



**Escuela de Doctorado
y Estudios de Posgrado**
Universidad de La Laguna

Trabajo de Fin de Máster

**DISEÑO Y FABRICACIÓN DE UN BRAZO ROBÓTICO
CON FINES EDUCATIVOS**

Titulación: Máster en Ingeniería Industrial

Estudiante:
Óscar Gutiérrez Álvarez

Tutora: Silvia Alayón Miranda
Co-Tutor: Miqueas Zurisadai Fortes González

20 de Mayo de 2024

La publicación de este Trabajo Fin de Máster sólo implica que el estudiante ha obtenido al menos la nota mínima exigida para superar la asignatura correspondiente, no presupone que su contenido sea correcto, aunque si aplicable. En este sentido, la ULL no posee ningún tipo de responsabilidad hacia terceros por la aplicación total o parcial de los resultados obtenidos en este trabajo. También pone en conocimiento del lector que, según la ley de protección intelectual, los resultados son propiedad intelectual del alumno, siempre y cuando se haya procedido a los registros de propiedad intelectual o solicitud de patentes correspondientes con fecha anterior a su publicación.

IMPRESO DE AUTORIZACIÓN DEL TRABAJO DE FIN DE MÁSTER POR EL/LA TUTOR/A

Curso 2023/2024

D^a **Silvia Alayón Miranda**, con D.N.I. 43812596B, como tutora del estudiante D.
Óscar Gutiérrez Álvarez en el Trabajo de Fin de Máster titulado

DISEÑO Y FABRICACIÓN DE UN BRAZO ROBÓTICO CON FINES EDUCATIVOS,

da su autorización, acreditada por la firma electrónica de este documento, para la presentación y defensa de dicho proyecto, a la vez que confirma que el estudiante ha cumplido con los objetivos generales y particulares que lleva consigo la realización del mismo.

La Laguna, 19 de mayo de 2024.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de manera significativa a la realización de este Trabajo de Fin de Máster.

En primer lugar, deseo agradecer a mi tutora, Silvia Alayón, por su orientación experta, su paciencia y su apoyo constante a lo largo de este proyecto. Su guía fue fundamental para mi desarrollo académico y profesional.

Asimismo, quiero reconocer el invaluable aporte de mi cotutor, Miqueas Fortes. Su conocimiento y experiencia fueron fundamentales para enriquecer este trabajo y llevarlo a buen término.

A mis compañeros de trabajo, Augusto Hernández y Kevin Rosa, les agradezco profundamente por su colaboración y por compartir su experiencia en el campo de la electrónica. Su ayuda fue crucial para superar los desafíos técnicos que surgieron durante el proceso.

No puedo dejar de mencionar el apoyo incondicional de mi familia. Agradezco especialmente a mi pareja, Alejandra, por su constante ánimo, comprensión y apoyo inquebrantable. Su presencia y aliento fueron mi mayor motivación para perseverar hasta el final.

Finalmente, quiero expresar mi gratitud a todas las personas que, de una forma u otra, contribuyeron a este proyecto y a mi formación académica. Vuestra confianza y respaldo fueron esenciales en este camino hacia la culminación de este trabajo.

¡A todos ustedes, mi más profundo agradecimiento!

Resumen

Este Trabajo Fin de Máster titulado "DISEÑO Y FABRICACIÓN DE UN BRAZO ROBÓTICO CON FINES EDUCATIVOS" se centró en la creación de un brazo robótico con aplicaciones educativas que imita los utilizados en entornos industriales. El proyecto se desarrolló siguiendo una metodología organizada en tareas específicas dentro de paquetes de trabajo, lo cual permitió una gestión eficiente del tiempo y los recursos. Durante los 8 meses de desarrollo, se realizó un detallado listado de materiales y un análisis de costos, resultando en una inversión total de 10.461,40 €, con un precio de venta del prototipo de 183,04 € sin IGIC. Este prototipo se considera viable comercialmente con un umbral mínimo de ventas de 58 unidades.

El proyecto produjo planos detallados del diseño y un estudio mecánico que evaluó el rendimiento de las piezas impresas en 3D, determinando que la pieza más crítica, el hombro, opera al 12,94% de su capacidad máxima bajo la tensión de 4,523 MPa. Además, se desarrollaron y documentaron los planos electrónicos y los códigos de prueba necesarios para el funcionamiento del brazo robótico. Aunque el proyecto dura hasta la fase de prototipo, los resultados indican un avance significativo hacia la producción de un producto terminado y comercializable.

Se identificaron diversas mejoras para futuras versiones, como la adición de movimientos más complejos, comunicación por Bluetooth y Wifi, y la integración de actuadores y sensores adicionales. Estas mejoras están destinadas a optimizar el uso educativo del brazo robótico, facilitando la enseñanza de conceptos STEAM y ofreciendo una base sólida para futuros desarrollos en la fabricación de brazos robóticos educativos. Este trabajo ha logrado no solo un prototipo funcional, sino que también ha establecido las bases para mejoras continuas y aplicaciones prácticas en el ámbito educativo.

Abstract

This Master's Thesis titled "DESIGN AND MANUFACTURE OF AN EDUCATIONAL ROBOTIC ARM" focused on the creation of a robotic arm with educational applications that mimics those used in industrial environments. The project was developed following a methodology organized into specific tasks within work packages, enabling efficient management of time and resources. Over the 8-month development period, a detailed list of materials and a cost analysis were conducted, resulting in a total investment of 10.461,40 €, with a prototype selling price of 183,04€ excluding IGIC. This prototype is considered commercially viable with a minimum sales threshold of 58 units.

The project produced detailed design plans and a mechanical study that evaluated the performance of 3D-printed parts, determining that the most critical piece, the shoulder, operates at 12,94% of its maximum capacity under a stress of 4,523 MPa. Additionally, electronic schematics and the necessary test codes for the robotic arm's operation were developed and documented. Although the project extends only to the prototype phase, the results indicate significant progress towards the production of a finished and marketable product.

Various improvements for future versions were identified, such as the addition of more complex movements, Bluetooth and Wifi communication, and the integration of additional actuators and sensors. These enhancements are aimed at optimizing the educational use of the robotic arm, facilitating the teaching of STEAM concepts, and providing a solid foundation for future developments in the manufacture of educational robotic arms. This work has achieved not only a functional prototype but also laid the groundwork for continuous improvements and practical applications in the educational field.

Índice de contenidos

Agradecimientos	3
Resumen	4
Abstract	5
1. Introducción	8
2. Antecedentes	12
2.1. Historia de los brazos robóticos	12
2.2. Tipos de brazos robóticos	14
2.3. Robótica en el ámbito docente	18
2.4. Brazos robóticos educativos comerciales	19
2.5. Antecedentes en la construcción de brazos robóticos educativos	25
2.6. Contribución del presente TFM tras el estudio de antecedentes.	28
3. Herramientas utilizadas	30
3.1. Herramientas software	30
3.2. Herramientas Hardware:	36
4. Desarrollo del prototipo	40
4.1. Paquetes de trabajo	40
4.2. Cronograma	44
4.3. Diseño mecánico	45
4.3.1. Primer diseño	45
4.3.2. Segundo diseño	55
4.3.3. Tercer diseño	59
4.4. Fabricación mecánica	65
4.5. Diseño electrónico	71
4.5.1. Microcontrolador principal ATmega 328P	75
4.5.2. Alimentación del dispositivo	76
4.5.3. Control del motor paso a paso	79
4.5.4. Control de servomotores	82
4.5.5. Pantalla informativa	83
4.5.6. Interfaz de control del dispositivo	84
4.5.7. Multiplexado y comunicaciones	85
4.5.8. Esquema de la PCB Auxiliar	88
4.5.9. Conjunto de PCBs diseñadas	89
4.6. Fabricación electrónica	90
4.7. Diseño software	94
4.7.1. Códigos de prueba realizados	95
4.7.2. Cálculo de cinemática inversa	95
4.7.3. Código principal del dispositivo	98
4.8. Implementación software	102
5. Presupuesto	104
5.1. Piezas mecánicas	104
5.2. Componentes electrónicos	105

5.3. Mano de obra	107
5.4. Coste total y umbral de rentabilidad	108
6. Resultados	109
7. Conclusiones	111
8. Referencias	112
9. Glosario de términos	116
Relación de Anexos	119

1. Introducción

En el panorama actual de la robótica, tanto en Industria como en Educación, los brazos robóticos se han convertido en dispositivos destacados. Estos dispositivos, que combinan mecánica, electrónica, automatización y programación, están transformando no solo la forma en que se llevan a cabo algunas tareas industriales, sino también la manera en que se enseñan y se aprenden conceptos fundamentales de Ciencia, Tecnología, Ingeniería y Matemáticas (STEM). Es esencial preparar a las generaciones futuras para el mundo actual impulsado por la tecnología. Los brazos robóticos educativos son una respuesta a esta necesidad urgente, fomentando la educación STEM desde edades tempranas y ayudando a desarrollar habilidades críticas para el futuro.

En las aulas de muchos países los brazos robóticos educativos como “Braccio” de Arduino se han convertido en herramientas esenciales para enseñar conceptos de programación, cinemática y resolución de problemas. Estos robots permiten a los estudiantes interactuar con la tecnología de una manera práctica y atractiva, lo que facilita la comprensión de conceptos STEM. Además, los brazos robóticos educativos están siendo también utilizados en una variedad de proyectos, desde la construcción de pequeños robots autónomos hasta la simulación de líneas de montaje automatizadas (figura 1).



Figura 1 - Dobot Magician Kit Educativo para Brazo robótico Stem, versión educativa con Correa transportadora. Imagen extraída de [1].

A pesar de los beneficios evidentes, existen desafíos importantes. La implementación de brazos robóticos en entornos educativos a menudo requiere una inversión significativa en hardware y capacitación de docentes. Además, la accesibilidad a esta tecnología puede ser limitada en algunas áreas, por motivos económicos, lo que plantea desafíos de equidad en la educación STEM: los precios que suelen tener este tipo de productos oscilan entre 90 euros [2] (productos de baja

calidad, difíciles de utilizar sin una preparación previa) y 2.000 euros [3,4] (productos de mejor calidad y con instrucciones y manual para aprender a utilizarlos).

El uso de un Brazo Robótico Educativo como herramienta de enseñanza para niños y niñas en edad escolar presenta una serie de desafíos y oportunidades. Uno de los problemas principales a abordar, es la falta de comprensión y experiencia de los estudiantes en el campo de la robótica y la programación. Muchos niños y niñas no tienen acceso a recursos educativos que les permitan explorar estos conceptos de manera práctica y tangible.

Este problema se agrava por la creciente demanda de habilidades en STEM en el mercado laboral actual. La falta de exposición temprana a estas disciplinas puede limitar las oportunidades futuras de los niños y las niñas y su capacidad para competir en un mundo cada vez más tecnológico.

Además, la enseñanza de conceptos abstractos como la programación puede resultar desafiante para los niños y niñas, especialmente si no tienen la oportunidad de aplicar esos conceptos en situaciones concretas y relevantes para ellos. La falta de herramientas y recursos adecuados puede dificultar aún más el proceso de aprendizaje y desalentar el interés de los estudiantes en estas áreas.

Por lo tanto, el diseño e implementación de un Brazo Robótico Educativo que sea accesible, intuitivo y atractivo para los niños y niñas puede abordar este problema de manera efectiva. Al proporcionar a los estudiantes la oportunidad de interactuar con un dispositivo real y aprender a programarlo para realizar tareas específicas, se puede fomentar el desarrollo de habilidades cognitivas y técnicas importantes, como el pensamiento computacional, la resolución de problemas y la creatividad.

En este Trabajo Fin de Máster (TFM) se aborda el diseño y la implementación de un brazo robótico educativo funcional, y un estudio de su viabilidad económica, para estimar si puede ser una solución práctica para el problema anteriormente comentado sobre la accesibilidad a la tecnología.

Este TFM se ha desarrollado en colaboración con la empresa Digital Codesign, empresa con sede en el Instituto Tecnológico de Canarias (ITC) de Santa Cruz de Tenerife dedicada a la investigación, el prototipado electrónico y el desarrollo de productos educativos tecnológicos. En estos momentos, la empresa está desarrollando diferentes kits y entrenadores educativos, y desea incluir el brazo robótico que se desarrolle en este TFM en uno de estos kits.

Tras un análisis exhaustivo de las opciones disponibles en el mercado para el ámbito educativo, y teniendo en cuenta que el dispositivo debe ser muy accesible para el

usuario, es objeto de este TFM el diseño y la implementación de un brazo robótico que pueda ser fabricado con una impresora 3D convencional, bajo un enfoque DIY¹. Es requisito también que su fabricación no implique un gasto significativo de materiales, para no elevar su coste. Teniendo en cuenta todos estos aspectos, se han fijado las siguientes especificaciones: el brazo debe tener una longitud aproximada de 30 cm, para que tenga un alcance medio, con 5 grados de libertad, debe disponer de una pinza para manipular objetos, y ser capaz de levantar el peso suficiente para poder realizar ejercicios de prácticas. También será objetivo del TFM diseñarlo para que pueda ser utilizado en un nivel de Educación Secundaria, en torno a edades entre 12 y 16 años, facilitando así la adquisición de las competencias básicas relacionadas con el funcionamiento, construcción y programación de brazos robóticos a estas edades.

En la presente memoria se abordará el desarrollo de un brazo robótico educativo, desde sus antecedentes históricos hasta su implementación práctica. En primer lugar, se explorará la evolución y diversidad de los brazos robóticos, desde sus primeras manifestaciones hasta las aplicaciones contemporáneas, destacando su creciente importancia en el ámbito educativo como herramienta para fomentar el aprendizaje STEAM.

A continuación, se detallarán las herramientas utilizadas en el proceso de desarrollo, tanto a nivel de software como de hardware, para proporcionar una visión completa del entorno tecnológico empleado en el proyecto.

El desarrollo del prototipo se dividirá en distintos paquetes de trabajo, abordando aspectos clave como el diseño mecánico, electrónico y de software. Cada etapa del proceso será descrita en detalle, desde la conceptualización inicial hasta la implementación práctica, destacando los desafíos enfrentados y las soluciones propuestas.

Posteriormente, se examinará el presupuesto asociado al proyecto, desglosando los costos en partes mecánicas, electrónicas y de mano de obra. Se calculará el coste total del proyecto y se establecerá el umbral de rentabilidad necesario para considerarlo económicamente viable.

Los resultados obtenidos durante el desarrollo del prototipo serán presentados y analizados en profundidad, destacando los logros alcanzados y las áreas de mejora identificadas.

Finalmente, se expondrán las conclusiones derivadas del proyecto, junto con las implicaciones para futuras investigaciones en el campo de la robótica educativa. Se proporcionarán referencias bibliográficas, así como un glosario de términos técnicos

para facilitar la comprensión del contenido. Los anexos complementarán la información presentada en la memoria, incluyendo detalles adicionales sobre el diseño y las pruebas realizadas.

2. Antecedentes

En este capítulo se ofrecerá una visión general de los avances más recientes en brazos robóticos educativos, destacando su importancia en la educación STEM. El capítulo comienza con la introducción de los brazos robóticos desde un punto de vista histórico y funcional, se centra posteriormente en su uso en el ámbito de la educación, y termina con unas conclusiones que apoyan la viabilidad del desarrollo robótico planteado en el presente TFM.

2.1. Historia de los brazos robóticos

La historia de los brazos robóticos ha evolucionado de forma constante y rápida en las últimas décadas. En 1961 la compañía estadounidense Unimation marcó un hito al liderar el desarrollo del primer robot de transferencia programable. De aquí surgieron los primeros brazos industriales, utilizados en fábricas de automóviles, destinados a realizar tareas pesadas y repetitivas con movimientos simples. General Motors, una de las empresas pioneras en la Robótica Industrial, adquirió licencias de Unimation para introducir un brazo robótico a su cadena de producción (figura 2), encargado de una de las tareas más peligrosas para los operarios: el levantamiento de grandes piezas de metal caliente para sumergirlas en líquidos refrigerantes [5].

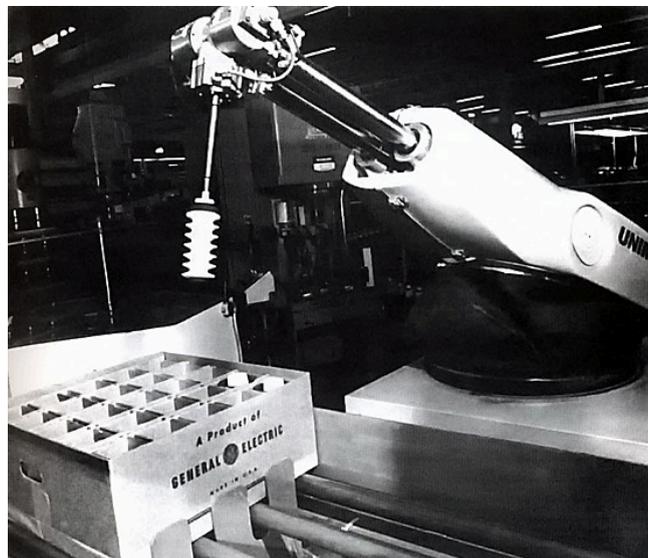


Figura 2 - Robot Unimate de General Motors en 1960. Imagen extraída de [5].

Posteriormente, en 1968, el grupo Kawasaki se destacó como el líder en robótica industrial en Japón, también mediante la adquisición de licencias de Unimation. A su vez, en 1969, Víctor Scheinman desarrolló el Stanford Arm, un robot ligero en forma de brazo multiprogramable para diversas tareas (figura 3) [5].



Figura 3 - Stanford Arm, desarrollado por Victor Scheinman en 1968. Imagen extraída de [5].

En la década de los 70, Unimation expandió su alcance a Europa, donde Suecia y la compañía Nokia lideraron la implantación de la Robótica Industrial. Se introdujeron robots de soldadura, capaces de ensamblar y soldar 110 automóviles por hora. En 1975, Europa vio nacer el ASEA IRB, un robot totalmente eléctrico que utilizaba tecnología Intel. En 1978, Unimation y Vicarm, la empresa de Víctor Scheinman, lanzaron al mercado el robot PUMA, utilizado en las líneas de ensamblaje de General Motors (figura 4) [6].



Figura 4 - Robot PUMA de Víctor Scheinman en 1978. Imagen extraída de [6].

Los años 80 marcan el comienzo de la Era Robótica. La fabricación y la venta de equipos robóticos aumentaron en un 80%, sentando las bases para la Robótica inteligente actual, con la incorporación de tecnología avanzada y la capacidad de toma de decisiones en tiempo real en colaboración con los seres humanos. En las décadas de los 90 y 2000, la robótica continuó creciendo de manera significativa. La globalización, impulsada por la internacionalización de las empresas, contribuyó a la creciente demanda y producción de bienes de consumo, hasta llegar a la actualidad, donde la robótica es ya una herramienta esencial para aumentar la productividad, reducir los costos y resolver problemas cotidianos, facilitando la vida de las personas [6].

2.2. Tipos de brazos robóticos

El término “brazo robótico” abarca una amplia categoría de mecanismos robóticos que se emplean en diversas aplicaciones. Aunque todos comparten características similares, existen ciertas particularidades que hacen que unos sean más adecuados para ciertas tareas que otros, como veremos a continuación [7-9].

Los **brazos articulados** son manipuladores robóticos de uso general con 5 o más articulaciones o grados de libertad (figura 5). Este término engloba varios subtipos de robots, como el robot de seis ejes, que cuenta con seis grados de libertad. Los brazos articulados abarcan una amplia variedad de robots utilizados en la industria, incluyendo los colaborativos.



Figura 5 - Brazo robótico articulado de precisión de entrenamiento para montar a caballo. Imagen extraída de [10].

Los **robots de seis ejes** son los brazos articulados más comunes y ampliamente utilizados en la industria actual (figura 6). Su versatilidad los convierte en una opción ideal para aplicaciones generales. Estos manipuladores son fácilmente identificables en entornos industriales.



Figura 6 - Brazo robótico con 6 ejes. Imagen extraída de [11].

Los **robots colaborativos** están diseñados específicamente para colaborar con humanos, con el objetivo de reducir riesgos. Aunque es un tipo de robot

relativamente nuevo, su presencia va aumentando a medida que se reconocen sus ventajas en diversas industrias (figura 7). Estos manipuladores permiten una mayor seguridad al alejar a los operarios humanos de tareas peligrosas.



Figura 7 - Robot colaborativo. Imagen extraída de [12].

Los **robots SCARA** son manipuladores con cumplimiento selectivo, lo que significa que tienen menos flexibilidad que los brazos articulados (figura 8). A pesar de estas limitaciones, presentan ventajas en ciertos escenarios.



Figura 8- Robot colaborativo. Imagen extraída de [13].

Los **robots cartesianos** son sistemas rígidos que se desplazan en un plano de coordenadas 3D mediante tres actuadores lineales (figura 9). Son ideales para aplicaciones más pequeñas y específicas.

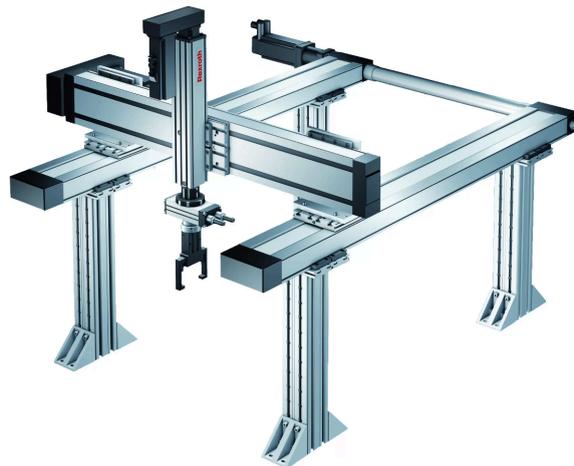


Figura 9 - Robot cartesiano. Imagen extraída de [14].

Los **brazos cilíndricos** están diseñados alrededor de un solo brazo que se mueve verticalmente y rota horizontalmente (figura 10). Compactos y eficientes, son ideales para tareas simples y pequeñas.



Figura 10 - Robot cilíndrico. Figura extraída de [15].

Los **robots esféricos/polares**, aunque menos comunes hoy en día, tienen un diseño similar a los cilíndricos, pero con un eje rotatorio adicional para movimientos verticales (figura 11).



Figura 11 - Robot polar. Imagen extraída de [16].

Diseñados para alta velocidad, los **robots paralelos/delta** destacan por su capacidad para alcanzar velocidades elevadas (figura 12). Son ideales para tareas ligeras y rápidas.



Figura 12 - Robot paralelo IRB 360 FlexPicker® series. Imagen extraída de [17].

Raros en entornos industriales, los **robots antropomórficos** tienen dos o más brazos y una apariencia más amigable (figura 13). Suelen utilizarse en colaboración estrecha con operarios humanos.

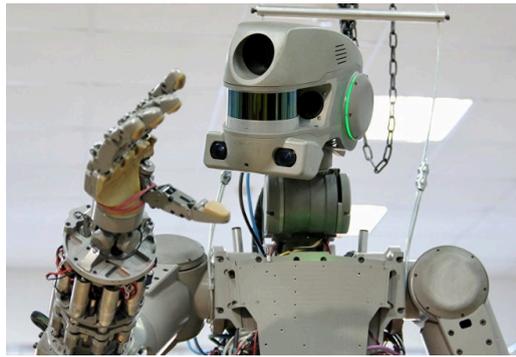


Figura 13 - Robot antropomórfico que conducirá el vehículo que enviará Rusia a la Luna. Imagen extraída de [18, 19].

Todos estos tipos de robots se aplican en una variedad de sectores: aeroespacial, automotriz, industria metalúrgica, alimentaria y de bebidas, farmacéutica, plásticos, ciencias de la salud, electrónica, etc. Las aplicaciones más comunes incluyen embalaje, paletización, manipulación de materiales, pintura, soldadura, montaje, inspección, corte y dispensación. La versatilidad de los manipuladores robóticos, con su equilibrio entre velocidad, capacidad de carga, alcance y precisión, los convierte en elecciones eficaces para diversas tareas en diferentes sectores. Su flexibilidad también permite a los fabricantes utilizarlos en aplicaciones más allá de su diseño original.

2.3. Robótica en el ámbito docente

La robótica puede ser una herramienta educativa con múltiples beneficios para el proceso de enseñanza y aprendizaje, entre los que destacan los siguientes [20-12]:

- **Aprendizaje interactivo:** el manejo de robots permite al estudiante aprender conceptos científicos, matemáticos y tecnológicos de forma práctica e interactiva.
- **Fomento de la Resolución de Problemas:** la programación y el control de robots requieren habilidades de resolución de problemas. Los estudiantes deben identificar y abordar desafíos específicos, lo que fomenta el pensamiento crítico y la creatividad.
- **Desarrollo de Habilidades de Programación:** la programación de robots es una introducción efectiva a la codificación. Los estudiantes pueden aprender conceptos de programación de manera práctica, desde lo básico hasta niveles más avanzados, dependiendo de la complejidad del robot.

- **Enfoque Práctico en STEM:** la robótica se integra perfectamente en las disciplinas STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), siendo un medio para aplicar y comprender estos principios de manera tangible.
- **Colaboración y Trabajo en Equipo:** trabajar en equipo para diseñar, programar y probar robots promueve habilidades de trabajo en equipo y comunicación.
- **Preparación para el Futuro:** puesto que la automatización y la robótica juegan un papel cada vez más importante en diversas industrias, la educación en robótica proporciona habilidades relevantes para el futuro mercado laboral.
- **Inclusión de la Diversidad de Estilos de Aprendizaje:** la robótica permite la inclusión de diferentes estilos de aprendizaje, beneficiando a los estudiantes que aprenden mejor a través de experiencias prácticas.
- **Motivación y Compromiso:** trabajar con robots puede aumentar la motivación y el compromiso de los estudiantes en el aula, debido a la emoción de ver sus creaciones en acción.
- **Exploración de Temas Complejos:** los robots pueden utilizarse para explorar temas complejos de manera más accesible. Por ejemplo, pueden simular procesos científicos o ayudar a visualizar conceptos matemáticos abstractos.
- **Adaptabilidad y Creatividad:** la robótica promueve la adaptabilidad y la creatividad al enfrentar a los estudiantes a desafíos que requieren soluciones únicas. Experimentar con diferentes enfoques para resolver problemas contribuye al desarrollo de estas habilidades.

Como se puede observar, la robótica en el ámbito docente no solo se trata de “enseñar sobre robots”, sino que también es una herramienta versátil para abordar una variedad de objetivos educativos. Proporciona un entorno de aprendizaje práctico y estimulante que puede preparar a los estudiantes para los desafíos tecnológicos del futuro.

2.4. Brazos robóticos educativos comerciales

En el mercado existen diversas propuestas educativas con brazos robóticos, dando lugar a una oferta variada, tanto en calidad como en funcionalidad y precio. A continuación, describimos algunos de los más populares.

El brazo robótico **Taidda** [23], modelo MG996R², es el más económico (figura 14). Está fabricado en aluminio y requiere el uso de una fuente de alimentación. Tiene 6 servomotores y una plancha de aluminio de 2mm de grosor para mejorar la estabilidad, los cojinetes de copa importados se utilizan en las juntas de dirección, lo que no solo hace que la dirección sea más flexible, sino que también permite que la dirección del mecanismo de dirección esté en el mismo centro, y el chasis tiene forma de disco, lo que hace que el manipulador gire hacia la izquierda y hacia la derecha de manera más flexible y suave. Este brazo está diseñado para la enseñanza a nivel universitario, pero no contiene instrucciones, lo que puede dificultar el montaje del mismo.



Figura 14 - Brazo robótico Taidda modelo MG996R. Imagen extraída de [23].

El brazo robótico **Youfang Smart** [2] es de acero inoxidable y funciona con un kit de arduino (figura 15). Tiene un tablero de control para el engranaje de dirección de 16 vías, adecuado para principiantes y utiliza un microcontrolador, el STM32³, el cual puede comunicarse con dispositivos externos utilizando varios protocolos de comunicación como SPI⁴, I2C⁵, UART⁶, etc., lo que permite a los desarrolladores incorporar funciones adicionales en sus diseños. Al igual que el brazo anterior, no contiene instrucciones de uso ni montaje, lo que puede dificultar su utilización.



Figura 15 - Brazo robótico YoufangSmart. Imagen extraída de [2].

El brazo robótico **Braccio** TinkerKit [24, 25] es un brazo montable muy popular fabricado por Arduino (figura 16). Es de plástico, funciona con una batería de 5V, tiene una rotación de 180° y opera con PWM⁷ analógico. Contiene dos tipos de servomotores, cuatro de ellos en las partes bajas del brazo, que tienen más potencia y sirven para desplazar el brazo, y dos más en el extremo superior del brazo para mover las pinzas. La empresa Arduino ofrece un contenido educativo propio orientado a estudiantes de secundaria y universitarios, por este motivo es de los más utilizados actualmente en Educación.



Figura 16 - Brazo robótico Braccio. Imagen extraída de [24].

El brazo robótico **RobotGeek** [26] es un brazo montable sin microprocesador, sólo contiene el hardware del brazo, con servos que tienen un giro de 180°, entre otras piezas para su ensamblaje (figura 17). A diferencia de los otros brazos, este cuenta

con 5 ejes de movimiento, puede controlarse con Geekduino y Arduino, e incluye instrucciones de montaje.

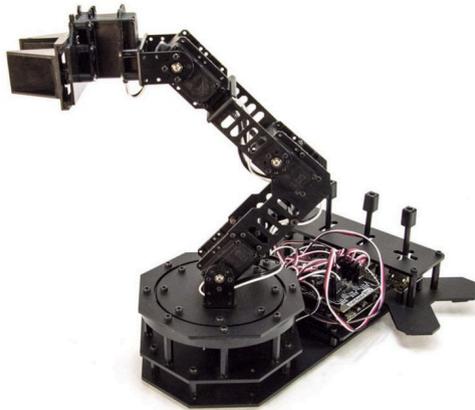


Figura 17 - Brazo robótico RobotGeek. Imagen extraída de [26].

El brazo **Minibot de WLkata** [3, 4] es el más completo de todos los analizados (figura 18). Incluye instrucciones de montaje y aprendizaje, y además es capaz de trabajar de manera colaborativa con otros brazos de la misma marca. Su gran inconveniente es que tiene un coste bastante elevado.



Figura 18 - Brazo robótico Minibot de WLkata. Imagen extraída de [3].

El brazo robótico inteligente ligero multifuncional **Dobot Magician Lite** [27] es un brazo diseñado para estudiantes y profesores, fácil de programar, utiliza su propio software de la propia marca (DobotStudio/DobotBlock), viene con un paquete completo de plan de estudios y tiene herramientas intercambiables para el extremo del mismo (figura 19). Es uno de los productos principales en el ecosistema de educación de inteligencia artificial personalizado K12 de DOBOT. Magician Lite disfruta de numerosos métodos de interacción de software y hardware e interfaces

de expansión para maximizar la libertad de creación de los estudiantes. A través de la construcción y el juego, los estudiantes pueden aprender cómo funciona la inteligencia artificial y la mecánica.



Figura 19 - Dobot Magician Lite. Imagen extraída de [27].

En la tabla 1 se muestran las características más importantes y el precio de cada brazo robótico descrito anteriormente.

Tabla 2.1 - Comparativa de brazos robóticos educativos existentes en el mercado.

Brazo robótico	Características	Precio
Youfang Smart [20]	Alimentación: Fuente eléctrica.	202,98 €
	Guía de Aprendizaje: No	
	Al funcionar con Arduino tiene un coste bajo, tiene potencial para la escalabilidad, pero los usuarios tienen que saber como programarlo.	
	Adecuado para principiantes: No	
	Cuenta con control remoto, garra de acero inoxidable, servos MG90 y 6 DOF.	
Taidda [2]	Alimentación: Fuente eléctrica.	91,83 €
	Guía de Aprendizaje: No	
	El manipulador 6 DOF es ampliamente utilizado en la enseñanza universitaria, la producción IDY.	
	Su sistema de potencia consta de 6 servomotores, que pueden realizar la demostración de agarre delantero y trasero, arriba y abajo y agarre izquierdo y derecho del robot.	
	Todos los accesorios de soporte están hechos de una placa de aluminio de 2 mm de espesor, que mejora efectivamente la estabilidad del robot.	

Brazo Robótico Braccio TinkerKit [22, 23]	Alimentación: Batería 5V y 5000 mA	269,81 €
	Guía de Aprendizaje: No	
	Señal de control: PWM analógico	
	Controlador: Arduino	
	Rango de rotación: 180°	
RobotGeek Core-Kit Robot Arm Snapper Gripper Arm [24]	Servos de 180°	324,99 €
	DOF: 5	
	Carga útil: 50 g	
	Alcance: 28,5 cm	
	Control: Geekduino/Arduino (no incluido)	
Kit Educativo de Mini Brazo Robótico Mirobot de WLkata [3, 4]	Ofrece varios métodos de control para usuarios principiantes y profesionales	1.936,04 €
	Ideal para aprender y demostrar la línea de producción	
	Se necesita adaptador para usuarios de la UE (se vende por separado)	
	Software para PC WLKATA Studio	
	Aplicación Móvil	
	Control Inalámbrico	
	Control Virtual 3D	
Dobot Magician Lite - Brazo robótico inteligente ligero multifuncional [27]	Voltaje: 100 V-240 V CA, 50/60 Hz.	1.301,26 €
	Potencia de entrada: 12 V ~ 5 A DC.	
	Potencia: 60 W máx.	
	Frecuencia principal: 168 MHz.	
	Comunicación: puerto serie virtual USB/puerto serie/Bluetooth	
	Lenguaje de Programación: MicroPython	
	Software: DobotStudio/DobotBlock	

Analizando las características de todos los modelos comerciales, se puede observar que los productos de menor precio suelen tener peor calidad y carecen de documentación que indique cómo trabajar con ellos. Otros datos destacables son que los brazos robóticos encontrados suelen tener 6 ejes de movimiento, un giro de 180° y están alimentados por una batería de 5V o por una fuente de alimentación externa.

2.5. Antecedentes en la construcción de brazos robóticos educativos

Durante el desarrollo de este TFM se ha realizado una búsqueda de información sobre la construcción de brazos robóticos educativos, explorando los niveles de tecnología necesarios para crear uno propio y evaluando su complejidad potencial. Se han encontrado un amplio conjunto de opciones disponibles para aquellos que deseen introducirse en la creación de su propio brazo robótico [28-34]. A continuación, se resumen las contribuciones que ofrecen más información.

En [28] se describe un proyecto de fabricación de un brazo robótico utilizando software libre y la plataforma de hardware Arduino. El objetivo de este brazo robótico es permitir a los estudiantes realizar diversas actividades prácticas, como el transporte de objetos o la selección de piezas defectuosas en una línea de producción. El artículo propone la implementación del brazo robótico como una alternativa a la simulación de procesos técnicos.

En [29] los autores comparten su experiencia en el diseño y construcción de un brazo robótico con fines educativos. Utilizando herramientas como Matlab, Simulink y SimMechanics, crearon un brazo robótico educativo de 6 grados de libertad, con una disposición de engranajes helicoidales para las articulaciones. El brazo puede trabajar con motores relativamente pequeños, no presenta retroceso y, por las propiedades de esta disposición, no consume energía cuando no realiza ningún movimiento. Los autores utilizaron un servomotor por cada articulación, con un codificador rotatorio absoluto, dos interruptores de límite para indicar los ángulos máximos y mínimos, y una disposición de tornillo sin fin (figura 20). El principal problema de esta solución es la baja velocidad del brazo debido a que las articulaciones del manipulador se accionan mediante una disposición de engranajes helicoidales, debido a la necesidad de añadir potencia a los motores.

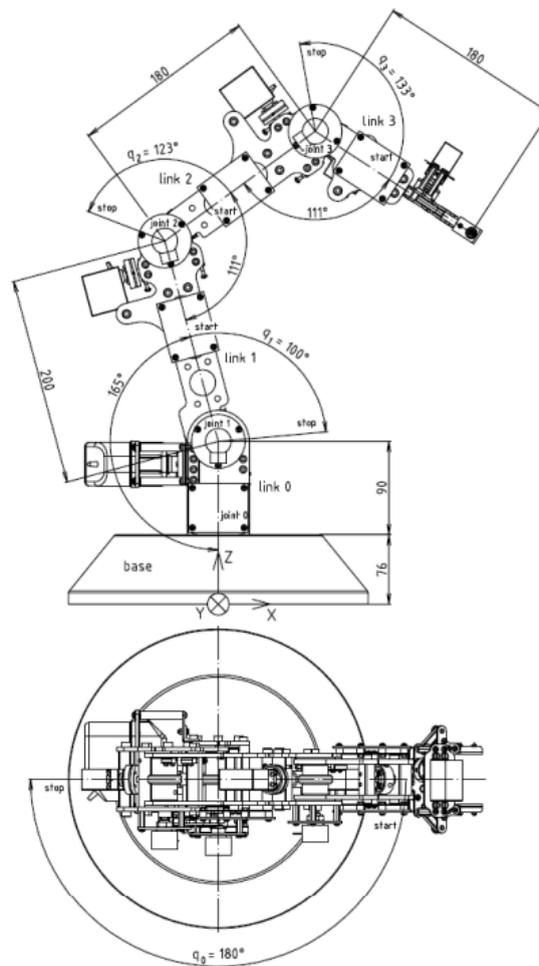


Figura 20 - Geometría del brazo robótico publicado en IEEE Xplore por un equipo de investigadores de la Facultad de Ingeniería Mecánica de la Universidad Tecnológica de Eslovaquia. Imagen extraída de [29].

En Project Hub se presenta un proyecto ingenioso. Utilizando componentes que se encuentran fácilmente en el mercado, y con una placa de desarrollo de tipo Arduino, se consigue controlar un brazo robótico mediante Bluetooth, haciendo uso de una aplicación móvil [30]. Con la ayuda de bibliotecas específicas para la programación, este proyecto demuestra que es completamente posible controlar un brazo robótico de esta manera, y ofrece una serie de ideas creativas para su implementación, lo que amplía las posibilidades en este campo (figura 21).

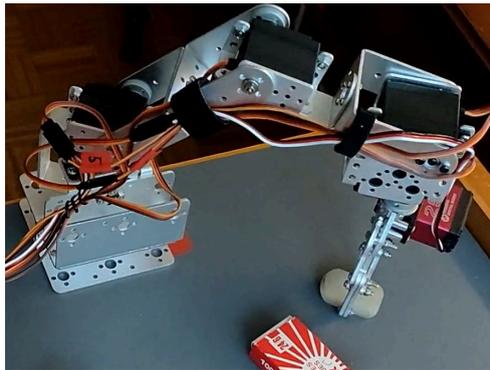


Figura 21 - Brazo robótico basado en Arduino que se encuentra en Project Hub. Imagen extraída de [30].

Existen otros proyectos que demuestran la posibilidad de fabricar un brazo robótico sin la necesidad de equipos sofisticados [31]. Estos proyectos hacen uso de tecnología accesible, como impresoras 3D FDM y componentes electrónicos sencillos, tales como micro servos de tipo SG90, servo motores de engranaje metálico de tipo MG996R y una placa de desarrollo Arduino Uno o equivalente, lo que demuestra que, a pesar de tratarse de una tarea compleja, la construcción de brazos robóticos es posible.



Figura 22 - Brazo robótico fabricado con impresora 3D publicado en [31].

En cuanto a la alimentación que pueda requerir el brazo robótico, existen diferentes formas de alimentarlos según el uso que se le vaya a dar. En [32] utilizan una batería de litio de 12V conectada a un convertidor BUCK⁸ reductor DC-DC, con un módulo capaz de suministrar una corriente máxima de 4A (figura 23). No obstante, existen otros planteamientos que hacen uso de fuentes de alimentación conectadas a red, o baterías LiPo de menor voltaje (3,7V) conectadas a módulos STEP-UP⁹.

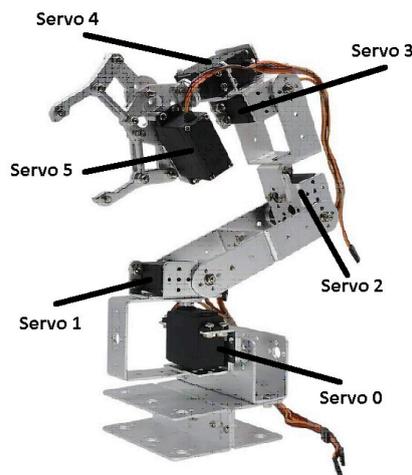


Figura 23 - Brazo robótico fabricado con piezas metálicas publicado en [32].

Por otra parte, los trabajos de [33, 34], proponen utilizar diferentes herramientas como *webcams*, aplicaciones para un mejor manejo del brazo, y la implementación de diferentes métodos de movimiento.

2.6. Contribución del presente TFM tras el estudio de antecedentes.

Una vez realizado el estudio de antecedentes se han llegado a diversas conclusiones importantes relacionadas con los brazos robóticos educativos y su impacto en el ámbito de la educación STEM, donde destacan las siguientes:

- El tema abordado en el TFM es de gran **relevancia en la Industria y en la Educación**. Los brazos robóticos combinan mecánica, electrónica, automatización y programación, y han transformado la forma en que se realizan tareas industriales y se enseñan conceptos STEM fundamentales.
- Existe una gran **variedad de opciones en el mercado**, con diferencias significativas en calidad, funcionalidad y precio. Los dispositivos de menor costo a menudo carecen de documentación adecuada, lo que puede dificultar su uso en entornos educativos, mientras que los dispositivos más costosos ofrecen recursos más complejos, aunque a un precio más elevado.
- Se ha podido demostrar la viabilidad de **construir brazos robóticos educativos personalizados**. Los recursos disponibles, como impresoras 3D y componentes electrónicos simples, hacen que la construcción de estos dispositivos sea accesible para aquellos interesados en explorar este campo sin la necesidad de equipos sofisticados.
- Existen **recursos de apoyo**, ya que se han identificado recursos valiosos que respaldan la construcción y programación de brazos robóticos educativos, como proyectos basados en Arduino y documentación detallada publicada en

fuentes académicas, lo que proporciona a los interesados una base sólida para sus propias investigaciones y desarrollos.

Por lo tanto, el propósito de este TFM es diseñar un brazo robótico educativo bien documentado y con un precio competitivo, útil para la enseñanza STEM a diferentes niveles educativos. El principal objetivo del TFM es diseñar un brazo funcional, un primer prototipo, y valorar la dificultad y el coste de su desarrollo.

3. Herramientas utilizadas

3.1. Herramientas software

En este apartado se comentan los diferentes programas utilizados durante el desarrollo del presente TFM, se explica por qué ha sido necesario su uso, y se describen brevemente sus principales funcionalidades.

- **Autodesk Fusion 360**

Autodesk Fusion 360 [35] es una herramienta CAD (diseño asistido por ordenador), modelado en 3D y CAM (fabricación asistida por ordenador) que se ha usado para el diseño mecánico de las piezas que componen el brazo. Este programa permite la creación de modelos tridimensionales complejos, ofrece la capacidad de parametrizar el modelo, e integra herramientas CAM, para fabricar directamente desde el diseño. Además, posibilita la colaboración de varios usuarios en la nube, es compatible con múltiples plataformas, y es económica.

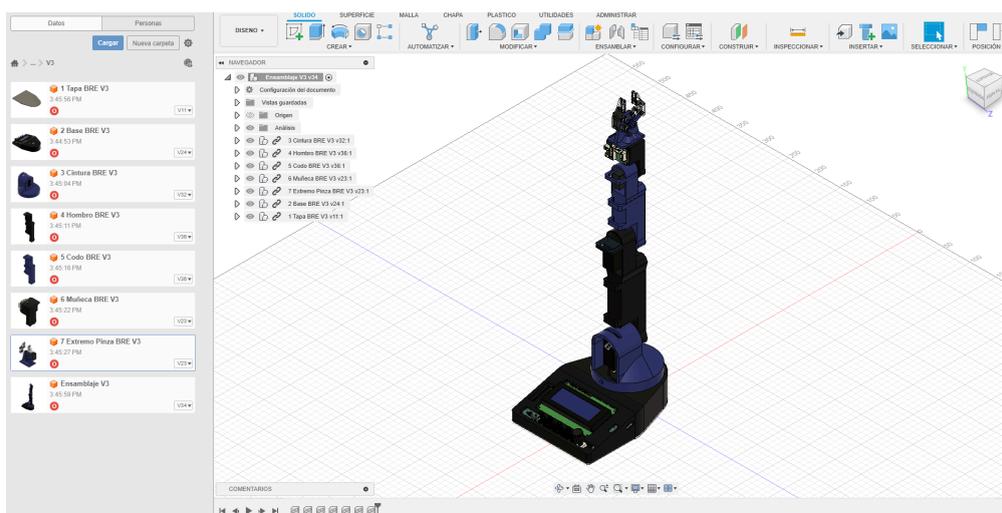


Figura 24 - Interfaz de Fusion 360.



Figura 25 - Logo de Fusion 360 [35].

- **Prusa Slicer**

El software **Prusa Slicer** [36], de código abierto y desarrollado por Prusa Research, permite la preparación de modelos 3D para fabricación aditiva en impresoras 3D

FDM¹⁰ y SLA¹¹. En este TFM este software CAM se ha utilizado para laminar las piezas diseñadas y fabricarlas en la impresora 3D FDM. Prusa Slicer permite a los usuarios ajustar una amplia variedad de parámetros de impresión: velocidad de impresión, densidad del relleno, temperatura del extrusor, las capas de los perímetros en la impresión, etc., y cuenta con herramientas inteligentes para la generación de soportes, lo que ayuda a minimizar el uso de material y facilita la eliminación de los soportes después de la impresión. Adicionalmente, ofrece la función de visualización de capas, que permite a los usuarios inspeccionar el modelo 3D y ver cómo se imprimirá capa a capa, lo que ayuda a identificar posibles problemas antes de imprimir. Es compatible con una gran variedad de formatos de archivo 3D, y existe una comunidad de apoyo a usuarios y desarrolladores.

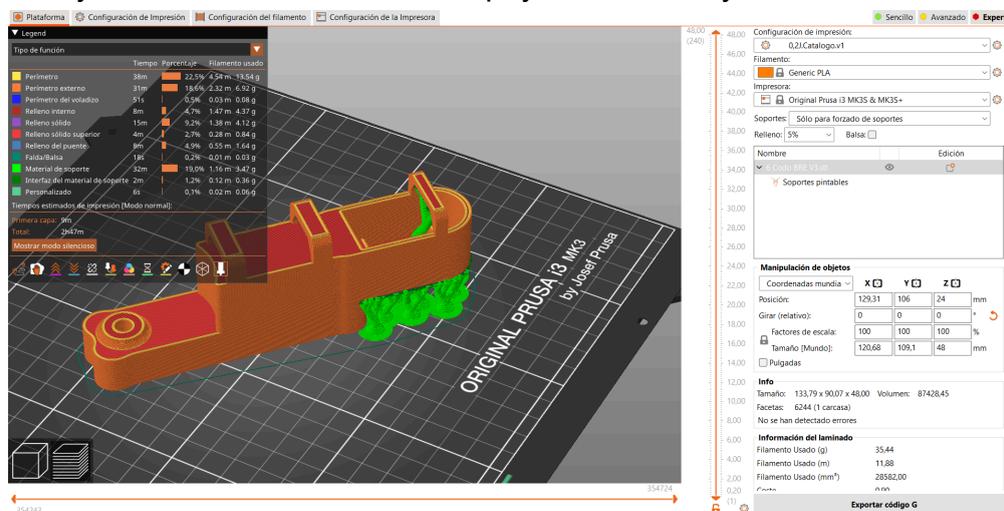


Figura 26 - Interfaz de Prusa Slicer.



Figura 27 - Logo de Prusa Slicer [36].

- **KiCad**

KiCad [37] es un software de diseño electrónico de código abierto que se utiliza para crear esquemas electrónicos y diseñar placas de circuito impreso (PCB). Para ello, cuenta con una amplia variedad de bibliotecas de componentes electrónicos estándar. Además, se puede integrar con otras herramientas de diseño electrónico y fabricación, lo que permite la exportación e importación de datos en varios formatos estándar, como Gerber, DXF y STEP. Es compatible con varias plataformas, y existe una comunidad de apoyo a usuarios y desarrolladores. Esta es la razón por la que se decidió utilizar este software para la creación de los esquemas electrónicos y el diseño de las diferentes PCBs del brazo.

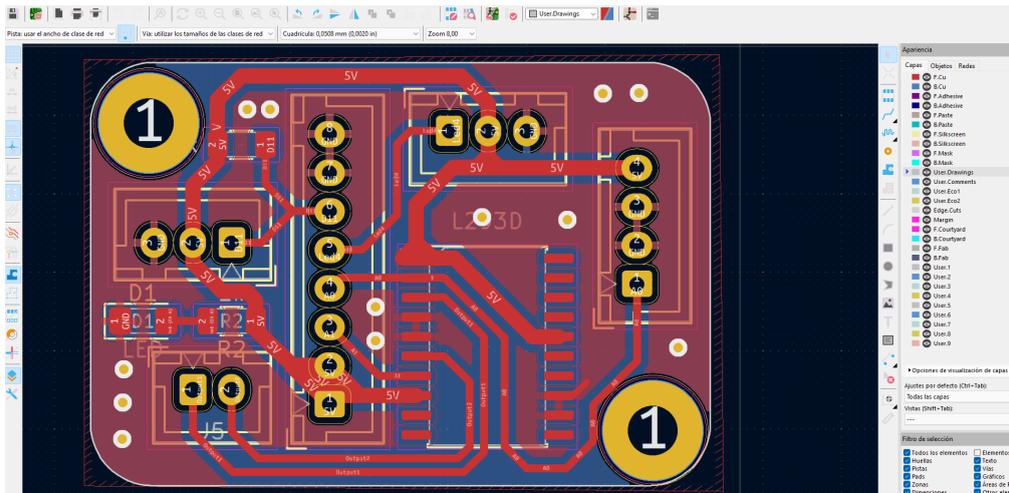


Figura 28 - Interfaz de KiCad.



Figura 29 - Logo de KiCad [37].

- **FlatCAM:**

FlatCAM [38] es un software de código abierto utilizado principalmente para preparar archivos de diseño de circuitos impresos (PCB) para el proceso de fabricación. Permite convertir archivos Gerber y de perforación NC (Numerical Control) en archivos G-code, que pueden ser utilizados por máquinas CNC (Control Numérico por Computadora) para fabricar los PCB. Entre sus características principales se incluyen la conversión de archivos Gerber y NC Drill a G-code, la visualización y edición de estos archivos, la generación de rutas de fresado, la simulación de rutas de herramientas para verificar el diseño antes de la fabricación y el soporte para diversas herramientas CNC. FlatCAM es especialmente útil para aficionados, ingenieros y pequeños fabricantes que buscan una solución económica y accesible para la creación de PCB, y al ser de código abierto, permite a los usuarios modificar y adaptar el software a sus necesidades específicas. En este proyecto se ha utilizado para hacer los preparativos de la fabricación de una de las placas diseñadas.

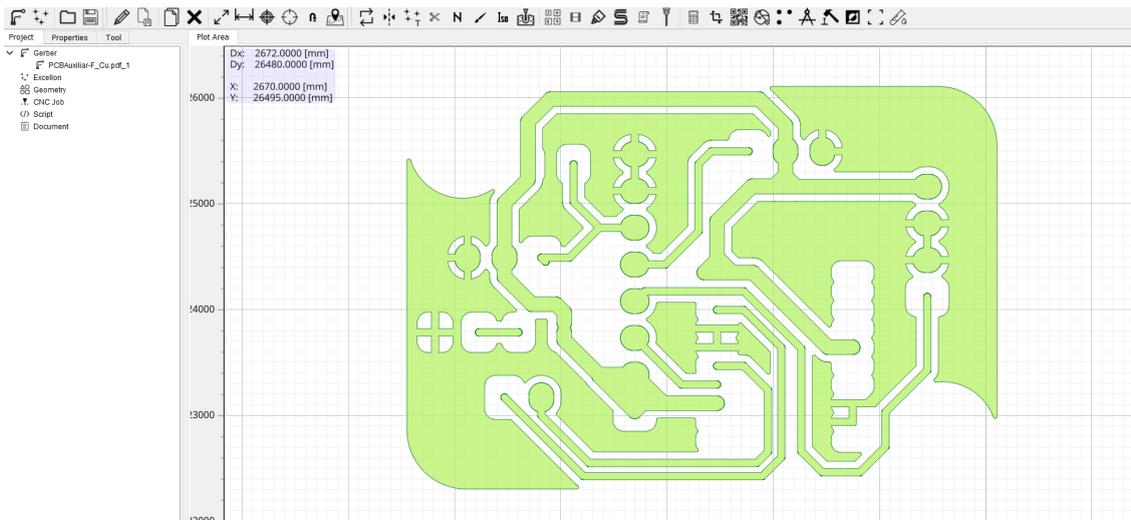


Figura 30 - Interfaz de FlatCAM

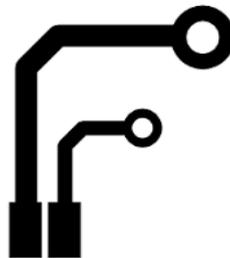


Figura 31 - Logo de FlatCAM [38]

- **RDWorks**

RDWorks [39] es un software de control y diseño utilizado principalmente en máquinas de corte y grabado láser fabricadas por Ruida Technology, una empresa china que produce controladores para una variedad de máquinas CNC, incluidas las máquinas láser. A través de su interfaz intuitiva, se pueden importar una variedad de formatos de archivo, controlar los parámetros de corte y grabado, y previsualizar el proceso de corte o grabado antes de enviar el trabajo a la máquina. En este TFM este programa se ha usado para realizar los diferentes cortes de las piezas que componen la pinza y la tapa de la base, y el grabado sobre una plancha de cobre para fabricar la PCB Auxiliar.

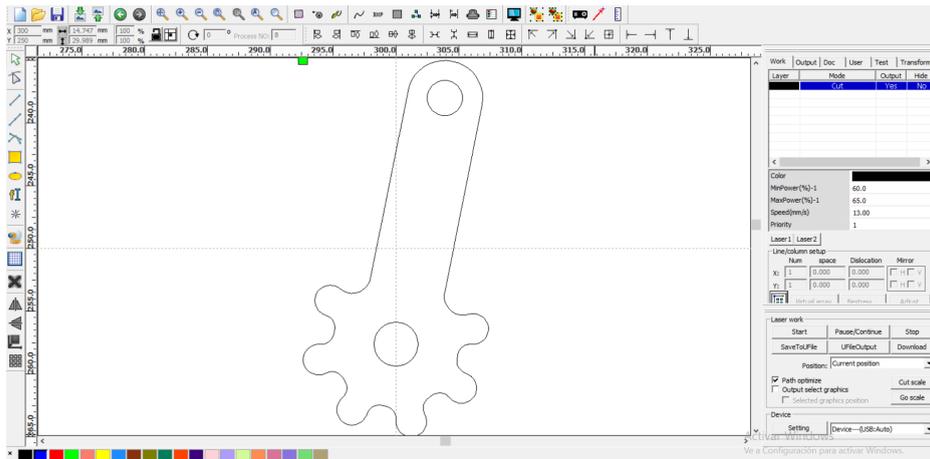


Figura 32 - Interfaz de RDWorks.



Figura 33 - Logo de RDWorks [39].

- **Arduino IDE:**

Otra herramienta utilizada es **ArduinoIDE** [40], un entorno de desarrollo de software para controladores Arduino, necesaria en este TFM puesto que se dispone de un Arduino ATmega328P integrado en PCB. Este entorno es fácil de usar, puesto que emplea una variante simplificada de C/C++, cuenta con una gran comunidad de usuarios activos y una amplia base de conocimientos en línea, es compatible con una amplia gama de placas de desarrollo Arduino, es compatible con varias plataformas, y es de código abierto y gratuito.

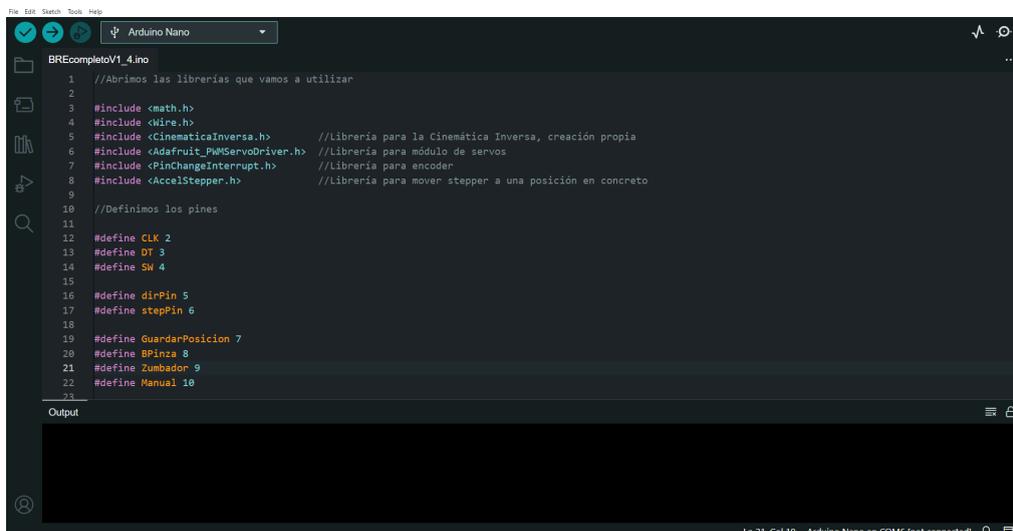


Figura 34 - Interfaz de Arduino IDE.



Figura 35 - Logo de Arduino IDE [40].

- **GitHub :**

El software GitHub [41] es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores trabajar juntos en proyectos de software, gestionar el código fuente, realizar un seguimiento de los cambios, y coordinar el trabajo de manera eficiente. Este software se ha utilizado en este TFM para organizar los diferentes archivos creados para este proyecto, contar con un registro de cambios y disponer de copias de seguridad en la nube.

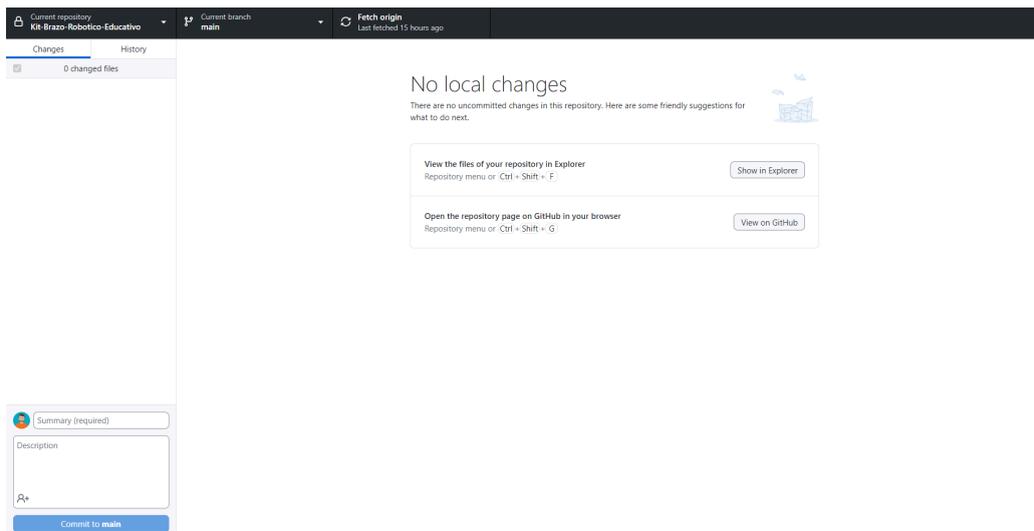


Figura 36 - Interfaz de GitHub.



Figura 37 - Logo de GitHub [41].

3.2. Herramientas Hardware:

En este apartado se comentan las diferentes herramientas hardware utilizadas durante el desarrollo del presente TFM, se explica por qué ha sido necesario su uso, y se describen brevemente sus principales características.

- **Impresora 3D de FDM:**

Una impresora 3D es una máquina que crea objetos tridimensionales capa por capa utilizando filamento de plástico fundido. Su utilización principal es el prototipado rápido, la fabricación de piezas personalizadas y producción a pequeña escala, por lo que es ideal para la fabricación de prototipos o piezas a un precio asequible. En este TFM se ha utilizado la impresora 3D Prusa i3 MK3S+ [42] para fabricar los componentes mecánicos diseñados con el programa Fusion 360. Las especificaciones de la impresora se muestran en la tabla 3.1.



Figura 38 - Impresora 3D Prusa i3 MK3S+ empleada.

Tabla 3.1 - Especificaciones técnicas Prusa MK3S+.

Volumen de Impresión	25 × 21 × 21 cm
Altura de capa	0,05 - 0,35 mm
Diámetro del filamento	1.75 mm
Materiales compatibles	PLA, PETG, ASA, ABS, PC, CPE, PVA/BVOH, PP (Polipropileno), Flex, Nylon
Velocidad máxima de	200 mm/s

desplazamiento	
Temperatura máxima de la boquilla	300 °C
Temperatura máxima de la base	120 °C
Extrusor	Accionamiento directo

- **Cortadora láser:**

Una cortadora láser es una máquina que utiliza un haz de luz láser para cortar o grabar diferentes materiales, como madera, acrílico, tela o papel. Se utiliza para cortar con precisión materiales en formas específicas, grabar diseños detallados o marcar superficies con gran exactitud. En este TFM se ha utilizado la cortadora láser de CO2 modelo K4060 de potencia 100W con controladora Ruida para la fabricación de las piezas que componen las pinzas del brazo y para el grabado de las pistas de la PCB Auxiliar como parte del proceso de fabricación por atacado químico. Sus características se recogen en la tabla 3.2.



Figura 39 - Cortadora láser empleada.

Tabla 3.2 - Especificaciones técnicas K4060 CO2 100W

Potencia del láser	100W
Espesor de corte	Según material: 0-12mm.
Resolución	4.000 dpi
Velocidad máxima de grabado	1.000 mm/s
Velocidad máxima de corte	500 mm/s
Tamaño de la base	60 x 40 cm

Precisión de posicionamiento	<0,01 mm
-------------------------------------	----------

- **Baño químico:**

El baño químico es una técnica de fabricación de PCBs, que implica la eliminación de cobre no deseado de placas de circuito impreso mediante la exposición a soluciones químicas, consiguiendo la separación y creación de rutas que conectan los diferentes componentes de una placa. Se utiliza para grabar o eliminar cobre, creando patrones de conexiones eléctricas necesarias para el circuito electrónico. En este TFM se ha utilizado esta técnica para marcar las pistas de la PCB Auxiliar.

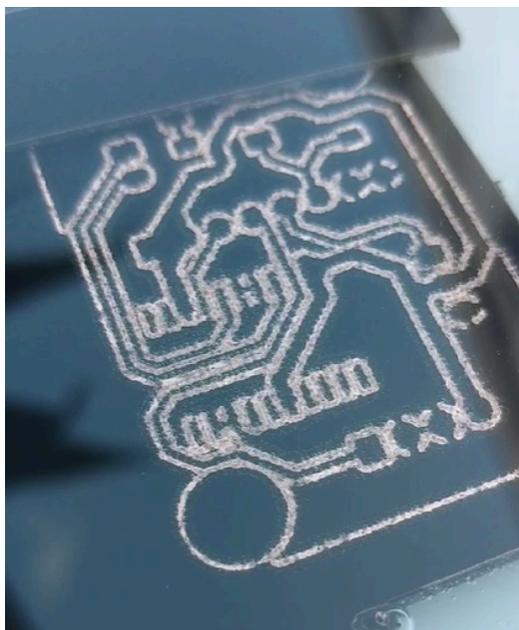


Figura 40 - PCB sumergida en el tanque de ataque químico.

- **Equipo de soldadura:**

En este TFM se ha utilizado una estación de soldadura con control de temperatura y puntas intercambiables, un soldador de precisión con punta y fina y delicada, alambre de soldadura de estaño, un extractor de humos con filtro de carbono activo para proteger, pinzas electroestáticas para prevenir descargas electrónicas y proteger los componentes, una lupa con luz y un microscopio especializado para soldar. Con dicho equipamiento se realizaron todas las soldaduras electrónicas de los diversos prototipos de PCBs. El equipo utilizado es del modelo Baku 878L [43].



Figura 41 - Equipo de soldadura.

4. Desarrollo del prototipo

El diseño y la implementación del brazo robot objeto de este TFM ha requerido la realización de múltiples tareas de diferentes disciplinas. Por motivos de claridad, en la presente memoria se presenta por separado el desarrollo mecánico, el electrónico y el de programación. En cada parte, se expone el objetivo, las decisiones de diseño tomadas para conseguirlo, y los cambios y las diferentes implementaciones que han dado lugar al prototipo final.

4.1. Paquetes de trabajo

El principal objetivo de este TFM es el diseño y la fabricación de un brazo robótico con fines educativos, pero que refleje de manera realista cómo son los brazos robóticos utilizados en los entornos industriales. Para alcanzar este objetivo, se ha realizado una planificación del trabajo siguiendo una estructura dividida por tareas, dentro de la actividad de la empresa que ha acogido la realización del TFM.

PT.0	Nombre del paquete de trabajo: PT.0 Coordinación y Gestión del proyecto. Responsables / Participantes: Estudiante, tutor y cotutor Inicio/Fin (meses): mes 1 / mes 8
Tarea	
T0.1 Investigación, previsión y vigilancia tecnológica El objetivo de esta tarea es la de buscar productos que puedan entrar en conflicto con el proyecto, de tal forma que podamos reaccionar a tiempo en la toma de decisiones respecto al desarrollo integral del brazo robótico. Consiste en la investigación, previsión y vigilancia tecnológica a través del estudio y consulta de publicaciones de artículos científicos y otros medios de difusión.	
T0.2 Gestión, control y supervisión del proyecto El objetivo de esta tarea es que el proyecto se lleve a cabo de acuerdo con el plan de trabajo y cronograma presentado, ajustado además al presupuesto previsto. Para ello, será necesario realizar de manera regular diferentes reuniones entre el estudiante, el tutor y el cotutor.	
T0.3 Gestión de adquisiciones y gastos Elaboración de las partidas económicas necesarias para llevar a cabo el proyecto. Adquisición y control del gasto en función del presupuesto asignado. Los gastos materiales necesarios se dividen en componentes electrónicos y mecánicos.	
T0.4 Preparación de informes técnicos y económicos Esta tarea contempla la elaboración de todo informe técnico y económico referido a la ejecución del proyecto, para todo su ciclo de vida.	

Entregables / Hitos

- D0.1** Lista de adquisiciones y gastos para la elaboración del proyecto.

PT.1

Nombre del paquete de trabajo: PT.1
 Identificación de las características del proyecto.
Responsables / Participantes: Estudiante, tutor y cotutor
Inicio/Fin (meses): mes 1 / mes 1

Tarea

T1.1 Desarrollo de ideas mediante técnicas creativas

Esta tarea trata de involucrar al estudiante en una dinámica de trabajo para desarrollar ideas creativas para el desarrollo del proyecto. Se utilizarán técnicas de *brainstorming*, *mind maps*, etc. para generar ideas implementables, desde el punto de vista técnico y didáctico, tomando como referencia el desarrollo del brazo robótico.

