

CARRERA DE ELECTRÓNICA

DESARROLLO DE UN ROBOT HUMANOIDE IMPRESO EN 3D CON MOTRICIDAD AVANZADA EN EXTREMIDADES SUPERIORES, INSPIRADO EN LA BIOMECÁNICA HUMANA.

**Trabajo de Integración Curricular previo a la obtención del Título de
Tecnólogo Superior en Electrónica**

Autor: PERLAZA BRIONES OSCAR ANDRÉS

Tutor: ING. ROSERO YUGSI RICARDO ELICIO

Marzo, 2025

SANGOLQUI – ECUADOR

Autor: Oscar Andrés Perlaza Briones	Título a obtener: Tecnólogo Superior en Electrónica Matriz: Sangolquí -Ecuador Correo electrónico: oscar.perlaza@ister.edu.ec
	
Dirigido por: Ricardo Elicio Rosero Yugsi	Matriz: Sangolquí -Ecuador Correo electrónico: ricardo.rosero@ister.edu.ec
	

TODOS LOS DERECHOS RESERVADOS.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2024 Tecnológico Universitario Rumiñahui

SANGOLQUÍ – ECUADOR

PERLAZA BRIONES OSCAR ANDRES

**DESARROLLO DE UN ROBOT HUMANOIDE IMPRESO EN 3D CON
MOTRICIDAD AVANZADA EN EXTREMIDADES SUPERIORES,
INSPIRADO EN LA BIOMECÁNICA HUMANA.**

CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.2

Sangolquí, 13 de marzo del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, **PERLAZA BRIONES OSCAR ANDRÉS** declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado **DESARROLLO DE UN ROBOT HUMANOIDE IMPRESO EN 3D CON MOTRICIDAD AVANZADA EN EXTREMIDADES SUPERIORES, INSPIRADO EN LA BIOMECÁNICA HUMANA**, de la Tecnología Superior **ELECTRÓNICA**; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,



PERLAZA BRIONES OSCAR ANDRÉS
C.I.: 1723305130

FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO
SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA:

TECNOLOGÍA SUPERIOR EN ELECTRÓNICA.

AUTOR /ES:

OSCAR ANDRÉS PERLAZA BRIONES.

TUTOR (METODOLÓGICO Y ACADÉMICO):

RICARDO ELICIO ROSERO YUGSI.

CONTACTO ESTUDIANTE:

0992777686

CORREO ELECTRÓNICO:

oscar.perlaza@outlook.com

TEMA:

DESARROLLO DE UN ROBOT HUMANOIDE IMPRESO EN 3D CON MOTRICIDAD
AVANZADA EN EXTREMIDADES SUPERIORES, INSPIRADO EN LA
BIOMECÁNICA HUMANA.

OPCIÓN DE TITULACIÓN:

TRABAJO DE INTEGRACIÓN CURRICULAR.

RESUMEN EN ESPAÑOL:

El presente trabajo de titulación presenta el desarrollo de un robot humanoide impreso en 3D, con motricidad avanzada en las extremidades superiores. El propósito es aportar una herramienta educativa accesible que permita a los estudiantes de carreras técnicas poner en práctica conceptos como: diseño mecánico, electrónica, automatización y programación. La metodología aplicada comprendió: diseño, implementación y evaluación. En la fase de diseño se analizaron principios teóricos de robótica humanoide y biomecánica. En la implementación se fabricaron las piezas 3D, se ensamblaron los componentes, y se elaboró el código que permite ejecutar movimiento. En la evaluación, se realizaron pruebas prácticas para validar los movimientos y la funcionalidad del sistema. Al final se obtuvo un prototipo funcional, que permite simular de forma realista los movimientos humanos. El robot puede ser una herramienta de aprendizaje para robótica, máquinas eléctricas o programación, contribuyendo a la formación de profesionales. Además, el proyecto demuestra que es posible construir herramientas didácticas de gran valor. Esta experiencia aporta significativamente a la innovación educativa y al fortalecimiento del perfil tecnológico del Instituto Tecnológico Rumiñahui.

PALABRAS CLAVE:

Robot humanoide, impresión 3D, biomecánica, Arduino Mega, servomotores, control de movimiento.

ABSTRACT:

This graduation project presents the development of a 3D-printed humanoid robot with advanced upper limb mobility. Its purpose is to provide an accessible educational tool that enables students in technical fields to apply concepts such as mechanical design, electronics, automation, and programming in practice. The applied methodology consisted of three stages: design, implementation, and evaluation. During the design phase, theoretical principles of humanoid robotics and biomechanics were analyzed. In the implementation phase, 3D parts were fabricated, components were assembled, and the control code was developed to enable movement. In the evaluation phase, practical tests were carried out to validate the robot's functionality and movement precision. The result was a functional

prototype capable of realistically simulating human movement. The robot serves as a valuable learning tool in subjects such as robotics, electric machines, and programming, contributing to the training of future professionals. Furthermore, the project demonstrates that it is possible to build high-value didactic tools with accessible resources. This experience significantly contributes to educational innovation and to strengthening the technological profile of Instituto Tecnológico Rumiñahui.

KEYWORDS:

Humanoid robot, 3D printing, biomechanics, Arduino Mega, servomotors, motion control.



Firma del Estudiante (AUTOR)

Oscar Andrés Perlaza Briones.

C.I.: 1723305130

SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 13 de marzo del 2025

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, **Perlaza Briones Oscar Andrés**, con C.I.: **1723305130** alumno de la Carrera **TECNOLOGO SUPERIOR EN ELECTRONICA**, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,



Firma del Estudiante

Perlaza Briones Oscar Andrés

C.I.: **1723305130**

DEDICATORIA

A mis padres, con toda admiración. Este proyecto es posible gracias a ustedes, que sin dudarlos me brindaron su respaldo económico, confiando plenamente en mis capacidades y en mis sueños. Valoro profundamente cada sacrificio que hicieron para que pudiera dedicarme por completo a construir este objetivo. Saber que me ofrecieron su apoyo y los frutos de su esfuerzo, que agradezco inmensamente. Este logro es mío así como suyo; lo construimos con esfuerzo, confianza y amor incondicional.

Oscar P.

AGRADECIMIENTO

Mi gratitud a los docentes del Instituto, quienes durante mi formación pusieron a mi disposición sus conocimientos y su compromiso en mi preparación académica. Agradezco también la orientación, sugerencias y colaboración con los detalles de mi proyecto, que fueron fundamentales para superar los desafíos que se presentaron. Gracias por la huella profunda que han dejado en mi formación profesional y en la calidad de este trabajo que aquí presento.

Oscar P.

RESUMEN EJECUTIVO

El presente trabajo de titulación presenta el desarrollo de un robot humanoide impreso en 3D, con motricidad avanzada en las extremidades superiores. El propósito es aportar una herramienta educativa accesible que permita a los estudiantes de carreras técnicas poner en práctica conceptos como: diseño mecánico, electrónica, automatización y programación.

La metodología aplicada comprendió: diseño, implementación y evaluación. En la fase de diseño se analizaron principios teóricos de robótica humanoide y biomecánica. En la implementación se fabricaron las piezas 3D, se ensamblaron los componentes, y se elaboró el código que permite ejecutar movimiento. En la evaluación, se realizaron pruebas prácticas para validar los movimientos y la funcionalidad del sistema.

Al final se obtuvo un prototipo funcional, que permite simular de forma realista los movimientos humanos. El robot puede ser una herramienta de aprendizaje para robótica, máquinas eléctricas o programación, contribuyendo a la formación de profesionales. Además, el proyecto demuestra que es posible construir herramientas didácticas de gran valor. Esta experiencia aporta significativamente a la innovación educativa y al fortalecimiento del perfil tecnológico del Instituto Tecnológico Rumiñahui.

PALABRAS CLAVE

Robot humanoide, impresión 3D, biomecánica, Arduino Mega, servomotores, control de movimiento.

ABSTRACT

This graduation project presents the development of a 3D-printed humanoid robot with advanced upper limb mobility. Its purpose is to provide an accessible educational tool that enables students in technical fields to apply concepts such as mechanical design, electronics, automation, and programming in practice.

The applied methodology consisted of three stages: design, implementation, and evaluation. During the design phase, theoretical principles of humanoid robotics and biomechanics were analyzed. In the implementation phase, 3D parts were fabricated, components were assembled, and the control code was developed to enable movement. In the evaluation phase, practical tests were carried out to validate the robot's functionality and movement precision.

The result was a functional prototype capable of realistically simulating human movement. The robot serves as a valuable learning tool in subjects such as robotics, electric machines, and programming, contributing to the training of future professionals. Furthermore, the project demonstrates that it is possible to build high-value didactic tools with accessible resources. This experience significantly contributes to educational innovation and to strengthening the technological profile of Instituto Tecnológico Rumiñahui.

KEYWORDS

Humanoid robot, 3D printing, biomechanics, Arduino Mega, servomotors, motion control.

Índice general

TODOS LOS DERECHOS RESERVADOS.	2
DEDICATORIA	8
AGRADECIMIENTO	8
RESUMEN EJECUTIVO	9
PALABRAS CLAVE	9
ABSTRACT.....	10
KEYWORDS	10
Índice general.....	11
INTRODUCCIÓN	17
1. CAPÍTULO I	18
1.1. Planteamiento del problema.....	19
1.2. Preguntas de directrices	20
1.3. Objetivo General	20
1.4. Objetivos Especifico	20
1.5. Justificación	21
1.6. Alcance	22
2. CAPÍTULO II	23
2.1. Fundamento teórico relacionado al trabajo de integración curricular.	23
2.1.1. Robots humanoides.....	23
2.1.2. Impresión 3D	24
2.1.3. Técnicas de Impresión 3D	24
2.1.4. Impresora 3D recomendable para el desarrollo.	27
2.1.5. Materiales para Impresión 3D.....	28
2.1.6. Software de Diseño 3D	28
2.1.7. Componentes Electrónicos.....	30
2.1.8. Controladores y Actuadores.....	31
2.1.8.1 Microcontroladores	31
2.1.8.2 Actuadores	32
2.1.9. Programación y Algoritmos de Control.....	33
2.1.10. Software de Programación	34
3. CAPÍTULO III.....	35
3.1. Fase de diseño	35
3.1.1. Etapa1: Análisis sobre robots humanoides.	35
3.1.2. Etapa 2: Diseño de piezas 3D.	35

3.1.3.	Etapa 3: Diseño de esquema eléctrico y electrónico.....	35
3.2.	Fase de implementación.....	36
3.2.1.	Etapa 1: Parametrización de archivos stl.	36
3.2.2.	Etapa 2: Impresión de las piezas 3D.	36
3.2.3.	Etapa 3: Ensamble del robot humanoide.	36
3.2.4.	Etapa 4: Montaje de elementos electrónicos.....	37
3.2.5.	Etapa 5: Programación de movimientos y reacciones.	37
3.3.	Fase de evaluación	37
3.3.1.	Etapa 1: Prueba de programación de movimiento sincronizados.	37
3.3.2.	Etapa 2: Demostración práctica de biomecánica y movimientos robóticos.	38
3.3.3.	Etapa 3: Programación de tareas prácticas para explorar sistemas electrónicos del robot.	38
4.	CAPÍTULO IV.....	39
4.1.	Diseño de la propuesta	39
4.1.1.	Análisis sobre robots humanoides	39
4.1.2.	Diseño de piezas 3D.....	41
4.1.3.	Diseño de esquema eléctrico y electrónico	43
4.2.	Implementación.....	45
4.2.1.	Parametrización de archivos stl.	45
4.2.2.	Impresión de piezas 3D.....	47
4.2.3.	Ensamble del robot humanoide.....	48
4.2.4.	Montaje de elementos electrónicos.....	72
4.2.5.	Programación de movimientos y reacciones.....	73
4.3.	Pruebas.....	76
4.3.1.	Prueba de programación de movimientos sincronizados.....	76
4.3.2.	Demostración práctica de biomecánica y movimientos robóticos.....	81
4.4.	Resultados	83
4.4.1.	Programación de tareas prácticas para explorar sistemas electrónicos del robot	83
5.	CAPITULO V	84
5.1.	Conclusiones	84
5.2.	Recomendaciones	86
5.3.	Referencias.....	88
6.		90
7.	ANEXOS	91

Índice de Figuras

Figura 1.	Impresora Bambu Lab	27
Figura 2.	Original Prusa XL.....	27
Figura 3.	PLA Printer Filament.....	28
Figura 4.	Arduino Uno	31
Figura 5.	Arduino Mega.....	31
Figura 6.	HK15298B.....	32
Figura 7.	Hitec HS-805BB.....	32
Figura 8.	Servo MG996R.....	33
Figura 9.	Clasificación de piezas	42
Figura 10.	Piezas específicas.....	42
Figura 11.	Piezas específicas.....	43
Figura 12.	Distribución de servos.....	44
Figura 13.	Conexión electrónica de servos.	45
Figura 14.	Stl. Mano izquierda.....	45
Figura 15.	Stl. Biceps.	46
Figura 16.	Proceso de impresión 1.	47
Figura 17.	Proceso de impresión 2.	47
Figura 18.	Ensamble cabeza.....	48
Figura 19.	Piezas cabeza	49
Figura 20.	Estructura ojos.	49
Figura 21.	Servos estructura ojos.	50
Figura 22.	Estructura ojos, final.....	50
Figura 23.	Estructura cuello.	51
Figura 24.	Estructura mandíbula.	51

Figura 25.	Ensamblaje cabeza posterior.....	52
Figura 26.	Ensamblaje cabeza final.....	52
Figura 27.	Ensamblaje torso.....	53
Figura 28.	Estructura trasera.	54
Figura 29.	Estructura frontal.	54
Figura 30.	Estructura intermedia.	54
Figura 31.	Estructura pecho.....	55
Figura 32.	Estructura espalda.	55
Figura 33.	Estructura espalda completa.	56
Figura 34.	Ensamblaje mano y brazo.	56
Figura 35.	Antebrazo.	57
Figura 36.	Estructura para servos, antebrazos.	57
Figura 37.	Antebrazo, servos ubicados.	58
Figura 38.	Servo muñeca.....	58
Figura 39.	Ensamble brazo y muñeca completa.....	59
Figura 40.	Piezas mano.	59
Figura 41.	Ensamble de dedos.....	60
Figura 42.	Mano, ensamble final.....	60
Figura 43.	Hilo nylon en manos.	61
Figura 44.	Hilo de nylon en muñecas.....	61
Figura 45.	Puntas de los dedos en manos.....	61
Figura 46.	Ensamblaje bíceps y hombros.....	62
Figura 47.	Piezas bíceps.	63
Figura 48.	Estructura base bíceps.	63
Figura 49.	Estructura para servo.....	64
Figura 50.	Estructura servo final.	64

Figura 51.	Protecciones de bíseps.	65
Figura 52.	Piezas de hombros.....	65
Figura 53.	Hombro completo.	66
Figura 54.	Lubricación de estructura de rotación, hombro.	66
Figura 55.	Ensamblaje estómago.....	67
Figura 56.	Piezas de base de rotación.....	68
Figura 57.	Estructura de movimiento lateral.	68
Figura 58.	Estructura base de movimiento lateral.	69
Figura 59.	Lubricación de movimiento lateral.	69
Figura 60.	Estructura torso y movimiento lateral.....	70
Figura 61.	Ensamblaje torso.....	70
Figura 62.	Base de rotación.....	71
Figura 63.	Ensamblaje completo del cuerpo.	71
Figura 64.	Ensamblaje electrónico.....	72
Figura 65.	Inicialización de servos.....	73
Figura 66.	Comandos básicos 1.....	74
Figura 67.	Comandos básicos 2.....	75
Figura 68.	Comandos básicos 3.....	75
Figura 69.	Inicialización de servos.....	76
Figura 70.	Estructura de control de comandos.....	77
Figura 71.	Estructura de control de servos.....	78
Figura 72.	Estado base del robot humanoide.	79
Figura 73.	Movimiento.....	80
Figura 74.	Limitación en hombro.....	81
Figura 75.	Limitación de movimiento por tiempo.	82
Figura 76.	Servos en brazo.....	83

Figura 77.	Movimiento 2.....	84
------------	-------------------	----

Lista de Anexos

Anexo 1.	2Servos.control	91
Anexo 2.	Control.Comandos	96

INTRODUCCIÓN

Desde siempre sentí atracción por la robótica y electrónica. Me fascinaba la idea de que una máquina pudiera imitar algo tan complejo como el movimiento humano. Por coincidencia, mi tutor propuso desarrollar un robot humanoide como proyecto de titulación, y tomé la oportunidad para cumplir este deseo: construir algo funcional que conectara la tecnología con la forma en que aprendemos.

Este trabajo aborda el diseño, construcción y programación de un robot humanoide impreso en 3D, con motricidad avanzada en las extremidades superiores, inspirado en la biomecánica humana. Con este proyecto se planea ofrecer una herramienta educativa que ayude a los estudiantes de carreras técnicas a comprender y aplicar conceptos como el diseño mecánico, la electrónica, la automatización y la programación.

La idea de construir pieza por pieza a este robot me abrumaba, y pronto me di cuenta de lo costoso que fue equivocarme al imprimir partes mal diseñadas. Además, no tenía claro desde el inicio qué servomotores u otros elementos electrónicos eran los adecuados para el desarrollo, lo que generó retrasos y algunos errores. Me sorprendió encontrarme con desafíos que no había previsto, como la necesidad de servos de mayor tamaño y la falta de una guía clara para el ensamblaje. Tuve que apoyarme en videos y fotos para identificar piezas faltantes, mal ubicadas o incluso inexistentes.

Durante el desarrollo aprendí algo más que solo teoría técnica: descubrí la importancia de planificar cuidadosamente, de detenerme a analizar los requisitos del proyecto antes de avanzar, y, sobre todo, de tener paciencia. A veces me desvelé ajustando detalles que podrían haberse resuelto con mejor organización desde el inicio.

Pese a todo, el resultado es un prototipo funcional que simula con realismo los movimientos humanos y que puede ser utilizado en clases de robótica, máquinas eléctricas o programación. Espero que este robot sirva a otros estudiantes como una plataforma práctica para explorar lo que yo mismo tuve que descubrir paso a paso. También deseo que lo mejoren, que se equivoquen, que aprendan, y que se sientan tan motivados como yo al ver que un diseño en papel puede cobrar vida.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

Uno de los mayores obstáculos en la enseñanza de la robótica es la falta de experiencias prácticas. Muchos estudiantes se enfrentan a la programación de motores, sensores y estructuras complejas sin haber pasado por una base progresiva que les permita entender paso a paso cómo funciona un sistema robótico. Ir directamente a construir sin ejemplos o demostraciones sencillas impide que puedan desarrollar sus ideas o entender el verdadero potencial de lo que están aprendiendo.

Este fue también mi caso. Al iniciar este proyecto, me encontré investigando sobre control de motores sin tener una idea clara de cómo se vería el resultado. Mover un servo parece sencillo en teoría, pero lograr que ese movimiento tenga un propósito claro, que se vea bien y que cumpla una función concreta, es mucho más difícil. De hecho, aprendí más equivocándome, dañando servos, adaptando piezas y corrigiendo errores que leyendo guías técnicas.

La robótica humanoide representa un reto aún mayor por su complejidad, pero también ofrece una ventaja educativa clave: permite que cada estudiante explore diferentes aspectos del diseño, la programación y los componentes electrónicos. A diferencia de un robot seguidor de línea, donde todos replican el mismo comportamiento, un robot humanoide puede dividirse en módulos como brazos, sensores o mecanismos de agarre. Así, cada estudiante puede contribuir con una parte distinta del proyecto general, promoviendo la creatividad, la colaboración y la resolución de problemas reales.

Hoy en día, no existen suficientes plataformas accesibles en las aulas que permitan a los estudiantes enfrentarse a este tipo de desafíos reales. Desarrollar un robot humanoide impreso en 3D, accesible y funcional, podría ser una herramienta clave para transformar la educación técnica. No solo serviría como apoyo para aprender conceptos complejos, sino que también motivaría a los estudiantes a experimentar, mejorar y adaptar tecnología existente a sus propias ideas.

1.1. Planteamiento del problema

La robótica humanoide es una de las áreas más fascinantes y complejas dentro del mundo tecnológico. Su desarrollo implica entender no solo mecánica y electrónica, sino también cómo se relacionan estas disciplinas con el movimiento humano. Sin embargo, en el ámbito educativo, esta complejidad se vuelve una barrera cuando no existen herramientas prácticas que permitan a los estudiantes experimentar de forma progresiva.

En mi experiencia personal, descubrí que no basta con tener acceso a motores y sensores. Me enfrenté a la tarea de programar movimientos sin una guía clara, sin saber siquiera si los componentes que usaba eran los más adecuados. Por ejemplo, trabajé con servomotores de 360° y 180°, y no entendí hasta mucho después que podía modificar algunos para lograr una mejor precisión. Esa falta de información concreta, aplicada al contexto educativo, genera frustración, pérdida de recursos y desmotivación.

Actualmente, muchos estudiantes reciben tareas complejas sin haber pasado por etapas previas que les permitan construir su conocimiento desde lo simple. Se espera que comprendan cómo hacer funcionar un robot sin haber tenido la oportunidad de ver uno funcionando en condiciones reales. Esta desconexión entre la teoría y la práctica limita gravemente el aprendizaje y hace que la robótica se perciba como algo inaccesible o reservado solo para expertos.

Por otro lado, la falta de una plataforma común de trabajo, como un robot humanoide modular, impide que los estudiantes participen activamente en clases prácticas. Si existiera un solo robot en el aula, cada grupo o individuo podría encargarse de desarrollar una parte: diseñar una nueva mano, mejorar el sistema de agarre, agregar sensores o corregir fallas estructurales, como las que yo mismo enfrenté al tener que adaptar piezas mal dimensionadas. Esta posibilidad de intervenir directamente fomenta el aprendizaje activo, la creatividad y la colaboración.

Por eso, el desarrollo de un robot humanoide accesible y funcional no solo responde a una necesidad técnica, sino también educativa. Ofrecer a los estudiantes una herramienta tangible y adaptable permitiría transformar el enfoque tradicional de enseñanza, llevando el conocimiento desde el papel hasta la experiencia real.

1.2.Preguntas de directrices

¿Cómo puede un robot humanoide con motricidad avanzada en extremidades superiores mejorar la comprensión de la biomecánica y la robótica en estudiantes de áreas técnicas y científicas?

¿Qué elementos de diseño y construcción debe tener un robot humanoide impreso en 3D para que realmente sea útil como herramienta educativa en clases de programación y control de robots?

¿Cómo puede el uso de algoritmos de control en un robot humanoide ayudar a que los estudiantes comprendan de forma práctica conceptos de automatización y sensores en su formación?

1.3.Objetivo General

Desarrollar un robot humanoide capaz de simular la motricidad esencial de brazos y manos humanas en el torso superior, empleando tecnologías como la impresión 3D, algoritmos de programación y control de robots, con el fin de facilitar la comprensión de conceptos complejos relacionados con el diseño.

1.4.Objetivos Especifico

OBJ 1: Analizar los fundamentos de la robótica humanoide, la biomecánica humana y la motricidad avanzada en las extremidades superiores.

OBJ 2: Diseñar un prototipo de robot humanoide impreso en 3D, considerando aspectos estructurales, funcionales y educativos.

OBJ 3: Elaborar un código de programación que permita controlar los movimientos del robot humanoide de forma precisa y funcional.

1.5. Justificación

Este proyecto nace de una motivación personal y educativa. Desde muy joven, soñé con construir un robot humanoide, y este trabajo de titulación representó la oportunidad de hacerlo realidad. Más allá del desafío técnico, el propósito también fue aportar al Instituto Tecnológico Rumiñahui una herramienta tangible con la que los estudiantes puedan aprender, experimentar y desarrollar sus habilidades en robótica de forma práctica.

El robot fue diseñado e impreso en 3D con el objetivo de simular los movimientos esenciales de los brazos y manos humanas. La impresión 3D fue elegida no solo por su accesibilidad y bajo costo, sino por su capacidad de liberar la creatividad. Hoy en día, esta tecnología permite que cualquier persona con ideas pueda materializarlas, probarlas y mejorarlas, convirtiéndose en un recurso ideal para fomentar la innovación dentro del aula.

Uno de los principales aportes del proyecto es que los estudiantes podrán utilizar este robot para practicar programación, entender cómo funcionan sus componentes internos, modificar gestos o incluso rediseñar las piezas desde software de modelado para adaptarlas a nuevas funciones. Esta libertad de modificación no solo mejora la comprensión de temas técnicos como la electrónica o el control de movimientos, sino que también fortalece habilidades creativas y de resolución de problemas.

A nivel institucional, este robot puede convertirse en un puente entre la teoría y la práctica. Su presencia en el aula permitirá a los docentes implementar metodologías activas de enseñanza, haciendo que los estudiantes comprendan de forma más rápida y clara los conceptos relacionados con diseño 3D, impresión, microcontroladores y biomecánica aplicada. El tamaño y la forma humanoide del robot hacen que su estudio sea intuitivo, ya que se puede observar claramente cómo interactúan las piezas, cómo se conectan, y cómo influye la estructura en el movimiento.

Además, este tipo de proyecto responde a una necesidad real del entorno académico y profesional. La industria actual demanda profesionales que no solo comprendan los conceptos técnicos, sino que sepan aplicarlos, tomar decisiones estructurales y cómo equilibrar lo estético con lo funcional. Todo esto forma parte del aprendizaje integral que puede marcar la diferencia en la preparación de futuros profesionales.

Finalmente, este proyecto está diseñado para evolucionar. Espero que otros estudiantes puedan replicar o mejorar las piezas, rediseñar los soportes de los servos para lograr una mejor integración, optimizar el cableado interno, e incluso incorporar sensores o inteligencia artificial que permitan una interacción más autónoma con el entorno. Lo que hoy es un prototipo, mañana puede convertirse en una plataforma de aprendizaje colaborativo y continuo, alineada con los retos actuales y futuros de la educación tecnológica.

1.6.Alcance

El presente proyecto tiene como objetivo el desarrollo de un robot humanoide con motricidad avanzada en las extremidades superiores, centrado en el torso y brazos debido a que esta zona del cuerpo humano es una de las más expresivas. A diferencia de construir extremidades individuales como piernas o una cabeza funcional, se optó por un diseño que permitiera simular gestos y movimientos preprogramados capaces de transmitir intenciones, con el fin de crear una herramienta educativa orientada a la interacción y al análisis funcional de sistemas complejos.

El prototipo fue fabricado utilizando impresión 3D con filamento PLA+ de la marca Jamg He, en conjunto con una placa Arduino Mega y servomotores metálicos MG996R y MG90S. La construcción tomó aproximadamente una semana y media, pero la programación de los gestos sincronizados (especialmente por el uso combinado de servos de 180° y 360°) requirió casi un mes de ajustes y pruebas. La mayor dificultad fue lograr que los movimientos de los servos de rotación continua se movieran de forma precisa, lo que exigió mucha experimentación para alcanzar una sincronía aceptable.

Aunque las piezas fueron impresas por especialistas en impresión 3D y se recibieron valiosas sugerencias por parte de profesores para resolver problemas estructurales o de compatibilidad, el ensamblaje completo del robot y toda su programación fueron desarrollados de forma individual.

Este robot está pensado como una herramienta de aplicación directa en varias asignaturas de la carrera de Electrónica en el Instituto Tecnológico Rumiñahui. Su diseño modular y funcional

lo hace útil para el aprendizaje práctico en áreas como Robótica, Electrónica Analógica, Programación, Microcontroladores, Control y Diseño. Su implementación permitirá a los estudiantes comprender de forma tangible conceptos complejos, al integrar diferentes áreas del conocimiento en un solo proyecto didáctico.

El alcance del proyecto también considera su uso como base para futuras mejoras. Los estudiantes podrán rediseñar piezas impresas, perfeccionar su integración mecánica, o bien añadir sensores e inteligencia artificial que permitan al robot interactuar con el entorno. De este modo, el robot no solo cumple una función formativa inicial, sino que también abre espacio a la exploración y evolución tecnológica dentro del entorno académico.

Aunque no existe una certeza sobre cómo será aprovechado a largo plazo por los docentes, se espera que el prototipo sirva como una plataforma abierta que pueda integrarse en diferentes asignaturas, promoviendo así la creatividad, el trabajo en equipo y la innovación dentro del proceso de enseñanza-aprendizaje.

CAPÍTULO II

MARCO DE REFERENCIAS

2.1. Fundamento teórico relacionado al trabajo de integración curricular.

2.1.1. Robots humanoides

El diseño de robots humanoides es un campo fascinante y multidisciplinario que combina la ingeniería mecánica, la robótica, la electrónica y la inteligencia artificial. Un elemento fundamental en este diseño es la creación de un sistema mecánico que permita al robot humanoide imitar las acciones y movimientos de un ser humano. Para lograr esto, se utilizan actuadores, sensores, motores y mecanismos que simulan el movimiento natural de las extremidades, lo que permite que el robot camine, se desplace y realice tareas similares a las de una persona. Esta complejidad en el diseño exige el uso de plataformas avanzadas de simulación y modelado, que ayudan a predecir cómo interactúan los componentes antes de construir el robot. Además, la tecnología de simulación es crucial para acortar el tiempo de diseño y las pruebas físicas, lo que facilita una iteración más rápida y precisa en el proceso de desarrollo. (*Diseño de robots | Siemens Software, 2025*)

Un gran reto en el diseño de robots humanoides es desarrollar un sistema de control que gestione de manera efectiva los movimientos y las interacciones del robot con su entorno. Estos robots necesitan ser capaces de tomar decisiones por sí mismos o de colaborar con los humanos en diversas tareas. Para lograrlo, se utilizan algoritmos avanzados de control y técnicas de inteligencia artificial que permiten al robot reconocer su entorno, planificar acciones y adaptarse a situaciones cambiantes. Mantener la estabilidad es crucial, ya que un robot humanoide debe ser capaz de equilibrarse mientras se mueve o interactúa con objetos y personas. Los algoritmos de control dinámico, especialmente aquellos que se basan en el aprendizaje por refuerzo, son fundamentales para optimizar la ejecución de tareas y mejorar la capacidad del robot para aprender de la experiencia. ((PDF) DISEÑO Y CONTROL DE ROBOTS INDUSTRIALES, 2025)

La fabricación de robots humanoides es un aspecto fundamental del diseño, ya que implica crear componentes de alta precisión que deben encajar a la perfección. La impresión 3D se ha convertido en una herramienta clave para producir piezas personalizadas con formas complejas que serían difíciles de hacer con métodos tradicionales. Esta tecnología permite diseñar partes de robots que son ligeras y resistentes, algo esencial para su eficiencia. Además, los procesos de fabricación deben tener en cuenta factores como la durabilidad de los materiales, el consumo de energía y la facilidad de mantenimiento, ya que los robots humanoides a menudo se utilizan en entornos industriales y necesitan funcionar durante largos períodos sin fallos. La fusión de un diseño innovador y una fabricación precisa es crucial para desarrollar robots humanoides que sean funcionales, duraderos y rentables. (*Diseño de robots* | Siemens Software, 2025)

2.1.2. Impresión 3D

La impresión 3D, que también se conoce como manufactura aditiva, es una tecnología que ha cambiado las reglas del juego. Permite crear objetos tridimensionales al añadir material en capas, todo a partir de un modelo digital. Aunque los plásticos y metales son los más comunes, también se pueden usar otros materiales como hormigón o tejidos vivos, lo que abre un abanico de posibilidades en campos como la construcción y la medicina. Este método no solo minimiza el desperdicio en comparación con las técnicas tradicionales, sino que también permite la creación de diseños complejos que serían imposibles de realizar de otra manera. (*¿Qué es la impresión 3D?* | Programa para impresora 3D | Autodesk, 2025)

2.1.3. Técnicas de Impresión 3D

La extrusión de material, que muchos conocen como FDM (Modelado por Deposición Fundida), es una de las tecnologías de impresión 3D más populares y accesibles. Este proceso implica calentar un filamento termoplástico en una boquilla, que luego se deposita capa por capa sobre una plataforma. Los materiales que se utilizan incluyen PLA, ABS y PETG, entre otros, lo que permite una amplia variedad de aplicaciones, desde la creación de prototipos hasta la fabricación de piezas funcionales. Sus ventajas incluyen bajo costo, facilidad de uso y una gran variedad de colores y materiales. Sin embargo, la precisión es limitada en comparación con otras tecnologías, lo que la hace ideal para proyectos básicos. *(Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025)*

La polimerización en tanque, representada por tecnologías como SLA y DLP, utiliza resinas fotopoliméricas que se curan selectivamente mediante luz. Este proceso logra un acabado superficial extremadamente liso y detalles precisos, lo que lo hace ideal para aplicaciones dentales, de joyería y prototipos complejos. Cada capa de material líquido se solidifica en un tanque utilizando láseres o proyectores, dependiendo del método. Aunque ofrece alta calidad, las impresoras y resinas asociadas suelen ser más caras y menos versátiles que las de extrusión de material. Esta tecnología destaca por su capacidad para producir piezas con detalles excepcionales y acabado profesional. *(Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025)*

La fusión en lecho de polvo utiliza fuentes térmicas, como láseres o haces de electrones, para fusionar partículas de polvo capa a capa y crear piezas sólidas. Este método es bastante común en sectores como la industria aeroespacial y automotriz, gracias a la alta resistencia y precisión dimensional de las piezas metálicas que se producen. Materiales como el nailon, titanio y acero se utilizan con frecuencia. Entre sus ventajas, se destaca la capacidad de crear geometrías complejas sin necesidad de estructuras de soporte, así como la producción de piezas funcionales muy robustas. Sin embargo, su alto costo y el consumo intensivo de energía pueden limitar su uso en aplicaciones que no son industriales. *(Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025)*

La inyección de material, conocida como MJ o DOD, consiste en depositar pequeñas gotas de un material fotopolimérico que se cura capa por capa con luz. Este método permite mezclar diferentes materiales y colores en un solo objeto, lo que resulta en un alto grado de personalización. Se utiliza mucho en la creación de prototipos a todo color, modelos médicos y moldes de inyección de baja tirada. Aunque proporciona un acabado superficial

impresionante, las piezas resultantes tienden a ser frágiles y no son adecuadas para aplicaciones mecánicas. Además, los costos de los equipos y materiales suelen ser mayores comparados con otras tecnologías de impresión 3D. (*Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025*)

La inyección de aglutinante emplea un aglutinante líquido para unir selectivamente regiones de un lecho de polvo. Este proceso destaca por su bajo costo y capacidad para producir piezas metálicas funcionales y modelos a todo color. Es comúnmente usado en la fundición en arena y en sectores como la ingeniería y la arquitectura. Aunque la tecnología es económica, las propiedades mecánicas de las piezas no igualan a las producidas por fusión en lecho de polvo, y requiere un proceso de curado y eliminación del polvo no unido, lo que añade tiempo y complejidad al pos procesamiento. (*Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025*)

La deposición de energía directa (DED) combina material en forma de polvo o alambre con una fuente de calor, como un láser, para crear piezas capa por capa. Es utilizada principalmente para reparar componentes de alto valor en las industrias automotriz y aeroespacial. Una de sus ventajas es la capacidad de mezclar materiales y trabajar en tres dimensiones sin necesidad de estructuras de soporte. Sin embargo, el proceso genera desperdicio de material y requiere un pos procesamiento significativo debido a la baja calidad del acabado superficial. Su aplicación es costosa, pero es esencial para trabajos de alta precisión y personalización. (*Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025*)

La impresión 3D a microescala se centra en crear piezas increíblemente pequeñas, que van desde micrones hasta nanómetros. Este proceso utiliza principalmente tecnologías de fotopolimerización, como μ SLA y 2PP, lo que permite fabricar componentes con detalles asombrosos para aplicaciones en medicina y electrónica. Además, algunas impresoras de última generación pueden trabajar con metales como el oro y el cobre. Aunque este método resulta más económico que otras técnicas de microfabricación, los equipos y materiales necesarios suelen ser bastante costosos. Sus aplicaciones abarcan desde parches de microagujas hasta implantes y circuitos, destacándose por su habilidad para producir componentes fuertes y precisos en tamaños diminutos. (*Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP, 2025*)

La laminación de hojas apila y fusiona láminas delgadas de material, como papel, polímeros o metales, para formar un objeto tridimensional. Se utilizan tecnologías como el LOM y UC, que

emplean calor o ultrasonidos para unir las capas. Este método es rápido y económico, ideal para prototipos no funcionales y piezas estéticas. Sin embargo, su precisión es limitada y genera más desperdicio en comparación con otros métodos de impresión 3D. Las piezas producidas suelen requerir un posprocesamiento significativo, lo que puede incrementar los costos y el tiempo total de fabricación. (*Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP*, 2025)

2.1.4. Impresora 3D recomendable para el desarrollo.

Las impresoras 3D son esenciales para construir robots humanoides como el InMoov, ya que permiten crear piezas personalizadas con alta precisión. Es importante utilizar una impresora con un volumen de impresión mínimo de 30 x 30 x 30 cm para ensamblar las partes más grandes del robot. La impresora debe ser compatible con archivos STL y contar con un software de slicing para convertirlos en instrucciones de impresión. (*The Best 3D Printers for Beginners in 2024*, 2024). Para el caso de los principiantes, hacer uso de una impresora como la Bambu Lab P1P en la **Figura 1** es perfectamente útil, sin embargo, ya que existe variedad de impresoras 3D se puede optar por un modelo más avanzado como lo es la Original Prusa XL de la **Figura 2**.

Figura 1. Impresora Bambu Lab



Nota: Adaptado de "Bambu Lab P1P", por All3DP, 2024, <https://all3dp.com/1/best-3d-printer-for-beginners/>.

Figura 2. Original Prusa XL



Nota: Adaptado de "Original Prusa XL", por All3DP, 2024, https://all3dp.com/1/best-premium-3d-printers/#google_vignette.

2.1.5. Materiales para Impresión 3D

El ABS es uno de los materiales más recomendados para la impresión de robots humanoides debido a su resistencia mecánica y a la facilidad de postprocesado, como el lijado o pegado. Sin embargo, el PLA, al ser más fácil de usar y biodegradable, es una opción popular entre principiantes. Otros materiales como el PETG y el nylon ofrecen alta resistencia, pero requieren configuraciones más avanzadas. La selección depende del propósito del componente y de las condiciones operativas, como temperatura o esfuerzo mecánico. (*Best PLA 3D Printing Filaments of 2019*, 2019)

Figura 3. PLA Printer Filament



Nota: Adaptado de "MakerBot PLA Printer Filament", por 3D Insider, 2024, <https://3dinsider.com/pla-3d-printing-filaments/>.

2.1.6. Software de Diseño 3D

Los programas de diseño 3D son herramientas clave para crear objetos tridimensionales, permitiendo a diseñadores, ingenieros y aficionados dar vida a sus ideas a través de impresoras 3D. Estos softwares ofrecen una variedad de funciones, desde la creación de modelos simples hasta simulaciones complejas que ayudan a perfeccionar el diseño antes de imprimir. Por

ejemplo, Blender es muy conocido por sus potentes capacidades de modelado y animación, mientras que TinkerCAD brilla por su facilidad de uso para principiantes, gracias a su interfaz intuitiva y basada en navegador. (*Mejores 8 programas para impresoras 3D que puedes descargar gratis*, 2025)

La flexibilidad que ofrecen estos programas es una de sus grandes ventajas, ya que permiten personalizar los diseños según las necesidades específicas de cada usuario. Además, hay muchos softwares gratuitos, como FreeCAD, que eliminan las barreras económicas para su uso. Sin embargo, algunas herramientas más avanzadas, como Fusion 360, pueden resultar costosas o requerir una curva de aprendizaje bastante pronunciada. Otro desafío común es la compatibilidad de formatos, que puede complicar la integración entre diferentes programas y dispositivos de impresión. (*Los softwares de impresión 3D: la Guía completa - 3Dnatives*, 2025)

Entre los programas más recomendados para principiantes se encuentran TinkerCAD y Ultimaker Cura. TinkerCAD permite a usuarios sin experiencia iniciarse en el modelado 3D con bloques básicos y herramientas simples. Por otro lado, Cura se especializa en la preparación de archivos para impresión, proporcionando configuraciones predefinidas según el modelo de impresora. Estas opciones han sido diseñadas para maximizar la facilidad de uso y minimizar el tiempo de aprendizaje. (*Mejores 8 programas para impresoras 3D que puedes descargar gratis*, 2025)

El crecimiento de tecnologías avanzadas ha dado lugar a programas como Meshmixer y Simplify3D, que ofrecen funciones para analizar y reparar modelos, lo que ayuda a mejorar la calidad de las impresiones. Por ejemplo, Meshmixer es perfecto para corregir errores en diseños complejos, mientras que Simplify3D se encarga de optimizar las trayectorias de impresión, aumentando la eficiencia y reduciendo el consumo de material. Sin embargo, es importante tener un conocimiento técnico intermedio para sacar el máximo provecho de estas herramientas. (*Mejores 8 programas para impresoras 3D que puedes descargar gratis*, 2025)

Cuando se trata de elegir el software de diseño 3D, es importante considerar tanto las necesidades como la experiencia del usuario. Si eres principiante, podrías empezar con herramientas más sencillas como TinkerCAD. Por otro lado, los profesionales suelen preferir opciones más avanzadas como Blender o Fusion 360. También es fundamental tener en cuenta la compatibilidad con las impresoras y los costos que esto puede implicar. Al final del día, la amplia variedad de programas disponibles garantiza que haya algo para cada nivel de habilidad

y cada objetivo en el emocionante mundo de la impresión 3D. (*Mejores 8 programas para impresoras 3D que puedes descargar gratis*, 2025)

2.1.7. Componentes Electrónicos

Los robots humanoides dependen de varios componentes electrónicos clave que les permiten imitar los movimientos humanos y llevar a cabo tareas complejas de manera autónoma. Primero, los actuadores son esenciales para la movilidad del robot. Estos dispositivos transforman la energía eléctrica en movimiento mecánico, lo que permite que las extremidades del robot se muevan de forma similar a los músculos humanos. Entre los actuadores más comunes se encuentran los motores eléctricos, que controlan los movimientos en las articulaciones, y los servomotores, que ofrecen un control preciso sobre los ángulos y la velocidad de los movimientos. Elegir los actuadores adecuados es fundamental para asegurar que los movimientos del robot sean fluidos, precisos y eficientes, algo vital para tareas que requieren gran destreza, como la manipulación de objetos. (arturo, 2020)

Las herramientas son fundamentales en la robótica humanoide, y uno de los elementos más importantes es el sistema de sensores. Estos sensores permiten que el robot perciba su entorno y reaccione de manera adecuada. Por ejemplo, los sensores de proximidad, como los ultrasónicos o infrarrojos, ayudan a evitar obstáculos y facilitan la navegación del robot. Además, los sensores de fuerza son cruciales para que el robot ajuste la presión al manipular objetos delicados. También están los sensores de posición, que monitorean la ubicación y el movimiento de las extremidades, lo que resulta esencial para mantener el equilibrio y coordinar los movimientos en tiempo real. Sin estos sensores, un robot humanoide no podría interactuar de manera efectiva con su entorno, lo que los convierte en una parte vital de su funcionamiento. (eduardo.ceo, 2024)

El sistema de alimentación de un robot humanoide juega un papel crucial al proporcionar la energía necesaria para que todos sus componentes electrónicos funcionen de manera óptima. Generalmente, estos robots utilizan baterías recargables, como las de litio, que ofrecen una excelente combinación de capacidad y peso. La gestión de la energía es fundamental, ya que debe ser optimizada para que el robot pueda operar durante largos períodos sin sobrecalentarse ni consumir más energía de la que realmente necesita. Los sistemas de gestión de energía son los encargados de regular el flujo eléctrico hacia los diferentes componentes, garantizando una distribución eficiente y evitando sobrecargas. Además, el sistema de alimentación debe contar con circuitos de protección para prevenir daños en caso de fluctuaciones de voltaje o

cortocircuitos. (*¿Cuáles son los componentes principales de los robots humanoides (caja de cambios de motor) ?*, 2023)

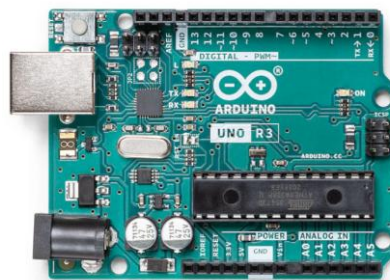
Finalmente, los sistemas de comunicación son fundamentales para que los robots humanoides interactúen con su entorno y con las personas. Gracias a estos sistemas, el robot puede recibir instrucciones y enviar información sobre su estado o lo que le rodea. La comunicación inalámbrica suele hacerse a través de Wi-Fi o Bluetooth, mientras que para conectar los diferentes sensores y actuadores del robot se utilizan protocolos de comunicación en serie como UART o I2C. Además, la interfaz de usuario (UI) juega un papel crucial, ya que permite a los operadores controlar el robot, ya sea mediante una pantalla táctil, un teclado o dispositivos de control remoto. La integración de todos estos componentes electrónicos no solo facilita un control más eficiente, sino que también hace que la interacción con el robot sea más intuitiva, mejorando así su funcionalidad y facilidad de uso. (*¿Cuáles son los componentes principales de los robots humanoides (caja de cambios de motor) ?*, 2023)

2.1.8. Controladores y Actuadores

2.1.8.1 Microcontroladores

El Arduino Uno de la **Figura 4** y Mega de la **Figura 5** son ideales para controlar robots como el InMoov. Estos microcontroladores destacan por su facilidad de programación y compatibilidad con una amplia variedad de sensores y actuadores. El Mega, en particular, es preferible para proyectos más grandes debido a su mayor número de pines y capacidad de memoria. Además, su integración con herramientas como MyRobotLab facilita la implementación de scripts en Python para la gestión del robot. (*Arduino Hardware*, 2025)

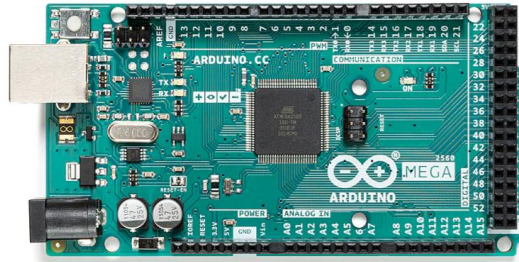
Figura 4. Arduino Uno



Nota: Adaptado de "Arduino Uno Rev3", por Arduino Store, 2024,

<https://store.arduino.cc/products/arduino-uno-rev3>.

Figura 5. Arduino Mega



Nota: Adaptado de "Arduino Mega 2560 Rev3", por Arduino Store, 2024,
<https://store.arduino.cc/products/arduino-mega-2560-rev3>.

2.1.8.2 Actuadores

Los servomotores, como el HK15298B de la **Figura 6** y el Hitec HS805BB de la **Figura 7**, son críticos para dotar a los robots de movimiento preciso. Estos dispositivos convierten señales eléctricas en desplazamientos angulares, lo que permite articular extremidades y realizar movimientos fluidos. El MG996, de la **Figura 8**, es adecuado para movimientos ligeros, mientras que el HS805BB, con mayor torque, maneja cargas pesadas. La elección de actuadores depende del peso y tipo de movimiento requerido. (Kamami.Pl - Electronics Store, 2025)

Figura 6. HK15298B



Nota: Adaptado de "HK15298B High Voltage Coreless Digital MG/BB Servo 66g / 20kg / 0.16s", por Kamami, 2024, <https://kamami.pl/en/retired-products/218773-phk-hk15298b-high-voltage-coreless-digital-mgbb-servo-66g--20kg--016s-16272p.html>.

Figura 7. Hitec HS-805BB



Nota: Adaptado de "Hitec HS-805BB - 180° giant servo", por Kamami, 2024, <https://kamami.pl/en/servos-giant/1183632-rob-11881-servo-hitec-hs-805bb-giant-scale.html>.

Figura 8. Servo MG996R



Nota: Adaptado de "Servo MG996R", por Kamami, 2024, <https://kamami.pl/en/servos-giant/204180-servo-mg996r-5906623440627.html>.

2.1.9. Programación y Algoritmos de Control

La programación en robótica humanoide es un proceso complejo que involucra lenguajes de programación específicos y algoritmos diseñados para controlar movimientos, decisiones y respuestas de los robots. Estos algoritmos están basados en modelos matemáticos como la cinemática y la dinámica, que permiten calcular trayectorias y posiciones precisas para las articulaciones. Lenguajes como Python, C++ y MATLAB son comúnmente utilizados, debido a su capacidad para manejar cálculos avanzados y su compatibilidad con bibliotecas especializadas en robótica. Estos lenguajes también son ideales para programar sistemas de control que aseguran movimientos suaves y precisos, integrando estrategias como el control PID (Proporcional-Integral-Derivativo) para ajustar la respuesta de los motores según los requerimientos del entorno. (*El ABC de la Programación de un Robot Humanoide*, 2025)

Además de los movimientos básicos, los algoritmos de control deben permitir que los robots humanoides interactúen de manera autónoma con su entorno. Esto se logra integrando sensores como cámaras, micrófonos y detectores de fuerza, que ofrecen datos en tiempo real para ayudar en la toma de decisiones. Los algoritmos de percepción analizan esta información para identificar obstáculos, personas y objetos, mientras que los de planificación trazan las rutas más efectivas para alcanzar objetivos específicos. Este nivel de autonomía se fundamenta en técnicas avanzadas de inteligencia artificial, como redes neuronales y aprendizaje por refuerzo, que permiten a los robots mejorar su rendimiento a medida que ganan experiencia en diferentes situaciones. (*Controladores de Robot - Documentación RoboDK*, 2025)

La interacción con humanos es un aspecto fundamental en la programación de robots humanoides. Los algoritmos deben permitir que los robots reconozcan gestos, expresiones faciales y comandos de voz, lo que facilita una comunicación efectiva. Esto se logra al combinar técnicas de procesamiento de señales con modelos de aprendizaje supervisado, que enseñan al robot a interpretar y responder de manera adecuada. Este tipo de interacción es crucial en aplicaciones como la asistencia personal y la educación, donde los robots se convierten en compañeros colaborativos. (*El ABC de la Programación de un Robot Humanoide*, 2025)

Por último, los sistemas de comunicación y las interfaces de usuario juegan un papel fundamental en la programación y el control de robots humanoides. Estos robots necesitan contar con protocolos de comunicación que les permitan intercambiar datos con otros dispositivos y sistemas, ya sea a través de redes locales o de la nube. Esto no solo facilita la coordinación entre varios robots, sino que también permite su integración en entornos inteligentes. Además, las interfaces de usuario, como los paneles de control o las aplicaciones móviles, son esenciales para configurar y monitorear el rendimiento del robot, lo que permite realizar ajustes en tiempo real y mejora la experiencia del usuario. (Guanajuato, 2023)

2.1.10. Software de Programación

El software MyRobotLab y los scripts en Python son herramientas fundamentales para programar robots humanoides. Estas plataformas permiten manejar actuadores, sensores y realizar tareas complejas, como el reconocimiento facial o el control autónomo. MyRobotLab incorpora bibliotecas de inteligencia artificial, mientras que Python hace que sea más fácil diseñar algoritmos personalizados para tareas específicas. (*Home | MyRobotLab*, 2025)

CAPÍTULO III

MARCO METODOLÓGICO

3.1. Fase de diseño

3.1.1. Etapa1: Análisis sobre robots humanoides.

En la primera etapa de la fase de diseño, que se centró en el análisis de robots humanoides, se llevó a cabo un estudio exploratorio para investigar y entender las características clave de los robots humanoides que ya existen y sus aplicaciones. Este tipo de investigación fue fundamental para obtener un conocimiento inicial sobre el estado actual de la tecnología en robótica humanoide, analizando los avances, limitaciones y desafíos en el diseño de robots que imitan la biomecánica humana. Se empleó el método lógico-deductivo para establecer los principios de diseño y construcción en robótica y biomecánica, utilizando diversos artículos y tesis relacionadas con el tema.

3.1.2. Etapa 2: Diseño de piezas 3D.

Durante el proceso de diseño de las piezas en 3D, se realizó un estudio detallado para definir con precisión las dimensiones, geometrías y características de cada componente, asegurando que todo encajara perfectamente en el diseño general. También se consideraron aspectos como la relación entre las piezas, sus ángulos de movimiento y las tolerancias necesarias para lograr una integración impecable con otras partes del robot. Utilizando un enfoque de investigación cualitativa, se analizó cómo el diseño de cada pieza influía en la biomecánica del robot y su capacidad para imitar los movimientos humanos. Esto incluyó una evaluación de la interacción entre las piezas, garantizando que cada componente cumpliera con las especificaciones requeridas para mantener la cohesión estructural y permitir los movimientos deseados.

3.1.3. Etapa 3: Diseño de esquema eléctrico y electrónico

En el proceso de diseñar los esquemas del robot humanoide, se realizó un estudio descriptivo que detalló los componentes electrónicos y las conexiones necesarias para asegurar que el sistema funcionara de manera eficiente. Este tipo de análisis fue clave para entender cómo se relacionan los actuadores, controladores y fuentes de alimentación, garantizando que el diseño

cumpliera con los requisitos técnicos y de seguridad. El enfoque de investigación cualitativa permitió examinar a fondo la funcionalidad e integración de cada componente, evaluando aspectos como la compatibilidad, la disposición de los elementos y la distribución de la energía. Este método ayudó a identificar posibles mejoras antes de llevar a cabo la implementación.

3.2. Fase de implementación

3.2.1. Etapa 1: Parametrización de archivos stl.

En relación con la parametrización de archivos STL, se llevó a cabo un estudio explicativo con el fin de comprender y optimizar cómo los parámetros ajustables en programas de impresión 3D, como Cura, impactan en la eficiencia y calidad de la fabricación de las piezas del robot. Este enfoque fue clave para ajustar aspectos como el tiempo de impresión, la cantidad de filamento requerido y las configuraciones de relleno, garantizando que las piezas se produjeran de manera precisa y funcional, sin desperdiciar recursos. Se utilizó el método de investigación hipotético-deductivo, formulando hipótesis sobre cómo modificar parámetros como la densidad de relleno, la velocidad de impresión y el grosor de capa para mejorar la precisión dimensional, la resistencia estructural o la velocidad de fabricación. A partir de estas hipótesis, se llevaron a cabo experimentos con el software Cura, ajustando las configuraciones de impresión y evaluando cómo cada ajuste afectaba el rendimiento de las piezas en el ensamblaje del robot.

3.2.2. Etapa 2: Impresión de las piezas 3D.

Enfocado en la impresión de piezas en 3D, se realizó un estudio exploratorio para identificar y abordar los desafíos que surgen en la fabricación de estos componentes. Esto permitió experimentar con diversas configuraciones de impresión, materiales y parámetros, asegurando que las piezas cumplieran con los estándares de calidad y funcionalidad necesarios para el robot humanoide. El enfoque de investigación, basado en la experimentación científica, fue clave, ya que incluyó pruebas prácticas con impresoras 3D para crear las piezas. Durante este proceso, se evaluaron aspectos como la precisión, la adherencia entre capas, la resistencia estructural y la calidad de la superficie. Estos experimentos facilitaron el ajuste de parámetros de impresión, como la temperatura, la velocidad, la densidad de relleno y la orientación de las piezas.

3.2.3. Etapa 3: Ensamble del robot humanoide.

En el proceso de ensamblaje del robot humanoide, se llevó a cabo un estudio descriptivo que documentó minuciosamente cómo se ensamblaban las piezas impresas y se integraban los

componentes electrónicos y mecánicos. Este enfoque ayudó a identificar posibles dificultades y a encontrar soluciones durante el montaje, asegurando que el diseño se convirtiera en un modelo físico que funcionara correctamente. La experimentación científica fue fundamental, ya que incluyó pruebas prácticas para ensamblar y ajustar las piezas. Durante este proceso, se revisó la compatibilidad entre los componentes, la alineación de las articulaciones y la estabilidad estructural del robot. Además, se hicieron los ajustes necesarios para solucionar problemas como holguras, falta de precisión o dificultades en el ensamblaje.

3.2.4. Etapa 4: Montaje de elementos electrónicos.

Para el montaje de elementos electrónicos, se realizó un estudio exploratorio que permitió probar diversas configuraciones y evaluar cómo funcionaban los componentes en condiciones reales, asegurando así que el sistema eléctrico operara correctamente. La experimentación científica fue clave en esta fase, ya que incluyó la instalación, conexión y prueba de elementos como actuadores, controladores y sistemas de alimentación. Durante este proceso, se revisaron aspectos como la polaridad correcta, la continuidad de las conexiones y la eficiencia en la distribución de energía.

3.2.5. Etapa 5: Programación de movimientos y reacciones.

En esta fase, que se centra en la programación de movimientos y reacciones, se llevó a cabo un estudio explicativo para entender cómo los algoritmos de control impactaron el rendimiento del robot humanoide en términos de motricidad y respuesta. Este tipo de investigación fue fundamental para analizar las relaciones de causa y efecto entre los comandos programados y las acciones físicas del robot, lo que permitió optimizar su funcionalidad. El enfoque de investigación hipotético-deductivo guió esta etapa, formulando hipótesis sobre cómo distintos patrones de programación podrían mejorar movimientos específicos o reacciones a estímulos. Estas hipótesis se pusieron a prueba a través de experimentos en los que el robot ejecutó secuencias de movimientos diseñadas para imitar acciones humanas, como agarrar objetos, mover extremidades o reaccionar a cambios en el entorno y la respuesta de los componentes bajo diversas condiciones de operación.

3.3. Fase de evaluación

3.3.1. Etapa 1: Prueba de programación de movimiento sincronizados.

En la prueba de programación de movimientos sincronizados, se llevó a cabo un estudio exploratorio para investigar cómo el robot humanoide realizaba movimientos coordinados en sus extremidades superiores. Este estudio resultó ser muy útil para detectar posibles fallos en la sincronización de los motores y en la precisión de la ejecución de los movimientos, lo cual fue fundamental para lograr una motricidad fluida y eficiente. El enfoque de investigación cualitativa se centró en un análisis detallado de las interacciones entre las distintas partes del robot mientras ejecutaba los movimientos sincronizados. Esto incluyó observar la fluidez de los movimientos, la alineación de las extremidades y la capacidad del robot para llevar a cabo tareas complejas, como levantar o manipular objetos.

3.3.2. Etapa 2: Demostración práctica de biomecánica y movimientos robóticos.

Para demostrar de manera práctica la biomecánica y los movimientos robóticos, se llevó a cabo un estudio explicativo que buscaba entender cómo un robot humanoide podía replicar las acciones humanas desde la perspectiva de la biomecánica. Este estudio fue fundamental para analizar cómo las articulaciones y estructuras del robot respondían a los comandos programados, logrando simular el movimiento humano de manera eficiente y precisa. Utilizando un enfoque de investigación deductivo, se partió de teorías previamente establecidas sobre cómo deberían ejecutarse los movimientos humanos, como la flexión de las articulaciones y la coordinación entre ellas. Con base en esta teoría, se realizaron experimentos para verificar si el robot podía imitar estos movimientos, deduciendo qué ajustes eran necesarios para mejorar su rendimiento.

3.3.3. Etapa 3: Programación de tareas prácticas para explorar sistemas electrónicos del robot.

En la planificación de tareas prácticas para explorar los sistemas electrónicos del robot, se llevó a cabo un estudio preliminar para entender cómo estos sistemas interactuaron durante la realización de las tareas. Esto ayudó a mejorar la funcionalidad del robot en un entorno educativo, lo cual es esencial para ofrecer a los estudiantes una experiencia práctica enriquecedora. Se utilizó un enfoque de investigación deductivo para poner a prueba teorías sobre el funcionamiento de los sistemas electrónicos del robot, aplicando tareas prácticas en áreas clave como electrónica, robótica, programación, microcontroladores, diseño asistido por computadora, instrumentación y sensores. Estas actividades permitieron evaluar cómo los componentes respondían a las señales de control y cómo se podía interactuar con el robot para entender los principios detrás de cada tecnología.

CAPÍTULO IV

DESARROLLO DEL PROYECTO TÉCNICO

4.1. Diseño de la propuesta

4.1.1. Análisis sobre robots humanoides

Los robots humanoides han cobrado una gran importancia en la robótica moderna gracias a su habilidad para replicar movimientos y comportamientos humanos. Este avance ha sido impulsado por la necesidad de mejorar la interacción entre personas y máquinas, lo que ha abierto la puerta a aplicaciones en educación, asistencia y manufactura. Un claro ejemplo de esto es el robot NAO, diseñado para ser accesible y versátil, lo que permite a investigadores y estudiantes experimentar con su programación y control. Entre las plataformas que se utilizan en su desarrollo se encuentran Choregraphe, CoppeliaSim y Matlab-Simulink, que facilitan la simulación y ejecución de algoritmos de control avanzados. (Enríquez Salinas, 2022)

El diseño estructural de un robot humanoide requiere tener en cuenta materiales que sean ligeros pero a la vez resistentes, como los polímeros y las aleaciones de aluminio. Para proyectos con un presupuesto ajustado, la impresión 3D se presenta como una opción muy viable, ya que permite crear piezas personalizadas y reducir costos. Además, los grados de libertad en las articulaciones son cruciales, ya que determinan la flexibilidad y el rango de movimiento del robot, lo cual es fundamental en la biomecánica robótica. Una distribución adecuada de sensores y actuadores facilita una interacción efectiva con el entorno, lo que mejora tanto la estabilidad como la precisión en los movimientos. (Domínguez López, 2019)

El control del movimiento en robots humanoides se fundamenta en modelos matemáticos que permiten predecir y ajustar su posición en tiempo real. Una de las estrategias más comunes es el control basado en Lyapunov, que asegura la estabilidad al reducir progresivamente el error en la trayectoria. Este enfoque es especialmente valioso para robots como el NAO, que necesitan ajustes constantes en su orientación y desplazamiento. La interacción entre simuladores y lenguajes de programación, como Python y C++, facilita la implementación de estos algoritmos en plataformas virtuales antes de probarlos en hardware real. (Enríquez Salinas, 2022)

Los sistemas de control en robots humanoides se pueden dividir en dos categorías: abiertos y cerrados. En los sistemas de lazo abierto, las órdenes se llevan a cabo sin recibir retroalimentación, lo que puede dar lugar a imprecisiones. Por otro lado, los sistemas de lazo cerrado utilizan sensores que ayudan a corregir desviaciones y a mejorar la precisión del movimiento. Además, el uso de algoritmos avanzados, como el control predictivo o el control adaptativo, puede optimizar la respuesta del robot ante cambios inesperados en su entorno. (Sánchez-Girón Coca, 2022)

El movimiento de un robot humanoide se fundamenta en la perfecta sincronización de motores eléctricos y servomecanismos. Los motores de corriente continua sin escobillas y los servomotores digitales son elecciones bastante comunes en el diseño de estos sistemas. Al implementar modelos dinámicos, se puede anticipar cómo reaccionará el robot ante diferentes tipos de terreno y obstáculos. Además, el uso de actuadores con retroalimentación de posición y torque contribuye a mejorar la estabilidad del robot, especialmente en movimientos complejos como la marcha bípeda. (Valdiviezo Romero & Valladares Yanqui, 2023)

El software juega un papel clave en la funcionalidad de los robots humanoides. Sistemas operativos como ROS (Robot Operating System) proporcionan una infraestructura robusta para la comunicación entre módulos de hardware y software. ROS permite la integración de bibliotecas especializadas para visión artificial, planificación de movimientos y aprendizaje automático. Además, su compatibilidad con simuladores como Gazebo y CoppeliaSim facilita la validación de algoritmos en entornos virtuales antes de su implementación física. (Domínguez López, 2019)

Los lenguajes de programación que se utilizan en robótica pueden variar bastante, dependiendo de cuán abstracto necesites que sea el enfoque. Python es muy popular, gracias a su sintaxis sencilla y a la gran cantidad de bibliotecas especializadas que ofrece, como OpenCV y TensorFlow. Por otro lado, C++ es la opción preferida cuando se trata de aplicaciones que demandan un alto rendimiento y un control en tiempo real. Matlab y Simulink, en cambio, son herramientas fantásticas para diseñar y simular sistemas de control a través de modelos gráficos, lo que resulta muy útil durante la fase de desarrollo de algoritmos. (Sánchez-Girón Coca, 2022)

Para mejorar el rendimiento de un robot humanoide, se pueden añadir sistemas de percepción más avanzados. Sensores de profundidad, como los que se encuentran en las cámaras RGB-D,

permiten crear mapas tridimensionales del entorno, lo que mejora la navegación autónoma. Además, los algoritmos de aprendizaje profundo pueden optimizar el reconocimiento de patrones, lo que permite que el robot interactúe de una manera más natural con su entorno. Por otro lado, la implementación de redes neuronales recurrentes ayuda a predecir trayectorias y a tomar decisiones en tiempo real. (Valdiviezo Romero & Valladares Yanqui, 2023)

La optimización del consumo energético es fundamental en el diseño de robots humanoides. Utilizar baterías de alta densidad energética, como las de ion-litio, permite aumentar la autonomía sin añadir peso extra al robot. Además, aplicar estrategias de control de potencia, como la modulación por ancho de pulso (PWM), puede mejorar la eficiencia de los motores. También, implementar mecanismos de recuperación de energía, como la frenada regenerativa, puede prolongar la duración de la batería en robots que consumen mucha energía. (Enríquez Salinas, 2022)

El desarrollo de interfaces de usuario intuitivas realmente mejora la forma en que interactuamos con los robots humanoides. Las aplicaciones de realidad aumentada pueden hacer que la programación y el monitoreo de estos robots en tiempo real sea mucho más sencillo. Además, el uso de asistentes de voz nos permite controlar funciones básicas solo con comandos hablados, lo que reduce la necesidad de tener interfaces físicas. Y al integrar estas tecnologías con plataformas de IoT, podemos gestionar los robots de manera remota, abriendo un mundo de nuevas posibilidades en automatización y asistencia personalizada. (Domínguez López, 2019)

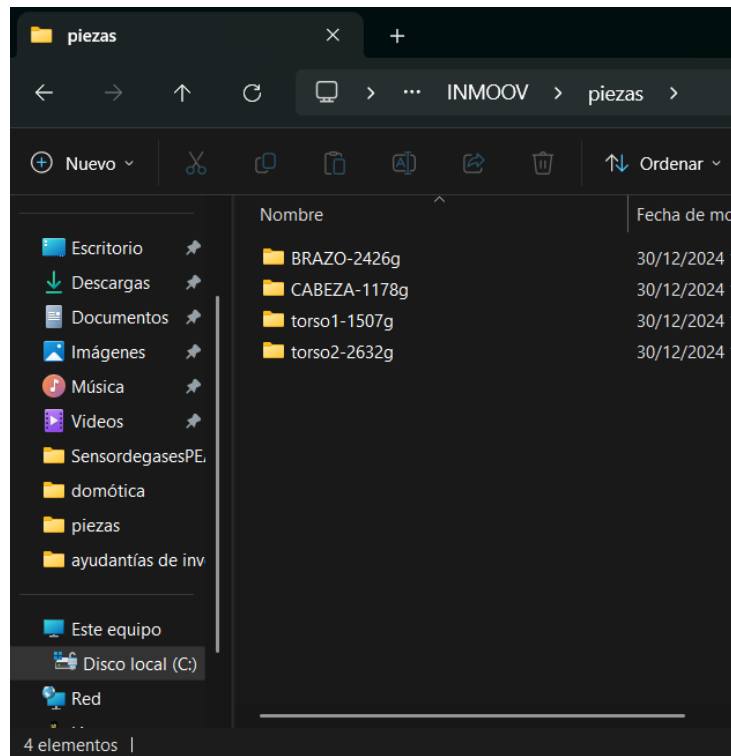
Finalmente, la evolución de los robots humanoides depende de la sinergia entre hardware y software. La combinación de algoritmos avanzados de control con materiales innovadores puede mejorar la eficiencia y versatilidad de estos dispositivos. La robótica colaborativa, donde los robots trabajan junto a humanos en entornos dinámicos, es un campo en crecimiento. Integrar inteligencia artificial en estos sistemas permitirá que los robots no solo ejecuten tareas programadas, sino que también aprendan y se adapten a nuevas situaciones, acercándonos cada vez más a una interacción natural entre humanos y máquinas. (Sánchez-Girón Coca, 2022)

4.1.2. Diseño de piezas 3D

Al obtener las piezas 3D del diseño del robot humanoide, primero se necesita clasificar en carpetas correctamente distribuidas de acuerdo a las secciones de la construcción, según la

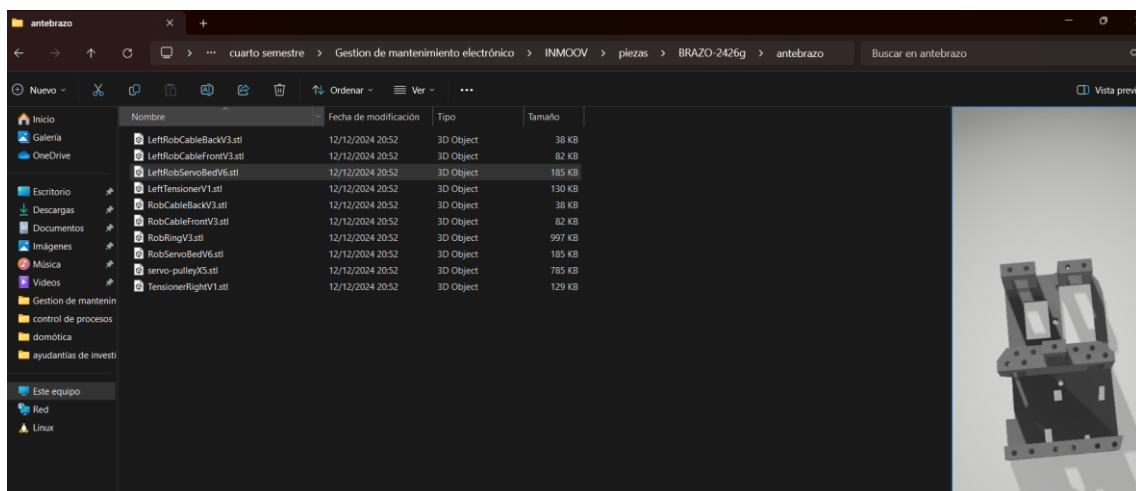
Figura 9, donde se dupliquen o aíslen, como el ejemplo en la Figura 10 y Figura 11 (según sea necesario) las partes que van a ser necesarias para ensamblarlo. Esto evitará gastos innecesarios y confusiones al momento de la construcción.

Figura 9. Clasificación de piezas



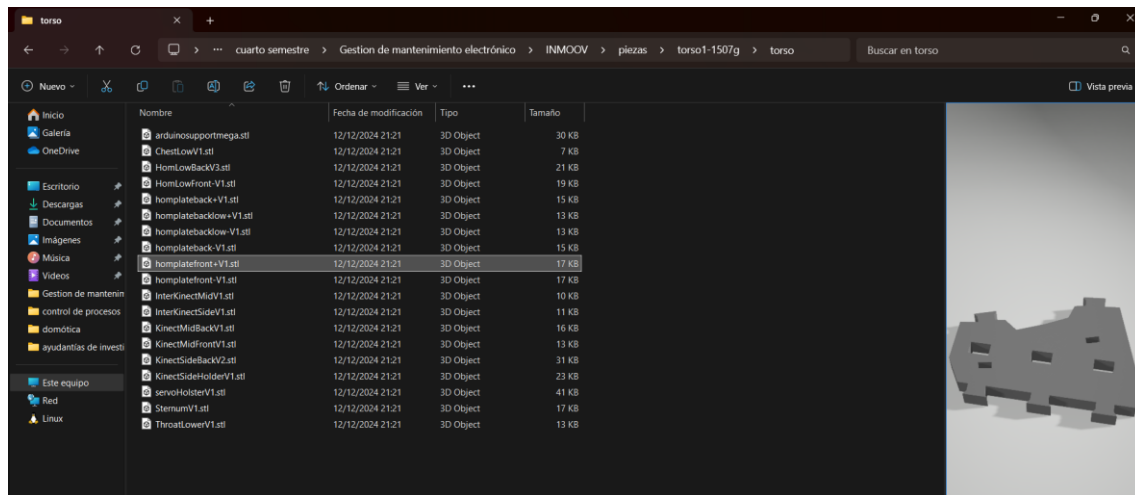
Nota: Elaboración propia.

Figura 10. Piezas específicas.



Nota: Elaboración propia.

Figura 11. Piezas específicas.

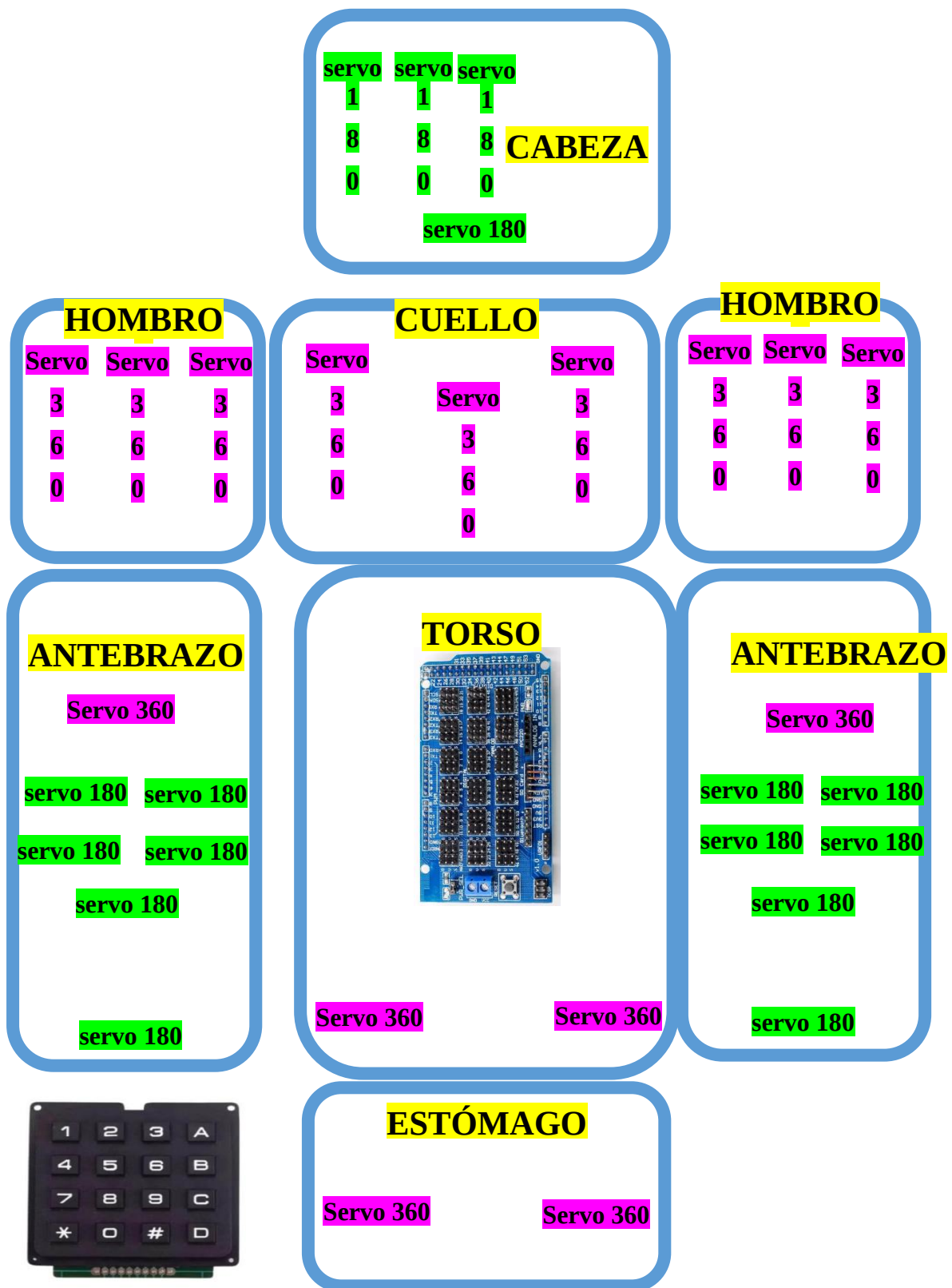


Nota: Elaboración propia.

4.1.3. Diseño de esquema eléctrico y electrónico

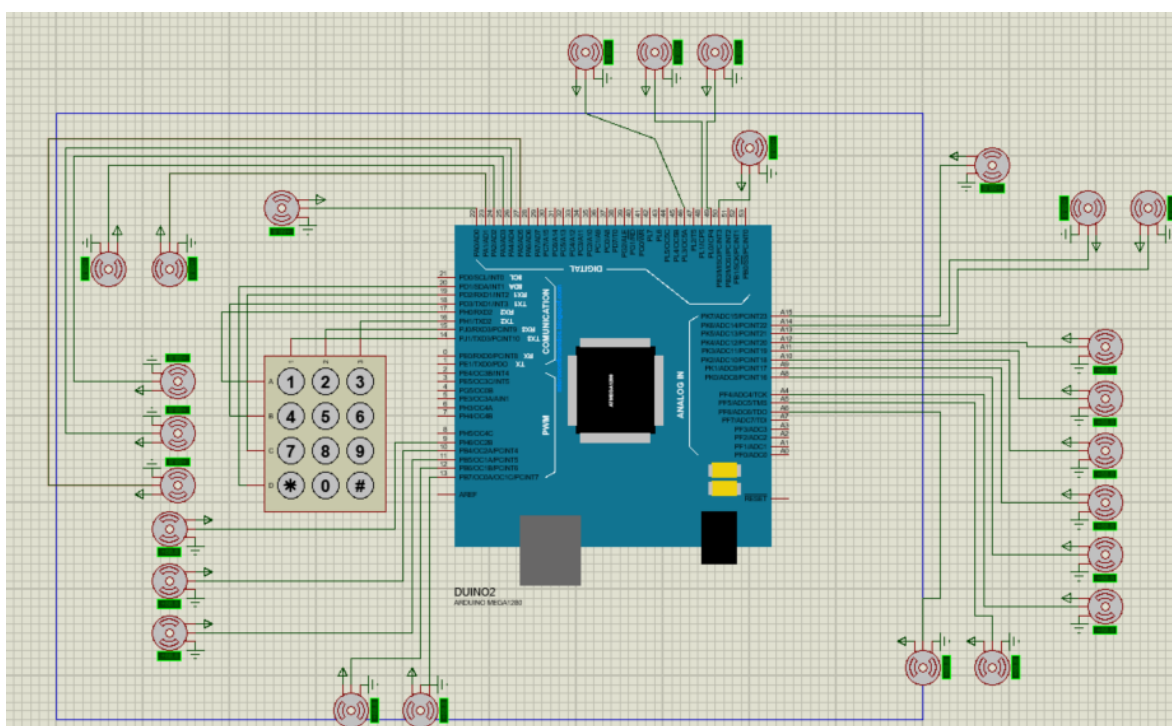
Se propone la distribución correspondiente a los servos, estos divididos: con movilidad de 180 grados y movilidad de 360 grados, tal como se visualiza en la Figura 12. Así también se plantea la conexión entre el Arduino MEGA con su correspondiente shield de sensores HW-857 V2, el teclado matricial de 4 x 4 y los diferentes servos, en un diagrama esquemático que permita visualizar la distribución de los pines correspondientes en la Figura 13.

Figura 12. Distribución de servos.



Nota: Elaboración propia. Servos en la estructura del robot humanoide y su característica de control.

Figura 13. Conexión electrónica de servos.



Nota: Elaboración propia.

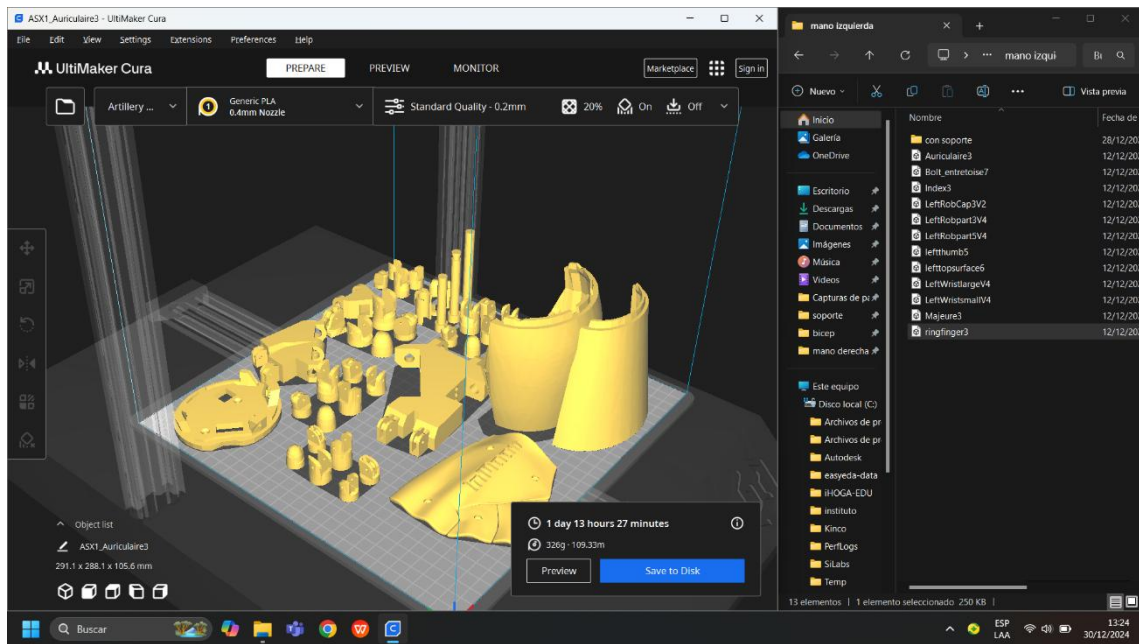
4.2. Implementación

4.2.1. Parametrización de archivos stl.

Se usa un programa para impresión 3D, en el caso de la Figura 14 y Figura 15 es el programa

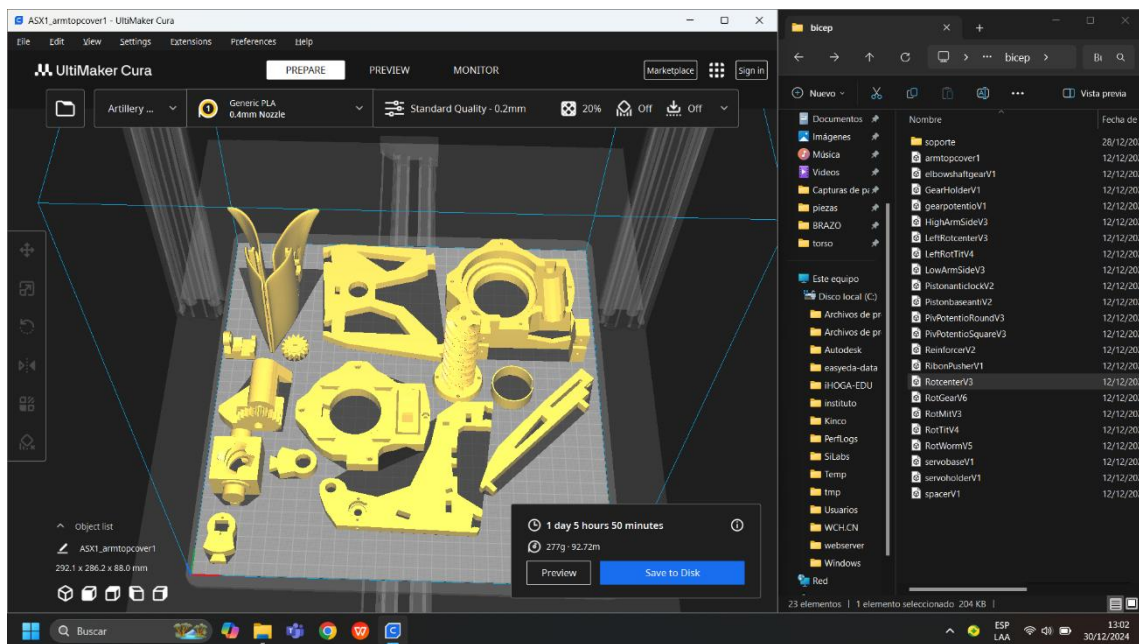
Cura, con el que podemos estimar el tiempo de impresión, la cantidad de material para la impresión y los valores necesarios para la impresión, como: relleno (entre 30% y 40% para mayor robustez) , material (PLA+, sin embargo un material con mejores características es el filamento ABS), velocidad, tipo de impresora y la posición adecuada en la que deberá ser impresa una pieza para darlos la mayor resistencia mecánica (en este último aspecto se refiere a la dirección en la que se imprimen las capas, pues, estas piezas pueden ser más resistentes si las capas están orientadas en contra de la fuerza a la que serán sometidas).

Figura 14. Stl. Mano izquierda.



Nota: Elaboración propia.

Figura 15. Std. Biceps.



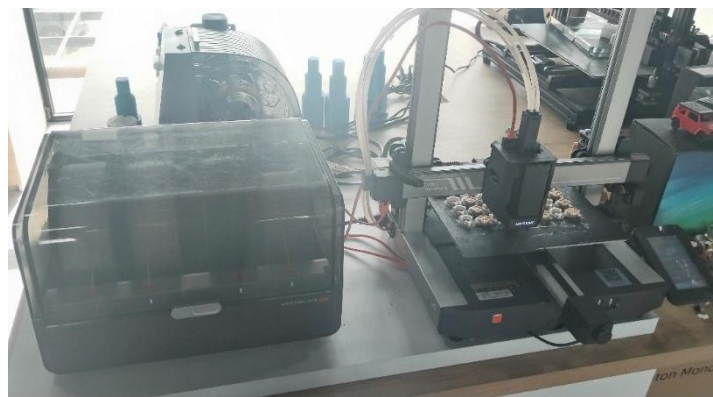
Nota: Elaboración propia.

4.2.2. Impresión de piezas 3D

Después de establecer las consideraciones necesarias para la impresión de cada pieza del robot humanoide, se procede a transferir los archivos a las impresoras 3D. Figura 16 y Figura 17.

Crear una carpeta con todos las piezas que va a necesitar el robot humanoide, tomando en cuenta los diagramas compartidos en: Ensamble del robot humanoide, para saber que piezas necesitan ser duplicadas para izquierda y derecha, así se podrá optimizar la impresión de las piezas y agilizar la construcción.

Figura 16. Proceso de impresión 1.



Nota: Elaboración propia.

Figura 17. Proceso de impresión 2.



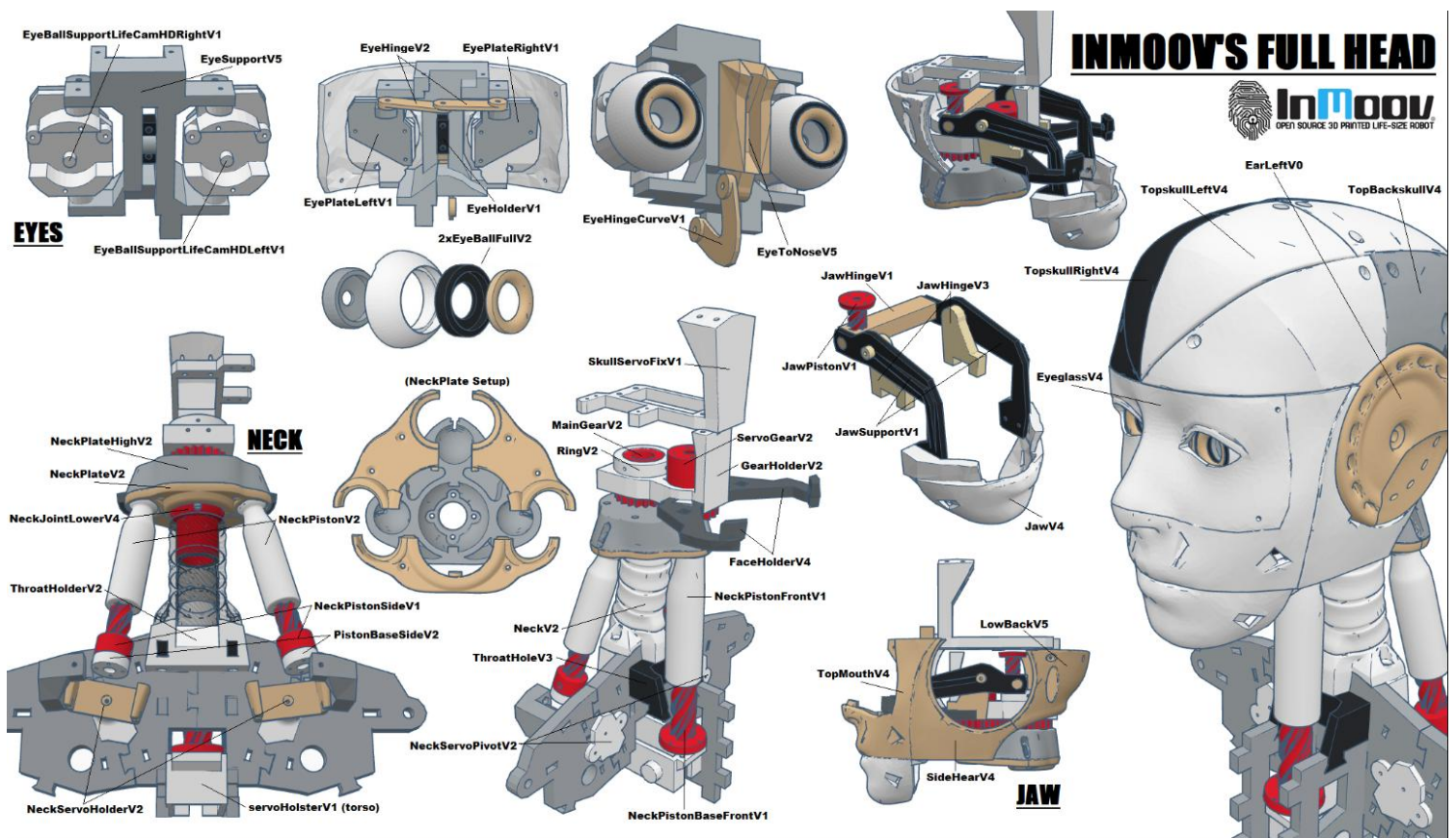
Nota: Elaboración propia.

4.2.3. Ensamble del robot humanoide

Conseguir de antemano: pegamento super glue o brujita, grasa de silicona o grasa multiuso de color neutro o amarillo y tornillos de cabeza avellanada: M3 40 de 10mm, 15mm, 20mm cada una con sus propias tuercas; 20 de 40mm con sus propias tuercas; 40 M4 de 20mm; varilla roscada M8 con 50 tuercas y 50 rodela.

Tomando en consideración la Figura 18, comprobamos tener todas las piezas necesarias para la construcción de la cabeza, cuello y mecanismo de ojos y mandíbula, como se muestra en la Figura 19. También, al comprobar las piezas, podemos saber los tornillos que hemos de necesitar: 20 tornillos M3 de 10mm, 15mm, 20mm con sus tuercas y 10 tornillos M4 de 20mm.

Figura 18. Ensamble cabeza.



Nota: Fuente: Shido Xen. Adaptado de "InMoov - Google Groups", por Google Groups, 2024, https://groups.google.com/g/inmoov/c/OTdm_bNy0CA.

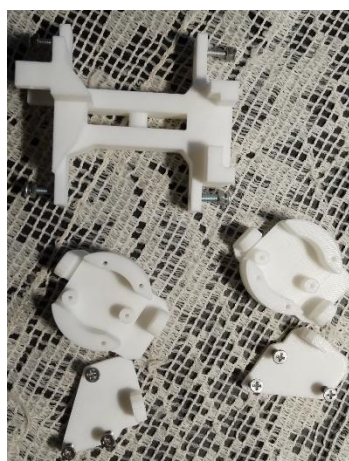
Figura 19. Piezas cabeza



Nota: Elaboración propia.

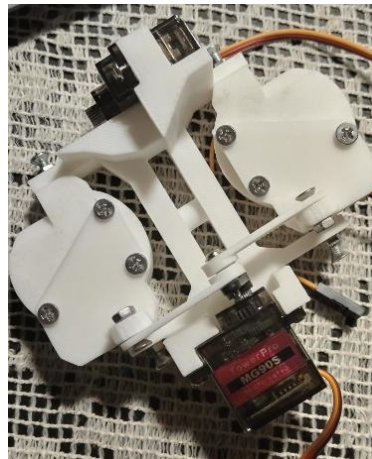
En las Figuras 20 y Figura 21 construimos el mecanismo para los ojos, teniendo cuidado en que la movilidad de las piezas de los ojos sea libre. Finalmente construimos la el mecanismo completo de ojos en la Figura 22.

Figura 20. Estructura ojos.



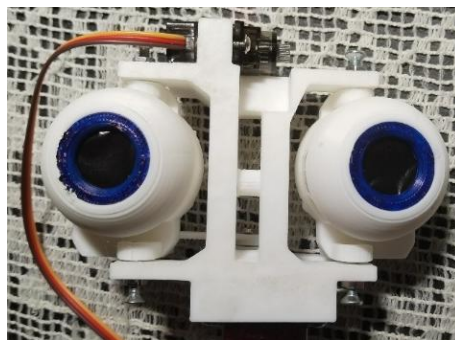
Nota: Elaboración propia.

Figura 21. Servos estructura ojos.



Nota: Elaboración propia.

Figura 22. Estructura ojos, final.



Nota: Elaboración propia.

La forma más eficiente de contruir el craneo es empezando desde la estructura base del cuello, como en la Figura 23 y a partir de ahí ensamblar las demás piezas, como se ve en la Figura 24, así no se tendrá problemas para incertar las piezas en su lugar, tal como se ve en la Figura 25. El resultado final del ensable de la cabeza deberia sostenerse por si mismo y encajar de manera ideal tal como se ve el resultad de la Figura 26. Hay que recordar usar las tuercas para cada tornillo, es recomendable fijarlas con una gota de pegamento líquido ya que cada agujero las ajusta perfectamente, no habrá ningún problema.

Figura 23. Estructura cuello.



Nota: Elaboración propia.

Figura 24. Estructura mandíbula.



Nota: Elaboración propia.

Figura 25. Ensamblaje cabeza posterior.



Nota: Elaboración propia.

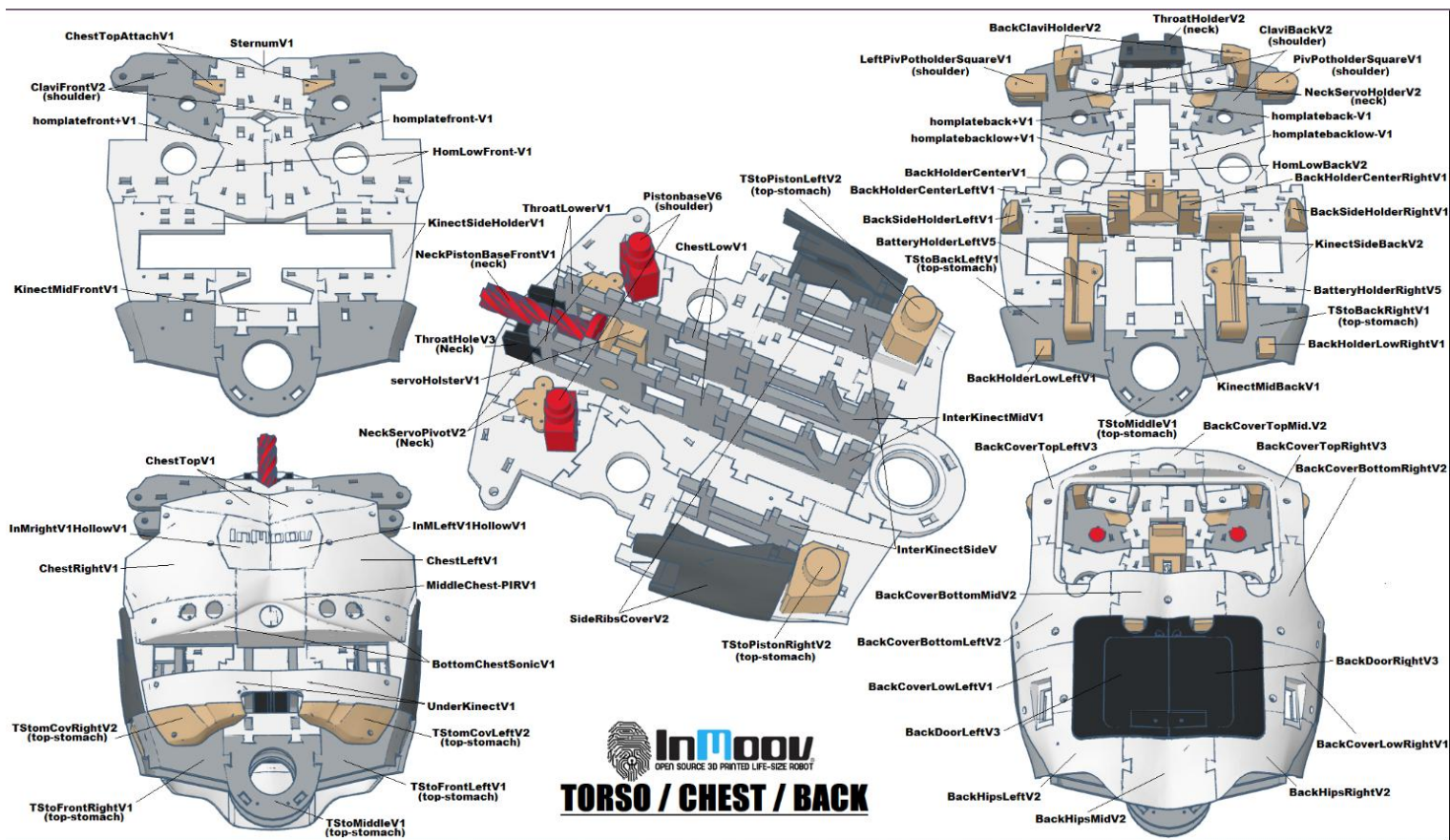
Figura 26. Ensamblaje cabeza final.



Nota: Elaboración propia.

El diagrama de la Figura 27 nos indicará las piezas necesarias para ensamblar el torso, pecho y espalda de nuestro robot humanoide.

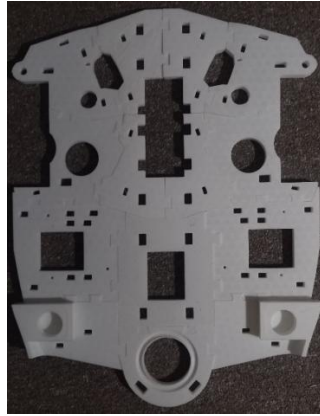
Figura 27. Ensamblaje torso.



Nota: Fuente: Shido Xen. Adaptado de "InMoov - Google Groups", por Google Groups, 2024, https://groups.google.com/g/inmoov/c/OTdm_bNy0CA.

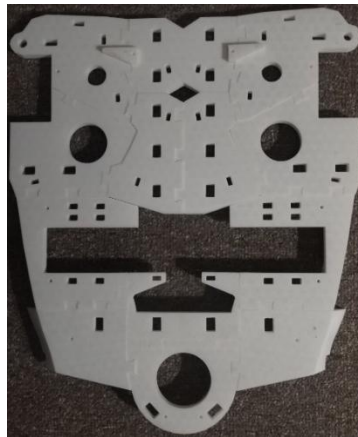
Las piezas de soporte principal en la estructura son una plancha como se ve en las Figuras 28, 29 y 30, por lo que cada pieza encaja al aplicar presión sobre la unión entre ellas (de preferencia usar una table sobre la pieza para que encajen de manera uniforme), si es difícil de ensamblar se puede rebajar las puntas de las secciones de encaje con un estilete, así estas encajaran de manera más fácil. Con lo rígidas que se vuelven las uniones no es necesario reforzarlas con pegamento.

Figura 28. Estructura trasera.



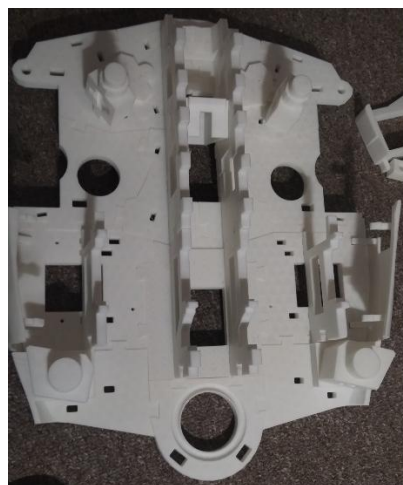
Nota: Elaboración propia.

Figura 29. Estructura frontal.



Nota: Elaboración propia.

Figura 30. Estructura intermedia.



Nota: Elaboración propia.

Las piezas que dan forma al pecho, Figura 31, y a la espalda Figura 32 se adhieren entre ellas y a la estructura principal con tornillos M3 y M4, de preferencia conseguirlo en cuanto se haya creado la estructura principal para poder definir los elementos de sujeción necesarios según la amplitud del orificio y la longitud necesaria para fijar una pieza a la otra, ya que varios de ellos tienen diferentes necesidades. Algunas piezas que tienen un ajuste único, sería recomendable asegurarlas con pegamento como los que se colocan entre la pieza de la Figura 32 y la parte trasera de la Figura 30 para obtener una pieza como en la figura 33.

Figura 31. Estructura pecho.



Nota: Elaboración propia.

Figura 32. Estructura espalda.



Nota: Elaboración propia.

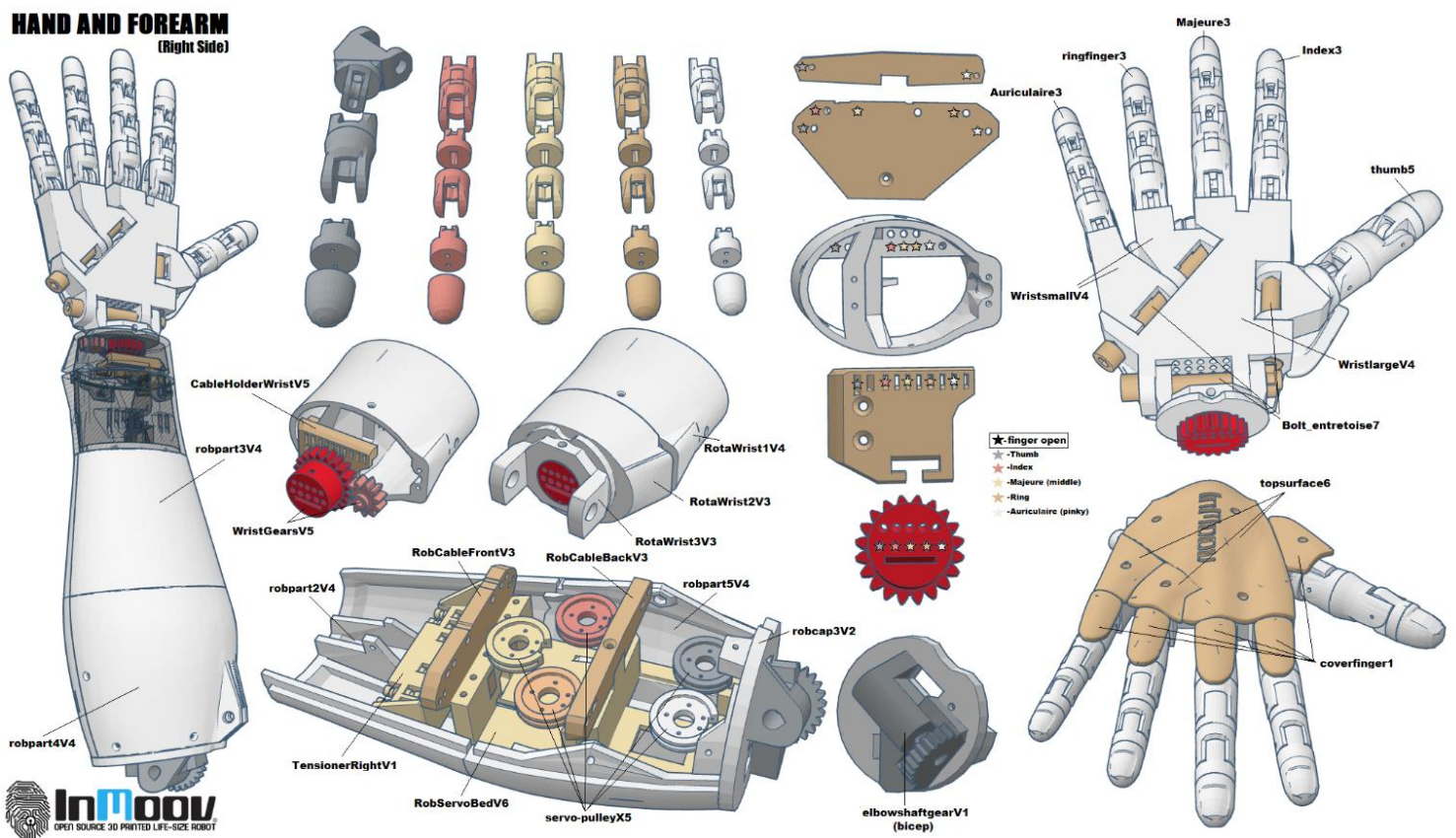
Figura 33. Estructura espalda completa.



Nota: Elaboración propia.

Para construir los brazos usaremos la Figura 34 para guiarnos, tomar en cuenta que algunas piezas necesitan ser duplicadas para poder hacer la mano derecha e izquierda.

Figura 34. Ensamblaje mano y brazo.



Nota: Fuente: Shido Xen. Adaptado de "InMoov - Google Groups", por Google Groups,

2024, https://groups.google.com/g/inmoov/c/OTdm_bNy0CA.

Las piezas que conforman el antebrazo deben ser fijadas con pegamento, tal como se ve en la Figura 35 y posteriormente se debe fijar con tornillos (considera la longitud mínima necesaria de acuerdo a los tornillos que conseguiste al principio) la estructura que sujeta los servos, como en la Figura 36 y Figura 37.

Figura 35. Antebrazo.



Nota: Elaboración propia.

Figura 36. Estructura para servos, antebrazos.



Nota: Elaboración propia.

Figura 37. Antebrazo, servos ubicados.



Nota: Elaboración propia.

Para conectar el servo al engranaje, se usa la pieza del WhrithGear, como en la Figura 38 y se adhiere con gran cantidad de pegamento bicomponente a un engrane de servo, ambos después se sujetarán con un tornillo M3 de 15mm. Lubricar la pieza con la grasa.

Al sujetar la estructura superior de la muñeca, como en la Figura 39, se debe sujetar con un tornillo M3 de 15mm, pero en caso de que no ajuste bien se puede usar tornillos M3.5 autoperforantes.

Figura 38. Servo muñeca.



Nota: Elaboración propia.

Figura 39. Ensamble brazo y muñeca completa.



Nota: Elaboración propia.

Para ajustar las piezas de los dedos, deberías limar las juntas entre ellas hasta que se muevan de manera libre como en la Figura 40 y para crear las uniones entre cada falange se puede usar el filamento plástico, un encendedor y precionarlas para que se sujete a ambos lados, como en la Figura 41, finalmente las piezas se ubican en la palma, como en la Figura 42.

Figura 40. Piezas mano.



Nota: Elaboración propia.

Figura 41. Ensamble de dedos.



Nota: Elaboración propia.

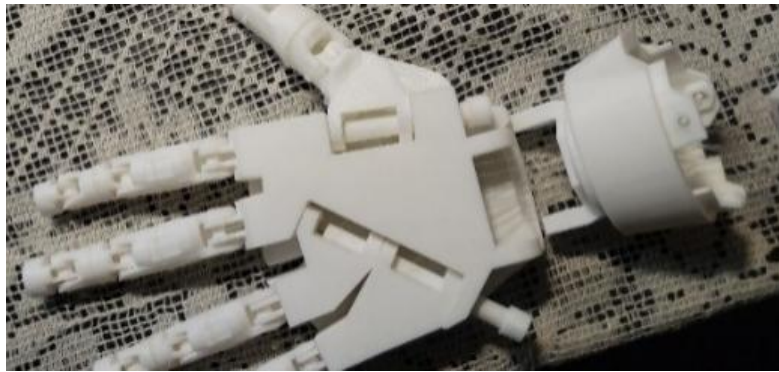
Figura 42. Mano, ensamble final.



Nota: Elaboración propia.

Para hacer el movimiento de los dedos se usará hilo nylon. Para tener una posición firme se debe usar nilón 0.8 elástico en la parte trasera del dedo que se extienda solo hasta el final de la palma y tensar hasta que el dedo tenga una tensión firme pero que se pueda flexionar el dedo Figura 43; mientras que en la posición contraria usar nylon rígido 0.4 Figura 44 y extenderlo hasta el servo correspondiente a cada dedo para finalmente cerrar el dedo con la punta correspondiente, como en la Figura 45.

Figura 43. Hilo nylon en manos.



Nota: Elaboración propia.

Figura 44. Hilo de nylon en muñecas.



Nota: Elaboración propia.

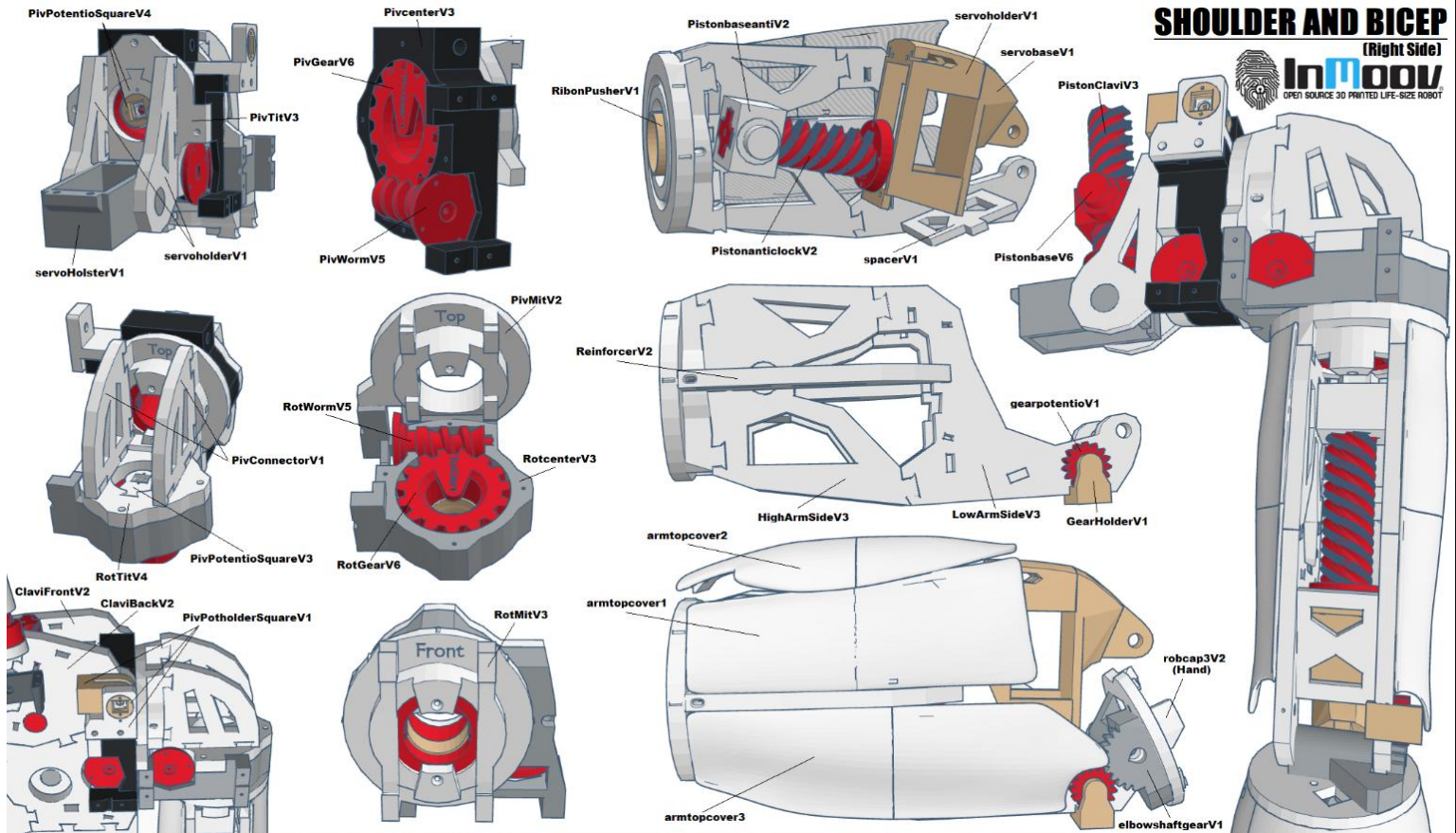
Figura 45. Puntas de los dedos en manos.



Nota: Elaboración propia.

El diagrama de la Figura 46 indica la construcción de solo un hombro y bíceps, por lo que es necesario confirmar que se tienen los duplicados necesarios para construir la segunda extremidad.

Figura 46. Ensamblaje bíceps y hombros.



Nota: Fuente: Shido Xen. Adaptado de "InMoov - Google Groups", por Google Groups, 2024, https://groups.google.com/g/inmoov/c/OTdm_bNy0CA.

Se compurban las piezas para el bicep de una extremidad Figura 47 y se ensamblan las piezas como en la Figura 48 para obtener la estructura base del bíceps y el contenedor del servo Figura 49.

Figura 47. Piezas bíceps.



Nota: Elaboración propia.

Figura 48. Estructura base bíceps.



Nota: Elaboración propia.

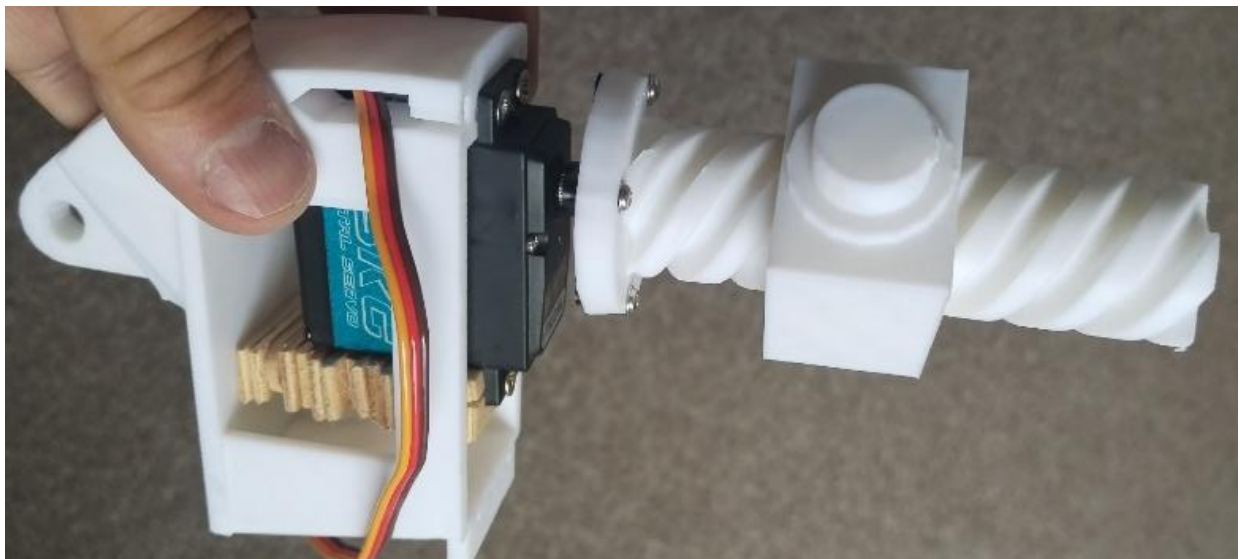
Figura 49. Estructura para servo.



Nota: Elaboración propia.

Para lograr que las estructuras del tornillo y el cubo en la Figura 50 se muevan de manera libre, es necesario desgastar hasta que se logre mover sin mucho esfuerzo, es por esto que se necesita un relleno grueso para las piezas de tornillos, así también, si los servos no caben en sus contenedores se puede adaptar el espacio con estructuras adaptadas como en este caso, con palitos de helado.

Figura 50. Estructura servo final.



Nota: Elaboración propia.

Se deben juntar las piezas que cubren el bíceps, así como el que conecta el antebrazo a este último Figura 51 con generoso pegamento bicomponente para darle resistencia a las uniones.

Figura 51. Protecciones de bíceps.



Nota: Elaboración propia.

Para ensamblar el hombro, tomar en cuenta las piezas que se dupliquen para ambos lados, como se ve en la Figura 52.

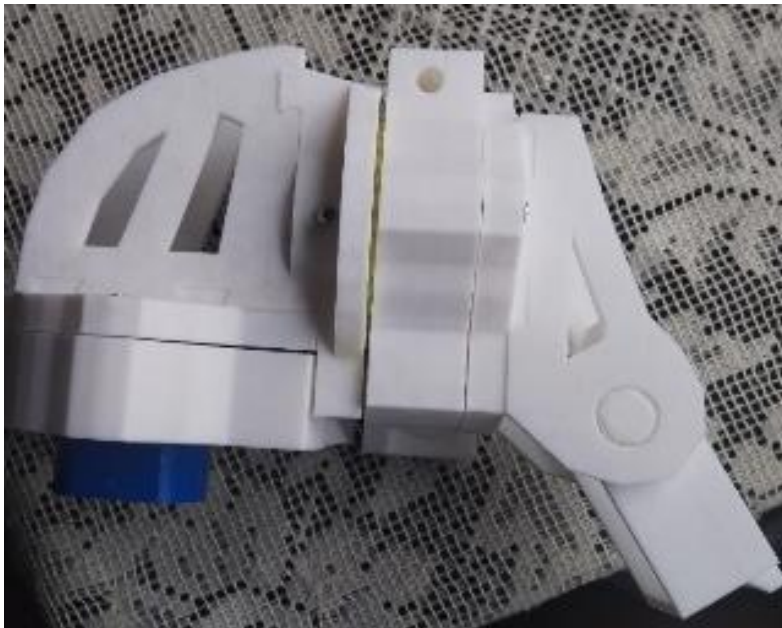
Fijarse correctamente en las piezas de rotación puesto que están tienen diferencias sutiles, usar de referencia la Figura 53 del hombro completo, especialmente las que necesitan ser lubricadas y en las que colocan balines a modo de rulimanes entre las uniones, como se aprecia en la Figura 54.

Figura 52. Piezas de hombros



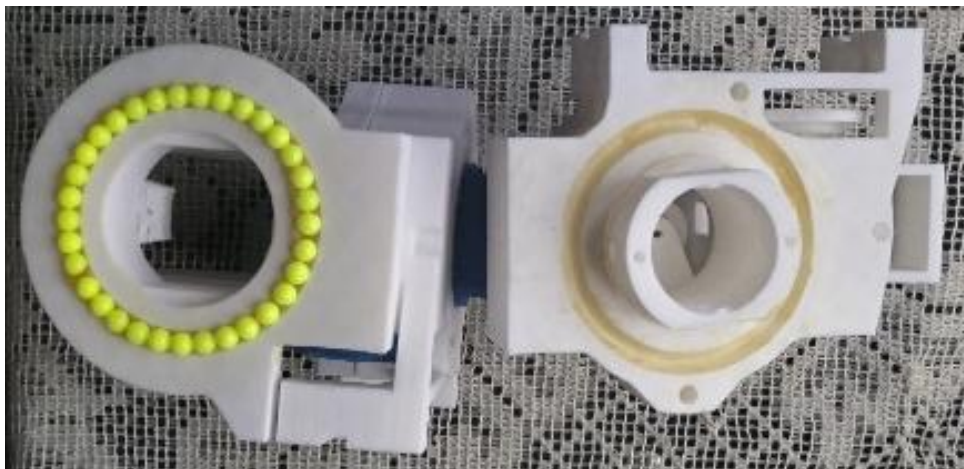
Nota: Elaboración propia

Figura 53. Hombro completo.



Nota: Elaboración propia.

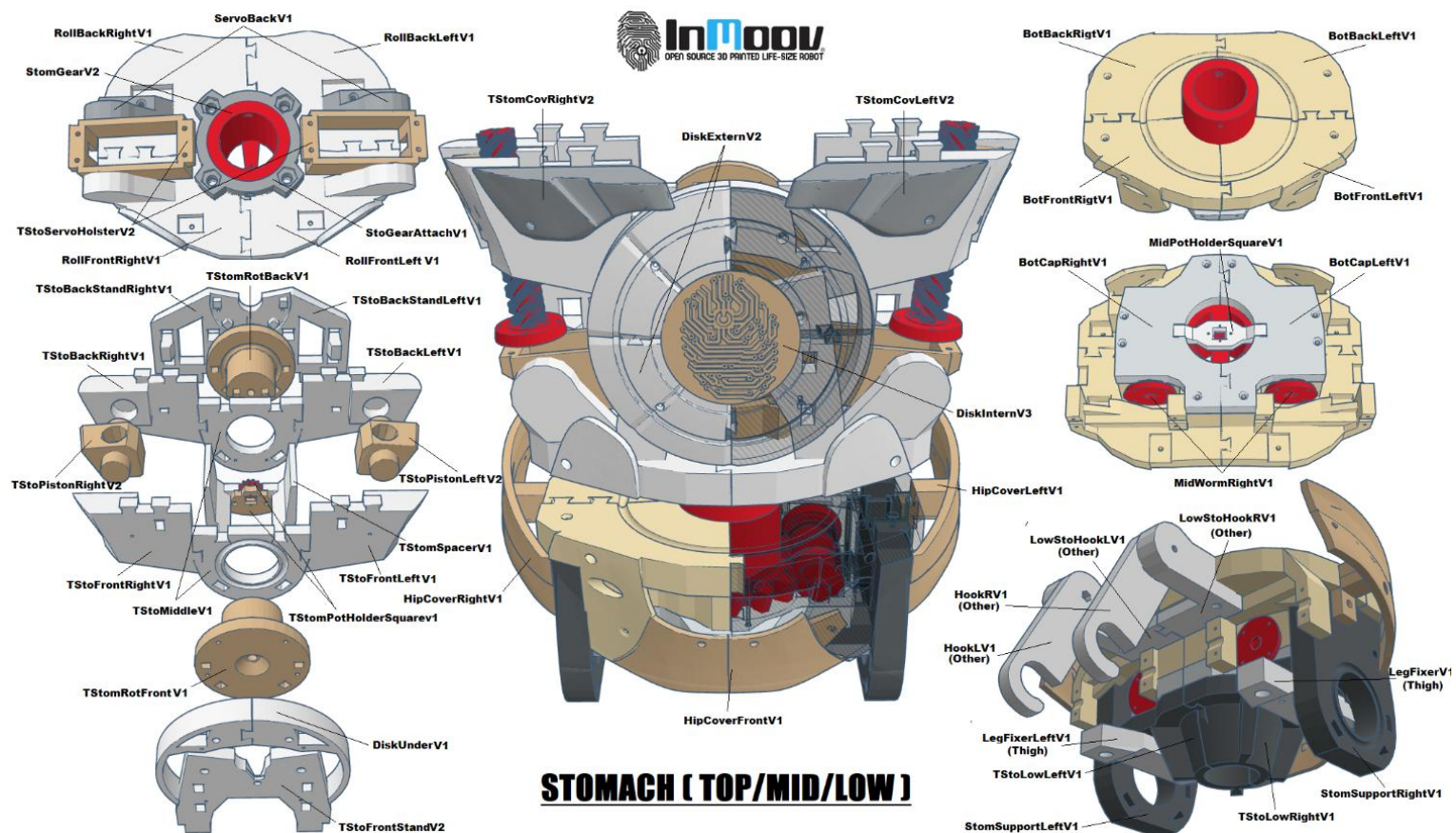
Figura 54. Lubricación de estructura de rotación, hombro.



Nota: Elaboración propia.

La estructura que se aprecia en la Figura 55, del estomago completo, debe ser ensamblada aparte para que sea más facil de manipular.

Figura 55. Ensamblaje estómago.



Nota: Fuente: Shido Xen. Adaptado de "InMoov - Google Groups", por Google Groups, 2024, https://groups.google.com/g/inmoov/c/OTdm_bNy0CA.

Las piezas más grades deben ser ajustadas entre ellas con el mismo principio que se usó para el torso.

Se debe reconocer las piezas de la estructura, como en la Figura 56, y comprobar que encajen de manera cómoda y fluida, especialmenten los tornillos sin fin conectados a los servos. Engradar especialmente los tornillos antes de cerrar la estructura.

Figura 56. Piezas de base de rotación.



Nota: Elaboración propia.

Se ve en la Figura 57 que se deben ajustar los servos, pese a que estos tienen la misma fuerza que los recomendados en la página del modelo humanoide, estos tienen un tamaño diferente, por lo que se vuelve necesario centrarlos con palitos de helado.

Figura 57. Estructura de movimiento lateral.



Nota: Elaboración propia.

No juntar las piezas circulares de la Figura 58 hasta que se ensamble de manera final toda la estructura de rotación del estómago.

Figura 58. Estructura base de movimiento lateral.



Nota: Elaboración propia.

Engrasar y colocar balines en las estructuras que se sujetan al torso, como se ve en la Figura 59 y acoplarlas juntas como se ve en la Figura 60. Finalmente incertar las piezas de la Figura 58 en la estructura base de la figura 57 para juntar la estructura del torso completa, tal como se ve en la Figura 61.

Figura 59. Lubricación de movimiento lateral.



Nota: Elaboración propia.

Figura 60. Estructura torso y movimiento lateral.



Nota: Elaboración propia.

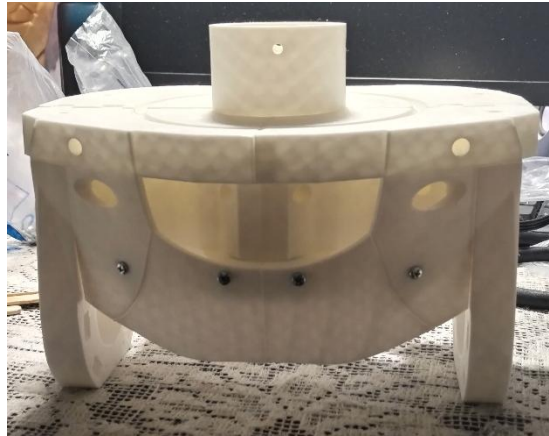
Figura 61. Ensamblaje torso.



Nota: Elaboración propia.

Finalmente junta las estructuras de la Figura 57 y la Figura 56 del estómago para formar la Figura 62, colocando en el surco que se forma entre ambas, grasa y balines para permitir el movimiento fluido, así concluir con el robot humanoide como se vé en la Figura 63.

Figura 62. Base de rotación.



Nota: Elaboración propia.

Figura 63. Ensamblaje completo del cuerpo.



Nota: Elaboración propia.

4.2.4. Montaje de elementos electrónicos

A lo largo de la construcción del robot humanoide, se deben ir colocando los servos necesarios para el movimiento del mismo. Es necesario realizar extensiones del cableado, o en su defecto crear adaptadores y shields, como sugiere la página de INMOOV, para evitar demasiados cables sueltos, de esta manera crear una apariencia más orgánica para el modelo. Esto se puede constatar en la Figura 64.

Figura 64. Ensamblaje electrónico.



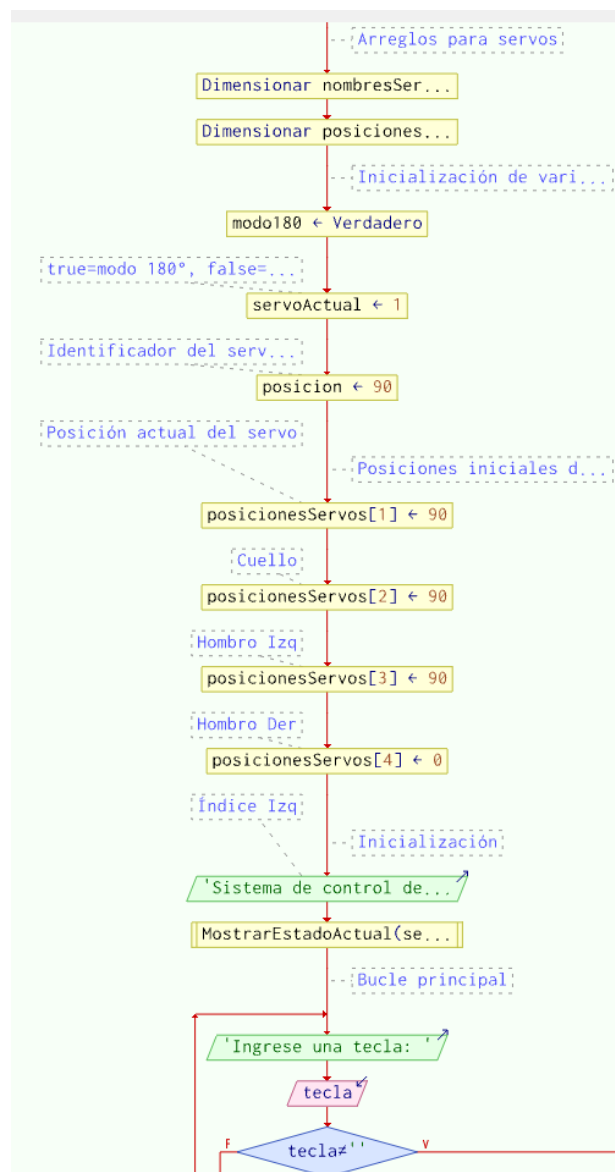
Nota: Elaboración propia.

4.2.5. Programación de movimientos y reacciones

Para la programación del robot humanoide se creó un programa inicial de control en el que se ejecuta cada servo de forma individual para determinar su correcto funcionamiento, la inicialización de cada servo y la orientación de cada movimiento.

En la Figura 65 podemos comprobar la inicialización de los servos del ejemplo y las funciones que el código usará para funcionar. En este caso la función de los servos 180 es la base del código y de acuerdo a los comandos ingresados posteriormente se podrá modificar la condición del servo, ya sea: 180 grados o 360 grados de funcionamiento.

Figura 65. Inicialización de servos.

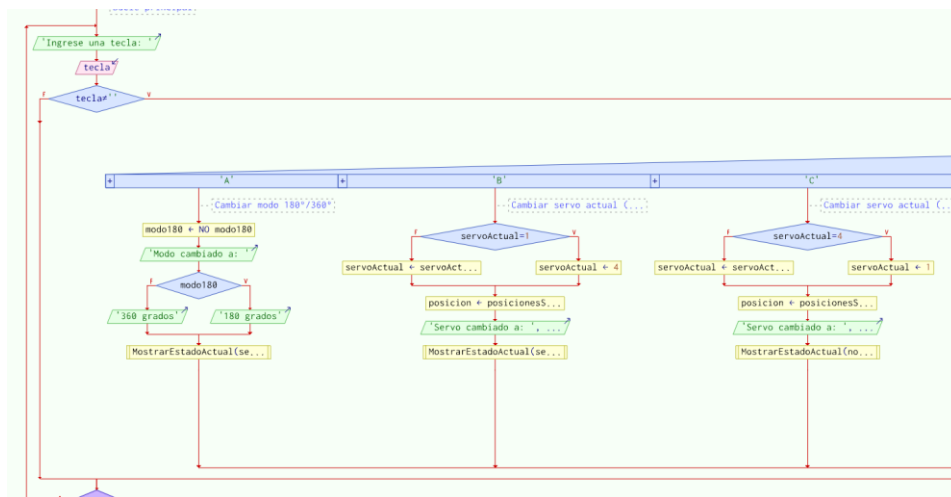


Nota: Elaboración propia.

En la Figura 66 podemos observar el primer bloque de comandos que podemos ingresar a través del teclado matricial. Inicialmente, para el comando “A” se puede intercambiar la función de control, que puede ser el control 180 o el control 360.

Los comandos “B” y “C” nos permiten cambiar de forma ascendente y descendente entre las diferentes opciones de servos disponibles en nuestra programación. Hay que tener en cuenta la organización de nuestros servos para poder saber cual estamos a punto de utilizar.

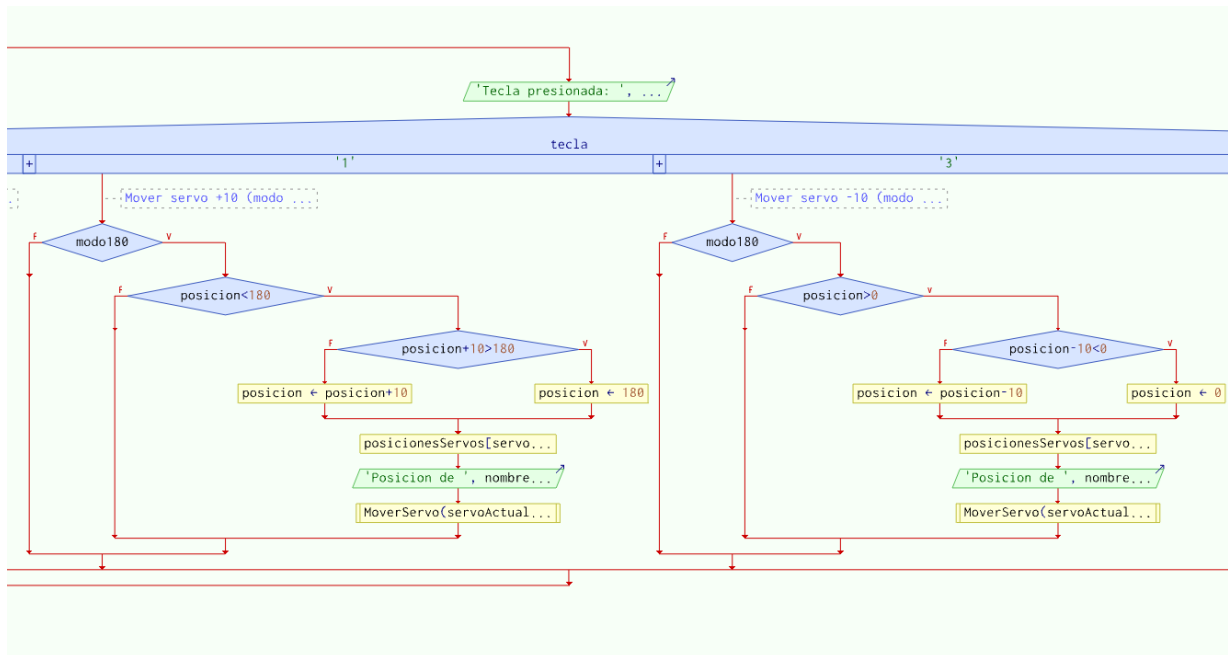
Figura 66. Comandos básicos 1.



Nota: Elaboración propia.

En la Figura 67 podemos observar las funciones asignadas a los números “1” y “3”, correspondientes al modo 180 de control de servos, donde la primera tecla suma 10 grados y la segunda resta 10 grados a la lectura del servo, siempre tomando en cuenta la inicialización particular del servo y guardando su ultima posición. En estos comandos se establece que los valores solo pueden llegar a ser iguales a 0 o 180 y sus valores intermedios.

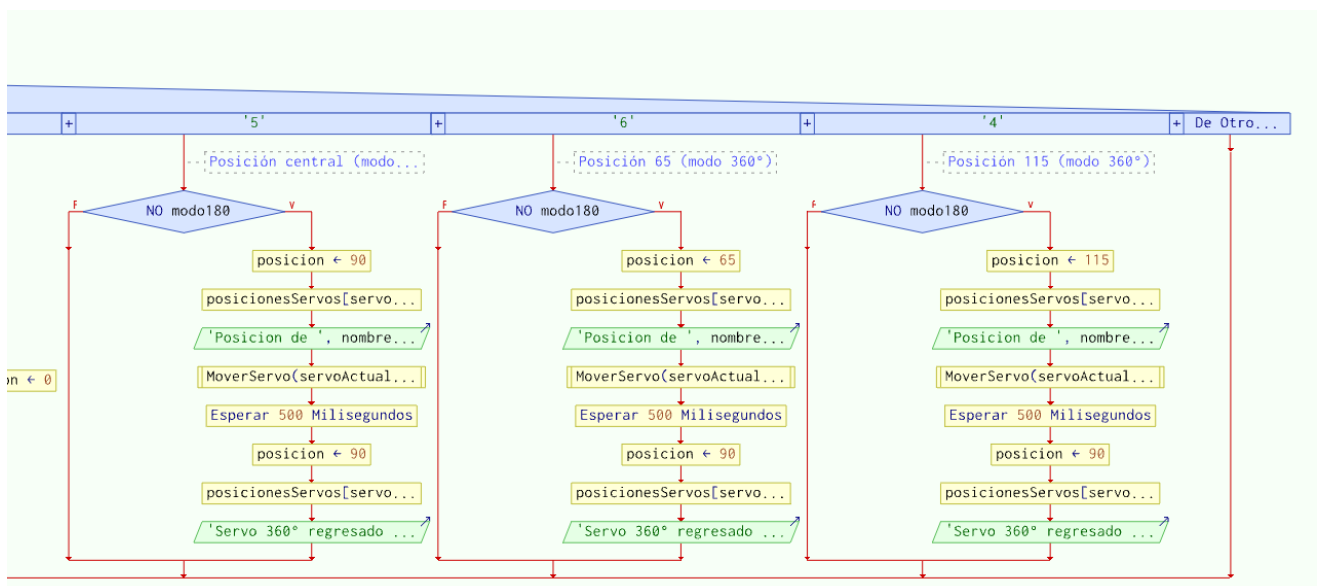
Figura 67. Comandos básicos 2



Nota: Elaboración propia.

En la Figura 68 tenemos los comandos correspondientes al modo 360, utilizando las teclas “4” y “6” para mover el servo en un sentido u otro (eso depende de la orientación del servo) durante un tiempo determinado, y “5” que es para detener el servo.

Figura 68. Comandos básicos 3.



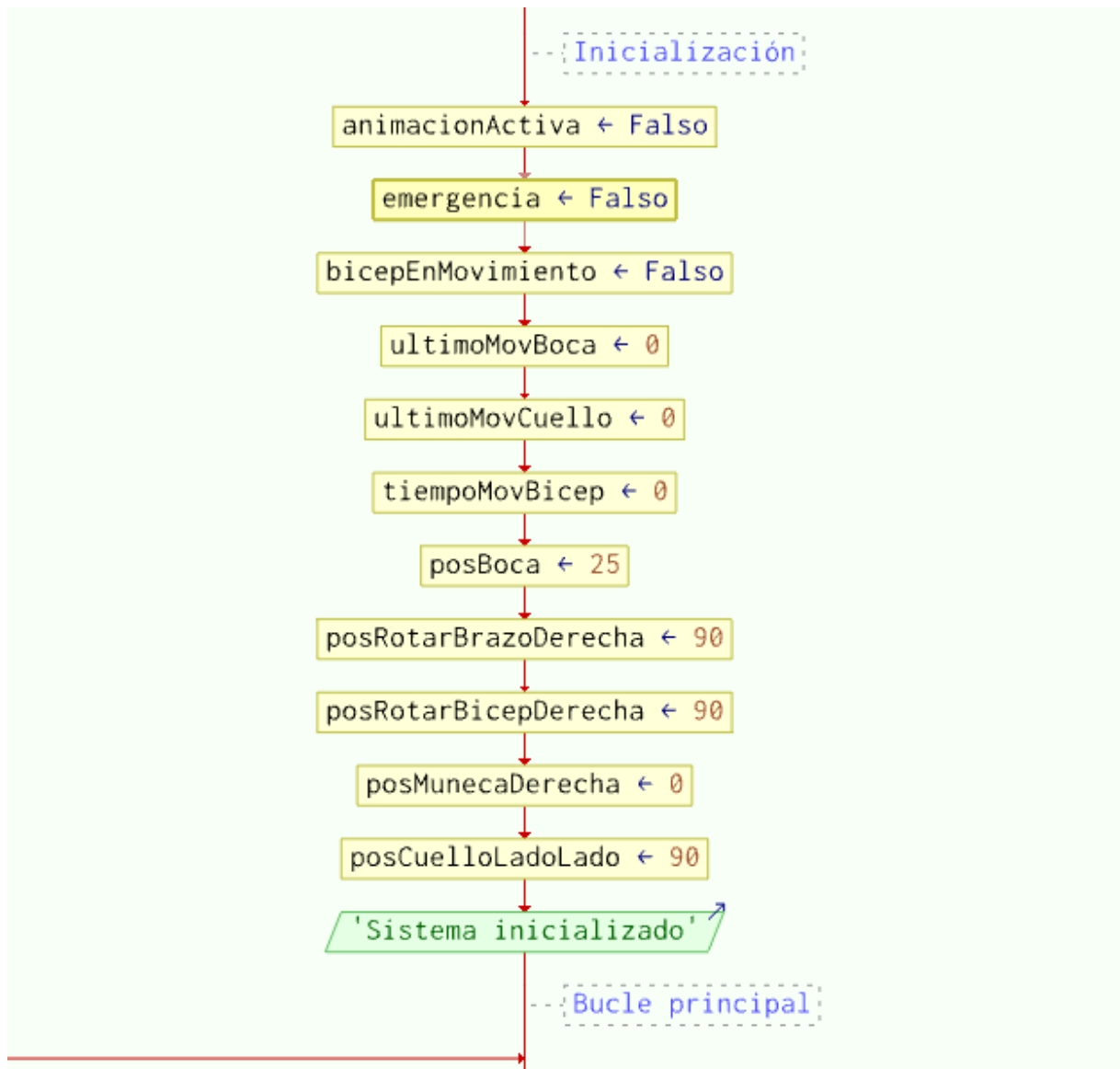
Nota: Elaboración propia.

4.3. Pruebas

4.3.1. Prueba de programación de movimientos sincronizados

A continuación, establecemos algunas funciones con movimientos sincronizados de los servos, en la Figura 69 establecemos la inicialización de aquellos que serán utilizados en el movimiento de nuestro robot humanoide.

Figura 69. Inicialización de servos.



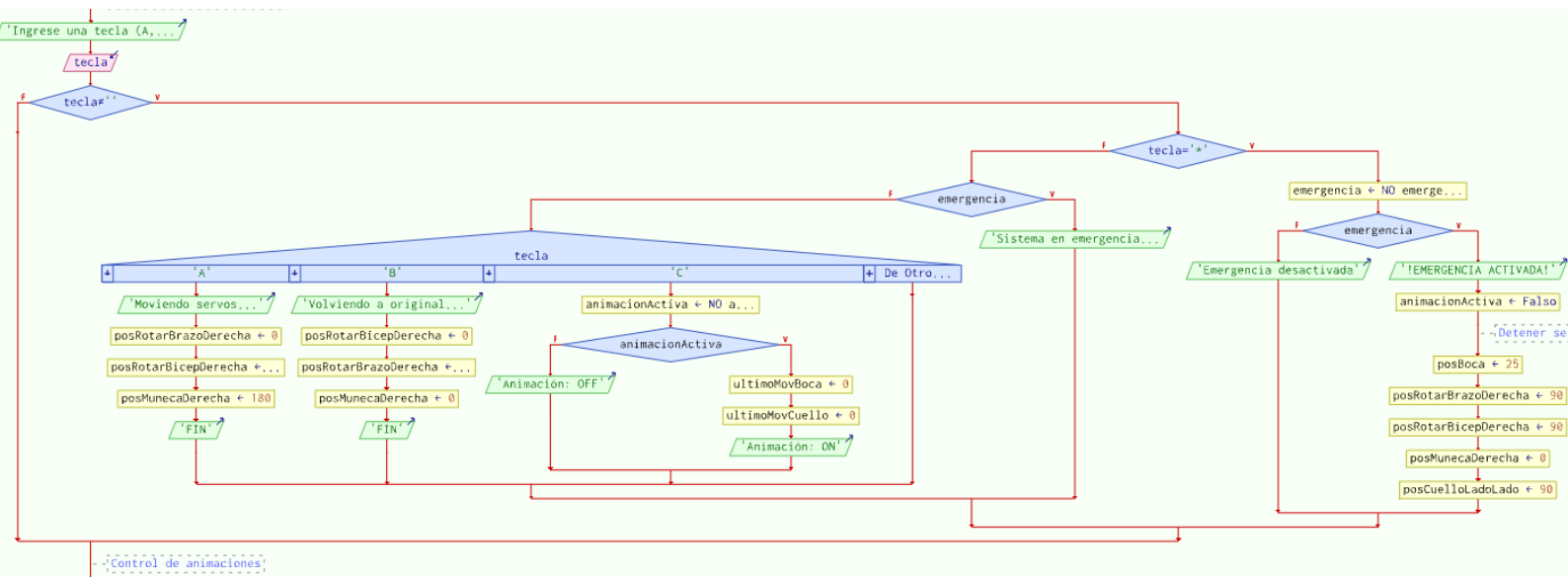
Nota: Elaboración propia.

Como se observa en la Figura 70, se utilizan las teclas: “A” para mover los servos a la posición predefinida y “B” para devolverlos a su posición original. Estas posiciones serán determinadas posteriormente en el código, como se verá en la Figura 71.

También el código establece el estado de emergencia con la tecla “ * ”, la cual puede ser activada en cualquier momento durante de la ejecución de cualquier movimiento.

Finalmente, la función de la tecla “C” que activa y desactiva la animación del movimiento del cuello y boca para simular diálogos en nuestro robot humanoide.

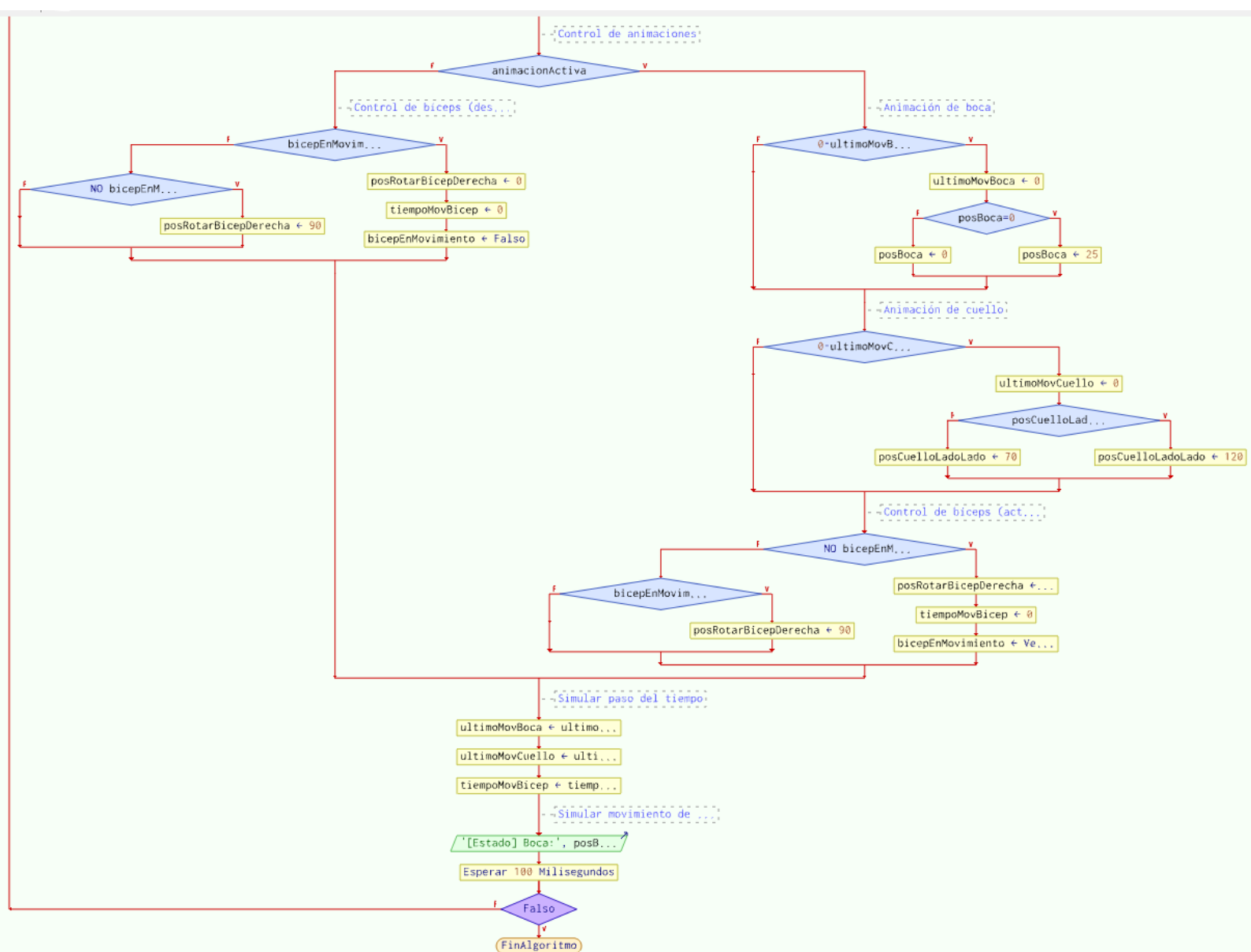
Figura 70. Estructura de control de comandos.



Nota: Elaboración propia.

En la Figura 71 observamos como se determina la posición de cada servo y su tiempo de activación de acuerdo al comando ingresado a través del teclado. Estas funciones se deben determinar por separado para cada servo, así no hay interferencias en el código y su ejecución.

Figura 71. Estructura de control de servos.



Nota: Elaboración propia.

