

Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería Mecánica, Mecánica-Eléctrica y Mecatrónica



**“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE
EXPLORACIÓN MÓVIL HEXÁPODO CONTROLADO
INALAMBRICAMENTE PARA ADQUISICIÓN DE DATOS DE
DIÓXIDO DE CARBONO Y TEMPERATURA CON INTERFAZ
INTERACTIVO VIRTUAL.”**

Tesis presentada por el Bachiller:

Portugal Quito, Boris

Para optar el Título Profesional de:

Ingeniero Mecatrónico

Asesor:

Dr. Quispe Ccachuco, Marcelo Jaime

Arequipa – Perú

2021

DICTAMEN APROBATORIO

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA MECANICA, MECANICA-ELECTRICA Y MECATRONICA
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 23 de Septiembre del 2021

Dictamen: 004079-C-EPIMMEM-2021

Visto el borrador del expediente 004079, presentado por:

2014192121 - PORTUGAL QUITO BORIS

Titulado:

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE EXPLORACIÓN MÓVIL HEXÁPODO
CONTROLADO INALAMBRICAMENTE PARA ADQUISICIÓN DE DATOS DE DIÓXIDO DE CARBONO
Y TEMPERATURA CON INTERFAZ INTERACTIVO VIRTUAL**

Nuestro dictamen es:

APROBADO

**1828 - GORDILLO ANDIA CARLOS ALBERTO
DICTAMINADOR**



**1936 - MESTAS RAMOS SERGIO ORLANDO
DICTAMINADOR**



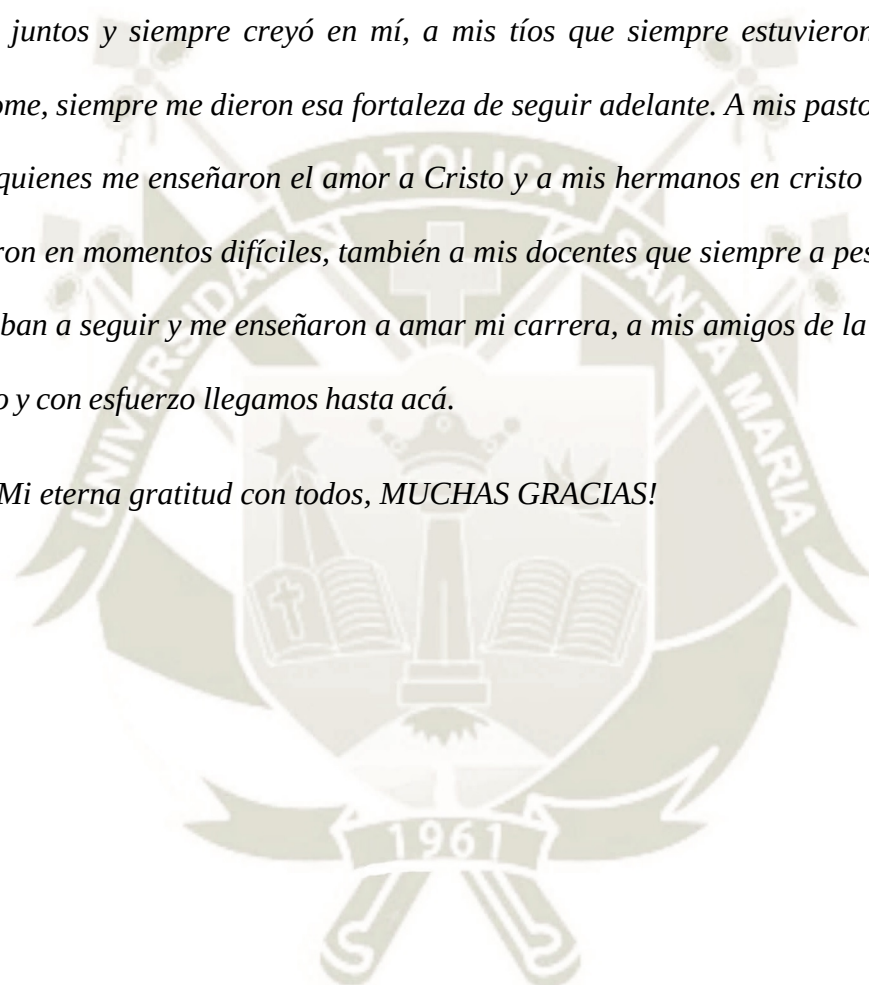
**3244 - SILES NATES FERNANDO DAVID
DICTAMINADOR**



DEDICATORIA

Dedico esta tesis primeramente a Dios que me sostuvo y que me acompañó en todo momento de mi vida, agradezco también a mi padre que en vida fue Antero, a quien siempre agradeceré por su apoyo y sus consejos, a mi mamá Dora quien me apoyó en todo momento y siempre estuvo para mí aconsejándome, a mi hermano Lenin que compartimos muchos momentos juntos y siempre creyó en mí, a mis tíos que siempre estuvieron alentándome y motivándome, siempre me dieron esa fortaleza de seguir adelante. A mis pastores Jorge, Hugo y Gerson quienes me enseñaron el amor a Cristo y a mis hermanos en Cristo quienes siempre me apoyaron en momentos difíciles, también a mis docentes que siempre a pesar de las caídas me motivaban a seguir y me enseñaron a amar mi carrera, a mis amigos de la universidad que estudiando y con esfuerzo llegamos hasta acá.

¡Mi eterna gratitud con todos, MUCHAS GRACIAS!



RESUMEN

El propósito principal del presente trabajo es el diseño y construcción de un robot hexápodo, el cual ha sido desarrollado tomando en cuenta la gran importancia que tienen los robots móviles especialmente cuando son utilizados para realizar tareas en ambientes de alto riesgo para los seres humanos y cuyo terreno presenta irregularidades.

El proyecto ha sido dividido en cinco capítulos e iniciamos con una clasificación de los robots, el planteamiento del problema, las justificaciones académica, económica, técnica y social, los objetivos de la investigación, las variables e independientes y dependientes, la hipótesis de la investigación y la situación actual.

En el capítulo dos se realiza una breve introducción a la robótica móvil y los diferentes sistemas utilizados para la exploración de ambientes empezando por los sistemas con ruedas, robots sin patas y con patas articuladas, haciendo particular énfasis en los robots móviles con seis patas articuladas que son objeto de este estudio. También esta el marco teórico de la presente investigación.

En el capítulo tres se encuentra en la descripción detallada del diseño construcción y ensamblaje del robot, también el diseño y construcción placa de control, las diferentes pruebas de materiales realizados, así como el acondicionamiento de los sensores y el hardware del microcontrolador. No obstante, se realiza la creación del entorno virtual para el control del robot mediante comunicación serial.

En el capítulo cuatro se someterá a las diferentes pruebas al robot, así como los resultados obtenidos de las mismas y las consecuentes limitaciones de la investigación.

El proyecto se diseñó tomando en cuenta la flexibilidad de trabajo de distintos materiales de construcción, hardware y software de control que pueda mejorarse en el futuro. El robot se

podrá usar con fines didácticos para experimentar con algoritmos de computación inteligente y técnicas de control avanzadas.

Palabras clave: Diseño, construcción, Ensamblaje, control, hexápodo, robot, tareas, entorno virtual, hardware y software.



ABSTRACT

The main purpose of this work is the design and construction of a hexapod robot, which has been developed taking into account the great importance of mobile robots, especially when they are used to perform tasks in high-risk environments for humans and whose terrain presents irregularities.

The project has been divided into five chapters and we begin with a classification of robots, the problem statement, the academic, economic, technical and social justifications, the research objectives, the independent and dependent variables, the research hypothesis and the current situation.

In chapter two there is a brief introduction to mobile robotics and the different systems used for the exploration of environments starting with systems with wheels, robots without legs and with articulated legs, with particular emphasis on mobile robots with six articulated legs that are the subject of this study. There is also the theoretical framework of the present research.

In chapter three there is a detailed description of the design, construction and assembly of the robot, as well as the design and construction of the control board, the different material tests performed, and the conditioning of the sensors and the microcontroller hardware. However, the creation of the virtual environment for the control of the robot by means of serial communication is carried out.

In chapter four, the robot will be subjected to the different tests, as well as the results obtained from them and the consequent limitations of the research.

The project was designed taking into account the flexibility of working with different construction materials, hardware and control software that can be improved in the future.

The robot can be used for educational purposes to experiment with intelligent computing algorithms and advanced control techniques.

Keywords: Design, construction, Assembly, control, hexapod, robot, tasks, virtual environment, hardware and software.



INTRODUCCIÓN

El desarrollo de robots a patas ha surgido como una alternativa a los vehículos con ruedas, debido a su fácil acceso a terrenos accidentados, basándose en la naturaleza donde animales como los insectos son capaces de adaptarse a terrenos agrestes o inestables.

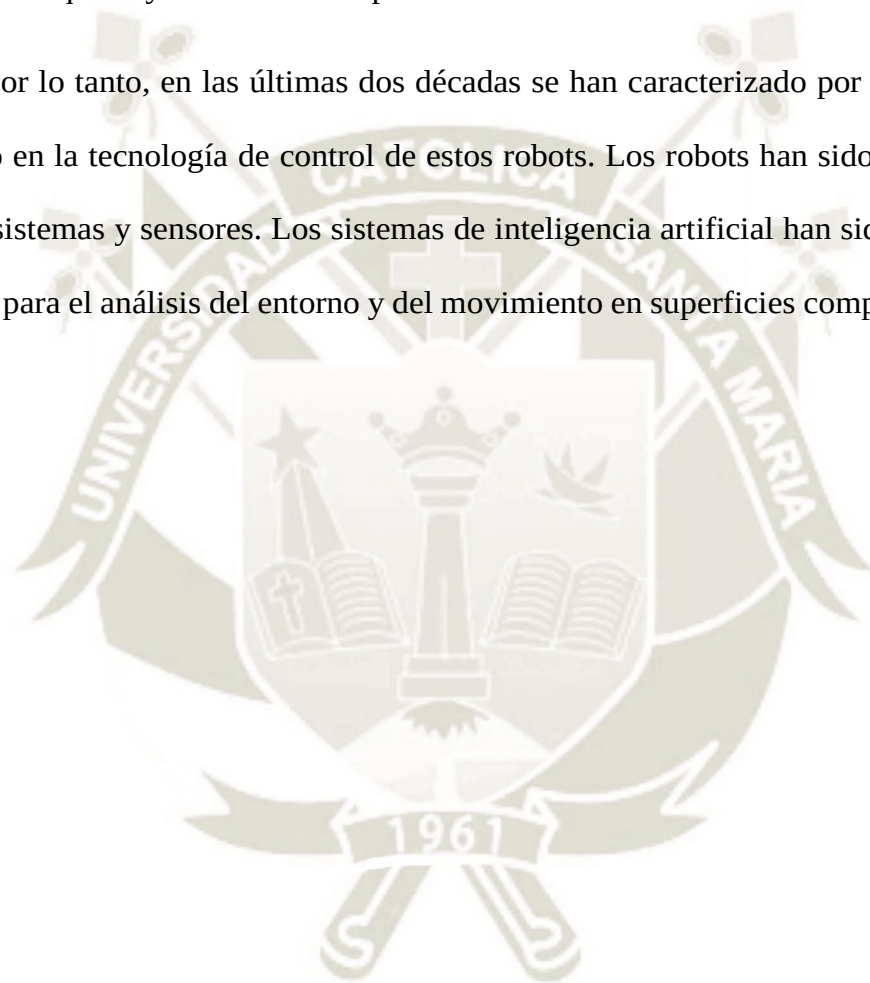
Como afirma Cárdenas Verdugo (2011): “El sistema de movilidad de un hexápodo permite mayor diversidad que un robot móvil convencional (a ruedas), la mayor dificultad es la complejidad en la coordinación de las patas, para mantener el equilibrio por lo cual es necesario conocer el modelo matemático.”

Fernández Fernández (2016) comenta: “Los robots hexápodos se definen como robots programables con seis patas unidas a su cuerpo. Las patas son controladas con un nivel de autonomía tal que el robot puede moverse en distintos terrenos para desempeñar distintas tareas. Dichos robots son útiles en aplicaciones terrestres y pueden incluir características como el movimiento omnidireccional, geometría variable, buena estabilidad y acceso a diversos terrenos.”

Ceccarelli (2005) indica que: “Uno de los factores importantes para el uso de estos robots es que pueden superar obstáculos mucho más grandes que su tamaño equivalente, característica que los distingue de los robots con ruedas. De hecho, el uso de ruedas limita el tamaño del obstáculo ya que solo pueden sortear obstáculos que tengan un tamaño menor a la mitad del diámetro de sus ruedas. Por el contrario, los robots caminantes, sí que pueden sortear obstáculos mayores, aunque, como es lógico, dependiendo del tamaño de sus patas. En el caso particular de los robots hexápodos, se benefician además de un impacto menor con el terreno y una mejor movilidad en ambientes naturales. Esto es especialmente importante en ambientes peligrosos como campos de minas u otro lugar donde sea necesario, por razones científicas, mantener un terreno sin dañar.”

En vista de lo anterior dicho, los robots hexápodos han sido usados en exploraciones lejanas y en ambientes hostiles como ambientes marinos, planetas en el espacio, plantas nucleares. Más allá de este tipo de aplicación, los robots hexápodos también pueden ser usados en una amplia variedad de tareas como recolección de cosechas, ayudas de búsqueda y salvamento de personas u otros servicios.

Por lo tanto, en las últimas dos décadas se han caracterizado por tener un rápido desarrollo en la tecnología de control de estos robots. Los robots han sido equipados con distintos sistemas y sensores. Los sistemas de inteligencia artificial han sido ampliamente aplicados para el análisis del entorno y del movimiento en superficies complejas para estos robots.



INDICE

DICTAMEN APROBATORIO	i
DEDICATORIA	ii
RESUMEN	iii
ABSTRACT	v
INTRODUCCIÓN	vii
CAPÍTULO I	1
1. PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1. ENUNCIADO DEL PROBLEMA	1
1.2. IDENTIFICACIÓN DEL PROBLEMA	1
1.3. DESCRIPCIÓN DEL PROBLEMA	2
1.3.1. ÁREA DE CONOCIMIENTO	3
1.3.2. JUSTIFICACIÓN ACADÉMICA	3
1.3.3. JUSTIFICACIÓN AMBIENTAL	4
1.3.4. JUSTIFICACIÓN TÉCNICA	4
1.4. OBJETIVOS DE LA INVESTIGACIÓN	4
1.4.1. Objetivo principal	4
1.4.2. OBJETIVOS ESPECÍFICOS	4
1.5. ALCANCES	5
1.6. Limitaciones	5
1.7. HIPÓTESIS DE LA INVESTIGACIÓN	6
1.8. VARIABLES DE LA INVESTIGACIÓN	6
1.9. SITUACIÓN ACTUAL	7
CAPÍTULO II	8
2. MARCO TEÓRICO	8
2.1. ANTECEDENTES	8

2.2. Revisión del estado del arte	13
2.2.1. ROBÓTICA	13
2.2.1.1. Robots hexápodos.....	13
2.2.1.2. Locomoción de los robots Móviles	14
2.2.1.3. Robots Caminantes.....	15
2.2.1.4. Robots Hexápodos Bioinspirados.....	17
2.2.1.5. Tipos de Locomoción	18
2.2.1.6. Clasificación de los hexápodos según su grado de libertad	20
2.2.2. Robótica Modular	21
2.2.2.1. Consideraciones.....	22
2.2.3. Cinemática Directa	23
2.2.4. CINEMÁTICA INVERSA	23
2.2.5. LOCOMOCIÓN.....	23
2.3. TEORÍA BÁSICA	23
2.3.1. Matlab.....	23
2.3.2. Processing.....	24
2.3.3. Arduino IDE	25
2.3.4. Atmega 2560	26
2.3.5. Solidworks.....	27
2.3.6. ISIS Proteus.....	27
2.3.7. Impresión 3D.....	28
2.3.8. Servo Tower pro mg995.....	29
2.3.9. Sensor de temperatura DHT22	30
2.3.10. Sensor de CO ₂ o calidad de aire MQ-135.....	31
2.3.11. Lectora de Micro SD	31
2.3.12. Módulo transceptor NRF24L01	32
2.3.13. PROCESSING ENTORNO DE DESARROLLO.....	35
CAPÍTULO III.....	38
3. DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO	38
3.1. METODOLOGÍA DE DISEÑO.....	38
3.1.1. NORMA VDI 2221	38

3.2.	ALCANCES Y LIMITACIONES	38
3.3.	BLACKBOX GENERAL	40
3.4.	ESTRUCTURA DE FUNCIONES	40
3.4.1.	Funciones	40
3.4.1.1.	Parte De Programación para contraerse.....	42
3.4.1.2.	Parte De Programación para pararse.....	43
3.5.	Función Optima.....	44
3.6.	PORTADORES DE FUNCIONES	45
3.7.	CRITERIOS DE SELECCIÓN	49
3.7.1.	Criterio Técnico.....	49
3.7.2.	Criterio Económico	49
3.7.3.	Concepto óptimo (Gráfica Técnico – Económico).....	50
3.8.	BLACKBOX DETALLADA	51
3.9.	DISEÑO PRELIMINAR	51
3.9.1.	ELABORACIÓN DE BOCETOS.....	51
3.10.	DETERMINACIÓN DE CARACTERÍSTICAS	53
3.10.1.	Controlador.....	53
3.10.2.	SELECCIÓN DE ACTUADORES	54
3.10.2.1.	Cálculo De Selección Del Servo Motor	57
3.10.2.2.	Selección del Servomotor.....	58
3.10.3.	Selección de Baterías.....	58
3.10.4.	Selección de comunicación inalámbrica.....	59
3.11.	Modelamiento Matemático del robot.....	61
3.11.1	MODELADO DEL ROBOT.....	61
3.11.2.	CINEMÁTICA DIRECTA.....	61
3.11.3.	Cinemática Inversa	67
3.11.4.	CINEMÁTICA DEL CUERPO.....	71
3.11.4.1.	Operador de Traslación	71
3.11.4.2	Operador de rotación	72

CAPÍTULO IV	78
4. PRUEBAS Y RESULTADOS.....	78
4.1. COMPROBACIÓN EN MATLAB.....	78
4.2. ANÁLISIS DE ESFUERZO.....	80
4.3. Resultado De Análisis De Esfuerzos – Pata Hexápodo.....	82
4.4. DISEÑO DEL CIRCUITO PCB	86
4.5. DISEÑO DEL CIRCUITO PCB MANDO DE CONTROL.....	89
4.5.1. DIAGRAMA DE FLUJO DE CONTROL INALÁMBRICA.....	91
4.6. DISEÑO DEL CIRCUITO PCB SENSADO DE TEMPERATURA Y Co2	92
4.7. PRUEBAS DE INTERFAZ INTERACTIVO VIRTUAL	93
4.7.1. Exportación de archivos SolidWorks a Processing	93
4.8. PRUEBAS DE SENSADO DE CO2 Y TEMPERATURA.	107
4.9. COSTOS DEL DESARROLLO DEL DISEÑO Y CONSTRUCCIÓN.....	109
CONCLUSIONES.....	112
RECOMENDACIONES	114
REFERENCIAS BIBLIOGRÁFICAS	115
ANEXOS.....	126

ÍNDICE DE FIGURAS

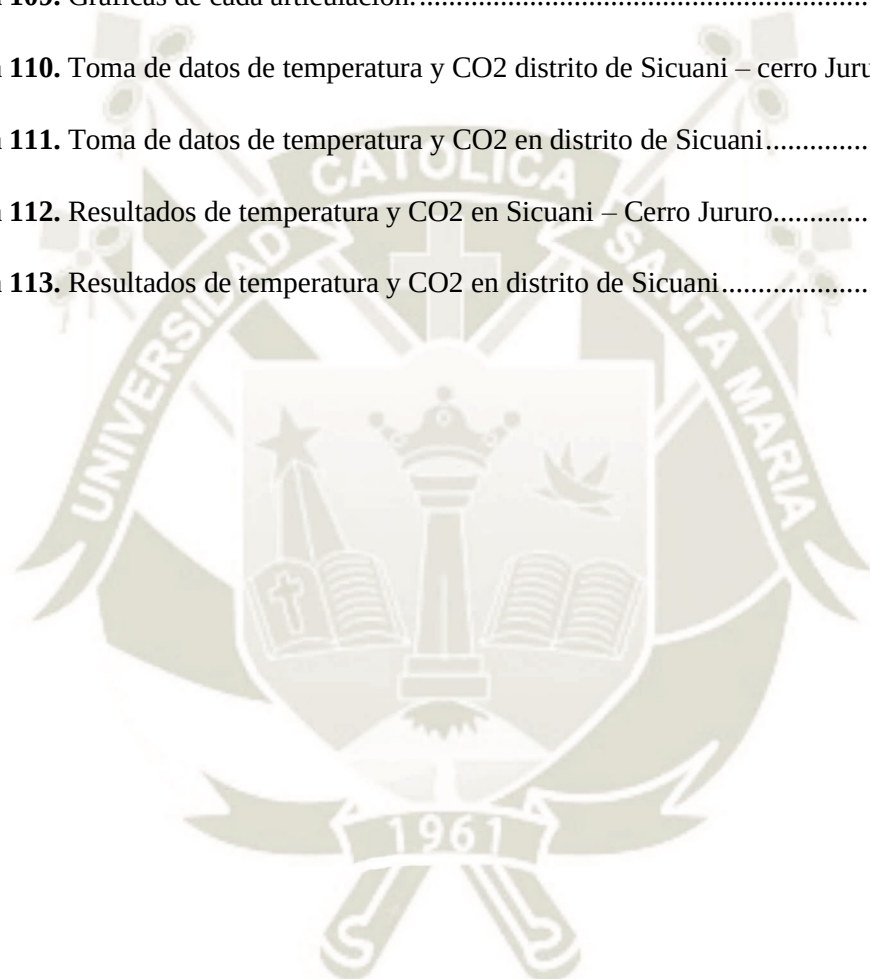
Figura 1. Dante.....	9
Figura 2. Dante II	9
Figura 3. Lauron V.....	10
Figura 4. Lemur.....	11
Figura 5. Athlete.....	12
Figura 6. AQUA Robot.....	12
Figura 7. Desplazamiento de un robot móvil terrestre	14
Figura 8. Ventajas de los robots caminantes	16
Figura 9. Triangulo de apoyo insecto tenebrio.....	17
Figura 10. Ventajas de la locomoción por patas topología hexápodo.....	18
Figura 11. Trayectoria que sigue una pata para que el robot se desplace.....	19
Figura 12. Hexápodos de a) 1 grado, b) 2 grados y c) más grados de libertad.....	20
Figura 13. Interfaz Proccesing IDE.....	24
Figura 14. Interfaz Arduino IDE.....	25
Figura 15. Microcontrolador Atmega 2560.....	26
Figura 16. Solidworks.....	27
Figura 17. Proteus y diseño de PCB Ares.....	28
Figura 18. Impresoras 3d.....	28
Figura 19. Servomotor Tower pro mg995.....	30
Figura 20. Sensor DHT22.....	30
Figura 21. Sensor MQ-135.....	31
Figura 22. Lectora Micro SD	32
Figura 23. NRF24L01	33
Figura 24. Canales de Onda	33
Figura 25. Modulo NRF24L01	34

Figura 26. Conexión con Arduino.....	35
Figura 27. Entorno de Processing.	36
Figura 28. pasos para el ejecutable.....	37
Figura 29. Blackbox general	40
Figura 30. Bosquejo de robot y numeración de articulaciones	40
Figura 31. Bosquejo de pata completa y sus articulaciones	41
Figura 32. Secuencia de movimiento para Contraerse - General	42
Figura 33. Secuencia de movimiento para Pararse - General.....	43
Figura 34. Secuencia de movimiento para Avanzar tipo TRIPOD	44
Figura 35. Criterio Técnico vs Criterio Económico.....	50
Figura 36. Blackbox.....	51
Figura 37. Boceto 1.....	51
Figura 38. Boceto 2.....	52
Figura 39. Boceto 3.....	52
Figura 40. Atmega 2560.....	54
Figura 41. Servo motor tower pro MG995.....	57
Figura 42. Pata Robot Hexápodo.	61
Figura 43. Posición cero Pata de robot.....	62
Figura 44. Posición creo primera elaboración.....	62
Figura 45. ángulo de inclinación real	63
Figura 46. distancias halladas	64
Figura 47. ángulo hallado.....	64
Figura 48. representación gráfica vista superior	67
Figura 49. Determinación de variables.	67
Figura 50. vista frontal pata de robot	68
Figura 51. vista frontal pata de robot, variable a1.....	68

Figura 52. vista frontal pata de robot, variable a_2	69
Figura 53. posición cualquiera z positivo q_2	69
Figura 54. vista frontal pata de robot, variable b_1	70
Figura 55. posición cualquiera z positivo para q_3	70
Figura 56. Traslación del vector u_1	71
Figura 57. Rotación del vector u_1	72
Figura 58. Posición de Caminata.	73
Figura 59. traslación en x.	74
Figura 60. traslación en y.	74
Figura 61. traslación en z	75
Figura 62. Rotación en x.	75
Figura 63. Rotación en y.	76
Figura 64. Rotación en z.	76
Figura 65. Volumen de Trabajo	77
Figura 66. Resultados de Cinemática directa e inversa.....	78
Figura 67. Valores Articulares	78
Figura 68. valores de posición en x, y, z.....	79
Figura 69. simulación peter Corke.	79
Figura 70. Trayectoria Parabólica	80
Figura 71. Robot Hexápodo diseñado en SolidWorks 2019.	80
Figura 72. Esfuerzos de Von Mises, desplazamiento y deformación Acople C1.....	83
Figura 73. Esfuerzos de Von Mises, desplazamiento y deformación Caja Servo.	84
Figura 74. Esfuerzos de Von Mises, desplazamiento y deformación Acople C2.....	84
Figura 75. Esfuerzos de Von Mises, desplazamiento y deformación Pata 1.....	85
Figura 76. Esfuerzos de Von Mises, desplazamiento y deformación.....	86
Figura 77. Circuito esquemático placa de control hexápodo.	87

Figura 78. PCB circuito de control Hexápodo.	88
Figura 79. Placa de control del robot Hexápodo.	88
Figura 80. Placa de control del robot Hexápodo Físico.	89
Figura 81. Circuito esquemático placa mando de control hexápodo.	89
Figura 82. Placa mando de control del robot Hexápodo Físico.	90
Figura 83. Diagrama de flujo de control de robot hexápodo.....	91
Figura 84. Circuito esquemático placa de sensado de temperatura y co2.	92
Figura 85. Placa de Sensado temperatura y co2.....	92
Figura 86. Opciones para cambiar una sola pieza.....	93
Figura 87. Unión de una pieza en archivo STL.....	94
Figura 88. Guardar archivo STL.	94
Figura 89. Exportación del archivo STL a blender	95
Figura 90. Dar color a la pieza exportada.	95
Figura 91. Exportar archivo con extensión. Obj	96
Figura 92. Entorno de trabajo de Processing.....	97
Figura 93. Entorno de trabajo de processing coordenadas	98
Figura 94. Vista isométrica robot ensamblado.....	98
Figura 95. Secuencia de configuración entorno virtual.....	99
Figura 96. conexión puerto COM.	99
Figura 97. Activación botón posición.	100
Figura 98. Activación de la cinemática directa.	100
Figura 99. Activación de la cinemática Inversa.	101
Figura 100. Activación de la cinemática del Cuerpo Traslación eje X.	101
Figura 101. Activación de la cinemática del Cuerpo Traslación eje Y.	102
Figura 102. Activación de la cinemática del Cuerpo Traslación eje Z.	102
Figura 103. Activación de la cinemática del Cuerpo Rotación eje X.	103

Figura 104. Activación de la cinemática del Cuerpo Rotación eje Y.	103
Figura 105. Activación de la cinemática del Cuerpo Rotación eje Z.....	104
Figura 106. Activación de Movimiento Circular	104
Figura 107. Activación de caminata Tripod.....	105
Figura 108. Activación de gráficas y caminata tripod.....	106
Figura 109. Graficas de cada articulación.....	106
Figura 110. Toma de datos de temperatura y CO2 distrito de Sicuani – cerro Jururo.	107
Figura 111. Toma de datos de temperatura y CO2 en distrito de Sicuani.....	107
Figura 112. Resultados de temperatura y CO2 en Sicuani – Cerro Jururo.....	108
Figura 113. Resultados de temperatura y CO2 en distrito de Sicuani.....	109



ÍNDICE DE TABLAS

Tabla 1. Tabla de Variables.....	6
Tabla 2. Patas en contacto con el suelo y suspendidas en cada forma de desplazamiento.	19
Tabla 3. Resultados de las secuencias de Bessonov y Umnov.	22
Tabla 4. Características de microcontrolador Atmega 2560	26
Tabla 5. Lista de Deseos y exigencias.....	39
Tabla 6. Tabla de portadores de funciones	45
Tabla 7. Tabla ponderaciones – Criterio Técnico.....	49
Tabla 8. Tabla ponderaciones – Criterio Económico.	49
Tabla 9. Tabla ponderaciones – Criterio Técnico y Económico.	50
Tabla 10. Tabla ponderaciones – BOCETOS.....	52
Tabla 11. Tabla características de selección de microcontrolador	53
Tabla 12. Selección de servo motor	58
Tabla 13. Selección de las baterías.....	58
Tabla 14. Tabla de selección de comunicación inalámbrica.	59
Tabla 15. Tabla de Características Robot hexápodo.	60
Tabla 16. Parámetros Denavit-Hartenberg para extremidades.	63
Tabla 17. Nuevos Parámetros Denavit-Hartenberg para extremidades.	65
Tabla 18. Componentes para el análisis de resistencia.....	82
Tabla 19. Costo para la placa electrónica de control	109
Tabla 20. Costo para la placa Estructura.	110
Tabla 21. Costo para la placa módulo de sensado.	110
Tabla 22. Costo para la placa del mando de control.....	110
Tabla 23. Costo total	111

CAPÍTULO I

1. Planteamiento de la investigación

1.1. Enunciado del problema

“Diseño y construcción de un prototipo de exploración móvil hexápodo controlado inalámbricamente para adquisición de datos de dióxido de carbono y temperatura con interfaz interactivo virtual.”

1.2. Identificación del problema

“Recientemente las investigaciones en robótica se enfocan hacia el desarrollo de máquinas caminantes de múltiples articulaciones (robots caminantes) y esto es debido a que en algunas aplicaciones se requiere de una considerable movilidad, sobre todo si existe la necesidad de locomoción en terrenos irregulares”, (Según Efren G. ,2002).

En la actualidad existe mucha información acerca de robots hexápodos que fueron desarrollados en diferentes países para el uso de exploración ya sea para terrenos regulares e irregulares, así como el modelamiento y validación de estos en simulación como Matlab, Python, Ros, etc. Las cuales pocos fueron plasmados a una forma real debido al precio muy elevado que es muy importante, también a la avanzada tecnología que se requiere, tanto en la integración de conocimientos que abarca este proyecto como diseño mecánico, telecontrol, electrónica, etc.; todo esto conlleva a una necesidad de adquirir los conocimientos antes mencionados para la construcción y comprobación de desarrollo de robot hexápodos y plantear alternativas de solución, porque se sabe que la simulación nos da un resultado que es irreal de la parte física debido a muchos factores que influyen.

En muchas investigaciones se hace el uso de robots móviles con ruedas para exploración y adquisición de datos ya sea en volcanes inactivos, nieve, terrenos superficiales planas, pero tuvo muchos inconvenientes al tener obstáculos, terrenos

irregulares, incluso para terrenos arenosos, hielo o nieve se tiene a fabricarse un tipo de rueda y esto implicaba el aumento de costo y fabricación del proyecto.

En el siguiente proyecto se propone al desarrollar el estudio de diseño mecánico, comunicación inalámbrica, comportamiento cinemático y locomoción de robots hexápodos tanto en simulación y parte física del robot; también en la adquisición de datos de Co2 y temperatura mediante sensores MQ-135 y DHT22 respectivamente.

1.3. Descripción del problema

Hoy en día la aplicación de robots para exploración y de adquisición de datos en lugar inaccesibles por el ser humano fueron desarrollados antes de los años 90, en la mayoría de las entidades como el IGP (Instituto Geofísico del Perú) que son los encargados de monitorear los efectos de contaminación y sucesos naturales, los cuales son encargados de adquirir datos ya sea de partículas de carbono negro en la nieve la cual produce el deshielamiento de los glaciares o datos de CO₂, temperatura, etc. Para que esto se pueda monitorear continuamente de los desastres naturales en nuestra localidad, también el traslado de equipos de medición las cuales tienen un costo elevado y llevarlo hacia un lugares difíciles de acceder sea un glaciar o volcán inactivo, surge la necesidad de crear un móvil para que pueda facilitar estas mediciones y pueda ser optima el trabajo de adquisición de datos sin exponer a los profesionales de sufrir algunos inconvenientes al hacer estas adquisiciones y/o sufrir un accidente, vemos muchos modelos y patentes de robots Hexápodos en el **Anexo 1**.

Medible:

En la actualidad la infinidad de robots hexápodos de exploración fueron evolucionando uno de los problemas en la mayoría de estos proyectos es abocarse solo a la construcción o simulación, pero en pocos casos se integró ambas partes, el siguiente

proyecto busca obtener estos resultados de la simulación y comprobarlo con el prototipo de robot hexápodo.

Solucionable:

El problema surge cuando no hay un buen diseño o calculo porque varía el funcionamiento y no hay estabilidad y por otra parte de un buen control del robot, lo cual se puede solucionar con un buen diseño y modelamiento matemático.

Flexible:

Todos los materiales necesarios para la construcción del robot son adquiribles en el mercado local.

1.3.1. Área de conocimiento

El área de conocimiento a desarrollar se basa en el diseño y construcción de un robot de morfología hexápoda con sistemas mecánicos, electrónicos y de programación, precisando en el método de la robótica.

- ÁREA: Ingeniería Mecatrónica
- SUB ÁREA: ingeniería mecánica, ingeniería electrónica, ingeniería de sistemas.
- LÍNEA: robótica

1.3.2. Justificación académica

La presente investigación busca mostrar los avances que se han desarrollado para adaptarse a las nuevas tecnologías del mercado y de la industria, y profundizar los conocimientos teóricos adquiridos sobre los procesos de robótica, control y automatización, utilizando conceptos y conocimientos ofreciendo una mirada integral sobre el conocimiento generado en tesis y papers ya existentes en la actualidad, así mismo añadir nuevas aplicaciones de la siguiente investigación.

1.3.3. Justificación ambiental

Mediante la elaboración del siguiente proyecto se desarrollará en la tecnología de impresión 3D la cual el material a utilizar será PLA (ácido poliláctico) la cual es un material biodegradable, es un políéster termoplástico y comúnmente se deriva de recursos renovables, como el almidón de maíz, las raíces de tapioca o la caña de azúcar. Una de las cosas más atractivas del plástico PLA es que se degrada naturalmente cuando se expone al medio ambiente. También este material tiene la certificación SGS es líder mundial en inspección, verificación, análisis y certificación.

1.3.4. Justificación técnica

Mediante el análisis detallado de los planos realizados en SolidWorks, teniendo en cuenta las medidas, basándonos en proyectos antepasados, así como la indagación acerca de los componentes necesarios para el diseño del robot hexápodo, concluimos que este prototipo de robot hexápodo es factible de ser utilizado, por los resultados obtenidos en la simulación en software. Además, permitirá complementar los conocimientos obtenidos para futuros proyectos relacionados a robot hexápodos.

1.4. Objetivos de la investigación

1.4.1. Objetivo principal

Diseñar y construir un prototipo de robot móvil hexápodo controlado inalámbicamente para adquisición de datos de dióxido de carbono y temperatura con interfaz interactivo virtual.

1.4.2. Objetivos específicos

1. Modelar matemáticamente la morfología hexápoda para un sistema de prototipo robótico.
2. Diseñar el modelo mecánico del robot hexápodo.

3. Implementar un control de interfaz interactiva para el robot.
4. Evaluar la funcionalidad controlando inalámbricamente la morfología hexápoda y monitoreo de gases CO₂ y temperatura mediante sensores MQ-135 y DHT22 respectivamente.

1.5. Alcances

Los alcances del proyecto son diseño, construcción y adquisición de datos de CO₂ y temperatura con sensores MQ-135 y DHT22 respectivamente.

Adicionalmente se añadirá un interfaz interactivo virtual para facilitar el entendimiento e indicación de los resultados obtenidos, también el de controlar el robot.

El proyecto se basa en cuatro conceptos: diseño y construcción del robot hexápodo, control inalámbricamente del robot, interfaz interactivo virtual que nos permitirá controlar de forma manual y obtener resultados del robot hexápodo, adquisición de datos de CO₂ y temperatura.

El proyecto se desarrollará con fines demostrativos la cual puede ser empleado para poder mejorar en los conocimientos y optimizar el proyecto.

1.6. Limitaciones

El proyecto se realizará con un presupuesto limitado de los productos obtenidos localmente y también con la importación por su bajo coste.

Se aplicará la metodología de diseño de máquinas norma VDI 2221, para este prototipo, también se realizará el modelamiento matemático e implementación de la cinemática directa e inversa al robot hexápodo.

El robot podrá desplazarse en una superficie plana la cual será controlada inalámbricamente con un mando de control o interfaz.

El robot se podrá controlar mediante un mando de control inalámbrico con una distancia máxima de 100 metros al aire libre.

El tiempo de autonomía es de 15 minutos continua por las baterías que vamos a utilizar, es uno de los inconvenientes grandes de un equipo o robot controlado.

1.7. Hipótesis de la investigación

Dado que, los robots hexápodos controlados, son muy eficientes al tener estabilidad en superficies irregulares ya que los robots móviles de ruedas para exploración no son muy eficaces a la hora de desplazarse por dichos terrenos, la cual dificulta la movilidad.

Es probable, reducir esos inconvenientes con el diseño y construcción del robot hexápodo.

1.8. Variables de la investigación

En la **tabla 1** se muestra la designación de variables dependiente e independientes.

Tabla 1. Tabla de Variables.

Fuente: Elaboración propia

VARIABLES	TEMA	INDICADORES
Variable Independiente	Modelamiento matemático	Distribución espacial de las extremidades del robot
	Diseño Mecánico	Tipo del material utilizado.
		Torque en los actuadores.
	Adquisición de datos	Sensores de Co2 MQ-135 y temperatura DHT22
Variable Dependiente	Interfaz interactiva virtual y física del robot hexápodo	Posición del robot en el plano cartesiano para su movimiento.
		Control del robot hexápodo inalámbricamente

1.9. Situación actual

Además de lo que pudimos observar en párrafos anteriores acerca de los trabajos que se han hecho con robots hexápodos a lo largo de la historia, queremos ahondar en trabajos realizados acerca de este tema, pero en un periodo menor de 10 años atrás para considerar la situación existente en el mundo y tener un conocimiento más específico de cómo se trabaja con estos robots y hasta dónde y cómo se puede llegar.



CAPÍTULO II

2. MARCO TEÓRICO

2.1. Antecedentes

El auge de la robótica móvil en los últimos años ha permitido el desarrollo de una gran cantidad de robots que pueden realizar distintas tareas tanto en entornos estructurados como en aquellos no tan estructurados. Así, los robots móviles son utilizados para realizar operaciones de búsqueda y rescate, exploración planetaria, operaciones de tipo militar y apoyo a actividades de agricultura, entre otras. Todas estas tareas tienen un común denominador: la navegación sobre terrenos irregulares con inclinaciones y obstáculos que deben ser sobrepasados por el robot. (Caicedo, 2015)

“El término ‘robot’ es usado por primera vez en 1921 por el escritor checo Karel Čapek, aunque atribuía la idea a su original a su hermano, en la obra R.U.R: Rossum’s Universal Robots derivado de robota donde su concepto trataba de seres humanos simplificados, fabricados en masa, libres de sentimientos y creatividad, creados única y exclusivamente para servir a la humanidad. Sin embargo, aún estábamos lejos de la conceptualización de robot como lo conocemos hoy en día; cuarenta años después en 1961 el primer robot programable industrial se instaló en General Motors desarrollado por Joe Engelberger y George Devol, UNIMATE apilaba piezas calientes de metal fundido.” (World Information, 2007).



Figura 1. Dante

Fuente: Tomado de Dante & Dante II, 1992

“En 1994, Dante II, una versión más robusta de su predicador, descendió al cráter del volcán de Alaska en el monte. Spurr y completó la misión con éxito.” ,(Bares ,2012).



Figura 2. Dante II

Fuente: Tomado de Dante & Dante II, 1992

Una serie de robots bio-inspirados fueron desarrollados en la universidad de Case Western Reserve (EEUU) a finales de los años 90 como, por ejemplo, Robot III. Este robot tenía un total de 24 grados de libertad y su arquitectura estaba basada en la cucaracha, intentando imitar su comportamiento. En particular, cada pata trasera

disponía de tres grados de libertad, cada pata del medio de cuatro grados y cada pata delantera de cinco grados de libertad.

En el año 2001, se comenzó el proyecto RHex que aún sigue vigente. Este robot nació del consorcio de varias universidades y su diseño consistía en un cuerpo rígido con seis patas articulares de un grado de libertad cada una de ellas. En consecuencia, tenía seis motores que giraban las patas como una rueda.” (U. Saranli, ,2001)

“El robot hexápodo Lauron V fue el resultado de 10 años (años 2003 - 2013) de progresivas mejoras en las previas configuraciones Lauron I, II, III y IV. Lauron estuvo inspirado en el insecto palo por lo que el robot tenía de seis patas acopladas al cuerpo central. Cada una de las seis patas disponía de cuatro articulaciones. Además, cada extremo de la pata tenía un sensor 3D y cada motor tenía un sensor de corriente que detectaba la fuerza que se oponía al movimiento.” (Roennau ,2013)



Figura 3. Lauron V

Fuente: Tomado de Roennau A., 2014

“En 2006, el robot Gregor I reprodujo la agilidad de una cucaracha estando basado el control del movimiento en la teoría de Generador Central de Patrones. Gregor I se caracterizó porque cada una de sus patas tenía un diseño único

disponiendo el par de patas delanteras y del medio de tres grados de libertad y las patas traseras de dos grados.” (B. Kennedy, 2005).

Una serie de robots llamados LEMUR fue desarrollada por el laboratorio Jet Propulsion con los objetivos de usarlos para reparar y mantener plataformas en zonas con gravedad cero como en la superficie de una nave espacial. Estos dieron lugar a la plataforma de investigación de robots móviles hexápodos llamada MARS (T. Hause, 2008), cuyas misiones actualmente están orientadas a la búsqueda de evidencia de habitabilidad, tafonomía (relativo a los fósiles) y carbono orgánico en el planeta Marte. (Showalter, 2009).

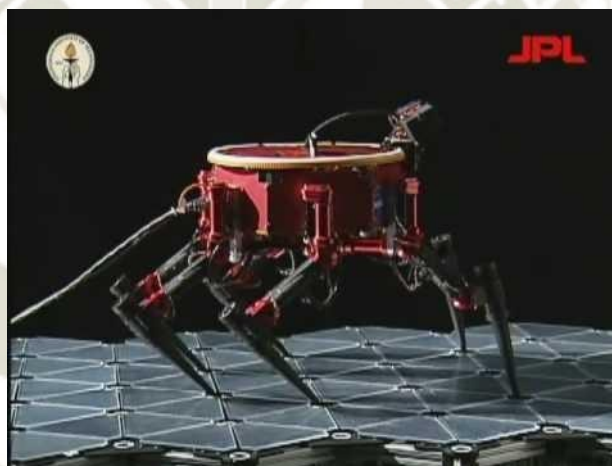


Figura 4. Lemur

Fuente: Tomado de Showalter, 2009

En 2006, el laboratorio Jet Propulsion, que también había desarrollado la serie de robot LEMUR, desarrolló el robot hexápodo ATHLETE. Este robot tenía la habilidad de rodar rápidamente sobre superficies planas con las ruedas que disponía en los extremos de las patas y andar cuidadosamente con las ruedas fijas sobre terrenos irregulares y escarpados. (C. Georgiadis, 2004).

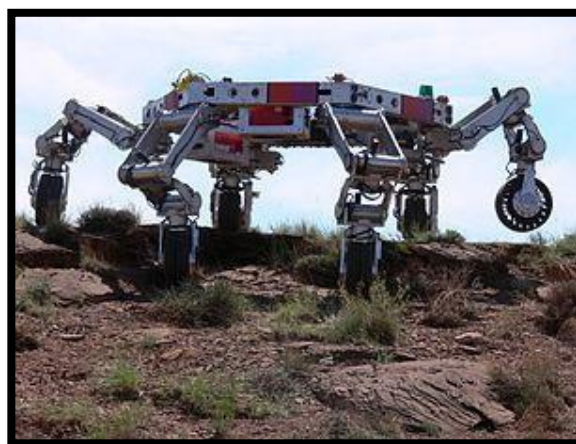


Figura 5. Athlete

Fuente: Tomado de Georgiadis, 2004

Por último, AQUA fue un robot hexápodo desarrollado en el año 2011 con seis actuadores, uno en cada pata, independientes. Una de las características más importantes de este robot era la capacidad de cambiar de modos de marcha andantes a nadadores conforme iba cambiando de zona con arena a agua profunda. (K. Nonami, 2001).

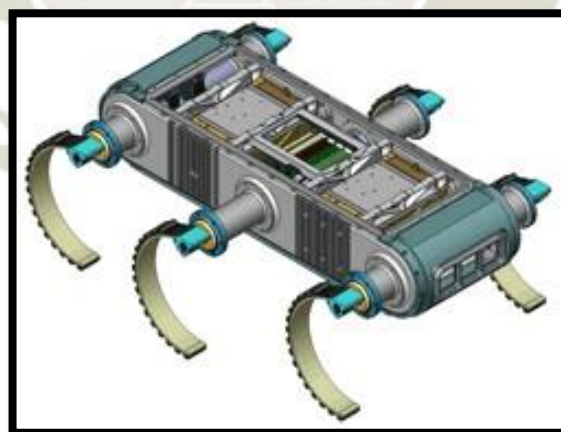


Figura 6. AQUA Robot

Fuente: Tomado de Nonami, 2001

“Actualmente ANYmal, es el robot móvil con patas totalmente autónomo equipado para tareas de búsqueda y rescate el cual puede desplazarse en terrenos

complicados con desniveles, escalones u obstáculos además de ser a prueba de agua y soportar temperaturas de entre los -30 y 40 grados centígrados.” (ETH Zurich, 2016).

2.2. Revisión del estado del arte

2.2.1. Robótica

La robótica es la rama de la ingeniería mecánica, de la ingeniería eléctrica, de la ingeniería electrónica, de la ingeniería biomédica, y de las ciencias de la computación, que se ocupa del diseño, construcción, operación, estructura, manufactura, simulación y aplicación de los robots. (Definición de robótica - RAE , 2008).

La robótica combina diversas disciplinas como: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física. Otras áreas importantes en robótica son el álgebra, metodologías de solución de los problemas algebraicos, los autómatas programables, la animatrónica y las máquinas de estados. (Robotics from Monster Career Advice , 2007).

2.2.1.1. Robots hexápodos

Los robots hexápodos han atraído considerablemente la atención durante décadas, habiéndose realizado numerosos estudios en centros de investigación, universidades e industrias. Sin embargo, solo en los últimos años los robots caminantes han sido diseñados y contruidos con estructuras suficientemente óptimas para ser útiles en diversas aplicaciones y terrenos. (Javier Fernandez F, 2016)

Los robots hexápodos se definen como robots programables con seis patas unidas a su cuerpo. Las patas son controladas con un nivel de autonomía tal que el robot puede moverse en distintos terrenos para desempeñar distintas tareas. Dichos robots son útiles en aplicaciones terrestres y pueden incluir características como el movimiento omnidireccional, geometría variable, buena estabilidad y acceso a diversos terrenos. (J. L. Villarroel, 2010)

Los robots hexápodos han sido usados en exploraciones lejanas y en ambientes hostiles como ambientes marinos, planetas en el espacio, plantas nucleares. Más allá de este tipo de aplicación, los robots hexápodos también pueden ser usados en una amplia variedad de tareas como recolección de cosechas, ayudas de búsqueda y salvamento de personas u otros servicios, (Javier Fernandez F, 2016).

2.2.1.2. Locomoción de los robots Móviles

La robótica móvil como su nombre lo indica, abarca todo sistema robótico destinado a realizar desplazamientos en su entorno, comprendiendo aquellos capaces de llegar a su destino por tierra, agua, aire o cualquier terreno para el cual el robot fue diseñado. Existen diversas formas para lograr movimientos, algunos usando modelos inspirados en la naturaleza como los robots humanoides que simulan la marcha humana y otros a partir de mecanismos comúnmente usados en vehículos como la rueda y orugas. Estos robots desempeñan trabajos de exploración, vigilancia, labores de rescate entre otros, por lo que la gran mayoría están equipados con sensores que le permiten conocer su ubicación y el tipo los objetos que lo rodean, (Cotera Berdugo, 2017).

Este trabajo se centra en un robot terrestre, este tipo de robots son diseñados para desplazarse por uno o varios tipos de superficie determinadas. Esta categoría comprende a los robots capaces de caminar, correr, escalar, deslizarse o rodar por lo que existen mecanismos para todo tipo de terreno entre los más comunes las patas, orugas y ruedas. En la **figura 7** se aprecian los tipos más comunes de robot terrestres. (Cotera Berdugo, 2017).

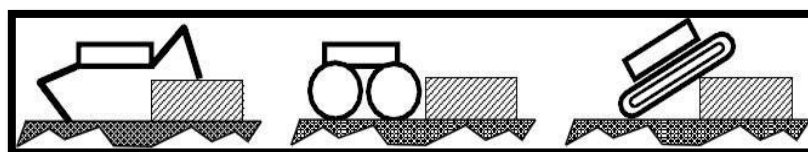


Figura 7. Desplazamiento de un robot móvil terrestre

Fuente: Cárdenas Verdugo, 2011

2.2.1.3. Robots Caminantes

Dentro del diseño de robots móviles, uno de los más utilizados es el que se da a base de marchas que implican la coordinación de patas para lograr desplazarse. El uso de patas en la robótica data de los diversos modelos bio inspirados presentes en la naturaleza y el éxito que han tenido las distintas especies para adaptarse a las condiciones más difíciles, logrando desplazarse de forma eficiente sobre superficies irregulares de todo tipo. (Cotera Berdugo, 2017).

El uso de extremidades implica una coordinación motriz rítmica en cada paso que puede apreciarse, como una señal cíclica que cuenta con periodo y fase. Esta señal depende también de la estructura del animal, dando como resultado una serie de movimientos conocidos como patrones de locomoción o patrones de marcha, en el que se logra especificar sus capacidades motoras, entre las que se encuentran el trote, correr, reptar, saltar entre otras (De León Melgarejo, 2010). Los robots caminantes aprovechan los diseños dados por la adaptación para determinar sus capacidades y limitantes. Un ejemplo de estos son los robots bípedos inspirados en la marcha humana, en donde uno de los mayores problemas es el equilibrio, problema que no se presenta en los cuadrúpedos por su estabilidad, pero se sigue trabajando en ellos debido a la facilidad que tienen los humanos para saltar, subir escalera y sortear obstáculos en comparación con especies de cuatro patas.

Los robots caminantes presentan varias ventajas en comparación de otros sistemas como los que utilizan ruedas, estas están dadas gracias a las capacidades como la flexibilidad y apoyo que otorga cada pata en las que se destacan:

- Mayor movilidad al ser capaces de cambiar de dirección sin importar la orientación del cuerpo.

- Capacidad para superar obstáculos al pasar sobre ellos subiendo las patas.
- Suspensión activa al mantenerse apoyado sobre el terreno en todo momento, esto es gracias a la capacidad de estirar y contraer cada pata.
- Marcha con independencia del terreno al no necesitar un espacio continuo para moverse, estos son capaces de moverse sobre arena, piedras, pasto en otras superficies difíciles para las ruedas.
- Se reduce el deslizamiento al levantar y apoyarse cada pata en la superficie en lugar de girar y verse afectado por la resistencia del terreno.
- Menor desgaste por el terreno debido a un menor tiempo de contacto.
- Velocidad promedio en terrenos irregulares, a pesar de ser más lentos que los que usan ruedas en superficies planas, estos son más constantes al marchar sobre cualquier terreno, (Cotera Berdugo, 2017).

En la **figura 8** se representan estas ventajas para dar un mejor entendimiento de los robots caminantes.

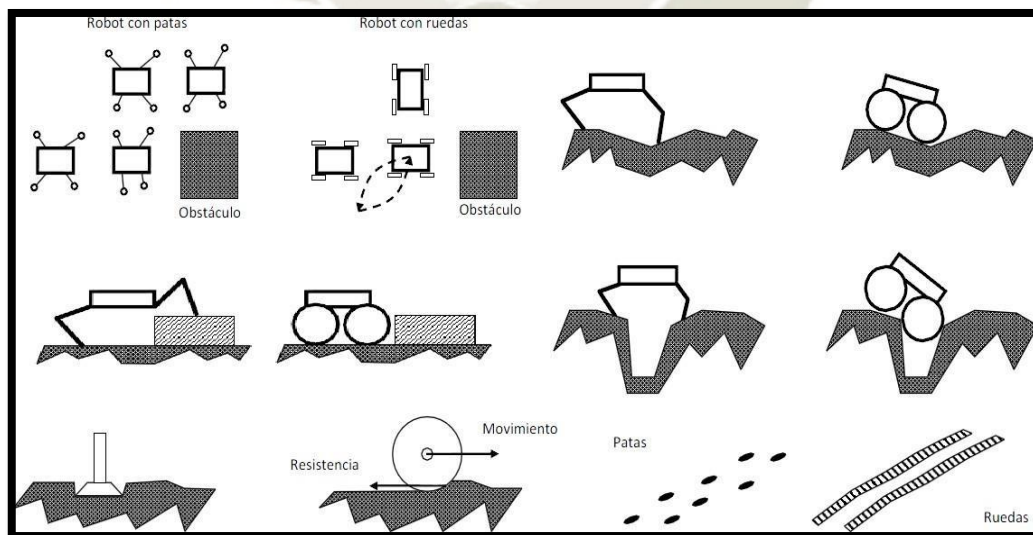


Figura 8. Ventajas de los robots caminantes.

Fuente: Cárdenas Verdugo, 2011

2.2.1.4. Robots Hexápodos Bioinspirados

Dentro de la gama de robots móviles, los insectos han sido uno de los modelos de inspiración más utilizados, debido al éxito que han tenido estos para adaptarse a todo tipo de terreno, destacando su velocidad de desplazamiento y la estabilidad en sus movimientos que mantienen incluso estando de cabeza. Su característica principal es que poseen 6 patas a lo que se denomina como morfología hexápoda, el número de patas implica mayores puntos de contacto con la superficie durante la locomoción, esta cualidad les permite mantenerse estables siempre, resolviendo uno de los problemas más comunes en el diseño de robots simplificando el control (Cárdenas Verdugo, 2011). Los robots hexápodos tienen en cuenta sus articulaciones y grados de libertad imitando los 3 grados por pata de los insectos, su marcha consiste en mantener apoyadas 3 de sus patas mientras usa las restantes para avanzar, tanto en los robots como en los insectos este movimiento genera un triángulo de apoyo que mantiene el centro de gravedad dentro del área de apoyo (Sedeño Bustos, 2011). En la **figura 9** se observa el triángulo de estabilidad que se forma sobre las tres patas apoyadas del insecto tenebrio que posee morfología hexápoda.

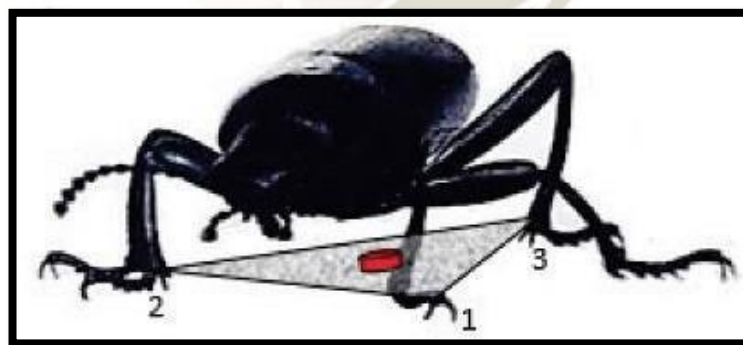


Figura 9. Triángulo de apoyo insecto tenebrio.

Fuente: Rodríguez Calderón, Ramos Parra, & Peña Giraldo, 2015

Los insectos y animales con más de cuatro patas son capaces de caminar inmediatamente nacen, teniendo resuelto el problema de balanceo a diferencia de los

cuadrúpedos, que demoran un tiempo más en ponerse de pie y algo más de tiempo para empezar a marchar, estando por último los bípedos como los humanos que tardan meses para ponerse de pie y aprender a caminar al no contar con posiciones estáticamente estables (Cárdenas Verdugo, 2011). Esto es tomado en cuenta por los investigadores, ya que a pesar de que los cuadrúpedos también forman ese polígono de estabilidad, los robots con mayor número de patas son más robustos y se les facilita lograr los movimientos, obteniendo todas las ventajas de los robots caminantes con mayor estabilidad (Cárdenas Verdugo, 2011). Estas ventajas y capacidades de movimiento se logran apreciar en la **Figura 10** que representa las ventajas de la topología hexápoda, (Cárdenas Verdugo, 2011).

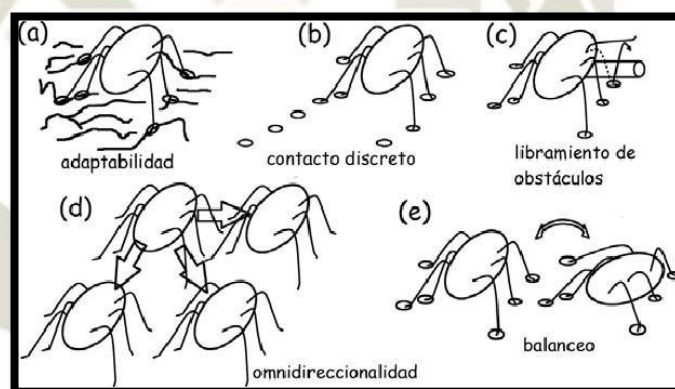


Figura 10. Ventajas de la locomoción por patas topología hexápoda.

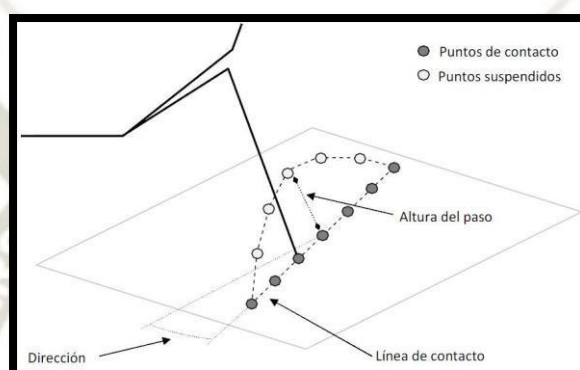
Fuente: De León Melgarejo, 2010

2.2.1.5. Tipos de Locomoción

Como ya se ha visto, la morfología hexápoda puede generar movimientos omnidireccionales como traslación y rotación, en donde el movimiento es observable con las posiciones del cuerpo del robot ya que las patas permanecen sujetas a este, y solo ejecutan un movimiento coordinado siguiendo una trayectoria en cortos periodos de caídas controladas en el que se levanta y se deja caer cada pata, cuidando que el sistema no se torne inestable (Sedeño Bustos, 2011). El movimiento de cada pata forma arcos con

respecto a la superficie, en donde se manejan parámetros como la línea en contacto directo con el suelo que define el desplazamiento y dirección que va a tomar el hexápodo, junto a los puntos suspendidos y de contacto que siguen a la trayectoria, (Cárdenas Verdugo, 2011).

En la **figura 11** se describe el movimiento de la pata para dar un paso, en donde la extremidad pasa de estar apoyada en el suelo, a ser levantada y llevada en la dirección.



Fuente: Cárdenas Verdugo, 2011

Figura 11. Trayectoria que sigue una pata para que el robot se desplace.

En un hexápodo, el número mínimo de patas sobre las superficies necesarias para lograr moverse es de tres, en donde la técnica de movimiento cambia dependiendo cuántas patas mueve dejando las demás estáticas. Las marchas agrupan las patas en cada paso, separando las que se mueven y las que no, si se mantienen cinco, cuatro o tres sobre el suelo moviendo 1, 2 y 3 respectivamente (Cárdenas Verdugo, 2011). La relación de patas en movimiento y las que permanecen quietas se describe en la **Tabla 2**.

Tabla 2. Patas en contacto con el suelo y suspendidas en cada forma de desplazamiento.

Fuente: Cárdenas Verdugo, 2011

Técnica	Número mínimo de patas en contacto con el suelo	Número máximo de patas separadas del suelo
3 patas por movimiento	3	3
2 patas por movimiento	4	2
1 pata por movimiento	5	1

2.2.1.6. Clasificación de los hexápodos según su grado de libertad

Los robots hexápodos son cualquier clase de robots que posea 6 patas, diferenciándose unos de otros por la complejidad del diseño, en donde las capacidades de cada hexápodo se definen por los grados de libertad que permiten sus articulaciones (Cotera Berdugo, 2017). El aumento de grados de libertad implica el uso de más actuadores, lo que complica los sistemas en aspectos como la alimentación y mecanismos, junto con el programa de comando que coordina los movimientos (Cotera Berdugo, 2017). Dentro de la gama de robots hexápodos se encuentran:

- **Hexápodos de un grado de libertad:** Son hexápodos con articulaciones rígidas que solo logran un tipo de movimiento, estos son de construcción simple y no usan muchos actuadores, no requieren una programación compleja.
- **Hexápodos de dos grados de libertad:** Estos hexápodos requieren el uso de más actuadores lo que vuelve más complejo todo el sistema de control para coordinar los movimientos.
- **Hexápodos de más grados de libertad:** Estos implican un sistema de control avanzado ya que tratan de imitar el diseño mecánico de los insectos con todas sus capacidades motoras.

En la **Figura 12** se muestran algunos robots pertenecientes a cada clasificación por grados de libertad.

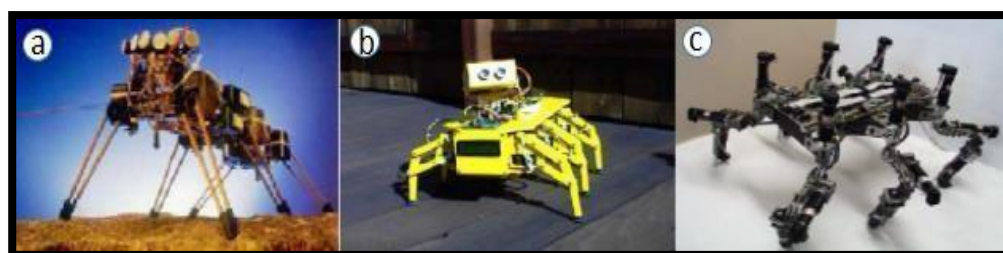


Figura 12. Hexápodos de a) 1 grado, b) 2 grados y c) más grados de libertad.

Fuente: Escobar Olarte & Silva Ortega, 2011

2.2.2. Robótica Modular

La robótica modular es la línea de la robótica dedicada a diseñar robots contruidos a partirde pequeños bloques, cada robot es independiente, con una construcción simple y geométrica que unido a otros robots iguales, pueden crean un robot más complejo y con mayores capacidades donde el número de módulos plantean posibilidades de diseño ilimitadas, (Cotera Berdugo, 2017).

Los robots modulares se hicieron con el fin de ser reconfigurables, surgiendo por primera vez en la década de los ochenta en el contexto de brazos manipuladores. La idea fue popularizadapor el trabajo de Mark Yim en Stanford a principios de los noventa, cuyo trabajo se extendió la idea de robots totalmente autónomos incluyendo su locomoción. Desde entonces, la comunidad que trabaja la robótica modular se ha centrado principalmente en robots homogéneos (los contruidos de muchas copias de un solo tipo de módulo), (Cotera Berdugo, 2017).

En 1995 Mark Yim propuso en su tesis doctoral un enfoque distinto al problema de locomoción presente en el diseño de robots, que en ese momento consistía en analizar el terreno ydiseñar un robot específico para dicho terreno. Pero en la práctica la topología de trabajo puede presentar variaciones, como podría ser pasar de un terreno plano a uno con imperfecciones para luego subir una escalera, por lo que se necesitarían tres robots distintos para llegar al punto deseado.Como solución a esto propuso el diseño de robots conformados por pequeños módulos, capaces deensamblarse uno con otros para formar distintas configuraciones. De esta forma al determinar que configuración es la más adecuada para cada terreno, estos solo necesitarían re ensamblarse para seguir su camino, como fue el caso del robot usado para llegar del edificio de robótica de Stanfordhasta el edificio cruzando la calle, en el que se usaron las configuraciones rueda, oruga y

cuadrúpedo para llegar a su destino (González Gómez, 2008, págs. 30-31).

2.2.2.1. Consideraciones

La complejidad del robot dependerá tanto de la movilidad que se desee tener en cada una de las extremidades como del número de estas. Cuantas menos patas tenga el robot menor será su estabilidad. Cuantos menos grados de libertad se tengan, menor será su movilidad.

Otro aspecto importante que debe tenerse en cuenta es la elección de los actuadores, los cuales, además de proporcionar la potencia para el movimiento de las piernas, deben de soportar el peso del robot.

Por último y no menos importante es el control de los movimientos. Hacer que el robot caminante se desplace consta de una serie de pasos sucesivos que se repite una y otra vez. El número de secuencias posibles depende del número de patas que tenga el robot pudiendo observarse en la **Tabla 3**. En dicha tabla se observa cómo crecen las combinaciones de secuencias conforme el robot pasa de tener una pata a tener diez.

Tabla 3. Resultados de las secuencias de Bessonov y Umnov.

Fuente: Cárdenas Verdugo, 2011

Número de patas	Número de secuencias	Número de secuencias que potencialmente presentan estabilidad estática
1	1	0
2	2	0
3	6	0
4	26	6
5	150	114
6	1,082	1,03
7	9,366	9,295
8	94,586	94,493
9	1'091,670	1'091,552
10	14'174,522	14'174,376

Para que una secuencia de movimientos presente potencialmente estabilidad estática es necesario que, por lo menos, haya más de tres patas en contacto con el suelo en todo momento. Es, por esto, por lo que en determinadas secuencias de movimientos

no se consigue dicha estabilidad. Esta idea cobrará gran importancia más adelante, al hablar de distintos modos de marcha en el robot hexápodo.

2.2.3. Cinemática Directa

El problema cinemático directo consiste en: “determinar la posición y orientación del extremo final del robot (efector final) con respecto al sistema base del mismo a partir de los valores de las articulaciones y los parámetros geométricos” (J. Ramírez, 2004).

2.2.4. Cinemática Inversa

Consiste en obtener los valores de las variables articulares para que el efector final de una cadena cinemática posea una determinada posición y orientación. (Evangelista, 2014).

2.2.5. Locomoción

un robot polípodo (múltiples articulaciones) la locomoción es el resultado de un movimiento coordinado de sus patas. Este movimiento se define por cierto paso que refleja determinadas especificaciones tales como velocidad, dirección, etc. Distingue dos fases claramente que contribuyen al movimiento de cada pata del robot: la fase de soporte y la fase de transferencia. Durante la fase de soporte o apoyo, cada pata debe ser capaz de ejercer cierta fuerza sobre la superficie en la que se encuentra el robot, de forma ordenada para proporcionar la fuerza necesaria al cuerpo del robot para así permitirle moverse según una trayectoria predeterminada. Después, durante la fase de transferencia o transición, la pata debe desplazarse de forma ordenada hacia el siguiente punto de soporte para reiniciar la secuencia de movimiento, (Cuaya, 2007).

2.3. Teoría básica

2.3.1. Matlab

MATLAB (abreviatura de MATrix LABoratory, «laboratorio de matrices») es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con

un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, macOS y GNU/Linux, (Goering, 2007).

El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets), (Goering, 2007).

2.3.2. Processing

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas, ambos miembros de Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda, Reas. (Casey; Fry, Ben ,2014).

Uno de los objetivos declarados de Processing es el de actuar como herramienta para que artistas, diseñadores visuales y miembros de otras comunidades ajenos al lenguaje de la programación, aprendieran las bases de la misma a través de una muestra gráfica instantánea y visual de la información. El lenguaje de Processing se basa en Java, aunque hace uso de una sintaxis simplificada y de un modelo de programación de gráficos. (Casey; Fry, Ben ,2014).



Figura 13. Interfaz Processing IDE.

Fuente: <https://processing.org/>

2.3.3. Arduino IDE

El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores, (Arduino announces FPGA board, 2018).

El código fuente para el IDE se publica bajo la Licencia Pública General de GNU, versión 2. El IDE de Arduino admite los lenguajes C y C ++ utilizando reglas especiales de estructuración de códigos.

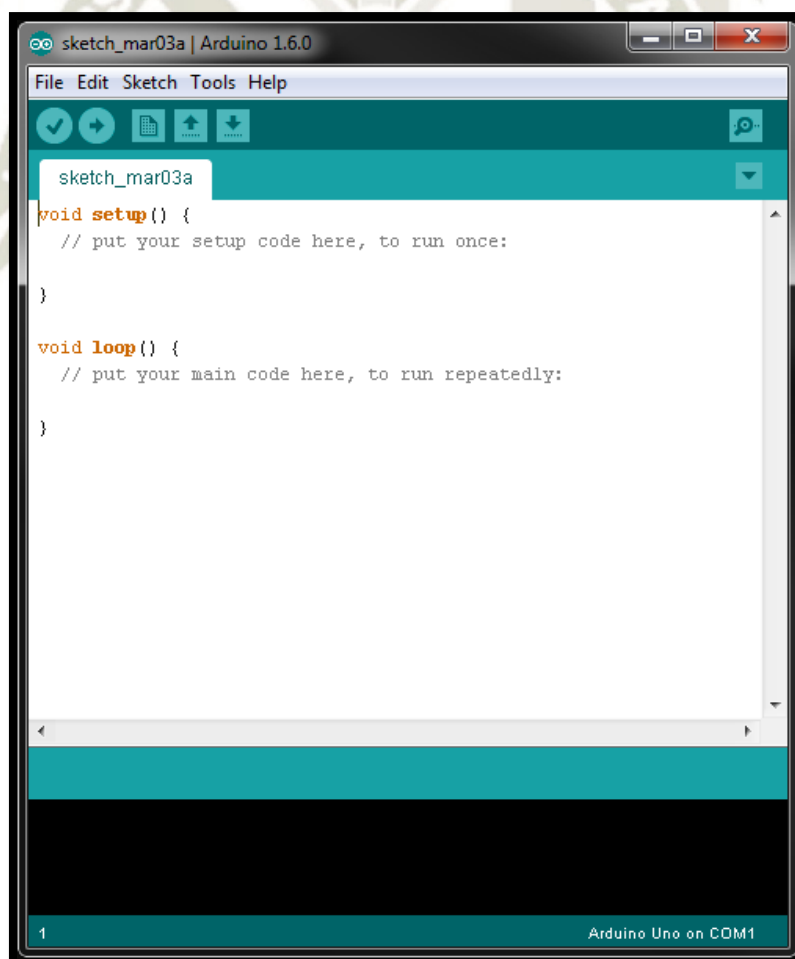


Figura 14. Interfaz Arduino IDE.

Fuente: <https://aprendiendoarduino.wordpress.com/category/ide/>

2.3.4. Atmega 2560

El microcontrolador de alto rendimiento y bajo consumo Microchip de 8 bits AVR® basado en RISC combina memoria flash ISP de 256 KB, SRAM de 8 KB, EEPROM de 4 KB, 86 líneas de E / S de propósito general, 32 registros de trabajo de propósito general, en tiempo real contador, seis temporizadores / contadores flexibles con modos de comparación, PWM, cuatro USART, interfaz en serie de dos cables orientada a bytes, convertidor A / D de 16 canales y 10 bits y una interfaz JTAG para depuración en chip. El dispositivo alcanza un rendimiento de 16 MIPS a 16 MHz y opera entre 4.5-5.5 voltios, se muestra las siguientes características en la **tabla 4**.

