



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

“DISEÑO E IMPLEMENTACIÓN DE UN BRAZO ROBÓTICO
INDUSTRIAL CON 5 GRADOS DE LIBERTAD GUIADO POR
KINECT”

TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO

Para optar al Grado Académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTORES: ALEX FERNANDO RICAURTE SEGOVIA
ISRAEL RIGOBERTO MINDA VALVERDE

TUTOR: ING. JAVIER JOSÉ GAVILANES CARRIÓN

Riobamba - Ecuador

2017

©2017, Alex Fernando Ricaurte Segovia y Israel Rigoberto Minda Valverde

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: “**DISEÑO E IMPLEMENTACIÓN DE UN BRAZO ROBÓTICO INDUSTRIAL CON 5 GRADOS DE LIBERTAD GUIADO POR KINECT**”, de responsabilidad de los señores Alex Fernando Ricaurte Segovia e Israel Rigoberto Minda Valverde, ha sido minuciosamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna E. DECANO FIE	-----	-----
Ing. Freddy Chávez V. DIRECTOR EIE CONTROL Y REDES INDUSTRIALES	-----	-----
Ing. Javier Gavilanes C. DIRECTOR DE TRABAJO DE TITULACIÓN	-----	-----
Ing. Edwin Altamirano S. MIEMBRO DEL TRIBUNAL DE TRABAJO DE TITULACIÓN	-----	-----

Nosotros, ALEX FERNANDO RICAURTE SEGOVIA E ISRAEL RIGOBERTO MINDA VALVERDE, somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio intelectual de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Alex Fernando Ricaurte Segovia

Israel Rigoberto Minda Valverde

DEDICATORIA

A mis padres por su apoyo incondicional, por ese empujón en los momentos difíciles, por acompañarme en todo mi trayecto estudiantil y de vida.

ISRAEL

La concepción de este proyecto está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar.

ALEX

AGRADECIMIENTO

A Dios por haberme dado la fuerza para culminar esta etapa de mi vida.

Agradezco a mi madre María Valverde que me ha demostrado su amor corrigiendo mis faltas y celebrando mis triunfos, a mi padre Rodrigo Minda por siempre confiar en mí a pesar de todo y que con sus consejos me ha ayudado a afrontar los retos de la vida, a mis hermanas Verónica e Itati por estar siempre conmigo a pesar de la distancia brindándome su apoyo. A mis amigos por todos los momentos compartidos en la carrera universitaria.

ISRAEL

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado fuerzas para seguir en momentos de la carrera donde tirar la toalla era más sencillo que lograr mis objetivos, brindándome esperanza y amor.

A mi madre Zoila Mercedes Segovia Murillo,

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi padre Gualberto Ricaurte Piedra,

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero principalmente por su amor.

A mi hermano Diego Mauricio Ricaurte Segovia,

Por su ayuda y consejos en temas relacionados a mi carrera y no afines a mi carrera que me permitieron superar un escalón más en mi vida.

Es por ellos es que soy lo que soy ahora. Mil gracias.

ALEX

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE GRÁFICOS.....	xvi
ÍNDICE DE ABREVIATURAS.....	xviii
RESUMEN	xx
SUMMARY	xxi
INTRODUCCIÓN	1
CAPITULO I.....	6
1. MARCO TEÓRICO REFERENCIAL	6
1.1. Visión Artificial	6
1.1.1. <i>Imagen digital</i>	6
1.1.2. <i>Estudio de color</i>	7
1.1.2.1. <i>Modelo RGB</i>	7
1.1.2.2. <i>Modelo HSV</i>	8
1.1.2.3. <i>Modelo HSL</i>	9
1.2. Elementos de un Sistema de Visión Artificial.....	9
1.2.1. <i>Módulo de Proceso</i>	10
1.2.2. <i>Sistema de respuesta en tiempo real</i>	10
1.2.3. <i>Fuente de luz</i>	10
1.2.4. <i>Sensor de Imagen</i>	12
1.2.5. <i>Tarjeta de adquisición</i>	12
1.2.6. <i>Transformaciones Morfológicas aplicadas al procesamiento de imágenes</i>	12
1.2.6.1. <i>Dilatación Binaria</i>	13
1.2.6.2. <i>Erosión Binaria</i>	13
1.2.6.3. <i>Apertura</i>	14
1.2.6.4. <i>Cierre</i>	14
1.2.6.5. <i>Esqueletos</i>	15

1.3.	Etapas de un sistema de Visión Artificial	15
1.4.	Cámara digital.....	17
1.5.	Tracking de objetos.....	18
1.5.1.	<i>Método de Umbralización.....</i>	<i>18</i>
1.5.2.	<i>Segmentación orientada a regiones.....</i>	<i>19</i>
1.5.3.	<i>Basada en bordes</i>	<i>20</i>
1.5.4.	<i>Por colores.....</i>	<i>20</i>
1.6.	Robótica	21
1.7.	Tipos de robots	21
1.7.1.	<i>Robots Industriales</i>	<i>21</i>
1.7.2.	<i>Robots inteligentes</i>	<i>22</i>
1.8.	Brazo robótico	22
1.8.1.	<i>Configuraciones para un brazo robótico</i>	<i>23</i>
1.8.2.	<i>Articulación.....</i>	<i>23</i>
1.8.3.	<i>Eslabón.....</i>	<i>24</i>
1.8.4.	<i>Grados de libertad</i>	<i>24</i>
1.8.5.	<i>Espacio de trabajo.....</i>	<i>24</i>
1.9.	Software	24
1.9.1.	<i>Python.....</i>	<i>25</i>
1.9.2.	<i>Opencv</i>	<i>25</i>
1.9.3.	<i>Matlab.....</i>	<i>26</i>
1.9.4.	<i>Msc Adams</i>	<i>27</i>
1.9.4.1.	<i>Visión general de Msc Adams.....</i>	<i>28</i>
1.10.	Kinect	28
1.10.1.	<i>Menú de componentes</i>	<i>29</i>
1.10.2.	<i>Tratamiento de la visión del Kinect Xbox 360</i>	<i>30</i>
1.10.3.	<i>Bits de información Depth.....</i>	<i>30</i>
1.10.4.	<i>Fórmula matemática de transformación.....</i>	<i>31</i>
1.11.	Motores	32

1.11.1.	<i>Motores DC</i>	32
1.11.2.	<i>Micromotores</i>	32
1.11.3.	<i>Servomotor</i>	33
1.11.4.	<i>Motor paso a paso</i>	34
1.12.	Puente H	35
1.13.	Sensores	36
1.13.1.	<i>Encoder</i>	36
1.14.	Impresión 3D	37
1.14.1.	<i>Tipos de impresión 3D</i>	38
1.15.	Raspberry PI3	38
CAPITULO II		40
2.	MARCO METODOLÓGICO	40
2.1.	Software utilizado para la programación y diseño del Robot	40
2.1.1.	<i>Instalación y Configuración de Raspbian Jessie</i>	41
2.2.	Selección de Motores	41
2.2.1.	<i>Metodología de LAGRANGE-EULER para el modelo dinámico de un robot</i> .	42
2.2.2.	<i>Controlador de motores</i>	44
2.2.3.	<i>Motores empleados en el brazo robótico según análisis dinámico</i>	44
2.3.	Selección de la cámara	45
2.4.	Diseño del brazo Zortrax 3D	46
2.4.1.	<i>Mesa Repositoria</i>	46
2.4.2.	<i>Base rotatoria</i>	46
2.4.3.	<i>Articulación giratoria</i>	47
2.4.4.	<i>Pieza contendora de brazo-antebrazo</i>	47
2.4.5.	<i>Pieza giratoria del antebrazo</i>	48
2.4.6.	<i>Antebrazo</i>	48
2.4.7.	<i>Pieza del actuador final</i>	49
2.4.8.	<i>Actuador final</i>	49
2.5.	Implementación del Brazo Zortrax 3D	50

2.6.	Diseño del brazo metálico.....	51
2.6.1.	<i>Diseño de la base y soporte de Nylon</i>	51
2.6.2.	<i>Diseño del primer eslabón</i>	52
2.6.3.	<i>Diseño del segundo eslabón</i>	52
2.6.4.	<i>Diseño del adaptador para el motor y el tecl</i>	53
2.7.	Método de cinemática inversa Denavit-Hartenberg del brazo metálico	53
2.8.	Cinemática inversa a través de matrices de transformación homogénea	55
2.9.	Método geométrico del brazo metálico	59
2.9.1.	<i>Resolución del Método Geométrico</i>	60
2.10.	Diagrama del algoritmo de umbralización	60
2.11.	Delimitación del área de trabajo según la aplicación planteada.....	61
2.12.	Diseño del diagrama electrónico.....	62
2.13.	Diagrama de bloques del brazo robótico	64
2.14.	Implementación del brazo robótico metálico	64
CAPITULO III		68
3.	MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN	68
3.1.	Pruebas al primer brazo Zortrax 3D	68
3.1.1.	<i>Problemas físicos y mecánicos</i>	68
3.2.	Pruebas realizadas en cada articulación.....	70
3.2.1.	<i>Eje base- primera articulación</i>	70
3.3.	Pruebas en la segunda articulación	72
3.4.	Prueba en la tercera articulación	73
3.5.	Pruebas en la cuarta articulación	75
3.6.	Prueba en la quinta articulación.....	77
3.7.	Conclusiones de resultados de funcionamiento	77
3.8.	Análisis del mecanismo en Msc Adams.....	78
3.9.	Análisis económico	83
CONCLUSIONES		84
RECOMENDACIONES		85

BIBLIOGRAFÍA..... 1

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-2:	Características del motor DC AmpFlow E30-400 con caja reductora.....	45
Tabla 2-2:	Características del servomotor	45
Tabla 3-2:	Parámetros D-H	54
Tabla 1-3:	Movimiento a 180° en la primera articulación del brazo robótico Zortrax .	70
Tabla 2-3:	Movimiento a 45° en la primera articulación del brazo robótico metálico...	71
Tabla 3-3:	Movimiento a 90° en la primera articulación del brazo robótico Zortrax ...	71
Tabla 4-3:	Movimiento a 90° en la primera articulación del brazo robótico metálico...	71
Tabla 5-3:	Movimiento a 45° en la segunda articulación del brazo robótico Zortrax ...	72
Tabla 6-3:	Movimiento a 45° en la segunda articulación del brazo robótico metálico ..	72
Tabla 7-3:	Movimiento a 90° en la segunda articulación del brazo robótico Zortrax ...	73
Tabla 8-3:	Movimiento a 90° del brazo robótico metálico.....	73
Tabla 9-3:	Movimiento a 45° del brazo robótico Zortrax.....	73
Tabla 10-3:	Movimiento a 45° del brazo robótico metálico.....	74
Tabla 11-3:	Movimiento a 90° del brazo robótico Zortrax.....	74
Tabla 12-3:	Movimiento a 90° del brazo robótico metálico.....	75
Tabla 13-3:	Movimiento a 45° del brazo robótico Zortrax.....	75
Tabla 14-3:	Movimiento a 45° del brazo robótico metálico.....	76
Tabla 15-3:	Movimiento a 180° del brazo robótico Zortrax.....	76
Tabla 16-3:	Movimiento a 180° del brazo robótico metálico.....	76
Tabla 17-3:	Movimiento de la quinta articulación del brazo robótico Zortrax.....	77
Tabla 18-3:	Movimiento de la quinta articulación del brazo robótico metálico.....	77
Tabla 19-3:	Comparación de resultados del brazo robótico Zortrax y brazo robótico metálico	78
Tabla 20-3:	Costo de implementación del brazo robótico Zortrax.....	83
Tabla 21-3:	Costo de implementación del brazo robótico metálico	83

ÍNDICE DE FIGURAS

Figura 1-1: Imagen digital.....	7
Figura 2-1: Colores primarios, secundarios y los terciarios.....	7
Figura 3-1: Modelo color RGB	8
Figura 4-1: Modelo HSV	9
Figura 5-1: Modelo HSV	9
Figura 6-1: Iluminación difusa	10
Figura 7-1: Iluminación Frontal.....	11
Figura 8-1: Iluminación Direccional	11
Figura 9-1: Iluminación estructurad.....	12
Figura 10-1: Imagen binarizada y su dilatación.....	13
Figura 11-1: Imagen binarizada y erosionada.....	14
Figura 12-1: Imagen con ruido vs imagen aplicada la apertura.....	14
Figura 13-1: Aplicación de cierre a una imagen binarizada	15
Figura 14-1: Imagen binarizada vs Imagen aplicada operación de esqueletos.	15
Figura 15-1: Diagrama de bloques de las etapas de un	16
Figura 16-1: Cámara digital.....	17
Figura 17-1: Seguimiento de objetos	18
Figura 18-1: Objetos claros sobre fondo oscuro.....	19
Figura 19-1: Segmentación orientada a regiones	19
Figura 20-1: Basada en bordes.....	20
Figura 21-1: Segmentación por color	20
Figura 22-1: Articulación de rotación y prismática.....	23
Figura 23-1: Kinect	29
Figura 24-1: Tratamiento de la visión del Kinect.....	30
Figura 25-1: Rango de bits	31
Figura 26-1: Datos en Kinect	31
Figura 27-1: Motor DC.....	32

Figura 28-1: Micromotor.....	33
Figura 29-1: Servomotores	33
Figura 30-1: Motor paso a paso	35
Figura 31-1: Controlador IFI VEX	36
Figura 32-1: Encoder	37
Figura 33-1: Raspberry Pi3.....	38
Figura 1-2: Representación de masas y velocidades	42
Figura 2-2: Diseño de la Mesa Repositorio	46
Figura 3-2: Base rotatoria	47
Figura 4-2: Articulación giratoria	47
Figura 5-2: Conector brazo-antebrazo.....	48
Figura 6-2: Pieza giratoria del antebrazo	48
Figura 7-2: Antebrazo.....	49
Figura 8-2: Pieza del actuador final	49
Figura 9-2: Actuador final.....	50
Figura 10-2: Brazo Zortrax 3D implementado.....	51
Figura 11-2: Brazo robotico SCADA de 5 GDL.....	51
Figura 12-2: Diseño de Base y soporte	52
Figura 13-2: Diseño del primer eslabón	52
Figura 14-2: Diseño segundo eslabón	53
Figura 15-2: Diseño del adaptador para el motor y el tecla	53
Figura 16-2: Metodología de Denavit-Hartenberg.....	54
Figura 17-2: Método geométrico.....	59
Figura 18-2: Algoritmo de umbralización	61
Figura 19-2: Rango de movimiento del brazo	62
Figura 20-2: Diagrama de conexión en ISIS Proteus.....	63
Figura 21-2: Esquema de conexión en Fritzing	63
Figura 22-2: Diagrama de bloques del brazo robótico	64
Figura 23-2: Dimensiones del brazo robótico	65

Figura 24-2: Motor acoplado al tecla	65
Figura 25-2: Disposición de sensores.....	66
Figura 26-2: Motores con caja reductora y controladores	66
Figura 27-2: Brazo robótico metálico.....	67
Figura 1-3: Motores del brazo Zortrax.....	69
Figura 2-3: Brazo del robot Zortrax	69
Figura 3-3: Simulación del brazo robótico en Mca Adams	78

ÍNDICE DE GRÁFICOS

Gráfico 1-3: Torque del primer eje.....	79
Gráfico 2-3: Torque del segundo eje.....	80
Gráfico 3-3: Torque del tercer eje.....	80
Gráfico 4-3: Velocidad del tercer eslabon	81
Gráfico 5-3: Velocidad angular del segundo eslabón	81
Gráfico 6-3: Posición del efector final.....	82
Gráfico 7-3: Posición del tercer eslabón	82

ÍNDICE DE ANEXOS

ANEXO A. BRAZO ROBÓTICO METÁLICO IMPLEMENTADO

ANEXO B. KINECT

ANEXO C. IMPLEMENTACIÓN DE DISPOSITIVOS EN EL BRAZO ROBÓTICO

ANEXO D. BRAZO ROBÓTICO ZORTRAX

ANEXO E. PARTES MODIFICADAS DEL BRAZO ROBÓTICO ZORTRAX

ANEXO F. ESLABONES Y ACOPLES IMPRESOS

ANEXO G. ALIMENTACION DE LA ESTRUCTURA

ANEXO H. SOFTWARE EN ARDUINO

ÍNDICE DE ABREVIATURAS

A	Amperios
C	tasa de descarga
CAMSHIFT	Continuously Adaptive Mean Shift
Cm	Centímetros
DC	Corriente Directa
g	Gramo
GDL	Grados de Libertad.
GPIO	General Purpose Input Output
HD	High Definition (Alta definición)
HDMI	High Definition Multimedia Interface
HSV	Hue, Saturation, Value (Matiz, Saturación, Valor)
IA	Inteligencia Artificial.
IDE	Entorno de desarrollo integrado
Kg	Kilogramo
m	Metros.
m A	Miliamperios
mm	Milímetros
mAh	Mili amperios hora
MP	Megapíxeles.
PIR	Passive Infra-Red

RAM	Random Access Memory (Memoria de acceso aleatorio)
RGB	Red, Green, Blue
SO	Sistema operativo
SoC	System on a Chip, Sistema en un Chip
μA	microamperios
V	Voltios.
VA	Visión Artificial.
VGA	Adaptador gráfico de Video
VNC	Virtual Network Computing (Red de computador virtual)
2D	Dos dimensiones.
3D	Tres dimensiones.

RESUMEN

El objetivo del presente proyecto fue diseñar e implementar un brazo robótico industrial con 5 grados de libertad guiado por Kinect, el mismo fue construido en un 90% con metal (hierro) para darle mayor resistencia y estabilidad, para el movimiento del robot se implementaron 3 motores DC de 2.1HP ubicados en las primeras 3 articulaciones debido al torque que se necesita en esta parte del brazo, en estos motores se colocaron sensores en sus ejes (encoders) para poder controlarlos con mayor precisión , y dos servomotores de 4.1 kg.cm en la parte final, dándole 5 grados de libertad al brazo robótico. Se utilizó como sensor encargado de guiar el prototipo, una cámara Kinect que permitió obtener una imagen tridimensional para calcular la profundidad con respecto a los objetos que la rodean, la misma que envía las imágenes al procesador Raspberry Pi 3, que fue programado para la realización del tracking de objetos a través de Python para obtener la posición del objeto donde llegara el actuador final en los tres ejes X, Y, Z. En el diseño se comprueba que el brazo robótico tiene una precisión del 90%. Después de la implementación del brazo mecánico se concluye que es necesario un sistema de iluminación adecuado para obtener resultados con un nivel de fiabilidad justo. Se recomienda tener como medida de seguridad finales de carrera ubicados estratégicamente para brindar protección a la estructura y al operador en caso de alguna falla del sistema.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INTELIGENCIA ARTIFICIAL>, <ROBÓTICA>, <GRADOS DE LIBERTAD>, <RASPBERRY PI 3 (SOFTWARE-HARDWARE)>, <BRAZO ROBÓTICO>, <KINECT (HARDWARE)>, <SEGUIMIENTO DE OBJETOS>

SUMMARY

The main purpose of this project was to design and implement an industrial robot arm with 5 degrees of freedom guided by Kinect. This robot arm was constructed in a 90% with metal (iron) to give it greater resistance and stability while for its movement three DC motors of 2.1 HP were implemented, and they were located in the first three joints due to the torque that is needed in this part of the arm. In the axes of these three motors, sensors (encoders) were placed to be able to control them with more precision, and two servomotors of 4.1kg. cm at the end, giving the robotic arm 5 degrees of freedom.

A Kinect camera was used as a sensor to guide the prototype, this, made possible to get a three-dimensional image to calculate the depth with respect to the objects that surround it. The camera sends the images to the processor Raspberry Pi 3, which was programmed for the realization of the tracking of objects through Python to determine the position of the object where the final actuator will arrive in the three axes X, Y, Z.

This design demonstrates that the robotic arm has an accuracy of 90%. After the implementation of the mechanical arm, it is concluded that a suitable lighting system is necessary in order to get outcomes with an appropriate level of reliability. It is recommended as a security measure; to have end-of-strokes strategically placed to provide structure and operator protection in case of a system failure.

KEYWORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <ARTIFICIAL INTELLIGENCE>, <ROBOTICS>, <DEGREES OF FREEDOM>, <RASPBERRY PI3 (SOFTWARE-HARDWARE)>, <ROBOTIC ARM>, <KINECT (HARDWARE)>, <TRACKING OF OBJECTS>

INTRODUCCIÓN

La tecnología ha venido acogiendo los distintos campos de la vida cotidiana del ser humano. Desde el hogar hasta la industria, pasando por los sistemas de control, de seguridad y de procesos. Sin duda es el industrial el que más progreso y dinero produce por lo que la necesidad de nuevas ideas es imperativa.

En este contexto la visión artificial ha venido a agolpar un gran punto de apoyo sobre los controles industriales y llegando a convertirse en los ojos del nuevo futuro que cada vez está más cerca. Pero no puede funcionar por su propia cuenta y en cuanto al controlador principal, ya la actualidad no quiere ocupar grandes espacios con computadoras portátiles o de mesas, por lo que se ha elegido la placa revolucionaria Raspberry Pi.

Raspberry Pi un microcomputador de costo muy modesto que salió al mercado en el año 2011 y que ha venido actualizándose automáticamente. Aunque en un principio se veía como una plataforma de prueba, dirigida a los estudiantes, actualmente sus recursos se han convertido en verdaderas fuentes de soluciones reales y aplicativas.

La Robótica también forma parte de este proyecto que une sus recursos a la visión artificial y en este caso le da una realidad al tracking de objetos, en donde la primera intención es reconocer un objeto y luego alcanzarlo. Por su valor potencial las aplicaciones han sido dirigidas al campo industrial. Y el presente proyecto también tiene esa misma base de operación.

El presente trabajo fusiona diferentes ciencias como: la Robótica, la Visión Artificial, la Electrónica. La Visión Artificial permite el tratamiento de las imágenes en tiempo real a través del módulo de cámara. Se escogió el algoritmo más óptimo para realizar la segmentación y seguimiento de la pelota.

El prototipo usa actuadores que le permiten desplazarse en el entorno en el que se encuentra y dispone de sensores que ayudan a conservar la integridad del niño y evitar que el robot no sufra averías por choques. A medida que el juego avance el robot emitirá mensajes pregrabados para así lograr un vínculo con el niño y estimularlo en su desarrollo

PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES

Desde el principio de los tiempos el hombre ha tratado de optimizar los trabajos manuales inventando herramientas que pueda lograr lo que a las manos les cuesta. El cobre, el hierro y demás metales han sido los principales aliados del hombre.

Con la era de la industrialización, los brazos hidráulicos, las computadoras y la ingeniería de procesos han automatizado gran parte de la manufactura de los productos. Esto ha hecho que se abastezca al ser humano con lo que necesita, se han abierto plazas de trabajo y se ha vuelto común todas las clases de productos ofertados al consumidor.

Entonces el hombre ya no quería cantidad sino calidad. Los procesos de gran calidad no pueden ser veloces y viceversa. Por lo que siempre se trata de llegar a un límite permitido superando los límites de las capacidades humanas y adoptando las tecnológicas. Es entonces en donde la robótica y la visión artificial se toman de la mano y se han venido ejecutando para los procesos industriales.

Hoy en día se pueden ver la tecnología corrigiendo casi todos los errores en las industrias manufacturas. Por lo que se necesita de nuevas ideas, que sean a la vez más eficientes y de un costo inmejorable para los inversionistas.

Los prototipos tienen que nacer a partir de investigaciones científicas recientes, por lo que el presente trabajo se asoma también a partir de una investigación sobre visión artificial reciente. Se pretende dar entonces un complemento a la investigación mencionada y una realidad robótica y funcional para el mismo.

Sobrado Malpartida Eddie Ángel en 2003 realizó una tesis de maestría titulada “SISTEMA DE VISIÓN ARTIFICIAL PARA EL RECONOCIMIENTO Y MANIPULACIÓN DE OBJETOS UTILIZANDO UN BRAZO ROBOT “, su estudio se basó en la visión artificial uso tracking de un objeto para su localización, procesamiento de imagen en movimiento.

Torres Vidal Ernesto en 2009 realizó un trabajo final de tesis de grado titulada “SISTEMA DE INTELIGENCIA ARTIFICIAL PARA EL CONTROL DE ROBOTS AUTÓNOMOS”, su trabajo se basó en la inteligencia artificial, en el cual encuentra un objeto y usa un sistema de control RF (joystick) para su localización.

Formulación del problema

- ¿Se puede implementar un brazo robótico industrial guiado por Kinect?
- ¿Cinco grados de libertad serán suficientes para alcanzar los objetivos?
- ¿Se puede encontrar de una manera precisa las coordenadas de un objeto en el espacio con Kinect?