T1.2 Estudio y consulta de normativas

Estudio y análisis de las normativas vigentes que deberá cumplir el brazo robótico a desarrollar en materia de seguridad (uso de materiales certificados, protecciones de circuitos electrónicos, uso de baterías y fuentes de alimentación, etc.).

T1.3 Estudio y rentabilidad económica del proyecto

Se realizará un estudio de viabilidad para el brazo robótico a desarrollar propuesto por los mecanismos de generación de ideas, donde se tendrán en cuenta aspectos técnicos de diseño y coste de fabricación.

Esta tarea tiene como objetivo facilitar la selección de los conceptos más oportunos para el proyecto.

T1.4 Selección de ideas implementables

A partir del estudio de viabilidad, se seleccionará la propuesta final para el proyecto, cuyos criterios de selección valorarán la capacidad didáctica del concepto, la complejidad de implementación, la disponibilidad de los materiales a corto y largo plazo y la rentabilidad económica total que supone frente a otros diseños.

Entregables / Hitos

- D1.1** Informe con condicionantes técnicos y económicos para la elaboración del brazo robótico
 D1.2 Informe con la solución final propuesta, con medios técnicos y valoración económica.

PT.2	Nombre del paquete de trabajo: PT.2 Diseño mecánico Responsables / Participantes: Estudiante Inicio/Fin (meses): mes 2 / mes 7
Tarea	
T2.1 Diseño, simulación e implementación de modelos, estructura y sujeciones Esta tarea comprende todas las actividades de diseño, simulación e implementación de los modelos, estructuras y sujeciones que se encuentren en el brazo robótico. Diseño de modelos 3D con herramientas CAD 3D. Modelado de estructuras 3D y simulaciones frente a cargas externas.	
T2.2 Selección y verificación de materiales Se seleccionarán los materiales necesarios para la elaboración del brazo robótico, teniendo en cuenta la seguridad y disponibilidad a corto y largo plazo de los materiales.	
T2.3 Fabricación de prototipos mediante técnicas aditivas y sustractivas Se realizarán prototipos a partir de los modelos 3D diseñados utilizando técnicas de fabricación aditivas y sustractivas, de tal manera que se puedan validar las dimensiones y comportamiento mecánico de las partes que conforman el brazo robótico.	
T2.4 Verificación, pruebas y validación del comportamiento mecánico Se verificará que cada diseño mecánico cumple con las especificaciones de seguridad adecuadas y cuya normativa sea aplicable. Se realizarán pruebas del comportamiento mecánico en condiciones nominales de trabajo, así como una validación, si aplica, frente a los resultados simulados previamente.	
Entregables / Hitos	
<input type="checkbox"/> D2.1 Documentación técnica sobre los diseños mecánicos del brazo robótico.	

PT.3	Nombre del paquete de trabajo: PT.2 Diseño electrónico Responsables / Participantes: Estudiante Inicio/Fin (meses): mes 2 / mes 7
Tarea	
T3.1 Diseño, simulación e implementación de circuitos electrónicos Esta tarea comprende todas las actividades de diseño, simulación e implementación de los circuitos electrónicos que se encuentren en el brazo robótico. Diseño de circuitos de potencia para alimentación de circuitos electrónicos y microcontroladores. Diseño de los esquemas eléctricos de control de motores. Implementación de circuitos en placas de prototipado, mediciones y verificaciones.	
T3.2 Diseño y prototipado de placas de circuito impreso Esta tarea comprende todas las actividades de diseño y prototipado de PCBs que estarán presentes en el brazo robótico a desarrollar.	

Diseño de PCB con herramientas de software EDA (*Electronic Design Automation*).
Fabricación de placas de circuito impreso mediante procesos manuales.

T3.3 Verificación, pruebas y validación del comportamiento eléctrico

Se verificará que cada diseño eléctrico cumple con las especificaciones de seguridad adecuadas y cuya normativa sea aplicable. Se realizarán pruebas del comportamiento eléctrico en condiciones nominales de trabajo, pruebas de verificación de fuentes de ruido, así como una validación frente a los resultados simulados previamente.

Entregables / Hitos

- D3.1** Documentación técnica sobre los diseños electrónicos del brazo robótico.

PT.4

Nombre del paquete de trabajo: PT.4

Diseño software

Responsables / Participantes: Estudiante

Inicio/Fin (meses): mes 2 / mes 7

Tarea

T4.1 Programación del software e interfaz de usuario

Esta tarea contempla el desarrollo de un software a medida que servirá para proveer una interfaz de usuario para el control y monitorización en tiempo real del brazo robótico, así como para la configuración del mismo a partir de múltiples variables que permitan efectuar pruebas y simulaciones para infinitas situaciones posibles.

T4.2 Verificación, pruebas y validación del software

Se verificará que el software cumpla con las especificaciones de seguridad y funcionalidad adecuadas. Se realizarán pruebas unitarias del código correspondiente para asegurar un comportamiento adecuado y predecible.

Entregables / Hitos

- D4.1** Documentación del funcionamiento del brazo robótico.

PT.5

Nombre del paquete de trabajo: PT.5

Redacción

Responsables / Participantes: Estudiante,
tutor y cotutor

Inicio/Fin (meses): mes 7 / mes 8

Tarea

T5.1 Implementación de contenidos

Esta tarea consiste en la creación de un documento donde esté toda la información relacionada con el proyecto, desde una introducción, antecedentes, desarrollo del proyecto, presupuesto y unas conclusiones.

Responsable: Estudiante

T5.2 Verificación, pruebas y validación de los contenidos

Se verificará que el contenido dentro del documento de la T5.1 es correcta y cumple con el reglamento y normativa de presentación de la Universidad de La Laguna

Entregables / Hitos

- D5.1** Documento a presentar como trabajo fin de máster

PT.6

Nombre del paquete de trabajo: PT.6

Cierre del proyecto

Responsables / Participantes: Estudiante

Inicio/Fin (meses): mes 8 / mes 8

Tarea

T6.2 Elaboración de ponencia para la defensa del proyecto

Se elaborará el material de apoyo necesario para realizar la presentación de la defensa del proyecto, para mostrar las especificaciones técnicas del brazo robótico desarrollado, su valor para la comunidad educativa en el campo de la robótica, así como sus ventajas en proyectos de investigación e innovación de ámbito tecnológico y educativo.

Entregables / Hitos

- D6.1** Material de apoyo para la defensa del TFM

4.2. Cronograma

Se presenta, a continuación, un cronograma de actuación en el que se contempla cada paquete de trabajo junto a sus actividades correspondientes frente al plazo de ejecución del proyecto. La temporalización del cronograma comienza en la fecha de inicio prevista del proyecto, y finaliza 8 meses después de dicha fecha.

Tarea	Q1				Q2			
	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8
PT.0								
T0.1								
T0.2								
T0.3								
T0.4								
PT.1								
T1.1								
T1.2								
T1.3								

T1.4								
PT.2								
T2.1								
T2.2								
T2.3								
T2.4								
PT.3								
T3.1								
T3.2								
T3.3								
PT.4								
T4.1								
T4.3								
PT.5								
T5.1								
T5.2								
PT.6								
T6.1								

4.3. Diseño mecánico

Para alcanzar el objetivo principal mencionado anteriormente, las características mecánicas funcionales fijadas para el brazo son las siguientes:

- Brazo articulado con 5 grados de libertad.
- Debe disponer de una pinza manipuladora y ser compatible con otros efectores finales a desarrollar.
- Debe moverse con 5 servomotores y 1 motor paso a paso.
- Debe ser capaz de mover la pinza de un punto a otro y transportar algún objeto ligero.
- El diseño debe ser fabricable con una impresora 3D tipo FDM.

Es importante indicar que, desde el punto de vista mecánico, se han diseñado e implementado tres versiones del brazo hasta llegar al prototipo final, tal y como se describe a continuación.

4.3.1. Primer diseño

El primer diseño mecánico que se realizó sirvió para colocar adecuadamente los servomotores y comprobar su funcionamiento. Del mismo modo, este diseño

permitió validar también el comportamiento electrónico mediante diferentes pruebas electromecánicas.

4.3.1.1 Descripción del diseño

En este diseño se utilizaron cuatro servomotores MG90S¹² [44], en las partes superiores del brazo (pinza, mano, muñeca y codo), y un servomotor MG996R [45] en el codo. Estos servomotores concretos se eligieron porque son de tamaño pequeño y están compuestos por engranajes metálicos, lo que los hace más resistentes que otros modelos disponibles fabricados totalmente en plástico. Suponiendo que los servomotores trabajan a una tensión de 4,8V como mínimo, podemos obtener de su ficha técnica el par mínimo que ejercerán. Estos valores son de 1,8 kgf·cm [47] y 9,4 kgf·cm [48] respectivamente. También se seleccionó un motor paso a paso, el NEMA17 [46] (cuyas características principales se muestran en la Tabla 4.11), para conseguir que el brazo pudiera girar más de 180° con un control muy preciso.

A continuación, se describe el diseño 3D de esta primera versión del robot. En la figura 42 se muestra una vista de conjunto del diseño, y en la tabla 4.1 se enumeran las piezas que lo componen.

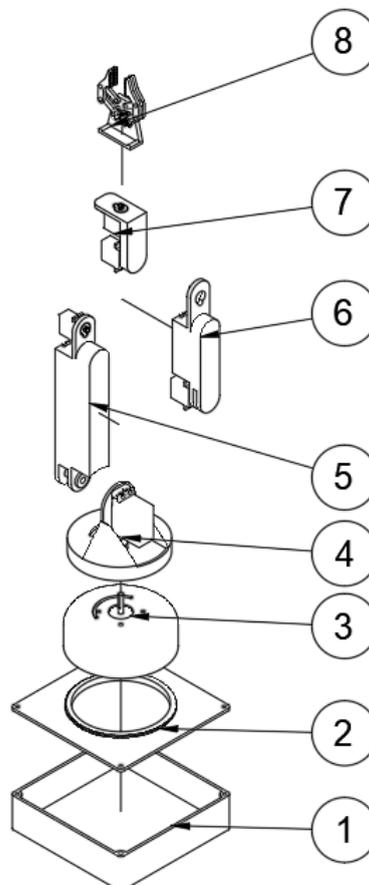


Figura 42 - Vista explosionada del primer diseño del Brazo Robótico Educativo.

Tabla 4.1- Piezas de la primera versión del brazo robótico.

Listado de piezas	
Elemento	Nombre de pieza
1	Base del brazo
2	Tapa
3	Base
4	Cintura
5	Hombro
6	Codo
7	Muñeca
8	Pinza

El elemento 1 (Tabla 4.1) es una **base** para el brazo (figura 43), donde irán los diferentes componentes electrónicos. Es una base cuadrada de dimensiones 15x15 cm y una altura 4 cm, capacidad suficiente para almacenar una protoboard y un Arduino Uno.



Figura 43 - Primera versión de la Base del brazo.

Esta base cuenta con una **tapa** que cubre los componentes electrónicos que se ubican en la base (figura 44), siendo esta el elemento 2 del brazo robótico (tabla 4.1). La tapa tiene un espesor de 3mm y una rosca para poder unirla con la base, haciendo uso de cuatro tornillos M3.

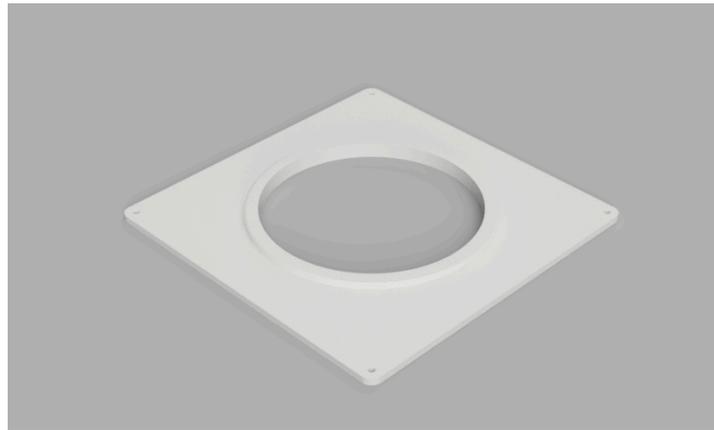


Figura 44 - Primera versión de la Tapa.

El elemento número 3 (tabla 4.1) es la **base** en la que se sujeta un **motor paso a paso** con cuatro tornillos M3 (figura 45). Presenta una altura de 4cm, para poder cubrir totalmente el motor paso a paso. Cuenta con un hueco que permite el paso de los cables del resto del dispositivo, de forma que puedan conectarse tanto los servomotores como futuros efectores finales.



Figura 45 - Primera versión de la Base.

La **cintura** es la parte que permite realizar el movimiento de todo el brazo respecto de su base, ya que establece una unión con el motor paso a paso para realizar los giros (figura 46). En este elemento se incorpora un servomotor MG996R, encargado del movimiento del hombro. La pieza tiene forma de una circunferencia de 10 cm de diámetro, cubre el motor paso a paso y contiene el hueco por donde se introducen los cables de los servomotores a la base.

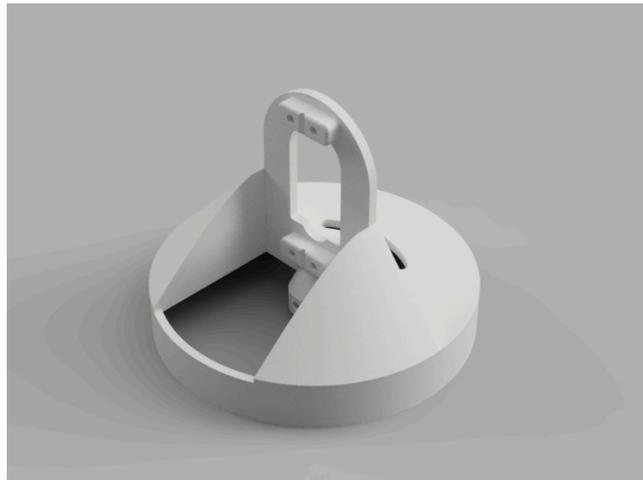


Figura 46 - Primera versión de la Cintura.

El elemento número cinco (tabla 4.1) es el **hombro** (figura 47), que es la parte que realiza el movimiento entre la *cadera* y el *codo*. Su longitud, medida desde el eje del servomotor del hombro al eje del servomotor del codo, es de 18 cm, con unas dimensiones totales de 4,7 x 2,6 cm a lo largo del hombro. También cuenta con un hueco para poder pasar los cables de los servomotores.



Figura 47 - Primera versión del Hombro.

El **codo**, el elemento número seis del conjunto del brazo, sirve de nexo entre el *hombro* y la *muñeca* (figura 48). Tiene una longitud desde el eje del servomotor del hombro al eje del servomotor del codo de 14 cm, con unas dimensiones de 3,8 x 2,6 cm a lo largo del codo. También cuenta con un hueco por donde poder pasar los cables de los servomotores, ubicado en el lado donde se posiciona el servomotor de la muñeca.



Figura 48 - Primera versión del Codo.

El séptimo elemento es la **muñeca** (figura 49), que permite el movimiento de la *pinza* respecto al *codo*. Tiene una longitud desde el eje del servomotor que lo mueve al extremo de la *muñeca* de 6 cm. Presenta un grosor de 1 x 2,6 cm en el servomotor que lo mueve y de 0,4 x 2,6 cm en el servomotor que tiene atornillado.



Figura 49 - Primera versión de la Muñeca.

El último elemento, la **pinza**, es el efector final (figura 50). Contiene diferentes partes fabricadas en metacrilato con corte láser, y otras fabricadas mediante impresión 3D FDM. Cuando la pinza se cierra, deja un espacio de 8 mm por seguridad. Tiene una longitud desde su base de 8 cm y una apertura máxima de 2,9 cm.

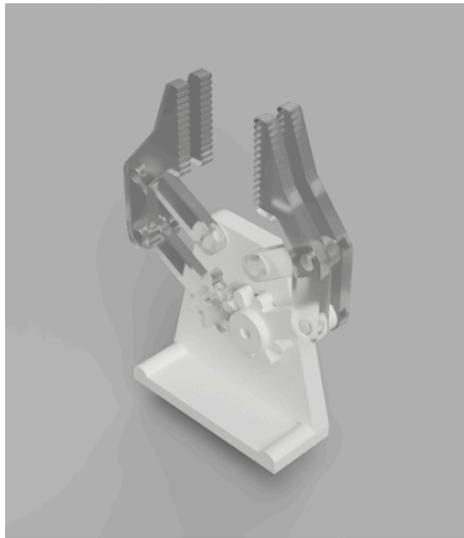


Figura 50 - Primera versión de la Pinza.

Una vez se han unido todos estos componentes mecánicos se tendría el resultado mostrado en el **ensamblaje** mostrado en la figura 51. Se cuenta, por tanto, con un brazo que presenta una longitud desde el hombro hasta la punta de la pinza de aproximadamente 40 cm y un peso aproximado de 450g, considerando únicamente el peso de los componentes fabricados en la impresora 3D (cuyo peso es estimado por el software PrusaSlicer), y el peso de los servomotores (cuyo peso se extrae de sus respectivos datasheet).

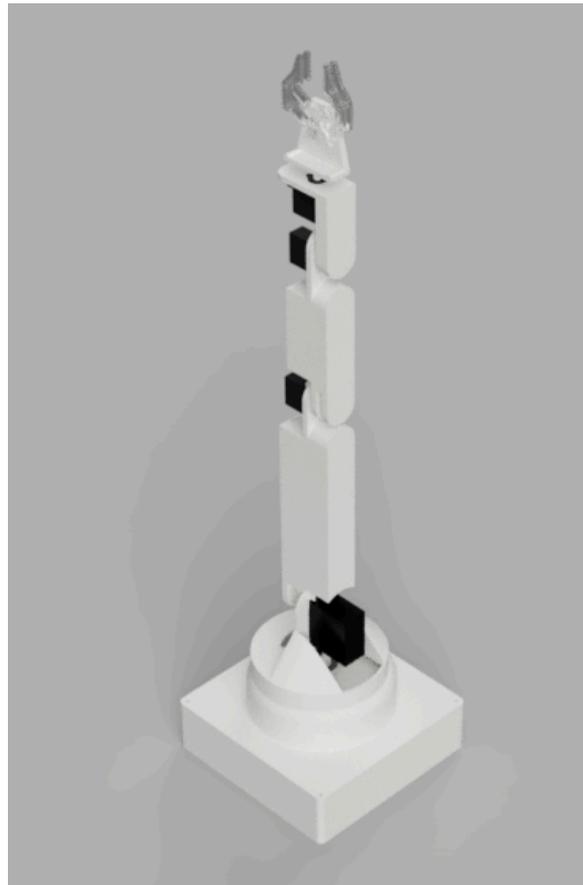


Figura 51 - Primera versión del Brazo Robótico Educativo ensamblado.

4.3.1.2 Funcionamiento del diseño

Una vez fabricado el primer diseño mecánico completo, se procedió a realizar un estudio de funcionamiento de cada actuador del dispositivo, analizando el momento máximo a soportar por cada motor y su porcentaje de trabajo. A continuación, se detalla la metodología de cálculo empleada, utilizando como ejemplo el primer elemento a comprobar: el servomotor que mueve el *hombro*, situado en la *cintura*.

En primer lugar, teniendo en cuenta que no se hace uso de piezas plásticas macizas, y que las piezas se fabrican mediante impresión 3D FDM, es necesario considerar el porcentaje de relleno, su patrón y orientación. Por dicho motivo, se recurre al software CAM, para obtener de forma rápida y precisa la masa del conjunto a fabricar y que debe ser movida por el actuador. Sumando las masas del resto de accesorios que acompañan a la pieza impresa se obtiene la masa total que debe soportar el actuador.

El cálculo se basa en la obtención del par o momento generado en el peor caso del brazo. Esta situación se dará con el brazo totalmente extendido y paralelo al plano XY. Por tanto, se obtendrá un sumatorio de los momentos generados por cada

conjunto pieza-actuador, considerando para ello que la fuerza se aplica en el centro de masas de dicho elemento, respecto al eje del servomotor a estudiar. Este centro de masas se obtiene utilizando el software de diseño y simulación Fusion 360.

Tomando el caso de ejemplo del *hombro*, el conjunto de piezas a evaluar son las siguientes:

- Hombro y su servomotor
- Codo y su servomotor
- Muñeca y su servomotor
- Pinza y su servomotor.

Se procede a continuación a obtener las masas, los centros de masas de cada conjunto y el par medido desde el centro de gravedad, como se muestra en la tabla 4.2.

Tabla 4.2 - Estudio de carga del servomotor que mueve el hombro

Elemento	Masa del actuador (g)	Elemento impreso (g)	Masa total (g)	Distancia CG (mm)	Momento (kgf·cm)
Hombro	13,4	51,69	65,09	86,78	0,565
Codo	13,4	29,63	43,03	233,17	1,003
Muñeca	13,4	11,82	25,22	321,65	0,811
Pinza	13,4	9,5	22,9	367,59	0,842

Se realiza el sumatorio de los momentos reflejados en la tabla 4.2:

$$\Sigma M = 0,565 + 1,003 + 0,811 + 0,842 = 3,221 \text{ kgf} \cdot \text{cm}$$

Se incluye un coeficiente de multiplicación del 50% con el que se considerará el efecto producido por cables, tornillería y demás accesorios del dispositivo, así como posteriormente, un coeficiente de seguridad del 25%. Considerando ambos factores, el par máximo obtenido es el siguiente:

$$M_{\text{máx}} = 3,221 \cdot (1 + 0,5 + 0,25) = 5,637 \text{ kgf} \cdot \text{cm}$$

Con dicho valor, se procede a consultar el par máximo admitido por el actuador. En este caso, se trata de un servomotor MG996R, que admite un momento máximo de 9,4 kgf·cm [42]. Se calcula el porcentaje de trabajo, obteniendo el siguiente valor:

$$\%_{Trabajo} = \frac{M_{m\acute{a}x}}{M_{m\acute{a}x\ MG996R}} = \frac{5,637\ kgf \cdot cm}{9,4\ kgf \cdot cm} = 59,97\%$$

Se comprueba, por tanto, que el servomotor seleccionado, debería ser apto para mover el brazo. Dicho procedimiento de cálculo se repite para cada elemento, obteniendo los resultados de la tabla 4.3.

Tabla 4.3 - Carga soportada por los servomotores en la primera versión

Servo de estudio	Momento a soportar (kgf · cm)	Porcentaje de trabajo (%)
Hombro	5,637	59,97%
Codo	1,772	98,46%
Muñeca	0,408	22,65%

Con el objetivo de comprobar de forma empírica el correcto funcionamiento del dispositivo, se fabricó y montó en su totalidad el brazo robótico, añadiendo todos los elementos necesarios como cableado, tornillería etc. Se posicionó el brazo en su posición más extendida horizontalmente, es decir, en el peor caso, y se procedió a ejecutar movimientos continuados del mismo. En este punto se observó que en la posición más extendida el comportamiento no es el estimado, ya que no es posible que el brazo recupere su posición. Esta situación demostró que era necesario realizar un ajuste del diseño, principalmente en la zona del *codo*, que es donde el brazo no era capaz de moverse desde la posición más extendida.

La deficiencia encontrada en el servo del *codo* no fue el resultado esperado, ya que los cálculos realizados indicaban que, a pesar de encontrarse en el límite de trabajo, podría moverse. Por este motivo, se decidió cambiar el servo que va colocado en el codo por el modelo MG996R, de mayor par motor. Este cambio hizo necesario introducir modificaciones en el brazo, dando lugar la segunda versión del diseño.



Figura 52 - Prueba realizada a la primera versión del brazo.

4.3.2. Segundo diseño

Como se ha explicado en el anterior apartado, a causa de las deficiencias detectadas en el primer diseño, fue necesario emplear un actuador con un mayor par motor en la zona del codo. Por este motivo se sustituyó el micro servo por un servomotor MG996R, con mayor peso, pero con un par de 9,4 kgf·cm (frente a los 1,8 kgf·cm del anterior). Este cambio de servomotor implicó realizar otros cambios en el hombro, para disponer de un adecuado emplazamiento para este servo de mayor volumen. Se aprovechó también este segundo diseño para optimizar las piezas que componían el *hombro*, el *codo* y la *muñeca*, para conseguir un diseño más ligero y resistente. A continuación, se detallarán bien todos estos cambios.

4.3.2.1 Descripción del diseño

Las piezas del primer diseño del brazo que se vieron modificadas en el segundo diseño fueron el *hombro*, el *codo* y la *muñeca*.

Con respecto al **hombro**, su principal modificación fue el cambio del servomotor MG90S al MG996R, por lo que fue necesario aumentar el tamaño en la parte superior de este (figura 53). A su vez, otro de los cambios realizados fue el aumento del tamaño del hueco donde pasarían los cables de los servomotores. Al realizar estos cambios se consiguió entre otras cosas, disminuir ligeramente el peso de la pieza, permitiendo alojar el nuevo servomotor manteniendo la distancia entre ejes de giro.



Figura 53 - Segunda versión del hombro adaptada al nuevo servomotor.

En el **codo** tuvo que modificarse el hueco por donde encajaba el eje del anterior servomotor para adaptarse al servo MG996R (figura 54). Al igual que en el hombro, se aumentó el tamaño del hueco por donde pasan los cables de los servomotores, manteniendo la misma distancia entre los ejes de giro de la pieza, debido a que la distancia entre las cabezas de los servomotores y la sección del cuerpo se mantiene. Esto provoca una pequeña reducción del peso de la pieza.



Figura 54 - Segunda versión del codo con modificación del hueco para el eje del MG996R.

La **muñeca** fue modificada a pesar de no observarse ningún problema en la prueba mecánica (figura 55). Sin embargo durante el montaje se notó cierta dificultad a la hora de atornillar el servomotor, al encontrarse uno de los tornillos pegado al cuerpo de la pieza. Por ello se decidió girar 90° la posición del servomotor para dejar libres dichas uniones y facilitar su montaje. Al realizar estos cambios, además de

conseguir la mejora de facilitar el montaje del brazo, también se redujo, al igual que en los demás casos, el peso de la pieza.



Figura 55 - Segunda versión de la muñeca para facilitar el montaje del actuador.

Una vez se han unido los componentes mecánicos modificados se tendría el resultado mostrado en el **ensamblaje** de la figura 56. En este segundo diseño el brazo tiene una longitud desde el hombro hasta la punta de la pinza igual que la versión anterior con una mejora en cuanto a su masa. Teniendo en cuenta únicamente con los componentes fabricados en la impresora 3D FDM y los servomotores, la nueva masa es de aproximadamente 426 g. Se logra, por tanto, una reducción de 24 g respecto al diseño original.

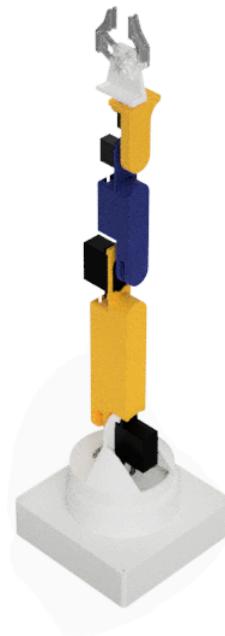


Figura 56 - Segunda versión del Brazo Robótico Educativo.

4.3.2.2 Funcionamiento del diseño

Tras estas modificaciones, para estimar el funcionamiento de este segundo diseño, se repitieron los cálculos realizados con el diseño 1, pero cambiando los valores del centro de masa de la pieza diseñada y de los servomotores, y manteniendo el mismo coeficiente de seguridad y de multiplicación. El resultado se muestra en la tabla 4.4.

Tabla 4.4 -Carga soportada por los servomotores en la segunda versión del diseño.

Servo	Momento a soportar (kgf · cm)	Porcentaje de trabajo
Hombro	6,973	74,18%
Codo	1,915	20,37%
Muñeca	0,470	26,1%

Observando estos resultados, se comprueba que el trabajo realizado por el servomotor que realiza el movimiento del *codo* se reduce significativamente, pasando del 98,46% (Tabla 4.3) al 20,37% (Tabla 4.4). También se observa un aumento del trabajo realizado por el servomotor que mueve la *muñeca*, incrementándose esta del 22,65% (Tabla 4.3) al 26,1% (Tabla 4.4), probablemente debido a la separación de los centros de masas de los diferentes componentes respecto a sus servomotores, dando un resultado más exacto, pero manteniendo el mismo funcionamiento. Como es de esperar, al cambiar el servomotor del *codo* de un MG90S a un MG996R, un servomotor más grande y pesado, el porcentaje de trabajo es mayor, pasando esta de un 59,97% (Tabla 4.3) a un 74,18% (Tabla 4.4), estando este valor dentro de las posibilidades de movimiento del servomotor.

Tras estos cálculos se fabricaron todas las piezas modificadas para proceder nuevamente al montaje del brazo y realizar una nueva verificación del funcionamiento. Igual que en el caso anterior, se colocó el dispositivo en su posición más extendida y se procedió a realizar diferentes movimientos. En esta ocasión se observó que el comportamiento mecánico es adecuado, no presentando ningún problema para realizar todos los movimientos con soltura.



Figura 57 - Prueba realizada a la segunda versión del brazo.

Durante la fase de montaje se descubrieron algunas mejoras funcionales que podrían realizarse y que ofrecerían un acabado de mayor calidad al dispositivo. Estas mejoras, que se detallarán en el siguiente subapartado, consistieron principalmente en la ocultación de los servomotores, la reducción de la longitud de algunos elementos para disminuir su masa, y la modificación de la forma de algunas de las piezas. Además, al disponerse ya de la dimensiones de la PCB, se puede diseñar de forma más adecuada la base que alberga la electrónica que controla el dispositivo educativo.

4.3.3. Tercer diseño

4.3.3.1 Descripción del diseño

Con las revisiones realizadas en el segundo diseño, se tomaron diferentes decisiones para fabricar el tercer (y último) diseño. Entre ellas, se unificaron los elementos uno (base del brazo) y tres (base) en un único elemento equivalente al elemento tres (figura 42). Al realizar este cambio, fue necesario modificar la forma de esa pieza, junto con la forma de la pieza del elemento dos (tapa) (figura 42). También se cambió el material de fabricación, siendo el nuevo material el metacrilato, y el medio de fabricación, que fue una cortadora láser.

En el tercer diseño se creó una nueva envolvente para los servomotores, para que estos quedaran lo más cubiertos posibles y no fueran tan accesibles. A su vez, se eliminaron los huecos por donde pasarían los cables y se sustituyeron por unas guías, evitando así que el cableado pudiera quedar atorado y facilitando su montaje.

Todos estos cambios provocaron un aumento en la masa de las piezas. Teniendo en cuenta que el servomotor que movía el *hombro* ya se encontraba trabajando al 74,18% (cerca de su límite), se decidió disminuir la distancia entre los servos, para reducir el momento que produciría el añadir estas nuevas partes. Por tanto, ahora la pieza que mueve el *hombro* y el *codo*, cambia sus dimensiones y pasa de 18 cm a 16,25 cm; la pieza que une el servomotor del *codo* y la *muñeca*, se reduce de 14 cm a 11,5 cm; y la que une el servomotor de la *muñeca* con el extremo del brazo, se modifica de 14 cm a 12,5 cm, alterando la longitud total del brazo, medida desde el servomotor que mueve el *hombro* hasta el extremo del brazo, se reduce en 6,75 cm (desde 47 cm a 40,25 cm).

Tras un uso prolongado del brazo realizado con las piezas de la anterior versión, se notó que en la unión por interferencia que había entre las *cabezas* de los servomotores y las piezas diseñadas en 3D, existía un desgaste que podría provocar con el tiempo que dicho encaje fuera perdiendo eficiencia. Debido a eso, se decidió incluir un hueco en el centro del eje donde poder atornillar un tornillo desde el cuerpo de la pieza a la *cabeza* del actuador.

Teniendo en cuenta las decisiones tomadas y las mejoras mencionadas en los párrafos anteriores, se procede a detallar los cambios realizados en esta tercera versión.

La **tapa** ahora de **metacrilato**, está adaptada para encajar y colocarse en la *base* y quedar totalmente sujeta mediante un único tornillo (figura 58). A su vez las dimensiones se ven cambiadas, midiendo ahora 20,3 cm x 15 cm.

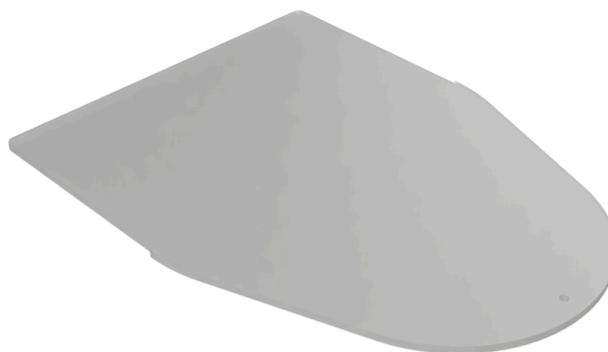


Figura 58 - Tercera versión de la Tapa

La **Base** ahora incorpora la electrónica del brazo, disponiendo de huecos para permitir interactuar con los potenciómetros, botones, interruptores y demás

componentes electrónicos de la Placa Base (figura 59). Presenta las mismas dimensiones que la *tapa* y una altura de 4,6 cm.



Figura 59 - Tercera versión de la Base con la electrónica integrada.

La **cintura** es la envolvente del servomotor, y ofrece un hueco para que puedan pasar los cables y conectarse con la PCB del brazo (figura 60).



Figura 60 - Tercera versión de la Cintura.

El **hombro** disminuye su longitud, pasando de 18 cm a 16,25 de eje a eje de los servomotores (figura 61). A su vez, se cambia en 180° la orientación de uno de los servomotores, con el fin de tener todos los servomotores orientados de igual modo. Se le añaden las guías de cables, y un hueco para el tornillo que sujeta el servomotor que lo mueve.



Figura 61 - Tercera versión del Hombro.

En el **codo**, como se mencionó anteriormente, se modifica la separación entre servomotores, pasando la distancia entre ejes de 14 cm a 11,5 cm. Se crea la envolvente del servomotor, y se le añade la guía para los cables y el hueco para atornillar la pieza al servomotor que lo mueve.



Figura 62 - Tercera versión del Codo.

En la **muñeca** se añaden unos orificios para permitir atornillar la PCB Auxiliar, se disminuye la distancia de separación entre servomotores y se añade el hueco para atornillar esta pieza al servomotor (figura 63).

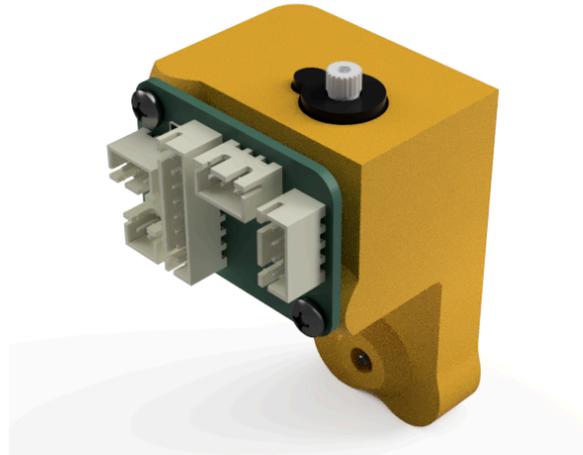


Figura 63 - Tercera versión de la Muñeca

La **pinza** no sufre demasiados cambios, solo se le reduce el tamaño en 0,5 cm y se le añade un hueco para atornillar la pieza al servomotor que lo rota (figura 64).

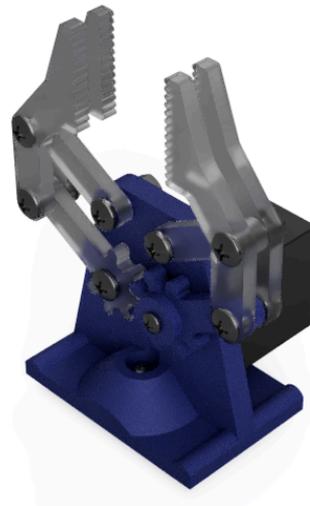


Figura 64 - Tercera versión de la Pinza

Con la unión de todos estos componentes mecánicos se obtiene el resultado mostrado en la figura 65.



Figura 65 - Tercera versión ensamblaje del Brazo Robótico Educativo.

4.3.3.2 Funcionamiento del diseño

Tras estas modificaciones se repitieron los cálculos mecánicos, dando como resultado lo mostrado en la Tabla 4.5.

Tabla 4.5 - Carga soportada por los servomotores en la tercera versión.

Servo	Momento a soportar (kgf · cm)	Porcentaje de trabajo
Hombro	6,873	73,12%
Codo	1,958	20,83%
Muñeca	0,361	20,06%

Aquí se puede observar cómo, a pesar de aumentar la masa de las piezas, en algunos servomotores se ha reducido la carga que soportan, como en el caso del *hombro* que pasa de 74,18% (Tabla 4.4) a 73,12% (Tabla 4.5) y la *muñeca* pasando de 26,1% (Tabla 4.4) a 20,06% (Tabla 4.5), o se han aumentado ligeramente, como en el caso del *codo*, aumentando del 20,37% (Tabla 4.4) al 20,83% (Tabla 4.5). Viendo estos datos se puede sospechar que el límite máximo que puede levantar el

brazo, vendrá limitado por el servomotor que mueve el *hombro*. Aún así se procede a calcular el límite máximo que puede levantar cada servomotor para alcanzar su carga máxima.

Para poder calcular esta carga máxima es necesario conocer la carga máxima que puede soportar cada servomotor, las cuales conocemos, recordando que son 1,8 kgf·cm [47], para el MG90S, y 9,4 kgf·cm [48], para el MG996R.

Cogiendo como ejemplo el servomotor que mueve el *hombro*, calculamos el momento que tiene que generar esta nueva carga sobre el brazo para llevarlo al límite, siendo esta:

$$M_{Diferencia} = M_{m\acute{a}x\ MG996R} - M_{m\acute{a}x} = 9,4\ kgf \cdot cm - 6,873\ kgf \cdot cm = 2,527\ kgf \cdot cm$$

Conociendo este valor, tenemos que dividirlo entre la distancia del eje del servomotor a estudiar y el centro de gravedad de la pieza a levantar. Como no conocemos la forma del objeto a levantar, suponemos que el punto del centro de masa se encuentra en el extremo del brazo, por tanto, para el caso del *hombro*, este punto se encuentra a una distancia de 40,25 cm. Así podemos saber cuánta masa puede llegar a levantar este servomotor:

$$Masa_{Objeto} = \frac{M_{Diferencia}}{CG_{Hombro}} = \frac{2,527\ kgf \cdot cm}{40,25\ cm} = 0,0628\ kg = 62,8\ g$$

Haciendo lo mismo con el resto de servomotores, nos encontramos con los resultados recogidos en la tabla 4.6.

Tabla 4.6 - Carga máxima que pueden levantar los servomotores en la tercera versión.

Servo	Distancia del centro de masa (cm)	Masa del objeto (g)
Hombro	40,25	62,8
Codo	24	310,1
Muñeca	12,5	723,1

4.4. Fabricación mecánica

En este apartado se detallan los pasos seguidos para fabricar las diferentes piezas del prototipo.

Para diseñar las diferentes partes del brazo robótico, se ha utilizado el software Fusion 360, exportando el cuerpo diseñado en el formato STL. Para esto, se debe seleccionar la pieza del componente en el programa y guardarla como malla.

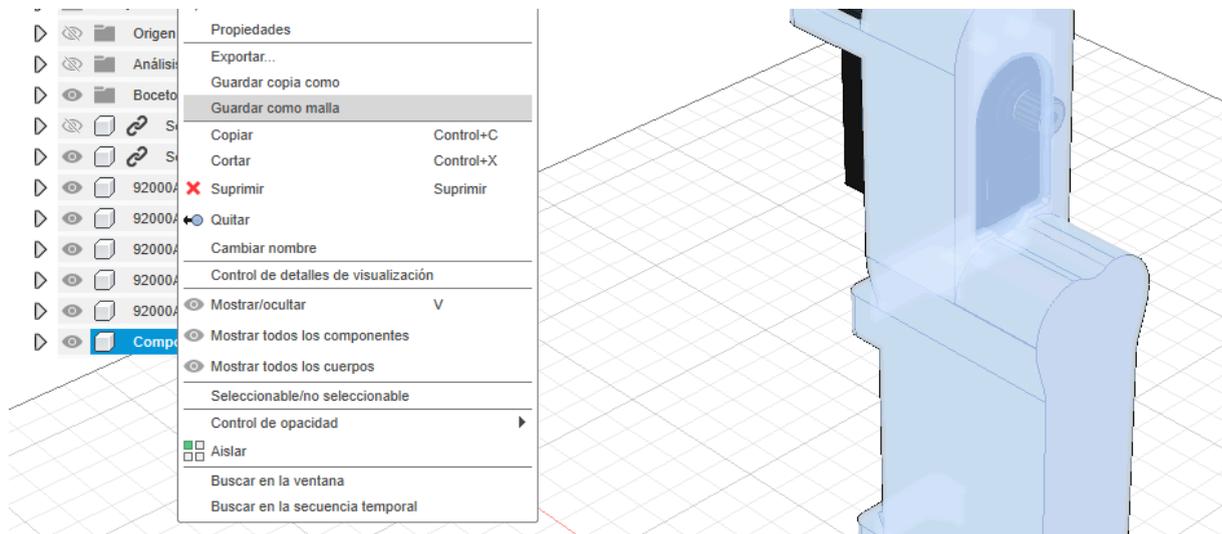


Figura 66 - Guardado de malla de pieza

Una vez seleccionada la opción de “**Guardar como malla**” (figura 66), se indica el formato STL y se puede seleccionar la calidad del mismo. En nuestro caso se ha elegido el refinado “Alto”. Aceptados los parámetros indicados, se descargará el archivo deseado (figura 67).

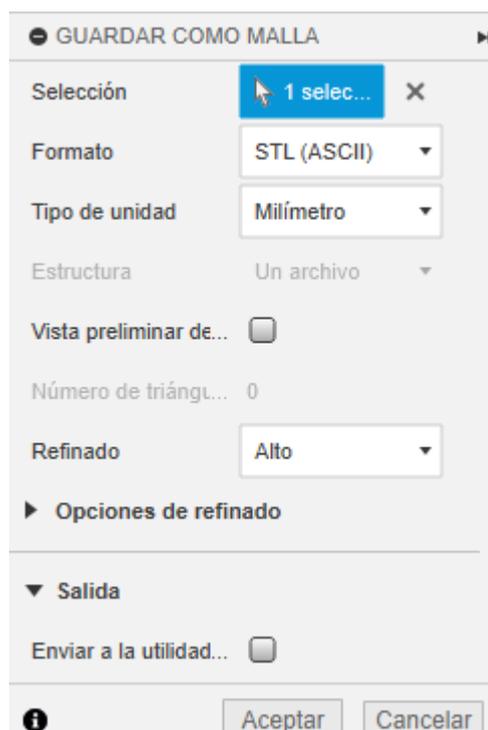


Figura 67 - Parámetros del guardado de malla

Cuando se tiene el archivo STL, abrimos el software Prusa Slicer (figura 26). Con el software abierto, se introduce el archivo exportado anteriormente (figura 67), y una vez tengamos la pieza le indicamos los diferentes parámetros de impresión.

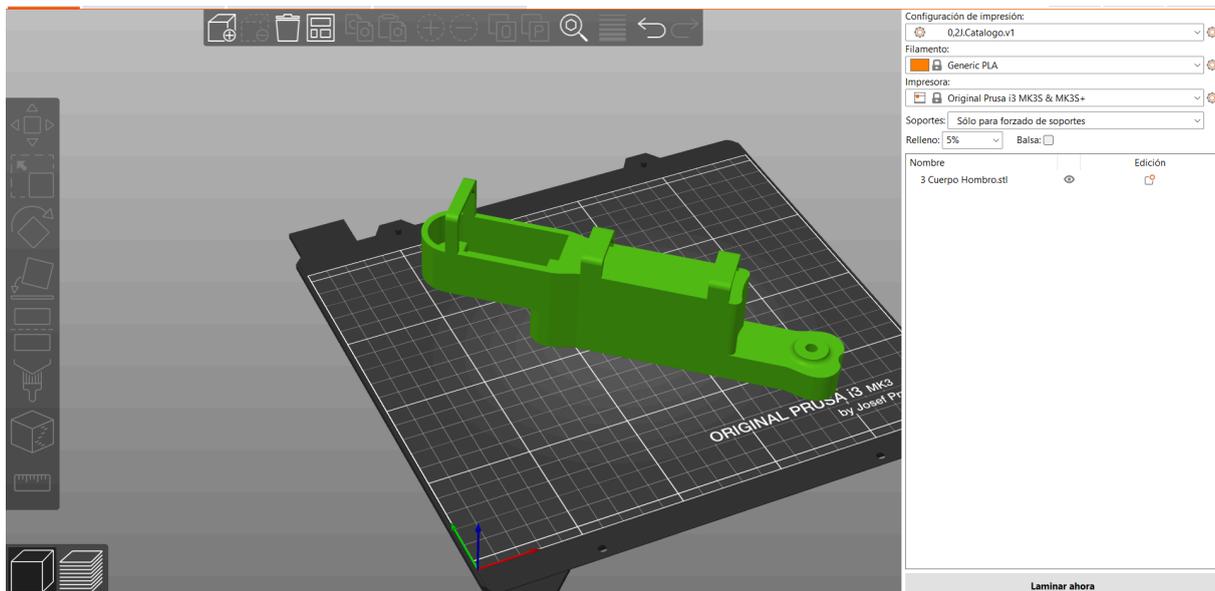


Figura 68 - Colocación de la pieza para su laminación.

Entre los parámetros importantes que deben ajustarse se encuentran los siguientes:

- El material a imprimir, que determinará la temperatura del extrusor y la base, así como la velocidad de impresión. En este caso, todas las piezas serán impresas en PLA¹³.
- El porcentaje de relleno, que en nuestro diseño será del 5% para reducir lo máximo posible el peso de las piezas.
- El número de capas en el perímetro, en las capas superiores y en las inferiores, que en nuestro caso serán 3.
- Los soportes para las partes de la pieza que se encuentren en voladizo.

Una vez indicados estos parámetros, el software podrá realizar el laminado (figura 69), y podemos observar una previsualización de cómo se fabricará la pieza en nuestra impresora 3D.

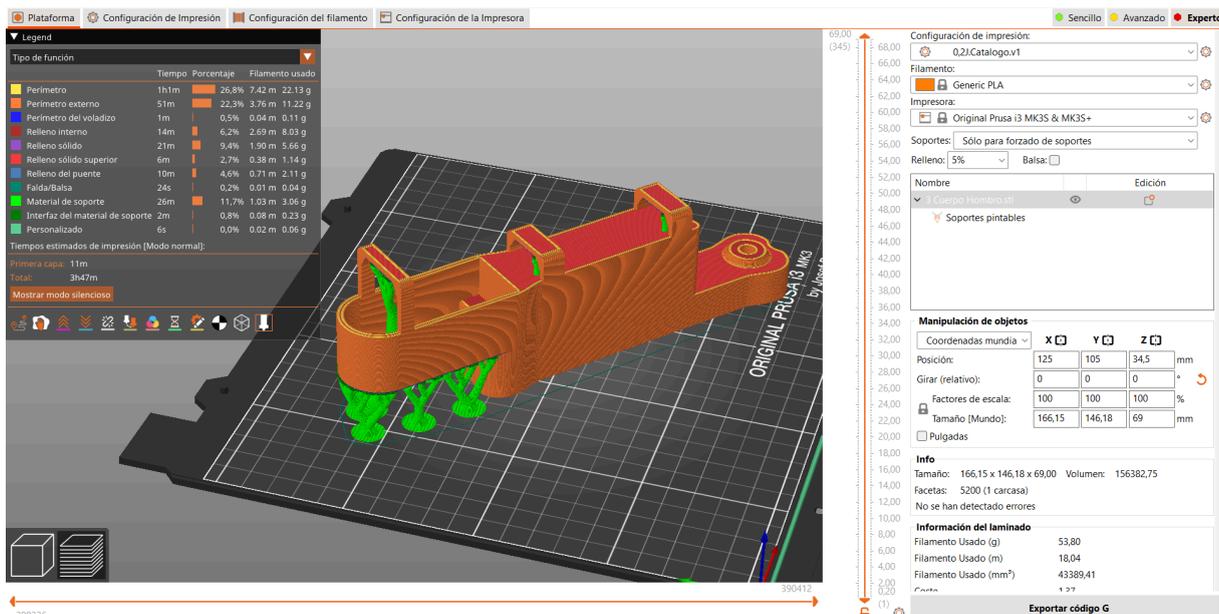


Figura 69 - Laminado de pieza.

Con la pieza laminada se procede a exportar el código G, que es un archivo en formato GCODE, que contiene los comandos de fabricación de la impresora 3D. Introducido este archivo en la impresora e insertado el material con el que fabricar en la máquina, se procede a su construcción (figura 70). Los tiempos y cantidad de filamento que se utilizará para la fabricación de las diferentes piezas se muestran en la Tabla 4.7 y se han extraído de los diferentes GCODEs generados.

Tabla 4.7 - Cantidad de filamento y tiempo de impresión de las piezas que componen el brazo robótico

Pieza a imprimir	Cantidad de filamento (g)	Tiempo en imprimir
Base	216,14	17 horas y 46 minutos
Cintura	93,43	7 horas y 48 minutos
Hombro	59,73	5 horas y 19 minutos
Codo	38,08	3 horas y 18 minutos
Muñeca	16,25	1 hora y 48 minutos
Pinza	7,97	57 minutos
Pieza con engranaje para servo	0,93	7 minutos

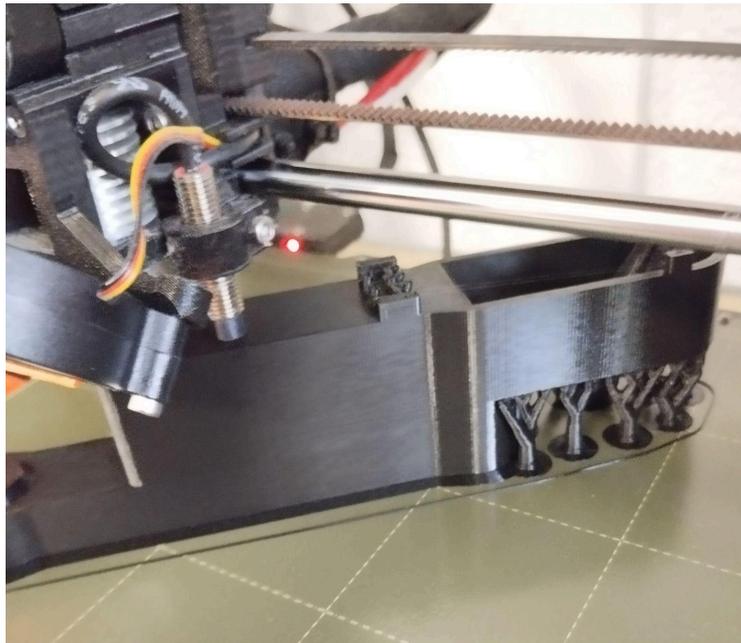


Figura 70 - Fabricación en impresora 3D de pieza del brazo robótico.

Mientras tanto, para la *tapa* del brazo robótico y las piezas que componen la *pinza*, es necesario obtener un archivo DXF para poder realizar su fabricación con métodos de fabricación sustractiva. Para ello, en el software CAD, hay que seleccionar el boceto donde se muestra la forma de la pieza y exportar su archivo DXF (figura 71).

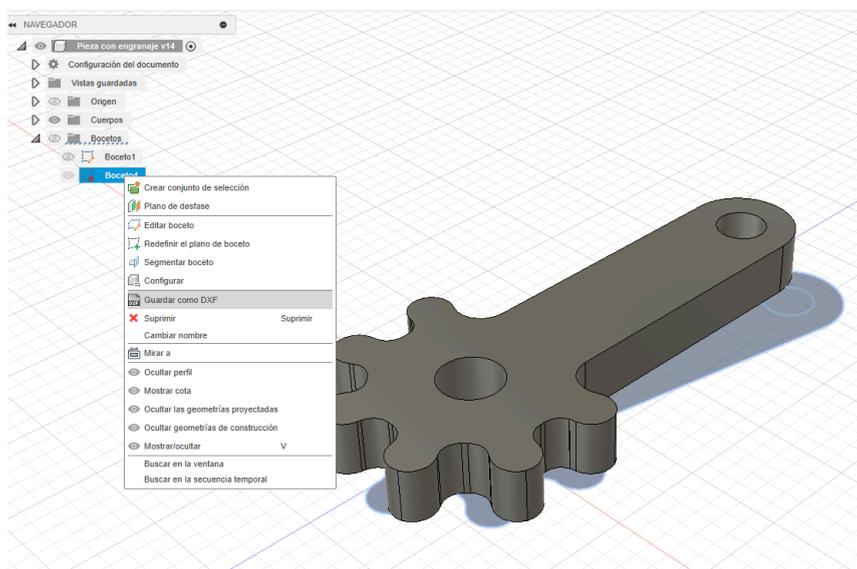


Figura 71 - Guardado de archivo dxf.

Ya obtenido este archivo, a continuación utilizamos el software CAM RDWorks, que indicará a la cortadora láser el conjunto de operaciones a realizar. En nuestro caso, se introducen los parámetros de corte (potencia y velocidad) adecuados para un

metacrilato de 3 mm de espesor (figura 72), y se exporta el archivo RD generado, que se introduce en la cortadora láser .

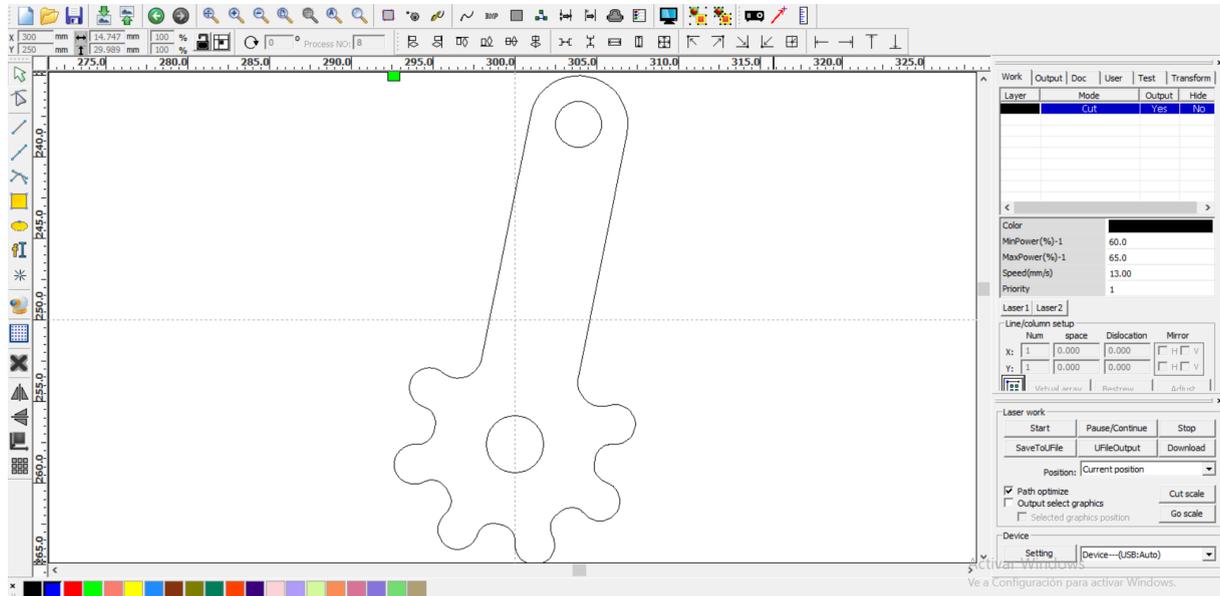


Figura 72 - Parámetros de corte para la cortadora láser.

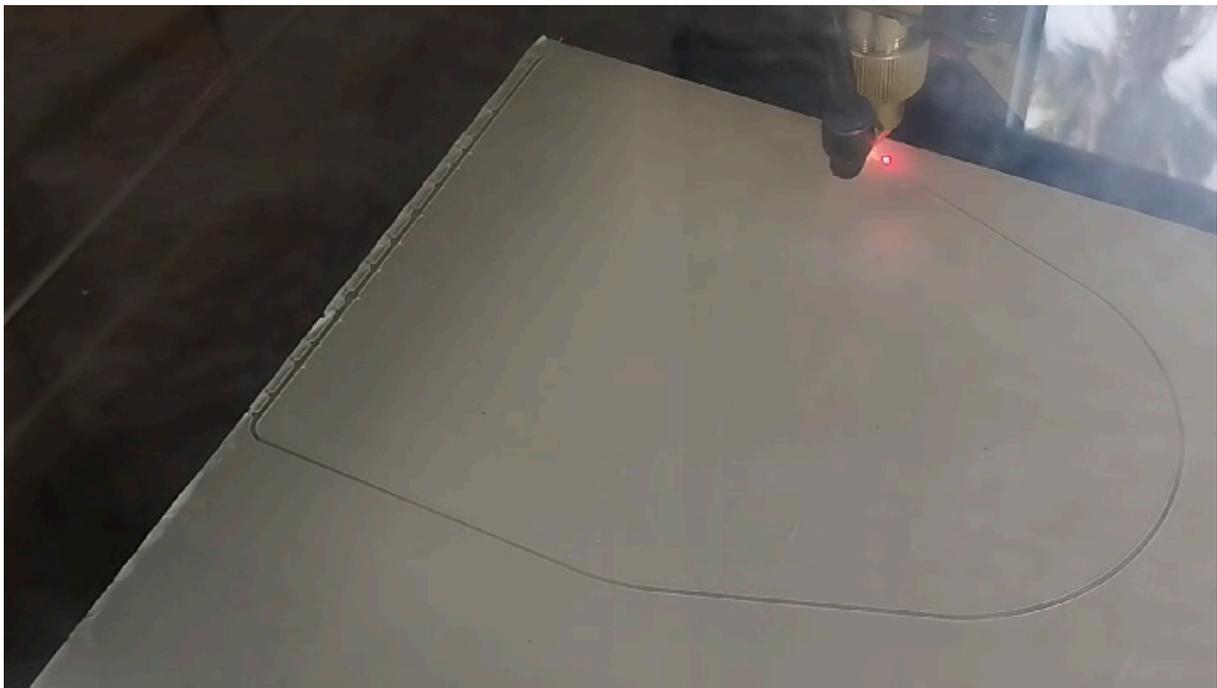


Figura 73 - Fabricación en corte láser de pieza del brazo robótico.

Con las piezas fabricadas, tanto las hechas con fabricación aditiva como las hechas con fabricación sustractivas (figura 74), se procede al montaje de las partes junto a los componentes electrónicos y tornillería, quedando como resultado final el mostrado en la figura 75.

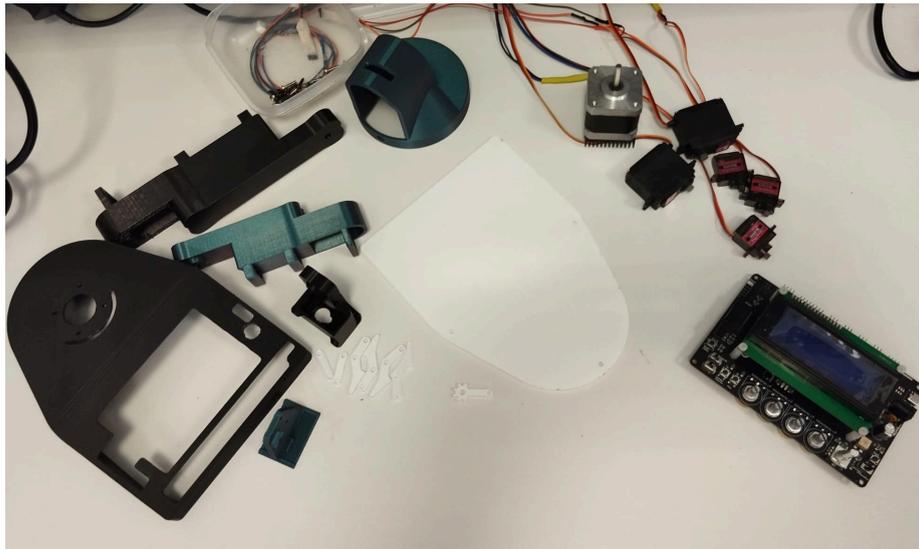


Figura 74 - Piezas del Brazo Robótico Educativo



Figura 75 - Brazo Robótico Educativo montado.

4.5. Diseño electrónico

Partiendo de las especificaciones mecánicas del Brazo Robótico Educativo, se procede a elaborar una Placa de Circuito Impreso que permita realizar un control completo del dispositivo. Para ello, se establecen las siguientes especificaciones mínimas de control:

- Debe contar con un microcontrolador compatible con Arduino, que permita su uso educativo y una rápida programación.
- Debe disponer de diversas interfaces de Entrada/Salida, que permitan realizar el control del brazo robótico y mostrar la suficiente información al usuario respecto a su estado y modo de funcionamiento.
- Debe ser capaz de controlar 5 servomotores y un motor paso a paso.
- Debe ser compatible con comunicaciones inalámbricas de tipo Bluetooth, WiFi o similar.
- Será configurable, permitiendo realizar un control manual del brazo (similar a un dispositivo de radiocontrol), pero permitiendo también programar secuencias de movimientos complejas. Por tanto, debe contar con dos modos de funcionamiento, a los que se les denominará modo 'manual' y modo 'automático'.

Teniendo en cuenta dichas especificaciones, se procede a realizar un diseño del circuito de control en protoboard para comprobar que todo funciona de forma adecuada, y tras la validación, se elabora la PCB que servirá de controlador del dispositivo y que posteriormente será fabricada e integrada en el dispositivo.

Durante este proceso, se realizaron 4 versiones de la PCB del brazo, de las cuales, la que llegó a fabricarse fue la última. Estas versiones, cuya información detallada puede observarse en el Anexo III, fueron diseños incrementales que se iban modificando debido a que, a medida que iba avanzando el proyecto, se fueron añadiendo nuevas funcionalidades hasta llegar a la última versión, denominada **Versión 1.4**.

Para una mejor comprensión del funcionamiento del dispositivo desde un punto de vista electrónico se muestra un diagrama de bloques en la figura 76.

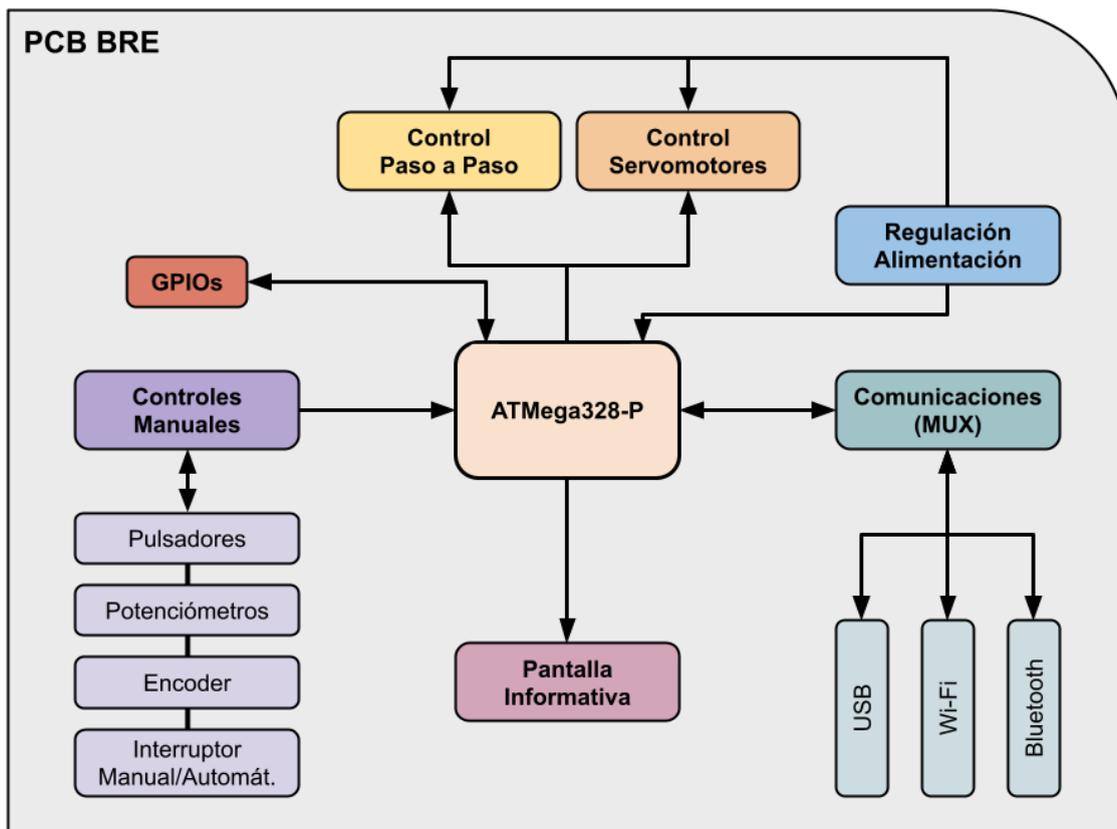


Figura 76 - Diagrama de bloques del funcionamiento electrónico de la PCB del brazo robótico.

La placa base del Brazo Robótico está formada por un microcontrolador ATmega328P con un oscilador de 16MHz, que, mediante el uso de diversos puertos digitales, permitirá el control de los servomotores y del motor paso a paso.

Se utiliza un controlador PCA9685PW [48], que es un controlador de PWM gestionado por I2C, conectado con el microcontrolador para aumentar el número de salidas PWM disponibles, debido a que, a lo largo del proyecto, se vio la necesidad de implementar un expansor como este para minimizar el número de pines necesarios para controlar los actuadores. Con este controlador es posible, a través de los 2 pines del bus I2C (SDA y SCL), controlar hasta 16 servomotores, frente a los 16 pines necesarios en caso de hacer un control directo. Las características del controlador PCA9685PW se muestran en la Tabla 4.8.

Tabla 4.8 - Características del controlador PCA9685PW.

Temperatura de operación (°C)	-40 a 85
Frecuencia de PWMs	Frecuencia ajustable de 1 kHz

Velocidad de I2C-bus (kHz)	1000
Voltaje de operación (VDC)	2.3 a 5.5
Corriente máxima de Sink por bit (mA)	25
Número de direcciones	126
Número de salidas	16

El control del motor paso a paso se realiza mediante un driver de motores DRV8825. Este controlador se escoge porque el motor paso a paso utilizado requiere 12V para su funcionamiento, y una corriente continua de 1A. Con este controlador se puede regular la corriente máxima que alimenta al motor. Las características del controlador DRV8825 [49] se muestran en la Tabla 4.9.

Tabla 4.9 - Características del controlador DRV8825.

Tensión mínima de funcionamiento	8.2 V
Tensión máxima de funcionamiento	45 V
Corriente continua por fase	1.5 A
Corriente máxima por fase	2.2 A
Tensión lógica mínima	2.5 V
Tensión lógica máxima	5.25 V
Resolución de micropasos	completo, 1/2, 1/4, 1/8, 1/16 y 1/32
¿Protección contra la tensión inversa?	No
Dimensiones	15,5 × 20,5 mm (0,6" × 0,8")

También se dispone de diversos potenciómetros, encoders y pulsadores que se conectan a distintos pines digitales y analógicos del ATmega. A través del bus I2C se controla una pantalla LCD de dimensiones 20x04 caracteres. Por último, para

permitir diversos modos de comunicación se dispone de un multiplexor desde el que es posible conmutar los pines de comunicación serial de la UART del microcontrolador hacia un módulo Bluetooth, módulo WiFi o el conector USB del dispositivo. En los siguientes subapartados se abordará en detalle cada uno de estos componentes.

4.5.1. Microcontrolador principal ATmega 328P

El elemento esencial del circuito del controlador es el microcontrolador ATmega328P. Este integrado es un microcontrolador perteneciente a la familia AVR¹⁴ con una arquitectura 8 bits. Sus características más destacadas se exponen en la Tabla 4.10.

Tabla 4.10 - Características del ATmega328P.

Arquitectura	8 bits
Memoria Flash	32 kB
Memoria SRAM	2 kB
Memoria EEPROM ¹⁵	1 kB
Número de pines E/S digitales	14 (6 de ellos PWM de 8 bits)
Número de pines de entrada analógica	6
Resolución del ADC	10 bits
Protocolos de comunicación disponibles	Serial TTL, SPI, I2C.
Voltaje de alimentación	5 V
Frecuencia del oscilador	8 MHz - 16 MHz

Este microcontrolador se ha montado junto con un cristal de cuarzo de 16MHz, con el objetivo de obtener el máximo aprovechamiento de sus características. En la figura 77 puede visualizarse el diseño electrónico del bloque microcontrolador, formado por el integrado ATmega328P trabajando a la frecuencia de reloj de 16MHz gracias al cuarzo Y1 de 16 MHz, la resistencia R14 de 1MΩ y los condensadores C5 y C6 de 22pF. Se dispone también de un circuito de reseteo del dispositivo formado por el pulsador SW5 y el circuito de *pull-up* con 5V formado por la resistencia R13 de 10 KΩ. Por otro lado, el resto de pines digitales y analógicos del microcontrolador son etiquetados con el GPIO¹⁶ correspondiente para conectarlos al resto de elementos del sistema.

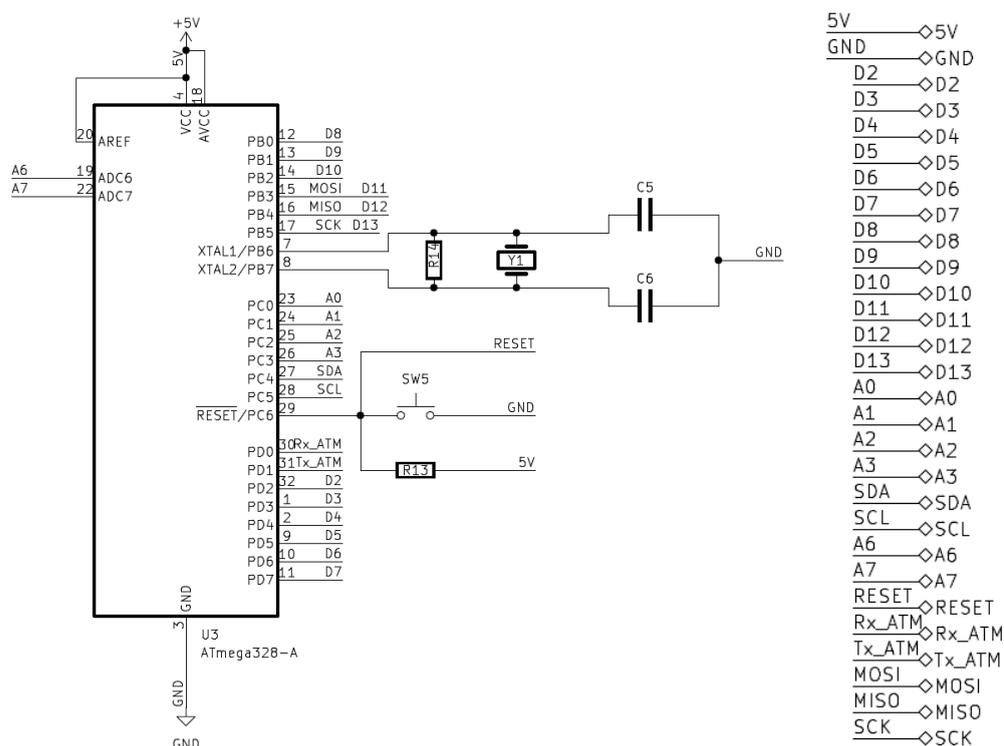


Figura 77 - Diseño electrónico del bloque del microcontrolador

4.5.2. Alimentación del dispositivo

Teniendo en cuenta la cantidad de actuadores de las que dispone el brazo, se hace muy necesario el correcto dimensionado del bloque de alimentación. En este caso, el dispositivo debe ser alimentado a partir de 12V a través de un conector DC-Jack, por lo que es necesario hacer uso de diversas etapas de conversión de voltaje para adaptar la tensión a la de trabajo del microcontrolador, que es de 5V, y a la de 3,3V que alimenta diversos periféricos y componentes encargados de las comunicaciones inalámbricas. Por otro lado, la tensión de 12V se usará de forma directa para alimentar el driver del motor paso a paso, que necesita operar con una tensión superior a 8V.

Para el bloque de conversión de 12V a 5V, se ha decidido hacer uso de un convertidor DC/DC de tipo Step-Down buck basado en el integrado LM2596 (integrado U5 de la figura 79). Es un regulador conmutado monolítico de alta corriente (hasta 3A) operado a 150 kHz que ofrece una alta eficiencia y una baja disipación de calor como puede verse en la figura 78, donde se muestran las características técnicas extraídas de su *datasheet*.

Specifications are for $T_J = 25^\circ\text{C}$, $I_{\text{LOAD}} = 500\text{ mA}$, $V_{\text{IN}} = 12\text{ V}$ for the 3.3-V, 5-V, and adjustable version, and $V_{\text{IN}} = 24\text{ V}$ for the 12-V version (unless otherwise noted).

PARAMETER		TEST CONDITIONS	MIN ⁽¹⁾	TYP ⁽²⁾	MAX ⁽¹⁾	UNIT
DEVICE PARAMETERS						
I_b	Feedback bias current	Adjustable version only, $V_{\text{FB}} = 1.3\text{ V}$	$T_J = 25^\circ\text{C}$	10	50	nA
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$		100	
f_o	Oscillator frequency ⁽³⁾	$T_J = 25^\circ\text{C}$		127	150	173
		$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$		110	173	
V_{SAT}	Saturation voltage ^{(4) (5)}	$I_{\text{OUT}} = 3\text{ A}$	$T_J = 25^\circ\text{C}$	1.16	1.4	V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$		1.5	
DC	Max duty cycle (ON) ⁽⁵⁾			100%		
	Min duty cycle (OFF) ⁽⁶⁾			0%		
I_{CL}	Current limit ^{(4) (5)}	Peak current	$T_J = 25^\circ\text{C}$	3.6	4.5	6.9
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	3.4	7.5	
I_L	Output leakage current ^{(4) (6)}	Output = 0 V, $V_{\text{IN}} = 40\text{ V}$			50	μA
		Output = -1 V		2	30	mA
I_Q	Operating quiescent current ⁽⁶⁾	See (6)		5	10	mA
I_{STBY}	Current standby quiescent	$\overline{\text{ON/OFF}}$ pin = 5 V (OFF) ⁽⁷⁾	$T_J = 25^\circ\text{C}$	80	200	μA
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$		250	μA
SHUTDOWN/SOFT-START CONTROL (see Figure 9-13 for test circuit)						
V_{IH}	$\overline{\text{ON/OFF}}$ pin logic input threshold voltage	Low (regulator ON)	$T_J = 25^\circ\text{C}$	1.3		V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$		0.6	
V_{IL}		High (regulator OFF)	$T_J = 25^\circ\text{C}$	1.3		V
			$-40^\circ\text{C} \leq T_J \leq 125^\circ\text{C}$	2		
I_H	$\overline{\text{ON/OFF}}$ pin input current	$V_{\text{LOGIC}} = 2.5\text{ V}$ (regulator OFF)		5	15	μA
I_L		$V_{\text{LOGIC}} = 0.5\text{ V}$ (regulator ON)		0.02	5	μA

Figura 78 - Características técnicas del el integrado LM2596

Para su diseño, se ha consultado el montaje recomendado por el fabricante Texas Instrument en su datasheet [50] que establece que para la versión de salida fija de 5V, es suficiente con acompañar el circuito integrado de una bobina de 33uH a la salida ($L1$) ubicada entre un diodo *schottky*¹⁷ en polarización inversa ($D5$) y un condensador de filtrado de al menos 220uF ($C13$) adecuado a la tensión de salida (5V). El pin de habilitación (pin número 5 del integrado $U5$) se conectará directamente a tierra de forma que al operar con lógica negada, el regulador se encuentre funcionando desde que haya tensión en el circuito. Finalmente, la realimentación (pin 4 del integrado $U5$) se conectará de forma directa a la salida de 5V deseada para monitorizar de forma continua la tensión de salida.

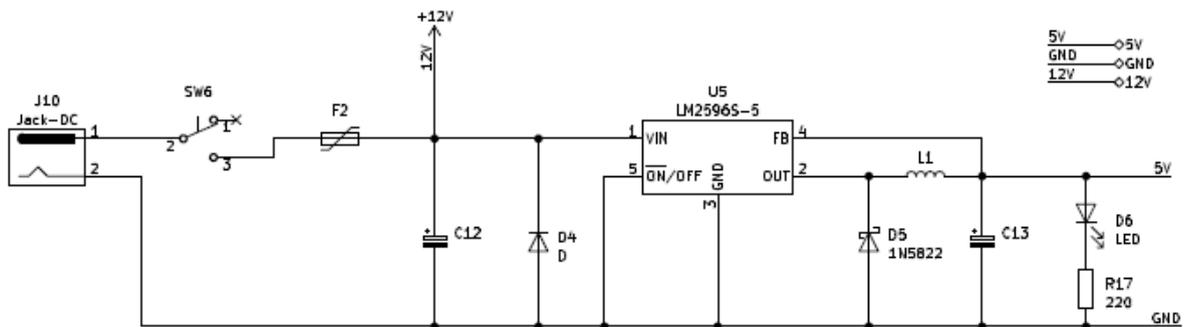


Figura 79 - Diseño electrónico del bloque de alimentación del dispositivo.

Para una mayor seguridad del circuito se incluye un interruptor de desconexión del sistema (SW6) y un sistema de protección frente a inversión de polaridad en el conector DC-Jack. Con este circuito formado por un fusible rearmable (F2) y un diodo en inversa (D4), se consigue que ante una colocación errónea de los pines de alimentación, de forma que la tensión de 12V se encuentre conectada a GND y el negativo conectado a la entrada positiva, dicho diodo D4 conduzca haciendo pasar una gran corriente durante un instante y procediéndose a la apertura del fusible de tipo PTC¹⁸ (F2). Se hace, por tanto, muy importante la elección de dicho fusible, por lo que teniendo en cuenta que la corriente máxima de salida del convertor es de 3A y considerando la relación de conversión de 12V a 5V con una eficiencia del 85%, se obtiene una corriente máxima de entrada de aproximadamente 490 mA, esto sumado al consumo producido por el motor paso a paso de 1 A, hace que se recurra a un fusible rearmable de 1,5A a 12V.

Por otro lado, la tensión de 3,3V se obtiene a partir de un regulador lineal de tensión de bajo *drop-out*¹⁹ conectado a la salida del convertor de 5V, concretamente el modelo LM1117-3.3 de Texas Instrument. Al existir una baja caída de tensión entre la entrada y la salida (se pasa de 5V a 3,3V) se consigue una baja disipación de calor en el regulador. Para su montaje se ha recurrido al diseño propuesto en el datasheet por el fabricante, quedando como se muestra en la figura 80. Es importante tener en cuenta que este dispositivo ofrece una corriente limitada a 800mA, valor muy inferior al proporcionado por otros reguladores de tipo conmutado, aunque suficiente para la aplicación deseada, que es la alimentación del módulo WiFi, con un consumo de menos de 500mA.

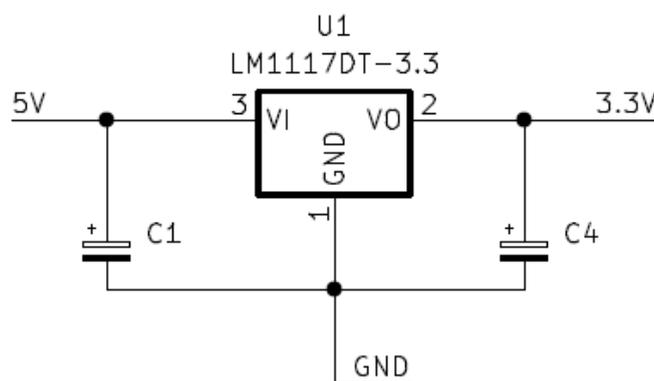


Figura 80 - Diseño electrónico del bloque Alimentación de 5V a 3.3V.

4.5.3. Control del motor paso a paso

Para la realización del movimiento de la *base* del brazo robótico se decide hacer uso de un motor paso a paso, concretamente el modelo NEMA 17. Este tipo de motor ofrece un control de alta precisión, con libertad de movimiento y un par alto. Se trata de un modelo de servomotor ampliamente utilizado en máquinas CNC como cortadoras láser, fresadoras o impresoras 3D, por lo que su fiabilidad está testada por toda la comunidad maker.

Los principales datos técnicos de este motor han sido extraídos de su datasheet [46], y se mostrarán en la Tabla 4.11.

Tabla 4.11 - Principales datos técnicos del motor NEMA17.

Número de fases	2
Tensión mínima de alimentación	8,2 V
Tensión máxima de alimentación	45 V
Pasos/Revolución	200
Velocidad máxima	1.000 rpm

Para realizar el control de este motor se hace indispensable el uso de un módulo DRV8825, que permite una alta precisión de control de 1/32 micropasos, es decir, para el motor de 200 pasos por revolución del que se dispone, se puede obtener un total de 6.400 pasos por revolución en la secuencia de 32 micropasos, o lo que es lo mismo, una precisión de movimiento de 0,056 grados. El uso de las secuencias de micropasos frente a los pasos completos, permite un control más preciso y suaviza el movimiento del brazo robótico. Sin embargo, para este proyecto, haremos uso del modo de paso completo que ofrece una resolución de 1,8°/paso, por la necesidad de simplificar el sistema de control, debido a que la precisión extrema no es requerida y los recursos computacionales son limitados. Al usar la secuencia de pasos completos, se reduce la complejidad del controlador y se simplifica la programación, lo que resulta beneficioso para esta aplicación educativa, donde la simplicidad y la fiabilidad son prioritarias sobre la precisión extrema.

Este módulo se instala en la PCB del dispositivo a través de dos tiras de pines hembra (*J5* y *J6*) que funcionan a modo de zócalo de conexión, como puede verse en la figura 81.

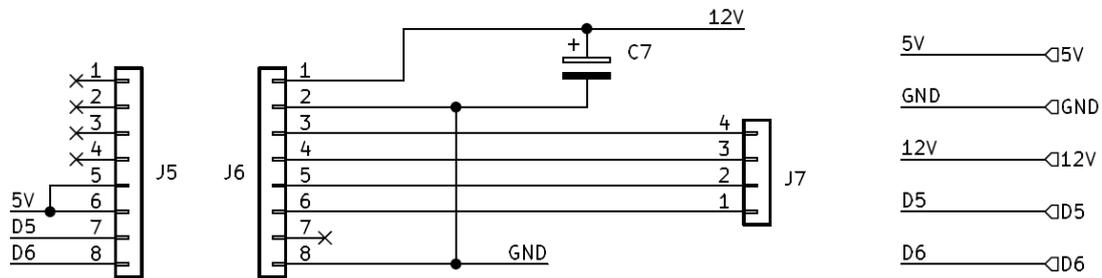


Figura 81 - Diseño electrónico del bloque del controlador del motor paso a paso.

La resolución en cuanto al control de motores puede ser definida haciendo uso de los pines denominados M0, M1 y M2 que configurarán el modo de trabajo de acuerdo a la tabla de verdad, de la Tabla 4.12.

Tabla 4.12 - Tabla de verdad del controlador DRV8825.

M0	M1	M2	Resolución
0	0	0	Paso completo
1	0	0	1/2 paso
0	1	0	1/4 paso
1	1	0	1/8 paso
0	0	1	1/16 paso
1	0	1	1/32 paso
0	1	1	1/32 paso
1	1	1	1/32 paso

Para utilizar correctamente el controlador DRV8825 con el ATmega328P y proteger tanto el motor paso a paso como el propio controlador, es crucial configurar el límite de corriente. Este proceso es esencial y se puede llevar a cabo realizando los siguientes pasos [51] (figura 82):

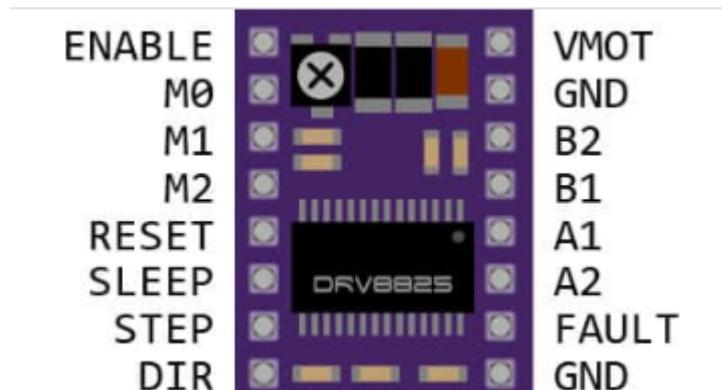


Figura 82 - Pines del DRV8825 [51]

- **Preparación de herramientas:** Antes de comenzar, hay que asegurarse de tener a mano un pequeño destornillador, un multímetro para medir la tensión de referencia y, opcionalmente, cables de prueba tipo cocodrilo para facilitar las mediciones.
- **Alimentación del controlador:** Conectar adecuadamente el controlador DRV8825, asegurándose de alimentarlo a través del pin VMOT con una tensión de entre 8.2 y 45 V. Además, aplicar 5 V a los pines RST y SLP para encender el driver. Es recomendable desconectar el motor paso a paso durante este proceso.
- **Cálculo del límite de corriente:** Hay que utilizar la fórmula de límite de corriente, donde el límite de corriente es igual a la tensión de referencia (V_{ref}) multiplicada por 2. Para este caso, trabajando con un límite de corriente de 1 A, la V_{ref} debe ser de 0.5 V.
- **Selección del límite de corriente adecuado:** Consultar la hoja de datos del motor paso a paso para determinar su corriente nominal. Si no se puede encontrar esta información, se recomienda comenzar con un límite de corriente de 1 A y ajustarlo según sea necesario.
- **Ajuste de la V_{ref} :** Medir la tensión de referencia (V_{ref}) entre los puntos marcados en el controlador (GND y el potenciómetro) y ajustar el potenciómetro hasta alcanzar el valor calculado previamente.
- **Medición y ajuste:** Es posible medir directamente el consumo de corriente del motor paso a paso, pero se recomienda el método de ajuste de la V_{ref} por su simplicidad. Si el motor emite mucho ruido, se puede considerar reducir el límite de corriente, aunque es importante mantenerlo lo suficientemente alto para evitar la pérdida de pasos.

Configurar correctamente el límite de corriente es crucial para garantizar un funcionamiento seguro y eficiente del motor paso a paso y del controlador DRV8825, y este proceso asegura que ambos componentes estén protegidos contra posibles daños por corriente excesiva.

4.5.4. Control de servomotores

Debido a que el brazo robótico cuenta con cinco grados de libertad, y al aumento del uso de pines de entradas y salidas del ATmega328P, es necesario la implementación de un controlador separado para controlar todos los servomotores que posee el brazo, que se comunique a través de I2C.

Por esta razón se decidió utilizar el módulo PCA9685PW (*U6*), debido a que cuenta con 16 salidas controladas PWM, utilizando 4 salidas para 4 servos. Debe ser conectado a una fuente de alimentación de 5V y sus señales de control I2C deben ser gestionadas por el ATmega328P. Este bloque del controlador de los servomotores quedaría como se observa en la figura 83.

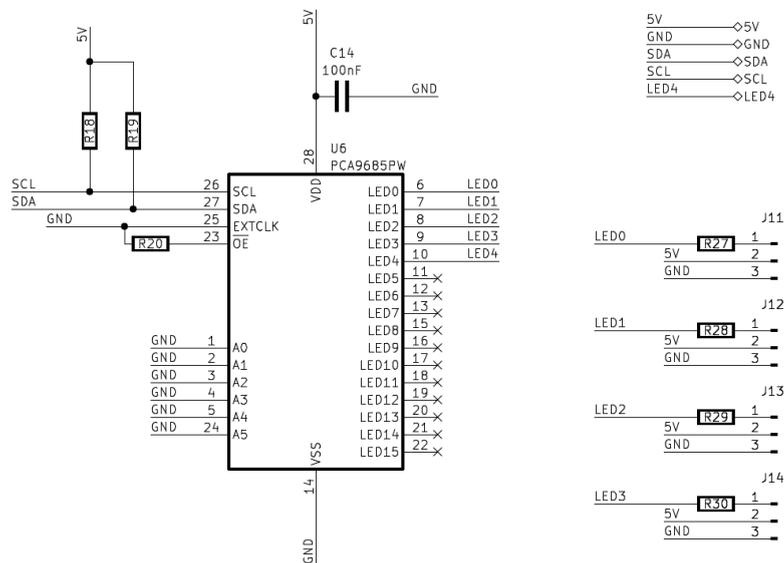


Figura 83 - Diseño electrónico del bloque del controlador de servomotores.

Al hacerse uso del bus I2C, cuyas líneas son de drenador abierto, es necesario añadir dos resistencias de polarización (*R18* y *R19*) en las líneas de datos (SDA) y reloj (SCL) de 10 kOhm. Con esto se garantiza el adecuado funcionamiento de este bus de comunicación.

Como nuestro diseño utiliza motores y elementos inductivos, puede aparecer inestabilidad en la alimentación y un alto ruido eléctrico. Por este motivo se implementa un filtro de ruido en la línea de alimentación de 5V lo más cercano posible al controlador de los servomotores. Haciendo uso de un condensador de filtrado de 100nF (*C14*) se dispone de una tensión nominal lo más estable posible.

4.5.5. Pantalla informativa

El dispositivo debe contar con una interfaz de salida informativa que muestre el estado de los parámetros del sistema. Para este proyecto se hace uso de una pantalla LCD de 20x04 (figura 84), conectada a la PCB a través de una tira de pines hembra (J3).



Figura 84 - Pantalla LCD de dimensiones 20x04.

Esta pantalla requiere 8 líneas de datos, por lo que se decide conectarla a un expansor de 8 bits modelo PCF8574 (U2). De esta manera es posible aprovechar el bus ya implementado para el control de los servomotores y gestionar completamente la pantalla sin ocupar pines adicionales. Se dispone también de un potenciómetro (RV5) que se utiliza para controlar el brillo y nivel de saturación de la pantalla. Este bloque queda definido como se muestra en la figura 85.

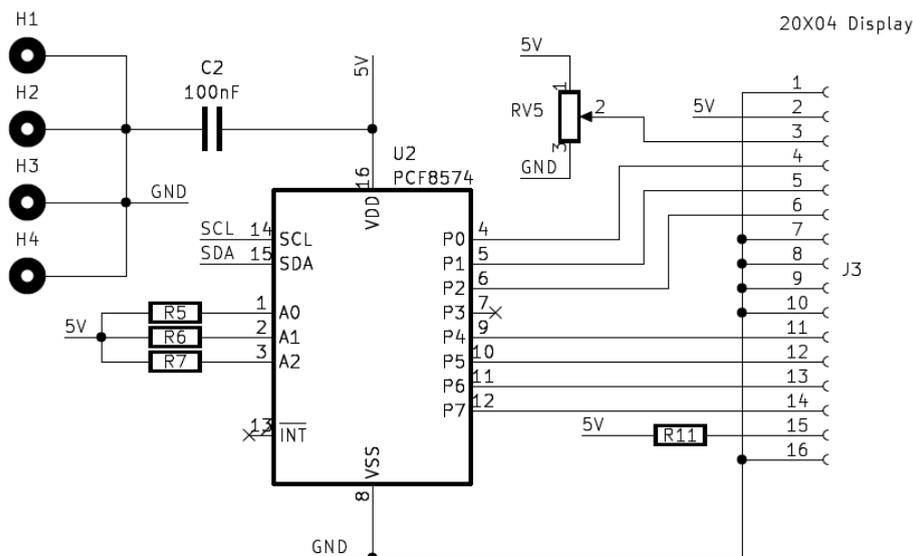


Figura 85 - Diseño electrónico del bloque de la pantalla LCD de dimensiones 20x04.

4.5.6. Interfaz de control del dispositivo

Uno de los bloques más importantes de este producto es el encargado de proporcionar al usuario la interfaz adecuada para el control del dispositivo. Esta interfaz debe permitir al usuario interactuar con el brazo robótico, controlando el movimiento de cada articulación por separado o en conjunto.

Se dispone de dos pulsadores (*SW2* y *SW3*) que permiten realizar el control de la *pinza*, permitiendo su apertura y cierre, y el guardado de posición de todo el brazo en dicho instante, almacenando en la RAM del microcontrolador los valores de los ángulos que tienen los servomotores, el número de pasos que lleva el motor paso a paso y el estado en el que se encuentra la *pinza*, abierta o cerrada. Ambos se encuentran montados en configuración pull-up con 2 resistencias de 10 KOhm (*R9* y *R10*).

Por otro lado, el encoder con pulsador (*SW1*) es el encargado de permitir al usuario moverse por la interfaz de la pantalla en el modo **Automático**, así como permitir el giro de la base del brazo, moviendo el motor paso a paso, en el modo **Manual**. Estos modos se explicarán más adelante. El encoder dispone de un pulsador conectado en configuración pull-up, así como dos líneas de datos del encoder rotativo (CLK y DT) conectadas a pines de interrupción del microcontrolador.

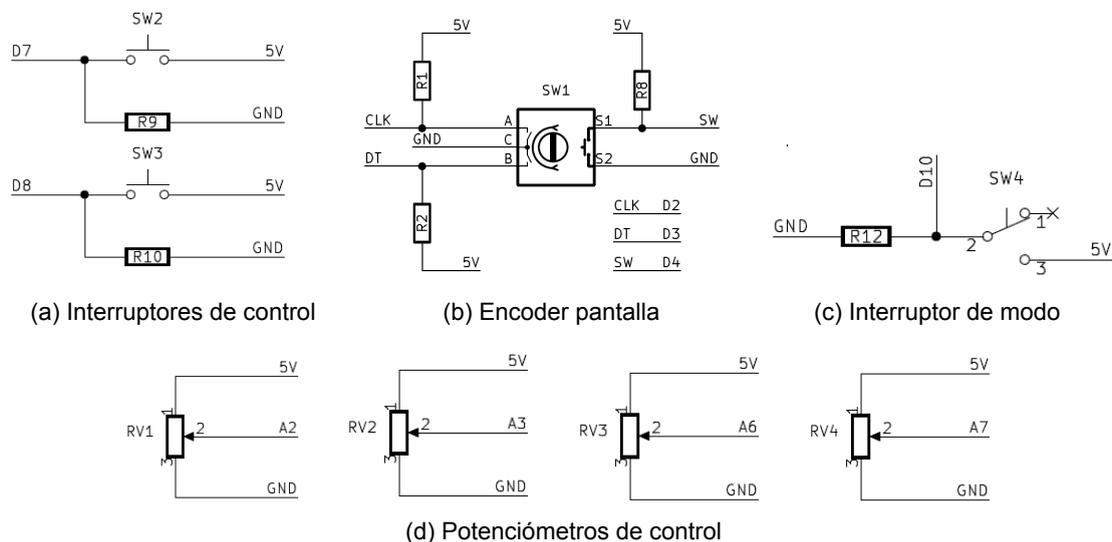


Figura 86 - Diseño electrónico del bloque de control del dispositivo.

Se dispone también de un interruptor (*SW4*) conectado al pin D10 del microcontrolador que permite intercambiar entre los modos **Automático** y **Manual**.

Finalmente, 4 potenciómetros de 10K Ohm, conectados a 4 entradas analógicas de 10 bits del microcontrolador, permiten al usuario mover en el rango de 0° a 180° los

servomotores instalados en las 4 articulaciones del brazo robótico educativo.

4.5.7. Multiplexado y comunicaciones

Este bloque es el encargado de permitir la programación y comunicación del dispositivo con otros equipos o herramientas. En primer lugar, se determina el protocolo de comunicación empleado, que en este caso será serial, haciendo uso de la salida de la UART del microcontrolador ubicada en los pines D0 y D1 del microcontrolador (Tx y Rx). Como se dispone únicamente de esta UART, es necesario hacer uso de una multiplexación (figura 87). Este dispositivo dispone de 3 medios de comunicación diferentes: comunicación USB, comunicación Bluetooth y comunicación WiFi.

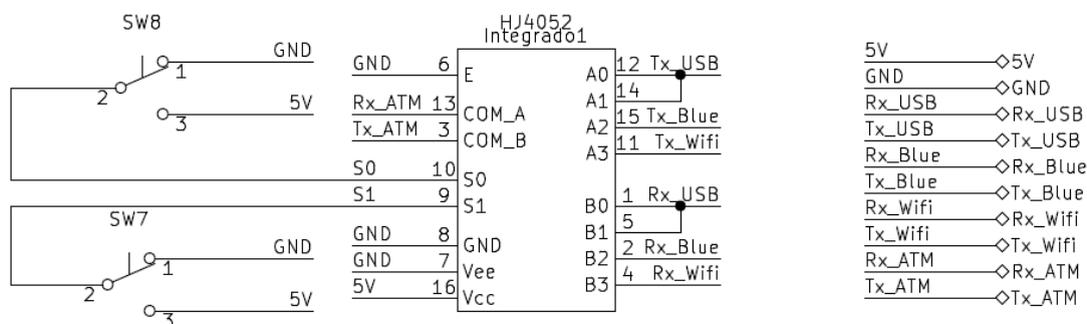


Figura 87 - Diseño electrónico del bloque del Multiplexor.

Dado que se tienen 3 opciones de multiplexación y deben multiplexarse en simultáneo 2 canales independientes (Tx y Rx), se hace necesario disponer de un multiplexor doble (de dos canales) de 2 bits. El integrado seleccionado es el HJ4052 [52], un multiplexor analógico dual con 2 bits de control que puede operar a partir de 3,3 V, cuyas características de operación se muestran en la figura 88.

			MIN	MAX	UNIT
$V_{CC} - V_{EE}$			-0.5	10.5	V
V_{CC}	DC Supply voltage		-0.5	7	V
V_{EE}			0.5	-7	V
I_{IK}	DC input diode current	$V_I < -0.5V$ or $V_I > V_{CC} + 0.5V$	-20	20	mA
I_{OK}	DC switch diode current	$V_I < V_{EE} - 0.5V$ or $V_I > V_{CC} + 0.5V$	-20	20	mA
	DC switch current ⁽²⁾	$V_I < V_{EE} - 0.5V$ or $V_I > V_{CC} + 0.5V$	-25	25	mA
I_{CC}	DC V_{CC} or ground current		-50	50	mA
I_{EE}	DC V_{EE} current		-20		mA
V_{SEL} or V_{EN}	Logic control input pin voltage (\overline{EN} , Ax, SELx)		-0.5	30	V
T_{JMAX}	Maximum junction temperature			150	°C
T_{LMAX}	Maximum lead temperature	Soldering 10 s		300	°C
T_{stg}	Storage temperature		-65	150	°C

Figura 88 - Características del HJ4052 [52].

Para la conmutación entre las diferentes interfaces de comunicación, se decide incluir dos conmutadores (*SW8* y *SW9*) que estarán conectados de forma directa a las líneas selectoras *S0* y *S1* y alternarán entre *GND* y *VCC* para proporcionar ambos valores binarios. Teniendo en cuenta el montaje mostrado en la figura 87, la tabla de verdad que representa la selección del medio de comunicación se muestra en la tabla 4.13.

Tabla 4.13 - Tabla de verdad del HJ4052

SW7 (S1)	SW8 (S0)	Comunicación empleada
0	0	USB
0	1	USB
1	0	Bluetooth
1	1	WiFi

Una vez definida la forma de alternar entre los diversos modos de comunicación, se debe implementar cada una de estas interfaces. Se comienza con la comunicación USB (figura 89), para la que se hace uso de un conector USB tipo C y un convertor de serial TTL a USB, concretamente el transceiver serial CH340C (*U4*), que toma de forma directa las líneas Tx y Rx del microcontrolador y las adapta para ser transmitidas por los pines *D+* y *D-* del conector USB tipo C (*J8*). Para una mejor interpretación de la comunicación, se han instalado dos leds indicadores (*D1* y *D2*) que ofrecen información del estado de las líneas Rx y Tx, respectivamente.

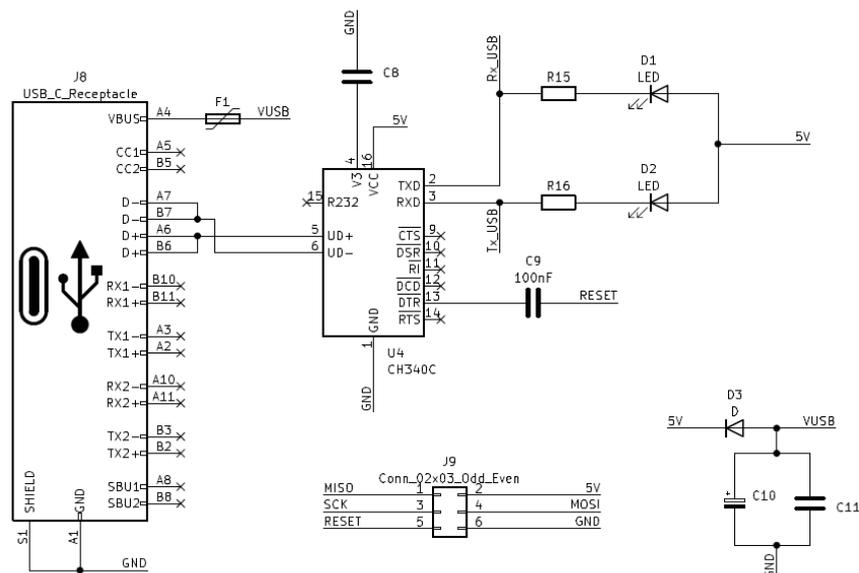
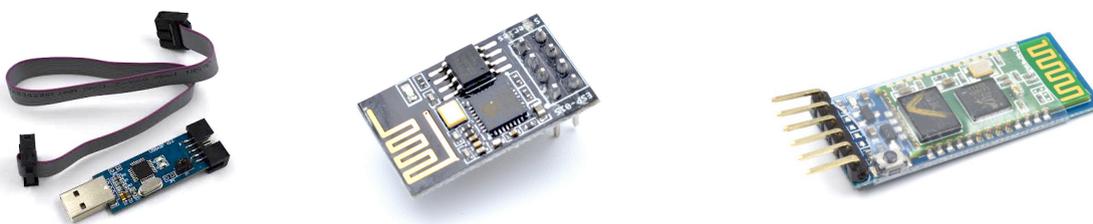


Figura 89 - Diseño electrónico del bloque de la conexión USB.

Asimismo, dado que el conector USB proporciona una alimentación de 5V y que el dispositivo podría estar alimentado desde el conector DC-Jack de forma simultánea, debe evitarse que fluya corriente desde la salida del regulador LM2596 hacia el conector USB. Para ello, se dispone un diodo (*D3*) entre la tensión del USB (*VUSB*) y la tensión de 5V proporcionada por el regulador conmutado, de forma que siempre que el LM2596 se encuentre trabajando, y por tanto ofreciendo una tensión de 5V exacta frente a los 4,5 o 4,8V que proporciona el USB, el diodo se encuentre en corte y no fluya corriente desde la fuente de alimentación hacia el USB del PC. Por el contrario, si la toma del DC-Jack no se encuentra conectada, la salida del regulador LM2596 será de 0V, por lo que el diodo *D3* conducirá desde que haya tensión en *VUSB*, y por tanto, el dispositivo quedará alimentado desde el USB.

Con el objetivo de permitir la adecuada carga del bootloader de Arduino, que permitirá que el microcontrolador del brazo robótico sea detectado por el IDE de Arduino como una placa Arduino UNO, es necesario disponer de un zócalo (*J9*) para hacer uso de un programador ICSP²⁰ (figura 90.a) que emplee los pines SPI y el pin de RESET y permita la carga del firmware del microcontrolador.

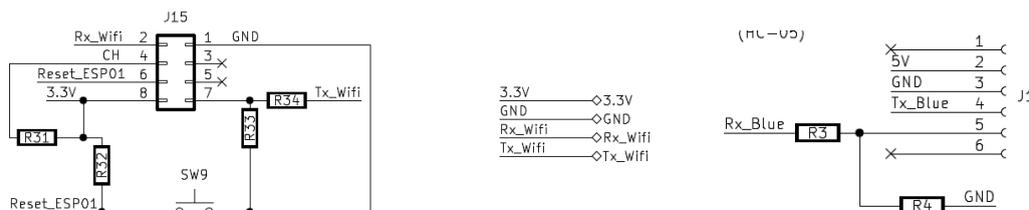
Por otro lado, las comunicaciones Bluetooth y WiFi se realizan haciendo uso de módulos específicos para ello como es el caso del módulo Bluetooth HC-05 (figura 90.c) y el módulo WiFi ESP-01 (figura 90.b).



(a) Programador ICSP (b) Módulo ESP-01 (WiFi) (c) Módulo HC-05 (Bluetooth)

Figura 90 - Programador y módulos WiFi y Bluetooth [53].

Para la conexión de estos módulos, la PCB dispone de zócalos que permiten de forma rápida conectar y emplear estos módulos, tal y como se muestra en la figura 91.



(a) Zócalo para módulo ESP-01 (WiFi) (b) Zócalo para módulo HC-05 (Bluetooth)

Figura 91 - Diseño electrónico de los módulos WiFi y Bluetooth.

4.5.8. Esquema de la PCB Auxiliar

Con el objetivo de permitir conectar diversos efectores finales en el brazo robótico y simplificar el proceso de montaje y desmontaje, se decide diseñar y fabricar una pequeña Placa de Circuito Impreso que irá ubicada en el extremo del brazo y proveerá al dispositivo de diversos conectores JST de 2,54mm que permitan de forma sencilla conectar servomotores y sensores analógicos y digitales que serán definidos en el futuro (fuera del alcance de este TFM). El esquemático de esta PCB se observa en la figura 92.

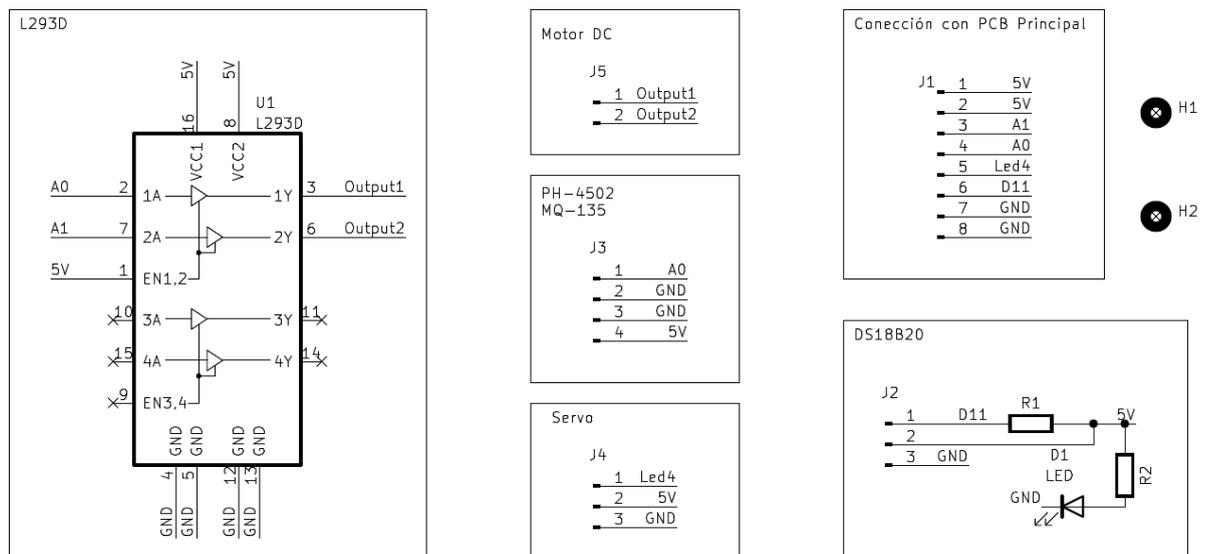


Figura 92 - Diseño electrónico de la PCB Auxiliar.

Esta PCB dispone de un conector JST de 8 pines (*J1*) que permite realizar la conexión entre los módulos instalados en el efector final y el controlador principal (la placa base), recibiendo las señales de control de motores desde el microcontrolador y enviando la información monitorizada por los sensores instalables en la punta. También incorpora un LED indicador de encendido (*D1*) con el que conocer si la placa electrónica está alimentada.

Adicionalmente se disponen conectores JST para sensores de tipo analógico como es el caso de los módulos medidores de PH para Arduino o sensores de gases de tipo MQ (*J3*) que están conectados de forma directa al pin A0 del ATmega 328P. Otros conectores importantes son los destinados a los motores, tanto el dedicado a un motor de corriente continua (*J5*) que proviene de la salida del driver de motores L293D (*U1*), como el destinado a un servomotor (*J4*) que proviene de la salida número 5 (*Led4*) del controlador PWM PCA9685PW. Finalmente, se ha dispuesto un conector para un sensor digital como podría ser el medidor de temperatura DS18B20, implementado con la resistencia de pull-up necesaria.

4.5.9. Conjunto de PCBs diseñadas

Una vez finalizado el esquemático global del dispositivo (tanto la Placa Base como la PCB Auxiliar), se procede a elaborar el diseño de las PCB, tratando de minimizar el tamaño de las mismas y buscando una disposición adecuada e intuitiva de los conectores y las interfaces de control. Tras diversas pruebas y diseños realizados hasta llegar a un modelo adecuado, el aspecto de las PCBs son los que se muestran en los renders de las figuras 93 y 94.

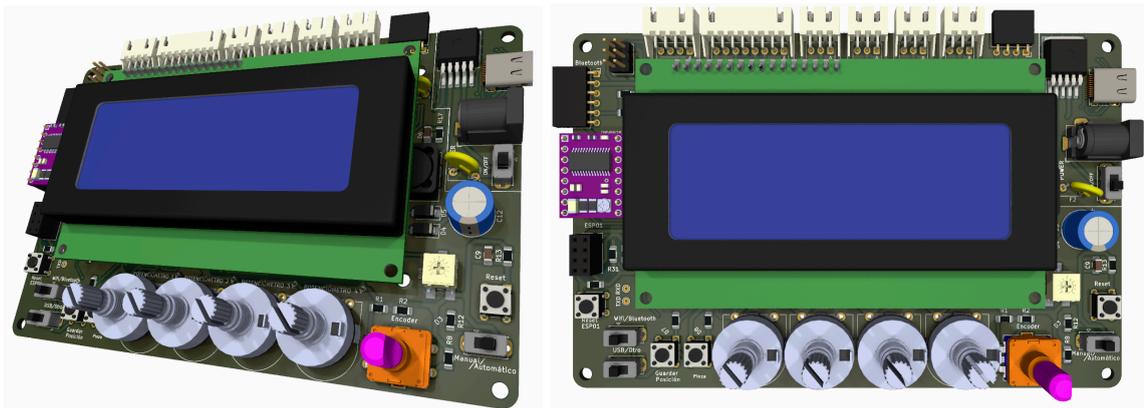


Figura 93 - Render de la PCB principal del brazo robótico.

Con respecto a la placa base del brazo robótico, los potenciómetros, el encoder y demás elementos de control e interacción se ubican en la franja inferior de la misma. Por otro lado, los dos interruptores que permiten conmutar entre los diferentes modos de comunicación se localizan en la esquina inferior izquierda, acompañados del zócalo para el módulo WiFi y el botón de *reset* de este dispositivo. En la zona inmediatamente superior al zócalo de dicho módulo se ubica el driver controlador de motores paso a paso y justo a su derecha, en la zona central del dispositivo, la pantalla LCD. Con respecto a las entradas de alimentación, en la esquina superior derecha se localizan los conectores USB (tipo C) y el conector DC-Jack. Finalmente, toda la franja superior de la placa base se destina a la ubicación de conectores JST acodados que permiten conectar los diversos actuadores (servomotores y motores paso a paso), así como la PCB auxiliar que se ubica en el efector final.

Por otro lado, la PCB auxiliar presenta un aspecto mucho más sencillo, con un conector JST central de 8 pines destinado a la conexión con la placa base y conectores JST distribuidos por la PCB de 2, 3 y 4 pines para motores de continua, sensores digitales o servomotores y sensores analógicos, respectivamente. Se observa, también en la zona central la ubicación del circuito integrado de transistores Darlington (L293) encargado de controlar el motor de corriente continua que puede ser conectado a esta PCB.

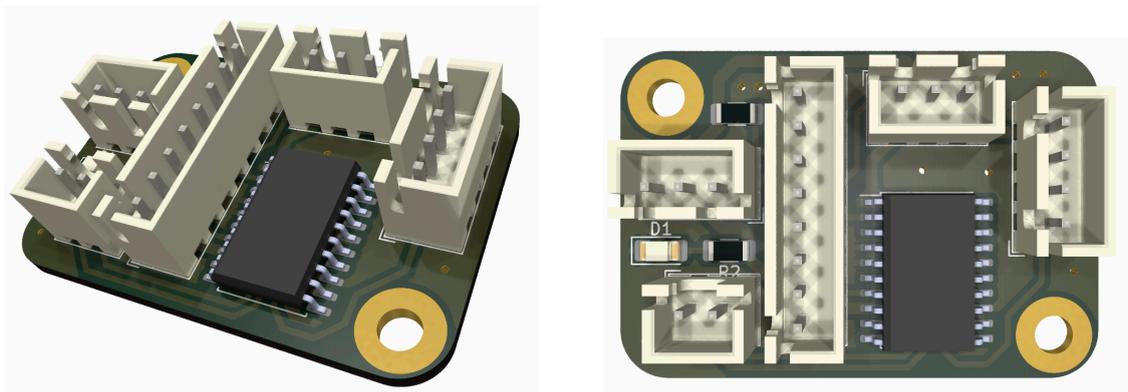


Figura 94 - Render de la PCB auxiliar para el efector final intercambiable del brazo robótico.

La primera de las PCBs mostradas, la del controlador principal, dada su complejidad y elevada cantidad de pines, pistas, espaciado entre pistas, *pitch*²¹ de los componentes SMD²², etc. debió ser fabricada a través de métodos de fabricación industriales para los que se recurrió a una empresa experta en esta área para la fabricación de la misma. Por otro lado, la PCB Auxiliar ha sido fabricada de forma artesanal en el fablab²³, tal y como podrá verse en apartados posteriores de esta memoria.

4.6. Fabricación electrónica

Como se ha mencionado anteriormente, en este TFM se ha fabricado directamente la PCB Auxiliar, debido a que tiene una distribución de componentes en una sola cara de la PCB, es de pequeñas dimensiones, y sus pistas tienen una separación adecuada para proceder a una fabricación 'artesanal'. En cambio, para la PCB situada en la base del brazo, al ser de un tamaño superior, con una mayor densidad de componentes y una distribución de pistas más complejas (grosor más pequeño, menor espaciado entre pistas, mayor número de taladros, etc.), se decide solicitar la fabricación a nivel industrial a través de un proveedor especializado. Sin embargo, para reducir costes de fabricación y ensamblaje, se decide pedir la PCB sin los componentes electrónicos THD²⁴, ya que éstos pueden ser soldados en el laboratorio. Por ello, una vez se tiene dicha PCB, el siguiente paso será soldar los componentes que faltan (figura 95).

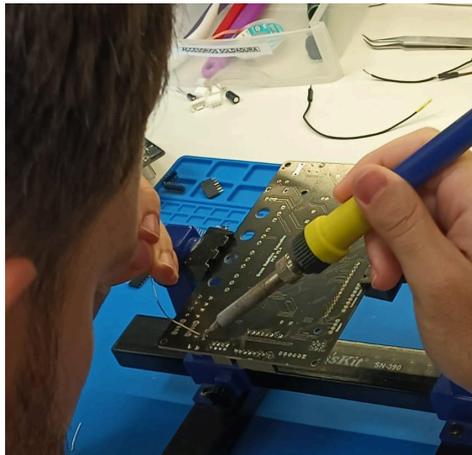


Figura 95 - Soldando componentes de la PCB

De esta forma, ya se tiene la placa completamente soldada en la figura 96.

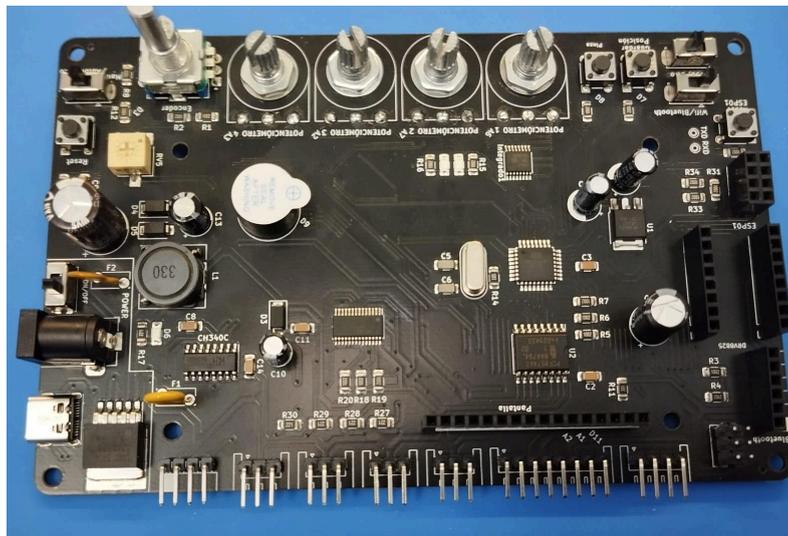


Figura 96 - PCB con componentes soldados

La fabricación de la PCB Auxiliar se realiza mediante un proceso de ataque químico en el laboratorio sobre un panel de cobre monocapa pintado de negro en el que se retirará mediante grabado láser la pintura de las zonas de cobre que deben ser atacadas por el ácido, dejando únicamente las pistas diseñadas sobre el cobre. Tras el ataque químico, se retira la pintura y se procede al taladrado manual de los agujeros de la PCB.

En primer lugar, es necesario obtener los ficheros GERBER de la PCB desde el propio software de diseño electrónico, el KiCAD. Al tratarse de un diseño monocapa que será fabricado mediante un proceso artesanal, sin máscara de soldadura ni serigrafía y que será taladrado a mano, solo se necesita la capa inferior de cobre (*capa bottom*), generando el fichero que se muestra en la figura 97.

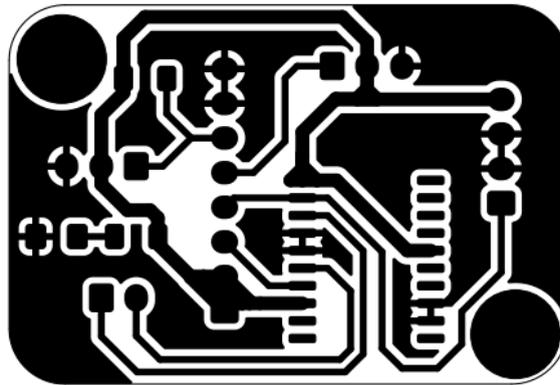


Figura 97 - Pistas de la PCB Auxiliar

Una vez exportado este archivo, es necesario procesarlo en el software FlatCAM con el fin de obtener los trazados necesarios para la cortadora láser.

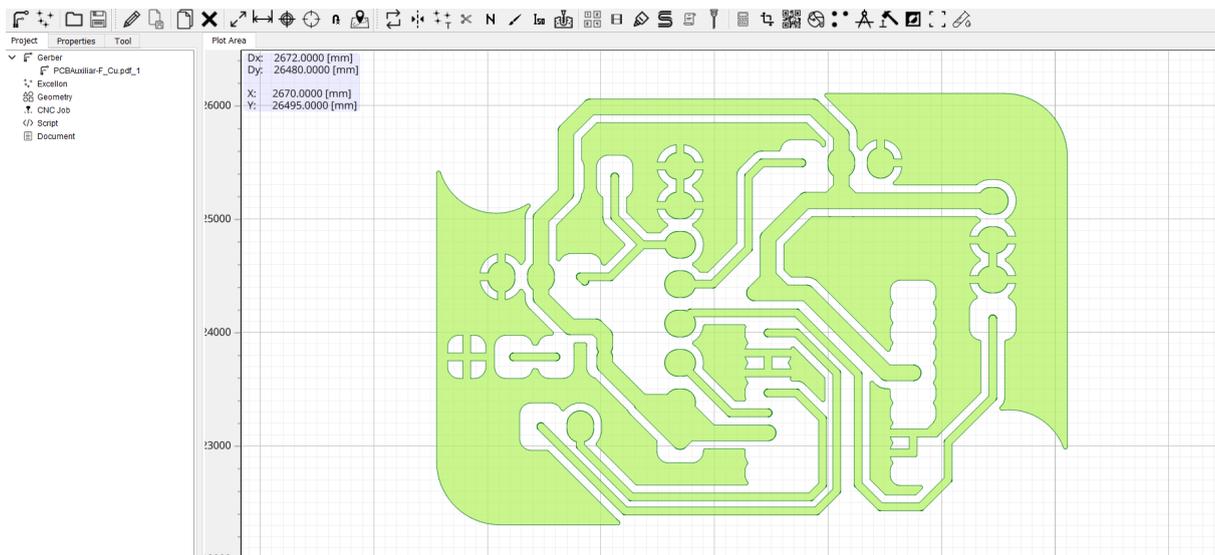


Figura 98 - Preparación del trazado de pistas de la PCB Auxiliar en FlatCAM.

Así, con el trazo realizado, se puede importar el archivo generado en el RDWorks, indicando los parámetros de potencia y velocidad necesarios para realizar el trabajo correctamente. Se debe garantizar con el ajuste de estos parámetros que la potencia sea suficiente y adecuada para eliminar la capa de pintura negra superficial sin dañar la placa de cobre.

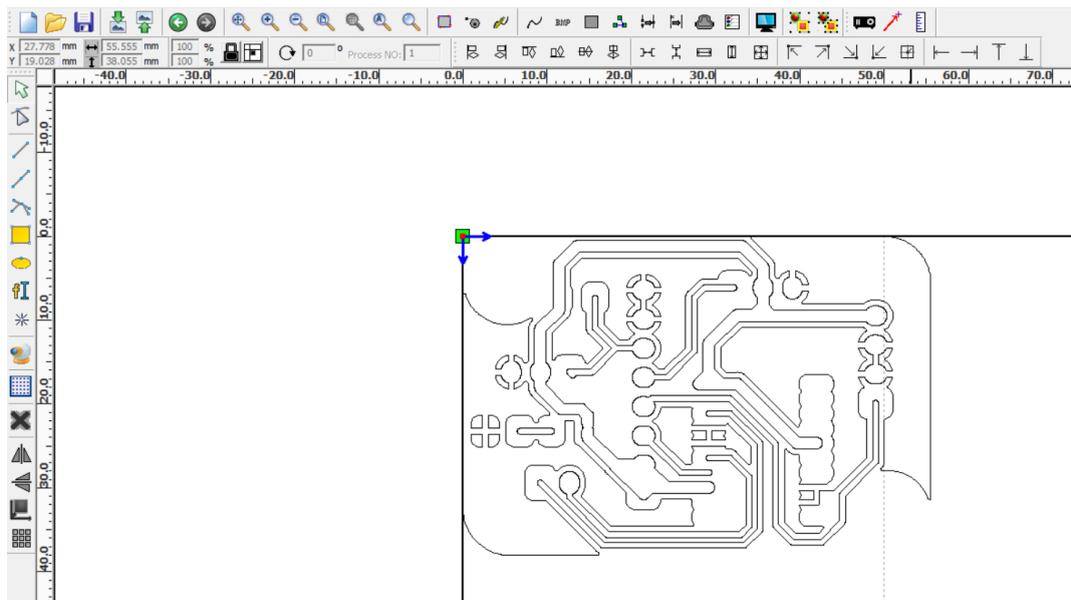


Figura 99 - Pistas de la PCB Auxiliar en RDWorks.

El siguiente paso consiste en colocar el panel de cobre pintado en la cortadora láser, y realizar el grabado. Cuando esté totalmente grabado el circuito impreso, se sumerge la placa en el baño químico, tal y como se muestra en la figura 100.

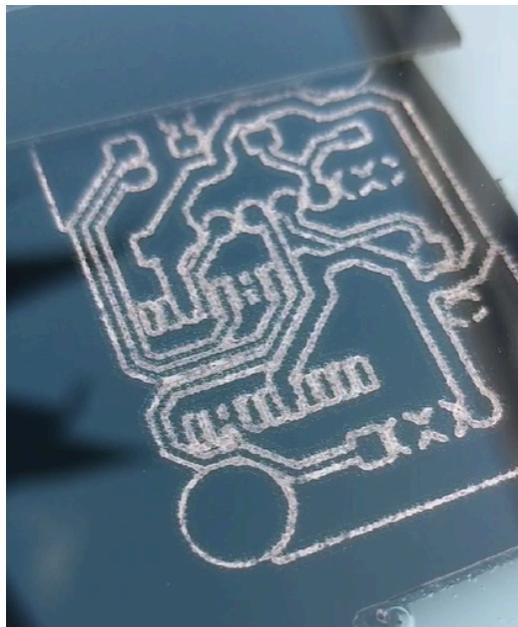


Figura 100 - PCB sumergida en el tanque de ataque químico.

Una vez ha finalizado la reacción química y todo el cobre de las zonas expuestas ha sido eliminado, se extrae la placa de cobre, se sumerge en agua destilada para su limpieza y se procede a eliminar la pintura de la superficie, quedando la PCB como en la figura 101.

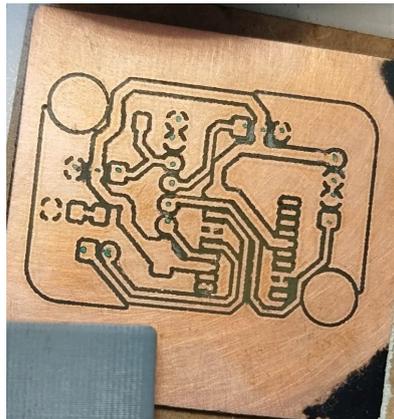


Figura 101 - Pistas de PB en cobre sin agujeros

Ya con la PCB totalmente grabada, se procede al taladrado de los agujeros necesarios para soldar los componentes. Tras ello se sueldan tanto los componentes THD en los agujeros realizados como los SMD en los pads dedicados para ello. El resultado se muestra en la figura 102.

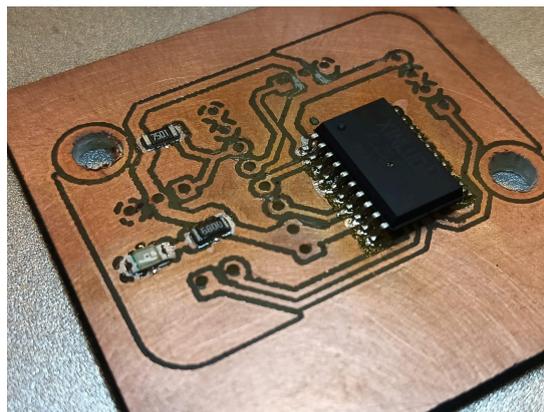


Figura 102 - PCB Auxiliar

4.7. Diseño software

Todo el desarrollo software de este dispositivo ha sido realizado en C++ haciendo uso del IDE de Arduino y las librerías específicas compatibles con el microcontrolador Arduino UNO. Al tener este producto un fin claramente educativo, es muy importante que todo el desarrollo software se pueda realizar con herramientas *OpenSource*, siendo accesible en el futuro para cualquier usuario. En este sentido, a la finalización de este TFM se obtiene un conjunto de librerías y códigos específicos, que permitirán que cualquier usuario que lo adquiera pueda ejecutarlo con el código diseñado o pueda realizar las modificaciones que desee para que el producto realice tareas personalizadas, de forma que sirva como una herramienta práctica con la que enseñar no solo robótica, sino electrónica y programación.

A lo largo de este apartado se definirán los diferentes códigos de pruebas realizados para comprobar el funcionamiento del prototipo, así como el código funcional del brazo y el cálculo de la cinemática inversa del dispositivo. A medida que la electrónica era mejorada y se añadían diferentes componentes y funcionalidades, se ha ido modificando el código principal hasta obtener una implementación completa y funcional de todo el brazo robótico. Todos los códigos realizados se pueden consultar en detalle en el Anexo IV.

4.7.1. Códigos de prueba realizados

A medida que se iba montando el prototipo de brazo robótico, se hacía necesario probar las diferentes partes del dispositivo, verificando que todo funcionaba de forma adecuada y permitiendo depurar el funcionamiento de cada elemento del dispositivo por separado. Para la realización de esta tarea, se realizaron en total 6 códigos en C++ diferentes, destinados a probar cada elemento del brazo. Los nombres de estos 6 códigos, localizados en el Anexo IV, se muestran a continuación acompañados de una breve descripción de las tareas que realizan.

- **Test_module_servos.ino:** Prueba de uso de módulo de control para servomotores.
- **Test_move_servos.ino:** Prueba de uso de varios servomotores en simultáneo.
- **Test_move_stepper.ino:** Prueba de uso de motor paso a paso controlado por el driver DRV8825.
- **Test_move_automático.ino:** Prueba de uso de movimiento automático.
- **Test_servos_velocidad.ino:** Prueba de modificación de velocidad de los servos.
- **Test_lcd.ino:** Prueba de uso de una pantalla LCD 20x04.

4.7.2. Cálculo de cinemática inversa

A parte de las pruebas descritas anteriormente, se desarrolla una librería para el cálculo de cinemática inversa llamada *CinematicaInversa*. Esta librería consiste en la introducción de las coordenadas X, Y y Z, a las que se quiere llevar el extremo del brazo, la longitud de cada segmento de los brazos, los límites máximos y mínimos a los que llegan los servomotores y los diferentes grados en los que se encuentra los servomotores en ese momento con el fin de calcular el ángulo en el que tienen que estar los diferentes servomotores para alcanzar el punto indicado.

Para poder crear la librería se crearon tres archivos diferentes:

- **CinematicaInversa.h:** Es el fichero de cabecera que contiene la definición de las funciones y las variables que se utilizarán en el código. En él se incluyen las diferentes variables que se comentaron anteriormente y que afectan al cálculo de la cinemática inversa. Este código es el siguiente:

```
#include "Arduino.h"

void CI (float x, float y, float z, int l1, int l2, int l3,
int limite_min, int limite_max, int &Paso0, float &Grado1,
float &Grado2, float &Grado3);
```

Dentro de este fichero, en la definición de la función se puede encontrar:

- Parámetros de entrada.
 - Coordenadas XYZ.
 - Distancia entre ejes de los servomotores del brazo robótico, siendo estas:
 - l1, del *hombro* al *codo*.
 - l2, del *codo* a la *muñeca*.
 - l3, de la *muñeca* al extremo del brazo.
 - El ángulo inferior del servomotor, en los servomotores utilizados es de 0°.
 - El ángulo superior del servomotor, en los servomotores utilizados es de 180°.
- Parámetros de entrada/salida (están indicadas con un “&” antes del nombre de la variable).
 - El número de pasos que lleva el motor paso a paso.
 - El ángulo en el que se encuentra el servomotor situado en el *hombro*.
 - El ángulo en el que se encuentra el servomotor situado en el *codo*.
 - El ángulo en el que se encuentra el servomotor situado en la *muñeca*.

Entre los parámetros mencionados, los más importantes son los parámetros de entrada/salida, debido a que son las variables que retornan una vez comience a ejecutarse la librería y las que se utilizarán posteriormente en el código principal para conseguir que el extremo del brazo alcance la nueva posición. Esto se hace gracias a que en la librería el paso de estas variables

se realiza por referencia y sus valores se sobrescriben directamente en el código fuente.

- **CinematicaInversa.cpp:** Es el archivo fuente que contiene el código básico de la librería. Aquí se incluyen el resto de librerías necesarias, así como variables y otras funciones. El funcionamiento de la función `void CI()` se muestra en el flujograma de la figura 103.

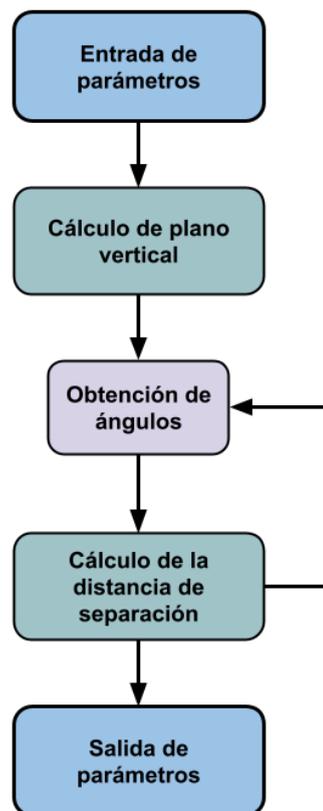


Figura 103 - Flujograma de la librería *CinematicaInversa*.

- Al comienzo, los datos de entrada son añadidos a los datos del código.
- Conocidas las coordenadas y los diferentes parámetros, se procede a calcular el número de pasos que tiene que dar el motor paso a paso para que el movimiento de los servomotores se encuentren en el mismo plano que el punto a alcanzar.
- Ya estando el brazo posicionado en el mismo plano que el punto a alcanzar, mediante cinemática inversa se van obteniendo los nuevos ángulos que tienen que tener los servomotores.
- Se calcula la distancia a la que se encuentra el brazo, y en caso de que no se alcance el punto final, se realiza otra iteración y se vuelven a calcular los nuevos ángulos que tienen que tener los servomotores.

- Una vez que se comprueba que el extremo del brazo alcanza el punto final, se sobrescriben los parámetros de pasos del motor paso a paso, y los ángulos de los servomotores.
- **keywords.txt:** Es el archivo de palabras clave. En este se indica el color de la sintaxis en el Arduino IDE según el grupo al que esté asociado. El grupo `KEYWORD1` resalta la palabra en color naranja y se emplea para clases y tipos de datos, mientras que `LITERAL1` la colorea en azul y se emplea para constantes. Este archivo queda tal que así:

```
CinematicaInversa    LITERAL1
CI                    KEYWORD1
```

Esta librería se destaca porque, en vez de trabajar en un plano XY, se trabajan en el espacio tridimensional (XYZ), calculando los ángulos de los servomotores y el número de pasos que tiene que tener el *stepper*, para alcanzar la coordenada indicada.

4.7.3. Código principal del dispositivo

A lo largo de este apartado, se presenta y explica el modo de funcionamiento del código principal que ejecuta el microcontrolador ATmega 328P y que gestiona el movimiento de todo el brazo robótico así como los distintos modos de funcionamiento que presenta. El algoritmo de funcionamiento del dispositivo puede observarse mediante el flujograma de la figura 104.

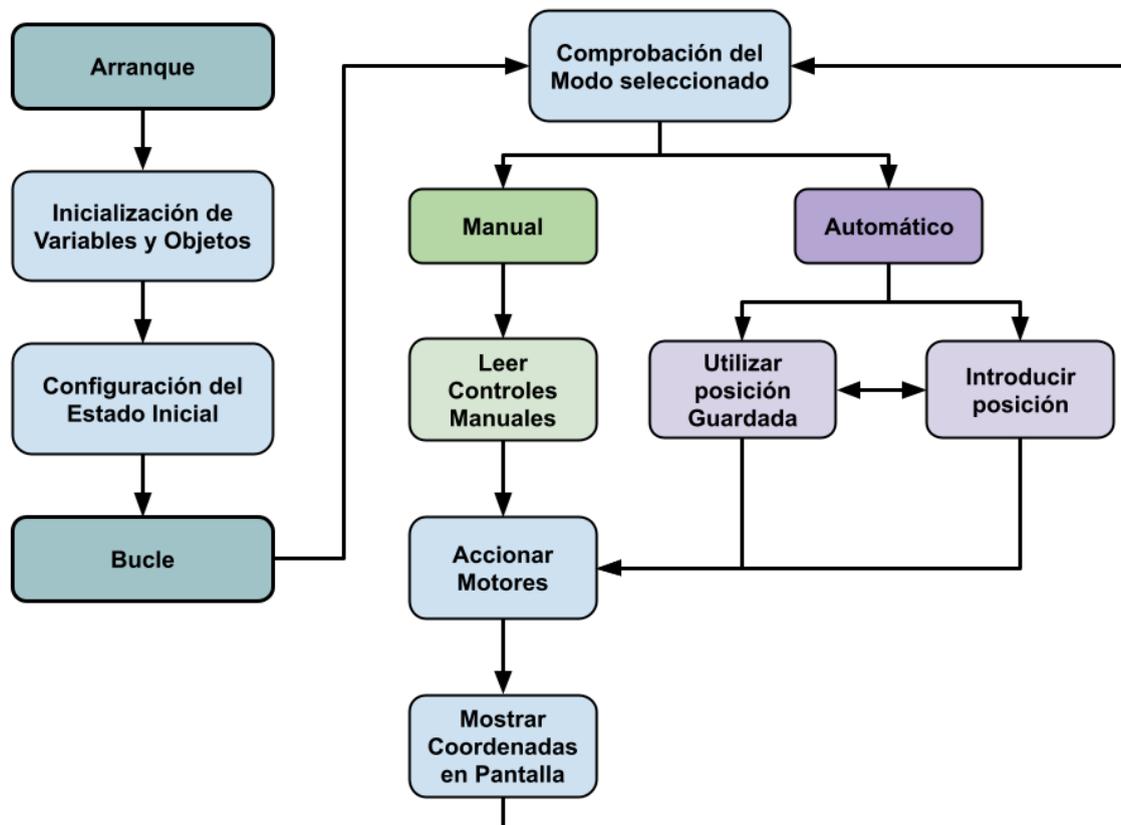


Figura 104 - Flujograma del funcionamiento del código principal.

En primer lugar, una vez el brazo es arrancado, se procede a la inicialización de variables y objetos, comenzando con incluir las librerías necesarias, las cuales son:

```
#include <math.h>
```

Se encarga de dar la posibilidad de utilizar fórmulas matemáticas.

```
#include <Wire.h>
```

Facilita la comunicación I2C entre el microcontrolador y otros dispositivos compatibles con I2C, como sensores, pantallas y otros microcontroladores.

```
#include <CinematicaInversa.h>
```

Librería creada para el cálculo de los ángulos de los servomotores para alcanzar un punto deseado.

```
#include  
<Adafruit_PWMServoDriver.h>
```

Permite utilizar el PCA9684PW para mover los servomotores.

<code>#include <PinChangeInterrupt.h></code>	Permite la utilización del encoder.
<code>#include <AccelStepper.h></code>	Sirve para mover el motor paso a paso indicando el número de pasos.

También definiendo los pines del microcontrolador que se utilizarán:

<code>#define CLK 2</code>	Pin utilizado para la rotación del encoder, cuando se mueve
<code>#define DT 3</code>	Pin utilizado para la rotación del encoder, dirección de movimiento
<code>#define SW 4</code>	Pin utilizado para el botón del encoder
<code>#define dirPin 5</code>	Pin utilizado para indicarle la dirección de giro al motor paso a paso
<code>#define stepPin 6</code>	Pin utilizado para indicar el número de pasos al motor paso a paso
<code>#define GuardarPosicion 7</code>	Pin utilizado por un botón para guardar posiciones
<code>#define BPinza 8</code>	Pin utilizado para un botón que abre o cierra la pinza
<code>#define Zumbador 9</code>	Pin utilizado para un zumbador
<code>#define Manual 10</code>	Pin utilizado para indicar cuando se encuentra en Manual o Automático

Finalmente, se crean las variables utilizadas a lo largo del código y los diferentes objetos como es el caso de los servomotores, el motor paso a paso, el encoder y la pantalla.

Tras ello, se realiza la configuración de los pines digitales y analógicos empleados, indicado si se trata de entradas o salidas

<code>pinMode (A2, INPUT)</code>	Potenciómetro
<code>pinMode (A3, INPUT);</code>	Potenciómetro
<code>pinMode (A6, INPUT);</code>	Potenciómetro

<code>pinMode(A7, INPUT);</code>	Potenciómetro
<code>pinMode(CLK, INPUT);</code>	Encoder
<code>pinMode(DT, INPUT);</code>	Encoder
<code>pinMode(SW, INPUT_PULLUP);</code>	Encoder
<code>pinMode(dirPin, OUTPUT);</code>	Motor Paso a Paso
<code>pinMode(stepPin, OUTPUT);</code>	Motor Paso a Paso
<code>pinMode(GuardarPosicion, INPUT);</code>	Pulsador
<code>pinMode(Pinza, INPUT);</code>	Pulsador
<code>pinMode(Zumbador, OUTPUT);</code>	Zumbador
<code>pinMode(Manual, INPUT);</code>	Interruptor

Posteriormente, se inician los objetos anteriormente mencionados, como la pantalla LCD, el encoder y el motor paso a paso. Y una vez se ha configurado el estado inicial, se comienza a ejecutar el código en bucle. En esta etapa se comienza comprobando el modo seleccionado entre los disponibles: **Manual** o **Automático**.

En caso de encontrarse el modo **Manual** activado, el código toma las lecturas de las señales proporcionadas por los controles, formado por los cuatro potenciómetros, el encoder y el pulsador para abrir y cerrar la pinza. Tras recibir estas instrucciones, mueve los actuadores, es decir, los servomotores y el motor paso a paso. Finalmente, muestra en pantalla las coordenadas en las que se encuentra el extremo del brazo.

En caso de encontrarse en modo **Automático**, aparecerá en la pantalla LCD un menú donde nos permitirá elegir si queremos introducir una coordenada, o utilizar una coordenada ya memorizada previamente. Las posiciones serán ejecutadas en el mismo orden en el que se memorizaron. Finalmente, se enviarán las instrucciones a los servomotores y al motor paso a paso para que se muevan a las posiciones memorizadas y se mostrará por pantalla las coordenadas de dicha posición.

Una vez han realizado los movimientos del modo **Manual** o finalizado las órdenes dadas en el modo **Automático**, se vuelve a comprobar el estado seleccionado y se repite el bucle.

4.8. Implementación software

El presente apartado tiene por objetivo mostrar el funcionamiento del dispositivo en sus diferentes modos de operación (manual y automático). Para ello, se mostrará la explicación de funcionamiento del dispositivo, acompañado de la imagen de la pantalla LCD con la información adecuada.

Al inicializar el dispositivo aparece en la pantalla LCD un mensaje de inicialización. Este es meramente decorativo y no influye en nada al funcionamiento.



Figura 105 - Interfaz, brazo inicializando

Una vez ha terminado la inicialización, existen dos opciones de funcionamiento: el modo manual o el modo automático. En caso de que el dispositivo se encuentre en modo Manual, se mostrará en la línea superior mediante el mensaje 'Manual = ON'. En este caso, haciendo diversos cálculos de cinemática directa, se muestra en la pantalla LCD las coordenadas en las que se encuentra el extremo del brazo, indicando también el estado de la pinza (abierta o cerrada).

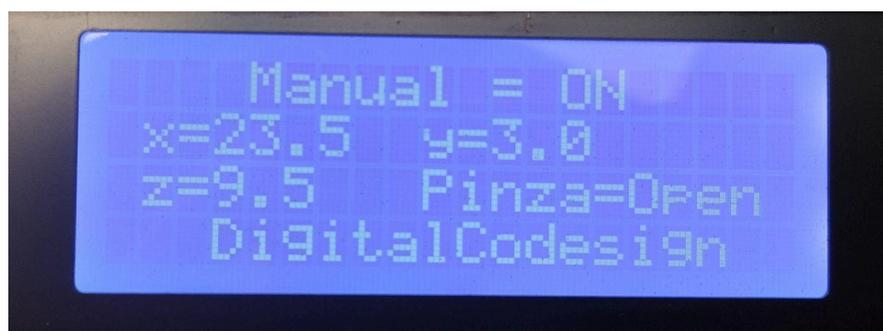


Figura 106 - Interfaz, indicando coordenadas en modo manual

Cuando el interruptor de modo de funcionamiento pasa de modo manual a modo automático, se indicará en la pantalla un mensaje que preguntará si se quiere que la posición a la que se va a mover el brazo sea una posición ya memorizada (estas irán apareciendo de forma ordenada) o si se desea introducir manualmente una nueva coordenada.

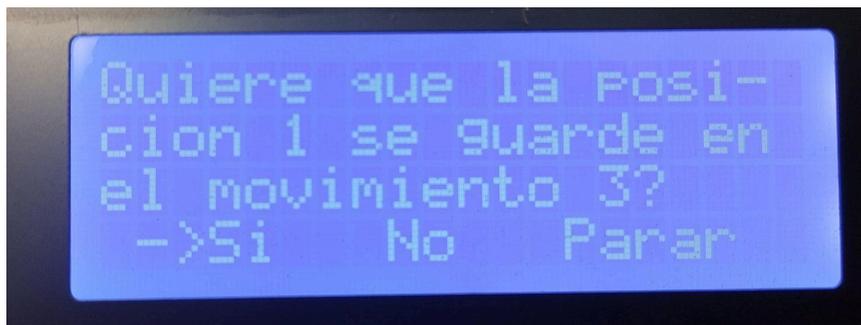


Figura 107 - Interfaz, elegir orden de posiciones

En caso de que indiquemos en la opción "No", irá preguntando las posiciones XYZ y el estado de la pinza, abierta o cerrada, las cuales iremos configurando con el encoder y su pulsador integrado.

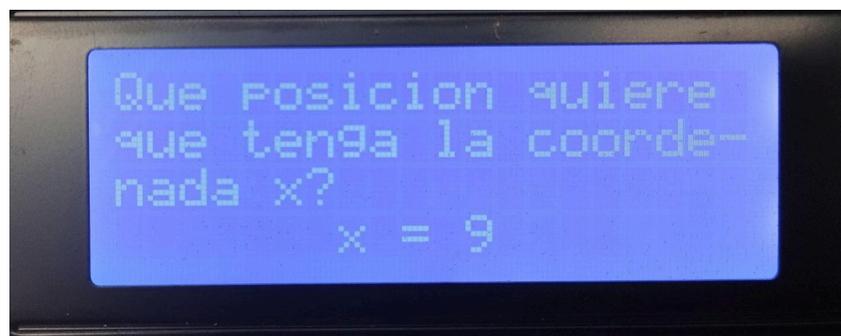


Figura 108 - Interfaz, solicitud de coordenadas XYZ

Una vez se han introducido todas las posiciones, aparecerá por pantalla el conjunto de coordenadas y el estado de la pinza para poder revisar si la configuración introducida es correcta.

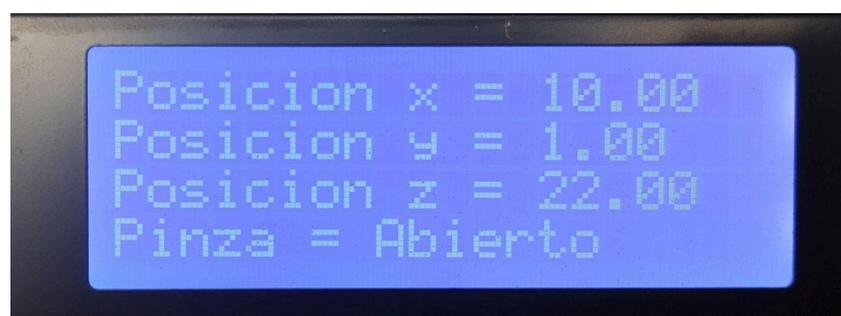


Figura 109 - Interfaz, muestras de coordenadas recogidas

A su vez, con estas coordenadas se utilizará la librería *CinematicaInversa* antes descrita para obtener los ángulos que deben tener los diferentes servomotores y los pasos que tiene que dar el motor paso a paso. En caso de que se seleccione la opción parar, preguntará si queremos que empiece a moverse el brazo de forma automática. En caso de indicar 'Si', el brazo comenzará a moverse.

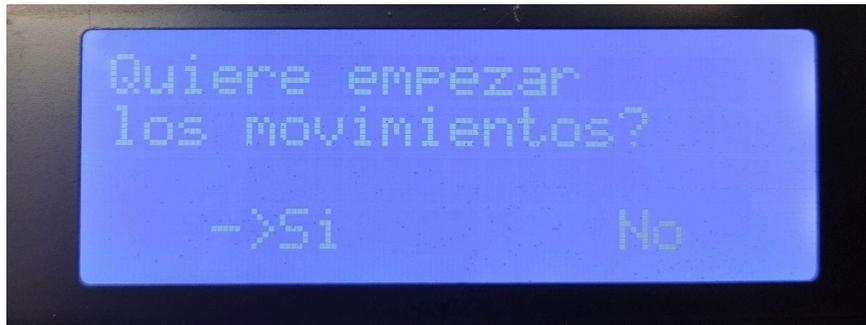


Figura 110 - Interfaz, comienzo de movimiento

5. Presupuesto

El presente apartado incluye un presupuesto detallado del coste de las partes mecánicas, los componentes electrónicos, la mano de obra y el beneficio industrial, acompañado de un breve estudio económico para ver el umbral de disponibilidad.

5.1. Piezas mecánicas

En primer lugar, se comienza calculando el precio de fabricación de los diseños mecánicos realizados. En la Tabla 5.1 y 5.2, se muestran el conjunto de piezas a fabricar y los costes de material, maquinaria y trabajador:

Tabla 5.1 - Coste de fabricación de piezas en impresora 3D

Pieza	CARACTERÍSTICAS					COSTES			
	Peso (g)	Tiempo impresión		Tiempo Trabajador (min)		Material	Tiempo		Total
		Horas	Minutos	Preparación	Postprocesado		Impresora	Trabajo	
Base	216,14	17	46	2	2	9,15 €	3,95 €	1,57 €	14,67 €
Cintura	93,43	7	48	2	2	3,96 €	1,74 €	1,57 €	7,27 €
Hombro	59,73	5	19	2	2	2,53 €	1,18 €	1,57 €	5,28 €
Codo	38,08	3	18	2	2	1,61 €	0,73 €	1,57 €	3,91 €
Muñeca	16,25	1	48	1	1	0,69 €	0,40 €	0,78 €	1,87 €
Pinza	7,97	0	57	1	1	0,34 €	0,21 €	0,78 €	1,33 €
Engranaje Pinza	0,93	0	7	1	1	0,04 €	0,03 €	0,78 €	0,85 €

Tabla 5.2 - Coste de fabricación de piezas en corte láser

Pieza	CARACTERÍSTICAS					COSTES			
	Área (cm ²)	Tiempo corte		Tiempo Trabajador (min)		Material	Tiempo		Total
		Horas	Minutos	Preparación	Postprocesado		Corte	Trabajo	
Tapa	300	0	3	1	1	1,08 €	0,02 €	0,78 €	1,88 €
Piezas Pinza	37,56	0	1,7	1	1	0,14 €	0,01 €	0,78 €	0,93 €

Para las piezas impresas en 3D, el material utilizado es PLA, mientras que el material para el corte láser son planchas de metacrilato de 3mm. A continuación, en la Tabla 5.3 se muestra el coste de cada pieza y el coste total que tiene el conjunto de piezas mecánicas.

Tabla 5.3 - Coste total del conjunto de piezas mecánicas

Pieza	Coste
Base	14,67 €
Cintura	7,27 €
Hombro	5,28 €
Codo	3,91 €
Muñeca	1,87 €
Pinza	1,33 €
Engranaje Pinza	0,85 €
Tapa	1,88 €
Piezas Pinza	0,93 €
Total	37,99 €

5.2. Componentes electrónicos

En cuanto a los componentes electrónicos, es necesario presupuestar el coste de la PCB principal del brazo, la cual fue pedida a un fabricante localizado en China denominado JLCPCB, los componentes que se soldaron en el fablab y los componentes necesarios para la fabricación de la PCB Auxiliar.

Con el objetivo de reducir los costes de la PCB principal, tuvo que pedirse un lote de 5 PCBs, de las cuales 3 de ellas se pidieron sin componentes soldados y 2 venían con todos los componentes SMD montados. El coste total del pedido de las 5 placas sin soldar fue de **10,20 €** mientras que las 2 placas soldadas tenían un precio extra de **88,40 €**. Esto hace que una PCB con el SMD soldador tenga un coste unitario de **46,24 €**. Mientras que los componentes electrónicos faltantes tendrán el precio que se muestra en la Tabla 6.1.

Tabla 6.1 - Coste total de componentes electrónicos de la PCB principal

Componente	Unidades	Coste
Potenciómetro de 10k Ω	4	3,60 €
Encoder rotativo	1	1,55 €

Interruptores deslizante de 2 posiciones	4	1,60 €
Pantalla Icd 2004	1	8,51 €
DC-Jack	1	0,32 €
Pulsador	3	1,05 €
Tira de pines hembra de 2x4	1	0,30 €
Tiras de pines hembra de 8 pines	2	0,46 €
Tira de pines hembra de 16 pines	1	0,25 €
Fusibles rearmable 500mA y 1,5 A	2	0,27 €
Total	20	17,91 €

En cuanto al coste de la PCB Auxiliar, hay que tener en cuenta tanto la fabricación de la placa como de los componentes electrónicos. En cuanto a la fabricación, el coste del material, teniendo en cuenta que se utilizó **30 cm²**, tiene un coste total de **1,12 €**. Mientras que la realización del grabado de las pistas con láser de CO₂ tiene un coste total de **0,85 €**. Finalmente, en la Tabla 6.2, se muestran los precios de los componentes eléctricos de la PCB Auxiliar.

Tabla 6.2 - Coste total de componentes electrónicos de la PCB Auxiliar

Componente	Unidades	Coste
Tira de pines hembra de 8 pines	1	0,23 €
Tira de pines hembra de 4 pines	1	0,30 €
Tira de pines hembra de 3 pines	2	0,36 €
Tira de pines hembra de 2 pines	1	0,17 €
Led rojo	1	0,12 €
Resistencia de 220 Ω	1	0,05 €
L293D	1	0,89 €
Total	8	2,12 €

Esto hace que el precio total de la PCB principal del brazo sea de **64,15 €**, el precio total de la PCB Auxiliar sea de **4,09 €** y el coste de 2 servomotores MG996R y 3 servomotores MG90S sea de **30,17€**. El coste total es de **98,41 €** como se muestra en la Tabla 6.3.

Tabla 6.3 - Coste total de componentes electrónicos

Componente	Coste
PCB principal	64,15 €
PCB Auxiliar	4,09 €
Servomotores	30,17 €
Total	98,41 €

5.3. Mano de obra

Para calcular el coste en mano de obra de la elaboración de este prototipo, es necesario tener en cuenta el coste del empleado para el desarrollo de este proyecto. Para el cálculo de dicho coste se toma la temporalización y la distribución por Paquetes de Trabajo indicadas en el Apartado 4.1. Teniendo en cuenta que el coste/hora de un ingeniero es de aproximadamente **25 €/hora**, en la Tabla 6.4 se mostrarán los paquetes de trabajo, el tiempo empleado en cada uno de ellos y su coste.

Tabla 6.4 - Coste total por paquete de trabajo realizado

Paquete de trabajo	Tiempo empleado	Coste total
PT1	52 horas	1.300 €
PT2	84 horas	2.100 €
PT3	115 horas	2.875 €
PT4	87 horas	2.175 €
PT5	60 horas	1.500 €
PT6	15 horas	375 €
Total	413 horas	10.325 €

Esto hace que el coste total de la mano de obra, para la realización de este proyecto, sea de **10.325 €**.

5.4. Coste total y umbral de rentabilidad

Sabiendo que el coste de las piezas fabricadas es de **37,99 €**, el coste de los componentes electrónicos es de **98,41 €** y la mano de obra es de **10.325,00 €**, hace que el coste total de proyecto ascienda a **10.461,40 €**.

Esto hace que, en caso de que se quiera poner un precio total del producto con impuestos incluidos de **195,90€**, es decir de **183,04€** sin IGIC (garantizando un margen del **35%** respecto al coste de los materiales sin IGIC), requiere de la venta de un mínimo de **58 unidades**. Por tanto, el umbral de rentabilidad con el coste fijo de elaboración del prototipo y los costes variables de fabricación de cada unidad se fija en 58 unidades, a partir de las cuales se empezaría a conseguir beneficio.

6. Resultados

A la finalización de este trabajo se ha logrado obtener un prototipo de brazo robótico totalmente funcional, basado en tecnología OpenSource que puede ser fabricado totalmente mediante técnicas modernas de fabricación digital como la impresión 3D o el corte láser. Este dispositivo es capaz de operar en diversos modos de operación, permitiendo al estudiantado trabajar con los brazos robóticos para el control directo de los mismos (en el modo manual) o indicar las posiciones que se desea que alcance el dispositivo y hacer uso de la cinemática inversa. Por tanto, este prototipo sirve como herramienta esencial para la formación en robótica manipulativa, permitiendo trabajar conceptos como la cinemática directa, inversa e incluso yendo más allá de la robótica y sirviendo como herramienta para explicar e integrar en el aula la ingeniería, la electrónica y la programación.

En el presente trabajo se han desarrollado planos detallados del prototipo, los cuales se presentan en el **Anexo I**. Además, se ha elaborado un informe exhaustivo de estudio mecánico, incluido en el **Anexo II**, que analiza el comportamiento mecánico de las piezas fabricadas mediante impresión 3D. En este estudio, se evaluó el rendimiento de las piezas bajo condiciones específicas de fabricación, incluyendo un relleno del 0% y paredes de 0,6 mm, junto con una carga de 60 g en el extremo del brazo.

Los resultados revelaron que la pieza más sometida a tensión es el hombro, con una resistencia máxima calculada de 4,523 MPa. En comparación, el material utilizado puede soportar una tensión máxima de 58,5 MPa. Esto indica que, en el peor de los casos, la pieza opera a tan solo el 12,94% de su capacidad máxima, lo que demuestra un margen significativo de seguridad y fiabilidad en el diseño.

El proyecto culmina con el diseño de 2 PCBs diferentes, una que sirve como controlador principal del dispositivo e incorpora el microcontrolador y las diversas interfaces de Entrada y Salida que sirven de control del brazo. La otra PCB permite la inclusión de diversos sensores y actuadores sustituibles en el efector final, por lo que se presenta un diseño totalmente modular y expandible en el futuro. Estos diseños se muestran en detalle en el **Anexo III**, donde se incluyen los planos electrónicos de las distintas versiones, acompañados de un renderizado de la PCB. Cada iteración realizada incorpora componentes electrónicos adicionales para ampliar las funcionalidades. Esto culmina en la última versión presentada.

La comprensión detallada del código empleado en las diversas pruebas resulta fundamental para el análisis exhaustivo de los resultados obtenidos. En el **Anexo IV** se encuentra el código correspondiente a cada una de las pruebas mencionadas en la sección 4.7.1. Este análisis permitió identificar posibles errores, optimizaciones y

mejoras en el proceso de desarrollo. Además, proporcionó una visión clara de cómo se implementan las diferentes funcionalidades y cómo interactuaron con el sistema en su conjunto, lo que fue crucial para la interpretación precisa de los resultados.

Se ha realizado un análisis exhaustivo de los costes asociados con la fabricación del brazo robótico educativo. Se detalla un presupuesto completo que abarca las partes mecánicas, los componentes electrónicos y la mano de obra involucrada en el proceso. Se calcula el coste de fabricación de las piezas mecánicas impresas en 3D y cortadas con láser, incluyendo el material utilizado y el tiempo de trabajo necesario. Asimismo, se presenta un desglose del coste de los componentes electrónicos, considerando tanto la PCB principal como la auxiliar, así como los servomotores requeridos. Además, se estima el coste en mano de obra, basado en la distribución de tiempo por paquetes de trabajo y el coste por hora de los ingenieros involucrados en el proyecto.

Llevando a cabo estos cálculos, se determina que el coste total del proyecto asciende a 10.461,40 €, lo que incluye el coste de las piezas fabricadas, los componentes electrónicos y la mano de obra. Con el objetivo de establecer un precio de venta viable para el prototipo, se fija en 183,04 €, garantizando un margen del 35% respecto al coste de los materiales sin IGIC. Esto sugiere que para alcanzar un umbral de rentabilidad, se necesitaría vender un mínimo de 58 unidades del producto.

Finalmente, cabe destacar que este documento no solo presenta la metodología y los resultados obtenidos a lo largo del proyecto, sino que también representa una contribución significativa al avance de la tecnología de brazos robóticos educativos. Al proporcionar una descripción detallada de los procesos de desarrollo y los resultados obtenidos, este trabajo no solo amplía el conocimiento en el campo de la robótica educativa, sino que también ofrece una base sólida para futuras investigaciones y desarrollos en esta área. La documentación exhaustiva de la metodología empleada y los resultados alcanzados se espera que beneficie a la comunidad científica y educativa, facilitando así el progreso continuo en el diseño y la implementación de brazos robóticos educativos.

7. Conclusiones

En este proyecto se logró desarrollar en 8 meses un prototipo de brazo robótico totalmente funcional y capaz de ser producido mediante técnicas de fabricación digital (como impresión 3D FDM o corte láser) y con capacidad para operar en diversos modos de funcionamiento (denominados en esta memoria modo *manual* y modo *automático*), sirviendo como herramienta para trabajar conceptos de robótica como la cinemática inversa, o conceptos relacionados con la electrónica y la programación. El dispositivo realizado es totalmente compatible con el *ecosistema Arduino*, por lo que se trata de un dispositivo altamente intuitivo y sencillo de programar. Asimismo, se ha garantizado la correcta resistencia del dispositivo mediante diversos análisis mecánicos de forma que el dispositivo sea totalmente seguro para su uso en el aula.

Este TFM representa un gran avance hacia la obtención de un producto finalizado que pueda posteriormente entrar en el mercado y servir como un recurso adecuado en el aula, principalmente en los niveles finales de Secundaria, Bachillerato y ciertas ramas de la Formación Profesional. A partir de aquí, se exploraron diferentes implementaciones que podrían añadirse para mejorar este prototipo y así alcanzar el objetivo final de llegar a un producto totalmente comercializable. Entre estas propuestas se incluyen:

- Implementación de diferentes tipos de movimientos para ir de un punto a otro.
- Implementación de protocolos de comunicación por Bluetooth empleando el zócalo para módulo Bluetooth disponible en la PCB.
- Implementación de protocolos de comunicación WiFi empleando el zócalo para módulo WiFi disponible en la PCB.
- Adición de diferentes actuadores y sensores en el extremo del brazo haciendo uso de la PCB Auxiliar diseñada.
- Implementación de la comunicación entre diferentes unidades de brazos robóticos permitiendo realizar tareas coordinadas e incrementando la complejidad de la programación del sistema.

Se destaca también la dificultad inherente a este tipo de proyectos, que requiere conocimientos mecánicos, electrónicos y de programación para su realización, lo que subraya la importancia de la multidisciplinariedad en la investigación y desarrollo tecnológico. Con estas propuestas de mejoras, se considera que este documento puede servir como referencia para la realización de otros proyectos de ámbito STEAM, especialmente aquellos relacionados con la fabricación de brazos robóticos, que pueden tomar diferentes formas como las mencionadas en el apartado 2.1.

8. Referencias

- [1] "Dobot Educativo - brazo robótico educativo con transportadora - Dobot Edu Magician Belt," Amazon. [En línea]. Disponible:
<https://www.amazon.com.mx/Dobot-Educativo-educativa-transportadora-Edu-Magician-Belt/dp/B07PPPFDST>.
- [2] "Brazo robótico," AliExpress. [En línea]. Disponible:
<https://es.aliexpress.com/item/1005005507359716.html>.
- [3] "WLKATA 6-Axis Mini Robotic Arm Mirobot Education Kit," RobotShop. [En línea]. Disponible:
<https://eu.robotshop.com/es/products/wlkata-6-axis-mini-robotic-arm-mirobot-education-kit>.
- [4] "Página principal," WLKATA. [En línea]. Disponible: <https://www.wlkata.com/es>.
- [5] "Historia de la robótica," Timetoast. [En línea]. Disponible:
<https://www.timetoast.com/timelines/historia-de-la-robotica-fbe7dbbf-39ac-4175-bd6d-aaa5724ded34>.
- [6] "Historia de la robótica," Timetoast. [En línea]. Disponible:
<https://www.timetoast.com/timelines/historia-de-la-robotica-8403140d-9d33-493d-ad02-43658eeb0465>.
- [7] "Brazos robóticos: diferentes tipos y cuándo utilizarlos," Descubre Arduino. [En línea]. Disponible:
<https://descubrearduino.com/brazos-roboticos-diferentes-tipos-y-cuando-utilizarlos/>.
- [8] "Industrial robots," IQS Directory. [En línea]. Disponible:
<https://www.iqsdirectory.com/articles/automation-equipment/industrial-robots.html>.
- [9] "La evolución industrial: los robots (parte 2)," Daneel Mechatronics. [En línea]. Disponible:
<https://daneel-mechatronics.com/blog/la-evolucion-industrial-los-robots-parte-2/>.
- [10] "Articulated precision robotic arm for entertainment riding with safety chair," Automatic Bag Packing Machine. [En línea]. Disponible:
<https://spanish.automaticbagpackingmachine.com/sale-9556838-articulated-precision-robotic-arm-for-entertainment-riding-with-safety-chair.html>.
- [11] "Brazo robótico con 6 ejes," Alamy. [En línea]. Disponible:
<https://www.alamy.es/brazo-robotico-con-6-ejes-robot-industrial-renderizacion-3d-aislada-sobre-fondo-blanco-image368047527.html>.
- [12] "Robot colaborativo," Industrial Robotics. [En línea]. Disponible:
<https://industrial-robotics.es/product-category/robot-colaborativo/>.

[13] "RT3401SS," Tecnoedu. [En línea]. Disponible:
<https://tecnoedu.com/Automatizacion/RT3401SS.php>.

[14] "Robots cartesianos," AMS Latam. [En línea]. Disponible:
<https://www.ams-latam.com/es/robots-cartesianos/>.

[15] "Cylindrical robots," Industrial Electronics. [En línea]. Disponible:
https://www.industrial-electronics.com/industrial-electricity-com/4_Cylindrical_Robots.html.

[16] "Describe spherical robot," JF505 Industrial Robotics Blog. [En línea]. Disponible:
<https://jf505industrialrobotics.blogspot.com/2013/12/223-describe-spherical-robot.html>.

[17] "Producto," DirectIndustry. [En línea]. Disponible:
<https://www.directindustry.com/prod/abb-robotics/product-30265-169123.html>.

[18] "¿Cómo es el robot antropomórfico ruso Fedor que va a hacer en el espacio?," Canal 13 San Juan. [En línea]. Disponible:
<https://www.canal13sanjuan.com/mundo/2019/8/24/1057omo-es-el-robot-antropomorfo-ruso-fedor-que-va-hacer-en-el-espacio-121820.html>.

[19] "Actualidad," El País. [En línea]. Disponible:
https://elpais.com/tecnologia/2019/08/22/actualidad/1566469024_955770.html.

[20] "Evolución de la robótica industrial," EDS Robotics. [En línea]. Disponible:
<https://www.edsrobotics.com/blog/evolucion-robotica-industrial/>.

[21] "Robótica educativa," UNIR. [En línea]. Disponible:
<https://www.unir.net/educacion/revista/robotica-educativa/>.

[22] "A robot-oriented approach for teaching computational thinking to kindergarteners," Springer. [En línea]. Disponible:
https://link.springer.com/chapter/10.1007/978-3-030-77040-2_3.

[23] "Abrazadera manipulador industrial," Amazon. [En línea]. Disponible:
<https://www.amazon.es/Taidda-Abrazadera-Manipulador-Industrial-universitaria/dp/B084FSVQYC>.

[24] "Brazo robótico Tinkerkit Braccio," Farnell. [En línea]. Disponible:
<https://es.farnell.com/arduino/t050000/brazo-rob-tico-tinkerkit-braccio/dp/2831002>.

[25] "Arduino Braccio," Arduino. [En línea]. Disponible:
<https://www.arduino.cc/education/braccio>.

[26] "RobotGeek Core Kit Robot Arm," ZeroDNA. [En línea]. Disponible:
<https://www.zerodna.de/RobotGeek-Core-Kit-Robot-Arm-Snapper-Gripper-Arm-controllable-by-Geekduino-Arduino-50g-payload-285cm-reach>.

[27] "Dobot Magician Lite," Amazon. [En línea]. Disponible:
<https://www.amazon.es/Dobot-Magician-Lite-inteligente-multifuncional/dp/B08HDKRWLM>.

[28] I. J. Henao, J. A. Giraldo, F. A. Meza, C. W. Sánchez y J. E. Ordoñez, "El brazo robótico como herramienta pedagógica en el aula de clase," Lumen Gentium, vol. 1, núm. 1, pp. 82-90, 2017. [En línea]. Disponible:
https://www.researchgate.net/publication/317413734_El_brazo_robotico_como_herramienta_pedagogica_en_el_aula_de_clase.

[29] "Teaching robotic concepts to K-12 students through hands-on projects," IEEE. [En línea]. Disponible: <https://ieeexplore.ieee.org/document/6581462>.

[30] "Robot Arm Automation," Arduino Project Hub. [En línea]. Disponible:
<https://projecthub.arduino.cc/danielgass/robot-arm-automation-c4e0cb>.

[31] "DIY Arduino Robot Arm With Smartphone Control," How To Mechatronics. [En línea]. Disponible:
<https://howtomechatronics.com/tutorials/arduino/diy-arduino-robot-arm-with-smartphone-control>.

[32] "Arduino Projects - Robotic Arm," Robotics and Energy. [En línea]. Disponible:
<https://roboticsandenergy.com/projects/arduino-projects/robotic-arm>.

[33] "Controlling Arduino Robot Arm With Arm Link Software," Instructables. [En línea]. Disponible:
<https://www.instructables.com/Controlling-Arduino-Robot-Arm-With-Arm-Link-Software>.

[34] "Guía de compra de brazos robóticos," MyBotRobot. [En línea]. Disponible:
<https://www.mybotrobot.com/guia-compra-brazos-roboticos>.

[35] "Fusion 360 for Personal Use," Autodesk. [En línea]. Disponible:
<https://www.autodesk.com/products/fusion-360/personal>.

[36] "PrusaSlicer," Prusa3D. [En línea]. Disponible:
https://www.prusa3d.com/en/page/prusaslicer_424.

[37] "KiCad," KiCad. [En línea]. Disponible: <https://www.kicad.org>.

[38] "FlatCAM," FlatCAM. [En línea]. Disponible: <http://flatcam.org>.

[39] "RDWorks," Software Informer. [En línea]. Disponible:
<https://rdworks.software.informer.com/8.0>.

[40] "Arduino Software," Arduino. [En línea]. Disponible: <https://www.arduino.cc/en/software>.

[41] "GitHub," GitHub. [En línea]. Disponible: <https://github.com>.

[42] "Original Prusa i3 MK3S+," Prusa3D. [En línea]. Disponible:
<https://www.prusa3d.com/es/producto/impresora-3d-original-prusa-i3-mk3s-3/#specs>.

[43] "MG90S Micro Servo," Baku Tool. [En línea]. Disponible:
<http://www.bakutool.com/product--id-29.html>.

[44] "Micro Servo MG90S 180 Grados," Canarias Digital Codesign. [En línea]. Disponible:
<https://canarias.digitalcodesign.com/shop/00010122-micro-servo-mg90s-180-grados-3357>.

[45] "Servomotor MG996R 180° Metal," Canarias Digital Codesign. [En línea]. Disponible:
<https://canarias.digitalcodesign.com/shop/00018333-servomotor-mg996r-180o-metal-4178>.

[46] "Bipolar 4-lead stepper motor," Amazon. [En línea]. Disponible:
<https://www.amazon.com/bipolar-4-plomo-conector-impresora-bricolaje/dp/B0CDRMHM82>.

[47] "MG90S Datasheet," AllDataSheet. [En línea]. Disponible:
<https://pdf1.alldatasheet.es/datasheet-pdf/view/1132104/ETC2/MG90S.html>.

[48] "MG996R Datasheet," AllDataSheet. [En línea]. Disponible:
<https://pdf1.alldatasheet.com/datasheet-pdf/view/1131873/ETC2/MG996R.html>.

[49] "PCA9685PW Datasheet," AllDataSheet. [En línea]. Disponible:
<https://html.alldatasheet.com/html-pdf/424209/NXP/PCA9685PW/180/3/PCA9685PW.html>.

[50] "DRV8825 Datasheet," Maker Guides. [En línea]. Disponible:
<https://www.makerguides.com/wp-content/uploads/2019/02/DRV8825-Datasheet.pdf>.

[51] "LM2596 Datasheet," Texas Instruments. [En línea]. Disponible:
<https://www.ti.com/lit/ds/symlink/lm2596.pdf>.

[52] "DRV8825 Tutorial," Maker Guides. [En línea]. Disponible:
<https://www.makerguides.com/es/drv8825-stepper-motor-driver-arduino-tutorial>.

[53] "CD74HC4051 Datasheet," Texas Instruments. [En línea]. Disponible:
<https://www.ti.com/lit/ds/symlink/cd74hc4051.pdf>.

[54] "Micro Servo MG90S 180 Grados," Canarias Digital Codesign. [En línea]. Disponible:
<https://canarias.digitalcodesign.com/shop>.

9. Glosario de términos

1. **DIY**: Siglas de 'Hazlo Tú Mismo' del inglés 'Do It Yourself'. Enfoque sobre la creación de producto centrado en que el usuario de forma autónoma o autodidacta pueda construirse sus propios elementos, herramientas o productos.
2. **MG996R**: Tipo de servomotor alimentado a 5V y con un par de 9,4 kgf-cm muy empleado en la robótica educativa.
3. **STM32**: Familia de microcontroladores de 32 bits basados en el núcleo ARM Cortex-M. Presenta una capacidad de cálculo muy superior a mis microcontroladores ATmega 328P y ATmega 2560. Está fabricado por STMicroelectronics.
4. **SPI**: Sus siglas son 'Serial Peripheral Interface'. Se trata de un estándar de bus de comunicaciones usado para transferencia de información entre circuitos integrados empleando un número reducido de conductores. Está formado por 4 líneas: una línea de reloj (SCK), otra de envío de datos (MOSI), otra de recepción de datos (MISO) y una de selección del dispositivo (SS).
5. **I2C**: Sus siglas provienen de 'Inter-Integrated Circuit'. Es un protocolo de bus de comunicación serial que emplea dos líneas para la comunicación: SDA (línea de datos) y SCL (línea de datos). Permite la conexión de hasta 128 dispositivos.
6. **UART**: Sus siglas provienen de 'Universal Asynchronous Receiver / Transmitter'. Es el dispositivo que controla los puertos y las comunicaciones seriales, permitiendo comunicaciones full-duplex con únicamente dos líneas de datos (Rx y Tx).
7. **PWM**: Sus siglas provienen de 'Pulse Width Modulation'. Se trata de una modulación de anchura de pulso, donde una señal de frecuencia constante varía su ciclo de trabajo de acuerdo a la amplitud de una señal. Es muy empleada en el control de la velocidad de motores de corriente continua.
8. **BUCK**: Tipo de convertor de corriente continua que permite obtener una tensión inferior a la salida respecto de su entrada. Se caracterizan por poseer una alta eficiencia en la conversión de voltaje.
9. **STEP-UP**: Tipo de convertor de corriente continua que permite obtener una tensión superior a la salida respecto de su entrada.
10. **FDM**: Tipo de impresora 3D. Su modo de operación consiste en la deposición de material fundido que va aplicándose capa a capa para generar la pieza solicitada.
11. **SLA**: Tipo de impresión 3D. La estereolitografía consiste en la exposición de un material líquido (generalmente resina) a imágenes con luz ultravioleta que solidifican las partes expuestas para construir una pieza capa a capa.
12. **MG90S**: Tipo de servomotor alimentado a 5V y con un par de 1,8 kgf-cm muy empleado en la robótica educativa.
13. **PLA**: Tipo de filamento para impresión 3D muy empleado por su bajo precio y toxicidad. Se trata de un Ácido Poliláctico que es un polímero que se obtiene a partir de la fermentación de material vegetal como el maíz, la yuca o la caña de azúcar.
14. **AVR**: Familia de microcontroladores del fabricante Atmel, que se caracteriza porque todas las CPUs de esta familia están formadas por 32 registros de 8 bits. En ella se incluyen desde los pequeños microcontrolados de 0,5 kB de memoria flash (ATtiny) hasta los ATmega de 256 kB de memoria flash (ATmega 2560).
15. **EEPROM**: 'Electrically Erasable Programmable Read-Only Memory'. Es un tipo de memoria no volátil utilizada en sistemas electrónicos para almacenar datos de

manera que puedan ser leídos, escritos y borrados eléctricamente. A diferencia de las memorias volátiles como la RAM, la EEPROM conserva su contenido incluso cuando se apaga la alimentación.

16. **GPIO**: 'General Purpose Input Output'. Conjunto de pines digitales que pueden ser configurados para diversas aplicaciones, pudiendo ser entradas o salidas según la funcionalidad deseada.
17. **Schottky**: Tipo de diodo con una muy baja tensión umbral y una velocidad de conmutación alta. Está especialmente recomendado para circuitos rectificadores.
18. **PTC**: 'Positive Temperature Coefficient'. Resistencia que varía según la temperatura de forma proporcional. Comúnmente es empleado como fusible rearmable, ya que al calentarse por un alto paso de corriente eleva tanto su resistencia que se comporta como un circuito abierto.
19. **Drop-out**: Expresión utilizada para referirse a la diferencia de tensión entre la entrada y la salida de un regulador de alimentación.
20. **Programador ICSP**: 'In Chip Serial Programmer'. Método de programación de microcontroladores que permite cargar de forma directa el firmware compilado desde el PC en el microcontrolador.
21. **Pitch**: Expresión que hace referencia a la distancia existente entre dos pistas o pads en una Placa de Circuito Impreso (PCB).
22. **SMD**: 'Surface-Mount Device'. Es una tecnología para fabricar circuitos electrónicos en la que los componentes se montan directamente sobre la superficie de placas de circuito impreso (PCB), en lugar de insertar los componentes en agujeros pasantes.
23. **Fablab**: 'Fabrication Laboratory', es un espacio de fabricación digital abierto al público, equipado con herramientas avanzadas como impresoras 3D, cortadoras láser y fresadoras CNC. Estos laboratorios permiten a individuos y pequeñas empresas diseñar y crear prototipos rápidamente. Los FabLabs también ofrecen formación y fomentan la colaboración y la innovación comunitaria, haciendo accesible la tecnología de fabricación a todos.
24. **THD**: 'Through-Hole Devices'. Son tipos de componentes electrónicos diseñados para ser montados en placas de circuito impreso mediante la inserción de sus pines o cables a través de agujeros perforados en la placa y posteriormente soldados en el lado opuesto.

 **Escuela de Doctorado
y Estudios de Posgrado**
Universidad de La Laguna

Trabajo de Fin de Máster

ANEXOS

**DISEÑO Y FABRICACIÓN DE UN BRAZO ROBÓTICO
CON FINES EDUCATIVOS**

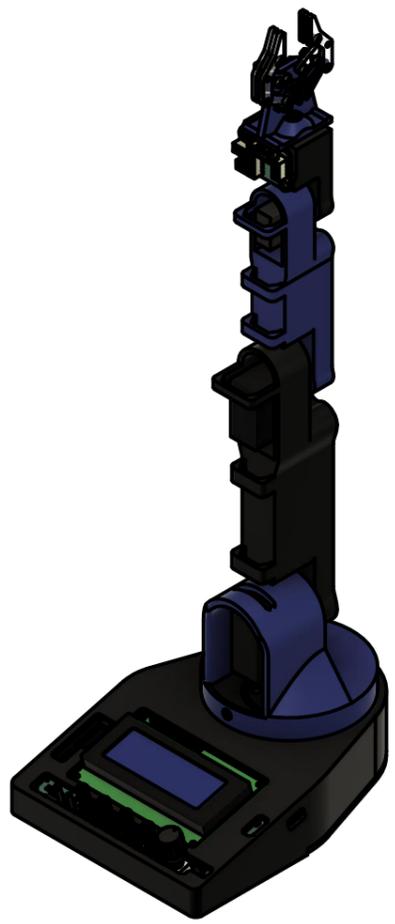
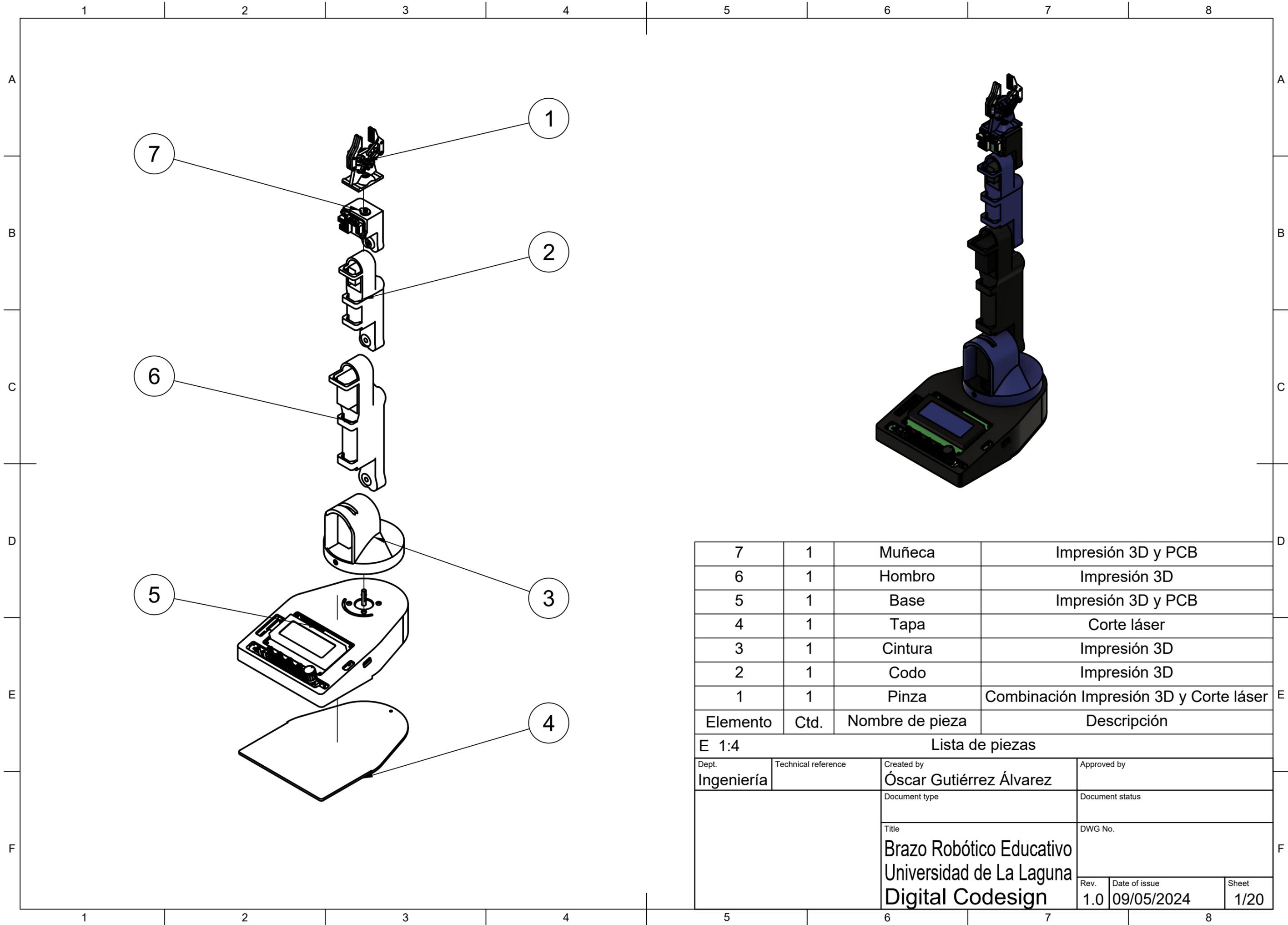
Titulación: Máster en Ingeniería Industrial

Estudiante:
Óscar Gutiérrez Álvarez

Tutor: Silvia Alayón Miranda
Co-Tutor: Miqueas Zurisadai Fortes González

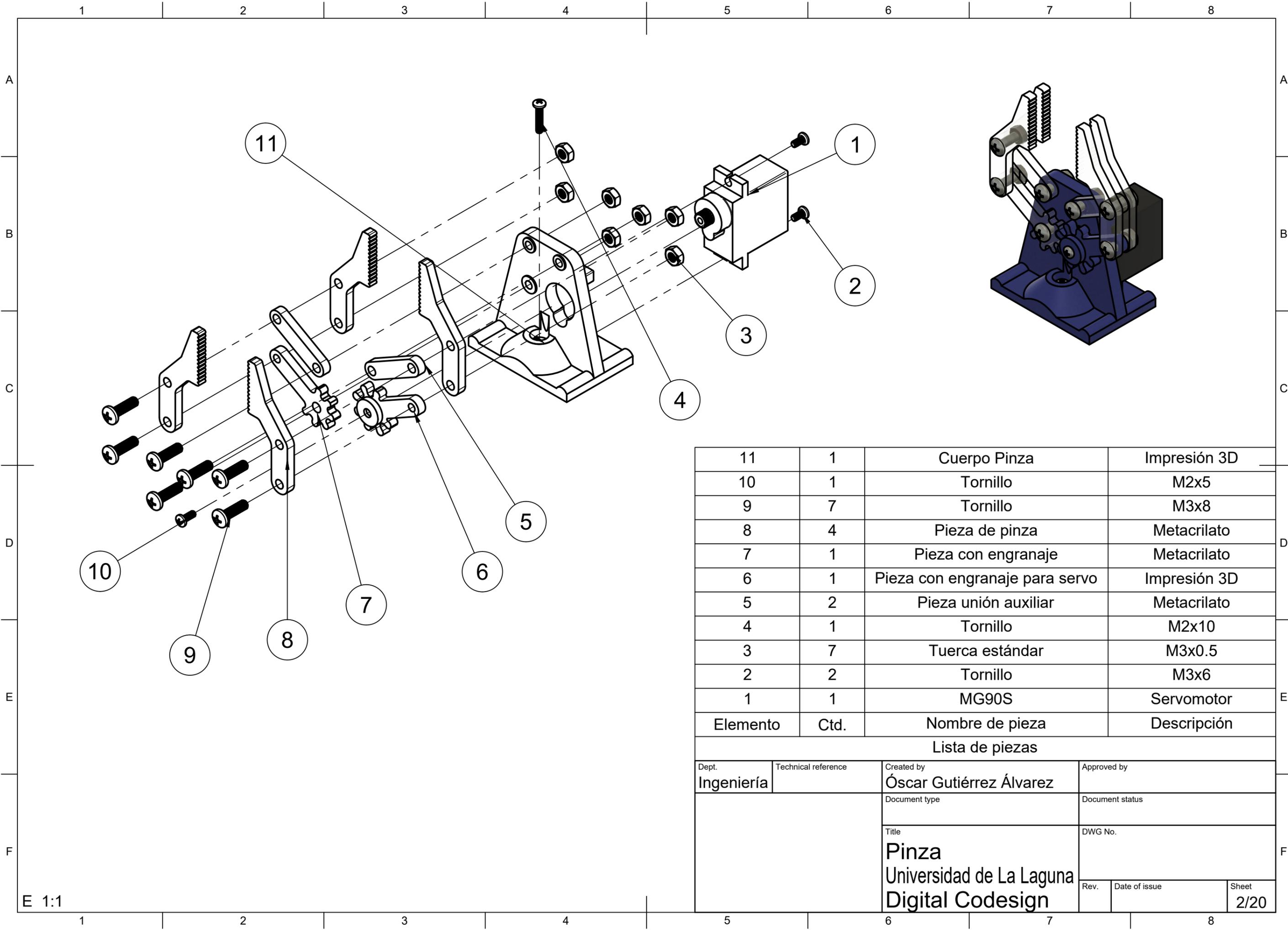
Relación de Anexos

- I. ANEXO I - PLANOS MECÁNICOS DEL PROTOTIPO
- II. ANEXO II - INFORME DE ESTUDIOS MECÁNICOS
- III. ANEXO III - PLANOS ELECTRÓNICOS DEL PROTOTIPO
- IV. ANEXO IV - SOFTWARE DEL DISPOSITIVO



7	1	Muñeca	Impresión 3D y PCB
6	1	Hombro	Impresión 3D
5	1	Base	Impresión 3D y PCB
4	1	Tapa	Corte láser
3	1	Cintura	Impresión 3D
2	1	Codo	Impresión 3D
1	1	Pinza	Combinación Impresión 3D y Corte láser
Elemento	Ctd.	Nombre de pieza	Descripción

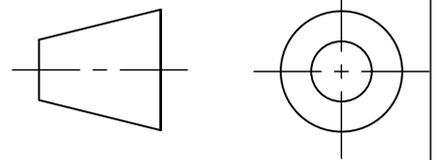
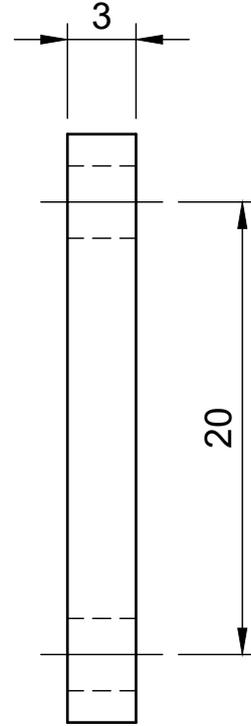
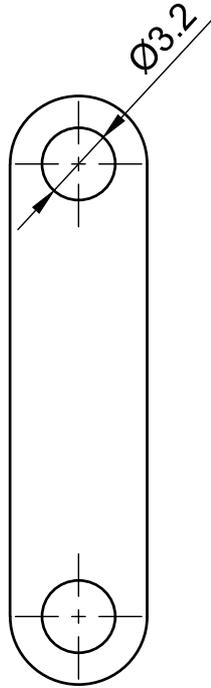
E 1:4 Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Brazo Robótico Educativo Universidad de La Laguna Digital Codesign	DWG No.
	Rev. 1.0	Date of issue 09/05/2024	Sheet 1/20



11	1	Cuerpo Pinza	Impresión 3D
10	1	Tornillo	M2x5
9	7	Tornillo	M3x8
8	4	Pieza de pinza	Metacrilato
7	1	Pieza con engranaje	Metacrilato
6	1	Pieza con engranaje para servo	Impresión 3D
5	2	Pieza unión auxiliar	Metacrilato
4	1	Tornillo	M2x10
3	7	Tuerca estándar	M3x0.5
2	2	Tornillo	M3x6
1	1	MG90S	Servomotor
Elemento	Ctd.	Nombre de pieza	Descripción

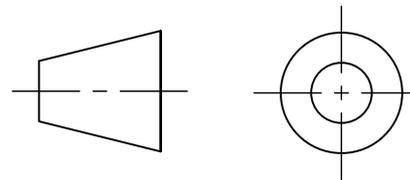
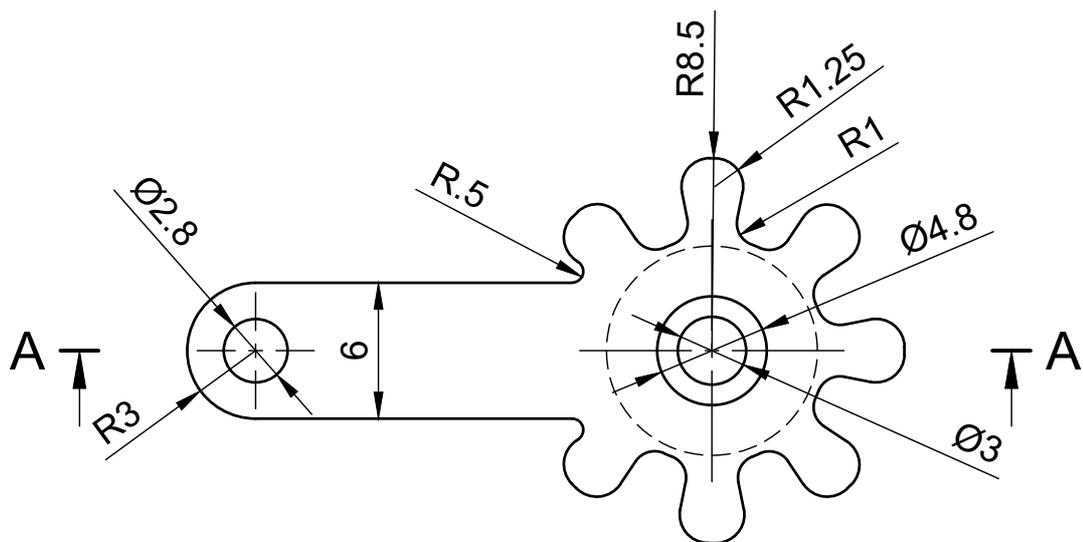
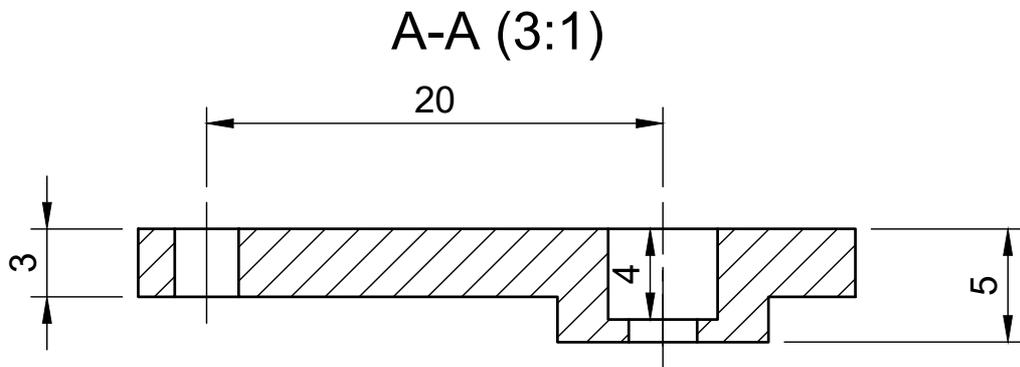
Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Pinza Universidad de La Laguna Digital Codesign	DWG No.
Rev.	Date of issue	Sheet 2/20	

E 1:1



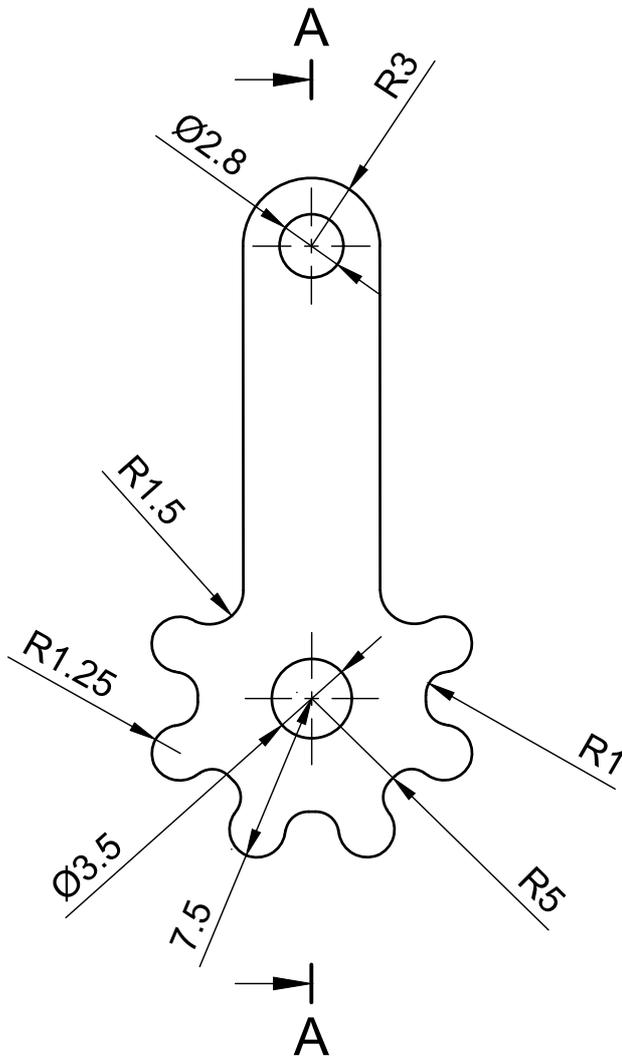
E 3:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title Pieza unión auxiliar Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 09/05/2024	Sheet 3/20	

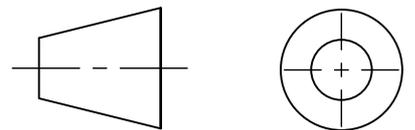
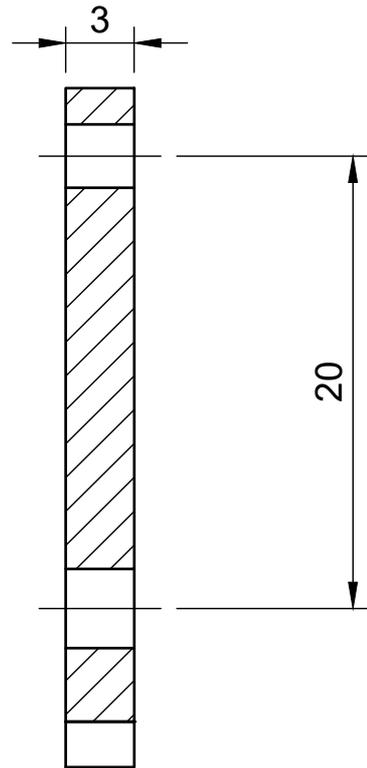


E 3:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title Pieza con engranaje para servo Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 09/05/2024	Sheet 4/20	

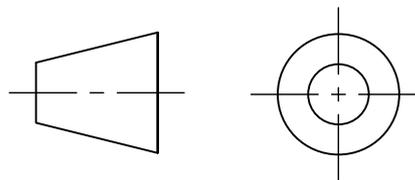
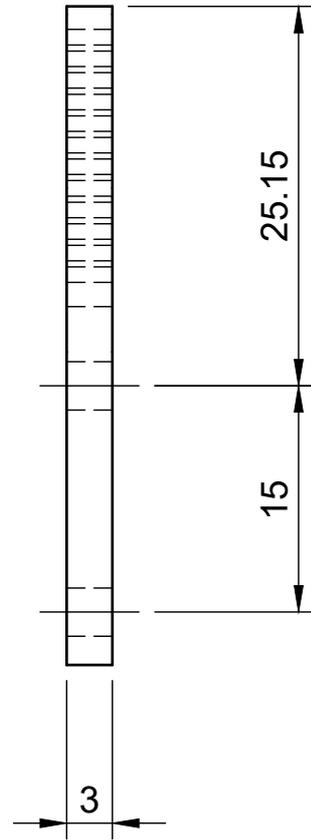
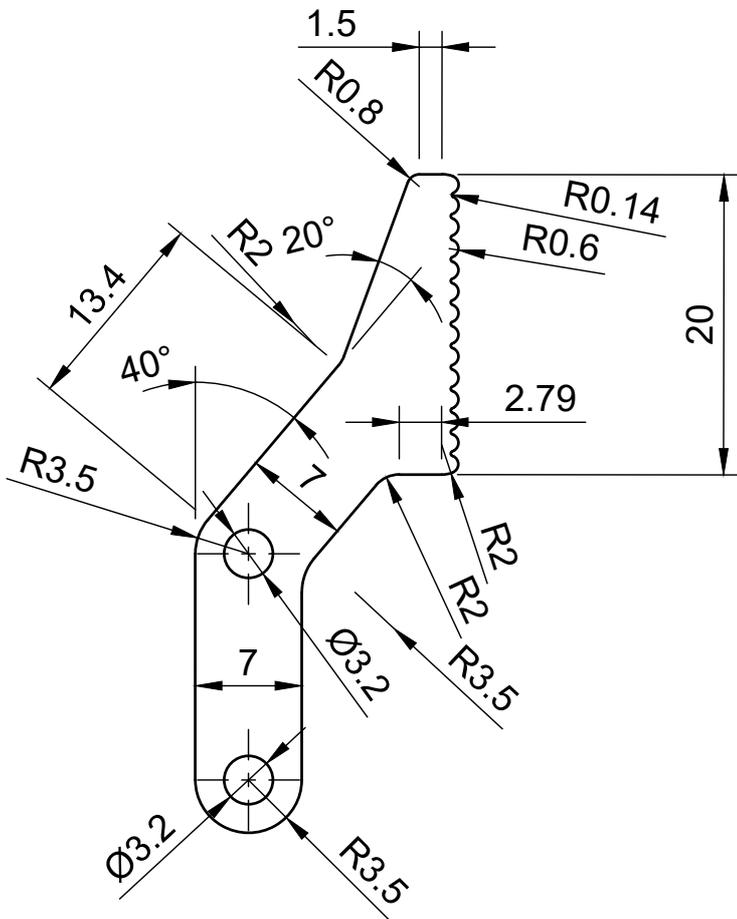


A-A (3:1)



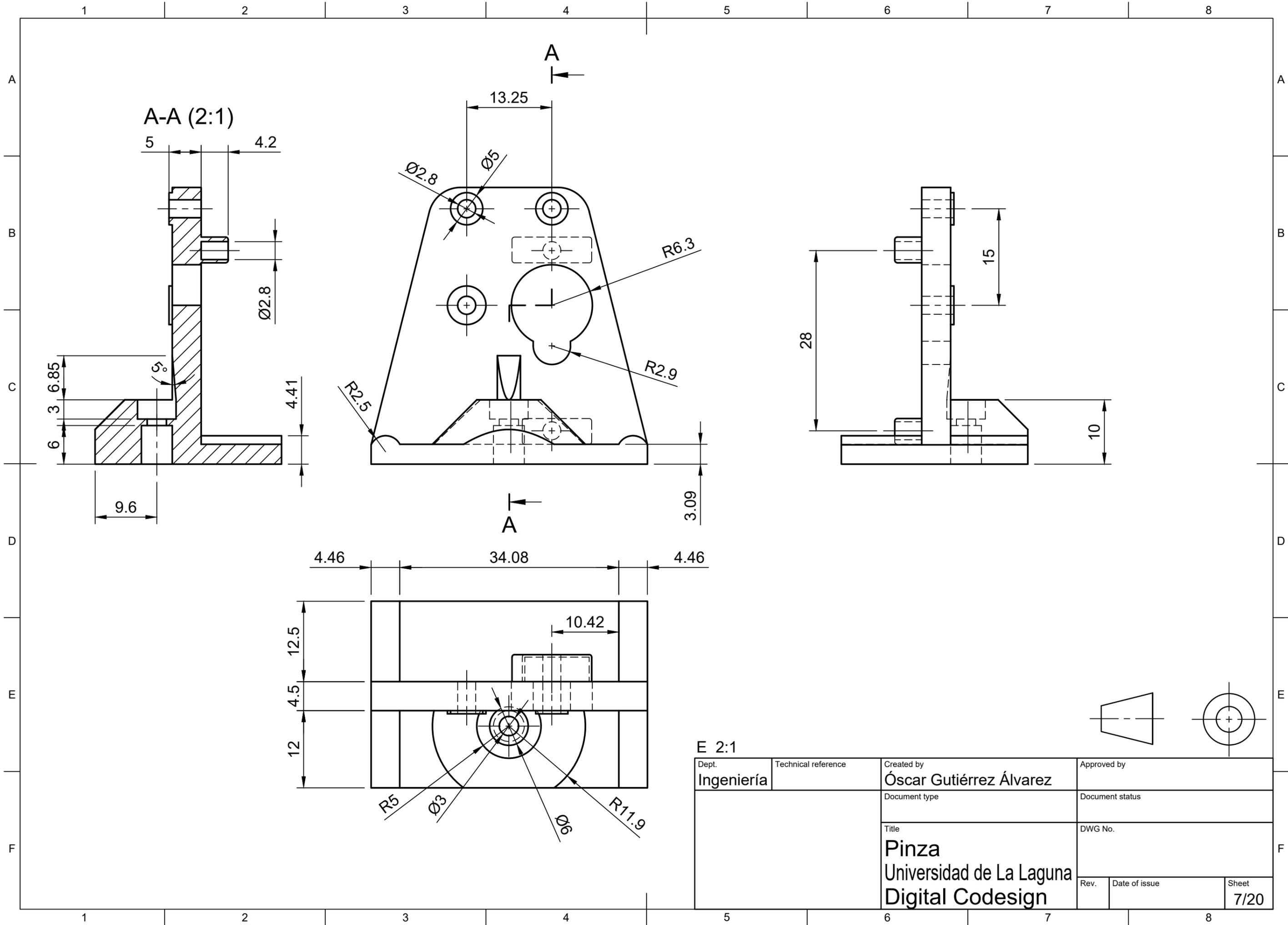
E 3:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by	
		Document type	Document status	
		Title Pieza con engranaje Universidad de La Laguna Digital Codesign	DWG No.	
		Rev. 1.0	Date of issue 09/05/2024	Sheet 5/20



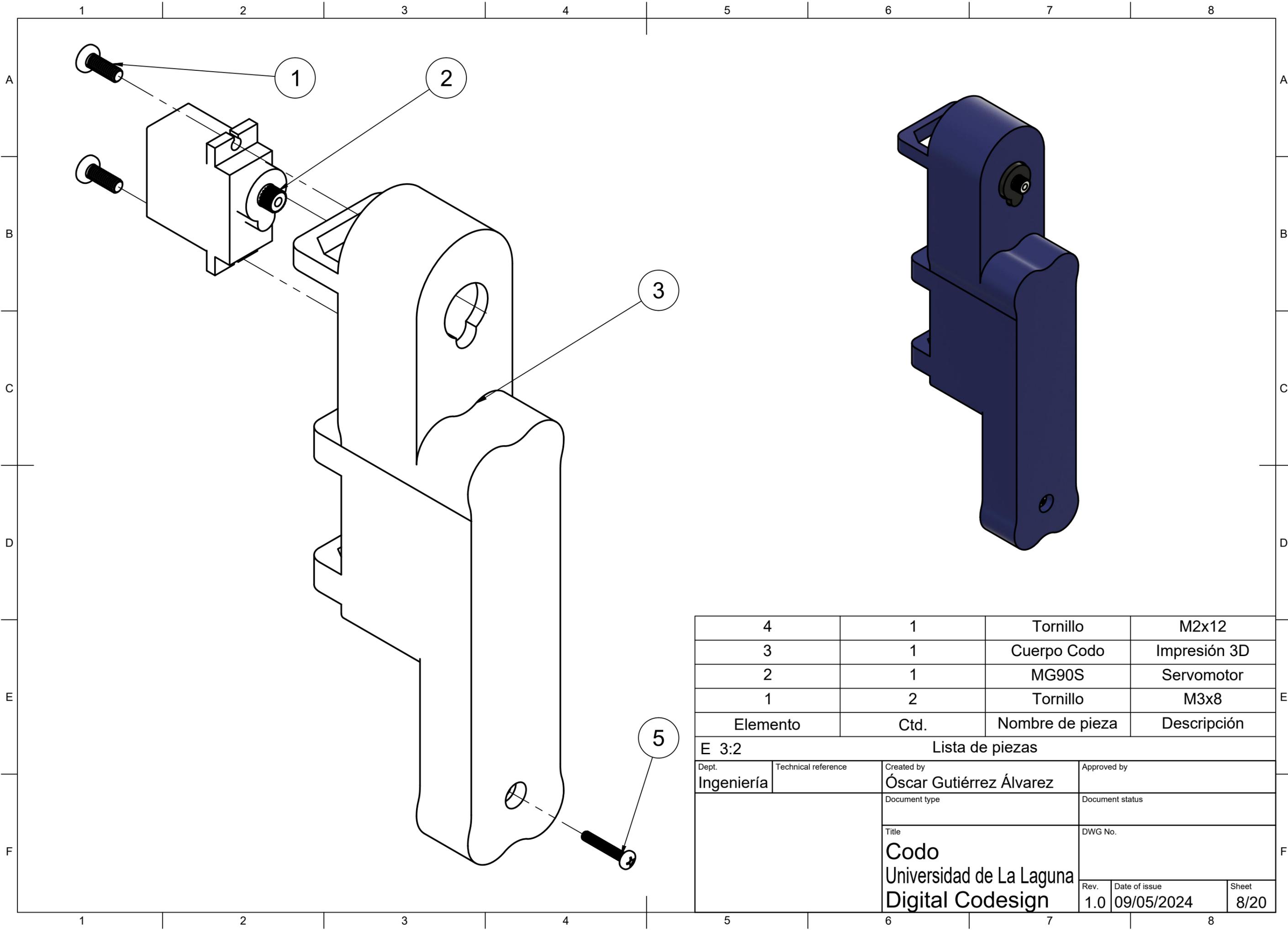
E 2:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title Pieza de Pinza Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 09/05/2024	Sheet 6/20	

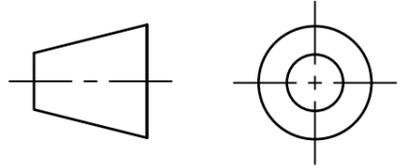
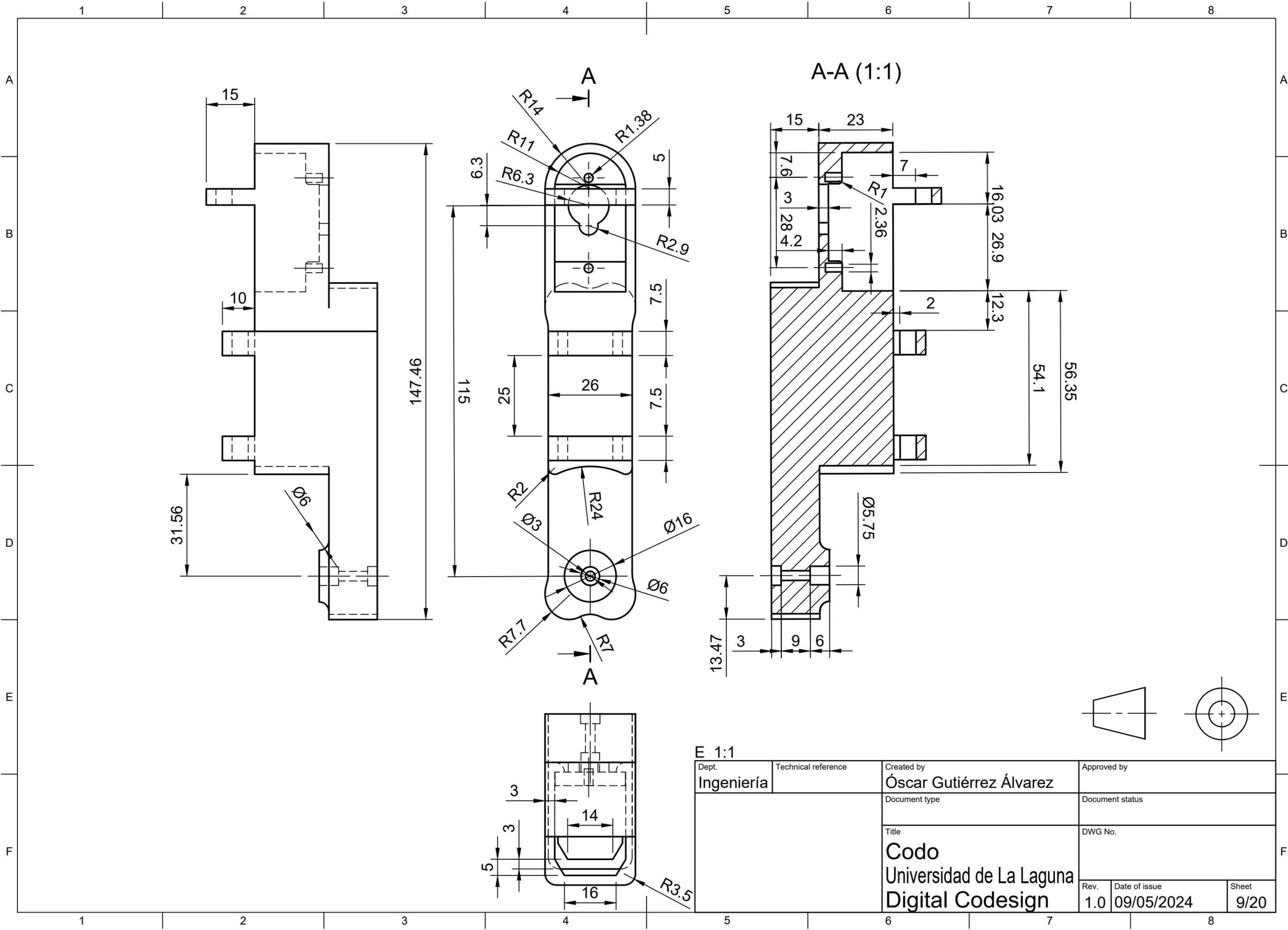


E 2:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Pinza Universidad de La Laguna Digital Codesign	DWG No.
Rev.	Date of issue	Sheet	7/20

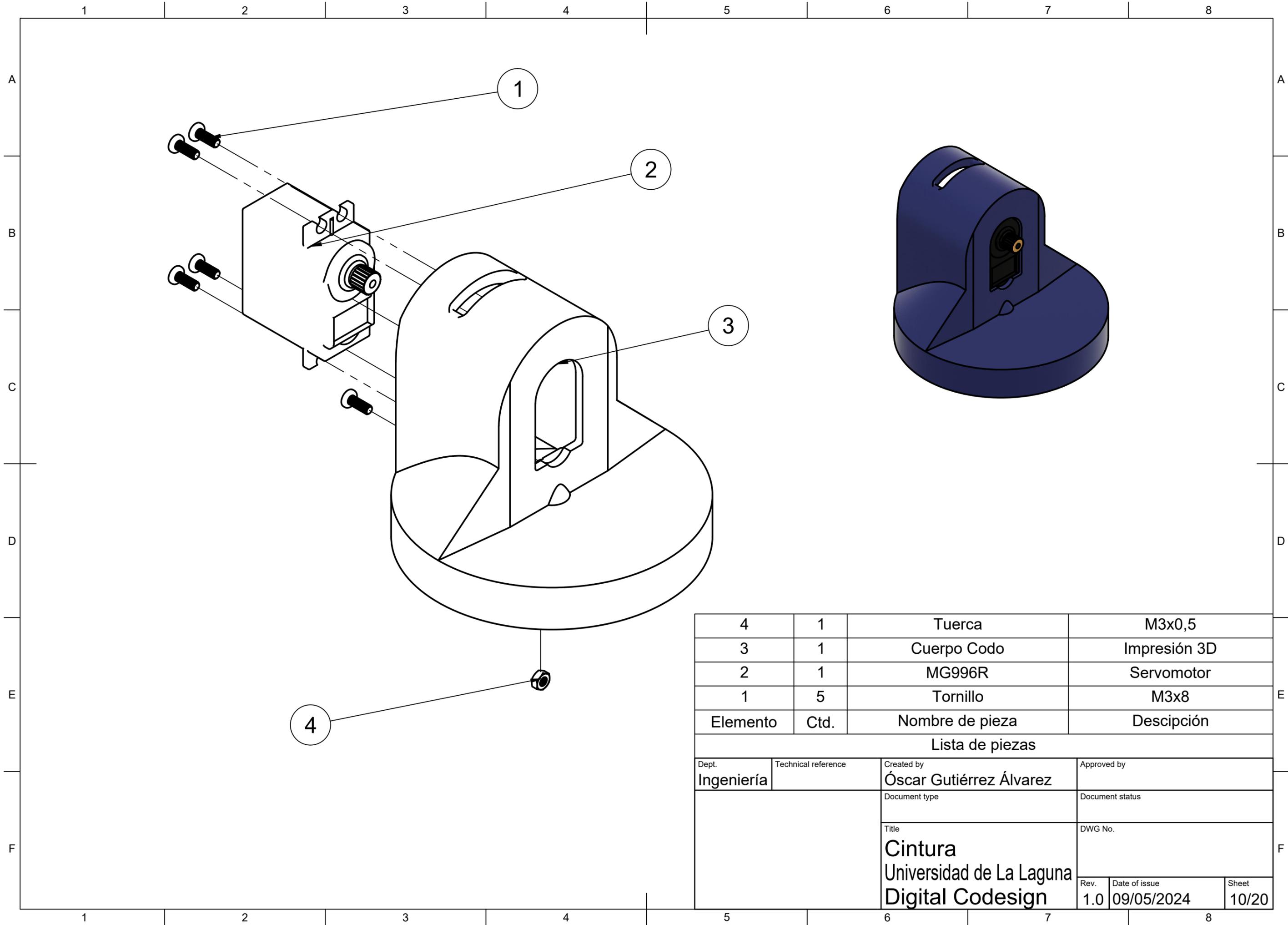


4	1	Tornillo	M2x12
3	1	Cuerpo Codo	Impresión 3D
2	1	MG90S	Servomotor
1	2	Tornillo	M3x8
Elemento	Ctd.	Nombre de pieza	Descripción
E 3:2			
Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Codo Universidad de La Laguna Digital Codesign	DWG No.
Rev. 1.0	Date of issue 09/05/2024	Sheet 8/20	

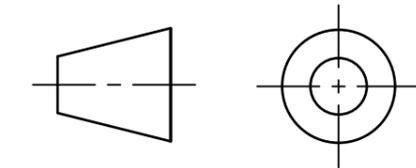
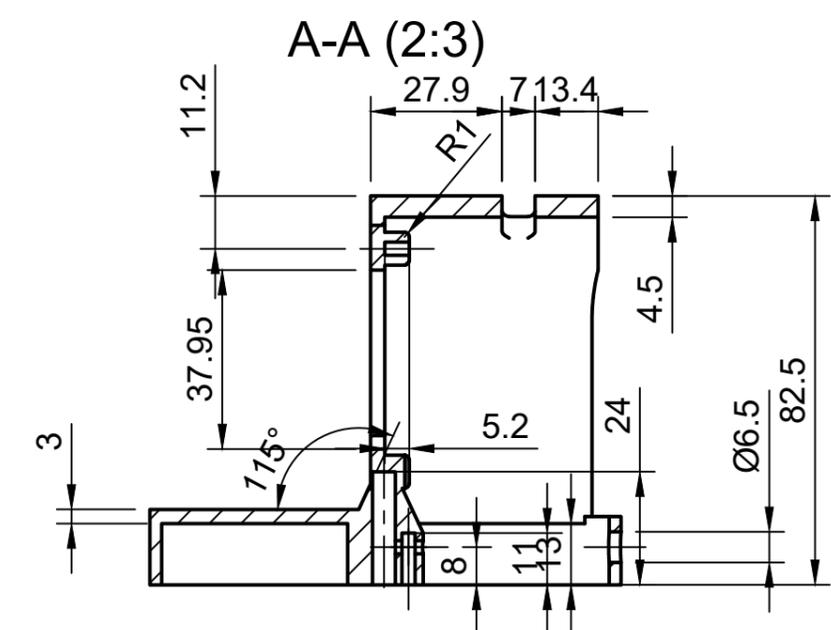
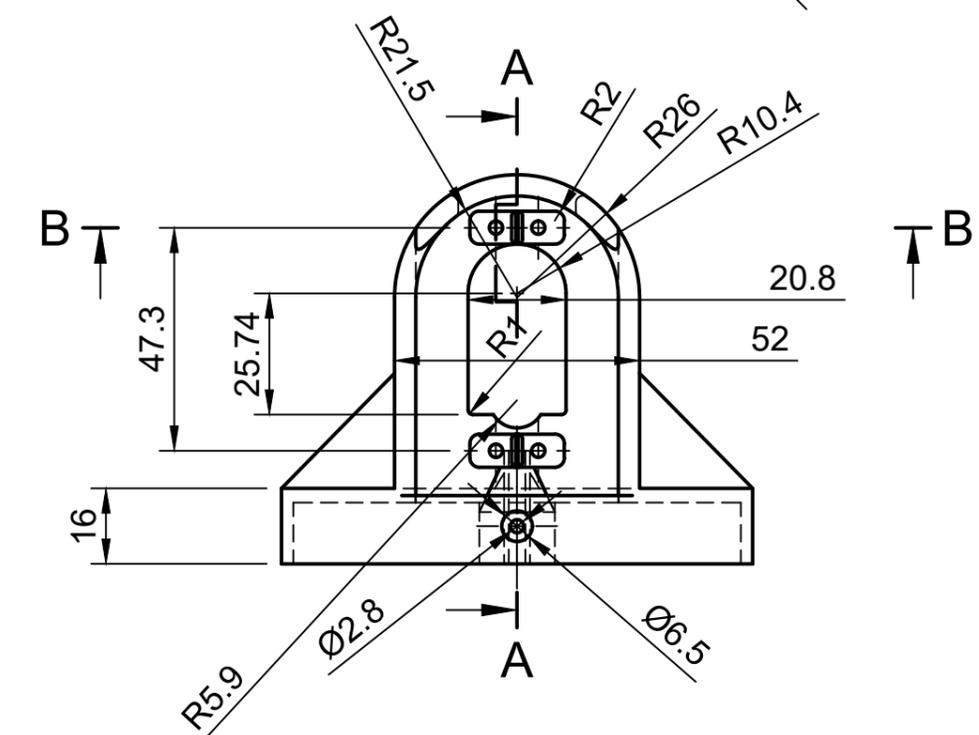
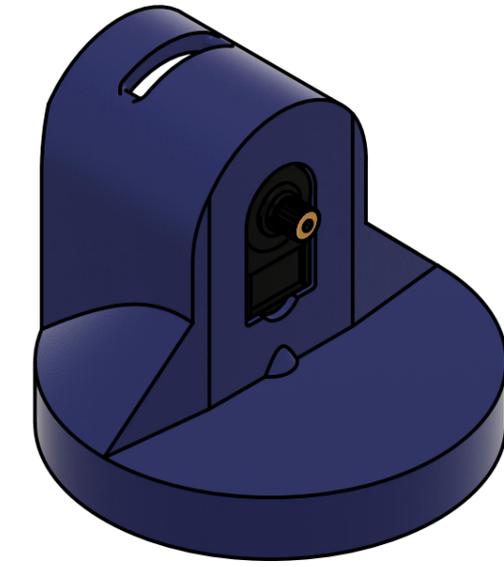
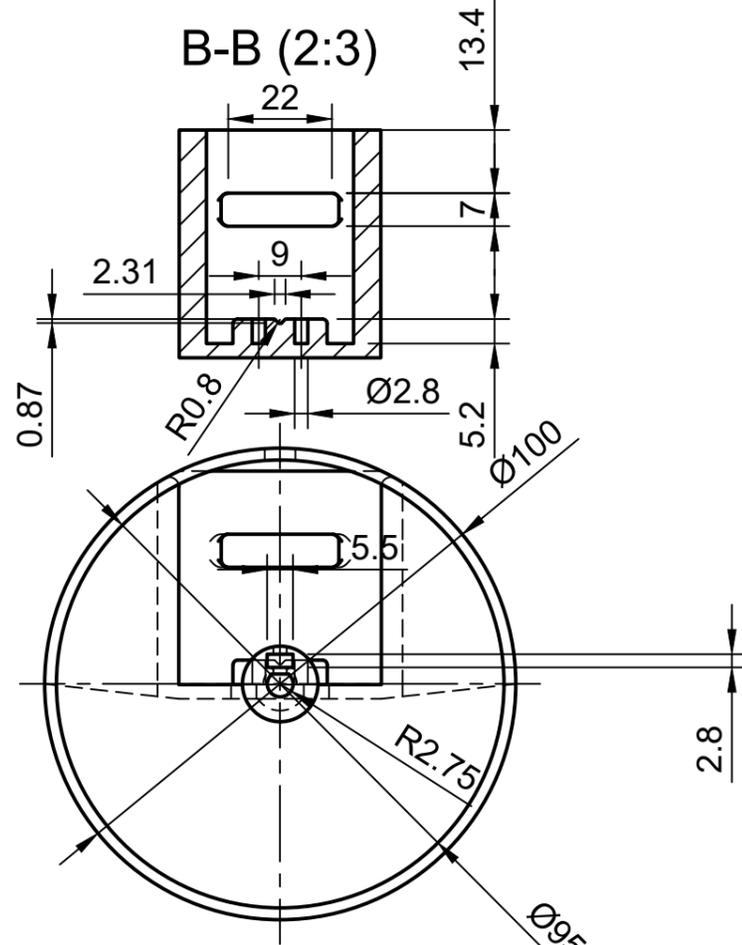


E 1:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by	
		Document type	Document status	
		Title Codo Universidad de La Laguna Digital Codesign	DWG No.	
Rev. 1.0	Date of issue 09/05/2024		Sheet 9/20	

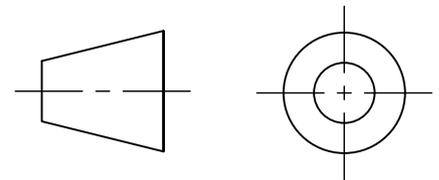
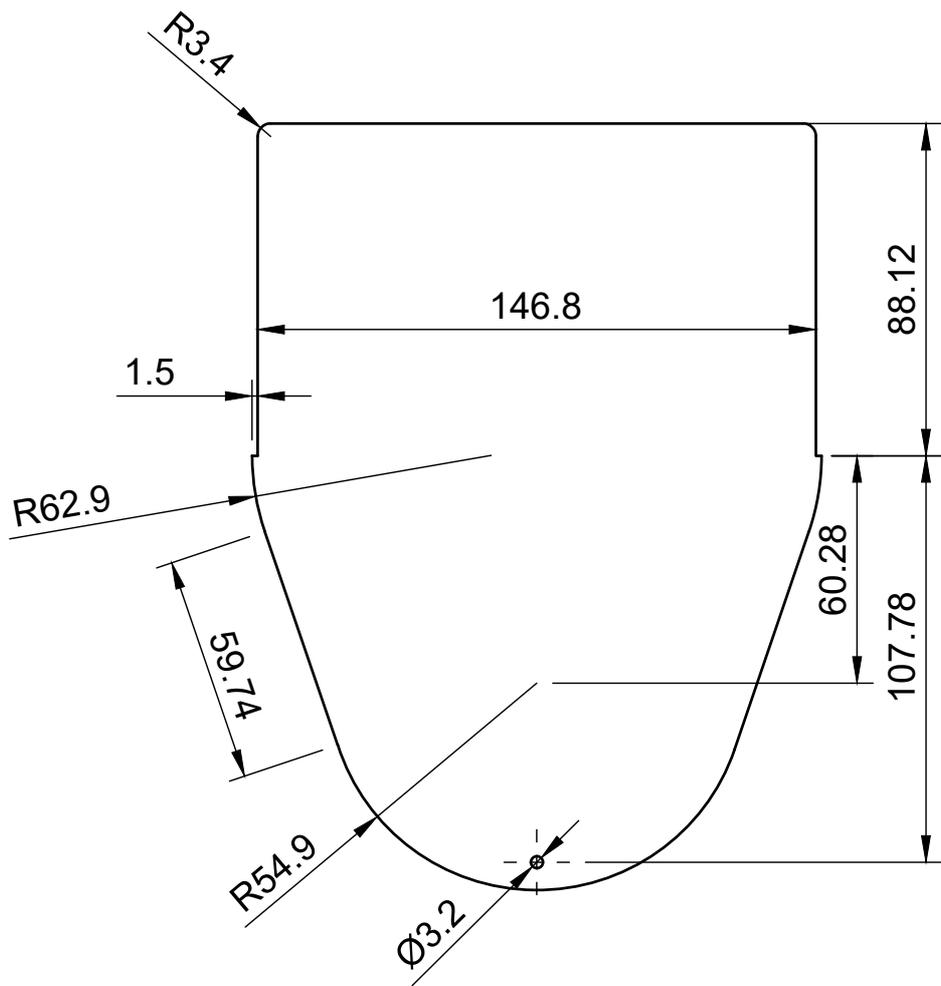
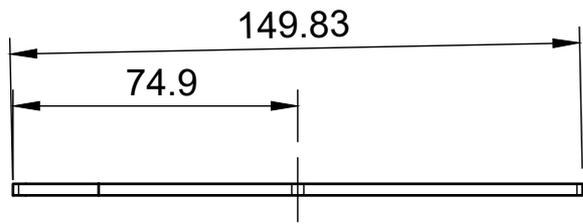


4	1	Tuerca	M3x0,5
3	1	Cuerpo Codo	Impresión 3D
2	1	MG996R	Servomotor
1	5	Tornillo	M3x8
Elemento	Ctd.	Nombre de pieza	Descipción
Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Cintura Universidad de La Laguna Digital Codesign	DWG No.
Rev. 1.0	Date of issue 09/05/2024	Sheet 10/20	



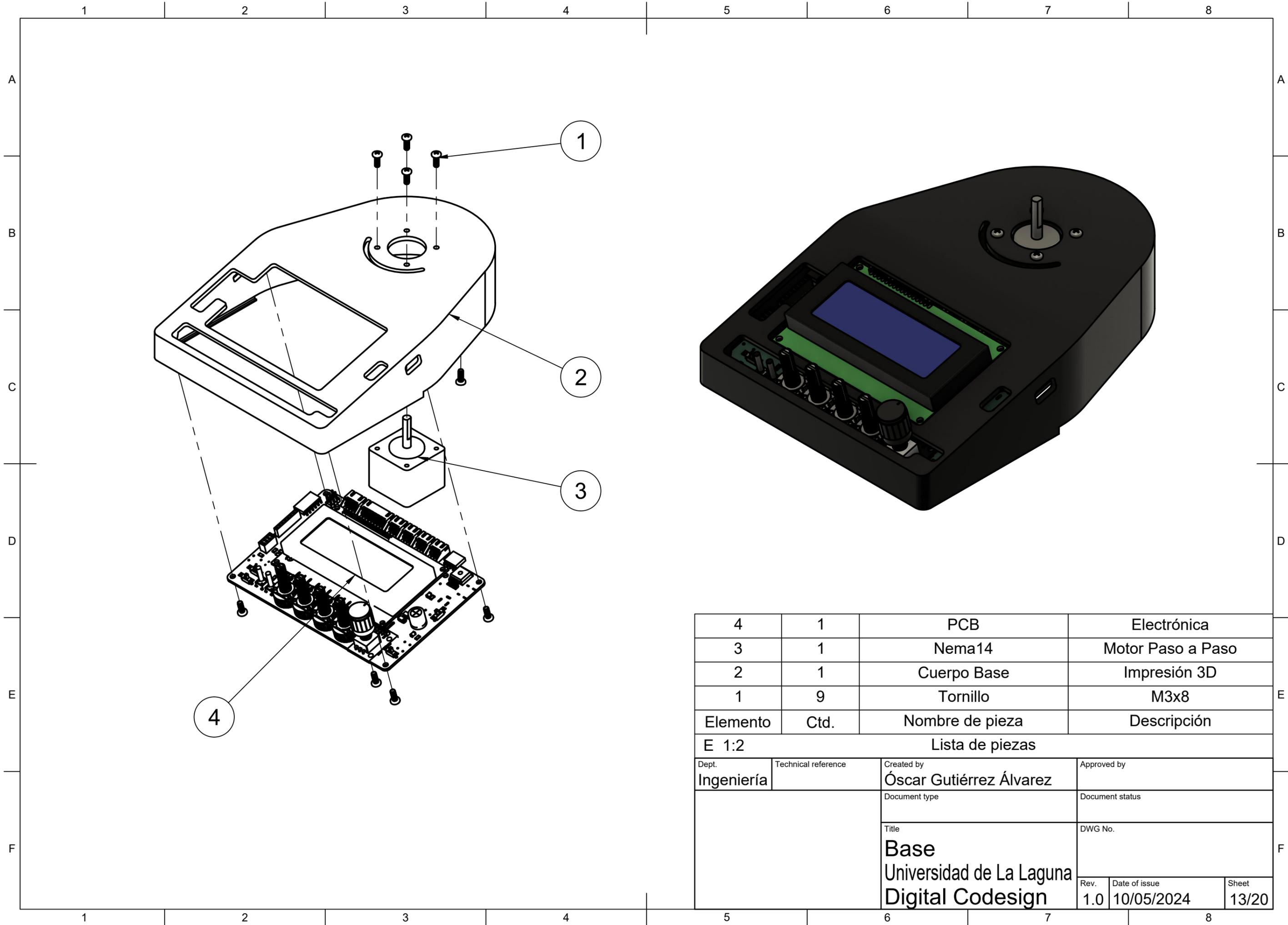
E 2:3

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by	
		Document type	Document status	
		Title Cintura Universidad de La Laguna Digital Codesign	DWG No.	
Rev. 1.0	Date of issue 09/05/2024	Sheet 11/20		



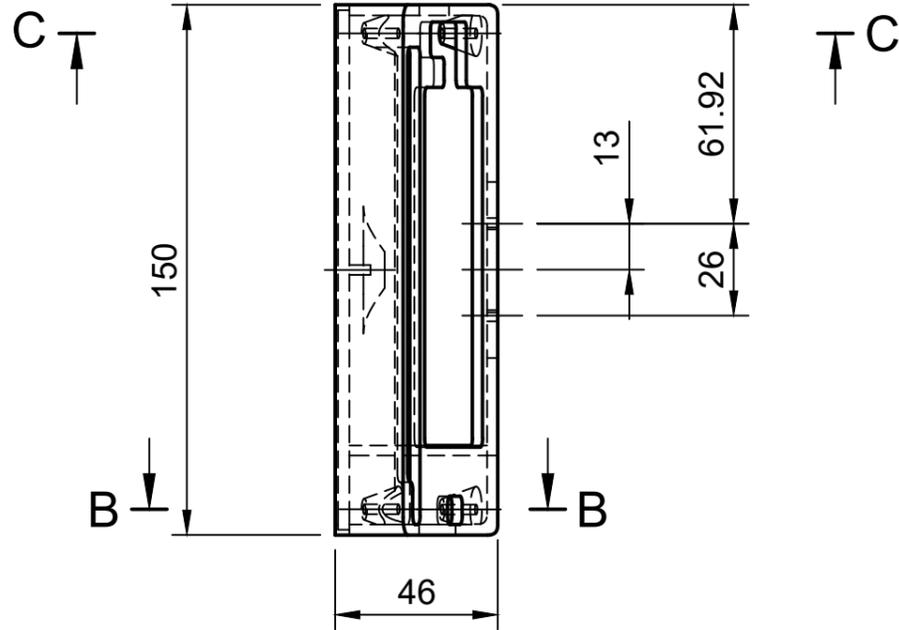
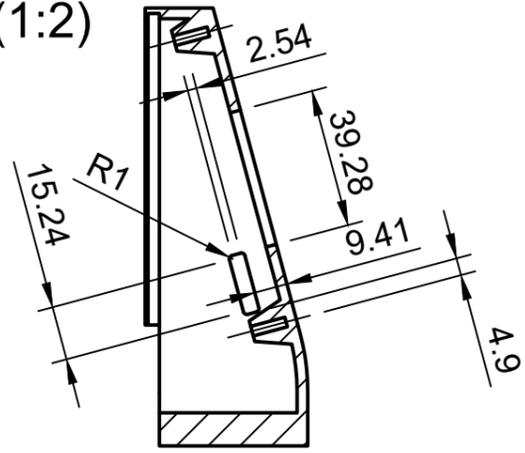
E 1:2

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title Tapa Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 09/05/2024	Sheet 12/20	

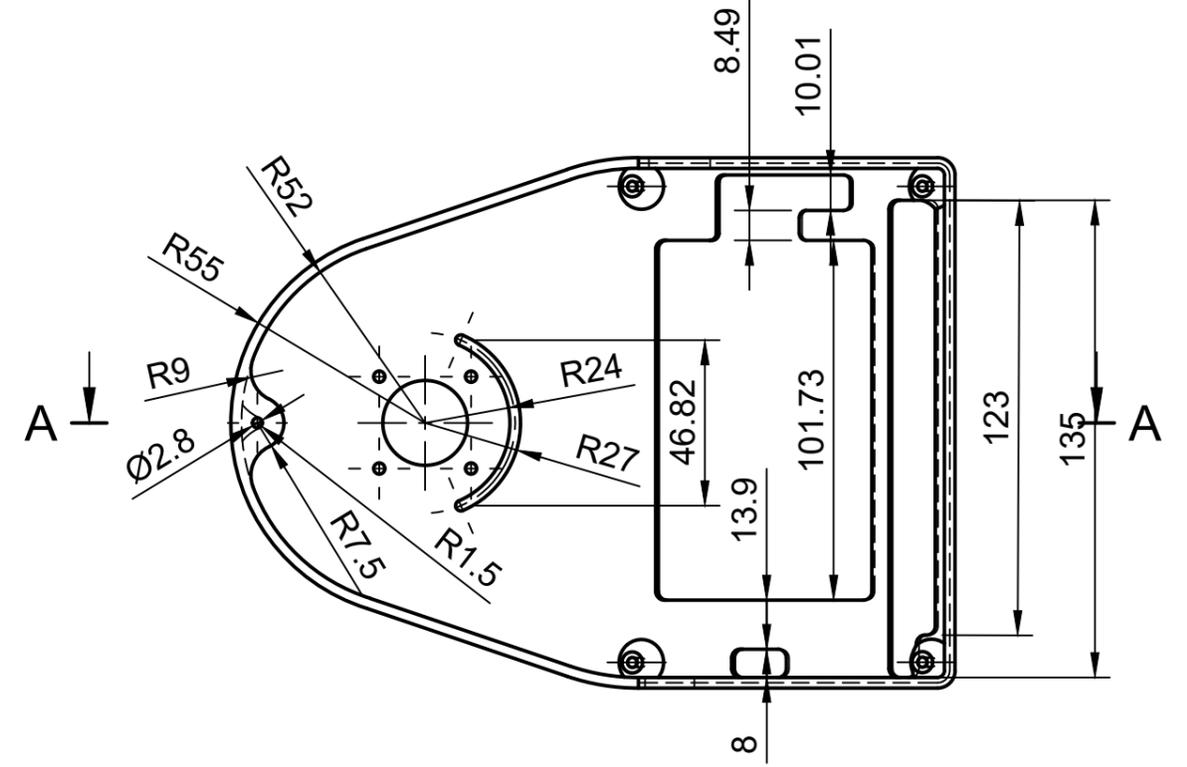
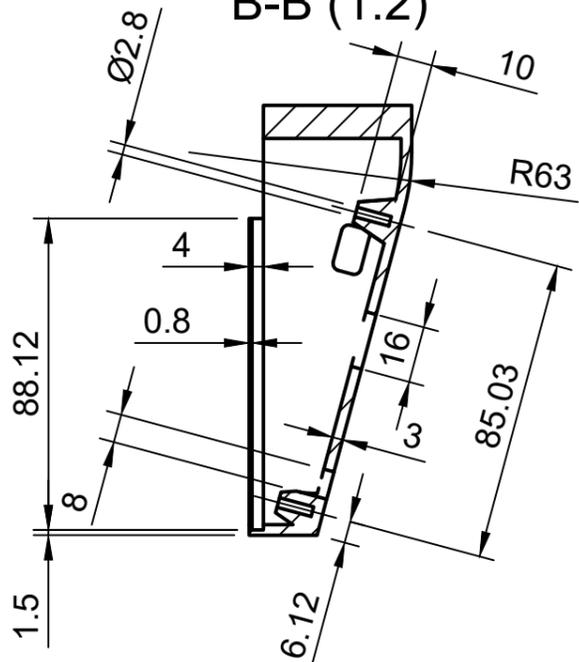


4	1	PCB	Electrónica
3	1	Nema14	Motor Paso a Paso
2	1	Cuerpo Base	Impresión 3D
1	9	Tornillo	M3x8
Elemento	Ctd.	Nombre de pieza	Descripción
E 1:2 Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Base Universidad de La Laguna Digital Codesign	DWG No.
		Rev. 1.0	Date of issue 10/05/2024
		Sheet 13/20	

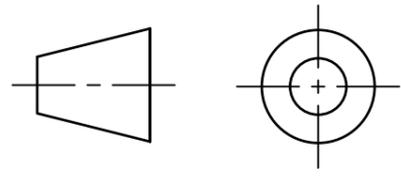
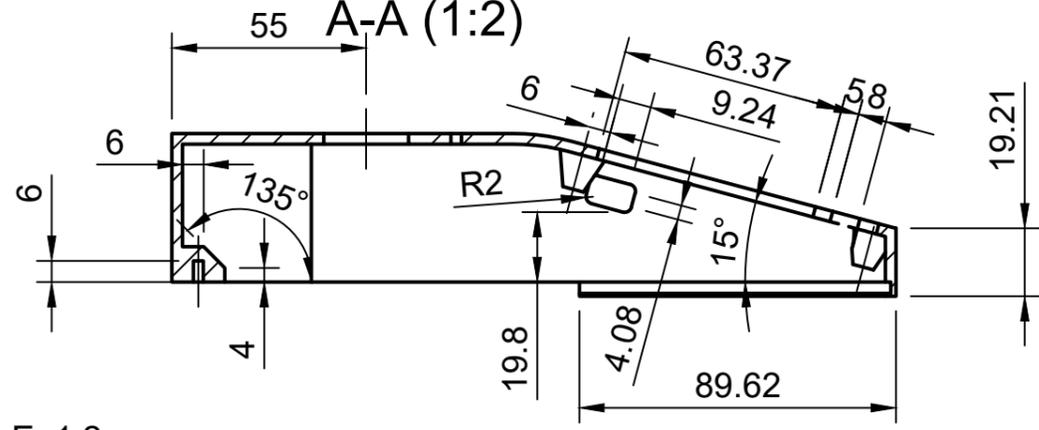
C-C (1:2)



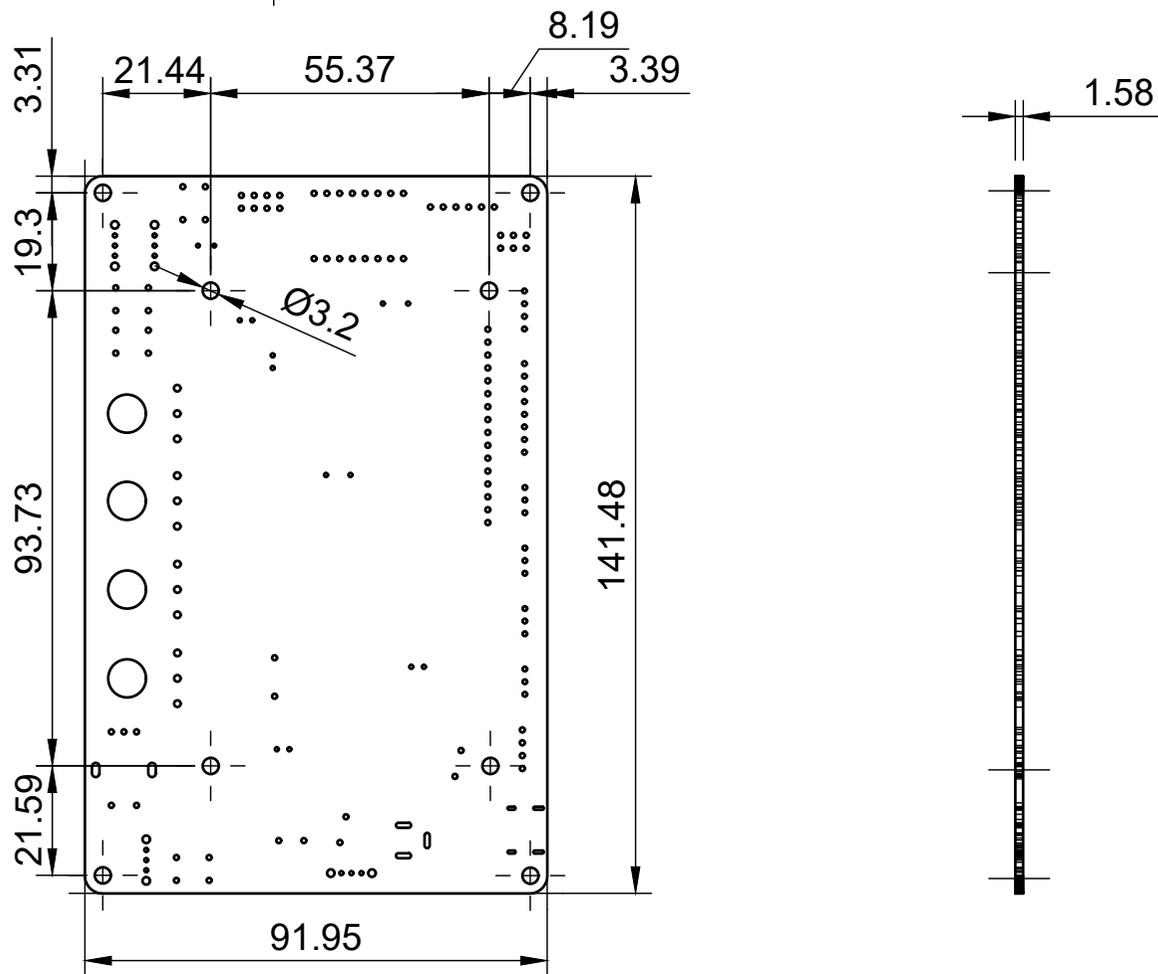
B-B (1:2)



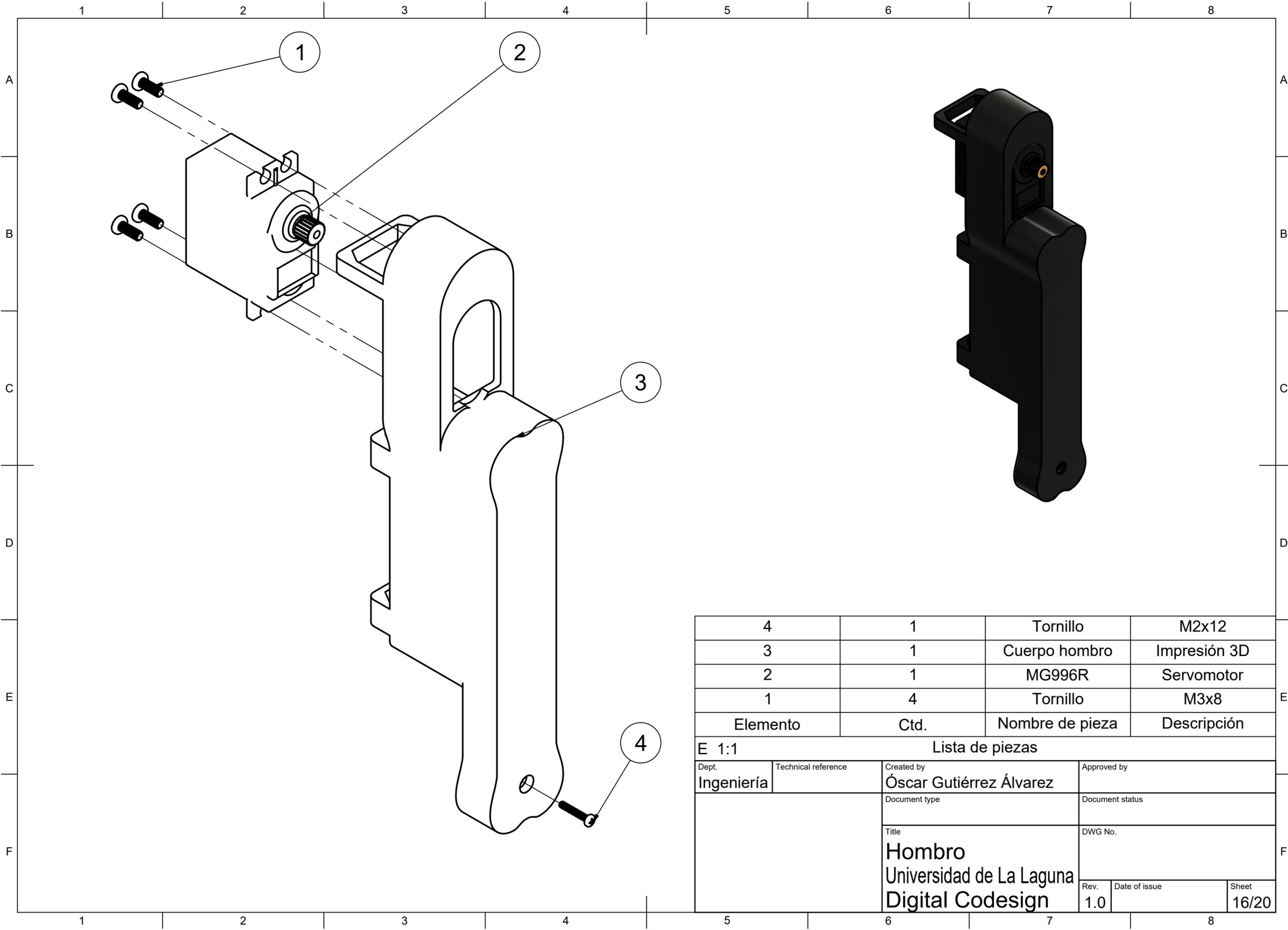
A-A (1:2)



E 1:2		Created by		Approved by	
Dept. Ingeniería		Technical reference		Óscar Gutiérrez Álvarez	
		Document type		Document status	
		Title		DWG No.	
		Base			
		Universidad de La Laguna			
		Digital Codesign			
Rev.	Date of issue	Sheet			
1.0	10/05/2024	14/20			

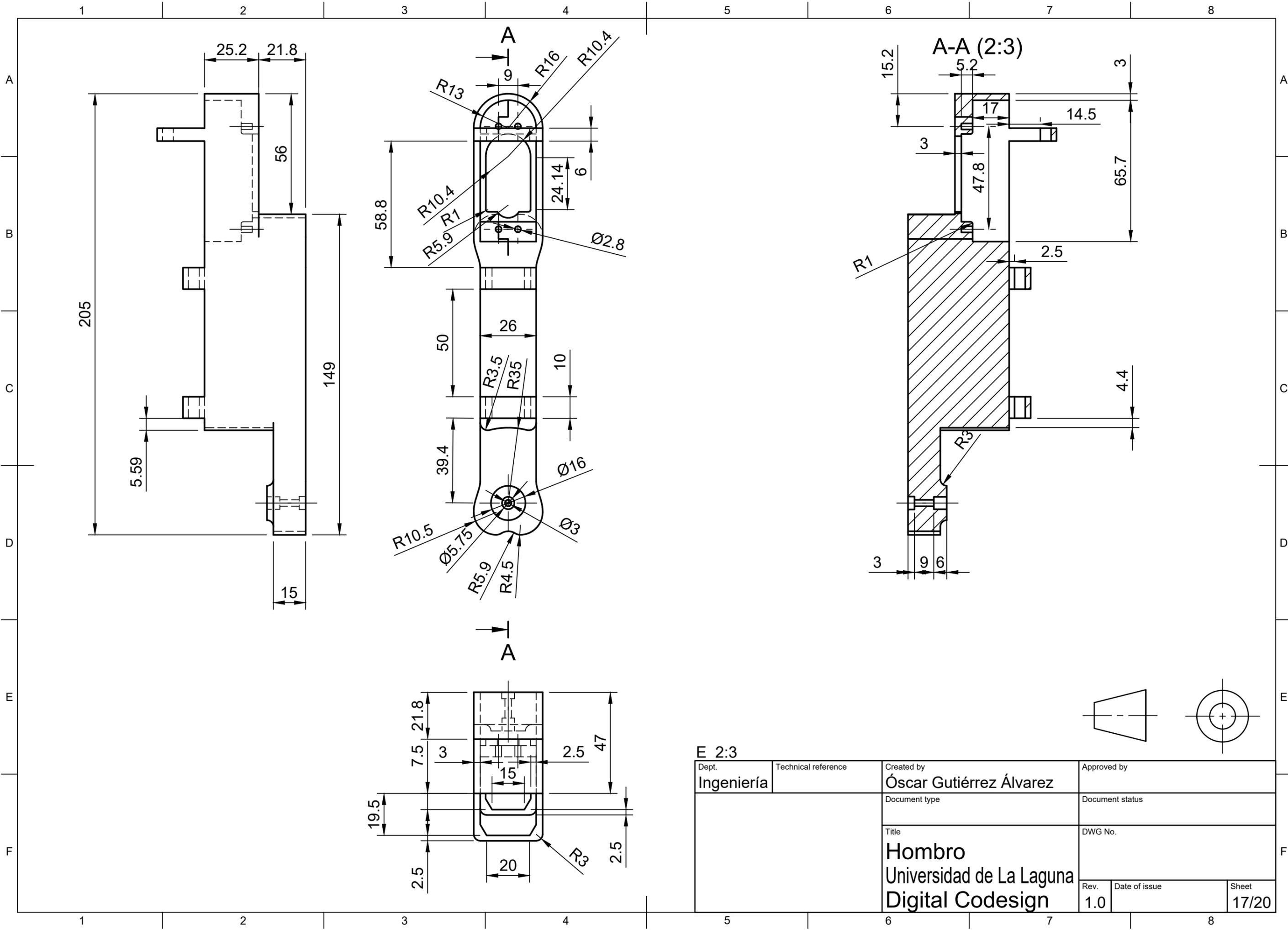


Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title PCB Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 10/05/2024	Sheet 15/20	

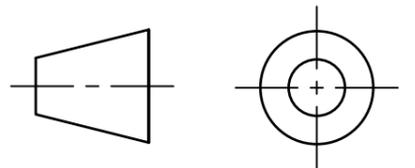


4	1	Tornillo	M2x12
3	1	Cuerpo hombro	Impresión 3D
2	1	MG996R	Servomotor
1	4	Tornillo	M3x8
Elemento	Ctd.	Nombre de pieza	Descripción

E 1:1				Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by				
		Document type	Document status				
		Title Hombro Universidad de La Laguna Digital Codesign	DWG No.				
		Rev. 1.0	Date of issue				Sheet 16/20



E 2:3			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Hombro Universidad de La Laguna Digital Codesign	DWG No.
Rev. 1.0	Date of issue	Sheet 17/20	



1

2

3

4

5

6

A

B

C

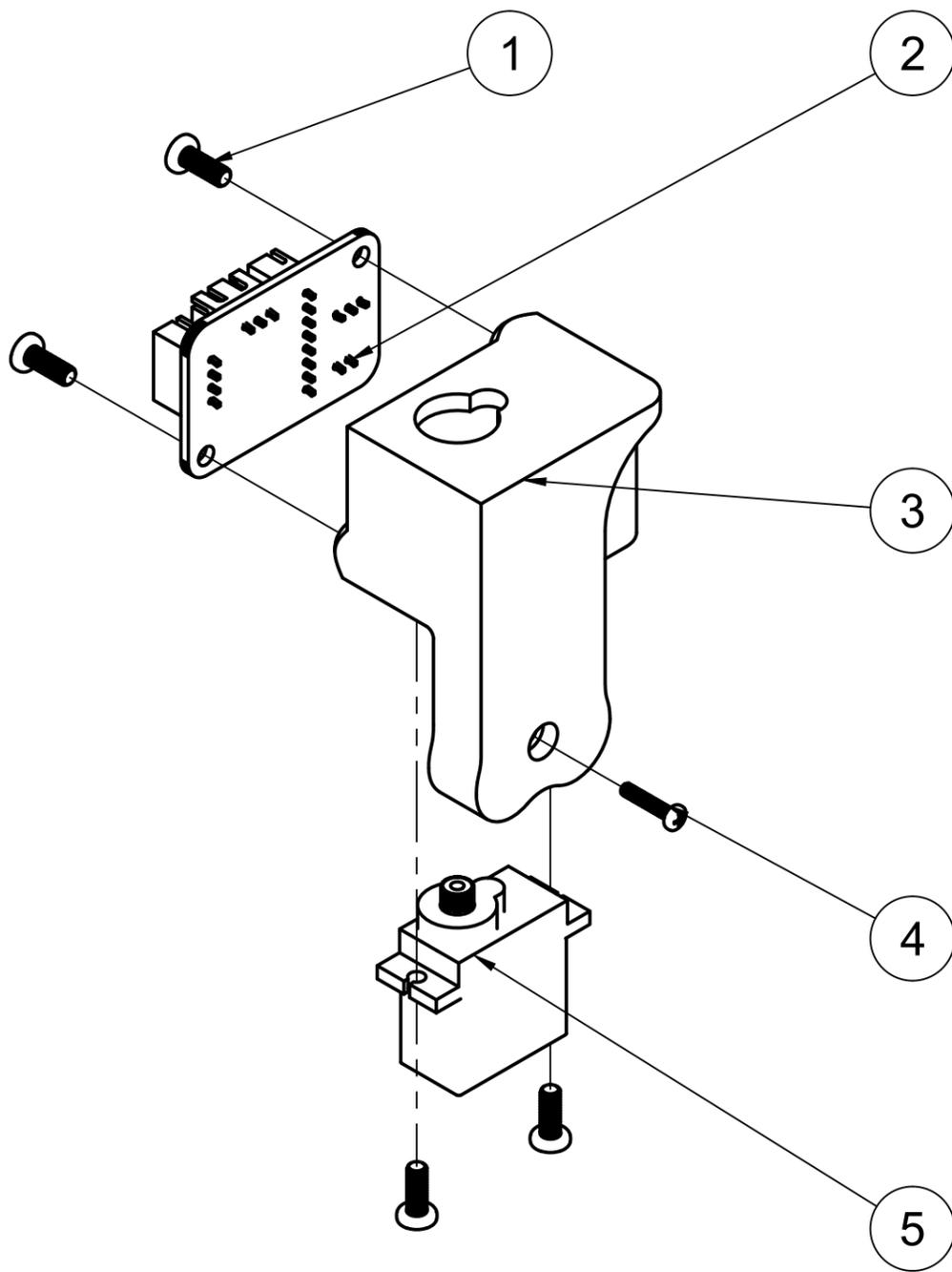
D

E

F

G

H



5	1	MG90S	Servomotor
4	1	Tornillo	M2x10
3	1	Cuerpo Muñena	Impresión 3D
2	1	PCB Auxiliar	Componente electrónico
1	4	Tornillo	M3x8
Elemento	Ctd.	Nombre de pieza	Descripción

E 1:1			
Lista de piezas			
Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by
		Document type	Document status
		Title Muñeca Universidad de La Laguna Digital Codesign	DWG No.
Rev. 1.0	Date of issue 10/05/2024	Sheet 18/20	

1

2

3

4

5

6

A

B

C

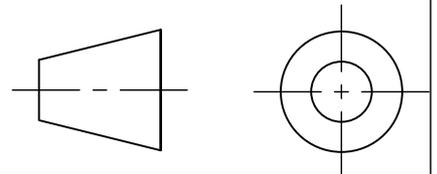
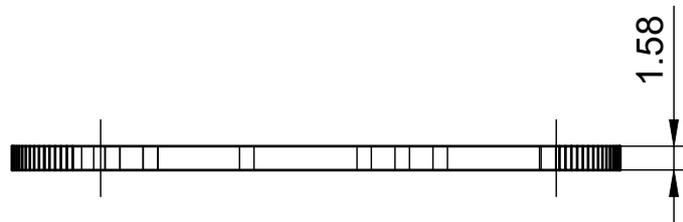
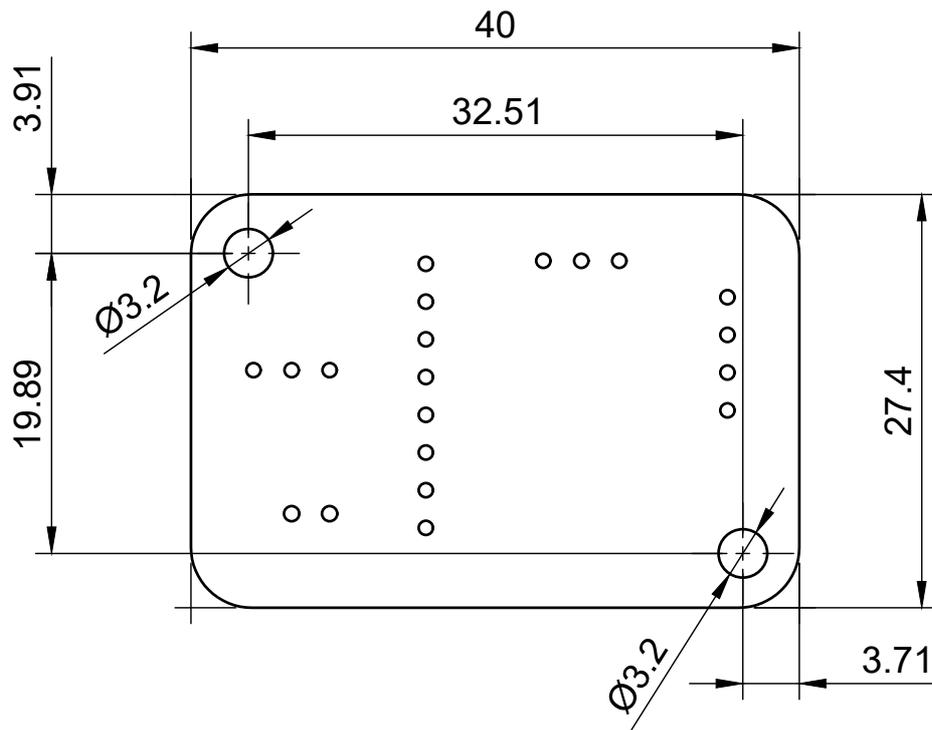
D

E

F

G

H



Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by		
		Document type	Document status		
		Title PCB Auxiliar Universidad de La Laguna Digital Codesign	DWG No.		
		Rev. 1.0	Date of issue 10/05/2024	Sheet 19/20	

1

2

3

4

5

6

A

B

C

D

E

F

G

H

A

B

C

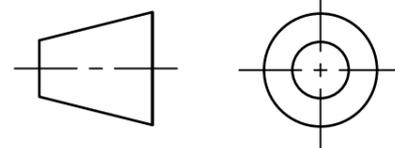
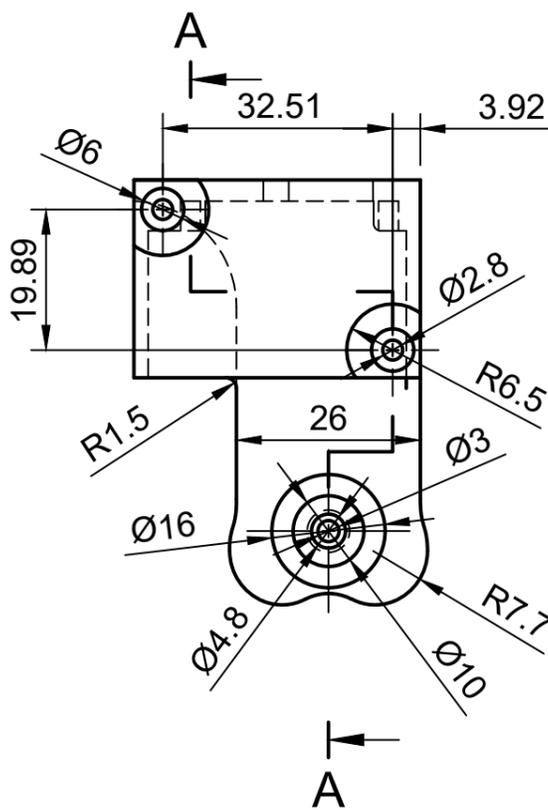
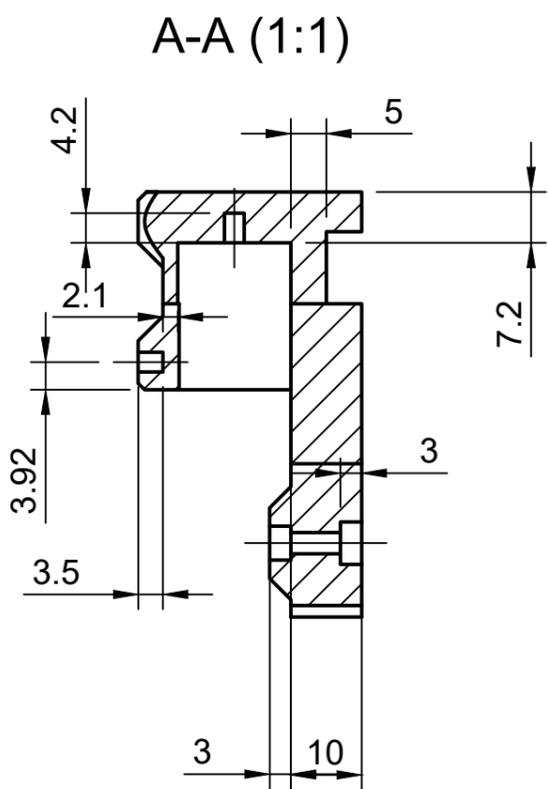
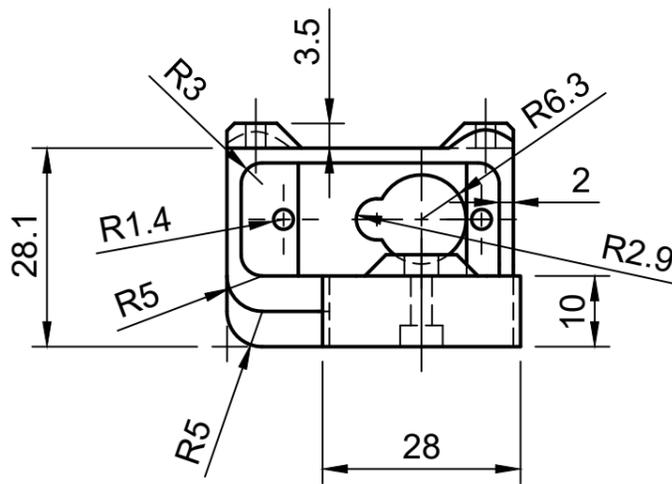
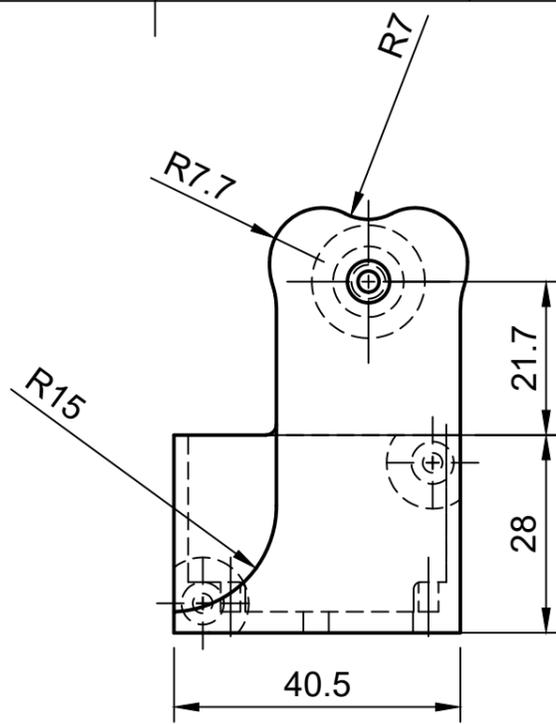
D

E

F

G

H



E 1:1

Dept. Ingeniería	Technical reference	Created by Óscar Gutiérrez Álvarez	Approved by	
		Document type	Document status	
		Title Muñeca Universidad de La Laguna Digital Codesign	DWG No.	
		Rev. 1.0	Date of issue 10/05/2024	Sheet 20/20

1

2

3

4

5

6

Anexo II - Informe de Estudios Mecánico

● Cuerpo Base

Archivo analizado	Cuerpo Base
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 11:15:15
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

○ Modelo de simulación 1:1

■ Estudio 1: tensión estática

● Propiedades del estudio

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 9:55:52

● Configuración

○ General

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

○ Malla

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Base	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa
Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)
Coefficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

- **Malla**

Tipo	Nodos	Elementos
Sólidos	15048	7501

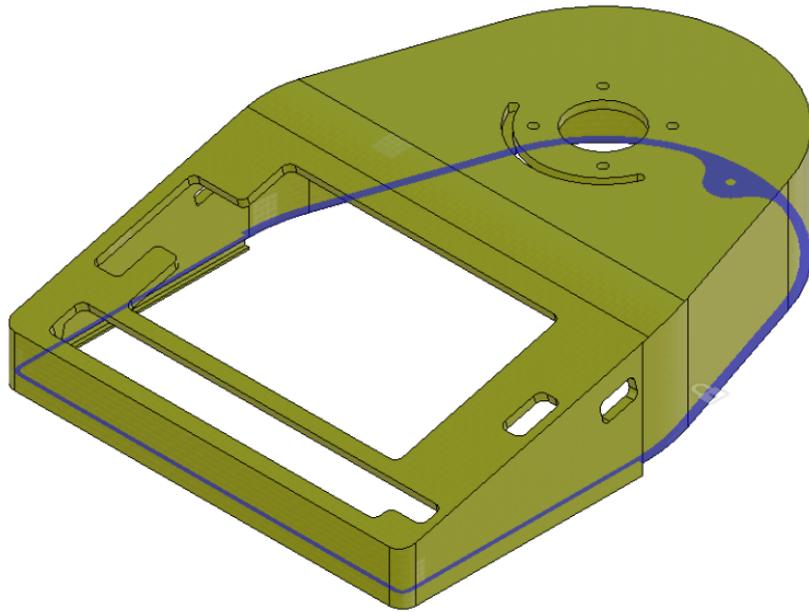
- **Caso de carga 1**

- **Restricciones**

- **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

- **Entidades seleccionadas**

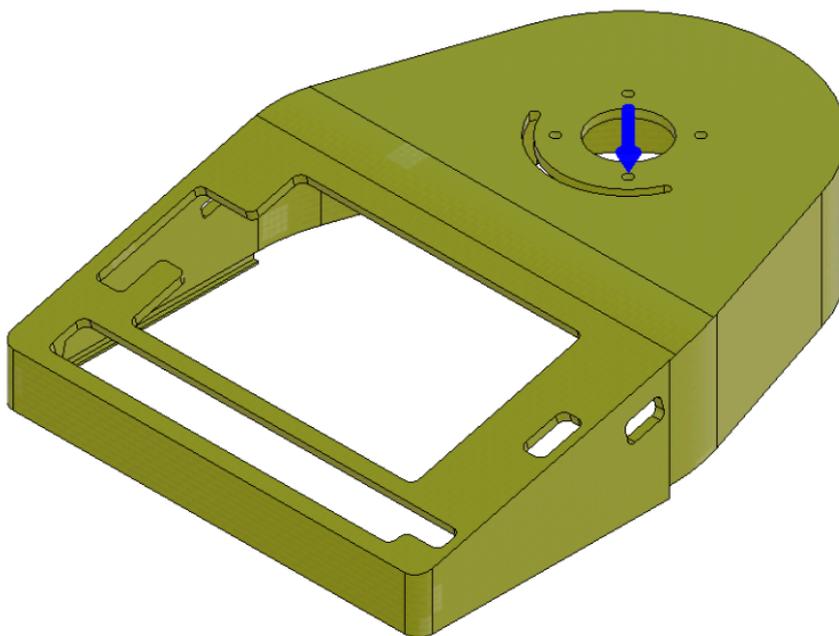


- **Cargas**

- **Gravedad**

Tipo	Gravedad
Magnitud	9.807 m / s ²
Valor X	0.00 m / s ²
Valor Y	-9.807 m / s ²
Valor Z	0.00 m / s ²

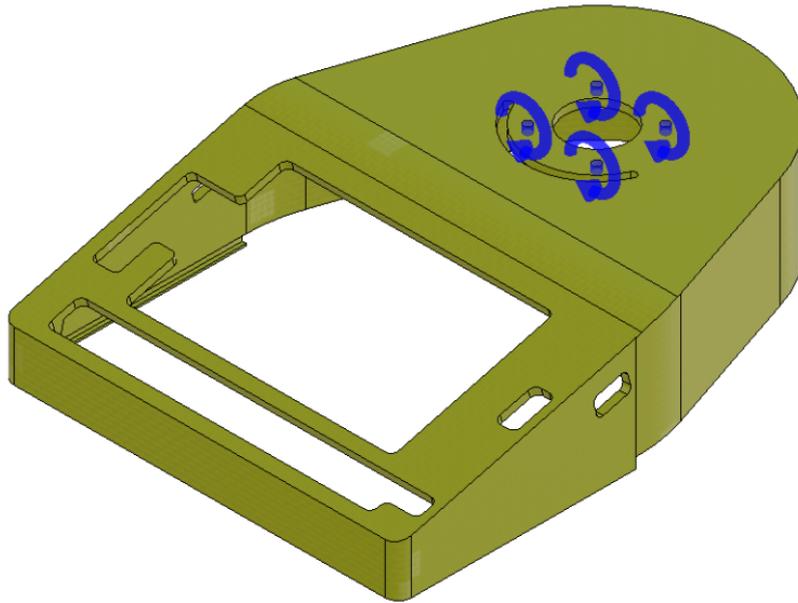
- **Entidades seleccionadas**



■ **Momento 1**

Tipo	Momento
Valor X	0.922 N mm
Valor Y	0.00 N mm
Valor Z	0.00 N mm

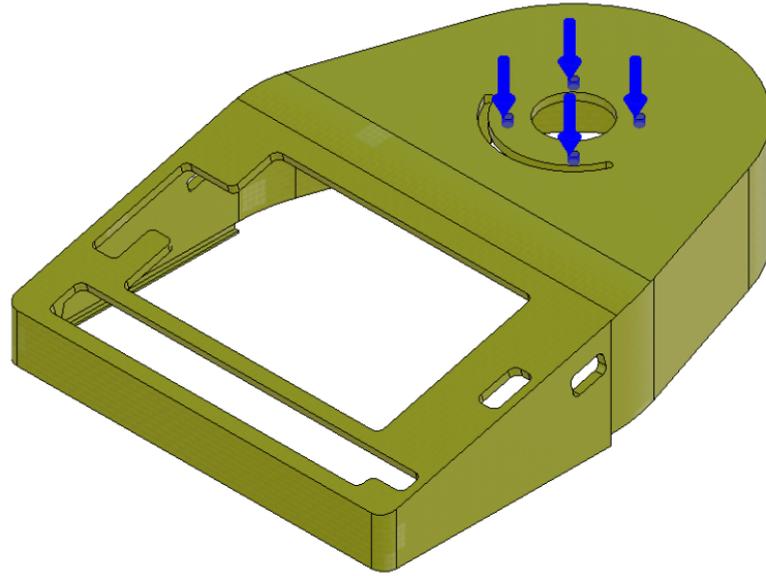
● **Entidades seleccionadas**



■ **Fuerza 1**

Tipo	Fuerza
Valor X	0.00 N
Valor Y	-1.99 N
Valor Z	0.00 N
Fuerza por entidad	No

- **Entidades seleccionadas**



- **Resultados**
 - **Resumen de resultados**

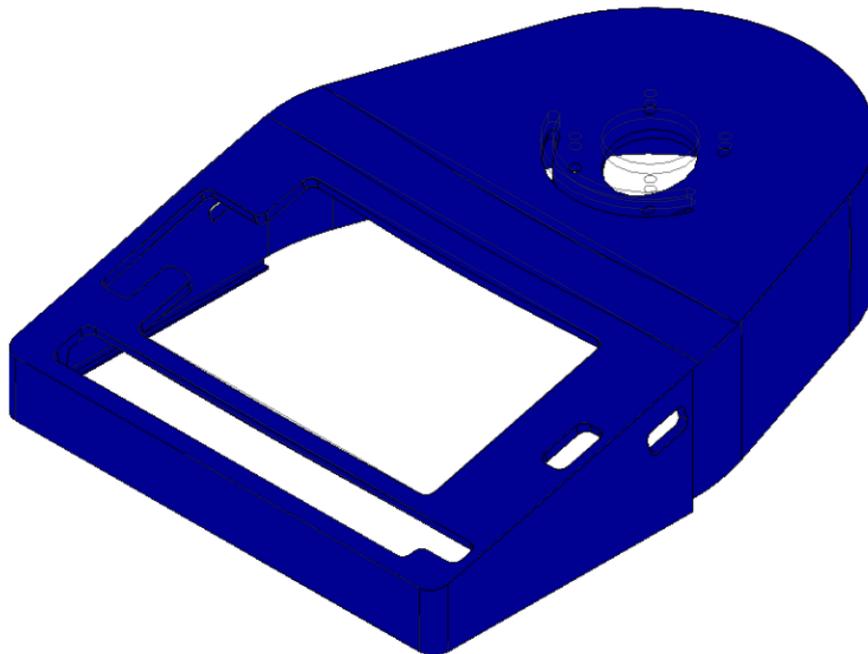
Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	1.342E-05 MPa	0.728 MPa
Primera principal	-0.154 MPa	0.724 MPa
Tercera principal	-0.913 MPa	0.042 MPa
Normal XX	-0.844 MPa	0.657 MPa
Normal YY	-0.264 MPa	0.211 MPa
Normal ZZ	-0.314 MPa	0.295 MPa
Corte XY	-0.223 MPa	0.098 MPa
Corte YZ	-0.205 MPa	0.234 MPa
Corte ZX	-0.278 MPa	0.284 MPa
Desplazamiento		
Total	0.00 mm	0.038 mm
X	-0.001 mm	9.835E-04 mm
Y	-0.038 mm	2.359E-04 mm
Z	-0.001 mm	0.001 mm
Fuerza de reacción		
Total	0.00 N	0.973 N

X	-0.172 N	0.272 N
Y	-0.338 N	0.334 N
Z	-0.824 N	0.913 N
Deformación		
Equivalente	4,60E-06	2,97E-01
Primera principal	-1,07E-05	2,71E-01
Tercera principal	-2,85E-01	-2,83E-06
Normal XX	-1,95E-01	1,73E-01
Normal YY	-5,88E-02	5,58E-02
Normal ZZ	-8,86E-02	8,74E-02
Corte XY	-1,80E-01	7,93E-02
Corte YZ	-1,65E-01	1,89E-01
Corte ZX	-2,24E-01	2,29E-01
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

■ **Factor de seguridad**

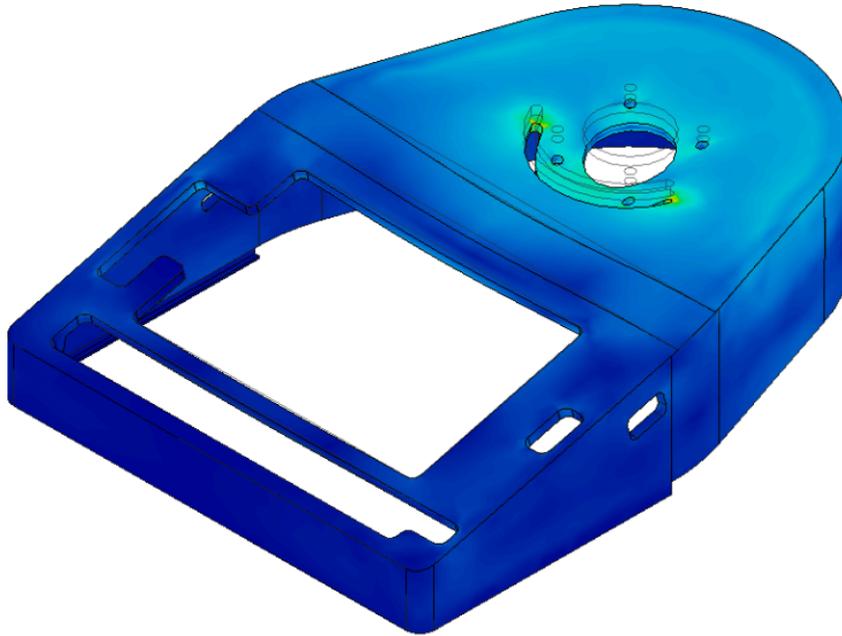
- **Coeficiente de seguridad (por cuerpo)**

0.00  8.00



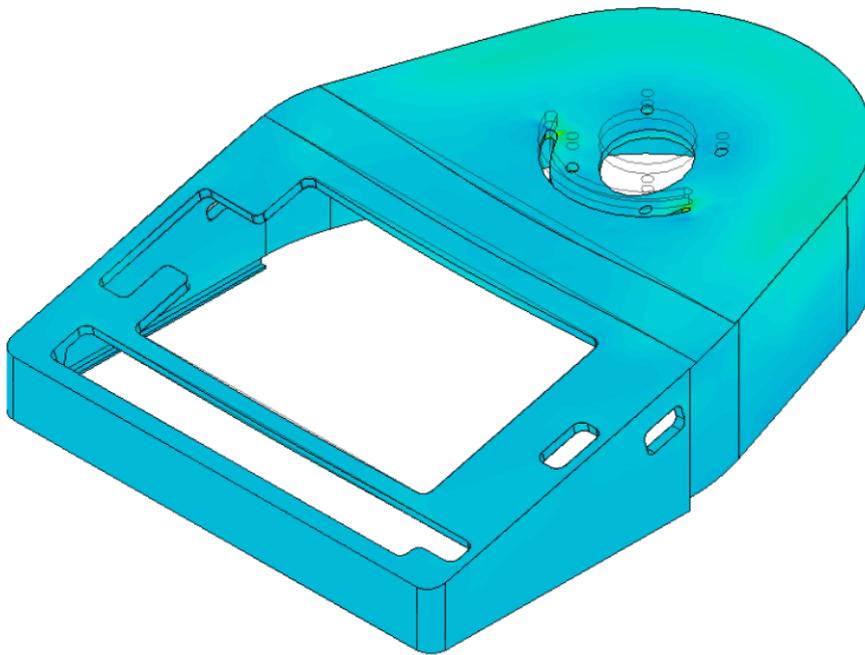
- Estrés
- von Mises

[MPa] 0.00  0.728



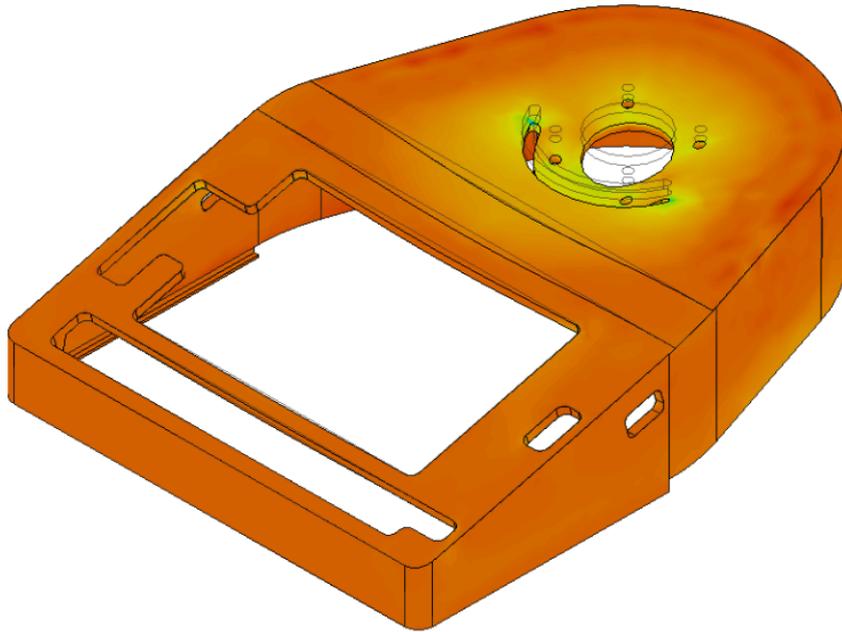
- Primera principal

[MPa] -0.154  0.724



● Tercera principal

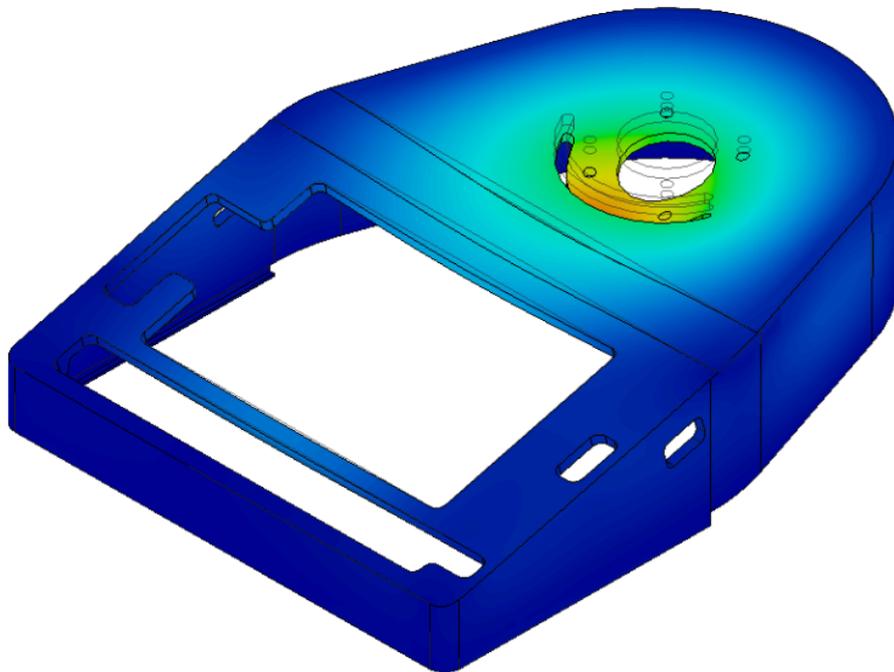
[MPa] -0.913 0.042



■ Desplazamiento

● Total

[mm] 0.00 0.038



● Cuerpo Cintura

Archivo analizado	Cuerpo Cintura
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 12:09:31
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

- **Modelo de simulación 1:1**
 - **Estudio 1: tensión estática**
 - **Propiedades del estudio**

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 09:56:38

- **Configuración**
 - **General**

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

○ Malla

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Cintura	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa
Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)
Coefficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

- **Malla**

Tipo	Nodos	Elementos
Sólidos	14841	7936

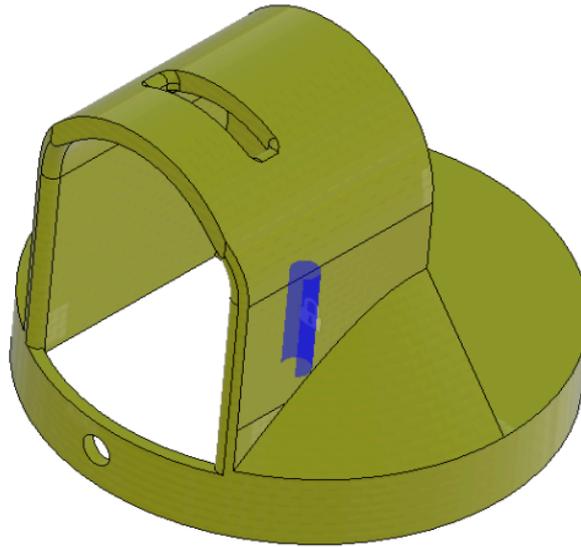
- **Caso de carga 1**

- **Restricciones**

- **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

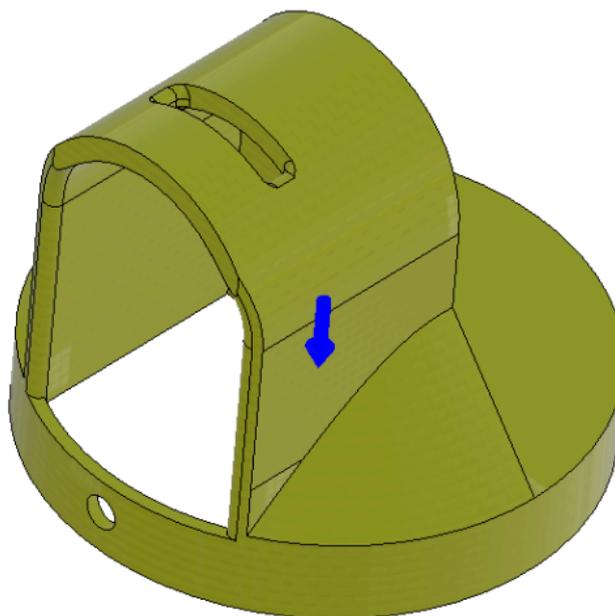
- Entidades seleccionadas



- Cargas
 - Gravedad

Tipo	Gravedad
Magnitud	9.807 m / s ²
Valor X	0.00 m / s ²
Valor Y	-9.807 m / s ²
Valor Z	0.00 m / s ²

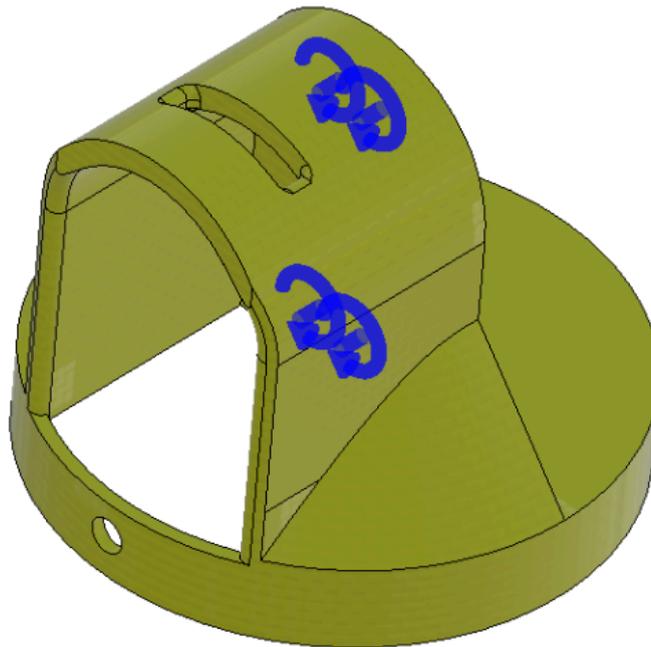
- Entidades seleccionadas



- **Momento 1**

Tipo	Momento
Valor X	0.922 N mm
Valor Y	0.00 N mm
Valor Z	0.00 N mm

- **Entidades seleccionadas**



- **Resultados**

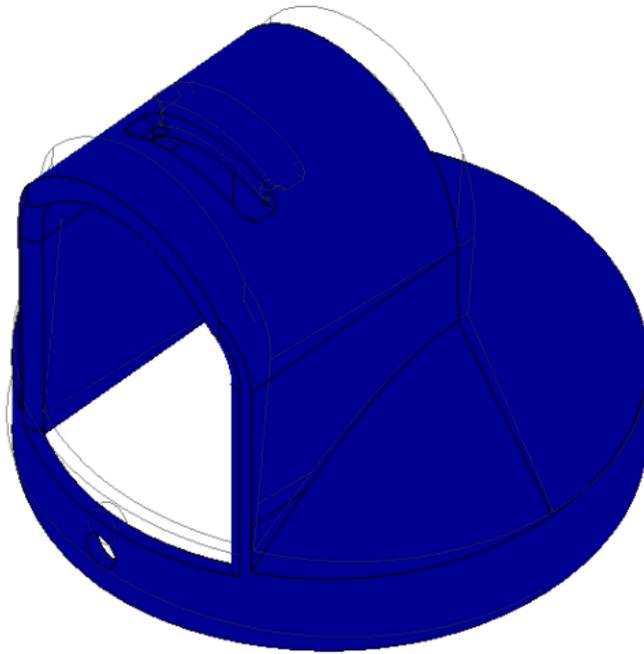
- **Resumen de resultados**

Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	9.167E-06 MPa	0.12 MPa
Primera principal	-0.038 MPa	0.084 MPa
Tercera principal	-0.134 MPa	0.015 MPa
Normal XX	-0.068 MPa	0.064 MPa
Normal YY	-0.115 MPa	0.038 MPa
Normal ZZ	-0.104 MPa	0.084 MPa
Corte XY	-0.034 MPa	0.017 MPa
Corte YZ	-0.042 MPa	0.041 MPa

Corte ZX	-0.04 MPa	0.032 MPa
Desplazamiento		
Total	0.00 mm	0.006 mm
X	-0.005 mm	0.001 mm
Y	-0.004 mm	0.002 mm
Z	-2.962E-04 mm	2.686E-04 mm
Fuerza de reacción		
Total	0.00 N	0.084 N
X	-0.036 N	0.052 N
Y	-0.04 N	0.05 N
Z	-0.063 N	0.058 N
Deformación		
Equivalente	3.162E-09	5.518E-05
Primera principal	3.216E-09	3.540E-05
Tercera principal	-5.745E-05	-2.010E-09
Normal XX	-1.762E-05	1.442E-05
Normal YY	-2.761E-05	8.805E-06
Normal ZZ	-2.315E-05	2.292E-05
Corte XY	-2.758E-05	1.398E-05
Corte YZ	-3.362E-05	3.327E-05
Corte ZX	-3.201E-05	2.600E-05
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

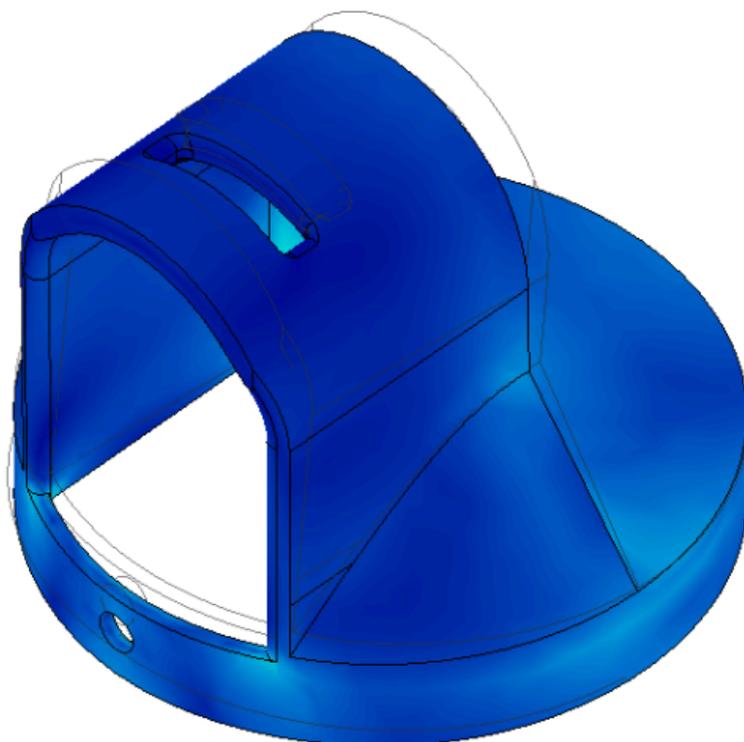
- Factor de seguridad
- Coeficiente de seguridad (por cuerpo)

0.00  8.00

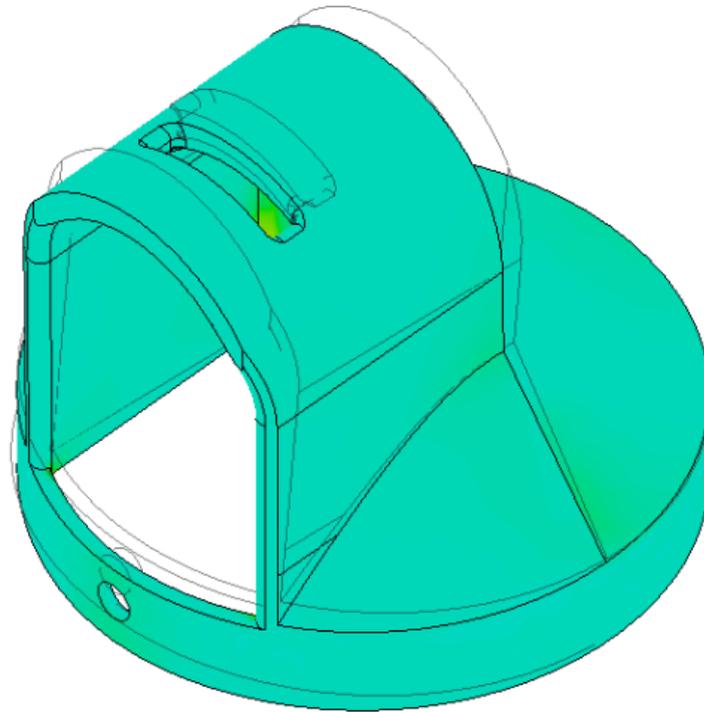


- Estrés
- von Mises

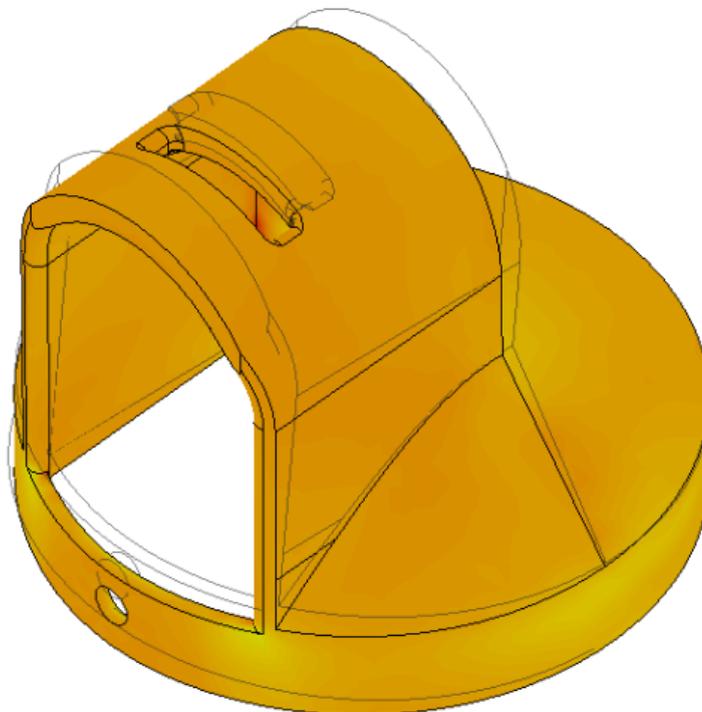
[MPa] 0.00  0.12



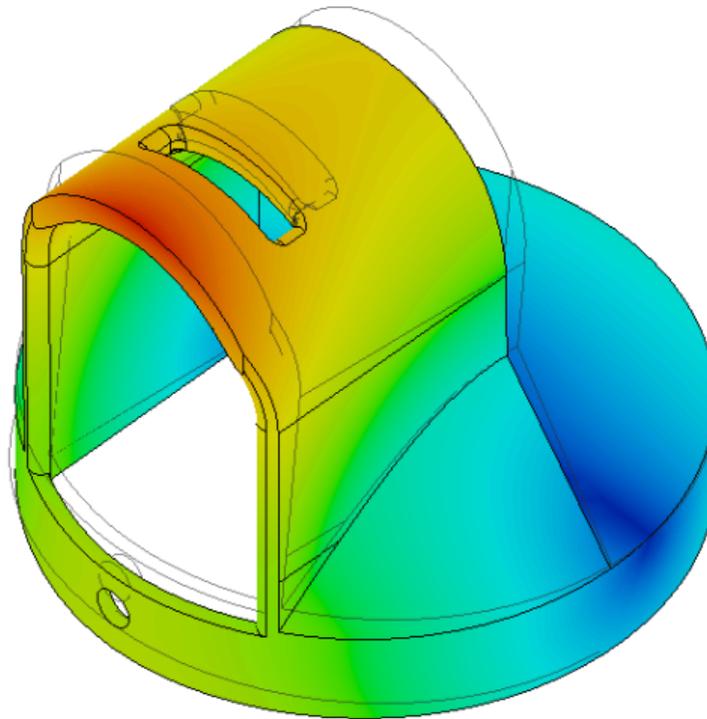
• Primera principal
[MPa] -0.038 0.084



• Tercera principal
[MPa] -0.134 0.015



■ Desplazamiento
 ● Total
 [mm] 0.00  0.006



● Cuerpo Hombro

Archivo analizado	Cuerpo Hombro
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 12:18:37
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

○ Modelo de simulación 1:1

■ Estudio 1: tensión estática

● Propiedades del estudio

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 09:57:09

- **Configuración**

- **General**

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

- **Malla**

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Base	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa
Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)

Coeficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

- **Malla**

Tipo	Nodos	Elementos
Sólidos	37726	18754

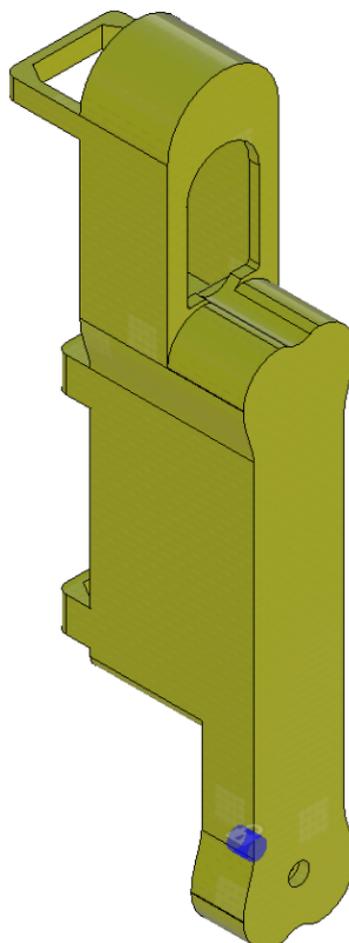
- **Caso de carga 1**

- **Restricciones**

- **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

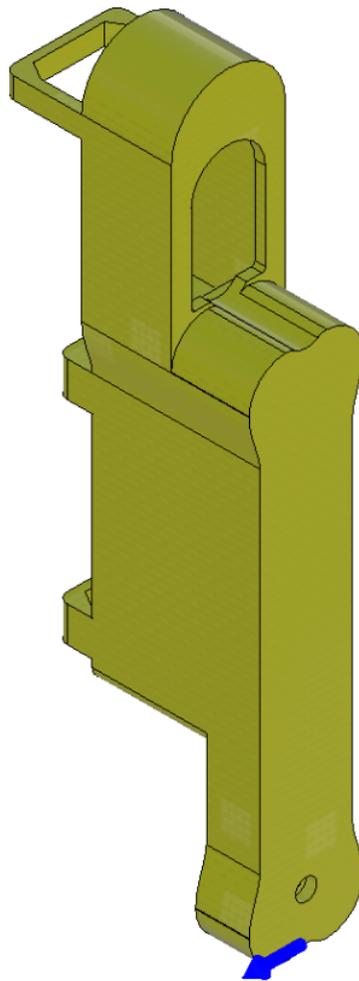
- **Entidades seleccionadas**



- **Cargas**
 - **Gravedad**

Tipo	Gravedad
Valor X	0.00 m / s ²
Valor Y	0.00 m / s ²
Valor Z	9.807 m / s ²

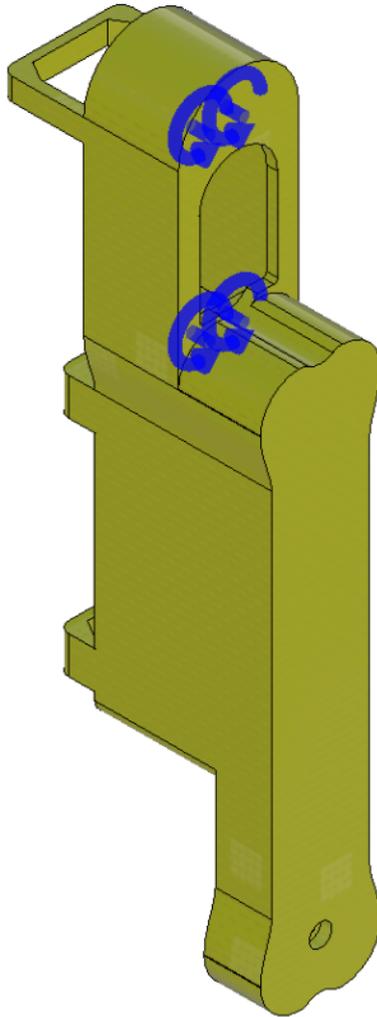
- **Entidades seleccionadas**



- **Momento 1**

Tipo	Momento
Valor X	0.338 N mm
Valor Y	0.00 N mm
Valor Z	0.00 N mm

- **Entidades seleccionadas**



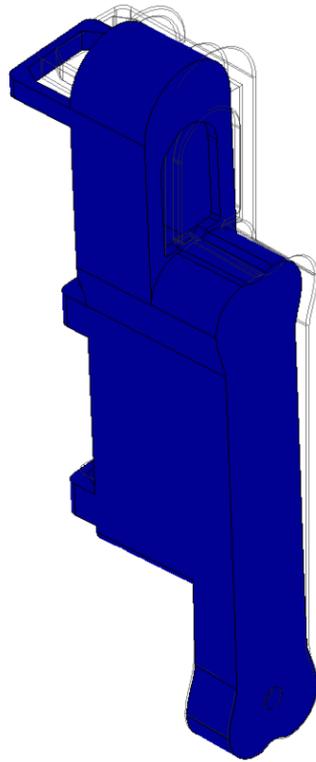
- **Resultados**
 - **Resumen de resultados**

Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	4.507E-05 MPa	0.893 MPa
Primera principal	-0.151 MPa	0.842 MPa
Tercera principal	-0.86 MPa	0.174 MPa
Normal XX	-0.426 MPa	0.41 MPa
Normal YY	-0.715 MPa	0.618 MPa
Normal ZZ	-0.706 MPa	0.834 MPa
Corte XY	-0.267 MPa	0.299 MPa
Corte YZ	-0.425 MPa	0.481 MPa

Corte ZX	-0.197 MPa	0.282 MPa
Desplazamiento		
Total	0.00 mm	0.11 mm
X	-0.008 mm	0.009 mm
Y	-0.008 mm	0.008 mm
Z	-0.007 mm	0.11 mm
Fuerza de reacción		
Total	0.00 N	0.399 N
X	-0.05 N	0.049 N
Y	-0.203 N	0.394 N
Z	-0.30 N	0.353 N
Deformación		
Equivalente	2.211E-08	4.498E-04
Primera principal	-5.058E-08	3.849E-04
Tercera principal	-3.945E-04	-9.617E-09
Normal XX	-8.808E-05	8.650E-05
Normal YY	-1.894E-04	1.715E-04
Normal ZZ	-2.211E-04	2.321E-04
Corte XY	-2.149E-04	2.408E-04
Corte YZ	-3.421E-04	3.874E-04
Corte ZX	-1.588E-04	2.276E-04
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

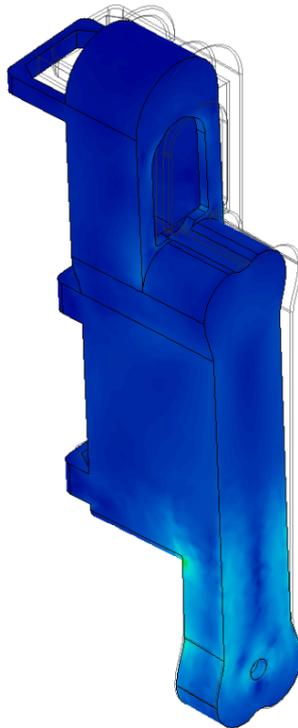
- Factor de seguridad
 - Coeficiente de seguridad (por cuerpo)

0.00  8.00



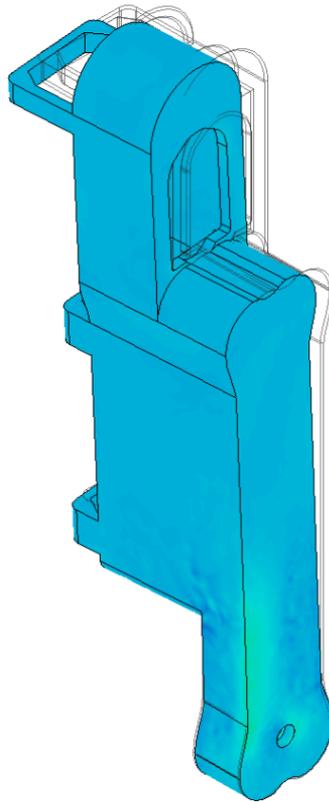
- Estrés
 - von Mises

[MPa] 0.00  0.893



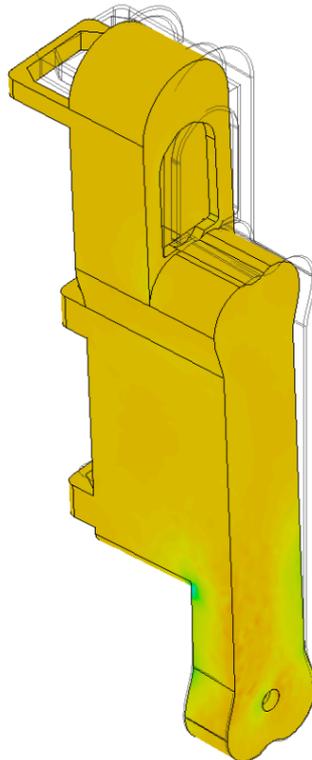
- Primera principal

[MPa] -0.151 0.842

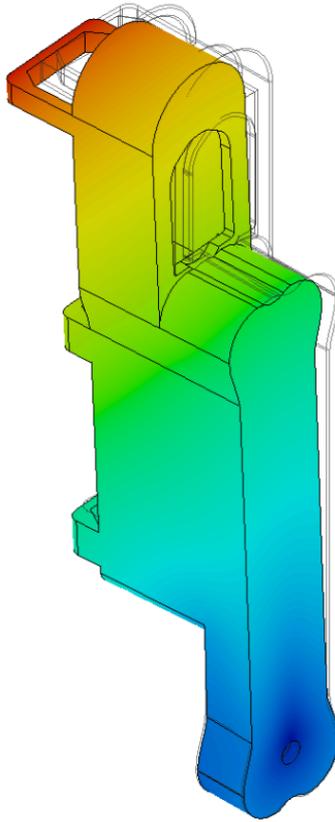


- Tercera principal

[MPa] -0.86 0.174



■ Desplazamiento
● Total
[mm] 0.00 0.11



● Cuerpo Codo

Archivo analizado	Cuerpo Codo
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 12:32:54
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

○ Modelo de simulación 1:1

■ Estudio 1: tensión estática

● Propiedades del estudio

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 10:25:34

- **Configuración**

- **General**

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

- **Malla**

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Base	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa

Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)
Coefficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

- **Malla**

Tipo	Nodos	Elementos
Sólidos	21431	10865

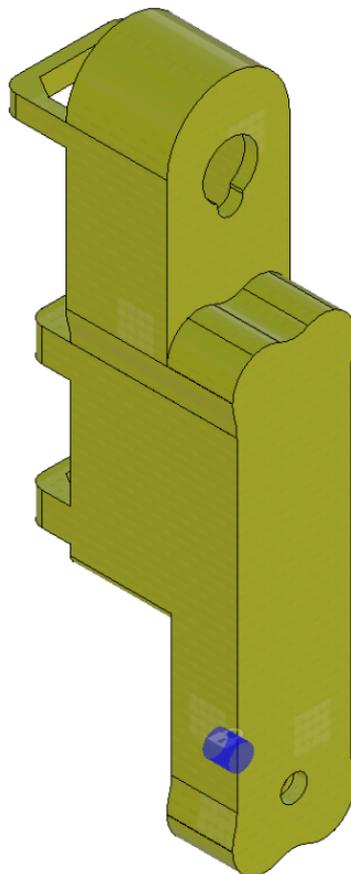
- **Caso de carga 1**

- **Restricciones**

- **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

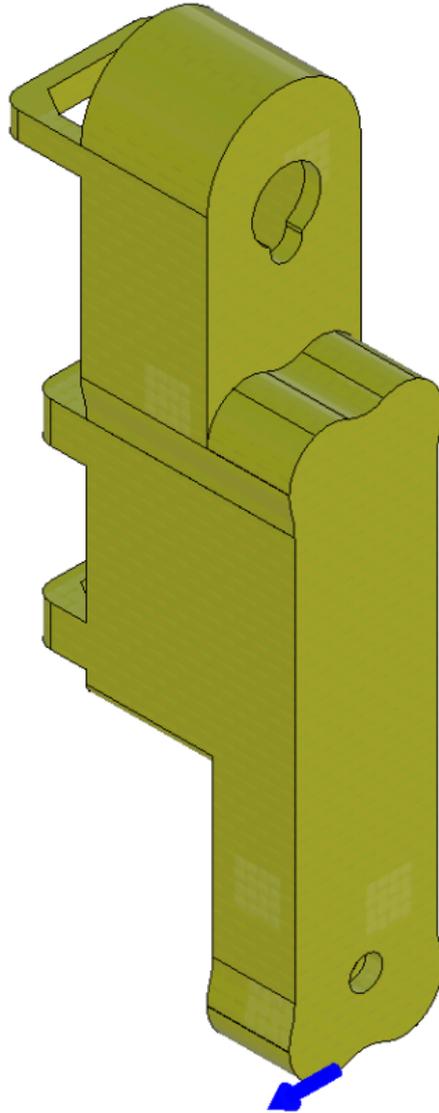
- **Entidades seleccionadas**



- **Cargas**
 - **Gravedad**

Tipo	Gravedad
Valor X	0.00 m / s ²
Valor Y	0.00 m / s ²
Valor Z	9.807 m / s ²

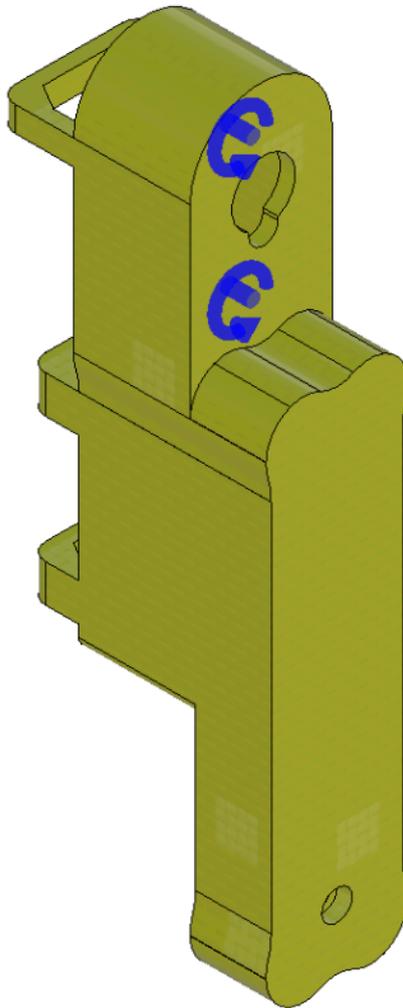
- **Entidades seleccionadas**



- **Momento 1**

Tipo	Momento
Valor X	0.109 N mm
Valor Y	0.00 N mm
Valor Z	0.00 N mm

- **Entidades seleccionadas**



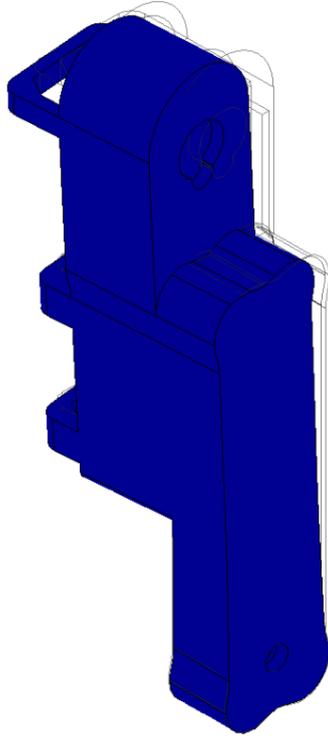
- **Resultados**
 - **Resumen de resultados**

Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	9.837E-05 MPa	0.348 MPa
Primera principal	-0.133 MPa	0.439 MPa
Tercera principal	-0.481 MPa	0.087 MPa
Normal XX	-0.305 MPa	0.269 MPa
Normal YY	-0.411 MPa	0.357 MPa
Normal ZZ	-0.301 MPa	0.327 MPa

Corte XY	-0.118 MPa	0.103 MPa
Corte YZ	-0.085 MPa	0.104 MPa
Corte ZX	-0.056 MPa	0.064 MPa
Desplazamiento		
Total	0.00 mm	0.025 mm
X	-0.003 mm	0.003 mm
Y	-0.002 mm	0.002 mm
Z	-0.001 mm	0.025 mm
Fuerza de reacción		
Total	0.00 N	0.455 N
X	-0.051 N	0.055 N
Y	-0.447 N	0.183 N
Z	-0.25 N	0.202 N
Deformación		
Equivalente	4.872E-08	1.449E-04
Primera principal	3.688E-08	1.568E-04
Tercera principal	-1.420E-04	-4.118E-08
Normal XX	-5.335E-05	4.879E-05
Normal YY	-8.180E-05	8.825E-05
Normal ZZ	-6.878E-05	8.219E-05
Corte XY	-9.469E-05	8.311E-05
Corte YZ	-6.887E-05	8.371E-05
Corte ZX	-4.530E-05	5.185E-05
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

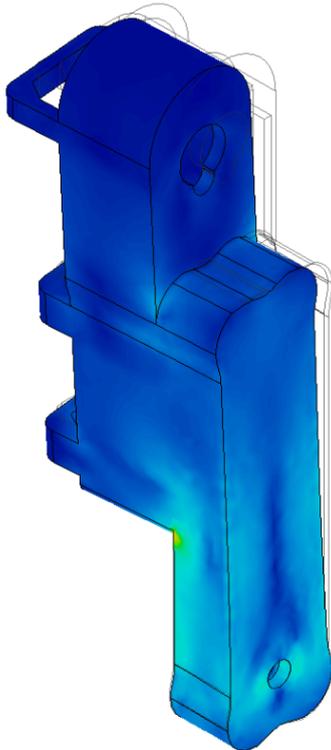
- Factor de seguridad
 - Coeficiente de seguridad (por cuerpo)

0.00  8.00



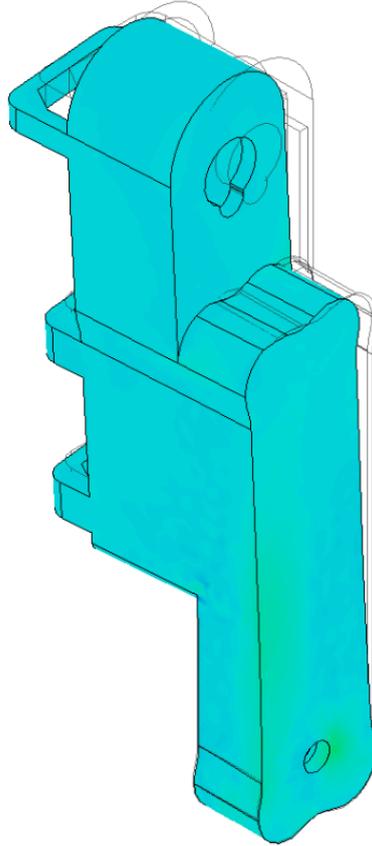
- Estrés
 - von Mises

[MPa] 0.00  0.348



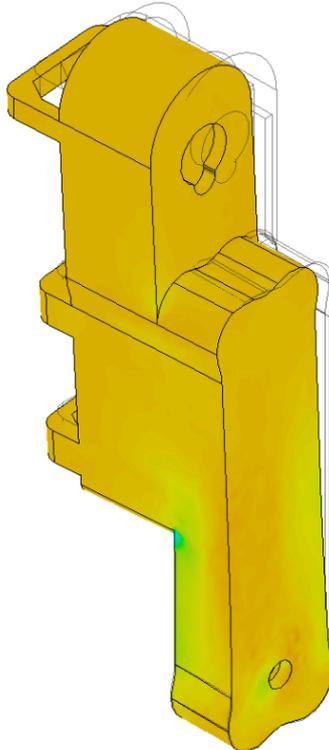
• **Primera principal**

[MPa] -0.133 0.439

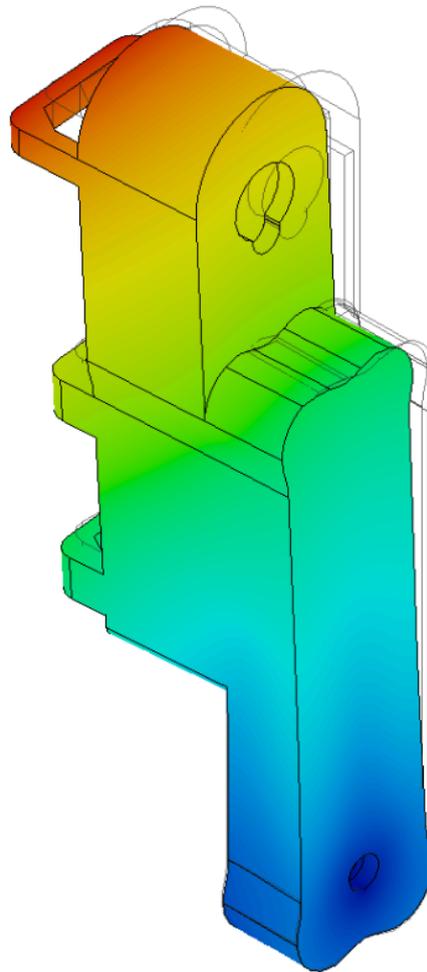


• **Tercera principal**

[MPa] -0.481 0.087



■ Desplazamiento
● Total
[mm] 0.00 0.025



● Cuerpo Muñeca

Archivo analizado	Cuerpo Muñeca
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 13:45:43
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

- **Modelo de simulación 1:1**
 - **Estudio 1: tensión estática**
 - **Propiedades del estudio**

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 10:35:08

- **Configuración**
 - **General**

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

- **Malla**

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Base	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
----------	--------------------------------

Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa
Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)
Coefficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

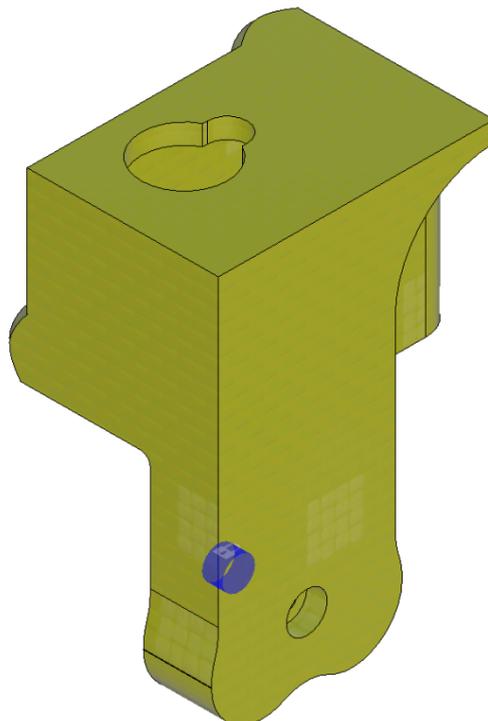
- **Malla**

Tipo	Nodos	Elementos
Sólidos	43302	22091

- **Caso de carga 1**
 - **Restricciones**
 - **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

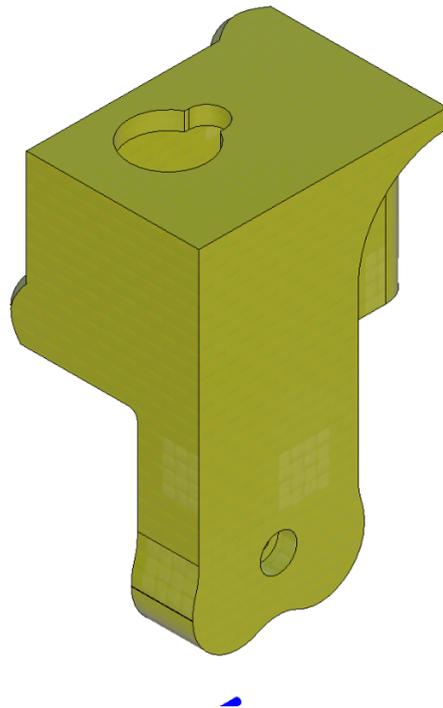
- **Entidades seleccionadas**



- **Cargas**
 - **Gravedad**

Tipo	Gravedad
Magnitud	9.807 m / s ²
Valor X	0.00 m / s ²
Valor Y	0.00 m / s ²
Valor Z	9.81 m / s ²

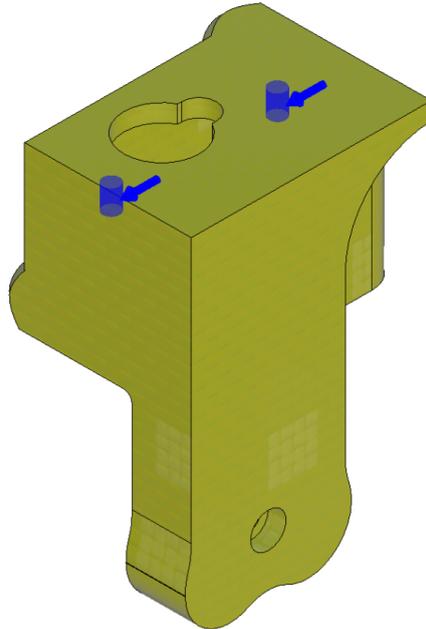
- **Entidades seleccionadas**



- **Fuerza 1**

Tipo	Fuerza
Valor X	0.00 N
Valor Y	0.00 N
Valor Z	1.15 N
Fuerza por entidad	No

- Entidades seleccionadas



- Resultados

- Resumen de resultados

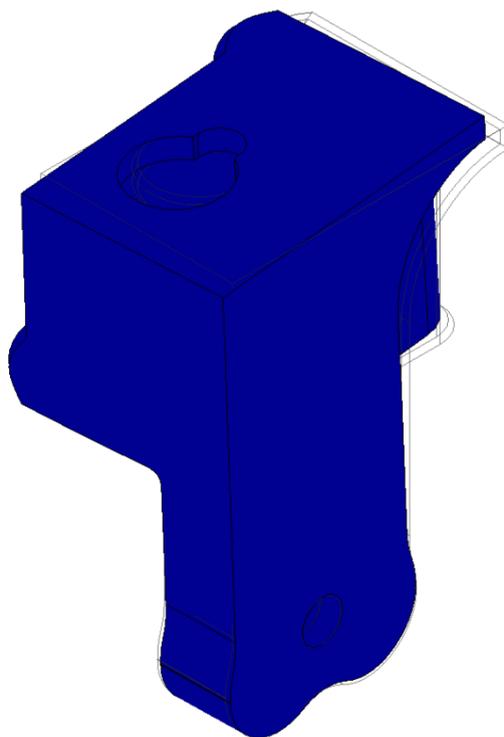
Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	2.769E-04 MPa	2.324 MPa
Primera principal	-0.362 MPa	1.666 MPa
Tercera principal	-1.714 MPa	0.379 MPa
Normal XX	-0.899 MPa	0.812 MPa
Normal YY	-1.449 MPa	1.264 MPa
Normal ZZ	-1.708 MPa	1.419 MPa
Corte XY	-0.471 MPa	0.53 MPa
Corte YZ	-1.302 MPa	1.316 MPa
Corte ZX	-0.59 MPa	0.396 MPa
Desplazamiento		
Total	0.00 mm	0.08 mm
X	-0.032 mm	0.011 mm
Y	-0.012 mm	0.026 mm
Z	-0.013 mm	0.072 mm
Fuerza de reacción		
Total	0.00 N	1.108 N

X	-0.241 N	0.274 N
Y	-1.081 N	1.056 N
Z	-1.049 N	0.991 N
Deformación		
Equivalente	1.150E-07	0.001
Primera principal	1.005E-07	0.001
Tercera principal	-0.001	-9.205E-08
Normal XX	-1.641E-04	1.650E-04
Normal YY	-5.324E-04	4.948E-04
Normal ZZ	-5.745E-04	5.217E-04
Corte XY	-3.798E-04	4.268E-04
Corte YZ	-0.001	0.001
Corte ZX	-4.753E-04	3.188E-04
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

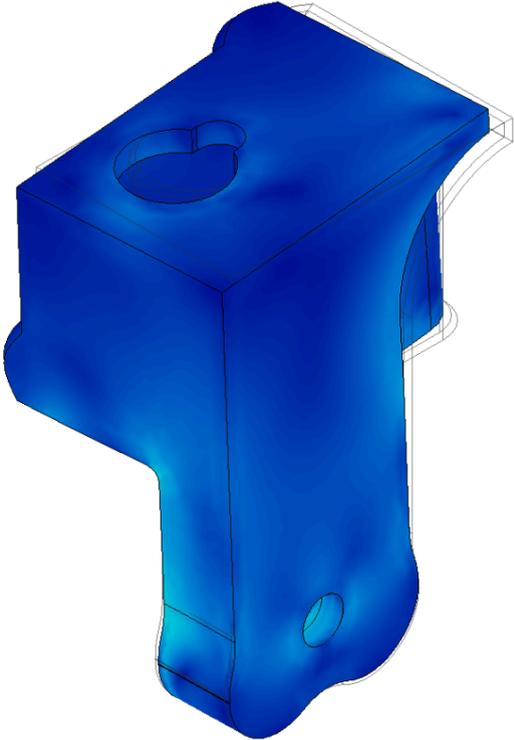
■ **Factor de seguridad**

- **Coficiente de seguridad (por cuerpo)**

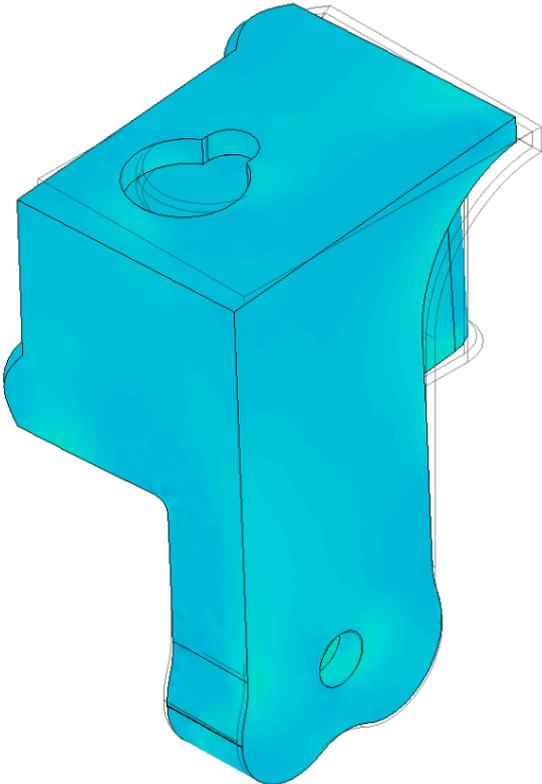
0.00  8.00



■ Estrés
• von Mises
[MPa] 0.00 2.324

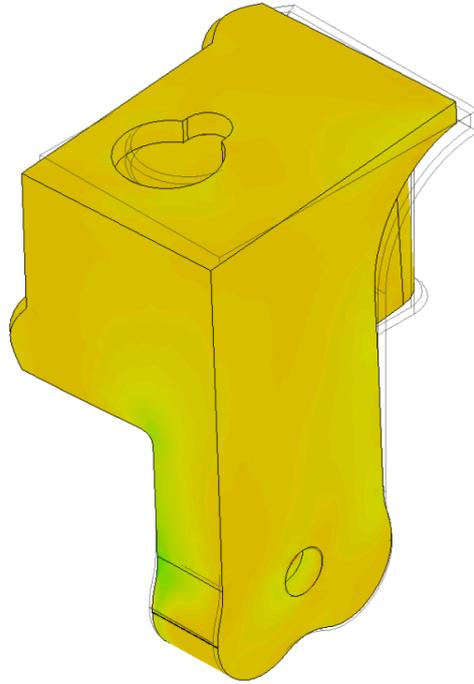


• Primera principal
[MPa] -0.362 1.666



• Tercera principal

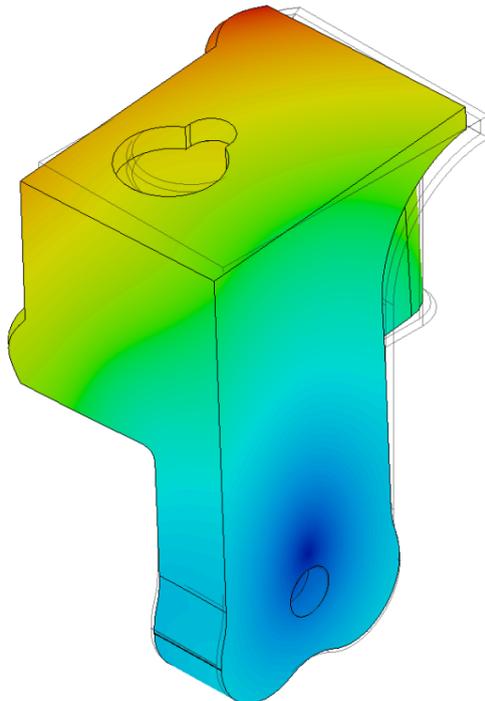
[MPa] -1.714 0.379



■ Desplazamiento

• Total

[mm] 0.00 0.08



● Cuerpo Pinza

Archivo analizado	Cuerpo Pinza
Versión	Autodesk Fusion (2.0.18961)
Fecha de creación	2024-05-13, 15:13:28
Autor	Óscar Gutiérrez Álvarez

○ Propiedades del informe

Título	Estudios
Autor	Óscar Gutiérrez Álvarez

- **Modelo de simulación 1:1**
 - **Estudio 1: tensión estática**
 - **Propiedades del estudio**

Tipo de estudio	Tensión estática
Fecha de la última modificación	2024-05-13, 09:34:20

- **Configuración**
 - **General**

Tolerancia de contacto	0.10 mm
Eliminar los modos de cuerpos rígidos	No

○ **Malla**

Tamaño medio de elemento (% del tamaño del modelo)	
Sólidos	10
Escala de tamaño de malla por pieza	No
Tamaño medio de elemento (valor absoluto)	-
Orden de elemento	Parabólico
Crear elementos de malla curva	Sí
Ángulo de giro máx. en curvas (grad.)	60
Relación de tamaño máx. de malla adyacente	1.5
Relación de aspecto máxima	10
Tamaño mínimo de elemento (% del tamaño medio)	20

- **Refinado de malla adaptable**

Número de pasos de refinado	0
Tolerancia de convergencia de resultados (%)	20
Parte de los elementos para refinar (%)	10
Resultados para la precisión de la línea base	Tensión de von Mises

- **Materiales**

Componente	Material	Factor de seguridad
Cuerpo Base	Plástico PLA	Límite de elasticidad

- **Plástico PLA**

Densidad	1.240E-06 kg / mm ³
Módulo de Young	3450.00 MPa
Coefficiente de Poisson	0.39
Límite de elasticidad	63.70 MPa
Resistencia máxima a tracción	58.50 MPa
Conductividad térmica	1.380E-04 W / (mm C)
Coefficiente de dilatación térmica	1.360E-04 / C
Calor específico	14998.00 J / (kg C)

- **Malla**

Tipo	Nodos	Elementos
Sólidos	15120	8849

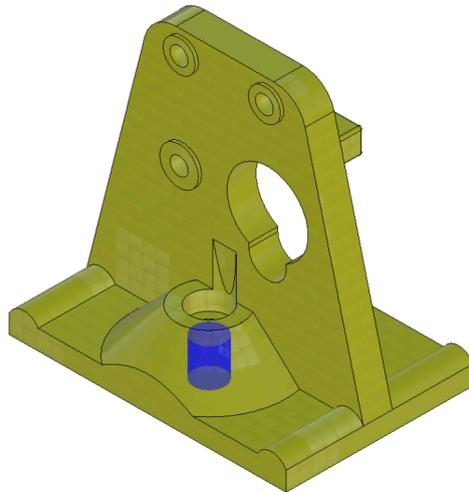
- **Caso de carga 1**

- **Restricciones**

- **Fijo 1**

Tipo	Fijo
Ux	Fijo
Uy	Fijo
Uz	Fijo

- **Entidades seleccionadas**

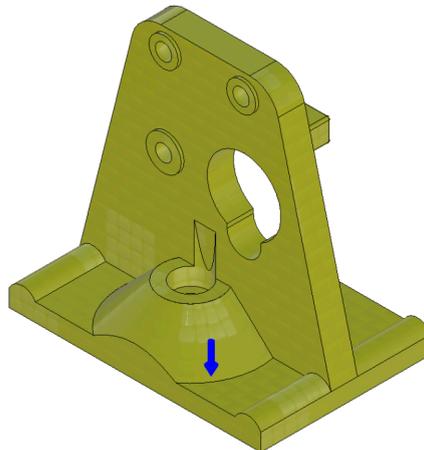


- **Cargas**

- **Gravedad**

Tipo	Gravedad
Magnitud	9.807 m / s ²
Valor X	0.00 m / s ²
Valor Y	-9.807 m / s ²
Valor Z	0.00 m / s ²

- **Entidades seleccionadas**

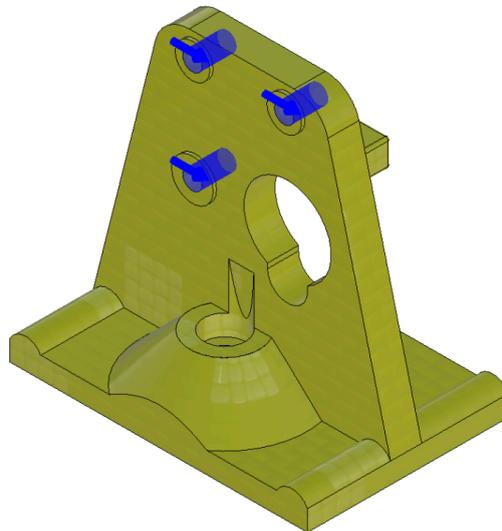


- **Fuerza 1**

Tipo	Fuerza
Valor X	1.17 N
Valor Y	0.00 N
Valor Z	0.00 N

Fuerza por entidad	No
--------------------	----

- **Entidades seleccionadas**



- **Resultados**

- **Resumen de resultados**

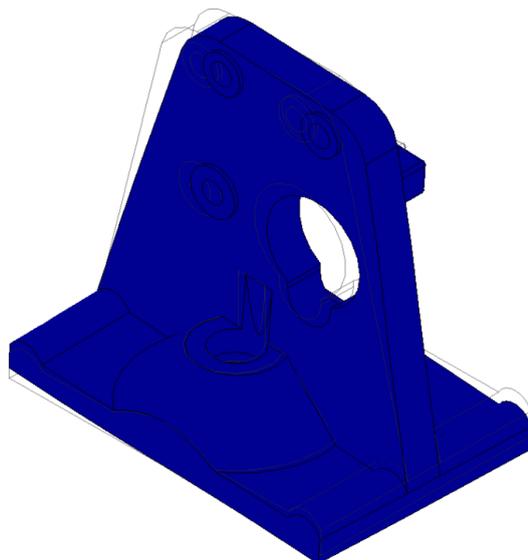
Nombre	Mínimo	Máximo
Factor de seguridad		
Coefficiente de seguridad (por cuerpo)	15.00	15.00
Estrés		
von Mises	6.920E-05 MPa	0.304 MPa
Primera principal	-0.207 MPa	0.389 MPa
Tercera principal	-0.486 MPa	0.138 MPa
Normal XX	-0.398 MPa	0.339 MPa
Normal YY	-0.27 MPa	0.183 MPa
Normal ZZ	-0.294 MPa	0.20 MPa
Corte XY	-0.114 MPa	0.125 MPa
Corte YZ	-0.055 MPa	0.055 MPa
Corte ZX	-0.126 MPa	0.126 MPa
Desplazamiento		
Total	0.00 mm	0.003 mm
X	-2.410E-04 mm	0.003 mm
Y	-0.002 mm	0.002 mm
Z	-2.375E-04 mm	2.647E-04 mm
Fuerza de reacción		
Total	0.00 N	0.506 N

X	-0.447 N	0.296 N
Y	-0.171 N	0.233 N
Z	-0.137 N	0.078 N
Deformación		
Equivalente	2.623E-08	1.630E-04
Primera principal	2.501E-08	1.388E-04
Tercera principal	-1.436E-04	-2.012E-08
Normal XX	-5.924E-05	5.565E-05
Normal YY	-2.617E-05	3.135E-05
Normal ZZ	-4.277E-05	3.246E-05
Corte XY	-9.209E-05	1.009E-04
Corte YZ	-4.459E-05	4.466E-05
Corte ZX	-1.015E-04	1.019E-04
Fuerza de contacto		
Total	0.00 N	0.00 N
X	0.00 N	0.00 N
Y	0.00 N	0.00 N
Z	0.00 N	0.00 N

■ **Factor de seguridad**

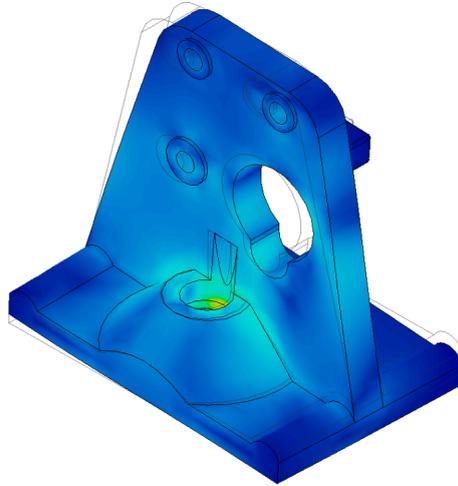
- **Coficiente de seguridad (por cuerpo)**

0.00  8.00



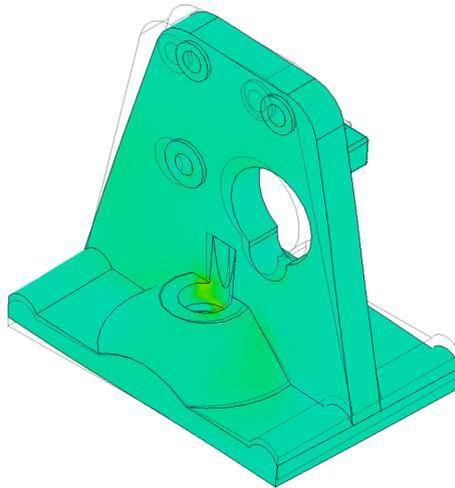
- Estrés
- von Mises

[MPa] 0.00  0.304



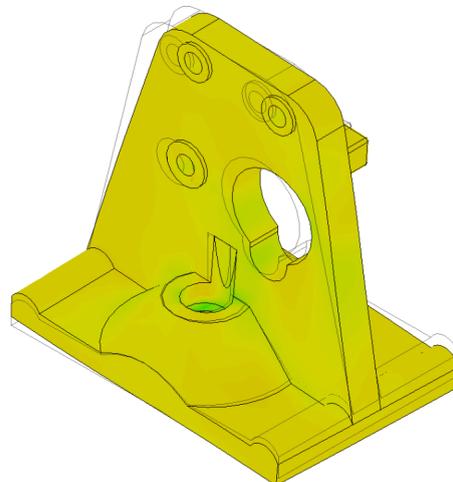
- Primera principal

[MPa] -0.207  0.389



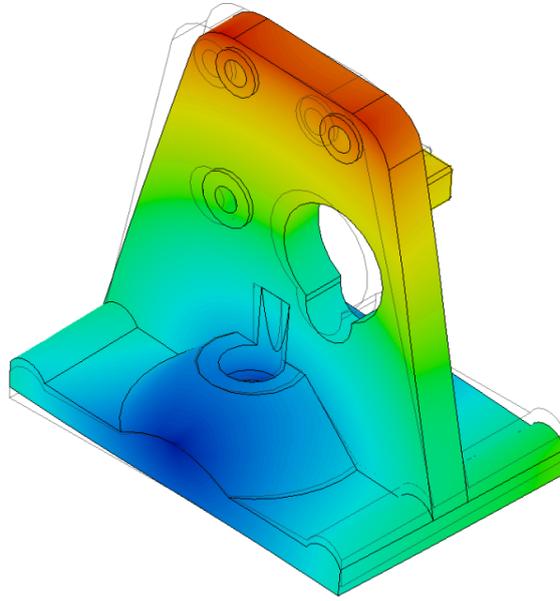
- Tercera principal

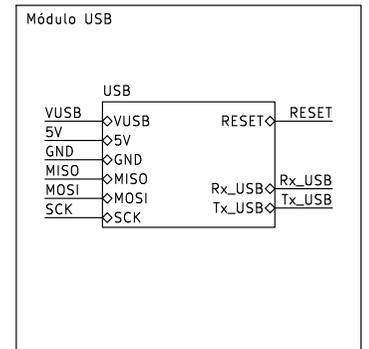
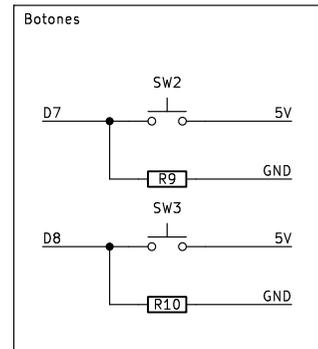
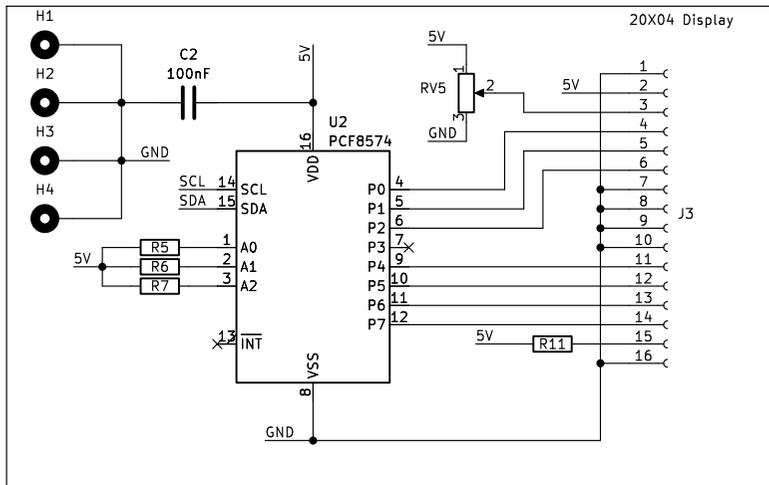
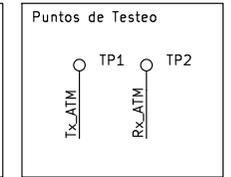
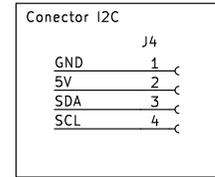
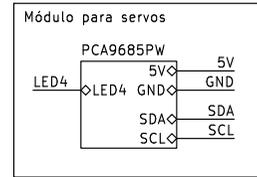
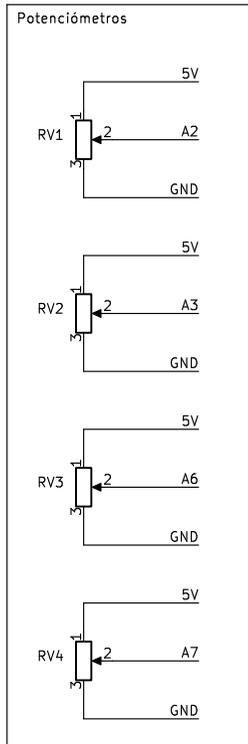
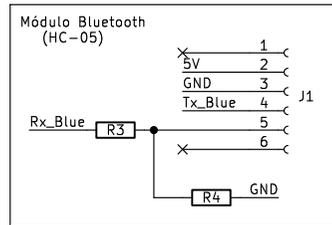
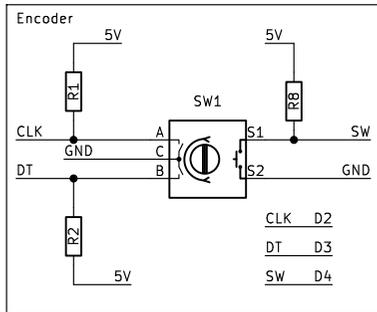
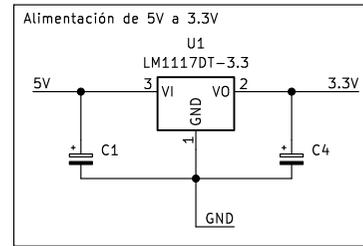
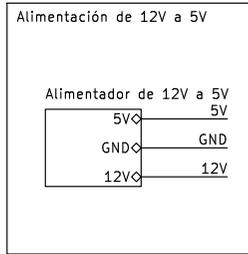
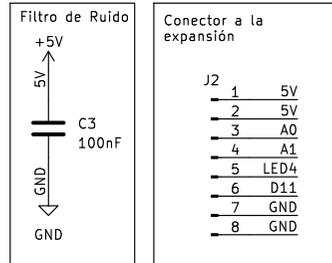
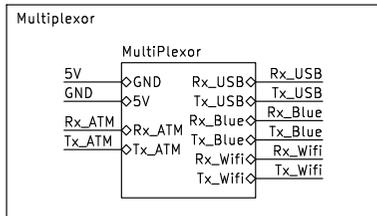
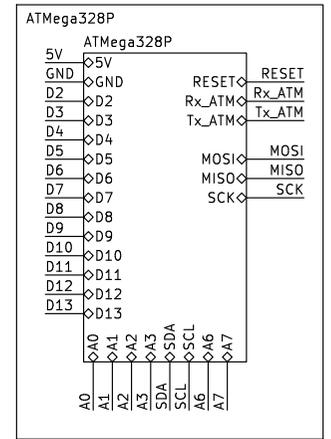
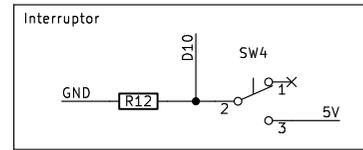
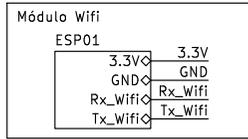
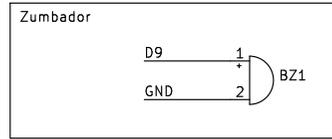
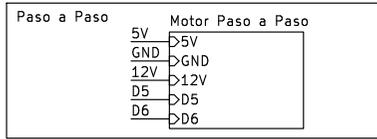
[MPa] -0.486  0.138

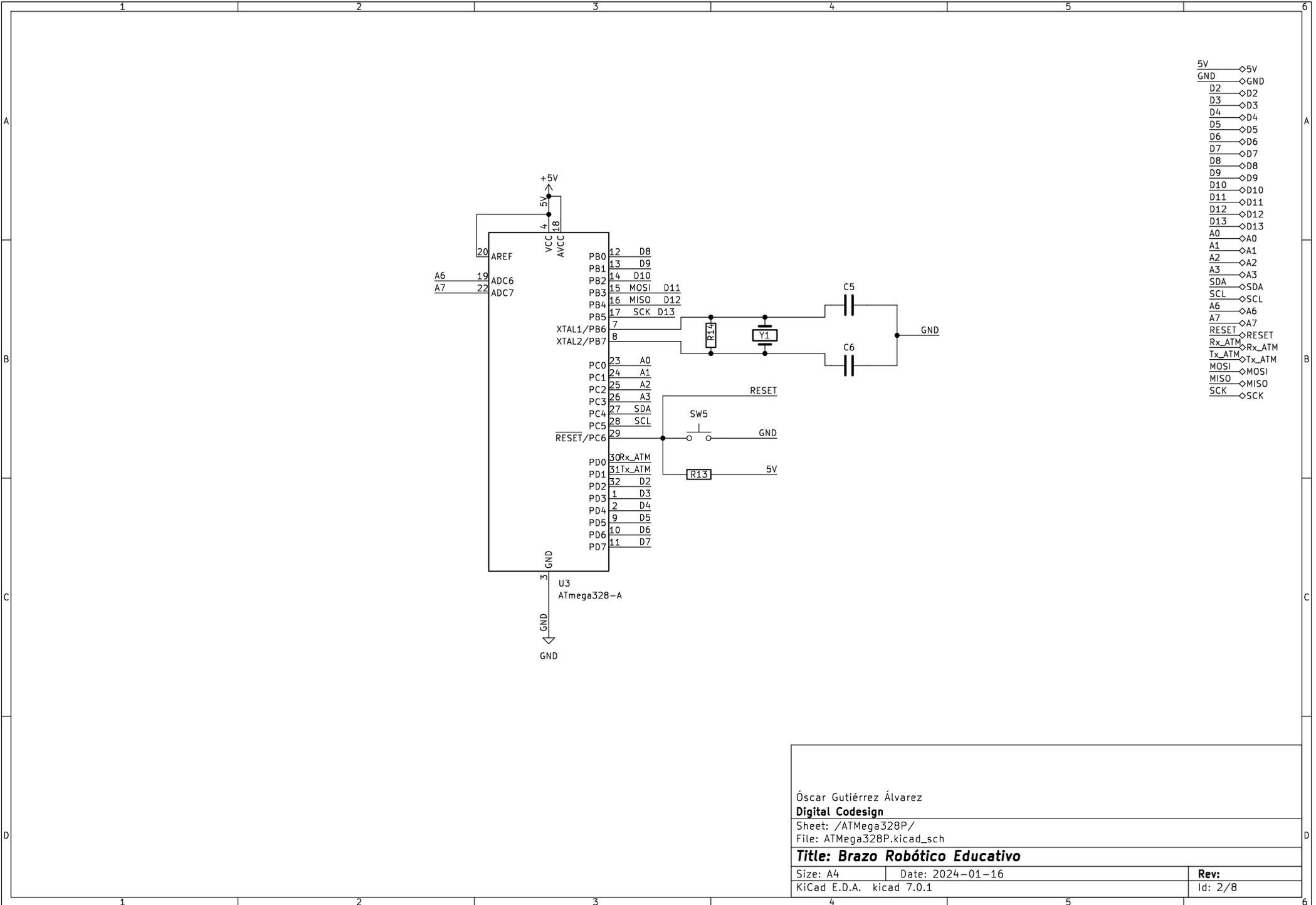


■ Desplazamiento
● Total

[mm] 0.00  0.003

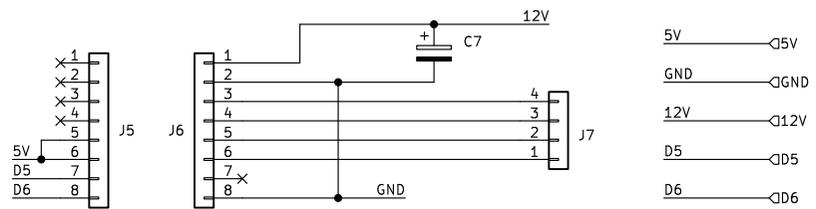






Óscar Gutiérrez Álvarez
Digital Codesign
 Sheet: /ATmega328P/
 File: ATmega328P.kicad_sch
Title: Brazo Robótico Educativo

Size: A4	Date: 2024-01-16	Rev:
KiCad E.D.A. kicad 7.0.1		Id: 2/8



Óscar Gutiérrez Álvarez
Digital Codesign

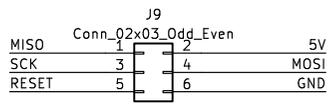
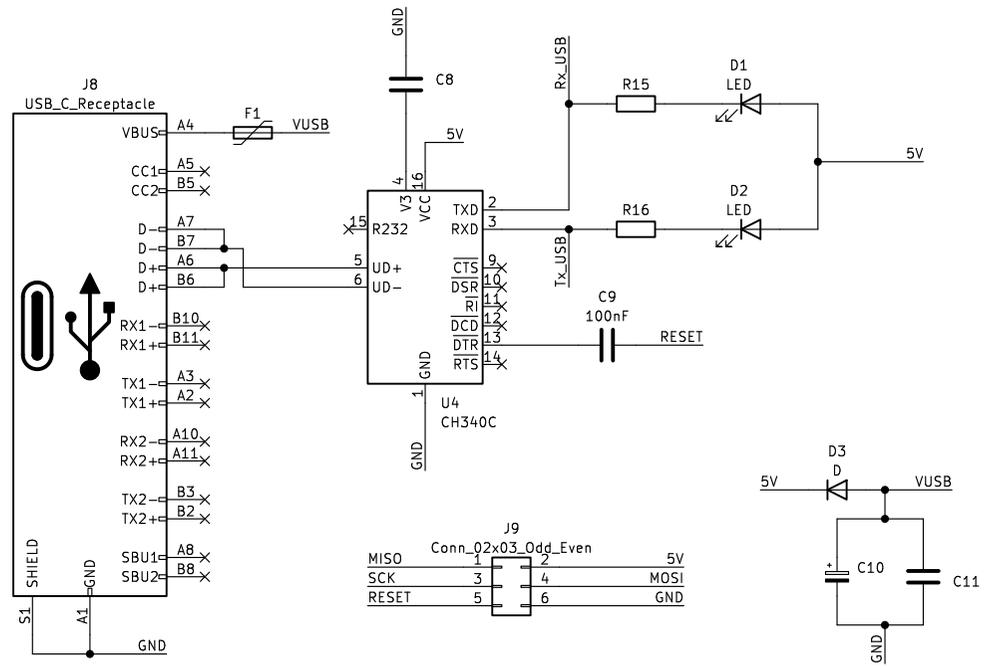
Sheet: /Motor Paso a Paso/
 File: MotorPasoPaso.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

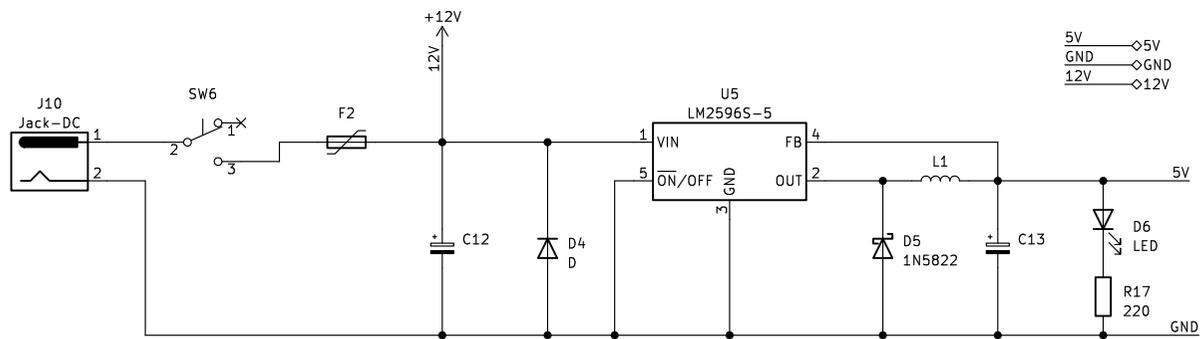
KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 3/8



- VUSB —◇ VUSB
- 5V —◇ 5V
- GND —◇ GND
- Rx_USB —◇ Rx_USB
- Tx_USB —◇ Tx_USB
- RESET —◇ RESET
- MISO —◇ MISO
- MOSI —◇ MOSI
- SCK —◇ SCK

Óscar Gutiérrez Álvarez	
Digital Codesign	
Sheet: /USB/	
File: USB.kicad_sch	
Title: Brazo Robótico Educativo	
Size: A4	Date: 2024-01-16
KiCad E.D.A. kicad 7.0.1	Rev: 4/8



Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /Alimentador de 12V a 5V/

File: JACK12Va5V.kicad_sch

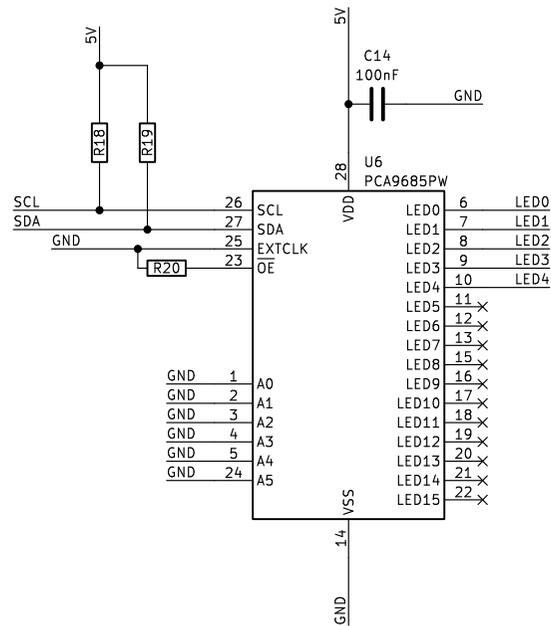
Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

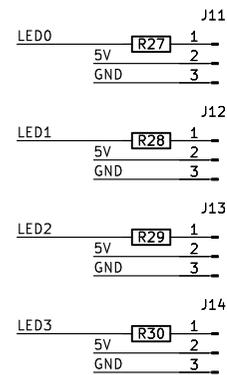
KiCad E.D.A. kicad 7.0.1

Rev:

Id: 5/8



5V ◇ 5V
 GND ◇ GND
 SDA ◇ SDA
 SCL ◇ SCL
 LED4 ◇ LED4



Óscar Gutiérrez Álvarez
Digital Codesign

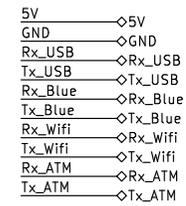
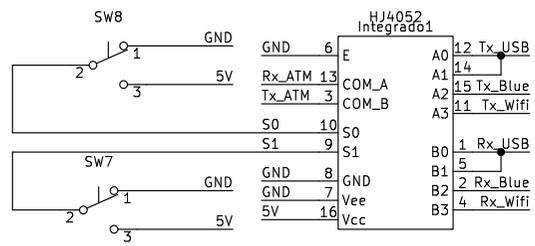
Sheet: /PCA9685PW/
 File: PCA9685PW.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 6/8



Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /MultiPlexor/

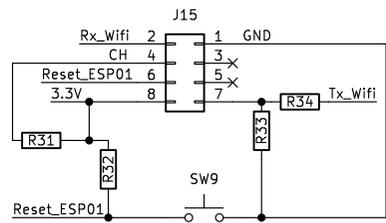
File: MultiFlexor.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
Id: 7/8



- 3.3V ◇ 3.3V
- GND ◇ GND
- Rx_Wifi ◇ Rx_Wifi
- Tx_Wifi ◇ Tx_Wifi

Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /ESP01/

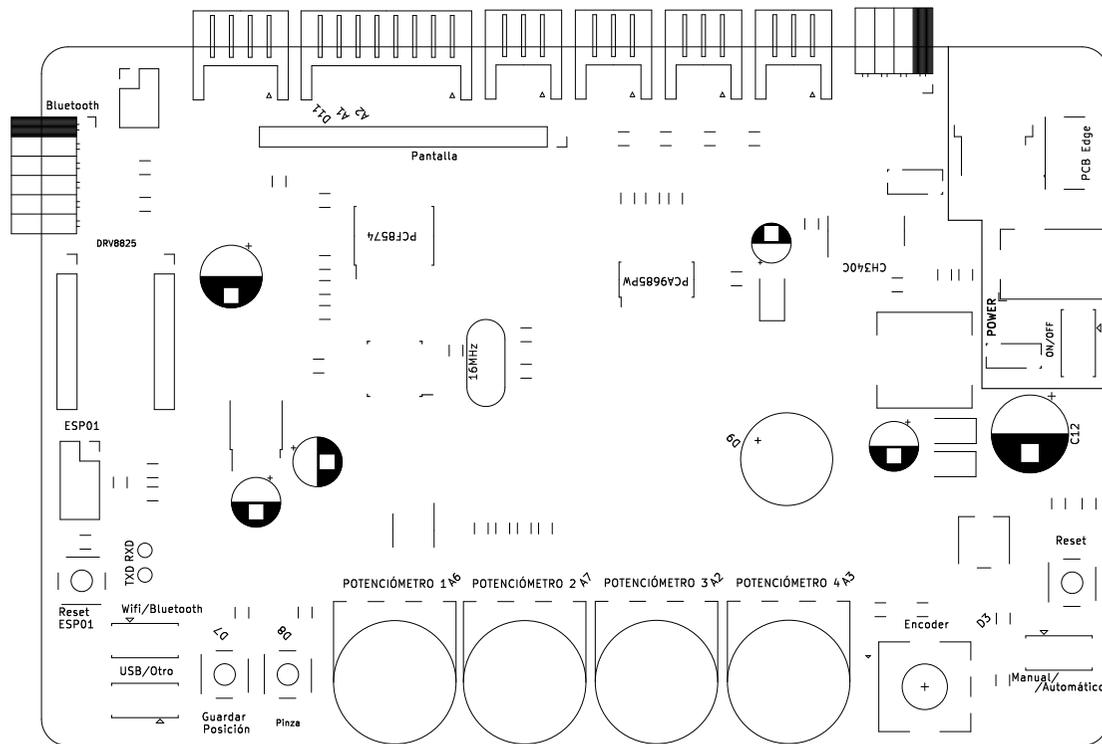
File: ESP01.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
Id: 8/8



Óscar Gutiérrez Álvarez
Digital Codesign

Sheet:
File: Proyecto BRE V1.4.kicad_pcb

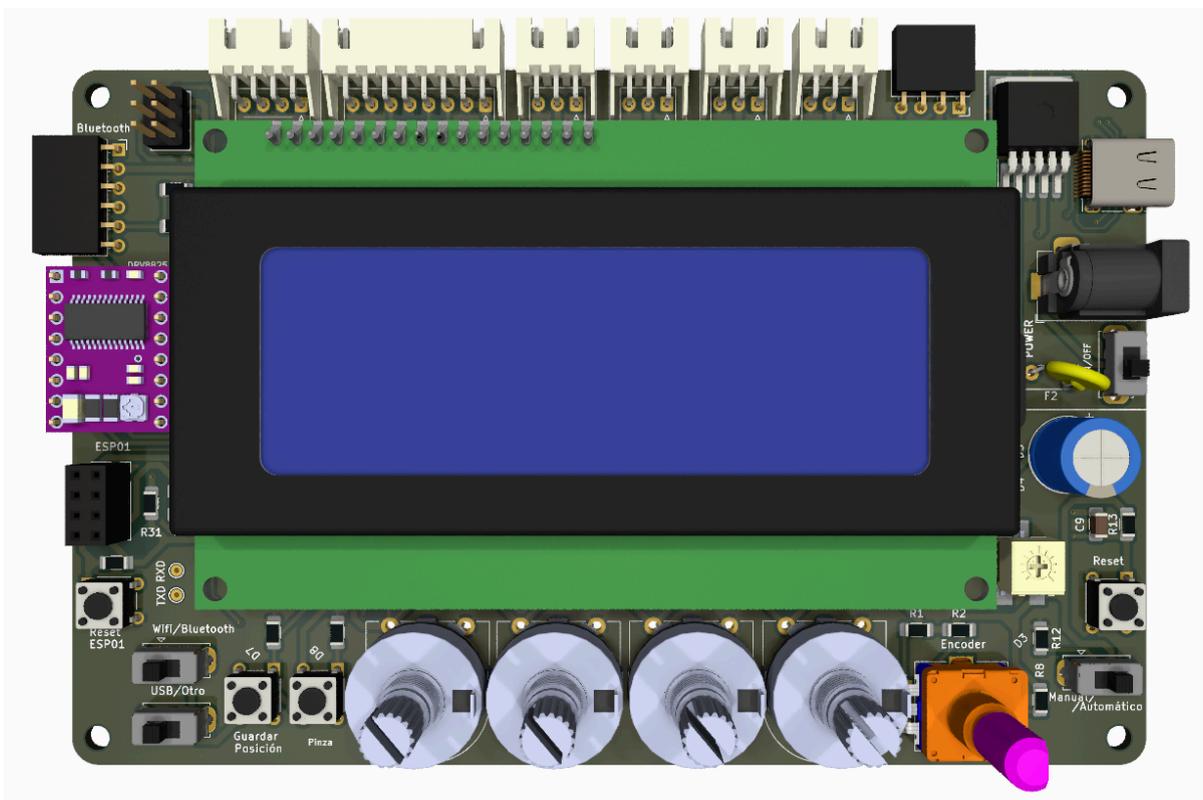
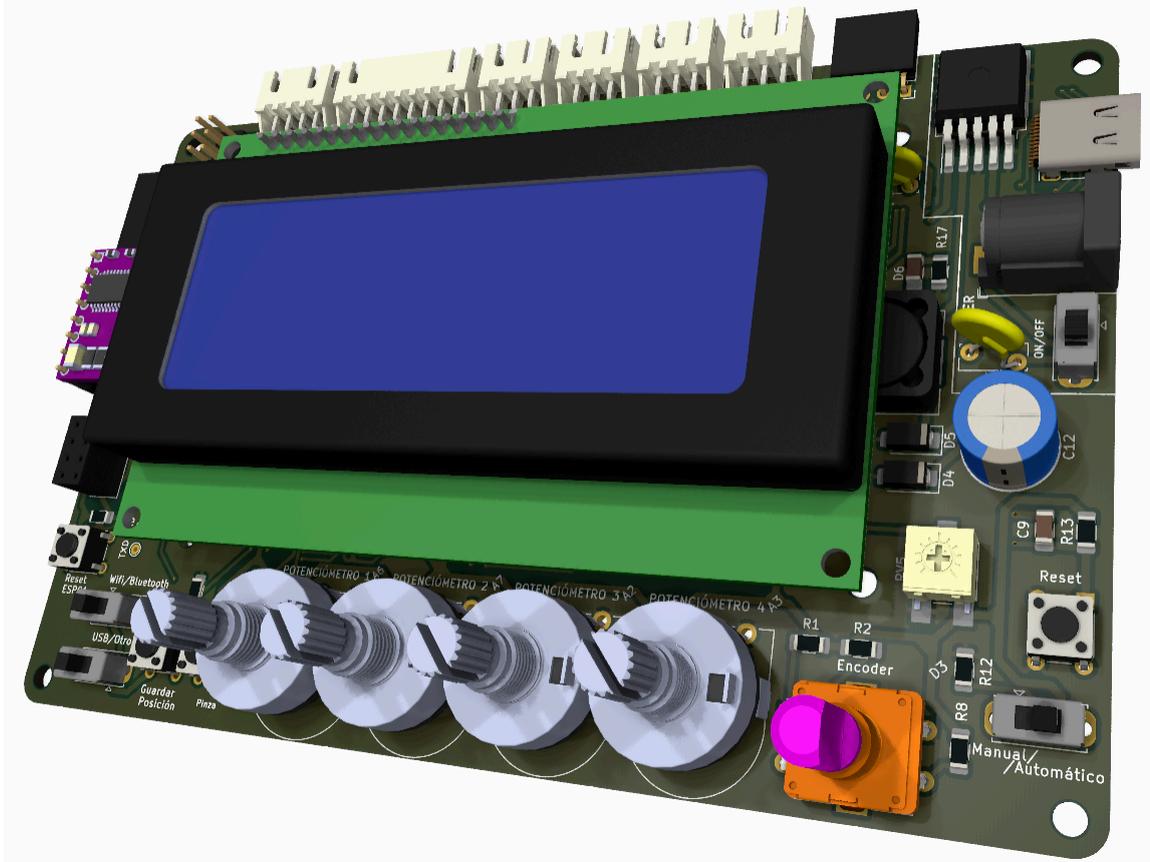
Title: Brazo Robótico Educativo

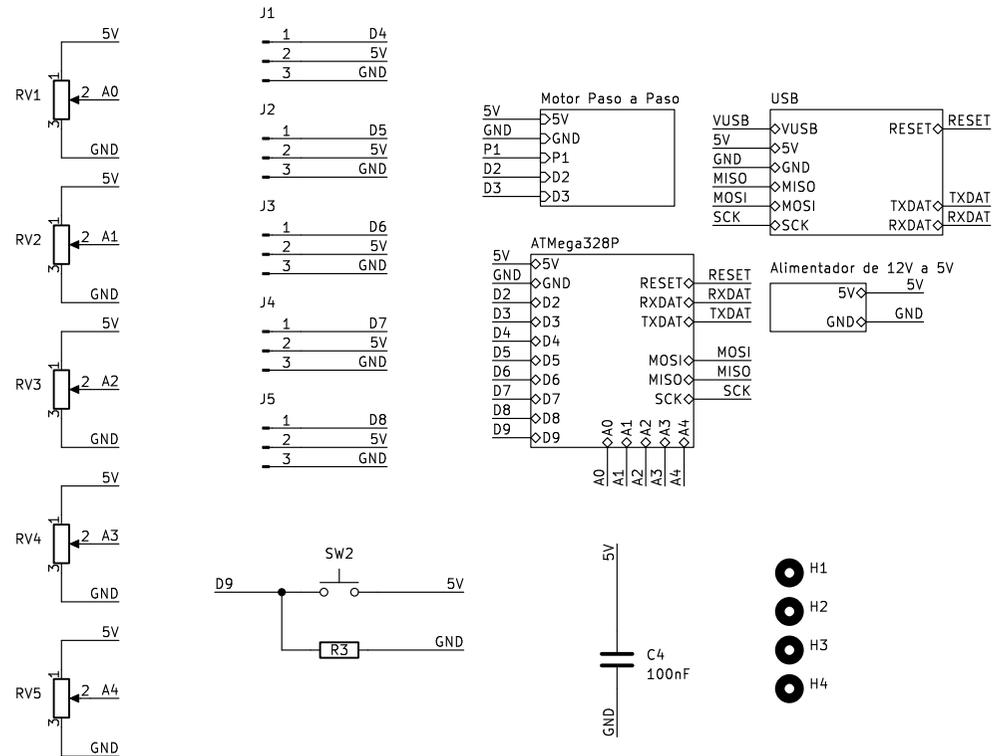
Size: A4 Date: 2024-05-05

KiCad E.D.A. kicad 7.0.1

Rev:
Id: 1/1

VISTAS 3D DE LA PCB PRINCIPAL DEL BRAZO ROBÓTICO (V1.4)





Óscar Gutiérrez Álvarez
Digital Codesign

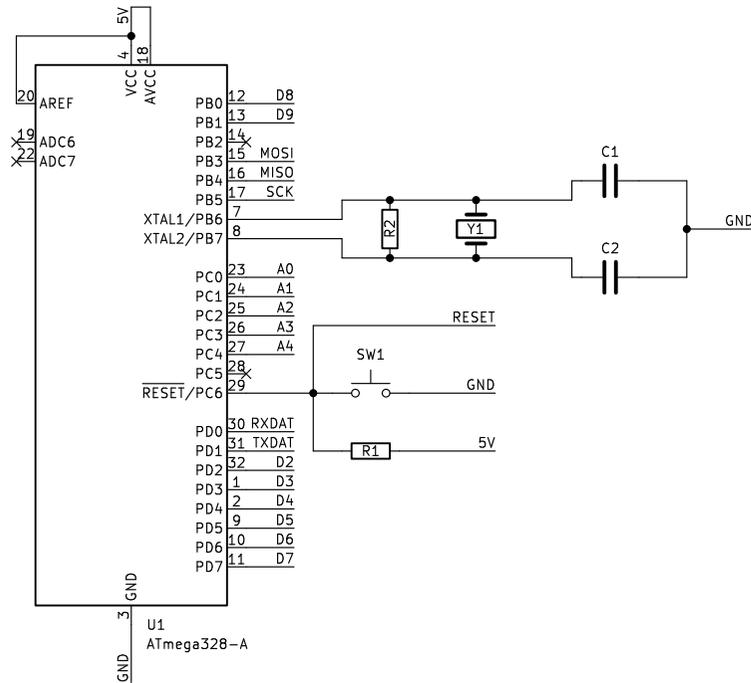
Sheet: /
 File: Proyecto BRE.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 1/5



- 5V ◇ 5V
- GND ◇ GND
- D2 ◇ D2
- D3 ◇ D3
- D4 ◇ D4
- D5 ◇ D5
- D6 ◇ D6
- D7 ◇ D7
- D8 ◇ D8
- D9 ◇ D9
- A0 ◇ A0
- A1 ◇ A1
- A2 ◇ A2
- A3 ◇ A3
- A4 ◇ A4
- RESET ◇ RESET
- RXDAT ◇ RXDAT
- TXDAT ◇ TXDAT
- MOSI ◇ MOSI
- MISO ◇ MISO
- SCK ◇ SCK

Óscar Gutiérrez Álvarez
Digital Codesign

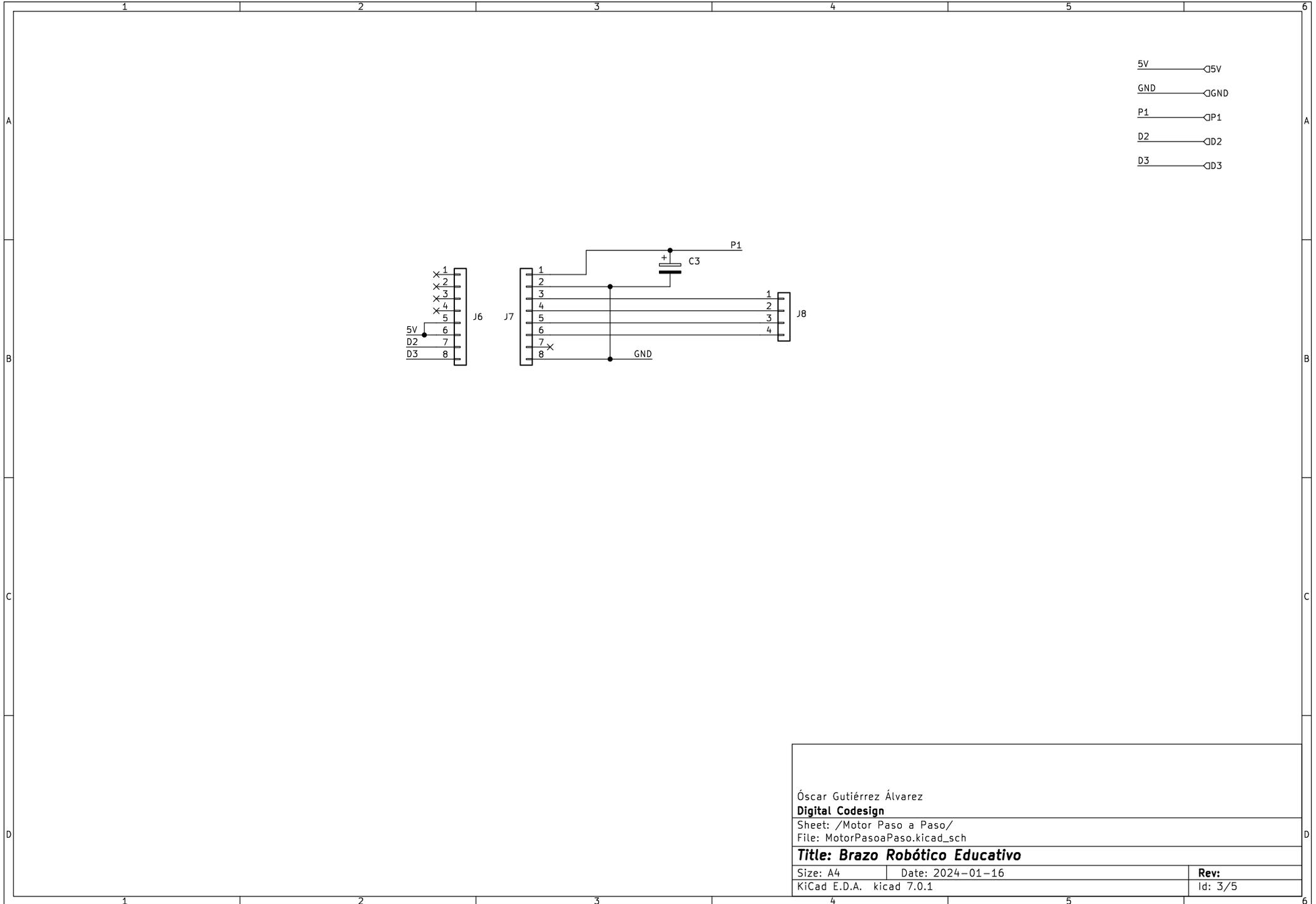
Sheet: /ATmega328P/
 File: ATmega328P.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

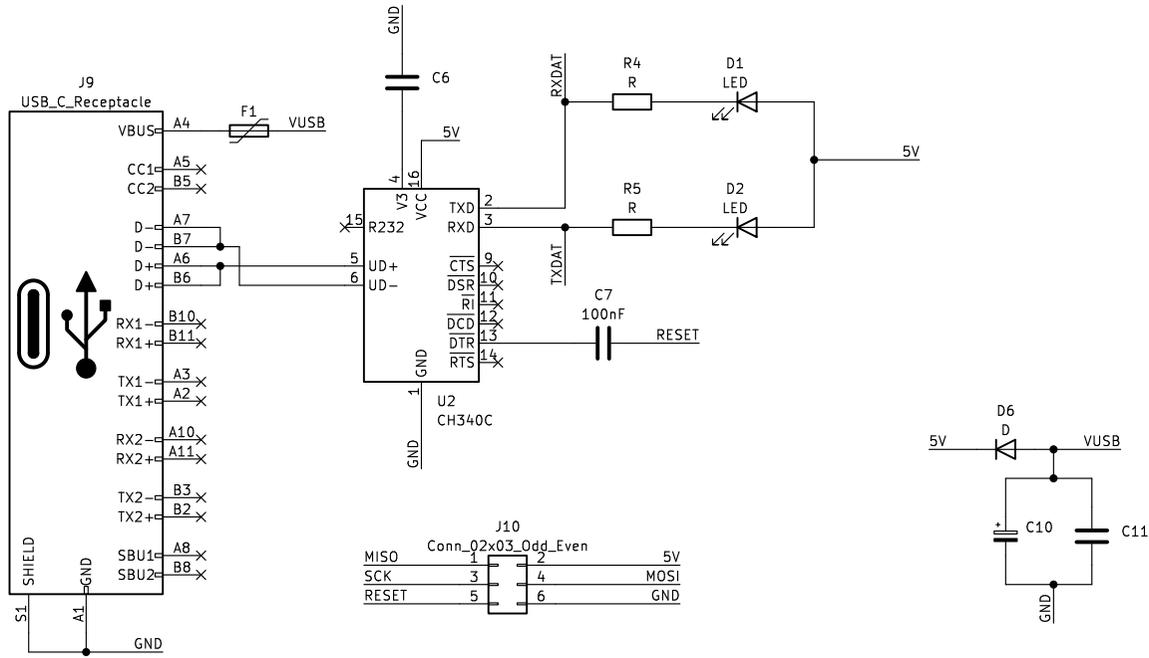
KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 2/5

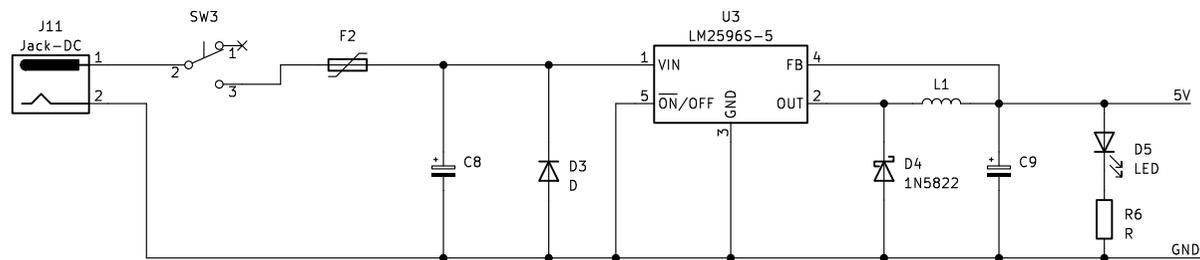


Óscar Gutiérrez Álvarez	
Digital Codesign	
Sheet: /Motor Paso a Paso/	
File: MotorPasoPaso.kicad_sch	
Title: Brazo Robótico Educativo	
Size: A4	Date: 2024-01-16
KiCad E.D.A. kicad 7.0.1	Rev: 3/5

- ◇— 5V
- ◇— GND
- ◇— RXDAT
- ◇— TXDAT
- ◇— RESET
- ◇— MISO
- ◇— MOSI
- ◇— SCK



Óscar Gutiérrez Álvarez	
Digital Codesign	
Sheet: /USB/	
File: USB.kicad_sch	
Title: Brazo Robótico Educativo	
Size: A4	Date: 2024-01-16
KiCad E.D.A. kicad 7.0.1	Rev: 4/5



5V \diamond 5V
 GND \diamond GND

Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /Alimentador de 12V a 5V/

File: JACK12Va5V.kicad_sch

Title: Brazo Robótico Educativo

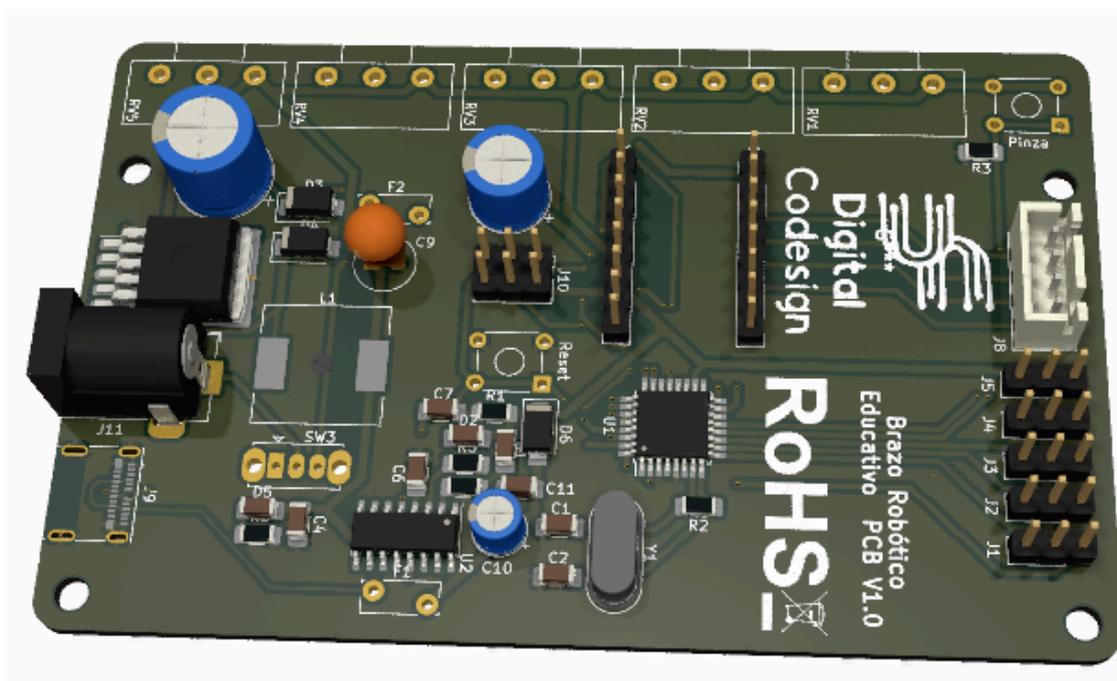
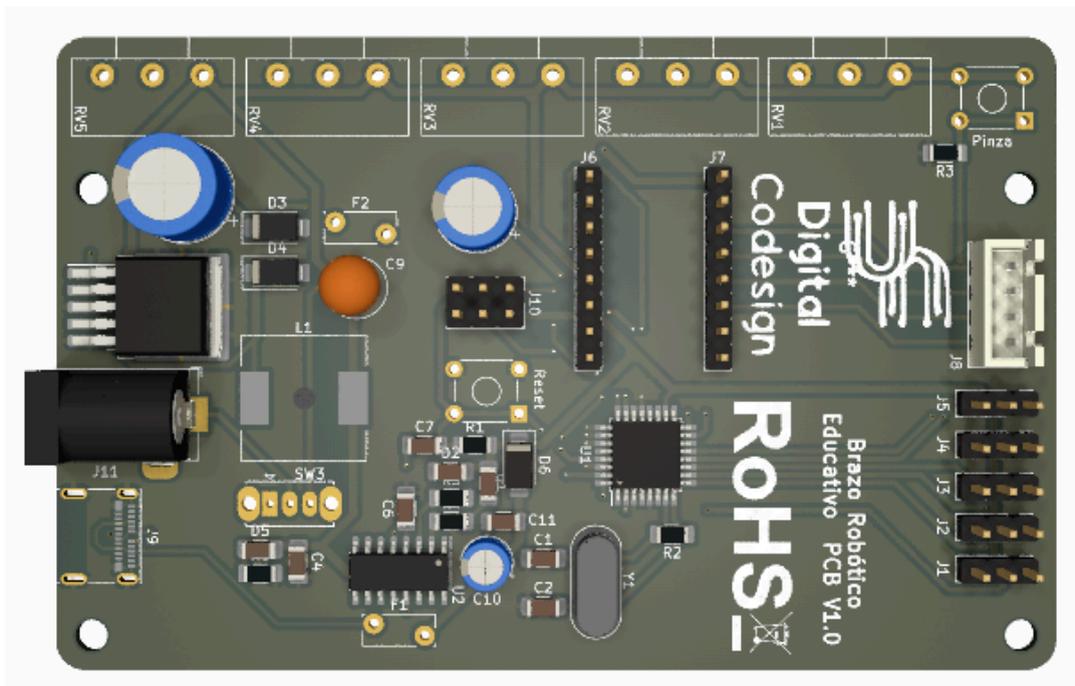
Size: A4 Date: 2024-01-16

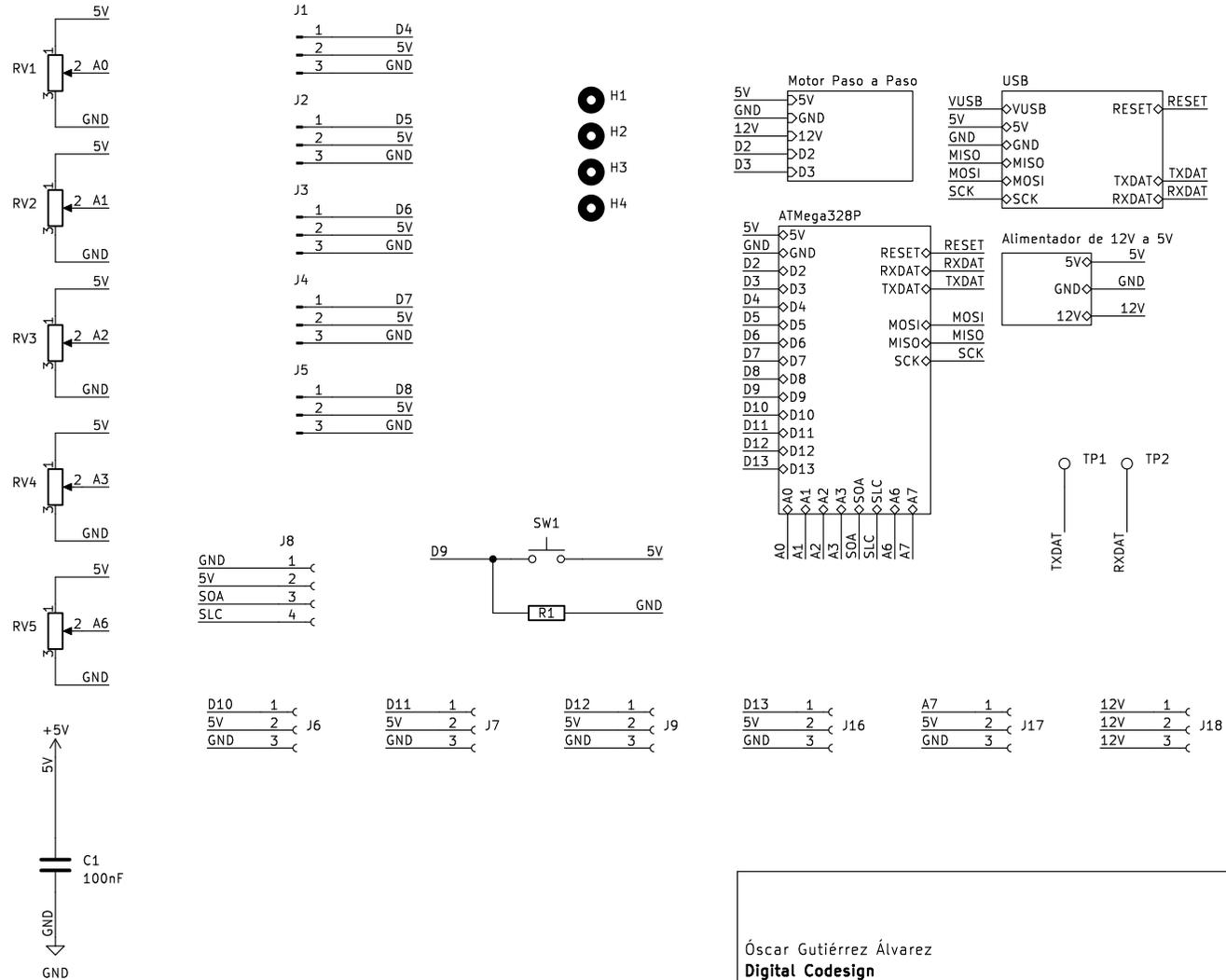
KiCad E.D.A. kicad 7.0.1

Rev:

Id: 5/5

VISTAS 3D DE LA PCB PRINCIPAL DEL BRAZO ROBÓTICO (V1.1)





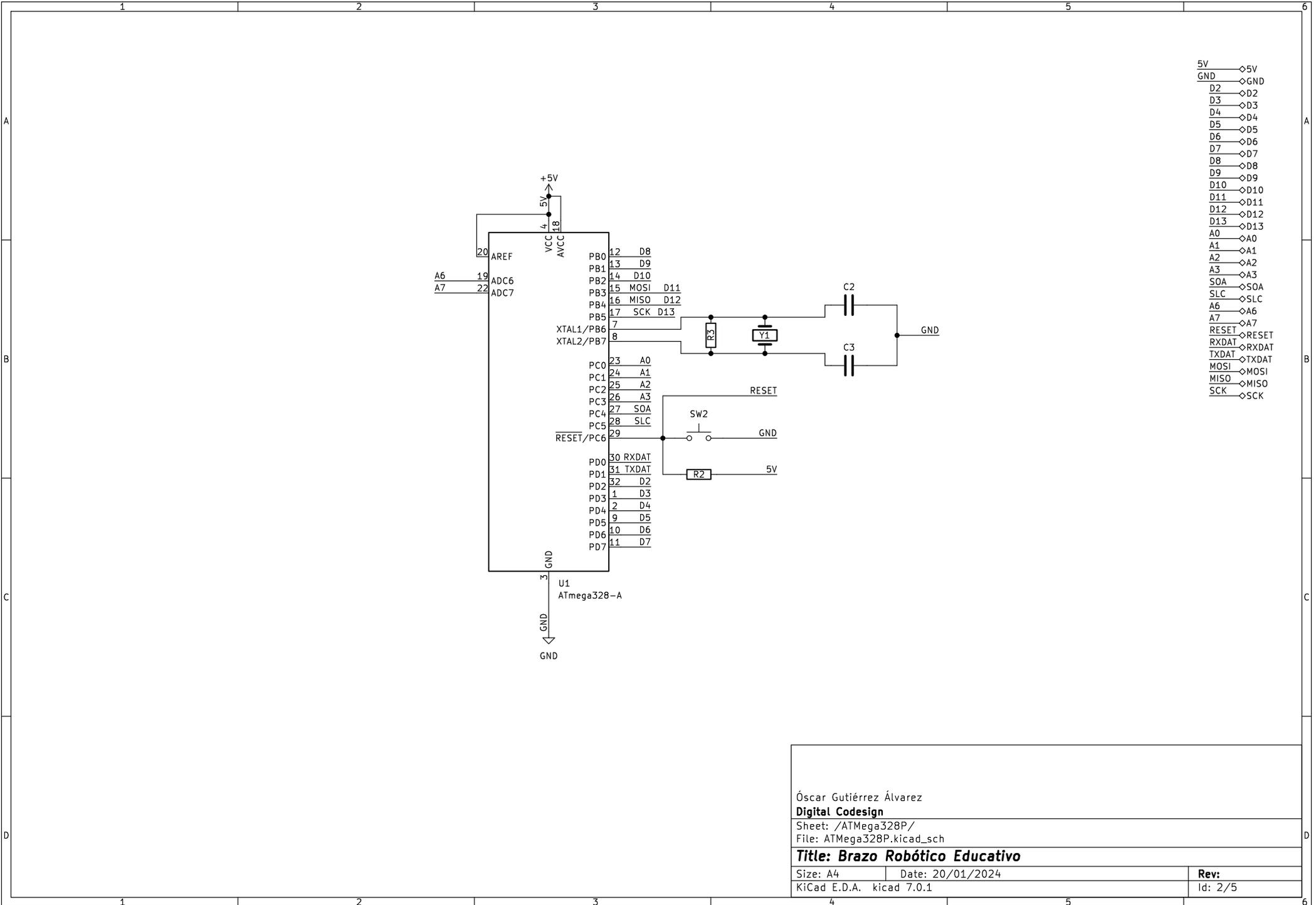
Óscar Gutiérrez Álvarez
Digital Codesign

Sheet: /
 File: Proyecto BRE V1.2.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 20/01/2024
 KiCad E.D.A. kicad 7.0.1

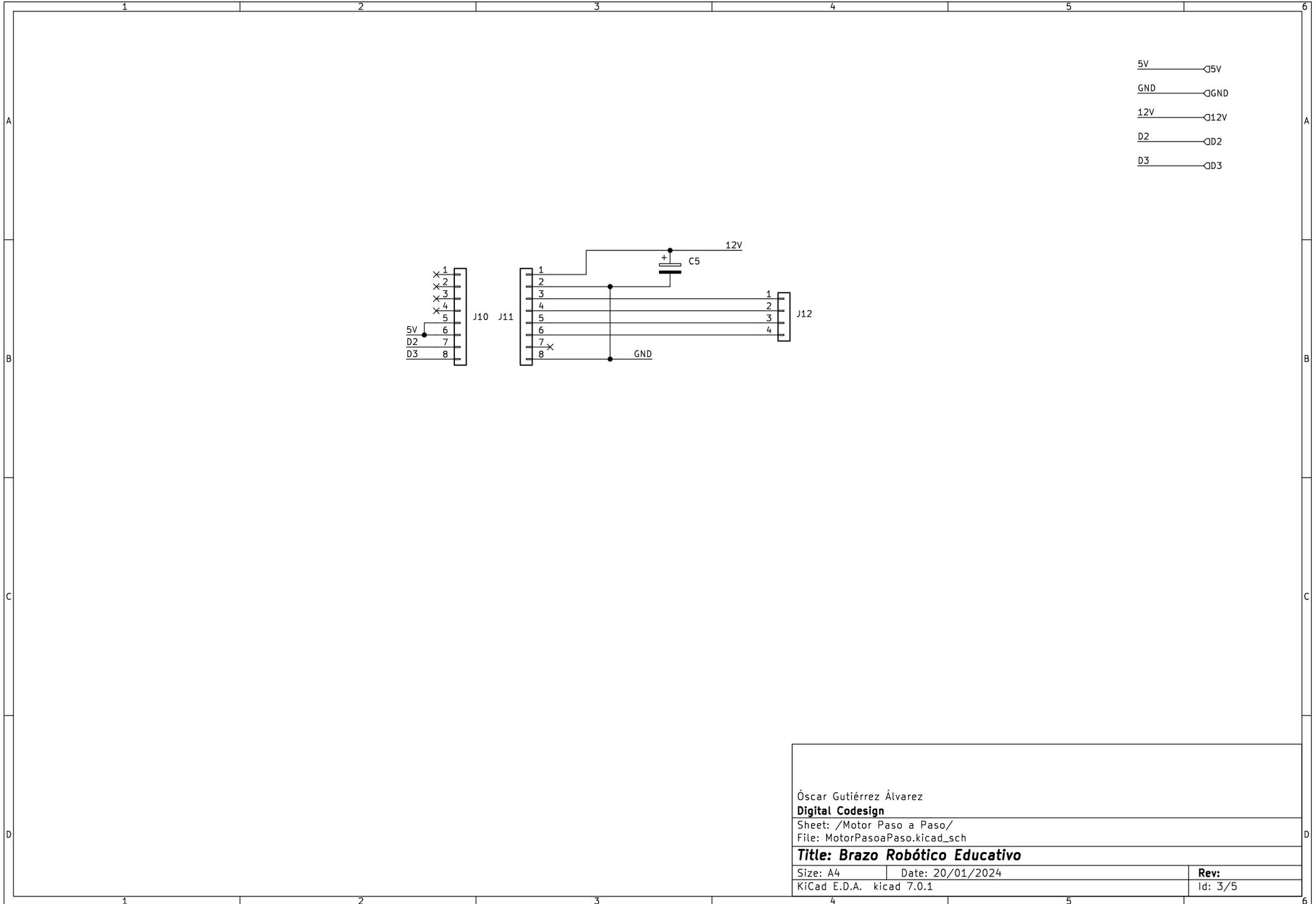
Rev:
 Id: 1/5



Óscar Gutiérrez Álvarez
Digital Codesign
 Sheet: /ATmega328P/
 File: ATmega328P.kicad_sch

Title: Brazo Robótico Educativo

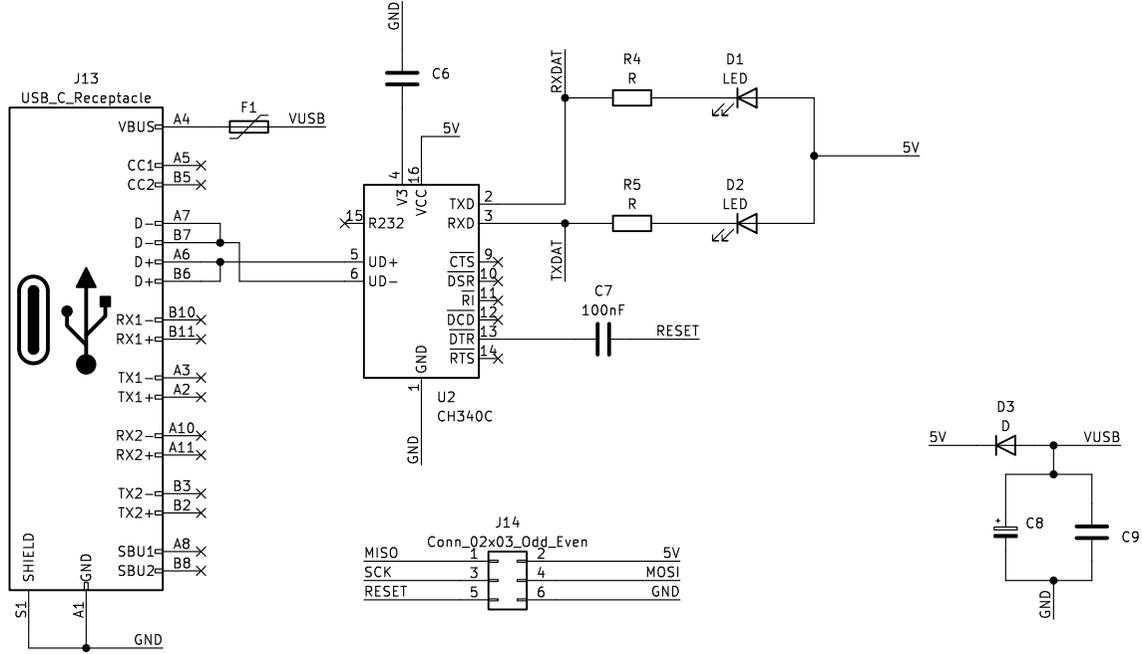
Size: A4	Date: 20/01/2024	Rev:
KiCad E.D.A. kicad 7.0.1		Id: 2/5



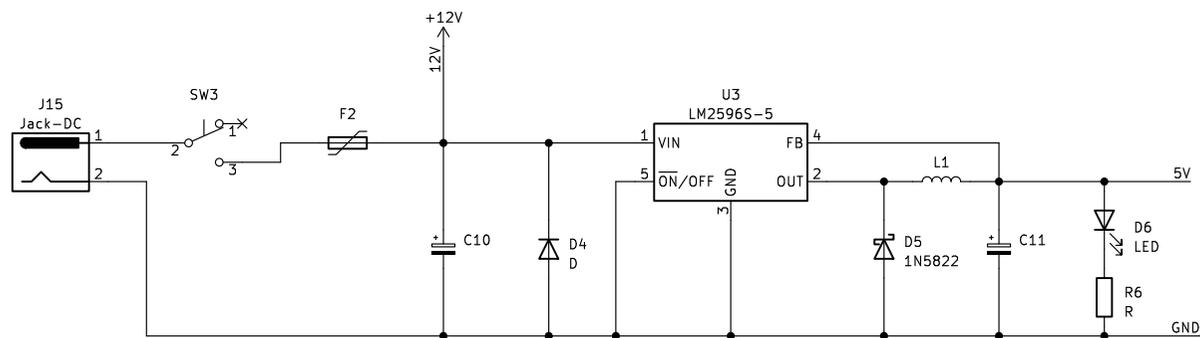
- 5V ———— ○D5V
- GND ———— ○GND
- 12V ———— ○D12V
- D2 ———— ○D2
- D3 ———— ○D3

Óscar Gutiérrez Álvarez	
Digital Codesign	
Sheet: /Motor Paso a Paso/	
File: MotorPasoPaso.kicad_sch	
Title: Brazo Robótico Educativo	
Size: A4	Date: 20/01/2024
KiCad E.D.A. kicad 7.0.1	Rev: 3/5

- VUSB — VUSB
- 5V — 5V
- GND — GND
- RXDAT — RXDAT
- TXDAT — TXDAT
- RESET — RESET
- MISO — MISO
- MOSI — MOSI
- SCK — SCK



Óscar Gutiérrez Álvarez	
Digital Codesign	
Sheet: /USB/	
File: USB.kicad_sch	
Title: Brazo Robótico Educativo	
Size: A4	Date: 20/01/2024
KiCad E.D.A. kicad 7.0.1	Rev: Id: 4/5



5V \diamond 5V
 GND \diamond GND
 12V \diamond 12V

Óscar Gutiérrez Álvarez
Digital Codesign

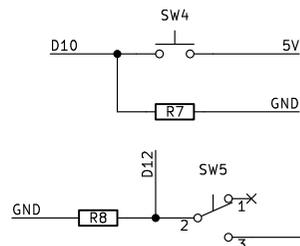
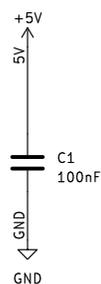
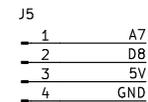
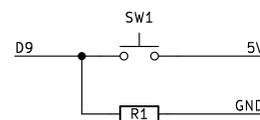
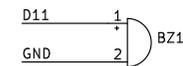
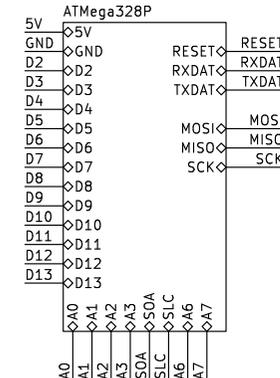
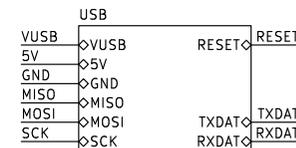
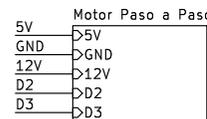
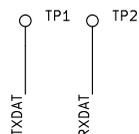
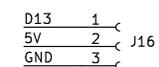
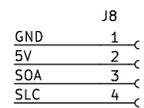
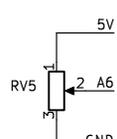
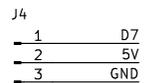
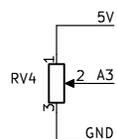
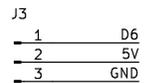
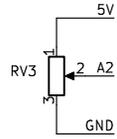
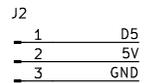
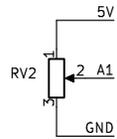
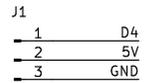
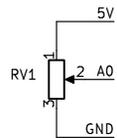
Sheet: /Alimentador de 12V a 5V/
 File: JACK12Va5V.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 20/01/2024

KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 5/5



Óscar Gutiérrez Álvarez
Digital Codesign

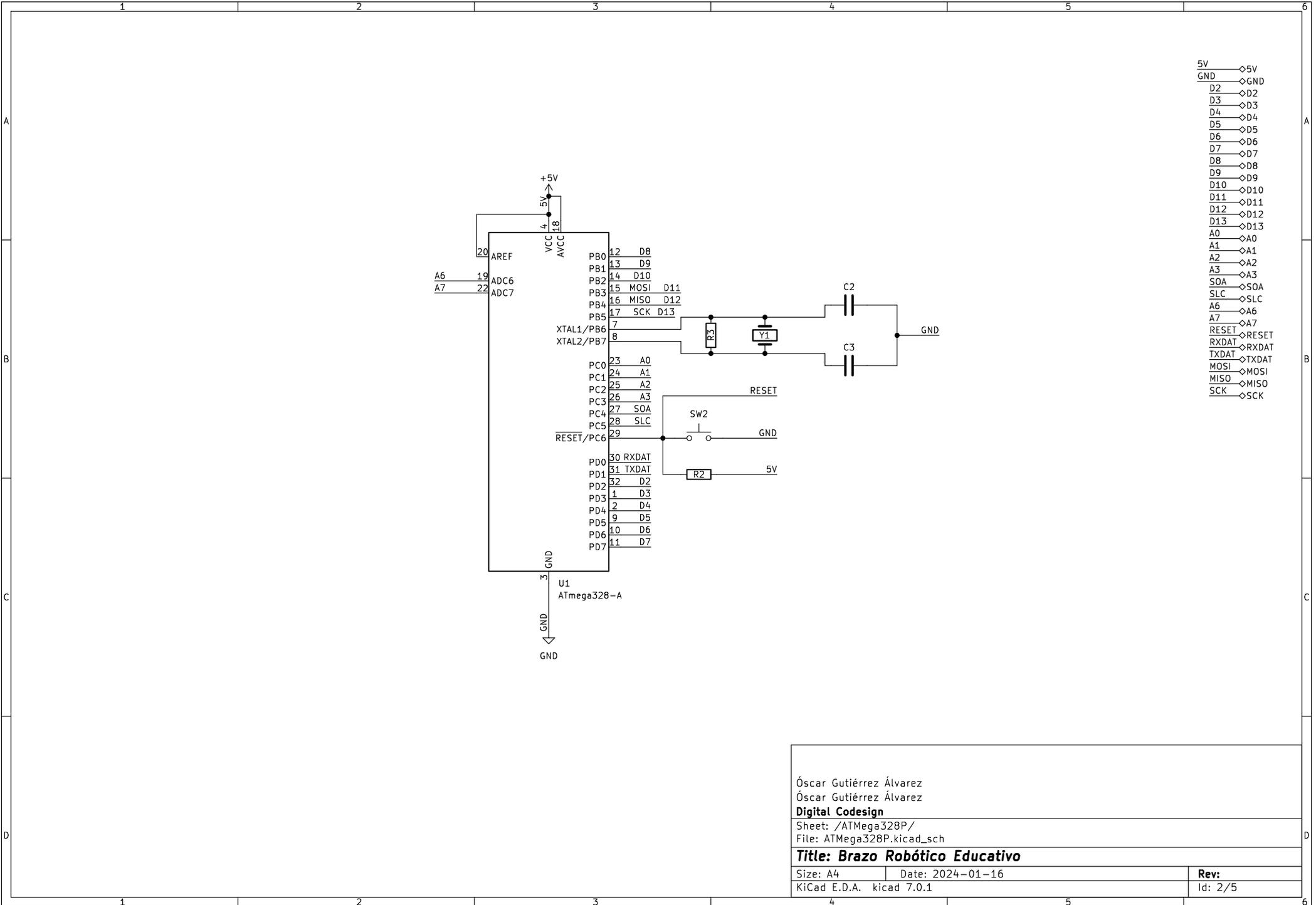
Sheet: /
 File: Proyecto BRE V1.3.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 1/5



Óscar Gutiérrez Álvarez
 Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /ATMega328P/
 File: ATMega328P.kicad_sch

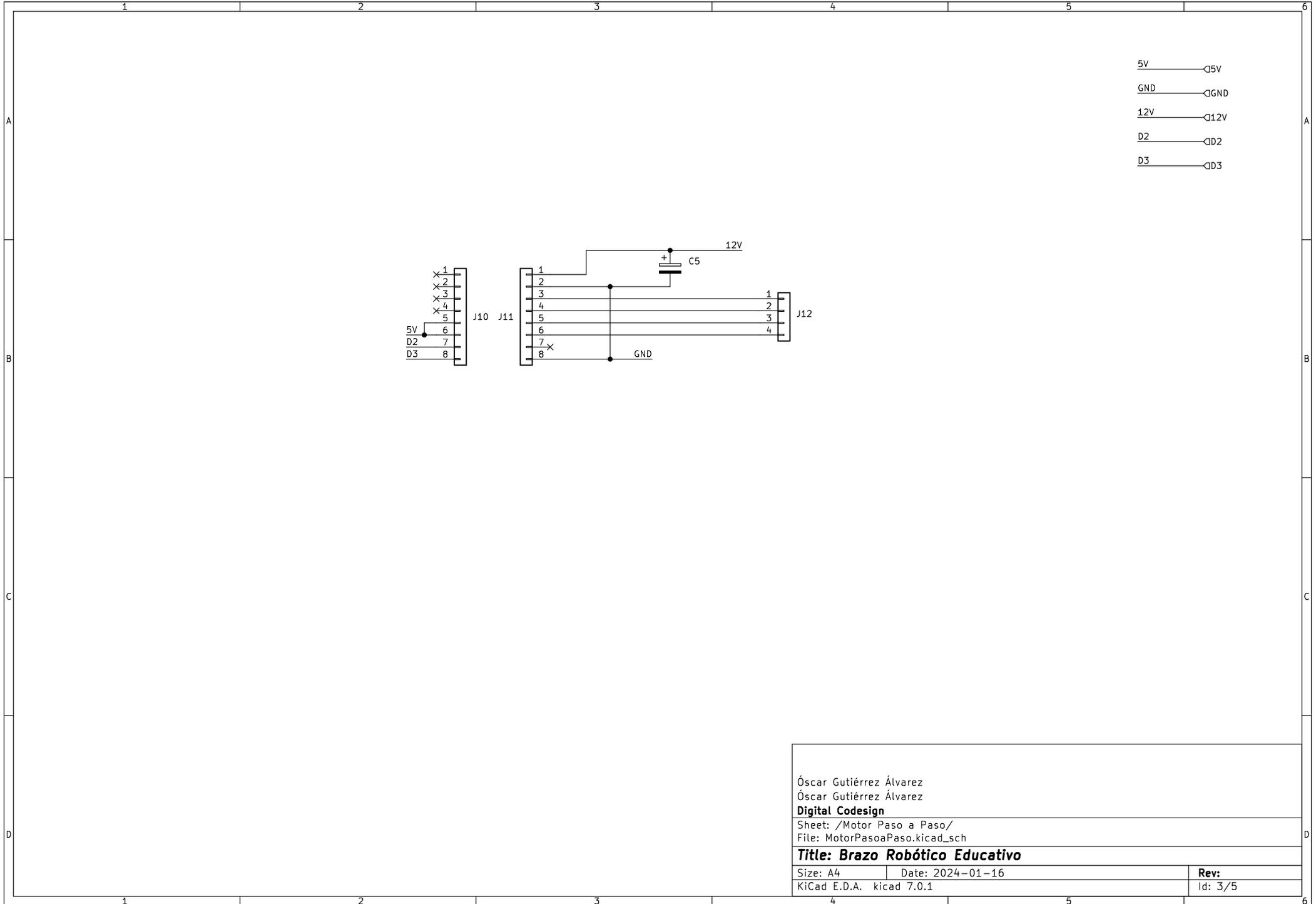
Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:

Id: 2/5



- 5V ———— ○D5V
- GND ———— ○GND
- 12V ———— ○D12V
- D2 ———— ○D2
- D3 ———— ○D3

Óscar Gutiérrez Álvarez
 Óscar Gutiérrez Álvarez

Digital Codesign

Sheet: /Motor Paso a Paso/
 File: MotorPasoPaso.kicad_sch

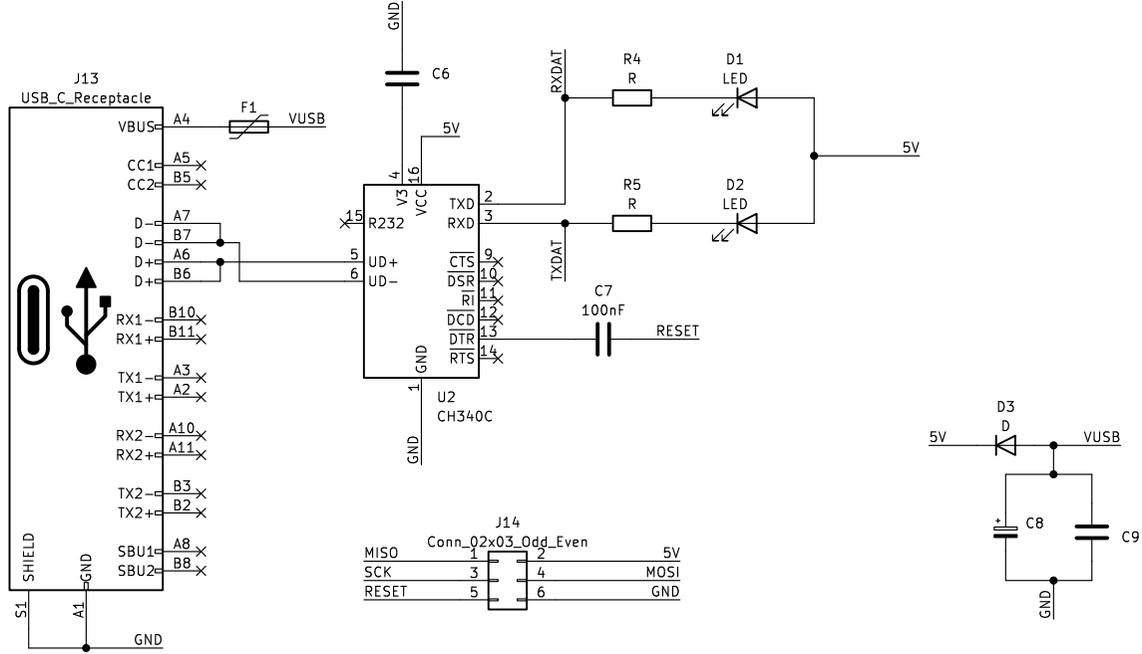
Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev: 3/5

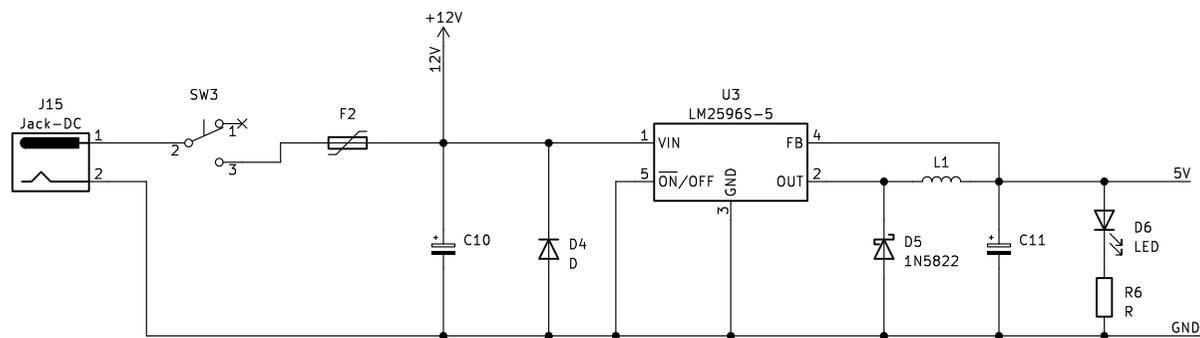
- VUSB — VUSB
- 5V — 5V
- GND — GND
- RXDAT — RXDAT
- TXDAT — TXDAT
- RESET — RESET
- MISO — MISO
- MOSI — MOSI
- SCK — SCK



Óscar Gutiérrez Álvarez
 Óscar Gutiérrez Álvarez
Digital Codesign
 Sheet: /USB/
 File: USB.kicad_sch

Title: Brazo Robótico Educativo

Size: A4	Date: 2024-01-16	Rev:
KiCad E.D.A. kicad 7.0.1		Id: 4/5



5V \diamond 5V
 GND \diamond GND
 12V \diamond 12V

Óscar Gutiérrez Álvarez
 Óscar Gutiérrez Álvarez

Digital Codesign

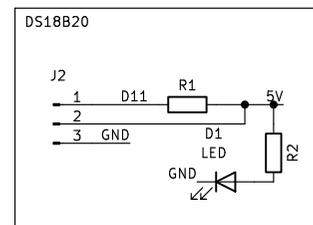
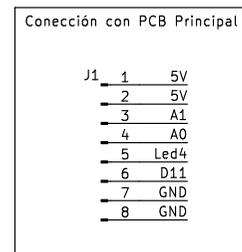
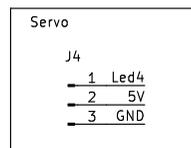
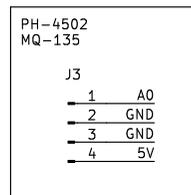
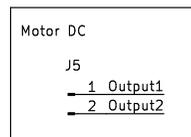
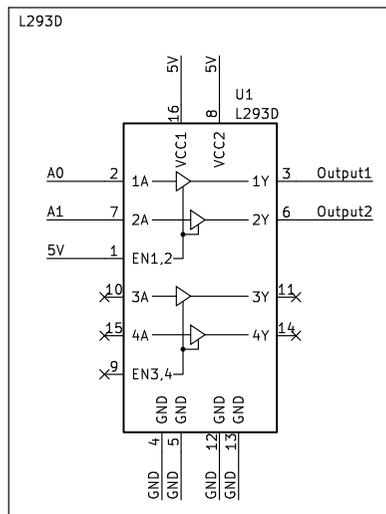
Sheet: /Alimentador de 12V a 5V/
 File: JACK12Va5V.kicad_sch

Title: Brazo Robótico Educativo

Size: A4 Date: 2024-01-16

KiCad E.D.A. kicad 7.0.1

Rev:
 Id: 5/5



Sheet: /
File: PCBAuxiliar.kicad_sch

Title:

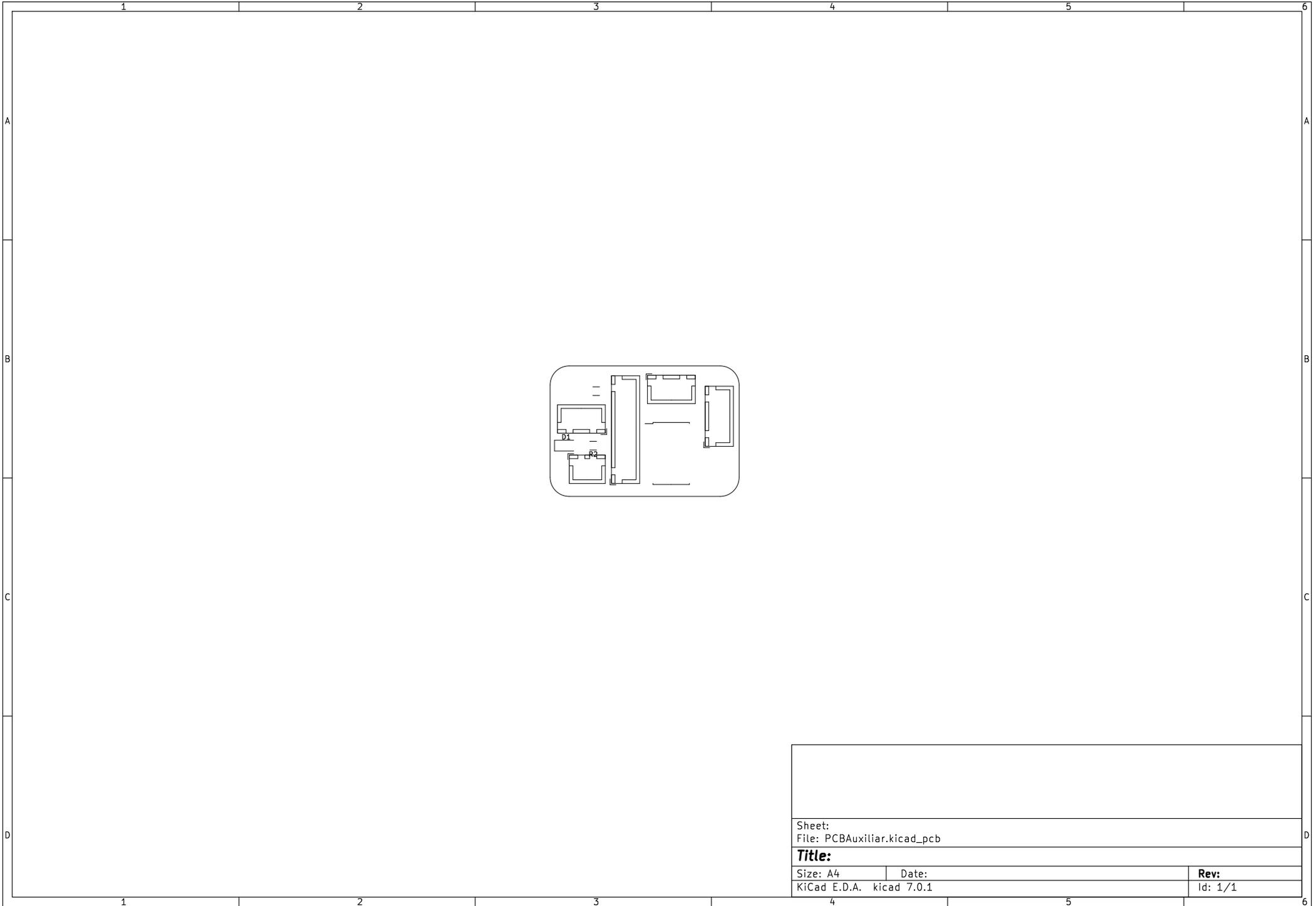
Size: A4

Date:

KiCad E.D.A. kicad 7.0.1

Rev:

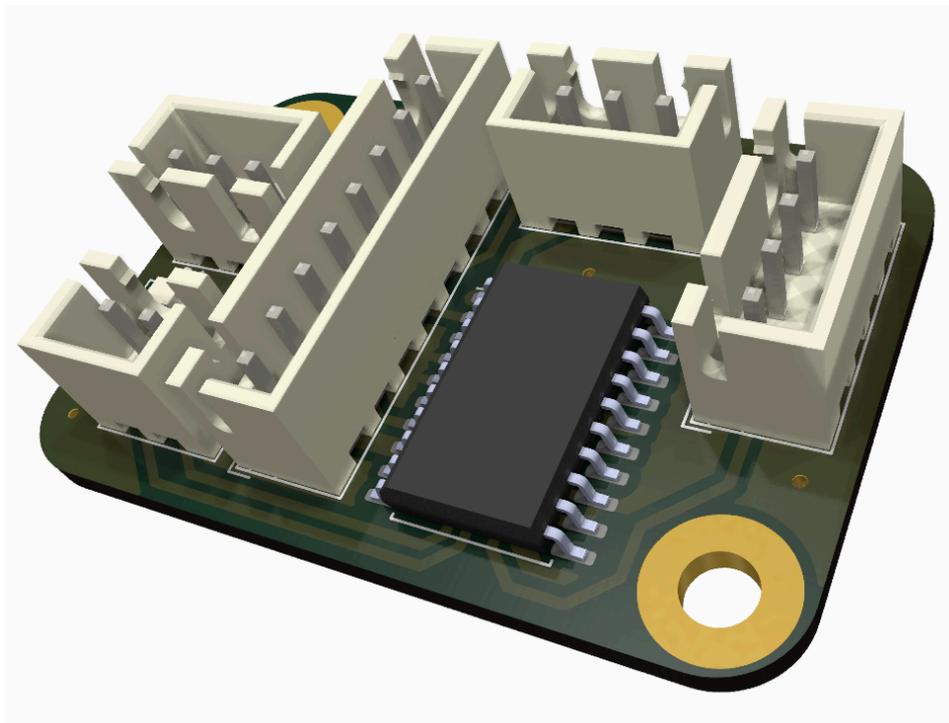
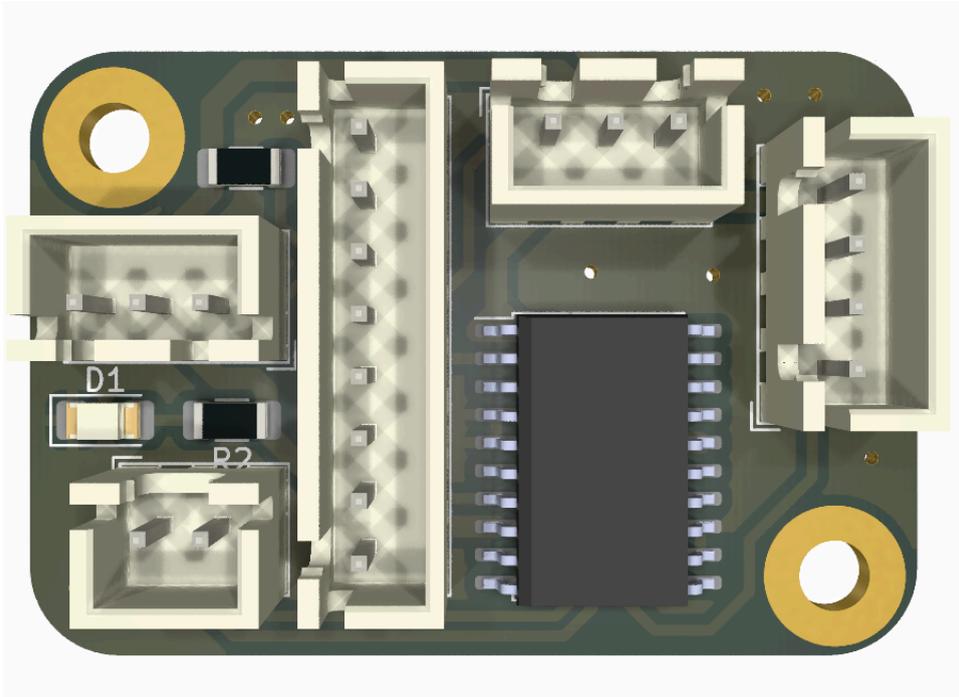
Id: 1/1



Sheet:
File: PCBAuxiliar.kicad_pcb

Title:	
Size: A4	Date:
KiCad E.D.A. kicad 7.0.1	Rev: 1/1

VISTAS 3D DE LA PCB AUXILIAR DEL BRAZO ROBÓTICO (V1.0)



ANEXO IV - SOFTWARE DEL DISPOSITIVO

IV.1 TEST_MODULE_SERVO

```
//Libraries
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

#include <PinChangeInterrupt.h>

#define CLK 2
#define DT 3
#define SW 4

//Constants
#define nbPCAServo 3

int counter = 0;
int currentState;
int initState;
unsigned long bebounceDelay = 0;

int Gradol;
int Boton = 0;
int Cambio = 0;

//Parameters
int MIN_IMP [nbPCAServo] ={500, 500, 500};
int MAX_IMP [nbPCAServo] ={2500, 2500, 2500};
int MIN_ANG [nbPCAServo] ={0, 0, 0};
int MAX_ANG [nbPCAServo] ={180, 180, 180};

//Objects
Adafruit_PWMServoDriver pca= Adafruit_PWMServoDriver(0x40);

void setup(){

    //Init Serial USB
    Serial.begin(9600);
    Serial.println(F("Initialize System"));
    pca.begin();
    pca.setPWMPFreq(60);    // Analog servos run at ~60 Hz updates

    pinMode(CLK, INPUT);
    pinMode(DT, INPUT);
    pinMode(SW, INPUT_PULLUP);

    initState = digitalRead(CLK);

    attachInterrupt(0, encoder_value, CHANGE);
    attachInterrupt(1, encoder_value, CHANGE);
    attachPCINT(digitalPinToPCINT(SW), button_press, CHANGE);

}

void loop(){

    Gradol = map(analogRead(A0), 0, 1023, 500, 2500);
    pcaScenario();
```

```

}

void pcaScenario(){/* function pcaScenario */

  //if(Boton == 0){

    pca.writeMicroseconds(0, Gradol);
    pca.writeMicroseconds(1, Gradol);
    pca.writeMicroseconds(2, Gradol);
    delay(10);

  //}

}

void button_press(){

  int buttonVal = digitalRead(SW);

  //If we detect LOW signal, button is pressed

  if (buttonVal == LOW) {

    if (millis() - bebounceDelay > 200) {

      Serial.println("Button pressed!");

      Cambio = 1;

      if(Boton == 0 && Cambio == 1){

        Boton = 1;
        Cambio = 0;

      }

      if(Boton == 1 && Cambio == 1){

        Boton = 0;
        Cambio = 0;

      }

    }

    bebounceDelay = millis();

  }

}

void encoder_value() {
//if(Boton == 0){

  // Read the current state of CLK
  currentState = digitalRead(CLK);

  // If last and current state of CLK are different, then we can be sure that
  the pulse occurred
  if (currentState != initState && currentState == 1) {

```

```

// Encoder is rotating counterclockwise so we decrement the counter
if (digitalRead(DT) != currentState) {

    counter = counter + 4;

    if (counter <= -60){
        counter = -60;
    }
    if (counter >= 60){
        counter = 60;
    }
} else {

    // Encoder is rotating clockwise so we increment the counter
    counter = counter - 4;

    if (counter <= -60){
        counter = -60;
    }
    if (counter >= 60){
        counter = 60;
    }
}

// print the value in the serial monitor window
Serial.print("Counter: ");
Serial.println(counter);

}

// Remember last CLK state for next cycle
initState = currentState;

//}
}

int jointToImp(double x,int i){/* function jointToImp */
////Convert joint angle into pwm command value

    int imp=(x - MIN_ANG[i]) * (MAX_IMP[i]-MIN_IMP[i]) /
(MAX_ANG[i]-MIN_ANG[i]) + MIN_IMP[i];
    imp=max(imp,MIN_IMP[i]);
    imp=min(imp,MAX_IMP[i]);

    return imp;
}

```

IV.2 TEST_MOVE_SERVO

```
#include <VarSpeedServo.h>

int Grado0 = 90;
int Grado1 = 90;
int Grado2 = 90;
int Grado3 = 90;

VarSpeedServo servo_3;
VarSpeedServo servo_4;
VarSpeedServo servo_5;
VarSpeedServo servo_6;

void setup() {

    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);
    pinMode(A3, INPUT);
    servo_3.attach(3);
    servo_4.attach(4);
    servo_5.attach(5);
    servo_6.attach(6);

}

void loop() {

    Grado0 = map(analogRead(A0), 0, 1023, 0, 180);
    Grado1 = map(analogRead(A1), 0, 1023, 0, 180);
    Grado2 = map(analogRead(A2), 0, 1023, 0, 180);
    Grado3 = map(analogRead(A3), 0, 1023, 0, 180);

    servo_3.write(Grado0, 25, false);
    servo_4.write(Grado1, 25, false);
    servo_5.write(Grado2, 25, false);
    servo_6.write(Grado3, 25, false);

    delay(20);

}
```

IV.3 TEST_MOVE_STEPPER

```
#include <AccelStepper.h>

#define dirPin 2
#define stepPin 3

int grado;

AccelStepper stepper1(1,stepPin,dirPin);

void setup() {

  Serial.begin(9600);

  stepper1.setMaxSpeed(1000);
  stepper1.setAcceleration(100);
  stepper1.moveTo(0);
  stepper1.moveTo(540);
  stepper1.moveTo(270);

  pinMode(stepPin,OUTPUT);
  pinMode(dirPin,OUTPUT);
  pinMode(A0,INPUT);

}

void loop() {

  grado = map(analogRead(A0), 0, 1023, 0, 540);
  stepper1.setSpeed(1000);
  stepper1.moveTo(grado);
  stepper1.runSpeedToPosition();

}
```

IV.4 TEST_MOVE_AUTOMATICO

```
#include <math.h>
#include <Wire.h>
#include <CinematicaInversa.h> //Librería para la Cinemática Inversa,
creación propia
#include <Adafruit_PWMServoDriver.h> //Librería para módulo de servos
#include <AccelStepper.h> //Librería para mover stepper a una
posición en concreto

#define nbPCAServo 5

#define dirPin 5
#define stepPin 6

#define Pinza 7
#define GuardarPosicion 8

//Creamos las variables que utilizaremos

int l1 = 20; //Longitud de la primera parte del brazo
int l2 = 12; //Longitud de la segunda parte del brazo
int l3 = 16; //Longitud de la tercera parte del brazo

//Variables para el monitor serial
String lectura; //Variable en letras para leer la lectura en el monitor
serial
bool leer; //Booleano para indicar cuando deja de leer el monitor
serial

//Variables para los ángulos de los servos y paso en el que se encuentra el
stepper
float Grado1;
float Grado2;
float Grado3;

int Paso0;

float Grado11;
float Grado21;
float Grado31;

//Coordenadas cartesianas

float x = 0;
float y = 0;
float z = 0;

//Posiciones del BRE

int Posicion[50][6];
int PosicionFinal[100][6];

//Pinza del BRE
int Abierto = 1; //Estado de la pinza (Abierto/Cerrado)
int IndicadorPinza = 0; //Indicador

//Variables de funcionamiento del código
int NumPos = 0;
int NumPosFinal = 1;
```

```

int CuentaNumPos;
int Menu = 0;
int PedirPosicion = 0;
int Moverse = 0;
int Boton = 0;
int Cambio = 0;
int Tiempo = 0;
int ContarTiempo = 0;

//Variables para el encoder

int counter = 150;
int counterpast = 150;
int currentState;
int initState;
unsigned long bebounceDelay = 0;

//Parameters
int MIN_IMP [nbPCAServo] ={500, 500, 500, 500, 500};
int MAX_IMP [nbPCAServo] ={2500, 2500, 2500, 2500, 2500};
int MIN_ANG [nbPCAServo] ={0, 0, 0, 0, 0};
int MAX_ANG [nbPCAServo] ={180, 180, 180, 180, 180};

//Objects
Adafruit_PWMServoDriver pca= Adafruit_PWMServoDriver(0x40);
AccelStepper stepper1(1, stepPin, dirPin);

void setup() {

    //Preparamos el monitor serial para controlar los diferentes grados

    Serial.begin(9600); Serial.begin(9600);

    Serial.println(F("Initialize System"));
    pca.begin();
    pca.setPWMPfreq(60); // Los servos en analógico se actualizan a 60 Hz

    pinMode(dirPin, OUTPUT);
    pinMode(stepPin, OUTPUT);

    //Abrimos la Pinza cuando iniciamos el programa

    pca.writeMicroseconds(4, 2500);
    delay(10);

}

void loop() {

    while (Menu == 0){

        Serial.println("Ponemos coordenada?");

        leer = true;

        while(leer == true){

            if (Serial.available()>0){

                lectura = Serial.readString();
                leer = false;
            }
        }
    }
}

```

```

    }

    delay(200);
}

//En caso de pulsar "n"
if (lectura == "y"){
    PedirPosicion = 1;
}

//En caso de pulsar otra tecla
if (lectura == "n"){
    //Pregunta para de colocar posiciones

    Serial.println("¿Quiere parar?    y/n ");
    leer = true;

    while(leer == true){
        if (Serial.available()>0){
            lectura = Serial.readString();
            leer = false;
        }

        delay(200);
    }

    //Parar

    if (lectura == "y"){
        Menu = 1;
        PedirPosicion = 0;
        Moverse = 1;
    }

    //Seguir

    if (lectura == "n"){
        Menu = 0;
        PedirPosicion = 0;
    }
}

//Pedir posiciones XYZ
while (PedirPosicion == 1){

```

```
Serial.println("Posición x = ");
leer = true;

while(leer == true){

    if (Serial.available()>0){

        lectura = Serial.readString();
        leer = false;

    }

    delay(100);

}

Serial.print("Lo leído es: ");
Serial.println(lectura);

float x = lectura.toFloat();

Serial.println("Posición y = ");
leer = true;

while(leer == true){

    if (Serial.available()>0){

        lectura = Serial.readString();
        leer = false;

    }

    delay(100);

}

Serial.print("Lo leído es: ");
Serial.println(lectura);

float y = lectura.toFloat();

Serial.println("Posición z = ");
leer = true;

while(leer == true){

    if (Serial.available()>0){

        lectura = Serial.readString();
        leer = false;

    }

    delay(100);

}

Serial.print("Lo leído es: ");
Serial.println(lectura);
```

```

float z = lectura.toFloat();

//Cinemática inversa del punto elegido

CI ( x, y, z, l1, l2, l3, 0, 180, Paso0, Grado1, Grado2, Grado3);

//Guarda las posiciones que tienen que tener los servos y stepper

Posicion[NumPosFinal][1] = map(Grado1, 0, 180, 500, 2500);
Posicion[NumPosFinal][2] = map(Grado2, 0, 180, 500, 2500);
Posicion[NumPosFinal][3] = map(Grado3, 0, 180, 500, 2500);

PedirPosicion = 0;
NumPosFinal = NumPosFinal + 1;

}
}

if (Menu == 1){

//El código bajando es para realizar los movimientos unos después de
otro.

for (int i = 1; i <= NumPosFinal; i++){

Serial.print("Moviendose a la posicion ");
Serial.println(i);

Grado1 = Posicion[i][1];
Grado2 = Posicion[i][2];
Grado3 = Posicion[i][3];

delay(500);

pcaScenario();

delay(1500);

}

Menu = 0;

}

}

void pcaScenario(){

pca.writeMicroseconds(0, (Grado1 - (Grado1-Grado11)/2));
pca.writeMicroseconds(1, (Grado2 - (Grado2-Grado21)/2));
pca.writeMicroseconds(2, (Grado3 - (Grado3-Grado31)/2));

delay(100);

pca.writeMicroseconds(0, Grado1);
pca.writeMicroseconds(1, Grado2);
pca.writeMicroseconds(2, Grado3);

delay(10);
}

```

IV.5 TEST_SERVO_VELOCIDAD

```
#include <Servo.h>

#include <PinChangeInterrupt.h>

#define CLK 2
#define DT 3
#define SW 4

//Constants
Servo servo_3;
Servo servo_5;
Servo servo_6;

int t = 10;
int Grado1;
int Grado2;
int Grado3;
int max;
int Grado1Anterior;
int Grado2Anterior;
int Grado3Anterior;
int DifGrado1;
int DifGrado2;
int DifGrado3;
float Coeficiente1;
float Coeficiente2;
float Coeficiente3;
int Mover1;
int Mover2;
int Mover3;

void setup() {

    //Init Serial USB
    Serial.begin(9600);
    Serial.println(F("Initialize System"));

    servo_3.attach(3, 500, 2500);
    servo_5.attach(5, 500, 2500);
    servo_6.attach(6, 500, 2500);

    Grado1 = 500;
    Grado2 = 500;
    Grado3 = 500;

    servo_3.write(int(Grado1));
    servo_5.write(int(Grado2));
    servo_6.write(int(Grado3));

    delay(600);
}

void loop() {

    Grado1Anterior = Grado1;
    Grado2Anterior = Grado2;
    Grado3Anterior = Grado3;
```

```

Gradol = 500;
Grado2 = 2200;
Grado3 = 1000;

DifGradol = Gradol - GradolAnterior;
DifGrado2 = Grado2 - Grado2Anterior;
DifGrado3 = Grado3 - Grado3Anterior;

if(abs(DifGradol) >= abs(DifGrado2) && abs(DifGradol) >= abs(DifGrado3)){

    if(DifGradol >= 0){

        Coeficiente1 = DifGradol/DifGradol;
        Coeficiente2 = DifGrado2/DifGradol;
        Coeficiente3 = DifGrado3/DifGradol;

    }

    if(DifGradol < 0){

        Coeficiente1 = -DifGradol/DifGradol;
        Coeficiente2 = -DifGrado2/DifGradol;
        Coeficiente3 = -DifGrado3/DifGradol;

    }

    max = abs(DifGradol);

}

if(abs(DifGrado2) >= abs(DifGradol) && abs(DifGrado2) >= abs(DifGrado3)){

    if(DifGrado2 >= 0){

        Coeficiente1 = DifGradol/DifGrado2;
        Coeficiente2 = DifGrado2/DifGrado2;
        Coeficiente3 = DifGrado3/DifGrado2;

    }

    if(DifGrado2 < 0){

        Coeficiente1 = -DifGradol/DifGrado2;
        Coeficiente2 = -DifGrado2/DifGrado2;
        Coeficiente3 = -DifGrado3/DifGrado2;

    }

    max = abs(DifGrado2);

}

if(abs(DifGrado3) >= abs(DifGrado2) && abs(DifGrado3) >= abs(DifGradol)){

    if(DifGrado3 >= 0){

        Coeficiente1 = DifGradol/DifGrado3;
        Coeficiente2 = DifGrado2/DifGrado3;
        Coeficiente3 = DifGrado3/DifGrado3;

    }

}

```

```

if(DifGrado3 < 0){

    Coeficiente1 = -DifGrado1/DifGrado3;
    Coeficiente2 = -DifGrado2/DifGrado3;
    Coeficiente3 = -DifGrado3/DifGrado3;

}

max = abs(DifGrado3);

}

for(int i = 1; i <= max; i++){

    Mover1 = Grado1Anterior + i*Coeficiente1;
    Mover2 = Grado2Anterior + i*Coeficiente2;
    Mover3 = Grado3Anterior + i*Coeficiente3;

    servo_3.write(Mover1);
    servo_5.write(Mover2);
    servo_6.write(Mover3);
    delay(t);

}

delay(1000);

Grado1Anterior = Grado1;
Grado2Anterior = Grado2;
Grado3Anterior = Grado3;

Grado1 = 2400;
Grado2 = 700;
Grado3 = 1500;

DifGrado1 = Grado1 - Grado1Anterior;
DifGrado2 = Grado2 - Grado2Anterior;
DifGrado3 = Grado3 - Grado3Anterior;

if(abs(DifGrado1) >= abs(DifGrado2) && abs(DifGrado1) >= abs(DifGrado3)){

    if(DifGrado1 >= 0){

        Coeficiente1 = DifGrado1/DifGrado1;
        Coeficiente2 = DifGrado2/DifGrado1;
        Coeficiente3 = DifGrado3/DifGrado1;

    }

    if(DifGrado1 < 0){

        Coeficiente1 = -DifGrado1/DifGrado1;
        Coeficiente2 = -DifGrado2/DifGrado1;
        Coeficiente3 = -DifGrado3/DifGrado1;

    }

    max = abs(DifGrado1);

}

```

```

if(abs(DifGrado2) >= abs(DifGrado1) && abs(DifGrado2) >= abs(DifGrado3)){

    if(DifGrado2 >= 0){

        Coeficiente1 = DifGrado1/DifGrado2;
        Coeficiente2 = DifGrado2/DifGrado2;
        Coeficiente3 = DifGrado3/DifGrado2;

    }

    if(DifGrado2 < 0){

        Coeficiente1 = -DifGrado1/DifGrado2;
        Coeficiente2 = -DifGrado2/DifGrado2;
        Coeficiente3 = -DifGrado3/DifGrado2;

    }

    max = abs(DifGrado2);

}

if(abs(DifGrado3) >= abs(DifGrado2) && abs(DifGrado3) >= abs(DifGrado1)){

    if(DifGrado3 >= 0){

        Coeficiente1 = DifGrado1/DifGrado3;
        Coeficiente2 = DifGrado2/DifGrado3;
        Coeficiente3 = DifGrado3/DifGrado3;

    }

    if(DifGrado3 < 0){

        Coeficiente1 = -DifGrado1/DifGrado3;
        Coeficiente2 = -DifGrado2/DifGrado3;
        Coeficiente3 = -DifGrado3/DifGrado3;

    }

    max = abs(DifGrado3);

}

for(int i = 1; i <= max; i++){

    Mover1 = GradolAnterior + i*Coeficiente1;
    Mover2 = Grado2Anterior + i*Coeficiente2;
    Mover3 = Grado3Anterior + i*Coeficiente3;

    servo_3.write(Mover1);
    servo_5.write(Mover2);
    servo_6.write(Mover3);
    delay(t);

}

delay(1000);

GradolAnterior = Gradol;

```

```

Grado2Anterior = Grado2;
Grado3Anterior = Grado3;

Grado1 = 2500;
Grado2 = 1000;
Grado3 = 700;

DifGrado1 = Grado1 - Grado1Anterior;
DifGrado2 = Grado2 - Grado2Anterior;
DifGrado3 = Grado3 - Grado3Anterior;

if(abs(DifGrado1) >= abs(DifGrado2) && abs(DifGrado1) >= abs(DifGrado3)){

    if(DifGrado1 >= 0){

        Coeficiente1 = DifGrado1/DifGrado1;
        Coeficiente2 = DifGrado2/DifGrado1;
        Coeficiente3 = DifGrado3/DifGrado1;

    }

    if(DifGrado1 < 0){

        Coeficiente1 = -DifGrado1/DifGrado1;
        Coeficiente2 = -DifGrado2/DifGrado1;
        Coeficiente3 = -DifGrado3/DifGrado1;

    }

    max = abs(DifGrado1);

}

if(abs(DifGrado2) >= abs(DifGrado1) && abs(DifGrado2) >= abs(DifGrado3)){

    if(DifGrado2 >= 0){

        Coeficiente1 = DifGrado1/DifGrado2;
        Coeficiente2 = DifGrado2/DifGrado2;
        Coeficiente3 = DifGrado3/DifGrado2;

    }

    if(DifGrado2 < 0){

        Coeficiente1 = -DifGrado1/DifGrado2;
        Coeficiente2 = -DifGrado2/DifGrado2;
        Coeficiente3 = -DifGrado3/DifGrado2;

    }

    max = abs(DifGrado2);

}

if(abs(DifGrado3) >= abs(DifGrado2) && abs(DifGrado3) >= abs(DifGrado1)){

    if(DifGrado3 >= 0){

        Coeficiente1 = DifGrado1/DifGrado3;
        Coeficiente2 = DifGrado2/DifGrado3;

```

```
    Coeficiente3 = DifGrado3/DifGrado3;
}

if(DifGrado3 < 0){
    Coeficiente1 = -DifGrado1/DifGrado3;
    Coeficiente2 = -DifGrado2/DifGrado3;
    Coeficiente3 = -DifGrado3/DifGrado3;
}

max = abs(DifGrado3);
}

for(int i = 1; i <= max; i++){
    Mover1 = Grado1Anterior + i*Coeficiente1;
    Mover2 = Grado2Anterior + i*Coeficiente2;
    Mover3 = Grado3Anterior + i*Coeficiente3;

    servo_3.write(Mover1);
    servo_5.write(Mover2);
    servo_6.write(Mover3);
    delay(t);
}

delay(1000);
}
```

IV.6 TEST_LCD

```
#include <math.h>
#include <Wire.h>
#include <CinematicaInversa.h>           //Librería para la Cinemática Inversa,
creación propia
#include <PinChangeInterrupt.h>         //Librería para encoder
#include <LiquidCrystal_I2C.h>

//Definimos los pines

#define CLK 2
#define DT 3
#define SW 4

int buttonVal;
int Boton = 0;
int Movimiento;
int Opcion = 1;
int Parar = 0;
float Posicion[6][4];
int Seguir = 0;

//Creamos las variables que utilizaremos

int l1 = 20; //Longitud de la primera parte del brazo
int l2 = 12; //Longitud de la segunda parte del brazo
int l3 = 16; //Longitud de la tercera parte del brazo

//Variables para los ángulos de los servos y paso en el que se encuentra el
stepper
int Paso0 = 150;
float Grado1;
float Grado2;
float Grado3;
int Grado4;

//Coordenadas cartesianas

float x = 0;
float y = 5;
float z = 0;
int Pinza = 1;

//Variables para el encoder

int counter = 150;
int counterpast = 150;
int currentState;
int initState;
unsigned long bebounceDelay = 0;

LiquidCrystal_I2C lcd(0x27,20,4);

int PedirPosicion = 1;

void setup() {

    Serial.begin(9600);

    // Inicializar el LCD
```

```

lcd.init();

//Encender la luz de fondo.
lcd.backlight();

pinMode(CLK, INPUT);
pinMode(DT, INPUT);
pinMode(SW, INPUT_PULLUP);

// Escribimos el Mensaje en el LCD.
lcd.setCursor(0, 0);
lcd.print("Inicializando");
delay(1000);
lcd.setCursor(0, 0);
lcd.print("Inicializando.");
delay(1000);
lcd.setCursor(0, 0);
lcd.print("Inicializando..");
delay(1000);
lcd.setCursor(0, 0);
lcd.print("Inicializando...");
delay(1000);
lcd.clear();
//Indicadores del encoder

initState = digitalRead(CLK);

attachInterrupt(0, encoder_value, CHANGE);
attachInterrupt(1, encoder_value, CHANGE);
attachPCINT(digitalPinToPCINT(SW), button_press, CHANGE);

}

void loop() {

  lcd.clear();
  lcd.setCursor(3, 1);
  lcd.print("Estableciendo");
  lcd.setCursor(3, 2);
  lcd.print("Posiciones");
  for(int i = 0; i <= 2; i++){

    delay(1000);
    lcd.print(".");

  }

  delay(1000);
  lcd.clear();

  while (Manual == 0){

    for(int i = 1; i <= 5; i++){

      Movimiento = 1;

      while(Movimiento == 1){

        if(Opcion == 1){

```

```

        lcd.setCursor(0,0);
        lcd.print("Quiere que la posi- ");
lcd.setCursor(0,1);
lcd.print("cion ");
lcd.print(i);
lcd.print(" se guarde en");
lcd.setCursor(0,2);
lcd.print("el movimiento ");
lcd.print(i);
lcd.print("? ");
lcd.setCursor(0,3);
lcd.print(" ->Si   No   Parar");

if (counter > counterpast){

    delay(200);
    lcd.clear();
    Opcion = 2;
    counterpast = counter;

}

if(Boton == 1){

    delay(200);
    lcd.clear();
    Movimiento = 0;
    Boton = 0;
    PedirPosicion = 0;
    Posicion[i][0] = x;
    Posicion[i][1] = y;
    Posicion[i][2] = z;
    Posicion[i][3] = Pinza;

}

}

if(Opcion == 2){

    lcd.setCursor(0,0);
    lcd.print("Quiere que la posi- ");
    lcd.setCursor(0,1);
    lcd.print("cion ");
    lcd.print(i);
    lcd.print(" se guarde en");
    lcd.setCursor(0,2);
    lcd.print("el movimiento ");
    lcd.print(i);
    lcd.print("? ");
    lcd.setCursor(0,3);
    lcd.print("   Si ->No   Parar");

    if (counter > counterpast){

        delay(200);
        lcd.clear();
        Opcion = 3;
        counterpast = counter;

    }

}

```

```

if (counter < counterpast){

    delay(200);
    lcd.clear();
    Opcion = 1;
    counterpast = counter;

}

if(Boton == 1){

    lcd.clear();
    PedirPosicion = 1;
    Movimiento = 0;
    Boton = 0;

}

}

if(Opcion == 3){

    lcd.setCursor(0,0);
    lcd.print("Quiere que la posi- ");
    lcd.setCursor(0,1);
    lcd.print("cion ");
    lcd.print(i);
    lcd.print(" se guarde en");
    lcd.setCursor(0,2);
    lcd.print("el movimiento ");
    lcd.print(i);
    lcd.print("? ");
    lcd.setCursor(0,3);
    lcd.print(" Si No ->Parar");

    if (counter < counterpast){

        delay(200);
        lcd.clear();
        Opcion = 2;
        counterpast = counter;

    }

    if(Boton == 1){

        lcd.clear();
        Parar = 1;
        Movimiento = 0;
        Boton = 0;

    }

}

}

while(PedirPosicion == 1){

    lcd.setCursor(0,0);

```

```

lcd.print("Que posicion quiere");
lcd.setCursor(0,1);
lcd.print("que tenga la coorde-");
lcd.setCursor(0,2);
lcd.print("nada x?");
lcd.setCursor(7,3);
lcd.print("x = ");
lcd.print(x,0);
lcd.print("  ");

if (counter < counterpast){

    x = x - 1;
    counterpast = counter;

    if (x <= -30){

        x = -30;

    }

}

if (counter > counterpast){

    x = x + 1;
    counterpast = counter;

    if (x >= 30){

        x = 30;

    }

}

if(Boton == 1){

    PedirPosicion = 2;
    Boton = 0;

}

}

lcd.clear();

while(PedirPosicion == 2){

    lcd.setCursor(0,0);
    lcd.print("Que posicion quiere");
    lcd.setCursor(0,1);
    lcd.print("que tenga la coorde-");
    lcd.setCursor(0,2);
    lcd.print("nada y?");
    lcd.setCursor(7,3);
    lcd.print("y = ");
    lcd.print(y,0);
    lcd.print("  ");

    if (counter < counterpast){

```

```

y = y - 1;
counterpast = counter;

if (y <= 0){
    y = 0;
}
}

if (counter > counterpast){

    y = y + 1;
    counterpast = counter;

    if (y >= 30){
        y = 30;
    }
}

if(Boton == 1){
    PedirPosicion = 3;
    Boton = 0;
}
}

lcd.clear();

while(PedirPosicion == 3){

    lcd.setCursor(0,0);
    lcd.print("Que posicion quiere");
    lcd.setCursor(0,1);
    lcd.print("que tenga la coorde-");
    lcd.setCursor(0,2);
    lcd.print("nada z?");
    lcd.setCursor(7,3);
    lcd.print("z = ");
    lcd.print(z,0);
    lcd.print("  ");

    if (counter < counterpast){

        z = z - 1;
        counterpast = counter;

        if (z <= -30){
            z = -30;
        }
    }
}

if (counter > counterpast){

```

```

z = z + 1;
counterpast = counter;

if (z >= 30){

    z = 30;

}

}

if(Boton == 1){
    PedirPosicion = 4;
    Boton = 0;
}

}

lcd.clear();

while(PedirPosicion == 4){

    if(Pinza == 1){

        lcd.setCursor(0,0);
        lcd.print("Quiere que la pinza ");
        lcd.setCursor(0,1);
        lcd.print("se mantenga:          ");
        lcd.setCursor(0,3);
        lcd.print("->Abierta    Cerrada");

        if (counter > counterpast){

            delay(200);
            lcd.clear();
            Pinza = 0;
            counterpast = counter;

        }

    }

}

if(Pinza == 0){

    lcd.setCursor(0,0);
    lcd.print("Quiere que la pinza ");
    lcd.setCursor(0,1);
    lcd.print("se mantenga:          ");
    lcd.setCursor(0,3);
    lcd.print("  Abierta  ->Cerrada");

    if (counter < counterpast){

        delay(200);
        lcd.clear();
        Pinza = 1;
        counterpast = counter;

    }

}

```

```

    }

    if(Boton == 1){

        PedirPosicion = 5;
        Boton = 0;

    }

}

lcd.clear();

while(PedirPosicion == 5){

    lcd.setCursor(0,0);
    lcd.print("Posicion x = ");
    lcd.print(x,1);
    lcd.setCursor(0,1);
    lcd.print("Posicion y = ");
    lcd.print(y,1);
    lcd.setCursor(0,2);
    lcd.print("Posicion z = ");
    lcd.print(z,1);
    lcd.setCursor(0,3);
    lcd.print("Pinza = ");

    if(Pinza == 1){
        lcd.print("Abierto");
    }
    if(Pinza == 0){
        lcd.print("Cerrado");
    }

    if(Boton == 1){
        PedirPosicion = 0;
        Boton = 0;
    }

    Posicion[i][0] = x;
    Posicion[i][1] = y;
    Posicion[i][2] = z;
    Posicion[i][3] = Pinza;

}

while(Parar == 1){
    lcd.setCursor(0,0);
    lcd.print(" Quiere que empiece ");
    lcd.setCursor(0,1);
    lcd.print(" a moverse el brazo?");
    lcd.setCursor(0,3);

    if (Seguir == 0){
        lcd.print("    ->Si        No    ");
    }
    if (Seguir == 1){
        lcd.print("        Si    ->No    ");
    }
}

if (counter > counterpast){

```

```

    delay(200);
    lcd.clear();
    Seguir = 1;
    counterpast = counter;
}

if (counter < counterpast){

    delay(200);
    lcd.clear();
    Seguir = 0;
    counterpast = counter;

}

if(Boton == 1){

    lcd.clear();
    Movimiento = 0;

    if (Seguir == 0){
        i = 7;
        Parar = 2;
    }

    if (Seguir == 1){
        Movimiento = 1;
        i= i - 1;
    }

    Parar = 0;
    Boton = 0;

}

}

}

Parar = 2;

while(Parar == 2){

for(int j = 1; j <= 5; j++){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Moviendo a la posi- ");
    lcd.setCursor(0,1);
    lcd.print("cion ");
    lcd.print(j);
    lcd.print(": ");
    lcd.setCursor(0,2);
    lcd.print(" x=");
    lcd.print(Posicion[j][0],1);
    lcd.print(" ");
    lcd.setCursor(8,2);
    lcd.print(" y=");
    lcd.print(Posicion[j][1],1);
    lcd.setCursor(0,3);

```

```

    lcd.print(" z=");
    lcd.print(Posicion[j][2],1);
    lcd.print(" ");
    lcd.setCursor(8,3);
    lcd.print(" Pinza=");

    if(Posicion[j][3] == 1){
        lcd.print("Open");
    }
    if(Posicion[j][3] == 0){
        lcd.print("Close");
    }

    delay(2000);

}

Parar = 0;

}
}
}

//Función del encoder
void encoder_value() {

    currentState = digitalRead(CLK);

    if (currentState != initState && currentState == 1) {

        if (digitalRead(DT) != currentState) {

            counter = counter + 1;

        } else {

            counter = counter - 1;

        }

    }

    initState = currentState;
}

void button_press(){

    buttonVal = digitalRead(SW);

    if (buttonVal == LOW) {

        if (millis() - bebounceDelay > 200) {

            Serial.println("Button pressed!");

            Boton = 1;

        }

        bebounceDelay = millis();
    } }

```

IV.7 CINEMÁTICA INVERSA

```
#include "Arduino.h"
#include "CinematicaInversa.h"
#include <math.h>

int l = 0;
float resta1[2];
float resta2[2];
float r[2];
float distancia = 1;
float dif_angulo = 0;
float AnguloPaso;
float angulo1;
float angulo2;
float limite;
int contador = 0;
float limite_bajo;
float limite_alto;

void CI (float x, float y, float z, int l1, int l2, int l3, int limite_min,
int limite_max, int &Paso0, float &Grado1, float &Grado2, float &Grado3) {

    //El código se queda en un bucle while hasta que alcanza el valor deseado

    distancia = 1;
    contador = 0;

    limite_bajo = limite_min*(2*M_PI/360);
    limite_alto = limite_max*(2*M_PI/360);

    //Cálculo del paso que tiene que alcanzar el stepper

    if( x == 0){

        if(z >= 0){

            AnguloPaso = M_PI/2;

        }

        if(z < 0){

            AnguloPaso = -M_PI/2;

        }

    }

    if(x > 0){

        if(z > 0){

            AnguloPaso = atan(z/x);

        }

        if(z < 0){

            AnguloPaso = - atan(abs(z)/x);

        }

    }

}
```

```

    }

    if(z == 0){
        AnguloPaso = 0;
    }
}

if(x < 0){
    if(z > 0){
        AnguloPaso = atan(z/abs(x)) + M_PI/2;
    }

    if(z < 0){
        AnguloPaso = - atan(z/x) - M_PI/2;
    }

    if(z == 0){
        AnguloPaso = M_PI;
    }
}

if (AnguloPaso >= 0){
    Paso0 = map(AnguloPaso, 0, M_PI, 0, 100);
}

if (AnguloPaso <= 0){
    Paso0 = map(AnguloPaso, 0, -M_PI, 200, 100);
}

Grado1 = M_PI/2;
Grado2 = M_PI/2;
Grado3 = M_PI/2;

while(distancia >= 0.01){
    //Comenzamos con un if con un contador para que haga los cálculos una
    sola vez

    if (contador == 0){

        //Se calcula los vectores para el punto final

        r[0] = sqrt(sq(x)+sq(z));
        r[1] = y;
        l = sqrt(sq(l1-l3)+sq(l2)); //Límite del propio brazo
        limite = sqrt(sq(r[0])+sq(r[1])); //Alcance del brazo
    }
}

```

```

}

float O00[2] = {0, 0};
float O01[2] = {cos(Grado1)*l1, sin(Grado1)*l1};
float O02[2] = {cos(Grado2-(M_PI/2 - Grado1))*l2 +
O01[0], sin(Grado2-(M_PI/2 - Grado1))*l2 + O01[1]};
float O03[2] = {cos(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Grado1))))*l3 +
O02[0], sin(Grado3-(M_PI/2 - (Grado2 - (M_PI/2 - Grado1))))*l3 + O02[1]};

resta1[0] = r[0] - O02[0];
resta1[1] = r[1] - O02[1];
resta2[0] = O03[0] - O02[0];
resta2[1] = O03[1] - O02[1];

if (resta1[0] == 0){

    if (resta1[1] >= 0){

        angulo1 = M_PI/2;

    }

    if (resta1[1] < 0){

        angulo1 = -M_PI/2;

    }

}

if (resta1[0] != 0){

    if(resta1[0] >= 0){

        if (resta1[1] >= 0){

            angulo1 = abs(atan(resta1[1] / resta1[0]));

        }

        if (resta1[1] < 0){

            angulo1 = - abs(atan(resta1[1] / resta1[0]));

        }

    }

    if(resta1[0] < 0){

        if (resta1[1] >= 0){

            angulo1 = abs(atan(resta1[1] / resta1[0]));

        }

        if (resta1[1] < 0){

            angulo1 = - abs(atan(resta1[1] / resta1[0]));

        }

    }

}

```

```

    }
}
}
if (resta2[0] == 0){
    if (resta2[1] >= 0){
        angulo2 = M_PI/2;
    }
    if (resta2[1] < 0){
        angulo2 = -M_PI/2;
    }
}
if (resta2[0] != 0){
    if(resta2[0] >= 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
        if (resta2[1] < 0){
            angulo2 = - abs(atan(resta2[1] / resta2[0]));
        }
    }
    if(resta2[0] < 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
        if (resta2[1] < 0){
            angulo2 = - abs(atan(resta2[1] / resta2[0]));
        }
    }
}
}
dif_angulo = angulo2 - angulo1;

```

```

Grado3 = Grado3 - dif_angulo;

if (Grado3 <= limite_bajo){
    Grado3 = limite_bajo;
}

if (Grado3 >= limite_alto){
    Grado3 = limite_alto;
}

O03[0] = cos(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Grado1))))*l3 + O02[0];
O03[1] = sin(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Grado1))))*l3 + O02[1];

resta1[0] = r[0] - O01[0];
resta1[1] = r[1] - O01[1];
resta2[0] = O03[0] - O01[0];
resta2[1] = O03[1] - O01[1];

if (resta1[0] == 0){
    if (resta1[1] >= 0){
        angulo1 = M_PI/2;
    }
    if (resta1[1] < 0){
        angulo1 = -M_PI/2;
    }
}

if (resta1[0] != 0){
    if(resta1[0] >= 0){
        if (resta1[1] >= 0){
            angulo1 = abs(atan(resta1[1] / resta1[0]));
        }
        if (resta1[1] < 0){
            angulo1 = - abs(atan(resta1[1] / resta1[0]));
        }
    }
    if(resta1[0] < 0){
        if (resta1[1] >= 0){
            angulo1 = abs(atan(resta1[1] / resta1[0]));
        }
    }
}

```

```

    }
    if (resta1[1] < 0){
        angulo1 = - abs(atan(resta1[1] / resta1[0]));
    }
}
}
if (resta2[0] == 0){
    if (resta2[1] >= 0){
        angulo2 = M_PI/2;
    }
    if (resta2[1] < 0){
        angulo2 = -M_PI/2;
    }
}
if (resta2[0] != 0){
    if(resta2[0] >= 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
        if (resta2[1] < 0){
            angulo2= - abs(atan(resta2[1] / resta2[0]));
        }
    }
    if(resta2[0] < 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
        if (resta2[1] < 0){
            angulo2 = - abs(atan(resta2[1] / resta2[0]));
        }
    }
}

```

```

    }
}
dif_angulo = angulo2 - angulo1;
Grado2 = Grado2 - dif_angulo;
if (Grado2 <= limite_bajo){
    Grado2 = limite_bajo;
}
if (Grado2 >= limite_alto){
    Grado2 = limite_alto;
}

O02[0] = cos(Grado2-(M_PI/2 - Grado1))*l2 + O01[0];
O02[1] = sin(Grado2-(M_PI/2 - Grado1))*l2 + O01[1];
O03[0] = cos(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Grado1))))*l3 + O02[0];
O03[1] = sin(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Grado1))))*l3 + O02[1];

restal[0] = r[0] - O00[0];
restal[1] = r[1] - O00[1];
resta2[0] = O03[0] - O00[0];
resta2[1] = O03[1] - O00[1];

if (restal[0] == 0){
    if (restal[1] >= 0){
        angulo1 = M_PI/2;
    }
    if (restal[1] < 0){
        angulo1 = -M_PI/2;
    }
}

if (restal[0] != 0){
    if(restal[0] >= 0){
        if (restal[1] >= 0){
            angulo1 = abs(atan(restal[1] / restal[0]));
        }
        if (restal[1] < 0){
            angulo1 = - abs(atan(restal[1] / restal[0]));
        }
    }
}

```

```

}

if(resta1[0] < 0){
    if (resta1[1] >= 0){
        angulo1 = abs(atan(resta1[1] / resta1[0]));
    }
    if (resta1[1] < 0){
        angulo1 = - abs(atan(resta1[1] / resta1[0]));
    }
}

}

if (resta2[0] == 0){
    if (resta2[1] >= 0){
        angulo2 = M_PI/2;
    }
    if (resta2[1] < 0){
        angulo2 = -M_PI/2;
    }
}

if (resta2[0] != 0){
    if(resta2[0] >= 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
        if (resta2[1] < 0){
            angulo2= - abs(atan(resta2[1] / resta2[0]));
        }
    }
    if(resta2[0] < 0){
        if (resta2[1] >= 0){
            angulo2 = abs(atan(resta2[1] / resta2[0]));
        }
    }
}

```

```

    }

    if (resta2[1] < 0){
        angulo2 = - abs(atan(resta2[1] / resta2[0]));
    }
}

}

dif_angulo = angulo2 - angulo1;
Gradol = Gradol - dif_angulo;

if (Gradol <= limite_bajo){
    Gradol = limite_bajo;
}

if (Gradol >= limite_alto){
    Gradol = limite_alto;
}

O01[0] = cos(Gradol)*l1;
O01[1] = sin(Gradol)*l1;
O02[0] = cos(Grado2-(M_PI/2 - Gradol))*l2 + O01[0];
O02[1] = sin(Grado2-(M_PI/2 - Gradol))*l2 + O01[1];
O03[0] = cos(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Gradol))))*l3 + O02[0];
O03[1] = sin(Grado3-(M_PI/2 - (Grado2-(M_PI/2-Gradol))))*l3 + O02[1];

distancia = sqrt(sq(abs(O03[0]-r[0]))+sq(abs(O03[1]-r[1])));
contador = 1;

}

Gradol = map(Gradol, 0, M_PI, 0, 180);
Grado2 = map(Grado2, 0, M_PI, 0, 180);
Grado3 = map(Grado3, 0, M_PI, 0, 180);

}

```