En las Figura 72 podemos observar el estado inicial de nuestro robot humanoide, en esta posición la inicialización general de todos los servos debe ser determinada de antemano, tomado en cuenta la posición natural de un cuerpo humano. Se debe tomar en consideración la diferencia de los servos 180 y los servos 360 al iniciar el programa, pues los servos 360 deben iniciar en 90 grados para no moverse y los servos 180 deben iniciar en una posición estándar de acuerdo a como estén dispuestos.

Figura 72. Estado base del robot humanoide.



Nota: Elaboración propia.

En la Figura 73 se observa la posición final que adopta el robot después de ejecutar el comando de la tecla “A”. La sincronía de los movimientos presenta problemas debido a la manera en que se mueven los servos: es más rápido mover el brazo hacia abajo de lo que es moverlo hacia arriba, por lo que el tiempo de ejecución entre el comando de posición y su retorno difieren.

Figura 73. Movimiento.

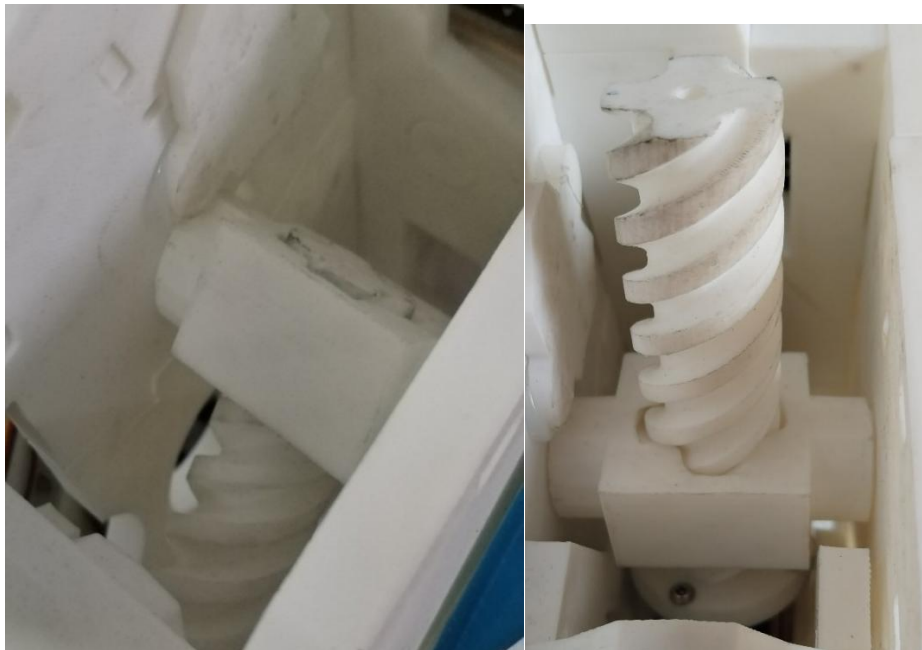


Nota: Elaboración propia. Movimiento programado que se detalla en el Anexo 2.

4.3.2. Demostración práctica de biomecánica y movimientos robóticos

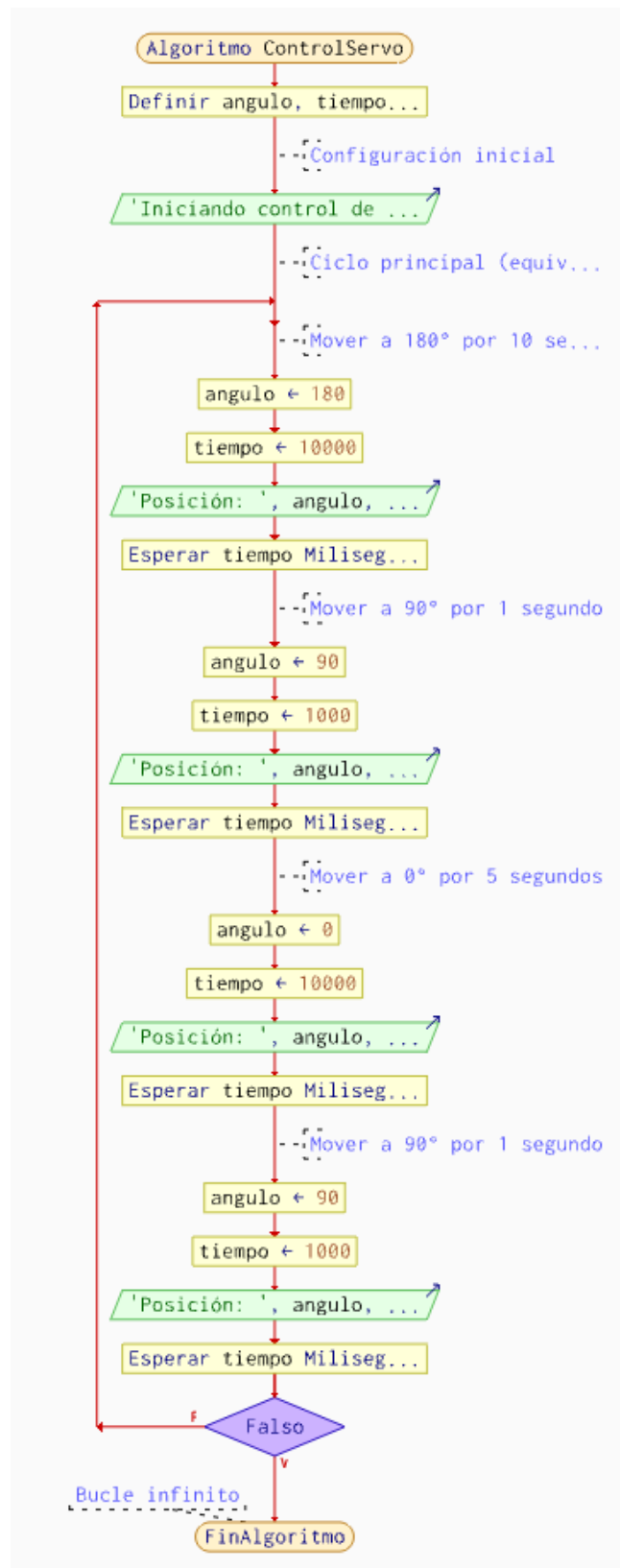
Las articulaciones en las que se usó servos de 360 grados deben ser controladas mediante tiempo; esto es necesario debido a que no hay manera directa de limitar el alcance del movimiento, como se puede observar en la Figura 74. Esto se determinará de acuerdo a la facilidad con la que se mueva determinada articulación entre la posición inicial y la posición final (de acuerdo a lo que se programe), tal como observamos en la Figura 75.

Figura 74. Limitación en hombro.



Nota: Elaboración propia.

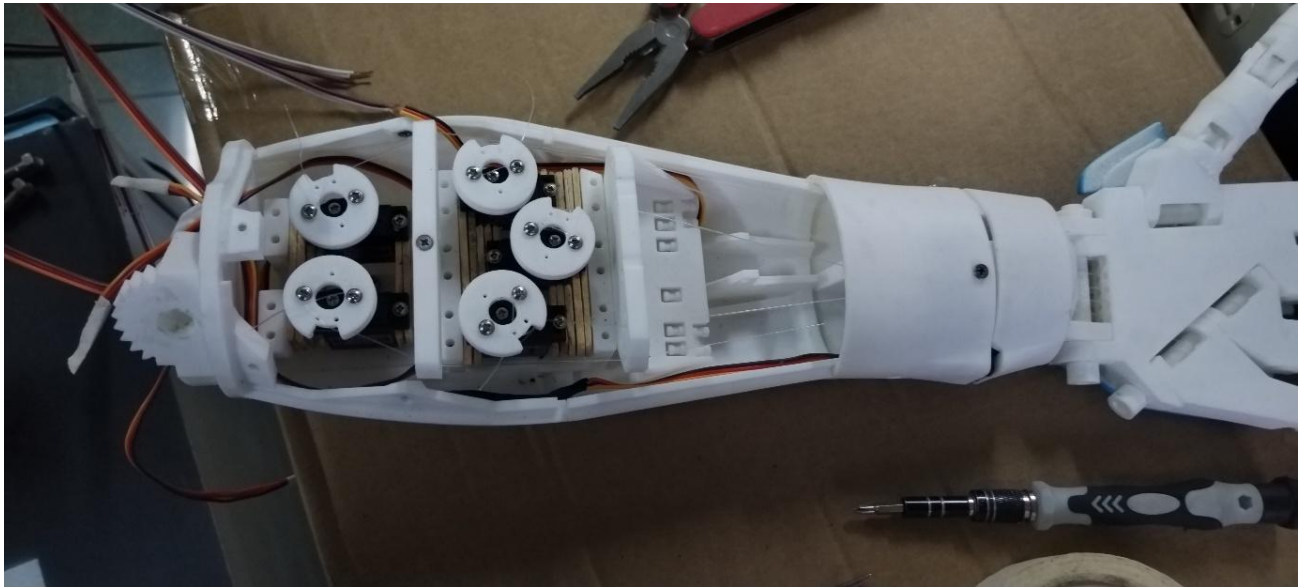
Figura 75. Limitación de movimiento por tiempo.



Nota: Elaboración propia.

Los servos de 180 grados, por su parte, requieren de un uso más cuidadoso. Lo recomendable sería moverlos de forma progresiva (sumar de 20 en 20 los grados) en lugar de súbitamente (hacer movimientos de 0 a 180 grados) para evitar arruinarlos, especialmente los servos pequeños, como en la Figura 76; ya que, pese a estar diseñados para soportar un torque de máximo 2 kg, estos no soportan el estrés del movimiento súbito.

Figura 76. Servos en brazo.



Nota: Elaboración propia.

4.4. Resultados

4.4.1. Programación de tareas prácticas para explorar sistemas electrónicos del robot

El robot humanoide ha sido programado con movimientos que asemejan la biomecánica humana, mediante este se puede entender el funcionamiento de los servos y sus diferentes características, así como sus limitantes. En la Figura 77 se puede observar como el dispositivo adopta una posición de acuerdo a lo programado y como se establecen los límites de su construcción.

La principal limitante del robot humanoide que se ha diseñado es la carencia de sensores que presenten límites en el movimiento o que permitan la interacción con el entorno, así también la fuerza de agarre limitada que tiene en el agarre, tal como vemos en la Figura 76, lo que nos permite entender el margen de mejora en cuanto diseño que tiene la construcción. Esto es un aspecto que permitirá a los educadores explorar con los estudiantes las maneras de incorporar estos medios de interacción con el entorno, haciendo dinámica la exploración de elementos,

medios y dispositivos que sirvan a ese propósito, incluso también la adaptación de nuevas piezas a la estructura con otras formas de control.

Figura 77. Movimiento 2.



Nota: Elaboración propia. Movimiento programado que se detalla en el Anexo 1

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- La integración exitosa de componentes electrónicos y de programación en el prototipo demuestra que se logró un desarrollo completo del robot humanoide con capacidades de motricidad avanzada. A través de técnicas de impresión 3D y una cuidadosa selección de servomotores, se logró replicar movimientos humanos auténticos en brazos y manos. La articulación de los dedos, la rotación de las muñecas, y los movimientos de los hombros y codos fueron controlados con gran precisión a través de un sistema programado desde una interfaz matricial. Esto demuestra la capacidad del robot para llevar a cabo funciones prácticas en experimentación y para entender conceptos de diseño robótico.
- La construcción del robot humanoide permitió alcanzar el objetivo general al ofrecer una plataforma funcional que facilita la comprensión de la robótica avanzada aplicada al diseño. Se utilizaron recursos accesibles, como el microcontrolador Arduino y software de código abierto, lo que resalta su valor como herramienta educativa. Gracias

a sus movimientos programables y a su capacidad para replicar funciones biomecánicas humanas, el prototipo se puede emplear en entornos educativos como un recurso tangible que facilita el aprendizaje de automatización, diseño tridimensional, programación estructurada y control de sistemas. Esta experiencia práctica actúa como un puente entre la teoría y la aplicación real, validando así el cumplimiento del objetivo propuesto.

- El análisis teórico que se llevó a cabo permitió establecer un marco conceptual sólido, el cual fue fundamental para el desarrollo del robot. Este estudio proporcionó las bases necesarias para entender cómo se mueven las articulaciones humanas y cómo esos movimientos pueden ser traducidos en sistemas mecánicos controlados. De esta manera, se logró cumplir con el objetivo al demostrar que las decisiones de diseño e implementación se fundamentaron en principios biomecánicos adecuados.
- A través de la comparación entre la biomecánica humana y los mecanismos robóticos, se demostró que la motricidad del robot humanoide se alinea bien con los movimientos naturales de las extremidades superiores. Al identificar puntos clave como ejes de rotación, grados de libertad y la distribución de fuerzas, se pudieron aplicar conceptos teóricos en el diseño de estructuras funcionales. Este enlace entre teoría y práctica confirmó que se entendieron los principios que rigen la robótica humanoide, lo que permitió crear un sistema que no solo imita los movimientos humanos, sino que lo hace de manera didáctica y estructurada.
- El prototipo fue creado completamente utilizando herramientas de modelado 3D, teniendo en cuenta los parámetros técnicos que surgieron del análisis realizado. La elección de materiales como el PLA y el ABS, junto con una planificación cuidadosa de la impresión, permitió obtener componentes que cuentan con la resistencia y flexibilidad necesarias. Cada pieza fue diseñada para encajar de manera modular, lo que facilitó tanto el ensamblaje como futuras mejoras o reemplazos. Esto demuestra que el diseño cumplió con los requisitos funcionales y mecánicos del objetivo propuesto.
- Se logró crear un diseño que no solo es funcional, sino también fácil de replicar, lo que lo hace ideal para que otros lo utilicen en entornos educativos o de investigación. Este diseño incorpora mecanismos complejos en áreas clave del robot, como las muñecas, los hombros y los dedos, garantizando un amplio rango de movimiento y estabilidad. Además, se tuvo en cuenta la ergonomía del prototipo, lo que permite que su estructura

sea intuitiva para ensamblar y sencilla de mantener. Esta accesibilidad refuerza su valor educativo y su utilidad como modelo de estudio.

- El sistema de programación que se desarrolló se fundamentó en el uso del entorno Arduino IDE junto con un teclado matricial, lo que permite controlar de manera intuitiva los movimientos del robot. Se creó un código que puede inicializar, identificar y operar varios servomotores. Este avance hizo posible traducir comandos digitales en acciones mecánicas precisas, demostrando que el código cumple con las funciones necesarias para operar el prototipo en situaciones reales de aprendizaje y demostración.
- El sistema de control permitió que el robot realizara movimientos sincronizados y autónomos en respuesta a las instrucciones del usuario, lo que validó su correcto funcionamiento. Se incorporaron funciones específicas para cambiar los modos de operación, seleccionar segmentos del robot y llevar a cabo tareas concretas, creando así una interfaz de control sólida y adaptable. Las pruebas realizadas confirmaron que los movimientos del robot eran precisos, fluidos y repetibles, aspectos clave para un sistema de control confiable. Por lo tanto, se logró el objetivo de desarrollar una programación funcional e intuitiva, asegurando la operatividad del prototipo desde una perspectiva educativa y técnica.

5.2. Recomendaciones

- Es recomendable optar por servomotores con mayor torque, al menos de 25 kg/cm, para lograr movimientos más precisos y fluidos, especialmente en las articulaciones que deben soportar más peso, como el torso y los hombros. En este proyecto, se utilizaron servos de 13 kg, lo que limitó algunos rangos de acción y respuesta. Además, es fundamental planificar desde el principio la compra de servos de 180° de buena calidad y con las dimensiones adecuadas para evitar restricciones en la programación por falta de tiempo, lo que puede afectar negativamente la precisión y sincronización del robot. Una mejor elección de componentes mejora directamente la funcionalidad del prototipo.
- Para sacar el máximo provecho educativo del robot en los entornos de aprendizaje, es fundamental documentar con detalle todo el proceso de ensamblaje y programación. Esto incluye recomendaciones sobre cómo elegir los servos, adaptar las piezas y controlar el movimiento. Este material será una guía valiosa para futuras réplicas del proyecto. Además, se sugiere ampliar el uso de servos de 180° modificando sus

potenciómetros, lo que permitirá un control más preciso de los ángulos y mejorará la respuesta del robot a comandos específicos. Estas mejoras hacen que el robot sea más intuitivo tanto para los estudiantes como para los docentes durante las prácticas técnicas.

- Es recomendable echar un vistazo a las experiencias previas de otros desarrolladores del InMoov para tomar decisiones más acertadas en el diseño. Esto incluye aspectos como la longitud de los cables, la ubicación de los actuadores y la distribución del peso. Al integrar teoría y simulación, se logrará una mayor precisión y eficiencia en el desarrollo mecánico del prototipo.
- En el desarrollo de este proyecto, se notó que los movimientos bruscos y continuos pueden dañar los servos de baja resistencia. Por lo tanto, se recomienda limitar la velocidad de movimiento en estas áreas, utilizando movimientos suaves de aceleración.
- Es recomendable hacer un inventario detallado de las piezas antes de comenzar el proceso de impresión, asegurándose de identificar qué componentes del robot necesitan ser duplicados para ambas extremidades. En este proyecto, no haber tenido en cuenta las piezas repetidas desde el principio complicó el ensamblaje y causó retrasos innecesarios. Además, es aconsejable etiquetar las piezas impresas al organizarlas para evitar confusiones durante la etapa de montaje. También se debe prever posibles problemas de ajuste, como ocurrió en este caso con piezas de tornillo que no encajaban correctamente; fue necesario limarlas manualmente, lo cual debilitó ciertas uniones. Se comprobó que el uso de filamento ABS, con el mismo porcentaje de relleno, facilitó significativamente el desgaste y la adaptación de las piezas móviles, resultando en un acabado más limpio y funcional. Esta experiencia resalta la importancia de elegir correctamente tanto el material como la orientación de impresión en piezas que estarán sometidas a movimiento frecuente.
- Conviene evaluar el rediseño de algunas piezas para que puedan integrarse de forma más sencilla con los servos alternativos disponibles en el mercado local, ya que la no disponibilidad de modelos específicos como los Hitec implicó adaptaciones que afectaron la funcionalidad del prototipo. Esta estrategia permitirá democratizar el acceso al robot humanoide, facilitando su aplicación en distintos contextos educativos o de investigación con recursos limitados.
- Para mejorar la gestión de los servos y la distribución de señales, se recomienda el diseño de una PCB personalizada que centralice las conexiones desde los

microcontroladores hacia los actuadores. Durante este proyecto, el exceso de cables afectó negativamente la estética y la movilidad del robot. Una PCB adaptada, como sugiere la misma página de InMoov, permitiría una distribución más ordenada y eficiente de los componentes electrónicos, lo que optimizaría el mantenimiento, la escalabilidad y la organización interna del sistema. Este paso, aunque técnico, representa una mejora notable en la estructura electrónica de cualquier robot educativo o prototipo avanzado.

- Se sugiere implementar un sistema de control mediante una interfaz gráfica o software que facilite pruebas de movimiento más seguras y precisas, especialmente en las etapas iniciales de programación. Aunque el uso de un teclado matricial ha sido útil, una interfaz visual podría ayudar a reducir errores y prevenir daños a los servos durante las pruebas de comandos. Además, sería recomendable incluir una rutina de calibración automática para los servos al inicio del programa, asegurando que todos comiencen desde la misma posición inicial, lo que minimizará errores de sincronización y prolongará la vida útil de los componentes.

5.3. Referencias

Arduino Hardware. (2025, enero 23). <https://www.arduino.cc/en/hardware>

arturo. (2020, enero 27). Los componentes eléctricos necesarios para construir un robot. *UNIT Electronics.* <https://uelectronics.com/los-componentes-electricos-necesarios-para-construir-un-robot/>

Best PLA 3D Printing Filaments of 2019. (2019, marzo 31). 3D Insider. <https://3dinsider.com/pla-3d-printing-filaments/>

Controladores de Robot—Documentación RoboDK. (2025, enero 23). <https://robodk.com/doc/es/Robot-Drivers.html>

¿Cuáles son los componentes principales de los robots humanoides (caja de cambios de motor)? (2023, diciembre 11). <https://doncenmotor.com/es/what-are-the-core-components-of-humanoid-robots/>

Diseño de robots | Siemens Software. (2025, enero 23). <https://resources.sw.siemens.com/es-ES/white-paper-overcome-complexity-accelerate-design-of-advanced-robots/>

Domínguez López, M. (2019). *Diseño de un robot humanoide de bajo coste mediante robot operating system.* <https://uvadoc.uva.es/handle/10324/37931>

eduardo.ceo. (2024, marzo 8). Partes de un robot: Identifica componentes con manualidades. *Aprender Creando.* <https://aprendercreando.com.pe/2024/03/08/partes-de-un-robot-identifica-componentes-con-manualidades/>

El ABC de la Programación de un Robot Humanoide. (2025, enero 23). SG Buzz. <https://sg.com.mx/revista/el-abc-la-programaci%C3%B3n-un-robot-humanoide>

Enríquez Salinas, A. J. (2022). *Diseño y simulación de un sistema de teleoperación de un robot humanoide NAO: Diseño y simulación del control de posicionamiento de un robot humanoide NAO.* [bachelorThesis, Quito : EPN, 2022.]. <http://bibdigital.epn.edu.ec/handle/15000/23251>

Guanajuato, U. de. (2023, junio 12). Clase digital 3. Programación y control de robots. *Recursos Educativos Abiertos.* <https://blogs.ugto.mx/rea/clase-digital-3-programacion-y-control-de-robots/>

Home | MyRobotLab. (2025, enero 23). <https://myrobotlab.org/>

Kamami.pl—Electronics store. (2025, enero 23). Kamami.Pl. <https://kamami.pl/en/>

Los softwares de impresión 3D: la Guía completa—3Dnatives. (2025, enero 23). <https://www.3dnatives.com/es/guia-programas-softwares-de-impresion-3d/>

Mejores 8 programas para impresoras 3D que puedes descargar gratis. (2025, enero 23).

<https://www.mundodeportivo.com/urbantecno/tecnologia/mejores-programas-para-impresoras-3d-que-puedes-descargar-gratis>

(PDF) DISEÑO Y CONTROL DE ROBOTS INDUSTRIALES: TEORÍA Y PRÁCTICA. (2025, enero 23). ResearchGate.

https://www.researchgate.net/publication/330449340_DISENO_Y_CONTROL_DE_ROBOTS_INDUSTRIALES_TEORIA_Y_PRACTICA

¿Qué es la impresión 3D? | Programa para impresora 3D | Autodesk. (2025, enero 23).

<https://www.autodesk.com/latam/solutions/3d-printing>

Sánchez-Girón Coca, C. (2022). *Simulación Dinámica de Robots usando Simscape (Simulink).*

Aplicación en el Movimiento de Robots Humanoides y Cuadrúpedos.

<https://uvadoc.uva.es/handle/10324/54232>

The Best 3D Printers for Beginners in 2024. (2024, enero 25). All3DP.

<https://all3dp.com/1/best-3d-printer-for-beginners/>

Tipos de impresoras 3D: las 7 tecnologías de impresión 3D | All3DP. (2025, enero 23).

<https://all3dp.com/es/1/tipos-de-impresoras-3d-tecnologia-de-impresion-3d/>

Valdiviezo Romero, C. A., & Valladares Yanqui, R. A. (2023). *Implementación de un robot humanoide con reconocimiento de objetos por color y forma* [bachelorThesis].

<http://dspace.ups.edu.ec/handle/123456789/24123>

ANEXOS

Anexo 1. 2Servos.control

```
#include <Keypad.h>
#include <Servo.h>

/* A: alto de emergencia
B - C: cambio de servos de la lista
1,3: control de servos 180
4,5,6: control de servos 360*/

// Configuración del teclado
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {21, 20, 19, 18};
byte colPins[COLS] = {17, 16, 15, 14};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Declaración de servos
Servo boca, ojoHorizontal, ojoVertical, cuelloLadoLado, cuelloCentro;
Servo aperturaHombroIzquierda, rotarBrazoIzquierda, bicepIzquierda,
munecaIzquierda, medioIzquierda;
Servo rotarBicepIzquierda, anularIzquierda, indiceIzquierda, pulgarIzquierda,
meniqueIzquierda;
Servo meniqueDerecha, pulgarDerecha, anularDerecha, indiceDerecha;
Servo rotarBicepDerecha, munecaDerecha, bicepDerecha, medioDerecha;
Servo rotarBrazoDerecha, aperturaHombroDerecha;
Servo cuelloIzquierda, cuelloDerecha;
Servo rotacionIzquierda, rotacionDerecha;
Servo lateralIzquierda, lateralDerecha;

// Variables de estado
bool modo180 = false;
bool modo360 = false;
bool sistemaActivo = false;
int posicion = 90;
int servoActual = 0;

// Temporizador para servos 360°
```

```

unsigned long tiempoMovimiento = 0;
bool servo360Activo = false;

// Lista de nombres y modos
const String nombresServos[] = {
  "boca", "ojoHorizontal", "ojoVertical", "cuelloLadoLado", "cuelloCentro",
  "cuelloIzqDer", "rotacionIzqDer", "lateralIzqDer",
  "aperturaHombroIzquierda", "rotarBrazoIzquierda", "bicepIzquierda",
  "munecaIzquierda", "medioIzquierda",
  "rotarBicepIzquierda", "anularIzquierda", "indiceIzquierda",
  "pulgarIzquierda", "meniqueIzquierda",
  "meniqueDerecha", "pulgarDerecha", "anularDerecha", "indiceDerecha",
  "rotarBicepDerecha", "munecaDerecha", "bicepDerecha", "medioDerecha",
  "rotarBrazoDerecha", "aperturaHombroDerecha"
};

const bool servosModo180[] = {
  true, true, true, true, false,
  false, false, false,
  false, false, false, true, true,
  false, true, true, true, true,
  true, true, true, true,
  false, true, false, true,
  false, false
};

void activarSistema(bool activar) {
  sistemaActivo = activar;
  inicializarServos();
  if (activar) {
    Serial.println("Sistema ACTIVADO");
  } else {
    Serial.println("Sistema DESACTIVADO");
  }
}

void inicializarServos() {
  boca.attach(7);          boca.write(0);
  ojoHorizontal.attach(8);  ojoHorizontal.write(90);
  ojoVertical.attach(9);    ojoVertical.write(110);
  cuelloLadoLado.attach(10); cuelloLadoLado.write(90);
  cuelloCentro.attach(13);  cuelloCentro.write(90);
  cuelloIzquierda.attach(12); cuelloIzquierda.write(90);
  cuelloDerecha.attach(11);  cuelloDerecha.write(90);
  rotacionIzquierda.attach(33); rotacionIzquierda.write(90);
  rotacionDerecha.attach(34); rotacionDerecha.write(90);
  lateralIzquierda.attach(32); lateralIzquierda.write(90);
  lateralDerecha.attach(35); lateralDerecha.write(90);
  aperturaHombroIzquierda.attach(22); aperturaHombroIzquierda.write(90);
}

```

```

rotarBrazoIzquierda.attach(23);    rotarBrazoIzquierda.write(90);
bicepIzquierda.attach(24);        bicepIzquierda.write(90);
munecaIzquierda.attach(25);       munecaIzquierda.write(90);
rotarBicepIzquierda.attach(27);   rotarBicepIzquierda.write(90);
medioIzquierda.attach(26);        medioIzquierda.write(0);
anularIzquierda.attach(28);       anularIzquierda.write(180);
indiceIzquierda.attach(29);       indiceIzquierda.write(0);
pulgarIzquierda.attach(31);       pulgarIzquierda.write(0);
meniqueIzquierda.attach(30);      meniqueIzquierda.write(180);
meniqueDerecha.attach(36);        meniqueDerecha.write(0);
pulgarDerecha.attach(37);         pulgarDerecha.write(180);
anularDerecha.attach(38);         anularDerecha.write(0);
indiceDerecha.attach(39);         indiceDerecha.write(180);
medioDerecha.attach(43);          medioDerecha.write(0);
rotarBicepDerecha.attach(40);     rotarBicepDerecha.write(90);
munecaDerecha.attach(41);         munecaDerecha.write(90);
bicepDerecha.attach(42);          bicepDerecha.write(90);
rotarBrazoDerecha.attach(44);     rotarBrazoDerecha.write(90);
aperturaHombroDerecha.attach(45); aperturaHombroDerecha.write(90);

posicion = (servoActual == 0) ? 0 : 90;
modo180 = servosModo180[servoActual];
modo360 = !modo180;

Serial.print("Servo actual: ");
Serial.println(nombresServos[servoActual]);
}

void moverServoActual(int pos) {
    if (!sistemaActivo) return;
    if (servoActual == 0) pos = constrain(pos, 0, 20);

    switch (servoActual) {
        case 0: boca.write(pos); break;
        case 1: ojoHorizontal.write(pos); break;
        case 2: ojoVertical.write(pos); break;
        case 3: cuelloLadoLado.write(pos); break;
        case 4: cuelloCentro.write(pos); break;
        case 5:
            cuelloIzquierda.write(pos == 0 ? 180 : (pos == 180 ? 0 : 90));
            cuelloDerecha.write(pos == 0 ? 0 : (pos == 180 ? 180 : 90));
            break;
        case 6:
            rotacionIzquierda.write(pos == 0 ? 180 : (pos == 180 ? 0 : 90));
            rotacionDerecha.write(pos == 0 ? 0 : (pos == 180 ? 180 : 90));
            break;
        case 7:
            lateralIzquierda.write(pos);
    }
}

```



```

        lateralDerecha.write(pos);
        break;
    case 8: aperturaHombroIzquierda.write(pos); break;
    case 9: rotarBrazoIzquierda.write(pos); break;
    case 10: bicepIzquierda.write(pos); break;
    case 11: muñecaIzquierda.write(pos); break;
    case 12: medioIzquierda.write(pos); break;
    case 13: rotarBicepIzquierda.write(pos); break;
    case 14: anularIzquierda.write(pos); break;
    case 15: indiceIzquierda.write(pos); break;
    case 16: pulgarIzquierda.write(pos); break;
    case 17: meniqueIzquierda.write(pos); break;
    case 18: meniqueDerecha.write(pos); break;
    case 19: pulgarDerecha.write(pos); break;
    case 20: anularDerecha.write(pos); break;
    case 21: indiceDerecha.write(pos); break;
    case 22: rotarBicepDerecha.write(pos); break;
    case 23: muñecaDerecha.write(pos); break;
    case 24: bicepDerecha.write(pos); break;
    case 25: medioDerecha.write(pos); break;
    case 26: rotarBrazoDerecha.write(pos); break;
    case 27: aperturaHombroDerecha.write(pos); break;
}

posicion = pos;
Serial.print("Posición actualizada: ");
Serial.println(posicion);

// Si es modo 360, iniciar temporizador para volver a 90°
if (modo360 && pos != 90) {
    tiempoMovimiento = millis();
    servo360Activo = true;
}
}

void cambiarServo(bool ascendente) {
    if (!sistemaActivo) return;
    servoActual = (ascendente) ? (servoActual + 1) % 28 : (servoActual - 1 + 28) % 28;
    modo180 = servosModo180[servoActual];
    modo360 = !modo180;
    if (modo360) {
        moverServoActual(90);
        posicion = 90;
    }
    Serial.print("Servo actual: ");
    Serial.print(nombresServos[servoActual]);
    Serial.print(" - Modo: ");

```

```

    Serial.println(modo180 ? "180°" : "360°");
}

void setup() {
    Serial.begin(9600);
    inicializarServos();
    Serial.println("Sistema listo. Presione * para activar o desactivar el control.");
}

void loop() {
    char tecla = keypad.getKey();

    if (tecla) {
        switch (tecla) {
            case '*':
                activarSistema(!sistemaActivo);
                break;
            case 'B':
                cambiarServo(true);
                break;
            case 'C':
                cambiarServo(false);
                break;
            case '1':
                if (modo180 && posicion > 0) moverServoActual(posicion - 10);
                break;
            case '3':
                if (modo180 && posicion < 180) moverServoActual(posicion + 10);
                break;
            case '4':
                if (modo360) moverServoActual(0);
                break;
            case '5':
                if (modo360) moverServoActual(90);
                break;
            case '6':
                if (modo360) moverServoActual(180);
                break;
        }
    }

    // Verifica si debe retornar a 90° en modo 360
    if (sistemaActivo && modo360 && servo360Activo && (millis() - tiempoMovimiento >= 1000)) {
        moverServoActual(90);
        servo360Activo = false;
    }
}

```

```
}
```

Anexo 2. Control.Comandos

```
#include <Keypad.h>
#include <Servo.h>

/* movimientos programados con:
1 para activar y desactivar la acción
2 para activar y desactivar la acción
3 para activar la acción
4 para activar y desactivar la acción*/

// Configuración del teclado matricial 4x4
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {21, 20, 19, 18}; // Pines de las filas
byte colPins[COLS] = {17, 16, 15, 14}; // Pines de las columnas

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// === Declaración de TODOS los servos ===
// Cara
Servo boca;
Servo ojoHorizontal;
Servo ojoVertical;

// Cuello
Servo cuelloLadoLado;
Servo cuelloCentro;
Servo cuelloIzquierda;
Servo cuelloDerecha;

// Hombro y brazo izquierdos
Servo rotacionIzquierda;
Servo lateralIzquierda;
Servo aperturaHombroIzquierda;
Servo rotarBrazoIzquierda;
Servo bicepIzquierda;
Servo muñecaIzquierda;
Servo rotarBicepIzquierda;
```

```

// Mano izquierda
Servo medioIzquierda;
Servo anularIzquierda;
Servo indiceIzquierda;
Servo pulgarIzquierda;
Servo meniqueIzquierda;

// Hombro y brazo derechos
Servo rotacionDerecha;
Servo lateralDerecha;
Servo aperturaHombroDerecha;
Servo rotarBrazoDerecha;
Servo bicepDerecha;
Servo munecaDerecha;
Servo rotarBicepDerecha;

// Mano derecha
Servo medioDerecha;
Servo anularDerecha;
Servo indiceDerecha;
Servo pulgarDerecha;
Servo meniqueDerecha;

// Variables de estado
bool estadoAccion1 = false;
bool estadoMovimiento2 = false;
int contadorPresion4 = 0;
bool bloqueo = false;

// Función para mover servos
void moverServo(Servo &servo, int angulo, int tiempo) {
    if (bloqueo) return;

    unsigned long inicio = millis();
    servo.write(angulo);
    delay(tiempo);
    servo.write(90); // Vuelve a posición central
}

// Función para mover múltiples servos simultáneamente con movimiento suave en
3 segundos
void moverDedosSimultaneos(Servo dedos[], int posInicial[], int anguloFinal[],
int numDedos) {
    if (bloqueo) return;

    const unsigned long duracionTotal = 3000; // 3 segundos
    const unsigned long inicio = millis();

```

```

while (millis() - inicio < duracionTotal) {
    float progreso = (float)(millis() - inicio) / duracionTotal;

    for (int i = 0; i < numDedos; i++) {
        int posicionActual = posInicial[i] + (anguloFinal[i] - posInicial[i]) *
progreso;
        dedos[i].write(posicionActual);
    }

    delay(20); // Pequeña pausa para suavizar el movimiento
}

// Asegurarse de que todos los servos llegaron exactamente a la posición
final
for (int i = 0; i < numDedos; i++) {
    dedos[i].write(anguloFinal[i]);
}
}

// Función para mover servos simultáneamente con tiempos independientes
void moverParejaServos(Servo &servo1, int angulo1, int tiempo1,
                        Servo &servo2, int angulo2, int tiempo2) {
    if (bloqueo) return;

    unsigned long startTime = millis();
    unsigned long endTime1 = startTime + tiempo1;
    unsigned long endTime2 = startTime + tiempo2;

    // Mover ambos servos a sus posiciones
    servo1.write(angulo1);
    servo2.write(angulo2);

    // Esperar hasta que ambos tiempos hayan terminado
    while(millis() < max(endTime1, endTime2)) {
        // Si el tiempo del servo1 ha terminado, volver a posición central
        if(millis() >= endTime1) {
            servo1.write(90);
        }
        // Si el tiempo del servo2 ha terminado, volver a posición central
        if(millis() >= endTime2) {
            servo2.write(90);
        }
    }

    // Asegurar que ambos estén en posición central al final
    servo1.write(90);
    servo2.write(90);
}

```

```

// Función para mover servos en sentidos opuestos
void moverServosOpuestos(int anguloIzq, int anguloDer, int tiempo) {
    if (bloqueo) return;

    unsigned long startTime = millis();

    rotacionIzquierda.write(anguloIzq);
    rotacionDerecha.write(anguloDer);

    while(millis() - startTime < tiempo) {
        // Espera activa
    }

    rotacionIzquierda.write(90);
    rotacionDerecha.write(90);
}

// Función para mover un dedo y la boca sincronizados (2 segundos)
void moverDedoYBoca(Servo &servo, int posInicial, int posFinal) {
    if (bloqueo) return;

    unsigned long tiempoInicio = millis();
    unsigned long duracion = 2000; // 2 segundos

    while (millis() - tiempoInicio < duracion) {
        float progreso = float(millis() - tiempoInicio) / duracion;
        int posDedo = posInicial + (posFinal - posInicial) * progreso;

        // Hacer que la boca complete 2 ciclos (0-20-0-20-0) durante el movimiento
        // del dedo
        float cicloBoca = fmod(progreso * 4.0, 2.0); // 4 ciclos (2 completos) en
        // el tiempo total
        int posBoca = (cicloBoca < 1.0) ? (20 * cicloBoca) : (20 * (2.0 -
        cicloBoca));

        servo.write(posDedo);
        boca.write(posBoca);
        delay(20);
    }

    // Asegurar posiciones finales exactas
    servo.write(posFinal);
    boca.write(0); // Asegurar que la boca termine en 0
}

void moverDedosSecuencial() {
    if (bloqueo) return;

```

```

// Calcular el incremento del cuello por cada dedo
float inicioCuello = 45;
float finCuello = 135;
float incrementoCuello = (finCuello - inicioCuello) / 10.0; // 10 dedos en
total
float posicionCuello = inicioCuello;
cuelloLadoLado.write(posicionCuello); // Posición inicial

// Mano izquierda
moverDedoYBoca(pulgarIzquierda, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(indiceIzquierda, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(medioIzquierda, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(anularIzquierda, 180, 0);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(meniqueIzquierda, 180, 0);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

// Mano derecha
moverDedoYBoca(meniqueDerecha, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(anularDerecha, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(medioDerecha, 0, 180);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(indiceDerecha, 180, 0);
posicionCuello += incrementoCuello;
cuelloLadoLado.write(posicionCuello);

moverDedoYBoca(pulgarDerecha, 180, 0);

```



```

    posicionCuello += incrementoCuello;
    cuelloLadoLado.write(posicionCuello);

    // Asegurar posición final exacta
    cuelloLadoLado.write(finCuello);
}

void regresarASuave() {
    if (bloqueo) return;

    int pasos = 100; // Más pasos para un movimiento más lento
    int delayPaso = 30;

    for (int i = 0; i <= pasos; i++) {
        float progreso = float(i) / pasos;

        // Mover dedos a posición inicial gradualmente
        pulgarIzquierda.write(180 + progreso * (0 - 180));
        indiceIzquierda.write(180 + progreso * (0 - 180));
        medioIzquierda.write(180 + progreso * (0 - 180));
        anularIzquierda.write(0 + progreso * (180 - 0));
        meniqueIzquierda.write(0 + progreso * (180 - 0));

        pulgarDerecha.write(0 + progreso * (180 - 0));
        indiceDerecha.write(0 + progreso * (180 - 0));
        medioDerecha.write(180 + progreso * (0 - 180));
        anularDerecha.write(180 + progreso * (0 - 180));
        meniqueDerecha.write(180 + progreso * (0 - 180));

        // Mover muñecas a posición inicial (90 grados)
        muñecaDerecha.write(0 + progreso * (90 - 0));
        muñecaIzquierda.write(180 + progreso * (90 - 180));

        delay(delayPaso);
    }
}

void setup() {
    Serial.begin(9600);

    // === Inicialización de servos del rostro ===
    boca.attach(7); boca.write(90);
    ojoHorizontal.attach(8); ojoHorizontal.write(90);
    ojoVertical.attach(9); ojoVertical.write(90);

    // === Inicialización de cuello ===
    cuelloLadoLado.attach(10); cuelloLadoLado.write(90);
    cuelloCentro.attach(13); cuelloCentro.write(90);
}

```

```

cuelloIzquierda.attach(12); cuelloIzquierda.write(90);
cuelloDerecha.attach(11); cuelloDerecha.write(90);

// === Inicialización de brazo izquierdo ===
rotacionIzquierda.attach(33); rotacionIzquierda.write(90);
lateralIzquierda.attach(32); lateralIzquierda.write(90);
aperturaHombroIzquierda.attach(22); aperturaHombroIzquierda.write(90);
rotarBrazoIzquierda.attach(23); rotarBrazoIzquierda.write(90);
bicepIzquierda.attach(24); bicepIzquierda.write(90);
munecaIzquierda.attach(25); munecaIzquierda.write(90);
rotarBicepIzquierda.attach(27); rotarBicepIzquierda.write(90);

// === Inicialización de mano izquierda ===
medioIzquierda.attach(26); medioIzquierda.write(0);
anularIzquierda.attach(28); anularIzquierda.write(180);
indiceIzquierda.attach(29); indiceIzquierda.write(0);
pulgarIzquierda.attach(31); pulgarIzquierda.write(0);
meniqueIzquierda.attach(30); meniqueIzquierda.write(180);

// === Inicialización de brazo derecho ===
rotacionDerecha.attach(34); rotacionDerecha.write(90);
lateralDerecha.attach(35); lateralDerecha.write(90);
aperturaHombroDerecha.attach(45); aperturaHombroDerecha.write(90);
rotarBrazoDerecha.attach(44); rotarBrazoDerecha.write(90);
bicepDerecha.attach(42); bicepDerecha.write(90);
munecaDerecha.attach(41); munecaDerecha.write(90);
rotarBicepDerecha.attach(40); rotarBicepDerecha.write(90);

// === Inicialización de mano derecha ===
medioDerecha.attach(43); medioDerecha.write(0);
anularDerecha.attach(38); anularDerecha.write(0);
indiceDerecha.attach(39); indiceDerecha.write(180);
pulgarDerecha.attach(37); pulgarDerecha.write(180);
meniqueDerecha.attach(36); meniqueDerecha.write(0);

Serial.println("Se inicializaron todos los servos");
}

void loop() {
    char tecla = keypad.getKey();

    if (tecla && !bloqueo) {
        Serial.print("Tecla presionada: ");
        Serial.println(tecla);

        switch(tecla) {
            case '1':
                bloqueo = true;

```

```

if(!estadoAccion1) {
    // Acción - Movimiento hacia adelante
    // Brazos
    moverServo(rotarBicepIzquierda, 0, 5000);
    moverServo(rotarBicepDerecha, 0, 5000);
    moverServo(aperturaHombroIzquierda, 0, 2000);
    moverServo(aperturaHombroDerecha, 0, 2000);

    // Dedos izquierdos
    Servo dedosIzq[] = {medioIzquierda, anularIzquierda,
meniqueIzquierda, indiceIzquierda, pulgarIzquierda};
    int posInicialIzq[] = {0, 180, 180, 0, 0};
    int anguloFinalIzq[] = {180, 0, 0, 180, 180};
    moverDedosSimultaneos(dedosIzq, posInicialIzq, anguloFinalIzq, 5);

    // Dedos derechos
    Servo dedosDer[] = {medioDerecha, anularDerecha, meniqueDerecha,
indiceDerecha, pulgarDerecha};
    int posInicialDer[] = {0, 0, 0, 180, 180};
    int anguloFinalDer[] = {180, 180, 180, 0, 0};
    moverDedosSimultaneos(dedosDer, posInicialDer, anguloFinalDer, 5);

} else {
    // Retorno - Movimiento hacia atrás
    // Brazos
    moverServo(aperturaHombroIzquierda, 180, 1500);
    moverServo(aperturaHombroDerecha, 180, 1500);
    moverServo(rotarBicepIzquierda, 180, 6000);
    moverServo(rotarBicepDerecha, 180, 6000);

    // Dedos izquierdos
    Servo dedosIzq[] = {medioIzquierda, anularIzquierda,
meniqueIzquierda, indiceIzquierda, pulgarIzquierda};
    int posInicialIzq[] = {180, 0, 0, 180, 180};
    int anguloFinalIzq[] = {0, 180, 180, 0, 0};
    moverDedosSimultaneos(dedosIzq, posInicialIzq, anguloFinalIzq, 5);

    // Dedos derechos
    Servo dedosDer[] = {medioDerecha, anularDerecha, meniqueDerecha,
indiceDerecha, pulgarDerecha};
    int posInicialDer[] = {180, 180, 180, 0, 0};
    int anguloFinalDer[] = {0, 0, 0, 180, 180};
    moverDedosSimultaneos(dedosDer, posInicialDer, anguloFinalDer, 5);
}
estadoAccion1 = !estadoAccion1; // Alterna el estado para la próxima
vez

bloqueo = false;
break;

```

```

case '2':
    bloqueo = true;
    Serial.println("Ejecutando 2...");

    if(estadoMovimiento2) {
        // SEGUNDA PULSACIÓN - Solo mover rotar brazos
        moverParejaServos(bicepIzquierda, 0, 2500, bicepDerecha, 0, 2500);
        moverParejaServos(aperturaHombroIzquierda, 180, 1500,
aperturaHombroDerecha, 180, 1500);
    } else {
        // PRIMERA PULSACIÓN - Secuencia completa
        // Brazos arriba (simultáneo con tiempos independientes)
        moverParejaServos(rotarBrazoIzquierda, 0, 13000, rotarBrazoDerecha,
0, 12000);
        moverParejaServos(bicepIzquierda, 180, 2300, bicepDerecha, 180,
2000);
        delay(3000);

        // Brazos frente (simultáneo)
        moverParejaServos(rotarBrazoIzquierda, 180, 4000, rotarBrazoDerecha,
180, 4000);
        delay(3000);

        // Brazos lados (simultáneo con movimiento combinado)
        moverParejaServos(aperturaHombroIzquierda, 0, 2000,
aperturaHombroDerecha, 0, 2000);
        moverParejaServos(rotarBrazoIzquierda, 180, 6000, rotarBrazoDerecha,
180, 6000);
    }

    estadoMovimiento2 = !estadoMovimiento2;
    Serial.println("Movimiento completado");
    bloqueo = false;
    break;

case '3':
    bloqueo = true;
    Serial.println("Ejecutando 3...");

    //saludo a la izquierda
    moverServosOpuestos(180, 0, 4000);
    rotarBrazoDerecha.write(0);
    delay(6000);
    rotarBrazoDerecha.write(90);
    // Movimiento saludo derecho
    for(int i = 0; i < 3; i++) {
        bicepDerecha.write(0);

```

```

        delay(500);
        bicepDerecha.write(90);
        delay(100);
        bicepDerecha.write(180);
        delay(500);
        bicepDerecha.write(90);
        delay(100);
    }
    bicepDerecha.write(0);
    delay(1000);
    bicepDerecha.write(90);
    rotarBrazoDerecha.write(180);
    delay(4500);
    rotarBrazoDerecha.write(90);

    //saludo a la derecha
    moverServosOpuestos(0, 180, 7000);
    rotarBrazoIzquierda.write(0);
    delay(7000);
    rotarBrazoIzquierda.write(90);
    // Movimiento saludo izquierdo
    for(int i = 0; i < 3; i++) {
        bicepIzquierda.write(0);
        delay(500);
        bicepIzquierda.write(90);
        delay(100);
        bicepIzquierda.write(180);
        delay(500);
        bicepIzquierda.write(90);
        delay(100);
    }
    bicepIzquierda.write(0);
    delay(1000);
    bicepIzquierda.write(90);
    rotarBrazoIzquierda.write(180); // Movimiento opuesto al derecho
    delay(5000);
    rotarBrazoIzquierda.write(90);
    // volver
    moverServosOpuestos(180, 0, 3000);

    Serial.println("Movimiento completado");
    bloqueo = false;
    break;

case '4':
    bloqueo = true;
    contadorPresion4++;

```

```

if (contadorPresion4 == 1) {
    Serial.println("Ejecutando 4...");

    cuelloLadoLado.write(45); // Colocar cuello a 45°

    unsigned long tiempoInicio = millis();
    while (millis() - tiempoInicio < 1000) {
        rotarBrazoIzquierda.write(0);
        rotarBrazoDerecha.write(0);
    }
    rotarBrazoIzquierda.write(90);
    rotarBrazoDerecha.write(90);

    unsigned long tInicioMuneca = millis();
    unsigned long duracion = 2000;
    int inicioDerecha = 90;
    int finDerecha = 0;
    int inicioIzquierda = 90;
    int finIzquierda = 180;
    while (millis() - tInicioMuneca < duracion) {
        float progreso = float(millis() - tInicioMuneca) / duracion;
        int posDerecha = inicioDerecha + (finDerecha - inicioDerecha) *
progreso;
        int posIzquierda = inicioIzquierda + (finIzquierda -
inicioIzquierda) * progreso;
        munecaDerecha.write(posDerecha);
        munecaIzquierda.write(posIzquierda);
        delay(20);
    }
    munecaDerecha.write(0);
    munecaIzquierda.write(180);

    cuelloCentro.write(180);
    delay(500);
    cuelloCentro.write(90);

    // Ahora mover los dedos uno por uno con la boca y el cuello
    moverDedosSecuencial();

    Serial.println("Secuencia completada");

} else if (contadorPresion4 == 2) {
    Serial.println("Regresando...");

    regresarASuave();

    cuelloCentro.write(0);
    delay(500);

```

```
    cuelloCentro.write(90);

    rotarBrazoIzquierda.write(180);
    rotarBrazoDerecha.write(180);
    delay(1000);
    rotarBrazoIzquierda.write(90);
    rotarBrazoDerecha.write(90);

    Serial.println("Completado");

    contadorPresion4 = 0;
}
bloqueo = false;
break;
}
}
}
```