Sistematización del problema

- ¿Cuáles serían los tipos de movimientos que podrían satisfacer al usuario del brazo robótico?
- ¿Los tiempos de reacción del brazo robótico industrial satisfacen las necesidades a la hora de actuar?
- ¿Es seguro usar un brazo robótico industrial con un sistema Kinect?
- ¿Cuáles son las ventajas de usar un brazo robótico guiado mediante Kinect?
- ¿Cuáles son las potenciales aplicaciones de un brazo robótico guiado por Kinect?

JUSTIFICACIÓN

Justificación teórica

A los brazos robóticos industriales se los considera como representantes de la mecatrónica. Así como también en muy pocas ocasiones se encuentra una variedad comparable de tecnologías y disciplinas científicas, enfocadas a la funcionalidad y ejecución de un sistema como en el desarrollo y aplicación de un robot.

El diseño de un manipulador sostiene diferentes criterios de funcionalidad y desempeño como son: la capacidad de carga, los grados de libertad, las herramientas de uso, la precisión, la repetitividad, las condiciones ambientales de operación, el costo de manufactura y desarrollo, el mantenimiento, la flexibilidad, etc.

Con el uso de la robótica se trata de hacer que la vida de personas con cierta discapacidad motriz se haga mucho sencilla, por esta razón se establece la creación de un brazo robótico de tipo asistencial, con el fin de que cumpla tareas básicas como acercamiento de objetos hacia lugares específicos, de modo que sirva para el realización de estas labores que se consideran fáciles para una persona sin este tipo de dificultades, por otro lado para alguien con capacidades especiales

son tareas que conllevan cierto grado de dificultad y presenta algún tipo de inconvenientes para el desenvolvimiento adecuado en el medio.

Empleando los conocimientos de robótica y a su vez conociendo la robótica industrial, se busca descubrir si al ser implementados con Kinect puede cumplir la función de identificación, localización y detección adecuada de objetos. Siendo multifuncional y práctico de modo que establezca una interacción.

Justificación aplicativa

Aunque muchos requerimientos pueden ser obvios o estar ya definidos, algunos parámetros centrales del diseño deben ser deducidos de un análisis de los esquemas de trabajo.

El parámetro con el mayor impacto en la complejidad, costo y apariencia del robot es su número de ejes independientes, esto es, los grados de libertad. Ya que son dados por estructuras predefinidas o eslabones como soporte del brazo robótico, mismos que serán diseñados en 3D para su posterior impresión y trayectorias espaciales que el efector final del robot debe seguir.

Los actuadores del dispositivo serán los servomotores con los cuales este se moverá en el espacio, dándoles una libertad de 360 grados, los cuales serán dirigidos por un sistema embebido, el Kinect se lo controlara desde la PC realizando una conexión entre en la computadora y el sistema embebido debido a los recursos bajos que posee el dicho sistema para las altas prestaciones que posee la cámara KINECT.

Hoy en día, con los avances a través del tiempo que ha transformado a la tecnología, se puede adquirir excelentes cámaras y a su vez recursos de hardware para el trabajo en inteligencia artificial, para ello se ha escogido como sensor principal el KINECT por ser uno del recurso más potente y de grandes prestaciones, además usará el software de openCV el cual es una librería de código libre para visión por computadora en tiempo real, escrita en C++ y compatible para Linux.

OBJETIVOS

Objetivos generales

- Diseñar e Implementar un brazo robótico industrial con 5 grados de libertad guiado por Kinect.

Objetivos específicos

- Realizar un estado del arte detallado referente al trabajo ejecutado.
- Diseñar el prototipo de tal manera que sus dimensiones satisfagan a las aplicaciones planteadas.
- Ensamblar los eslabones con los actuadores y realizar la conexión entre el controlador y el software de seguimiento del prototipo.
- Recolectar datos del entorno respecto al tiempo de acción.
- Planificar trayectorias para llegar al espacio predeterminado.
- Desarrollar una aplicación con el Kinect para controlar el brazo robótico.

CAPITULO I

1. MARCO TEÓRICO REFERENCIAL

1.1. Visión Artificial

La visión artificial es una de las ramas de las ciencias que estudian la inteligencia artificial. La V.A, modela las fases de percepción de la imagen, crea programas que puedan reproducirlos a través de un computador, su versatilidad puede apoyar el desarrollo de varios campos y aplicaciones como reconocimiento de caracteres, interpretación de imágenes y escenas, robótica, medicina, etc.

Permite la interpretación de la luz, si esta es buena los resultados serán excelente para analizarlos, convierte a las cámaras inmóviles en dispositivos inteligentes que pueden detectar eventos específicos tales como el seguimiento de un objeto o de una persona dentro de un campo visual, consiste en intentar simular la visión humana, su objetivo es mejorar localidad de una imagen. Sostiene explícita relación con el hardware, software, la V.A demanda de muchos recursos en torno a ciencia, la investigación sobre este campo es rentable.

1.1.1. *Imagen digital*

Una imagen digital es una representación bidimensional de una imagen, mediante el uso de bits ya sean 0 o 1; dichas imágenes son capturadas a través de una cámara fotográfica. Una imagen digital presenta características como: dimensión y espacio de color.

Una imagen digital está formada por una retícula rectangular, esta a su vez está formada por celdas, que son la menor unidad en color de una imagen digital. A la mencionada celda se le conoce con el nombre de pixel. Cada pixel está formado por tres subpixeles: rojo, verde y azul; cada uno con diferente intensidad de brillo (Enrique Benimeli, 2011, p.1).

La resolución de una imagen digital es la cantidad de pixeles por pulgada. A una mayor resolución se obtendrá una información digital más detallada d la imagen. Existen diversos códigos diseñados para representar espacios o modelos de colores de una imagen digital. Para el proyecto se utilizará dos tipos de modelos para el procesamiento de imágenes: el Modelo RGB y el modelo HSV.

La **Figura 1-1:** muestra una imagen digital.

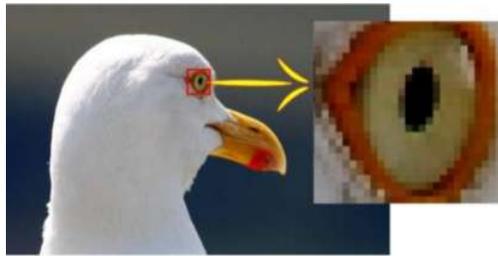


Figura 1-1: Imagen digital

Fuente: Los colores en Informática, 2011.

1.1.2. Estudio de color

Dentro del estudio del color se debe tener en cuenta su punto de partida ya que estos están formados por los tres colores primarios, mismos que permiten el origen de diferentes tipos de colores, de ellos se genera una subdivisión de estos en primarios, secundarios y los terciarios. Los formatos del color tienen una gran variedad, su desarrollo ha ido avanzando gracias a la mejora del hardware y el Software de los cuales cabe mencionar algunos como: HSI, CMY Y RGB.

La Figura 2-1: muestra los colores primarios, secundarios y terciarios.

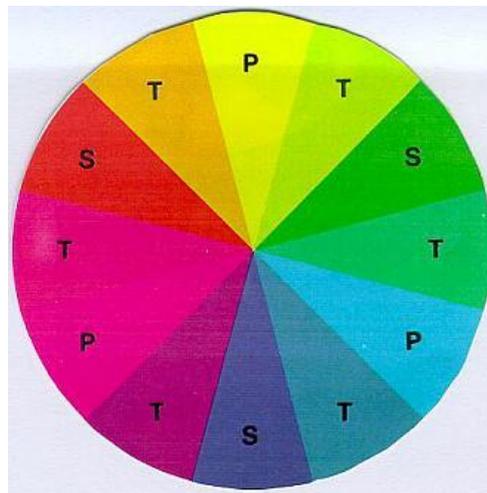


Figura 2-1: Colores primarios, secundarios y los terciarios.

Fuente: Salper Viridiana, 2016

1.1.2.1. Modelo RGB

Es el considerado como uno de los más simples para utilizar, el modelo RGB es el que utilizan las cámaras y sensores que contienen colores primarios es decir el Rojo, Verde, Azul. La

representación de la cantidad de color es de un rango exacto para cada uno de los colores y esto es desde el 0 a 255. Este tipo de formato se usa para enseñar imágenes en una pantalla o un monitor. Al mezclar dos colores a su máxima intensidad sin la presencia de un tercer color obtenemos la siguiente combinación: Rojo más Verde = Amarillo, Azul más Verde = Cian, Rojo más Azul = Magenta.

La **Figura 3-1:** muestra el Modelo de color RGB.

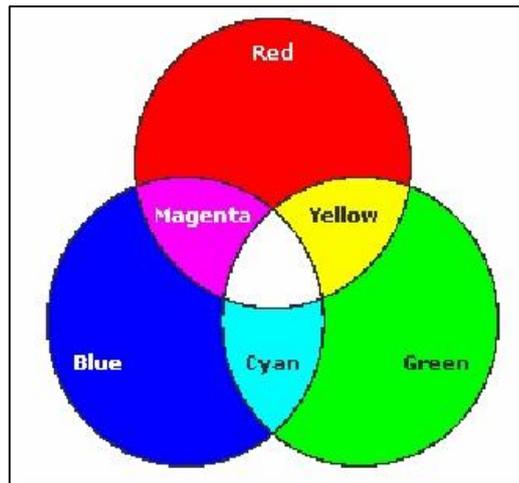


Figura 3-1: Modelo color RGB

Fuente: Salper Viridiana, 2016

1.1.2.2. *Modelo HSV*

El formato HSV: se trasforma de forma no lineal partiendo de RGB, su derivación proviene de las siglas Hue, Saturation, Value, que quiere decir los siguiente:

Matiz: es un grado cuyos valores se encuentran dentro de un rango de 0 a 360 de esta forma los grados ayudan a la representación del color.

Saturación: es la distancia del eje simple o básico de brillo (blanco _ negro), el rango de medida se encuentra entre 0-100 por ciento, es decir que mientras más bajo sea este dato la imagen puede ser más mas gris y menos colorida.

Valor: altura que tiene entre el color blanco y el negro, de la misma forma se puede llegar a medir en grados de c-100, quedando el negro como el cero.

La **Figura 4-1:** muestra el Modelo HSV

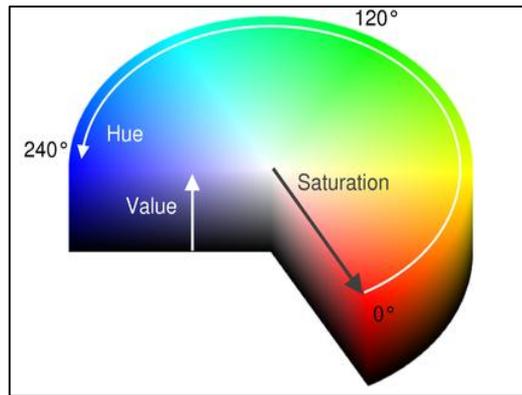


Figura 4-1: Modelo HSV

Fuente: <http://edujoaomiguel.blogspot.com/2015/11/modelos-de-cor.html>

1.1.2.3. Modelo HSL

Este modelo tipo solo se mide por parámetros, (hue, saturation, light) se encarga de precisar la ubicación dentro del cilindro de colores, dentro de sus tres parámetros, el horizontal, el vertical y el centro.

La **Figura 5-1:** muestra el Modelo HSV

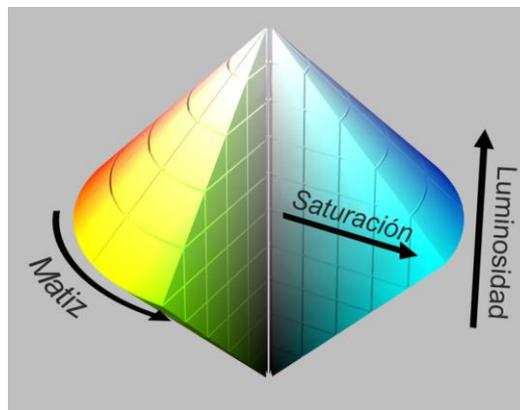


Figura 5-1: Modelo HSV

Fuente: https://es.wikipedia.org/wiki/Modelo_de_color_HSL

1.2. Elementos de un Sistema de Visión Artificial

Un sistema de Visión Artificial está compuesto por cinco elementos principales, denominados: Módulo de proceso, sistema de respuesta en tiempo real, fuente de luz, sensor de imagen, tarjeta de adquisición, en los siguientes apartados se describen la función que cumplen cada uno en el sistema de VA.

1.2.1. Módulo de Proceso

Es el sistema encargado de recibir y analizar las imágenes. La selección del módulo de proceso depende de las necesidades del sistema de Visión Artificial, hay que tener en cuenta la velocidad al procesar la imagen en tiempo real (Aspectos de un Sistema de Visión Artificial, 2005, p.7-8). En el presente trabajo se utilizó como Módulo de Proceso a la placa Raspberry Pi.

1.2.2. Sistema de respuesta en tiempo real

Este sistema usa la información recibida desde la escena para tomar decisiones con respecto al entorno, como pueden ser de movimiento o desplazamiento; mediante actuadores, sensores, baterías, entre otros (Aspectos de un Sistema de Visión Artificial, 2005, p.9-10).

1.2.3. Fuente de luz

La fuente de Luz es la encargada de iluminar la escena a visualizar, este es un apartado de vital importancia; ya que tiene gran influencia en el óptimo desempeño del sistema de VA; pues dependiendo de la cantidad de luz emitida, el fondo y la posición de la cámara en la escena; los colores pueden variar. Dichos parámetros deben ajustarse al objeto que se desea iluminar (Eddie Sobrado, 2003, p 2-3,2-6).

Los tipos de iluminación son los siguientes:

- **Iluminación de Retroalimentación Difusa:** En este tipo de iluminación se coloca la fuente de luz contra la cámara y el objeto a iluminar se encuentra en medio. Por lo general esta iluminación se usa para obtener las características de un objeto, sin embargo, se pierde los detalles del resto de la escena. Además, el objeto o pieza a iluminar debe encontrarse fija.

En la **Figura 6-1:** se muestra la Iluminación Difusa aplicada a una escena u objeto.

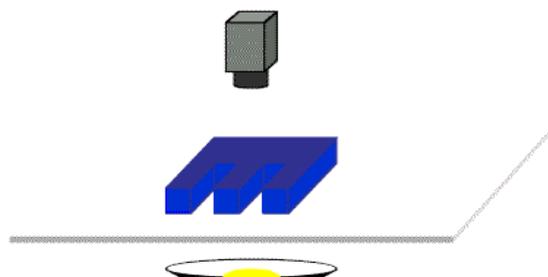


Figura 6-1: Iluminación difusa

Fuente: Sistema de VA de un brazo robot, 2003.

- **Iluminación Frontal:** Es la más usada ya que se enfoca al objeto o escena frontalmente. Sin embargo, se puede tener problemas con objetos que sean demasiado reflectores, provocando brillos.

En la **Figura 7-1:** se muestra la Iluminación Frontal aplicada a una escena u objeto.

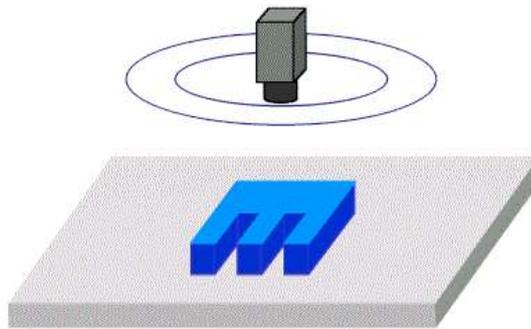


Figura 7-1: Iluminación Frontal

Fuente: Sistema de VA de un brazo robot, 2003.

- **Iluminación Direccional:** Consiste en iluminar desde algún sentido la escena, puede ser desde alguna esquina o desde el centro de la habitación; para así obtener características específicas del objeto.

En la **Figura 8-1:** se muestra la Iluminación Direccional aplicada a una escena u objeto.

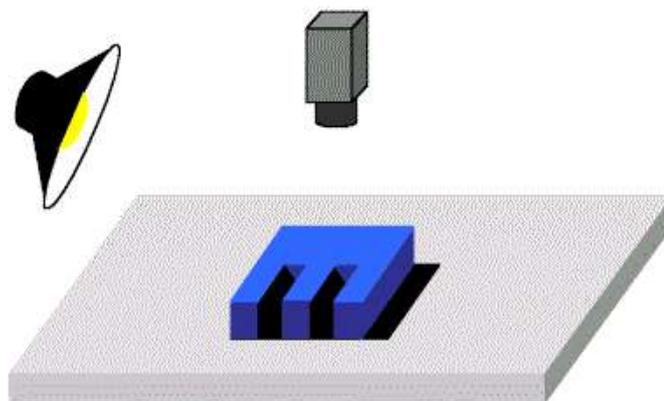


Figura 8-1: Iluminación Direccional

Fuente: Sistema de VA de un brazo robot, 2003.

- **Iluminación Estructurada:** En este tipo de iluminación se usa por lo general una fuente de luz láser; la misma que atraviesa unas rejillas para posteriormente enfocar el objeto o escena. Generalmente se usa para conocer la forma de objetos.

En la **Figura 9-1:** se muestra la Iluminación Estructurada aplicada a una escena u objeto

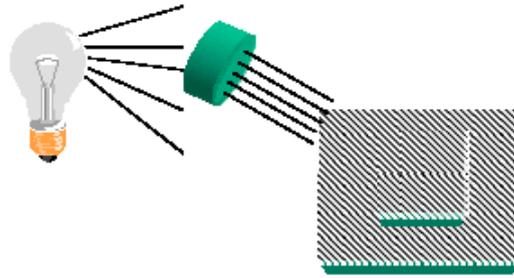


Figura 9-1: Iluminación estructurad

Fuente: Sistema de VA de un brazo robot, 2003.

1.2.4. *Sensor de Imagen*

Este subsistema está destinado a la adquisición y captura de las características del objeto o escena a analizar. Para dicho sistema se utilizan diversos tipos de cámara dependiendo de la necesidad de cada proyecto (Universidad Nacional de Quilmes, Cátedra: Visión Artificial, 2005, p.3).

1.2.5. *Tarjeta de adquisición*

La tarjeta de adquisición es la encargada de transportar la imagen capturada por la cámara hacia el módulo de proceso o computador.

En este componente radican aspectos importantes para el procesamiento óptimo de la imagen como son: la velocidad de transmisión la profundidad del pixel, la velocidad de transferencia de datos.

Todos estos parámetros aseguran que la imagen recibida por el computador conserve en su mayoría las características reales del objeto o escena (Aspectos de un Sistema de Visión Artificial, 2005, p.4).

1.2.6. *Transformaciones Morfológicas aplicadas al procesamiento de imágenes*

Las Transformaciones Morfológicas se encargan de modificar la geometría y forma de un objeto, mediante el uso de herramientas matemáticas que permiten obtener valores en cuanto a las características de forma y estructura de la imagen tratada.

Entre las principales utilidades que brindan las Transformaciones Morfológicas tenemos: separación de objetos unos de otros, obtención de contornos, descomposición de formas de

objetos complejas en formas de objetos simples, supresión de ruidos y simplificación de formas, descripción de objetos basado en área, perímetros y la reconstrucción de elementos distorsionados en una imagen (Técnicas y Algoritmos Básicos de Visión Artificial, 2006, p 55-60).

A continuación, se detallan las principales herramientas matemáticas que ofrecen las Transformaciones Morfológicas.

1.2.6.1. Dilatación Binaria

La dilatación Binaria produce un efecto de engrosamiento de los bordes de los objetos, con esto se logra aumentar el contorno de objetos y unir líneas discontinuas.

Dada una imagen A y un elemento estructural B; ambas son imágenes binarias con fondo blanco; se procede a desplazar el elemento estructural B por toda la imagen A; en caso de coincidir un elemento de la matriz de la imagen A con un pixel del elemento estructural B el pixel en común se pone a uno (Edmans, 2006, p 55).

En la **Figura 10-1:** se ilustra una imagen binarizada y su dilatación.

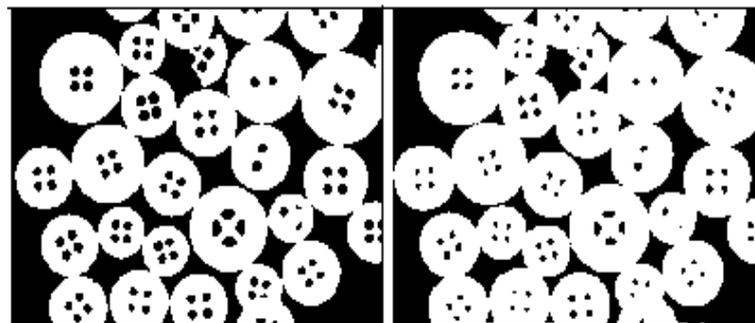


Figura 10-1: Imagen binarizada y su dilatación

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

1.2.6.2. Erosión Binaria

La erosión binaria es una función dual que permite combinar los resultados obtenidos de la dilatación y erosión; mientras que la dilatación engrosa y expande los bordes de los objetos reduce los contornos de los objetos para así separar los objetos que están unidos en pequeñas secciones de sus contornos (Edmans, 2006, p 57).

En la **Figura 11-1:** se ilustra una imagen binarizada y su erosión.

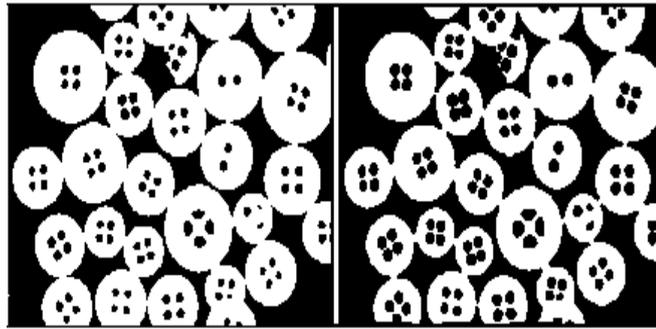


Figura 11-1: Imagen binarizada y erosionada

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

1.2.6.3. Apertura

La operación de apertura de una imagen consiste en realizar primero la erosión y luego la dilatación de la misma. La apertura puede estar formada por una combinación de números de erosiones y dilataciones diferentes y es utilizada para separar objetos de diferentes formas, descomponer un objeto en partes más sencillas, aumentar agujeros que están dentro de objetos, extracción de formas (Edmans, 2006, p 58).

La **Figura 12-1:** se ilustra una imagen binarizada y su apertura.

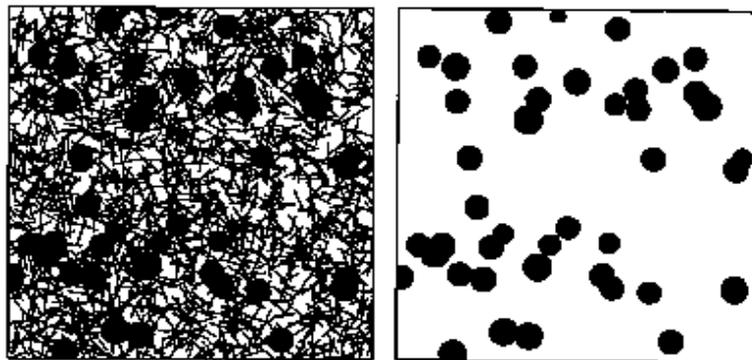


Figura 12-1: Imagen con ruido vs imagen aplicada la apertura.

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

1.2.6.4. Cierre

El cierre es la operación opuesta la apertura, realizando primero la dilatación y luego la erosión de la imagen. La combinación de la dilatación y erosión en segundo lugar ayuda en el filtrado, segmentación de la imagen (Edmans, 2006, p 58).

En la **Figura 13-1:** se ilustra la aplicación de cierre a una imagen binarizada.

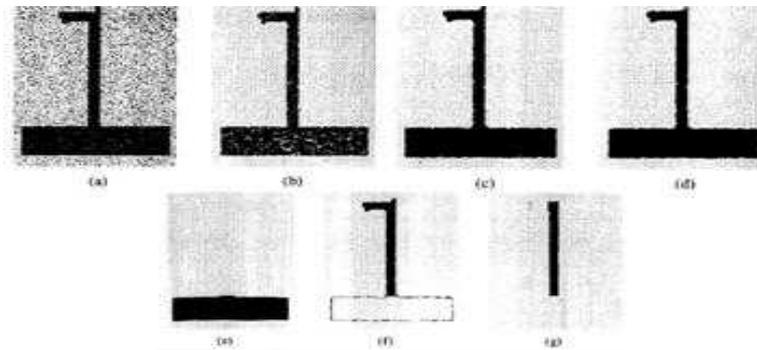


Figura 13-1: Aplicación de cierre a una imagen binarizada

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

1.2.6.5. *Esqueletos*

En esta operación se reducen los objetos de una imagen aun esqueleto del grosor de un pixel. En este tipo de operación el tiempo juega un papel importante ya que solo se puede implementar si se cuenta con el hardware específico para acelerar los cálculos de dicha operación (Edmans, 2006, p 60).

En la **Figura 14-1:** se ilustra el resultado de aplicar la operación de esqueletos a la imagen de la izquierda.



Figura 14-1: Imagen binarizada vs Imagen aplicada operación de esqueletos.

