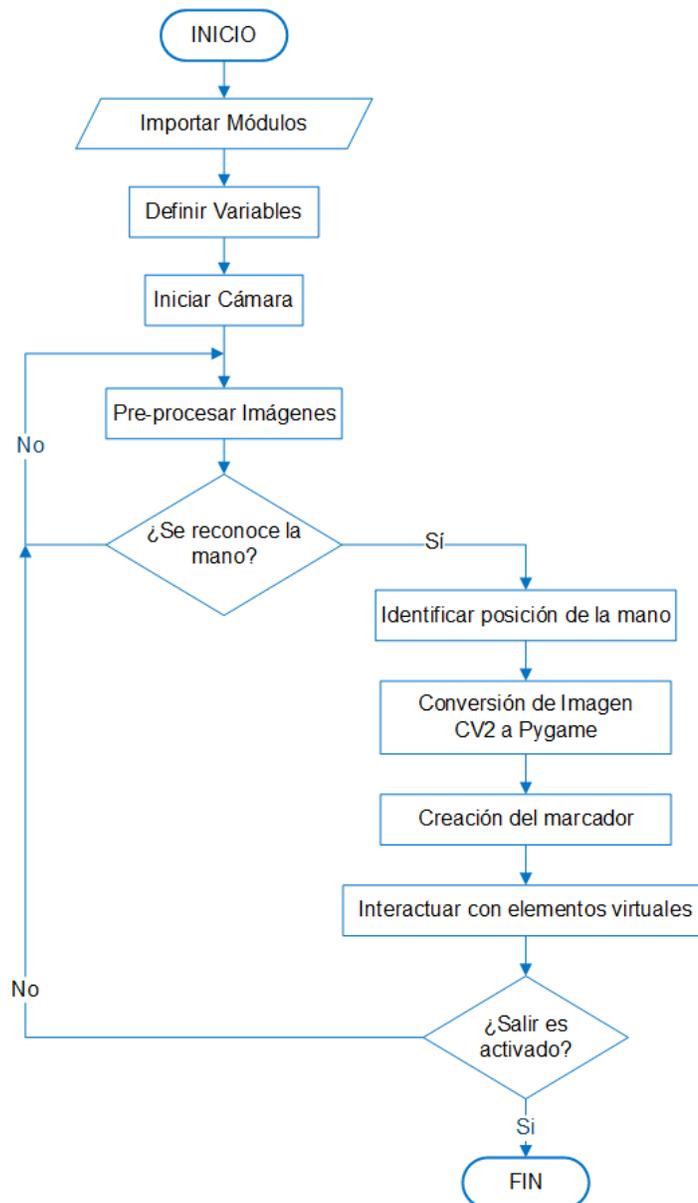


Figura 108 Proceso General de los juegos RA

En la tabla 35, se muestra la descripción de los algoritmos más importantes que se emplean para la creación de los juegos.

Tabla 35**Principales algoritmos de pygame y opencv para desarrollo de juegos RA.**

Código	Descripción
<code>import</code>	Permite importar módulos como por ejemplo los módulos de pygame o librerías de opencv.
<code>cv2.CascadeClassifier()</code>	Permite cargar un clasificador Haar entrenado.
<code>cv2.VideoCapture()</code>	Inicializa la cámara de video.
<code>cap.read()</code>	Permite obtener las imágenes capturadas por medio de la cámara.
<code>file.detectMultiScale()</code>	Permite detectar los patrones característicos de un objeto en particular.
<code>cv2.cvtColor()</code>	Permite cambiar el espacio de color de las imágenes, por ejemplo, transformar a escala de grises
<code>cv2.GaussianBlur()</code>	Permite suavizar las imagines y eliminar el ruido de las imagines.
<code>pygame.init()</code>	Inicializa todos los módulos importados de pygame.
<code>pygame.display.set_mode()</code>	Configura una ventana o pantalla para mostrar.
<code>pygame.image.load()</code>	Permite cargar una imagen de un archivo
<code>pygame.Rect()</code>	Permite crear un objeto mediante coordenadas rectangulares.
<code>pygame.sprite.Sprite()</code>	Define un objeto de tipo "Sprite", un sprite está compuesto de una imagen y una posición además permite realizar operaciones de colisión.
<code>pygame.draw.rect()</code>	Dibuja un objeto en forma de rectángulo
<code>pygame.time.Clock()</code>	Crear un objeto para ayudar a controlar el tiempo
<code>pygame.font.SysFont()</code>	Permite crear un objeto de tipo fuente desde las fuentes del sistema
<code>display.blit()</code>	Dibuja una imagen en la pantalla.
<code>pygame.display.update()</code>	Actualiza la pantalla
<code>pygame.event.get()</code>	Permite obtener los eventos del teclado y mouse.
<code>pygame.quit()</code>	Permite salir o quitar los módulos inicializados.

A continuación, se describe cada uno de los subprocesos del diagrama de flujos de la figura 108, tomando como referencia uno de los juegos de realidad aumentada que se ha desarrollado llamado “Números Naturales”, es un juego de matemáticas que consiste en seleccionar el número correspondiente al número de objetos que se presentan, este juego tiene el objetivo de enseñar a contar e identificar los números naturales.

- a. **Importar módulos.** Se hace un llamado de las librerías más importantes que se van a utilizar, en este caso las librerías de pygame y opencv (cv2) son indispensables para el desarrollo de los juegos (ver figura 109).

```
# Módulos
import pygame
import cv2
import cv2.cv as cv
import random
import numpy
import numpy as np
from pygame.locals import *
```

Figura 109 Importación de módulos

- b. **Definir variables.** Estas variables tienen la función de almacenar los valores de los parámetros que sirven como argumento para la aplicación, la ventana de trabajo se configura a 1280x720 pixeles (ver figura 110).

```
# Constantes
SIZE = (1280, 720) #tamaño de la ventana
# Colores
verde = (0,255,0)
rojo = (255,0,0)
azul = (0,0,255)
# Datos Mano
hand = cv2.CascadeClassifier("hand.xml")
#Fuentes
smallfont = pygame.font.SysFont("comicsansms", 25)
medfont = pygame.font.SysFont("comicsansms", 50)
largefont = pygame.font.SysFont("comicsansms", 85)
# Cargar imagenes
image1 = pygame.image.load(config.imagenes+"1caja.png")
image2 = pygame.image.load(config.imagenes+"2mariposas.png")
```

Figura 110 Definición de variables

- c. **Iniciar cámara.** Se inicia los parámetros para abrir la cámara y poder obtener las secuencias de imágenes de video (ver figura 111), en este paso también se configura la resolución de la cámara a (864x480) pixeles.

```
# Puerto y Resolucion de Camara
cap = cv2.VideoCapture(0)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_WIDTH, 864)
cap.set(cv2.cv.CV_CAP_PROP_FRAME_HEIGHT, 480)
```

Figura 111 Inicialización de cámara de video

- d. **Pre-procesar imágenes.** Se realiza la captura de la imagen, la primera operación es la transformación de a escala de grises y luego se aplica un filtro gaussiano para eliminación de ruido, como la imagen que se obtiene esta girada se voltea la imagen 180 grados, las operaciones realizadas se muestran en la figura 112.

```
# Captura de imagen
ret, cv2_image = cap.read()
# Transformacion a escala de grises
gray = cv2.cvtColor(cv2_image, cv2.COLOR_BGR2GRAY)
# Eliminacion de ruido (filtrado Gaussiano)
blur = cv2.GaussianBlur(gray, (5,5), 0)
# Giro de la imagen a 180 grados
cv2_image = cv2.flip(blur, 1)
```

Figura 112 Procesamiento de imagen

- e. **Reconocimiento de la mano.** Se realiza la detección de la mano mediante el clasificador entrenado, en la figura 113 se muestra el algoritmo para la detección.

```
# Deteccion de mano
handdetect= hand.detectMultiScale(cv2_image, 1.3, 5)
```

Figura 113 Detección de la mano cerrada

- f. Identificación de la posición de la mano.** Se extrae las coordenadas y el tamaño de la zona donde se localiza la mano dentro de las imágenes, esta información se emplea para crear un objeto que sirva como marcador para la interacción con los elementos virtuales (ver figura 114), para obtener las coordenadas (x_hand, y_hand) se realiza una interpolación lineal, puesto a que inicialmente se obtienen las coordenadas de una imagen de 864x480 pixeles, pero la ventana de trabajo real es de 1280x720 pixeles.

```
# Identificacion posicion de la mano
for (x,y,w,h) in handdetect:
    cv2.rectangle(cv2_image, (x,y), (x+w,y+h), (255,0,0), 2)
    (x_hand, y_hand)=((x*1280/864)+w*0.2), ((y*720/480)+h*0.2))
    (w_hand, h_hand) = (w, h)
```

Figura 114 Extracción de posición y tamaño de la mano

- g. Conversión de imagen CV2 a Pygame.** La imagen obtenida después del procesamiento e identificación de las coordenadas, para creación del marcador es un array (matriz) de pixeles, pero pygame trabaja únicamente con superficies, por lo tanto, esta imagen cv2 se debe convertir a una imagen de superficie y luego escalar la imagen al tamaño de la ventana de trabajo (1280x720 pixeles) para poder utilizarla en pygame, el conjunto de algoritmos para realizar esta tarea se muestra en la figura 115.

```
# Conversion imagen opencv a imagen pygame
frame=cv2.cvtColor(cv2_image,cv2.COLOR_BGR2RGB)
frame=numpy.rot90(frame)
pygame_image = pygame.surfarray.make_surface(frame)
pygame_image = pygame.transform.flip(pygame_image,1,0)
pygame_image = pygame.transform.scale(pygame_image,(1280,720))
```

Figura 115 Conversión imagen cv2 a imagen de superficie (pygame)

- h. Creación del marcador.** El marcador se crea a partir de las coordenadas (x, y) y tamaño (w, h) de la mano, es básicamente un cuadrado con

propiedades de colisión que sigue la posición de la mano cerrada, el algoritmo para la creación del marcador se muestra en la figura 116.

```
cursor_hand = pygame.Rect(x_hand, y_hand, w_hand, h_hand)
```

Figura 116 Creación del marcador

- i. **Interacción con los elementos virtuales.** Es aquí donde se diferencian cada uno de los juegos de realidad aumentada, ya que en este punto se definen como se va a realizar la interacción con los elementos virtuales, es decir que fin va a tener el juego dependiendo a que sección este enfocado: aprendizaje, entretenimiento o razonamiento. En cada juego se definen clases y funciones que realizan tareas específicas, para este caso del juego de matemáticas se definen algunas clases y funciones que ejecutan diferentes tareas.
- **Clase botón.** Esta clase se encarga de la creación de los botones de la interfaz, el algoritmo se aprecia en la figura 117.

```
class boton(pygame.sprite.Sprite):
    def __init__(self, imagen1, imagen2, x=200, y=200):
        self.imagen_normal=imagen1
        self.imagen_seleccion=imagen2
        self.imagen_actual=self.imagen_normal
        self.rect=self.imagen_actual.get_rect()
        self.rect.left, self.rect.top=(x, y)

        self.temp2 = 0

    def update(self, pantalla, cursor, cursor2):
        if cursor.colliderect(self.rect):
            self.temp2 += 1
            self.imagen_actual=self.imagen_seleccion
            if self.temp2 == 30:
                self.temp2 = 0
                pygame.quit()
        else:
            self.imagen_actual=self.imagen_normal
            self.temp2 = 0

    display.blit(self.imagen_actual, self.rect)
```

Figura 117 Clase para creación de botones

- **Clase escena.** Esta clase administra todo, está compuesta por dos funciones la función `init()` para inicializar y definir los parámetros que van a contener las escenas, y la función `update()` que se encarga de administrar las variables e ir actualizando constantemente las escenas, en la figura 118 se muestra una parte del algoritmo de la clase escena.

```
class escena(object):
    def __init__(self, numero):
        self.numero = numero
        self.contador = 0
        self.lista = vector(self.numero)

    def update(self, display, mouse):
        if self.contador==10:
            display.blit(congra, (250, 0))
            message_to_screen("FELICITACIONES", verde_claro, 0,
                              size="large")
            message_to_screen("CULMINASTE EL JUEGO", verde_claro, 100,
                              size="large")
            message_to_screen("MUY BIEN", verde_claro, 200,
                              size="large")
        else:
            if self.numero==1:
                display.blit(image1, (20, 180))
                message_to_screen(m1, verde_claro, -300, size="medium")
                message_to_screen(m, verde_claro, -250, size="medium")
                display.blit(s1.image, s1.rect)
```

Figura 118 Algoritmo clase escena

Esta clase también se vale de funciones externas para mostrar los mensajes los mensajes en la pantalla y definir los “sprites”.

- **Función `message_to_screen`.** Se encarga de definir y mostrar los mensajes de texto dentro de las escenas, en la figura 119 se muestra el algoritmo empleado para mostrar los mensajes.

```
def message_to_screen(msg, color, y_displace = 0, size = "small"):
    textSurf, textRect = text_objects(msg, color, size)
    textRect.center = (int(SIZE[0] / 2), int(SIZE[1] / 2) + y_displace)
    display.blit(textSurf, textRect)
```

Figura 119 Función para agregar texto

- **Función sprite.** Permite crear objetos que poseen propiedades de colisión para interactuar en las escenas, el algoritmo se muestra en la figura 120.

```
def sprite(imagen):
    sprite = pygame.sprite.Sprite()
    sprite.image = imagen
    sprite.rect = imagen.get_rect()
    sprite.rect.left = random.randrange(750,1100)
    sprite.rect.top = random.randrange(180,610)
```

Figura 120 Función para crear sprites

El algoritmo completo del juego "Números Naturales" se encuentra en el anexo E-6, en la figura 121 se muestra la interfaz del juego terminado

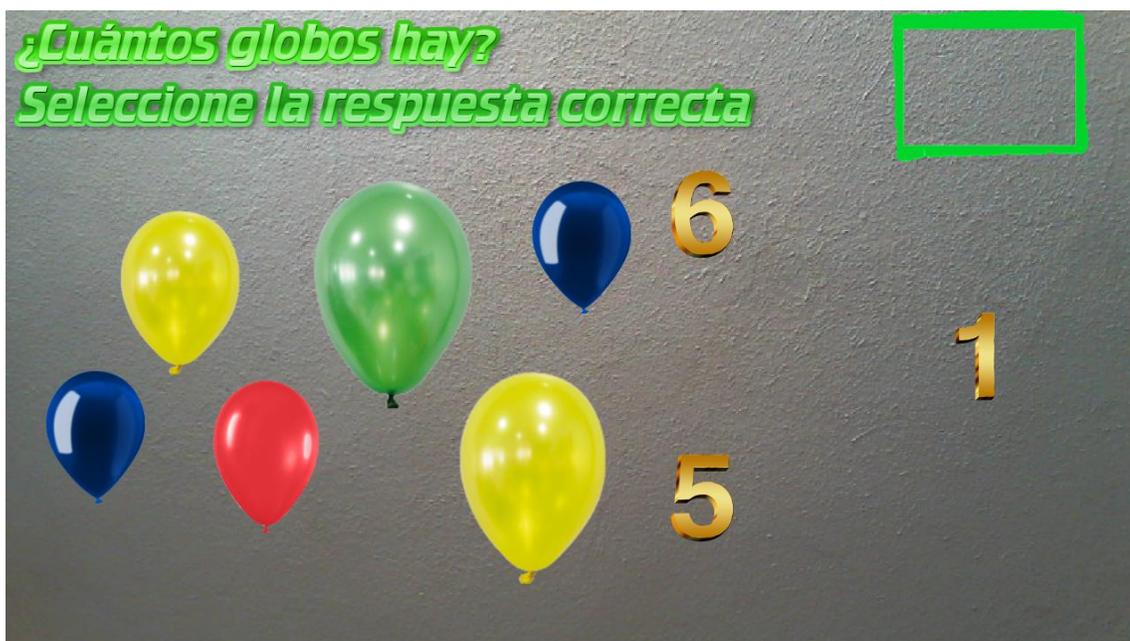


Figura 121 Juego "Números Naturales" desarrollado

De esta manera se van desarrollando cada uno de los juegos de realidad aumentada, en el anexo G se muestra el listado de todos los juegos que se han desarrollado clasificados según el nivel de educación básica, una vez desarrollado los juegos, se debe comunicar estos juegos con el robot didáctico.

4.2.4. Creación de pistas de Audio

En esta sección se realizará la creación de pistas de audio para generar las instrucciones tanto en los menús principales, sub menú y juegos. Las mismas que servirán de guía a los niños y niñas. Cabe destacar que las pistas se grabaron en el software llamado grabador de voz que viene instalado por defecto en todos los sistemas operativos ver figura 122.

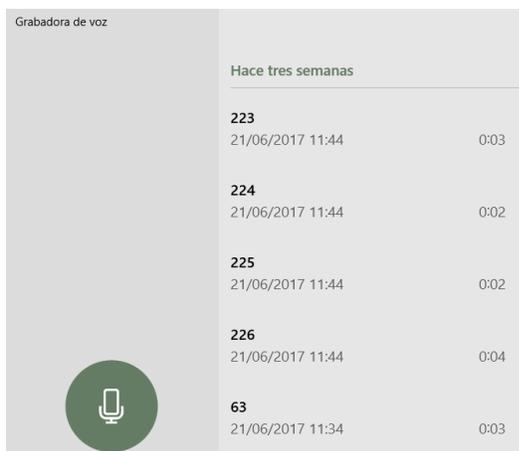


Figura 122 Grabador de voz

Una vez grabado todos los audios se procede a transformarlos en archivos mp3 con el programa “aTube Catcher” como se puede observar en la siguiente figura 123.