Tabla 4. Características de microcontrolador Atmega 2560

Fuente: Elaboración propia

Nombre	Valor
Tipo de memoria de programa	Destello
Tamaño de la memoria del programa (KB)	256
Velocidad de CPU (MIPS / DMIPS)	dieciséis
SRAM (B)	8.192
Datos EEPROM / HEF (bytes)	4096
Periféricos de comunicación digital	4-UART, 5-SPI, 1-I2C
Capturar / Comparar / Periféricos PWM	Captura de 4 entradas, 4 CCP, 16PWM
Temporizadores	2 x 8 bits, 4 x 16 bits
Numero de comparadores	1
Rango de temperatura, ° C)	-40 hasta 85
Seis / Doce canales PWM con resolución programable	2 a 16 bits
Rango de voltaje de funcionamiento (V)	1.8 hasta 5.5
Número de pines	100



Figura 15. Microcontrolador Atmega 2560.

Fuente: <https://www.microchip.com/wwwproducts/en/ATmega2560>

2.3.5. Solidworks

SOLIDWORKS es un software de diseño CAD 3D (diseño asistido por computadora) para modelar piezas y ensamblajes en 3D y planos en 2D. El software que ofrece un abanico de soluciones para cubrir los aspectos implicados en el proceso de desarrollo del producto. Sus productos ofrecen la posibilidad de crear, diseñar, simular, fabricar, publicar y gestionar los datos del proceso de diseño, (Solidworks Company History,2021).



Figura 16. Solidworks.

Fuente: <https://es.wikipedia.org/wiki/SolidWorks>

2.3.6. ISIS Proteus

2.3.6.1. Isis

Isis proteus es un programa de diseño electrónico que sirve para simular un circuito electrónico, tiene varias librerías de componentes electrónicos desde lo más básico hasta lo más complejo.

2.3.6.2. Ares

Ares es un adicional al programa de Isis proteus ya que nos permite elaborar un Board PCB, o placa impresa para nuestro circuito esto permite facilitar a la persona a no tener que utilizar otro software de diseño de PCB.



Figura 17. Proteus y diseño de PCB Ares.

Fuente:

[https://www.enerxia.net/portal/index.php?option=com_content&view=article&id=406:electronic a-proteus-simulador-digital-y-analogico&catid=61&Itemid=142](https://www.enerxia.net/portal/index.php?option=com_content&view=article&id=406:electronic-a-proteus-simulador-digital-y-analogico&catid=61&Itemid=142)

2.3.7. Impresión 3D

La impresión 3D, también conocida como manufactura por adición, es un proceso por el cual se crean objetos físicos colocando un material por capas en base a un modelo digital. Todos los procesos de impresión 3D requieren que el software, el hardware y los materiales trabajen en conjunto.

La impresión 3D es una tecnología increíble que se extiende a una gran variedad de industrias y usos. Estamos aquí para ayudarte a descubrir que es, cómo y cuándo se usa. Nosotros te mostraremos el impacto que tiene sobre la vida de las personas, sistemas de producción y los negocios, (jmgoluciones, 2018).



Figura 18. Impresoras 3d.

Fuente: <https://latinoamerica.autodesk.com/solutions/3d-printing>

2.3.8. Servo Tower pro mg995

Los servomotores son parte de un sistema de circuito cerrado y se componen de varias partes, a saber, un circuito de control, servomotor, eje, potenciómetro, engranajes de accionamiento, amplificador y un codificador o resolutor. Un servomotor es un dispositivo eléctrico autónomo que gira partes de una máquina con alta eficiencia y con gran precisión. El eje de salida de este motor se puede mover a un ángulo, posición y velocidad particulares que un motor normal no tiene. El servomotor utiliza un motor normal y lo acopla con un sensor para retroalimentación posicional. El controlador es la parte más importante del servomotor diseñado y utilizado específicamente para este propósito. (Salazar, 2020).

Características:

- Product Model: MG995
- Product Size: 40.7 * 19.7 * 42.9mm
- Product weight: 55g
- Operating torque: 15KG/cm
- Reaction speed: 53-62R / M
- Operating Temperature: -30 ~ +60 °
- Deadband settings: 4 microseconds
- Plug Type: JR, FUTABA General
- Rotation angle: A maximum of 180 degrees
- Servo: Servo
- Operating current: 100mA
- Using the voltage: 3-7.2V
- Materials of Construction: Copper metal teeth, coreless motor, double ball bearing

- No load operating speed: 0.17 seconds / 60 degrees (4.8V); 0.13 seconds / 60 degrees (6.0V)

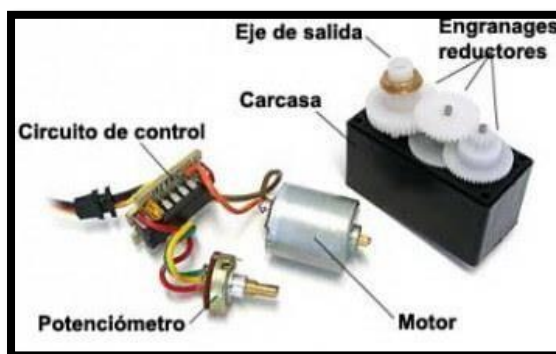


Figura 19. Servomotor Tower pro mg995.

Fuente: <https://clr.es/blog/es/servomotor-cuando-se-utiliza/>

2.3.9. Sensor de temperatura DHT22

El DHT22 es un sensor básico digital de medición de temperatura y humedad. Este sensor está basado en un termistor que sirve para medir el aire circundante (temperatura) e implementa un sensor interno capacitivo para la medición de humedad. DHT22 utiliza una técnica exclusiva de recolección de señal digital y tecnología de detección de humedad y puede superar la señal digital calibrada. El tamaño pequeño, el bajo consumo y la larga distancia de transmisión (20 metros) permiten que AM2303 se adapte a todo tipo de aplicaciones difíciles, (sparkfun, 2019) más detalle en el **Anexo 8**.

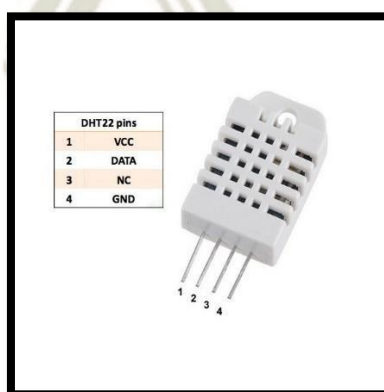


Figura 20. Sensor DHT22.

Fuente: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/58-sensor-de-temperatura-y-humedad-relativa-dht22-am2302.html>

2.3.10. Sensor de CO₂ o calidad de aire MQ-135

Se trata de un sensor electroquímico. Tiene la capacidad de variar su resistencia en función de las partes por millón de partículas de gas en el ambiente. Dispone de un calefactor interno que tiene la finalidad de mantener estable la temperatura y la humedad dentro del encapsulado, ya que estas afectan a la sensibilidad del sensor. Por este motivo, se recomienda usar el sensor en unas condiciones ambientales óptimas y tenerlo encendido unos minutos antes de usarlo. De esta forma, conseguiremos que el calefactor interno haga su trabajo, (iberotecno, 2021) más detalle en el **Anexo 9**.



Figura 21. Sensor MQ-135.

Fuente: <https://naylampmechatronics.com/sensores-gas/73-sensor-calidad-aire-mq135.html>

2.3.11. Lectora de Micro SD

Un lector SD es un dispositivo que permite emplear como almacenamiento una tarjeta SD, que podemos incorporar en nuestros proyectos de electrónica y Arduino. Las tarjetas SD y micro SD se han convertido en un estándar, desplazando a otros medios de almacenamiento de datos debido a su gran capacidad y pequeño tamaño. Por este motivo han sido integradas en una gran cantidad de dispositivos, siendo en la actualidad componentes frecuentes en ordenadores, tablets y smartphones, entre otros. (llamas, 2016)

Dentro del mundo de Arduino, es posible encontrar lectores de bajo coste tanto para tarjetas SD como micro SD. Los primeros en aparecer fueron los lectores SD y posteriormente los micro SD. Por tanto, en general, los módulos con micro SD son modelos más modernos que los de SD. En ambos tipos de lectores, la lectura puede realizarse a través de bus SPI. Aunque pueden disponer de otros interfaces, como bus I2C o UART, normalmente es preferible emplear SPI por su alta tasa de transferencia. (llamas, 2016).

Respecto a las tarjetas empleadas, podemos emplear tarjetas SD o SDSC (Standard Capacity) o SDHC (High Capacity), pero no SDXC (Extended Capacity). Deberá estar formateada en sistema de archivos FAT16 o FAT32, (llamas, 2016)

La tensión de alimentación es de 3.3V, pero en la mayoría de los módulos se incorpora la electrónica necesaria para conectarlo de forma sencilla a Arduino, lo que frecuentemente incluye un regulador de voltaje que permite alimentar directamente a 5V. (llamas, 2016)

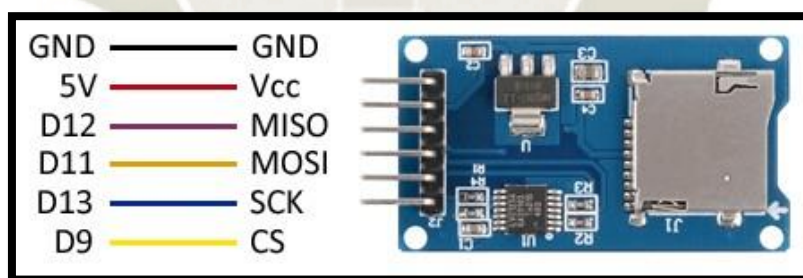


Figura 22. Lectora Micro SD.

Fuente: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

2.3.12. Módulo transceptor NRF24L01

El módulo transceptor NRF24L01. Utiliza la banda de 2,4 GHz y puede operar con velocidades de transmisión de 250 kbps hasta 2 Mbps. Si se usa en espacios abiertos y, con menor velocidad de transmisión, su alcance puede llegar hasta los 100 metros. Para

mayores distancias, hasta 1000 metros, existen módulos provistos con una antena externa en lugar de una antena trazada sobre la misma placa, como se observa en la **figura 23**.

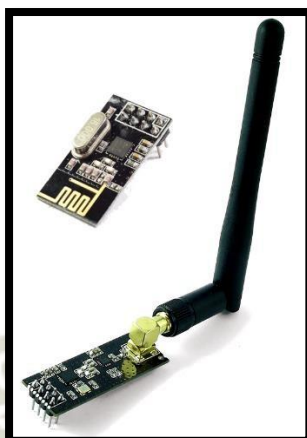


Figura 23. NRF24L01

Fuente: <http://robots-argentina.com.ar/didactica/arduino-comunicacion-inalambrica-con-nrf24l01/>

El módulo puede usar 125 canales diferentes, lo que da la posibilidad de tener una red de 125 módems que funcionen con independencia uno de otro en un solo lugar. Cada canal puede tener hasta 6 direcciones, es decir, cada unidad puede comunicarse con hasta otras 6 unidades al mismo tiempo. (Robots Didácticos, 2019).

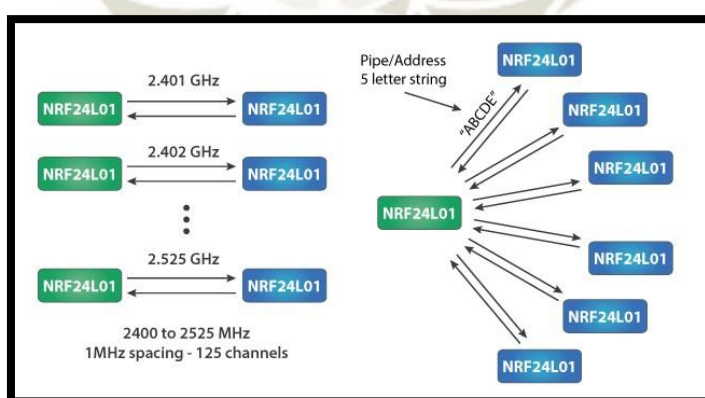


Figura 24. Canales de Onda

Fuente: <http://robots-argentina.com.ar/didactica/arduino-comunicacion-inalambrica-con-nrf24l01/>

El consumo de este módulo es de alrededor de 12 mA durante la transmisión, un valor menor al de un LED encendido. El voltaje de operación del módulo es de 1,9 a 3,6V, pero lo bueno es que los demás pines toleran la lógica de 5V, por lo que podemos conectarlo sin problemas a un Arduino sin necesidad de un convertidor de niveles lógicos, se muestra en la **Figura 25**. (Robots Didácticos, 2019).

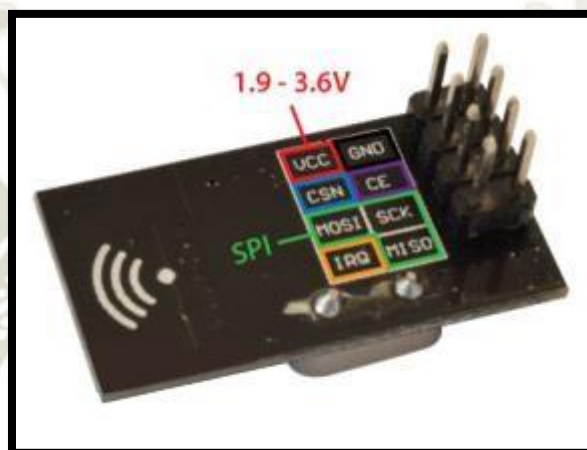


Figura 25. Modulo NRF24L01

Fuente: <http://robots-argentina.com.ar/didactica/arduino-comunicacion-inalambrica-con-nrf24l01/>

Tres de estos pines son para la comunicación SPI: MOSI, MISO y SCK, que deben conectarse a los pines de la interfaz SPI del Arduino. Se debe tener en cuenta que diferentes placas Arduino pueden tener los pines del interfaz SPI en diferentes posiciones. Los pines CSN y CE se pueden conectar a cualquier pin digital de la placa Arduino, y su función es configurar el módulo en modo de espera o activo, así como para alternar entre modo de transmisión o de comando. El último pin, IRQ, es un pin de interrupción que no es necesario utilizar. (Robots Didácticos, 2019).

Una vez conectados los módulos NRF24L01 a las placas Arduino, llegará el momento de escribir los programas para el transmisor, y para el receptor, se muestra en la **Figura 26**. (Robots Didácticos, 2019).

Conexionado para utilizar Arduinos Uno R3 en ambos lados de la comunicación:

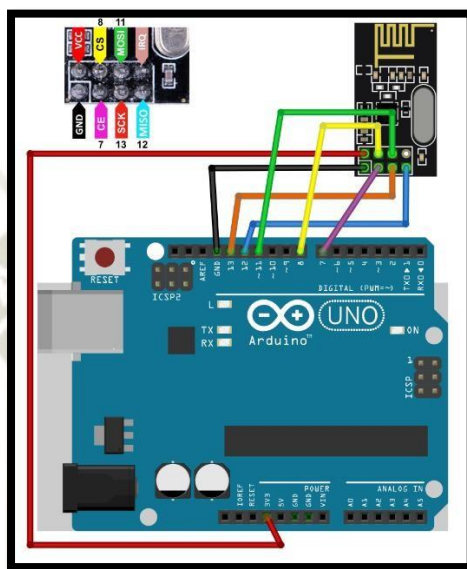


Figura 26. Conexión con Arduino.

Fuente: <http://robots-argentina.com.ar/didactica/arduino-comunicacion-inalambrica-con-nrf24l01/>

2.3.13. Processing entorno de desarrollo

Tiene su propio entorno de desarrollo. Cuando lo veas entenderás porque se dice que la plataforma Arduino se nutre de Processing. Se llama PDE (Processing Development Enviroment) desarrollado en Java. Es muy sencillo y fácil de usar, ya te he dicho que es una plataforma plug and play como Arduino. (Hernández, 2020)

Pero no solo tenemos PDE, si nuestro proyecto es complejo podemos utilizar otros entornos de desarrollo como Eclipse o Netbeans. Lo aconsejable es empezar por el propio

entorno de desarrollo y solo cuando tengamos la suficiente experiencia pasar a uno más complejo y potente, se muestra en la **figura 27**. (Hernández, 2020)

Desde el propio entorno de desarrollo podemos generar varios ejecutables en forma de Applet de Java, para ejecutar en navegador con las limitaciones de acceso a recursos del sistema o, como ya he comentado, ejecutables para diferentes plataformas.

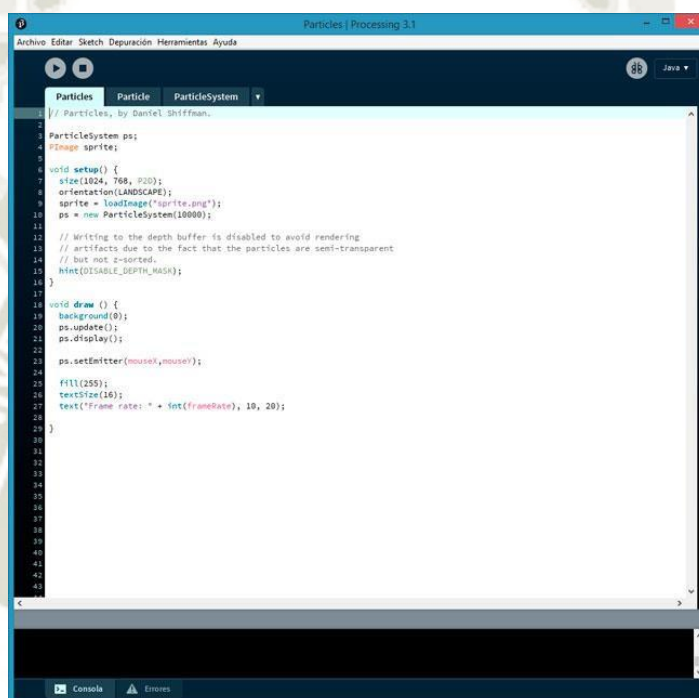


Figura 27. Entorno de Processing.

Fuente: <https://programarfacil.com/podcast/80-processing-lenguaje-para-graficos/>

Processing está basado en Java, más concreto en la versión 1.4.2. Esto no quiere decir que podamos utilizarlo con las versiones más actuales. Para ello debemos utilizar un entorno de desarrollo más potente como por ejemplo Eclipse. En este caso lo utilizaríamos como una librería gráfica que importamos a nuestro proyecto. (Hernández, 2020)

El proceso por el cual conseguimos un ejecutable te lo muestro en la **figura 28**.

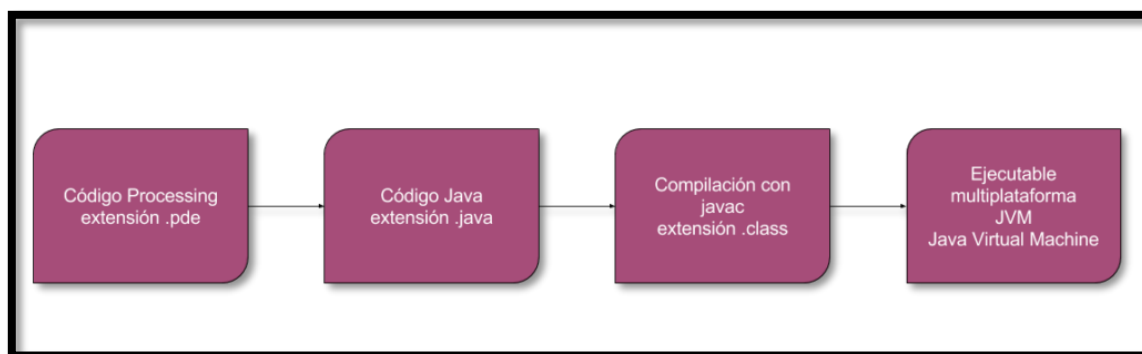


Figura 28. pasos para el ejecutable.

Fuente: <https://programarfacil.com/podcast/80-processing-lenguaje-para-graficos/>

Pero, aunque Processing esté basado en Java, encontramos diferencias entre estos dos lenguajes de programación. (Hernández, 2020)

- La sintaxis es más relajada, plataforma plug and play y sin preparativos.
- En el modo básico programamos sin funciones, sin clases, sólo líneas de código con variables globales. En Java no podemos.
- En el modo procedural al estilo de la programación en C, podemos definir nuestras propias funciones a las que llamamos.
- En el modo más avanzado, programación orientada a objetos, podemos crear clases de una forma muy sencilla que luego son transformadas e completas clases Java.

CAPÍTULO III

3. Diseño y Construcción del prototipo

3.1. Metodología de Diseño

Definir un proceso para llegar a un diseño completamente perfecto, en donde el cliente y el diseñador estén satisfechos es difícil o a veces hasta imposible llegar, sin embargo existen ciertos puntos en los que debemos de poner atención como diseñadores para que la interacción con el cliente sea más efectiva, eficaz y satisfactoria. (Olachea, 2013)

3.1.1. Norma VDI 2221

El modelo **VDI 2221** centra sus actividades en la búsqueda de soluciones, con el fin de obtener la información precisa para el desarrollo de un **diseño** eficaz que permita un prototipo que satisfaga en su totalidad las necesidades requeridas, (ARIAS, Ríos, Cortez, & Gabriela, 2017).

Con más detalle y los pasos que se tiene que seguir lo vemos en el anexo 5.

3.2. Alcances y limitaciones

En la **tabla 5** se muestra la lista de deseos y exigencias para nuestro diseño del robot Hexápodo.

Tabla 5. Lista de Deseos y exigencias.

Fuente: Elaboración propia basado en la norma VDI 2221

PROYECTO		DISEÑO DE ROBOT HEXAPODO CONTROLADO A DISTANCIA
Deseo o Exigencia		EXIGENCIA
		1. Función principal:
E		Desplazarse:
E		El robot debe poder ir hacia adelante, retroceder, girar (de esta forma puede ir hacia los costados)
E		
		2. Geometría:
D		El robot debe medir hasta 75 cm de largo, 75 cm de ancho y 25 cm de alto
		3. Cinemática:
E		Cada pata debe tener 3GDL
		4. Velocidad:
D		El robot debe avanzar hasta 0.077 cm/s
		5. Energía:
D		La fuente de alimentación para que el robot funcione debe durar mínimo hasta 30 min.
		6. Señales:
E		El robot debe poder recibir señales a distancia en un radio de hasta 2 metros
		7. Terreno de transporte:
E		El robot debe poder desplazarse en superficies planas
		8. Montaje:
E		El montaje de las patas debe ser por medio de articulaciones.
		9. Fabricación:
D		Debe poder fabricarse de manera fácil. Y el material del que esté hecho el robot debe poder soportar temperaturas de entre 0 hasta 40 grados.
		10. Mantenimiento:
D		Correctivo, para que cuando las piezas se dañen o presenten algún desperfecto sólo puedan ser reemplazadas por otras nuevas.
		11. Entrega de trabajo:
E		El robot no será construido sólo diseñado, para lo cual se entregará planos, análisis y simulación (sábado 06 de Julio)

3.3. Blackbox General

En la **figura 29** se muestra el blackbox general del robot hexapodo.



Figura 29. Blackbox general.

Fuente: Elaboración propia

3.4. Estructura De Funciones

3.4.1. Funciones

ARRANQUE: Un pulso eléctrico permite que la alimentación de una fuente de energía, energice a nuestro robot. Para luego este haga una secuencia de pasos. Para entender la locomoción debemos definir las patas y sus grados de libertad, así como la numeración de las mismas para así poder entender de una mejor manera la descripción del movimiento.

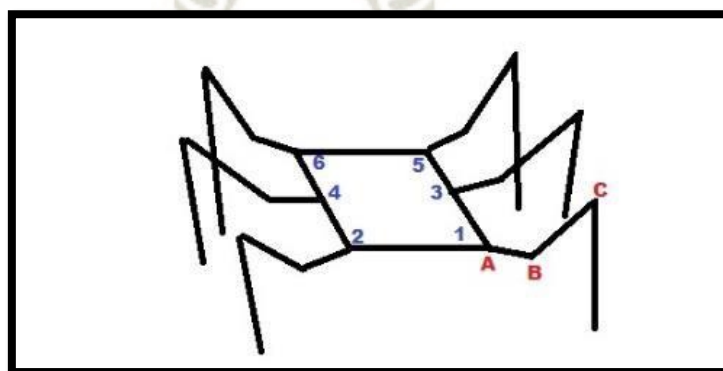


Figura 30. Bosquejo de robot y numeración de articulaciones.

Fuente: Elaboración propia

La articula A gira en torno al plano Z de manera horaria y anti horaria; las articulaciones B y C giran en torno al plano Y de manera horaria y anti horaria. Se plantea un cuadro de tiempos para entender la secuencia de movimientos

articulaciones: Este tipo de movimiento consiste en la posición inicial que debe adoptar el robot nos referimos a que estando en la posición que se encuentre debe de contraerse, es decir, debe de disminuir su altura en el eje Z.

PARTE MECÁNICA:

Cada una de las patas cuenta con 3 articulaciones (0, 1 y 2) como se ve en la figura siguiente, y de 3 segmentos o eslabones. Para esta función (Contraer) el segmento 0 (Hombro) debe realizar un movimiento linear en el eje Z dependiendo de la posición en que se encuentre hacia la posición inicial mostrada en la **Figura 31** para ello el segmento 1 debe moverse con respecto a la articulación 2 mientras que el segmento 0 se mantiene estable y si no lo está, el segmento 0 debe moverse alrededor de la articulación 1. Asimismo, el segmento 2 debe moverse alrededor de la articulación 2 hasta que se encuentre en 90° con respecto a la superficie plana.



Figura 31. Bosquejo de pata completa y sus articulaciones.

Fuente: Espinoza ,2014

3.4.1.1. Parte De Programación para contraerse

CONTRAERSE - GENERAL	
ART.	T1
1 A	Perpendicular al plano X
1 B	-
1 C	
2 A	Perpendicular al plano X
2 B	+
2 C	
3 A	Perpendicular al plano X
3 B	-
3 C	
4 A	Perpendicular al plano X
4 B	+
4 C	
5 A	Perpendicular al plano X
5 B	-
5 C	
6 A	Perpendicular al plano X
6 B	+
6 C	

Figura 32. Secuencia de movimiento para Contraerse - General.

Fuente: Elaboración propia

- Establecer comunicación: Necesitamos confirmar que se puede controlar al robot, para esto se necesitará de una señal que nos verifique la comunicación con el hexápodo, esta señal debe ser periódica y síncrona.
- Pararse:

PARTE MECANICA: Para poder pararse el brazo debe moverse en sentido antihorario a través del codo mientras que el hombro sólo se mantiene de forma lineal (solo de $-z$ a $+z$), al terminar el antebrazo debe moverse en sentido antihorario también a través del codo en pocos grados.

3.4.1.2. Parte De Programación para pararse

PARARSE - GENERAL	
ART.	T1
1 A	
1 B	+
1 C	
2 A	
2 B	-
2 C	
3 A	
3 B	+
3 C	
4 A	
4 B	-
4 C	
5 A	
5 B	+
5 C	
6 A	
6 B	-
6 C	

Figura 33. Secuencia de movimiento para Pararse - General.

Fuente: Elaboración propia

Estas sub funciones son repetitivas y genéricas pues el movimiento es en simultáneo

1. Avanzar: La función de avanzar, puede ser representada por diferentes secuencias según las patas y las articulaciones (dadas en la parte de programación)

PARTE MECANICA:

Se desarrolla un bucle. Antes y después de cada paso, el robot deberá encontrar en pie. Durante cada paso se hará que avance cierta distancia en la dirección especificada. El bucle se repetirá hasta que se le retire la orden de avanzar. El algoritmo que sigue en su avance es el trípode alternante, que se muestra a continuación a grandes rasgos:

- Las patas 1,4 y 5 se desplazan en forma parabólica hacia adelante. Las patas 2, 3 y 6 simultáneamente hacen un movimiento en la articulación A, en función del avance o retroceso del robot.

- Las patas 2, 3 y 6 se desplazan en forma parabólica hacia adelante. Las patas 1, 4 y 5 simultáneamente hacen un movimiento en la articulación A, en función del avance o retroceso del robot.

PARTE DE PROGRAMACION

En este caso utilizaremos la caminata tripod como se muestra en la **figura 34**.

ART.	AVANCE			TRIPODE								
	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
1 A	+	+			+	+			+	+		
1 B			-	+			-	+			-	+
1 C												
2 A			+	+			+	+			+	+
2 B	-				-	+			-	+		
2 C												
3 A			+	+			+	+			+	+
3 B	-	+			-	+			-	+		
3 C												
4 A	+	+	-	+	+	+	-	+	+	+	-	+
4 B												
4 C												
5 A	+	+	-	+	+	+	-	+	+	+	-	+
5 B												
5 C												
6 A			+	+	-	+	+	+	-	+	+	+
6 B	-	+			-	+			-	+		
6 C												

Figura 34. Secuencia de movimiento para Avanzar tipo TRIPOD

Fuente: Elaboración propia

3.5. Función Optima

TRIPODE: Este modo de marcha es el que presenta mayor velocidad, pero mayor inestabilidad, debido a que únicamente se sustenta sobre tres de sus patas.

Por tanto, su aplicación podría ser fundamentalmente en suelos muy uniformes.

La longitud de la carrera es la mitad de la longitud de zancada

El polígono de apoyo es un triángulo.

La marcha es periódica, simétrica y regular.

Tiempo de funcionamiento=1/2

3.6. Portadores De Funciones

Tabla 6. Tabla de portadores de funciones.

Fuente: Elaboración propia

FUNCIÓN PARCIAL	PRINCIPIOS TECNOLOGICOS				
	OPCIÓN 1	OPCIÓN 2	OPCIÓN 3	OPCIÓN 4	OPCIÓN 5
Encender controlador	Directo: Pulso (1) llega a controlador con su fuente de alimentación conectada todo el tiempo a este	Indirecto: Pulso (1) llega a un dispositivo que enciende la fuente de alimentación del controlador	Conexión de energía directa a controlador		
Mandar señal	Wifi	Rf	Bluetooth	Xbee	Comunicación serial
Procesar señal	Arduino Mega	Raspberry Pi	Arduino Uno	Circuitos integrados	
Articular	Eléctrico	Neumático	Hidráulico	Electromagnético	
Energizar robot	Batería química reusable	Batería química desechable	Paneles solares	Alimentación monofásica	Fuente de alimentación del controlador
Controlar señal	HMI	Control de mando inalámbrico	Control de mando alámbrico	Infrarrojo	

Caso 1: →

- Encender controlador: 2
- Mandar señal: 3
- Procesar señal: 2
- Articular: 1
- Energizar robot: 2
- Controlar señal: 1

El caso 1 de la **tabla 6**, el encendido del controlador será de forma directa, es decir, llegará un pulso (1) al controlador el cual ya estará alimentado por una fuente externa todo el tiempo. Entonces el cerebro del robot ya estará en encendido para recibir señales, las cuales serán mandadas por Bluetooth de un dispositivo (Controlador de señal) a otro dispositivo (Procesador de señales); las señales que recibirá de ahora en adelante son de mando para poder articular las patas lo cual se hará de forma eléctrica, la fuente de alimentación para energizar este movimiento (articular) será con una batería química

desechable. Y por último para comandar las señales se usará HMI (una interfaz hombre máquina). Entonces desde la interfaz se podrá elegir lo que queremos que haga el robot como avanzar retroceder o girar.

Caso 2:

- Encender controlador: 2
- Mandar señal: 1
- Procesar señal: 2
- Articular: 1
- Energizar robot: 2
- Controlar señal: 2

El caso 2 de la **tabla 6**, El encendido del controlador será de forma indirecta, es decir, llegará un pulso (1) al controlador el cual tendrá que recibir una señal para encender la fuente del controlador. Entonces el cerebro del robot ya estará en encendido para recibir señales, las cuales serán mandadas por wifi de un dispositivo (Controlador de señal) a otro dispositivo (Procesador de señales del microcontrolador); será alimentado por una batería química desechable, las señales que recibirá será de manera inalámbrica las cuales envaremos una señal para poder articular las patas del robot.

Caso 3:

- Encender controlador: 1
- Mandar señal: 5
- Procesar señal: 1
- Articular: 2
- Energizar robot: 4
- Controlar señal: 4

El caso 3 de la **tabla 6**, el encendido del controlador será de forma directa, es decir, llegará un pulso (1) al controlador el cual ya estará alimentado por una fuente externa todo el tiempo. Entonces el cerebro del robot ya estará en encendido para recibir señales, las cuales serán mandadas por comunicación serial, es decir cableada desde un microcontrolador externo. Las articulaciones del robot serán ejecutadas por actuadores neumáticos. La alimentación será externa directamente de una fuente de alimentación monofásica y controlada por un mando de control alámbrico.

Caso 4: →

- Encender controlador: 1
- Mandar señal: 1
- Procesar señal: 1
- Articular: 1
- Energizar robot: 1
- Controlar señal: 2

El caso 4 de la **tabla 6**, el encendido del controlador será de forma directa, es decir, llegará un pulso (1) al controlador el cual ya estará alimentado por una fuente externa todo el tiempo. Entonces el cerebro del robot ya estará en encendido para recibir señales, las cuales serán mandadas de forma inalámbrica de un dispositivo (Controlador de señal) a otro dispositivo (Procesador de señales del microcontrolador); las señales que recibirá serán de manera inalámbrica RF las cuales podemos enviar una señal para poder articular las patas del robot.

Caso 5: →

- Encender controlador: 2
- Mandar señal: 2

- Procesar señal: 1
- Articular: 1
- Energizar robot: 1
- Controlar señal: 2

El caso 5 de la **tabla 6**, El encendido del controlador será de forma directa, es decir, llegará un pulso (1) al controlador. El módulo de radio frecuencia será el medio por el cual se puedan dar las comunicaciones con el microprocesador que domina al robot, mediante un mando inalámbrico, el cual lo podrá mover y girar. La alimentación del robot se dará por medio de una batería química reusable. Las uniones articulables del robot son controladas por el microcontrolador por medio de actuadores eléctricos.

Caso 6:



- Encender controlador: 2
- Mandar señal: 2
- Procesar señal: 1
- Articular: 1
- Energizar robot: 1
- Controlar señal: 2

El caso 6 de la **tabla 6**, el encendido del controlador será de forma directa, es decir, llegará un pulso (1) al controlador. El módulo de WIFI será el medio por el cual se puedan dar las comunicaciones con el microcontrolador que domina al robot, mediante un mando inalámbrico, el cual lo podrá mover y girar. La alimentación del robot se dará por medio de una batería química reusable. Las uniones articulables del robot son controladas por el microcontrolador por medio de actuadores eléctricos.

3.7. Criterios De Selección

3.7.1. Criterio Técnico

En la **tabla 7** se muestra las ponderaciones del criterio técnico por cada caso con un máximo de ponderación de 5.

Tabla 7. Tabla ponderaciones – Criterio Técnico.

Fuente: Elaboración propia

CRITERIOS TÉCNICOS																
Nº	CRITERIO NOMBRE	PONDERACIÓN	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6	P*Caso 1	P*Caso 2	P*Caso 3	P*Caso 4	P*Caso 5	P*Caso 6	Max. De Ponderación	Max
1	Buen uso de fuerza y energía	4	5	3	5	4	4	5	20	12	20	16	16	20	5	20
2	Seguridad	2	5	4	4	5	5	5	10	8	8	10	10	10	5	10
3	Velocidad	2	2	2	2	2	2	2	4	4	4	4	4	4	5	10
4	Estabilidad	5	5	5	3	5	5	5	25	25	15	25	25	25	5	25
5	Rigidez	4	5	5	5	5	5	5	20	20	20	20	20	20	5	20
6	Manipulación	5	5	5	2	5	4	5	25	25	10	25	20	25	5	25
7	Facilidad de manejo	5	5	4	4	5	5	5	25	20	20	25	25	25	5	25
8	Transportabilidad	4	5	5	2	5	5	5	20	20	8	20	20	20	5	20
9	Calidad de trabajo	4	5	5	4	4	4	4	20	20	16	16	16	16	5	20
10	Complejidad	5	5	2	2	3	3	2	25	10	10	15	15	10	5	25
Sumatorias Totales									194	164	131	176	171	175	TOTAL MÁXIMO	200

3.7.2. Criterio Económico

En la **tabla 8** se muestra las ponderaciones del criterio económico por cada caso con un máximo de ponderación de 5.

Tabla 8. Tabla ponderaciones – Criterio Económico.

Fuente: Elaboración propia

CRITERIOS ECONÓMICOS																
Nº	CRITERIO NOMBRE	POND	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6	P*Caso 1	P*Caso 2	P*Caso 3	P*Caso 4	P*Caso 5	P*Caso 6	Max. De Ponderación	
1	Número de piezas	4	4	4	4	4	4	4	16	16	16	16	16	16	5	20
2	Facilidad de montaje	5	5	5	2	5	5	5	25	25	10	25	25	25	5	25
3	Productividad	5	5	5	4	4	5	5	25	25	20	20	25	25	5	25
4	Costos diversos	5	5	4	5	5	4	4	25	20	25	25	20	20	5	25
5	Pocos desperdicios	2	1	2	5	1	4	4	2	4	10	2	8	8	5	10
Sumatorias Totales									93	90	81	88	94	94	TOTAL MÁXIMO	105

Según los resultados, para este rubro la solución 4 se perfila como la mejor de todas con una ligera ventaja sobre la solución 1.

3.7.3. Concepto óptimo (Gráfica Técnico – Económico)

En la **tabla 9**, se presenta un resumen de los valores de X_i e Y_i para cada solución planteada, tanto para el criterio económico y el criterio técnico.

Tabla 9. Tabla ponderaciones – Criterio Técnico y Económico.

Fuente: Elaboración propia

Caso	EJE X (CRITERIO ECONOMICO)	EJE Y (CRITERIO TÉCNICO)
1	0.89	0.97
2	0.86	0.82
3	0.77	0.66
4	0.84	0.88
5	0.90	0.86
6	0.90	0.88

En la **figura 35** se muestra el criterio técnico y el criterio económico.

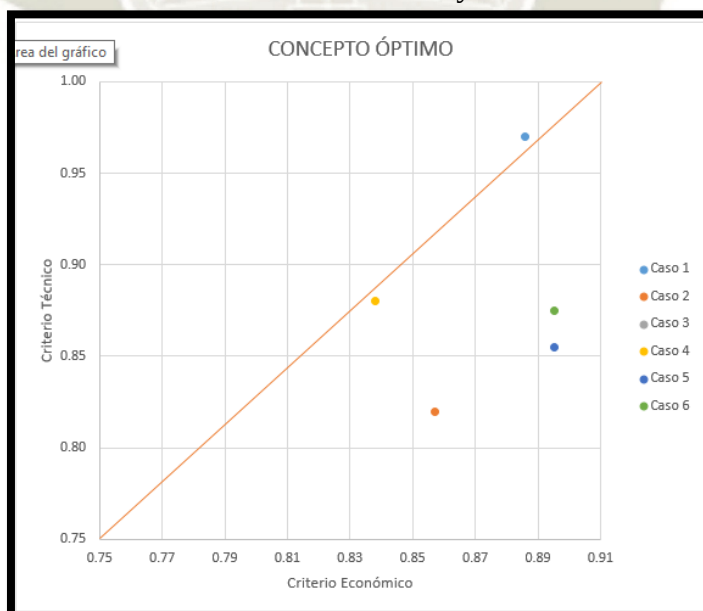


Figura 35. Criterio Técnico vs Criterio Económico.

Fuente: Excel

De acuerdo a las ponderaciones obtenidas según nuestros indicadores técnico-económicos,

vemos en la gráfica que la que más se acerca a la línea traza de 45° es el Concepto del caso planteado número 4 y la número 1, nos guiaremos por el precio ya que debe ser accesible, por lo tanto, se escoge la Solución 4 como la óptima para realizar el diseño de un Hexápodo.

3.8. Blackbox Detallada

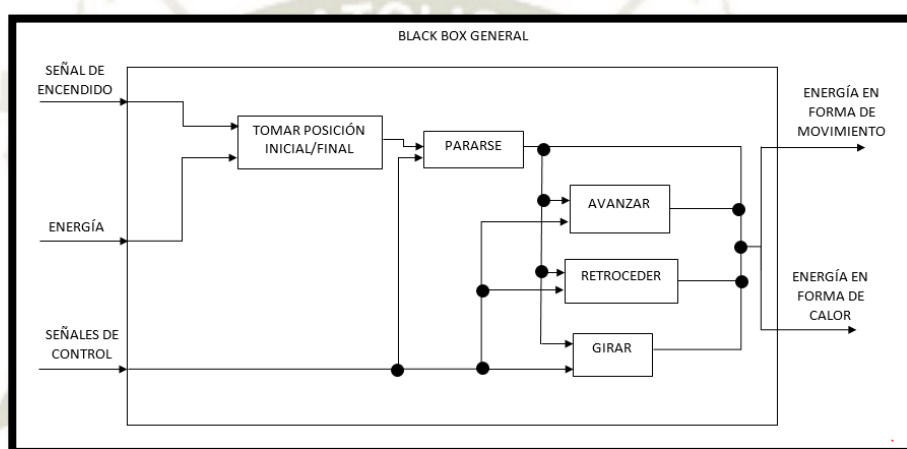


Figura 36. Blackbox.

Fuente: Elaboración propia

3.9. Diseño Preliminar

3.9.1. Elaboración De Bocetos

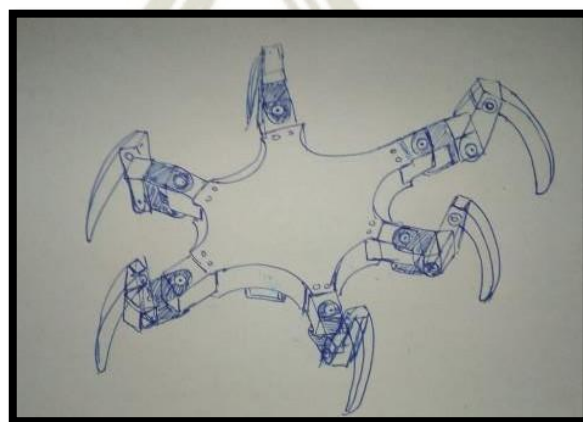


Figura 37. Boceto 1.

Fuente: Elaboración propia

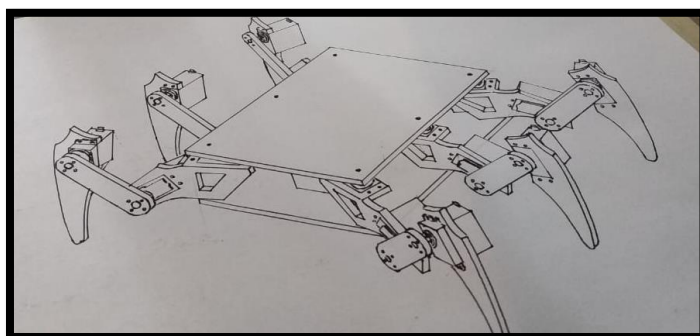


Figura 38. Boceto 2.

Fuente: Elaboración propia

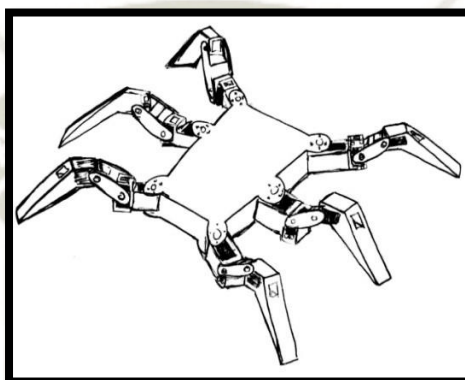
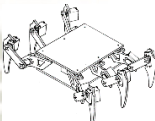


Figura 39. Boceto 3.

Fuente: Elaboración propia

Tabla 10. Tabla ponderaciones – BOCETOS.

Fuente: Elaboración propia

BOCETO	BOCETO 1	BOCETO 2	BOCETO3
			
Función	2	3	4
Forma	2	4	5
Diseño	3	3	3
Seguridad	3	3	3
Ergonomía	3	3	3
Fabricación	3	3	3
Montaje	2	4	4
Transporte	4	4	4
Uso	2	3	4
Costos	2	4	4
Plazos	2	3	3
TOTAL	28	37	40

De acuerdo a la **Tabla 10**, Se escogió el boceto 3 ya que presenta más puntuación en la tabla de ponderación por lo que se rige más a nuestros objetivos y lista de especificaciones; por lo que se convierte en nuestro diseño preliminar y está abierto a nuevos ajustes que se necesiten posteriormente.

3.10. Determinación De Características

3.10.1. Controlador

Se realizó las características de cada microcontrolador en el mercado y se muestra en la **tabla 11**.

Tabla 11. Tabla características de selección de microcontrolador.

Fuente: Elaboración propia

CARACTERÍSTICAS / ARDUINO	UNO	LEONARDO	NANO	MINI	MEGA
MICROCONTROLADOR	Atmega 328P	Atmega 32u4	Atmega 328P	Atmega 328P	Atmega 2560
VOLTAJE DE ENTRADA	7-12 V	7-12 V	7-12 V	7-9 V	7-12 V
VOLTAJE DE ENTRADA LIMITE	6-20V	6-20V	-	No más de 9 V	6-20V
PINES DIGITALES I/O	14	20	22	14	54
PINES ANALOGICOS	6	12	8	8	16
CORRIENTE DC PINES	20 mA	40 mA	40 mA	40 mA	20 mA
MEMORIA FLASH	32 KB	32 KB	32KB	32KB	256 kB
SRAM	2 KB	2.5 KB	2 KB	2 KB	8 KB
EEPROM	1 KB	1 KB	1 KB	1 KB	4 KB
PESO	25 g	20 g	7 g	-	37 g
TAMAÑO	68.6x53.4 mm	68.6x53.3 mm	18x45 mm	30x18 mm	101.52x53.3 mm
CONEXIÓN A COMPUTADORA	USB	USB	USB mini B	Adaptador USB	USB
COSTO	22	19.80	22	-	38.80
OTROS	*Compatible a adaptador AC/DC *Batería como F.A	* Botón de Reinicio *Comunicación USB *Encabezado ICSP	*Botón de Reinicio	*Versión compacta del NANO	*13 Led incorporados * Botón de Reinicio *Encabezado ICSP

Se escogió el Controlador Mega 2560 ya que este cuenta con la cantidad de patas disponibles para poder controlar los 18 servomotores en cuestión y tener además patas adicionales por si ocurrieran algún problema con las dispuestas. Existe una opción para poder controlar los 18 servo motores con los demás Arduinos con un módulo externo dedicado al control de servomotores (PCA9685) pero requiere mayor peso, volumen y consumo energético. Gracias al protocolo de comunicación síncrono I2C, el cual usa sólo dos cables, uno para el reloj (SCL) y otro para el dato (SDA), manteniendo así comunicación entre el maestro y esclavo por el mismo cable, el cual es controlado por el maestro que crea la señal de reloj, la cual se muestra en la **figura 40**.

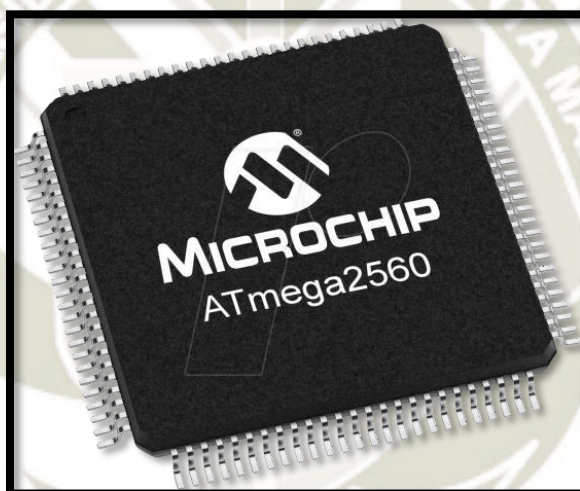


Figura 40. Atmega 2560.

Fuente: Geekbot electronics,s.f.

3.10.2. Selección De Actuadores

Luego de haber visto gran parte del diseño del robot y ya haber definido longitudes, material y masas es posible realizar un análisis más completo para la selección de los actuadores. Es sabido que se estableció con anterioridad el uso de servomotores como actuadores, pero escoger un servomotor en especial requiere de varios factores tales como torque, velocidad, precisión, tamaño, volumen, peso y funcionalidad; siendo el

torque el factor de más interés puesto a que este parámetro determina la fuerza de giro que se requiere para efectuar los movimientos del robot, (Lazo Pinto, 2018).

En el mercado actual los servomotores que predominan sobre los demás son los de corriente directa (DC) y esto se debe a que estos dispositivos de accionamiento rinden un mejor desempeño frente a otros dispositivos con convertidores de frecuencia, ya que estos no tienen un control óptimo respecto a control de posición y resultan poco efectivos a bajas velocidades, (Lazo Pinto, 2018).

Antes de entrar en consideración, es necesario especificar las características necesarias que mejor se adapten al robot Hexápodo. Primero es necesario que exista una realimentación continua de las posiciones alcanzadas por el servomotor, este también debe tener un tamaño mínimo que se adapte a los componentes del robot, debe contar con una buena resolución para un control más óptimo. Respecto al sistema de control, este debe tener una velocidad de transmisión de datos óptima, capaz de enviar todos los datos necesarios al controlador sin ningún problema, (Lazo Pinto, 2018).

Los servomotores deberán contar con una velocidad de operación lo suficientemente grande como para ejercer los movimientos del robot en el tiempo estimado y lo más importante deben de tener un torque lo suficientemente grande, capaz de ejercer todos los movimientos del robot sin dificultad ni problema alguno, (Lazo Pinto, 2018).

Partiendo de las consideraciones anteriores y luego de una extensa revisión por diferentes catálogos de fabricantes de servomotores, se decidió optar por los servomotores de la marca TowerPro. Se optó por usar los Servomotores TowerPro, porque: Poseen diferentes modos de control, siendo capaz de moveré libremente como una rueda si es necesario o modo articular limitando los movimientos de servo hasta donde sea necesario.

- Gozan de una alta resolución lo que genera gran precisión a la hora de ejercer los movimientos del robot.
- Posee gran torque y consta de una alta eficiencia.

Si bien es cierto que se escogió usar los servomotores TowerPro, aún no se especificó qué modelo se usara puesto que no se sabe el torque necesario para ejercer los movimientos del robot. Dicho esto, es necesario calcular los torques y esto se obtiene a partir del modelo dinámico.

Con el fin de obtener una correcta selección del actuador, se tiene que asegurar una velocidad y aceleración en cada movimiento del robot. Para tal efecto, se usará funciones sinusoidales como posiciones a alcanzar del servomotor, de esta manera se asegura las derivadas de la posición.

En la parte de modelamiento se determinó que el Torque máximo alcanzado por articulación es el siguiente: $r_1 = 0.7824 \text{ Kg/cm}$

$$r_2 = 10.9610 \text{ Kg/cm}$$

$$r_3 = 0.6700 \text{ Kg/cm}$$

Siendo el más crítico el mayor obtenido de 10.9610 Kg/cm.

A este valor le damos un factor de seguridad de 1.2, obteniendo un torque de 13.1532 Kg/cm

Partiendo de los resultados anteriores, se optó por usar el servomotor TowerPro MG995, porque:

- Posee un torque de funcionamiento de 15 Kg/cm.
- Posee una corriente de funcionamiento de 100 mA.

- Utiliza una tensión de funcionamiento entre 3-7.2 V.
- Las dimensiones van de acuerdo a nuestros requerimientos.
- Tiene un ángulo de rotación de 180 grados.

3.10.2.1. Cálculo De Selección Del Servo Motor

$$E_{SG90} = 6.0 \times 0.8 = 4.8 W \times 18 = 86.4 W \times 1 = 86.4 Wh$$

$$E_{MG90s} = 6.0 \times 0.7 = 4.2 W \times 18 = 75.6 W \times 1 = 75.6 Wh$$

$$E_{MG946R} = 6.0 \times 1.2 = 7.2 W \times 18 = 129.6 W \times 1 = 129.6 Wh$$

$$E_{MG995} = 6.0 \times 1.2 = 7.2 W \times 18 = 129.6 W \times 1 = 129.6 Wh$$

Basándose en los cálculos y torque necesario, en el desarrollo de nuestro trabajo necesitaremos un servomotor con un torque mínimo de 13 kg-cm, a lo que nuestros dos primeros servos quedan descalificados, teniendo en cuenta el precio y el parecido en las características técnicas del tercero y el cuarto, optaremos por el cuarto que nos ofrece un mayor torque.

Nuestro servomotor final será TowerPro MG995, con alto Torque, hasta 15Kg-cm. Robusto, alto rendimiento, compacto, con engranes de Metal, tamaño estándar, incluye accesorios y tornillos para montarlo en cualquier proyecto.



Figura 41. Servo motor tower pro MG995.

Fuente: naylampmechatronics.com

3.10.2.2. Selección del Servomotor

Se realizó las características de cada servomotor para la adquisición y se muestra en la **tabla 12**.

Tabla 12. Selección de servo motor.

Fuente: Elaboración propia

TIPOS DE SERVO Características importantes	MicroServo SG90	MicroServo MG90s	Servo TowerPro MG946R	Servo TowerPro MG995
Total, de Servos	18	18	18	18
Dimensiones	22*11.5*27 mm	22*11.5*27 mm	40.7*19.7*42.9 mm	40.7*19.7*42.9 mm
Par de trabajo	1.6Kgxcmm	2.5Kgxcmm	14Kgxcmm	15Kgxcmm
Voltaje Max de trabajo	6.0V	6.0V	6.0V	6.0V
Corriente Max de trabajo	800 mA	700mA	1200mA	1200mA
Precio (x18)	S/. 198.00	S/. 324.00	S/. 720.00	S/. 630.00
Peso (x18)	265 g	241.2 g	990 g	990 g
Energía consumida por hora de funcionamiento (x18)	86.4 Wh	75.6 Wh	129.6 Wh	129.6 Wh

3.10.3. Selección de Baterías

Se realizó las características de cada servomotor para la adquisición y se muestra en la **tabla 13**.

Tabla 13. Selección de las baterías.

Fuente: Elaboración propia

TIPOS DE BATERIAS	Ni/Cd	Ni/MH	Li-Ion	LiPO 2 celdas
Características importantes				
Voltaje	4.8 VDC	4.8 VDC	7.2VDC	7.4V
Dimensiones	30.2*54.1*30.2 mm	41 x 11 x 31 mm	84*49*13 mm	13x34x102mm
Peso	72.6 g	53.9 g	110 g	170 g
Capacidad de Carga	600 mAh	750 mAh	3100mah	3300mAh
Recargable	Si	Si	Si	Si
EcoFriendly	No	Si	SI	Si
Precio	Barato pero discontinuado	Barato pero discontinuado	Caro (70 soles)	Caro

Se escogió la batería de iones de litio de 2 celdas porque además de ser liviana, tiene muy buena densidad de carga y corriente nominal de trabajo, y entre sus ventajas esta que no tiene químicos contaminantes y que no es tan inflamable como las baterías de plomo, también que se pueden recargar más rápido sin dañar la estructura de la batería.

3.10.4. Selección De Comunicación Inalámbrica

Después de haber visto gran parte del diseño del robot y ya haber definido longitudes, material y masas es posible realizar un análisis más completo para la selección de los componentes (Lazo Pinto, 2018). Se hizo la selección de la comunicación inalámbrica, la que nos permitirá poder controlar el robot, como se muestra en la **Tabla 14**.

Tabla 14. Tabla de selección de comunicación inalámbrica.

Fuente: Elaboración propia

TIPOS DE Comunicación inalámbricas	Wi fi	Bluetooth	RF	Xbee	Comunicación serial
Características					
Frecuencia	2.4, 5 GHz	2.4 GHz	2.4 GHz	2.4 GHz	0.2 Hz
Transferencia de Datos	0.1-1 Mb/s	1,2,1,3 Mb/s	2 Mb/s	250 kb/s	19.2 kb/s
Alcance	indefinido	100m	100 - 1k m	100m	15 m
Energía consumida	medio	bajo	bajo	bajo	bajo
costo	medio	bajo	bajo	alto	medio

Tabla 15. Tabla de Características Robot hexápodo.

Fuente: Elaboración propia

TIPO	CARACTERISTICA	Descripción - Fundamento
Estructura	Material	Según la Tabla N°3 el que más nos conviene es el PLA ya que tiene una baja densidad (1.25) con respecto a la fibra de carbono, un límite de elasticidad de 8840 de lo cual no necesitamos un alto nivel, tiene un punto de fusión de 49° y lo que nos concierne más es su maquinabilidad que es MEDIA ya que usaremos una impresora 3D. Y PLA es más barato que ABS
	Fabricación	Haremos el uso de impresión 3D, ya que la forma de los componentes tiene más detalles (curvas, lugares de poco alcance - C)
	Acabado	El acabado con respecto a los que da una impresora 3D va de la mano con el costo total ya que a un acabado más fino y relleno más costoso; es por ello que se usará un 30%
	Puntas de las patas	Para evitar un desgaste y aumentar la fricción entre la superficie del suelo y la pata del robot hexápodo, se colocarán gomas antideslizantes.
Control	Cerebro	Microcontrolador MEGA 2560, según la Tabla N°27 escogimos este microcontrolador ya tiene las suficientes entradas y salidas que necesitamos, y un voltaje de entrada de 7-12 V que es con lo que pensamos trabajar. Aun siendo el más grande, no significa un impacto significativo en el diseño (101.52 mm x 53.3 mm), pesa 37g.
	Modulo Comunicación inalámbrica	ESP8266-01S, este módulo permite realizar una conexión Wifi que permite ser utilizado como un host o cliente. Incorpora una memoria caché que permite mejorar el rendimiento del sistema bajo tales aplicaciones. Incorpora protocolos de comunicación digital SPI/SDIO o I2C/UART. Incluye antena, switches, un amplificador de potencia, receptores de bajo ruido, módulos de administración de energía. Su diseño se pensó para ocupar un área de PCB mínima (14,30 mm x 24.80 mm). Alimentación de 3.3V.
	Conexionado	Se usarán cables de calibre 22 con terminales Dupont, siendo hembra o macho según se requiera.
Articulación	Motor	Según la tabla N°11, el servomotor TowerPro MG995 resalta frente a los otros por contar con un alto Torque, hasta 15Kg-cm, el cual, junto a su precio, lo hacen ser el idóneo para nuestro proyecto.
	Uniones	Este es Robusto, tiene alto rendimiento, compacto, cuenta con engranes de Meta, entre otros atributos.
Alimentación	Batería	Según la tabla 12, se escogió la batería de iones de litio de 2 celdas porque además de ser liviana, tiene muy buena densidad de carga y corriente nominal de trabajo, y entre sus ventajas esta que no tiene químicos contaminantes y que no es tan inflamable como la baterías lipo, también que se pueden recargar más rápido sin dañar la estructura de la batería.
	Conexionado	Placas PCB, y cables

Nota: se muestra los resultados obtenidos del diseño

3.11. Modelamiento Matemático Del Robot

3.11.1 Modelado del robot

En cada extremidad del robot hexápodo se realizará el análisis como un manipulador de 3 grados de libertad ya que tiene la misma configuración y los mismos grados de libertad, bastará con solo realizar el análisis en una de las extremidades del robot.

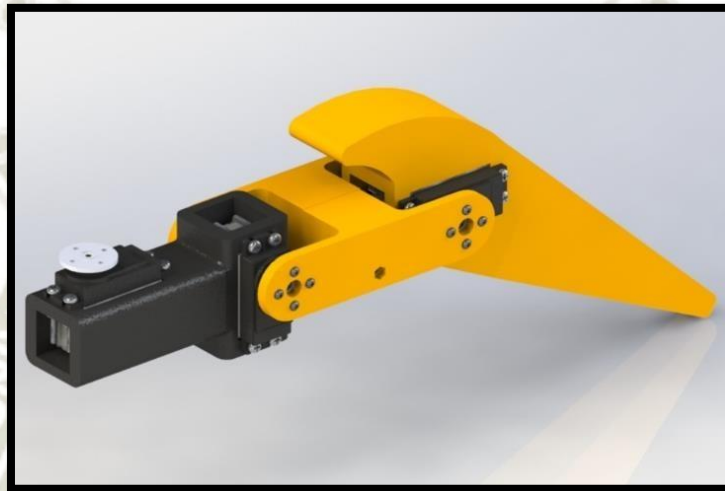


Figura 42. Pata Robot Hexápodo.

Fuente: Elaboración propia

3.11.2. Cinemática Directa

El problema cinemático directo consiste en determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot, (Craig J John, 2006).

El propósito de este procedimiento es calcular la posición en el efector final tomando en cuenta las variables angulares. (Q_1 , Q_2 , Q_3) y cada articulación es representada por los motores servo, se muestra en la **figura 43**.

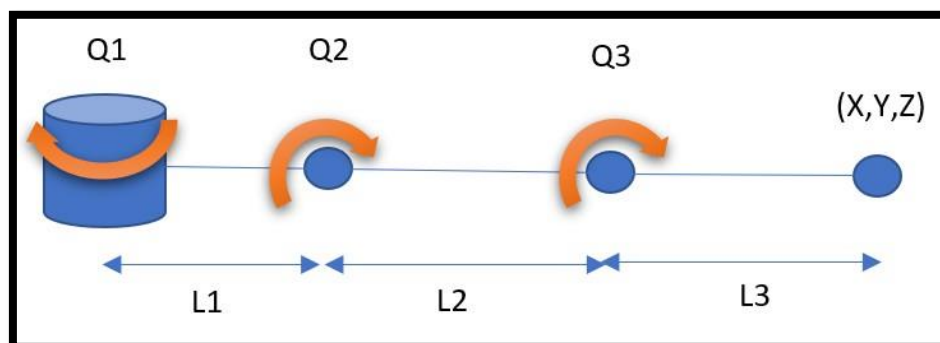


Figura 43. Posición cero Pata de robot.

Fuente: Elaboración propia

Con la base de conocer los procedimientos establecidos para realizar los parámetros de Denavit-Hartenberg inicial, se deberá obtener los valores para describir la estructura cinemática como se muestra en la **figura 44**.

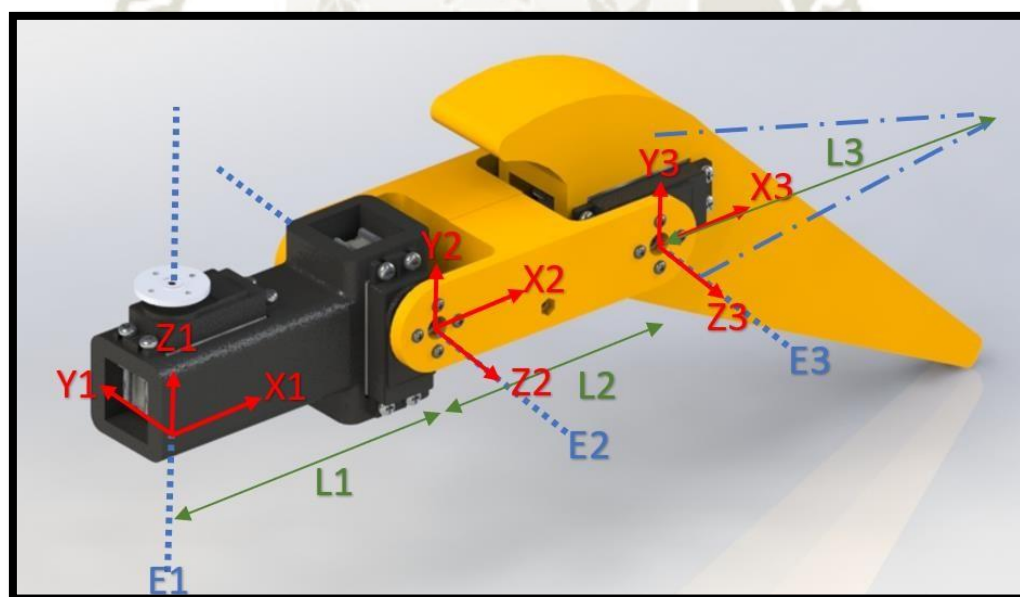


Figura 44. Posición creo primera elaboración.

Fuente: Elaboración propia

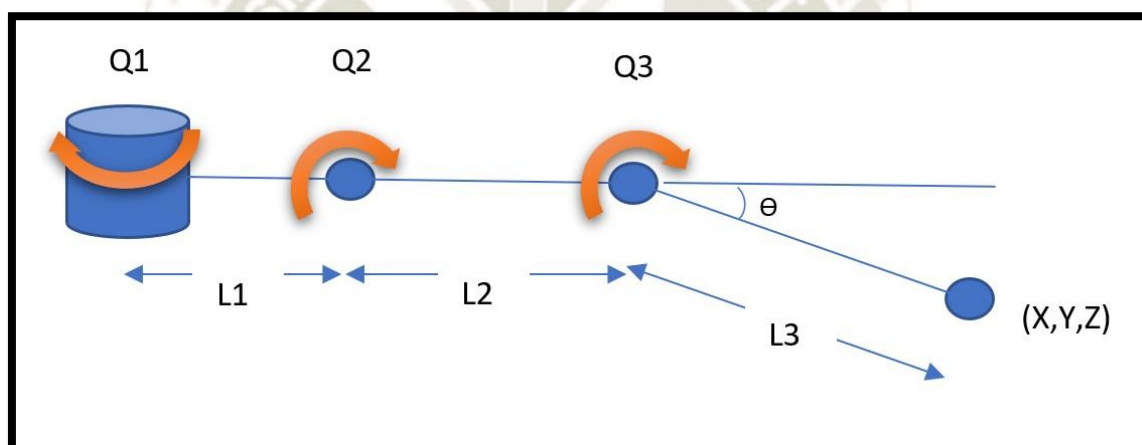
En vista de que empezamos con una configuración de que cada eslabón este en posición recta obtenemos los resultados en la **tabla 16**.

Tabla 16. Parámetros Denavit-Hartenberg para extremidades.

Fuente: Elaboración propia.

Articulación	θ_i	d_i	a_i	α_i
1	q_1	0	l_1	90
2	q_2	0	l_2	0
3	q_3	0	l_3	0

Viendo desde la parte del diseño ejecutando la primera configuración el efector final estaría en otro punto de coordenada lo cual no cumpliría con nuestro punto real, para esto hay un Angulo lo cual debemos hallar y proceder a restar a nuestro ángulo Θ_3 como se muestra en la **figura 45**.


Figura 45. ángulo de inclinación real.

Fuente: Elaboración propia

Para esto procedemos a hallar el ángulo Θ , sabiendo la distancia $L_3=18.3$ cm y la altura $H = 10.21468$ cm, lo cual se obtuvo con la medida con un vernier, se muestra en la **figura 46**.

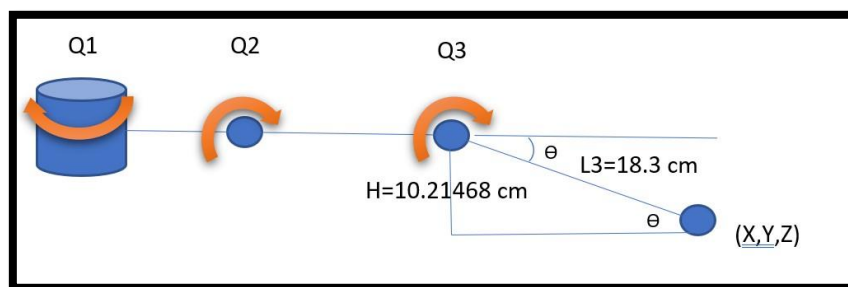


Figura 46. distancias halladas.

Fuente: Elaboración propia

Para poder hallar el ángulo Procedemos a hallar con la función seno.

$$\sin(\theta) = \frac{10.21468}{18.3}$$

$$\sin(\theta) = 0.55817 \text{ cm}$$

Luego procedemos a desarrollar el Arcosin del valor obtenido.

$$\theta = \arcsin(0.55817)$$

$$\theta = 33.93^\circ$$

Una vez obtenido el valor del angulo Θ que será negativa por que la orientación es hacia abajo, vemos en la imagen 47 y procedemos a desarrollar la nueva **tabla 17**.

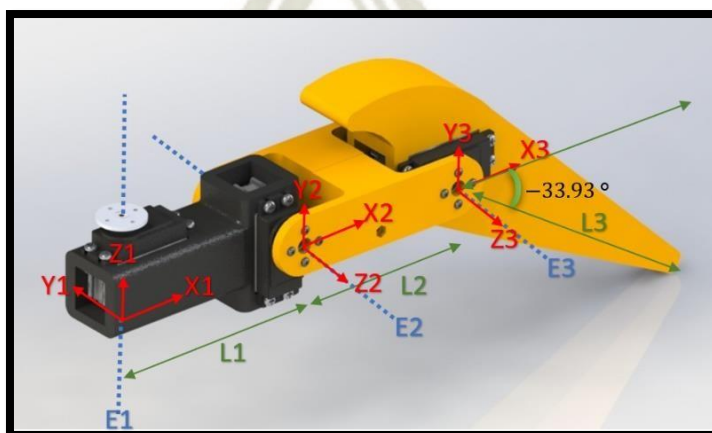


Figura 47. ángulo hallado.

Fuente: Elaboración propia

Tabla 17. Nuevos Parámetros Denavit-Hartenberg para extremidades.

Fuente: Elaboración propia.

Articulación	θ_i	d_i	a_i	α_i
1	q_1	0	l_1	90
2	q_2	0	l_2	0
3	$q_3 - 33.93$	0	l_3	0

Las transformaciones básicas para relacionar los sistemas i e $i - 1$, son cuatro:

- Rotación alrededor del eje Z_i un ángulo θ_i
- Traslación a lo largo de Z_i una distancia d_i ; vector $d_i(0,0,d_i)$
- Traslación a lo largo de X_i una distancia a_i ; vector $a_i(a_i,0,0)$
- Rotación alrededor del eje X_i , un ángulo α_i .

$${}^{i-1}A_i = Rot(Z_i, \theta_i) \cdot Tras(0,0,d_i) \cdot Tras(a_i,0,0) \cdot Rot(X_i, \alpha_i)$$

Con fines de abreviación se cambiará $c \theta_i$ y $s \theta_i$ para representar a $\cos \theta_i$

Y $\sin \theta_i$ respectivamente.

$${}^{i-1}A_i = \begin{bmatrix} c \theta_i & -s \theta_i & 0 & 0 & 1 & 0 & 0 & 0 \\ s \theta_i & c \theta_i & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha_i & 1 & 0 & 0 \\ 0 & c \alpha_i & -s \alpha_i & 0 \\ 0 & s \alpha_i & c \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}A_i = \begin{bmatrix} c \theta_i & -s \theta_i \cdot c \alpha_i & s \theta_i \cdot s \alpha_i & \alpha_i \cdot c \theta_i \\ s \theta_i & c \theta_i \cdot c \alpha_i & -c \theta_i \cdot s \alpha_i & \alpha_i \cdot s \theta_i \\ 0 & s \alpha_i & c \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De lo siguiente podemos obtener: Cambiaremos los θ_i a q_i .

$${}^0A_1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & L_1 \cos q_1 \\ \sin q_1 & 0 & -\cos q_1 & L_1 \sin q_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & L_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & L_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & L_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & L_3 \sin q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n = 3$$

$$T = {}^0A_n = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3$$

$$T = {}^0A_3 = \begin{bmatrix} C_{q_1} \cdot C(q_2 + q_3) & -C_{q_1} \cdot S(q_2 + q_3) & S_{q_1} & C_{q_1}(L_3 \cdot C(q_2 + q_3) + L_2 C_{q_2} + L_1) \\ S_{q_1} \cdot C(q_2 + q_3) & -S_{q_1} \cdot S(q_2 + q_3) & -C_{q_1} & S_{q_1}(L_3 \cdot C(q_2 + q_3) + L_2 C_{q_2} + L_1) \\ S(q_2 + q_3) & C(q_2 + q_3) & 0 & L_3 \cdot S(q_2 + q_3) + L_2 S_{q_2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De la matriz anterior, son de interés los factores que determinen la posición final de la extremidad.