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

1.3. **Etapas de un sistema de Visión Artificial**

Un sistema de Visión Artificial necesita de un conjunto de procesos ordenados sistemáticamente; los mismos que están destinados a procesar las características de una escena u objeto, para que

posteriormente en conjunto con sensores, actuadores; se pueda cumplir con el objetivo específico del sistema (Técnicas y Algoritmos Básicos de Visión Artificial, 2006, p 15).

En la **Figura 15-1**: se muestra el diagrama de un Sistema Visión Artificial.

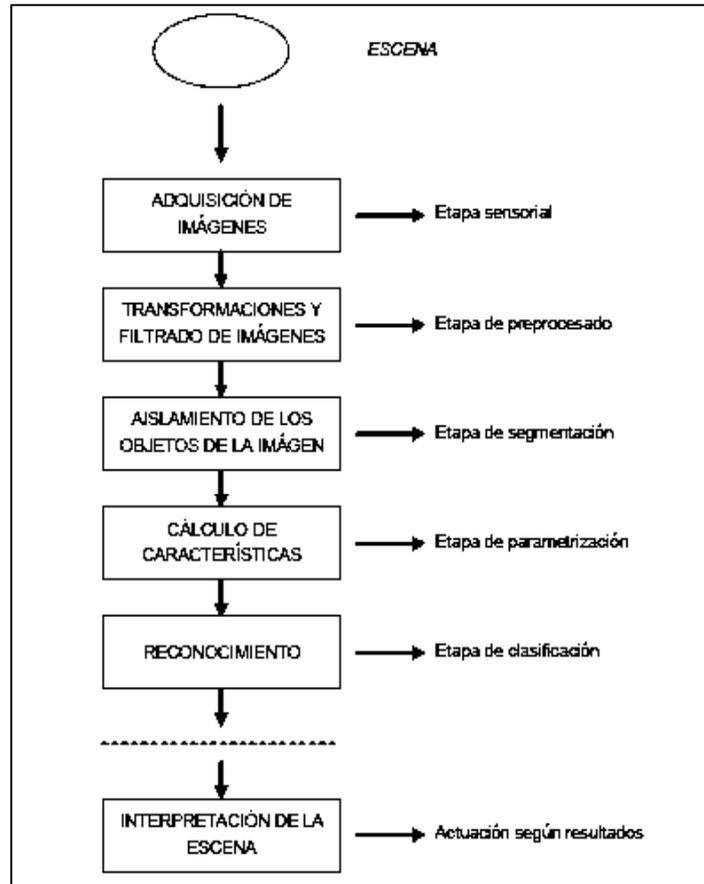


Figura 15-1: Diagrama de bloques de las etapas de un Sistema de Visión Artificial.

Fuente: Técnica y Algoritmos Básicos de Visión Artificial

Un sistema de Visión Artificial debe seguir un conjunto de procesos consecutivos. En primer lugar, se adquiere la imagen o escena a través de la cámara. Luego la imagen capturada es procesada; aquí se obtiene una información más precisa al eliminar ruido en la misma. En estas dos primeras etapas se realiza un tratamiento y análisis de la imagen más simple y global (Cayambe, Castañeda, 2017, p.18-19).

Posteriormente se realiza la segmentación de la imagen; aquí se separa en regiones u objetos de ser el caso. Luego de tener la escena separada por regiones se analiza los objetos característicos como el color, la textura, la forma, el diámetro, el área; etc. En estas etapas se le da un tratamiento a la imagen más personalizado de las regiones que la conforman (Cayambe, Castañeda, 2017, p.18-19).

Al tener la imagen separada en regiones es posible seleccionar la región u objeto a la que se desea interpretar en un entorno estructurado y en tiempo real.

Una vez cumplido el proceso descrito actuará un sistema de respuesta, que en conjunto con el Sistema de Visión Artificial cumplirán el objetivo final del procedimiento (Cayambe, Castañeda, 2017, p.18-19).

En la Visión Artificial intervienen dos subsistemas principales; el primero se encarga de adquisición de la imagen, mientras que el segundo se encarga de procesar dichas imágenes.

1.4. Cámara digital

La cámara digital es un tipo de cámara fotográfica que al obtener una imagen también la guarda, cuentan con una variada de funciones, además de los dispositivos que permiten grabar sonidos e incluso videos.

La primera cámara que registraba imágenes dentro de un computador aparece en los años 1988, es capaz de grabar dentro de una tarjeta de memoria de unos 16 mg, además de estar incorporada de una pequeña batería para el almacén de estos datos, las primeras en salir al mercado en el año 1991 eran las Dycam Model 1, para el proceso de traslado de los archivos de la cámara al computador se lo realizaba por medio del cable de datos.

En el mismo año kodak lanzó al mercado un nuevo modelo del DCS-100, luego su línea se fue extendiendo, a veces la marca Nikon trabajaba con un sensor de 1.3 megapíxeles, los cambios a los formatos digitales como lo son JPEG y MPEG, comprimían los archivos para que estos puedan ser almacenados.

En la **Figura 16-1**: se muestra la cámara digital



Figura 16-1: Cámara digital

Fuente: <http://www.nikon.com.mx/>

1.5. Tracking de objetos

El tracking de objetos o seguimiento de objetos, es un proceso que permite en determinando tiempo obtener la ubicación de uno o de varios objetos móviles, este proceso se realiza mediante el uso de una cámara, que a su vez tiende a tardar un poco debido a la cantidad de información que tiene un video, conjuntamente con las técnicas de reconocimiento, siendo determinante en este procedimiento se aumenta su complejidad.

La cámara determina la ubicación del objeto y la estructura a través de píxeles, obteniendo la posición y el tamaño del objeto en tiempo real, de esta forma se disponen a calcular su movimiento, considerando la compleja relación que tiene la proyección de la figura con el objeto, sin embargo, no tiene solo dependencia del moviendo del objeto, también influyen otros factores, esto conlleva a que su desplazamiento se convierta en una tarea un tanto complicada.



Figura 17-1: Seguimiento de objetos

Fuente: wikipedia.org/wiki/Seguimiento_de_objetos

1.5.1. Método de Umbralización

El método de umbralización es considerado como uno de los más principales, su propósito es transformar una imagen de una escala de gris a otra con solo dos niveles, de esta forma los elementos serán independientes del fondo, clasificando los píxeles de la imagen como fragmentos de ella.

En la **Figura 18-1:** muestra los Objetos claros sobre fondo oscuro



Figura 18-1: Objetos claros sobre fondo oscuro

Fuente: https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_03_04/sonificacion

Su extracción se la realiza seleccionando el umbral de intensidad T que divide los dos objetos. Con esta forma cualquiera de los puntos (x, y) para $f(x, y) > T$, integrará al objeto, de lo contrario se integrará al fondo. Usualmente se pueden dar dos casos el umbralización: thresholding y histograming. En ambos se necesita aplicar un valor igual de gris, para evaluar la intensidad en cada uno de los pixeles.

En Thresholding: si el fondo no es tan claro que el objeto entonces es lo determina como parte del elemento, si no es así, pasa a formar parte del resto de la imagen.

El Histograming: la parte importante de la imagen se la haya por medio de un análisis de la misma tomando un rango para encontrar el objeto.

1.5.2. Segmentación orientada a regiones

Este tipo de segmentación se centra en regiones predilectas de una imagen, parte de las regiones que tengan pixeles con características homogéneas, Las regiones que se obtienen están vinculadas dependiendo de la longitud escogida. Los pixeles deben cumplir similitudes para poder pertenecer a una determinada región como: el color, la textura entre otros.

La **Figura 19-1:** muestra la Segmentación orientada a regiones



Figura 19-1: Segmentación orientada a regiones

Fuente: <http://slideplayer.es/slide/3767498/>

1.5.3. Basada en bordes

Las regiones y los bordes mantienen una relación entre sí, a los bordes se los expresa como cambios de 2 regiones en una etapa gris distinta la una de la otra, y se usa para segmentar bien. En su resultado se logra precisar cálculos con relación a los bordes de un objeto.

La **Figura 20-1:** muestra la imagen basada en bordes

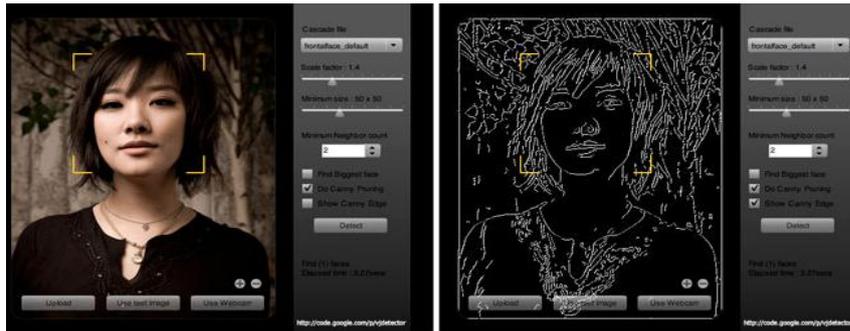


Figura 20-1: Basada en bordes

Fuente: <https://visionartificialparatodos.files.wordpress.com/2011/01/binary.png>

1.5.4. Por colores

La segmentación por colores es usada por su viabilidad y variabilidad al conjuntarse diferentes métodos. Cuenta con algunos procedimientos con un mismo fin: segmentar el objeto por el color requerido. El más práctico y sencillo es exponer un determinado color y valorar cada uno de los píxeles para observar su similitud con relación a la tonalidad. Uno de sus inconvenientes suele aparecerse cuando el fragmento contiene demasiados objetos con tonos iguales, causando que la segmentación no sea exclusiva. La **Figura 21-1:** muestra la segmentación por color

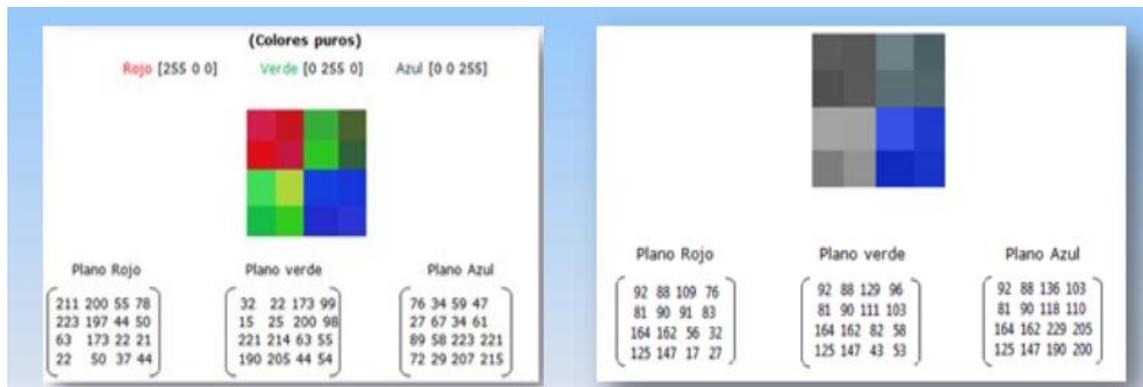


Figura 21-1: Segmentación por color

Fuente: <http://slideplayer.es/slide/4109617/>

1.6. Robótica

La robótica es la combinación de múltiples tareas, Estudia el robot, ha ido evolucionando progresivamente, se la podría definir con la elaboración y utilización de aparatos mecánicos, automáticos, capaces de realizar tareas automáticas entre otras labores.

Su concepto hace referencia a labores que requieren esfuerzo. La mayoría trabaja en la industria, bajo un sistema adaptado para ellos. Las características de la robótica tienen que ver en sus grados de Libertad, estos determinan su movimiento y la posición del robot. Sus derivaciones pueden ser: álgebra, los autómatas de programación, la mecánica, la informática, electrónica etc. la robótica se encarga de todo aquello que abarca a los robots como: prueba de motores, los neumáticos, sensores de movimiento entre otros.

1.7. Tipos de robots

Los robots son utilizados para realizar trabajos de tipo mecánico, su monitoreo se realiza través de un computador. Facilitan las tareas rutinarias de un proceso; por ende existen diversos tipos de robots, enfocados a una aplicación en particular. En los siguientes apartados se describen los principales.

1.7.1. Robots Industriales

La creciente utilización de robots industriales en el proceso productivo, ha dado lugar al desarrollo de controladores industriales rápidos y potentes, basados en microprocesadores, así como un empleo de servos en bucle cerrado que permiten establecer con exactitud la posición real de los elementos del robot y su desviación o error (Diseño y Programación de Algoritmos para Robot Móviles, p. 14).

Los robots industriales se clasifican en:

- **Manipuladores:** Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos mediante los modos: manual, de secuencia fija y de secuencia variable.
- **Robots de repetición o aprendizaje:** Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario durante la

fase de enseñanza se vale de una pistola de programación con diversos pulsadores o teclas, o bien de joysticks, o bien utiliza un maniquí, o desplaza directamente la mano del robot.

- **Robot con control por computador:** Son manipuladores o sistemas mecánicos multifuncionales, controlados por un computador, que habitualmente suele ser un microordenador. El control por computador dispone de un lenguaje específico de programación, compuesto por varias instrucciones adaptadas al hardware del robot, con las que se puede diseñar un programa de aplicación utilizando solo el ordenador. A esta programación se le denomina “textual” y se crea sin la intervención del manipulador.

1.7.2. Robots inteligentes

Son similares a los del grupo anterior, pero tienen la capacidad de poder relacionarse con el mundo que les rodea a través de sensores y de tomar decisiones en función de la información obtenida en tiempo real. De momento, son muy poco conocidos en el mercado y se encuentran en fase experimental, donde grupos de investigadores se esfuerzan por hacerlos más efectivos, al mismo tiempo que más económicamente asequibles. El reconocimiento de imágenes y algunas técnicas de inteligencia artificial son los campos que más se están estudiando para su posible aplicación en estos robots (Diseño y Programación de Algoritmos para Robot Móviles, p. 14).

1.8. Brazo robótico

El brazo robótico o también llamado mecánico usualmente se lo puede programar, sus funciones son similares a las que tiene un brazo humano, de ahí su complejidad. Sus partes están interconectadas por medio de articulaciones, las cuales ayudan a un movimiento coordinado.

El ordenador se encarga de su control, realizando la rotación de varios motores individuales, permitiendo su precisión en los movimientos, formando parte indispensable de este proceso, a su vez los sensores de movimiento se encargan de que se mueva lo necesario, a la hora de desplazarse de un punto a otro.

El sistema de un brazo robótico está compuesto por una estructura mecánica, transmisiones, actuadores, sensores, elementos finales y un controlador. Su constitución física es similar a la anatomía de los brazos de un ser humano y por lo general se hace referencia a los componentes del robot con los nombres de su parte correspondiente en la extremidad de una persona. Como por ejemplo hombro, codo, brazo, muñeca, entre otros (Romero, s.f.).

1.8.1. Configuraciones para un brazo robótico

Existen cuatro configuraciones básicas para un brazo robótico. La primera es la configuración cartesiana, que consta de tres articulaciones prismáticas. La segunda configuración es la cilíndrica, la cual tiene dos articulaciones prismáticas y una de rotación. La configuración polar o esférica está formada por dos articulaciones rotacionales y una prismática. Finalmente, la configuración angular tiene tres articulaciones rotacionales (Ollero, 2001).

1.8.2. Articulación

Una articulación es la parte de la estructura del robot mediante los cuales se unen los eslabones y permiten un movimiento relativo entre los mismos. Por lo general cada articulación que se aumenta en el robot, incrementa también un grado de libertad en el mismo. La mayoría de robots industriales cuentan con seis o menos articulaciones, para así tener un mejor control y precisión de las mismas (Romero, s.f).

Existen dos tipos de articulaciones, las que se usan más comúnmente en robótica son las de rotación, que proveen al robot un grado de libertad rotacional alrededor del eje de la articulación. Las otras articulaciones usadas son las prismáticas, las cuales permiten realizar un desplazamiento a lo largo del eje de la articulación (Ollero, 2001).

Estos dos tipos de articulación se muestran en la **Figura 22-1:**

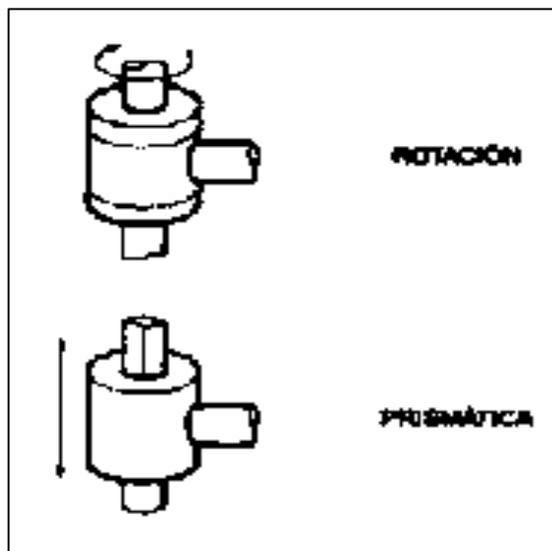


Figura 22-1: Articulación de rotación y prismática

Fuente: Ollero, 2011

1.8.3. *Eslabón*

Son elementos estructurales sólidos que forman parte del sistema mecánico del robot. Para juntar varios eslabones entre sí, es necesario tener articulaciones entre ellos. Los eslabones del robot junto con las articulaciones van a definir el alcance del robot y su espacio de trabajo, y también van a ser factores determinantes de la maniobrabilidad de los mismos (Romero, s.f.).

1.8.4. *Grados de libertad*

Los grados de libertad son la cantidad de parámetros independientes que determinan la posición del elemento terminal del brazo robótico, el número de grados de libertad por lo general coincide con el número de eslabones de la cadena cinemática (Ollero, 2001).

Para posicionar y orientar un objeto en el espacio de cualquier manera deseada, se necesitan seis parámetros, tres para la posición y tres para la orientación. Es por esto que por lo general los brazos robóticos industriales en su mayoría tienen seis grados de libertad. Cuando el número de grados de libertad del robot excede los necesarios para que cumpla con su tarea se suele decir que un robot es redundante (Romero, s.f.).

1.8.5. *Espacio de trabajo*

Se define el espacio de trabajo de un robot el volumen dentro del cual puede desplazarse su efector final. Este volumen está restringido por el tamaño de los eslabones del brazo robótico y los límites de giro de sus articulaciones. El espacio de trabajo es clasificado como regular o irregular dependiendo de la configuración del brazo robótico. Un espacio regular es característica de las configuraciones cartesiana y cilíndrica, mientras que para una configuración polar, el espacio de trabajo por lo general es irregular (Romero, s.f.).

1.9. Software

Para la implementación de un brazo robótico es necesario el uso de varios softwares propios de un Sistema de Visión Artificial y de la robótica. Los softwares empleados en el presente proyecto son: Python, Raspbian Jessie, OpenCV, SolidWorks; a continuación, se describen cada uno.

El lenguaje de programación sirve para configurar un dispositivo de control para que éste pueda cumplir con un funcionamiento requerido.

1.9.1. Python

Es un lenguaje de programación que hace énfasis en una síntesis que favorece a un código completamente claro, este tipo de lenguaje soporta más de un paradigma en programación. Las principales ventajas de este software son:

- La ubicación de objetos
- La programación imperativa y un tanto de la funcional
- Multiplataforma utiliza un tipado completamente dinámico, dirigido por Python software foundation tiene su licencia de código abierto, misma que lleva el nombre de su administrador.
- Además, es compatible con GNU, aparte de las versiones 2.1.1 existen ciertos problemas de compatibilidad en versiones anteriores.

Fue creado a finales de los 80, por el Guido Van Rossum su nombre fue dado por la afición de sus creadoras al humorista Monty Python.

1.9.2. Opencv

Es una potente librería de visión artificial y machine learning, tiene licencia BSD, lo que permite utilizar y modificar el código. Se encuentra escrito en C/C++ mismo que lo hace `practico para su desarrollo, mantiene una muy buena comunidad alrededor de unos 50000 usuarios y por lo menos unos 7 millones en descargas.

OpenCV es bastante comercial y tiene aproximadamente unos 2500 algoritmos.

Entre sus funciones principales se encuentra la identificación de los objetos, las caras, la clasificación de aquellos actos humanos, el seguimiento de objetos etc. Open Source de Visión Artificial prefiere a OpenCV para su desarrollo por su capacidad.

Las aplicaciones más conocidas son las siguientes:

- Detector de movimiento
- Detección de parámetros en dos y tres dimensiones.
- Segmentación y reconocimiento.
- Detección de Objetos.
- Calibración de cámaras.

Funciones básicas:

- **Core:** Se definen las estructuras de datos básicas que usan el resto de los módulos.
- **Imgproc:** módulo procesamiento de imágenes como: filtrado lineal / no lineal, transformaciones afines, conversión del espacio de color, histogramas.
- **Video:** módulo de análisis de video que incluye algoritmo para la estimación del movimiento.
Extracción de fondo.
- **Calib3d:** Algoritmos básicos de visión múltiple como calibración de cámaras estéreo y otras cámaras.
- **Features 2d:** detectores de características.
- **Objdetect:** detección de objetos instancias o clases predefinidas como: caras, ojos, gente, coches.
- **Highgui:** Todo lo relacionado a la interfaz gráfica de OpenCV. Permiten importar imágenes y video.
- **GPU:** algoritmos acelerados x hardware para distintos módulos OpenCV.

1.9.3. *Matlab*

El tipo de trabajo que realiza dinámico para desarrollar algoritmos analizar y visualizar datos y cálculos numéricos, tiene la capacidad de solucionar problemas computación usando lenguaje ya sea en CC++, Fortran, no se necesita detallar las variables a usarse, por ello se debes ser cuidadosos y asegurarse que el tipo de variables no contengan el mismo nombre de otras en diferentes programas, para evitar las posibles dificultades.

- Dentro de sus aplicaciones está también procesar imágenes, comunicaciones, diseño de control, prueba y medición, etc.
- A su vez, es necesario guardar un poco de memoria, en el caso de las matrices muy grandes atribuyéndole cualquier valor, si existe un excesivo uso en la memoria también es posible su eliminación siempre y cuando estas variables no se vayan a usar.
- Los programas que generalmente se escriben en Matlab admiten en su mayoría estructuras de programación muy estándar ej: If, for White.
- Su creación se dio por Cleve Moler en el 1984, se realiza la invención de diferentes versiones su objetivo principal fue la creación de subrutinas escritas en fortran, se estimaba que en 2014 su uso fue de alrededor de un millón de usuarios, en un entorno académico y dentro del mundo empresarial.

- Como programa de cálculo numérico, su efectividad es mucho mejor si se diseñan los algoritmos, en matrices y en vectores. Su versión más actual es Matlab R2016.

Operaciones que realiza:

- Vectores y matrices
- Cálculos
- Creación de vectores y matrices
- Operación y gráficos

Sus características:

- Lenguaje de alta nivel para computación técnica
- Entorno de desarrollo para gestión de código, archivos y datos
- Herramientas interactivas para exploración iterativa, diseño y resolver problemas
- Funciones matemáticas para algebra lineal, estadística, análisis de Fourier, filtrado, optimización e integración numérica.
- Gráficos en 2-D y 3-D de las funciones de visualización de los datos
- Herramientas para creación de interfaces personalizadas graficas
- Funciones que integran algoritmos con base en MARLAB que contienen aplicaciones extras. Idiomas, como C, C++, Fortran, Java, COM, y Microsoft Excel

1.9.4. Msc Adams

MSC.Adams es un software de simulación de sistemas mecánicos por medio de modelado, análisis, y capacidades de visualización. Esto permite a los usuarios crear prototipos virtuales reales simulando el comportamiento de movimientos lleno de sistemas mecánicos complejos, sobre sus ordenadores y rápidamente analiza múltiples variaciones de diseño hasta que un diseño óptimo sea alcanzado. Esto reduce el número de prototipos físicos costosos, mejora la calidad de diseño, y reduce radicalmente el tiempo de desarrollo de producto.

En 1977 se demostró por primera vez, mediante la simulación de un sistema tridimensional de mecánica y dinámica, las predicciones de las simulaciones que fueron confirmados a través de la verificación experimental. Desde entonces, MSC.Adams ha demostrado y comprobado con una amplia y diversa gama de aplicaciones.

1.9.4.1. *Visión general de Msc Adams*

Como la mayoría de los programas multicuerpos, MSC Adams toma como entrada el número de cuerpos en el sistema mecánico que está siendo modelado. Los organismos se denominan partes, que pueden ser rígidas o flexibles, y las propiedades de masa para cada parte debe estar definida. Las piezas que están conectadas entre sí y se mueven con respecto a una, que están conectadas a través de las articulaciones (Hernández Fraile, Alejandro, 2011, p. 56-70).

El grado de libertad Modelado, Simulación y Análisis con MSC.Adams. Aplicación a Rodamientos con Defectos en una Pista de Rodadura ~ 72 ~ de una parte tiene relación con otra parte, que depende de las características de la articulación, por ejemplo, un control deslizante permite la traducción de una parte en relación a otra a lo largo de un eje, una rotación de revolución conjunta sobre un solo eje, una rotación esférica conjunta sobre los tres ejes, entre OTROS (Hernández Fraile, Alejandro, 2011, p. 56-70).