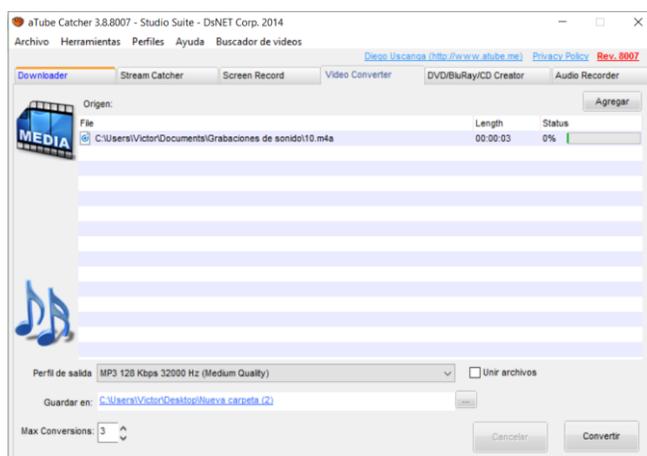


Figura 123 Convertir audios a mp3

Con la ayuda del software “MP3Gain” se procede nivelar el volumen a 82 decibelios de todos los audios .mp3, ver en la figura 124.

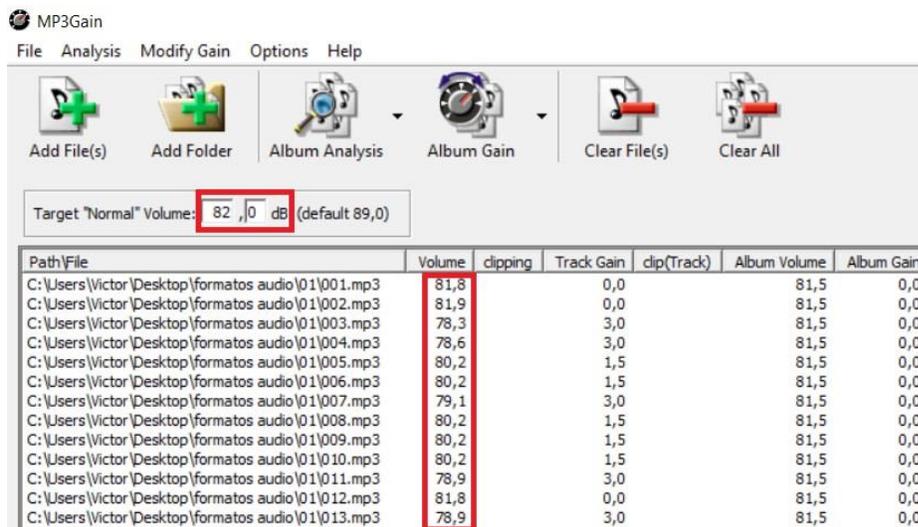


Figura 124 Nivelación de volumen de los audios mp3

Una vez transformadas las pistas en formato .mp3 se procede a almacenar los audios en una tarjeta micro SD de 8 Gb, en total se crearon 197 pistas de audio, como se puede ver en la figura 125, finalmente se coloca la micro SD en el módulo DFPlayer mini Mp3 para su correcta ejecución.

En el anexo H se encuentran el número de pistas con una pequeña descripción del audio grabado.

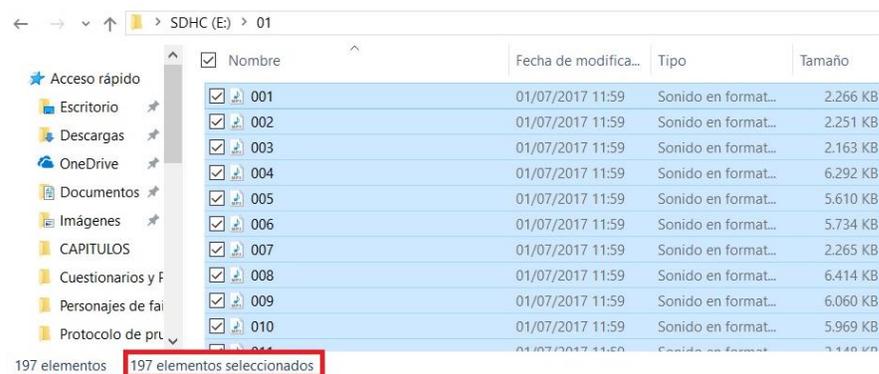


Figura 125 Audios mp3 almacenado es la tarjeta micro SD

4.3. Sistema de Comunicación

El sistema de transmisión de datos entre la Realidad Aumentada y el Robot Interactivo se lo hace mediante dos etapas; la primera etapa es la comunicación entre el sistema RA y Arduino Maestro, esta comunicación se realiza por medio del puerto serial, la segunda etapa consiste en una comunicación maestro-esclavo entre el Arduino del pc y la tarjeta Arduino mega del robot, esta comunicación es inalámbrica vía bluetooth. En la figura 126 se muestra cómo se estructura el sistema de comunicación.

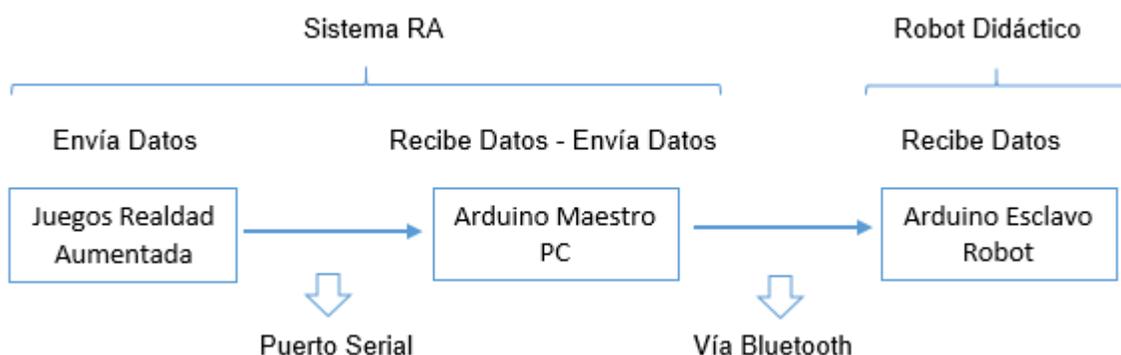


Figura 126 Estructura de Comunicación entre los Juegos RA y el Robot

Los Juegos RA envían los datos por medio del puerto serial al Arduino maestro, éste recibe el dato y por medio de un módulo bluetooth instalado en el Arduino se envía el dato nuevamente al Arduino esclavo que es la tarjeta de control del robot, este recibe el dato la interpreta y ejecuta las instrucciones.

4.3.1. Comunicación entre Sistema RA y Arduino Maestro

Debido a que los juegos RA están desarrollados en Python-Pygame, la comunicación entre los juegos y el Arduino maestro, se realiza mediante programación en lenguaje Python empleando la librería Pyserial para poder utilizar el puerto serial como medio de comunicación.

En la tabla 36, se muestran una lista de comandos más utilizados para realizar la comunicación.

Tabla 36
Lista de comandos para la comunicación entre Python-Pygame y Arduino

COMANDO	FUNCIÓN
<code>import serial</code>	Importa la librería serial, necesario para la transmisión de datos por el puerto serial.
<code>serial.Serial()</code>	Abre el puerto serial
<code>arduino.write()</code>	Permite escribir información en el puerto serial

En la figura 127, se muestra el comando `serial.Serial()` que sirve para abrir el puerto serial y establecer con que velocidad de transmisión se va a realizar la comunicación.

```
arduino = serial.Serial('COM3', 9600)
```

Figura 127 Abrir el puerto serial en Python

El comando `arduino.write()` cumple la función de escribir información en el puerto serial para comunicarse con el Arduino maestro (ver figura 128).

```
arduino.write('1')
```

Figura 128 Escribir información en el puerto serial

4.3.2. Comunicación entre Arduino Maestro y Esclavo.

La configuración maestro- esclavo vía bluetooth, el dispositivo bluetooth maestro se encuentra instalada en la tarjeta PC de control encargado de procesar y ejecutar los juegos, mientras que el dispositivo bluetooth esclavo encargado de recibir los datos se encuentra instalada en la tarjeta de control del robot. En la tabla 37 se puede observar los comandos AT más utilizados para realizar la configuración.

Tabla 37
Lista de comandos AT

COMANDO	FUNCIÓN
AT	Test de conexión UART
AT+ NAME	Consulta / Establecer nombre de dispositivo
AT+ PSWD	Consulta / Establecer contraseña de emparejamiento
AT+ UART	Parámetro UART de consulta / configuración de la velocidad de transmisión estándar
AT+ CMODE	Establecer modo conexión MAC específica
AT+ BIND	Consulta / Establecer la dirección Bluetooth de enlace
AT+ ADDR	Consulta/ Dirección Bluetooth del dispositivo de la MAC
AT+ ROLE	Consulta / Configurar función del dispositivo (1= maestro; 0= esclavo)

Configuración Bluetooth Esclavo - Maestro

La configuración del módulo esclavo HC-06 se realiza a través de los pines RXD y TXD mediante la conexión Arduino Mega y Módulo Bluetooth HC-06 (ver figura 129).

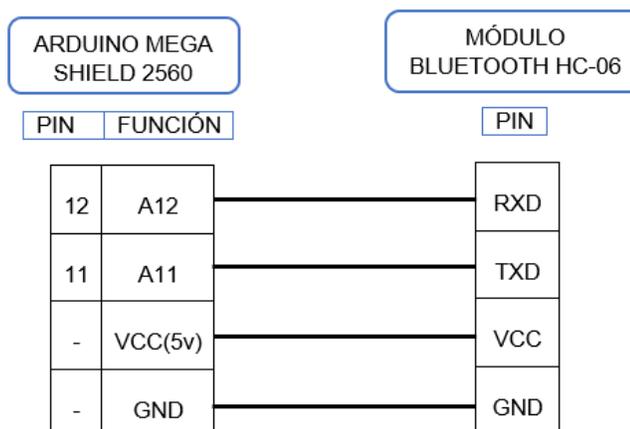


Figura 129 Conexión Arduino Mega y Módulo Bluetooth HC-06

En el anexo D-7 se puede observar a detalle el diagrama de conexión del circuito electrónico y los elementos empleados en la configuración del módulo esclavo HC-06.

Una vez abierto el programa Arduino se procede a seleccionar la tarjeta en este caso Arduino Mega 2560 (ver figura 130).

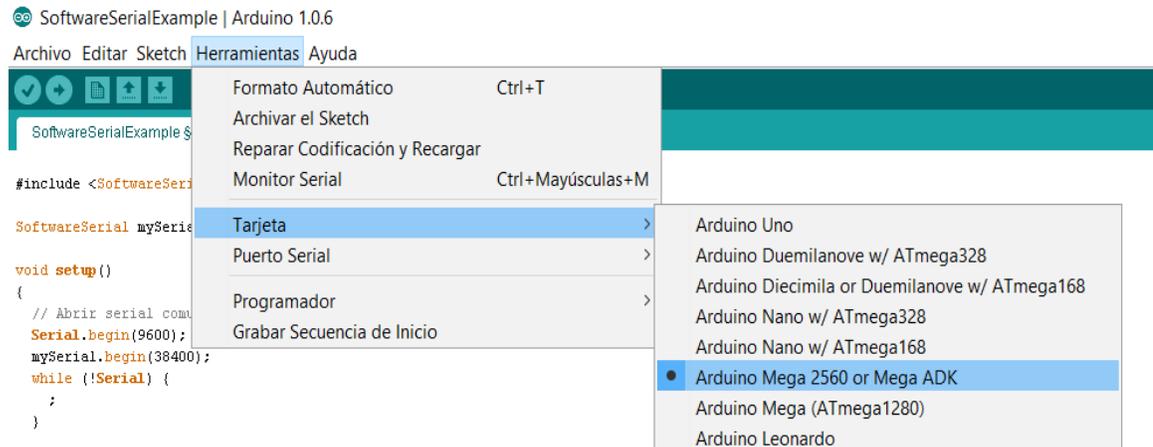


Figura 130 Selección tarjeta Arduino Mega 2650

Además, se selecciona el puerto de comunicaciones (COM) como se puede ver en la figura 131.

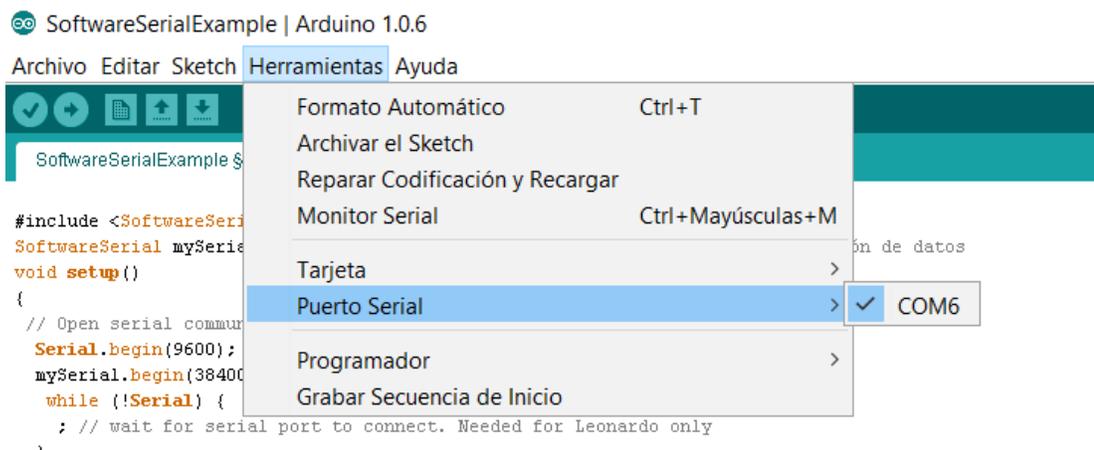


Figura 131 Selección Puerto Serial

Se realiza un pequeño programa para configurar el módulo bluetooth HC-06 esclavo. Este consta de tres variables tipo `char`, para poder cambiar tanto el nombre como la contraseña para esto se escribe los nuevos nombres dentro de las comillas dobles, además se debe de tener en cuenta de no sobrepasar la dimensión del vector creado en las variables. Mientras que para elegir la velocidad de transmisión se escoge el número cuatro, que corresponde a los 9600 bps que es la más ideal para este tipo de comunicación (ver figura 132).

```
char NOMBRE[10] = "Esclavo";  
char BPS       = '4';  
char PASS[10]  = "1234";
```

Figura 132 Configuración del nombre y contraseña módulo HC-06

En la figura 133, se puede observar la ejecución ciertos comandos AT dentro del `void loop()` del programa, con el fin de poder verificar la modificación del nombre, velocidad y contraseña antes ingresados mediante la ventana del monitor serial.

```
Serial.print("AT");  
delay(1000);  
  
Serial.print("AT+NAME");  
Serial.print(NOMBRE);  
delay(1000);  
  
Serial.print("AT+BAUD");  
Serial.print(BPS);  
delay(1000);  
  
Serial.print("AT+PIN");  
Serial.print(PASS);  
delay(1000);
```

Figura 133 Configuración del módulo HC-06

Una vez configurado y cargado los datos en el módulo bluetooth con la ayuda de la aplicación Android BluetoothRSSI se obtiene la MAC, que servirá para realizar la conexión maestro-esclavo (ver figura 134).

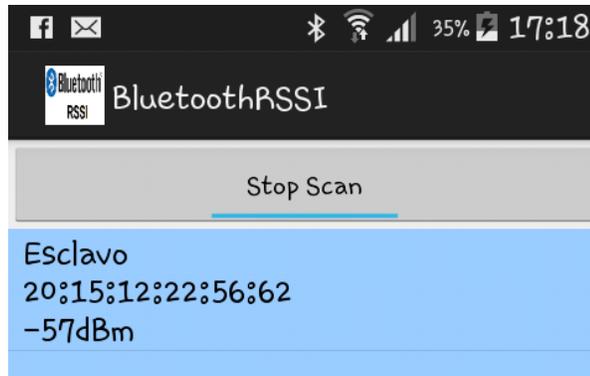


Figura 134 MAC del módulo bluetooth esclavo

La configuración en modo maestro se realiza a través de los pines de transmisión RXD y TXD mediante la conexión Arduino Mega y Módulo Bluetooth HC-05 (ver figura 135).

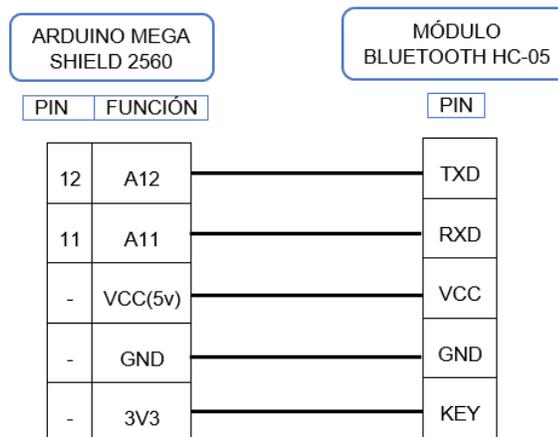


Figura 135 Conexión Arduino Mega y Módulo Bluetooth HC-05

En el anexo D-1 se puede observar el circuito electrónico de la configuración del módulo maestro HC-05.

Una vez abierto el programa Arduino se dirige a la pestaña Archivo>> Ejemplos >> Software Serial >> SoftwareSerialExample y finalmente se da Click en esa opción (ver figura 136).