Para poder ser exacto que nuestro efector final este en el punto dado por q_3 se le resta una variable $DES=33.93^\circ$ que representa el desfase angular de nuestro último eslabón.

$$X = L_1 \cos q_1 + L_2 \cos q_1 \cos q_2 + L_3 \cos q_1 \cos(q_2 + (q_3 - DES)) \quad (1.1)$$

$$Y = L_1 \sin q_1 + L_2 \sin q_1 \cos q_2 + L_3 \sin q_1 \cos(q_2 + (q_3 - DES)) \quad (1.2)$$

$$Z = L_2 \sin q_2 + L_3 \sin(q_2 + (q_3 - DES)) \quad (1.3)$$

Comprobación de dichas fórmulas en Matlab véase en el **Anexo 2**.

3.11.3. Cinemática Inversa

De la figura 48 esta vista superior, hallamos la tangente de q_1 .

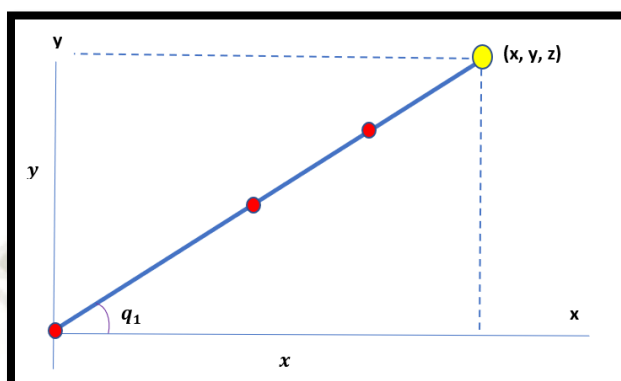


Figura 48. representación gráfica vista superior.

Fuente: Elaboración propia

$$\tan(q_1) = \frac{y}{x}$$

Despejamos q_1 .

$$q_1 = \arctan\left(\frac{y}{x}\right)$$

De la figura 49 de vista superior, hallamos por Pitágoras la distancia L.

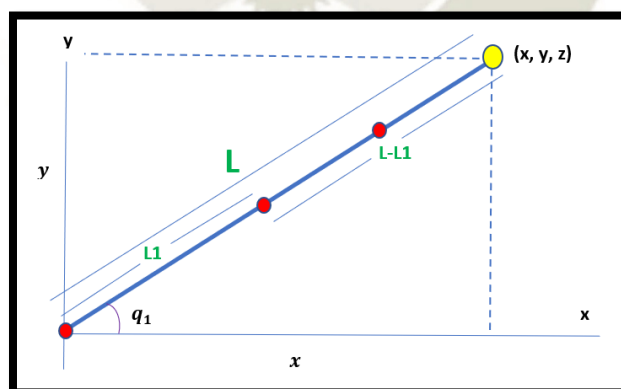


Figura 49. Determinación de variables.

Fuente: Elaboración propia

$$L = \sqrt{x^2 + y^2} \quad (1.4)$$

De la **figura 50** de vista frontal hallamos por Pitágoras la distancia ML.

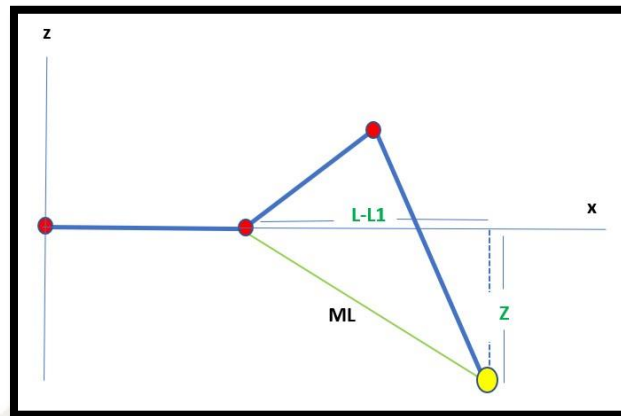


Figura 50. vista frontal pata de robot.

Fuente: Elaboración propia

$$ML = \sqrt{(L - L1)^2 + z^2} \quad (1.5)$$

De la figura 51 de vista frontal, hallamos la tangente del ángulo a_1

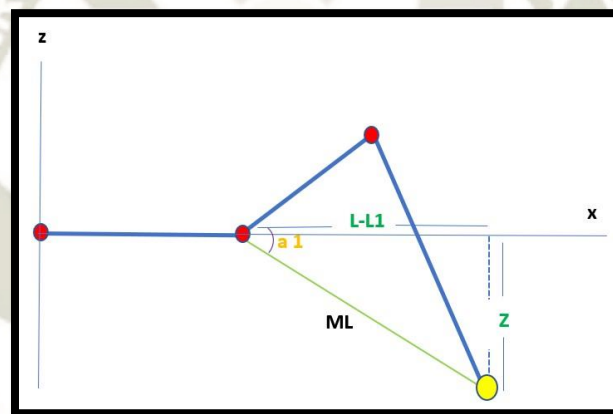


Figura 51. vista frontal pata de robot, variable a_1 .

Fuente: Elaboración propia

$$a_1 = \arctan\left(\frac{z}{L-L1}\right) \quad (1.6)$$

De la figura 52 de vista frontal hallamos por ley de cosenos el ángulo a_2 y despejamos el ángulo a_2 .

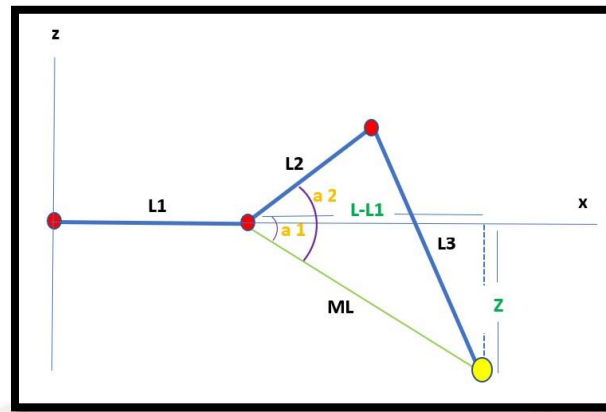


Figura 52. vista frontal pata de robot, variable a_2 .

Fuente: Elaboración propia

$$a_2 = \arccos\left(\frac{L_3^2 - L_2^2 - ML^2}{-2 * L_2 * ML}\right) \quad (1.7)$$

Reemplazando las ecuaciones (1.6) (1.7) obtenemos la siguiente ecuación para

q_2 bajo dos condiciones.

Si $Z \geq 0$ cuando el z es positivo

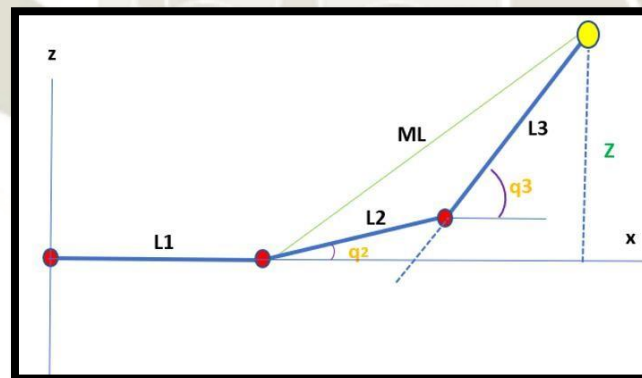


Figura 53. posición cualquiera z positivo q_2 .

Fuente: Elaboración propia

$$q_2 = \arctan\left(\frac{z}{L - L_1}\right) - \arccos\left(\frac{L_3^2 - L_2^2 - ML^2}{-2 * L_2 * ML}\right)$$

Si $Z < 0$

$$q_2 = \arctan\left(\frac{z}{L - L_1}\right) + \arccos\left(\frac{L_3^2 - L_2^2 - ML^2}{-2 * L_2 * ML}\right)$$

De la **figura 54** de vista frontal, hallamos por ley de cosenos el ángulo b_1 y despejamos el ángulo b_1 con la resta del desfase angular hallado anteriormente “DES”.

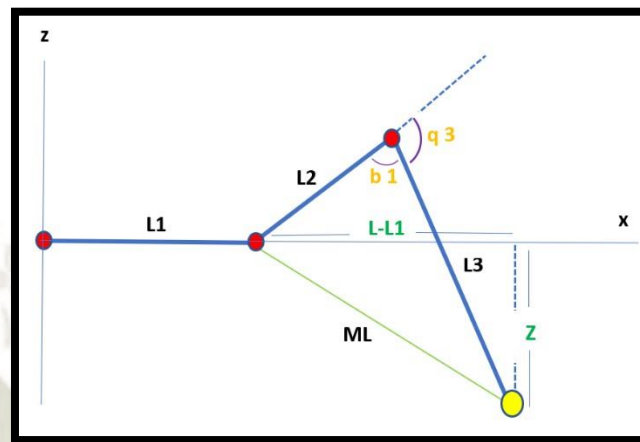


Figura 54. vista frontal pata de robot, variable b_1 .

Fuente: Elaboración propia

$$b_1 = \arccos\left(\frac{ML^2 - L2^2 - L3^2}{-2 * L2 * L3}\right) \quad (1.8)$$

$$q_3 = (180 - b_1) - DES$$

Reemplazando las ecuaciones (1.8) obtenemos la siguiente ecuación para q_3 bajo dos condiciones.

Si $Z \geq 0$ cuando el z es positivo

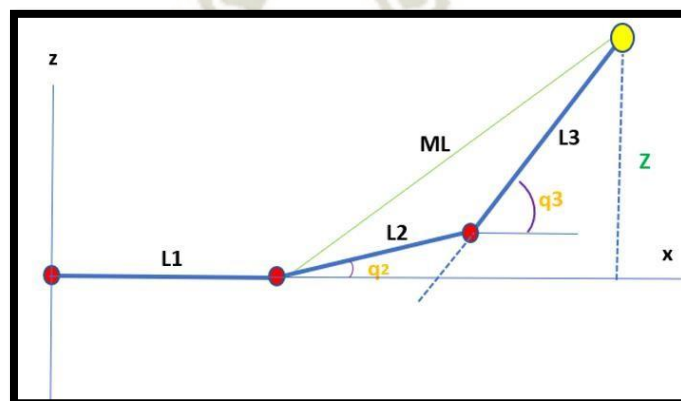


Figura 55. posición cualquiera z positivo para q_3 .

Fuente: Elaboración propia

$$q_3 = (180 - \arccos(\frac{ML^2 - L2^2 - L3^2}{-2 * L2 * L3})) - DES$$

$$q_3 = (b1 - 180) - DES$$

Si $Z < 0$

$$q_3 = (\arccos(\frac{ML^2 - L2^2 - L3^2}{-2 * L2 * L3}) - 180) + DES$$

Comprobación del cálculo matemático y aplicación de las fórmulas en el Anexo 2.

3.11.4. Cinemática del cuerpo

3.11.4.1. Operador de Traslación

Una traslación mueve un punto en el espacio una distancia finita en dirección de un vector dado. Con esta interpretación de lo que es una traslación lo único que se necesita para realizarla es un sistema de coordenadas en base al cual se llevará a cabo la transformación y un vector que indique tanto la dirección como la magnitud de la misma. En la **Figura 56** se puede ver gráficamente como el vector u_1 es trasladado por el vector v .

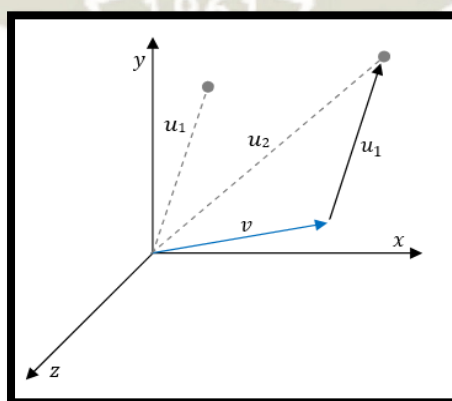


Figura 56. Traslación del vector u_1 .

Fuente: Barrientos (2014).

El resultado de la operación es un nuevo vector u_2 , que se calcula como

$$u_2 = u_1 + v$$

Esta misma operación puede realizarse con un operador matricial

$$u_2 = D(v)u_1$$

Donde v es el vector que indica la magnitud y dirección de la traslación, y

D el operador detraslación

$$D(v) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} V_x \\ V_y \\ V_z \\ 1 \end{matrix}$$

Donde v_x , v_y y v_z son los componentes del vector v en el sistema coordenado.

3.11.4.2. Operador de rotación

La rotación de un punto en el espacio puede verse como un movimiento angular sobre un eje. Lo que se necesita para poder llevar a cabo dicha transformación es la dirección del eje de rotación y la magnitud angular de la propia rotación.

En la **Figura 57** se puede ver gráficamente como el vector u_1 es rotado alrededor del vector v , que en este caso en particular tiene la misma dirección del vector z del sistema de coordenadas.

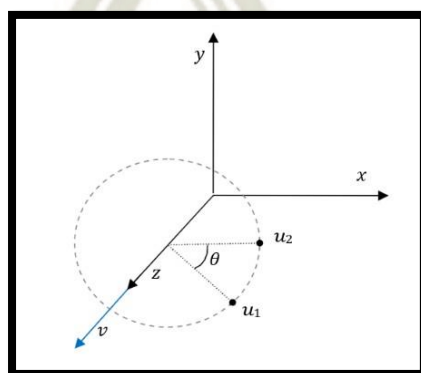


Figura 57. Rotación del vector u_1 .

Fuente: Barrientos (2014).

El resultado de la rotación es un nuevo vector u_2 que se calcula como

$$u_2 = Rv(\theta) u_1$$

Donde R_v es el operador de rotación aplicado al vector u_1 , y θ es la magnitud de la rotación. Las rotaciones más utilizadas son las realizadas sobre los ejes coordenados.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Una vez que se tiene la cinemática inversa de cada pata del robot el control del robot se hace teniendo los puntos que se alcanzan con cada pata en la superficie.

Primero se tiene que poner en posición de caminata lo cual no tiene traslaciones ni rotaciones como se muestra en la **figura 58**.

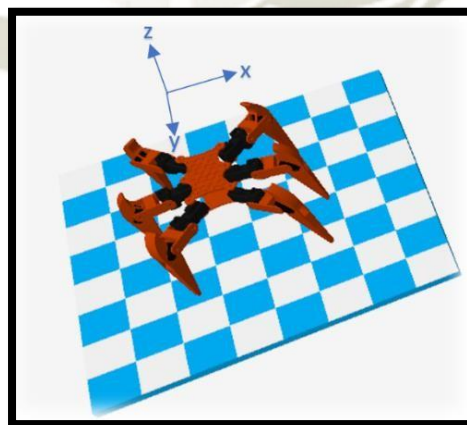


Figura 58. Posición de Caminata.

Fuente: Elaboración propia.

Transformación (traslación en x) a los puntos finales de cada pata, se muestra en la **figura 59**.

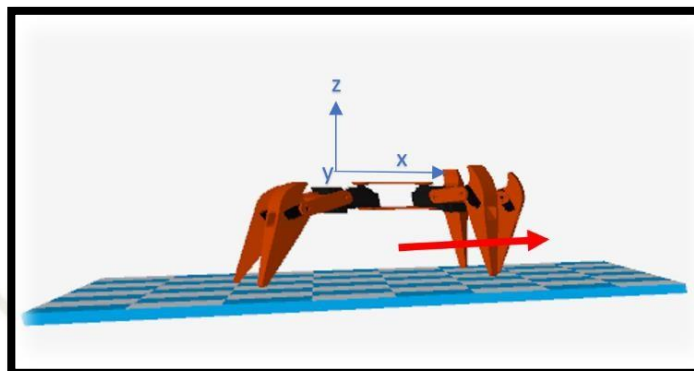


Figura 59. traslación en x .

Fuente: Elaboración propia.

Transformación (traslación en Y) a los puntos finales de cada pata, se muestra en la **figura 60**.

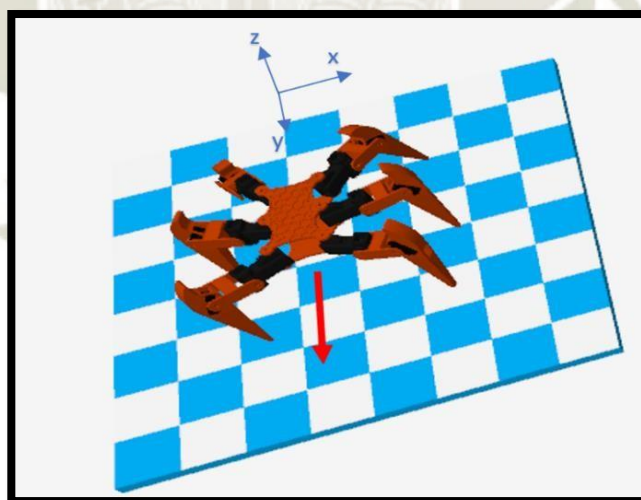


Figura 60. traslación en y .

Fuente: Elaboración propia.

Transformación (traslación en z) a los puntos finales de cada pata, se muestra en la **figura 61**.

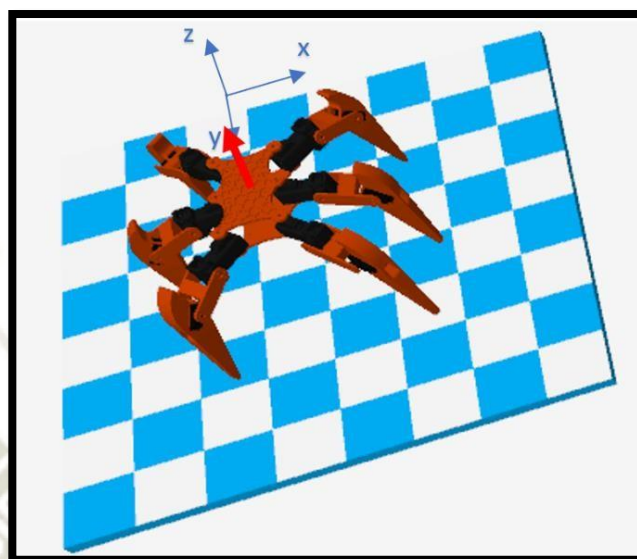


Figura 61. traslación en z.

Fuente: Elaboración propia.

Se aplica una transformación (Rotación en x) a los puntos finales de cada pata, se muestra en la **figura 62**.

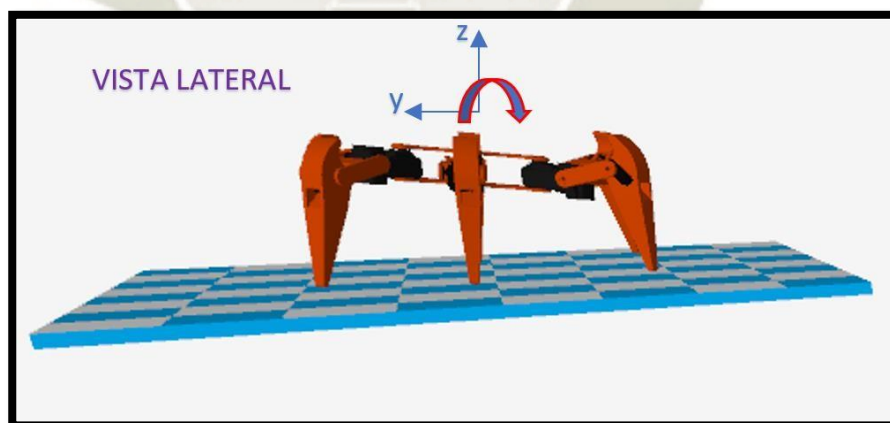


Figura 62. Rotación en x.

Fuente: Elaboración propia.

Se aplica una transformación (Rotación en Y) a los puntos finales de cada pata,
se muestra en la **figura 63**.

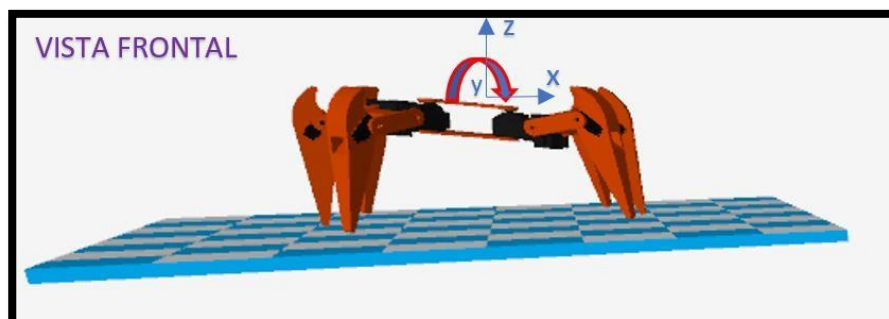


Figura 63. Rotación en y.

Fuente: Elaboración propia.

Se aplica una transformación (Rotación en Z) a los puntos finales de cada pata,
se muestra en la **figura 64**.

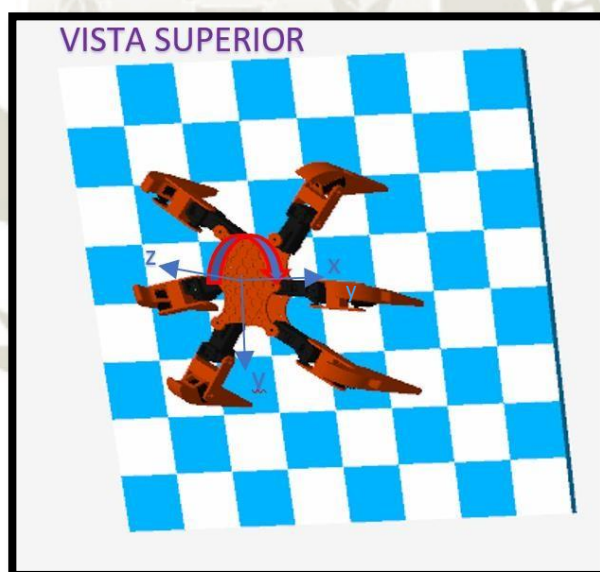


Figura 64. Rotación en z.

Fuente: Elaboración propia.

Comprobación y aplicación del código en proccesing en el **Anexo 3**.

También se halló el espacio de trabajo de una pata con los siguientes parámetros:

`%TIEMPO PARA COMPLETAR MOVIMIENTO`

`h_1=3;`

```
k=0;  
muestreo=0.1;  
tiempo=3;  
% PARAMETROS DE TRAYECTORIA PARABÓLICA  
y_mov = 5;  
z_mov = 8;
```

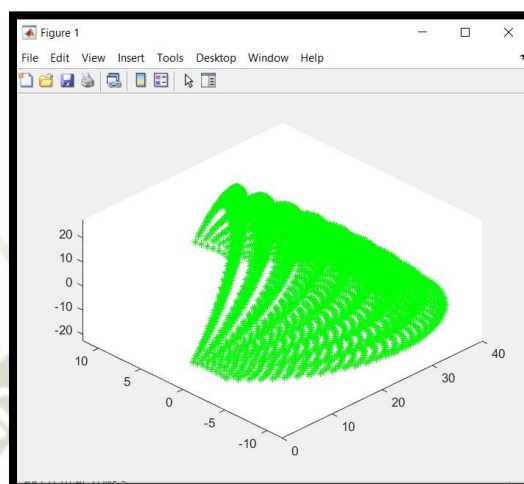


Figura 65. Volumen de Trabajo.

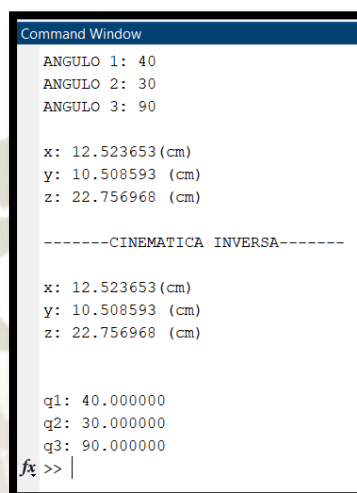
Fuente: Matlab 2020a.

CAPÍTULO IV

4. Pruebas y Resultados

4.1. Comprobación en Matlab

En el **Anexo 2** se muestra el código en Matlab la comprobación e ingreso de la fórmula de cinemática directa e inversa, resultados obtenidos en la **figura 66**.



```
Command Window
ANGULO 1: 40
ANGULO 2: 30
ANGULO 3: 90

x: 12.523653 (cm)
y: 10.508593 (cm)
z: 22.756968 (cm)

-----CINEMATICA INVERSA-----

x: 12.523653 (cm)
y: 10.508593 (cm)
z: 22.756968 (cm)

q1: 40.000000
q2: 30.000000
q3: 90.000000
fx >> |
```

Figura 66. Resultados de Cinemática directa e inversa.

Fuente: Matlab 2020a.

En la **figura 67** nos muestra la variación de ángulos de cada articulación q_1 , q_2 , q_3 .

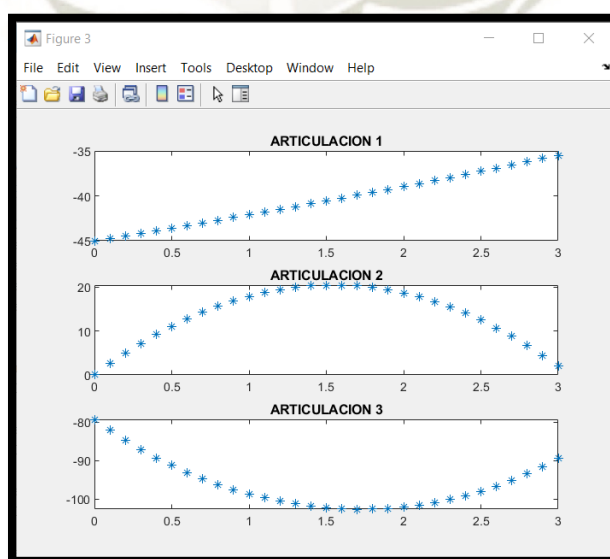


Figura 67. Valores Articulares.

Fuente: Matlab 2020a.

En la **figura 68** nos muestra la variación de posición en cada articulación coordenadas x, y, z.

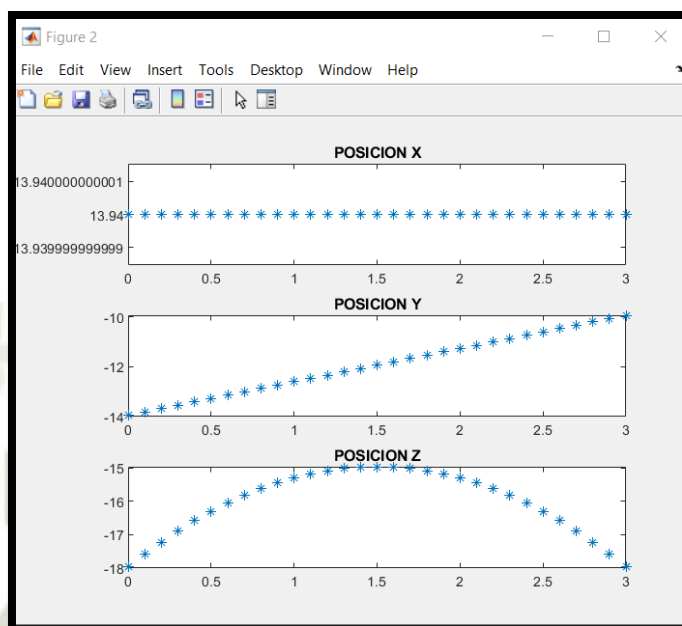


Figura 68. valores de posición en x, y, z.

Fuente: Matlab 2020a.

En la **figura 69** nos muestra la simulación de una pata del robot con librería de Peter corke en Matlab.

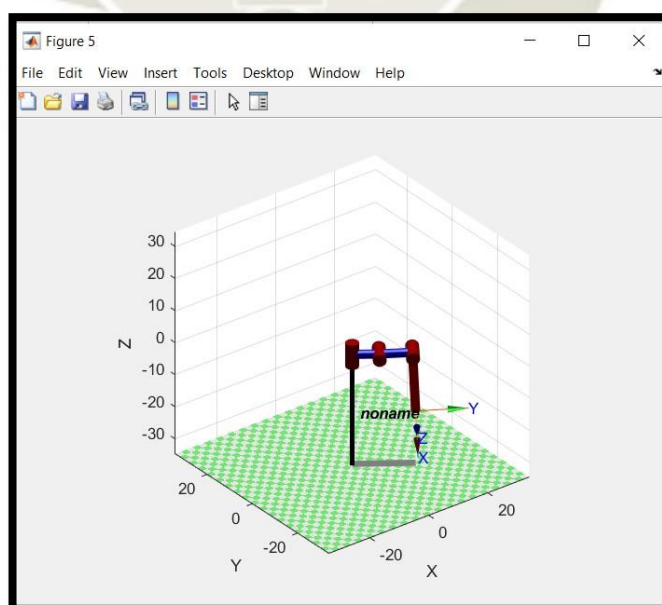


Figura 69. simulación peter Corke.

Fuente: Matlab 2020a.

En la **figura 70** se muestra la trayectoria parabólica generada para una pata del robot.

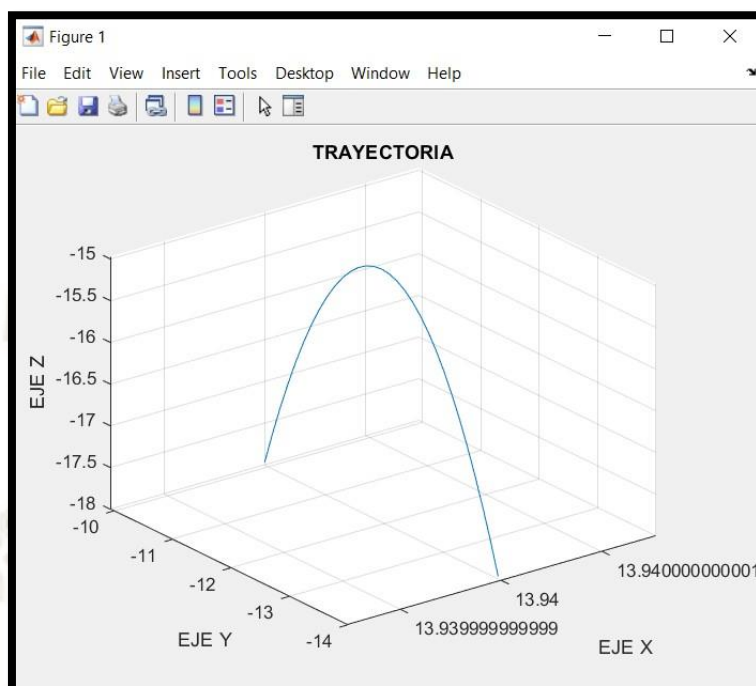


Figura 70. Trayectoria Parabólica.

Fuente: Matlab 2020a.

4.2. Análisis De Esfuerzo

El objetivo principal de este análisis se centra en estudiar el comportamiento de los esfuerzos que actúan sobre los componentes y la deformación plástica obtenidos de aplicar diversas cargas en cada componente, (Lazo Pinto, 2018).



Figura 71. Robot Hexápodo diseñado en SolidWorks 2019.

Fuente: Elaboración propia.

Para este estudio y simulación de esfuerzos se usará un Software capaz de generar y validar la información sobre el análisis de cada componente, siendo este el Software de diseño asistido por computadora Solidworks 2019. Este software cuenta con la opción de elementos finitos y genera resultados con valores muy cercanos a los reales de esfuerzos, deformaciones y factores de seguridad. (Lazo Pinto, 2018).

A simple vista podemos afirmar que las cargas aplicadas a los componentes del robot no son significativamente grandes y al usar un material como el ABS, los resultados en cuanto a deformación serán minúsculos si es comparado con un robot industrial. En tal caso, este análisis y simulación de esfuerzos busca reforzar y corroborar lo ya sabido, así como obtener información complementaria de todos los componentes. (Lazo Pinto, 2018).

Para la simulación solo serán tomados en cuenta los componentes más propensos a fallar siendo estos la base fija, base móvil, base del servomotor, base del tórax y las patas.

Todos los componentes a analizar cuentan con un cuerpo sólido y están hechos de plásticos PLA.

Las restricciones de movimiento se dan en las áreas donde cada componente es apoyado.

Las cargas que se tomarán en cuenta en la simulación, son las ejercidas por el peso de los componentes, la gravedad y la carga que moverá el robot. Las fuerzas originadas por la masa de los componentes se hallan en la **Tabla 18**.

Se plantea una simulación en el supuesto caso más crítico, siendo esta cuando la pata del robot hexápodo se encuentra extendido a 90° en todo, ya que en esta posición se genera el mayor momento. (Lazo Pinto, 2018).

Tabla 18. Componentes para el análisis de resistencia.

ELEMENTO	CANTIDAD	MASA (G)
PATA	6	235.8
CAJA	6	79.76
SERVO		
BASE	1	91.1
TÓRAX 1		
BASE	1	92.27
TÓRAX 2		
ACOPLE C1	6	33.75
ACOPLE C2	6	47.26

Fuente: Elaboración propia.

Los planos de cada componente al detalle y ensamble en el **Anexo 7**.

En cuanto a los resultados ofrecidos de la simulación por elementos finitos se obtienen 3, siendo estos:

Tensión de Von Mises: Indica el máximo esfuerzo aplicado al componente.

STRN: Indica la cantidad de movimiento que ejerce la carga.

URES: Indica el desplazamiento en mm.

4.3. Resultado De Análisis De Esfuerzos – Pata Hexápodo

En el análisis realizado a brazo se tomó en cuenta la carga que ejerce las rotulas, antebrazos, base móvil y carga. Luego de la simulación se obtienen los siguientes resultados, (Lazo Pinto, 2018).

Resumiendo, los datos obtenidos obtenemos:

Máximo esfuerzo normal: MPa

Máximo desplazamiento: mm

Al comparar estas evidencias, en la **Figura 72** se obtiene que los esfuerzos y desplazamientos son mínimos y por lo tanto despreciables, como Figura 67. Apreciamos un resultado de $7.804e+04$ (N/m²) máx. De esfuerzo de tensión nodal y un mín. de $5.800e-00$ (N/m²), deformación máx. es de $3.366e-01$ y min. De $3.508e-03$, un desplazamiento de $2.447e-07$ mm máx. Y $1.000e-30$ mm min, plano en el **Anexo 7**.

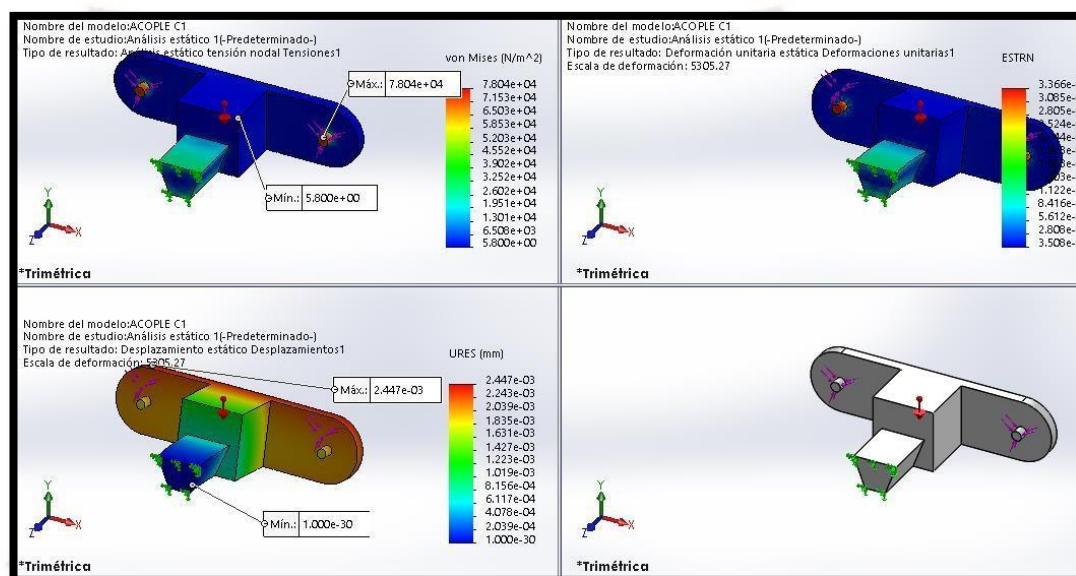


Figura 72. Esfuerzos de Von Mises, desplazamiento y deformación Acople C1.

Fuente: Solidworks 2019.

Al comparar estas evidencias, en la **Figura 73** se obtiene que los esfuerzos y desplazamientos son mínimos y por lo tanto despreciables, como Figura 68. Apreciamos un resultado de $2.285e+05$ (N/m²) máx. De esfuerzo de tensión nodal y un mín. De $7.769e-02$ (N/m²), deformación máx. es de $9.381e-01$ y min. De $3.510e-03$, un desplazamiento de $2.218e-02$ mm máx. Y $1.000e+30$ mm min, plano en el **Anexo 7**.

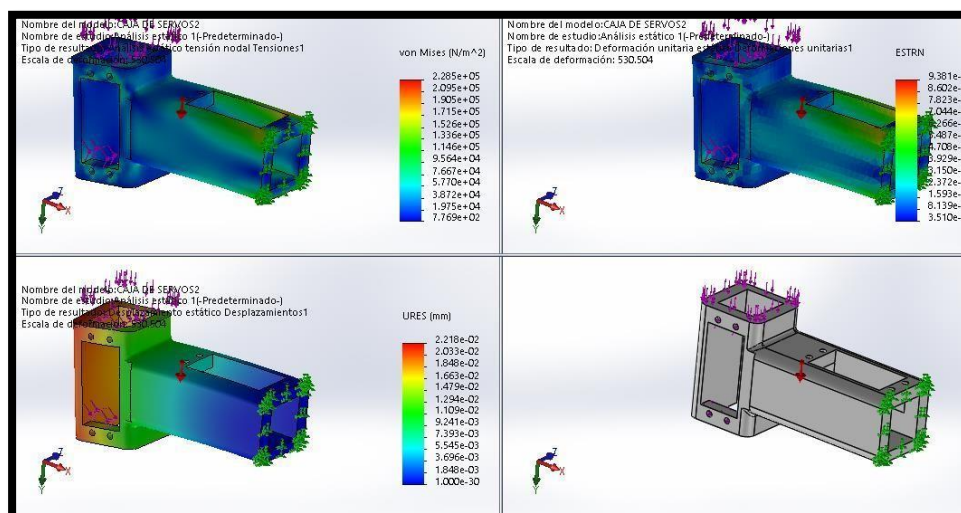


Figura 73. Esfuerzos de Von Mises, desplazamiento y deformación Caja Servo.

Fuente: Solidworks 2019.

Al comparar estas evidencias, en la **Figura 74** se obtiene que los esfuerzos y desplazamientos son mínimos y por lo tanto despreciables, como Figura 69. Apreciamos un resultado de $7.050e+04$ (N/m²) máx. De esfuerzo de tensión nodal y un mín. de $1.081e-01$ (N/m²), deformación máx. es de $2.802e-05$ y min. De $2.850e-08$, un desplazamiento de $1.385e-04$ mm máx. Y $1.000e-30$ mm min, plano en el **Anexo 7**.

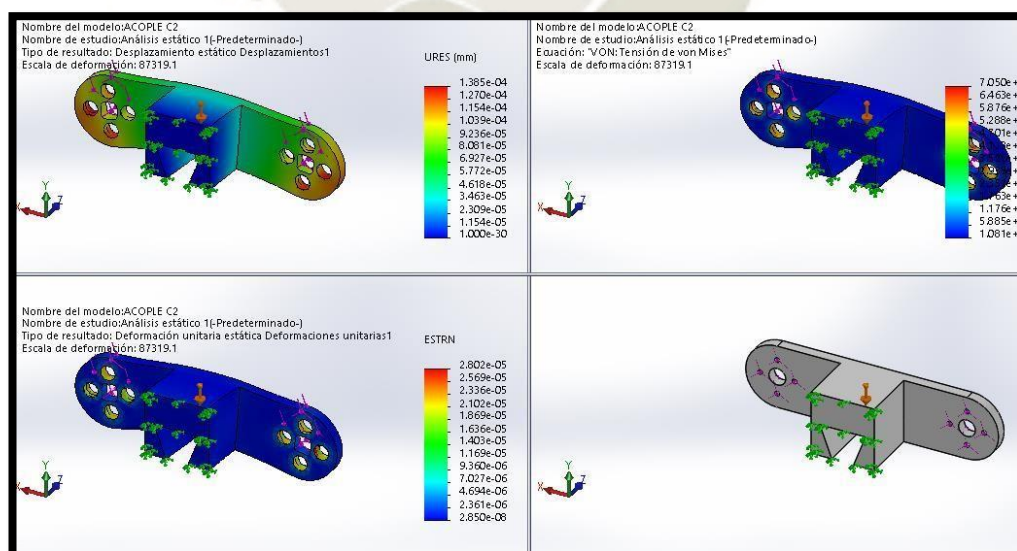


Figura 74. Esfuerzos de Von Mises, desplazamiento y deformación Acople C2.

Fuente: Solidworks 2019.

Al comparar estas evidencias, en la **Figura 75** se obtiene que los esfuerzos y desplazamientos son mínimos y por lo tanto despreciables, como Figura 70. Apreciamos un resultado de $3.684e+05$ (N/m²) máx. De esfuerzo de tensión nodal y un mín. de $1.430e-01$ (N/m²), deformación máx. es de $1.462e-01$ y min. De $8.356e-03$, un desplazamiento de $4.912e-02$ mm máx. Y $1.000e-30$ mm min, plano en el **Anexo 7**.

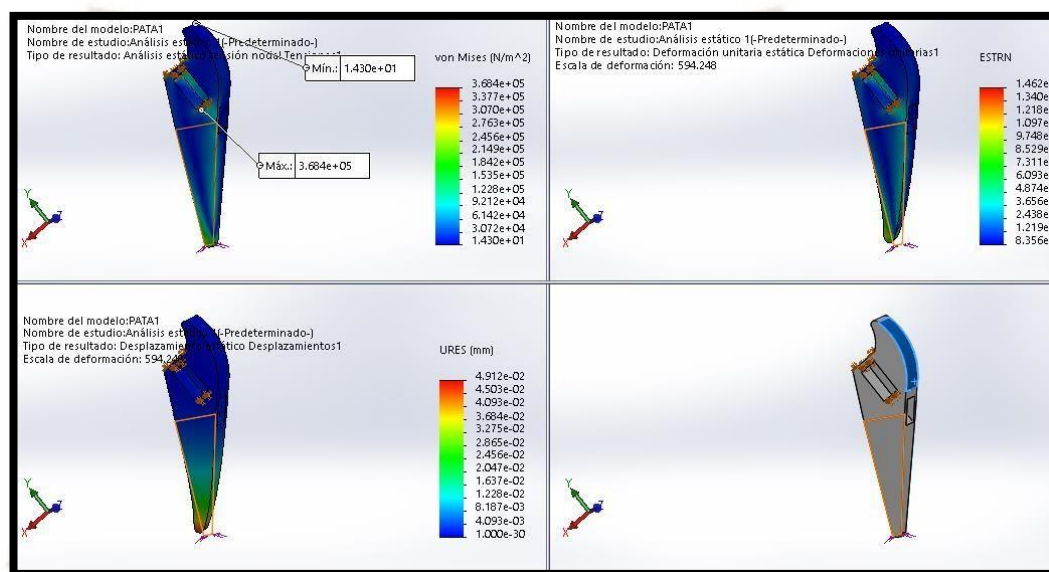


Figura 75. Esfuerzos de Von Mises, desplazamiento y deformación Pata 1.

Fuente: Solidworks 2019.

Al comparar estas evidencias, en la **Figura 76** se obtiene que los esfuerzos y desplazamientos son mínimos y por lo tanto despreciables, como Figura 71. Apreciamos un resultado de $1.238e+04$ (N/m²) máx. De esfuerzo de tensión nodal y un mín. de $1.363e-01$ (N/m²), deformación máx. es de $4.828e-01$ y min. De $1.182e-03$, un desplazamiento de $1.1035e-03$ mm máx. Y $1.000e-30$ mm min, plano en el **Anexo 7**.

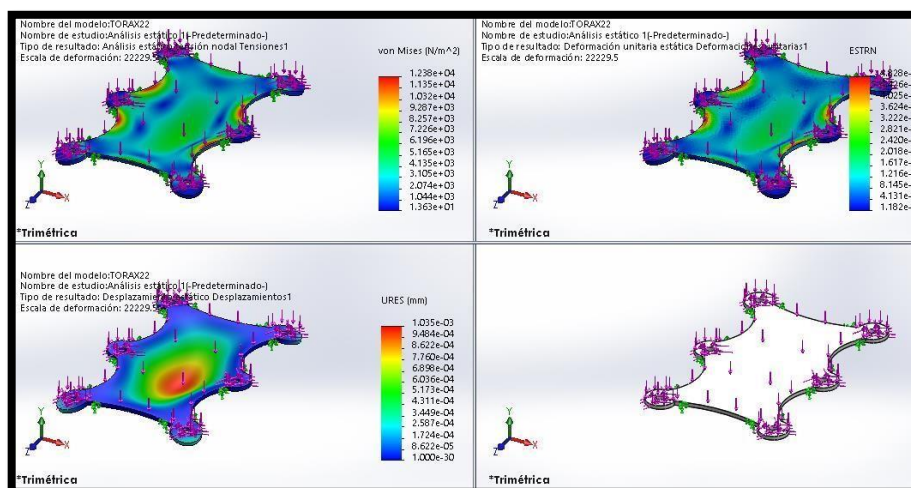


Figura 76. Esfuerzos de Von Mises, desplazamiento y deformación.

Fuente: Solidworks 2019.

4.4. Diseño Del Circuito PCB

Luego de tener definido todos los componentes mecánicos, es posible realizar la electrónica del robot Hexápodo. Se utilizó el diagrama esquemático del Arduino mega pero modificado a nuestras condiciones de uso.

Antes de entrar en consideración de que componentes electrónicos, es necesario recordar que este proyecto está enfocado en diseñar un robot capaz desplazarse por terrenos planos y que sea controlado con programas de libre uso, es decir, que no se desea estar limitado por licencias ni programas especiales que requieren de hardware especial, como, por ejemplo; el uso del software de MATLAB; es muy usado en el ambiente educativo, pero muy poco visto en las industrias, este software requiere de una PC o Laptop con requerimientos especiales tales como la memoria RAM, tarjeta de video, procesadores avanzados, etc.

Atendiendo estas consideraciones y como un medio para agregarle peso al presente proyecto, se decidió usar como controlador principal a un microcontrolador.

El hecho de usar un microcontrolador como dispositivo de control y no como medio para adquirir datos implica:

- Diseñar un programa capaz de cubrir las necesidades del robot.
- Trabajar bajo los requerimientos y limitantes del microprocesador tales como la memoria RAM, velocidad de transmisión, voltaje de operación, etc.
- Establecer la electrónica en torno al microprocesador.
- Finalmente, luego de definir las consideraciones del uso del microcontrolador es posible proceder con el diseño.

Una vez que se pudo definir todos los componentes necesarios, es posible realizar un esquema o diagrama que indique como es que se integran los componentes. Este diseño estará echo para los 18 servomotores , puesto que es el elemento que requiere más esfuerzo en el proyecto, se muestra el diagrama esquemático en la **figura 77**, más detallado en el anexo 5 y código de programación en Arduino **Anexo 4**.

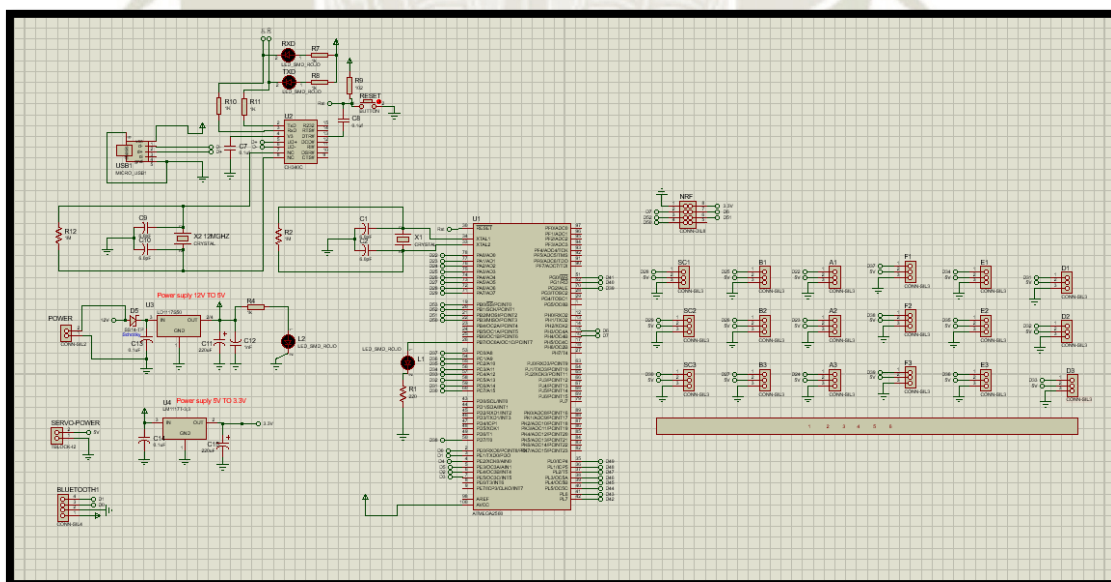


Figura 77. Circuito esquemático placa de control hexápodo.

Fuente: Proteus 8 profesional.

Los servomotores TowerPro se conectarán uno a uno a una placa PCB diseñada (con una alimentación en paralelo) con el fin de ofrecer estabilidad en la tensión de alimentación a nuestros servomotores y principalmente brindar comodidad y orden a todas nuestras conexiones, se muestra en la **figura 78**.

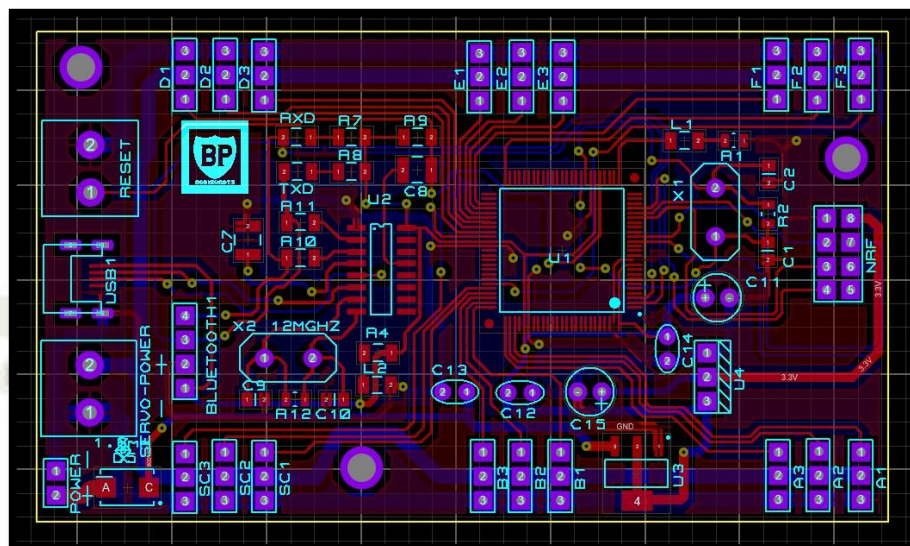


Figura 78. PCB circuito de control Hexápodo.

Fuente: Proteus (ARES) 8 profesional.

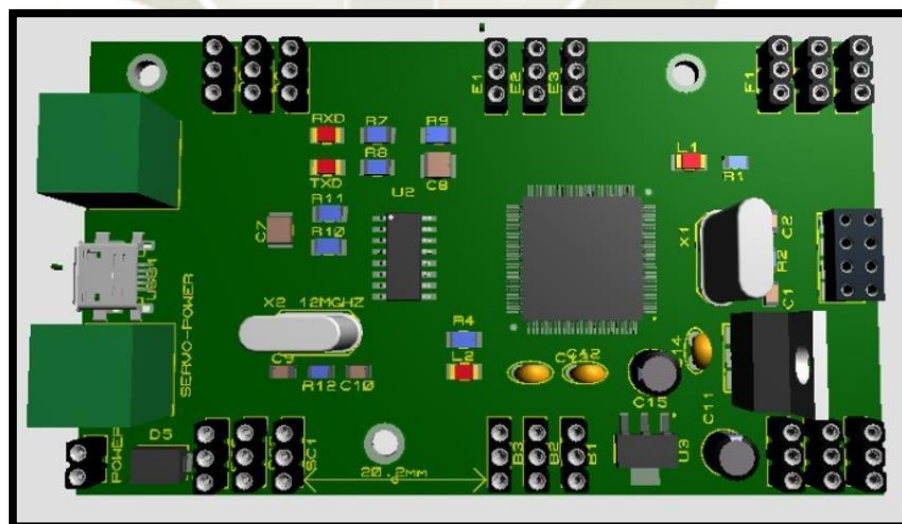


Figura 79. Placa de control del robot Hexápodo.

Fuente: Proteus (ARES) 8 profesional.

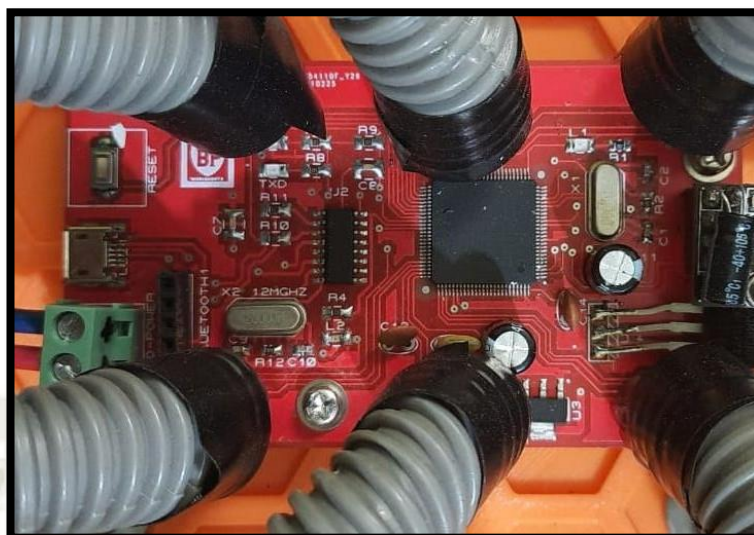


Figura 80. Placa de control del robot Hexápodo Físico.

Fuente: Elaboración propia.

4.5. Diseño Del Circuito PCB Mando De Control

Despues de tener todos los componentes necesarios, Este diseño estará basado en el control a distancia del robot hexápodo en la **Figura 81**, detallado en el **anexo 6** y código de programación en Arduino **Anexo 4**.

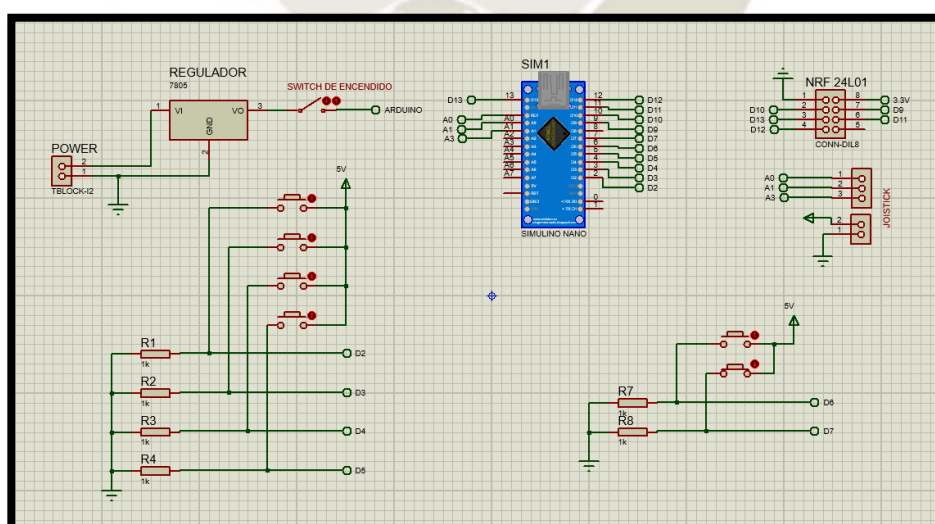


Figura 81. Circuito esquemático placa mando de control hexápodo.

Fuente: Proteus 8 profesional.

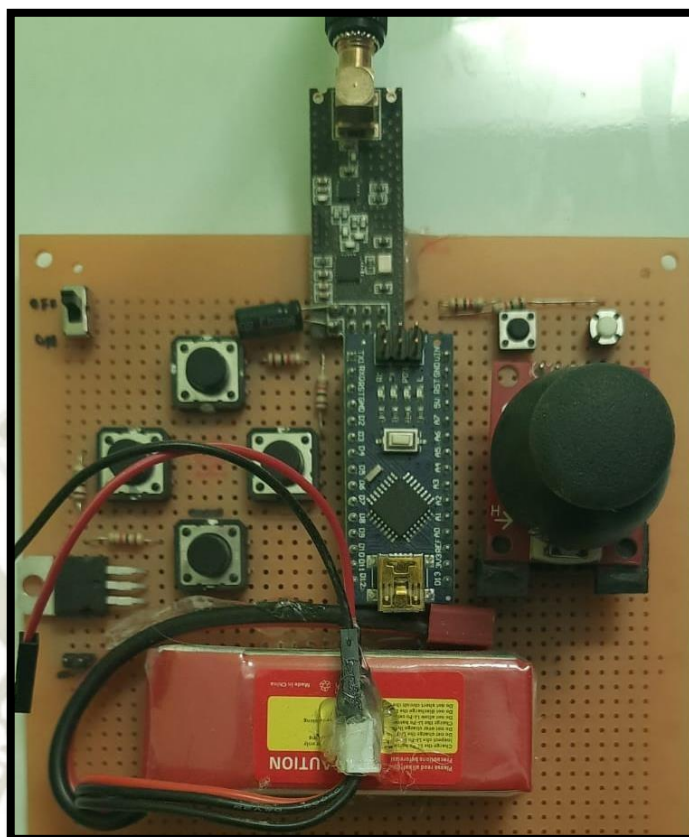


Figura 82. Placa mando de control del robot Hexápodo Físico.

Fuente: Elaboración propia

4.5.1. Diagrama de Flujo de control Inalámbrica

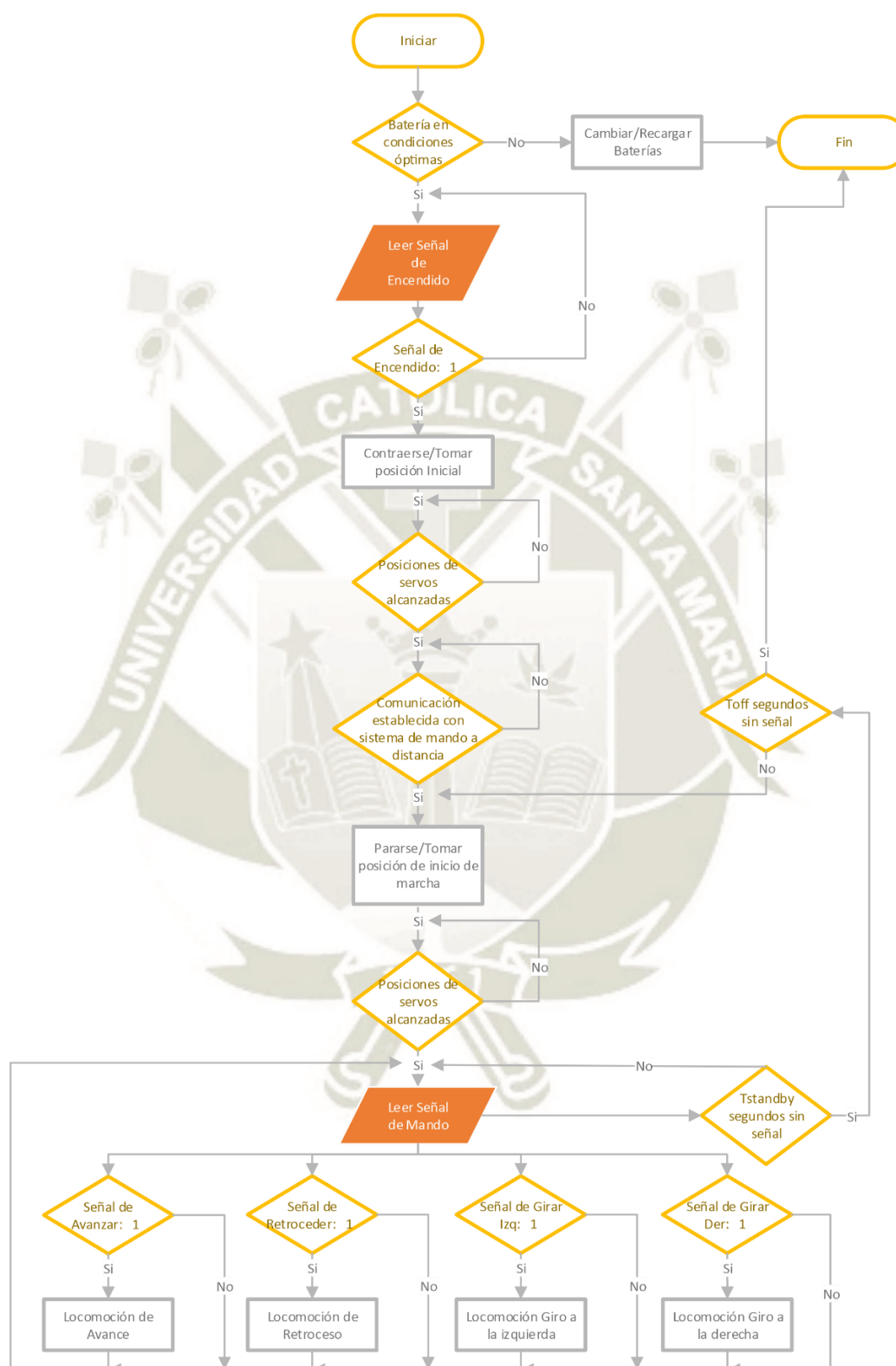


Figura 83. Diagrama de flujo de control de robot hexápodo.

Fuente: Elaboración propia

4.6. Diseño Del Circuito PCB Sensado De Temperatura y Co2

Una vez definidos todos los componentes necesarios, es posible realizar un esquema o diagrama que indique como es que se integran los componentes. Este diseño estará basado en el sensado de temperatura y co2 en la **Figura 84**, mas detalle en el **anexo 6** y código de programación en Arduino **Anexo 4**.

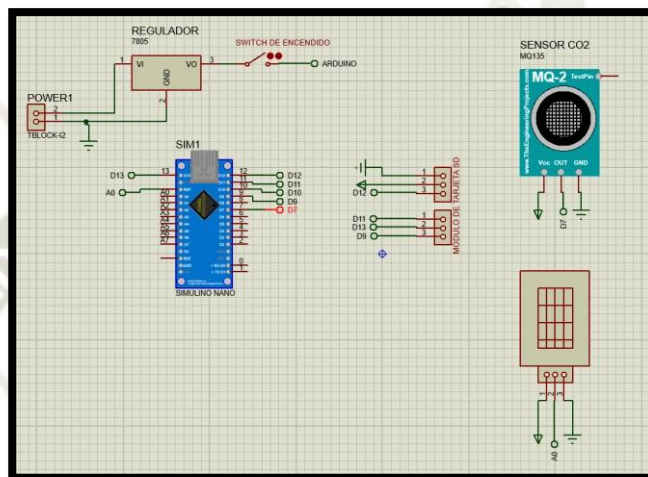


Figura 84. Circuito esquemático placa de sensado de temperatura y co2.

Fuente: Proteus 8 profesional.

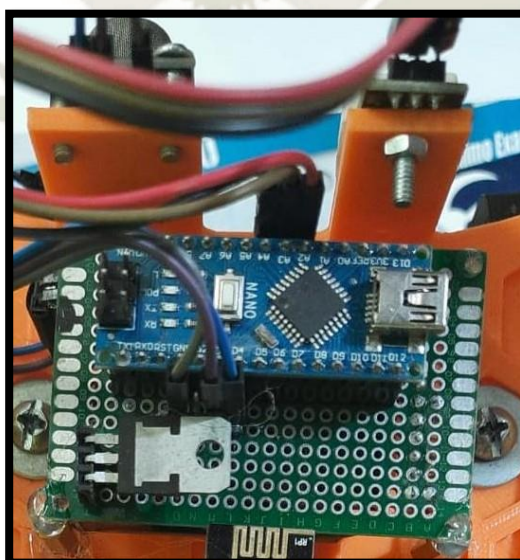


Figura 85. Placa de Sensado temperatura y co2.

Fuente: Elaboración propia.

4.7. Pruebas de Interfaz interactivo virtual

4.7.1. Exportación de archivos SolidWorks a Processing

Primeramente creamos el ensamble en solidworks en archivo de extension .STL, lo que nos permite exportarlo hacia el entorno de trabajo de Blender, pero antes vamos a “opciones”, esto nos permite exportarlo en una sola pieza, como se muestra en la figura 86.

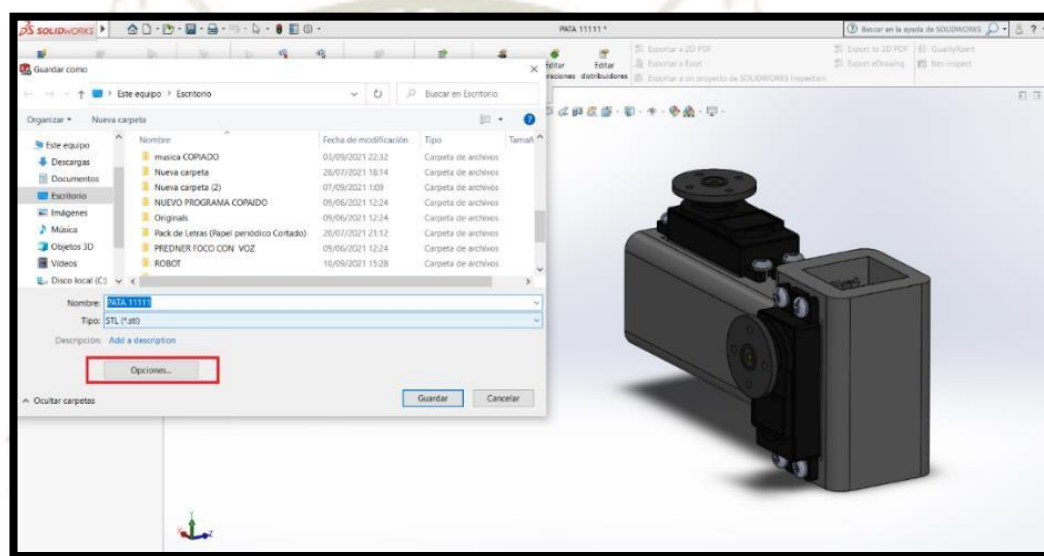
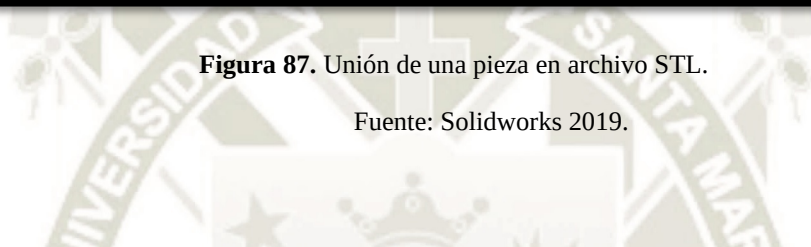


Figura 86. Opciones para cambiar una sola pieza

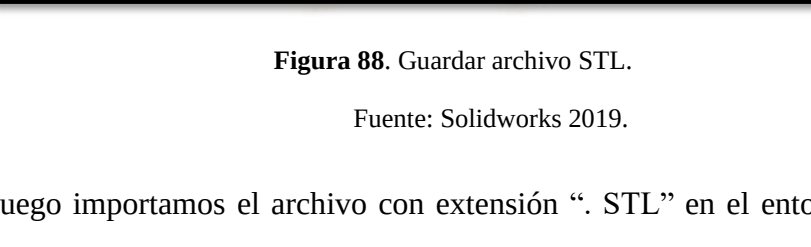
Fuente: Solidworks 2019.

Una vez seleccionado marcamos “Guardar todos los componente de ensamble en un unico archivo” lo que nos permite guardar en una sola pieza con extension .STL, como se muestra en la figura , luego precionamos en “guardar” como se muestra en la figura 87 .



Fuente: Solidworks 2019.

Fuente: Solidworks 2019.



Fuente: Solidworks 2019.

Fuente: Solidworks 2019.

Luego importamos el archivo con extensión “. STL” en el entorno de trabajo de Blender, como se muestra en la **figura 89**.

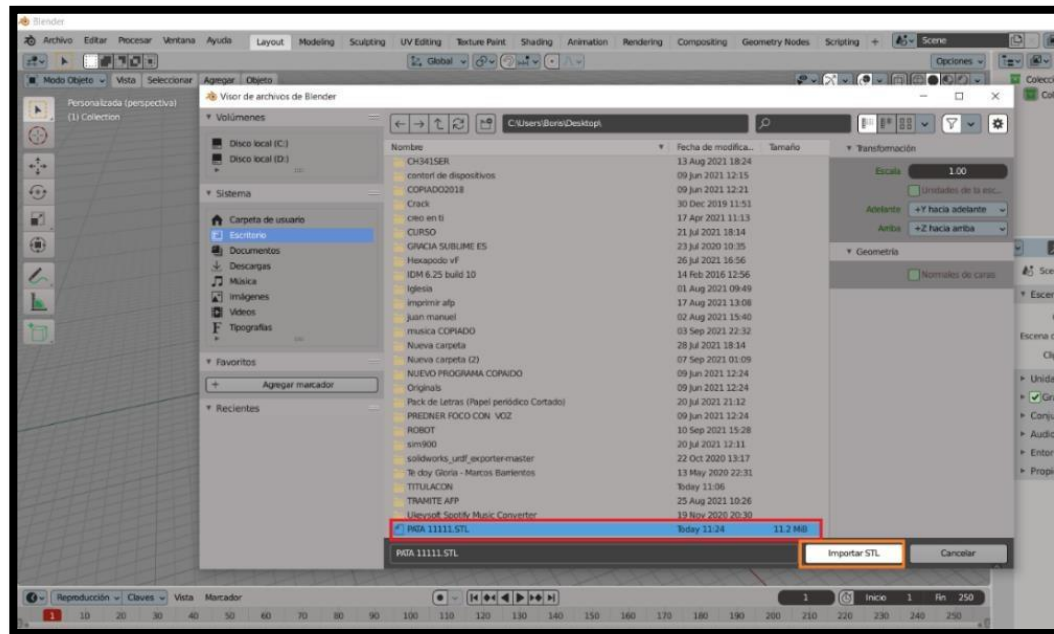


Figura 89. Exportación del archivo STL a blender.

Fuente: Blender.