La conexión de elementos de fuerza, tales como muelles de suspensión y amortiguadores de simulaciones basadas en el vehículo, también pueden incluirse, así como los detalles de componentes, tales como casquillos flexibles, cinemáticos, espacios comunes o juego libre.

Además, en los generadores de simulaciones basadas en vehículos en movimiento y los controladores se pueden desarrollar para impulsar, dirigir y guiar. Los módulos de software diseñados para esta tarea están disponibles (Hernández Fraile, Alejandro, 2011, p. 56-70).

El software MSC.Adams genera las ecuaciones matemáticas que describen la dinámica del sistema mecánico, y el software procede a encontrar soluciones para cada paso de tiempo en la simulación. La simulación construye una imagen de cómo, en cada paso de tiempo, cada parte en el modelo se mueve en el espacio de 3 dimensiones. Cuando la simulación se ha completado, todos los movimientos y las fuerzas están a disposición del usuario para su revisión y para su posterior análisis, ya sea dentro MSC.Adams o utilizando el software de ingeniería (Hernández Fraile, Alejandro, 2011, p. 56-70).

1.10. Kinect

El código de Kinect se ha usado como una plataforma en algunas investigaciones en V.A. permite la relación con la consola sin contacto físico solo por medio del reconocimiento de la voz, de los gestos, del cuerpo, también recrea un avatar a través de escaneo del rostro, el sensor ayuda al proyector para el movimiento corporal de una forma más profunda (www.xbox.com).

Microsoft lanzó al público su propio controlador de Kinect mismo que funciona en Windows 7 y 8. El hardware se basó por el diseño tecnológico 3D color, Se ha usado como una clase o una función en el lenguaje orientado a objetos C#, facilitando la construcción de aplicaciones (www.xbox.com).

El convenio realizado con Linux fue un controlador de código abierto. Sin embargo, los controladores libres aún se siguen desarrollando y son muchos los programadores que lo usan (www.xbox.com).

La **Figura 23-1:** muestra el dispositivo Kinect.



Figura 23-1: Kinect

Fuente: <http://www.xbox.com/en-US/xbox-360/accessories/kinect>

1.10.1. Menú de componentes

Kinect posibilita la interacción con el computador por medio del adaptador USB. Tiene la facilidad de ajustar la cámara según una inclinación deseada (www.xbox.com). Posee unos 27° de trabajo verticales.

- **Generales:** Infrarrojo (IR) CMOS camera y protector Infrarrojo -830, 60mW diodo laser.
- **Audio:** Micrófonos
- **Controladores:** Motor y Acelerómetro (3- axes)
- **Procesador y Memoria:** Prime Sense Chip PSI080 -A2 y 64 MB DDR2 SDRAM

- **Kinect:** Color Camera 640 x 480; 640x480 @fps output, IR Camera: 1280 x 1024, 640x480 @30 fps output, Rango de operación (Sensor Depth). Rango= 0,8 - 3,5, FOV= 58° H, 45 V, 70D, Resolución espacial (@ 2m distancia) = 3mm y Resolución Depth (@ 2m distancia) = 1cm.

1.10.2. Tratamiento de la visión del Kinect Xbox 360

En la doble visión que posee el KINECT se lo considera una de las características más importante de su creación, por otro lado, también se la evidencias como uno de sus aspectos negativos, luego de las ganancias que se logran en las funciones de su código, al no existir precisión en la visión RGB y la Depth con respecto a su dimensión y posición. Por defecto las cámaras no mantienen una misma resolución provocando errores en los cálculos de terminación.

La **Figura 24-1:** muestra el tratamiento de la visión del Kinect.

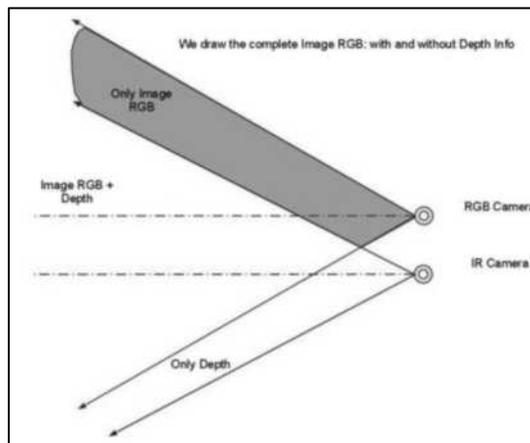


Figura 24-1: Tratamiento de la visión del Kinect

Fuente: <https://landerpfc.files.wordpress.com/2011/03/verticalblackwideline.jpg>

1.10.3. Bits de información Depth

El retorno de la información del sensor infrarrojo de Kinect constituye un total de 16 bits con una distancia corta, sus tres primeros son empleados por el SDK Kinect en aplicaciones mismo que comprende como gestos a los movimientos corporales para la ejecución de una determinada tarea, y aun así quedan disponibles 12 bits restantes.

Dentro del trabajo de aplicación son necesarios los próximos cinco bits, ya que sus datos son suficientes para la indagación requerida. El examen realizado sobre el desarrollo del Kinect en

sus diferentes categorías o rangos, se las puede apreciar en la imagen 4 con algunos resultados aproximados de distancia.

La **Figura 25-1:** muestra el rango de bits.

RANGO DE BITS	DISTANCIAS APROXIMADAS(cm)
0 [0 - 256]	[0 - 74]cm
1 [256 - 512]	[74 - 110] cm
2 [512 - 768]	[110 - 172]cm
3 [768 - 1024]	[172 - 328]cm
4 [1024 - ?]	-

Figura 25-1: Rango de bits

Fuente: Ilbay Luis, 2015

1.10.4. *Fórmula matemática de transformación*

Los cambios en los que mayormente se evidencia su profundidad, está presente dentro de las distancias cortas, las distancias extensas mantienen una diferencia paulatina, su estudio no satisface una sola formulan, por lo tanto, se ha denotado su descripción dentro de una curva no lineal.

La **Figura 26-1:** muestra los datos en Kinect.

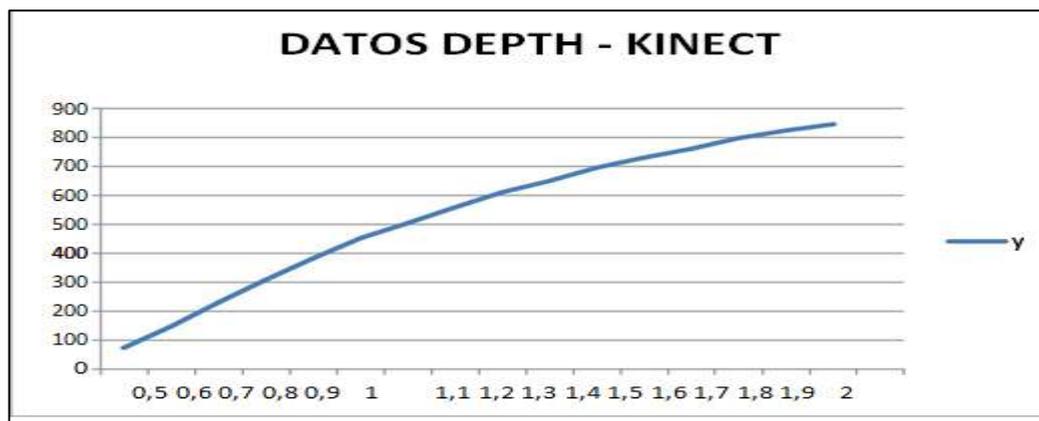


Figura 26-1: Datos en Kinect

Fuente: Ilbay Luis, 2015

Los resultados poco costosos y sencillos, gracias a la generación de una formulan creada matemáticamente para cada grupo de bits, como se muestra en la imagen 5. El uso de una ecuación

de línea es imprescindible ya que cada uno de los grupos de bits tiene cierta linealidad, no se ocupan formulas numéricas su uso ha provocado más errores que aciertos, a este dato de profundidad se lo conoce como PVAL.

1.11. Motores

Un motor es la parte sistemática de una máquina capaz de hacer funcionar el sistema, transformando algún tipo de energía (eléctrica, de combustibles fósiles, entre otras.), en energía mecánica capaz de realizar un trabajo.

1.11.1. Motores DC

Es un aparato que permite transformar energía eléctrica en mecánica, generando que su movimiento sea rotativo. A su vez se produce cuando una corriente electrónica se abre paso por el cable conductor inmerso en un campo magnético, la fuerza magnética crea un par y este a su vez ocasiona que se dé el giro del motor.

La **Figura 27-1:** muestra un motor DC.



Figura 27-1: Motor DC

Fuente: tienda.bricogeek.com/motores/

1.11.2. Micromotores

Los Micromotores son implementos que se utilizan en una variedad de campos, es muy práctico y tienen una rapidez que lo ha transformado en una herramienta única para la realización de distintas acciones, ya sean estos en el ámbito de pulir o perforar, generalmente su uso es de vital utilidad en la odontología, joyería y además podología.

La **Figura 28-1**: muestra un micromotor.



Figura 28-1: Micromotor

Fuente: <http://www.medicaexpo.es/>

1.11.3. *Servomotor*

Este modelo de motor se caracteriza especialmente porque se posicionan por medio de un Angulo que va de entre 0 -180°, de esta forma posibilita que su control pueda tener una mayor precisión, el PWM o también llamada amplitud de pulso, señala el ángulo de rotación que debe hacer el motor, los datos de los valores van desde 1 a 2 ms, de esta forma al número 1 le tocara el (0 grados) y al número 2 (180 grados), para sostener a este modelo de motor en una posición adecuada, es importante él envió de los pulsos ya que estos deben ser de forma continua del ancho al posicionamiento deseado.

La **Figura 29-1**: muestra un servomotor



Figura 29-1: Servomotores

Fuente: <http://www.directindustry.es/>

1.11.4. Motor paso a paso

A este tipo de motores también se los llama motores de pasos, sus bobinados constituyen parte de estator, su motor también logra ser un imán estable, sus controladores han sido creados de tal forma que pueda mantener una posición fija, asimismo pueda rotar de un sentido a otro.

En su mayoría su avance se da por frecuencias de audio, permitiéndoles rotar con mayor velocidad, un adecuado controlado permite que avance o se detenga en un instante y en una determinada posición.

Su comportamiento es totalmente diferente a de los de corriente continua, no rotan con libertad por si solos, su avance se da a pasos pequeños, se diferencia de los CC, con relación a la relación y el torque, en su mayoría tu torque o giro de baja velocidad.

Una de sus características es la detención de par motor o par giro, produciendo la firmeza en su detención cuando este no se encuentra rotando, al estar inmerso en este estado su fuerza de trabajo continúa aplicada en su eje.

El control de este tipo de motor no está aplicando directamente al pulso eléctrico, para el avance tiene a las bobinas deberán estar alimentadas es una secuencia apropiada, asimismo al invertir la frecuencia gira al lado opuesto.

Característica de un Motor pasó a paso

- Voltaje
- Resistencia eléctrica
- Grados por paso

Tipos de motores pasó a paso

- Imán permanente
- Reluctancia variable
- Hídricos
- Unipolares
- Bipolares
- Reluctancia variable

La **Figura 30-1**: muestra un motor paso a paso.



Figura 30-1: Motor paso a paso

Fuente: <http://www.directindustry.es/>

1.12. Puente H

Un puente H es un controlador electrónico que permite establecer la dirección de giro de un motor, estableciendo de esta forma el giro en sus dos sentidos, es decir delante/ atrás. Son utilizados ampliamente en proyectos de robótica y convertidores de potencia.

Los puentes H se puede conseguir fácilmente en el mercado bajo especificaciones estandarizadas tanto en corriente y voltaje de alimentación, no obstante, también se puede diseñar y crear uno con dispositivos electrónicos existentes en el mercado.

El control de velocidad y sentido de giro de los motores se realiza mediante el uso de un puente h el cual se encarga de la administración del sentido e intensidad del flujo de corriente para así poder controlar tanto la velocidad como el torque generado por el motor (Cadena Contreras, y otros, 2011, p.55).

Para la selección de un puente h se debe tener en cuenta la potencia entregada por los motores a plana carga y el voltaje de alimentación de los mismos, de esta forma no se tendrá el riesgo de dañar o en el peor de los casos quemar el dispositivo.

En la **Figura 31-1**: se muestra el controlador IFI VEX Pro cictor 885 empleado para el control de los motores con caja reductora del brazo robótico metálico.



Figura 31-1: Controlador IFI VEX

Pro Víctor 885

Fuente: <https://images-na.ssl-images-amazon.com>

1.13. Sensores

Los sensores son dispositivos que tienen la capacidad de detectar todo tipo de magnitudes como las físicas y las químicas, y convertirlas en variables eléctricas para su análisis respectivo.

En la actualidad existen todo tipo de sensores como, por ejemplo: sensores de velocidad, sensores de aceleración, sensores de temperatura, sensores humedad, sensores de movimiento, sensores ópticos, sensores de radiación, sensores de fuerza y deformación, sensores de distancia, sensores de caudal, sensores táctiles, sensores acústicos, sensores de desplazamiento, sensores de presión, entre otros.

1.13.1. *Encoder*

Los encoder son sensores que se utiliza para medir la velocidad de giro de un motor. Estos dispositivos convierten el movimiento angular en un código electrónico digital para su análisis respectivo. Existen dos tipos de encoder: Encoder absolutos y Encoder incrementales.

Los encoder tienen una gran variedad de aplicaciones como se describe a continuación:

- Son utilizados en robótica.
- Se utiliza para los plotters.
- En equipos y dispositivos médicos.
- Para los indicadores en x/y.
- Maquina taladradoras, entre otras.

La **Figura 32-1:** muestra el encoder utilizado en el brazo robótico metálico.

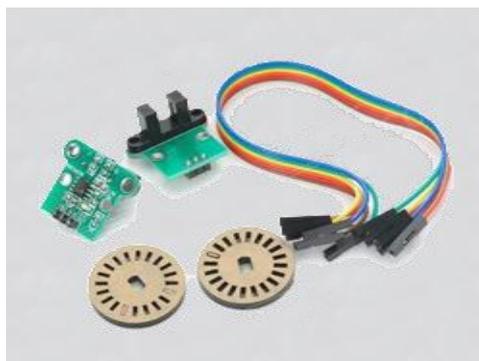


Figura 32-1: Encoder

Fuente: <http://mco-s1-p.mlstatic.com/>

1.14. Impresión 3D

La impresión en 3d ha ido ganando terreno dentro del mundo tecnológico, haciendo que un objeto tridimensional sea creado por medio de añadiduras de capas continuas. Este tipo de impresiones permite dar volumen a una figura, y tienen cierta relación con los Plotter, mismos que están compuestos por un cabezal y la tinta, que implica movimientos dentro de un eje, posibilitando la creación de puntos y líneas. A su vez el cabezal se mueve facilitando la producción de las imágenes en dos dimensiones.

Dentro de las impresiones en 3d se implica una tercera dimensión, admitiendo la generación del volumen en los dibujos, con ayuda del software específico se realiza la formación de las capas a partir de un modelo 3d virtual, quiere decir que, al diseñar un objeto en algún programa como el CAD, estos pueden ser impresos a través de unos impresos en 3d.

Existen diferentes tipos de materiales para la realización de una impresión, su costo depende de la calidad del material a utilizarse.

Entre algunos tipos de materia que se usan están los siguientes:

- PLA (ácido politactico)
- ABS (acrilinotrilo butadieno estireno)
- Metal
- Fotopolímeros
- Resina líquida

1.14.1. Tipos de impresión 3D

Existen diversos métodos para la impresión 3D, a continuación, se describen los principales:

- **Impresoras 3d. Estereolitografía:** Implica a utilización de una aza de luz ultravioleta o resina líquida.
- **Impresoras 3d. Sinterización Selectiva Por Láser:** la impresión de los objetos es por medio de la utilización de láser.
- **Impresoras 3d. Por Inyección:** las impresiones se realizan a través de la inyección de fopolímero líquido.
- **Impresoras 3d. Por Deposición De Material Fundido:** para su impresión es necesario insertar polímero fundido dentro de una base completamente plana por capas.

1.15. Raspberry PI3

Al Raspberry pi 3 se lo considera como un computador de placa simple o reducida, su desarrollo fue dado en reino Unido, uno de los objetivos fue alentar que se enseñe este tipo de ciencias como lo es la computación,

Explícitamente no indica si es software libre o con algún tipo de marca, ya que se pueden adquirir y a su vez cualquier persona puede convertirse en un distribuidor de Raspberry, de esta forma se lo considera de libre uso.

En la **Figura 33-1:** se muestra la placa Raspberry Pi3.



Figura 33-1: Raspberry Pi3

Fuente: www.amazon.es/Raspberry-Pi-modelo-sobremesa-Quad-Core/

Esta versión es una de las más potentes y convenientes para los usuarios, manteniendo su costo mínimo de \$ 35 dólares, lo que más sobresale de este modelo es su procesador el ARM CORTEX A53, que comprende cuatro núcleos de 1.2 GHZ de 64 bits, además de su productividad que es diez veces mayor al Raspberry original, y a sus vez un cincuenta por ciento más que Raspberry pi2 del modelo anterior, este modelo se encuentra integrado en su placa de conectividad de wifi y de bluetooth.

CAPITULO II

2. MARCO METODOLÓGICO

El proceso de diseño del brazo robótico del proyecto tuvo tres métodos principales que influyeron en su ejecución:

Los métodos Inductivo, Heurístico y Experimental se distribuyen en tres etapas del trabajo. El primer método es usado en la etapa de recolección de información, el segundo método se aplica para la programación y diseño del robot y el tercer método se utilizó en la etapa de prueba y corrección de fallas y errores en el robot.

Se aplicó el método inductivo para la etapa de recolección de información, ya que este método obtiene ideas generales a partir de premisas particulares. El conjunto de toda la información recogida y la correcta manera de estudiarla, han hecho que se pueda alcanzar un compendio correcto de la ciencia electrónica, de visión artificial y robótica, tras el proyecto.

Se recurrió al método heurístico para realizar la programación en OpenCV, el diseño de la estructura en SolidWorks, disposición de los sensores y actuadores; el método se basa en el uso de reglas y conocimientos empíricos para explorar procedimientos que permitan hallar la solución.

La experiencia al momento de construir el robot se ha dado desde el diseño digital que ha venido dado por un ejemplo anterior sobre el cual se ha tenido que trabajar y en el momento de la construcción se ha debido corregir los errores del primer plan.

Para las pruebas de funcionamiento se ha tenido que separar por partes a la construcción del robot, para que, en caso de haber problemas, se pueda tener una clara idea de los procesos o sitios en donde hubo de darse los errores y para al final, tener mínimo impacto de errores en la consecución del proyecto.

2.1. Software utilizado para la programación y diseño del Robot

Para la realización del brazo robótico se hace necesario diversos softwar, desde el sistema operativo Raspbian Wheezy que sirve de plataforma al software de programación, software de

compilación. El sistema operativo Windows 8.1 servirá de plataforma para el software de diseño del robot. SolidWorks permitió diseñar el brazo robótico.

2.1.1. *Instalación y Configuración de Raspbian Jessie*

El raspbian Jessie es el proporcional al Windows o Ubuntu, o cualquier otro sistema operativo de computadora. Esto separa al “Raspbi” del resto de las placas de desarrollo. La instalación es muy sencilla, a continuación, se describen los pasos:

- Descargar la imagen ISO de la página oficial de la Raspberry.
- Una vez instalado y grabado en la tarjeta de memoria, se procede a encender la placa Raspberry Pi con sus debidas conexiones.
- Se realizan las configuraciones básicas como son fecha y hora, distribución del teclado.
- Finalmente se accede al escritorio para empezar trabajo en la placa Raspberry Pi.

Las demás partes como el entorno de programación Python ya llega en instalado y preconfigurado en el sistema operativo. Los demás complementos se irán revisando conforme avance el proyecto.

2.2. Selección de Motores

En este apartado se explica los parámetros que se tomaron en cuenta para la selección y dimensionamiento de cada elemento empleado en el sistema automatizado; como son las boquillas, elementos de protección y conductores.

Para escoger los motores adecuados en el presente proyecto, y considerando la gran variedad de opciones a nuestra disposición se debió tener en cuenta las siguientes consideraciones:

Fuerza: Debido al material con el que está construido el brazo, este adquiere un peso considerable por lo que los motores requieren tener un gran torque con respecto a su eje y además se necesita la precisión del movimiento del mismo. Por esta razón los motores no deben ser pequeños ni convencionales, se ha visto la necesidad de la adquisición de motores especiales para el trabajo antes explicado.

Peso: El peso de un motor suele ir de la mano con la fuerza que este puede aplicar, y por lo mencionado con anterioridad, el peso de los motores va a ser considerable.

2.2.1. Metodología de LAGRANGE-EULER para el modelo dinámico de un robot

En este apartado se diseñará y analizará la estructura mecánica del robot, se determinará el torque de los motores. Existen dos metodologías que permiten desarrollar el modelo matemático de la dinámica de un robot industrial: La metodología de Newton-Euler y la metodología de LaGrange – Euler.

Para el análisis de la dinámica del robot diseñado se utilizará la metodología de LaGrange-Euler ya que las expresiones resultantes de estas son más compactas con respecto a las encontradas con la metodología Newton-Euler. En la **Figura 1-2:** se muestra la representación de masas y velocidades.

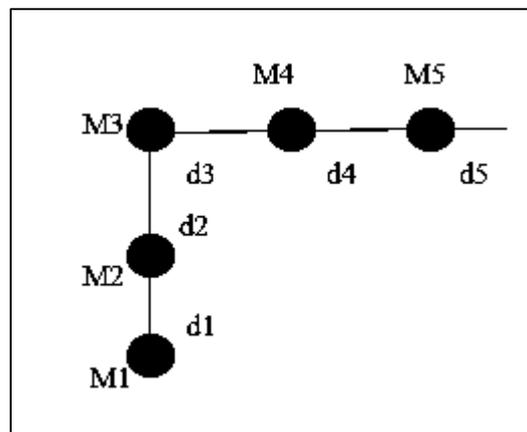


Figura 1-2: Representación de masas y velocidades

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = T$$

$$L=K-P$$

Donde:

L= Representa el Lagrangiano y es la diferencia entre la energía cinética y la energía potencial.

K= Energía cinética del robot.

P= Energía potencial del robot.

\ddot{d} = aceleración de gravedad

q= Variables de la articulación compuestas por ángulos Θ_i ó desplazamientos d_i .

τ = Es el vector con componente T_i de torques de los ángulos de las articulaciones y F_i de los desplazamientos en las articulaciones.

$$\frac{l\partial}{\partial q_1} = (m_1 + m_2)\dot{d}_1$$

$$\frac{d}{dt} \frac{l\partial}{\partial \dot{q}} = (m_1 + m_2)\ddot{d}_1$$

$$\frac{l\partial}{\partial \dot{q}_1} = 0$$

$$\mathbf{F1}=(\mathbf{m1+m2})\ddot{\mathbf{d1}}$$

$$\frac{l\partial}{\partial q_2} = (m_2 + m_3)\dot{d}_2$$

$$\frac{d}{dt} \frac{l\partial}{\partial \dot{q}} = (m_2 + m_3)\ddot{d}_2$$

$$\frac{l\partial}{\partial \dot{q}_2} = 0$$

$$\mathbf{F2}=(\mathbf{m2+m3})\ddot{\mathbf{d2}}$$

$$\frac{l\partial}{\partial q_3} = (m_3)\dot{d}_3$$

$$\frac{d}{dt} \frac{l\partial}{\partial \dot{q}} = (m_3)\ddot{d}_3$$

$$\frac{l\partial}{\partial \dot{q}_3} = 0$$

$$\mathbf{F3}=(\mathbf{m3})\ddot{\mathbf{d3}}$$

$$\frac{l\partial}{\partial q_4} = (m_4)\dot{d}_4$$

$$\frac{d}{dt} \frac{l\partial}{\partial \dot{q}} = (m_4)\ddot{d}_4$$

$$\frac{l\partial}{\partial \dot{q}_4} = 0$$

$$\mathbf{F4}=(\mathbf{m4})\ddot{\mathbf{d4}}$$

$$\frac{l\partial}{\partial q_5} = (m_5)\dot{d}_5$$

$$\frac{d}{dt} \frac{l\partial}{\partial \dot{q}} = (m_5)\ddot{d}_5$$

$$\frac{l\partial}{\partial \dot{q}_5} = 0$$

$$\mathbf{F4}=(\mathbf{m5})\ddot{\mathbf{d5}}$$

Matriz del modelo dinámico del robot

$$\begin{matrix} F1 \\ F2 \\ F3 \\ F4 \\ F5 \end{matrix} = \begin{pmatrix} F1=(m1+m2) \ddot{d}1 & 0 & 0 & 0 & 0 \\ 0 & F2=(m2+m3)\ddot{d}2 & 0 & 0 & 0 \\ 0 & 0 & F3=(m3)\ddot{d}3 & 0 & 0 \\ 0 & 0 & 0 & F4=(m4)\ddot{d}4 & 0 \\ 0 & 0 & 0 & 0 & F4=(m5)\ddot{d}5 \end{pmatrix}$$

Fuerza requerida para movilizar cada eslabón

M1= 2kg	F1=88.2 NM
M2 =5kg	F2=90.16NM
M3= 4.2kg	F3=41.16NM
M4= 0.kg	F4=0.98NM
M5=0.1kg	F5=0.98NM

2.2.2. Controlador de motores

La ejecución del programa de visión artificial se ha desarrollado en la placa Raspberry pi, en cambio se ha visto necesario que para el control del movimiento de los motores se utilice una placa externa; la placa seleccionada ha sido Arduino Uno. En los siguientes apartados se describen los motores empleados para la elaboración del brazo robótico.