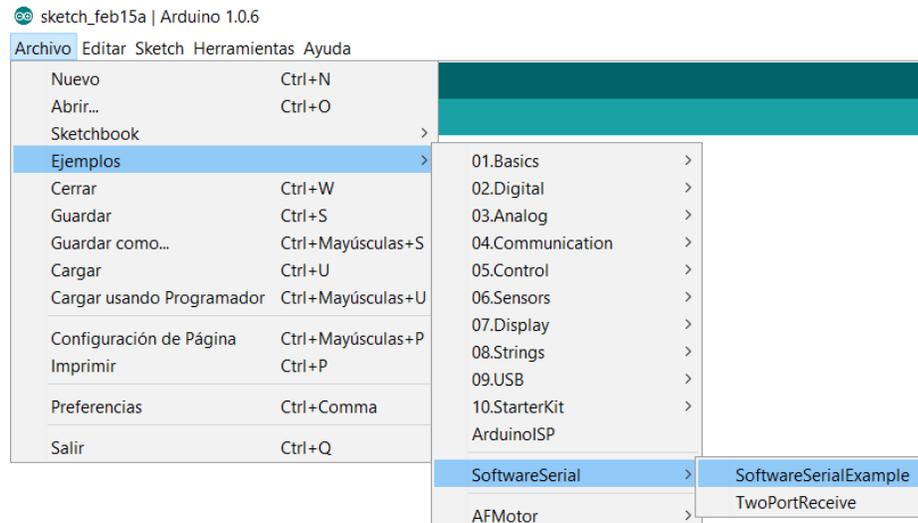


Figura 136 Ejemplo Software Serial

El programa para realizar la configuración del módulo bluetooth HC-05 se divide en tres partes. En la figura 137 se define tanto la librería como los pines de transmisión de datos en los que se van a conectar el módulo físicamente 12 y 11 del módulo Arduino Mega.

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(12, 11);
```

Figura 137 Definición de los pines de transmisión de datos

En la figura 138 se establece la velocidad de transmisión 9600 bps y se manda a ejecutar un código para que el puerto serial del computador se abra.

```
Serial.begin(9600); //
mySerial.begin(38400);
while (!Serial) {;
```

Figura 138 Velocidad y comunicación del puerto serial

Una vez cargado el programa se abre la ventana del monitor serial (ver figura 139), posteriormente se verifica la comunicación bidireccional y la velocidad de transmisión (9600 baud). Para configurar el módulo se escribe AT y luego se da click en el botón enviar.

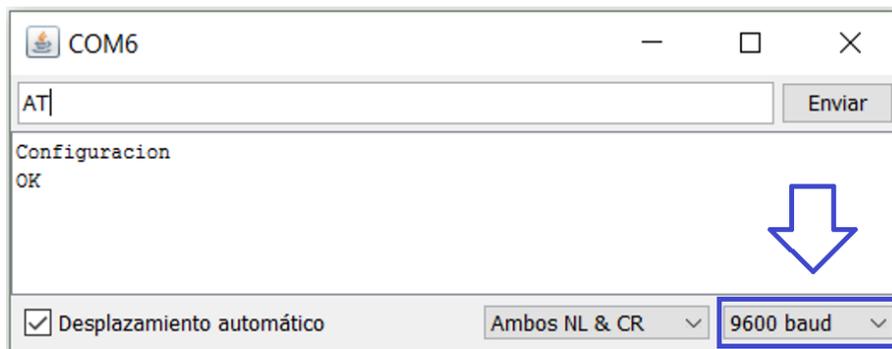


Figura 139 Comando AT

Se digita el comando AT+ ROLE=1 para configurar el módulo bluetooth como maestro (ver figura 140).

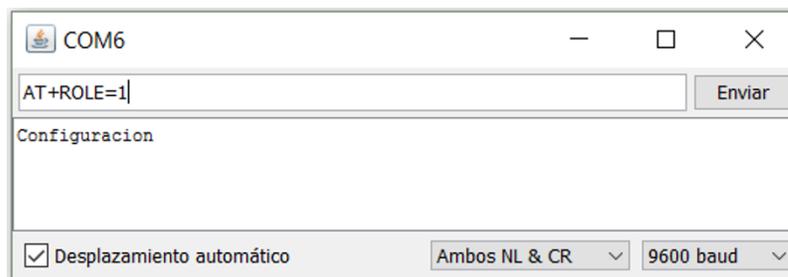


Figura 140 Comando configurar función del dispositivo

Se escribe el comando AT+ CMODE=0 para conectar el módulo bluetooth maestro a una MAC específica, además se debe comprobar la velocidad de transmisión 9600 bps mediante el comando AT+UART (ver figura 141).

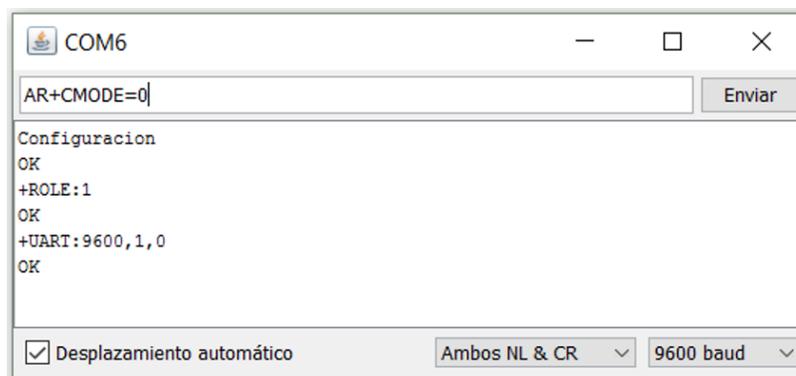


Figura 141 Comando para conectar a una MAC específica

Finalmente se digita el comando `AT+ BIND=20,15,12,22,56,62` con el fin de establecer la comunicación inalámbrica entre el módulo esclavo HC-06 y el módulo maestro Hc-05 (ver figura 142).

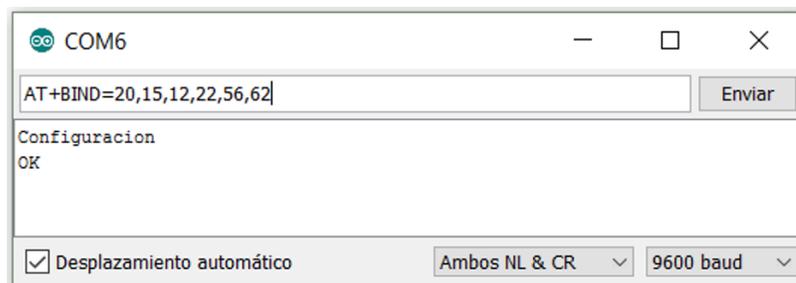


Figura 142 Comando enlazar el módulo maestro con el esclavo

Una vez realizado el diseño mecánico, electrónico y eléctrico del robot, al igual que el diseño de las interfaces de los juegos de Realidad Aumentada, y las configuraciones de los dispositivos bluetooth para la transmisión de datos, se procede a ensamblar todos los subsistemas, para su posterior implementación y realización de pruebas.

CAPÍTULO V

ENSAMBLE, IMPLEMENTACION Y PRUEBAS

5.1. Ensamble del Robot

Una vez manufacturado las piezas se procede a ensamblar cada una de las partes conjuntamente con los componentes electrónicos y la tarjeta de control, en el anexo C se encuentra el manual de ensamble del robot.

5.1.1 Ensamble de las piernas.

Las piernas son los soportes del robot, se empieza ensamblando los pies los cuales están compuestos por el motor, la caja reductora y la carcasa, en la figura 143 se muestra los componentes de los pies del robot.

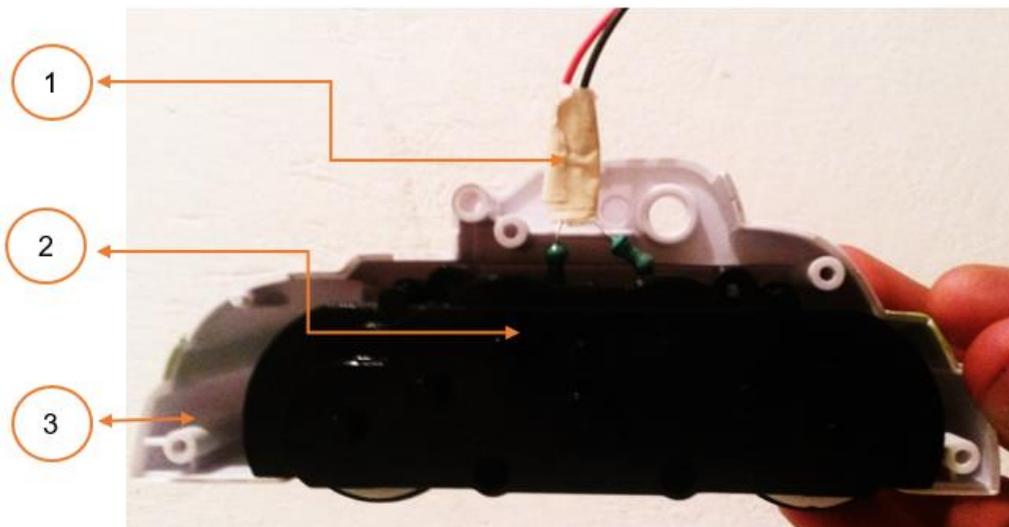


Figura 143 Ensamble del Pie

Elementos: (1) Conectores del motor, (2) Motor y Reductor de engranes, (3) Carcasa del pie.

La parte superior de la pierna está compuesta por el eslabón de soporte, además de soportar este elemento sirve para unir la parte del pie con el muslo, otro elemento es el eje de acero, éste complementa la sujeción, permite el desplazamiento y evita la colisión entre el pie y el muslo (ver figura 144).

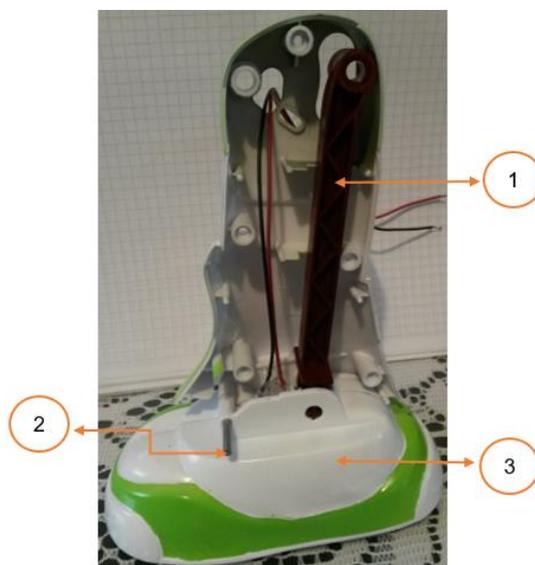


Figura 144 Ensamble de la pierna

Elementos: (1) Eslabón de soporte de la pierna, (2) Eje de unión, (3) Pie

El ensamble final de la pierna derecha se puede apreciar en la figura 145, para la pierna izquierda se sigue el mismo procedimiento.

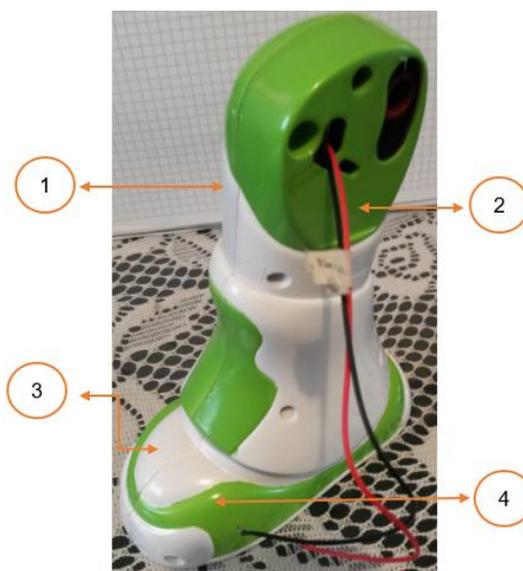


Figura 145 Ensamble final de la pierna derecha

Elementos: (1) Carcasa muslo anterior, (2) Carcasa Muslo Posterior, (3) Carcasa pie anterior, (4) Carcasa pie posterior

5.1.2. Ensamble de los Brazos

Los brazos cumplen la función de equilibrar el cuerpo del robot, posee dos grados de libertad cada uno de 30 grados de movimiento, y se componen por dos articulaciones una de ellas para la unión al torso, la segunda articulación permite la unión entre el brazo y el antebrazo, el elemento final se compone de un puño cuya función es simplemente estético, cabe recalcar que este elemento es reciclado, en la figura 146 se aprecia el ensamble del brazo derecho.



Figura 146 Ensamble del brazo

Elementos: (1) Articulación del hombro, (2) Carcasa hombro-brazo anterior, (3) Carcasa hombro-brazo posterior, (4) Articulación brazo-antebrazo, (5) Carcasa brazo anterior, (6) Carcasa brazo posterior, (7) Puño

Los brazos son simétricos entre si, es decir el brazo izquierdo está compuesto por los elementos que son simétricos al brazo derecho, por lo cual el proceso de ensamble e instalación es el mismo.

5.1.3. Ensamble del Torso

El torso alberga la mayor cantidad de componentes mecánicos para la generación de movimientos, ahí se encuentran los mecanismos para mover los brazos y el cuello, al igual van instalados los componentes electrónicos como la tarjeta de control y el Módulo Shield L293D para controlar los motores.

Se empieza instalando el interruptor para encendido y apagado del robot, en la figura 147 se muestra el lugar donde se ubica el interruptor.

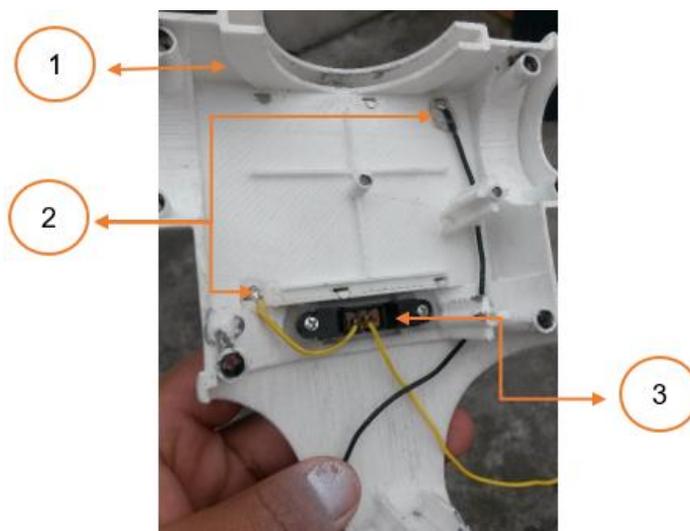


Figura 147 Implementación del interruptor de encendido-apagado

Elementos: (1) Carcasa torso posterior, (2) Puntos de alimentación positivo-negativo, (3) Interruptor

Se ensambla las partes de la columna y el torso, la columna cumple la función de servir como estructura de soporte de las demás piezas, va fijada a la carcasa posterior (espalda) del robot mediante un eje de acero, el torso se compone de dos partes posterior-anterior va unida a la carcasa y a la columna, cumple la función de servir de soporte del eje de los brazos, además posee una cavidad donde se instala el motor para el movimiento de los brazos, en la figura 148 se muestra la instalación de los mecanismos de soporte, los ejes y el motor.

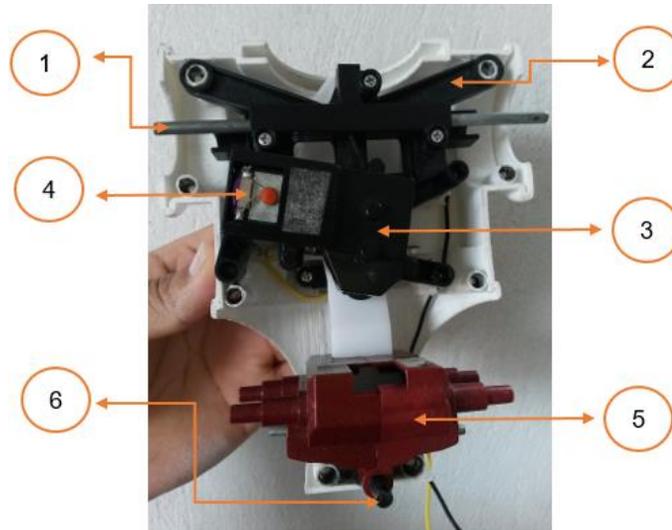


Figura 148 Implementación de los mecanismos de soporte

Elementos: (1) Eje de soporte de brazos, (2) Torso posterior, (3) Torso anterior, (4) Motor de brazos, (5) Base de columna, (6) Eje de unión columna-carcasa

Se procede a ensamblar las partes de los brazos y las piernas al torso, en la figura 149 se muestra el proceso.

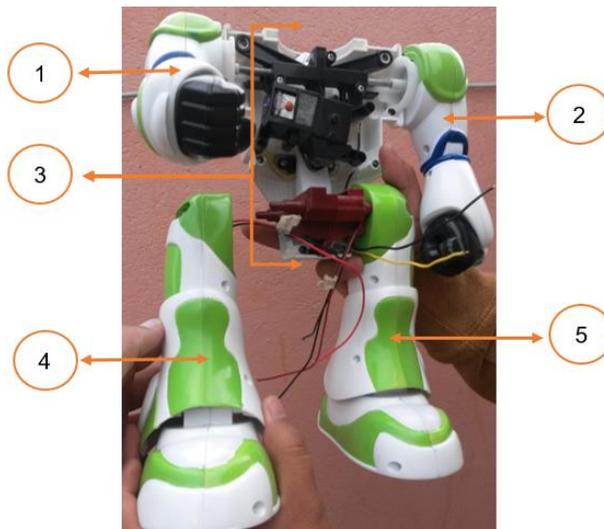


Figura 149 Ensamble del cuerpo del robot

Elementos: (1) Brazo Derecho, (2) Brazo izquierdo, (3) Torso, (4) Pierna derecha, (5) Pierna izquierda.

La carcasa anterior del torso (pecho) posee unas ranuras donde se inserta la tarjeta Arduino Mega conjuntamente con el módulo L293D donde van conectados los tres motores para el control de movimientos (ver figura 150), la carcasa igualmente posee una ranura en la parte inferior izquierda por donde se puede introducir el cable serial de la tarjeta de control y conectarlo a la PC para realizar la sincronización de todos los dispositivos electrónicos mediante la programación del Arduino.