Una vez importado la pieza a trabajar, le damos color como se muestra en la **figura 90.**

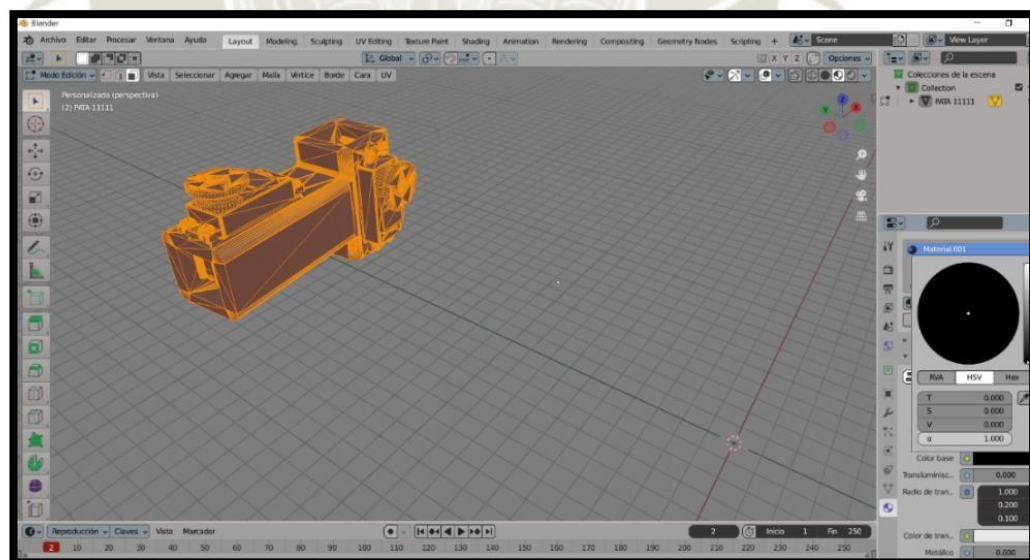


Figura 90. Dar color a la pieza exportada.

Fuente: Blender.

Por último, exportamos la pieza ya terminada en una extensión “.obj” la cual nos servirá para nuestra simulación en processing, como se muestra en la **figura 91.**

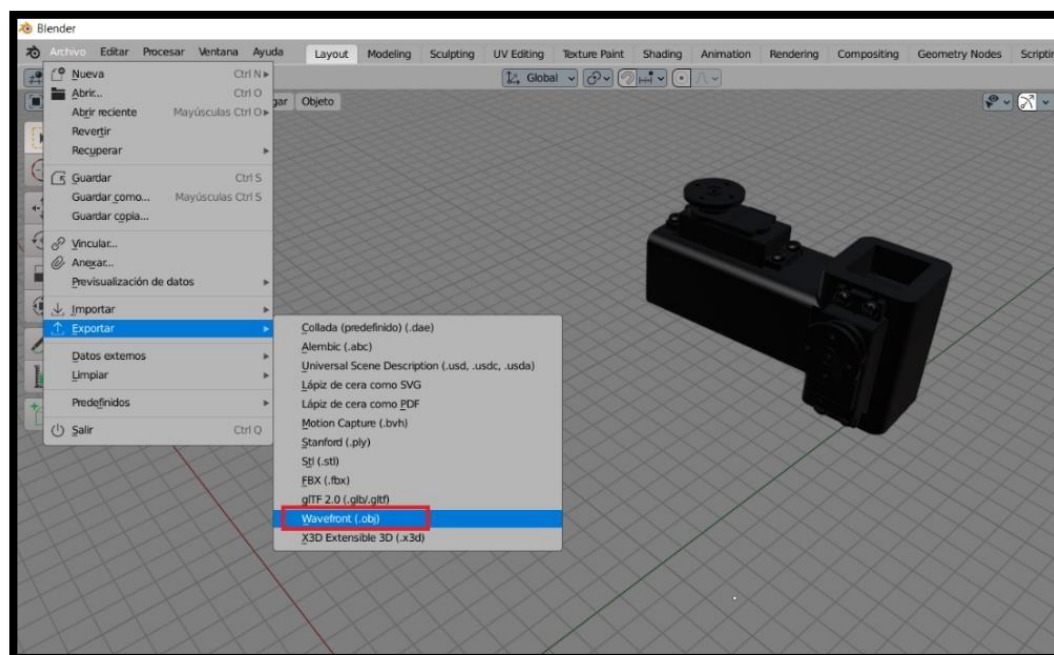
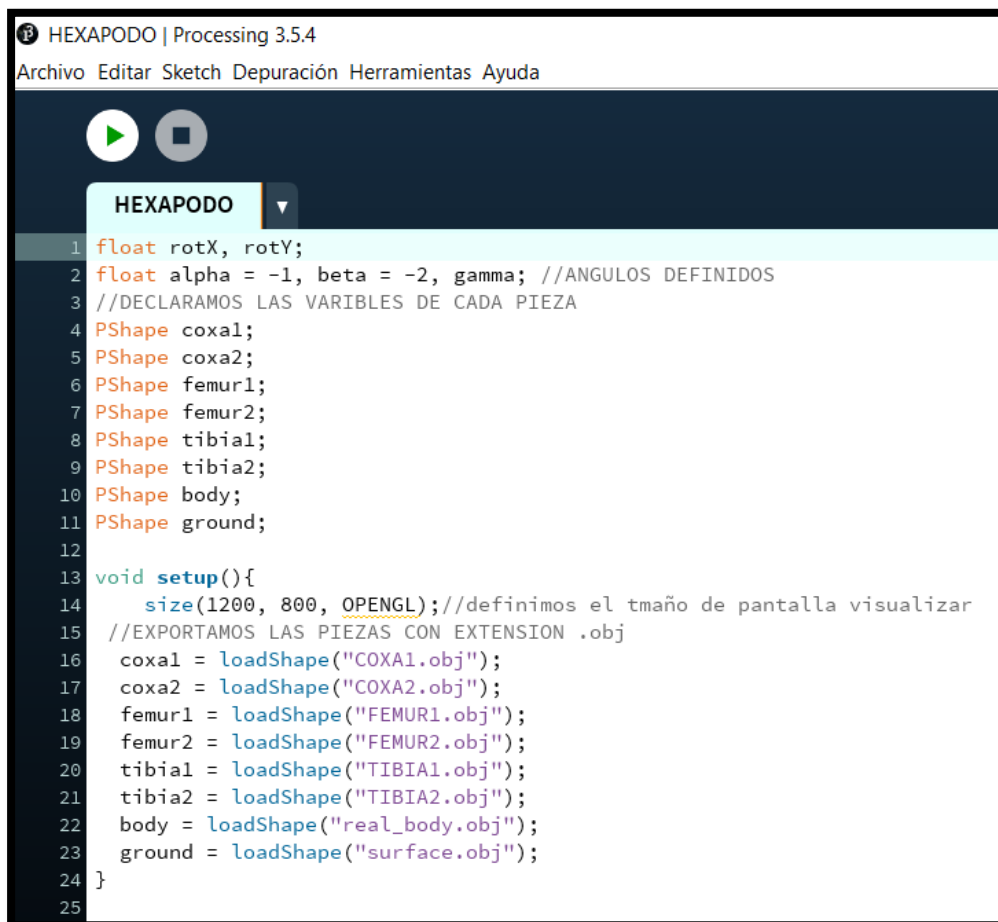


Figura 91. Exportar archivo con extensión. Obj.

Fuente: Blender.

Antes de extraer la pieza diseñada, necesitamos cambiar a un renderizado 3D agregando `OPENGL` a la función de tamaño. La rotación se realiza usando `rotateX` (ángulo), `rotateY` (ángulo) y `rotateZ` (ángulo). `width` y `height` hacen referencia a los valores que agregamos a la función de tamaño, `translate (width / 2, height / 2)`, debemos llamar a la función `smooth ()`. Esto no funcionará sin agregar fondo (color), `Lights ()` enciende las luces y agrega sombras a la pieza.

Tenemos que crear una nueva carpeta en la misma carpeta asignar el nombre "data". Colocar los archivos .obj en esa carpeta, ahora podemos importar las piezas a nuestro boceto creando una `PShape` para cada uno de ellos y usando `loadShape` ("archivo") para asignar el archivo obj. Reemplazamos la caja con forma (base) y Processing dibujará la geometría, como se muestra en la **figura 92**.



```

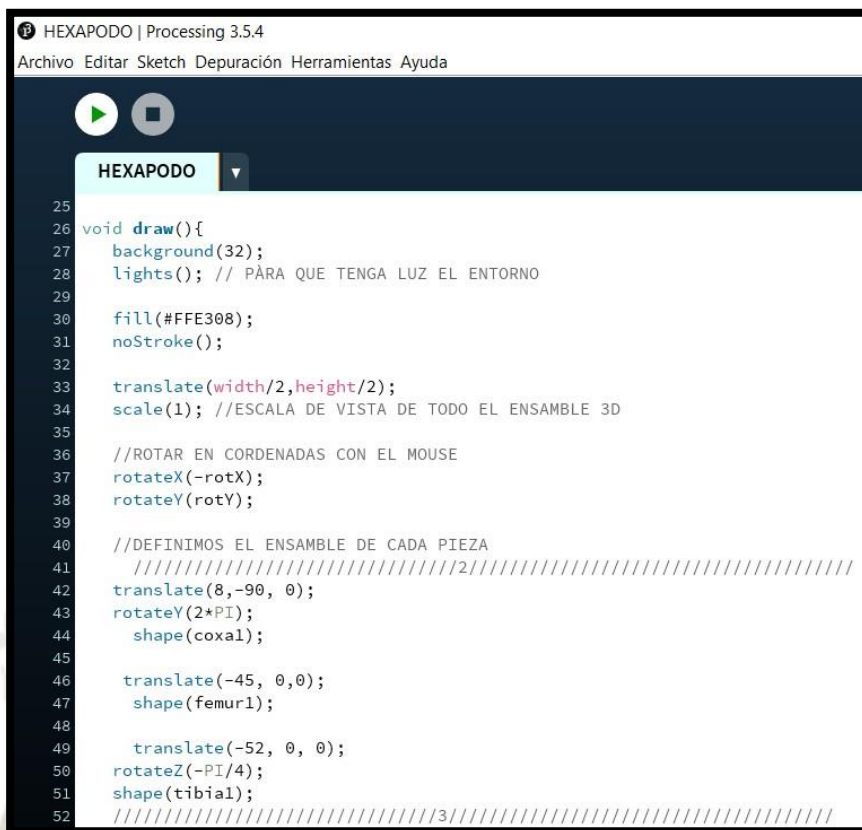
HEXAPODO
1 float rotX, rotY;
2 float alpha = -1, beta = -2, gamma; //ANGULOS DEFINIDOS
3 //DECLARAMOS LAS VARIABLES DE CADA PIEZA
4 PShape coxa1;
5 PShape coxa2;
6 PShape femur1;
7 PShape femur2;
8 PShape tibial1;
9 PShape tibial2;
10 PShape body;
11 PShape ground;
12
13 void setup(){
14     size(1200, 800, OPENGLE); //definimos el tamaño de pantalla visualizar
15     //EXPORTAMOS LAS PIEZAS CON EXTENSION .obj
16     coxa1 = loadShape("COXA1.obj");
17     coxa2 = loadShape("COXA2.obj");
18     femur1 = loadShape("FEMUR1.obj");
19     femur2 = loadShape("FEMUR2.obj");
20     tibial1 = loadShape("TIBIA1.obj");
21     tibial2 = loadShape("TIBIA2.obj");
22     body = loadShape("real_body.obj");
23     ground = loadShape("surface.obj");
24 }
25

```

Figura 92. Entorno de trabajo de Processing.

Fuente: Processing.

Ahora ensamblaremos el robot agregando las partes restantes utilizando las funciones de traslación / rotación para colocar las piezas. Los valores de traslación y rotación siempre se sumarán. Eso significa que todas las partes están en una cadena en la que cada eslabón se mueve en relación con su predecesor. Los valores de rotación están en radianes se muestra en la **figura 93**.



```

25
26 void draw(){
27   background(32);
28   lights(); // PARA QUE TENGA LUZ EL ENTORNO
29
30   fill(#FFE308);
31   noStroke();
32
33   translate(width/2,height/2);
34   scale(1); //ESCALA DE VISTA DE TODO EL ENSAMBLE 3D
35
36   //ROTAR EN CORDENADAS CON EL MOUSE
37   rotateX(-rotX);
38   rotateY(rotY);
39
40   //DEFINIMOS EL ENSAMBLE DE CADA PIEZA
41   ///////////////////////////////////2////////////////////////////////////
42   translate(8,-90, 0);
43   rotateY(2*PI);
44   shape(coxa1);
45
46   translate(-45, 0,0);
47   shape(femur1);
48
49   translate(-52, 0, 0);
50   rotateZ(-PI/4);
51   shape(tibia1);
52   ///////////////////////////////////3////////////////////////////////////

```

Figura 93. Entorno de trabajo de processing coordenadas.

Fuente: Processing.

Vista isométrica del robot hexápodo ya ensamblado se muestra en la **figura 94**.

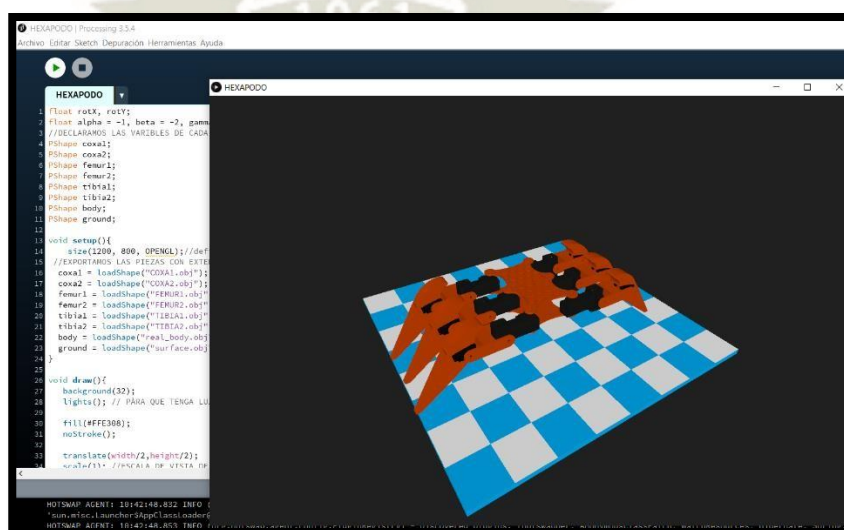


Figura 94. Vista isométrica robot ensamblado.

Fuente: Processing.

El Interfaz interactivo ha sido desarrollado bajo la secuencia de la **figura 95** permite validar los cálculos de la plataforma a tiempo real.



Figura 95. Secuencia de configuración entorno virtual.

Fuente: Elaboración propia.

En la **figura 96** mostrada, enlazamos la conexión el robot y la plataforma virtual, presionamos “conexión” dependiendo del COM puerto que se conectó. Por default se pondrá en la posición cero inicial.

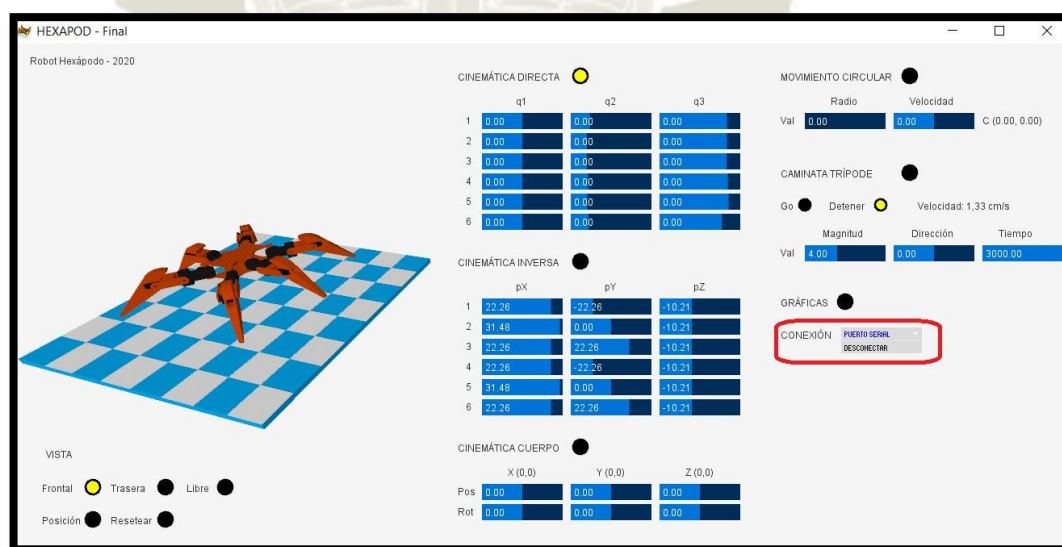


Figura 96. conexión puerto COM.

Fuente: Elaboración propia.

En la **figura 97** que se muestra, presionamos el botón “libre” la cual nos sirve para poder mover en cualquier vista el robot. Luego presionamos el botón “posición” esta nos permite que el robot se ponga en posición de caminata.

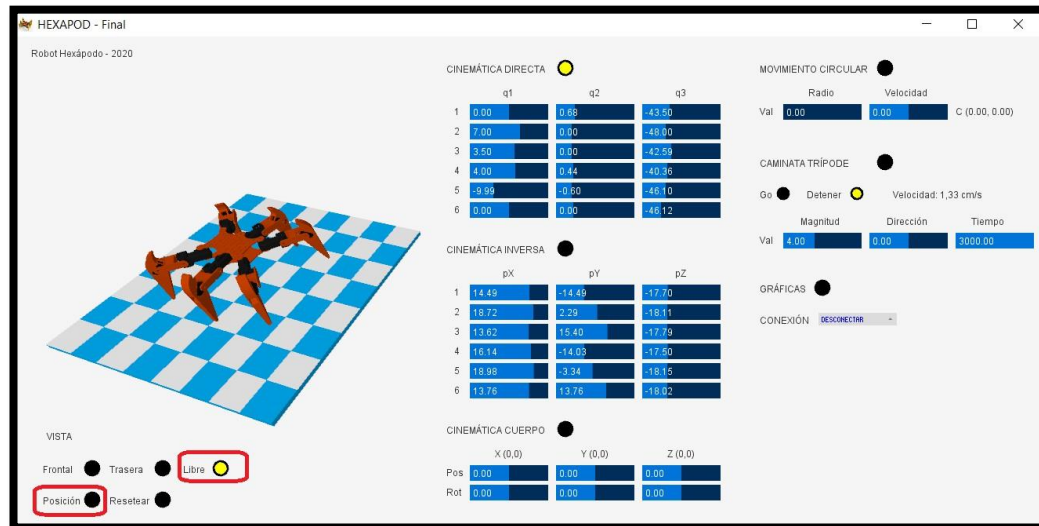


Figura 97. Activación botón posición.

Fuente: Elaboración propia.

En la **figura 98** se muestra la prueba de “Cinemática Directa” lo que nos permite manipular mediante las sliders de las 6 patas y cada una con 3 articulaciones.

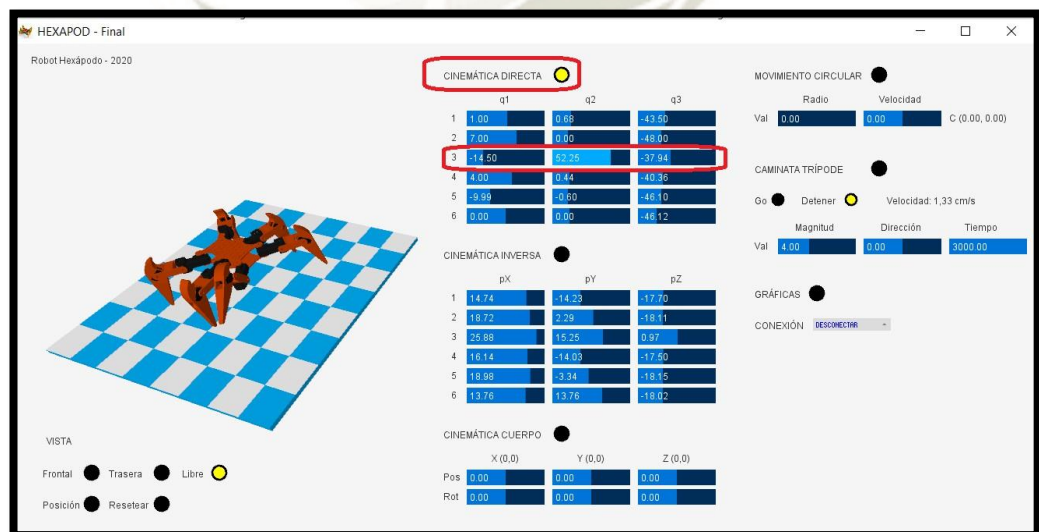


Figura 98. Activación de la cinemática directa.

Fuente: Elaboración propia.

En la **figura 99** se muestra la prueba de “Cinemática Inversa” lo que nos permite manipular mediante las sliders de las 6 patas y cada una con 3 articulaciones

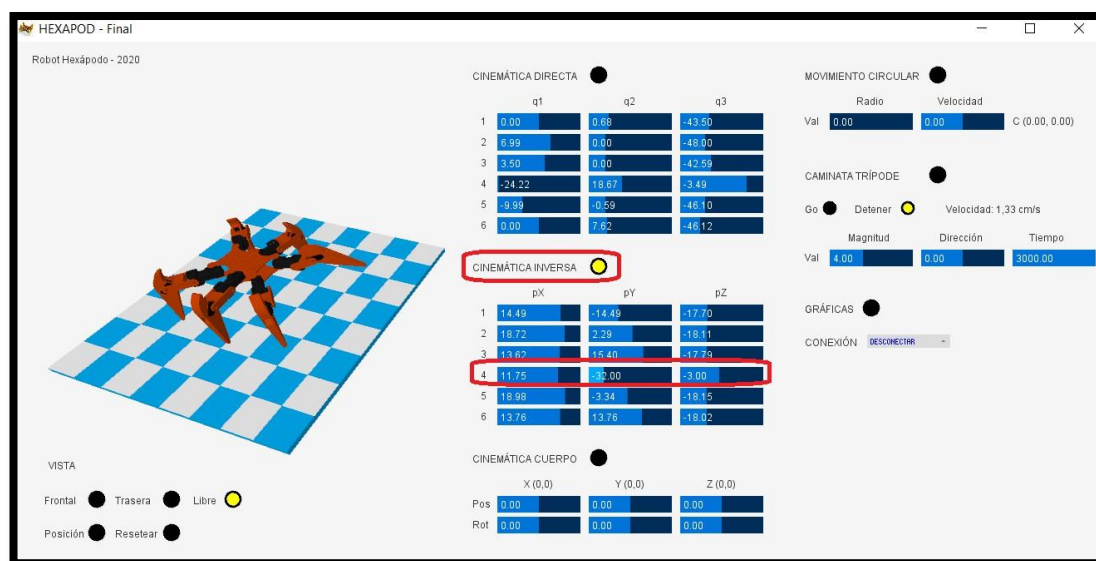


Figura 99. Activación de la cinemática Inversa.

Fuente: Elaboración propia.

En la **figura 100** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante las sliders posición de desplazamiento en el eje “x”.

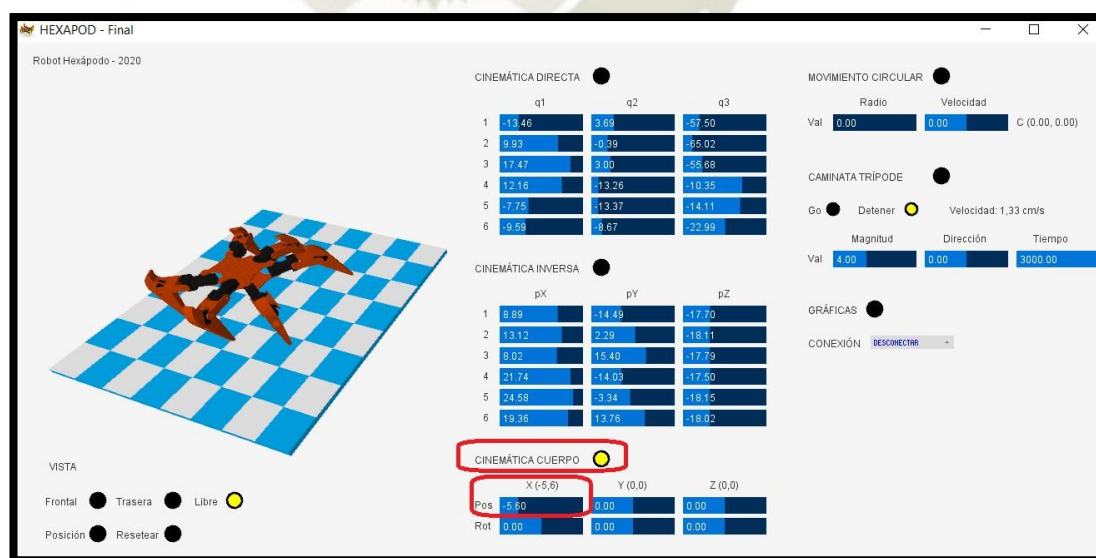


Figura 100. Activación de la cinemática del Cuerpo Traslación eje X.

Fuente: Elaboración propia.

En la **figura 101** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante las sliders posición de desplazamiento en el eje “y”.

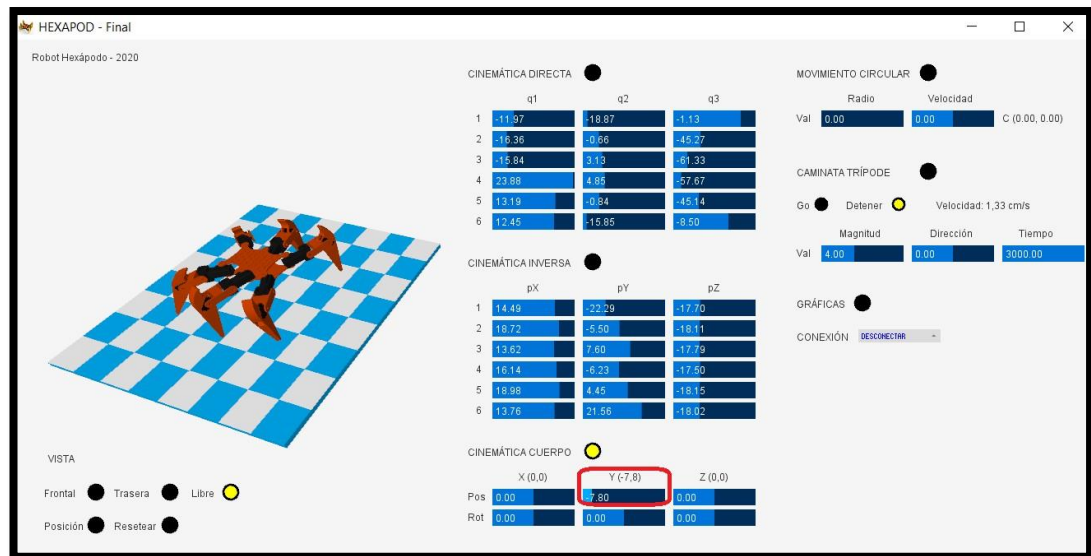


Figura 101. Activación de la cinemática del Cuerpo Traslación eje Y.

Fuente: Elaboración propia.

En la **figura 102** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante las sliders posición de desplazamiento en el eje “z”.

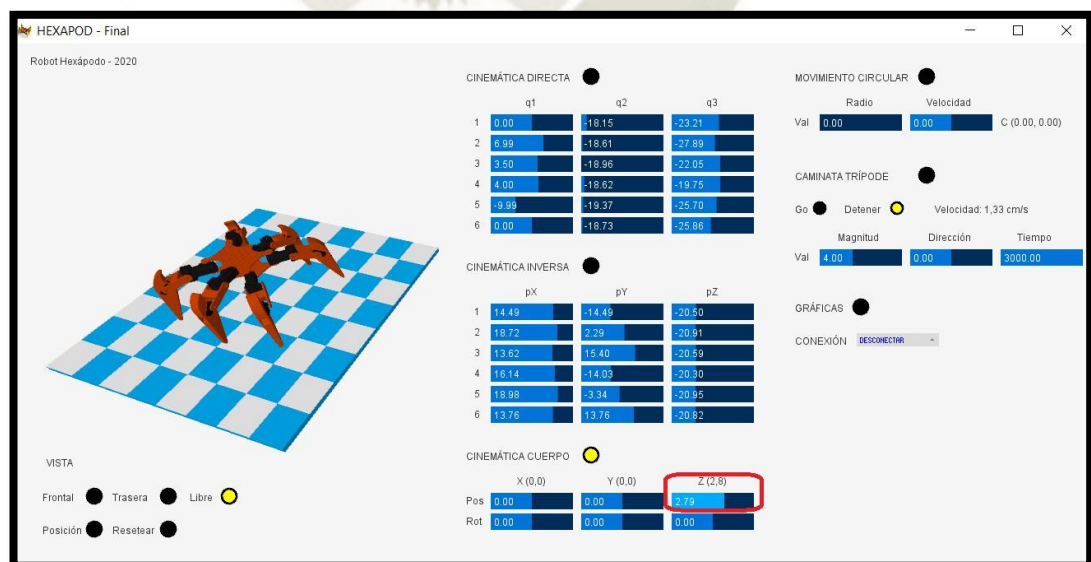


Figura 102. Activación de la cinemática del Cuerpo Traslación eje Z.

Fuente: Elaboración propia.

En la **figura 103** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante los sliders posición de rotación en el eje “x”.

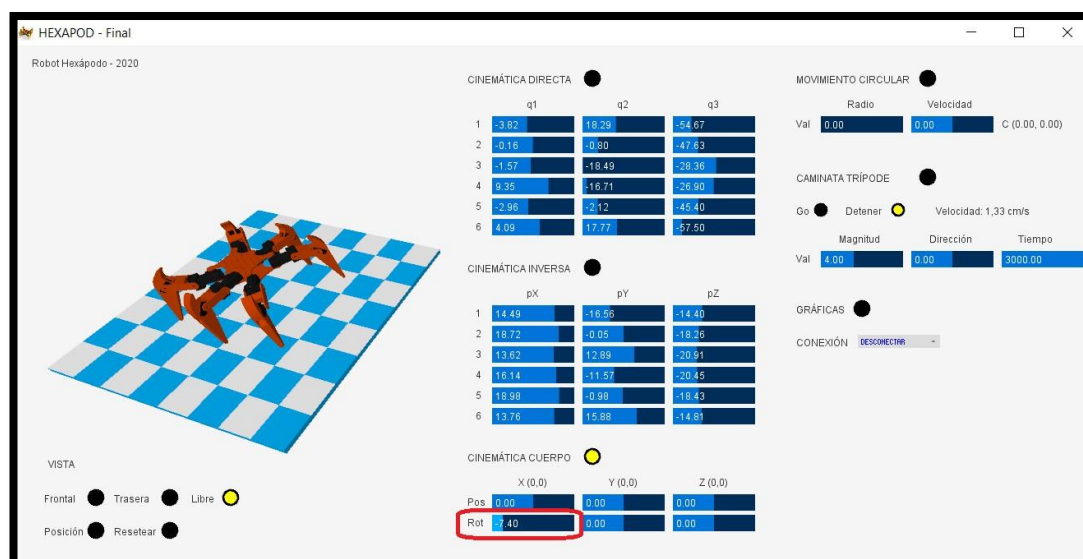


Figura 103. Activación de la cinemática del Cuerpo Rotación eje X.

Fuente: Elaboración propia.

En la **figura 104** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante las sliders posición de rotación en el eje “y”.

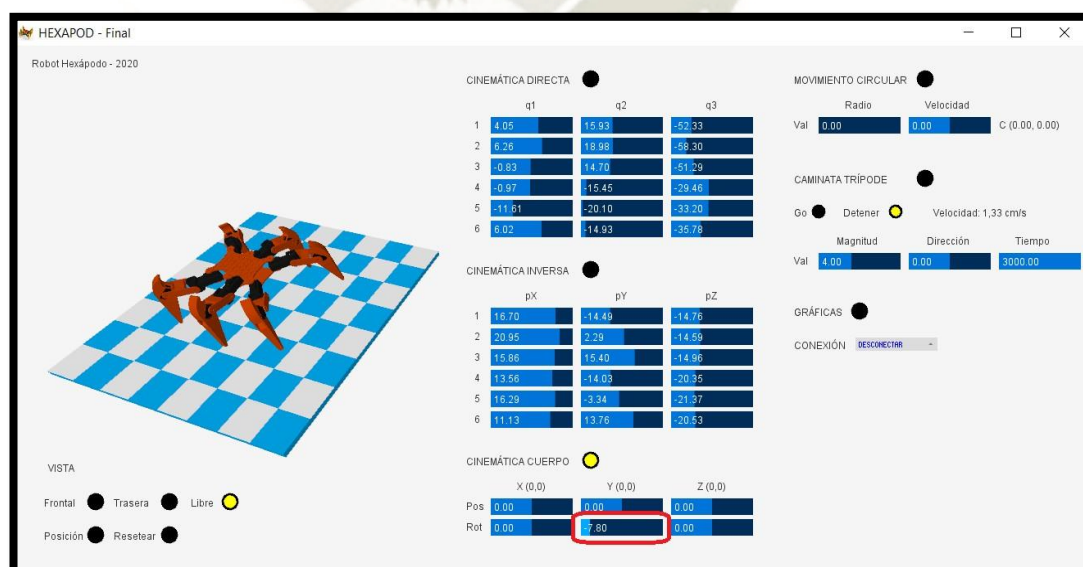


Figura 104. Activación de la cinemática del Cuerpo Rotación eje Y.

Fuente: Elaboración propia.

En la **figura 105** se muestra la prueba de “Cinemática del cuerpo” lo que nos permite manipular mediante las sliders posición de rotación en el eje “z”.

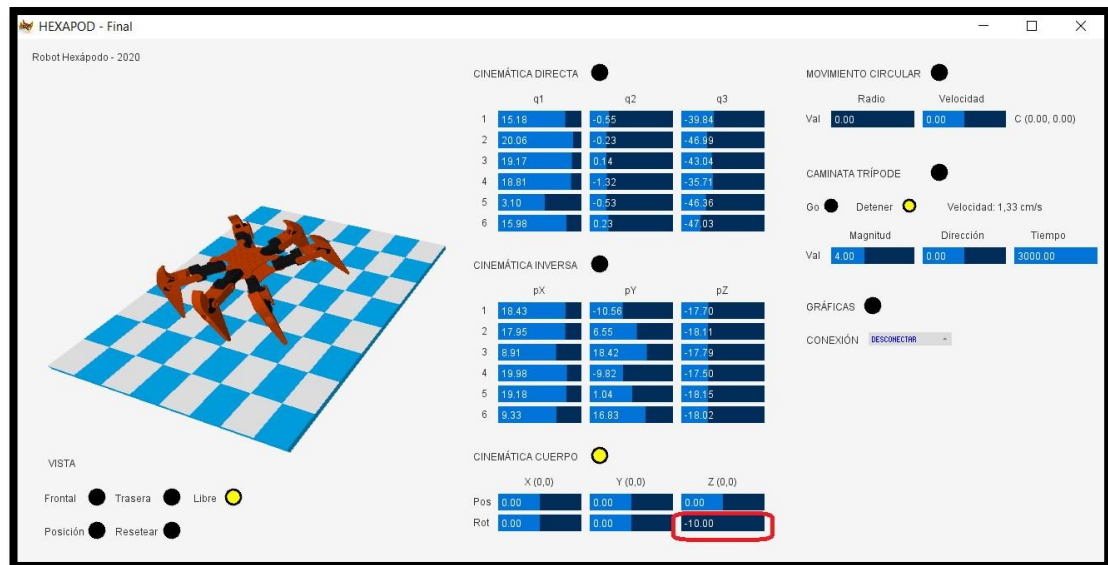


Figura 105. Activación de la cinemática del Cuerpo Rotación eje Z.

Fuente: Elaboración propia.

En la **figura 106** se muestra la prueba de “Movimiento circular” lo que nos permite manipular mediante las sliders, “Radio” determina el radio de circunferencia en cm, “velocidad” determina la velocidad de la trayectoria circular en cm/s.

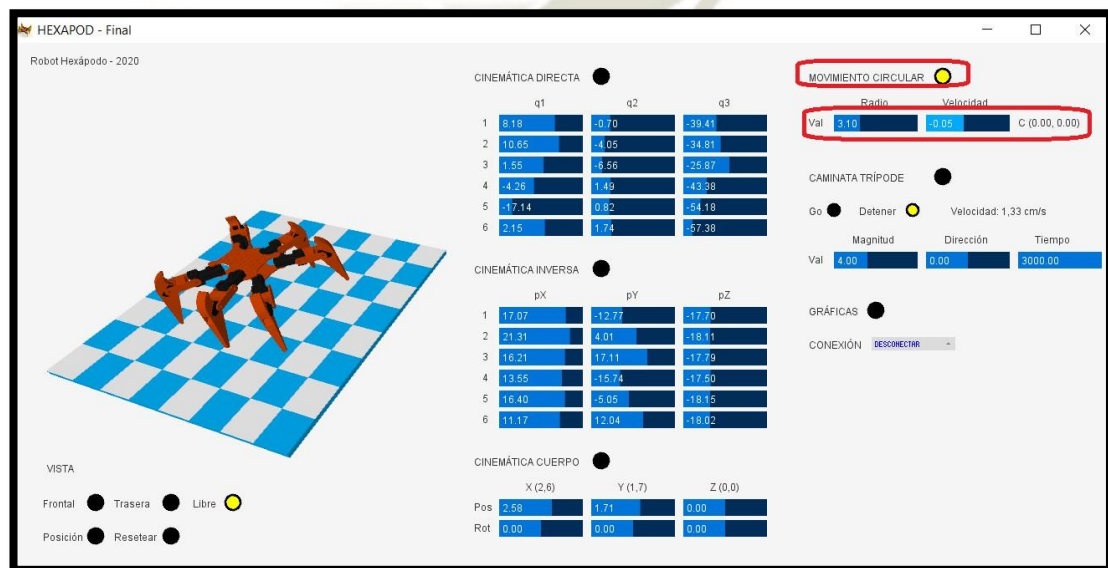


Figura 106. Activación de Movimiento Circular.

Fuente: Elaboración propia.

En la **figura 107** se muestra la prueba de “Caminata Trípole” lo que nos permite manipular mediante las sliders, “Magnitud” determina la magnitud de la trayectoria parabólica en cm, “Dirección” determina la dirección donde avanza el robot, ya sea izquierda, derecha, adelante y atrás, “Tiempo” determina el tiempo de trayectoria parabólica, por lo tanto, la velocidad de avance del robot. “Go” hace que el robot camine, “Detener” hace que el robot se pare completando su ciclo.

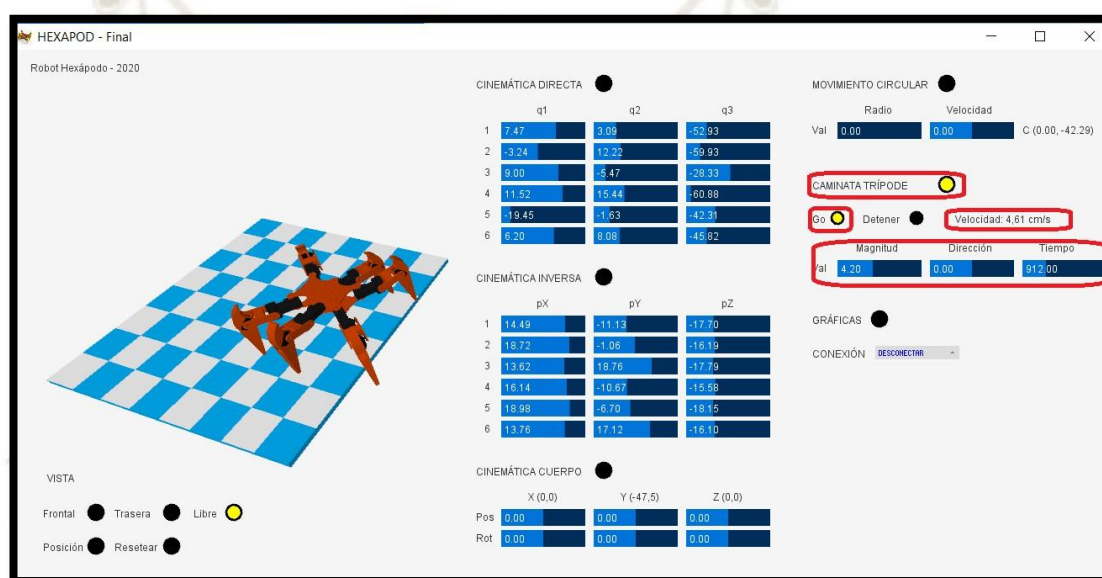


Figura 107. Activación de caminata Trípod.

Fuente: Elaboración propia.

En la **figura 108** se muestra la prueba de “Caminata Trípole” al presionar este botón Hace que el robot camine, pero cuando presionamos la tecla “Gráficas” nos permite visualizar las gráficas de cada articulación del robot.

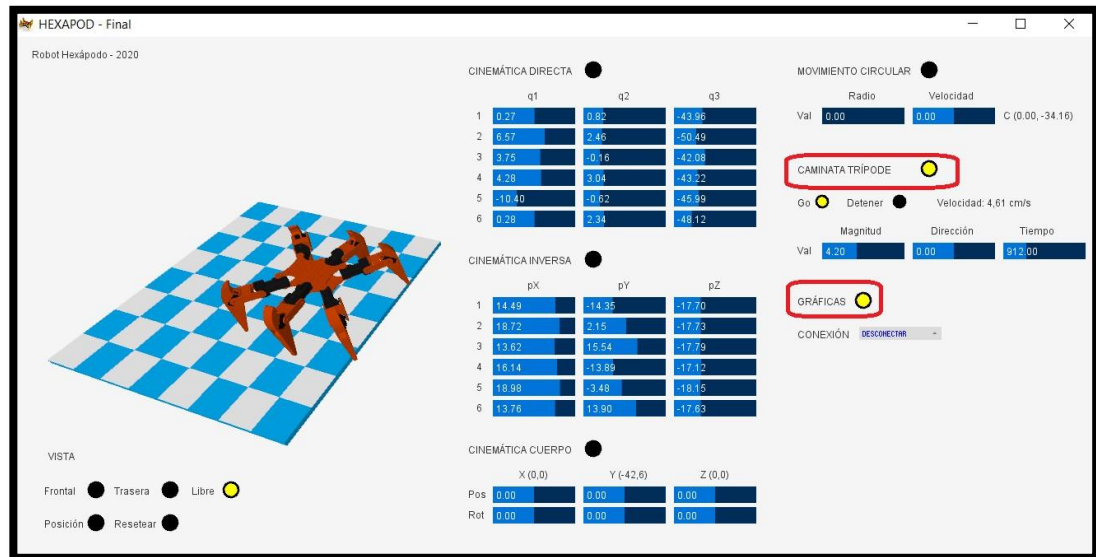


Figura 108. Activación de gráficas y caminata tripod.

Fuente: Elaboración propia.

En la **figura 109** se muestra las gráficas las posiciones de cada articulación del robot al presionar el botón “Exportar” se crea un archivo Excel con extensión .xls, con la cual podemos extraer los valores articulares para fines demostrativos, código de toda la simulación en proccesing en el Anexo 3.

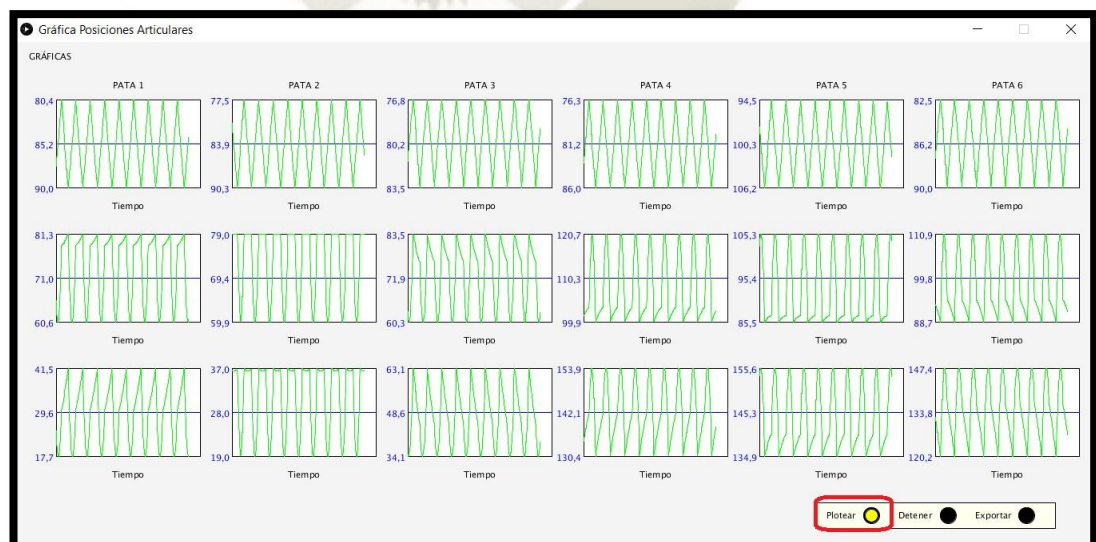


Figura 109. Graficas de cada articulación.

Fuente: Elaboración propia.

4.8. Pruebas de Sensado de CO₂ y Temperatura.

En la **figura 110** se muestra tomando los datos en tiempo real de Co₂ y temperatura en el distrito de Sicuani – cerro Jururo.



Figura 110. Toma de datos de temperatura y CO₂ distrito de Sicuani – cerro Jururo.

Fuente: Elaboración propia.

Posteriormente para su comparación de resultados se hizo otra prueba de toma de datos en la ciudad de Sicuani esto se muestra en la **figura 111**.



Figura 111. Toma de datos de temperatura y CO₂ en distrito de Sicuani.

Fuente: Elaboración propia.

Por consiguiente, se obtuvo los siguientes resultados resultados que se obtuvo en el distrito de Sicuani – cerro Jururo. Hace referencia de una temperatura de 23° constante y una variación de co2 Max de 131 ppm y un Min de 70 ppm. Las cuales se hizo en 1000 muestras en 100 milisegundos se muestra en la **figura 112**.

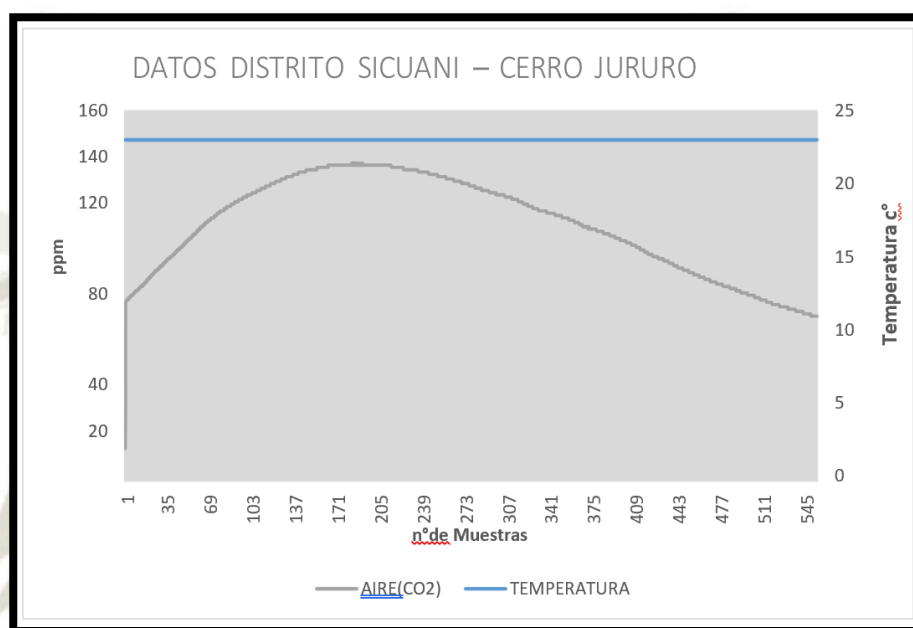


Figura 112. Resultados de temperatura y CO2 en Sicuani – Cerro Jururo.

Fuente: Elaboración propia

En la **figura 113** se muestra los resultados que se obtuvieron en el distrito de Sicuani. Hace referencia de una temperatura de 22° constante y una variación de co2 Max de 314 ppm y un Min de 98 ppm. Las cuales se hizo en 1000 muestras en 100 milisegundos.

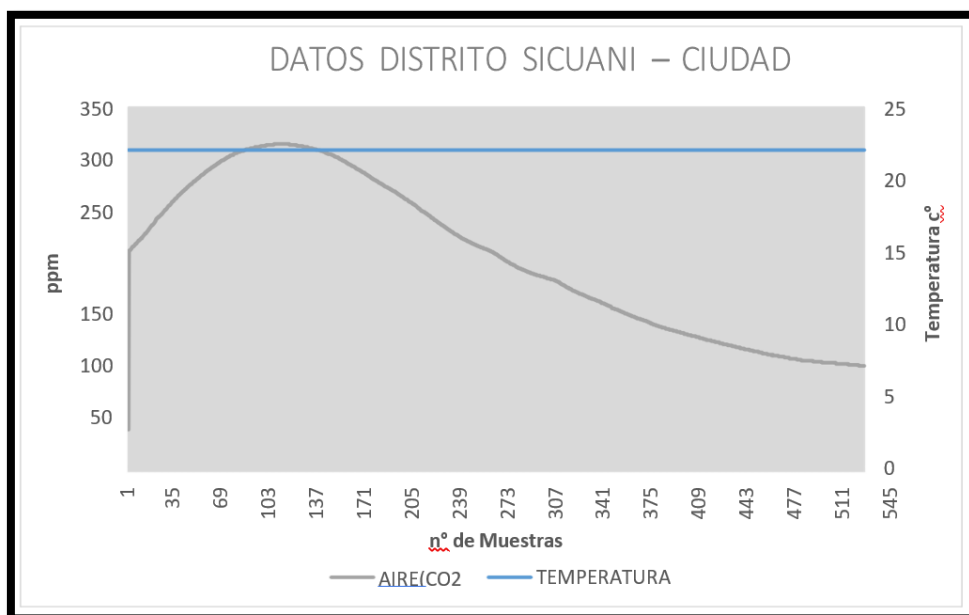


Figura 113. Resultados de temperatura y CO2 en distrito de Sicuani.

Fuente: Elaboración propia

4.9. Costos del desarrollo del diseño y Construcción

se describe en las siguientes **tablas 19,20,21 y 22** los componentes y gastos generales usados en todo el proyecto, detallando la forma de adquisición y el costo, y la **tabla 23** costo total.

Tabla 19. Costo para la placa electrónica de control

Fuente: Elaboración propia

Recurso	Descripción	Fuente	Cantidad	Precio	Monto s/
Placa Electrónica de Control	LEDs	Tienda	5	4	20
	Pulsadores	Tienda	1	3	3
	Atmega 2560	Tienda	1	30	30
	Cables	Tienda	4	3	12
	Estaño	Tienda	1	2	2
	Borneras	Tienda	2	4	8
	Resistencias 220	Tienda	10	0.2	2
	Regulador 3.3v	Tienda	1	2	2
	Regulador 5v 7805	Tienda	1	2	2
	Espadines Macho	Tienda	2	1	2
	Espadines Hembra	Tienda	2	1	2
	Módulo NRF24L01 2.4 GHZ	Tienda	1	10	10
	Batería LIPO 11.1V 5000 MAH	Tienda	1	185	185
	Condensador de 10uf	Tienda	2	1	2
	regulador Step dow	Tienda	1	30	30
	Placa -china	Tienda	1	15	15
	Cable usb - interfaz	Tienda	1	15	15
SUBTOTAL					S/ 342,00

Tabla 20. Costo para la placa Estructura.

Fuente: Elaboración propia

Recurso	Descripción	Fuente	Cantidad	Precio	Monto s/
Estructura	Pernos 5mm x2" 1/2	Tienda	8	1	8
	Pernos 3mm x 1"	Tienda	72	0.3	21.6
	Tuercas para Pernos 1"	Tienda	72	0.1	7.2
	Tuercas para Pernos 2" 1/2	Tienda	8	0.1	0.8
	Rodamientos Mz10	Tienda	18	2	36
	Impresión 3D	Empresa	1	350	350
	Servos Tower pro-mg995	Tienda	18	25	450
SUBTOTAL					S/ 844,00

Tabla 21. Costo para la placa módulo de sensado.

Fuente: Elaboración propia

Recurso	Descripción	Fuente	Cantidad	Precio	Monto s/
Módulo Sensado	Arduino Nano	Tienda	1	18	18
	Regulador 7805 5V	Tienda	1	1	1
	Placa perforada	Tienda	1	1	1
	Sensor DHT22 Temperatura	Tienda	1	10	10
	sensor MQ-135 CO2	Tienda	1	15	15
	Batería Lipo Turnigy 7.4V	Tienda	1	50	50
	Modulo lector Micro SD Arduino	Tienda	1	15	15
	Micro SD 2GB	Tienda	1	25	25
SUBTOTAL					S/ 135,00

Tabla 22. Costo para la placa del mando de control.

Fuente: Elaboración propia

Recurso	Descripción	Fuente	Cantidad	Precio	Monto s/
Mando De Control	Batería Lipo Turnigy 7.4V	Tienda	1	50	50
	Módulo NRF24L01 Receptor	Tienda	1	22	22
	Arduino Nano	Tienda	1	18	18
	joystick arduino	Tienda	1	8	8
	Pulsadores	Tienda	4	1	4
	resistencia de 1kΩ	Tienda	8	0.1	0.8
	interrupor	Tienda	1	1	1
	Placa perfo 10x10cm	Tienda	1	2	2
SUBTOTAL					S/ 105,00

Tabla 23. Costo total.

Fuente: Elaboración propia

Recurso	Monto s/
Placa Electrónica de Control	342
Estructura	844
Módulo Sensado	135
Mando De Control	105
	S/
Total	1.426,00



Conclusiones

- Se implementó los resultados obtenidos en la caminata trípode y se puede apreciar un comportamiento similar a las descritas en el entorno virtual, también la comprobación de cinemática directa e inversa como se muestra en los resultados obtenidos con el software Matlab, de las cuales nos dará la trayectoria parabólica de cada extremidad del robot.
- Se implementó las fórmulas de matrices de rotación y traslación de manera analítica en la cinemática del cuerpo, se obtuvo buenos resultados con el entorno virtual tanto en la parte física y simulación.
- Al realizar el entorno virtual en processing nos ayudó a poder comprobar los resultados obtenidos de todos los cálculos realizados, incorporando las dimensiones exactas de diseño del robot y de las ecuaciones halladas, para luego poder implementar físicamente.
- El diseño final del robot en SolidWorks se pudo integrar en el programa processing mediante el software Blender ya que la lectura de processing es con extensión. “. Obj “y no con extensión “. SLDPRT “, lo cual se tuvo inconvenientes al exportar el diseño, pero se solucionó ese problema obteniendo buenos resultados.
- En el diseño mecánico, se realizó la metodología de diseño NORMA VDI 2221 lo que permitió elaborar el diseño, también en los resultados de análisis de esfuerzo, URES, VON MISES y ESTRN de las piezas analizadas del robot.

- Se implementó físicamente los sensores Mq-135 y DHT22 para CO₂ y temperatura respectivamente, obteniendo los resultados en el distrito de Sicuani una temperatura de 23° constante una variación de co₂ Max de 131 ppm y un Min de 70 ppm comparando con los resultados en el cerro jururo del distrito de Sicuani, 22° constante y una variación de co₂ Max de 314 ppm y un Min de 98 ppm, ambos con 1000 muestras en 100 milisegundos.



Recomendaciones

- Para optimizar y tener mayor precisión en los actuadores y accionamiento se puede adquirir servomotores de gran precisión como los servo Dynamixel son mucho más precisos y cada servo tiene la capacidad de controlar su velocidad, posición, entre otras variables, por lo que puede optimizar el funcionamiento del robot.
- En el ámbito de diseño físico del robot debe optarse que el material sea de mayor resistencia y menor peso, lo que se recomienda utilizar fibra de carbono, aluminio o policarbonato, para aplicaciones donde las condiciones son más críticas.
- Para futuros trabajos de investigación se pretende incorporar sensores en las extremidades del robot lo cual permitirán moverse en terrenos irregulares, esto no alterará la formulación del análisis lo cual tendría solo la variación en el eje z, también de un control de trayectorias y redes neuronales para que el robot tenga autonomía.

Referencias Bibliográficas

Afzali-Far, B., & Lidström, P. (2018). Analytical index of dynamic isotropy and its application to hexapods. *Precision Engineering*, 52(January), 242–248. <https://doi.org/10.1016/j.precisioneng.2018.01.001>

Agheli, M., Qu, L., & Nestinger, S. S. (2014). SHeRo: Scalable hexapod robot for maintenance, repair, and operations. *Robotics and Computer-Integrated Manufacturing*, 30(5), 478–488. <https://doi.org/10.1016/j.rcim.2014.03.008>

Akdağ, M., Karagülle, H., & Malgaca, L. (2012). An integrated approach for simulation of mechatronic systems applied to a hexapod robot. *Mathematics and Computers in Simulation*, 82(5), 818–835. <https://doi.org/10.1016/j.matcom.2011.12.002>

Belter, D., & Skrzypczyński, P. (2011). Integrated Motion Planning for a Hexapod Robot Walking on Rough Terrain. *IFAC Proceedings Volumes*, 44(1), 6918–6923. <https://doi.org/10.3182/20110828-6-IT-1002.02234>

Chen, J., Liu, Y., Zhao, J., Zhang, H., & Jin, H. (2014). Biomimetic Design and Optimal Swing of a Hexapod Robot Leg. *Journal of Bionic Engineering*, 11(1), 26–35. [https://doi.org/10.1016/S1672-6529\(14\)60017-2](https://doi.org/10.1016/S1672-6529(14)60017-2)

Cirillo, A., Cirillo, P., De Maria, G., Marino, A., Natale, C., & Pirozzi, S. (2017). Optimal custom design of both symmetric and unsymmetrical hexapod robots for aeronautics applications. *Robotics and Computer-Integrated Manufacturing*, 44, 1–16. <https://doi.org/10.1016/j.rcim.2016.06.002>

Cárdenas Verdugo, M. A. (2011). Diseño, Construcción y Control de un Robot hexápodo. ciudad universitaria, México d. f: universidad nacional autónoma de México.

Javier Fernández Fernández. (2016). Análisis, desarrollo y evaluación de modos de marchapara un robot hexápodo, universidad politécnica de Madrid

Deng, H., Xin, G., Zhong, G., & Mistry, M. (2017). Gait and trajectory rolling planning and control of hexapod robots for disaster rescue applications. *Robotics and Autonomous Systems*, 95, 13–24. <https://doi.org/10.1016/j.robot.2017.05.007>

Deng, H., Xin, G., Zhong, G., & Mistry, M. (2018). Object carrying of hexapod robots with integrated mechanism of leg and arm. *Robotics and Computer-Integrated Manufacturing*, 54(April 2017), 145–155. <https://doi.org/10.1016/j.rcim.2017.11.014>

Ding, X., Wang, Z., Rovetta, A., & Zhu, J. M. (2010). Locomotion Analysis of Hexapod Robot. In *Climbing and Walking Robots* (Issue March). InTech. <https://doi.org/10.5772/8822>

Evangelista, G. (2014). Design and modeling of a mobile research platform based on hexapod robot with embedded system and interactive control. *2014 19th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 294–299. <https://doi.org/10.1109/MMAR.2014.6957367>

A. Rentería. (2001). *Robotica Industrial: Fundamentos y aplicaciones*. McGraw-Hill Interamericana.

Antonio, C. J. (Noviembre de 2012). Tesis. *Brazo Robotico Planar con dos grados de Libertad*. Mexico, México.

B. Kennedy, A. O. (2005). The Lemur II - Class Robots for Inspection and Maintenance of Orbital Structures: A System Description. *Proceedings of the 8th International Conference on Climbing and Walking Robots*, (págs. 1069-1076). Reino Unido.

Bares, J. E. (17 de Enero de 2012). *cyberneticzoo.com*. Obtenido de <http://cyberneticzoo.com/walking-machines/1992-4-dante-dante-ii-john-e-bares-william-red-whittaker-american/>

Bessonov, A. (1976). *Choice of Geometric Parameters of Walking Machines*.

Brooks, R. A. (1989). Fast, Cheap and out of control: A robot invasion of the solar system. *Journal of the British Interplanetary Society*, 478-485.

C. Georgiadis, A. G. (2004). E. AQUA: An Aquatic Walking Robot. *Proceedings of the IEEE/RST International Conference on Intelligent Robots and Systems*. Sendai.

Espen Knoop, A. C. (Abril de 2017). Hypocycloid Mechanism for Efficient Bioinspired Robotic Gaits.

ETH Zurich. (2016). *ANYbotics*. Obtenido de <https://www.anybotics.com/anymal-legged-robot/>

Huffman, C. (26 de Junio de 2003). Archytas. (E. N. Zalta, Ed.) *Stanford Encyclopedia of Philosophy*.

Jan Faigl, P. Č. (13 de Enero de 2019). Adaptive Locomotion Control of Hexapod Walking Robot for Traversing Rough Terrains with Position Feedback Only. Praga, Praga, República Checa.

K. Nonami, Q. J. (2001). Humanitarian Mine Detection Six-Legged Walking Robot COMET II with Two Manipulators. *Proceedings of the 4th International Conference on Climbing and Walking Robots* (págs. 989-996). Alemania: Karlsruhe.

Kosow, I. L. (1993). *Máquinas Eléctricas y transformadores*. Pearson Education.

Ludovico Minati, M. F. (26 de Enero de 2018). Versatile Locomotion Control of a Hexapod Robot Using a Hierarchical Network of Nonlinear Oscillator Circuits. Yokohama, Tokio, Japón.

Martinez, N. (28 de Mayo de 2003). Robot Hexápodo. Catalunya, España.

Murray, R. M. (1994). *A Mathematical Introduction to Robotic Manipulation*.

November, C. (16 de Junio de 2010). *Hipertextual*. Obtenido de <https://hipertextual.com/2010/06/rur-robots-universales-de-rossum>

Programa Interactivo Tutorial sobre Robotica. (Abril de 2033).

Roennau, A. (2013). Lauron V: Optimized Leg Configuration for the Design of a Bio-inspired Walking Robot. *Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, (págs. 563-571). Sidney.

Santos, P. G. (2006). An Introduction to the control of Four-Legged Robots. *Springer-Verlag*.

Showalter, M. (2009). Hexapod Work Space and Walking Algorithm.
Verlag:Saarbrücken, 20-34.

SRI International. (s.f.). Obtenido de <https://www.sri.com/work/timeline-innovation/timeline.php?timeline=computing-digital#!&innovation=shakey-the-robot>

Strimpel, O. (25 de Diciembre de 2009). *cyberneticzoo.com*. Obtenido de <http://cyberneticzoo.com/cyberneticanimals/1960-standford-cart-american/>

T. Hause, K. B. (2008). Motion Planning for legged Robots on Varied Terrain. *J. Robot.*

U. Saranlı, M. B. (2001). A simple and Highly Mobile Hexapod Robot. *J.Robot.*

Valero, R. (2011). Cirugía Robótica: Historia e impacto en la enseñanza. *Scielo*.

2014 Cyberbotics Ltd. (2014). *Webots User Guide release 7.4.3*.
www.cyberbotics.com: Cyberbotics©.

Arredondo, R. (2006). *Design and simulation of locomotion of self-organising modular robots for adaptive furniture. Master's thesis*. Swiss Federal Institute of Technology Lausanne:Biologically Inspired Robotics Group.

Chih-Han, Y., & Radhika, N. (2009). *Engineering Self-Adaptive Modular Robotics: A Bio-Inspired Approach*. Cambridge, USA: The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Choque Aspiazu, G. (2009). *Robótica modular reconfigurable*. LA PAZ-BOLIVIA: www.eldiario.net.

D. Hancher, M., & S. Hornby, G. (2006). *A Modular Robotic System with Applications to Space Exploration*. California: 2nd IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT'06).

D. Hancher, M., & S. Hornby, G. (2007). *Evolving Quadruped Gaits with a Heterogeneous Modular Robotic System*. NASA Ames Research Center, University of California Santa Cruz: 2007 IEEE Congress on Evolutionary Computation (CEC 2007).

De León Melgarejo, L. A. (2010). *Generación de patrones de marcha de un hexápodo por CPG's mediante autómatas de Mealy*. Mexico D.F: Instituto Politécnico Nacional.

Ekawahyu, S., Jianing, L., Alvarado Rayo, Y., Peck, A. M., Montenegro, J., Gonyea, M., & Valdastri, P. (2016). *An Affordable Modular Robotic Kit for Integrated Science, Technology Engineering, and Math Education*. IEEE robotics & automation magazine.

Electronics, P. R. (n.d.). *Pololu Robotic & Electronics*. Retrieved from <https://www.pololu.com/product/2858>

Escobar Olarte, J. M., & Silva Ortega, J. D. (2011). *Diseño de sistema electrónico jerárquico mediante comandos para robot hexápodo*. Bogotá doc. Colombia: pontificia universidad javeriana . facultad de ingeniería.

Feczko, J., Manka, M., Krol, P., Giergiel, M., Uhl, T., & Pietrzyk, A. (2015). *Review of the modular self reconfigurable robotic systems*. Poznan,Polonia: Proceedings of the 10th International Workshop on Robot Motion and Control, Poznan University of Technology.

Gonzáles Gómez, J. (2008). *robótica modular y locomoción: aplicación y robots ápodos*. Madrid, España: universidad autónoma de madrid escuela politécnica superior.

Gonzáles Gómez, J. (2009). *Robótica Modula y locomoción*. Albacete-España: Robótica Móvil: Una apuesta de futuro. Curso de Verano.

Hurtado Herasco, C. A., & Rubiano Montaña, O. G. (2013). *Diseño y simulación de un robot modular reconfigurable*. Bogotá, Colombia: Universidad Militar Nueva Granada.

Jie, Z., Xindan, C., Yanhe, Z., & Shufeng, T. (2011). *A New Self-Reconfigurable Modular RoboticSystem UBot: Multi-mode locomotion and Self-reconfiguration*. Shanghai,China: 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center.

Kamimura, A., Kurokawa, H., Yoshida, E., Murata, S., Kokaji, S., & Tomita, K. (2005). *Diseño Automático de locomoción y experimentos para un sistema robótico modular*. iee/asme transactions on mechatronics, vol.10.

Lancheros Guzmán, P. N., & Sanabria Galvis, L. B. (2016). *Simulación e Implementación de Movimientos para Sistema Robótico Modular Considerando Diferentes Configuraciones*. Bogotá-Colombia: Universidad Militar Nueva Granada.

Limitado, P. P. (n.d.). *Tdrobotica.co*. Retrieved from <http://tdrobotica.co/servo-mini-digital-giro-limitado-hd-1810mg-39-kg-cm/523.html>

Rai, V., Correll, N., & Anne van Rossum. (2011). *Self-Assembly of Modular Robots from Finite Number of Modules Using Graph Grammars*. San Francisco, CA, USA: IEEE/RSJ International Conference on Intelligent Robots and Systems.

Rodríguez Calderón, J. P., Ramos Parra, M. F., & Peña Giraldo, M. V. (2015). *Simulación de un Robot Hexápodo Bioinspirado en el Tenebrio*. Bogotá, Colombia: Universidad Libre.

Russo, S., Harada, K., Ranzani, T., Manfredi, L., Stefanini, C., Menciassi, A., & Dario, P. (2013). *Design of a Robotic Module for Autonomous Exploration and Multimode Locomotion*. *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, december 2013.

Sedeño Bustos, E. (2011). *Locomoción de un Robot Cuadrúpedo: Un Enfoque a Celdas Neuronales Analógicas*. Cuernavaca, Morelos, México: Centro Nacional de Investigación y Desarrollo Tecnológico.

Van Tung, L., & Trung Dung, N. (2014). *Virtual Pheromone based Information Foraging in Modular Robotics*. Kuala Lumpur, Malaysia: The 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2014).

Vonásek, V., Neumann, S., Oertel, D., & Worn, H. (2015). *Online Motion Planning for Failure Recovery of Modular Robotic Systems*. Washington, U.S.A: 2015 IEEE International Conference on Robotics and Automation (ICRA) Washington State Convention Center Seattle.

Weng, J. (2012). *Creación del Modelo del Robot Humanoide Teo para el Simulador WEBOTS*. Madrid, España: Universidad Carlos III de Madrid escuela politécnica ~~apir~~

Worldwide, P. E. (n.d.). Retrieved from
<https://www.pjrc.com/teensy/teensy31.html#specs>

Zhang, H., Gonzales Gomez, j., Chen, S. Y., & Zhang, W. (2008). *Embedded Intelligent Capability of a Modular Robotic System*. Bangkok, Thailand: IEEE, International Conference on Robotics and Biomimetics.

Zykov, V., Mytilinaios, E., Desnoyer, M., & Lipson, H. (2007). *IEEE Transactions on robotics*, vol. 23, no. 2, . ieee.

G.Cuaya (2007), “*Procesos de Decisión de Markov Aplicados a la Locomoción de Robots Hexápodos*”.

Matlab. (2017). El lenguaje del cálculo técnico.
Recuperado de <https://es.mathworks.com/products/matlab.html>

Definición de robótica - RAE (2008). definición
https://es.wikipedia.org/wiki/Rob%C3%B3tica#cite_note-3

Definición de robótica (2008).
https://es.wikipedia.org/wiki/Rob%C3%B3tica#cite_note-3

Industry Spotlight: Robotics from Monster Career Advice (2007).
https://es.wikipedia.org/wiki/Rob%C3%B3tica#cite_note-3

Reas, Casey; Fry, Ben (2014). *Processing: A Programming Handbook for Visual Designers and Artists (en inglés)*. The MIT Press. ISBN 978-0262028288.
Consultado el 1 de febrero de 2016.

Arduino announces FPGA board, ATmega4809 in Uno Wi-Fi mk2, cloud-based IDE and IoT hardware» (en inglés británico). 18 de mayo de 2018.
Consultado el 14 de junio de 2018.

Solidworks Company History 23 may 2021 en Wikipedia

<https://es.wikipedia.org/wiki/SolidWorks>

sparkfun <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT2>

2.pdf, Consultado octubre 2019

llamas, L. (16 de octubre de 2016). LEER Y ESCRIBIR EN UNA TARJETA SD O MICRO SD CON ARDUINO. Obtenido de <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Hernández, L. d. (2020). Processing, el lenguaje para gráficos. Obtenido de <https://programarfacil.com/podcast/80-processing-lenguaje-para-graficos/>

iberotecno. (21 de febrero de 2021). El sensor de CO2 MQ-135 analizado al detalle. Obtenido de <https://iberotecno.com/tienda/blog/electronica/el-sensor-de-co2-mq-135-analizado-al-detalle>

llamas, L. (16 de octubre de 2016). LEER Y ESCRIBIR EN UNA TARJETA SD O MICRO SD CON ARDUINO. Obtenido de <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>

Robots Didácticos. (25 de diciembre de 2019). Arduino: Comunicación inalámbrica con NRF24L01. Obtenido de <http://robots-argentina.com.ar/didactica/arduino-comunicacion-inalambrica-con-nrf24l01/>

iberotecno. (21 de febrero de 2021). El sensor de CO2 MQ-135 analizado al detalle. Obtenido de <https://iberotecno.com/tienda/blog/electronica/el-sensor-de-co2-mq-135-analizado-al-detalle>

Caicedo, J. M. (octubre de 2015). Navegación de un robot móvil sobre terreno irregular con contacto de su brazo con el suelo.

Craig J John, J. (2006). *Robótica. México: pearson education.* . Obtenido de https://es.wikipedia.org/wiki/Cinem%C3%A1tica_directa

Goering, R. (9 de setiembre de 2007). *Matlab edges closer to electronic design automation world.* Obtenido de <https://es.wikipedia.org/wiki/MATLAB>

Lazo Pinto, A. A. (2018). Diseño e implementación de un robot paralelo tipo delta de 3 grados de libertad controlado por medio de lógica difusa para la clasificación de objetos”. Arequipsa, Arequipa, Perú.

Olachea, O. (9 de agosto de 2013). *7 pasos para el proceso de un buen diseño.* Obtenido de <https://www.paredro.com/7-pasos-para-el-proceso-de-un-buen-diseno/>.