2.2.3. Motores empleados en el brazo robótico según análisis dinámico

Para el presente proyecto se emplearon dos tipos de motores: 3 motores DC con caja reductora y 2 servomotores. A continuación, se describen su función en el brazo robótico.

Motor con caja reductora: se eligió el Motor DC de 2.1HP de 24 [V], para el control de las tres primeras articulaciones, debido a que satisfacen con el torque requerido para estos 3 primeros grados de libertad.

En la Tabla 1-2: se describen las características del motor con caja reductora de tipo DC.

Tabla 1-2:Características del motor DC AmpFlow E30-400 con caja reductora

CARACTERÍSTICA	DESCRIPCIÓN
MODELO	E30-400
VOLTAJE [V]	24
RPM	680
POTENCIA	2.1 HP
TORQUE	8.8 KG

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Servomotor: se eligió un servomotor ya que no se necesitaba de mucha precisión sino más bien de un motor ligero y sin mucho peso. Se emplearon dos servomotores para el control de las dos últimas articulaciones del brazo robótico. De toda la variedad de motores que se pueden encontrar en el mercado se ha seleccionado el servomotor de 5kg.

Tabla 2-2:Características del servomotor

CARACTERÍSTICA	DESCRIPCIÓN
MODELO	HITEC
VOLTAJE [V]	4.8 -12VDC
CORRIENTE QUE CONSUME	180mA a 6VDC
POTENCIA	2.1 HP
TORQUE	3.53 kg-cm a 6VDC

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.3. Selección de la cámara

Debido a la gran capacidad del Kinect y sus variadas bondades para con la visión artificial. En el capítulo I se ha detallado las características del Kinect, sin embargo, para justificar mejor la elección del Kinect se exponen los siguientes puntos:

- Mejor resolución de cámara RGB: La cámara RGB que posee el Kinect es mejor sobre su competencia VGA que posee las cámaras convencionales y ocupadas para la Raspberry Pi.
- Se debe tener en cuenta que por tener mejor resolución esto implica menor control sobre las imágenes por segundo que puede acceder a estudiar el programa. Por lo que, a su vez, la gran ventaja del Kinect puede convertirse en una contrariedad.

- Permite trabajar desde la raíz del procesador y así no pasarlo por muchos entornos, pues esto implicaría mucha demora en la impresión de sus resultados.

2.4. Diseño del brazo Zortrax 3D

A continuación, se presentan los diseños digitales de las piezas del brazo robótico Zortrax 3D de 5 grados de libertad. El robot final tiene el mismo diseño sin embargo no se ocupan las dimensiones digitales presentadas a continuación por motivo de la invalidez del presente modelo, el cual se vio en la necesidad de mejorar y que se detallará en el apartado de la construcción y pruebas.

2.4.1. Mesa Repositoria

Un brazo robótico no puede tener como su centro de acción el suelo directo porque no podría tener uno ideal siempre. Entonces se necesita una mesa repositorio que sea parte del brazo todo el tiempo y en cualquier tipo de trabajo.

La **Figura 2-2:** muestra el Diseño de la Mesa Repositorio.

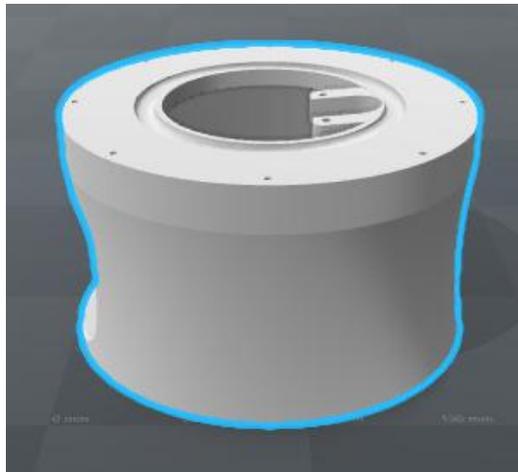


Figura 2-2: Diseño de la Mesa Repositorio

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.2. Base rotatoria

Equipada con un motor NEMA 17, la base giratoria significa el primer grado de libertad y el movimiento total del brazo, por lo que en la práctica debe sobrellevar todo el peso del brazo robótico. El material con el que se ha impreso este y todos los demás componentes es ABS.

En la **Figura 3-2:** se muestra el diseño de la base rotatoria



Figura 3-2: Base rotatoria

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.3. *Articulación giratoria*

Encapsulada en la base giratoria y llevada por el movimiento del motor, esta pieza viene a mostrar el segundo grado de libertad y además su diseño le permite comunicar con la siguiente articulación. En la **Figura 4-2:** muestra el diseño de la articulación giratoria.

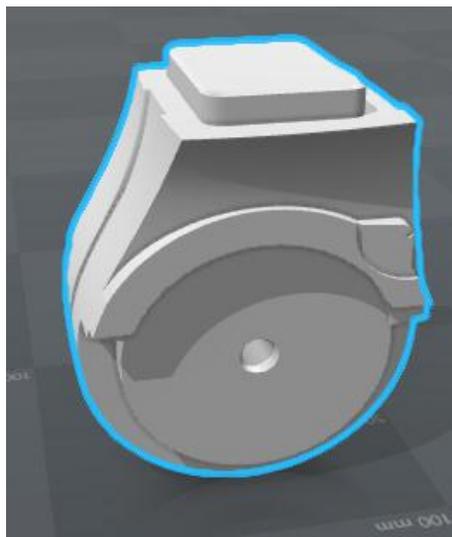


Figura 4-2: Articulación giratoria

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.4. *Pieza contendora de brazo-antebrazo*

A la postre la pieza más importante pues es la conectora del peso base del brazo y antebrazo, es fundamental pues de no tener un motor adecuado no va a poder realizar los movimientos correctos.

En la **Figura 5-2:** se muestra el diseño del conector brazo-antebrazo.

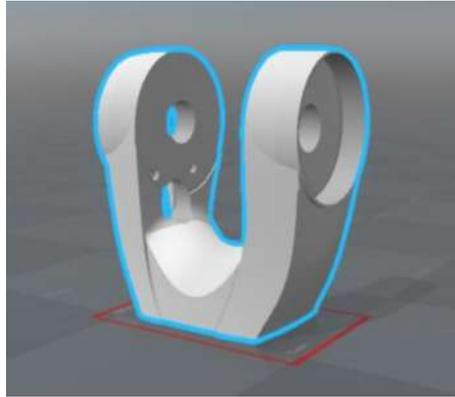


Figura 5-2: Conector brazo-antebrazo

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.5. *Pieza giratoria del antebrazo*

Aunque todos los ejes se mueven basados en los mismos motores los diseños varían un poco debido a la encapsulación física que necesita la pieza.

En la **Figura 6-2:** el diseño de la pieza giratoria del antebrazo.

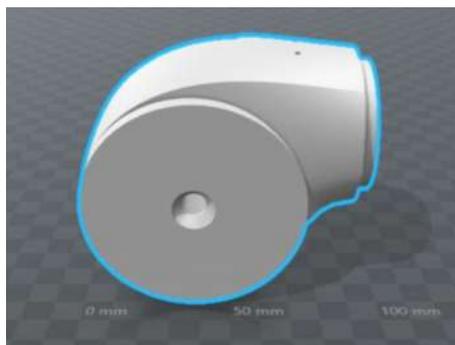


Figura 6-2: Pieza giratoria del antebrazo

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.6. *Antebrazo*

Pieza que sirve de eslabón y también de punto y soporte del eje final y del actuador. Sus dimensiones deben tener la correcta longitud, aun cuando el espacio de trabajo requiera de más distancia de pieza, esta no puede excederse pues quizá la fuerza del motor no abastezca al movimiento.

En la **Figura 7-2**: se muestra el diseño del antebrazo.

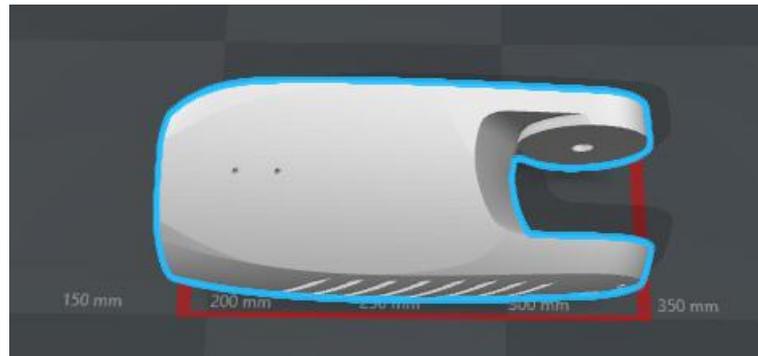


Figura 7-2: Antebrazo

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.7. *Pieza del actuador final*

Es la pieza que contiene al actuador final, es decir el cuarto grado de libertad del brazo robótico. Conviene mencionar que tiene el mismo tamaño y dimensión que la articulación anterior para facilitar el movimiento. Se encarga de generar un movimiento circular que no amplía el espacio de trabajo sino más bien que posiciona de forma correcta al actuador final.

En la Figura 8-2: se observa el diseño de la pieza del actuador final

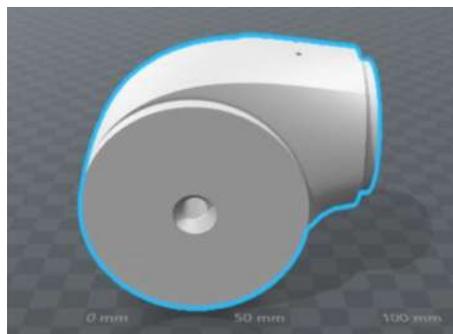


Figura 8-2: Pieza del actuador final

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.4.8. *Actuador final*

El actuador final trata de unas pinzas de ABS que suponen un actuador final del brazo robótico, estos son movidos por servomotores.

En la **Figura 9-2:** se muestra el diseño del actuador final

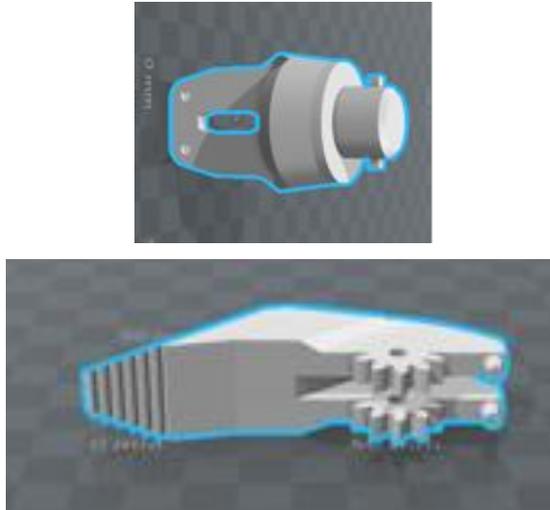


Figura 9-2: Actuador final

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.5. Implementación del Brazo Zortrax 3D

La **Figura 10-2:** se visualiza el Brazo Zortrax 3D implementado, cabe mencionar que este fue el primer diseño para el prototipo de brazo robótico, sin embargo debido limitaciones mecánicas se procedió a la implementación de un segundo brazo de material metálico. En el Capítulo III se describen las limitaciones del brazo Zortrax 3D



Figura 10-2: Brazo Zortrax 3D implementado

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.6. Diseño del brazo metálico

Para el diseño del robot se ha usado como base un clásico diseño de brazo robótico SCADA de 5 grados de libertad. Como en la siguiente imagen. En la **Figura 11-2:** se muestra el brazo robótico SCADA de 5 GDL. Por motivos de la implementación del brazo robótico se han necesitado elementos adicionales a la estructura base.

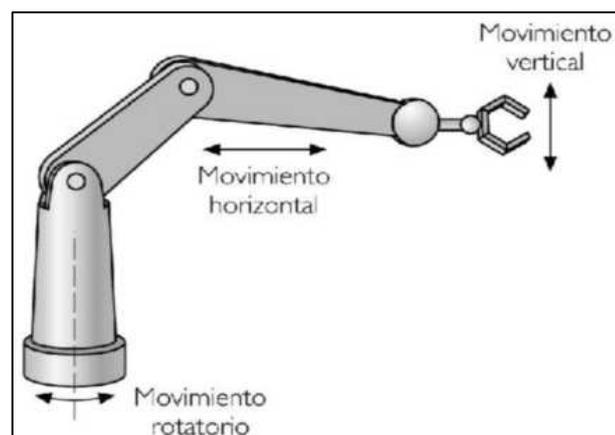


Figura 11-2: Brazo robotico SCADA de 5 GDL

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.6.1. *Diseño de la base y soporte de Nylon*

En la base se ha colocado una mesa metálica con suficiente peso para que puede resistir el movimiento del brazo. Sus dimensiones son: 60 cm x 60 cm x 60 cm.

En la **Figura 12-2**: se muestra el diseño de la base y soporte.

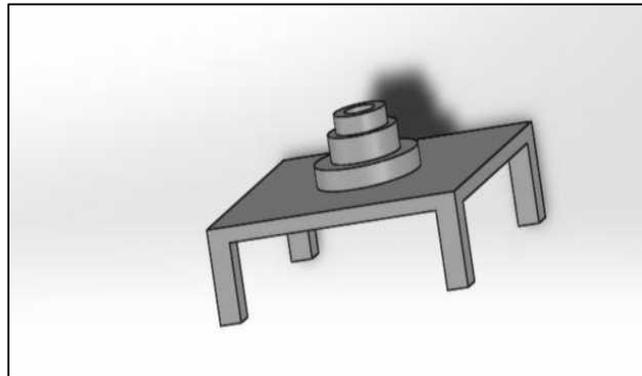


Figura 12-2: Diseño de Base y soporte

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.6.2. *Diseño del primer eslabón*

El primer eslabón cuenta en lo físico con un motor en su base de 2 toneladas, y de las dimensiones que se presentan a continuación en el dibujo. Su material es de una línea de acero que no resulta tan pesada por delgadez, pero sí de gran resistencia.

En la **Figura 13-2**: se muestra el diseño del primer eslabón.

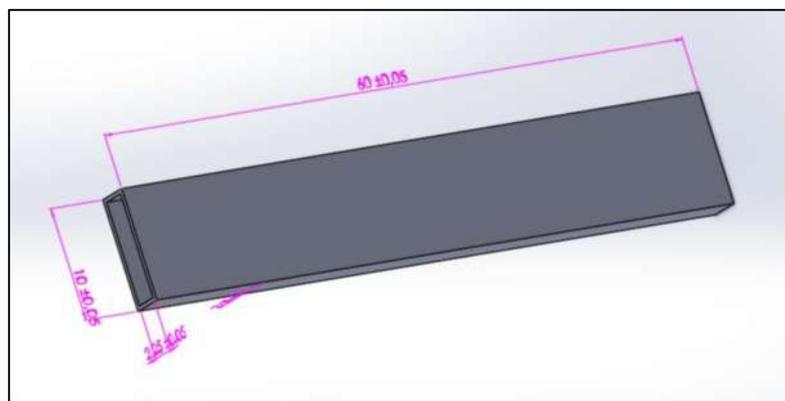


Figura 13-2: Diseño del primer eslabón

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.6.3. *Diseño del segundo eslabón*

De la misma forma el segundo eslabón tiene un gran parecido, ya que la articulación es la misma y el movimiento es circular. Su movimiento lo genera un motor de 1 tonelada.

En la **Figura 14-2:** se visualiza el Diseño segundo eslabón.

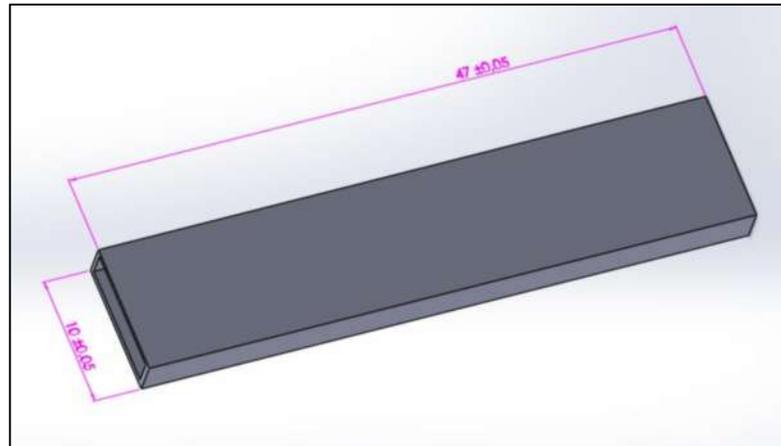


Figura 14-2: Diseño segundo eslabón

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.6.4. *Diseño del adaptador para el motor y el tecla*

El movimiento de los motores al ser demasiado rápidos y fuertes, necesitaba de cajas reductoras que disminuyan esta velocidad y pudieran ser controladas. Además, se han necesitado de encoders para poder posicionar los puntos finales.

La **Figura 15-2:** muestra el diseño del adaptador para el motor y el tecla

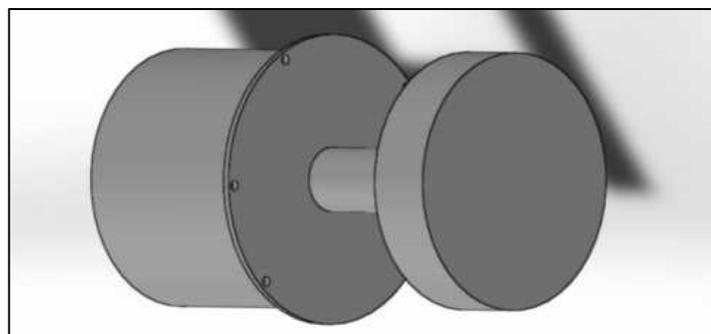


Figura 15-2: Diseño del adaptador para el motor y el tecla

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.7. Método de cinemática inversa Denavit-Hartenberg del brazo metálico

Este es un método usado para el posicionamiento espacial del robot, aplicando las leyes de la cinemática inversa, con los cuales se usa el tipo de movimiento que se quiere imprimir en cada uno de los ejes y según sea el caso llegar a la posición final. En la **Figura 16-2:** se ilustra la metodología de Denavit-Hartenberg

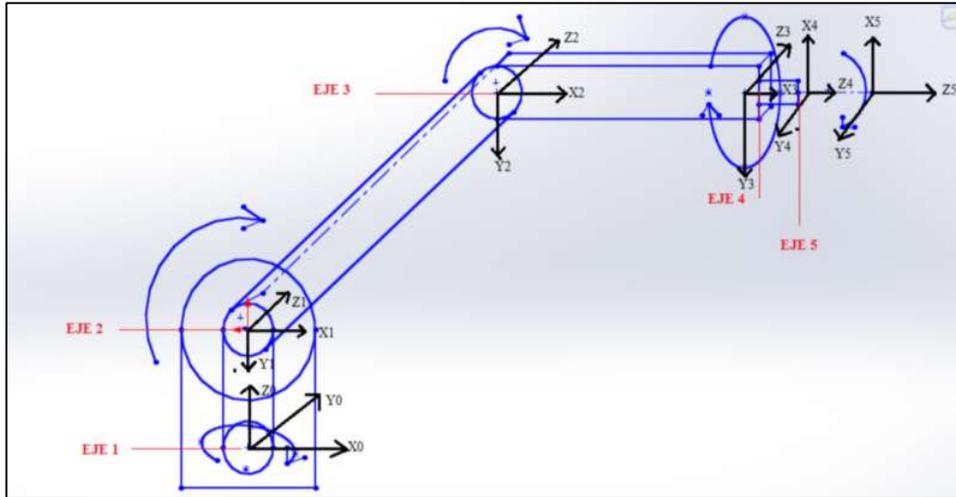


Figura 16-2: Metodología de Denavit-Hartenberg

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Para este brazo robótico de 5GDL se ha desarrollado la siguiente tabla de posicionamiento.

Tabla 3-2: Parámetros D-H

TABLA Denavit-Hartenberg (D-H)				
ELEMENTO	α_i°	q_i	θ_i°	d_i
1	90	0	θ_1	l_1
2	0	l_2	θ_2	0
3	0	l_3	θ_3	0
4	0	0	θ_4	0
5	90	0	θ_5	l_4

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Las matrices correspondientes a cada eslabón son las siguientes:

$$A_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ S_2 & C_2 & 0 & l_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 & l_3 C_3 \\ S_3 & C_3 & 0 & l_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} C_5 & S_5 & 0 & 0 \\ S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.8. Cinemática inversa a través de matrices de transformación homogénea

Mediante el uso de las matrices de transformación homogénea se realiza la cinemática inversa del robot. Utilizando las matrices obtenidas con los parámetros DH y sus inversas se procede a calcular las siguientes ecuaciones.

$$\mathbf{T} = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4$$

$$\frac{\mathbf{T}}{A_1^0} = A_2^1 * A_3^2 * A_4^3 * A_5^4$$

$$(A_1^0)^{-1} \mathbf{T} = A_2^1 * A_3^2 * A_4^3 * A_5^4$$

$$A_1^0 = \mathbf{T} (A_2^1)^{-1} * (A_3^2)^{-1} * (A_4^3)^{-1} * (A_5^4)^{-1}$$

$$\mathbf{T} (A_1^0)^{-1} = A_2^1 * A_3^2 * A_4^3 * A_5^4$$

$$\mathbf{T} = \begin{bmatrix} n_x & O_x & a_x & P_x \\ n_y & O_y & a_y & P_y \\ n_z & O_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & l_1 & 1 \end{bmatrix} = A_2^1 * A_3^2 * A_4^3 * A_5^4$$

ECUACIONES

- 1) $C_1 f_x + S_1 f_y = C_2 L_3 C_3 - C_2 L_3 C_3 + L_2 C_2$
- 2) $P_3 = S_2 L_3 C_3 + C_2 L_3 C_3 + L_2 S_2$
- 3) $S_1 f_x - C_1 f_y = 0$

(3)

$$S_1 p_x = C_1 p_y$$

$$\frac{S_1}{C_1} = \frac{P_Y}{P_X}$$

$$tg_1 = \frac{P_Y}{P_X} \quad q_1 = \arctg\left(\frac{P_Y}{P_X}\right)$$

$$T(A_1^0)^{-1}(A_2^1)^{-1} = A_3^2 * A_4^3 * A_5^4$$

$$(4) (C_1 C_2 - S_1 S_2)(P_x) + (C_1 S_2 + S_1 C_2)(P_y) = L_3 C_3$$

$$P_z = L_3 S_3$$

$$(5) (S_1 C_2 + C_1 S_2)(P_x) + (S_1 S_2 - C_1 C_2)(P_y) = L_4$$

(4)

$$C_1 C_2 P_x - S_1 S_2 P_x + C_1 S_2 P_y + S_1 C_2 P_y = L_3 C_3$$

$$C_2(C_1 P_x + S_1 P_y) - S_2(+S_1 P_x - C_1 P_y) = L_3 C_3 \quad (4.1)$$

$$C_2 A - S_2 B = L_3 C_3$$

(5)

$$S_1 C_2 + C_1 S_2 P_x + S_1 S_2 P_y - C_1 C_2 P_y = L_4$$

$$C_2(S_1 P_x - C_1 P_y) + S_2(C_1 P_x + S_1 P_y) = L_4 \quad (5.1)$$

$$C_2 B + S_2 A = L_4$$

(4.1) + (5.1)

$$A(C_2 + S_2) + B(C_2 - S_2) = L_3 C_3 + L_4$$

Elevamos al cuadrado (4.1) y (5.1)

$$C_2^2 A^2 - 2C_2 A S_2 B + S_2^2 B^2 = L_3^2 C_3^2 \quad (*1)$$

$$S_2^2 A^2 + 2S_2 A C_2 B + C_2^2 B^2 = L_4^2 \quad (*2)$$

(*1) + (*2)

$$A^2(C_2^2 + S_2^2) + B^2(S_2^2 + C_2^2) = L_3^2 C_3^2 + L_4^2$$

$$A^2 + B^2 = L_3^2 C_3^2 + L_4^2$$

$$C_3 = \frac{A^2 + B^2 - L_4}{L_3^2} \quad (*3)$$

$$A^2 = (C_1 P_x + S_1 P_y)^2$$

$$S_1 P_x - P_y C_1 = 0 \quad \text{Ecuación (3)}$$

$$\longrightarrow A^2 = P_x^2 + P_y^2 \quad B^2 = P_x^2 - P_y^2$$

$$C_3 = \frac{P_x^2 + P_y^2 + (P_x^2 - P_y^2) - L_4}{L_3^2}$$

$$q_3 = \arccos \left(\frac{P_x^2 + P_y^2 + (P_x^2 - P_y^2) - L_4}{L_3^2} \right)$$

Se divide (4.1) de (5.1)

$$\frac{1/C_2(C_2 A + S_2 B)}{1/C_2(S_2 A + C_2 B)} = \frac{L_4}{L_3 C_3}$$

$$\frac{A + B \frac{2}{C_3}}{B + A \frac{-S_2}{C_2}} = \frac{L_4}{L_3 C_3}$$

$$L_3 S_3 A + L_3 S_3 B = -T_2 A L_3 C_3 - T_2 A L_2 + B L_3 C_3 + B L_2$$

$$L_3 S_3 A + L_3 S_3 B = -T_2 A L_3 C_3 - T_2 A L_2 + B L_3 C_3 + B L_2$$

$$q_2 = \arctg \frac{P_x - P_y}{P_x + P_y} - \arctg \frac{L_3 C_3}{L_3 + L_3 C_3}$$

Para el cálculo de q4 y q5 se procede a separar el manipulador. Se utiliza las sub-matrices de rotación $A_1^0 * A_2^1 * A_3^2$

$$R_1^0 = \begin{bmatrix} C_1 & 0 & S_1 \\ S_1 & 0 & -C_1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_3^0 = \begin{bmatrix} -C_1 S_{23} & S_1 & C_1 C_{23} \\ -S_1 S_{23} & -C_1 & S_1 S_{23} \\ C_{23} & 0 & C_1 S_{23} \end{bmatrix}$$

$$R_3^2 = \begin{bmatrix} C_4 & 0 & S_4 \\ S_4 & 0 & -C_4 \\ 0 & 1 & 0 \end{bmatrix}$$

$$(R_2^0)^{-1} = \begin{bmatrix} -C_1 S_{23} & -S_1 S_{23} & C_{23} \\ S_1 & -C_1 & 0 \\ C_1 C_{23} & S_1 C_{23} & S_{23} \end{bmatrix}$$

$$R_5^4 = \begin{bmatrix} C_5 & 0 & S_5 \\ S_5 & 0 & -C_5 \\ 0 & 1 & 0 \end{bmatrix}$$

$$R_5^5 = (R_4^3)(R_5^4)$$

$$T(R_3^0)^{-1} = (R_4^3)(R_5^4)$$

$$T(R_3^0)^{-1} = \begin{bmatrix} S_{45} & S_4 & C_4 S_5 \\ S_4 C_5 & C_4 & S_{45} \\ S_5 & 0 & -C_5 \end{bmatrix}$$

ECUACIONES

- 1) $-C_1 S_{23} P_x - S_1 S_{23} P_y + C_{23} P_z = C_4 S_5$
- 2) $S_1 P_x - C_1 P_y = S_4 S_5$
- 3) $C_1 C_{23} P_x + S_1 C_{23} P_y + S_{23} P_z = C_5$

(3)

$$\mathbf{tg}_5 = \frac{P_y}{P_x} \quad \mathbf{q}_5 = \mathbf{arctg} \left(\frac{P_y}{P_x} \right)$$

$$\mathbf{q}_5 = \mathbf{arctg} \frac{\pm \sqrt{\mathbf{1} - C_3^2}}{C_5}$$

(2) dividido (1)

$$\frac{S_1 P_x - C_1 P_y}{-C_1 S_{23} P_x - S_1 S_{23} P_y + C_{23} P_z} = \frac{S_4 S_5}{C_4 S_5}$$

$$\frac{S_4}{C_4} = \frac{S_1 P_x - C_1 P_y}{-C_1 S_{23} P_x - S_1 S_{23} P_y + C_{23} P_z}$$

$$q_4 = \text{arctg} \frac{S_1 P_x - C_1 P_y}{-C_1 S_{23} P_x - S_1 S_{23} P_y + C_{23} P_z}$$

2.9. Método geométrico del brazo metálico

Es un método sencillo y muy aplicado para brazos de pocos grados de libertad y con una sola forma de articulación. Su proceso y funcionalidad se basa en usar la posición final a donde se desea hacer llegar al actuador final. A partir de allí, se ocupan funciones trigonométricas senos y cosenos, con los cuales se obtiene los ángulos de inclinación que necesita cada una de las articulaciones

En la **Figura 17-2:** se muestra la resolución del brazo robótico mediante el método geométrico.

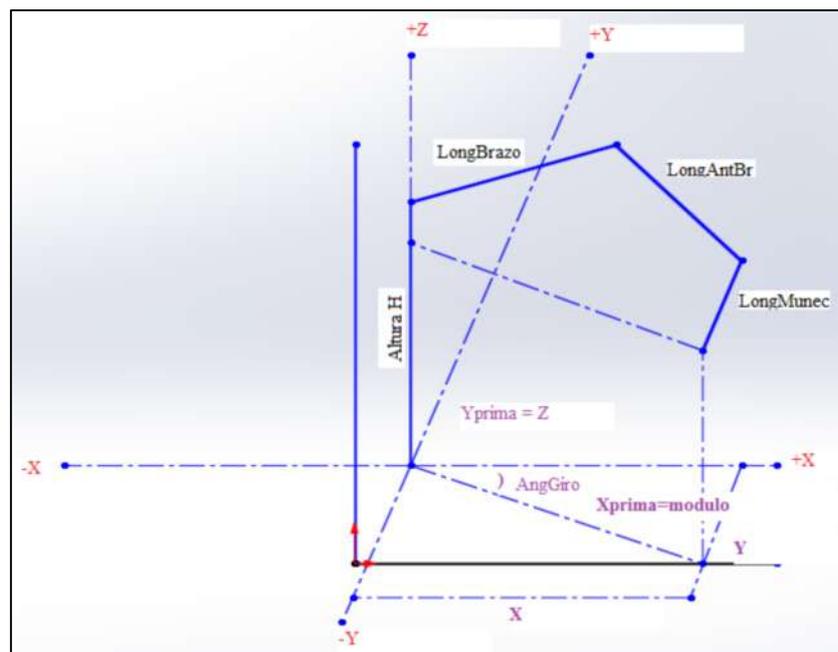


Figura 17-2: Método geométrico

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Como podemos apreciar en la imagen, necesitamos conocer (a parte de la posición final del actuador) la posición de la base o punto de inicio. Además, las dimensiones de la altura de la barra, longitud del brazo, longitud del antebrazo y longitud de la muñeca. Ya que el resultado final de esta resolución depende directamente de estos datos físicos que deben ser exactos.

2.9.1. Resolución del Método Geométrico

A continuación, se describe la resolución del método geométrico para el brazo robótico:

$$Afx = \cos(\text{Cabeceo}) * \text{LongMunec}$$

$$\text{LadoB} = X - Afx$$

$$Afy = \sin(\text{Cabeceo}) * \text{LongMunec}$$

$$\text{LadoA} = Y - Afy$$

$$\text{Hipotenusa} = \sqrt{(\text{LadoA})^2 + (\text{LadoB})^2}$$

$$\text{Alfa} = \tan^{-1}\left(\frac{\text{LadoA}}{\text{LadoB}}\right)$$

$$\text{Beta} = \cos^{-1}\frac{(\text{LongBrazo}^2 - \text{LongAntBr}^2 + \text{Hipotenusa}^2)}{2 * \text{LongBrazo} * \text{Hipotenusa}}$$

$$\text{AngBrazo} = \text{Alfa} + \text{Beta}$$

$$\text{Gamma} = \cos^{-1}\frac{(\text{LongBrazo}^2 + \text{LongAntBr}^2 - \text{Hipotenusa}^2)}{2 * \text{LongBrazo} * \text{LongAntBr}}$$

$$\text{AngAntBr} = -(180 - \text{Gamma})$$

$$\text{AngMunec} = \text{Cabeceo} - \text{AngBrazo} - \text{AngAntBr}$$

2.10. Diagrama del algoritmo de umbralización

En la **Figura 18-2:** Se puede observar el diagrama del algoritmo de umbralización empleado para realizar el código de programación del sistema de visión artificial que guiara al brazo robótico.



Figura 18-2: Algoritmo de umbralización

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Se presenta el diagrama de búsqueda de una imagen a través del algoritmo de umbralización que inicia con la adquisición de la imagen para luego convertirla en un espacio HSV también llamado HSB. Dicho modelo muestra la matiz, saturación y valor de la imagen a ser tratada, y a la vez define un modelo de color en términos de sus componentes, para luego realizar su umbralización y finalmente combinar la información existente para conseguir nuevos resultados con procesos morfológicos.

2.11. Delimitación del área de trabajo según la aplicación planteada.

El área de trabajo está limitada a 80cm por 80cm es decir un plano cuadrado en el espacio, distancia que está dentro de los rangos de movimientos posibles del prototipo, la distancia máxima es de 1.10cm desde el eje ubicado en la base rotatoria hasta el efector final. La aplicación simulada es la de un robot soldador (suelda eléctrica punto a punto), el cual tendrá que realizar su trabajo en las juntas de las barras a ser soldadas, llegando a las extremidades del cuadrado localizado.

En la **Figura 19-2:** se muestra el rango de movimiento del brazo

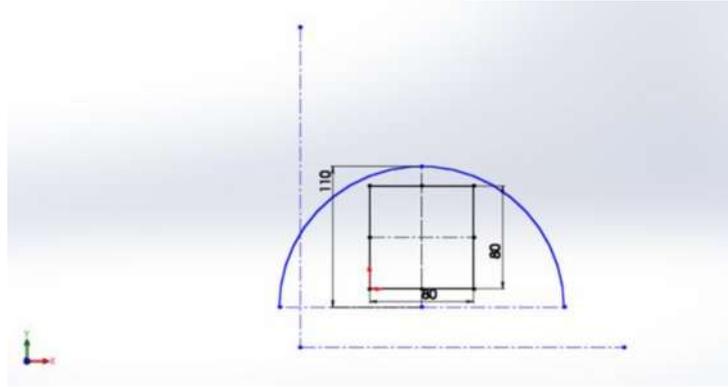


Figura 19-2: Rango de movimiento del brazo

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.12. Diseño del diagrama electrónico

La placa electrónica utilizada en el diseño está simulada en ISIS Proteus, consta de un controlador Arduino nano, que es el encargado de conectar vía serial el sistema mecánico con el sistema de visión artificial, además controla las funciones de los finales de carrera, los controladores de los motores, y los sensores(encoders).

En la **Figura 20-2:** se muestra el diagrama de conexión de los elementos conectados al Arduino Nano en ISIS Proteus. La placa Arduino nano es la encargada de controlar los servomotores que se encuentran en las dos últimas articulaciones, y también los motores con caja reductora, sin embargo, estos últimos tienen adicionados un controlador.

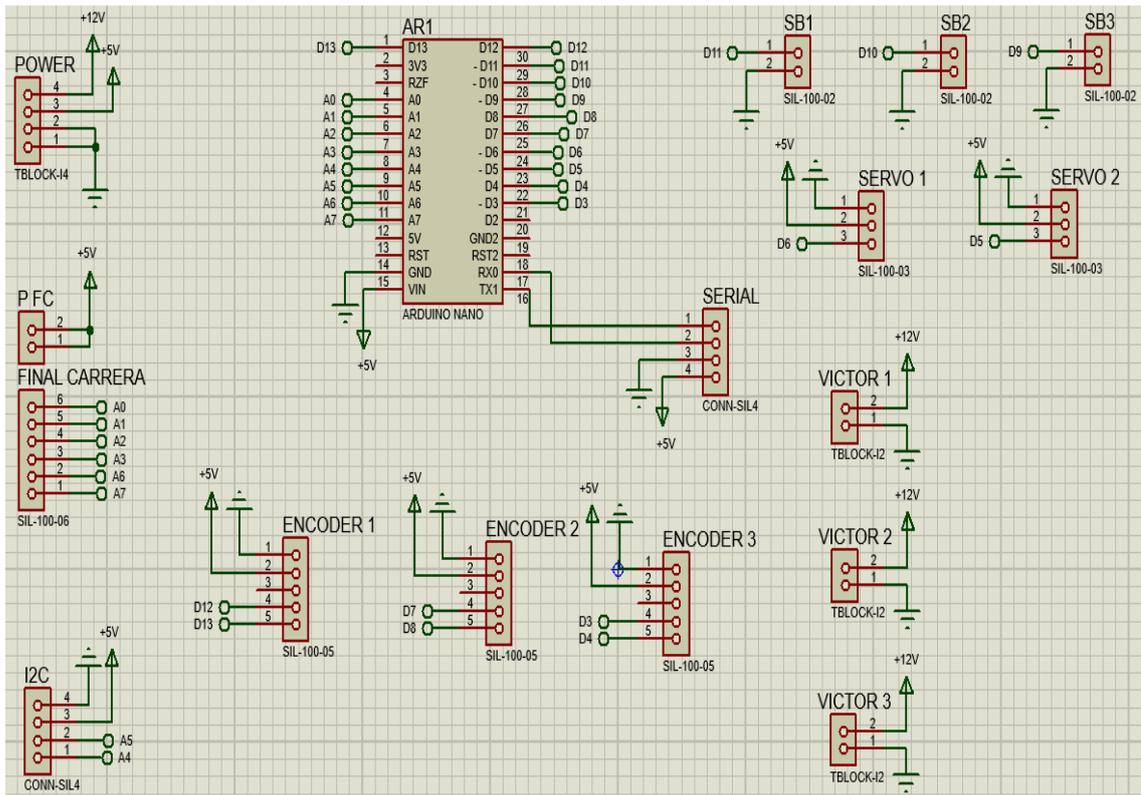


Figura 20-2: Diagrama de conexión en ISIS Proteus

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

En la **Figura 21-2:** se visualiza el esquema de conexión en Fritzing de todos los dispositivos empleados en el brazo robótico.

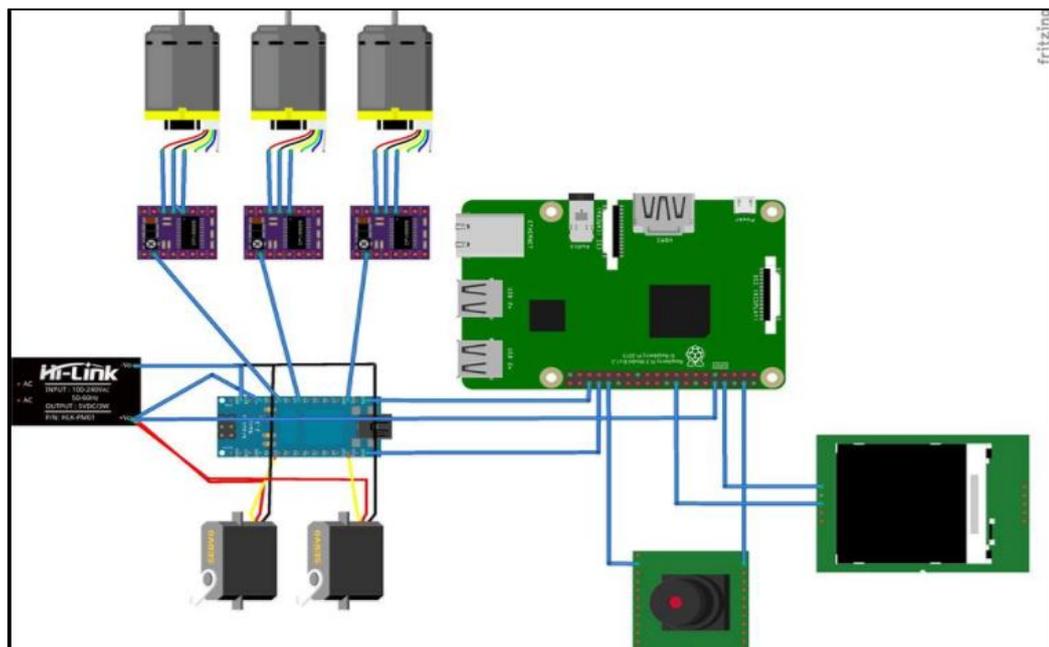


Figura 21-2: Esquema de conexión en Fritzing

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.13. Diagrama de bloques del brazo robótico

En la **Figura 22-2**: se muestra el diagrama de bloques de los principales componentes del diseño. En primer lugar se encuentra el sensor en este caso kinect que obtiene los datos del entorno que está conectado al procesador el cual codifica los datos recibidos para entregarlos al controlador secundario que a su vez da señal a los cuatro actuadores.

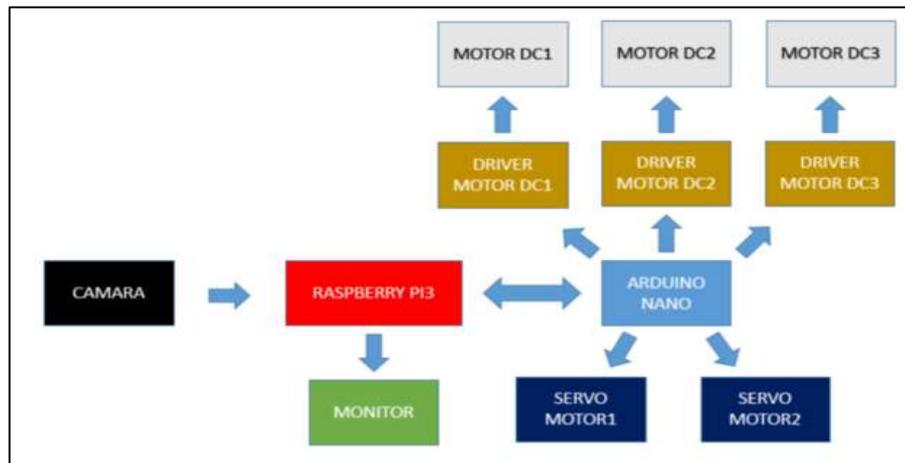


Figura 22-2: Diagrama de bloques del brazo robótico

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

2.14. Implementación del brazo robótico metálico

Previo a la implementación del brazo robótico se describen las características como dimensiones y peso de cada uno de sus componentes.

Dimensiones y peso:

- Mesa: 60 x 60 x 60
- Eslabón cero 20cm
- Primer eslabón 60 x 2.5cm x 4cm
- Segundo 40cm x 2.5cm x 4cm
- Tercer eslabón 7cm x 2.5cm x 4cm
- Cuarto eslabón 7cm
- Peso total del brazo robótico 75 [lb].

Descripción de elementos utilizados:

- Motor de la primera articulación con un torque de 8.15 [kgm].
- Motor de la segunda articulación con un torque de 8.15 [kgm].

- Motor de la base del brazo con un torque de 8.15 [km].
- Primera articulación con una caja reductora de 30:1.
- Segunda articulación con una caja reductora de 20:1.
- Alcance máximo 120 [cm].

En la **Figura 23-2:** se visualiza el esquema del brazo robótico implementado.

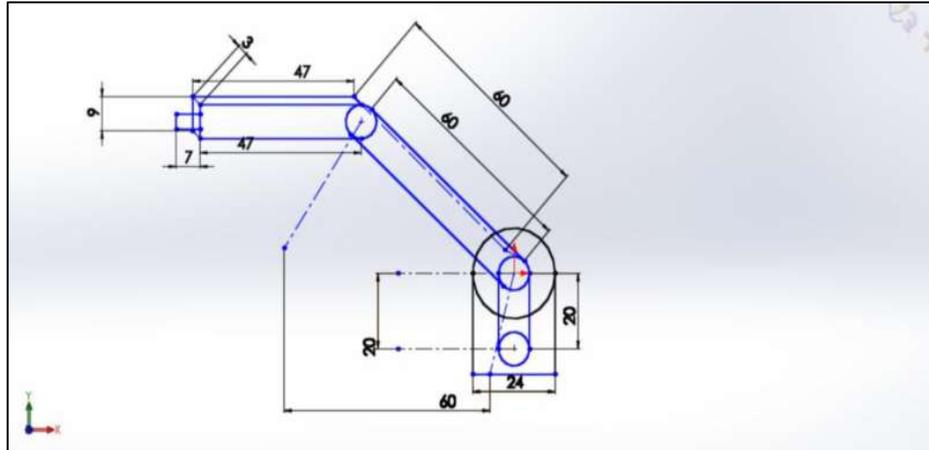


Figura 23-2: Dimensiones del brazo robótico

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Para la construcción del segundo prototipo se ha optado por materiales metálicos, para que este pueda tener una resistencia de nivel industrial, por este motivo también, se ha usado nuevos motores. En la **Figura 24-2:** se muestra que los motores tienen capacidades totalmente distintas. Se han ajustado los motores con reductora con relación 20:1 y 30:1 en la segunda y tercera articulación respectivamente.



Figura 24-2: Motor acoplado al tele

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Los 3 primeros motores al ser DC cuentan con sensores (encoders) ubicados en sus ejes posibilitando ubicar al efector final en una posición requerida por el control de la visión artificial. Para limitar los movimientos de todas las articulaciones se precisó también colocar finales de carrera y con ello limitar el espacio de trabajo, y brindar protección a la estructura y al operador.

En la **Figura 25-2:** se pueden ver los sensores que limitan este trabajo y permite que el brazo no se exceda en sus movimientos.



Figura 25-2: Disposición de sensores

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Una parte esencial del diseño fueron los controladores de los motores. Estos se obligaban pues los mismos al requerir cambios bruscos y altos de corriente podían generar, contracorriente y demás situaciones que pueden afectar el funcionamiento del sistema. En la **Figura 26-2:** se muestra la conexión física de los controladores a los motores con caja reductora.

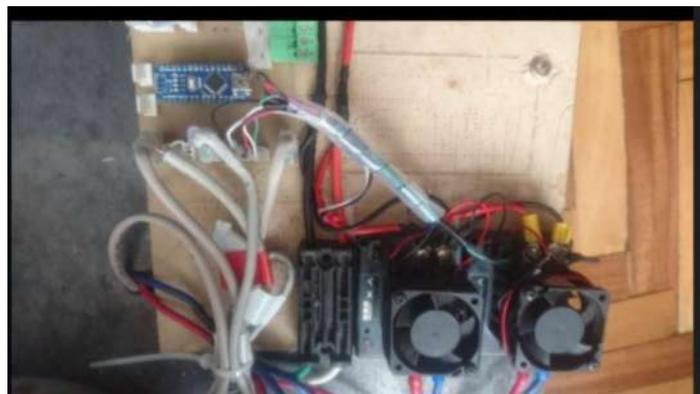


Figura 26-2: Motores con caja reductora y controladores

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

En la **Figura 27-2:** se puede ver el robot ya terminado de ensamblar. Y se puede ubicar y detallar la posición de los controladores que se hallan implementados en la base. También se puede apreciar en la misma base el Arduino nano, el cual sirve de transmisión de datos serial mente.

En el actuador final se pueden apreciar los servomotores y la totalidad del brazo robótico, que tiene la misma forma que el primer brazo pero con un material y actuadores distintos.



Figura 27-2: Brazo robótico metálico

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

CAPITULO III

3. MARCO DE RESULTADOS, ANÁLISIS Y DISCUSIÓN

En este apartado se muestran las pruebas finales, resultados y conclusiones sobre el proceso de construcción, implementación y prueba del brazo robótico orientado por la mira en tres dimensiones del Kinect.

En los siguientes subtemas se presentan las realidades obtenidas con el avance del proyecto, así como las ideas fallidas que tuvieron que ser totalmente hechas a un lado para dar paso a unas más funcionales.

3.1. Pruebas al primer brazo Zortrax 3D

El primer brazo robótico implementado se definió por un prototipo de ZORTRAX. Una vez concluido el ensamblaje se procedió a probar el movimiento del robot, el cual no tuvo el funcionamiento esperado. Por tal razón se ha visto necesario exponer algunos de los problemas más evidentes sobre el brazo. Posteriormente se realizarán comparaciones con otro brazo robótico.

3.1.1. *Problemas físicos y mecánicos*

El primer gran problema que se tuvo con este brazo es que las piezas al encajar en sus lugares y al momento de tratar de dar movilidad no tiene el control que el manual promete.

- El peso del brazo es demasiado mayor a lo que puede soportar el motor, por lo que al momento de movilizar el brazo este se descontrola. A parte de esto la prueba solo se realiza con su articulación, sin el brazo completo, por lo que no se puede esperar que funcione mejor con aun más desequilibrio de peso.
- El resorte añadido al diseño original busca amortiguar algo del movimiento tan desordenado que tiene. En la **Figura 1-3:** se refleja otro problema, los motores no encajan completamente en los sitios que el diseño promete, por lo que en primer lugar no se tendrá la tracción

adecuada para el movimiento de los ejes y tampoco tendrá la distribución de peso ideal para un brazo que presenta muchas debilidades.



Figura 1-3: Motores del brazo Zortrax

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- La longitud del brazo ha sido demasiado extensa como para que el motor resista el movimiento y menos aún que lo controle por ángulos. Por tal motivo, como se nota en la imagen, el brazo ha sido mutilado y modificado ciertas piezas del diseño original para su posterior impresión y no tenga tanta longitud y por ende demasiado peso. Sin embargo, aun cuando las modificaciones tratan de solventar estos errores. El brazo no funciona como se espera.



Figura 2-3: Brazo del robot Zortrax

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- Los problemas encontrados tienen que ver con el error en las dimensiones del brazo con respecto al peso y la fuerza que pueden producir los motores NEMA 17. A este peso, que de por sí imposibilita el movimiento armónico del brazo, se le tiene que añadir el de la estructura de plástico que afecta directamente los resultados finales. Cabe recalcar que es en los actuadores finales en donde el peso abate de forma más desproporcionada.

3.2. Pruebas realizadas en cada articulación

En los siguientes apartados, se presentan la comparación de pruebas que se hicieron en los movimientos de cada uno de los ejes entre el brazo robótico Zortrax y el brazo robótico metálico. Para las pruebas se realizan 10 intentos de dos ángulos, el primero de 45 grados y el segundo de 90 grados, en el caso de las articulaciones rotacionales se toman también una medida de 180 grados, entregando resultado que están basados en cuanto a precisión y tiempo

3.2.1. Eje base- primera articulación

En el caso de la primera articulación del primer prototipo se toma una medida de 180 grados dado que este no tenía restricción de movimiento, como sucede con el segundo prototipo.

En la **Tabla 1-3:** se muestra los valores obtenidos mediante las pruebas realizadas al brazo robótico Zortrax con movimiento de 180°. Como se puede observar tiene movimiento bastantes regulares, en los intentos 5 y 6, no se ha movido el brazo.

Tabla 1-3: Movimiento a 180° en la primera articulación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX– PRIMERA ARTICULACIÓN 180°		
INTENTO	GRADOS	TIEMPO(SEG)
1	30	1,67
2	30	1,6
3	35	1,5
4	40	1,70
5	0	0
6	0	0
7	35	1,5
8	45	1,5
9	30	1,6
10	35	1,5

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 2-3:** se muestra los valores obtenidos mediante las pruebas realizadas al brazo robótico metálico con movimiento de 45°

Tabla 2-3: Movimiento a 45° en la primera articulación del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO – PRIMERA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	35	1,01
2	10	0,98
3	35	1,2
4	40	1,38
5	40	1,6
6	40	1,55
7	40	1,6
8	40	1,5
9	40	1,47
10	35	1,45

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 3-3:** se muestra los valores obtenidos mediante las pruebas realizadas al brazo robótico Zortrax con movimiento de 90°

Tabla 3-3: Movimiento a 90° en la primera articulación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – PRIMERA ARTICULACIÓN 90°		
INTENTO	GRADOS	TIEMPO(SEG)
1	85	0,80
2	95	0,80
3	75	0,80
4	85	0,80
5	70	0,80
6	80	0,35
7	85	0,5
8	80	0,70
9	80	0,90
10	75	1,1

- En la **Tabla 4-3:** se muestra los valores obtenidos mediante las pruebas realizadas al brazo robótico metálico con movimiento de 90°

Tabla 4-3: Movimiento a 90° en la primera articulación del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO - PRIMERA ARTICULACIÓN 90°		
INTENTO	INTENTO	INTENTO
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.3. Pruebas en la segunda articulación

A continuación, se detallan las pruebas realizadas en los dos prototipos realizados, con ángulos de 45° y 90°.

- En la **Tabla 5-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la segunda articulación del brazo robótico Zortrax con movimiento de 90°

Tabla 5-3: Movimiento a 45° en la segunda articulación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – SEGUNDA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 6-3:**Tabla 4-3: se muestra los valores obtenidos mediante las pruebas realizadas en la segunda articulación del brazo robótico metálico con movimiento de 45°

Tabla 6-3: Movimiento a 45° en la segunda articulación del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO -SEGUNDA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	50	2,35
2	55	2,40
3	-20	2,20
4	55	2,35
5	40	2,30
6	45	2,35
7	45	2,40
8	40	2,35
9	45	2,40
10	50	2,30

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 7-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la segunda articulación del brazo robótico Zortrax con movimiento de 90°

Tabla 7-3: Movimiento a 90° en la segunda articulación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – SEGUNDA ARTICULACIÓN 90°		
INTENTO	GRADOS	TIEMPO(SEG)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 8-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la segunda articulación del brazo robótico metálico con movimiento de 90°

Tabla 8-3: Movimiento a 90° del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO -SEGUNDA ARTICULACIÓN 90°		
INTENTO	GRADOS	TIEMPO(SEG)
1	80	3,85
2	90	3,90
3	80	3,10
4	80	3,80
5	80	3,95
6	80	3,75
7	85	3,80
8	85	3,85
9	90	3,80
10	70	3,80

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.4. Prueba en la tercera articulación

En este apartado se detallan las pruebas realizadas en la tercera articulación de los dos protoipos implementados, con ángulos de 45° y 90°.

- En la **Tabla 9-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la tercera articulación del brazo robótico Zortrax con movimiento de 45°

Tabla 9-3: Movimiento a 45° del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – SEGUNDA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 10-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la tercera articulación del brazo robótico metálico con movimiento de 45°

Tabla 10-3: Movimiento a 45° del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO -SEGUNDA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	30	1,55
2	35	1,55
3	30	1,55
4	40	1,50
5	40	1,55
6	35	1,55
7	45	1,50
8	55	1,55
9	35	1,40
10	35	1,40

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 11-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la tercera articulación del brazo robótico Zortrax con movimiento de 90°. Esta articulación tampoco se ha movido.

Tabla 11-3: Movimiento a 90° del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – SEGUNDA ARTICULACIÓN 90°		
INTENTO	GRADOS	TIEMPO(SEG)
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 12-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la tercera articulación del brazo robótico metálico con movimiento de 90°

Tabla 12-3: Movimiento a 90° del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO -SEGUNDA ARTICULACIÓN 90°		
INTENTO	GRADOS	TIEMPO(SEG)
1	75	2,70
2	75	2,80
3	80	2,80
4	90	2,80
5	80	2,80
6	75	2,80
7	75	2,80
8	75	2,85
9	70	2,80
10	80	2,80

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.5. Pruebas en la cuarta articulación

En el presente apartado se detallan las pruebas realizadas en la cuarta articulación de ambos prototipos, con ángulos 45° y 180°.

- En la **Tabla 11-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la cuarta articulación del brazo robótico Zortrax con movimiento de 45°.

Tabla 13-3: Movimiento a 45° del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – CUARTA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	35	0,9
2	40	0,87
3	35	0,88
4	35	0,76
5	35	0,70
6	35	0,80
7	35	0,80
8	55	0,80
9	50	0,85
10	50	0,85

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 14-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la cuarta articulación del brazo robótico metálico con movimiento de 45°

Tabla 14-3: Movimiento a 45° del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO - CUARTA ARTICULACIÓN 45°		
INTENTO	GRADOS	TIEMPO(SEG)
1	55	0,45
2	45	0,45
3	40	0,45
4	40	0,40
5	35	0,45
6	40	0,45
7	40	0,30
8	40	0,45
9	40	0,40
10	40	0,50

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 15-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la cuarta articulación del brazo robótico Zortrax con movimiento de 180°.

Tabla 15-3: Movimiento a 180° del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – CUARTA ARTICULACIÓN 180°		
INTENTO	GRADOS	TIEMPO(SEG)
1	180	0,75
2	180	0,75
3	180	0,75
4	180	0,65
5	180	0,65
6	180	0,70
7	180	0,70
8	180	0,75
9	180	0,70
10	180	0,75

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 16-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la cuarta articulación del brazo robótico metálico con movimiento de 180°

Tabla 16-3: Movimiento a 180° del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO - CUARTA ARTICULACIÓN 180°		
INTENTO	GRADOS	TIEMPO(SEG)
1	180	0,70
2	180	0,80
3	180	0,80
4	180	0,80
5	180	0,80
6	180	0,80
7	180	0,80
8	180	0,85
9	180	0,80
10	180	0,80

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.6. Prueba en la quinta articulación

Para esta última prueba se realiza varios intentos para probar la fidelidad en el control del último eje, dado que es el que nos proporcionará la posición real requerida.

- En la **Tabla 17-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la quinta articulación del brazo robótico Zortrax con movimiento

Tabla 17-3: Movimiento de la quinta articulación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX – QUINTA ARTICULACIÓN	
INTENTO	MOVIMIENTO
1	SI
2	SI
3	SI
4	SI
5	SI
6	SI
7	SI
8	SI
9	SI
10	SI

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- En la **Tabla 18-3:** se muestra los valores obtenidos mediante las pruebas realizadas en la quinta articulación del brazo robótico metálico con movimiento

Tabla 18-3: Movimiento de la quinta articulación del brazo robótico metálico

BRAZO ROBÓTICO METÁLICO - QUINTA ARTICULACIÓN	
INTENTO	MOVIMIENTO
1	SI
2	SI
3	SI
4	SI
5	SI
6	SI
7	SI
8	SI
9	SI
10	SI

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.7. Conclusiones de resultados de funcionamiento

En la **Tabla 19-3:** se muestra las comparaciones de los resultados obtenidos en la prueba realizada tanto al primer prototipo denominado brazo robótico Zortrax y el segundo prototipo brazo robótico metálico.

Tabla 19-3: Comparación de resultados del brazo robótico Zortrax y brazo robótico metálico

PRIMER PROTOTIPO: BRAZO ROBÓTICO ZORTRAX	SEGUNDO PROTOTIPO: BRAZO ROBÓTICO MECÁNICO
Carece de la funcionalidad total necesaria para el trabajo completo.	Tiene una funcionalidad completa y una fuerza de tipo industrial.
Solo en la quinta articulación los dos brazos se pueden comparar en la funcionalidad correcta.	Uno y otro poseen buena movilidad en la última articulación proporcionando los movimientos esperados.
No puede presentar ninguna seguridad de funcionabilidad completa por su falta de movilidad en sus articulaciones.	Se puede usar este modelo para la continuación de proyectos por su capacidad completa en el movimiento de sus 5 articulaciones.
En peso neto es más liviano, pero a pesar de eso no tiene un estudio adecuado en los motores para mover sus articulaciones.	Su peso es extremadamente mayor al primer brazo, pero con un buen análisis de motores se permite la movilidad de todas sus articulaciones.

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

3.8. Análisis del mecanismo en Msc Adams

A continuación, en la **Figura 3-3:** se visualiza la simulación del brazo robótico en Msc Adams; en se tiene la simulación del prototipo con las dimensiones, peso y material real para su análisis.

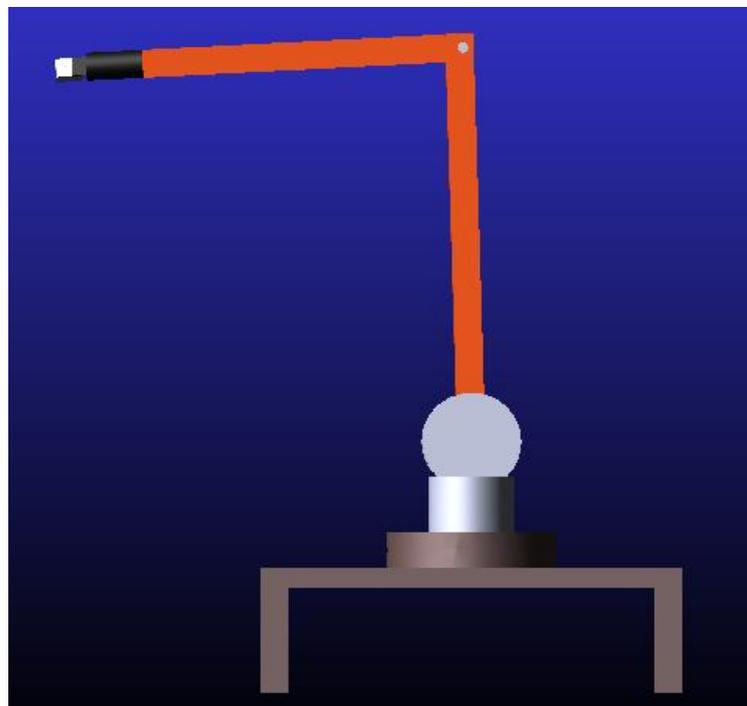


Figura 3-3: Simulación del brazo robótico en Msc Adams

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Los resultados del mecanismo con un movimiento general del modelo, con los siguientes grados en cada articulación, se detallan a continuación:

- **1er Eje Base rotatoria**

STEP (time, 0, 0, 8,180d)

- **2do Eje**

STEP (time,1,0,4,80d)

- **3er Eje**

STEP (time, 0, 0, 2,45d)

- **4to Eje**

STEP (time, 0, 0, 2,180d) + STEP(time,6,0,8,180d)

- **5to Eje**

STEP (time,1,0,4,20d)

Graficas de las medias más sobresalientes, tomadas al tiempo de 10 seg.

- **Torque del primer eje:** En este movimiento se tiene el pico más alto de torque a los 2.3 seg, con 640 N.mm, además parte desde el reposo con un torque de 100N.mm, para luego hacerse cero en el instante 8 segundos. La **Gráfico 1-3:** visualiza el torque del primer eje

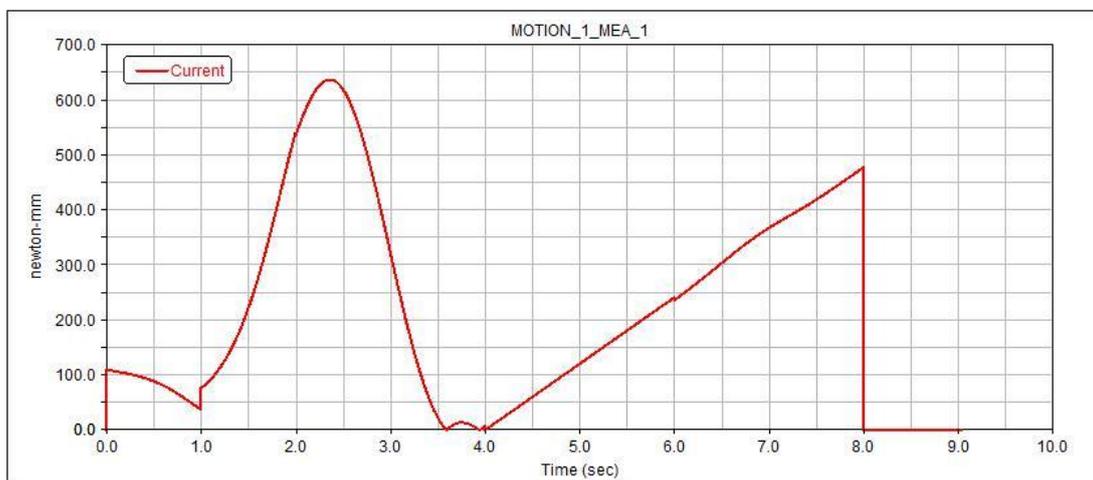


Gráfico 1-3: Torque del primer eje

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Torque Segundo eje:** en la **Gráfico 2-3:** Se puede apreciar un torque negativo de 690 N.mm a los 2.4 seg debido al movimiento en sentido de las agujas del reloj del eje.

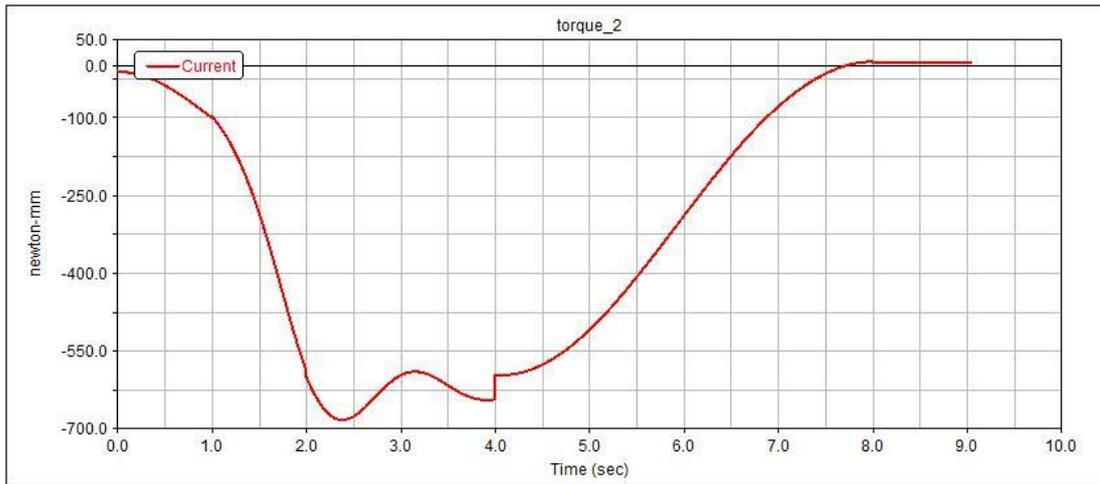


Gráfico 2-3: Torque del segundo eje

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Torque tercer eje:** en la **Gráfico 3-3:** se muestra el torque en el tercer eje.

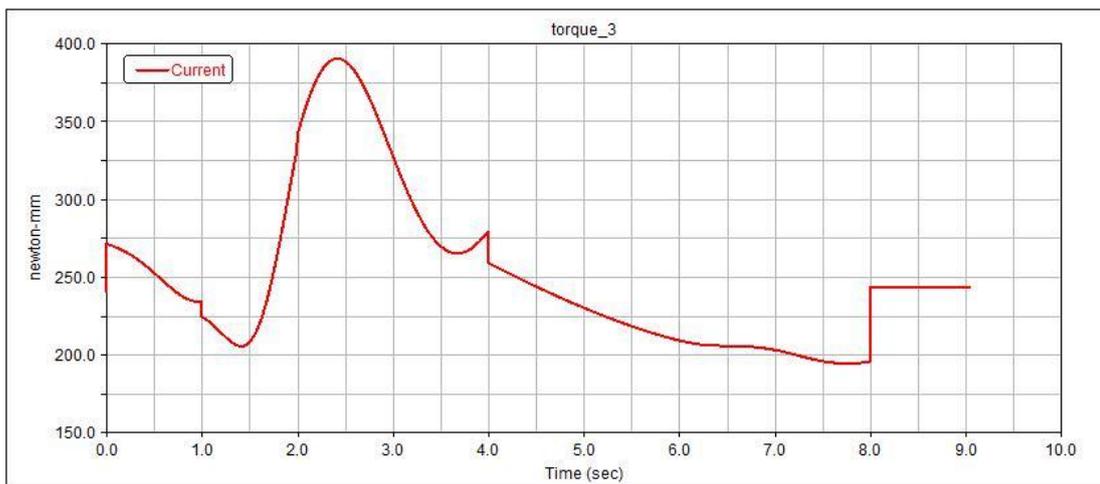


Gráfico 3-3: Torque del tercer eje

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Velocidad angular del tercer eslabón:** En la **Gráfico 4-3:** se tiene una medida de velocidad de rotación que interpreta el ángulo girado por una unidad de tiempo aquí se observa que desde el estado de reposo la curva asciende rápidamente hasta el primer segundo para luego declinar e incrementarse nuevamente a los 3 segundos.

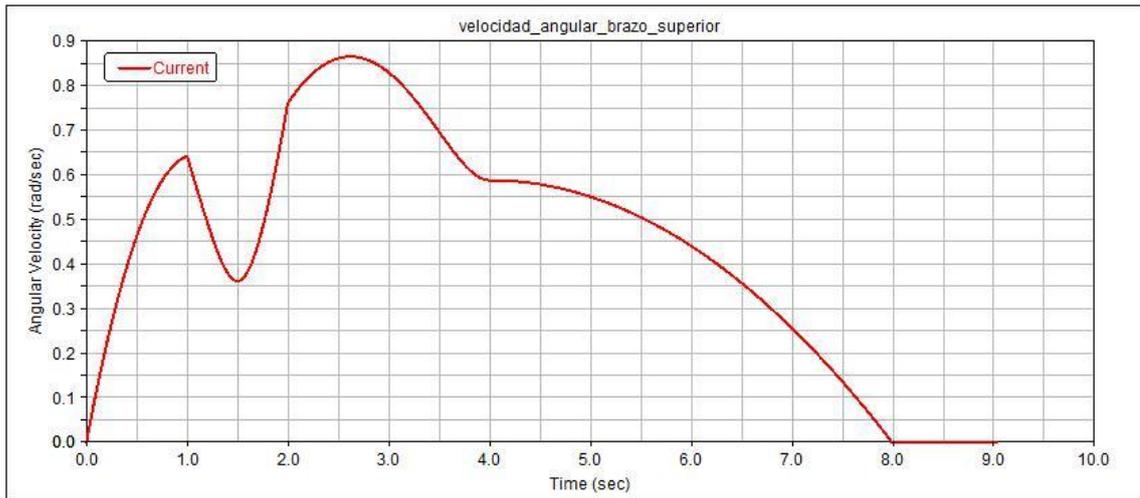


Gráfico 4-3: Velocidad del tercer eslabón

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Velocidad angular del segundo eslabón:** La **Gráfico 5-3:** es muy distinta a la anterior por la rápida estabilización de la curva, esta llega al pico máximo a las 2,4 seg, y a partir del instante 4 seg esta se hace cero debido a que el movimiento en el primer eslabón se ha detenido. En los dos casos se tiene la velocidad angular en rad/seg.

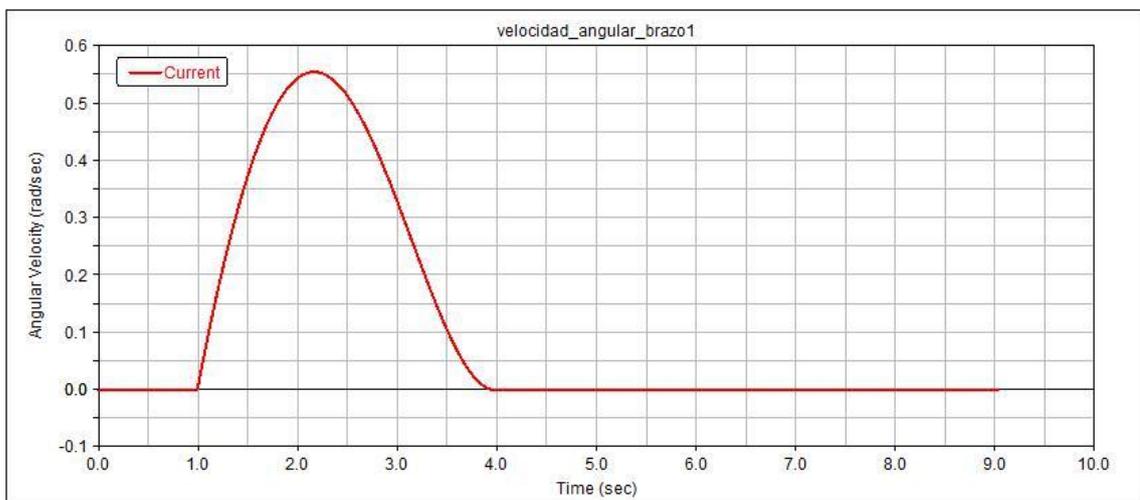


Gráfico 5-3: Velocidad angular del segundo eslabón

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Posición del efector final:** la **Gráfico 6-3:** muestra la posición de la partícula en el círculo al que está limitado el movimiento del eslabón final en un instante de tiempo, esta encuentra su pico máximo a los 2.5seg, se puede apreciar que se estabiliza rápidamente dado al movimiento estable que este último eslabón produce, este dato es muy importante dado que se necesita de la precisión del efector final.

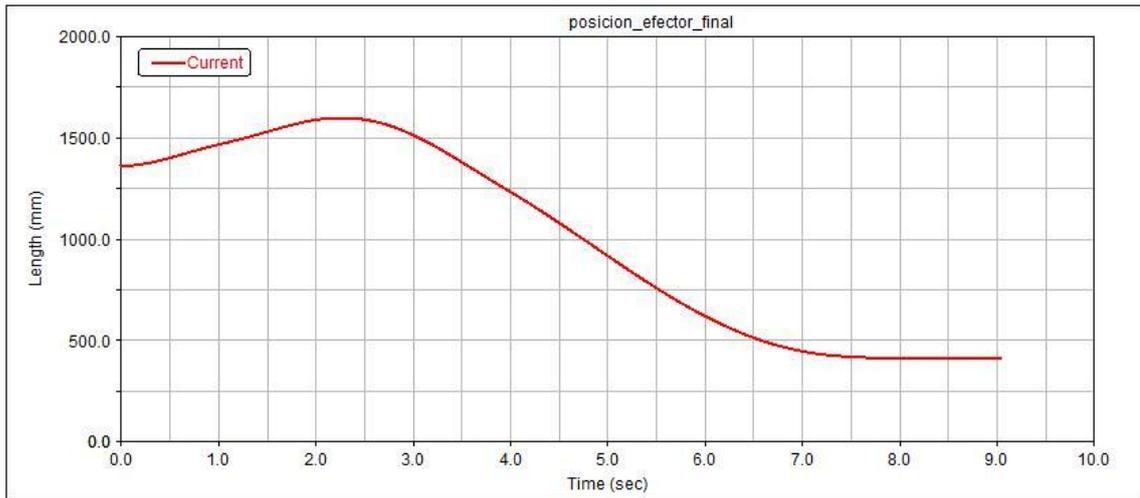


Gráfico 6-3: Posición del efector final

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

- **Posición del tercer eslabón:** En la **Gráfico 7-3:** se observa que la curva del tercer eslabón y el efector final son muy parecidas, encuentran su pico máximo a los 2.5seg y se estabilizan a los 7seg, se ha tomado las medidas en estos dos puntos del prototipo debido a la longitud del tercer eslabón y su cercanía con el último.

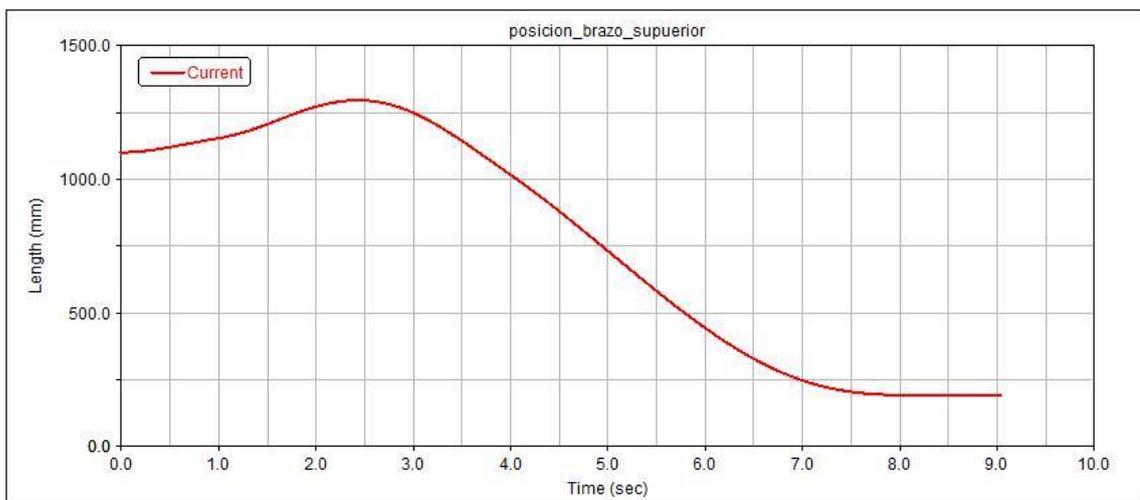


Gráfico 7-3: Posición del tercer eslabón

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Apreciación de los resultados.

- Rápidamente se puede decir observando las gráficas que los picos más altos y bruscos de torque se generan en el primer eje dado por N.mm y en los tiempos que el mecanismo así lo requiere.

- Se ha tomado la velocidad angular del segundo y tercer eslabón por ser los que más movimiento en este tipo despliegan dándonos en el segundo una constante a los 4seg.
- La posición del efector final y tercer eslabón se muestran constantes al final.

3.9. Análisis económico

En las **Tabla 20-3:** y **Tabla 21-3:** se resume los gastos reales del proyecto del brazo robotico Zortrax y del brazo robótico metálico respectivamente.

Tabla 20-3: Costo de implementación del brazo robótico Zortrax

BRAZO ROBÓTICO ZORTRAX	
OBJETO	COSTO (\$)
Impresiones 3d	600
Utilería	100
Motores nema 17	160
Controladores	32
Servomotores	45
Arduino uno	30
TOTAL	967

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Tabla 21-3: Costo de implementación del brazo robótico metálico

BRAZO ROBÓTICO METALICO	
OBJETO	COSTO (\$)
3 controladores	900
Arduino nano	10
Raspberry pi	120
Final de carrera	60
3 motores y caja	1800
Caja de engranes	90
Encoders rotativos	20
2 servomotores	30
1 pinza	20
Material	500
Base de nylon	100
Utilería	100
Transporte	100
Impresiones	50
TOTAL	3900

Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

Dando una totalidad del proyecto de: \$ **4867**

CONCLUSIONES

- La implementación del brazo fue posible y completa, logrando culminar exitosamente con el objetivo esperado.
- Se ha demostrado la capacidad de la Raspberry pi 3 aplicada en la visión artificial, al movilizar un brazo robótico en dirección de un objeto a ser detectado mediante software.
- Se demuestra que para la ejecución de un brazo robótico (soldador) en tiempo real se necesita la correcta armonía entre fuerza y peso de la estructura del robot.
- Se concluye que una inadecuada iluminación en los sistemas de visión artificial altera los resultados en la precisión.
- Los brazos robóticos que cuentan con 5 grados de libertad tienen la capacidad de llegar a todos los rincones dentro de su espacio de trabajo.
- Se determinó la importancia de tomar medidas de seguridad al momento de maniobrar el brazo. (riegos mecánicos, eléctricos y físicos).
- La dificultad del brazo robótico radica en la construcción y utilidad de los componentes mecánicos dándole un grado de complejidad alto.
- Se concluye que en el brazo ZORTRAX no se realizó una cinemática y dinámica apropiada para la utilidad del proyecto.
- La recolección de datos por parte de la cámara Kinect ha sido totalmente eficaz permitiendo ubicar el objeto en los tres ejes (X, Y, Z).

RECOMENDACIONES

- Al momento de la construcción trabajar con cálculos reales (peso, altura, rotación, inclinación, torque) ya que todo influye en el resultado final.
- Se recomienda usar motores paso a paso o servomotores en lugar de motores DC por su manejo de precisión.
- Es necesario hacer un análisis dinámico para dimensionar la capacidad de los motores y demás componentes.
- Para mayor eficiencia se podría implementar la alimentación independiente para cada motor.
- Es conveniente realizar un correcto estudio del presupuesto mecánico y eléctrico para que el sistema no se vea limitado al momento de su implementación.
- Se recomienda analizar profundamente investigaciones pasadas ya que no siempre proveen los datos reales.
- La luz en el área de trabajo para Kinect debe ser la adecuada y el cuerpo poseer óptimas propiedades de forma y/o color.
- La alimentación del dispositivo raspberry pi 3 debe ser el adecuado (datos de placa), en caso de no hacerlo puede presentar varios problemas de funcionamiento.

BIBLIOGRAFÍA

ARBONA, A., Análisis de Imágenes de profundidad para aplicaciones de realidad aumentada mediante el uso de la Kinect., Valencia – España., Universidad Politécnica de Valencia, 2013., pp. 1 -17

ALBA, A., Ejercicios con OpenCV, Facultad de Ciencias, UASLP, Septiembre 2011., pp. 1-5. España, Escuela Universitaria de Ingeniería Técnica Industrial de Madrid., Técnicas de Segmentación de Imágenes., pp. 5.6 – 5.9

ANDUJAR C., Kinect, Agosto., 2015pp. 9-16

CAYAMBE YAMBAY, Diana Isabel; & CASTAÑEDA CHATE, Juan Andrés. Inteligencia artificial aplicada a robot asistencial para la interacción con niños en edad temprana [En línea], **(TESIS DE PREGRADO)**. Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales Riobamba- Ecuador. 2015. pp. 14-30. [Consulta: 05 de septiembre2017]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/6036>

DEPARTAMENTO DE ELECTRÓNICA. *Sistema de locomoción de robots móviles* [web]. Sevilla-España. Automatización y Robótica Industrial, 2010 pp. 1-20.

DIAZ CELIS C. Y ROMERO MOLANO C., Navegación de Robot móvil usando Kinect, OpenCV y Arduino., Meta – Colombia., Universidad de Llanos, 2012

GRUPO DE INVESTIGACIÓN EDMANS. Técnicas y algoritmos básicos de visión artificial, Universidad de la Rioja, Servicio de Publicaciones., 2006. pp. 1-91.

ILBAY LLANGARI, Luis Guido. Evaluación de algoritmos de tracking 3d para la simulación de un brazo robótico, mediante Kinect [En línea], **(TESIS DE PREGRADO)**. Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales Riobamba- Ecuador. 2015. pp. 24-127. [Consulta: 05 de enero 2016].

Disponible en:<http://dspace.esPOCH.edu.ec/bitstream/123456789/4402/1/108T0124.docx.pdf>

MACHUCA MENDOZA, Cristian Rodrigo; & LÓPEZ AYALA, Luis Fernando. Locomoción de un robot cuadrúpedo basada en redes neuronales artificiales (**TESIS**). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales Riobamba- Ecuador. 2015. pp. 20-106.

[Consulta: 05 de enero 2016].

Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/3803/1/108T0113.pdf>

Ollero, A. Robótica, manipuladores y robots móviles. Barcelona, España: Marcombo, 2001

POLOLU ROBOTICS & ELECTRONICS. A-Star 32U4 Micro. [Pdf].

[Consulta: 20 de noviembre 2015]. Disponible en:

<https://www.pololu.com/product/3101/resources>.

RAMIREZ A., Reporte de Búsqueda, Detección y Conteo de Objetos., Guanajuato – México, Centro de Investigación en Matemáticas A.C

RASPBERRY PI. Introducción a la Raspberry [Pdf].

[Consulta: 20 de agosto 2017]. Disponible en:

<https://www.raspberrypi.org/>

Romero, A. Estructura de un robot industrial. (s.f.). [Consulta: 20 de septiembre 2017]. Disponible en:

http://www.juntadeandalucia.es/averroes/iesalfonso_romero_barcojo/departamentos/tecnologia/unidades_didacticas/ud_controlroboticav1/morfologia%20de%20un%20robot.pdf

SALCEDO TOVAR, Marco Luis. Minicomputador educacional de bajo costo Raspberry Pi. *Revista Ethos Venezolana* [web]. Vol. 7 No. 1, Caracas- Venezuela, 2015, pp. 28 – 45 ISSN 1856-9862 ~ Depósito legal pp 200902ZU3258 [Consulta: 20 de noviembre 2015]. Disponible en: http://www.uniojeda.edu.ve/files/upload/ethos/volumen-7-n1/Revista-Ethos-Venezolana_Vol.-7-No.-1.pdf

SERVOCITY. HS-311 STANDARD. [Pdf].

[Consulta: 25 de septiembre 2017]. Disponible en:

https://www.servocity.com/html/hs-311_standard.html#.ViQOwyu-njA.

VIGUERAS, Ángel Soriano. Diseño y Programación de Algoritmos para Robot Móviles. Aplicación al robot LEGO-NXT (**TESIS**). Universidad Politécnica de Valencia, Ingeniería Informática. Valencia-España, 2015, pp. 7-92.

[Consulta: 15 de enero 2016].

Disponible en: <https://riunet.upv.es/bitstream/handle/10251/11167/Memoria.pdf>

VÁZQUEZ Martha. *Niños tímidos, seis consejos para que superen su timidez* [blog]. Vasco-España, 13 de septiembre de 2013. [Consulta: 15 de agosto 2015]. Disponible en: <http://www.consumer.es/>

VIGUERAS, Ángel Soriano. Diseño y Programación de Algoritmos para Robot Móviles. Aplicación al robot LEGO-NXT (**TESIS**). Universidad Politécnica de Valencia, Ingeniería Informática. Valencia-España, 2015, pp. 7-92. [Consulta: 15 de enero 2016]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/11167/Memoria.pdf>

VILLEGAS, Jaime. *Que es un detector de movimiento* [blog]. Medellín-Colombia: TecnoSeguro, 02 de febrero del 2012. [Consulta: 20 de enero 2016]. Disponible en: <http://www.tecnoseguro.com/faqs/alarma/que-es-un-detector-de-movimiento-pasivo-o-pir.html>

YAMANOOR Sai; & YAMANOOR Srihari. Raspberry Pi Mechatronics Projects HOTSH T [web]. Birmingham: Packt Publishing, 2015, pp. 10-47. [Consulta: 15 de noviembre 2015]. Disponible en: <http://PacktLib.PacktPub.com>

ANEXOS

ANEXO A. BRAZO ROBÓTICO METÁLICO IMPLEMENTADO



Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

ANEXO B. KINECT



Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

ANEXO C. IMPLEMENTACIÓN DE DISPOSITIVOS EN EL BRAZO ROBÓTICO



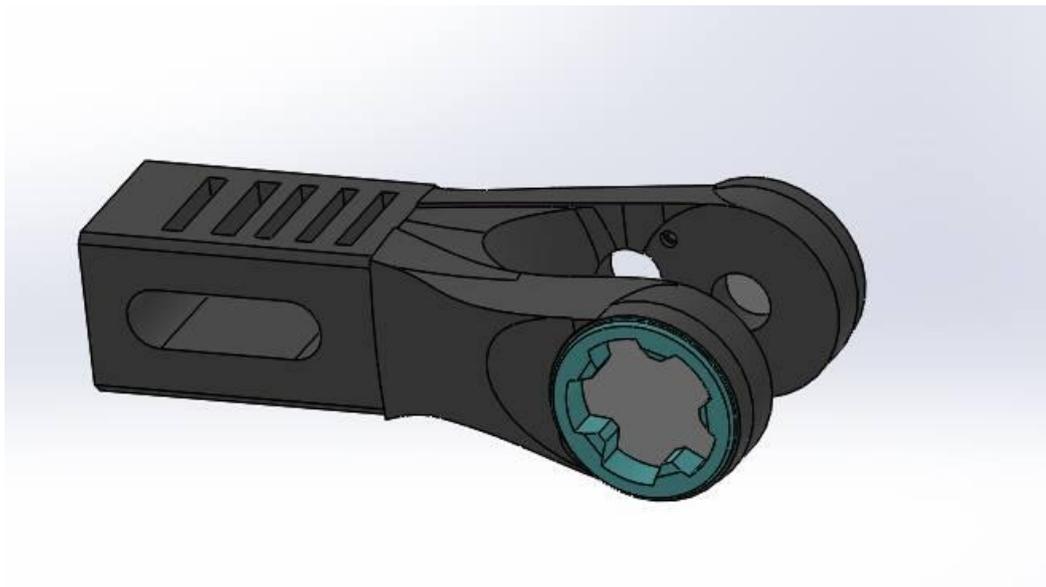
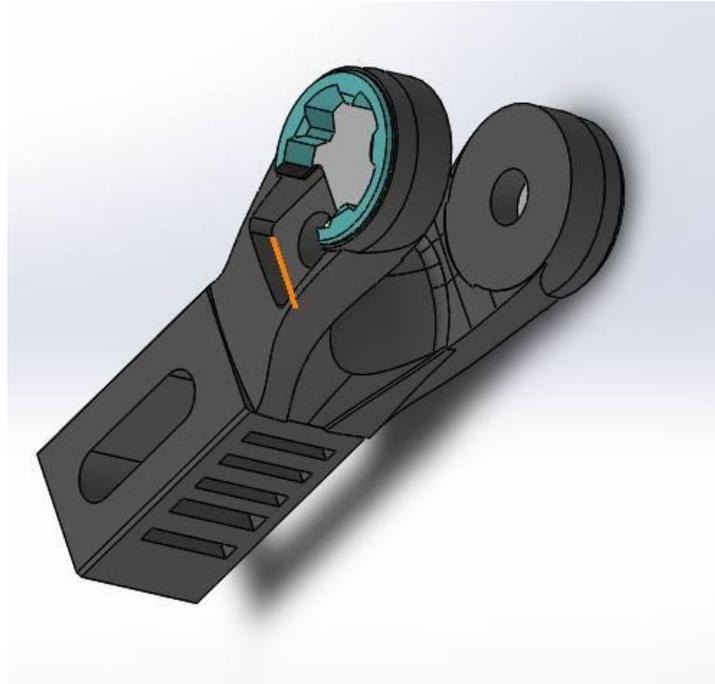
Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

ANEXO D. BRAZO ROBÓTICO ZORTRAX



Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

ANEXO E. PARTES MODIFICADAS DEL BRAZO ROBÓTICO ZORTRAX



Realizado por: MINDA, Israel; RICAURTE, Alex, 2017

ANEXO F. ESLABONES Y ACOPLER IMPRESOS



ANEXO G. ALIMENTACION DE LA ESTRUCTURA



ANEXO H. SOFTWARE EN ARDUINO

```
#include <Servo.h>
Servo M5;
Servo M4;
Servo M3;
Servo M2;
Servo M1;
int serialvalue = 0;
int pinA = 12; // Connected to CLK on KY-040 ----12-----3
int pinB = 2; // Connected to DT on KY-040 -----2-----4
int encoderPosCount = 0;
int pinALast;
int aVal;
boolean bCW;
int led=13;
int pinA2 = 7; // Connected to CLK on KY-040 ----7-----3
int pinB2 = 8; // Connected to DT on KY-040 -----8-----4
int encoderPosCount2 = 0;
int pinALast2;
int aVal2;
boolean bCW2;
int pinA3 = 3; // Connected to CLK on KY-040 ----7-----3
int pinB3 = 4; // Connected to DT on KY-040 -----8-----4
int encoderPosCount3 = 0;
int pinALast3;
int aVal3;
boolean bCW3;
String DATO="";
String SID="";
String SGRADO="";
int ID,GRADO;
int GD4;
int GD5;
int GD3;
int GD2;
int GD1;
int var=0;
int FN1;
int FN2;
int FN3;
int FN4;
int FN5;
int FN6;
int BOT;
void setup() {
  Serial.begin (9600);
  M5.attach(5);
  M4.attach(6);
  M3.attach(11);
  M2.attach(10);
  M1.attach(9);
  pinMode (pinA,INPUT);
  pinMode (pinB,INPUT);
```

```

pinMode(A0, INPUT_PULLUP);
pinMode(A1, INPUT_PULLUP);
pinMode(A2, INPUT_PULLUP);
pinMode(A3, INPUT_PULLUP);
  pinMode (led, OUTPUT);
Serial.println("Grado 90");
M5.write(90);
M4.write(90);
M3.write(90);
M2.write(90);
M1.write(90);
/*
  delay(4000);
  Serial.println("Grado 30");
  M3.write(30);
  delay(2000);
  delay(2000);
  Serial.println("Grado 90");
  M3.write(90);
  Read Pin A
  Whatever state it's in will reflect the last position
*/
pinALast = digitalRead(pinA);
}
void loop()
{
finalcarrera();
if(BOT>=50)
{
  botonera();
}
  while (Serial.available())
  {
    DATO=Serial.readString().substring(0,4);
    SID=DATO.substring(0,2);
    SGRADO=DATO.substring(2,4);
    ID = SID.toInt(); // Asigno "serialvalue" a
Serial.parseInt
    GRADO = SGRADO.toInt(); // Asigno "serialvalue" a
Serial.parseInt
// Me imprime por el puerto serie la variable
serialvalue
    if(ID>=0)
    {
      Serial.print("ID: ");
      Serial.print(ID);
      Serial.print(" GRADO: ");
      Serial.println(GRADO);
      if(ID==55)
      {
        GD3=GRADO;
        encoder3p();
      }
      if(ID==66)
      {
        GD3=GRADO;

```



```

Serial.print("Encoder Position: ");
Serial.println(encoderPosCount);
}
        pinALast = aVal;
    }
}
M3.write(90);
Serial.println("Grado 90");
//encoderPosCount=0;
}
void encoder2p()
{
    Serial.println("Here 1");
    if(GD2<=90)
    {
        Serial.println("Here 2");
        while( encoderPosCount2<=GD2 && FN4<=500 ){
            finalcarrera();
            Serial.println("Grado 70");
            M2.write(150);
            /* if(encoderPosCount<=0)
            {
                encoderPosCount=encoderPosCount*-1;
            }
            */
            Serial.print("    Encoder Position2: ");
            Serial.println(encoderPosCount2);
            aVal2 = digitalRead(pinA2);
            if (aVal2 != pinALast2)
            {
we need to determine direction
if (digitalRead(pinB2) != aVal2) {
encoderPosCount2++;
bCW2 = true;
}
else { // Otherwise B changed first and we're moving CCW
bCW2 = false;
encoderPosCount2++;
}
        Serial.print ("Rotated: ");
if (bCW2){
Serial.println ("clockwise");
}
else{
Serial.println("counterclockwise");
}
Serial.print("Encoder Position: ");
Serial.println(encoderPosCount2);
}
        pinALast2 = aVal2;
    }
}
Serial.println("Grado 100");
M2.write(90);
}
void encoder2n()

```

```

{
    if(GD2<=90)
    {
        while( encoderPosCount2>=GD2 && FN3==LOW){
            finalcarrera();
            Serial.println("Grado 70");
            M2.write(70);
            /* if(encoderPosCount<=0)
            {
                encoderPosCount=encoderPosCount*-1;
            }
            */
            Serial.print("    Encoder Position2: ");
            Serial.println(encoderPosCount2);
            aVal2 = digitalRead(pinA2);
            if (aVal2 != pinALast2)
            { // Means the knob is
rotating
// if the knob is rotating,
we need to determine direction
// We do that by reading
pin B.
if (digitalRead(pinB2) != aVal2)
{ // Means pin A Changed first - We're Rotating Clockwise
encoderPosCount2 --;
bCW2 = true;
}
else {// Otherwise B changed first and we're moving CCW
bCW2 = false;
encoderPosCount2--;
}
Serial.print ("Rotated: ");
if (bCW2){
Serial.println ("clockwise");
}
else{
Serial.println("counterclockwise");
}
Serial.print("Encoder Position: ");
Serial.println(encoderPosCount2);
}
pinALast2 = aVal2;
}
}
Serial.println("Grado 100");
M2.write(90);
}
void encoder1p()
{
    Serial.println("Here 1");
    if(GD1<=90)
    {
        Serial.println("Here 2");
        while( encoderPosCount3<=GD1 && FN2<=LOW ){
            // Serial.println("Here 3");
            finalcarrera();

```

```

Serial.println("Grado 150");
M1.write(130);

/* if(encoderPosCount<=0)
{
encoderPosCount=encoderPosCount*-1;
}
*/
Serial.print("Encoder Position3: ");
Serial.println(encoderPosCount3);
aVal3 = digitalRead(pinA3);
if (aVal3 != pinALast3)
{ // Means the knob is
rotating
we need to determine direction
pin B.
if (digitalRead(pinB3) != aVal3)
{ // Means pin A Changed first - We're Rotating Clockwise
encoderPosCount3++;
bCW3 = true;
}
else { // Otherwise B changed first and we're moving CCW
bCW3 = false;
encoderPosCount3++;
}
Serial.print ("Rotated: ");
if (bCW3){
Serial.println ("clockwise");
}else{
Serial.println("counterclockwise");
}

Serial.print("Encoder Position3: ");
Serial.println(encoderPosCount3);
}
pinALast3 = aVal3;
}
}
Serial.println("Grado 90");
M1.write(90);
}
void encoder1n()
{
if(GD1<=90)
{
while( encoderPosCount3>=GD1 && FN1==LOW){
finalcarrera();
Serial.println("Grado 70");
M1.write(70);
if(encoderPosCount<=0)
{
encoderPosCount=encoderPosCount*-1;
}
Serial.print("Encoder Position3: ");
Serial.println(encoderPosCount3);

aVal3 = digitalRead(pinA3);

```

```

                                if (aVal3 != pinALast3)
                                {
if (digitalRead(pinB3) != aVal3)
{ // Means pin A Changed first - We're Rotating Clockwise
encoderPosCount3 --;
bCW3 = true;
}
else {// Otherwise B changed first and we're moving CCW
bCW3 = false;
encoderPosCount3--;
}
Serial.print ("Rotated: ");
if (bCW3){
Serial.println ("clockwise");
}
else{
Serial.println("counterclockwise");
}
else{
Serial.print("Encoder Position: ");
Serial.println(encoderPosCount3);
}
                                pinALast3 = aVal3;
                                }
                                }
Serial.println("Grado 100");
                                M1.write(90);
}
void finalcarrera(){
//A0 == Grado1
//A1 == Grado1
//A2 == Grado2
//A6 == Grado2
//A3 == Grado3
//A7 == Grado3

FN1= digitalRead(A0);
FN2= digitalRead(A1);
FN3= digitalRead(A2);
FN4= analogRead(A6);
FN5= analogRead(A7);
FN6= digitalRead(A3);
BOT=analogRead(A5);
Serial.print("FN1: ");
Serial.print(FN1);
Serial.print(" FN2: ");
Serial.print(FN2);
Serial.print(" FN3: ");
Serial.print(FN3);
Serial.print(" FN4: ");
Serial.print(FN4);
Serial.print(" FN5: ");
Serial.print(FN5);
Serial.print(" FN6: ");
Serial.println(FN6);
}

```