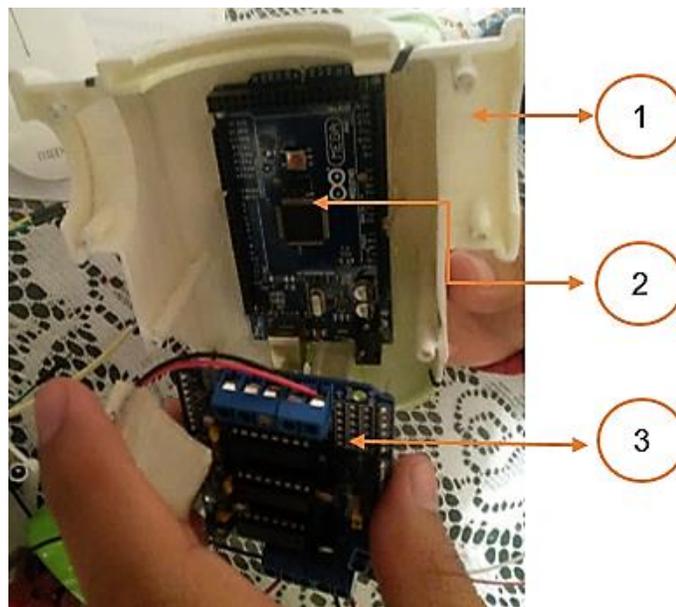


Figura 150 Implementación de la tarjeta de control

Elementos: (1) Carcasa torso anterior, (2) Tarjeta Arduino Mega, (3) Módulo Shield L293D

En la figura 151 se aprecia en forma global los elementos del cuerpo del robot ya ensamblados, se ha realizado un punto de alimentación que permita brindar por separado la energía eléctrica a la tarjeta de control, motores y el sistema de iluminación que en este caso se ubican en la cabeza. Como se puede observar todo el cableado de los dispositivos electrónicos que van a la cabeza pasan por un orificio del cuello, esto no altera en nada su movilidad.



Figura 151 Ensamble Torso parte electrónica

Elementos: (1) Cuello, (2) Carcasa torso anterior, (3) Punto de alimentación, (4) Tarjeta de control, (5) Brazos, (6) Piernas

5.1.4. Ensamble de la cabeza

La cabeza es la parte donde están instalados la mayor cantidad de dispositivos electrónicos, en la figura 152, se puede visualizar como están distribuidos los componentes, el sistema de iluminación se ubica tanto en la parte frontal para iluminación de los ojos como en las dos partes laterales para iluminación de las orejas, la matriz de leds encargada de simular los gestos igual está ubicada en la parte frontal en la apertura de la boca, el módulo bluetooth se ubica en la parte lateral derecha alejado un poco de los demás componentes para no alterar su función de receptor datos.

El módulo de audio está ubicado en la parte lateral izquierda a la cual a su vez están conectados los dos parlantes localizados en los lados laterales de la cabeza estos cumplen la función de reproducir los sonidos almacenados en el

módulo de audio, en la parte trasera se ubica un punto de conexión para todo el sistema de iluminación, todos estos elementos electrónicos están protegidos del medio exterior por las carcasas anterior y posterior de la cabeza.

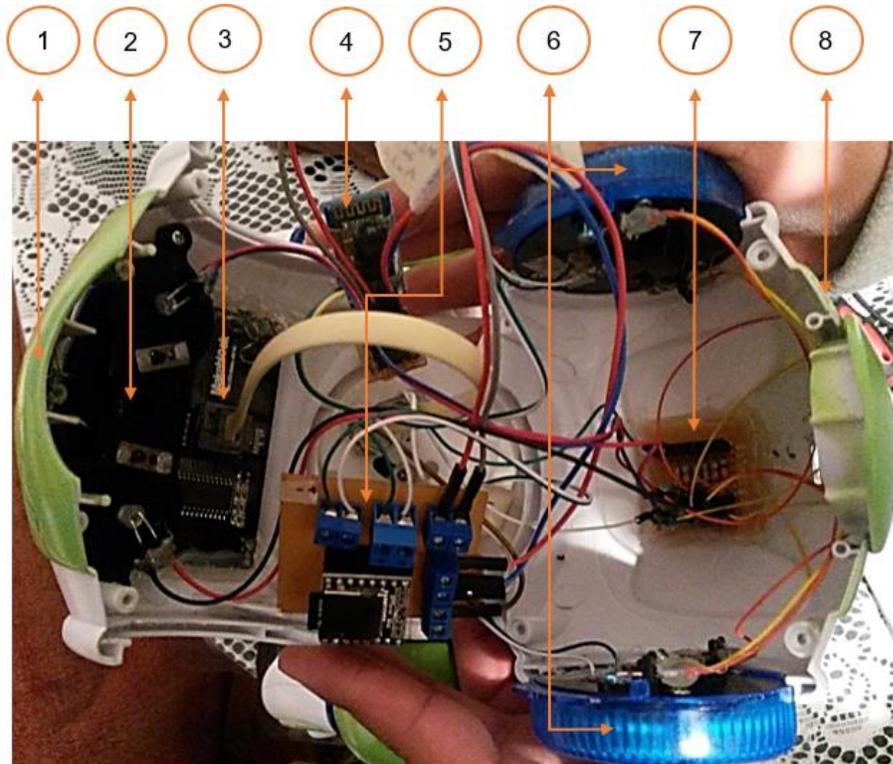


Figura 152 Ensamble Cabeza

Elementos: (1) Carcasa cabeza anterior, (2) Sistema de Iluminación ojos, (3) Matriz de leds Boca, (4) Módulo Bluetooth, (5) Módulo de Audio, (6) Parlantes orejas, (7) Punto de alimentación del sistema de iluminación, (8) Carcasa cabeza posterior.

Una vez ensamblado todos los componentes mecánicos, electrónicos, y la carcasa, el prototipo del robot didáctico está listo, y para entrar a funcionar se debe cargar todos los subprogramas de control, subprogramas que ya se describieron en el capítulo 3, sección 3.3.1 Configuración-Programación de los módulos y tarjeta Arduino. En la figura 153 se aprecia el ensamble final vista desde diferentes perspectivas, frontal, lateral y posterior.



(a) Vista Frontal

(b) Vista Lateral

(c) Vista Posterior

Figura 153 Ensamble final del Robot

5.2. Implementación del sistema de Realidad Aumentada

El sistema de realidad aumentada es básicamente un computador con componentes adicionales, en la figura 154 se muestra todos los componentes ensamblados del sistema de RA.

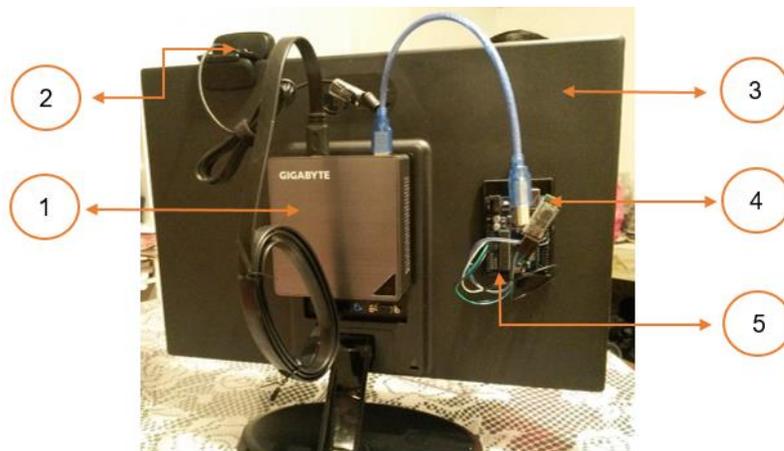


Figura 154 Componentes del Sistema RA

Elementos: (1) Mini PC - Procesador, (2) Cámara, (3) Monitor, (4) Módulo Bluetooth, (5) Arduino Uno – Tarjeta de control Maestro

Los juegos RA desarrollados y los programas necesarios para su ejecución se describe en el capítulo 4, por lo tanto, el siguiente paso es instalar todos los programas requeridos al igual que los drivers de los dispositivos de la cámara y tarjeta Arduino. En el Anexo F se describe el proceso de instalación de todos los programas requeridos para la ejecución de los juegos RA.

5.3. Especificaciones de los sistemas implementados

Una vez culminado la implementación de todos los sistemas, se obtiene un software de realidad aumentada con un robot asistente. Las especificaciones finales de los sistemas se mencionan en la tabla 38.

Tabla 38

Especificaciones finales de los sistemas

Sistema	Especificación	Descripción
Realidad de Aumentada	Juegos	Se han desarrollado 62 RA, distribuidos en: <ul style="list-style-type: none"> • Primero de básica: 14 juegos • Segundo de básica: 18 juegos • Tercero de básica: 15 juegos • Cuarto de básica: 15 juegos
	Marcador Virtual	Se puede interactuar con juegos, utilizando el puño cerrado de la mano.
	Audio	Se implementa 197 pistas de audio, para los mensajes de instrucciones
Robot Asistente	Dimensiones	(alto, ancho, grueso) = (37x20x12) cm
	Movimientos	El robot cuenta con 5 grados de libertad: <ul style="list-style-type: none"> • Camina: Adelante, atrás • Giros: Izquierda, derecha • Giro de cabeza 45 grados

CONTINÚA 

	Expresiones	El programa 6 expresiones emocionales: felicidad, tristeza, impresión, indiferencia, alegría, sonriente
	Peso	2 kilogramos
	Voltaje de Trabajo	6 - 9 Voltios

5.4. Pruebas y Análisis de Resultados

5.3.1. Pruebas de etapa de Preprocesamiento y Reconocimiento.

La imagen que se adquiere por medio de la cámara pasa por una etapa de pre-procesamiento antes de ser utilizado por el clasificador, todas las operaciones y algoritmos de la etapa de pre-procesamiento se describen en la sección 4.2.3 Diseño de los Algoritmos.

La primera operación que se realiza a la imagen es la transformación a escala de grises, esta operación es muy importante puesto a que el clasificador trabaja con imágenes en escala de grises. En la figura 155 (b), se muestra la imagen final una vez aplicada esta operación.



(a) Imagen Original

(b) Imagen en escala de Grises

Figura 155 Transformación a escala de grises

La siguiente operación es la eliminación de ruido, esto se consigue mediante un filtro de convolución que suaviza la imagen, en la figura 156 (b) se presenta la imagen resultante después de aplicar un kernel promedio de 5x5.



(a) Imagen en escala de grises

(b) Imagen suavizada

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

(c) Kernel utilizado

Figura 156 Filtro de Convolución

Una vez pre-procesada la imagen, esta pasa al clasificador para realizar el reconocimiento de la mano, el funcionamiento del clasificador se describe en la figura 111 de la sección 4.2.2 Reconocimiento de la mano cerrada. El algoritmo del clasificador trabaja con dos parámetros `ScaleFactor` y `MinNeighbors` (ver figura 157), el primer parámetro especifica cuánto se reduce el tamaño de la imagen en cada escala, y el segundo cuántos pixeles vecinos debe tener cada rectángulo, estos dos parámetros pueden ser modificados para obtener resultados favorables acorde al entorno (iluminación) en el que se aplique.