Chica, M. R. (2018). *StuDocu.* Obtenido de <https://www.studocu.com/es/document/universidad-de-malaga/metodologia-del-diseno/tema-01-introduccion-apuntes-1/9856648>

DÍAZ, F. V. (11 de DICIEMBRE de 2017). *Scribd* . Obtenido de <https://es.scribd.com/document/366869321/Normas-VDI-2222-2221-Anexos>

ANEXOS

Anexo 1: FABRICANTES Y PATENTES

Anexo 2: CODIGO MATLAB

Anexo 3: CODIGO PROCESSING

Anexo 4: CODIGO ARDUINO

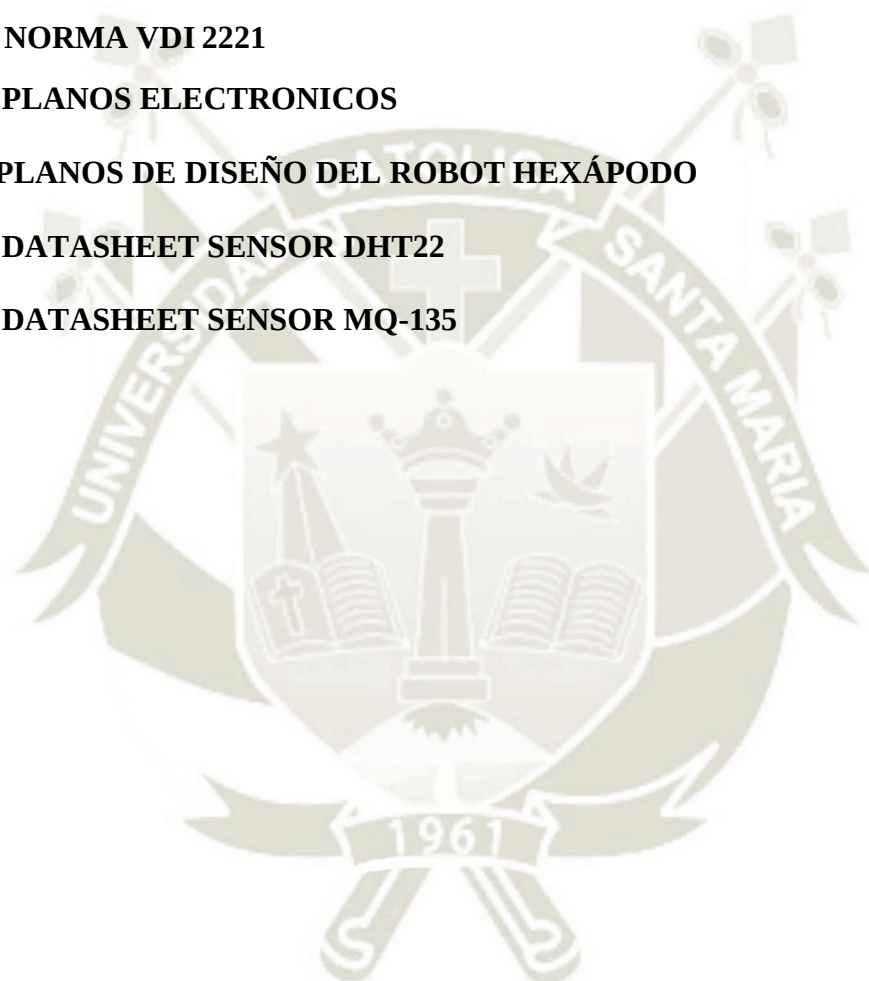
Anexo 5: NORMA VDI 2221

Anexo 6: PLANOS ELECTRONICOS

Anexo 7: PLANOS DE DISEÑO DEL ROBOT HEXÁPODO

Anexo 8: DATASHEET SENSOR DHT22

Anexo 9: DATASHEET SENSOR MQ-135



Anexo 1: FABRICANTES Y PATENTES

INFORMACION OBTENIDA DE FABRICANTES

- ArcBotics es uno de los fabricantes de robots hexápodos que pueden ser usados para aprender, no son fabricados de forma industrial.

Hexapod Robot Hexy: El Hexapod Robot Hexy - Blue es un pequeño robot de seis patas, y es el primer kit de robótica abierta de ArcBotics. Es un kit de robot hexápodo totalmente articulado que le brinda una plataforma para explorar la robótica avanzada, la cinemática y la programación. Hexy tiene seis patas, 19 servomotores y es impulsado por Arduino. Este robot es ofrecido para estudiantes de 15 años o más que exploran robótica avanzada, cinemática y entusiastas del bricolaje que buscan una plataforma de código abierto.

Se debe tener en cuenta que este robot no viene armado, cuenta con una guía dada por ArcBotics para su armado casero.

Este robot cuenta además con un soporte de batería para pilas 2A, un sensor ultrasónico de distancia (ojos del robot), y un módulo de bluetooth. Todas sus piezas son hechas de acrílico, cortadas con láser, además que se vende también archivos en solidWorks acerca de su estructura, archivos de programación y electrónicos; su costo final es de \$212.46.

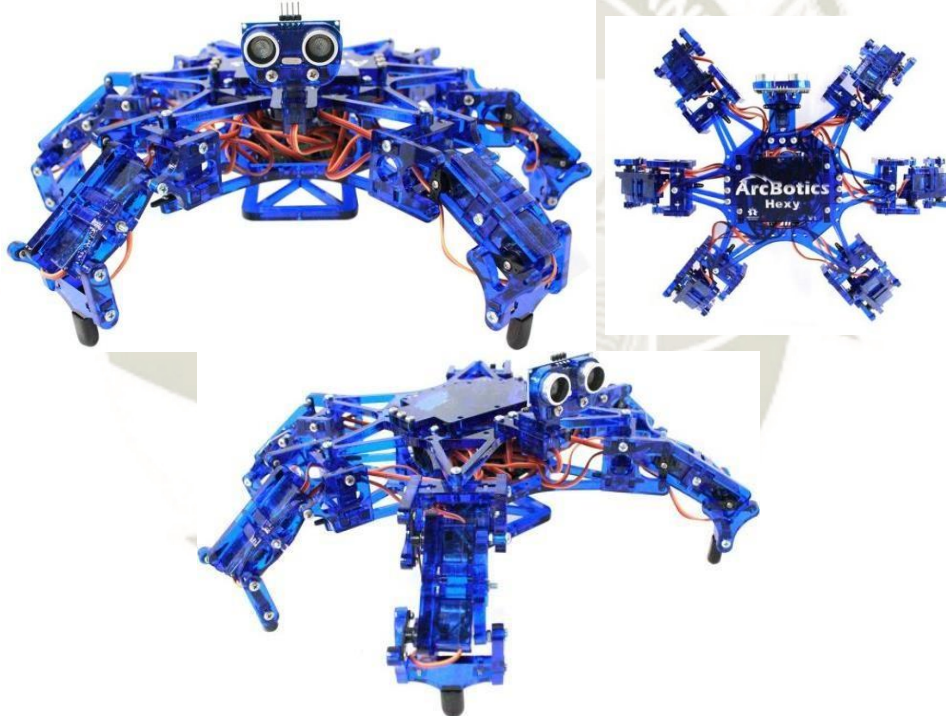


Figura 22. Robot Hexy
Recuperado de Arcbotics Shop

- DFRobot Spider Robot Kit: El cual es solo la estructura de un robot hexápodo con forma de araña que incluye una batería con el precio de \$5.90
 - Voltaje de funcionamiento 1.5V
 - Corriente de funcionamiento 180mA
 - Motor de accionamiento 130RPM 11000r / min (por motor)
 - Tamaño: 18x24x6 cm

Se necesita:

 - Romeo-All in one Controller (Arduino Compatible Atmega 328)
 - DFRduino UNO R3

- Sensor de distancia Sharp GP2Y0A21 (10-80cm)
- Control remoto de rango completo con 2 bandas.
- tirador de pistola de juguete



Figura 23. Robot Spider Kit
Recuperado de ArcBotics Shop

- RobotShop ofrece fabricar y distribuir kits de robot educativo, en su página se pueden encontrar diferentes proyectos los cuales no sólo ofrecen comprar el robot sino también los códigos de programación, archivos con la estructura, datasheets, guías y demás para poder ensamblar y poner en funcionamiento estos robots.
- Lynxmotion A-Pod Hexapod Robot Kit (Hardware Only)
Está inspirado en los insectos. Las patas en ángulo proporcionan un rango adicional de movimiento. El diseño de las tres patas DOF (grado de libertad) significa que el robot puede caminar en cualquier dirección. El robot ha sido diseñado para usar 18 servos HS-645 para las piernas y 7 servos adicionales para la cabeza y la cola. El robot está fabricado con soportes de servo erectores de aluminio de alta calidad y corte de precisión de PVC.
 - Servo control de movimiento: bucle cerrado local
 - Dirección: diferencial
 - Número de piernas: 6
 - Velocidad de movimiento: 10 in/s
 - Distancia al suelo: hasta 7.50 in
 Requiere (se vende por separado)
 - 25 x Servos estándar HS-645MG (133 oz. in.)
 - Microcontrolador BotBoarduino.
 - Servo controlador SSC-32U (incluye cable USB)
 - Batería de 6V
 - Controlador inalámbrico PS2
 Valor total: \$303.20



Figura 24. A-Pod Hexapod
Recuperado de Lynxmotion

- EZ-Robot Revolution Six Wifi Hexapod:
Con 6 patas y 12 grados de libertad. Las articulaciones son accionadas por 12 servomotores de servicio pesado para habilidades dinámicas únicas de movimiento. Las juntas extraíbles EZ-Bit y la cámara montada en la parte superior mejoran la interactividad con el mundo.
- Procesamiento a 32MHz a 200MHz (procesador Cortex M3 ARM de 120MHz a 32MHz y microchip de 80MHz PIC32)
- Tamaño pequeño (2.1 "x 2.2" = 36x36x18)
- Peso: 2.600 kg
- Conectividad WiFi (Ad-Hoc / Infraestructura / WEP / WPA / WPA2)
- Servidor web incorporado
- Fuente de alimentación de conmutación digital que ahorra energía
- Protección de fusible reajutable
- Protección contra polaridad inversa
- 1 x Altavoz Amplificado para Discurso y Música
- 8 puertos analógicos tolerantes (ADC) de 5 voltios
- Puertos digitales tolerantes de 24 x 5 voltios (servos, PWM y más)
- Hasta 73 Servos (combinando PWM y Dynamixel)
- Soporte Servo Dynamixel
- 3 x puerto i2c
- 1 x puerto para cámara de video EZ-Robot
- 3 x puertos UART de alta velocidad
- Batería y monitor de temperatura
- Velocidad de operación: 0.24sec / 60degree (7.4V)
- Par de apriete: 15.0kg-cm / 529 oz-in. (7.4V)

- Rango de temperatura: -20C a + 60C
- Voltaje de operación: 4.8V a 7.4V
- Ferrita de 3 polos
- Doble rodamiento de bolas de metal.
- Todos los engranajes de metal.
- Longitud del cable conector: 32cm
- Futaba 3f (25 dientes) para la conducción de estrías / ejes
- Spline de 6.2mm (incluyendo dientes)



*Figura 25. E-Z Robot Revolution
Recuperado de RobotShop*

- **Lynxmotion CH3-R Hexapod Robot Kit (BotBoarduino) w/ HS-645 Servo Upgrade:**

Incluye todo lo que necesita para construir un robot para controlar el uso del CH3 actualizado con los servos HS-645 y un controlador BotBoarduino. La simetría de su cuerpo redondo hace que este andador de hexápodos sea único. El diseño de las tres patas DOF (grado de libertad) proporciona la flexibilidad necesaria para caminar en cualquier dirección. El robot utiliza 18 servos HS-475 / HS-485 para las piernas. El kit combinado incluye todo lo que necesita para hacer un robot funcional, excepto las baterías. Esta es una de las ofertas de robots para caminar más sofisticadas.

- CH3 actualizado con servos HS-645 y un controlador BotBoarduino
- Proporciona la flexibilidad requerida para caminar en cualquier dirección. •
- Componentes estructurales Lexan cortados con láser ultra resistentes
- Programas prescritos disponibles
- Peso: 4.60 kg
- Cargador: NiCad & Ni-MH Universal Smart Charger
- Baterías: 6.0 V, Ni-MH 2800 mAh



*Figura 26. CH3-R Hexapod Robot Kit
Recuperado de RobotShop*

PATENTES

Patente: CN202115613U de CHINA

Inventor: 关楠楠, 刘岩, 华磊, 宋天宇, 李婉宁, 金江, 陈宏伟

Año: 2011

Limitaciones:

- Volumen del Robot: 45 cm x 30 cm x 22 cm
- Material del Robot: plástico 6 mm de espesor
- Número de grados de libertad en las patas: 3 GDL
- Número de Actuadores: 6 servomotores
- Tipo de Actuadores: no especificado
- Peso: no especificado
- Sensores: dos sensores infrarojo, una cámara y su sujetador
- Control: Coordina sus patas automáticamente para cruzar caminos muy complicados.

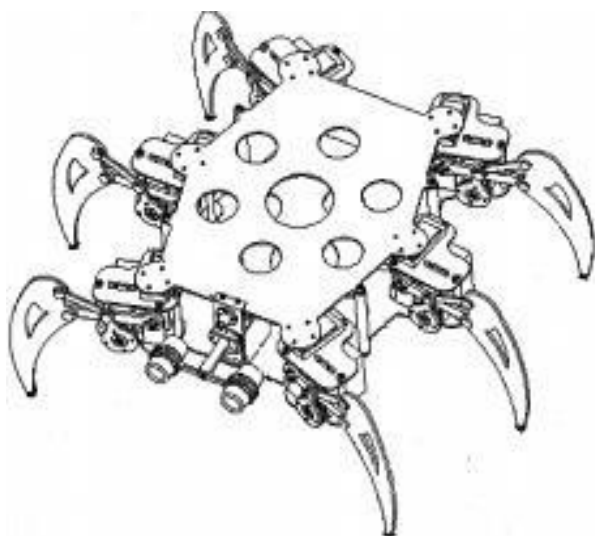


Figura 27. Cuerpo Entero
Recuperado de China Patente n° CN202115613U, 2011

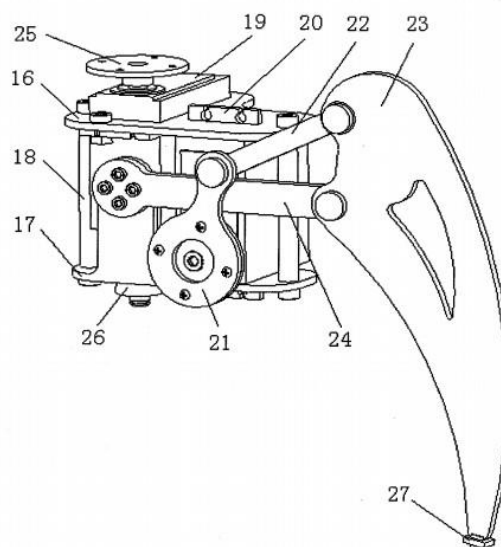


Figura 28. Detalle de las patas
Recuperado de China Patente n° CN202115613U, 2011



Patente: CN107140052A de CHINA

Inventor: 丁希仑, 潘希祥, 徐坤, 郑羿, 齐静, 陈佳伟

Año: 2017

Limitaciones:

- Volumen del Robot: no especificado
- Material del Robot: aluminio
- Número de grados de libertad en las patas: 3 GDL
- Número de Actuadores: 12 servomotores, 12 amortiguadores, 2 motores DC
- Tipo de Actuadores: no especificado
- Peso: no especificado
- Sensores: dos sensores infrarojo, una cámara y su sujetador
- Control: cuenta con una articulación de pierna, rodilla, talón y cadera; que le permiten al robot tener una libertad de movimiento de 0 a 25°. Además, dispone de un sistema de amortiguamiento, equipado con ruedas para un desplazamiento más rápido en terreno plano.

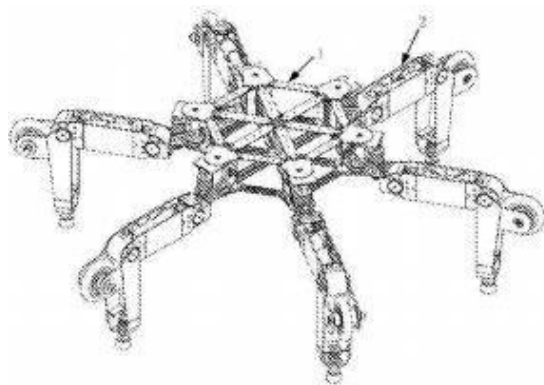


Figura 29. Cuerpo entero parado.
Recuperado de China Patente n° CN107140052A,
2017

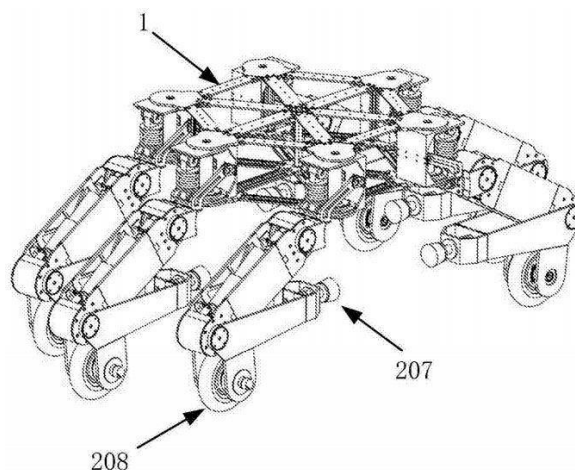


Figura 30. Cuerpo entero de ruedas.
Recuperado de China Patente n° CN107140052A,
2017

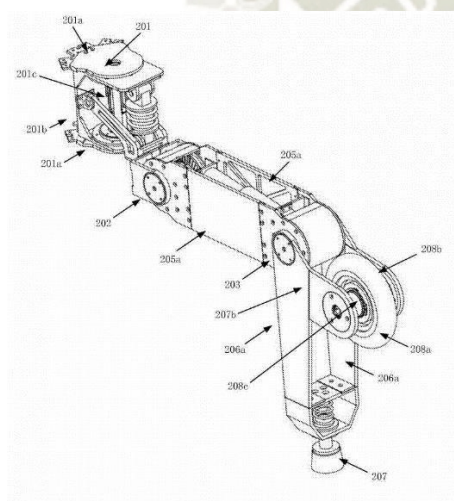


Figura 31. Detalle PATA.

Recuperado de China Patente n°
CN107140052A, 2017

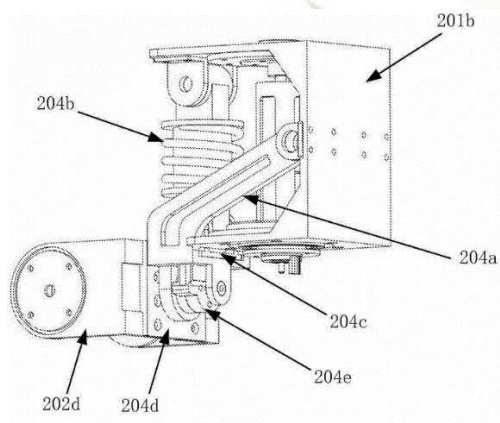


Figura 32. Detalle CADERA.

Recuperado de China Patente n°
CN107140052A, 2017

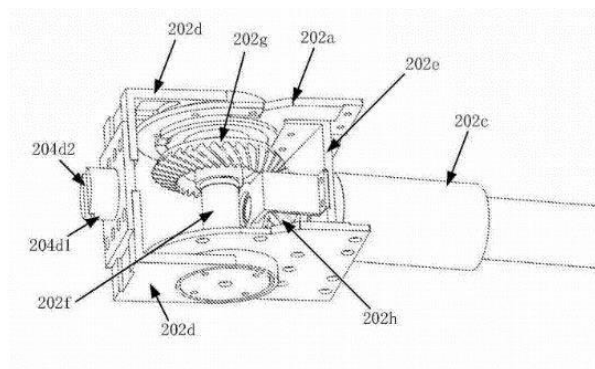


Figura 33. Detalle RODILLA.

Recuperado de China Patente n°
CN107140052A, 2017

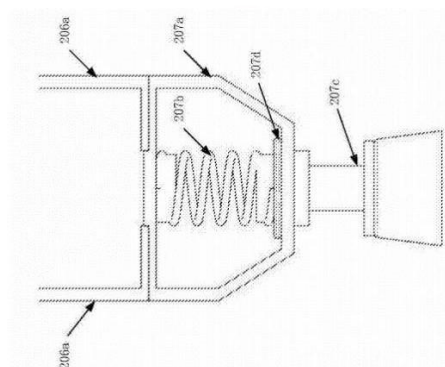


Figura 34. Detalle TALON.

Recuperado de China Patente n°
CN107140052A, 2017

ALCANCES Y LIMITACIONES

- Geometría:
 - Torax: hal 70x100x130 mm³
 - Conjunto: 250x750x750 mm³
- Estructura: ABS (aplicaciones industriales)
- Cinemática: Directa e Inversa.
 - 3 gld por pata.
- Fabricación: Impresora 3d - Diseño Inventor
- Energía: Batería de 10000mAh
- Montaje: Fácil ensamblaje
- Transporte: Portable
- Uso: Adquisición de datos de calidad de aire para terrenos inaccesibles
- Mantenimiento:
 - Limpia contactos
 - Polvo
 - Intercambio de piezas
- Controlador: Arduino Nano
- Módulos:
 - Controlador de servos
 - Sensor de gases
 - Regulador de Voltaje
 - Módulo SD
- Limitación: Terreno a acceder visible

Anexo 2: CODIGO MATLAB

CIENEMATICA DIRECTA E INVERSA

```
% LIMPIAR DATOS, GRAFICAS, VARIABLES
close all, clc, clear all

% ROBOT HEXAPODO
% CINEMÁTICA INVERSA - PATA

% MORFOLOGÍA DE UNA PATA
L1=7.3; %LONGITUD 1 - [cm]
L2=9; %LONGITUD 2 - [cm]
L3=18.3; %LONGITUD 3 - [cm]

% ENTRADA DE VALORES ANGULARES PRUEBA
q1=input('ANGULO 1: ');
q2=input('ANGULO 2: ');
q3=input('ANGULO 3: ');
des=33.93;

% CINEMATICA DIRECTA ECUACIONES
x=L1*cosd(q1)+L2*cosd(q1)*cosd(q2)+L3*cosd(q1)*cosd(q2+q3-
des);
y=L1*sind(q1)+L2*sind(q1)*cosd(q2)+L3*sind(q1)*cosd(q2+q3-
des);
z=L2*sind(q2)+L3*sind(q2+q3-des);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x=-7;
% y=-19;
% z=-7;

% POSICION DEL EFECTOR FINAL [X Y Z]
fprintf('\nx: %f (cm) ',x)
fprintf('\ny: %f (cm) ',y)
fprintf('\nz: %f (cm) \n',z)

% ECUACIONES DE CINEMÁTICA INVERSA
L = sqrt(x^2+y^2);
ML = sqrt((L-L1)^2+z^2);

% PRIMER GRADO DE LIBERTAD
q1 = atan2d(y,x);

%SEGUNDO GRADO DE LIBERTAD
% a1 = atan2d((L-L1),z);
```



```

a1=atand(z/(L-L1));
a2 = acosd((L3^2-L2^2-ML^2)/(-2*L2*ML));
if(z>=0)
A = a1-a2;
elseif(z<0)
A = a1+a2;
end
% q2 = 90-(A);
q2 =A;

%TERCER GRADO DE LIBERTAD
b1 = acosd((ML^2-L2^2-L3^2)/(-2*L2*L3));
if(z>=0)
q3 = 180-b1+des;
elseif(z<0)
q3=b1-180+des;
end

disp(' ')
disp(' ')
disp('-----CINEMATICA INVERSA -----')
fprintf('\nq1: %f',q1)
fprintf('\nq2: %f',q2)
fprintf('\nq3: %f\n',q3)

% CINEMATICA DIRECTA ECUACIONES
x=L1*cosd(q1)+L2*cosd(q1)*cosd(q2)+L3*cosd(q1)*cosd(q2+q3-
des);
y=L1*sind(q1)+L2*sind(q1)*cosd(q2)+L3*sind(q1)*cosd(q2+q3-
des);
z=L2*sind(q2)+L3*sind(q2+q3-des);

% POSICION DEL EFECTOR FINAL [X Y Z]
fprintf('\nqx: %f(cm)',x)
fprintf('\ny: %f (cm)',y)
fprintf('\nz: %f (cm)\n',z)

```

CINEMATICA DEL CUERPO

```
% ROBOT HEXAPODO
% CINEMÁTICA INVERSA - CUERPO

% MORFOLOGÍA DE UNA PATA
L1=7.3; %LONGITUD 1 - [cm]
L2=9; %LONGITUD 2 - [cm]
L3=18.3; %LONGITUD 3 - [cm]

% MORFOLOGÍA CUERPO-PATA

% PATA 1
% b = 0;
% xof = 5.0;
% yof = 0;
% off = [xof yof 0]';

% PATA 2
% b = 0;
% xof = 35.76;
% yof = 0;
% off = [xof yof 0]';

% PATA 3
b = 45;
xof = 35.76;%%modificar
yof = 55;%%modiifcar
off = [xof yof 0]';

% ENTRADA DE VALORES ANGULARES PRUEBA
q1=input('ANGULO 1: ');
q2=input('ANGULO 2: ');
q3=input('ANGULO 3: ');

% CINEMATICA DIRECTA ECUACIONES
x=L1*cosd(q1)+L2*cosd(q1)*cosd(q2)+L3*cosd(q1)*cosd(q2+q3);
y=L1*sind(q1)+L2*sind(q1)*cosd(q2)+L3*sind(q1)*cosd(q2+q3);
z=L2*sind(q2)+L3*sind(q2+q3);

% POSICION DEL EFECTOR FINAL [X Y Z]
fprintf('\nx: %f(cm)',x)
fprintf('\ny: %f (cm)',y)
fprintf('\nz: %f (cm)\n',z)

P = [x y z]';
```

```
% ROTACIÓN YAW
t1 = input('YAW: ');

% POSICIÓN RESPECTO A EJES DE CUERPO
Pp = MR('z',b)*P

% % POSICIÓN RELATIVA RESPECTO AL CENTRO DE MASA
Pr = off+Pp

% DISTANCIA RELATIVA POSICION PATA - CENTRO DE MASA
% RF = sqrt(Pr(1)^2+Pr(2)^2)
%
% ANGULO POSICION RELATIVA
% t0 = atan2d(Pr(2),Pr(1))
%
% ANGULO DE GIRO COMPLETO
% ang = t1+t0;

% POSICIÓN FINAL LUEGO DE ROTACIÓN YAW
g = MR('z',-b)*Pr
Pf = MR('z',b+t1)*g

% DESFASE POSICIÓN RELATIVA FINAL E INICIAL
D = [Pf(1)-Pr(1) Pf(2)-Pr(2) Pf(3)-Pr(3)]'

% DIFERENCIA PARA RESTAR VALOR
Dm = MR('z',-b-t1)*D

% POSICIÓN DE PATA ABSOLUTA
Pabs = [P(1)+Dm(1) P(2)+Dm(2) P(3)+Dm(3)]'

x = Pabs(1);
y = Pabs(2);
z = Pabs(3);

% ECUACIONES DE CINEMÁTICA INVERSA
L = sqrt(x^2+y^2);
ML = sqrt((L-L1)^2+z^2);

% PRIMER GRADO DE LIBERTAD
q1 = atan2d(y,x);

%SEGUNDO GRADO DE LIBERTAD
a1 = atan2d((L-L1),z);
a2 = acosd((L3^2-L2^2-ML^2)/(-2*L2*ML));
if(z>=0)
A = a1+a2;
```



```
elseif(z<0)
A = a1-a2;
end
q2 = 90-(A);

%TERCER GRADO DE LIBERTAD
b1 = acosd((ML^2-L2^2-L3^2)/(-2*L2*L3));
if(z>=0)
q3 = 180-b1;
elseif(z<0)
q3=b1-180;
end

disp('-----CINEMATICA INVERSA -----')
fprintf('\nq1: %f',q1)
fprintf('\nq2: %f',q2)
fprintf('\nq3: %f\n',q3)

% CINEMATICA DIRECTA ECUACIONES
x=L1*cosd(q1)+L2*cosd(q1)*cosd(q2)+L3*cosd(q1)*cosd(q2+q3);
y=L1*sind(q1)+L2*sind(q1)*cosd(q2)+L3*sind(q1)*cosd(q2+q3);
z=L2*sind(q2)+L3*sind(q2+q3);

% POSICION DEL EFECTOR FINAL [X Y Z]
fprintf('\nx: %f(cm)',x)
fprintf('\ny: %f (cm)',y)
fprintf('\nz: %f (cm)\n',z)
```

TRAYECTORIA PARABÓLICA

```
clc, clear all, close all

%TIEMPO PARA COMPLETAR MOVIMIENTO
tiempo=3;
muestreo=0.1;

% COMPROBAR SI TRAYECTORIA ES POSIBLE
k=0;

% PARAMETROS DE TRAYECTORIA PARABÓLICA
y_mov = 4; %6
x_mov = 0;
z_mov = 3;
fx_mov = 0;

q3_d = 33.93;

% LIMITE ARTICULACIONES
q1_min = -70; %-65 / -20 / 25 / -65
q1_max = -20; %-25 / 20 / 65 / -25
q2_min = -25;
q2_max = 82;
q3_min = -75-q3_d;
q3_max = 15-q3_d;

%ECUACION DE UNA PARABOLA EN R3
%PUNTO INCIAL
disp('PUNTO INICIAL')
% PATA 1
x0=13.94;
y0=-13.94;
z0=-17.97;

% PATA2
% x0=24.88;
% y0=0;
% z0=-16.47;

% PATA 3
% x0=17.59;
% y0=17.59;
% z0=-16.47;

% PATA 4
% x0=17.59;
```

```
% y0=-17.59;
% z0=-16.47;

% PATA 5
% x0=24.88;
% y0=0;
% z0=-16.47;

% PATA 6
% x0=17.59;
% y0=17.59;
% z0=-16.47;

xx0 = [x0 y0 z0]'

%PUNTO INTERMEDIO
disp('PUNTO INTERMEDIO')
xi=x0+fx_mov;
yi=y0+y_mov/2;
zi=z0+z_mov;
xxi = [xi yi zi]'

%PUNTO FINAL
disp('PUNTO FINAL')
xf=xi-fx_mov;
yf=y0+y_mov;
zf=z0;
xxf = [xf yf zf]'

%ECUACIONES PARAMETRICAS DE TRAYECTORIA EN EL ESPACIO DE LA
TAREA
t=0:muestreo:tiempo;
[a1,b1,c1]=para(x0,xi,xf,tiempo);
x=c1+b1.*t+a1.*t.*(t);
[a2,b2,c2]=para(y0,yi,yf,tiempo);
y=c2+b2.*t+a2.*t.*(t);
[a3,b3,c3]=para(z0,zi,zf,tiempo);
z=c3+b3.*t+a3.*t.*(t);

%GRAFICA DE TRAYECTORIA ESPACIO DE LA TAREA
figure(1)
plot3(x,y,z)
xlabel('EJE X')
ylabel('EJE Y')
zlabel('EJE Z')
title('TRAYECTORIA')
grid on
```



```
figure(2)
subplot(3,1,1)
plot(t,x,'*');
title('POSICION X')
subplot(3,1,2)
plot(t,y,'*');
title('POSICION Y')
subplot(3,1,3)
plot(t,z,'*');
title('POSICION Z')

%CINEMATICA INVERSA
for i=1:1:size(x,2)
[q1(i),q2(i),q3(i)]=inv_hex(x(i),y(i),z(i));
if (q1(i)>=q1_min&&q1(i)<=q1_max&&q2(i)>=q2_min&&q2(i)<=q2_max&&q3(i)>=q3_min&&q3(i)<=q3_max)
else
    k=1;
end
end

if (k==0)

%GRAFICA DE TRAYECTORIA ESPACIO ARTICULAR
figure(3)
subplot(3,1,1)
plot(t,q1,'*')
title('ARTICULACION 1')
subplot(3,1,2)
plot(t,q2,'*')
title('ARTICULACION 2')
subplot(3,1,3)
plot(t,q3,'*')
title('ARTICULACION 3')

%ROBOT ESFERICO MORFOLOGIA
L1=7.3;
L2=9;
L3=18.3;

% PARAMETROS DE DH
L(1)=Link('revolute','d',0,'a',L1,'alpha',pi/2); %MATRIZ A01
L(2)=Link('revolute','d',0,'a',L2,'alpha',0); %MATRIZ A12
L(3)=Link('revolute','d',0,'a',L3,'alpha',0); %MATRIZ A23
P=SerialLink(L);

%POSICION EN DEGREES
```

```
q1f = q1;  
q2f = q2;  
q3f = q3;  
  
%CONVERSION A RADIANTES ANGULOS OBTENIDOS  
q1=degtorad(q1);  
q2=degtorad(q2);  
q3=degtorad(q3);  
  
%ANIMACION DE TRAYECTORIA  
figure(5)  
for i=1:1:size(x,2)  
P.plot([q1(i) q2(i) q3(i)], 'trail', '-')  
end  
  
end  
  
if(k==1)  
    disp('MOVIMIENTO NO POSIBLE')  
end
```



ESPACIO DE TRABAJO

```
% MORFOLOGÍA DE PATA
```

```
L1=7.3;
```

```
L2=9;
```

```
L3=18.3;
```

```
%TIEMPO PARA COMPLETAR MOVIMIENTO
```

```
h_1=3;
```

```
k=0;
```

```
muestreo=0.1;
```

```
tiempo=3;
```

```
% PARAMETROS DE TRAYECTORIA PARABÓLICA
```

```
y_mov = 5;
```

```
z_mov = 8;
```

```
% LIMITE ARTICULACIONES
```

```
q1_min = -20; %-65 / -20
```

```
q1_max = 20; %-25 / 20
```

```
q2_min = -30;
```

```
q2_max = 90;
```

```
q3_min = -90; %-90
```

```
q3_max = 0;
```

```
% VOLUMEN DE TRABAJO
```

```
figure(1)
```

```
for q1=q1_min:5:q1_max
```

```
for q2=q2_min:5:q2_max
```

```
for q3=q3_min:5:q3_max
```

```
x=L1*cosd(q1)+L2*cosd(q1)*cosd(q2)+L3*cosd(q1)*cosd(q2+q3);
```

```
y=L1*sind(q1)+L2*sind(q1)*cosd(q2)+L3*sind(q1)*cosd(q2+q3);
```

```
z=L2*sind(q2)+L3*sind(q2+q3);
```

```
plot3(x,y,z,'* g')
```

```
hold on
```

```
end
```

```
end
```

```
end
```

```
%ECUACION DE UNA PARABOLA EN R3
```

```
%PUNTO INCIAL
```

```
disp('PUNTO INICIAL')
```

```
x0=24.88;
```

```
y0=0;
```

```
z0=-16.47;
```

```
%PUNTO INTERMEDIO
```

```
disp('PUNTO INTERMEDIO')
```



```
xi=x0;  
yi=y0+y_mov/2;  
zi=z0+z_mov;
```

```
%PUNTO FINAL  
disp('PUNTO FINAL')  
xf=xi;  
yf=y0+y_mov;  
zf=z0;
```

```
%ECUACIONES PARAMETRICAS DE TRAYECTORIA EN EL ESPACIO DE LA  
TAREA  
t=0:muestreo:tiempo;  
[a1,b1,c1]=para(x0,xi,xf,tiempo);  
x=c1+b1.*t+a1.*t.*(t);  
[a2,b2,c2]=para(y0,yi,yf,tiempo);  
y=c2+b2.*t+a2.*t.*(t);  
[a3,b3,c3]=para(z0,zi,zf,tiempo);  
z=c3+b3.*t+a3.*t.*(t);  
  
plot3(x,y,z);
```



Anexo 3: CODIGO PROCESSING

GRAFICAS

//VARIABLES

boolean graph_on = false;

//VARIBLE DE ENTRADA

int cn = 0;

//BOTONES

boolean run_graph = false;

//LEG SAVE VARIABLES (HISTORIC)

int samples = 600;

float [][] q_l1 = new float [3][samples+1];

float [][] q_l2 = new float [3][samples+1];

float [][] q_l3 = new float [3][samples+1];

float [][] q_l4 = new float [3][samples+1];

float [][] q_l5 = new float [3][samples+1];

float [][] q_l6 = new float [3][samples+1];

float [] time_plot = new float[samples+1];

float [] time_sc1 = new float[samples+1];

float [] time_sc2 = new float[samples+1];

float [] time_sc3 = new float[samples+1];

float [] time_sc4 = new float[samples+1];

float [] time_sc5 = new float[samples+1];

float [] time_sc6 = new float[samples+1];

float [] g_max1 = {300, 300, 300, 300, 300, 300};

float [] g_min1 = {0, 0, 0, 0, 0, 0};

float [] g_min2 = {300, 300, 300, 300, 300, 300};

float [] g_max2 = {0, 0, 0, 0, 0, 0};

float [] g_min3 = {300, 300, 300, 300, 300, 300};

```
float [] g_max3 = {0, 0, 0, 0, 0, 0};
```

```
//GRAPH WINDOW
```

```
WINDOW_GRAPH GRAPH;
```

```
//TABLE EXCEL
```

```
Table q_table;
```

```
class WINDOW_GRAPH extends PApplet {
```

```
    WINDOW_GRAPH() {
```

```
        super();
```

```
        PApplet.runSketch(new String[] {this.getClass().getSimpleName()}, this);
```

```
        surface.setTitle("Gráfica Posiciones Articulares");
```

```
    }
```

```
    //void settings() {
```

```
        //size(300, 300, P3D);
```

```
    //}
```

```
    void setup() {
```

```
        //size(300, 300);
```

```
        frameRate(30);
```

```
        surface.setSize(1350, 630);
```

```
        surface.setLocation(displayWidth/2-680, displayHeight/2-340);
```

```
        surface.setAlwaysOnTop(true);
```

```
        surface.setAlwaysOnTop(false);
```

```
        rectMode(CORNER);
```

```
        background(240);
```

```
        for (int i=0; i <= samples; i++) {
```

```
            time_plot[i] = 0.01*i;
```



```
}

q_table = new Table();

}

void draw() {
    //BACKGROUND
    background(244);

    //TEXT CONFIGURATION
    textAlign(LEFT, CENTER);
    textSize(11);
    fill(0);

    //TITLE
    text("GRÁFICAS", 16, 16);

    fill(255, 255, 240);
    int jk = 1000;
    int jh = 590;
    rect(jk, jh-14, 300, 32);
    fill(0);
    text("Plotear", 1014, 590);
    text("Detener", 1104, 590);
    text("Exportar", 1200, 590);

    //BUTTONS
    fill(0);
    ellipse(1070, 592, 20, 20);
    ellipse(1166, 592, 20, 20);
    ellipse(1264, 592, 20, 20);
}
```

```

fill(255, 255, 0);
if (run_graph==false) {
    ellipse(1166, 592, 16, 16);
} else {
    ellipse(1070, 592, 16, 16);
}

//POSITION
//text("q1", 12, 134);
//text("q2", 12, 134+168);
//text("q3", 12, 134+168*2);

//GRAPH
//fill(255);
//int k = 40;
//int wth = 200;
//for (int j=0; j < 3; j++) {
//  for (int i=0; i < 6; i++) {
//    rect(k+wth*i+10*i, 84+110*j, wth, 100);
//  }
//}

//VARIABLES INPUT

//PLOT LEG 1
int in_posx = 50;
int sep = 220;
int zy = 72;

plot(q_temp[0][0], q_temp[1][0], q_temp[2][0], q_l1, time_sc1, cn,
in_posx+sep*0, zy, 0);

plot(q_temp[0][1], q_temp[1][1], q_temp[2][1], q_l2, time_sc2, cn,
in_posx+sep*1, zy, 1);

```

```
plot(q_temp[0][2], q_temp[1][2], q_temp[2][2], q_l3, time_sc3, cn,
in_posx+sep*2, zy, 2);
```

```
plot(q_temp[0][3], q_temp[1][3], q_temp[2][3], q_l4, time_sc4, cn,
in_posx+sep*3, zy, 3);
```

```
plot(q_temp[0][4], q_temp[1][4], q_temp[2][4], q_l5, time_sc5, cn,
in_posx+sep*4, zy, 4);
```

```
plot(q_temp[0][5], q_temp[1][5], q_temp[2][5], q_l6, time_sc6, cn,
in_posx+sep*5, zy, 5);
```

```
if (run_graph) {
    cn++;
}

if (cn>samples) {
    cn = 0;
    for (int i=0; i < 6; i++) {
        g_min1[i] = 0;
        g_max1[i] = 300;
        g_min2[i] = 300;
        g_max2[i] = 0;
        g_min3[i] = 300;
        g_max3[i] = 0;
    }
}

//cn=0;
}
```

```
void mousePressed() {
    if (mouseButton==LEFT) {
        if (dist(mouseX, mouseY, 1070, 592)<=10) {
            cn = 0;
            for (int i=0; i < 6; i++) {
                g_min1[i] = 0;
            }
        }
    }
}
```



```

        g_max1[i] = 300;
        g_min2[i] = 300;
        g_max2[i] = 0;
        g_min3[i] = 300;
        g_max3[i] = 0;
    }

    run_graph = true;
} else if (dist(mouseX, mouseY, 1166, 592)<=10) {
    cn = cn - 1;
    run_graph = false;
} else if (dist(mouseX, mouseY, 1264, 592)<=10&&run_graph==false) {
    String file_name = "";
    file_name = showInputDialog(frame, "Nombre", "Exportar",
    INFORMATION_MESSAGE);
    println(file_name);

    if (file_name!=null) {
        try {
            for (int i=0; i < 19; i++) {
                q_table.removeColumn(0);
                q_table.removeColumn(1);
                q_table.removeColumn(2);
                q_table.removeColumn(3);
                q_table.removeColumn(4);
                q_table.removeColumn(5);
                q_table.removeColumn(6);
                q_table.removeColumn(7);
                q_table.removeColumn(8);
                q_table.removeColumn(9);
                q_table.removeColumn(10);
                q_table.removeColumn(11);
            }
        }
    }
}

```

```
q_table.removeColumn(12);
q_table.removeColumn(13);
q_table.removeColumn(14);
q_table.removeColumn(15);
q_table.removeColumn(16);
q_table.removeColumn(17);
q_table.removeColumn(18);
}
}
catch(Exception e) {
}

q_table.addColumn("Time (ms)");
q_table.addColumn("q11");
q_table.addColumn("q12");
q_table.addColumn("q13");
q_table.addColumn("q21");
q_table.addColumn("q22");
q_table.addColumn("q23");
q_table.addColumn("q31");
q_table.addColumn("q32");
q_table.addColumn("q33");
q_table.addColumn("q41");
q_table.addColumn("q42");
q_table.addColumn("q43");
q_table.addColumn("q51");
q_table.addColumn("q52");
q_table.addColumn("q53");
q_table.addColumn("q61");
q_table.addColumn("q62");
q_table.addColumn("q63");
```

```
q_table.clearRows();
```

```
for (int i=0; i < cn; i++) {
```

```
    TableRow R = q_table.addRow();
```

```
    R.setFloat("Time (ms)", speed_gait*i);
```

```
    R.setFloat("q11", q_l1[0][i]);
```

```
    R.setFloat("q12", q_l1[1][i]);
```

```
    R.setFloat("q13", q_l1[2][i]);
```

```
    R.setFloat("q21", q_l2[0][i]);
```

```
    R.setFloat("q22", q_l2[1][i]);
```

```
    R.setFloat("q23", q_l2[2][i]);
```

```
    R.setFloat("q31", q_l3[0][i]);
```

```
    R.setFloat("q32", q_l3[1][i]);
```

```
    R.setFloat("q33", q_l3[2][i]);
```

```
    R.setFloat("q41", q_l4[0][i]);
```

```
    R.setFloat("q42", q_l4[1][i]);
```

```
    R.setFloat("q43", q_l4[2][i]);
```

```
    R.setFloat("q51", q_l5[0][i]);
```

```
    R.setFloat("q52", q_l5[1][i]);
```

```
    R.setFloat("q53", q_l5[2][i]);
```

```
    R.setFloat("q61", q_l6[0][i]);
```

```
    R.setFloat("q62", q_l6[1][i]);
```

```
    R.setFloat("q63", q_l6[2][i]);
```

```
}
```

```
    saveTable(q_table,
```

```
    program_path+"/graphs/"+file_name+second()+year()+".csv");
```

```
q_table.removeColumn("Time (ms)");
```

```
q_table.removeColumn("q11");
```

```
q_table.removeColumn("q12");
```

```
q_table.removeColumn("q13");
```



```

q_table.removeColumn("q21");
q_table.removeColumn("q22");
q_table.removeColumn("q23");
q_table.removeColumn("q31");
q_table.removeColumn("q32");
q_table.removeColumn("q33");
q_table.removeColumn("q41");
q_table.removeColumn("q42");
q_table.removeColumn("q43");
q_table.removeColumn("q51");
q_table.removeColumn("q52");
q_table.removeColumn("q53");
q_table.removeColumn("q61");
q_table.removeColumn("q62");
q_table.removeColumn("q63");
}
}
}
}

```

```

void exit() {
    dispose();
    graph_on = false;
    cn = 0;
    //for (int i=0; i < 6; i++) {
    // g_min1[i] = 0;
    // g_max1[i] = 300;
    // g_min2[i] = 300;
    // g_max2[i] = 0;
    // g_min3[i] = 300;
    // g_max3[i] = 0;
    //}
}

```

```

    GRAPH = null;
}

void plot(float q1, float q2, float q3, float[][] qs, float [] time_sc, int n, int
posx, int posy, int leg) {
    //GRAPHIC
    int wth = 180;
    int hth = 110;
    int desf = hth+58;
    textAlign(CENTER, CENTER);
    fill(0);
    text("PATA "+(leg+1), wth/2+posx, posy-20);

    if (run_graph) {
        if (q1>g_min1[leg]) {
            g_min1[leg] = q1;
        }
        if (q1<g_max1[leg]) {
            g_max1[leg] = q1;
        }
        if (q2<g_min2[leg]) {
            g_min2[leg] = q2;
        }
        if (q2>g_max2[leg]) {
            g_max2[leg] = q2;
        }
        if (q3<g_min3[leg]) {
            g_min3[leg] = q3;
        }
        if (q3>g_max3[leg]) {
            g_max3[leg] = q3;
        }
    }
}

```

}

```

for (int i=0; i < 3; i++) {
    stroke(0);
    fill(255);
    rect(posx, posy+desf*i, wth, hth);
    fill(0);
    textAlign(CENTER, CENTER);
    text("Tiempo", wth/2+posx, (posy+hth+21)+desf*i);
    fill(0, 0, 255);
    stroke(0, 0, 255);
    textAlign(RIGHT, CENTER);
    if (i==0) {
        text(nf(g_max1[leg], 1, 1), posx-3, posy+desf*i);
        text(nf(g_min1[leg], 1, 1), posx-3, posy+hth+desf*i);
        //float yy = map(abs(g_max1[leg]-g_min1[leg])/2+g_max1[leg],
g_min1[leg], g_max1[leg], hth+posy, posy);
        line(posx, posy+hth/2, posx+wth, posy+hth/2);
        text(nf(abs(g_max1[leg]-g_min1[leg])/2+g_max1[leg], 1, 1), posx-3,
posy+hth/2+desf*i);
    } else if (i==1) {
        text(nf(g_max2[leg], 1, 1), posx-3, posy+desf*i);
        text(nf(g_min2[leg], 1, 1), posx-3, posy+hth+desf*i);
        if (g_min2[leg]>g_max2[leg]) {
            text(nf(abs(g_max2[leg]-g_min2[leg])/2+g_min2[leg], 1, 1), posx-3,
posy+hth/2+desf*i);
        } else {
            text(nf(-abs(g_max2[leg]-g_min2[leg])/2+g_max2[leg], 1, 1), posx-3,
posy+hth/2+desf*i);
        }
        line(posx, posy+hth/2+desf, posx+wth, posy+hth/2+desf);
    } else if (i==2) {
        text(nf(g_max3[leg], 1, 1), posx-3, posy+desf*i);
    }
}

```



```

text(nf(g_min3[leg], 1, 1), posx-3, posy+hth+desf*i);
if (g_min3[leg]>g_max3[leg]) {
    text(nf(abs(g_max3[leg]-g_min3[leg])/2+g_max3[leg], 1, 1), posx-3,
posy+hth/2+desf*i);
} else {
    text(nf(-abs(g_max3[leg]-g_min3[leg])/2+g_max3[leg], 1, 1), posx-3,
posy+hth/2+desf*i);
}
line(posx, posy+hth/2+desf*2, posx+wth, posy+hth/2+desf*2);
}
}

//GET DATA AND SAVE IT
//qs[0][n] = map(q1, q1_min[leg], q1_max[leg], posy+hth, posy);
//qs[1][n] = map(q2, q2_min[leg], q2_max[leg], posy+desf+hth, posy+desf);
//qs[2][n] = map(q3, q3_min[leg], q3_max[leg], posy+desf*2+hth,
posy+desf*2);

if (run_graph) {
    qs[0][n] = q1;
    qs[1][n] = q2;
    qs[2][n] = q3;

    time_sc[n] = map(time_plot[n], 0, time_plot[samples], posx, posx+wth);
}

//PLOT DATA
stroke(0, 255, 0);
for (int i=0; i < cn; i++) {
    float u1 = map(qs[0][i], g_min1[leg], g_max1[leg], posy+hth, posy);
    float u11 = map(qs[0][i+1], g_min1[leg], g_max1[leg], posy+hth, posy);
    float u2 = 0, u22 = 0, u3 = 0, u33 = 0;
    //if (leg==0||leg==1||leg==2) {

```

```

u2 = map(qs[1][i], g_min2[leg], g_max2[leg], posy+ht+desf, posy+desf);
u22 = map(qs[1][i+1], g_min2[leg], g_max2[leg], posy+ht+desf,
posy+desf);
u3 = map(qs[2][i], g_min3[leg], g_max3[leg], posy+ht+desf*2,
posy+desf*2);
u33 = map(qs[2][i+1], g_min3[leg], g_max3[leg], posy+ht+desf*2,
posy+desf*2);
    //} else {
        // u2 = map(qs[1][i], g_min2[leg], g_max2[leg], posy+ht+desf,
posy+desf);
        // u22 = map(qs[1][i+1], g_min2[leg], g_max2[leg], posy+ht+desf,
posy+desf);
        // u3 = map(qs[2][i], g_max3[leg], g_min3[leg], posy+ht+desf*2,
posy+desf*2);
        // u33 = map(qs[2][i+1], g_max3[leg], g_min3[leg], posy+ht+desf*2,
posy+desf*2);
    //}
    line(time_sc[i], u1, time_sc[i+1], u11);
    line(time_sc[i], u2, time_sc[i+1], u22);
    line(time_sc[i], u3, time_sc[i+1], u33);
    //point(time_sc[i], u1);
    //point(time_sc[i], u2);
    //point(time_sc[i], u3);
}

//RESET
fill(0);
stroke(0);
textAlign(LEFT, CENTER);
}
}

```

CAMINATA TRIPODE

```
float[] para(float p0, float pi, float pf, float t) {
    float[] result = new float[3];

    result[0] = p0;
    result[1] = (4*pi-3*p0-pf)/t;
    result[2] = (pf-p0-result[1]*t)/(pow(t, 2));

    return result;
}

void tripod_gait(float mag_l, float ang, float tmp, float speed) {

    ////////////
    //VARIABLES
    float [][] p0 = new float[3][6];
    float [][] pi = new float[3][6];
    float [][] pf = new float[3][6];
    speed_gait = int(speed/(int(time/sample)));

    // SLIDERS
    for (int i=0; i < 3; i++) {
        for (int j=0; j < 6; j++) {
            if (i==0) {
                pos[i][j] = x[j].getValue();
            } else if (i==1) {
                pos[i][j] = y[j].getValue();
            } else if (i==2) {
                pos[i][j] = z[j].getValue();
            }
        }
    }
}
```


//ANGULO DE DIRECCION DEL VECTOR

x_mov = mag_l*sin(radians(ang));

y_mov = mag_l*cos(radians(ang));

//println(x_mov, y_mov);

//DETERMINE INICIAL MID AND FINAL POSITION

for (int i=0; i < 6; i++) {

p0[0][i] = pos[0][i];

p0[1][i] = pos[1][i];

p0[2][i] = pos[2][i];

pi[2][i] = pos[2][i] + z_mov;

pf[2][i] = pos[2][i];

if (i!=3&&i!=4&&i!=5) {

pf[1][i] = pos[1][i] + y_mov;

pi[1][i] = pos[1][i] + y_mov/2;

pf[0][i] = pos[0][i] + x_mov;

pi[0][i] = pos[0][i] + x_mov/2;

} else {

pf[1][i] = pos[1][i] - y_mov;

pi[1][i] = pos[1][i] - y_mov/2;

pf[0][i] = pos[0][i] - x_mov;

pi[0][i] = pos[0][i] - x_mov/2;

}

}

//for (int i=0; i < 6; i++) {

// println(p0[0][i], p0[1][i], p0[2][i], pi[0][i], pi[1][i], pi[2][i], pf[0][i], pf[1][i], pf[2][i]);

//}

int cont = 0;

//LINEAR MOVEMENT TOWARD Y AXIS

```
for (int i=0; i < 6; i++) {
    cont = 0;
    for (float j=0; j < tmp; j = j + sample) {

        if (i==0||i==1||i==2) {
            xg_body[i][cont] = pos[0][i] - x_mov/(int(tmp/sample))*cont;
            yg_body[i][cont] = pos[1][i] - y_mov/(int(tmp/sample))*cont;
        } else {
            yg_body[i][cont] = pos[1][i] + y_mov/(int(tmp/sample))*cont;
            xg_body[i][cont] = pos[0][i] + x_mov/(int(tmp/sample))*cont;
        }
        zg_body[i][cont] = pos[2][i];
        cont++;
    }
}
```

//CALCULATE X Y Z CHANGES ACCORDING TO TIME

```
for (int i=0; i < 6; i++) {
    float [] xk = new float[3];
    float [] yk = new float[3];
    float [] zk = new float[3];
    xk = para(p0[0][i], pi[0][i], pf[0][i],tmp);
    yk = para(p0[1][i], pi[1][i], pf[1][i],tmp);
    zk = para(p0[2][i], pi[2][i], pf[2][i],tmp);
    cont = 0;
    for (float j=0; j < tmp; j = j + sample) {
        xg_leg[i][cont] = xk[0]+xk[1]*j+xk[2]*j*(j);
        yg_leg[i][cont] = yk[0]+yk[1]*j+yk[2]*j*(j);
        zg_leg[i][cont] = zk[0]+zk[1]*j+zk[2]*j*(j);
        cont++;
    }
}
```

```

    }
}

//CALCULATE FINAL MATRIX 6 X 62
for (int i=0; i < 6; i++) {
    for (int j=0; j < (2*(int(tmp/sample)+1)); j++) {
        if (j<((2*(int(tmp/sample)+1))/2)) {
            if (i==0||i==2||i==4) {
                xg_final[i][j] = xg_leg[i][j];
                yg_final[i][j] = yg_leg[i][j];
                zg_final[i][j] = zg_leg[i][j];
            } else {
                xg_final[i][j] = xg_body[i][j];
                yg_final[i][j] = yg_body[i][j];
                zg_final[i][j] = zg_body[i][j];
            }
        } else {
            if (i==0||i==2||i==4) {

                if (i==0||i==2) {
                    yg_final[i][j] = yg_body[i][j]-((2*(int(tmp/sample)+1))/2)]+y_mov;
                    xg_final[i][j] = xg_body[i][j]-((2*(int(tmp/sample)+1))/2)]+x_mov;
                } else {
                    yg_final[i][j] = yg_body[i][j]-((2*(int(tmp/sample)+1))/2)]-y_mov;
                    xg_final[i][j] = xg_body[i][j]-((2*(int(tmp/sample)+1))/2)]-x_mov;
                }
                zg_final[i][j] = zg_body[i][j]-((2*(int(tmp/sample)+1))/2)];
            } else {

                if (i==3||i==5) {
                    yg_final[i][j] = yg_leg[i][j]-((2*(int(tmp/sample)+1))/2)]+y_mov;
                    xg_final[i][j] = xg_leg[i][j]-((2*(int(tmp/sample)+1))/2)]+x_mov;

```



```
} else if (i==1) {  
    yg_final[i][j] = yg_leg[i][j-((2*(int(tmp/sample)+1))/2)]-y_mov;  
    xg_final[i][j] = xg_leg[i][j-((2*(int(tmp/sample)+1))/2)]-x_mov;  
}  
zg_final[i][j] = zg_leg[i][j-((2*(int(tmp/sample)+1))/2)];  
}  
}  
}  
}  
}  
////////////////////  
}
```



ESTRUCTURA DE INTERFAZ

//SLIDERS

import controlP5.*;

ControlP5 cp5;

//FKINE

Slider[] q1;

Slider[] q2;

Slider[] q3;

//IKINE

Slider[] x;

Slider[] y;

Slider[] z;

//BODY

Slider[] body_q;

Slider[] body_pos;

//CIRCULAR MOTION

Slider[] circ_param; //0 Radius of motion - 1 Speed

//TRIPOD GAIT

Slider[] tripod;

//SLIDERS LIMITS

//float q1_mn = -90; //70

//float q1_mx = 90; //110

float q1_mn = -25; //70

float q1_mx = 25; //110

//float q2_min = 60; //60

//float q3_min = 0; //0

//float q2_max = 180; //180

//float q3_max = 90; //90

//float initial_pos1 = 90;

//float initial_pos2 = 90;

//float initial_pos3 = 90;

//POSITION DEFINED 07/10/2020

//float initial_pos1[] = {0, 0, 0, 0, 0, 0};

//float initial_pos2[] = {18.45, 14.03, 15.66, -4, 1.75, -1.97};

//float initial_pos3[] = {-70.08, -64.83, -67.71, -34.41, -43.88, -37.72};

//NEW POSITION DEFINED 14/10/2020

//float initial_pos1[] = {0, 0, 0, 0, 0, 0};

//float initial_pos2[] = {15.44, 14.37, 6.21, 4.52, 16.51, 9.54};

//float initial_pos3[] = {-45.18, -44.51, -40.7, -29.54, -40.64, -29.04};

//ALL Z ARRANGE TO ONE PLANE (NEEDS MORE PRECISION)

//float initial_pos1[] = {0, 0, 0, 0, 0, 0};

//float initial_pos2[] = {15.47, 15.67, 18.46, 4.51, 9.02, 4.88};

//float initial_pos3[] = {-45.21, -45.62, -51.56, -24.18, -32.53, -24.74};

//POSITION 19/10/2020

float initial_pos1[] = {0, 7, 3.5, 4, -10, 0};

float initial_pos2[] = {0.68, 0, 0, 0.44, -0.6, 0};

float initial_pos3[] = {-43.5, -48, -42.59, -40.36, -46.1, -46.12};

//configuracion de los eslaiders de cada articulacion .. min o max

float q1_min[] = {115, 109.5, 112, 115, 109.5, 115};//POSITION 24/11/2020

float q1_max[] = {65, 59.5, 62, 65, 59.5, 65};

float q2_min[] = {107, 99, 95, 77.43, 65.6, 74.29};

float q2_max[] = {0, -1, 0, 179.43, 172.6, 164.29};

float q3_min[] = {10, 10, 10, 155, 165, 157};


```
float q3_max[] = {100, 100, 103, 78, 82, 70};
```

```
float q2_mn[] = {-25, -20, -19, -22, -22, -20};
```

```
//float q2_mn[] = {-40, -40, -40, -40, -40, -40};
```

```
float q2_mx[] = {82, 80, 76, 80, 85, 70};
```

```
//float q2_mx[] = {90, 90, 90, 90, 90, 90};
```

```
float q3_mn[] = {-75, -75, -77, -65, -71, -67};
```

```
//float q3_mn[] = {-100, -100, -100, -100, -100, -100};
```

```
float q3_mx[] = {15, 15, 16, 12, 12, 20};
```

```
//float q3_mx[] = {30, 30, 30, 30, 30, 30};
```

```
//FONT
```

```
PFont txt_font;
```

```
void UI() {
```

```
    //TITLE
```

```
    fill(50);
```

```
    textSize(12);
```

```
    textAlign(LEFT, CENTER);
```

```
    text("Robot Hexápodo - 2020", 20, 20);
```

```
    //VISTA DE CONTROL
```

```
    //////////////////////////////////////
```

```
    //fill(50);
```

```
    //textSize(12);
```

```
    //textAlign(LEFT, CENTER);
```

```
    text("VISTA", 40, 508);
```

```
    //BUTTONS
```

```
    fill(0);
```

```
ellipse(98, 550, 20, 20);  
ellipse(188, 550, 20, 20);  
ellipse(262, 550, 20, 20);  
ellipse(98, 590, 20, 20);  
ellipse(188, 590, 20, 20);
```

```
fill(255, 255, 0);  
if (world_sel==0) { //FRONT  
    ellipse(98, 550, 16, 16);  
} else if (world_sel==1) { //BACK  
    ellipse(188, 550, 16, 16);  
} else if (world_sel==2) { //FREE  
    ellipse(262, 550, 16, 16);  
}
```

```
fill(50);  
text("Frontal", 35, 550);  
text("Trasera", 120, 550);  
text("Libre", 214, 550);  
text("Posición", 35, 590);  
text("Resetear", 120, 590);  
////////////////////////////////
```

```
//KINEMATICS
```

```
////////////////////////////////
```

```
//MENU
```

```
text("CINEMÁTICA DIRECTA", 550, 40);  
text("CINEMÁTICA INVERSA", 550, 270);  
text("CINEMÁTICA CUERPO", 550, 500);  
text("MOVIMIENTO CIRCULAR", 950, 40);  
text("CAMINATA TRÍPODE", 950, 160);  
text("GRÁFICAS", 950, 320);
```

```

text("CONEXIÓN", 950, 360);

for (int i=0; i < 3; i++) {
    text("q" + (i+1), 622+110*i, 70);
}

//fill(50);
text("pX", 622, 300);
text("X (" + nf(trans_x2, 1, 1) + ")", 612, 530);
text("pY", 732, 300);
text("Y (" + nf(trans_y2, 1, 1) + ")", 722, 530);
text("pZ", 842, 300);
text("Z (" + nf(trans_z2, 1, 1) + ")", 832, 530);
text("Radio", 1012, 70);
text("Velocidad", 1110, 70);
text("Magnitud", 1002, 234);
text("Dirección", 1112, 234);
text("Tiempo", 1220, 234);
text("Velocidad: " + nf(tripod[0].getValue()/(tripod[2].getValue()/1000), 1,
2) + " cm/s", 1120, 200);

//textAlign(LEFT, CENTER);
for (int i=0; i < 6; i++) {
    text((i+1), 560, 94+25*i);
    text((i+1), 560, 324+25*i);
}

//textAlign(LEFT, CENTER);
text("Pos", 550, 554);
text("Rot", 550, 579);
text("Val", 950, 94);

```



```
text("C (" + nf(trans_x1, 1, 2).replace(',', '.') + ", " + nf(trans_y1, 1, 2).replace(',', '.') + ")", 1200, 94);
```

```
text("Go", 950, 200);
```

```
text("Detener", 1010, 200);
```

```
text("Val", 950, 258);
```

```
//BUTTONS
```

```
fill(0);
```

```
ellipse(702, 40, 20, 20); //FORWARD KINEMATICS
```

```
ellipse(702, 270, 20, 20); //INVERSE KINEMATICS
```

```
ellipse(702, 500, 20, 20); //BODY KINEMATICS
```

```
ellipse(1110, 40, 20, 20); //CIRCULAR MOTION
```

```
ellipse(1110, 160, 20, 20); //TRIPOD GAIT
```

```
ellipse(980, 200, 16, 16); //GO
```

```
ellipse(1074, 200, 16, 16); //STOP
```

```
ellipse(1030, 320, 20, 20); //GRAPH
```

```
fill(255, 255, 0);
```

```
if (opt_sel==0) { //FKINE
```

```
    ellipse(702, 40, 16, 16);
```

```
} else if (opt_sel==1) { //IKINE
```

```
    ellipse(702, 270, 16, 16);
```

```
} else if (opt_sel==2) { //BODY
```

```
    ellipse(702, 500, 16, 16);
```

```
} else if (opt_sel==3) { //CIRCULAR MOTION
```

```
    ellipse(1110, 40, 16, 16);
```

```
} else if (opt_sel==4) { //TRIPOD GAIT
```

```
    ellipse(1110, 160, 16, 16);
```

```
}
```

```
if (tripod_sel==0) { //GO
```

```
    ellipse(980, 200, 12, 12);
```

```
} else if (tripod_sel==1) { //STOP  
    ellipse(1074, 200, 12, 12);  
}
```

```
if (graph_on==true) {  
    ellipse(1030, 320, 16, 16);  
}
```

```
fill(0);
```

```
////////////////////
```

```
}
```



PROGRAMA PRINCIPAL

//HEXAPODO PARTES

PShape coxa1;

PShape coxa2;

PShape femur1;

PShape femur2;

PShape tibia1;

PShape tibia2;

PShape body;

PShape ground;

//DIMENSIONES

float L1 =7.3;

float L2 = 9;

float L3 =18.3; //18.3

float xof = 6;

float yof = 10;

float zof = 0; //15

//PLANO

float rotX = -0.6, rotY = 3.42;

float xpos = 280;

float ypos = 270;

boolean world_rot = false;

boolean outside_click = false;

int world_sel = 0;

//boolean front_sel = true; //0

//boolean back_sel = false; //1

//boolean free_sel = false; //2

//CINEMATICA

int opt_sel = 0; //0 KFINE - 1 IKINE

//ESCALA

float sc = 0.8;

//SOFTWARE INFO

String version = "Final";

// ANGULOS DE ROTACION

float [][] q = new float [3][6];

float [][] qi = new float[3][6];

//POSICION CINEMTICA INVERSA

float [][] pos = new float[3][6];

//CINEMATICA DEL CUERPO

float trans_x1 = 0;

float trans_y1 = 0;

float trans_z1 = 0;

float trans_x2 = 0;

float trans_y2 = 0;

float trans_z2 = 0;

float Tx = 0;

float Ty = 0;

float Tz = 0;

float rX = 0;

float rY = 0;

float rZ = 0;

float dx_rot[] = new float[6];

float dy_rot[] = new float[6];

float dz_rot[] = new float[6];

//MOVIMEINTO CIRCULAR

float anf = 0;

float xp = 0;

float yp = 0;

//CAMINATA TRIPOD

float time = 3;

float sample = 0.1;

float y_mov = 0;

float x_mov = 0;

float z_mov = 3;

float ang_mov = 0;

float mag = 4;

float [][] xg_leg = new float[6][int(time/sample)+1];

float [][] yg_leg = new float[6][int(time/sample)+1];

float [][] zg_leg = new float[6][int(time/sample)+1];

float [][] xg_body = new float[6][int(time/sample)+1];

float [][] yg_body = new float[6][int(time/sample)+1];

float [][] zg_body = new float[6][int(time/sample)+1];

float [][] xg_final = new float[6][2*(int(time/sample)+1)];

float [][] yg_final = new float[6][2*(int(time/sample)+1)];

float [][] zg_final = new float[6][2*(int(time/sample)+1)];

int num_sample = 0;

boolean one_time = false;

boolean tripod_run = false;

boolean tripod_finish = false;

int tripod_sel = 1;

//TIEMPO DE CONTROL

int pv_time = 0;

int pv_time2 = 0;

int ac_time = 0;

int ct2 = 0;

//CAMINATAS

int speed_gait = 0;

//int speed = 10;

//LIBRERIAS

import static javax.swing.JOptionPane.*;

//DIRECTORIO

String program_path = "";

void settings() {

//TAMAÑO DE PANTALLA (CONFIGURATION)

size(1310, 630, P3D);

PJOGL.setIcon("HEX.png");

}

void setup() {

//SKETCH PATH

program_path = sketchPath();

//NOM,BRE DEL PROGRAMA

surface.setTitle("HEXAPOD - "+version);

surface.setLocation(displayWidth/2-655, displayHeight/2-350);

//texto

//txt_font = loadFont("Verdana-12.vlw");

//textFont(txt_font, 12);

//HEXAPOD PARTS

```
coxa1 = loadShape("COXA1.obj");  
coxa2 = loadShape("COXA2.obj");  
femur1 = loadShape("FEMUR1.obj");  
femur2 = loadShape("FEMUR2.obj");  
tibia1 = loadShape("TIBIA1.obj");  
tibia2 = loadShape("TIBIA2.obj");  
body = loadShape("real_body.obj");  
ground = loadShape("surface.obj");
```

//MATRIZ DE INICIALIZacion

```
for (int i=0; i < 3; i++) {  
    for (int j=0; j < 6; j++) {  
        q[i][j] = 0;  
        qi[i][j] = 0;  
        pos[i][j] = 0;  
    }  
}
```

// SLIDERS INICIALES

```
cp5 = new ControlP5(this);  
cp5.setAutoDraw(false);
```

//FKINE

```
q1 = new Slider[6];  
q2 = new Slider[6];  
q3 = new Slider[6];
```

//IKINE

```
x = new Slider[6];  
y = new Slider[6];
```

```
z = new Slider[6];
```

```
//BODY IKINE
```

```
body_q = new Slider[3];
```

```
body_pos = new Slider[3];
```

```
//movimiento circular
```

```
circ_param = new Slider[2];
```

```
//caminata tripod
```

```
tripod = new Slider[3];
```

```
for (int i=0; i < 6; i++) {
```

```
    // CINEMATICA DIRECTA
```

```
    q1[i] = cp5.addSlider("q1_" + i).setPosition(580, 86+25*i)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(q1_mn).setMax(q1_mx).setLabel("");
```

```
    q2[i] = cp5.addSlider("q2_" + i).setPosition(690, 86+25*i)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(q2_mn[i]).setMax(q2_mx[i]).setLabel("");
```

```
    q3[i] = cp5.addSlider("q3_" + i).setPosition(800, 86+25*i)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(q3_mn[i]).setMax(q3_mx[i]).setLabel("");
```

```
//CINEMATICA INVERSA
```

```
    x[i] = cp5.addSlider("x_" + i).setPosition(580, 316+25*i)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(-50).setMax(L1+L2+L3).setLabel("").setLock(true);
```

```
    y[i] = cp5.addSlider("y_" + i).setPosition(690, 316+25*i)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(-50).setMax(50).setLabel("").setLock(true);
```

```
    z[i] = cp5.addSlider("z_" + i).setPosition(800, 316+25*i)
```

```
.setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
.setMin(-50).setMax(50).setLabel("").setLock(true);
}

//CINEMATICA DEL CUERPO
for (int i=0; i < 3; i++) {
    body_pos[i] = cp5.addSlider("body_pos_"+i).setPosition(580+110*i, 546)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(-10).setMax(10).setLabel("").setLock(true);
    body_q[i] = cp5.addSlider("body_q_"+i).setPosition(580+110*i, 571)
        .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
        .setMin(-10).setMax(10).setLabel("").setLock(true);
}

//MOVIMIENTO CIRCULAR
circ_param[0] = cp5.addSlider("circ_param_0").setPosition(980, 86)
    .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
    .setMin(0).setMax(10).setLabel("").setLock(true);
circ_param[1] = cp5.addSlider("circ_param_1").setPosition(1090, 86)
    .setSize(100, 20).setValue(0).setFont(createFont("arial", 12))
    .setMin(-0.5).setMax(0.5).setLabel("").setLock(true);
```