```

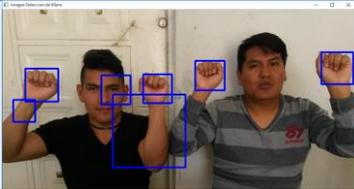
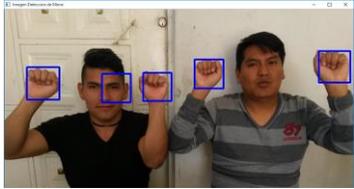
MinNeighbors
ScaleFactor
handetect= hand.detectMultiScale(cv2_image, 1.3, 5)

```

Figura 157 Algoritmo del Clasificador

En la tabla 39, se muestra los resultados aplicando diferentes valores a los parámetros del clasificador.

Tabla 39
Determinación de los parámetros del clasificador

N.º	Factor escala	Min Vecindad	Resultado	Descripción
1	1.1	1		Existen mucha presencia de falsos negativos, el clasificador no diferencia la mano con otras partes.
2	1.1	3		Mejora el reconocimiento, pero siguen existiendo falsos negativos.
3	1.1	4		El clasificador, logra reconocer al 100 % todas las partes de la mano.
4	1.3	3		El clasificador unicamente reconoce un 75 % de las partes.
5	1.3	5		Unicamente 75 % de las partes son reconocidas.
6	1.4	5		El clasificador reconoce unicamente el 50% de las partes.

De la tabla 38, se deduce que los valores más adecuados para los parámetros del clasificador son 1.1 y 4.

5.3.2. Pruebas de funcionamiento del Robot

a) Sincronización de Motores

Luego de haber terminado el ensamble del robot se procede a realizar las primeras pruebas de sincronización de los motores, las pruebas se basan en realizar una serie de movimientos predeterminados, el código de programación para el control de los motores se analiza en el apartado C de la sección “3.3.1. Configuración-programación de los módulos y la tarjeta Arduino”, las mismas que son cargados en la tarjeta de control Arduino. En la tabla 40 se aprecia los resultados obtenidos de las pruebas.

Tabla 40

Primera prueba de movimiento en el robot

MOVIMIENTO	EVALUACIÓN	OBSERVACIÓN
Camina hacia adelante	Reprobada	Completar la secuencia
Camina hacia atrás	Reprobada	Completar la secuencia
Gira a la derecha	Reprobada	Revisar giro completo
Gira a la izquierda	Reprobada	Revisar giro completo
Desplazamiento para adelante	Aprobada	
Desplazamiento para atrás	Aprobada	
Cabeza	Reprobada	Completar el giro

Los resultados que se aprecian en la tabla 39, no satisfacen los movimientos que debe realizar el robot, los motores son controlados por tiempos de activación, es decir para realizar una secuencia de movimientos se debe sincronizar los tiempos de estado activo e inactivo de los motores.

En la tabla 41, se aprecia las correcciones que se realizan al código de programación de cada uno de los movimientos.

Tabla 41

Corrección del código de control de los motores

MOVIMIENTO	ERROR	CORRECCIÓN
Camina hacia adelante	<pre>motor1.run(BACKWARD); motor1.setSpeed(250); motor2.run(RELEASE); motor2.setSpeed(250); delay(1000); motor2.run(FORWARD); motor2.setSpeed(250); motor1.run(RELEASE); motor1.setSpeed(250); delay(1000); motor1.run(BACKWARD); motor1.setSpeed(250); motor2.run(RELEASE); motor2.setSpeed(250); delay(1000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor1.run(BACKWARD); motor1.setSpeed(150); motor2.run(RELEASE); motor2.setSpeed(150); delay(400); motor2.run(FORWARD); motor2.setSpeed(150); motor1.run(RELEASE); motor1.setSpeed(150); delay(800); motor1.run(BACKWARD); motor1.setSpeed(150); motor2.run(RELEASE); motor2.setSpeed(150); delay(400); motor1.run(RELEASE); motor2.run(RELEASE);</pre>
Camina hacia atrás	<pre>motor2.run(BACKWARD); motor2.setSpeed(250); motor1.run(RELEASE); motor1.setSpeed(250); delay(1000); motor1.run(FORWARD); motor1.setSpeed(250); motor2.run(RELEASE); motor2.setSpeed(250); delay(1000); motor2.run(BACKWARD); motor2.setSpeed(250); motor1.run(RELEASE); motor1.setSpeed(250); delay(1000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor2.run(BACKWARD); motor2.setSpeed(150); motor1.run(RELEASE); motor1.setSpeed(150); delay(400); motor1.run(FORWARD); motor1.setSpeed(150); motor2.run(RELEASE); motor2.setSpeed(150); delay(800); motor2.run(BACKWARD); motor2.setSpeed(150); motor1.run(RELEASE); motor1.setSpeed(150); delay(400); motor1.run(RELEASE); motor2.run(RELEASE);</pre>
Gira a la derecha	<pre>motor1.run(BACKWARD); motor1.setSpeed(250); motor2.run(BACKWARD); motor2.setSpeed(250); delay(5000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor1.run(BACKWARD); motor1.setSpeed(150); motor2.run(BACKWARD); motor2.setSpeed(150); delay(1500); motor1.run(RELEASE); motor1.setSpeed(150); motor2.run(FORWARD); motor2.setSpeed(150); delay(600); motor1.run(RELEASE); motor2.run(RELEASE);</pre>

CONTINÚA



Gira a la izquierda	<pre>motor1.run(FORWARD); motor1.setSpeed(250); motor2.run(FORWARD); motor2.setSpeed(250); delay(5000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor1.run(FORWARD); motor1.setSpeed(150); motor2.run(FORWARD); motor2.setSpeed(150); delay(1500); motor1.run(BACKWARD); motor1.setSpeed(150); motor2.run(RELEASE); motor2.setSpeed(150); delay(600); motor1.run(RELEASE); motor2.run(RELEASE);</pre>
Desplazamiento para adelante	<pre>motor1.run(BACKWARD); motor1.setSpeed(250); motor2.run(FORWARD); motor2.setSpeed(250); delay(6000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor1.run(BACKWARD); motor1.setSpeed(150); motor2.run(FORWARD); motor2.setSpeed(150); delay(500); motor1.run(RELEASE); motor2.run(RELEASE);</pre>
Desplazamiento para atrás	<pre>motor1.run(FORWARD); motor1.setSpeed(250); motor2.run(BACKWARD); motor2.setSpeed(250); delay(6000); motor1.run(RELEASE); motor2.run(RELEASE);</pre>	<pre>motor1.run(FORWARD); motor1.setSpeed(150); motor2.run(BACKWARD); motor2.setSpeed(150); delay(500); motor1.run(RELEASE); motor2.run(RELEASE);</pre>
Cabeza	<pre>motor3.run(FORWARD); motor3.setSpeed(250); delay(2000); motor3.run(RELEASE);</pre>	<pre>motor3.run(FORWARD); motor3.setSpeed(240); delay(1000); motor3.run(RELEASE); delay(500); motor3.run(BACKWARD); motor3.setSpeed(240); delay(1200); motor3.run(RELEASE);</pre>

Una vez modificado, revisado y completado el código de programación para la secuencia de movimiento de los motores, se procede a cargar la programación en la tarjeta Arduino Mega para continuar con las pruebas, se realiza de igual manera las pruebas de los motores ejecutando las secuencias de movimientos predeterminados, los resultados obtenidos una vez realizado las correcciones respectivas se aprecia en la tabla 42.

Tabla 42

Corrección de la primera prueba de movimiento en el robot

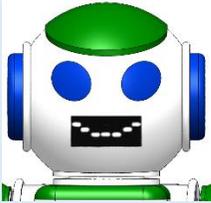
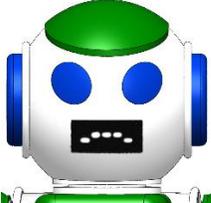
MOVIMIENTO	EVALUACIÓN
Camina hacia adelante	Aprobada
Camina hacia atrás	Aprobada
Gira a la derecha	Aprobada
Gira a la izquierda	Aprobada
Desplazamiento para adelante	Aprobada
Desplazamiento para atrás	Aprobada
Cabeza	Aprobada

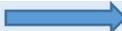
b) Expresiones o gestos

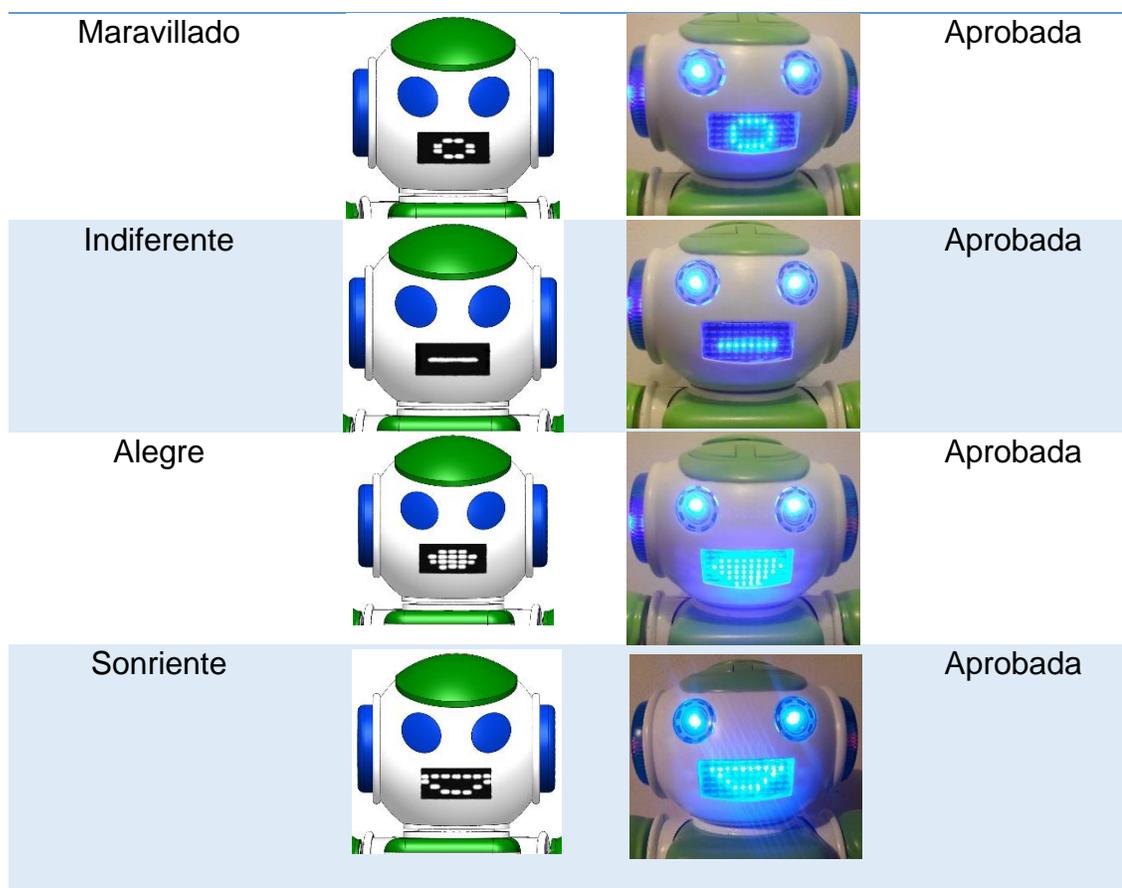
La programación de los gestos se procede a cargar el código en la tarjeta Arduino Mega, esto con el fin de observar los gestos que se generan en matriz de leds. En la tabla 43 se observa los resultados de los diferentes gestos emocionales que se obtienen.

Tabla 43

Gestos o expresiones del robot

GESTO	GESTO ESPERADO	GESTO OBTENIDO	EVALUACIÓN
Feliz			Aprobada
Triste			Aprobada

CONTINÚA 



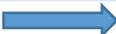
5.3.3. Pruebas de comunicación entre la RA y el Robot

El sistema RA envía datos al robot y éste ejecuta las instrucciones, en base a esto, se procede a enviar varios datos (números) y verificar si los datos recibidos son iguales a los enviados. En la tabla 44 se muestran los resultados realizados de transmisión de datos.

Tabla 44

Pruebas preliminares de transmisión de datos

N.º	Dato Enviado Pygame	Dato Recibido Maestro	Dato enviado Maestro	Dato Recibido Esclavo	Observación (Recibido)
1	1	1	1	1	Correcto
2	3	3	3	3	Correcto
3	6	6	6	6	Correcto

CONTINÚA 

4	8	8	8	8	Correcto
5	10	1	1	1	Incorrecto
6	20	2	2	2	Incorrecto
7	80	8	8	8	Incorrecto
8	100	1	1	1	Incorrecto
9	150	1	1	1	Incorrecto
10	180	1	1	1	Incorrecto

De la tabla 43 se deduce que se está recibiendo únicamente el primer dígito de los datos que se envían, esto se debe a que el Arduino maestro lee un carácter a la vez y lo reenvía al Arduino esclavo.

Para solucionar esto, el Arduino debe leer uno a uno los caracteres del dato enviado e ir almacenando en una variable string hasta tener el dato completo y después enviarlo. Para esto se modifica la programación del Arduino maestro, en la figura 158 se muestra las correcciones que se realiza para solucionar este inconveniente.

Error

```
void loop()
{
  if(Serial.available()) {
    option = Serial.read();

    if(option == "001"){
      BTMasterHC05.write(option);
      delay(3);
    }
    if(option == "002"){
      BTMasterHC05.write(option);
      delay(3);
    }
  }
}
```

Correccion

```
void loop()
{
  if(Serial.available()) {
    option = Serial.read();
    string += option;
    cont += 1;

    if (cont == 3){
      cont = 0;
      if(string == "001"){
        BTMasterHC05.write("001");
        delay(3);
      }
      if(string == "002"){
        BTMasterHC05.write("002");
        delay(3);
      }
    }
  }
}
```

Figura 158 Corrección del código de recepción y reenvío de datos

Una vez corregido el programa se carga nuevamente en el Arduino y se realiza la prueba, en la tabla 45 se muestra los resultados que se obtuvieron tras realizar los cambios.

Tabla 45

Pruebas finales de Transmisión de datos

N.º	Dato Enviado	Dato Recibido	Dato enviado	Dato Recibido	Observación
	Pygame	Maestro	Maestro	Esclavo	(Recibido)
1	1	1	1	1	Correcto
2	3	3	3	3	Correcto
3	6	6	6	6	Correcto
4	8	8	8	8	Correcto
5	10	10	10	10	Correcto
6	20	20	20	20	Correcto
7	80	80	80	80	Correcto
8	100	100	100	100	Correcto
9	150	150	150	150	Correcto
10	180	180	180	180	Correcto

Los resultados obtenidos de la transmisión de datos son correctos, el dato recibido es igual al dato enviado.

5.3.4. Pruebas de funcionamiento

a) Pruebas Preliminares

Con la finalidad de evaluar el grado de impacto que tiene el proyecto sobre los niños y niñas se realiza la primera prueba de funcionamiento del sistema a los alumnos de segundo de básica de la “Unidad Educativa Carlos Cisneros” de la ciudad de Riobamba con la colaboración de la Licenciada Nereida Caiza docente de la institución, en la figura 159 se muestra el sistema completo implementado, donde se aprecia a uno de los alumnos interactuando con el sistema.



Figura 159 Pruebas de funcionamiento en alumna

Los estudiantes tienen cuatro categorías de juegos, matemáticas, lenguaje, ciencias naturales y robótica, cada uno de ellos contienen un conjunto de juegos referentes a los temas y acordes al nivel de educación, en la figura 160, se aprecia al niño interactuando en el menú de categorías.



Figura 160 Menú de Categorías de Juegos RA

En cada juego, el niño interactúa con los elementos virtuales acorde al enunciado o a las instrucciones brindadas por el robot, en la figura 161 se aprecia al niño interactuando con uno de los juegos RA de matemática.

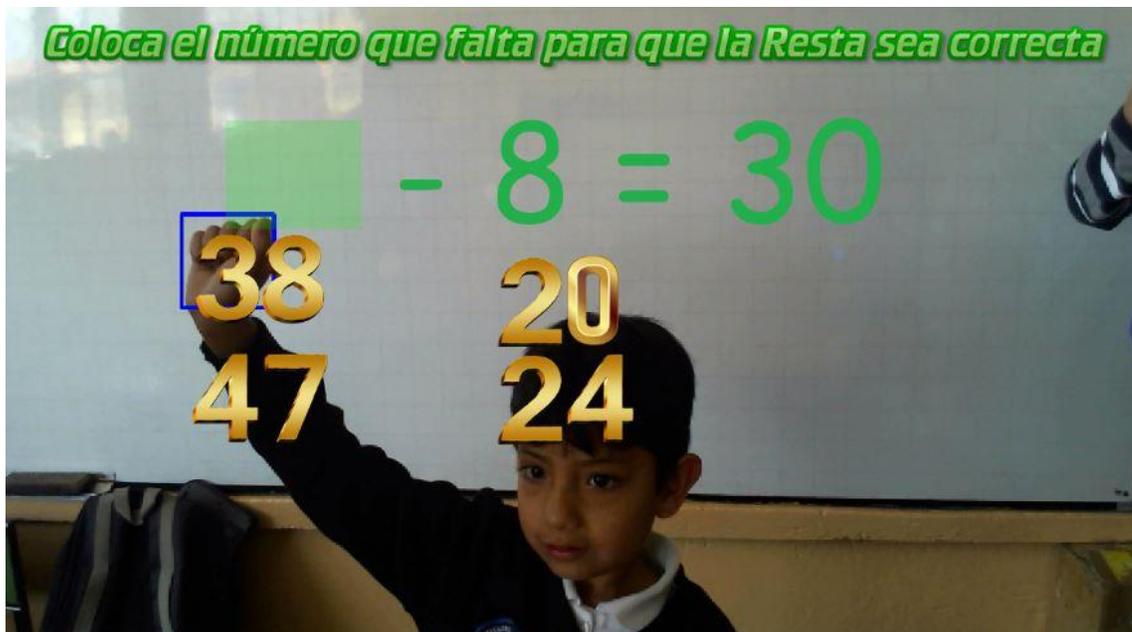


Figura 161 Niño interactuando con los juegos RA

b) Observaciones

Tras observar a los niños y niñas interactuar con el sistema, se identificaron algunos puntos claves que pueden ayudar a mejorar el proyecto.

- El audio del robot es muy bajo, en el aula los estudiantes hacen mucho ruido y opacan las indicaciones del robot.
- El reconocimiento de la mano, algunos niños y niñas colocan el puño muy apegado al cuerpo opacando la silueta de la mano, causa por la cual en ocasiones el sistema no identifica de manera correcta.
- En las escenas de los juegos, los botones para cambio de escena tienen un tiempo de activación muy largo.
- Prefieren pocas iteraciones en cada juego y saltar rápidamente a un juego de otro tema.

c) Correcciones del sistema.

Una vez definido los puntos inestables del sistema, se procede a corregir y mejorar para las pruebas definitivas.

- 1) Para corregir, el bajo sonido del robot didáctico, se procede a cambiar los parlantes por unos de mayor potencia, en este caso se cambia los parlantes de 8Ω 0,25w que estaban instalados inicialmente en el robot por unos parlantes de 8Ω 0,5w (ver figura 162).



Figura 162 Parlantes de 8Ω 0,5w

- 2) Para mejorar la experiencia de usuario con respecto a la interacción de los juegos, especialmente para los más pequeños que en ocasiones se les dificulta colocar de manera correcta la mano se procede a introducir otro marcador, es decir, ahora se tendrá dos posibilidades de manipular los elementos virtuales, mediante la mano cerrada y también con un objeto, en este caso, según las observaciones realizadas y por recomendación de la docente se va a utilizar una pelotita de color verde. Este reconocimiento se lo realiza por medio de detección de color, y está determinada por el rango del color en el espacio hsv (tinte, saturación y brillo), en este caso el parámetro hsv para el color verde es:

- Alto: (107, 255, 255)
- Bajo: (49, 50, 50)

En la figura 163, se observa la parte del código que se mejora, para tener dos posibilidades de interacción y mejorar la experiencia de usuario.

```

# Variable para creacion de marcador
(x_hand, y_hand)= (0, 0) # hand
(w_hand, h_hand) = (0, 0)# hand

(x_ball, y_ball)= (0, 0) # ball
(w_ball, h_ball) = (0, 0)# ball
#-----
#Captura de imagen
ret, cv2_image = cap.read()
# Transformacion a escala de grises
gray = cv2.cvtColor(cv2_image, cv2.COLOR_BGR2GRAY)
#Kernel para convolucion con imagen
kernel = np.ones ((5,5), np.float32) / 25
#Eliminacion de Ruido con filtro
blur = cv2.filter2D (gray, -1, kernel)

```

Reconocimiento Mano

```

handdetect= hand.detectMultiScale(blur, config.ScaleFactor, config.MinNeighbors)

for (x,y,w,h) in handdetect:
    cv2.rectangle(cv2_image, (x,y), (x+w,y+h), (255,0,0),2)
    (x_hand, y_hand)= ((1280-w)-((x*1280/864)+w*0.2), ((y*720/480)+h*0.2))
    (w_hand, h_hand) = (w, h)
#

```

```

# Eliminacion de Ruido con filtro
blured = cv2.filter2D (cv2_image, -1, kernel)
# convertir a espacio de color HSV
hsv = cv2.cvtColor(blured, cv2.COLOR_BGR2HSV)
# Construir una mascara para el color
# Realizar erosiones y dilataciones para remover particulas pequeñas de la mascara
mask = cv2.inRange(hsv, Lower, Upper)
mask = cv2.erode(mask, None, iterations=2)
mask = cv2.dilate(mask, None, iterations=2)
# Encontrar los contornos en la mascara e inicializar la corriente
# (x, y) centro de la bola
cnts = cv2.findContours (mask.copy(), cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE) [-2]
center = None

# Solo si el contorno ha sido encontrado
if len(cnts) > 0:

    # Encontrar la longitud del contorno de la mascara
    # Utilizar esto para encontrar el radio y el centroide
    c = max(cnts, key=cv2.contourArea)
    ((x, y), radius) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)
    center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))

    # solo si el radio es mayor a un valor determinado
    if radius > 10:

        # Dibujar el circulo y el centroide en la imagen
        # Actualizar la lista de puntos de seguimiento
        cv2.circle(cv2_image, (int(x), int(y)), int(radius), (0, 255, 255), 2)
        cv2.circle(cv2_image, center, 5, (0, 0, 255), -1)

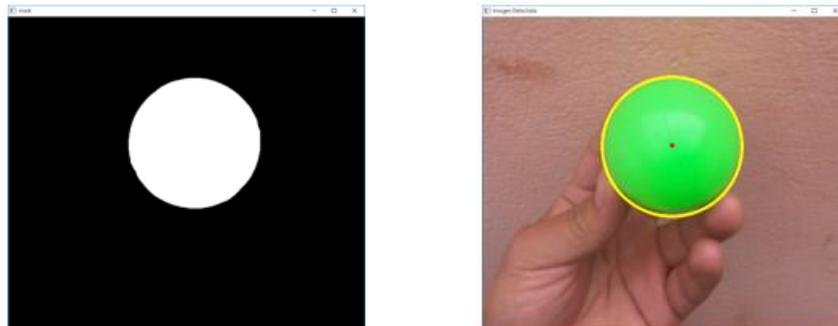
        # Parametros para creacion del marcador
        (x_ball, y_ball) = (1280-((x*1280/864)+radius), ((y*720/480)-radius))
        (w_ball, h_ball) = (2*radius, 2*radius)

```

Reconocimiento Objeto

Figura 163 Código de programación mejorado para interacción de juegos

Los resultados obtenidos de la detección del objeto de color verde se pueden apreciar en la figura 164.



(a) Máscara del Color Verde (b) Detección objeto de color verde

Figura 164 Detección mejorada del marcador

En la figura 165, se muestra cómo responde el reconocimiento del objeto implementado como segundo marcador, esto se realiza en uno de los juegos que se han desarrollado, denominado “Provincias del Ecuador”.



Figura 165 Implementación del segundo marcador

- 3) Para el reducir el tiempo de activación de los botones de cada una de las escenas, se procede a modificar en la función `update` de la clase `boton` el parámetro `self.temp2` que es el encargado de controlar el tiempo de activación, en la figura 166 se aprecia un fragmento del código.

```
def update(self, cursor, opcion):
    if cursor.collidirect(self.rect):
        self.temp2 += 1
        self.imagen_actual=self.imagen_seleccion

        if self.temp2 == 30:
            self.temp2 = 0

        if opcion == 1:
            pygame.quit()

        if opcion == 2:
            main()
```

Figura 166 Parámetro del tiempo de activación de botones

Mediante las observaciones realizadas en las pruebas, el valor de la variable `self.temp2` debe reducirse en un tercio de la original, por lo tanto, se procede a modificar esta variable a un valor de 20.

- 4) Para controlar el número de iteraciones de cada juego, se procede a cambiar el parámetro `self.contador` de la función `update` de la clase `escena`, en la figura 167 se aprecia un fragmento del código donde actúa el parámetro.

```
def update(self, mouse, boton_back, boton_again):
    if self.contador == 10:
        display.blit(congra, (300, 0))
        boton_back.update(mouse, opcion = 1)
        boton_again.update(mouse, opcion = 2)
        display.blit(mvm, (50, 340))
        display.blit(mjo, (670, 345))
```

Figura 167 Parámetro para control de iteraciones

Por medio de las pruebas realizadas, el parámetro `self.contador` debe reducirse a la mitad del valor original, motivo por el cual se modifica el parámetro a un valor de 5.

d) Pruebas Finales

Una vez realizadas todas las correcciones del sistema, se procede a realizar nuevamente las pruebas con los niños y niñas en la “Unidad Educativa Carlos Cisneros” de la ciudad de Riobamba con la colaboración de la Licenciada Nereida Caiza docente de la Institución. Estas pruebas se realizan por un periodo de dos semanas con una duración de 20 a 30 minutos diarios.

También se realizan las pruebas con los niños y niñas del sector de Belisario Quevedo de la ciudad de Latacunga, en colaboración con un grupo de estudiantes de la “Universidad de las Fuerzas Armadas ESPE Latacunga”, pertenecientes al proyecto de vinculación “Rescate de la lengua Ancestral”, estas pruebas se realizaron en las aulas de la Institución, por un periodo de dos días.

Pruebas de aprendizaje en la Unidad Educativa Carlos Cisneros

Las pruebas de aprendizaje se realizaron en base a un cuestionario, para ello se tomó una evaluación inicial para saber el nivel de conocimiento que poseen los niños y niñas, el test de preguntas utilizado fue revisado y aprobado por la docente, este cuestionario se puede ver en el Anexo A-3, conjuntamente los resultados se muestran en el anexo A-4. Una vez conocido los resultados se procede a dividir aleatoriamente a los estudiantes en dos grupos, cada grupo consta de 5 estudiantes por nivel, en este caso se realiza las pruebas con tres niveles de educación básica, primero, tercero y cuarto.

Las asignaturas que se toma como referencia para las pruebas son: Matemáticas, Lenguaje y Ciencias Naturales, en cada grupo se exponen los mismos temas con la finalidad de realizar una evaluación final de conocimientos y comparar los resultados obtenidos entre los dos métodos.

- **Método tradicional**

El primer grupo realiza las actividades educativas aplicando el método tradicional, en la figura 168 se muestra al grupo de niños y niñas resolviendo ejercicios de las materias.



Figura 168 Pruebas aplicando el método tradicional

- **Método aplicando RA y Asistente robótico**

El segundo grupo de estudiantes realiza las actividades utilizando la herramienta didáctica propuesta, ver figura 169.



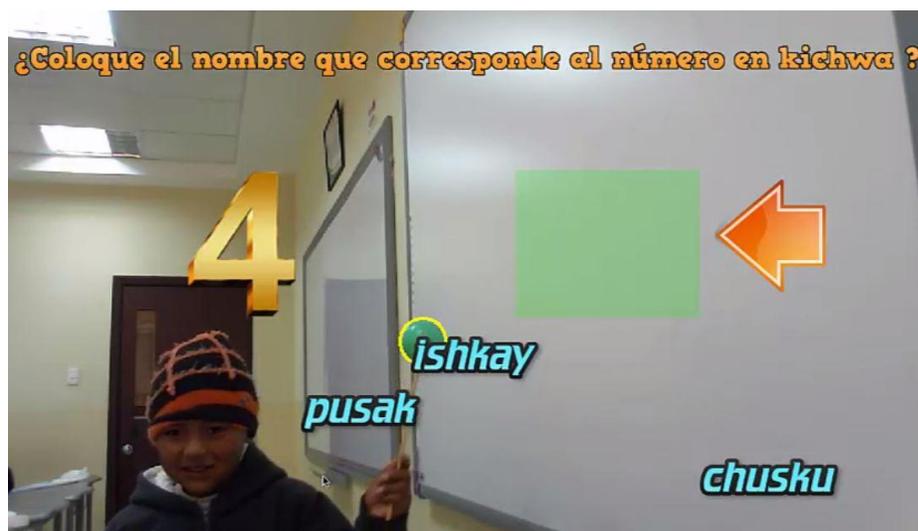
Figura 169 Pruebas aplicando herramienta didáctica RA y Robot

Pruebas de aprendizaje en el sector de Belisario Quevedo.

Estas pruebas se realizan utilizando la herramienta didáctica, a niños y niñas de 6 a 8 años de edad del sector, para ayudar a contar los números en kichwa, en la figura 170 se muestra las actividades realizadas.



(a) Explicación de funcionamiento de la herramienta didáctica



(b) Juego de Números en Kichwa para reforzar conocimientos.

Figura 170 Pruebas en el sector de Belisario Quevedo

En la tabla 46 se muestra el promedio obtenido de los estudiantes de cada nivel con el fin de poder evaluar los resultados y conocer si la implementación de la herramienta didáctica ayuda a mejorar el aprendizaje y conocimiento en los niños y niñas de educación básica. Los resultados de los test y los cálculos de la tasa de mejora de cada estudiante, se puede ver en el anexo A-4.

Tabla 46

Resultados del cuestionario de aprendizaje

Estudiantes Nivel	Método tradicional			Método aplicando RA y Asistente Robótico		
	Promedio Test Inicial	Promedio Test Final	Taza de mejora (%)	Promedio Test Inicial	Promedio Test Final	Taza de mejora (%)
MATEMÁTICAS						
Primero	16	16,8	4,76	15,8	16,8	5,95
Tercero	15,4	16,2	4,94	16	17,1	6,43
Cuarto	16	16,7	3,61	16	16,7	4,19
LENGUAJE						
Primero	16	16,6	3,61	16,2	16,9	4,14
Tercero	16	16,7	4,19	16	16,9	5,33
Cuarto	15,6	16,2	3,70	15,8	16,6	4,82
CIENCIA NATURALES						
Primero	15,8	16,6	4,82	16	16,9	5,33
Tercero	15,8	16,6	4,82	16,2	17,4	6,90
Cuarto	15,6	16,3	4,29	15,8	16,7	5,39
ROBÓTICA						
Primero	N/A	N/A	N/A	15,6	16,3	4,29
Tercero	N/A	N/A	N/A	16,2	16,9	4,14
Cuarto	N/A	N/A	N/A	15,8	16,5	4,24

A continuación, se analiza a detalle la tasa de mejora en cada asignatura como: Matemática, Lenguaje, Ciencias Naturales y Robótica.

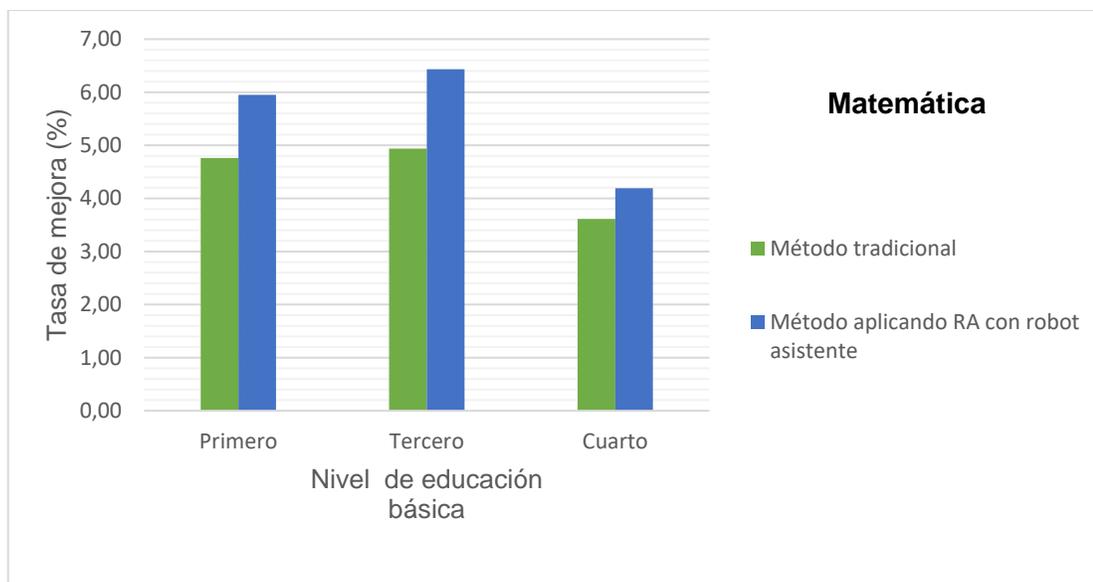


Figura 171 Análisis de la tasa de mejora en Matemática

De acuerdo a los datos obtenidos en la figura 171 en la asignatura de Matemáticas en el primer nivel se obtuvo una tasa de mejora del 1,19% con respecto al método tradicional, en el tercer nivel se alcanzó un 1,49% de mejora y por último en cuarto nivel se logró una tasa de mejora del 0,58% siendo de esta manera el Método RA con asistente robótico superior al método tradicional.

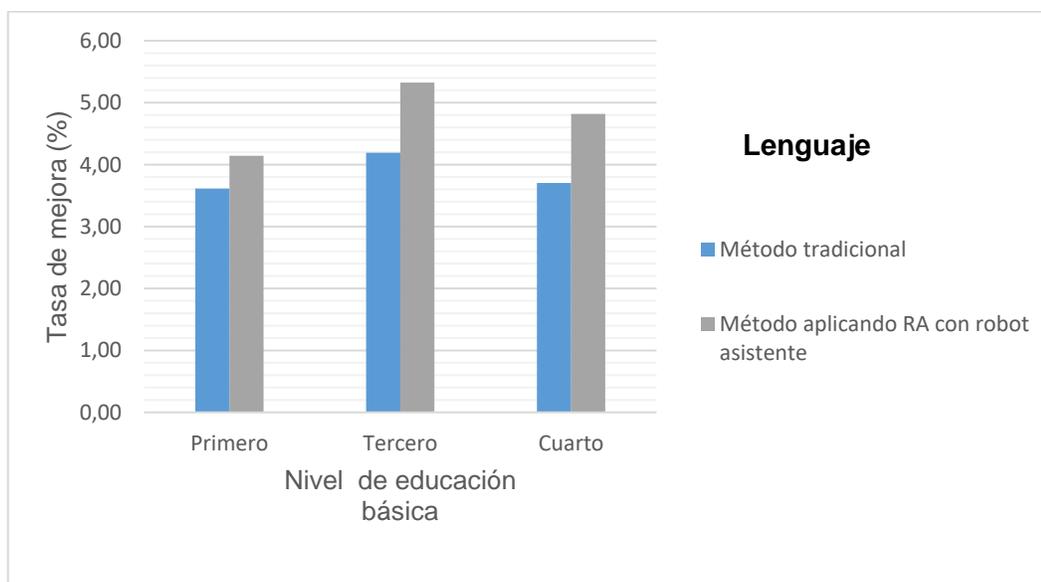


Figura 172 Análisis de la tasa de mejora en Lenguaje

De acuerdo a los datos obtenidos en la figura 172 en la asignatura de Lenguaje en el primer nivel se obtuvo una tasa de mejora del 0,53% con respecto al método tradicional, en el tercer nivel se alcanzó un 1,14% de mejora y por último en cuarto nivel se logró una tasa de mejora del 1,12% de esta manera se puede deducir que el ganador indudablemente es el Método RA con asistente robótico.

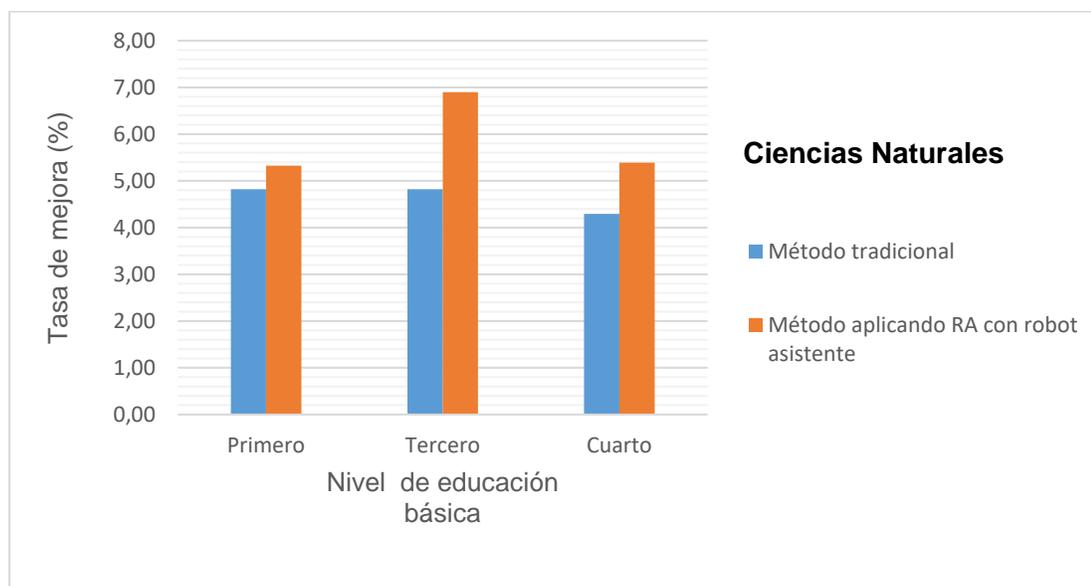


Figura 173 Análisis de la tasa de mejora en Ciencias Naturales

De acuerdo a los datos obtenidos en la figura 173 en la asignatura de Ciencias Naturales en el primer nivel de educación se logró una tasa de mejora del 0,51%, en cuanto al tercer nivel fue de un 2,08% y por último en un cuarto nivel se consiguió una tasa de mejora del 1,1% siendo de esta manera el Método RA con asistente robótico ganador.

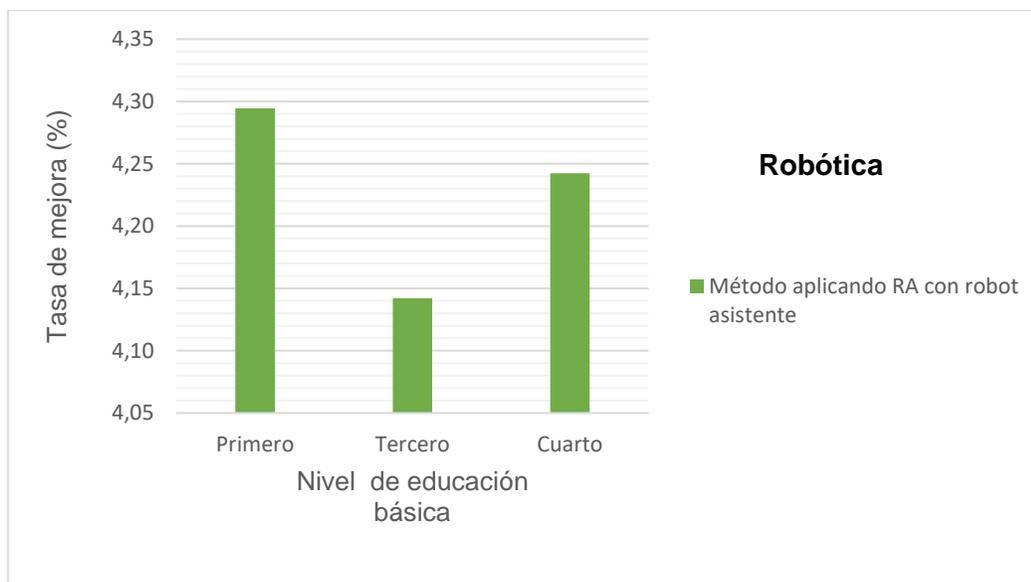


Figura 174 Análisis de la tasa de mejora en Robótica

De acuerdo a los datos obtenidos en la figura 174 en la asignatura de Robótica en el primer nivel de educación se alcanzó una tasa de mejora del 4,29%, en el tercer nivel fue un 4,14% y finalmente para el cuarto nivel se consiguió una tasa de mejora del 4,24% estos porcentajes altos se deben a que los niños y niñas no reciben esta materia en las aulas de clase en el método tradicional y por lo tanto es un nuevo conocimiento el que aprenden.

5.3.5. Pruebas de usabilidad

Con la finalidad de obtener los primeros resultados del sistema, se realizó dos pruebas de funcionamiento. En la primera prueba se utilizó el método de observación, que consiste en ver a los usuarios en este caso un grupo de niños y niñas de segundo de básica interactuar con el sistema y el robot asistente. Una vez culminada la primera prueba se notó algunos puntos claves ver (sección 5.3.4 Pruebas con el usuario) que ayudaron a mejorar el sistema.

Para la segunda prueba de usabilidad se contó con dos grupos de 5 niñas de diferentes niveles de educación excepto segundo nivel ya que con ellos se realizaron las primeras pruebas del sistema.

Cuestionario

Una manera sencilla y practica de evaluar tanto la interfaz RA como el robot es un cuestionario de preguntas ver anexo A-5. En el cuestionario los usuarios evaluarán el grado de impacto del mismo y se tendrán futuras correcciones.

La revisión y validación del cuestionario fueron realizadas por la docente colaboradora, los datos del cuestionario serán de ayuda para la calificación final del sistema. Para ello se presentan tres sentencias importantes.

- Me gustan los juegos RA
- Me encanto los movimientos, gestos e instrucciones del robot
- Las imágenes o iconos del sistema RA me ayudaron a entender mejor el contenido.

Los resultados obtenidos del cuestionario de usabilidad se pueden observar en el anexo A-6. La tabla 47 muestra el resultado promedio de usabilidad, cabe recalcar que las ponderaciones realizadas fueron del 1 al 5.

Tabla 47

Promedio del cuestionario de usabilidad

	USUARIO	Promedio
PREGUNTAS	Me agrado los juegos de RA	4,6
	Me gustaría interactuar de nuevo con la RA y robot	4,5
	Aprendí y descubrí nuevas cosas	4,3
	El sistema RA con el robot son entretenidos	4,6
	Existe una variedad de juegos RA	4,3
	Escuche claramente las instrucciones del robot	4,4
	Me encanto las canciones mientras jugaba	4,9
	Me gustan los movimientos que realizó el robot	4,4
	Los gestos del robot son divertidos	4,4
	Me agradan los iconos del menú de juegos	4,7
	Las imágenes me ayudaron a comprender mejor el juego	4,7
	Recomendaría el sistema RA con el robot a compañeros de otra clase	4,6
	PROMEDIO POR USUARIO	4,53

De acuerdo a los datos obtenidos en la tabla 46 se tiene el resultado total y el promedio por usuario de cada pregunta. Estos datos analizados se pueden ver en la figura 175.

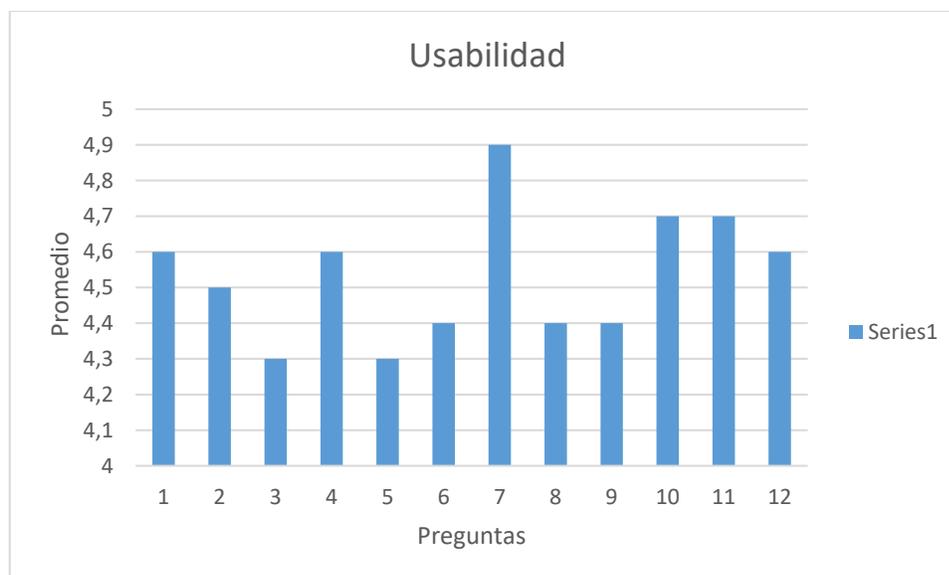


Figura 175 Resultados obtenidos del Cuestionario de usabilidad

Preguntas: (1) Me agrado los juegos de RA, (2) Me gustaría interactuar de nuevo con la RA y el robot, (3) Aprendí y descubrí nuevas cosas, (4) El sistema RA con el robot son entretenidos, (5) Existe una variedad de juegos RA, (6) Escuche claramente las instrucciones del robot, (7) Me encanto las canciones mientras jugaba, (8) Me gustan los movimientos que realizo el robot, (9) Los gestos del robot son divertidos, (10) Me agradan los iconos del menú de juegos, (11) Las imágenes me ayudaron a comprender mejor el juego, (12) Recomendaría el sistema RA con el robot a compañeros de otra clase.

De la figura 175 se tiene que 10 de las doce preguntas tuvieron mucha influencia con un promedio mayor al 4.4 lo cual indica que a los niños y niñas sintieron agrado, aprendieron nuevas cosas y disfrutaron de la variedad de canciones así también entendieron claramente las instrucciones del robot.

La pregunta 3 (*Aprendí y descubrí nuevas cosas*) y 5 (*Existe una variedad de juegos RA*) obtuvieron la calificación más baja 4,4 lo que significa que muchos de ellos ya tuvieron con anterioridad conocimientos previos o ya aprobaron ese nivel y se encontraban realizando un refuerzo de los temas. Mientras que la pregunta 7 (*Me encanto las canciones mientras jugaba*) obtuvo un puntaje alto 4,9 con lo que corrobora que las canciones de los juegos les mantiene activos.

5.5. Validación de la hipótesis

La hipótesis propuesta en el proyecto es:

¿El uso de un sistema de realidad aumentada con robot asistente permitirá mejorar el aprendizaje, creatividad y entretenimiento en niños y niñas de educación primaria?

Para la validación también se toma en cuenta la variable dependiente e independiente.

Variable independiente

- Sistema de realidad aumentada con robot asistente.

Variable dependiente

- Mejora del aprendizaje, creatividad y entretenimiento en niños y niñas de educación primaria.

Los resultados obtenidos de la sección 5.3.4, se analizan en la tabla 48, donde se muestra una comparación entre la tasa de mejora de los estudiantes de cada nivel, con la finalidad de conocer el impacto que causa la utilización de la herramienta didáctica propuesta con respecto al método tradicional.

Tabla 48**Taza de comparación entre los dos métodos**

		Matemática (%)	Lenguaje (%)	Ciencias Naturales (%)	Robótica (%)
1 nivel	Método Tradicional	4,76	3,61	4,82	N/A
	Método RA con robot asistente	5,95	4,14	5,33	4,29
3 nivel	Método Tradicional	4,94	4,19	4,82	N/A
	Método RA con robot asistente	6,43	5,33	6,90	4,14
4 nivel	Método Tradicional	3,61	3,70	4,29	N/A