Anexo 4: CODIGO ARDUINO

INTERFAZ

```
#include <Servo.h>
```

```
//INCOMING DATA
```

```
float num[18];
```

```
float val[18];
```

```
int cont = 0;
```

```
//SERVOMOTOR
```

```
Servo servos[18];
```

```
/*
```

```
PATA 1 -> (SC1) -> 28,29,30
```

```
PATA 2 -> (B1) -> 25,26,27
```

```
PATA 3 -> (A1) -> 22,23,24
```

```
PATA 4 -> (F1) -> 37,38,39
```

```
PATA 5 -> (E1) -> 34,35,36
```

```
PATA 6 -> (D1) -> 31,32,33
```

```
*/
```

```
//PINES DE LOS SERVOS
```

```
int pins[18] = {28,29,30,25,26,27,22,23,24,37,38,39,34,35,36,31,32,33};
```

```
void setup() {
```

```
    //SERIAL
```

```
    Serial.begin(115200);
```

```
    //INIT IN DATA
```

```
    for (int i = 0; i < 18; i++) {
```

```
        num[i] = 0;
```

```
        val[i] = 0;
```

```
    }
```

```
    for (int i = 0; i < 18; i++) {
```

```
        servos[i].attach(pins[i]);
```

```
    }
```

```
}
```

```
void loop() {
```

```
    //RECEIVER
```

```
    //RECEPCION DATOS
```

```
    //////////////////////////////////
```

```
    while (Serial.available() > 0) {
```

```

num[cont] = Serial.parseInt();
cont++;

if (Serial.read() == '\n') {
    for (int i = 0; i < cont; i++) {
        val[i] = num[i];
    }
    cont = 0;
}
}
}

////////////////////

for (int i = 0; i < 18; i++) {
    servos[i].write(val[i]);
    //servos[i].write(90);
    //Serial.println(val[i]);
}
}

```

MANDO DE COTROL

//SEND DATA

float send_data[8] = {0, 0, 0, 0, 0, 0, 0, 0};

//RF TRANSMITTER (NRF24L01 + PA/LNA)

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

RF24 hexapod(9, 8); //CE AND CSN

const byte address[6] = "16078";

//BOTONES

int botones[6] = {2, 3, 4, 5, 7, 6};

int digital_value[7] = {0, 0, 0, 0, 0, 0, 0};

//ANALOG

int analog_value[2] = {0, 0};

```
//JOYSTICK

int h_init = 497;
int v_init = 520;
float angle_joystick = 0;

void radio_config() {

    hexapod.begin();
    hexapod.setPALevel(RF24_PA_LOW);
    hexapod.openWritingPipe(address);
}

void setup() {

    //SERIAL INITIALIZE
    //  Serial.begin(9600);

    //INITIALIZE RF24
    radio_config();

    //INITIALIZE BUTTONS
    for (int i = 0; i < 6; i++) {
        pinMode(botones[i], INPUT);
    }

}

void loop() {

    //READ DATA FROM BUTTONS
```



```

for (int i = 0; i < 7; i++) {
    if (i < 6) {
        digital_value[i] = digitalRead(botones[i]);
    }
    else {
        digital_value[6] = 1 - constrain(analogRead(A3), 0, 1); //POSE SEND 0
    }
}

//READ DATA JOYSTICK
analog_value[0] = analogRead(A0);
analog_value[1] = analogRead(A1);

//CALCULATE ANGLE
float temp_h = analog_value[0] - h_init;
float temp_v = analog_value[1] - v_init;

if (abs(temp_h) < 5 && abs(temp_v) < 5) {
    angle_joystick = -1607;
}
else {
    if (temp_h > 0 && temp_v < 0) {
        angle_joystick = degrees(atan(temp_v / temp_h)) + 90;
    }
    else if (temp_h > 0 && temp_v > 0) {
        angle_joystick = degrees(atan(temp_v / temp_h)) + 90;
    }
    else if (temp_h < 0 && temp_v > 0) {
        angle_joystick = degrees(atan(temp_v / temp_h)) - 90;
    }
    else if (temp_h < 0 && temp_v < 0) {
        angle_joystick = degrees(atan(temp_v / temp_h)) - 90;
    }
}

```

```
}  
}
```

//SEND DATA ACCORDING TO BUTTONS ON THE CONTROLLER

```
send_data[0] = digital_value[0];  
send_data[1] = digital_value[1];  
send_data[2] = digital_value[2];  
send_data[3] = digital_value[3];  
send_data[4] = digital_value[4];  
send_data[5] = digital_value[5];  
send_data[6] = digital_value[6];  
send_data[7] = angle_joystick;
```

//SEND TO RF

```
hexapod.write(send_data, sizeof(send_data));
```

```
delay(10);
```

```
}
```

PROGRAMA PRINCIPAL

```
//SERVOMOTORS

#include <Servo.h>

const int num_servos = 18;

Servo servos[num_servos];

int pins[num_servos] = {42, 44, 46, 36, 38, 40, 33, 32, 34, 15, 17, 19, 21, 23, 25,
27, 29, 31};

//RF RECEIVER COMMUNICATION (NRF24L01)

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

RF24 hexapod(7, 6); //CE AND CSN
const byte address[6] = "16078";
float value[8] = {0, 0, 0, 0, 0, 0, 0, 0};

//PARAMETROS DEL HEXAPODO

float L1 = 7.3;
float L2 = 9;
float L3 = 18.3;
float xof = 6;
float yof = 10;
float diff_q1 = 45;
float diff_q3 = 33.93;

//RESTRICCION DE MOVIMIENTO (VARIAR)

float q1_mn = -25;
float q1_mx = 25;
float q2_mn [6] = { -25, -20, -19, -22, -22, -20};
float q2_mx [6] = {82, 80, 76, 80, 85, 70};
```


float q3_mn [6] = { -75, -75, -77, -65, -71, -67};

float q3_mx [6] = {15, 15, 16, 12, 12, 20};

//REAL SERVO ANGLES (variarnos minimos y maximos)

float q1_min[6] = {119.5, 110.51, 105.5, 113.01, 110.01, 100.5};

float q1_max[6] = {69.5, 60.51, 55.5, 63.01, 60.01, 50.5};

float q2_min[6] = {107, 99, 95, 77.43, 65.6, 74.29};

float q2_max[6] = {0, -1, 0, 179.43, 172.6, 164.29};

float q3_min[6] = {10, 10, 10, 155, 165, 157};

float q3_max[6] = {100, 100, 103, 78, 82, 70};

//POSICION DE CAMINATA

float initial_pos [3][6] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, { -43.5, -43.5, -42.59, -41.13, -43.61, -38.29}};

//PISICIO N DE RESET

float reset_pos [3][6] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};

//ANGULOS DE ROTACION DE CADA EXTREMIDAD

float q [3][6] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};

float qi [3][6] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};

//POSICION DE LAS EXTREMIDADES

float pos [3][6] = {{0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0}};;

//CINEMATICA INVERSA

float temp_pos [3][6];

//cinematica del cuerpo

float xt, yt, zt;

float qX, qY, qZ;

float trans_x1 = 0;

```
float trans_y1 = 0;
float trans_z1 = 0;
float trans_x2 = 0;
float trans_y2 = 0;
float trans_z2 = 0;
float Tx, Ty, Tz;
float rX, rY, rZ;
float dx_rot [6] = {0, 0, 0, 0, 0, 0};
float dy_rot [6] = {0, 0, 0, 0, 0, 0};
float dz_rot [6] = {0, 0, 0, 0, 0, 0};

//MOVIMIENTO CIRCULAR
float anf = 0;
float xp = 0;
float yp = 0;
float rotation_speed = 0.05;
float radius = 3.5;

// TRAJECTORIA PARABOLICA
float tmt = 3;
float sample = 0.1;
float y_mov = 0;
float x_mov = 0;
float z_mov = 3;
float p0[3][6];
float pi[3][6];
float pf[3][6];

//CAMINATA TRIPOD
float ang_mov = 0; //DIRECTION °
float mag = 4; //MOVEMENT TOWARDS DIRECTION [cm]
//int ta1 = int(tmt / sample) + 1;
```

```
//int tb1 = 2 * size1;
const int size1 = 31; //ta1
const int size2 = 62; //tb1
int num_sample = 0;
boolean execute_once = false;
boolean tripod_run = false;
boolean tripod_finish = false;
int tripod_speed = 0;
float time_completion = 3; //tiempo completado [sec]

//CONTROL DE TIEMPO
unsigned long pv_time = 0;
unsigned long ac_time = 0;

//SELECCION
int option = 0;
bool once = false;
bool once2 = false;
int cont = 0;

//ONE TIME
bool one_only = false;
int cont_only = 0;
int mode = -1;
bool already_pressed = false;

//POSE TIME
unsigned long pose_prev = 0;
bool pose_count = false;
bool pose_pressed = false;

//RESET
```



```
bool reset_pressed = false;
```

```
//MODO
```

```
bool mode_pressed = false;
```

```
//BOTONES
```

```
bool y_pressed = false;
```

```
bool x_pressed = false;
```

```
bool b_pressed = false;
```

```
bool a_pressed = false;
```

```
//ROTATION / TRASLATION
```

```
bool rot_tras = false;
```

```
//CONECTAR
```

```
unsigned long data_prev = 0;
```

```
boolean reconnect = false;
```

```
void radio_config() {
```

```
    hexapod.begin();
```

```
    hexapod.openReadingPipe(0, address);
```

```
    hexapod.setPALevel(RF24_PA_LOW);
```

```
    hexapod.startListening();
```

```
}
```

```
void setup() {
```

```
    //COMUNICACIÓN SERIAL
```

```
    // Serial.begin(9600);
```

```
//INITIALIZE SERVO MOTORS
for (int i = 0; i < num_servos; i++) {
    servos[i].attach(pins[i]);
}

//INITIALIZE RF24
radio_config();

pinMode(13, OUTPUT);
}

void loop() {

//ACTUAL TIME PASSED
ac_time = millis();

//RF RECEIVER
if (one_only == true) {
    RF_RECEIVE();
}

//MENU OPTIONS
switch (option) {
    //RESET
    case 0:
        assign(q, reset_pos);
        option = 2;
        break;

    //POSE
    case 1:
```

```
assign(q, initial_pos);
```

```
option = 2;
```

```
break;
```

```
//CINEMATICA DIRECTA
```

```
case 2:
```

```
FKINE(q, pos);
```

```
break;
```

```
// CINEMATICA INVERSA
```

```
case 3:
```

```
IKINE(pos, q);
```

```
break;
```

```
// CINEMATICA DEL CUERPO
```

```
case 4:
```

```
BKINE_CALC(pos, q);
```

```
break;
```

```
//MOVIMIENTO ROTACION
```

```
case 5:
```

```
CIRCULAR(pos, q, radius, rotation_speed);
```

```
break;
```

```
//CAMINATA TRIPOD
```

```
case 6:
```

```
TRIPOD(pos, q);
```

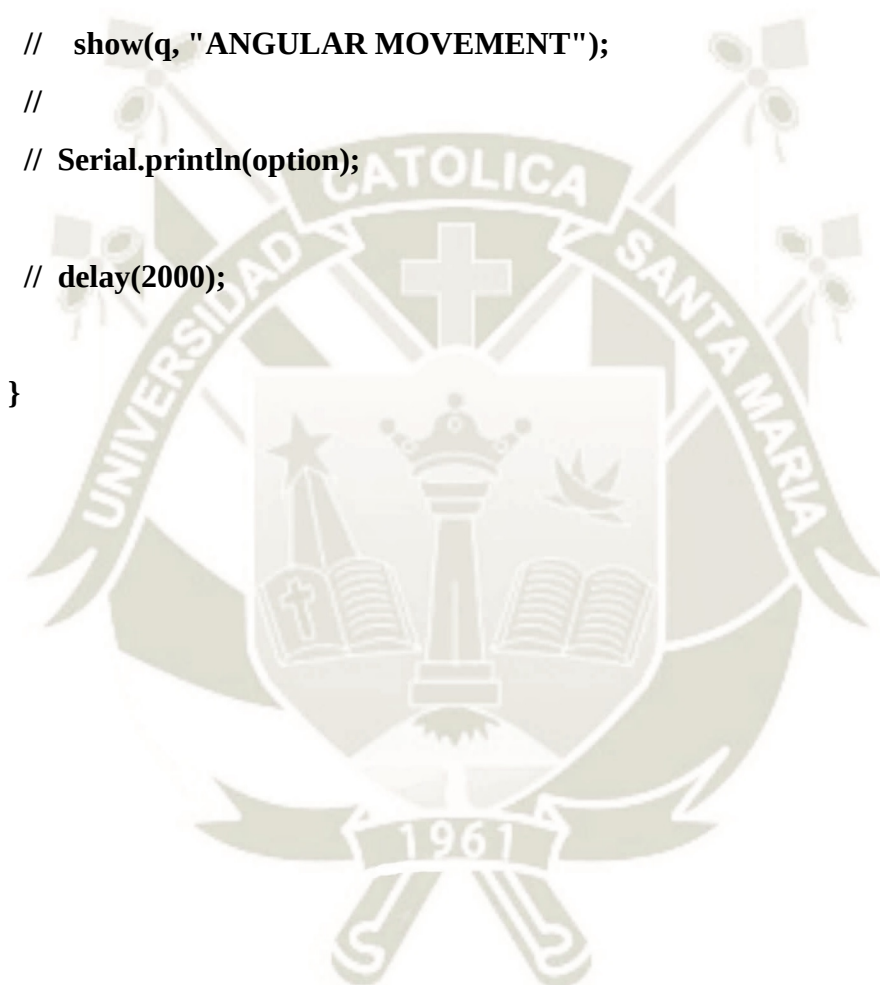
```
break;
```

```
}
```

```
SERVO_MOVEMENT(q);
```



```
if (cont_only == 1) {  
    one_only = true;  
}  
else {  
    cont_only++;  
}  
  
// show(q, "ANGULAR MOVEMENT");  
//  
// Serial.println(option);  
  
// delay(2000);  
}
```



PROGRAMA AUXILIAR

```
#include <Servo.h>
#include <SoftwareSerial.h>

ServoAo1; // declaramos variables para los 18 servos.
ServoAo2;
Servo Ao3;
Servo Bo1;
Servo Bo2;
Servo Bo3;
Servo Do1;
Servo Do2;
Servo Do3;
Servo Eo1;
Servo Eo2;
Servo Eo3;
const int pos=80;
char dataIn = 'S';
char determinant;
char det;

void setup()
{
  Serial.begin(9600); char det;
  Ao1.attach(2); Ao2.attach(3); Ao3.attach(4);
  Bo1.attach(5); Bo2.attach(6); Bo3.attach(7);
  Do1.attach(8); Do2.attach(9); Do3.attach(10);
  Eo1.attach(11); Eo2.attach(12); Eo3.attach(13);
}
```

```
void loop()  
{  
posinicial();  
delay (500);  
det=check();  
while(1)  
{  
while (det == 'F') //if incoming data is a F, move forward  
{  
paso1();  
delay(400);det = check();  
}  
while (det == 'B') //if incoming data is a B, move back  
{  
paso2();  
delay(400);  
det = check();  
}  
while (det == 'L') //if incoming data is a L, move left  
{  
paso4();  
delay(400);  
det = check();  
}  
while (det == 'R') //if incoming data is a R, move right  
{  
paso5();  
delay(400);  
det = check();
```



```
}  
  
while (det == 'S') //if incoming data is a S, stop  
{  
    posinicial();  
    det = check();  
}  
  
}  
  
void paso1()  
{  
    //PASO 1  
    delay(200);  
    Bo1.write(60); // PATA DEL MEDIO derecho  
    Bo2.write(120);  
    Bo3.write(110);  
    Do1.write(70); //PRIMERAS PATAS lado IZQUIERDO  
    Do2.write(30);  
    Do3.write(10);  
    delay(200);  
    Bo1.write(90); // PATA DEL MEDIO derecho  
    Bo2.write(120);  
    Bo3.write(110);  
    Do1.write(50); //PRIMERAS PATAS lado IZQUIERDO  
    Do2.write(30);  
    Do3.write(10);  
    delay(200);  
    Bo1.write(90); // PATA DEL MEDIO derecho  
    Bo2.write(100);  
    Bo3.write(110);
```

Do1.write(50); // PRIMERAS PATAS lado IZQUIERDO

Do2.write(50);

Do3.write(10);

delay(200);

//return

Bo1.write(60);// PATA DEL MEDIO derecho

Bo2.write(100);

Bo3.write(110);

Do1.write(70); // PRIMERAS PATAS lado IZQUIERDO

Do2.write(50);

Do3.write(10);

delay(200);

Ao1.write(80);// PRIMERAS PATAS lado DERECHO

Ao2.write(100);

Ao3.write(110);

Eo1.write(70); // PATA DEL MEDIO izquierdo

Eo2.write(40);

Eo3.write(40);

delay(200);

Ao1.write(100);// PRIMERAS PATAS lado DERECHO

Ao2.write(100);

Ao3.write(110);

Eo1.write(40); // PATA DEL MEDIO izquierdo

Eo2.write(40);

Eo3.write(40);

delay(200);

Ao1.write(100);// PRIMERAS PATAS lado DERECHO

Ao2.write(80);

Ao3.write(110);

```
Eo1.write(50); // PATA DEL MEDIO izquierdo
Eo2.write(60);
Eo3.write(40);
delay(200);
//return

Ao1.write(80);// PRIMERAS PATAS lado DERECHO
Ao2.write(80);
Ao3.write(110);
Eo1.write(70); // PATA DEL MEDIO izquierdo
Eo2.write(60);
Eo3.write(40);
delay(200);
}
void paso2()//atras
{
//PASO 2
delay(200);
Bo1.write(60);// PATA DEL MEDIO derecho
Bo2.write(120);
Bo3.write(110);
Do1.write(70); //PRIMERAS PATAS lado IZQUIERDO
Do2.write(30);
Do3.write(10);
delay(200);
Bo1.write(30);// PATA DEL MEDIO derecho
Bo2.write(120);
Bo3.write(110);
Do1.write(90); //PRIMERAS PATAS lado IZQUIERDO
Do2.write(30);
```



```
Do3.write(10);  
delay(200);  
Bo1.write(30);// PATA DEL MEDIO derecho  
Bo2.write(100);  
Bo3.write(110);  
Do1.write(90); // PRIMERAS PATAS lado IZQUIERDO  
Do2.write(50);  
Do3.write(10);  
delay(200);  
//return  
Bo1.write(60);// PATA DEL MEDIO derecho  
Bo2.write(100);  
Bo3.write(110);  
Do1.write(70); // PRIMERAS PATAS lado IZQUIERDO  
Do2.write(50);  
Do3.write(10);  
delay(200);  
Ao1.write(80);// PRIMERAS PATAS lado DERECHO  
Ao2.write(100);  
Ao3.write(110);  
Eo1.write(70); // PATA DEL MEDIO izquierdo  
Eo2.write(40);  
Eo3.write(40);  
delay(200);  
Ao1.write(60);// PRIMERAS PATAS lado DERECHO  
Ao2.write(100);  
Ao3.write(110);  
Eo1.write(100); // PATA DEL MEDIO izquierdo  
Eo2.write(40);
```

```
Eo3.write(40);
delay(200);
Ao1.write(60);// PRIMERAS PATAS lado DERECHO
Ao2.write(80);
Ao3.write(110);
Eo1.write(100); // PATA DEL MEDIO izquierdo
Eo2.write(60);
Eo3.write(40);
delay(200);
//return
Ao1.write(80);// PRIMERAS PATAS lado DERECHO
Ao2.write(80);
Ao3.write(110);
Eo1.write(70); // PATA DEL MEDIO izquierdo
Eo2.write(60);
Eo3.write(40);
delay(200);
}
void paso4()//IZQUIERDA
{
//PASO 4
delay(200);
Do1.write(70); //PRIMERAS PATAS lado IZQUIERDO
Do2.write(30);
Do3.write(10);
Bo1.write(60);// PATA DEL MEDIO derecho
Bo2.write(120);
Bo3.write(110);
delay(200);
```

Do1.write(50); //PRIMERAS PATAS lado IZQUIERDO

Do2.write(30);

Do3.write(10);

Bo1.write(30);// PATA DEL MEDIO derecho

Bo2.write(120);

Bo3.write(110);

delay(200);

Do1.write(50); // PRIMERAS PATAS lado IZQUIERDO

Do2.write(50);

Do3.write(10);

Bo1.write(30);// PATA DEL MEDIO derecho

Bo2.write(100);

Bo3.write(110);

delay(200);

//return

Bo1.write(60);// PATA DEL MEDIO derecho

Bo2.write(100);

Bo3.write(110);

Do1.write(70); // PRIMERAS PATAS lado IZQUIERDO

Do2.write(50);

Do3.write(10);

delay(200);

Ao1.write(80);// PRIMERAS PATAS lado DERECHO

Ao2.write(100);

Ao3.write(110);

Eo1.write(70); // PATA DEL MEDIO izquierdo

Eo2.write(40);

Eo3.write(40);

delay(200);

Ao1.write(60);// PRIMERAS PATAS lado DERECHO

Ao2.write(100);

Ao3.write(110);

Eo1.write(40); // PATA DEL MEDIO izquierdo

Eo2.write(40);

Eo3.write(40);

delay(200);

Ao1.write(60);// PRIMERASPATAS lado DERECHO

Ao2.write(80);

Ao3.write(110);

Eo1.write(50); // PATA DEL MEDIO izquierdo

Eo2.write(60);

Eo3.write(40);

delay(200);

//return

Ao1.write(80);// PRIMERAS PATAS lado DERECHO

Ao2.write(80);

Ao3.write(110);

Eo1.write(70); // PATA DEL MEDIO izquierdo

Eo2.write(60);

Eo3.write(40);

delay(200);

}

void paso5()//DERECHA

{

//PASO 5

delay(200);

Ao1.write(80);// PRIMERAS PATAS lado DERECHO

Ao2.write(100);

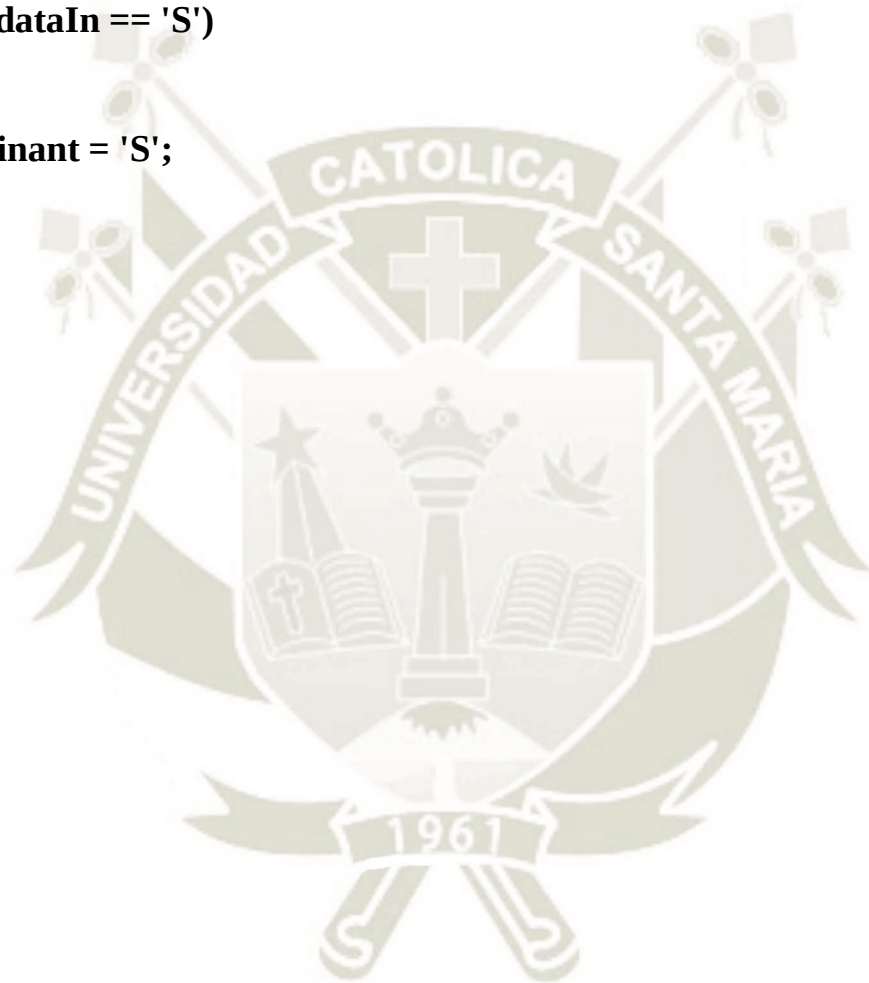
```
Ao3.write(110);  
Eo1.write(70); // PATA DEL MEDIO izquierdo  
Eo2.write(40);  
Eo3.write(40);  
delay(200);  
Ao1.write(100);// PRIMERAS PATAS lado DERECHO  
Ao2.write(100);  
Ao3.write(110);  
Eo1.write(100); // PATA DEL MEDIO izquierdo  
Eo2.write(40);  
Eo3.write(40);  
delay(200);  
Ao1.write(100);// PRIMERAS PATAS lado DERECHO  
Ao2.write(80);  
Ao3.write(110);  
Eo1.write(100); // PATA DEL MEDIO izquierdo  
Eo2.write(60);  
Eo3.write(40);  
delay(200);  
//return  
Ao1.write(80);// PRIMERAS PATAS lado DERECHO  
Ao2.write(80);  
Ao3.write(110);  
Eo1.write(70); // PATA DEL MEDIO izquierdo  
Eo2.write(60);  
Eo3.write(40);  
delay(200);  
Bo1.write(60);// PATA DEL MEDIO derecho  
Bo2.write(120);
```

```
Bo3.write(110);
Do1.write(70); //PRIMERAS PATAS lado IZQUIERDO
Do2.write(30);
Do3.write(10);
delay(200);
Bo1.write(90);// PATA DEL MEDIO derecho
Bo2.write(120);
Bo3.write(110);
Do1.write(90); //PRIMERAS PATAS lado IZQUIERDO
Do2.write(30);
Do3.write(10);
delay(200);
Bo1.write(90);// PATA DEL MEDIO derecho
Bo2.write(100);
Bo3.write(110);
Do1.write(90); // PRIMERAS PATAS lado IZQUIERDO
Do2.write(50);
Do3.write(10);
delay(200);
//return
Bo1.write(60);// PATA DEL MEDIO derecho
Bo2.write(100);
Bo3.write(110);
Do1.write(70); // PRIMERAS PATAS lado IZQUIERDO
Do2.write(50);
Do3.write(10);
delay (200);
}
void posinicial();//PARADA
```



```
{  
  Ao1.write(80);// PRIMERAS PATAS lado DERECHO  
  Ao2.write(80);  
  Ao3.write(110);  
  Bo1.write(60);// PATA DEL MEDIO derecho  
  Bo2.write(100);  
  Bo3.write(110);  
  Do1.write(70); // PRIMERAS PATAS lado IZQUIERDO  
  Do2.write(50);  
  Do3.write(10);  
  Eo1.write(70); // PATA DEL MEDIO izquierdo  
  Eo2.write(60);  
  Eo3.write(40);  
}  
int check()  
{  
  if (Serial.available() > 0)  //Check for data on the serial lines.  
  {  
    dataIn = Serial.read(); //Get the character sent by the phone and store it in  
'dataIn'. if (dataIn == 'F')  
    {  
      determinant = 'F';  
    }  
    else if (dataIn == 'B')  
    {  
      determinant = 'B';  
    }  
    else if (dataIn == 'L')  
    {
```

```
determinant = 'L';  
}  
else if (dataIn == 'R')  
{  
    determinant = 'R';  
}  
else if (dataIn == 'S')  
{  
    determinant = 'S';  
}  
}  
}
```



Programa de adquisición de datos de CO2 y temperatura.

```
#include "DHT.h"

#include <SPI.h>  // incluye libreria interfaz SPI
#include <SD.h>   // incluye libreria para tarjetas SD
#define DHTPIN 2  // Pin donde está conectado el sensor

#define DHTTYPE DHT22  // Sensor DHT22
int VALOR;  // almacena valor leído del sensor MQ135
DHT dht(DHTPIN, DHTTYPE);

#define SENSOR 4  // constante SENSOR en pin digital 4 (senal de DHT22)
int TEMPERATURA; // variable para almacenar valor de temperatura

#define SSpin 10  // Slave Select en pin digital 10

File archivo;  // objeto archivo del tipo File

void setup() {
  Serial.begin(9600); // inicializa monitor serie a 9600 bps
  dht.begin();       // inicializacion de sensor

  Serial.println("Iniciando tarjeta ..."); // texto en ventana de monitor
  if (!SD.begin(SSpin)) { // inicializacion de tarjeta SD
    Serial.println("fallo en inicializacion !");// si falla se muestra texto
correspondiente y
    return;  // se sale del setup() para finalizar el programa
  }

  Serial.println("inicializacion correcta"); // texto de inicializacion correcta
```



```
archivo = SD.open("datos.txt", FILE_WRITE); // apertura para  
lectura/escritura de archivo datos.txt
```

```
if (archivo) {
```

```
    for (int i=1; i < 31; i++){ // bucle repite 30 veces
```

```
        TEMPERATURA = dht.readTemperature(); // almacena en variable  
valor leído de temperatura
```

```
        HUMEDAD = dht.readHumidity(); // almacena en variable valor leído  
de humedad
```

```
        archivo.print(i); // escribe en tarjeta el valor del indice
```

```
        archivo.print(","); // escribe en tarjeta una coma
```

```
        archivo.print(TEMPERATURA); // escribe en tarjeta el valor de  
temperatura
```

```
        archivo.print(","); // escribe en tarjeta una coma
```

```
        archivo.println(VALOR); // escribe en tarjeta el valor de co2 y salto de  
linea
```

```
        Serial.print(i); // escribe en monitor el valor del indice
```

```
        Serial.print(","); // escribe en monitor una coma
```

```
        Serial.print(TEMPERATURA); // escribe en monitor el valor de  
temperatura
```

```
        Serial.print(","); // escribe en monitor una coma
```

```
        Serial.println(VALOR); // escribe en monitor el valor de co2 y salto de  
linea
```

```
        delay(1000); // demora de 1 segundo
```

```
    }
```

```
    archivo.close(); // cierre de archivo
```

```
    Serial.println("escritura correcta"); // texto de escritura correcta en  
monitor serie
```

```
} else {
```

```
    Serial.println("error en apertura de datos.txt"); // texto de falla en  
    apertura de archivo
```

```
}
```

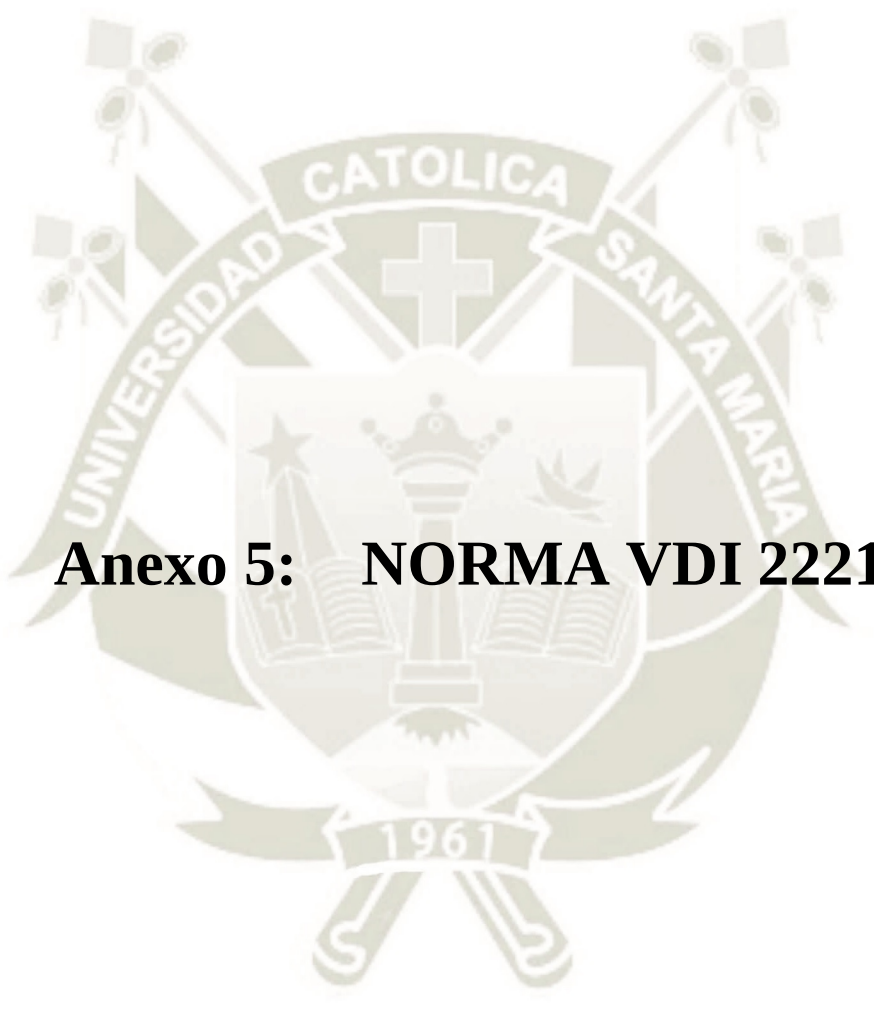
```
}
```

```
void loop() { // funcion loop() obligatoria de declarar pero no utilizada
```

```
    // nada por aqui
```

```
}
```





Anexo 5: NORMA VDI 2221

Según (DÍAZ, 2017) nos habla de la metodología de diseño VDI221.

VDI 2221

La VDI 2221 es una variación de la famosa 2222, esta analiza y entiende el problema a profundidad, es especialmente útil para la realización de productos pero principalmente para el caso de rediseños o mejoras de productos.

Se fracciona en 7 etapas generales.

1. Especificaciones

- Clarificar y definir la tarea

2. Estructura de funciones

- Definir funciones
- Definir estructuras

3. Búsqueda de soluciones

- Buscar principales soluciones y combinaciones

4. Estructura de modelo

- Decidir en módulos realizables

5. Arreglos

- Arreglos preliminares
 - Desarrollar arreglos de módulos claves □

Arreglos definitivos

- Completar arreglo general

6. Documentación

- Preparar instrucciones de operación y producción

7. Realización

- Realización del producto

Karl T. Ulrich

El método de Karl T. Ulrich abarca todos los campos que intervienen en el desarrollo de un proyecto de diseño, desde el grupo de diseñadores, hasta mercadeo y finanzas.

Para el proceso de desarrollo de producto, Ulrich emplea una metodología estructurada fundamentada en seis fases. Cada fase describe las actividades a desarrollar en los cuatro departamentos que comúnmente existen en una empresa (Mercadeo; diseño; manufactura; departamento de administración, investigación y finanzas).

Es considerado como uno de los métodos más completo y descriptivo.

1. Planeación

- Articular y definir segmentos de mercado
- Considerar arquitectura de producto □ Identificar restricciones del producto
- Finanzas, investigación y administración

Resultado: Se hace la planeación total del proyecto, y se obtiene una aprobación que precede al desarrollo del producto.

2. Desarrollo del concepto

- Identificar necesidad del cliente
- Identificar usuarios
- Identificar productos

- Desarrollo de conceptos de diseño
- Construir y probar prototipos
- Calcular costos y factibilidad de producción
- Estudiar finanzas y patentes

Resultado: Se identifican las necesidades del mercado objetivo, se generan y evalúan conceptos de productos alternativos, y se seleccionan uno o más conceptos para el desarrollo y para prueba.

3. Diseño a nivel de sistema

- Desarrollar plan para opciones de producto
- Generar arquitecturas alternativas del producto
- Identificar proveedores, establecer costos
- Análisis de fabricar contra adquirir (finanzas)

Resultado: Se define la arquitectura del producto y el desglose de este en subsistemas y componentes. Se define también el esquema de ensamble final para el sistema de producción. Da como resultado una distribución geométrica del producto, una especificación funcional de cada subsistema y un diagrama de flujo de proceso preliminar para la secuencia del ensamble final.

4. Diseño de detalles

- Desarrollar plan de mercadotecnia
- Definir geometría, materiales y documento de diseño industrial
- Definir procesos de producción y comenzar abastecimiento de herramientas

Resultado: Se establece la especificación completa de la geometría, materiales y tolerancias de todas las partes que sean únicas en el producto, y la identificación de todas las partes estándar que se van a adquirir de los proveedores.

Se establece un plan del proceso y se designa el herramental para cada parte que se va a fabricar dentro del sistema de producción. Además se abordan los costos de producción y la confiabilidad.

5. Prueba y refinamiento

- Desarrollar materiales de producción y lanzamiento
- Prueba de fiabilidad, duración , desempeño y aprobación legal
- Facilitar el arranque del proveedor, depurar los procesos de fabricación y ensamble
- Desarrollar plan de ventas

Resultado: Involucra la construcción y evaluación de múltiples versiones de producción previas del producto. Se desarrollan los prototipos alfa y beta.

6. Producción piloto

- Primera producción a disposición del cliente
- Evaluar resultados
- Comenzar la operación de todo el sistema de producción

Resultado: En esta fase el producto se fabrica utilizando el sistema de producción pretendido. Tiene como finalidad capacitar a la fuerza laboral y resolver problemas que persistan en los procesos de producción. Los productos que resultan de esta producción piloto son enviados a los clientes preferidos y evaluados de manera cuidadosa para eliminar cualquier defecto que aun exista. La transición de la producción piloto a la continua es por lo general gradual.

Pahl y Beitz

Secuencia de trabajo en el diseño según Pahl y Beitz

1. Problema

- Clarificar el problema
- Generar listado de requerimientos

2. Concepción

- Identificar los problemas esenciales
- Establecer las estructuras de las funciones
- Buscar principios de solución

- Combinar y confirmar variantes de conceptos
- Evaluar criterios técnicos y económicos

3. Proyecto

- Determinar arreglos preliminares y diseños de forma
- Seleccionar los mejores arreglos preliminares
- Refinar y evaluar contra los criterios técnicos y económicos
- Optimizar y completar los diseños de forma
- Verificar si hay errores y la eficiencia en costos
- Preparar la lista de partes preliminar y documentos de producción
- Finalizar los detalles
- Completar los dibujos de detalle y los documentos de producción
- Verificar todos los documentos de fabricación

4. Elaboración

- Elaboración
- Documentar información
- Solución

QFD

Quality function deployment o despliegue de la función de calidad.

Podemos fabricar un producto con unas excelentes prestaciones, a un bajo precio y, sin embargo, fracasar por no tener la acogida esperada en el mercado. Esto nos indica que debemos conocer las expectativas del cliente, y traducirlas adecuadamente a las características del producto. El QFD supone una metodología que permite sistematizar la información obtenida del usuario hasta llegar a definir las características de calidad del producto/servicio.

1. Identificar y jerarquizar a los clientes

- Comprender a los clientes
- Seleccionar segmentos de usuarios

2. Identificación de las expectativas del cliente

- Conocer las expectativas del cliente por medio de:
 - Grupos de discusión
 - Informes sobre quejas
 - Estudios existentes en base a encuestas realizadas
 - Informes de responsables de puntos de venta
 - Publicaciones y artículos
 - Informaciones sobre la competencia

3. Conversión de la información en descripciones verbales específicas

- Obtener medidas concretas

4. Elaboración y administración de la encuesta a clientes

- Administrar una encuesta a usuarios (usuarios nuestros, que conozcan también la competencia)

5. Despliegue de la calidad demandada

- Obtener datos
- Generar cuadro de la calidad demandada
- Generar cuadro de la calidad planificada
- Determinar en qué aspectos hay que comprometer mayor esfuerzo

6. Despliegue de las características de calidad

- Identificar elementos susceptibles a mejora
- Elaboración de lista de indicadores (hecha por un grupo interdisciplinar) □ La lista resultante deberá ser exhaustiva y consistente

Roth

Diseño metódico según el proceso algorítmico de elección con catalogo de Roth, separado por fases.

1. Fase de formulación del problema

- Planteamiento
- Análisis medio del producto
- Formulación de la función
- Listado de requerimientos

Resultado: formulación satisfactoria del problema con requerimientos primarios y secundarios.

2. Fase funcional (desarrollo)

- Desarrollo de la estructura (estructura general de la función)
- Flujograma lógico
- Elementos funcionales
- Funciones físicas (generales y específicas)

Resultado: Funciones que se cumplan.

3. Fase de determinación de formas

- Determinación de forma de acuerdo a la geometría del producto
- Determinación de forma de acuerdo al material del producto
- Determinación de forma de acuerdo a la fabricación del producto
- Función y costos efectivos nominales

4. Documentación

- Documentación para la fabricación (resultados de la actividad de diseño)

TEMA 1. Introducción a la Metodología de Diseño Industrial

- 1.1. Introducción al Diseño Industrial. Diseñar y Proyectar.
- 1.2. Concepto y evolución de la Metodología en el Diseño Industrial
- 1.3. Propuesta y clasificación de las metodologías de Diseño Industrial
- 1.4. Modelo organizativo
- 1.5. Los productos industriales. Ciclo de vida de un producto.
- 1.6. El Diseño en la Empresa.
- 1.7. Norma VDI 2221.

Según (Chica, 2018) en su tema de introducción de diseño industrial nos habla de la siguiente norma VDI2221.

En este primer capítulo se van a repasar algunos conceptos importantes para el seguimiento de la asignatura.

Introducción al Diseño Industrial. Diseñar y Proyectar

La actividad profesional del Diseño Industrial, encara un complejo proceso a seguir en la generación de ideas o conceptos de diseño, ya que los productos a concebir cuentan con distintos caracteres de uso, funcionales, estructurales, productivos y de mercado, -lo que origina que los proyectos por desarrollar sean diferentes, "no es posible establecer hasta no conocer el problema de diseño a resolver, criterios o normas para solucionarlos, pues todos y cada uno de ellos poseen características únicas que responden a requerimientos o restricciones específicas; sin embargo, sí es posible enunciar ciertas etapas generales aplicables a todo proyecto.

Un título para ganar competitividad

INVERTIR EN DISEÑO ES UN BUEN NEGOCIO

Esta afirmación, que muchos pensarán proviene de algún profesional del sector del diseño, es sin embargo una idea de la que gobiernos de países ya desarrollados como países en vías de desarrollo, y muchos empresarios participan, y por supuesto se benefician. La prueba es que revistas tan alejadas de los círculos de diseñadores, como las prestigiosas Business Week, o el Wall Street Journal, tienen espacios permanentes para el negocio del diseño.

¿Por qué podemos afirmar esto ahora, y no hace 15 años cuando se programaron las carreras de diseño actuales?

Fundamentalmente por dos razones. Por un lado el cambio de paradigma descrito en el apartado anterior hace enfocar los esfuerzos en consolidar lo alcanzado, como fue la introducción en los años 80 del diseño como factor de innovación en nuestras empresas y productos, y afrontar los nuevos retos de competitividad que plantea la ampliación de la Unión Europea y los efectos de la globalización y deslocalización.

Por otro lado, hasta mediados y finales de los 90 no se habían recogido datos de los beneficios que habían aportado las inversiones que las empresas hicieron en diseño. Las cifras resultantes son especialmente reveladoras.

A pesar de ello, tanto la administración, como los empresarios, los formadores y los profesionales deben tener datos convincentes de los beneficios que puede aportar invertir en diseño en el sentido más amplio del término, desde la contratación de servicios de diseño profesionales, mejorar la gestión del diseño y desarrollar una estrategia de diseño antes de invertir los siempre limitados recursos en tiempo y dinero.

El cambio de paradigma y los datos de los beneficios del diseño nos permiten hacer dos tipos de valoraciones se podrían hacer para fundamentar el diseño industrial del futuro: una cualitativa y otra cuantitativa

Valoración cualitativa

En estas últimas décadas se ha constatado que el diseño antes que una actividad profesional debe entenderse como un proceso, una tecnología o un saber hacer, una disciplina y finalmente un servicio.

Efectivamente, coincidiremos en que el diseño es un proceso analítico, técnico y creativo que nos lleva a determinar un producto concreto. A la vez se puede entender como tecnología por cuanto coordina unas habilidades intelectuales con las instrumentales para obtener el fin. Es obvio que esa coordinación obedece a una disciplina proveniente de una formación de capacidades analíticas, de conocimientos técnicos sumados a una sensibilidad cultural y capacidad creativa.

Estamos hablando por lo tanto del diseño como un saber-hacer del cual dependemos para minimizar el riesgo en la generación, desarrollo y lanzamiento de nuevos productos.

Por esta razón, una mayor afinidad entre la definición de estrategias empresariales y una gestión correcta del proceso de desarrollo de nuevos productos, nos permitirá optimizar todos los recursos que en esta tarea se inviertan.

Valoración cuantitativa

Hasta hace algunos años no se disponía de información sobre la amortización y, generación de beneficios de la inversión en diseño y en una efectiva gestión del mismo.

Solo se conocen por los medios de comunicación los casos de marcas más populares, sean nacionales o extranjeras, pero que, normalmente, se desconocen los datos del conjunto empresarial que es la que describiría nuestra realidad industrial.

El valor del diseño en el Reino Unido

Sin embargo, recientemente se han hecho públicos algunos estudios que vienen a demostrar que un buen diseño es un buen negocio.

Los primeros resultados, aunque fueron realizados en el Reino Unido y en los Estados Unidos respectivamente, pueden ser útiles por la proximidad del tipo de empresas analizadas y los productos desarrollados. En el Reino Unido, la investigación fue desarrollada por el Design Innovation Group de la Open University y la Universidad de Manchester, e hizo un seguimiento sobre 221 proyectos de diseño de producto, diseño industrial y gráfico de los 300 que el gobierno británico, a través del Design Council había subvencionado parcialmente entre 1982 y 1987. La mayoría de las empresas tenían experiencia en contratar servicios de diseño, más de la mitad tenían menos de 100 empleados y el resto tenían menos de 500 trabajadores. Fueron proyectos desarrollados, por lo tanto, por pequeñas y medianas empresas, de las cuales la mitad eran del sector del mueble y textil, una tercera parte eran proyectos de diseño industrial en general, desde ventiladores, componentes electrónicos, objetos de cocina y calzado; y el resto eran diseño de packaging y manuales de uso.

En líneas generales el análisis reveló que:

- Alrededor del 90% de los proyectos generaron beneficio, recuperando las inversiones en un término medio de 15 meses desde el lanzamiento del producto.

- El 48 % de los proyectos recuperaron sus costes totales, incluyendo utillajes, en menos de un año después de haber lanzado el producto.
- En aquellos casos donde se pudo comparar los nuevos productos con los antiguos, las ventas aumentaron el 41 % por término medio.
- Más del 40 % de las ventas resultantes de los proyectos de diseño industrial eran de exportación.
- La cuarta parte de los proyectos abrieron nuevos mercados nacionales, y el 12 % abrieron mercados extranjeros.
- Se percibieron otros beneficios tangibles como la reducción de costes de producción, el ahorro de stocks, el beneficio social de la creación de empleo, la diversificación industrial y apertura de nuevos mercados, y hasta la creación de nuevas empresas.

Pero también se reconocieron otros beneficios intangibles como la mejora en la imagen de la compañía, la apertura de expectativas al cliente habitual, el cambio de actitud de la dirección hacia el diseño que pasó de verlo como un gasto a verlo como una inversión rentable. Este aspecto puede ser especialmente significativo en la actualidad en la que el consumidor más culto o más informado, y por lo tanto más exigente, se fija en la cultura de producto que genera la empresa emisora, pudiendo ser factor determinante en la elección del consumidor.

Preguntando a las compañías sobre la influencia del diseño con respecto a otros factores en los resultados finales, el 85% afirmaban que de todos los factores, el diseño suponía más de la mitad del éxito comercial.

La mitad de las empresas aumentaron la relación con los servicios de diseño, y una tercera parte mejoraron su actitud frente al diseño. En particular notaron que habían aprendido las bases de la gestión del diseño, especialmente lo que se refería a seleccionar, encargar y dirigir a diseñadores profesionales. Este hecho es especialmente significativo porque viene a confirmar la valoración de lo intangible, al enriquecer el saber-hacer de la empresa con este concepto del diseño como tecnología.

El valor del diseño en Estados Unidos

En el caso norteamericano, varios son los estudios que se han realizado y que vale la pena mencionar. El primero de ellos se publicó a principios de los ochenta recogiendo la experiencia de más de 14.000 productos en 1.000 empresas y que permitió establecer los criterios para la gestión de nuevos productos para la década siguiente.

Estos factores de éxito para nuevos productos se podrían resumir en:

- Orientación hacia el mercado: el consumidor debe percibir una mejora sustancial con respecto a los competidores. De no existir esa mejora, puede ser preferible abandonar el desarrollo del producto.
- Estudio de factibilidad anticipado y especificaciones claras: viene a corroborar la idea de que es preferible dedicar atención al concepto (qué hacer) antes de ponerse a diseñar (cómo hacerlo). Las encuestas arrojaban datos significativos: para que cinco productos dieran beneficios había que generar
- Calidad en el proceso del desarrollo de nuevos productos: se alcanza cuando el equipo de desarrollo está en total sintonía con la capacidad tecnológica de la empresa y con el departamento de marketing y ventas.

El segundo estudio fue en 1995 con ocasión del décimo aniversario del Premio a la Excelencia en Diseño Industrial (IDEA, Industrial Design Excellence Award) que otorga la Sociedad de Diseñadores Industriales de América (IDSA, Industrial Designers Society of America). En aquel aniversario pasaron de aplicar el criterio inicial de premiar al diseño que había convertido al cliente en líder del mercado a analizar cómo se producía ese liderazgo, aplicando las siguientes preguntas a los que optaban a premio:

- En cuanto aumentó la cuota de mercado de la empresa en el segmento en el que se lanzó el producto.
- Si hubieron otros factores además del diseño (como una nueva promoción) que contribuyeran al éxito del producto.
- Cuál es el crecimiento anual de ventas por cada año que el producto está en el mercado (Comparado con la media del mercado)
- En cuanto aumentó el diseño las ventas del producto en su segmento.
- En cuanto aumentó la publicidad.
- Como ha satisfecho el diseño al consumidor.
- Identificar el incremento de los márgenes de beneficios como resultado de la disminución de costes en materiales y/o procesos comparados con los anteriores productos.
- Identificar mejoras para el usuario en términos funcionales, de servicio y de vigencia del producto que justifique su precio.
- Si se utilizaron patentes, si ayudaron a mantener la cuota de mercado o si habían sido bien defendidas.

De nuevo este último dato, viene a revalorar la protección de la innovación como forma de generar un saber-hacer propio de la empresa, que en un momento dado puede convertirse en un valor transferible. Las patentes son por tanto activos de la empresa.

Tres características cumplían los premiados desde que se implantó el premio en 1985.

- Trabajo interdisciplinar
- Estética integrada con la función y la utilidad, facilitando especialmente el uso.
- El éxito comercial y la innovación vienen al solventar las necesidades reales del consumidor.

Estas coincidencias las podemos ver incluidas entre los factores clave para el éxito de un producto resultante de un proyecto de innovación:

A nivel estratégico:

1. Orientación al mercado. Conocimiento de las necesidades del usuario.
2. Coherencia con la orientación de la empresa.
3. Control de las ventajas competitivas del producto.
4. Apoyo total de la dirección y toda la organización al proyecto.
5. Eficacia en el sistema de selección y validación de proyectos. Creatividad.
6. Equipo interdisciplinar y coordinación entre departamentos.
7. Eficacia en la dirección del proyecto. Control del proceso de desarrollo.
8. Diferenciación del producto por el diseño.
9. Adaptación del producto a las capacidades tecnológicas, productivas y de marketing de la empresa.

Nivel de lanzamiento

10. Previsiones de tiempos para la introducción en el mercado
11. Comercialización eficiente.

Definición de Diseño

La palabra diseño proviene del término italiano disegno, que significa delineación de una figura, realización de un dibujo.

En la actualidad, el concepto diseño tiene una amplitud considerable, de tal modo que especifica su campo de acción acompañándose de otros vocablos. Así tenemos: diseño industrial, diseño artesanal, diseño gráfico, diseño textil, diseño mecánico, diseño estructural, diseño de asentamientos humanos, diseño arquitectónico, diseño de plantas industriales, diseño de proceso, etc.

La producción masiva a partir de la revolución industrial sentó los principios básicos para que el término diseño se entendiera como un nuevo concepto internacional desde los primeros años del presente siglo.

De acuerdo a lo que plantean Cross, Elliott y Roy; Diseño en la actualidad se toma como innovación, como creación, como avance, como solución renovadora, como un nuevo modo de relacionar un número de variables o factores, como una nueva forma de expresión, como el logro de una mayor eficacia.

La dificultad de escribir (y de hablar) sobre diseño reside en que esta palabra tiene diferentes significados, y según quien la emplee, puede significar:

- Un producto (p. ej.: "Este nuevo modelo de papel pintado es un diseño mío");
- Un plano (p. ej.: "Este dibujo es mi diseño para el nuevo edificio");
- Un proceso (p. ej.: "Voy a diseñar una nueva forma de hacer el trabajo").
- Un servicio (e. ej.: "Se ha diseñado un nuevo servicio como oferta para los clientes").

Y puede conceptuarse en forma:

Racional:

"Una actividad orientada a determinados fines, para la solución de problemas" (L. Bruce y Archer);

Administrativa:

"El esfuerzo consciente de imponer un orden significativo" (Víctor Papanek);

Mística:

"La realización de un acto de fe muy complicado" (J. Christopher Jones).

En vista de esta diversidad de significados e intuiciones respecto al diseño, tal vez sea mejor adoptar la definición de J. Christopher Jones: "El efecto de diseñar es iniciar un cambio en las cosas realizadas por el hombre". Aquí se desplaza claramente el problema de la definición de ¿qué es diseñar? a ¿cuál es el efecto de diseñar? Cualquier actividad que inicia un cambio en las cosas realizadas por el hombre es, pues, una actividad de diseño.

La adopción de esta definición general de diseño, justifica la amplia gama de epígrafes a que se puede referir dicho concepto, de tal modo que el diseñador no tiene que ser necesariamente un ingeniero o un arquitecto o cualquiera de los profesionales del

diseño (industrial, gráfico, textil, de asentamientos humanos); el diseñador puede ser administrador, político, consumidor, abogado, sindicalista, carnicero, panadero o fabricante de velas.

En ocasiones, con toda seguridad, el diseñador será usted mismo.

Definición de Diseño Industrial

Así como es problemático dar una definición del concepto diseño, más lo es cuando se trata del término diseño industrial, ya que una somera mirada a la bibliografía en que se utiliza dicho concepto, nos dará una idea de las muy diversas maneras de concebir el papel y los objetivos de esta actividad.

Sin embargo, a continuación se expresan dos concepciones del diseño industrial, no para que sean adoptadas sin más, sino para que se analicen a lo largo de la preparación académica en la universidad, de tal manera que al concluir su formación profesional cuente con una concepción personal.

En primer lugar se enunciará la definición del término diseño industrial oficialmente reconocido por el ICSID (International Council of Societies of Industrial Design), cuyo autor es el reconocido maestro de la teoría del diseño. Tomás Maldonado. Él la dio a conocer en el año de 1961, en Venecia, Italia, durante una conferencia titulada Education for Design, en los siguientes términos:

El diseño industrial es una actividad proyectual que consiste en determinar las propiedades formales de los objetos producidos industrialmente. Por propiedades formales no hay que entender tan sólo las características exteriores, sino, sobre todo, las relaciones funcionales y estructurales que hacen que un objeto tenga una unidad coherente desde el punto de vista tanto del productor como del usuario, puesto que, mientras la preocupación exclusiva por los rasgos exteriores de un objeto determinado conlleva el deseo de hacerlo aparecer más atractivo o también disimular sus debilidades constitutivas, las propiedades formales de un objeto son siempre el resultado de la integración de factores diversos, tanto si son de tipo funcional, cultural, tecnológico o económico. Dicho de otra manera, así como los caracteres exteriores hacen referencia a cualquier cosa como una realidad extraña, es decir, no ligada al objeto y que no se ha desarrollado con él, de manera contraria las propiedades formales constituyen una realidad que corresponde a su organización interna, vinculada a ella y desarrollada a partir de ella.

La segunda concepción que sobre el término diseño industrial se transcribe es producto del autor Gerardo Rodríguez:

El diseño industrial es una disciplina proyectual, tecnológica y creativa, que se ocupa tanto de la proyección de productos aislados o sistemas de productos, como del estudio de las interacciones inmediatas que tienen los mismos con el hombre y con su modo particular de producción y distribución; todo ello con la finalidad de colaborar en la optimización de los recursos de una empresa, en función de sus procesos de fabricación y comercialización (entendiéndose por empresa cualquier asociación confines productivos). Se trata de proyectar productos o sistemas de productos que tengan una interacción directa con el usuario (pudiendo ser bienes de consumo, de capital, o de uso público); que se brinden como servicio; que se encuentren estandarizados, normalizados y seriados en su producción, y que traten de ser innovadores o creativos dentro del terreno tecnológico (en cuanto a funcionamiento, técnica de realización y manejo de recursos), con la pretensión de incrementar su valor de uso. Estos productos y sistemas de productos deben ser concebidos a través

de un proceso metodológico interdisciplinario y un modo de producción de acuerdo con la complejidad estructural y funcional que los distingue y los convierte en unidades coherentes.

En resumen, sus características destacadas son:

- Es un proceso de innovación cuyo objetivo es introducir el máximo valor añadido en el producto, con el fin de asegurar su rentabilidad económica, social y ambiental.
- Básicamente consiste en determinar adecuadamente las propiedades que definen el producto innovado.
- Se refiere exclusivamente a objetos industriales, es decir, seriales, así como al entorno de intangibles relacionados con el producto, si los hubiere, como son la marca, la publicidad, etc.

Esta última característica donde se hace mención a los intangibles asociados al producto incide de manera notable en la diferencia de concepto entre Diseño Industrial y Desarrollo del Producto, donde el primer término incluye el plan de comunicación e imagen en las empresas considerado el “diseño corporativo”.

El Diseño Industrial ha existido desde la producción en serie y su importancia ha ido en aumento al ritmo de la exigencia de los consumidores, el desarrollo de las tecnologías de fabricación y la competitividad de los mercados. Normalmente, el Diseño Industrial suele ser el responsable directo del éxito del producto en el mercado, de ahí su importancia.

Características esenciales de la actividad del Diseño Industrial

La gran mayoría de los teóricos del diseño, como es el caso de Bonsiepe, se establecen en común las siguientes características para definir la actividad del diseño industrial:

- Actividad que satisface las necesidades de la colectividad social mediante productos desarrollados (aislados o sistemas de productos) en interacción directa con los usuarios.
- Actividad innovadora en el ámbito de las disciplinas que constituyen el gran campo de la proyección ambiental.
- Actividad que trata ante todo de incrementar el valor de uso de los productos (función del producto y utilización por parte del usuario).
- Actividad que determina las propiedades formales (estéticas, estructurales y funcionales) de los productos.
- Actividad que pretende ser una instancia crítica en la estructuración del mundo de los objetos.
- Actividad que pretende ser un instrumento para el incremento de la productividad o para el fomento de nuevas industrias.
- Actividad coordinadora del desarrollo y planificación de productos.
- Actividad planteada como procedimiento para incrementar el volumen de las exportaciones.

Evolución histórica del Diseño Industrial

Las decisiones en el proceso de diseño se tomarán en base a la funcionalidad y estética de aquello que se diseña, y funcionarán gracias a ese proceso. Para ello será necesaria una metodología de trabajo basada en sistemas de información, y

fundamentada en la idea de convergencia, simultaneidad o concurrencia, obtenida del análisis del ciclo de vida del producto/servicio a diseñar.

La principal tarea de los ingenieros es aplicar su conocimiento científico y tecnológico a la solución de problemas técnicos, y posteriormente, optimizar las soluciones dentro de los requerimientos y restricciones impuestos por los materiales, aspectos relacionados con temas tecnológicos, económicos, legales, de medio ambiente y humanos.

Haciendo un recorrido histórico, fue en 1920 cuando J. Sinel utilizó la palabra *diseño* por primera vez, y aún así no fue hasta 1940 cuando Raymond Loewy dibujó el paquete de tabaco de la marca comercial Lucky Strike, donde los teóricos del Diseño Industrial toman como punto de partida para el nacimiento del Diseño Industrial.

Ahondando en la breve reseña histórica, se considera el Diseño Industrial español como uno de los primeros a nivel mundial. Tomando como referencia el diseño pionero en la ciudad de Barcelona. En 1775 se inauguró la Escuela Gratuita de Diseño en Barcelona bajo la protección de la Real junta Particular de Comercio. En 1783 se crearon las escuelas de Madrid, Zaragoza y en 1786 en Gerona.

Realmente el verdadero arranque del diseño en España comenzó tras la Guerra Civil, desde los años 70 hasta el fin de la Dictadura, el Diseño tuvo cierto desarrollo gracias a la aparición de la SEDI (Sociedad Española de Diseño Industrial) en Cataluña, Madrid y País Vasco, además, de la ADIFAD (Agrupación del Diseño Industrial del Fomento de las Artes Decorativas). En la década de los ochenta se propició el boom del diseño debido a la situación favorable de España, tanto en aspectos económicos como por incorporación a la Comunidad Económica Europea (CEE), de esta forma se daba a conocer a nivel mundial la creatividad del diseño español.

Concepto y evolución de la Metodología en el Diseño Industrial

Para tener un concepto adecuado sobre las acepciones a trabajar en la asignatura como es metodología y Diseño Industrial será necesario tener un conocimiento previo de ambos términos. El segundo de ellos ya ha quedado definido con una breve introducción, por lo que será necesario tener en consideración ahora una definición del primer término.

El método, a diferencia de “procedimiento” y “técnica”, aunque pueden tener valores semánticos apenas diferenciables, se define como *el modo de hacer con orden una cosa*, esta definición aporta una primera aproximación, siendo evidente la presencia de dos conceptos asociados como acción y ordenación, supuestamente racional, de las acciones.

De acuerdo con Newell, las características principales de un método son las que se detallan a continuación,

- Método es una forma específica de proceder.
- Método es una forma racional de proceder, de tal forma que la probabilidad de alcanzar el objetivo con éxito es mayor que si se intentara conseguir de cualquier otra manera.
- El objetivo del método es necesariamente general, es decir, el método debe ser aplicable a más de un problema de la misma clase.
- El uso del método es observable.

Con estas premisas, Roozenburg & Eeckels proponen que el método es la *estructura de relaciones – incluso temporales – conscientemente aplicada, de un proceso de acción*.

Un concepto más sencillo de entender para los neófitos en el tema, podría ser: **la enumeración descriptiva y racional de las dimensiones de un proceso de acción y de las relaciones entre ellas, que atiende a un propósito predeterminado**. Es decir, el método describe y relaciona **qué** hay que hacer, **quién** ha de hacerlo, **dónde**, **cuándo**, **cómo**, **con qué**, etc. para conseguir un objetivo concreto. Este proceso queda resumido en la siguiente tabla,

¿Qué?	Tareas-Actividades
¿Quién?	Actores
¿Cuándo?	Flujo (Secuencia de Tareas)
¿Dónde?	Ubicación
¿Con qué?	Herramientas
¿Cómo?	Técnicas

La agrupación de los conceptos previamente definidos forman un tándem que a continuación se define como

Metodología del Diseño Industrial: La enumeración descriptiva y racional de las dimensiones del proceso de diseño de producto innovadores y de las relaciones entre ellas, que atiende –al menos – a los siguientes objetivos predeterminados:

El **objetivo principal** del método de diseño industrial será “**definir correctamente** (nivel de calidad exigido a un producto industrial) **y completamente** (nivel de detalle exigido que permita establecer un proceso productivo) **un objeto innovador de manera que pueda ser fabricado en serie con unos determinados medios** (de fabricación que dispone el cliente que suponen una importante restricción y que influye notablemente en las especificaciones del producto a diseñar), **y ofrezca beneficios por su venta** (aunque importante, no es el único indicador del éxito del producto diseñado, que justifican los honorarios profesionales del diseñador)”.

Los **objetivos adyacentes** dependen de cada profesional, entre ellos se pueden enumerar e interpretar como básicos,

- La **originalidad**, independientemente del plagio, acto a descartar por el compromiso ético y legal, el diseñador industrial debe ser consciente de que la obra, tanto a nivel global como individual, debe ser original. Olvidar este objetivo puede hacer caer al diseñador en el autoplagio, por lo que el diseñador se encuentre en una pérdida de actualidad y de competitividad del producto en el mercado.
- El **compromiso ético**, se ha de tener en cuenta que la premisa fundamental para el empresario es la del beneficio industrial, frente a aquellos requerimientos del diseño tan importantes como la lucha contra el plagio, la seguridad de los objetos, tanto a nivel funcional como materiales utilizados, así como del proceso productivo, la utilidad de los productos de consumo, la protección del menor y la protección del medio ambiente.

El Diseñador Industrial debe huir de aquellas propuestas de trabajo en las que la seguridad de los objetos no sea uno de los requisitos básicos de los mismos. En la actualidad, las exigencias de los mercados marcan esta pauta, aunque todavía existen mercados incipientes o grupos de consumidores menos favorecidos, en los que la capacidad de autoregulación no se ha desarrollado y, por ello, exista mayor posibilidad de introducir productos de baja calidad o seguridad para obtener mayores beneficios.

En la misma línea tampoco se debe permitir el diseño de productos cuyo sistema productivo pudiera suponer riesgo para la integridad física de los trabajadores.

El compromiso ético más complicado de alcanzar es la de dotar al producto de un valor de uso claro, diferenciado y necesario. Uno de los casos con un alto nivel de sensibilización es la protección del menor, por lo que los aspectos legales sobre los mismos deben ser un punto de partida en las especificaciones del diseño, por lo que el diseñador deberá exigir a sus creaciones un respeto al correcto desarrollo físico, psíquico y social del menor. Un tema similar al mencionado es la protección del medio ambiente, siendo las consecuencias, de un mal uso, mucho más predecibles y palpables. Es por ello, que todos los requerimientos legales que afecten a este tema pasarán directamente a formar parte de la lista de especificaciones de diseño. Así, un diseño que pueda llevar el apelativo de ecológico, deberá tener en cuenta el impacto de la extracción de las materias primas, los residuos que genera y las emisiones de los procesos de fabricación, el consumo energético de dichos procesos y el reciclaje de los productos entre otros y actuar en consecuencia.

- Otro de los objetivos adyacentes, es la **rapidez** y la **calidad de la respuesta**. El cliente valora estas dos cualidades del servicio sin que para ello se vea afectada la calidad del producto.
- La **automatización** del proceso de diseño y del proceso productivo reducen los tiempos de ejecución y aportan seguridad en el proceso, conocimiento y estado de ejecución, permitiendo una mayor planificación de las tareas, así como de la resolución de problemas que se puedan acontecer.
- Y por último, el **propio beneficio** donde el diseñador profesional debe obtener, consolidar y expandir los beneficios propios de su actividad, de lo contrario se trataría de un diseñador industrial aficionado. Este beneficio particular hace que se incremente el beneficio empresarial.

La necesidad de utilizar un método o no, siempre ha estado cuestionado por los distintos investigadores y autores, debido a que el uso de método puede coartar la creatividad de la investigación. Atendiendo a esta afirmación queda patente que incluso el método más rudimentario asegura un resultado, sin embargo trabajar sin método deja abierta la posibilidad de un doble fracaso, el no obtener un resultado y el haber perdido el tiempo dedicado a una búsqueda infructuosa.

Es por ello, que el seguimiento de un método permita, desde un punto de vista económico,

- Aportar unos resultados tangibles
- Cumplir con los plazos establecidos
- Optimizar los recursos

Según el autor, Óscar Olea en su obra “Metodología para el Diseño”, expone que “...la actividad de diseñar se convierte en una serie de acciones sucesivas que permiten llegar a resultados en el trabajo con mayor rapidez y seguridad, con economía de esfuerzo y, sobre todo, con un alto grado de responsabilidad de respuesta...”

Se puede determinar que los ingenieros y particularmente los de Diseño Industrial son profesionales, que se diferencian de otros especialistas en su obligado compromiso con la materialización de las ideas, aunque sean en modo conceptual. Es decir, la ingeniería no es únicamente concepción, sino gestación, alumbramiento y seguimiento.

En general, se puede admitir la necesidad de seguir un método, aunque un exceso de detalle en la aplicación del mismo puede alcanzar unos resultados no esperados. Por lo que, método y creatividad no deben ser elementos excluyentes sino coetáneos y complementarios con el fin de obtener el mayor beneficio.

Propuesta y clasificación de las metodologías de Diseño Industrial

Antes de analizar las diversas propuestas metodológicas dadas por diferentes autores, será necesario aclarar que estas deben cumplir los siguientes aspectos,

- Estar dirigida hacia la resolución de cualquier problema, sin importar su campo de especialidad
- Promover la inventiva y la comprensibilidad
- Aumentar la cantidad de resultados
- Facilitar la búsqueda de soluciones óptimas
- Ser compatible con los conceptos, métodos y hallazgos de otras disciplinas
- No estar basada en hallar soluciones por azar
- Facilitar la aplicación de soluciones conocidas a tareas determinadas
- Compatible con herramientas informáticas
- Fácil de enseñar y aprender
- Reflejar los logros de la psicología cognitiva y la ergonomía moderna reduciendo la carga de trabajo, ahorrando tiempo, previendo el error humano y ayudando a mantener el interés activo

Esta metodología cuyas características son dividir la tarea en etapas o fases y desarrollar algoritmos que permitan alcanzar una solución de forma iterativa, permite,

- *Asegurar la calidad*, el proceso de desarrollo especifica las fases y los controles por los que tendrá que pasar el desarrollo del proyecto.
- *Coordinación*, este proceso actúa como plan maestro que define el papel de cada uno de los miembros del equipo.
- *Planificación*, este proceso contiene hitos naturales correspondientes con la finalización de cada fase.
- *Dirección*, comparando los sucesos reales con los planificados, el director del proyecto puede identificar las posibles áreas problemáticas.
- *Mejora*, una cuidadosa documentación de todo el proceso puede ayudar a identificar oportunidades de mejora.

A continuación se muestran un conjunto de metodologías propuestas por diversos autores, donde se puede observar que existen dos tendencias por el grado de análisis del problema: un grupo compuesto por teorías simplistas y otro que analiza, con cierto nivel de detalle, el problema desde su origen hasta la consumación de la venta del producto. Entre los autores relacionados se encuentran Gómez Senent 1998, Pugh

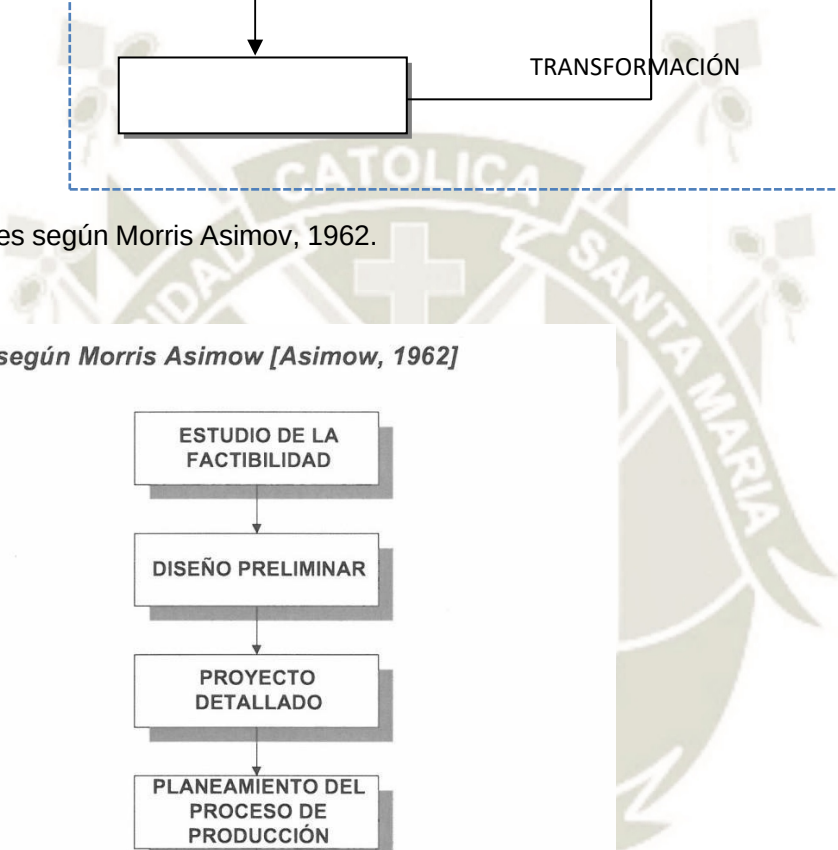
Dentro de las actividades del diseño, y en un modo realista, se deberán incluir aquellas acciones que conducen a la captación de clientes y de encargos, debido a que estas pertenecen a una fase previa e investigativa de un contrato de servicios con el fin de desarrollar un Diseño Industrial.

Teoría de las Dimensiones del Proyecto (Eliseo Gómez-Senent)		Total Design (Stuart Pugh)		Engineering Design (Pahl y Beitz)	
		Análisis	Investigación de Mercado	Planificación del producto y Clarificación de la tarea	
		Síntesis	Especificaciones		
ProcesoProyectual	Evaluación de soluciones	Diseño Conceptual		Diseño conceptual	
		Diseño de detalle		Diseño de conjunto (Embodiment Design)	
Fases	Creativas	Etapas	Investigación de Mercado	Principios de Trabajo (Working principles)	
				Especificaciones	Combinación de principios Estructuras de Trabajo (Working structures)
				Diseño Conceptual	
				Diseño de Detalle	Principio de Solución (Solution principle)
				Fabricación (DFX)	Diseño para ... (DFX)
	Transformación		Venta		
	Explotación				
Factores		Documentos de las EDP		Lista de requerimientos	
Metaproyecto		Gestión del diseño			
Técnicas específicas		Técnicas independientes Técnicas dependientes		Métodos generales, VDI.	
Instrumentos operativos		CAD y similares			

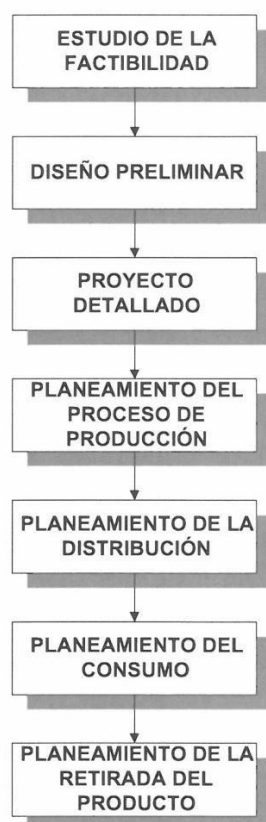
Sistemas Técnicos (Hubka y Eder)	Planteamiento Axiomático (SUH)
Obtención y clarificación de información básica	Detección de necesidades
Especificaciones de diseño	Planteamiento de los FR's
Estructura de función	Búsqueda de los Parámetros de Diseño(DP) Establecimiento de la matriz de diseño
Estructura de órgano	
Estructura de componente	
Fabricación y prueba	
Distribución	
	Requerimientos funcionales (FR)
Técnicas de gestión y organización	
Métodos de diseño, metodologías	

A continuación se presentan las actividades organizadas cronológicamente según diversos autores.

Actividades según J.C. Jones, 1992.

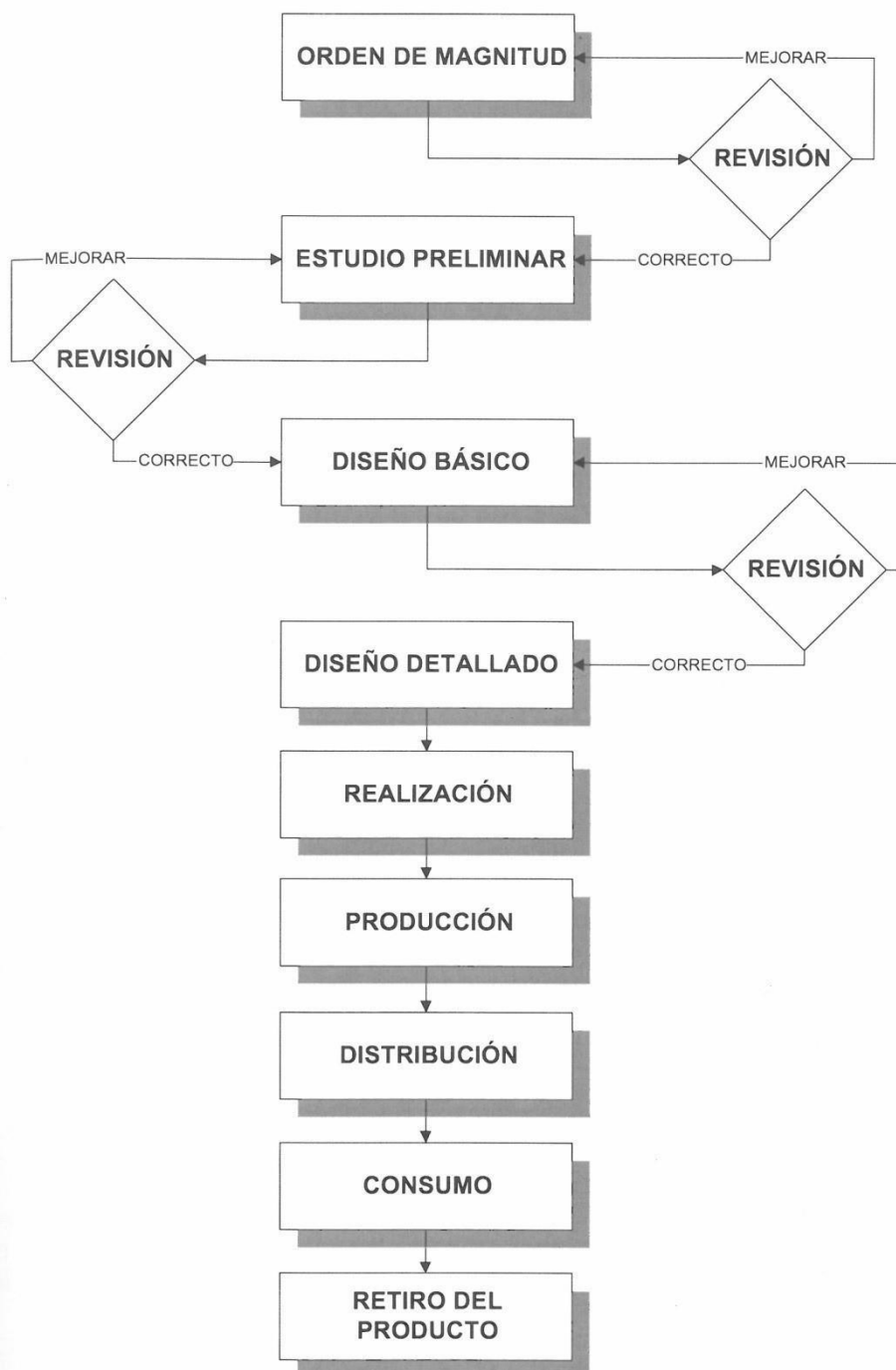


Actividades según Morris Asimow [Asimow, 1962]



Actividades según Eliseo Gómez-Senent, 1997.

Actividades según Eliseo Gómez-Senent [GómezSenent, 1997]

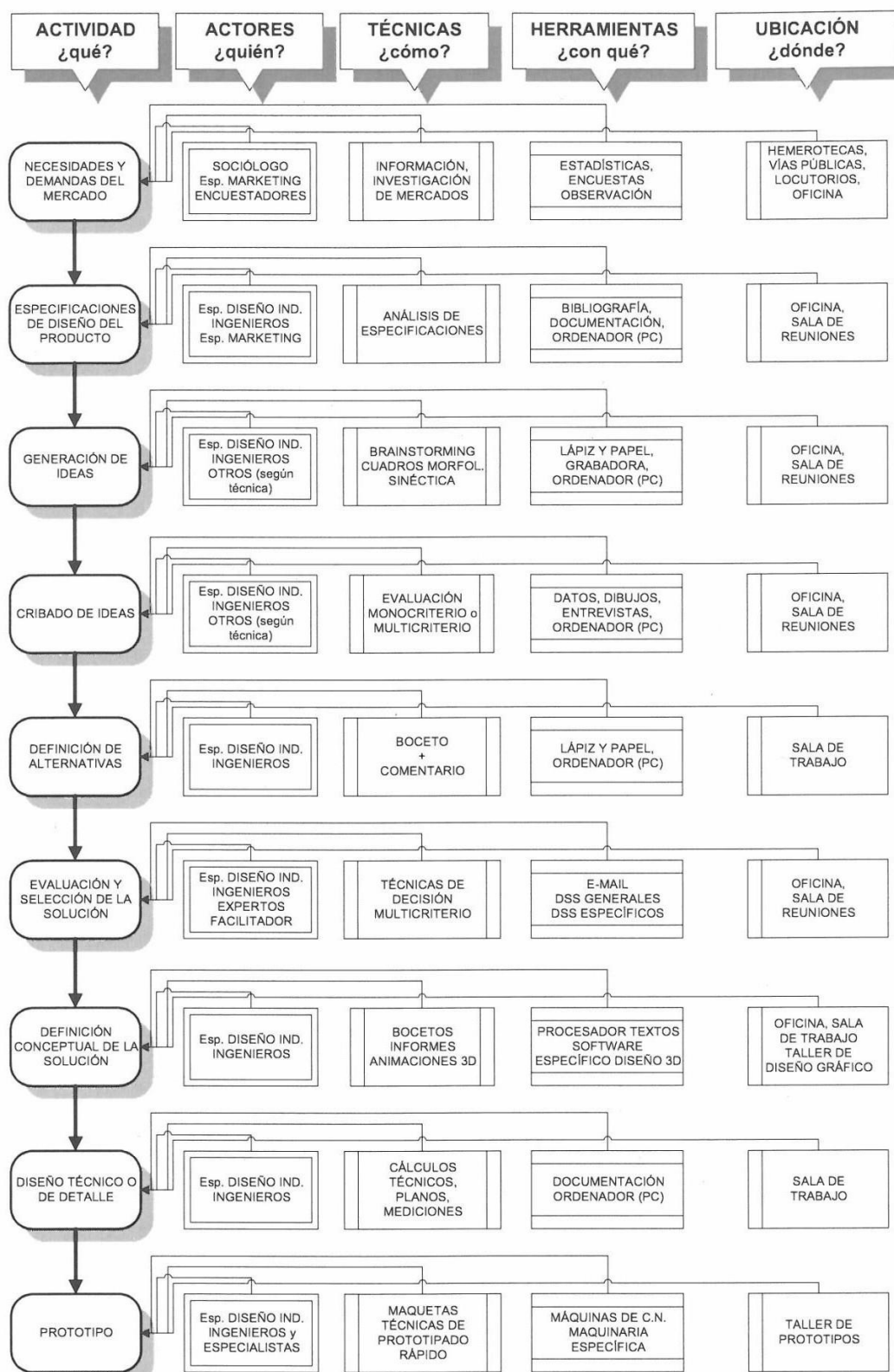


La propuesta desarrollada para la asignatura está basada en el trabajo planteado por Stuart Pugh., esquema básico de diseño de obligada referencia, dividiendo las fases según se recogen a continuación,

Fase o etapa		Descripción
1	Investigación de las necesidades y demandas del mercado	Recopilación de la información de los mercados y análisis de la misma, con el objeto de obtener una visión rigurosa de las necesidades o demandas existentes.
2	Especificaciones de diseño del producto	Decisión sobre las características mínimas del producto. El número de estas dependerá de la novedad del producto, aunque en muchas ocasiones será complicado determinarlas sobre un producto inexistente que pretende cubrir una necesidad concreta.
3	Generación de ideas	Actividad creativa del proceso de diseño y no debe verse afectada por prejuicios o restricciones innecesarias. El resultado de la misma es un conjunto de soluciones con un grado de definición meramente descriptivo.
3	Cribado de ideas	La riqueza y originalidad de un diseño dependen en gran medida de la fertilidad de la generación de ideas aunque no todas estas ideas serán viables. Por ello, esta actividad devolverá una lista de alternativas viables desde el punto de vista de las especificaciones de diseño.
3	Definición de alternativas	Se incrementa el nivel de definición de las alternativas para su posterior evaluación. De este modo se realizará la descripción de las funciones y materiales de los subsistemas, varias perspectivas del conjunto y la previsión de los parámetros (coste, fiabilidad, reciclabilidad, etc.) necesarios en la evaluación de las alternativas.
4	Evaluación y selección de la solución	Tiene como resultado la elección de la alternativa sobre la que se desarrollará el producto deseado. Los criterios de evaluación estarán íntimamente relacionados con las especificaciones de diseño descritas anteriormente.
5	Definición conceptual de la solución	Una determinada la solución final, se procederá a aumentar el nivel de definición, proporcionando modelos gráficos tridimensionales con animación, un estudio cromático, definición concreta de las entradas y salidas de cada subsistema, y cuantas descripciones sean necesarias para cumplir con las especificaciones.
6	Diseño técnico o de detalle	En esta fase se llevarán a cabo todos los estudios, cálculos, selecciones y definiciones sean necesarios para confeccionar los planos de fabricación y montaje del producto, además de redactar una memoria técnica del mismo. Esta etapa es la que confiere un carácter técnico al diseño industrial, incluyendo la selección de materiales y acabados, la resolución de los subsistemas y la decisión del proceso de fabricación.
7	Prototipado	Se construirá un modelo físico del producto y se le someterá a los ensayos correspondientes para determinar su bondad respecto de las especificaciones. Una vez superadas con éxito dichas pruebas se procederá a la fabricación masiva.

Las actividades del Diseño Industrial también incluyen las fases de fabricación, expedición, venta y retirada del producto. En esta visión global se puede observar los beneficios considerados en campos como satisfacción del usuario y el medio ambiente.

En la siguiente imagen se recoge un cuadro resumen de la metodología propuesta con las diferentes etapas, así como los actores que participan, las técnicas utilizadas, las herramientas necesarias y la ubicación de su ejecución.



Modelo organizativo

Una vez descritas las funciones será necesario establecer el modelo de organización empresarial propio de los recursos humanos que forme parte del gabinete de diseño. A continuación se analizan las ventajas e inconvenientes de tres modelos concretos.

Modelo jerárquico

Las funciones, responsabilidades y cadena de mando de los distintos puestos de trabajo están claramente definidos, estables según el tamaño y alcance de la organización y no dependen del proyecto concreto a desarrollar.

Ventajas

- Correcta definición de las características de los puestos de trabajo que tiene un efecto positivo sobre el control de la ejecución de los proyectos y la motivación del personal, debido a que cada trabajador conoce sus funciones y responsabilidades y recibe órdenes de una sola persona.

Inconvenientes

- Tiene una tendencia a la excesiva burocratización de los trabajos, la rigidez frente al cambio, escasa agilidad de las comunicaciones internas y la inclinación a la uniformidad de las retribuciones por escalas, lo que supone un efecto desmotivador.

Modelo adhocrático

Este modelo empresarial se conforma "ad hoc", es decir, a propósito de cada producto. Esto conlleva a una dirección por objetivos y la formación de equipos de trabajo según el fin a conseguir. Así mismo, será necesario establecer un responsable-gestor por proyecto (Project leader o Project manager) que asuma las tareas de organización interna del equipo de trabajo.

Ventajas

- Son modelos ágiles y con una comunicación interna rápida. Es altamente motivante ya que supone conocer en todo momento los objetivos del trabajo. Facilita la integración de los recursos humanos más adecuados, por sus características, a un fin concreto y permite establecer retribuciones diferenciadas en función de los logros alcanzados por cada trabajador.

Inconvenientes

- Tiene un importante aumento de las tareas de organización. Debido a que cada cargo supone una reestructuración del equipo de trabajo además de los conflictos generados por la utilización de un mismo recurso por varios equipos.

La organización por divisiones

Se caracteriza por ser similar en los conceptos al modelo jerárquico pero especializado según áreas de actividad con un propósito común o unidades de negocio.

Ventajas

- Similares a las del modelo jerárquico con un añadido de la especialización.

Inconvenientes

- Añadir a las del modelo jerárquico las dificultades que conlleva la reasignación de destinos para aprovechar la capacidad de un recurso de forma completa y constante.

Una vez descritos y analizadas las ventajas e inconvenientes de cada modelo, se puede afirmar que el más conveniente y extendido en los gabinetes de diseño, además de flexible es el modelo ahocrático.

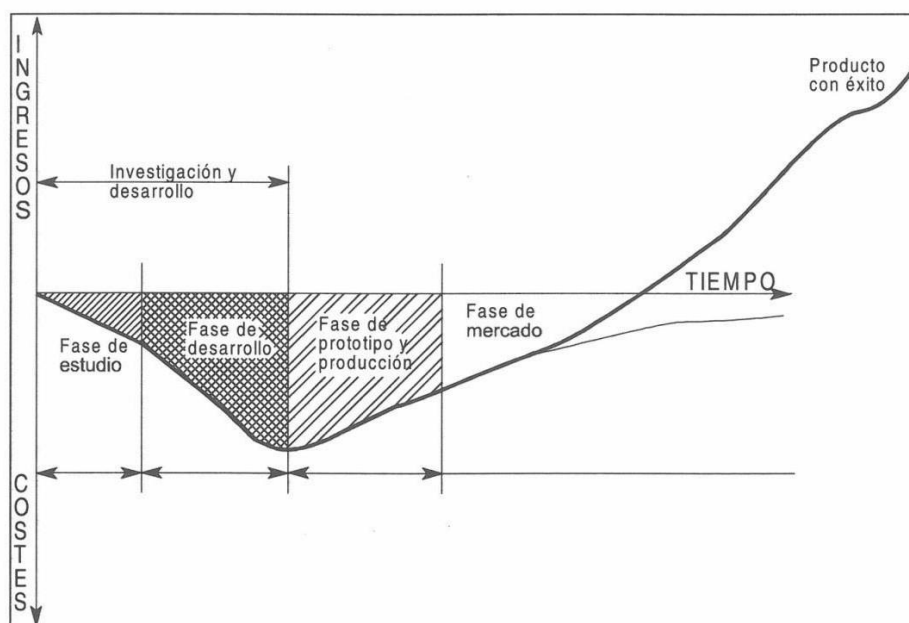
Los productos industriales. Ciclo de vida del un producto

Se considera producto industrial al grupo de atributos tangibles e intangibles que incluyen el envase o embalaje, el color, el precio, la calidad y la marca; incluyendo los servicios, la reputación del vendedor, etc., constituyendo en sí un producto sistema. Todo ello, diseñado y desarrollado en el contexto empresarial para su fabricación en pequeñas o grandes series. Un aspecto que caracteriza y distingue a los productos es su valor añadido, este proviene de la incorporación de otra serie de productos, uso de la energía y mano de obra, además de los servicios añadidos, como transporte, tiempo de entrega, servicio técnico, financiación, garantía, etc.

Ciclo de vida de un producto

Las etapas del ciclo de vida de un producto se establecen en las fases de introducción o lanzamiento, crecimiento, madurez y declive. Estos ciclos en los productos industriales son algo más largos que los productos de consumo. Se pueden diferenciar dichos productos según sean sus usuarios y para qué se utilicen.

Dentro de estas etapas, existe una en la que no se generan ingresos, sino que por el contrario el empresario deberá disponer de fuentes de financiación para poder hacer frente a esta situación hasta que la comercialización de producto haga recuperar dicha inversión en un periodo determinado de tiempo. Estas etapas comprenden las fases de estudio, desarrollo, prototipo y producción.



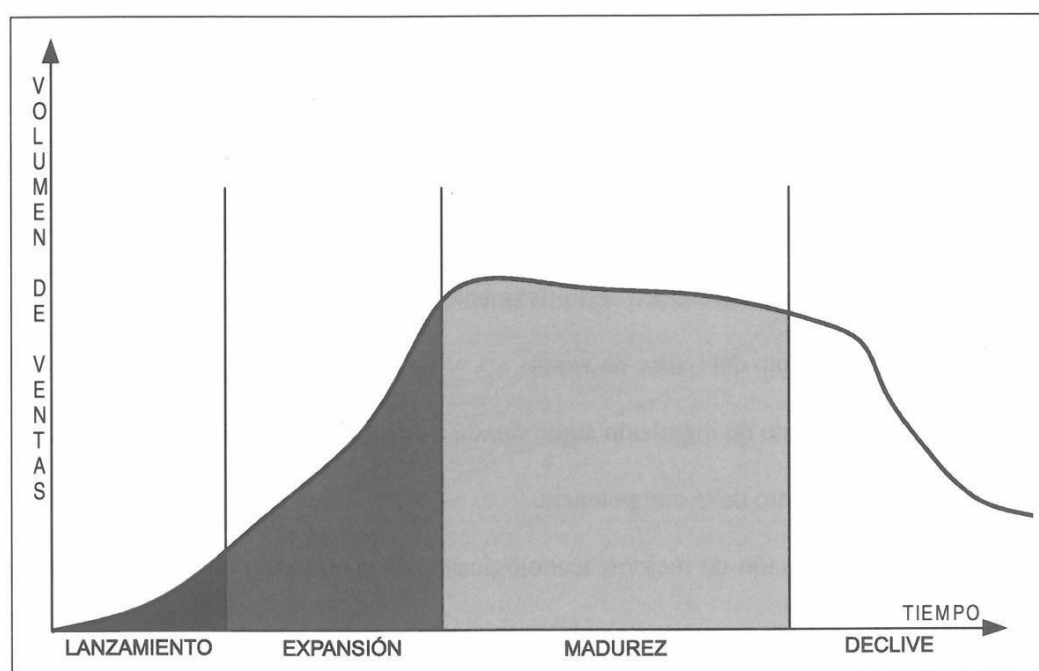
Fase de Estudio: varía en función del proyecto y de la política, estrategia y fines de la empresa. Las organizaciones por lo general se mueven por criterios de rentabilidad y procuran vigilar el tiempo y coste de realización, vinculando el concepto de producto a la consecución de resultados económicos.

Fase de Desarrollo: Se realizan las actividades de puesta a punto, prueba y control de necesidades para llevar a la práctica el nuevo producto.

Estas dos fases constituyen acorde con la metodología propuesta con el proceso de Investigación y Desarrollo, es decir, fases de la 1 a la 6, véase figura XX.

Fase de Producción: Esta última fase incluye la construcción de un prototipo, la preparación del sistema de fabricación y el inicio de la producción, fabricación de preseries.

El Proceso de Diseño y Desarrollo del Producto finaliza cuando se introduce el producto en el mercado, iniciándose una nueva etapa que genera ingresos para la empresa y está constituida por las siguientes fases,



Fase de introducción o lanzamiento del producto: el producto es ofrecido al mercado en general o particular según los segmentos analizados en la fase anterior. Si el producto responde las expectativas de sus creadores, es aceptado por un grupo de compradores proclives a las novedades y con alta capacidad de gasto, aunque con volúmenes de venta aún bajos, se genera un incremento paulatino de la demanda. En esta fase el producto comienza a ser conocido. Los ingresos por ventas aún no cubren la totalidad de los costes, dado que la actividad todavía se encuentra en la fase de pérdidas.

Características de la fase:

- Producción a gran escala bajo un programa exhaustivo de mercadotecnia.
- Normalmente el incremento en las ventas se produce de forma lenta.
- Las operaciones del periodo introductorio se caracterizan por costes elevados,

pérdidas netas y distribución limitada.

- El factor precio índice excesivamente en los productos nuevos, ya que suelen ser más elevados que cuando ya están introducidos en el mercado, debido a varios factores:
 - o Elevados costes por producciones reducidas.
 - o Problemas tecnológicos de producción y grandes gastos de promoción.
 - o Existencia de pocos competidores.
 - o Introducción a clientes de renta elevada.
 - o Reducida distribución.
 - o Exceso de inversión en técnicas impulsivas.

Fase de crecimiento: a medida que el producto es aceptado en el mercado, se estimula la demanda y se registran incrementos muy fuertes en las ventas.

Características de la fase:

- El producto se conoce definitivamente en el mercado.
- Incremento del índice de ventas.
- El capítulo de impulsión sigue siendo elevado.
- Incremento de la competencia.
- Introducción de mejoras tecnológicas y de producción en los productos y servicios.
- Penetración en distintos niveles del mercado.
- Potenciación de la demanda selectiva.
- Captación de distribuidores idóneos en función del mercado.

Fase de madurez: El producto se consolida en el mercado.

Características.

- El producto llega a una cota de saturación en el mercado.
- El ritmo de ventas continúa ascendiendo pero a un ritmo menor, llegando a cota en la que las ventas se estabilizan y los beneficios empiezan a descender.
- La competencia al producto es agresiva.
- Se incrementa el gasto en técnicas impulsivas para mantener la vigencia del producto.
- Posibilidad de ampliación de la vida del producto, si rejuvenece mediante modificaciones (rediseño o restyling), se realiza una nueva promoción y/o un cambio de los precios.

Fase de declive: El producto entra en su última fase motivado por diferentes razones como competencia, productos sustitutivos, cansancio de los consumidores, nuevas tecnologías, etc.

Características de la fase:

- Se requiere un estudio de rentabilidad del producto en sus posibles rediseños.
- Abandono de la producción e introducción del producto en el mercado, o renovación e innovación del mismo.
- Las actuaciones que se llevarán a cabo en el caso de la renovación e innovación del producto serán:
 - o Mejora del producto en forma funcional o búsqueda de actuaciones que lo revitalicen de alguna forma.
 - o Depurar la gama de productos, eliminando los tamaños y modelos menos rentables.
 - o Cerciorarse de que los programas de mercadotecnia y producción sean los más eficientes posibles.

- o Profundizar en el estudio y control de los gastos que genera el producto, para reducir costes al nivel mínimo, optimizando así la rentabilidad durante el resto del tiempo.

Esta fase también conocida como obsolescencia, los productos necesitan ser mejorados y modificados de acuerdo a las nuevas demandas del mercado, o a las posibilidades que ofrece la tecnología. Esta actividad se denomina mejora continua de los productos existentes.

Otra de las opciones para evitar el declive del producto, es la apertura hacia nuevos mercados no explorados ni explotados.

El Diseño en la Empresa

La condición para formular una estrategia competitiva consiste en relacionar la empresa con su medio ambiente. La situación de competencia en un sector industrial depende de cinco fuerzas competitivas básicas, como son: los cambios de nuevas empresas y productos, las empresas competidoras, el peso comercial de los clientes, el peso comercial de los proveedores y la competitividad del sector.

El objetivo de una empresa industrial, dentro de la filosofía del libre mercado, donde la industria debe vivir de sus propios recursos, es diseñar productos funcionales y estéticamente agradables en un plazo de lanzamiento lo más corto posible, con el mínimo coste, y con el objetivo de mejorar la calidad de vida del usuario final.

Se puede entender la empresa como un ente que responde a uno o varios objetivos que generan sistemas, objetos, servicios y productos que lleva a satisfacer una necesidad humana. La empresa entendida como un servicio al cliente, debe incorporar los métodos y metodologías del diseño que le permitan dar soluciones eficientes dentro de su entorno.

Los requisitos previos para la gestión efectiva de un diseño pasan por el conocimiento de las necesidades y del interés de los clientes potenciales, por tanto, se deben tener en cuenta los compromisos que la empresa diseñadora está dispuesta a asumir respecto a la calidad y la fiabilidad, debiendo preguntarse hasta qué punto la organización está dispuesta a llevar a cabo un buen diseño. Es importante que la dirección de la organización se implique plenamente en el diseño para reaccionar a tiempo frente a los competidores en cuanto a los recursos disponibles, innovaciones en otros campos y posibles limitaciones de la organización.

Norma VDI 2221

La VDI 2221 (Verein Deutscher Ingenieure) denominada "Enfoque sistemático para el diseño de sistemas técnicos y productos", norma desarrollada por los ingenieros profesionales, es una variación de la VDI 2222, esta analiza y entiende el problema a profundidad, es especialmente útil para la realización de productos pero principalmente para el caso de rediseños o mejoras de productos.

La estructura de este enfoque general del diseño se basa en siete etapas y se muestra en la figura 15, cada una de ellas con un resultado particular. El resultado de la primera etapa, la especificación es de particular importancia y constantemente se

revisa y se mantiene actualizada, se utiliza como una referencia en todas las etapas subsecuentes.

La segunda etapa del proceso consiste en determinar las funciones requeridas del diseño, y producir una estructura de funciones diagramáticas. En la etapa número tres se hace una búsqueda de principios de solución para todas las funciones secundarias, y éstas se combinan de acuerdo con la estructura global de funciones en la solución principal. Esta se divide, en la etapa 4, en módulos realizables y una estructura de módulos que representan la descomposición de la solución en ensambles fundamentales. En la etapa 5 se desarrollan módulos claves en un conjunto de arreglos preliminares. Éstos se refinan y desarrollan en la etapa 6 para llegar a un arreglo definitivo; y en la etapa 7 se generan los documentos del producto final.

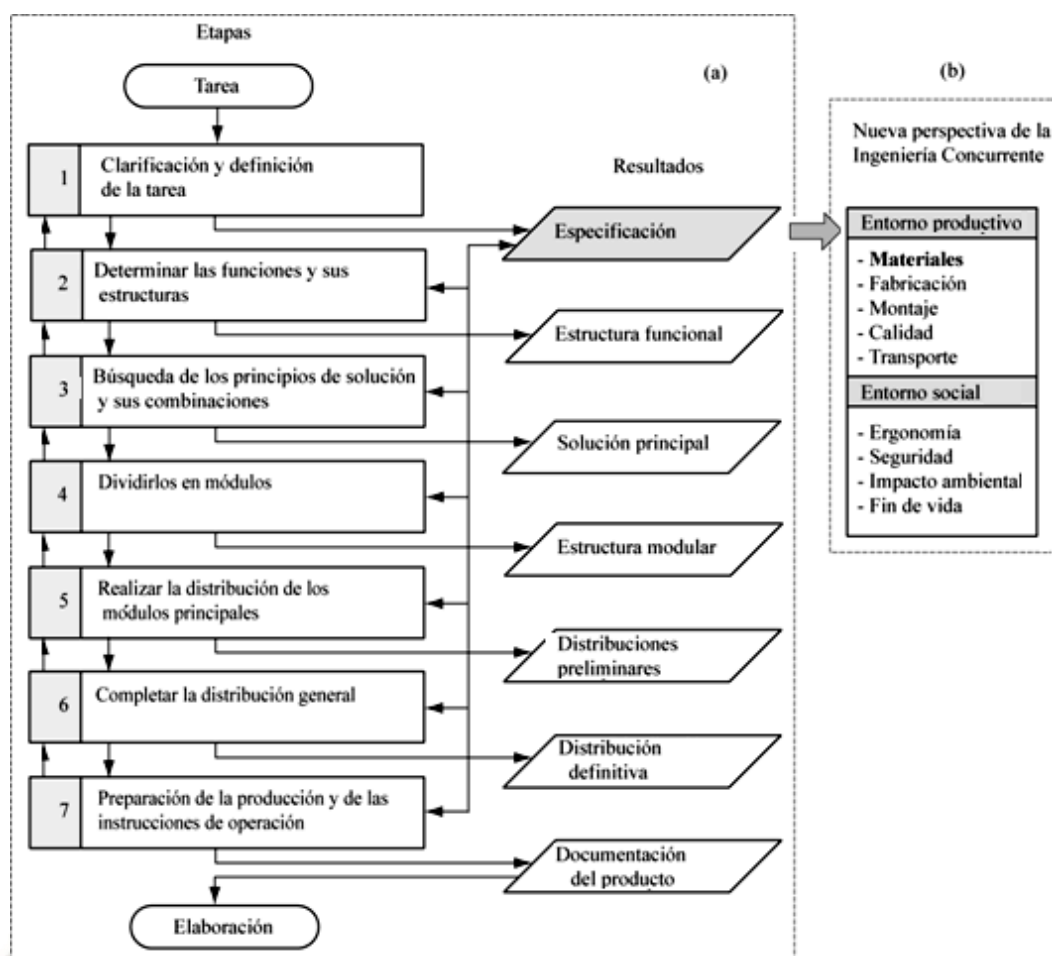
Esta guía incide en que se deben analizar y evaluar en cada etapa diversas variantes de solución, y que existe mucho más detalle en cada etapa de lo que se muestra en el diagrama. Entre las advertencias que arroja se encuentra la siguiente:

Es importante hacer notar que las etapas no necesariamente se cumplen de manera rígida una después de la otra. Es frecuente que se lleven a cabo de manera iterativa, regresando a las anteriores, y logrando de esta forma una optimización paso a paso.

La guía VDI sigue un procedimiento sistemático general en el que primero se analiza y entiende el problema lo más completamente posible, luego se descompone en problemas secundarios, se encuentran soluciones secundarias apropiadas y éstas se combinan en una solución general.

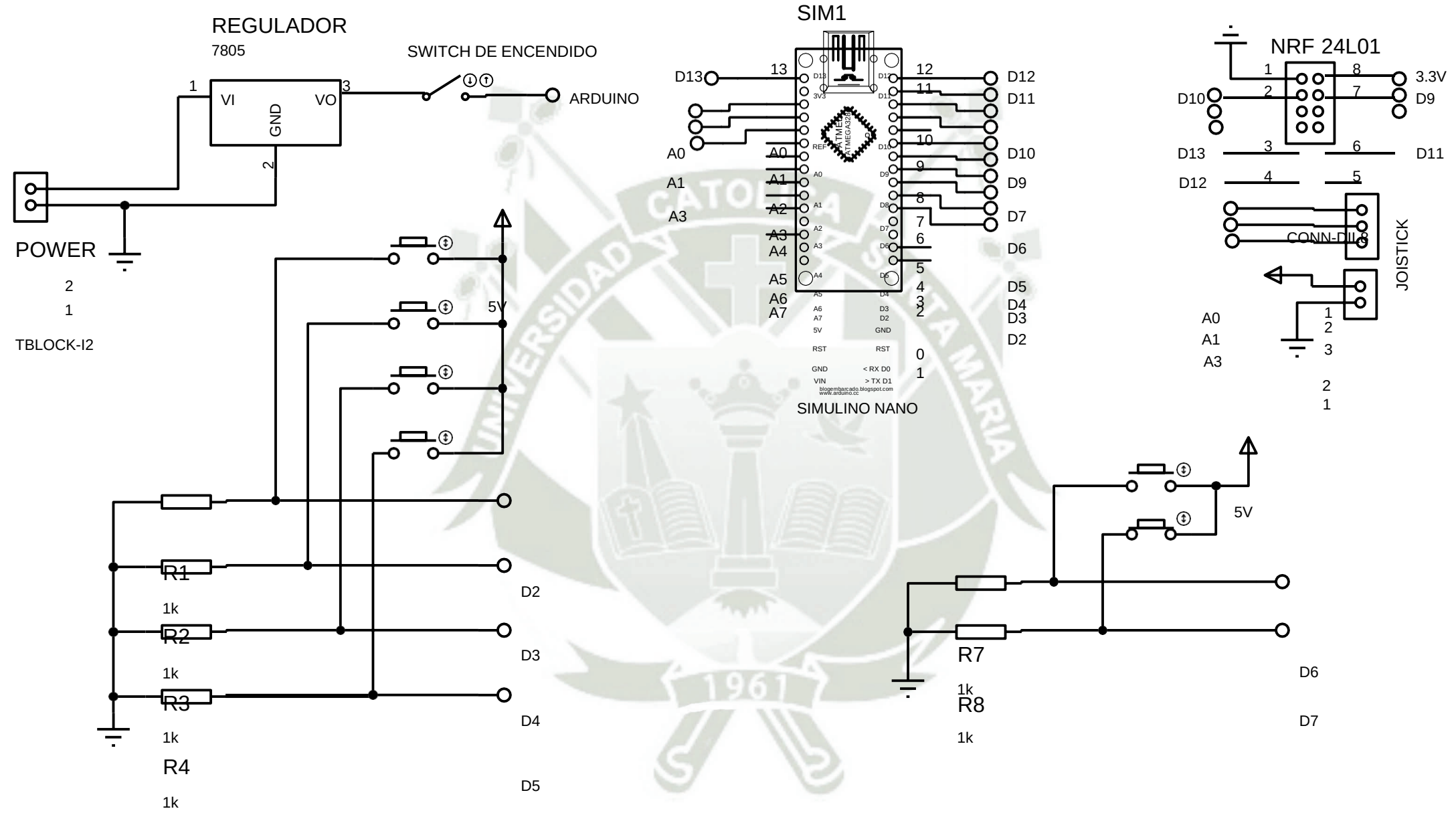
Las 7 etapas generales.

1. Especificaciones Clarificar y definir la tarea
2. Estructura de funciones
Definir funciones Definir estructuras
3. Búsqueda de soluciones
Buscar principales soluciones y combinaciones
4. Estructura de modelo
Decidir en módulos realizables
5. Arreglos
Arreglos preliminares
 - Desarrollar arreglos de módulos claves
 Arreglos definitivos
 - Completar arreglo general
6. Documentación
Preparar instrucciones de operación y producción
7. Realización
Realización del producto

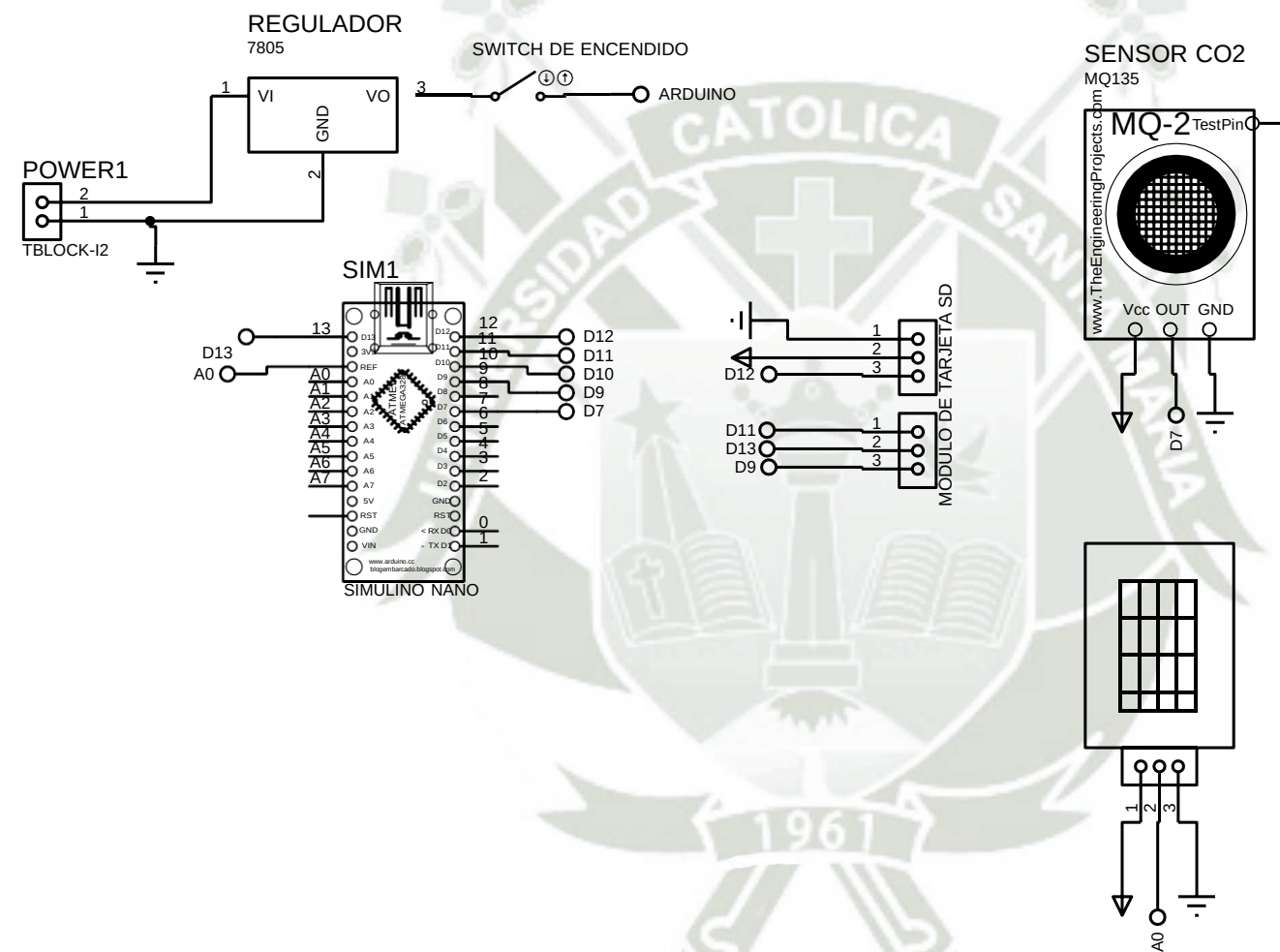




Anexo 6: PLANOS ELECTRONICOS



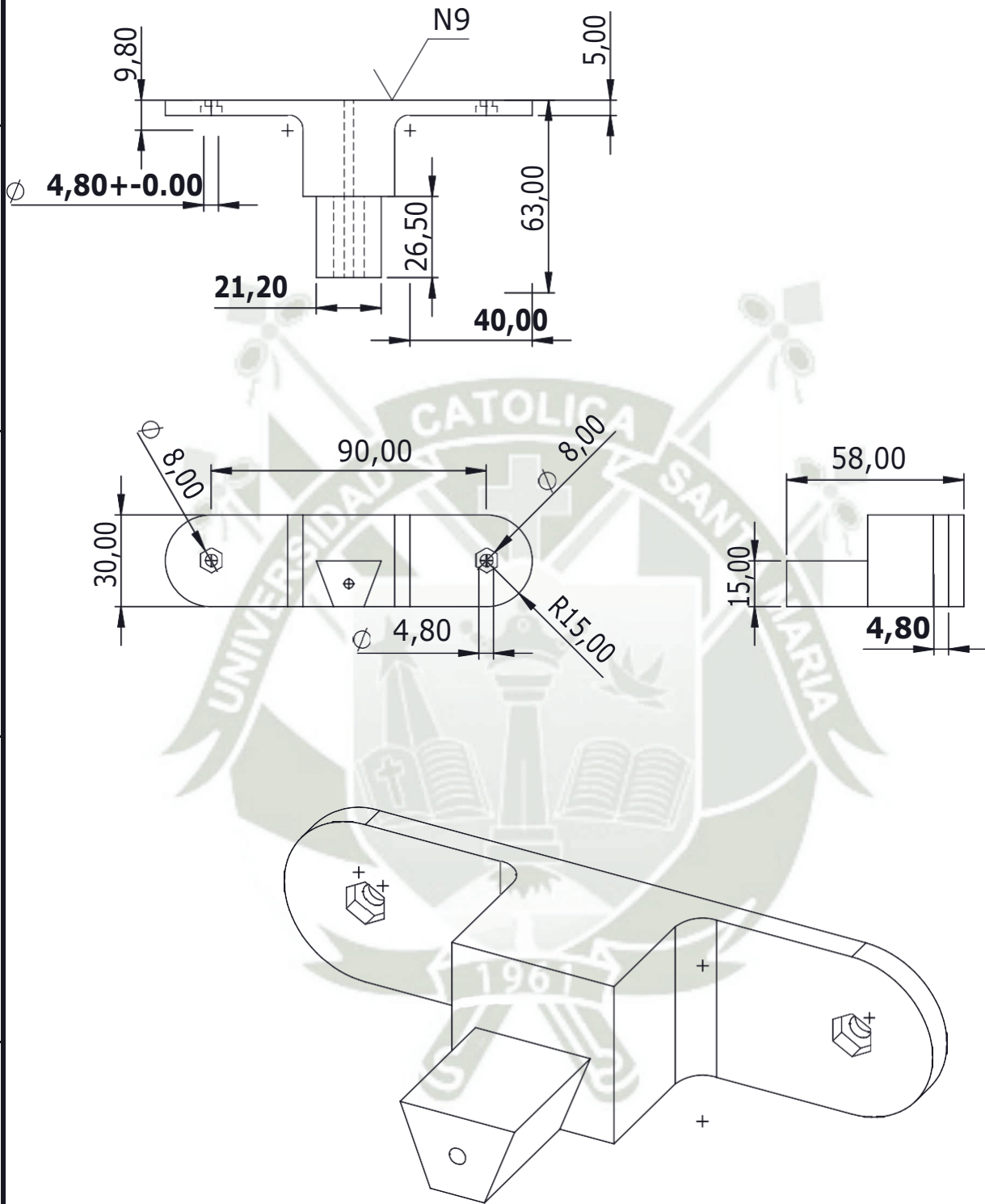
	Nombre	Fecha	Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala:	CIRCUITO ESQUEMATICO MANDO DE CONTROL		Pag: 2
1:1			Págs: -



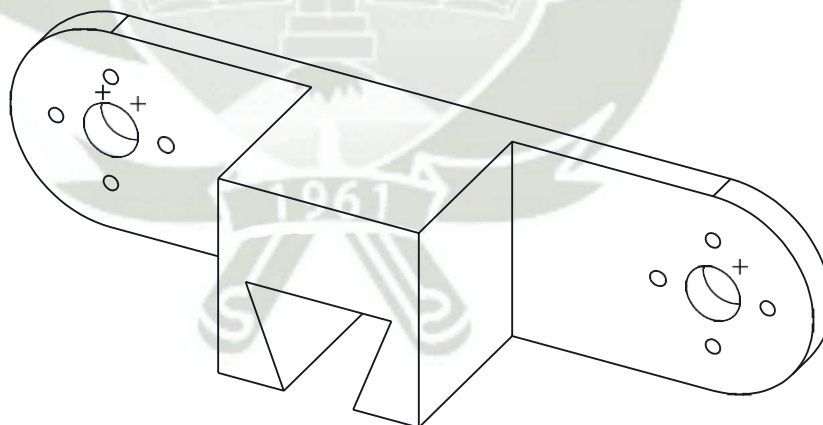
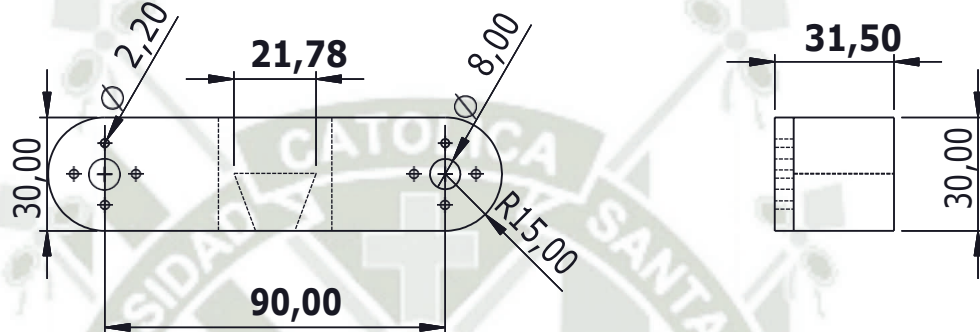
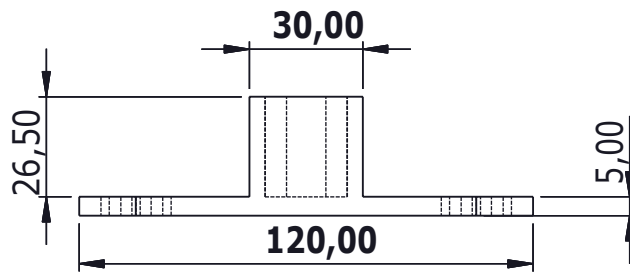
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	A
Elaboró:	Boris Portugal Quito	01/09/2021		
Escala: 1:1	CIRCUITO ESQUEMATICO SENSADO TEMPERATURA Y CO2			Pag: 3
				Págs: -



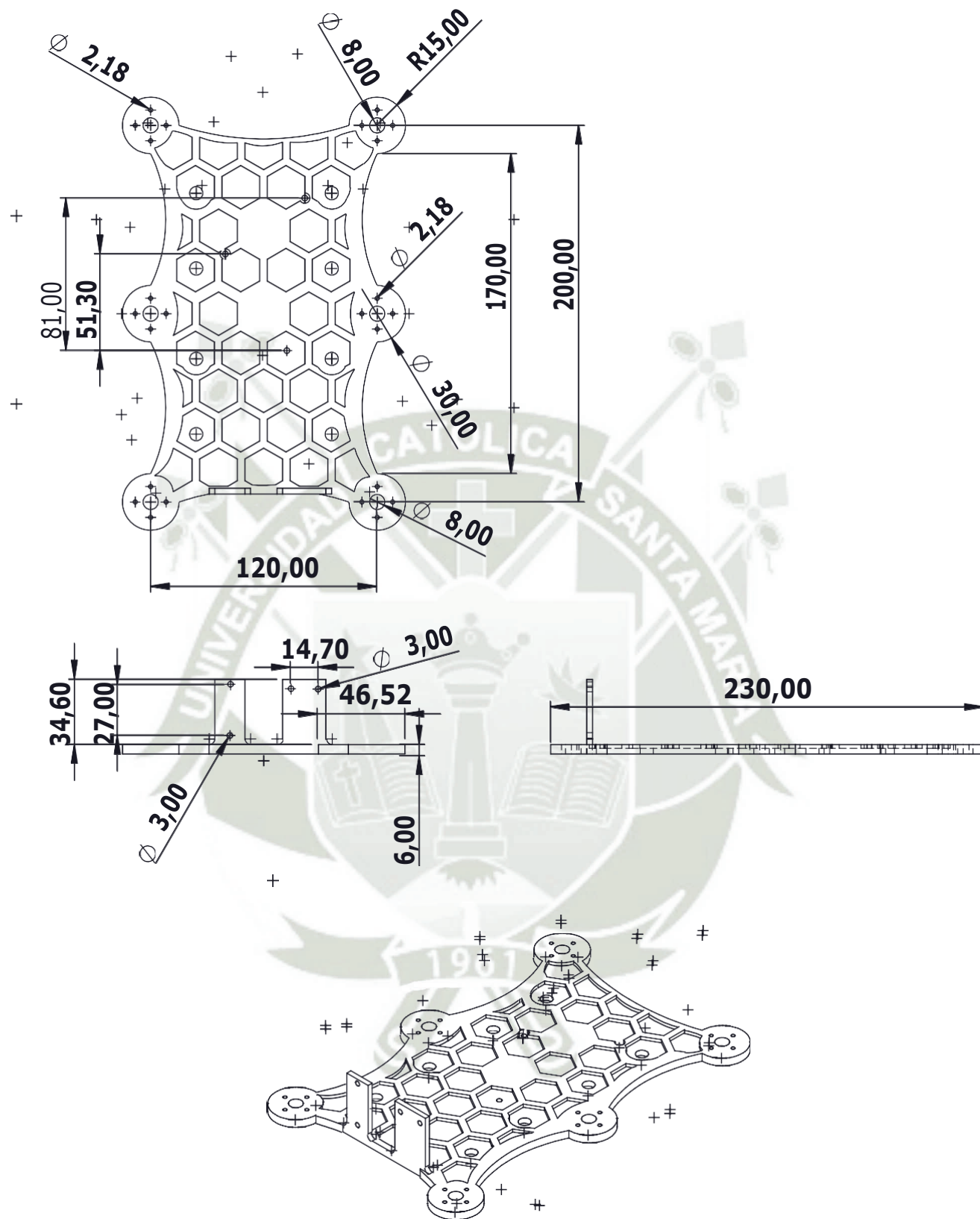
Anexo 7: PLANOS DE DISEÑO DEL ROBOT HEXÁPODO



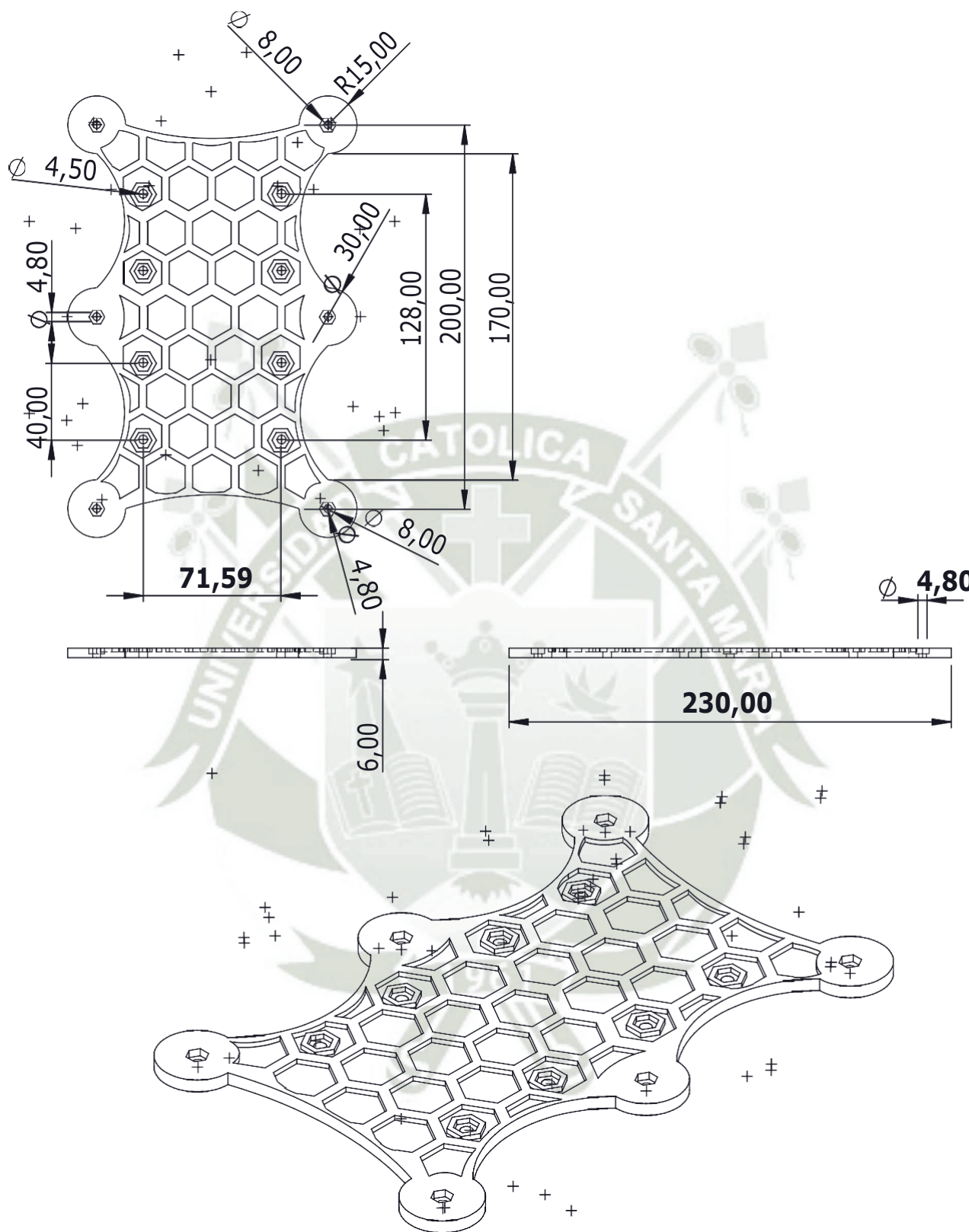
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.1 Acople c 1		Pag: 1
			Págs: 1-14



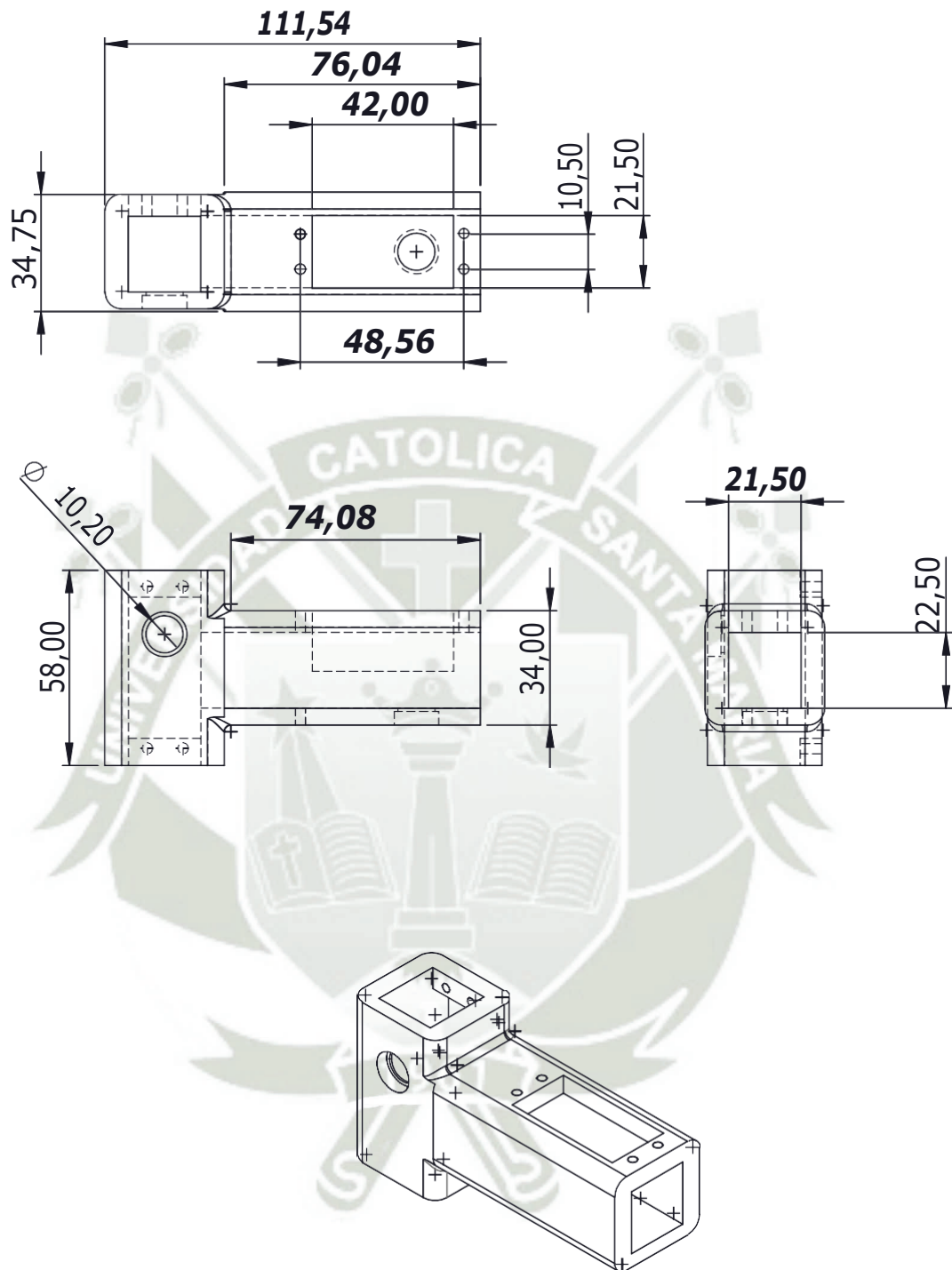
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.2 Acople C 2		Pag: 2 Págs: 1-14



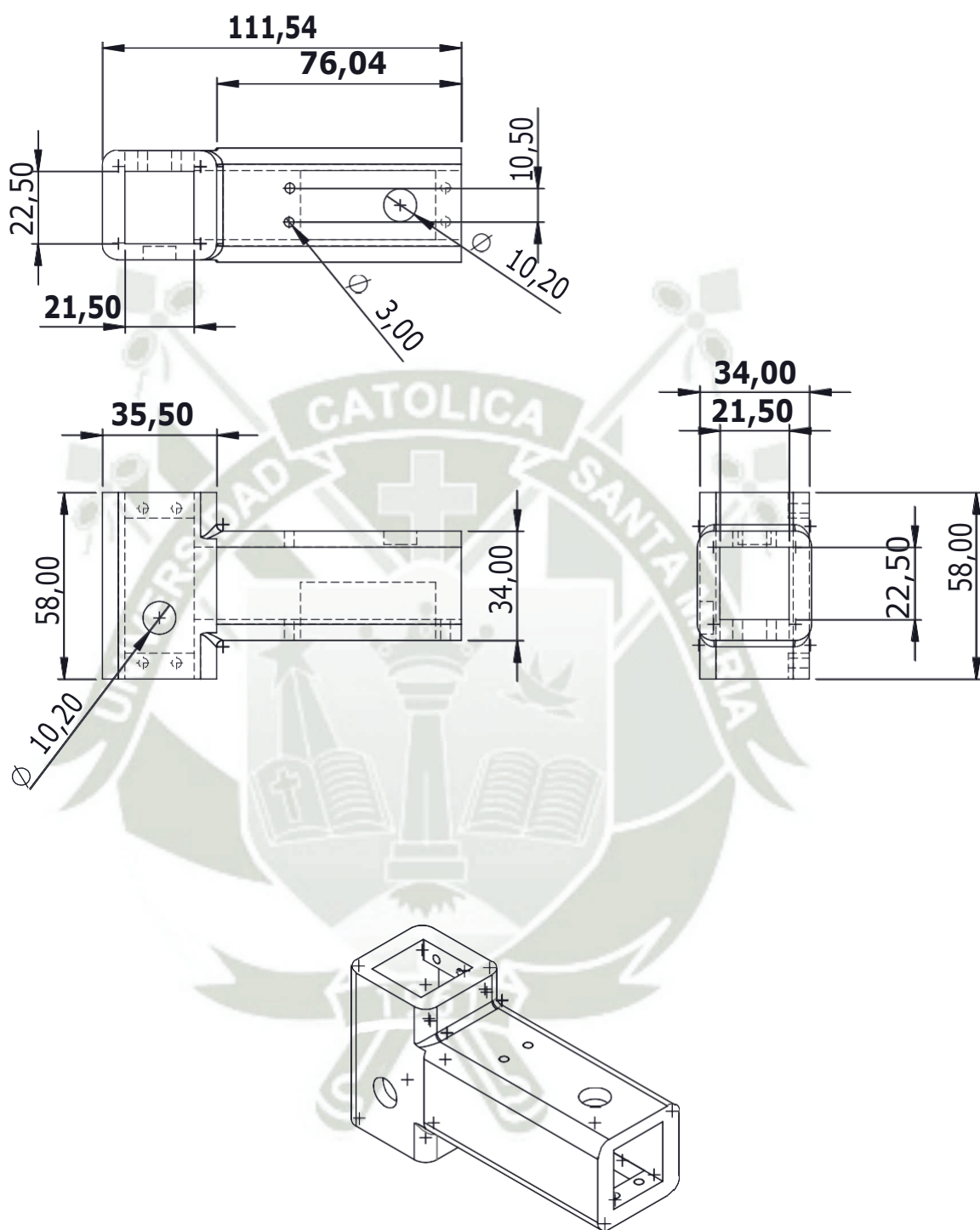
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	⊕ 1.3 Torax2		Pag: 3 Págs: 1-14



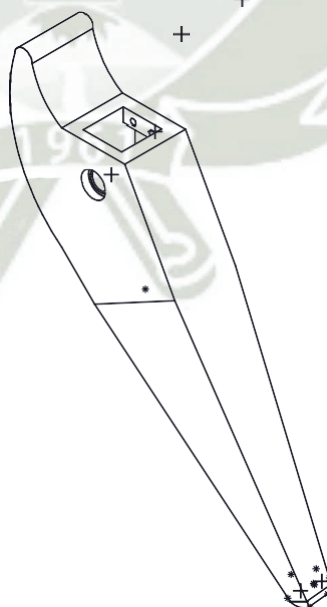
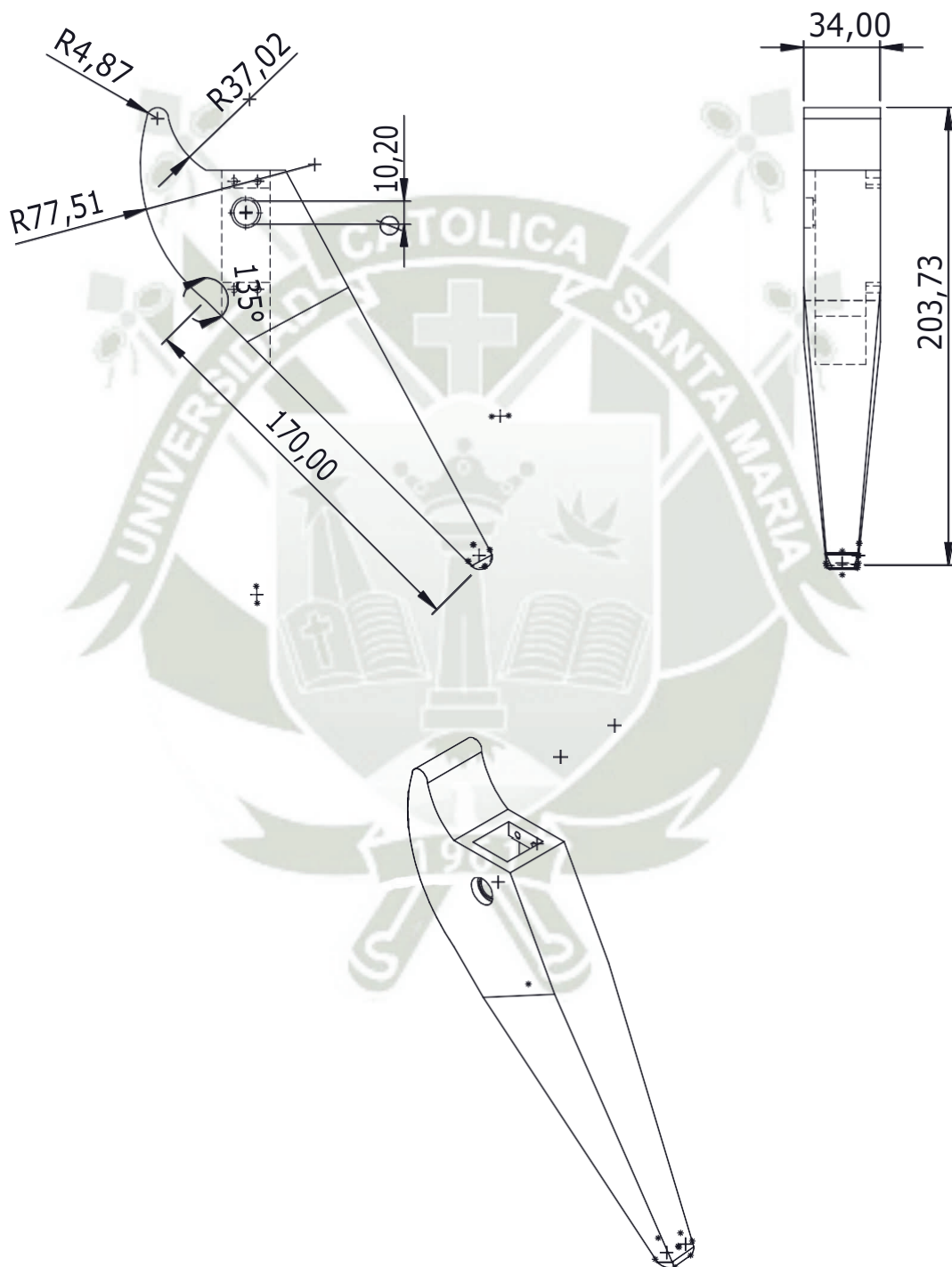
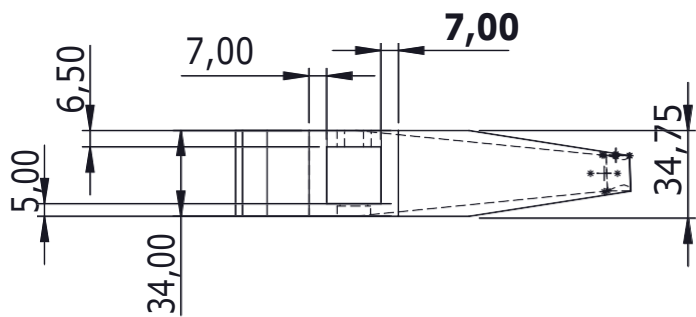
	Nombre	Fecha	Universidad Católica de Santa María  Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.4 Torax1		Pag: 4
			Págs: 1-14



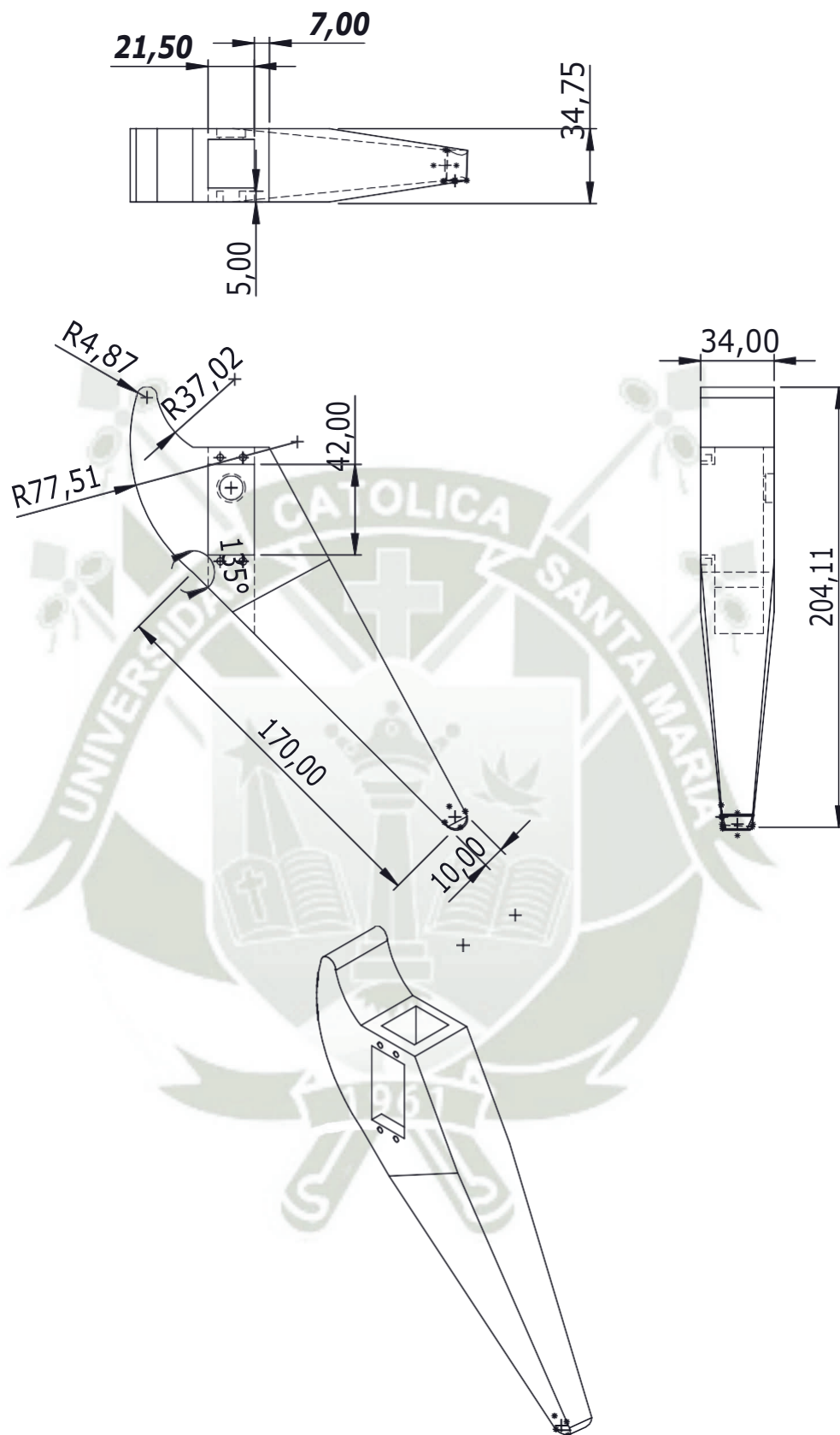
	Nombre	Fecha	Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.5 Caja de servos 1		Pag: 5 Págs: 1-14



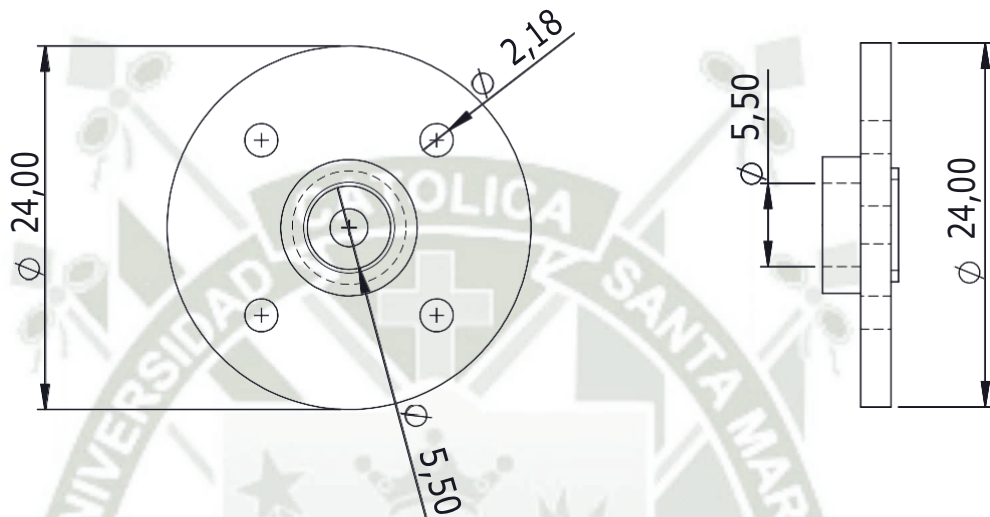
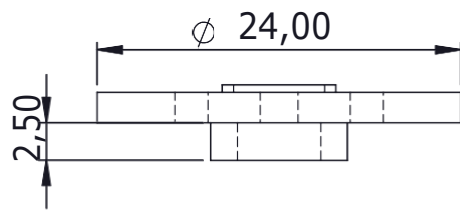
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.6 Caja de Servos 2		Pag: 6 Págs: 1-14



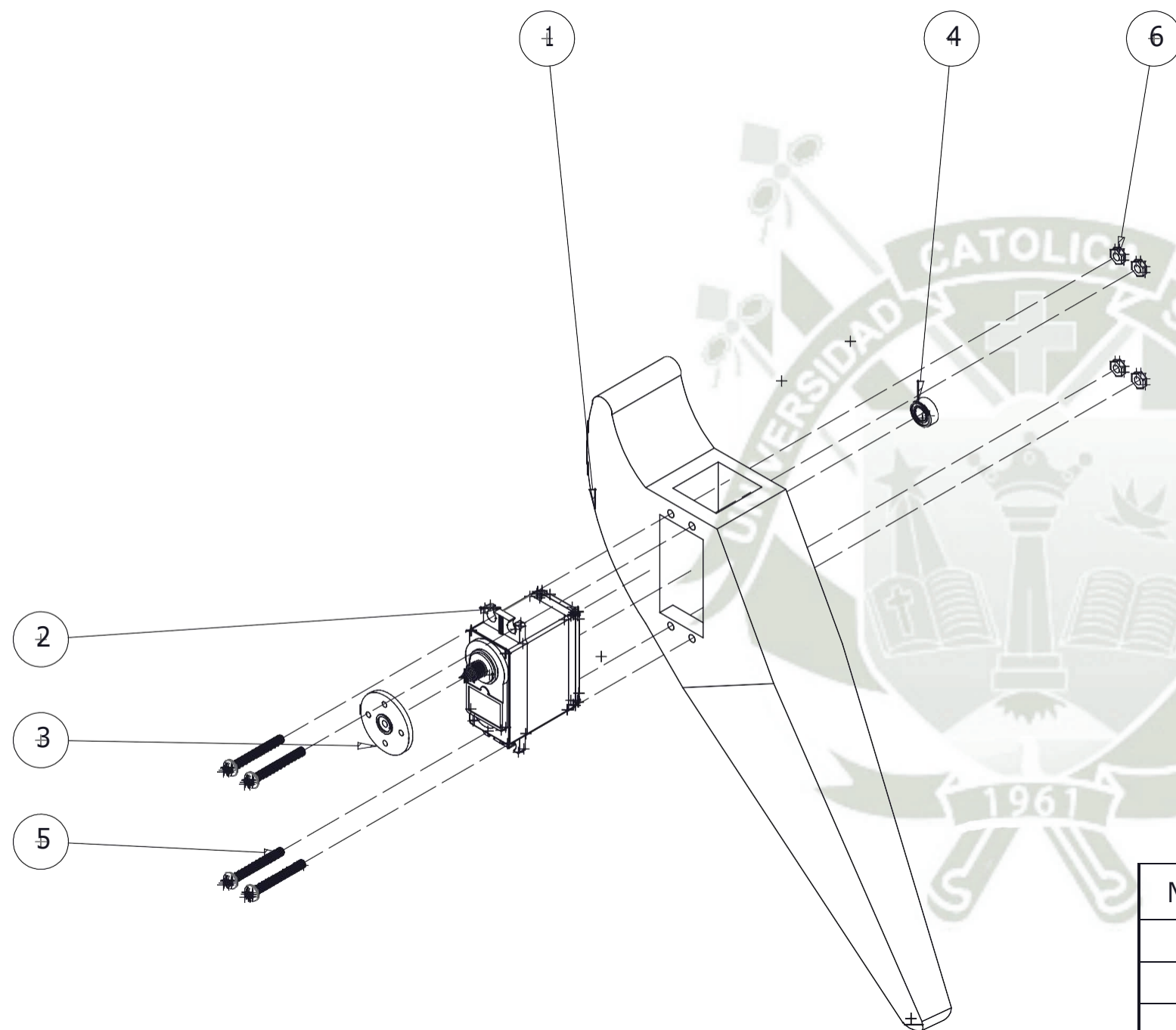
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.7 Pata 1		Pag: 7 Págs: 1-14



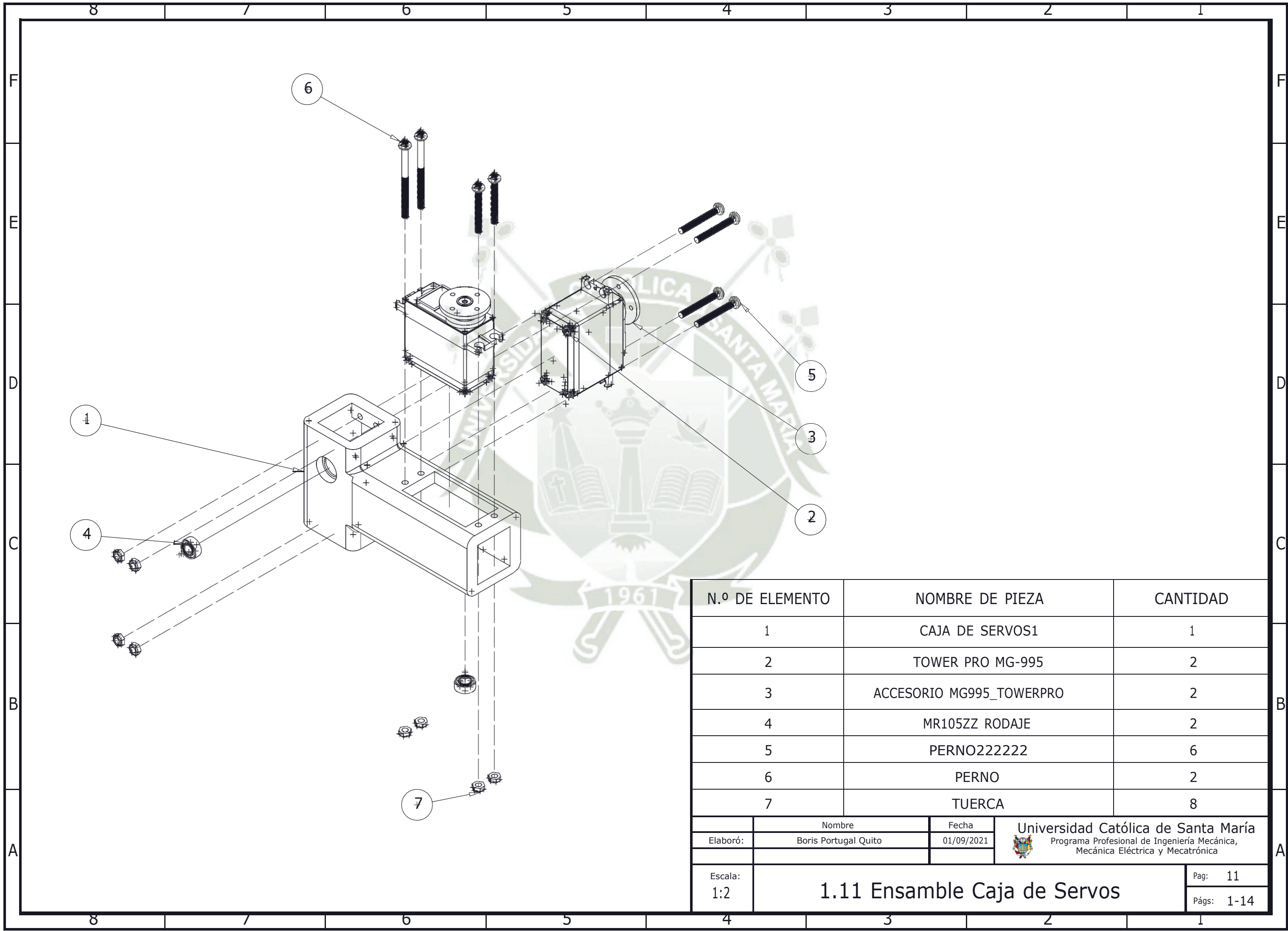
	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.8 Pata 2		Pag: 8 Págs: 1-14



	Nombre	Fecha	Universidad Católica de Santa María  Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica	
Elaboró:	Boris Portugal Quito	01/09/2021		
Escala: 1:2	1.9 ACCESORIO MG995_TowerPro			Pag: 9
				Págs: 1-14



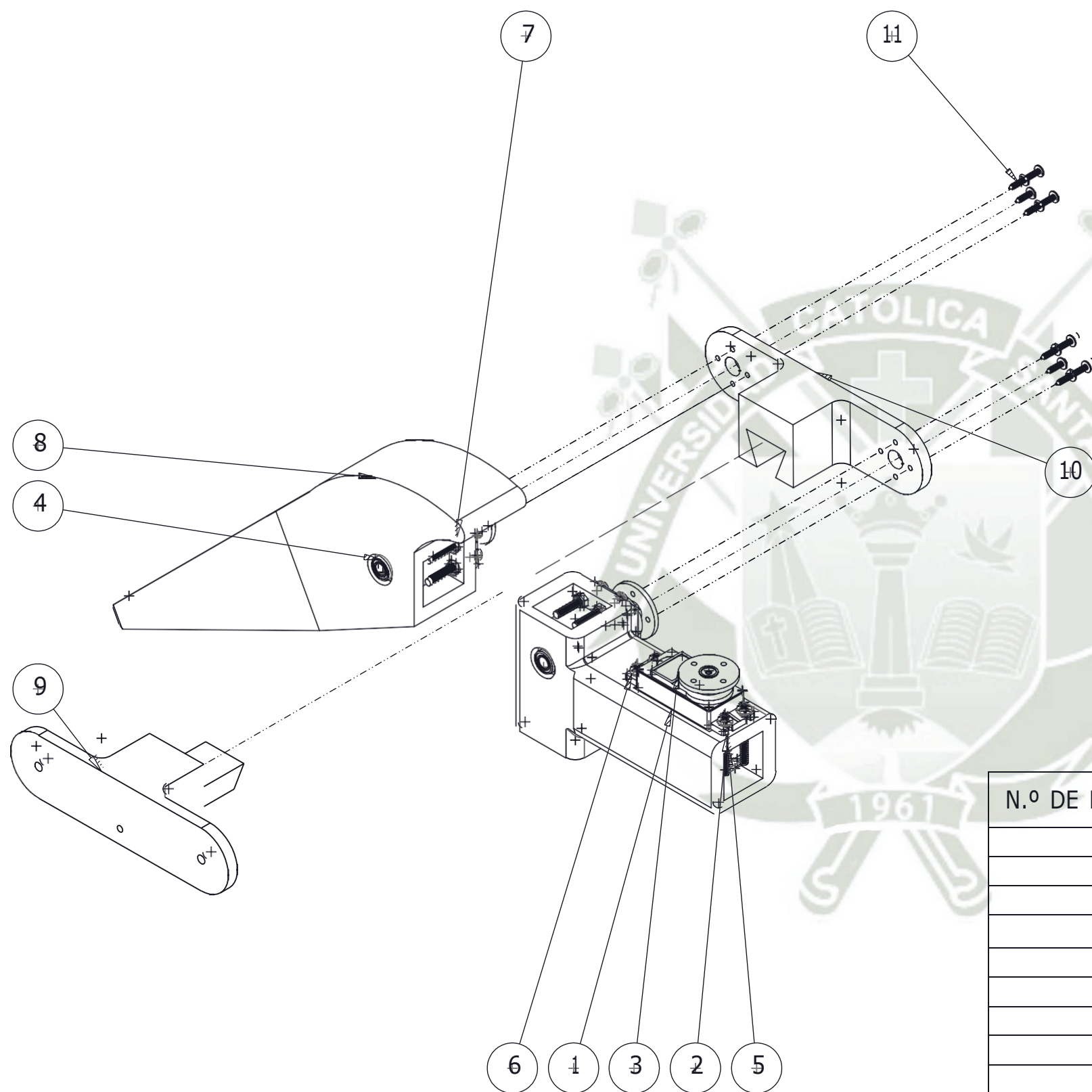
N.º DE ELEMENTO		NOMBRE DE PIEZA		CANTIDAD
1		PATA2		1
2		TOWER PRO MG-995		1
3		ACCESORIO MG995_TOWERPRO		1
4		MR105ZZ RODAJE		1
5		PERNO222222		4
6		TUERCA		4
	Nombre		Fecha	Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito		01/09/2021	
Escala: 1:2	1.10 Ensamble Pata			
	Pag: 10			
	Págs: 1-14			



N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	CAJA DE SERVOS1	1
2	TOWER PRO MG-995	2
3	ACCESORIO MG995_TOWERPRO	2
4	MR105ZZ RODAJE	2
5	PERNO222222	6
6	PERNO	2
7	TUERCA	8

Elaboró:	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
	Boris Portugal Quito	01/09/2021	

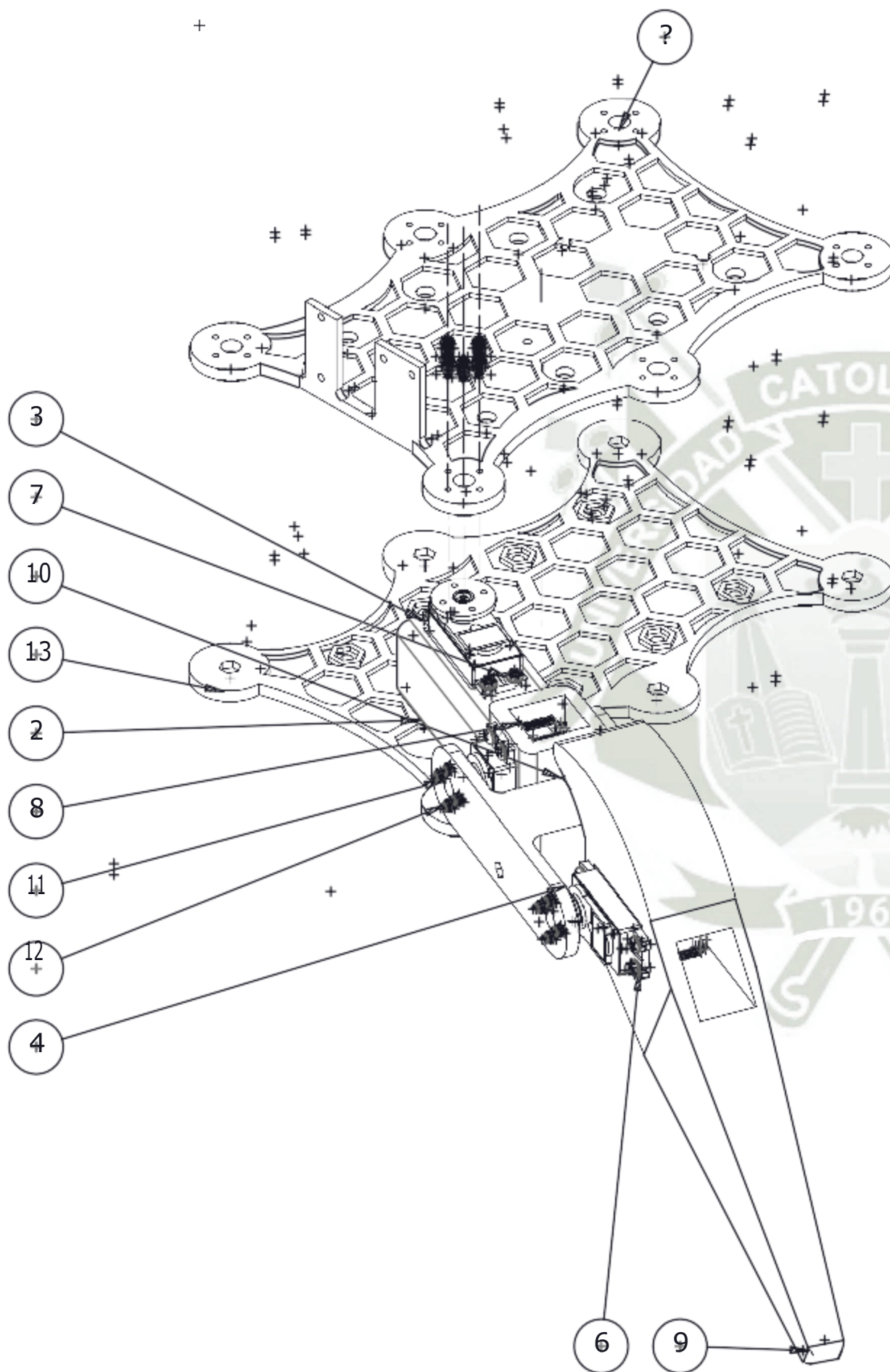
Escala: 1:2	1.11 Ensamble Caja de Servos	Pag: 11
		Págs: 1-14



N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	CAJA DE SERVOS1	1
2	TOWER PRO MG-995	3
3	ACCESORIO MG995_TOWERPRO	3
4	MR105ZZ RODAJE	3
5	PERNO222222	10
6	PERNO	2
7	TUERCA	12
8	PATA1	1
9	ACOPLE C1	1
10	ACOPLE C2	1
11	TORNILLO	10

	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	
Escala: 1:2	1.12 Ensamble Acople C		Pag: 12
			Págs: 1-14

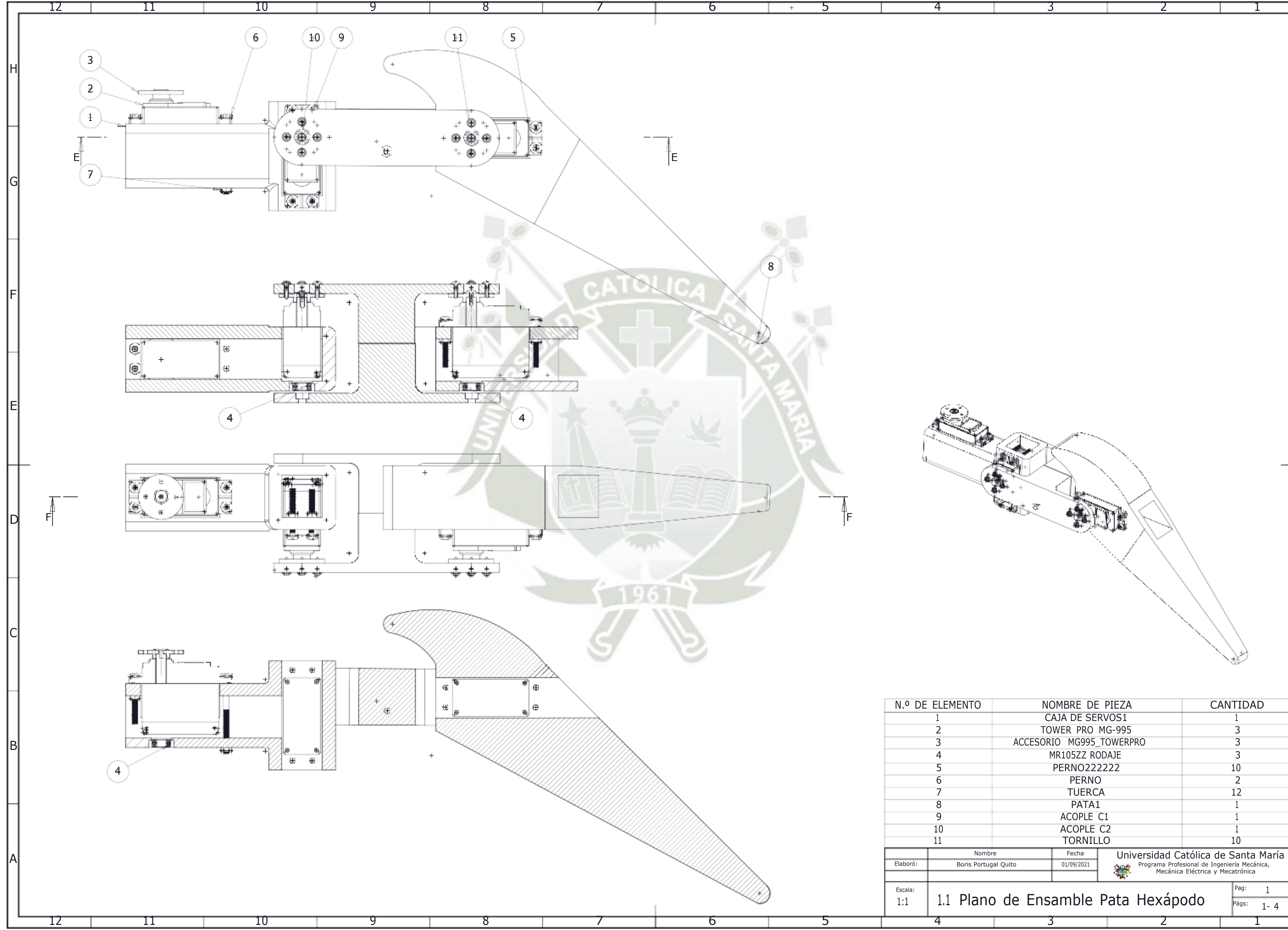
1.12 Ensamble Acople C



N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	TORAX22	1
2	CAJA DE SERVOS1	1
3	TOWER PRO MG-995	3
4	ACCESORIO MG995_TOWERPRO	3
5	MR105ZZ RODAJE	3
6	PERNO222222	10
7	PERNO	2
8	TUERCA	12
9	PATA1	1
10	ACOPLE C1	1
11	ACOPLE C2	1
12	TORNILLO	15
13	TORAX11	1

Elaboró:	Nombre Boris Portugal Quito	Fecha 01/09/2021	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Escala:	1:2		

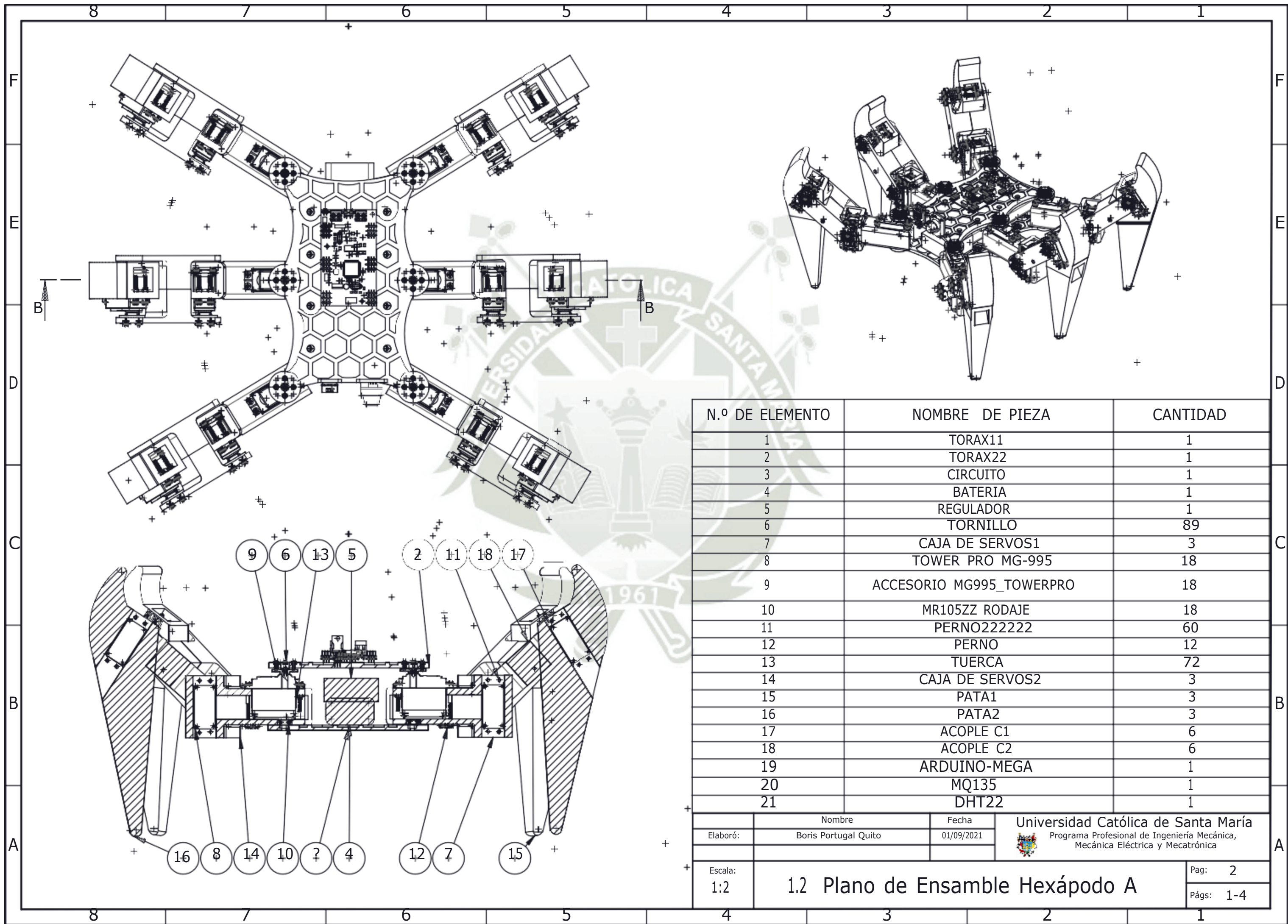
1.13 Ensamble Torax			Pag: 13
			Págs: 1-14



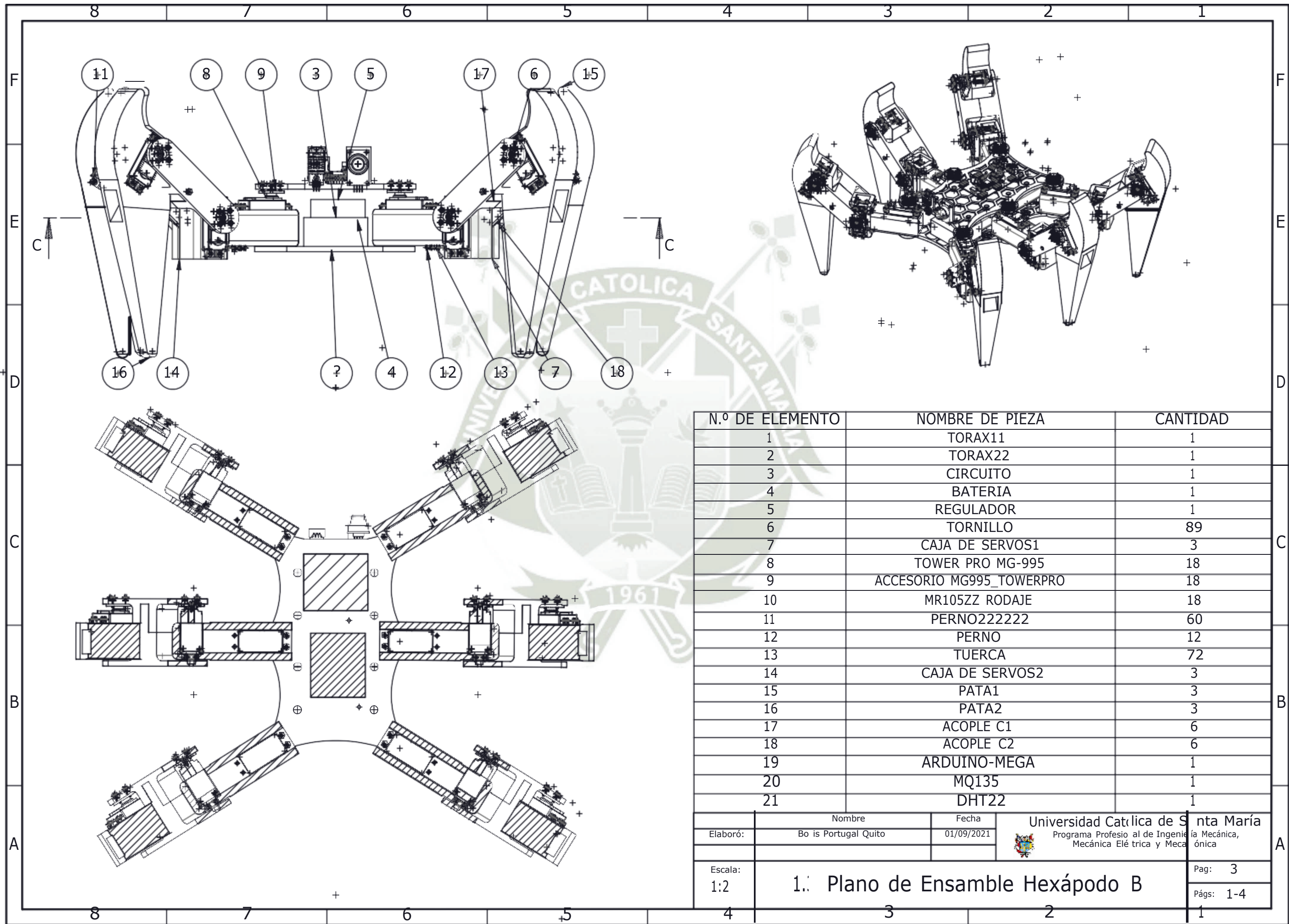
N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	CAJA DE SERVOS1	1
2	TOWER PRO MG-995	3
3	ACCESORIO MG995_TOWERPRO	3
4	MR105ZZ RODAJE	3
5	PERNO222222	10
6	PERNO	2
7	TUERCA	12
8	PATA1	1
9	ACOPLE C1	1
10	ACOPLE C2	1
11	TORNILLO	10

	Nombre	Fecha	 Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica
Elaboró:	Boris Portugal Quito	01/09/2021	

Escala: 1:1	1.1 Plano de Ensamble Pata Hexápodo	Pag: 1
		Págs: 1- 4

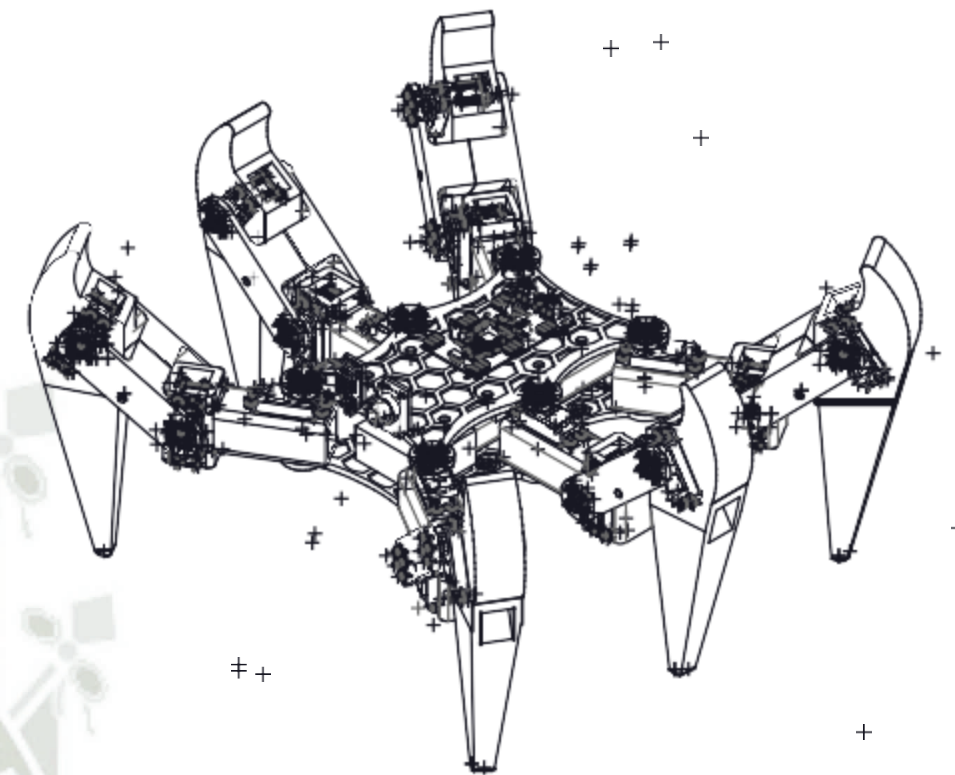
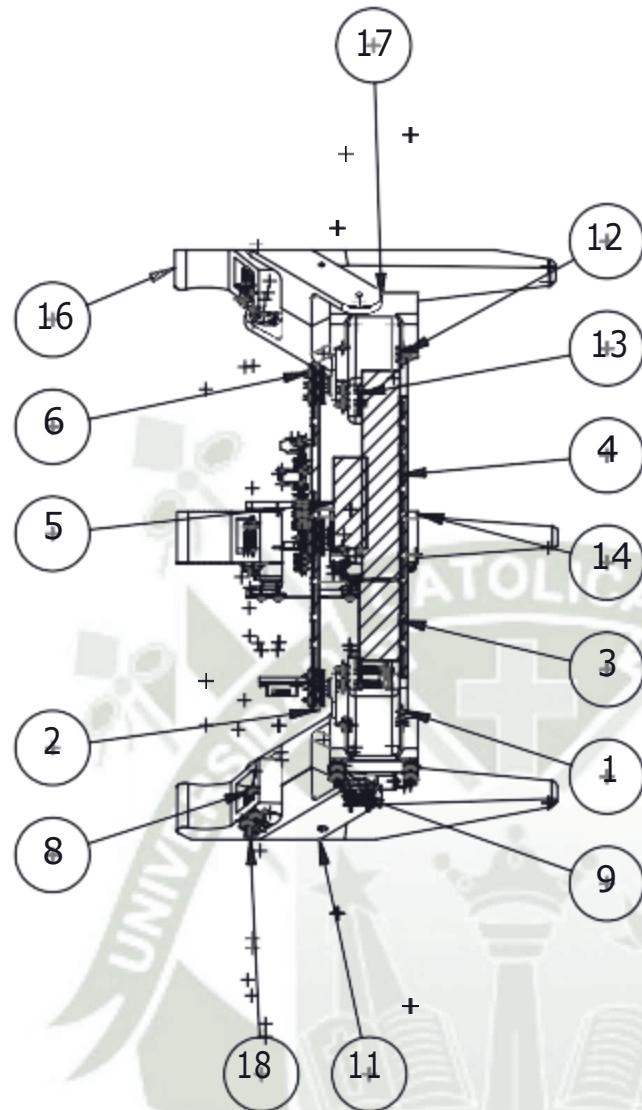
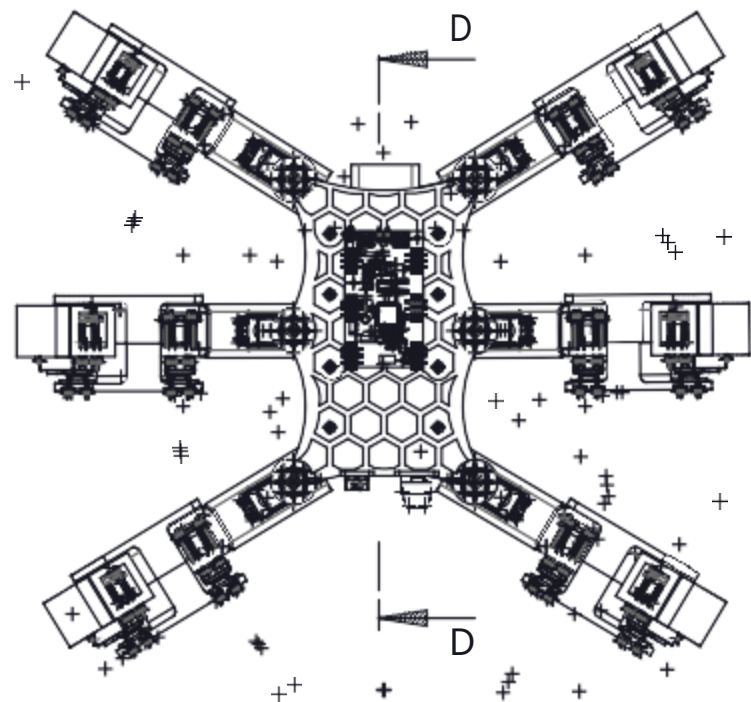


N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	TORAX11	1
2	TORAX22	1
3	CIRCUITO	1
4	BATERIA	1
5	REGULADOR	1
6	TORNILLO	89
7	CAJA DE SERVOS1	3
8	TOWER PRO MG-995	18
9	ACCESORIO MG995_TOWERPRO	18
10	MR105ZZ RODAJE	18
11	PERNO222222	60
12	PERNO	12
13	TUERCA	72
14	CAJA DE SERVOS2	3
15	PATA1	3
16	PATA2	3
17	ACOPLE C1	6
18	ACOPLE C2	6
19	ARDUINO-MEGA	1
20	MQ135	1
21	DHT22	1



N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	TORAX11	1
2	TORAX22	1
3	CIRCUITO	1
4	BATERIA	1
5	REGULADOR	1
6	TORNILLO	89
7	CAJA DE SERVOS1	3
8	TOWER PRO MG-995	18
9	ACCESORIO MG995_TOWERPRO	18
10	MR105ZZ RODAJE	18
11	PERNO222222	60
12	PERNO	12
13	TUERCA	72
14	CAJA DE SERVOS2	3
15	PATA1	3
16	PATA2	3
17	ACOPLE C1	6
18	ACOPLE C2	6
19	ARDUINO-MEGA	1
20	MQ135	1
21	DHT22	1

Elaboró:	Nombre Bo is Portugal Quito	Fecha 01/09/2021	Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecánica
----------	--------------------------------	---------------------	--



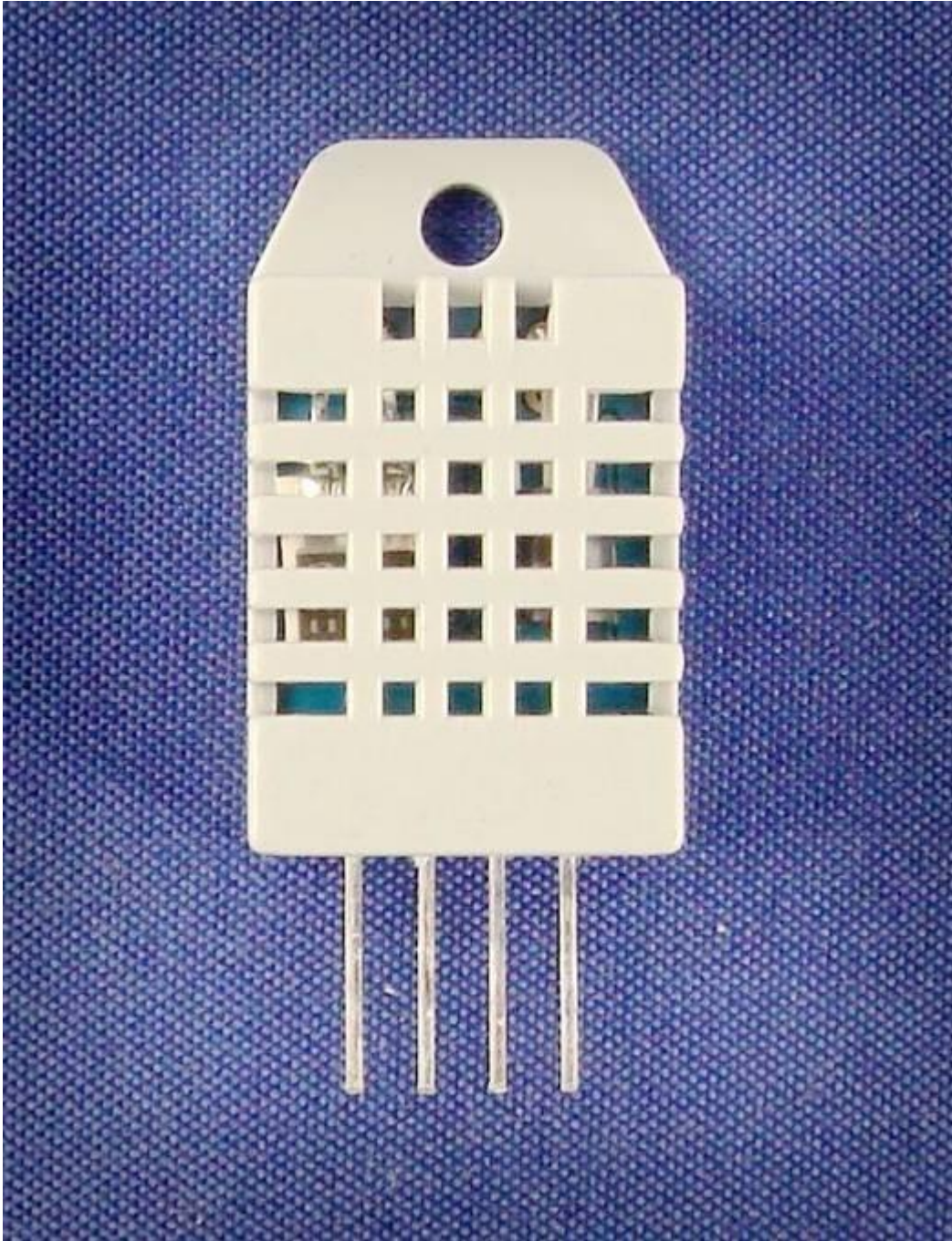
N.º DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	TORAX11	1
2	TORAX22	1
3	CIRCUITO	1
4	BATERIA	1
5	REGULADOR	1
6	TORNILLO	89
7	CAJA DE SERVOS1	3
8	TOWER PRO MG-995	18
9	ACCESORIO MG995_TOWERPRO	18
10	MR105ZZ RODAJE	18
11	PERNO222222	60
12	PERNO	12
13	TUERCA	72
14	CAJA DE SERVOS2	3
15	PATA1	3
16	PATA2	3
17	ACOPLE C1	6
18	ACOPLE C2	6
19	ARDUINO-MEGA	1
20	MQ135	1
21	DHT22	1

Elaboró:	Nombre Bois Portugal Quito	Fecha 01/09/2021	Universidad Católica de Santa María Programa Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecánica
Escala:	1:5		
14 Plano de Ensamble Hexápodo C			Pag: 4 Págs: 1-4

Anexo 8: DATASHEET SENSOR DHT22

Digital-output relative humidity & temperature sensor/module

DHT22 (DHT22 also named as AM2302)



Capacitive-type humidity and temperature module/sensor

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

- * Full range temperature compensated
- * Relative humidity and temperature measurement
- * Calibrated digital signal
- * Outstanding long-term stability
- * Extra components not needed
- * Long transmission distance
- * Low power consumption
- * 4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements are connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity $\pm 2\%$ RH (Max $\pm 5\%$ RH); temperature ± 0.5 Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1 Celsius
Repeatability	humidity $\pm 1\%$ RH; temperature ± 0.2 Celsius
Humidity hysteresis	$\pm 0.3\%$ RH
Long-term Stability	$\pm 0.5\%$ RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

4. Dimensions: (unit ---mm)

1) Small size dimensions: (unit--- mm)

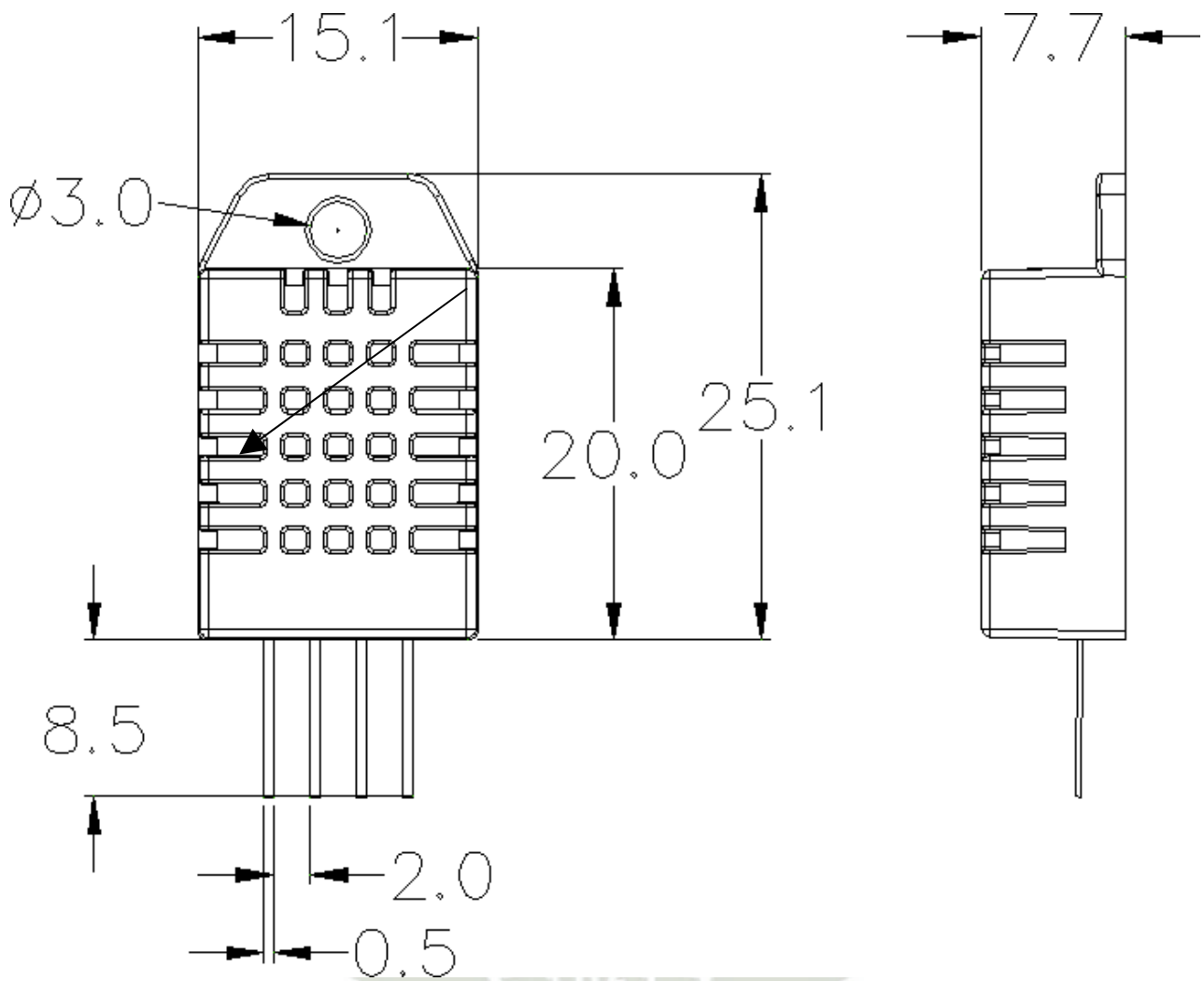
Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Aosong Electronics Co.,Ltd

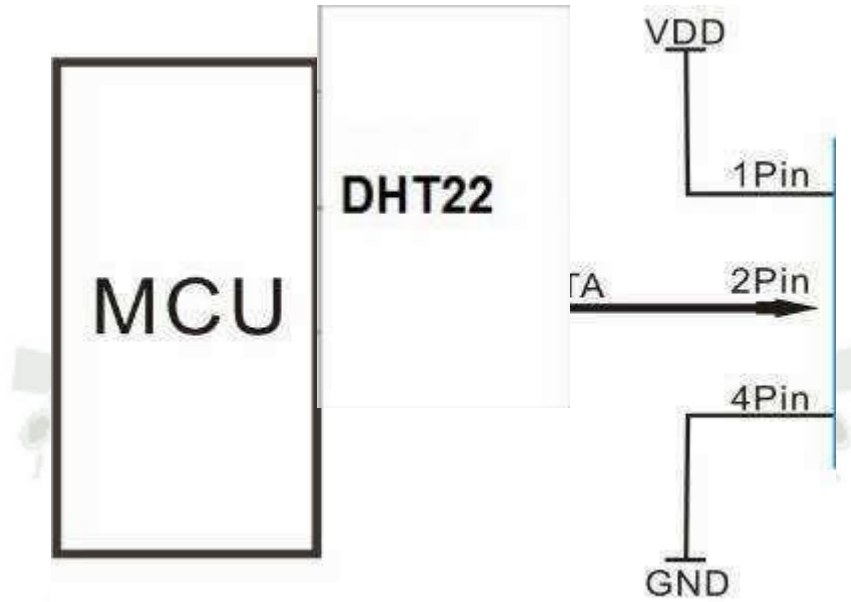
Your specialist in innovating humidity & temperature sensors



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD ---power supply
2	DATA--signal
3	NULL
4	GND

5. Electrical connection diagram:



3Pin---NC, AM2302 is another name for DHT22

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT22, it costs 5mS for single time communication.

Data is comprised of integral and decimal part, the following is the formula for data.

DHT22 send out higher data bit firstly!

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum
If the data transmission is right, check-sum should be the last 8 bit of "8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data".

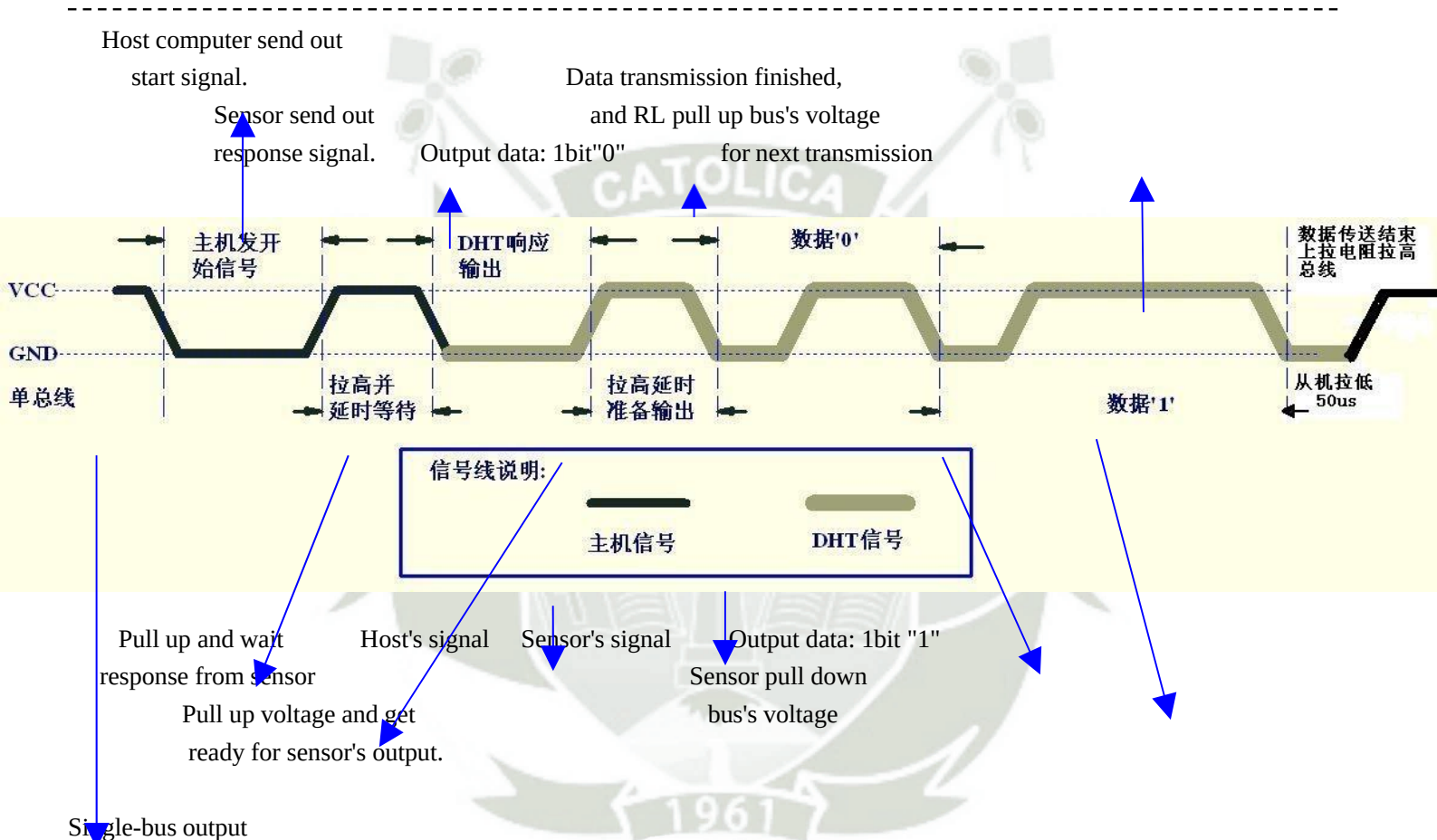
When MCU send start signal, DHT22 change from low-power-consumption-mode to running-mode. When MCU finishes sending the start signal, DHT22 will send response signal of 40-bit data that reflect the relative humidity

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

and temperature information to MCU. Without start signal from MCU, DHT22 will not give response signal to MCU. One start signal for one time's response data that reflect the relative humidity and temperature information from DHT22. DHT22 will change to low-power-consumption-mode when data collecting finish if it don't receive start signal from MCU again.

1) Check bellow picture for overall communication process:



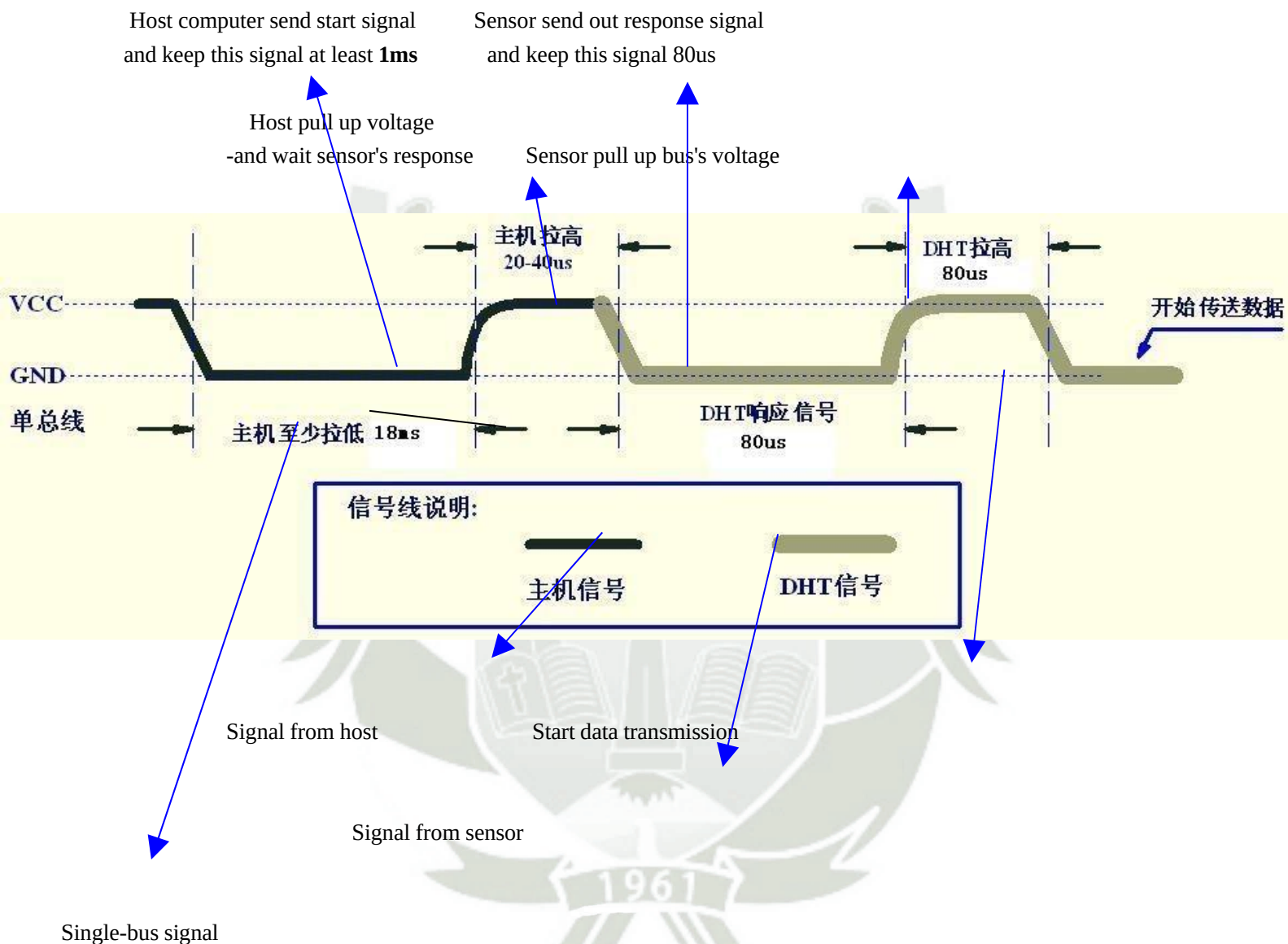
2) Step 1: MCU send out start signal to DHT22

Data-bus's free status is high voltage level. When communication between MCU and DHT22 begin, program of MCU will transform data-bus's voltage level from high to low level and this process must beyond at least 1ms to ensure DHT22 could detect MCU's signal, then MCU will wait 20-40us for DHT22's response.

Check bellow picture for step 1:

Aosong Electronics Co.,Ltd

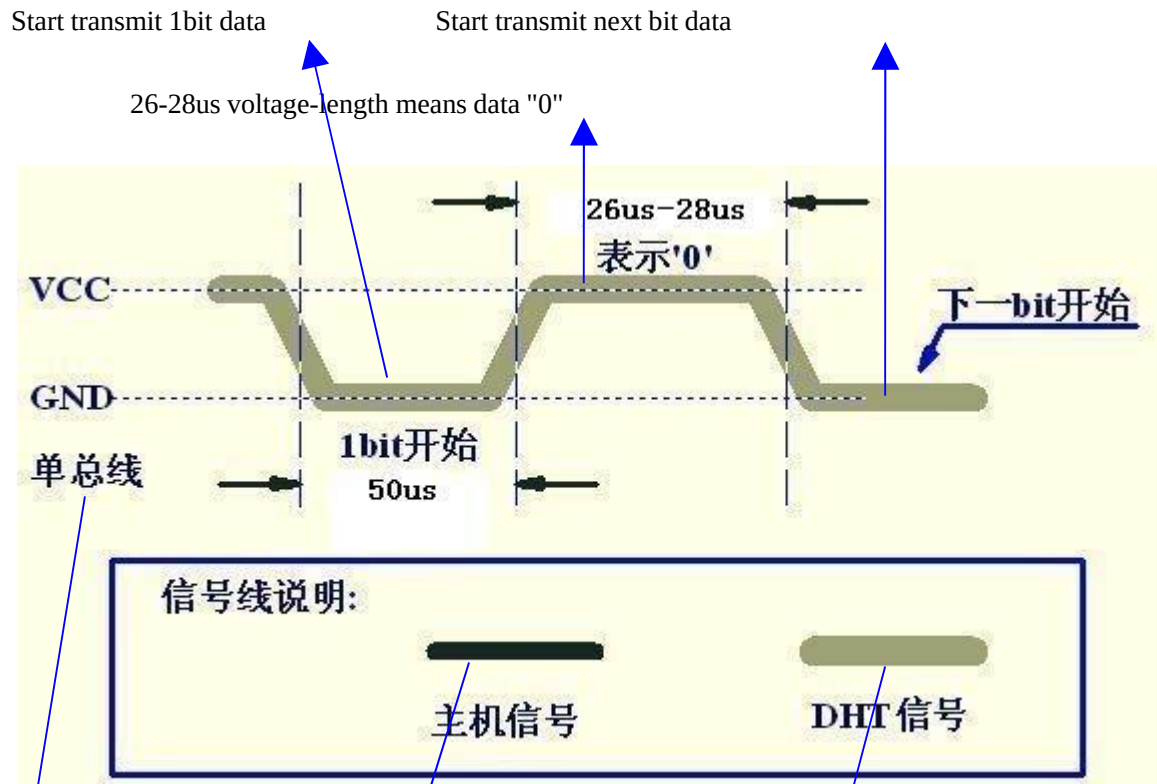
Your specialist in innovating humidity & temperature sensors



Step 2: DHT22 send response signal to MCU

When DHT22 detect the start signal, DHT22 will send out low-voltage-level signal and this signal last 80us as response signal, then program of DHT22 transform data-bus's voltage level from low to high level and last 80us for DHT22's preparation to send data.

Check bellow picture for step 2:



Host signal

Sesnor's signal

Single-bus signal

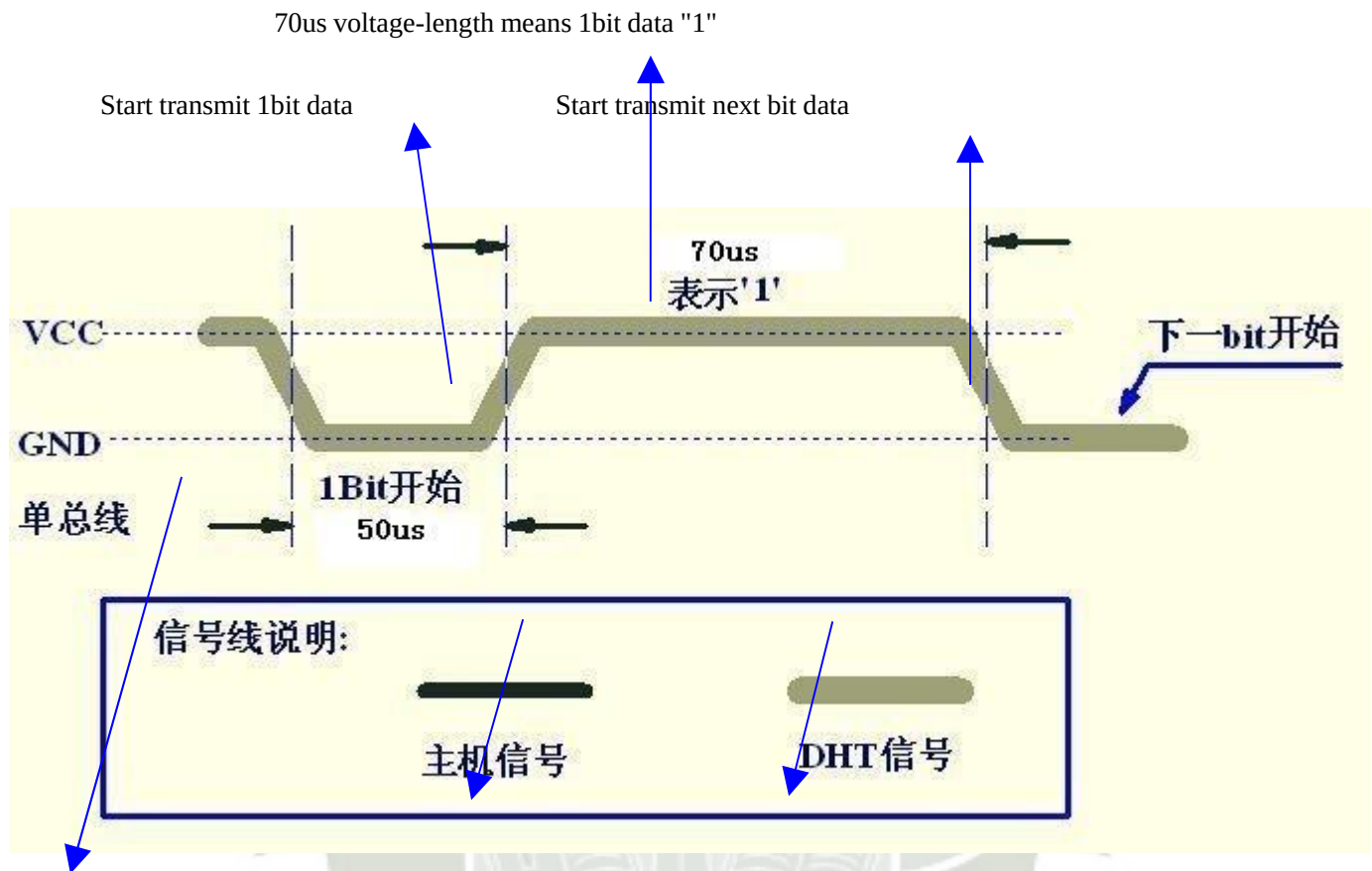
Step 3: DHT22 send data to MCU

When DHT22 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0".

Check bellow picture for step 3:

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors



Host signal

Sesnor's signal

Single-bus signal

If signal from DHT22 is always high-voltage-level, it means DHT22 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

*Collecting period should be : >2 second.

Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The DHT22 sensor can recover after working in non-normal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere DHT22's sensitive-elements and debase DHT22's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the DHT22 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the DHT22 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

DHT22 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase DHT22's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

- * Welding temperature should be bellow 260Celsius.
- * Avoid using the sensor under dew condition.
- * Don't use this product in safety or emergency stop devices or any other occasion that failure of DHT22 may cause personal injury.

Anexo 9: DATASHEET SENSOR MQ-135

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃, NO_x, alcohol, Benzene, smoke, CO₂, etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _C	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33Ω ±5%	Room Tem
P _H	Heating consumption	less than 800mw	

B. Environment condition

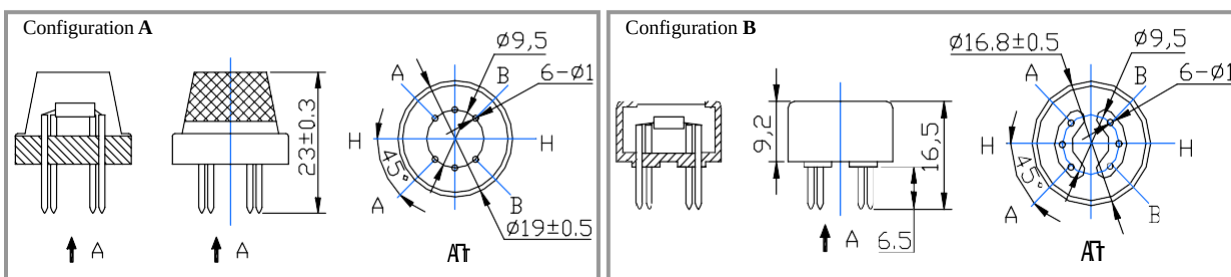
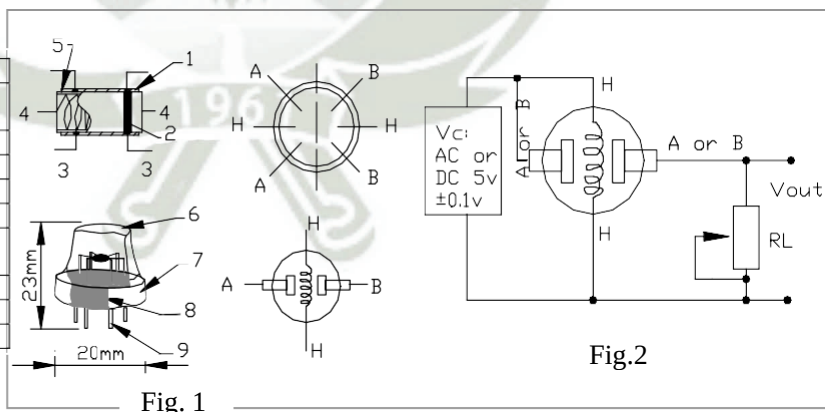
Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10℃ ... +45℃	
T _{as}	Storage Tem	-20℃ ... +70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
Rs	Sensing Resistance	30K Ω -200K Ω (100ppm NH ₃)	Detecting concentration scope: 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤ 0.65	
Standard Detecting Condition	Temp: 20℃ ±2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit

Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

sensitive components. The enveloped MQ-135 have 6 pins ,4 of them are used to fetch signals, and other 2 are used for providing heating current.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

Fig.2 sensitivity characteristics of the MQ-135

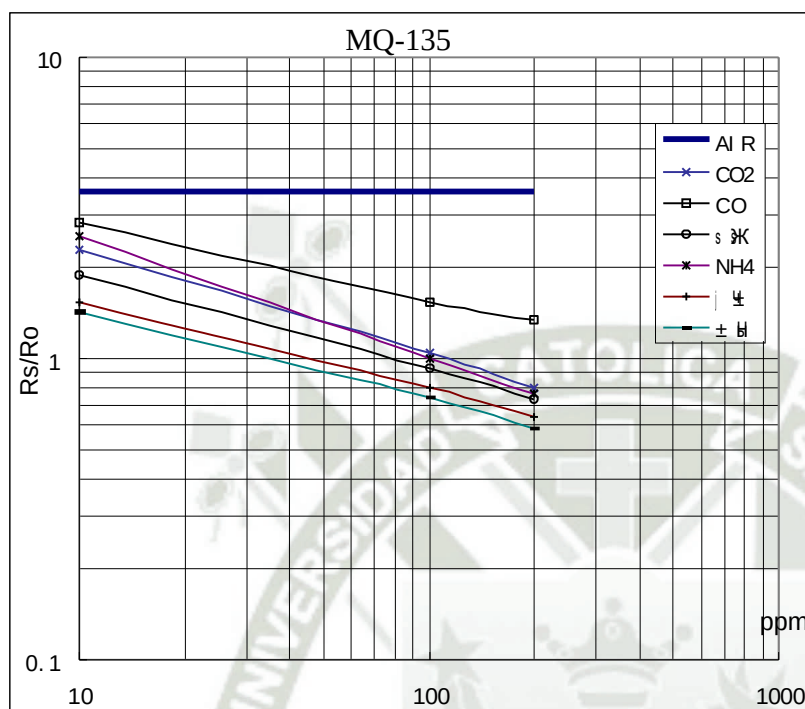


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases.

in their: Temp: 20℃、

Humidity: 65%、

O₂ concentration 21%

RL=20k Ω

Ro: sensor resistance at 100ppm of NH₃ in the clean air.

Rs: sensor resistance at various concentrations of gases.

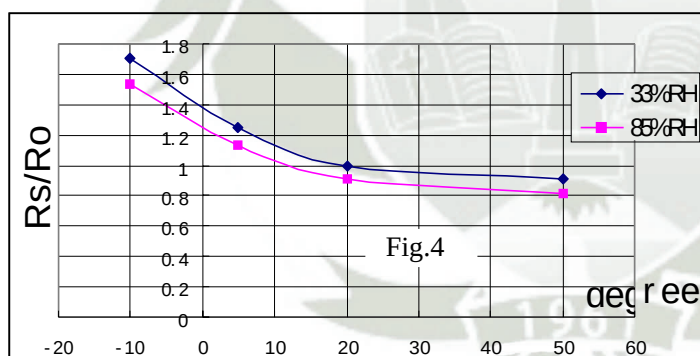


Fig.4 is shows the typical dependence of the MQ-135 on temperature and humidity.

Ro: sensor resistance at 100ppm of NH₃ in air at 33%RH and 20 degree.

Rs: sensor resistance at 100ppm of NH₃ at different temperatures and humidities.

SENSITIVITY ADJUSTMENT

Resistance value of MQ-135 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use value of Load resistance that (R_L) about 20 K Ω (10K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence.