	Método RA con robot asistente	4,19	4,82	5,39	4,24

Como se puede ver en la tabla 47, existe mejora en ambos casos tanto en el método tradicional como en el método RA con Robot Asistente, de igual manera se observa que los alumnos que utilizaron la herramienta didáctica tienen una tasa de mejora superior con respecto a los estudiantes que aprendieron con el método tradicional.

Pero para hacer válida la hipótesis general, se debe aplicar una prueba de hipótesis la cual demuestre estadísticamente que los resultados son concluyentes y no son producto de factores externos que hayan influenciado la mejora del alumno, por ejemplo, si los estudiantes repasaron las materias en casa con sus padres, o hayan utilizado otro tipo de herramienta didáctica como el teléfono u otros juguetes lúdicos.

Para ello la hipótesis general, se divide en tres hipótesis.

Hipótesis 1

Existe una diferencia significativa en el rendimiento académico de los estudiantes que aprendieron utilizando la herramienta didáctica con respecto a los estudiantes del método tradicional.

En la tabla 49, se muestra los resultados de los test finales que obtuvieron los estudiantes.

Tabla 49**Datos del promedio utilizando el método tradicional y experimental**

	Método Tradicional	Método Ra con robot Asistente
N.º	Test Final	Test Final
1	16,83	16,88
2	16,83	16,88
3	16,83	17,25
4	17,5	15,63
5	15,33	17
6	17,67	18,25
7	16	17,38
8	16,33	16,37
9	15,5	16,87
10	17	16,5
11	18,67	17,87
12	16,17	16,87
13	15,5	16,37
14	15,33	15,5
15	16,17	16,5

Primeramente, se debe conocer si los datos corresponden a una distribución normal, para lo cual se emplea el método de Shapiro Wilk, se asume una confiabilidad del 95% con nivel de significancia de $\alpha=0,05$, en la tabla 50, se muestra los resultados calculados en el software Excel.

Tabla 50**Resultados del test Shapiro Wilk en las pruebas finales**

<i>Shapiro-Wilk Test</i>		
	<i>Test Final</i>	<i>Test Final</i>
W	0,93615562	0,96026956
p-valor	0,33648013	0,69708241
alpha	0,05	0,05

Puesto que se obtienen datos normales con la prueba Shapiro Wilk ($p = 0,336 > 0,05 = \alpha$) y ($p = 0,697 > 0,05 = \alpha$) ver tabla 49, se procede a utilizar el método T Student con el objetivo de encontrar diferencias significativas entre ambos test finales. (González, Ágel Ramos, Hernandez, & López, 2013)

Se establecen la hipótesis nula y alternativa.

H_0 : Existe una diferencia significativa en el rendimiento académico de los estudiantes que utilizaron la herramienta didáctica con respecto a los del método tradicional.

H_1 : No Existe una diferencia significativa en el rendimiento académico de los estudiantes que utilizaron la herramienta didáctica con respecto a los del método tradicional.

Se establecen las siguientes condiciones con la finalidad de validar la hipótesis.

Si $p\text{-valor} < \alpha \rightarrow$ Se rechaza H_0

Si $p\text{-valor} > \alpha \rightarrow$ Se acepta H_0

A continuación, se realiza el cálculo de la varianza agrupada, grados de libertad y t estadístico.

En la tabla 51, se resumen los datos obtenidos mediante el software Excel, los cuales se pueden ver en el anexo I-1

Tabla 51

Resumen de datos para el análisis del t estadístico

n_1	n_2	s_1^2	s_2^2	\bar{x}_1	\bar{x}_2	$\mu_1 - \mu_2$
15	15	0,9205	0,5326	16,510	16,808	0

Se calcula de la varianza agrupada:

$$s^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{(n_1 - 1) + (n_2 - 1)}$$

$$s^2 = \frac{(15 - 1)(0,9205) + (15 - 1)(0,5326)}{(15 - 1) + (15 - 1)}$$

$$s = \sqrt{0,72655} = 0,8523$$

Calculo del t estadístico:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$t = \frac{(16,510 - 16,808) - 0}{(0,8523)\sqrt{\frac{1}{15} + \frac{1}{15}}} = -0,95$$

Calculo de los grados de libertad:

$$G_l = n_1 + n_2 - 2 = 15 + 15 - 2 = 28$$

Tabla 52

Datos del promedio utilizando el método T student

Método RA con robot asistente		
	<i>Test Final</i>	<i>Test Final</i>
Media	16,51066667	16,808
Varianza	0,920520952	0,532617143
Grados de libertad	28	
Estadístico t	-0,955291217	
P(T<=t) dos colas	0,347605953	
Valor crítico de t (dos colas)	2,048407142	
alpha	0,05	

De la tabla 52 se observa un valor estadístico $t = -0,95$ el cual se ubicaría dentro de la zona de aceptación, por lo cual se realiza un análisis de distribución t de dos colas ver figura 176. Puesto que valor estadístico ($p = 0,347 > 0,05 = \alpha$) se retiene la hipótesis nula y se rechaza la hipótesis alternativa. Es decir, se está 95% seguro que existe una diferencia significativa en el rendimiento académico de los estudiantes cuando interactúan con la herramienta didáctica propuesta.

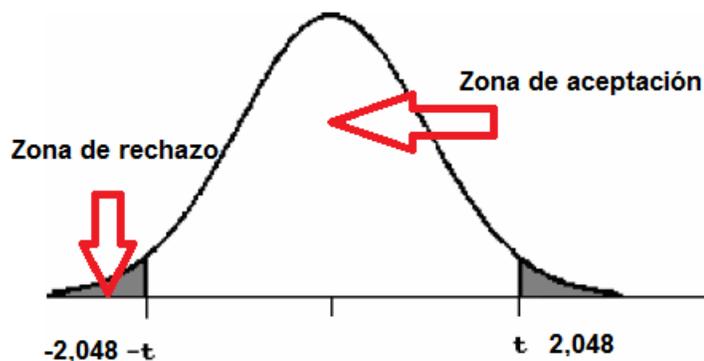


Figura 176 Análisis de distribución de t de dos colas

Hipótesis 2

Las actividades fuera de clases como repasar las materias en casa, utilizar algún tipo de herramienta de apoyo o incluso el nivel educativo de los padres tienen un impacto significativo en la mejora académica de los estudiantes durante el proceso de las pruebas.

Para validar esta hipótesis se emplea el análisis de varianza (ANOVA), la cual permite analizar más de dos grupos o factores influyentes (González, Ágel Ramos, Hernandez, & López, 2013), en este caso se analizan, si los cuatro factores externos mencionados en la hipótesis pueden influir en la mejora de los estudiantes.

Se establecen la hipótesis nula y la alternativa.

H_0 : Los factores externos tienen impacto en la mejora de los estudiantes.

H_1 : Los factores externos no tienen impacto en la mejora de los estudiantes.

Se asume un nivel de confiabilidad de 95% y el nivel de significancia $\alpha = 0,05$.

Puesto a que se analizan varios factores la conclusión de la hipótesis está determinada por la siguiente condición.

Si $p\text{-valor} > \alpha \rightarrow$ Se rechaza H_0

Si $p\text{-valor} < \alpha \rightarrow$ Se acepta H_0

Para realizar este análisis se hace uso del software SPSS, los cálculos correspondientes se observan en el anexo I-2, los datos fueron tomados mediante encuesta a los estudiantes con los cuales se obtienen los resultados que se muestran en la tabla 53.

Tabla 53

Resultado Análisis de Varianza para prueba de hipótesis

Pruebas de efectos inter-sujetos					
Variable dependiente: Porcentaje de Mejora del Estudiante					
Origen	Suma de cuadrados	GL	Media cuadrática	F	Sig. (p-valor)
Repaso	3,539	2	1,769	,732	,508
Lúdica	7,939	1	7,939	3,284	,103
Nivel Educativo Padre	4,121	2	2,060	,852	,458
Nivel Educativo Madre	2,185	2	1,093	,452	,650

De la tabla 52, se observa que ningún valor de significancia de los factores externos es menor que 0,05; por lo tanto, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, es decir, con un 95% de confiabilidad se puede afirmar que ninguno de los factores afectó directamente en el mejoramiento del aprendizaje de los estudiantes.

Hipótesis 3

La Herramienta didáctica propuesta, causa asombro y curiosidad en los estudiantes.

Para la validación de esta hipótesis se recurre a la figura 179 de la sección 5.3.4 Pruebas de usabilidad con los estudiantes, donde se puede observar que el sistema tuvo un gran impacto en los niños y niñas causando asombro e interés por utilizar la herramienta didáctica propuesta.

Con lo expuesto anteriormente se puede corroborar la hipótesis general exponiendo que, el uso de un sistema de realidad aumentada con robot asistente permite mejorar el aprendizaje, creatividad y entretenimiento en niños y niñas de educación primaria

5.6. Análisis Técnico – Económico

En este capítulo se realiza un análisis económico detallado de cada elemento: electrónico, mecánico, material impreso, comunicación y utilería empleado para el sistema de RA con asistente robótico. Debido a que el análisis contiene un sinnúmero de elementos se ha visto la necesidad de dividirlo en cuatro secciones en la que se describirán detalladamente cada componente.

En la tabla 54 se especifica la cantidad y precio invertido en la adquisición de cada elemento que componen el sistema de RA.

Tabla 54

Detalle de costos de materiales de la Realidad Aumentada

Ítem	Cantidad	Precio Unitario (\$)	Total
Mini Pc (Giga byte)	1	500,00	500,00
Cámara	1	80,00	80,00
Regulador de voltaje	1	16,00	16,00
Pantalla de Visualización	1	110,00	110,00
Teclado	1	10,00	10,00
Mouse	1	10,00	10,00
Batería	6	2,50	15,00
Cargador de Batería	1	10,00	10,00
Cable HMI	1	5,00	5,00
Micro SD	1	20,00	20,00
Total			776,00

El costo de los componentes que forman parte de la Realidad Aumentada para la creación de los juegos virtuales es de \$776 dólares.

El costo del robot se detalla en dos tablas, en la tabla 54, se describen todos los componentes electrónicos y mecánicos necesarios para el diseño del

robot humanoide mientras que en tabla 55 se detallan todos los componentes que se imprimieron en 3D.

Tabla 55

Detalle de costos del Robot interactivo

Ítem	Cantidad	Precio Unitario (\$)	Total
Robot interactivo			
Motor DC	3	2,00	6,00
Matriz de led 8 x16	2	12,00	24,00
Tarjeta de control Arduino Mega	1	20,00	20,00
Módulo Shield L293D control de motores	1	12,00	12,00
Tarjeta de audio DFPlayer Mini Mp3	1	12,00	12,00
Caja reductora de engranes	2	15,00	30,00
Diodos led	6	0,25	1,50
Subtotal1			105,50

El sub total de los componentes que conforman la parte electrónica del robot es de \$105,50 dólares.

Tabla 56

Detalle de costos de las impresiones 3D

Impresiones 3D	Costo/hora (\$)	Tiempo (h)	Total
Cabeza	3,00	14	42,00
Barriga	3,00	20	60,00
Extremidades superiores	3,00	10	30,00
Extremidades inferiores	3,00	10	30,00
Columna	3,00	5	15,00
Articulaciones de brazos	3,00	4	12,00
Soporte de piernas	3,00	6	18,00
Tapa de la batería	3,00	5	15,00
Subtotal2			222,00
Total (Subtotal1+Subtotal2)			327,50

El costo empleado en la fabricación e impresiones 3D con material Acido Poli láctico (PLA) tiene un monto de \$ 327,50 dólares.

La tabla 57 se detalla el costo de cada elemento utilizado en el sistema de comunicación de datos.

Tabla 57

Detalle de costos de la comunicación inalámbrica

Ítem	Cantidad	Precio Unitario (\$)	Total
Módulo bluetooth HC-05	1	10,00	10,00
Módulo bluetooth HC-06	1	12,00	12,00
Total			22,00

El gasto que conforman la comunicación o transmisión de datos entre el Sistema de Realidad Aumentada y el robot interactivo es de \$22 dólares.

La tabla 58 detalla los gastos varios e imprevistos que se obtuvieron en el transcurso del desarrollo del proyecto como por ejemplo el daño de un material o reemplazo en de un material defectuoso.

Tabla 58

Detalle de gastos varios

Ítem	Cantidad	Precio Unitario (\$)	Total
Pinturas Spary diferentes colores	3	1,50	4,50
Baquelita circuito Impreso	1	1,00	1,00
Materiales electrónicos	1	30,00	30,00
Cables	1	5,00	5,00
Tornillos	-----	-----	6,25
Otros	-----	-----	30,00
Total			76,75

Los gastos varios o imprevistos como pinturas Spray, circuitos impresos, materiales electrónicos, cables, tornillos y otros es de \$76,75 dólares

En el costo general del proyecto se resumen todos los costos detallados anteriormente como es el caso de componentes del sistema de Realidad Aumentada, construcción del Robot interactivo e impresiones 3D, transmisión de datos y gastos varios ver tabla 59.

Tabla 59

Costo general del proyecto

Descripción	Totales
Realidad Aumentada	\$ 776,00
Robot Asistente	\$ 327,50
Transmisión de datos	\$ 22,00
Gastos varios	\$ 76,75
Total	\$ 1202,25

El costo total del proyecto tiene como gasto 1202,25 dólares americanos, gasto que es sumamente bajo en comparación a la adquisición de otras herramientas tecnológica educativas que superan este valor, por ejemplo, un robot Nao Educativo llega a superar los 7000 dólares o la adquisición de un Sistema Virtual que aproximadamente llega a los 1000 dólares, precios que en la actualidad las instituciones educativas no pueden costear.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

Una vez culminado la etapa de diseño e implementación del proyecto se establecen las respectivas conclusiones y recomendaciones, las cuales sirven de aporte para futuros proyectos similares.

6.1. Conclusiones

- Se investigó, diseñó e implementó un sistema de realidad aumentada con asistente robótico, que sirve como herramienta didáctica de apoyo a los métodos de enseñanza existentes permitiendo el mejoramiento del aprendizaje, creatividad y entretenimiento en los estudiantes de educación básica.
- A partir de librerías de OpenCv y softwares libres, se logró diseñar un software de realidad aumentada, que consiste en un conjunto de juegos educativos que poseen contenido didáctico sobre asignaturas de matemática, lenguaje, ciencias naturales y robótica para estudiantes.
- Mediante el reconocimiento de objetos y procesamiento digital de imágenes, se implementó un marcador virtual que sirve para la interacción con los elementos en el software RA, esto a partir de las coordenadas (x, y) y las dimensiones (w, h), obtenidas de la detección de la mano cerrada o detección del objeto de color verde.
- Se diseño y construyó un robot humanoide interactivo que está compuesto por tres módulos: módulo de matriz 8x16 encargado de generar gestos de: felicidad, tristeza, impresión, indiferencia, alegría y sonriente, módulo L293D que controla los motores para realizar movimientos básicos y el módulo DFPlayer mini mp3 que brinda instrucciones de voz a los estudiantes.

- El uso del material Acido Poli láctico (PLA), permite que la estructura y carcasa del robot tengan un factor de seguridad óptimo e ideal para su construcción y ensamble, además de ser un material no tóxico que puede ser manipulado por los niños y niñas sin riesgo alguno.
- Se implementó un protocolo de comunicación inalámbrica maestro – esclavo entre dos módulos bluetooth, los cuales permiten un sistema de transmisión de datos rápido, seguro y libre de señales interferencias entre el sistema de realidad aumentada y el robot asistente.
- Se realizó pruebas de funcionamiento con los niños y niñas de educación básica, obteniendo resultados positivos con una gran aceptación tanto por parte de los docentes como de los estudiantes, por lo que el uso del Sistema de Realidad Aumentada con Asistente Robótico sirve como herramienta didáctica de apoyo, permitiendo reforzar los conocimientos adquiridos en clases, al igual que promueve la curiosidad y el interés de los estudiantes para aprender sobre tecnologías de aprendizaje.

6.2. Recomendaciones

- Para la ejecución del software de realidad aumentada, se considera necesario usar una cámara que tenga una resolución mínima de 720p, puesto que todos los objetos virtuales, tamaño de la pantalla y cálculo de coordenadas de la posición del marcador, fueron calibradas para funcionar correctamente a esa resolución o superior.
- Para obtener una fluidez de imágenes aceptable, en las escenas de los juegos de realidad aumentada, es aconsejable utilizar un procesador mínimo de tres núcleos y memoria RAM mínima de 2 Gigabyte.
- El sistema funciona mejor en un entorno iluminado, puesto a que la iluminación es un factor que afecta directamente al procesado de imágenes y reconocimiento del marcador, pues la variación de este factor permite realzar los rasgos característicos del objeto.
- Se sugiere trabajar con la velocidad de transmisión de 9600 bps que viene configurada por defecto y no posee ningún inconveniente al momento de enviar los datos de emisor a receptor.
- Para el correcto funcionamiento del dispositivo DFPlayer mini mp3 se recomienda nombrar las pistas mp3 como también la carpeta de canciones con un formato de tres dígitos y dos dígitos respectivamente, de esta forma el módulo no presentará ningún inconveniente en reconocer las carpetas y canciones.
- Para una correcta comunicación y transmisión de datos, además de configurar los módulos bluetooth como maestro o esclavo es aconsejable configurar su dirección MAC, pues es la encargada de generar una dirección única, permanente y segura a la comunicación.
- Es recomendable que después de haber utilizado el robot interactivo, este sea puesto en un lugar cerrado (libre de polvo y suciedades) ya que estos

factores a lo largo del tiempo pueden dañar o provocar fallos en los elementos electrónicos.

- Se puede mejorar al robot con la implementación de sensores de movimientos, cámaras con la finalidad de que posea un mayor grado de inteligencia y autonomía.

REFERENCIAS BIBLIOGRÁFICAS

- Adafruit. (30 de Mayo de 2014). *Adafruit.com*. Recuperado el 13 de Noviembre de 2016. Obtenido de Adafruit.com: https://blog.adafruit.com/2014/05/30/weekend-family-project-building-rapiro-piday-raspberrypi-raspberry_pi-projectrapiro/
- Alonso, M. (11 de Noviembre de 2013). *aumenta.me*. Recuperado el 13 de Noviembre de 2016. Obtenido de aumenta.me: <http://aumenta.me/taxonomy/term/21>
- Alonso, V. (31 de Octubre de 2013). Proyecto de Titulación. *Detección y Reconocimiento de semáforos por Visión Artificial*. Madrid, España.
- Barrientos, E. (18 de Marzo de 2013). *pedagogica*. Recuperado el 12 de Noviembre de 2016. Obtenido de pedagogica: <http://www.pedagogica.edu.sv/index.php/revista-uperspectiva/item/46-robotica-educativa-tecnologia-al-servicio-de-la-educacion?start=460>
- Blanco, J. (10 de Abril de 2013). <http://miradadelgolem.blogspot.com/>. Recuperado el 12 de Noviembre de 2016. Obtenido de <http://miradadelgolem.blogspot.com/>: <http://miradadelgolem.blogspot.com/2013/04/librerias-de-vision-artificial.html>
- Bravo, A. (03 de 08 de 2014). *Robótica al descubierto*. Recuperado el 20 de Agosto de 2017. Obtenido de Robótica al descubierto: <http://solorobotica.blogspot.com/2011/08/motores-de-corriente-continua.html>
- Cadillo, J. (5 de Septiembre de 2016). *Conocimientos sistemas.wordpress*. Recuperado el 14 de Noviembre de 2016. Obtenido de <https://conocimientoysistemas.wordpress.com/category/realidad-aumentada-peru-2/>

- Cantos, L. (15 de Enero de 2015). *www.dineib.gob.ec*. Recuperado el 12 de Noviembre de 2016. Obtenido de *www.dineib.gob.ec*: <http://sistemaeducativoecuador.blogspot.com/>
- Chuquimarca, M. (20 de Agosto de 2016). *dspace*. Recuperado el 12 de Noviembre de 2016. Obtenido de *dspace*: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/11454/1/TESIS%2021-04-2016.pdf>
- Colegio Juan de Valdés. (14 de Mayo de 2015). *Lo que suponen las nuevas tecnologías en nuestro colegio*. Obtenido de *Lo que suponen las nuevas tecnologías en nuestro colegio*: http://www.fliedner.es/media/modules/editor/juandevaldes/docs_v2/proyecto_educativo/201505-nuevas-tecnologias-articulo.pdf
- Educación 3.0. (19 de Julio de 2016). *educaciontrespuntocero*. Recuperado el 12 de Noviembre de 2016. Obtenido de *educaciontrespuntocero*: <http://www.educaciontrespuntocero.com/recursos/apps-para-crear-contenidos-con-realidad-aumentada/25200.html>
- El Estanquillo. (31 de Mayo de 2013). *elestanquillo.wordpress.com*. Recuperado el 12 de Noviembre de 2016. Obtenido de *elestanquillo.wordpress.com*: <https://elestanquillo.wordpress.com/2013/05/31/crean-un-ojo-bionico-para-las-personas-ciegas-de-nacimiento/>
- Garcia, E. M. (2013). *Visión Artificial*. Catalonia: FUOC.
- González, Y., Ángel Ramos, J. M., Hernandez, H., & López, J. (2013). Impacto de los Juegos Didacticos como Herramienta Metodológica en el aprendizaje y la enseñanza de la Ingeniería Industrial. *Educación en Ingeniería*, 37 - 48.
- Heredia, S. (8 de Enero de 2016). *Conferencia: Reflexión sobre los usos de la realidad aumentada en educación*. Recuperado el 22 de Enero de 2017. Obtenido de *The Magical World of Technology*: http://themagicalworldoftechnology.blogspot.com/2016_01_01_archive.html

- Hernández, J. (2 de Septiembre de 2014). *Creaconlaura*. Recuperado el 12 de Noviembre de 2016. Obtenido de Creaconlaura: <http://creaconlaura.blogspot.com/2014/02/arcrowd-aplicacion-online-para-crear.html>
- Informatica. (12 de Marzo de 2016). *Diferencias entre monitores LED y LCD*. Recuperado el 1 de Febrero de 2017. Obtenido de INFORMATICAHOY: <http://www.informatica-hoy.com.ar/hardware-monitores/Diferencias-monitores-LED-LCD.php>
- Keerthi, A. (26 de Junio de 2015). *gamedevelopment*. Recuperado el 15 de Noviembre de 2016. Obtenido de gamedevelopment: <https://gamedevelopment.tutsplus.com/es/tutorials/how-to-learn-pygame--cms-24184>
- Marin, C. (16 de Octubre de 2015). *acodigo.blogspot.com*. Recuperado el 18 de Febrero de 2017. Obtenido de acodigo.blogspot.com: <http://acodigo.blogspot.com/2015/12/entrenar-opencv-en-deteccion-de-objetos.html>
- Matallana, C. (11 de Marzo de 2011). *wordpress.com*. Recuperado el 13 de Noviembre de 2016. Obtenido de <https://carlosmatallana.wordpress.com/2011/03/15/realidad-aumentada-otra-forma-de-ver-el-mundo/>
- Mendez, R. (8 de Diciembre de 2015). *robotsinaction.es*. Recuperado el 3 de Octubre de 2016. Obtenido de <http://www.robotsinaction.es/rob%C3%B3tica-educativa/>
- Ministerio de Educación. (16 de Febrero de 2016). *educacion.gob.ec*. Recuperado el 12 de Noviembre de 2016. Obtenido de [educacion.gob.ec](https://educacion.gob.ec/generalidades/): <https://educacion.gob.ec/generalidades/>
- Mocholí, A. (15 de Octubre de 2014). *Yeeply*. Recuperado el 12 de Noviembre de 2016. Obtenido de Yeeply: <https://www.yeeply.com/blog/desarrollar-aplicaciones-moviles-de-realidad-aumentada/>

- Morcillo, C. G., Fernández, D. V., Jiménez, J. A., & Sanchez, J. C. (2014). *Realidad Aumentada: Un enfoque practico con ARToolKit y Blender* (Vol. I). España: ISBN.
- Mosquera, S. (2 de Abril de 2015). *elcomercio*. Recuperado el 12 de Noviembre de 2016. Obtenido de elcomercio: <http://www.elcomercio.com/tendencias/teebot-robot-tecnologia-educacion-ecuador.html>
- Narváez, D. (13 de Enero de 2012). *scribd*. Recuperado el 24 de Marzo de 2017, de scribd: <https://es.scribd.com/doc/78108343/El-acido-polilactico>
- Nuñoz, V. (14 de Abril de 2015). *Crifacacias*. Recuperado el 15 de Noviembre de 2016. Obtenido de Crifacacias: <http://www.crifacacias.es/camarote/2015/04/14/ponencia-educacion-y-robotica-el-robot-nao-itic15/>
- Olea, P. (05 de Mayo de 2012). *slideshare*. Recuperado el 13 de Noviembre de 2016. Obtenido de slideshare: <http://es.slideshare.net/PEDAIDO1225/tecnologia-bluetooth>
- Pineda, A. (10 de Junio de 2014). *slideshare.net*. Recuperado el 12 de Noviembre de 2016. Obtenido de slideshare.net: <http://es.slideshare.net/Arita1005/exposicion-estructura-del-sistema-educativo>
- Rivera, E. (6 de Marzo de 2015). *UNOCERO*. Recuperado el 3 de Octubre de 2016. Obtenido de <https://www.unocero.com/2015/03/06/ninos-y-robots-aprenden-a-escribir-juntos/>
- Robotlab. (10 de Junio de 2017). *robotlab.com*. Recuperado el 17 de Abril de 2017. Obtenido de robotlab.com: <http://www.robotlab.com/store/nao-standard-edition>
- Robotrónica. (17 de Noviembre de 2015). *AliveRobots*. Recuperado el 3 de Octubre de 2016. Obtenido de <http://aliverobots.com/nao-educacion/>
- Rodriguez, J. (19 de Junio de 2014). *AR Example*. Recuperado el 24 de Enero de 2017. Obtenido de pygame.org: <http://www.pygame.org/project-AR+example-1622-.html>

- Rodriguez, R. (2011). *Procesamiento y Análisis Digital de Imágenes*. Habana - Cuba: RA-MA.
- RS Components. (01 de Abril de 2016). *IOT Protocolos de comunicación*. Recuperado el 13 de Noviembre de 2016. Obtenido de IOT Protocolos de comunicación: <http://redeweb.com/ficheros/articulos/p62a65.pdf>
- Sánchez, A. (16 de Diciembre de 2015). *Diwo*. Recuperado el 13 de Noviembre de 2016. Obtenido de Diwo: <http://diwo.bq.com/zowi-taller-madres-blogueras/>
- Softigal. (18 de Mayo de 2014). *softigal.com*. Recuperado el 14 de Noviembre de 2016. Obtenido de softigal.com: <http://www.softigal.com/redes-internet/redes-wifi/>
- Suarez, P. (16 de Abril de 2015). *innouae*. Recuperado el 13 de Noviembre de 2016. Obtenido de <http://realidadaumentada.info/tecnologia/>
- Ulrich, K. T., & Eppinger, S. D. (2013). *Diseño y desarrollo de productos* (Vol. V). México: McGraw-Hill.
- Urrutia, M. (11 de Septiembre de 2013). *america-retail*. Recuperado el 13 de Noviembre de 2016. Obtenido de america-retail: <http://www.america-retail.com/tendencias-e-innovacion/realidad-aumentada-y-probadores-virtuales-las-tecnologias-que-apuntan-al-retail/>
- Vasquez, W. (24 de Abril de 2013). *Realidad Aumentada.eu*. Recuperado el 13 de Noviembre de 2016. Obtenido de <http://www.realidad-aumentada.eu/es/elementos-de-la-realidad-aumentada/>

ANEXOS



DEPARTAMENTO DE CIENCIAS DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRONICA

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores: **CRISTIAN DAVID CHIMBO TAMAMI y VÍCTOR HUGO JIMÉNEZ CAIZA**

En la ciudad de Latacunga, a los **22 días del mes de agosto del 2017.**



Ing. Patricia Constante

DIRECTORA DEL PROYECTO

Aprobado por:



Ing. Vicente Hallo

DIRECTOR DE CARRERA



Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO