



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño, montaje y programación del control cinemático de un brazo robot manipulador de 5 grados de libertad

TRABAJO DE FIN DE GRADO

16 de agosto de 2021, València

Autor: Jarque Pérez, Diego

Tutor: González Sorribes, Antonio

4º Curso de Ingeniería Mecánica: Especialidad Diseño de Máquinas

RESUMEN

El presente proyecto consiste en el diseño de un brazo robot de 3 grados de libertad con la finalidad de emular una tarea de posicionamiento del órgano terminal del robot, consistente en una pinza de agarre. En primer lugar, se procederá con el modelado del conjunto de partes del brazo mediante el programa SolidWorks, y posteriormente se aplicarán técnicas de modelado cinemático para finalmente poder implementar una tarea de control cinemático. La programación del robot se basará en Arduino IDE, en particular Arduino UNO R3, para controlar los movimientos del robot y el accionamiento remoto de la pinza.

Palabras Clave: Brazo robot, modelado, control cinemático, Arduino IDE.

ABSTRACT

The following Project consists of the design of a 3 degrees of freedom robotic arm, aiming to simulate an end effector positioning task, consistent on a gripper. In first place, the assembly will be design using the program SolidWorks, and subsequently different kinematic modeling techniques will be applied to be able to implement a kinematic control task. The robot programming will be based on Arduino IDE, particularly an Arduino UNO R3 will be used to control the robot movements and the remote drive of the gripper.

Key words: Robotic arm, modeling, cinematic control, Arduino IDE.

Índice general

DOCUMENTO MEMORIA

1	INTRODUCCIÓN	19
1.1	CONTEXTO.....	19
1.2	POSIBLES APLICACIONES.....	19
1.3	REVISIÓN DE DEFINICIONES Y CONCEPTOS	19
2	OBJETIVO	23
3	SOLUCIONES ALTERNATIVAS	23
4	DESCRIPCIÓN DETALLADA DEL MODELO	25
5	JUSTIFICACIÓN DE COMPONENTES	28
5.1	BASE.....	28
5.2	HOMBRO	30
5.3	BRAZO	33
5.4	ANTEBRAZO	35
5.5	PINZA.....	38
5.6	TRASMISIONES POR CORREA Y RUDEAS DENTADAS	39
6	ELECTRÓNICA	48
7	CINEMÁTICA.....	50
7.1	MODELO CINEMÁTICO DIRECTO.....	50
7.2	MODELO CINEMÁTICO INVERSO.....	58
8	PROGRAMACIÓN Y ANÁLISIS.....	64
9	RESULTADOS Y CONCLUSIONES	88
10	LÍNEAS FUTURAS DE INVESTIGACIÓN.....	89
11	BIBLIOGRAFÍA	89
	ANEXO I - Hojas técnicas de los motores.....	92
	ANEXO II – Obtención y cálculo de presupuestos.....	97

DOCUMENTO PLANOS

1.	ENSAMBLAJE ROBOT.....	101
1.1.	ENSAMBLAJE BASE.....	102
1.1.1.	BASE INFERIOR	103

1.1.2.	BASE SUPERIOR.....	104
1.1.3.	TROCO DE LA BASE.....	105
1.2.	ENSAMBLAJE HOMBRO	106
1.2.1.	POLEA HOMBRO	107
1.2.2.	SEPARADOR POLEA HOBRO	108
1.2.3.	BASE HOMBRO.....	109
1.2.4.	COBERTURA HOMBRO	110
1.2.5.	SOPORTE MOTOR.....	111
1.2.6.	ADAPTADOR POLEA.....	112
1.3.	ENSAMBLAJE BRAZO.....	113
1.3.1.	POLEA BRAZO.....	114
1.3.2.	SEPARADOR POLEA BRAZO	115
1.3.3.	BASE BRAZO	116
1.3.4.	SEPARADOR.....	117
1.3.5.	COBERTURA BRAZO	118
1.4.	ENSAMBLAJE ANTEBRAZO	119
1.4.1.	ANTEBRAZO PRINCIPAL	120
1.4.2.	ACOPLE EJE PRINCIPAL.....	122
1.4.3.	ADAPTADOR MOTOR.....	123
1.4.4.	CONEXIÓN MUÑECA	124
1.4.5.	POLEA ANTEBRAZO	125
1.4.6.	ACOPLE EJE SECUNDARIO	126
1.5.	ENSAMBLAJE PINZA.....	127
1.5.1.	SOPORTE PINZA	128
1.5.2.	PINZA.....	129
1.5.3.	ENGRANAJE 1	130
1.5.4.	ENGRANAJE 2	131
1.5.5.	FIJADOR PINZA.....	132

DOCUMENTO PLIEGO DE CONDICIONES TÉCNICAS

1	OBJETO.....	135
2	CONDICIONES DE LOS MATERIALES.....	135
2.1	ENSAMBLAJE BASE.....	135
2.2	ENSAMBLAJE HOMBRO	136

2.3	ENSAMBLAJE BRAZO.....	137
2.4	ENSAMBLAJE ANTEBRAZO	137
2.5	ENSAMBLAJE PINZA.....	138
3	CONDICIONES DE EJECUCIÓN	139
3.1	ENSAMBLAJE BASE.....	139
3.2	ENSAMBLAJE HOMBRO	139
3.3	ENSAMBLAJE BRAZO.....	140
3.4	ENSAMBLAJE ANTEBRAZO	140
3.5	ENSAMBLAJE PINZA.....	141
3.6	MONTAJE.....	141
4	PRUEBAS Y AJUSTES FINALES DE SERVICIO	145

DOCUMENTO PRESUPUESTO

1	MATERIALES	148
2	MANO DE OBRA	149
3	EQUIPO INDUSTRIAL	149
4	ENSAMLAJE BASE.....	150
4.1	BASE INFERIOR.....	150
4.2	BASE SUPERIOR.....	151
4.3	TRONCO BASE	151
5	ENSAMBLAJE HOMBRO.....	152
5.1	POLEA HOMBRO.....	153
5.2	SEPARADOR POLEA HOMBRO	153
5.3	BASE HOMBRO	153
5.4	COBERTURA HOMBRO.....	154
5.5	SOPORTE MOTOR.....	154
5.6	ADAPTADOR POLEA	154
6	ENSAMBLAJE BRAZO.....	155
6.1	POLEA BRAZO.....	156
6.2	SEPARADOR POLEA BRAZO	156
6.3	BASE BRAZO	156
6.4	SEPARADOR.....	157
6.5	COBERTURA BRAZO.....	157
7	ENSAMBLAJE ANTEBRAZO.....	158

7.1	ANTEBRAZO PRINCIPAL.....	159
7.2	ACOPLE EJE PRINCIPAL.....	159
7.3	ADAPTADOR MOTOR.....	159
7.4	CONEXIÓN MUÑECA.....	160
7.5	POLEA ANTEBRAZO.....	160
7.6	ACOPLE EJE SECUNDARIO.....	160
8	ENSAMBLAJE PINZA.....	161
8.1	SOPORTE PINZA.....	161
8.2	PINZA.....	162
8.3	ENGRANAJE 1.....	162
8.4	ENGRANAJE 2.....	162
8.5	FIJADOR PINZA.....	163
9	COSTE TOTAL.....	164

Índice de figuras

Figura 1. Esquema de un robot cartesiano y su movimiento articular. Fuente: UdeSantiagoVirtual [22].	20
Figura 2. Esquema de un robot cilíndrico y su movimiento articular. Fuente: community.fortunecity.ws [2].	20
Figura 3. Esquema de un robot esférico y su movimiento articular. Fuente: Ingeniería de Sistemas y Automática [13].	21
Figura 4. Esquema de un robot angular y su movimiento articular. Fuente: UdeSantiagoVirtual [23].	21
Figura 5. Esquema de un robot SCARA, sus ejes y sistemas de referencia y su movimiento articular. Fuente: ResearchGate [17].	21
Figura 6. Esquema de un robot paralelo y sus dimensiones y eslabones principales. Fuente: ResearchGate [16].	22
Figura 7. Ensamblaje del brazo en posición en reposo – Vista Isométrica. Fuente: Elaboración Propia.	26
Figura 8. Ensamblaje del brazo en posición en reposo – Vista Perfil Derecho. Fuente: Elaboración Propia.	26
Figura 9. Ensamblaje del brazo en posición en reposo – Vista Perfil Izquierdo. Fuente: Elaboración propia.	27
Figura 10. Diagrama de bloques por subensamblajes y componentes de cada subensamblaje. Fuente: Elaboración propia.	27
Figura 11. Conjunto base diferenciando sus 3 piezas principales, base inferior (rojo), tronco de la base (azul) y base superior (verde), así como el posicionamiento del motor. Fuente: elaboración propia.	28
Figura 12. Vista de sección de la base y posicionamiento de los rodamientos y el eje. Modelo 51104 parte superior y 6002-2RSH en la inferior. Fuente: elaboración propia.	29
Figura 13. Conjunto hombro diferenciando sus 4 piezas principales, base del hombro (azul), soporte del motor (amarillo), rueda dentada (rojo) y cubierta del hombro (verde), así como el posicionamiento del motor. Fuente: elaboración propia.	30
Figura 14. Vista trasera del hombro, se observa la cobertura en detalle, el posicionamiento del eje y el sistema de transmisión. Fuente: Elaboración propia	31
Figura 15. Eje adaptado para la polea HTD de 16 dientes con acople para el motor NEMA 17 con tren planetario 5:1. Fuente: elaboración propia.	32

Figura 16. Conjunto brazo diferenciando sus 4 piezas principales, base del brazo (azul), cobertura del brazo (amarillo), polea brazo (verde) y separador (rojo), así como el posicionamiento del motor. Fuente: elaboración propia.....	33
Figura 17. Vista trasera del conjunto del brazo, sin la cobertura, permitiendo ver los componentes del sistema de transmisión. Fuente: Elaboración propia.	34
Figura 18. Conjunto antebrazo diferenciando sus 4 piezas, antebrazo principal (azul), rueda dentada (amarillo), conexión muñeca (verde) y adaptadores motores (rojo), así como el posicionamiento de los motores. Fuente: elaboración propia.	35
Figura 19. Sección del antebrazo, muestra la varilla roscada y su unión al servomotor (amarillo) y a la conexión con la muñeca. Fuente: Elaboración propia.	36
Figura 20. Conexión muñeca y sistema de unión entre servomotor y varilla roscada (pieza amarilla). Fuente: Elaboración propia.....	36
Figura 21. Pieza de acople de la varilla roscada con el servomotor DS04-NFC. Fuente: Elaboración propia.	37
Figura 22. Pieza de acople de la varilla roscada con el servomotor MG90D. Fuente: Elaboración propia.....	37
Figura 23. Conjunto pinza diferenciando sus 3 partes principales, las pinzas (rojo), el sistema de transmisión de movimiento (verde) y el soporte de la pinza (amarillo), así como el posicionamiento del motor. Fuente: Elaboración propia.....	38
Figura 24. Dimensiones de la correa GT2. Fuente: Brico Geek, Correa dentada GT2 – 6mm.	39
Figura 25. Distancia horizontal entre ejes de las poleas de la base. Fuente: elaboración propia.	41
Figura 26. Distancia vertical entre ejes de las poleas del codo. Fuente: elaboración propia. ..	41
Figura 27. Esquema que permite la deducción de la ecuación trigonométrica. Fuente: BIRT LH, Correas trapeciales [5].....	42
Figura 28. Trasmisión del codo, polea menor y poleas encargadas de aumentar el ángulo de agarre. Fuente: Elaboración propia.....	42
Figura 29. Esquema gráfico de la trasmisión del codo junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.	43
Figura 30. Esquema gráfico de la trasmisión de la base junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.	44
Figura 31. Dimensiones de las correas HTD (caso particular HTD M3). Fuente: Talleres auxiliares del motor, Selección del paso de la correa dentada [20].....	44
Figura 32. Distancia horizontal entre ejes de las poleas del hombro. Fuente: elaboración propia.	46

Figura 33. Trasmisión del hombro, polea menor y rodamientos encargados de aumentar el ángulo de agarre. Fuente: Elaboración propia.	46
Figura 34. Esquema gráfico de la trasmisión del hombro junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.	47
Figura 35. Conexionado y componentes electrónicos del brazo robot. Fuente: Elaboración propia.	48
Figura 36. Representación de ejes articulares (rojo), eslabones del brazo (naranja y verde) y sentido de giro de las articulaciones. Fuente: Elaboración propia.	50
Figura 37. Representación de ejes locales y distancias articulares para la obtención de la tabla D-H. Fuente: Elaboración propia.	51
Figura 38. Base diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	53
Figura 39. Hombro diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	54
Figura 40. Brazo diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	54
Figura 41. Antebrazo diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	55
Figura 42. Muñeca diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	55
Figura 43. Pinza diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.	56
Figura 44. Ensamblaje del robot en RobotScene con sus sistemas de referencia, la tabla D-H y su posición de reposo. Fuente: Elaboración propia.	57
Figura 45. Comparativa de los sistemas de referencia obtenidos. Fuente: Elaboración propia.	57
Figura 46. Representación del modelo geométrico del robot junto a sus incógnitas (azul), parámetros D-H (rojo) y elementos auxiliares (verde). Fuente: Elaboración propia.	59
Figura 47. Valores y posición de las articulaciones del robot seleccionados para la verificación. Fuente: Elaboración propia.	62
Figura 48. Inicio del programa de Trazado de triángulo en el Monitor Serial (arriba) y vista del programa tras la introducción de las coordenadas del punto 1 por parte del usuario (abajo). Fuente: Elaboración propia.	84
Figura 49. Definición gráfica del cálculo vectorial para la obtención de los movimientos articulares para realizar la trayectoria Punto 1 – Punto 2. Fuente: Elaboración propia.	85

Figura 50. Ejemplo de perfil de velocidad rectangular. Fuente: elaboración propia.	86
Figura 51. Ejemplo de perfil de velocidad trapezoidal. Fuente: elaboración propia.....	86
Figura 52. Ejemplo de perfil de velocidad escalonado (coincidente con el empleado en el programa de trazado de triángulo). Fuente: elaboración propia.	87
Figura 53. Conexión de un motor paso a paso con el módulo L298n, lugar de conexión redondeado en amarillo. Fuente: Elaboración propia.....	142
Figura 54. Esquema de conexionado entre los módulos L298n (dispositivos rojos) y la placa Arduino UNO (dispositivo azul), puntos de conexión redondeados en amarillo. Fuente: Elaboración propia.	143
Figura 55. Conexión entre servomotores y Arduino UNO (cable amarillo). Fuente: Elaboración propia.....	143
Figura 56. Conexiones entre alimentación de 5V de Arduino (cable gris), microservos (cables rojos) y módulos L298n (cables grises), puntos de conexión redondeados en amarillo. Fuente: Elaboración propia.	144
Figura 57. Esquema de conexionado de tierra representado con los cables negros y conexionado de la alimentación de los motores con los cables rojos. Zonas de conexión con la placa de conexionado redondeadas en amarillo. Fuente: Elaboración propia.....	144

Índice de tablas

Tabla 1. Matriz de decisión y resultados de las alternativas del proyecto.	24
Tabla 2. Características de las transmisiones base, hombro y codo. Fuente: Elaboración propia.	47
Tabla 3. Parámetros de la matriz D-H.	51
Tabla 4. Posibles configuraciones de las articulaciones.	61
Tabla 5. Rangos articulares. Fuente: Elaboración propia.	68
Tabla 6. Lista de comprobaciones previas al uso del brazo robot, referentes a la estructura. Fuente: Elaboración propia.	145

Índice de códigos

Código 1. Código comentado del programa de comprobación de máximo rango de los motores paso a paso. Fuente: Elaboración propia.	65
Código 2. Código comentado del programa de comprobación de máximo rango del servomotor de giro continuo de 360°. Fuente: Elaboración propia.	66
Código 3. Código comentado del programa de comprobación de máximo rango de servomotor de giro de 180°. Fuente: Elaboración propia.	67
Código 4. Código comentado del programa de recorrido articular, sección de definición de variables y Void setup(). Fuente: Elaboración propia.	68
Código 5. Código comentado del programa de recorrido articular, sección de Void loop(). Fuente: Elaboración propia.	69
Código 6. Ejemplo del sistema de variables de control en el caso del menú principal. Fuente: Elaboración propia.	70
Código 7. Código comentado del programa de Trazado de triángulo, sección de definición de variables, Void setup() y Menú del punto 1. Fuente: Elaboración propia.	72
Código 8. Código comentado del programa de recorrido articular, sección Menú del punto 1 y Menú del punto 2. Fuente: Elaboración propia.	73
Código 9. Código comentado del programa de recorrido articular, sección de Menú del punto 2, Menú del punto 3 y Cálculos. Fuente: Elaboración propia.	74
Código 10. Código comentado del programa de recorrido articular, sección de Cálculos. Fuente: Elaboración propia.	75
Código 11. Código comentado del programa de recorrido articular, sección de Movimiento al punto 1. Fuente: Elaboración propia.	76
Código 12. Código comentado del programa de recorrido articular, sección de Movimiento al punto 1 y Movimiento al punto 2. Fuente: Elaboración propia.	77
Código 13. Código comentado del programa de recorrido articular, sección de Movimiento al punto 2. Fuente: Elaboración propia.	78
Código 14. Código comentado del programa de recorrido articular, sección de Movimiento al punto 2 y Movimiento al punto 3. Fuente: Elaboración propia.	79
Código 15. Código comentado del programa de recorrido articular, sección de Movimiento al punto 3. Fuente: Elaboración propia.	80
Código 16. Código comentado del programa de recorrido articular, sección de Movimiento al punto 3 y Movimiento al reposo. Fuente: Elaboración propia.	81

Código 17. Código comentado del programa de recorrido articular, sección de Movimiento al reposo. Fuente: Elaboración propia.	82
Código 18. Código comentado del programa de recorrido articular, sección de Reset de variables. Fuente: Elaboración propia.	83
Código 19. Sistema de lectura y guardado de datos introducidos por el usuario en el Monitor Serial. Fuente: elaboración propia.....	85



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

DOCUMENTO MEMORIA

Índice documento memoria

1	INTRODUCCIÓN.....	19
1.1	CONTEXTO.....	19
1.2	POSIBLES APLICACIONES.....	19
1.3	REVISIÓN DE DEFINICIONES Y CONCEPTOS.....	19
2	OBJETIVO.....	23
3	SOLUCIONES ALTERNATIVAS.....	23
4	DESCRIPCIÓN DETALLADA DEL MODELO.....	25
5	JUSTIFICACIÓN DE COMPONENTES.....	28
5.1	BASE.....	28
5.2	HOMBRO.....	30
5.3	BRAZO.....	33
5.4	ANTEBRAZO.....	35
5.5	PINZA.....	38
5.6	TRASMISIONES POR CORREA Y RUDEAS DENTADAS.....	39
6	ELECTRÓNICA.....	48
7	CINEMÁTICA.....	50
7.1	MODELO CINEMÁTICO DIRECTO.....	50
7.2	MODELO CINEMÁTICO INVERSO.....	58
8	PROGRAMACIÓN Y ANÁLISIS.....	64
9	RESULTADOS Y CONCLUSIONES.....	88
10	LÍNEAS FUTURAS DE INVESTIGACIÓN.....	89
11	BIBLIOGRAFÍA.....	89
	ANEXO I - Hojas técnicas de los motores.....	92
	ANEXO II – Obtención y cálculo de presupuestos.....	97

1 INTRODUCCIÓN

1.1 CONTEXTO

A lo largo de este proyecto se aborda una de las áreas más punteras de la tecnología en la actualidad, la robótica.

Siguiendo la información aportada en [27], puede definirse la robótica como la ciencia que estudia el uso, diseño y aplicación de los robots, un término que aparece por primera vez en 1921, en las escrituras del filósofo austrohúngaro Karel Čapek, y que ha ido desarrollándose a lo largo de la historia hasta alcanzar la concepción por la que lo conocemos hoy, una máquina o entidad virtual programable capaz de realizar diversas operaciones o tareas con un propósito definido por el programador.

Existen un gran número de tipos de robots, como puede observarse en [1], entre los que pueden destacarse robots industriales, robots de investigación, robots destinados a medicina, robots de uso militar, etc. En este proyecto, se estudiará un robot perteneciente al primer grupo nombrado, robots industriales.

Hoy en día, la aplicación de la robótica en la industria es algo muy común, ya que el uso de robots en ciertas aplicaciones simplifica la producción y la hace más eficiente, en términos económicos y de tiempo. Algunas de las aplicaciones entre las que puede emplearse la robótica industrial son la manipulación de objetos (de distintos pesos, tamaños, materiales y formas), tareas de montaje, tareas específicas en cadenas de producción, soldaduras, etc. Además, permite la realización de acciones en ambientes peligrosos o nocivos para el ser humano.

1.2 POSIBLES APLICACIONES

Este proyecto podría emplearse como base para el desarrollo de prototipos de mayor tamaño de brazos robots manipuladores para la industria, teniendo en cuenta que los cálculos realizados y la programación podría trasladarse, realizando cambios poco significativos, a otros modelos de brazos de 5gdl. También podría emplearse el modelo como brazo manipulador de pequeños objetos en cualquier cadena de producción, o como brazo asistente en ciertas tareas de investigación, ya que su tamaño y peso no suponen ningún peligro para su uso cerca de personas.

Otra posibilidad entre las aplicaciones podría ser la de robot educacional. Al tratarse de un modelo que engloba una gran cantidad de aplicaciones de asignaturas vistas a lo largo de la carrera de Ingeniería Mecánica, como la propia robótica, el diseño asistido por ordenador, aplicaciones electrónicas o diseño de máquinas sería de gran utilidad para mostrar una aplicación práctica de todas estas materias. Del mismo modo, podría emplearse para mostrar a distintos estudiantes el proceso de ensamblaje de la maqueta y su posterior estudio cinemático y programación, partiendo del programa base implementado en el controlador Arduino, encargado de manejar el robot.

1.3 REVISIÓN DE DEFINICIONES Y CONCEPTOS

Brazo robot

Un brazo robot consiste en un tipo de robot capaz de realizar operaciones y procesos similares a los que haría un brazo humano, como tareas de manipulación, o las de un brazo humano con una herramienta, como procesos de soldadura, atornillado, etc.

Como se especifica en [26], se trata de un componente mecánico programable, incluido dentro de la categoría de robots de base estática. Estos robots están formados por diversos componentes unidos entre sí por juntas llamadas articulaciones. Estas articulaciones pueden ser de tipo rotacional (pares R) o de tipo prismático (pares P). El número de articulaciones define el número de grados de libertad del robot, y en función del tipo de articulaciones, los robots de base estática pueden clasificarse en 6 categorías principales.

- Cartesianos: Aquellos cuyas articulaciones son únicamente prismáticas.

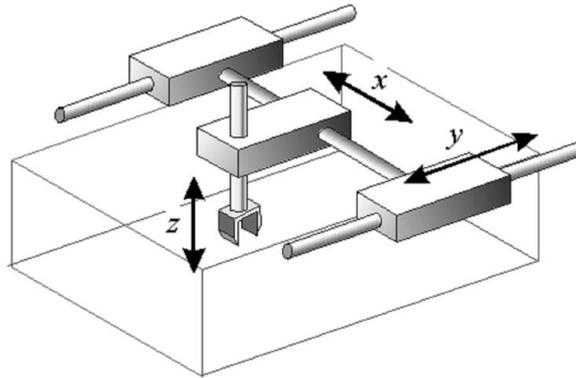


Figura 1. Esquema de un robot cartesiano y su movimiento articular. Fuente: UdeSantiagoVirtual [22].

- Cilíndricos: Presentan una articulación rotacional en la base y dos articulaciones prismáticas.

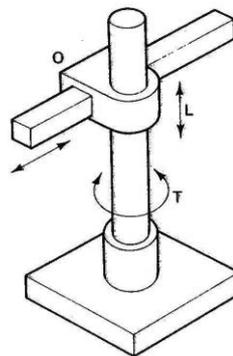


Figura 2. Esquema de un robot cilíndrico y su movimiento articular. Fuente: community.fortunecity.ws [2].

- Esféricos o polares: Presentan 2 articulaciones rotacionales, situadas en sus dos primeras articulaciones, y una tercera prismática.

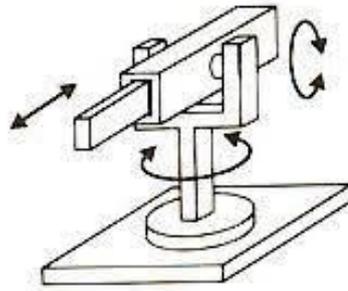


Figura 3. Esquema de un robot esférico y su movimiento articular. Fuente: Ingeniería de Sistemas y Automática [13].

- Angulares: Todas sus articulaciones son rotacionales.

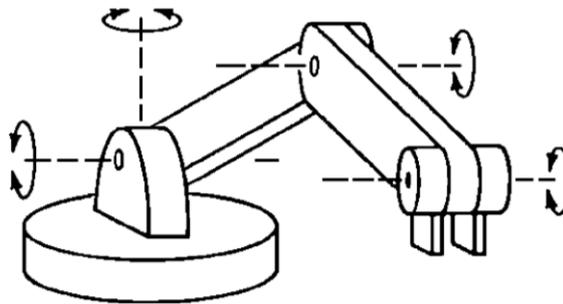


Figura 4. Esquema de un robot angular y su movimiento articular. Fuente: UdeSantiagoVirtual [23].

- SCARA: Presenta 3 articulaciones rotacionales, situadas en las tres primeras, y una cuarta prismática.

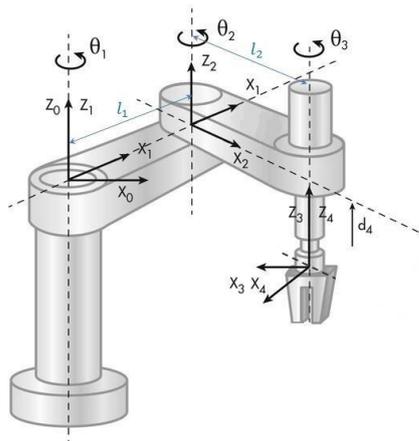


Figura 5. Esquema de un robot SCARA, sus ejes y sistemas de referencia y su movimiento articular. Fuente: ResearchGate [17].

- Paralelos o deltas: formados por una configuración de 3 articulaciones rotacionales y prismáticas, llamada articulación de paralelogramo, que permiten a su órgano terminal mantenerse en todo momento paralelo a la base.

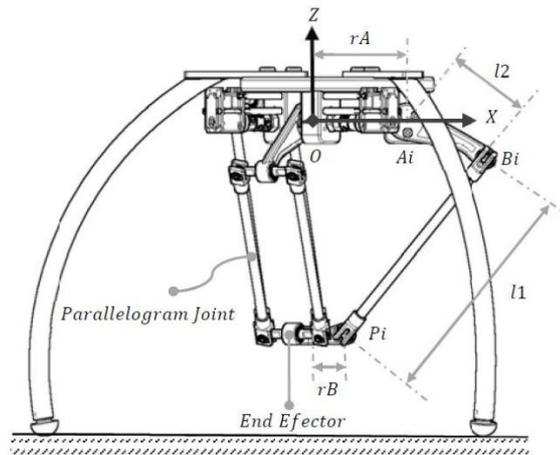


Figura 6. Esquema de un robot paralelo y sus dimensiones y eslabones principales. Fuente: ResearchGate [16].

Los brazos robots pertenecen a la categoría de angulares ya que, como norma general, sus articulaciones son de tipo rotacional. Dentro de los brazos robóticos, los más empleados son los brazos antropomorfos o de 6gdl, ya que su configuración permite al robot abarcar todo el espacio tridimensional dentro de su entorno de trabajo. Este tipo de brazos son empleados principalmente en la industria, aunque también pueden emplearse en otro tipo de actividades como investigación o educación.

Arduino

Arduino [3] es un plataforma libre y abierta basada en un hardware y software fácil de usar. Para ello, Arduino implementa un lenguaje de programación propio, basado en Wiring, una plataforma de programación de microcontroladores basada en lenguaje C/C++ pensada para programadores, y un software llamado Arduino Software (IDE), basado en Processing, un entorno de código abierto que permite importar librerías y ejemplos de programas disponibles en la web.

Gracias a la implementación de estas dos herramientas, Arduino logra generar un entorno de programación intuitivo y sencillo para el autoaprendizaje en el que el código puede simplificarse en gran medida gracias a las distintas librerías disponibles y programas creados por la comunidad. Cabe remarcar que esta es la mayor ventaja de Arduino sobre otras plataformas y microcontroladores, la comunidad activa existente de miles de estudiantes, profesores, investigadores y programadores que son usuarios de Arduino y que colaboran en la red generando programas y librerías, respondiendo dudas y haciendo tutoriales en webs, foros y plataformas de vídeo.

Además, Arduino presenta una gama de productos de hardware compatibles entre ellos que simplifican en gran medida los montajes y reducen los costes, al ser, en su gran mayoría, componentes reutilizables. Entre estos productos pueden destacarse distintos microcontroladores, como Arduino UNO, Arduino Mega o Arduino Nano, y también sensores de todo tipo.

Robótica en Arduino

Existen cientos de proyectos de robótica basados en Arduino, y realizados por su comunidad, en Internet. Se pueden encontrar proyectos desde la fabricación hasta programación del robot para realizar una tarea determinada.

El acceso libre a estos proyectos permite tener una base de partida para nuevas ideas, así como generar proyectos de mayor complejidad basados en aplicaciones ya creadas. Es por esto por lo que la realización de trabajos relacionados con la robótica mediante Arduino supone una gran ventaja.

2 OBJETIVO

El objetivo final de este proyecto consiste en la programación de un brazo robot, plenamente operativo y de diseño propio, para la realización de una tarea de posicionamiento de su órgano terminal.

Para alcanzar dicha finalidad, el proyecto se divide en varios subobjetivos, de igual importancia, que van a desarrollarse y explicarse a lo largo de este documento y que se nombran a continuación:

- Diseñar mediante el programa informático SolidWorks un modelo de brazo robótico.
- Realizar el estudio cinemático para comprender el movimiento del robot.
- Montar el modelo, mediante partes adquiridas de terceros y/o partes diseñadas.
- Estudiar el rango articular, así como programar una tarea de posicionamiento del órgano terminal del brazo robótico.

3 SOLUCIONES ALTERNATIVAS

Las alternativas planteadas previas a la realización del proyecto giran en torno a la elección del tipo de robot que quiere diseñar, con el consecuente montaje y programación. Los motivos que justifican este enfoque a la hora de definir el concepto de proyecto a desarrollar se basan en que el modelo de robot diseñado será el que determine las características cinemáticas a estudiar, el modo de ensamblaje, el coste y la programación de la tarea final.

El método de elección del diseño se basará en un estudio mediante matriz de decisión de ciertas características fundamentales, determinadas por el director del proyecto, que el brazo robot debe cumplir. Estas características serán puntuadas en un rango definido a continuación.

- Grados de libertad: en esta característica se valora entre un 1 y un 4 la dificultad de realización del modelo en función de los grados de libertad (gdl) del brazo robótico, siendo 1 el valor asignado a un robot de 6gdl (más complicado de diseñar) y 4 el asignado a un robot de 3gdl (diseño sencillo) o adquirido de terceros.
- Valor académico: al tratarse de un proyecto TFG, se valora en gran medida la iniciativa del alumno a la hora de escoger y diseñar sus trabajos, de tal forma que cuanto mayor sea el reto autoimpuesto por el alumno, siempre que la realización sea la correcta, se tendrá en cuenta en sus calificaciones. Por ello esta característica tendrá un mayor peso y se valorará entre un 1, proyecto extremadamente simple, y un 10, proyecto extremadamente complejo.
- Coste del ensamblaje: el presupuesto disponible es limitado, por tanto, es conveniente tener esto en cuenta y valorar los precios de los posibles modelos a realizar. Por

ejemplo, un mayor número de grados de libertad implicará la necesidad de comprar un mayor número de motores, o la implementación de dichos motores directamente sobre las articulaciones implicarán la necesidad de motores de mayor potencia (más caros) que si se emplean métodos de transmisión por correa y poleas. Se valorará esta característica entre 1 (presupuesto mínimo de 50€) y 5 (presupuesto máximo de 500€).

- Tiempo: esta característica tiene en cuenta el tiempo de realización del proyecto, de forma que un diseño más sencillo en el que los grados de libertad son bajos o las articulaciones tienen los motores incorporados a los ejes requieren de un menor tiempo de diseño y montaje. Por tanto, se asignará un 1 a ensamblajes que cuesten más de 96 horas en diseñar y montar y un 5 a ensamblajes que cuesten menos de 5 horas.
- Consumo eléctrico: el consumo eléctrico del ensamblaje final puede llegar a suponer un problema, ya que será necesario realizar pruebas y verificaciones de programas implementados para cumplir con los objetivos impuestos. A mayor consumo, mayor será la instalación necesaria a la cual conectar el brazo y mayores problemas electrónicos pueden suceder a consecuencia. Por esto, se asignará un con un 1 a aquellas alternativas que requieran un consumo eléctrico de más de 20V y 3A y con un 5 a aquellos que requieran menos de 10V y 1A.

Las alternativas que se tuvieron en cuenta a la hora seleccionar un proyecto fueron:

1. Brazo robótico de 3gdl con los motores implementados en las articulaciones.
2. Brazo robótico de 5gdl con un sistema de transmisión por correas y ruedas dentadas.
3. Brazo robótico de 6gdl con un sistema de transmisión por correas y ruedas dentadas.
4. Brazo de 6gdl adquirido de terceros.

La matriz de decisión obtenida se muestra a continuación:

Alternativa	Grados de libertad	Valor académico	Coste del ensamblaje	Tiempo	Consumo eléctrico	Resultados
1	4	3	2	3	4	16
2	2	9	3	2	2	18
3	1	10	4	1	1	17
4	4	1	5	1	3	14

Tabla 1. Matriz de decisión y resultados de las alternativas del proyecto.

Como puede observarse en la **Tabla 1**, la alternativa escogida como solución final es la realización de un brazo robótico de 5gdl empleando como sistema de movimiento articular el uso de correas y ruedas dentadas accionadas por motores.

4 DESCRIPCIÓN DETALLADA DEL MODELO

El ensamblaje del brazo robótico de 5gdl presenta un total de 5 articulaciones rotacionales, clasificándolo como robot angular, y un órgano terminal de tipo pinza, atribuyéndole la función de robot manipulador. Las dimensiones generales del robot, en la posición de reposo representada en las **Figuras 7 8 y 9**, son 440x340x180mm y tiene un peso de 1691g.

El ensamblaje está dividido en 5 subensamblajes, base, hombro, brazo, antebrazo y pinza, que aparecen desglosados en la **Figura 10**. La división permite un montaje individual de cada una de las partes simplificando el ensamblaje general. Además, los distintos subensamblajes están unidos entre sí mediante un tubo de aluminio de 15mm de diámetro exterior, para el caso de la base, el hombro y el brazo, mediante un eje de métrica 8, para el caso del brazo y el antebrazo, y mediante un eje de métrica 4 para el caso del antebrazo y la pinza.

El sistema de movimiento de las articulaciones del robot ser llevarán a cabo mediante motores paso a paso NEMA 17 y servomotores. Los motores NEMA 17 empleados son, 2 motores Bipolares de 45Ncm de par y 1 motor Bipolar de 2Nm con tren planetario de engranajes acoplado de relación 5:1. Respecto a los servomotores, se han empleado 1 servomotor DS04-NFC de 52Ncm de par y 2 servomotores MG90D de 24Ncm de par. Para los subensamblajes de la base, el hombro y el brazo se ha realizado, además, un sistema de transmisión por correas para aumentar la potencia de dichas articulaciones. Esto ha permitido también situar el centro de masa del robot bajo y centrado, manteniendo los tres motores NEMA cercanos a la base.

Con el objetivo de mejorar la fluidez del movimiento articular y de las correas de transmisión, se han introducido rodamientos 4 6002-2RSH, en base y hombro, 4 rodamientos 608-2RSH, en brazo y hombro, 3 rodamientos 624-2RS1, en el antebrazo, un rodamiento axial 51104 en la base, y un rodamiento 625-2RS1 en el antebrazo.

Para el control del brazo se ha empleado el Arduino UNO R3 como microcontrolador, debido a sus ventajas comentadas en el apartado *Introducción*. Así mismo, para coordinar el manejo de los motores paso a paso, se han empleado dispositivos L298n. Estos *drivers* permiten controlar la velocidad y el sentido de giro de los motores paso a paso, pudiendo así administrar el par de giro de las articulaciones Base, Hombro y Codo y permitiendo al brazo adaptarse de forma eficiente a las distintas posibles tareas que se desea que desempeñe.

Los motores empleados para operar el movimiento articular son dos NEMA 17 modelo 17HS16-2004S1, para las articulaciones de la base y el codo, un motor NEMA 17 con tren de engranajes planetario modelo 17HS15-1684S-PG5, para la articulación del hombro, dos servomotores MG90D, para la articulación de la muñeca y apertura y cierre de la pinza, y un servomotor DS04-NFC, para la rotación de la muñeca.

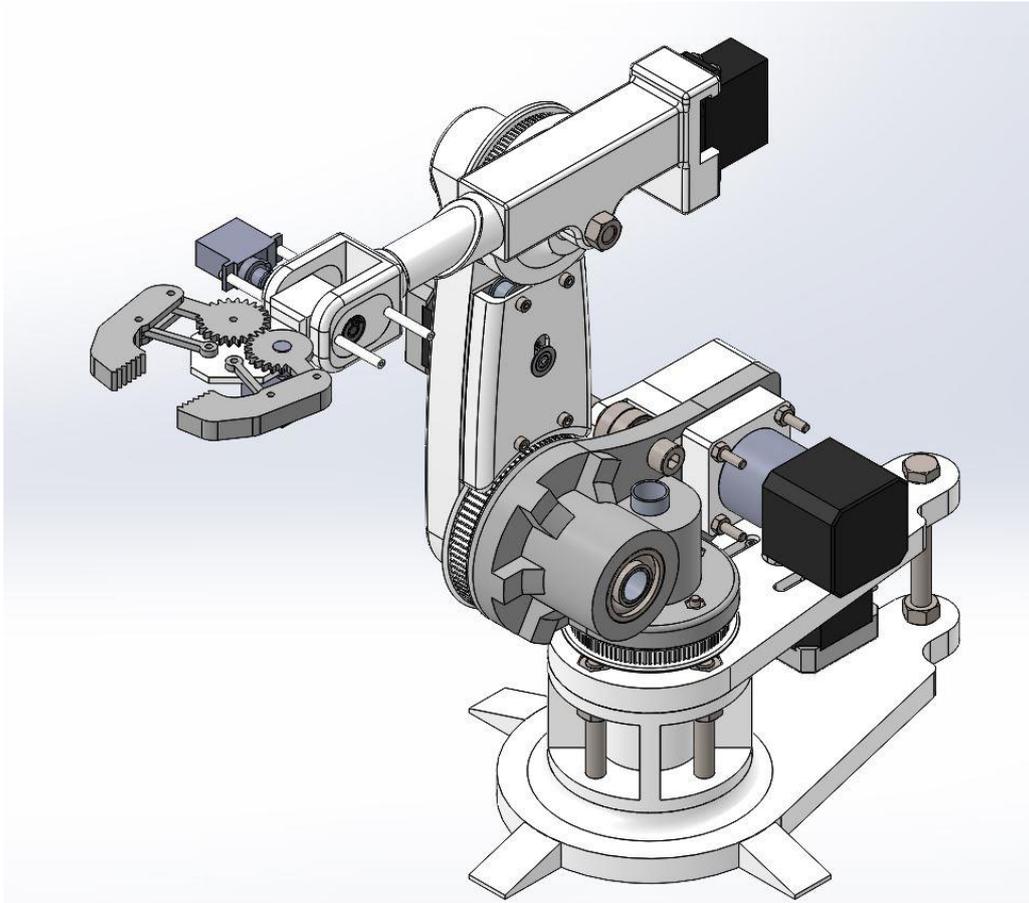


Figura 7. Ensamblaje del brazo en posición en reposo – Vista Isométrica. Fuente: Elaboración Propia.

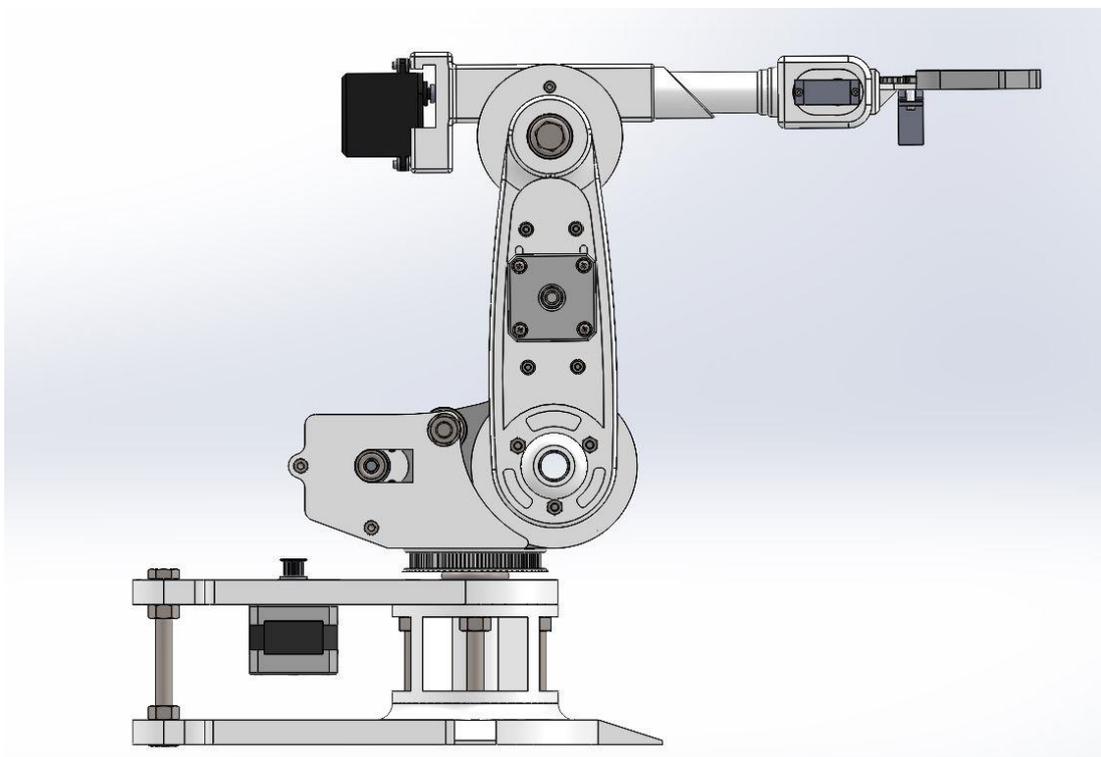


Figura 8. Ensamblaje del brazo en posición en reposo – Vista Perfil Derecho. Fuente: Elaboración Propia.

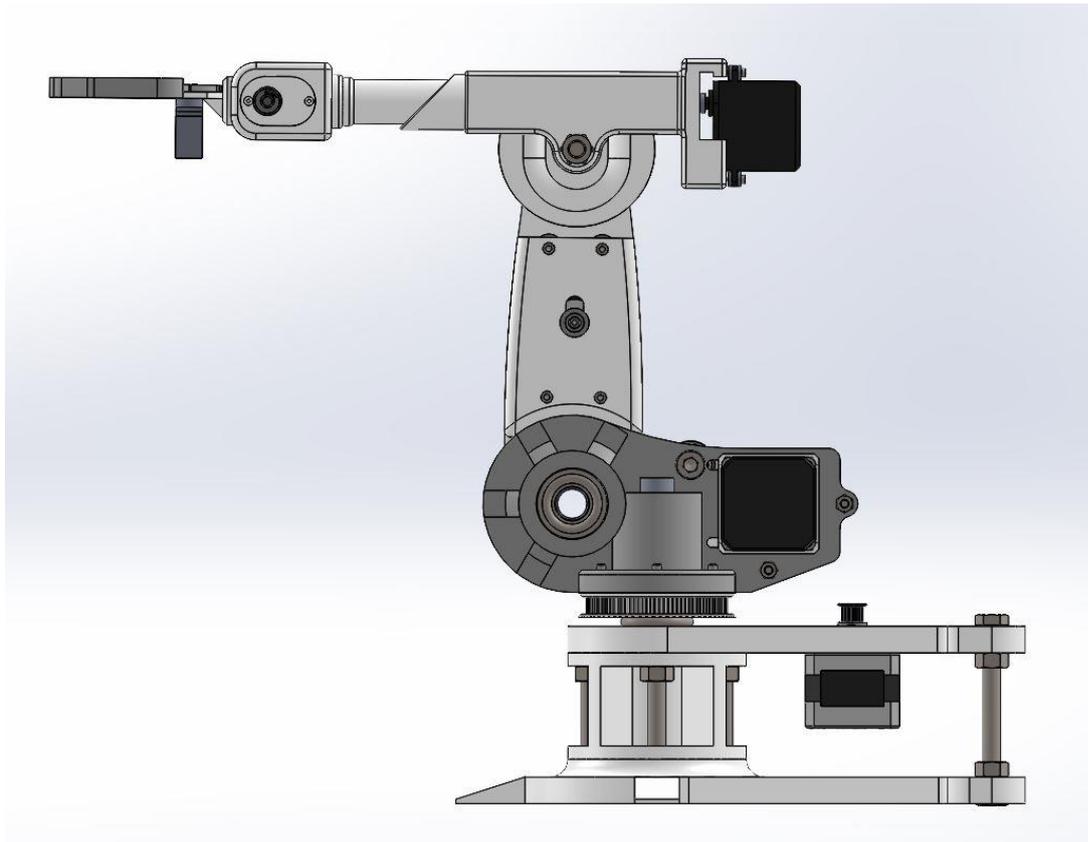


Figura 9. Ensamblaje del brazo en posición en reposo – Vista Perfil Izquierdo. Fuente: Elaboración propia.

Diagrama de bloques del proyecto

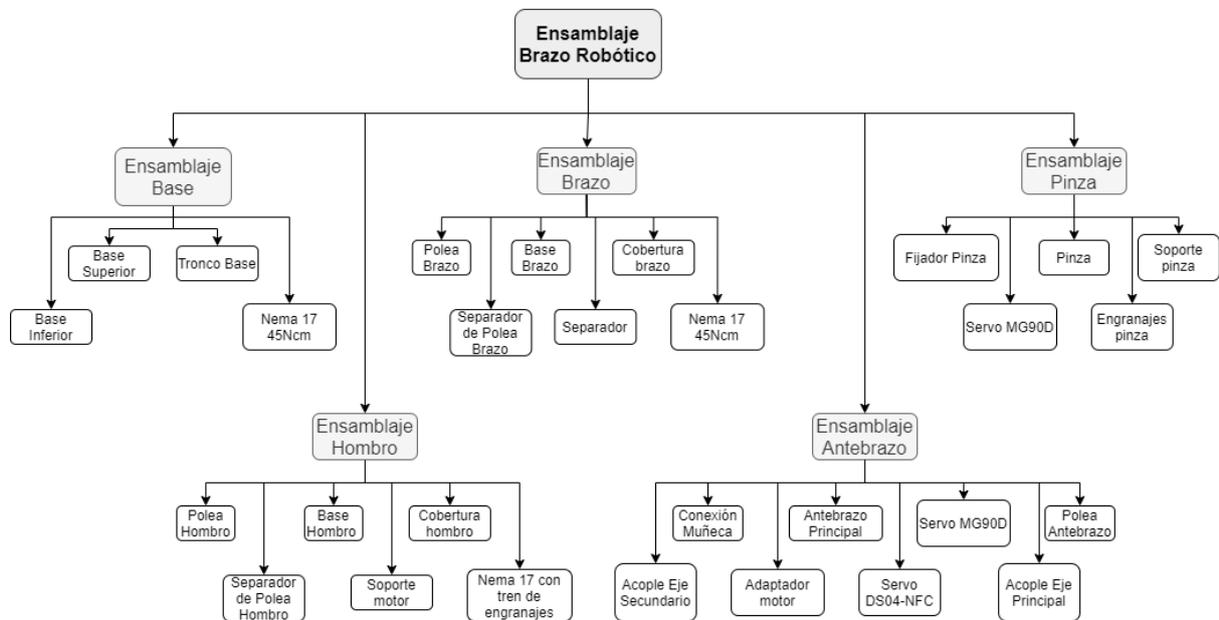


Figura 10. Diagrama de bloques por subensamblajes y componentes de cada subensamblaje. Fuente: Elaboración propia.

5 JUSTIFICACIÓN DE COMPONENTES

5.1 BASE

Para la base del robot, se busca un diseño estable, capaz de equilibrar el resto de los conjuntos durante el movimiento del robot. Además, en la base ha de acoplarse el primer motor, encargado de conferir al brazo un movimiento rotacional paralelo a la superficie de apoyo del robot.

Debido a esto, se optó por un diseño amplio y abierto, compuesto por 3 componentes principales, impresos en 3D con poliestireno de alto impacto (Z-HIPS) y relleno del 30%, y una zona adaptada para el acople del motor, tal y como se muestra en la **Figura 11**. El sistema de movimiento de la articulación se realiza por medio de una transmisión por correa y ruedas dentadas. A consecuencia de ello el posicionamiento del motor se encuentra ligeramente alejado del eje articular. Esto permite además equilibrar el conjunto, estableciendo el centro de masas del subensamblaje a 50mm del eje de la articulación de la base.

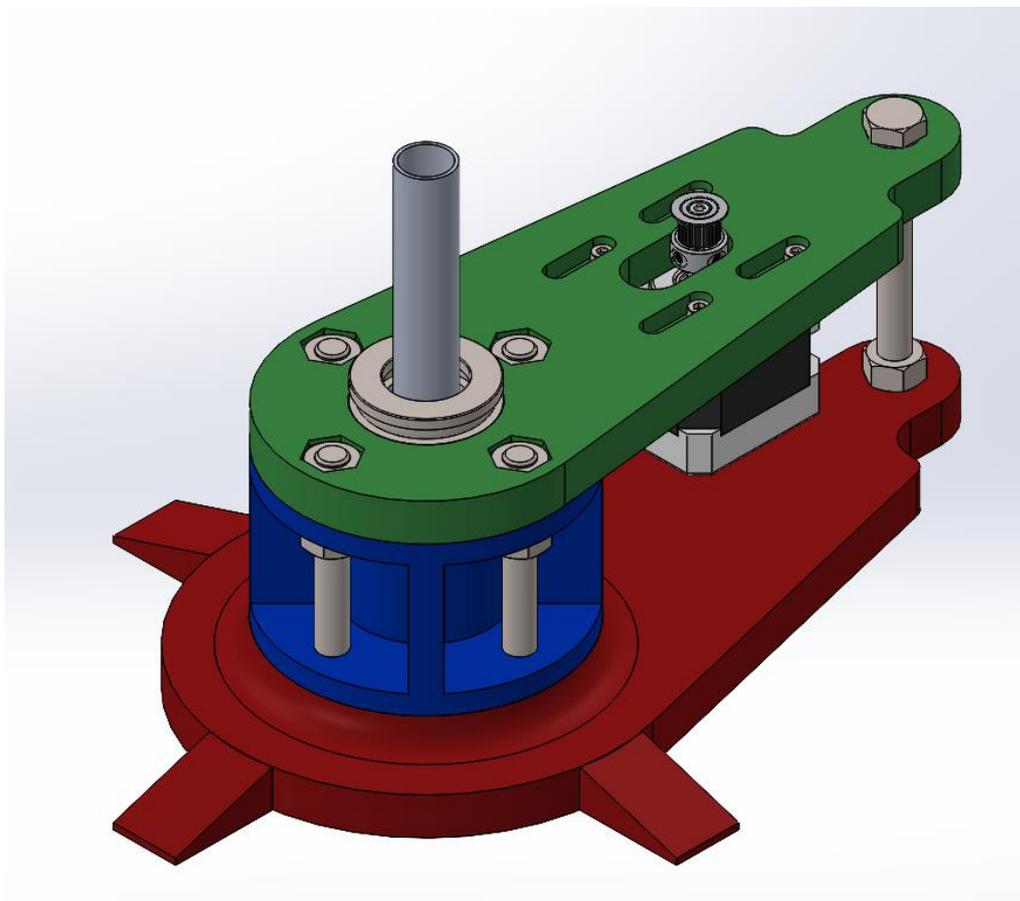


Figura 11. Conjunto base diferenciando sus 3 piezas principales, base inferior (rojo), tronco de la base (azul) y base superior (verde), así como el posicionamiento del motor. Fuente: elaboración propia.

Además del acople del motor, la base presenta un sistema de unión al conjunto del hombro mediante un tubo de aluminio de 15mm de diámetro externo. Este tubo debe poder rotar respecto a su eje para facilitar el movimiento del robot. Por ello, se han colocado 2 rodamientos de bola rígidos obturados rodantes de 15x32x9mm (modelo 6002-2RSH) para facilitar el giro del eje, y un rodamiento axial de bolas de 20x35x10mm (modelo 51104) para mejorar el giro

entre los conjuntos base-hombro. La disposición del eje y los rodamientos se muestra en la **Figura 12**.

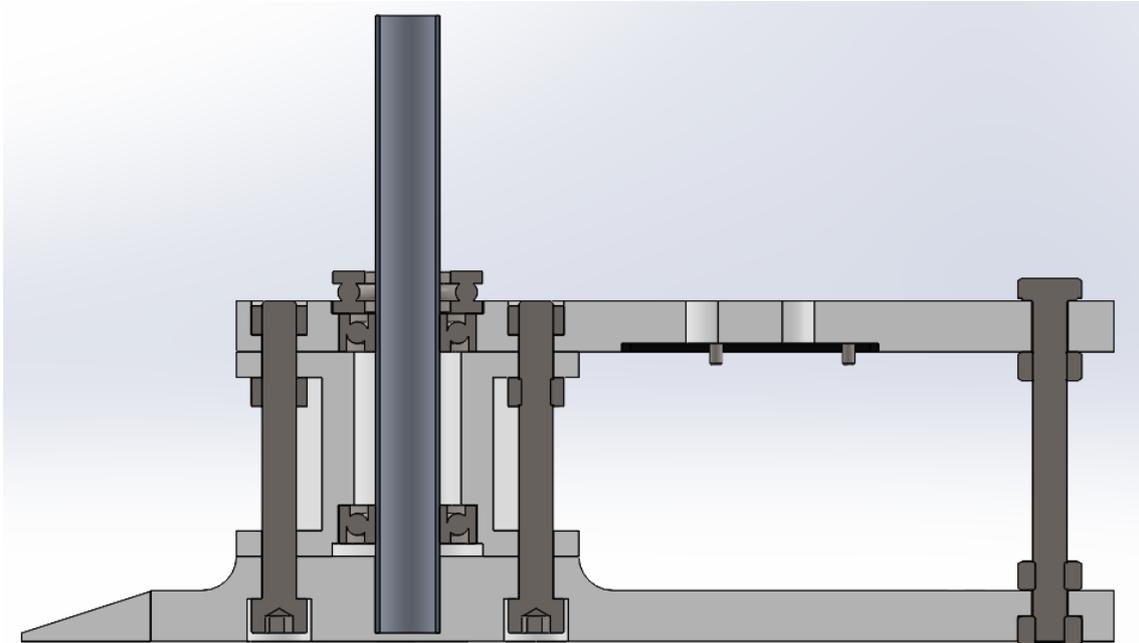


Figura 12. Vista de sección de la base y posicionamiento de los rodamientos y el eje. Modelo 51104 parte superior y 6002-2RSH en la inferior. Fuente: elaboración propia.

El motor empleado para el movimiento de la articulación de la base es un NEMA 17 de 45Ncm, bipolar de 2A y dimensiones 42x42x40mm (modelo 17HS16-2004S1). Fue seleccionado debido a sus prestaciones y con un par rotativo superior al necesario para evitar problemas futuros durante la realización de las tareas programadas. Su hoja de especificaciones técnicas y su curva de comportamiento de par de rotación en función de la velocidad y frecuencia aparecen en el **ANEXO I**.

Para transmitir el movimiento de rotación al conjunto del hombro se emplea un sistema de correa y ruedas dentadas adaptado para una correa cerrada GT2 de 6mm de ancho y 280mm de longitud. Las ruedas dentadas empleadas son una polea GT2 de 16 dientes ajustable al eje del NEMA 17 y una polea de 100 dientes y 63mm de diámetro (también adaptada a la correa GT2) unida al subensamblaje del hombro.

Las dimensiones generales del conjunto de la base son 260x185x85mm (sin tener en cuenta la altura del eje) y una masa total aproximada de 462g (contabilizando el motor y la tornillería). Si se desean conocer más detalles acerca de los componentes o el montaje del ensamblaje, consultar el **Documento Pliego de Condiciones**.

5.2 HOMBRO

El hombro es el conjunto encargado de levantar todo el peso del brazo, por tanto, se busca que tenga un diseño robusto, con un motor de elevada potencia y par de rotación. También se ha de tener en cuenta que, para mantener el equilibrio del robot, la mayor parte de su peso debe concentrarse en la parte contraria en la cual estará montado el resto del brazo.

Para cumplir con estas características, se ha realizado un diseño centrado en una pieza principal, la base del hombro, a la cual se acoplan el resto de las piezas, la rueda dentada para el movimiento articular de la base, el motor y la cobertura del hombro, encargada de proteger la transmisión por correa y rudas dentadas de la articulación del hombro. Todas las piezas diseñadas en este conjunto se han impreso en 3D con poliestireno de alto impacto (Z-HIPS) y relleno del 30%, a excepción del Separador Polea Hombro, fabricado con metacrilato de extrusión (PMMA-XT) mediante cortadura láser. Puede observarse la estructura del conjunto en las **Figura 13**.

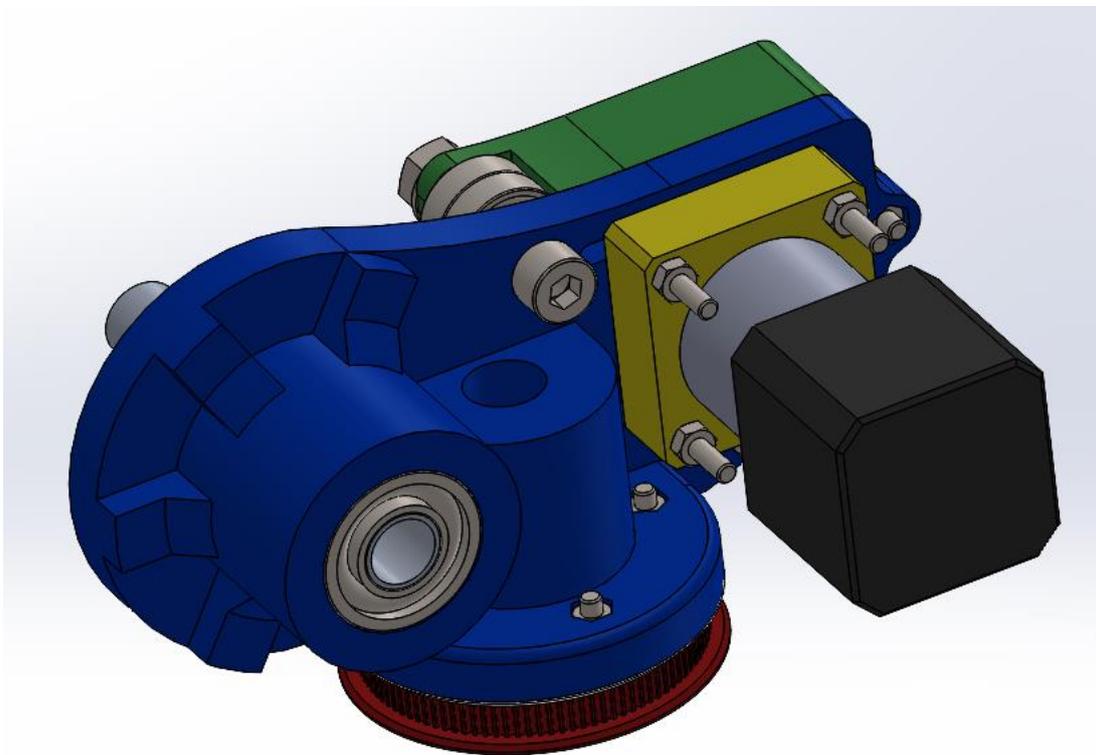


Figura 13. Conjunto hombro diferenciando sus 4 piezas principales, base del hombro (azul), soporte del motor (amarillo), rueda dentada (rojo) y cubierta del hombro (verde), así como el posicionamiento del motor. Fuente: elaboración propia.

El eje empleado para unir el hombro al resto de componentes es un tubo de aluminio de 15mm de diámetro externo, ayudado en la rotación por dos rodamientos 6002-2RSH (iguales a los empleados en el eje de la base).

El motor, es un NEMA 17 bipolar con un tren de engranajes planetarios de relación 5:1 incorporado, con par de rotación 4Nm, 1,68A de corriente de alimentación y unas dimensiones de 42x42x87,3mm (modelo 17HS15-1684S-PG5). En este caso también se seleccionó un motor con características y potencia superiores a las mínimas exigidas por el modelo para evitar problemas futuros en el rango articular o a la hora de la realización de tareas

programadas. Para consultar más información, se ha dejado su ficha técnica a disposición en el ANEXO I.

Con respecto al sistema de transmisión, se ha empleado nuevamente el conjunto correa – ruedas dentadas, con el objetivo de mejorar la estabilidad general del ensamblaje (centrando el centro de masas del conjunto) y para aumentar el par generado para levantar el resto del brazo. El sistema se ha implementado mediante una correa cerrada HTD de 11mm de grosor y 330mm de longitud, una polea HTD 3M16T de 16 dientes y 6 mm de diámetro y una rueda dentada de 80 dientes y 75mm de diámetro (perteneciente al conjunto del brazo), también adaptada para correa HTD de 11mm de grosor.

Para facilitar la transmisión, se ha hecho uso de dos rodamientos rígidos de bolas obturados rodantes de 8x22x7mm (modelo 608-2RSH), encargados de reducir el ángulo de agarre de la polea menor. Con el objetivo de alinear correctamente la polea HTD, se ha diseñado un eje adaptado al motor NEMA 17 con tren planetario, representado en la **Figura 15**, y un rodamiento rígido de bolas obturado rodante de 5x16x5mm (modelo 625-2RS1) que estará apoyado sobre la cobertura del hombro, tal como puede observarse en la **Figura 14**, encargado de reducir los esfuerzos de flexión del eje de la polea y mantenerlo alineado a la polea de 80 dientes.

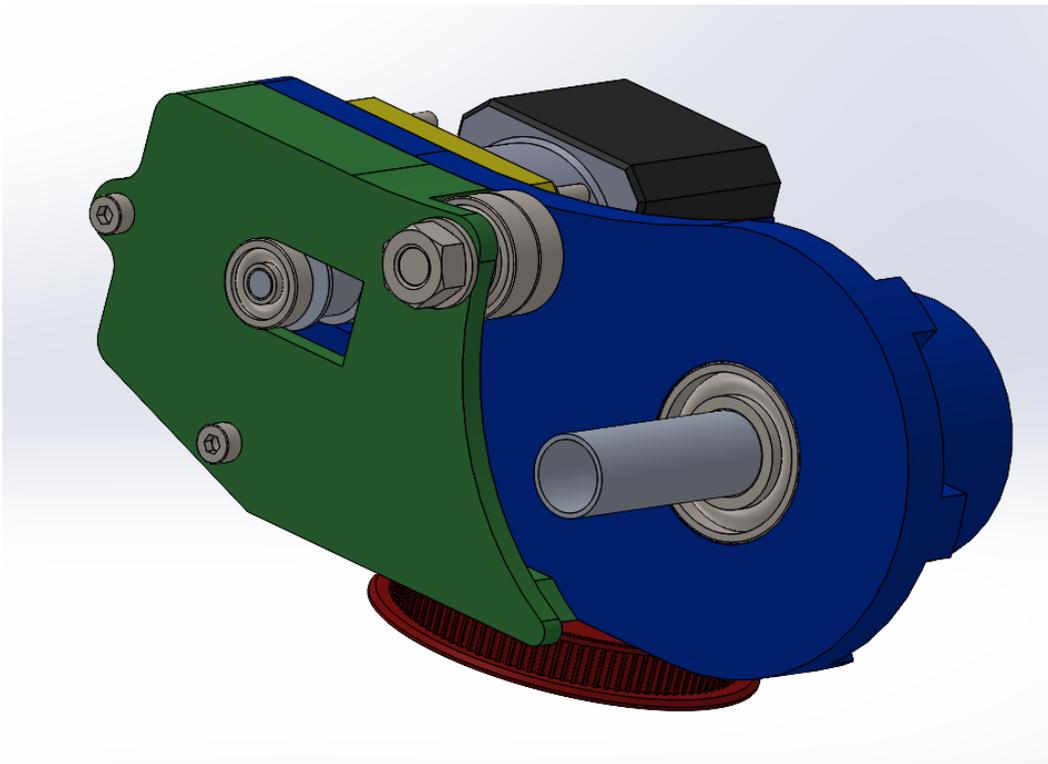


Figura 14. Vista trasera del hombro, se observa la cobertura en detalle, el posicionamiento del eje y el sistema de transmisión. Fuente: Elaboración propia

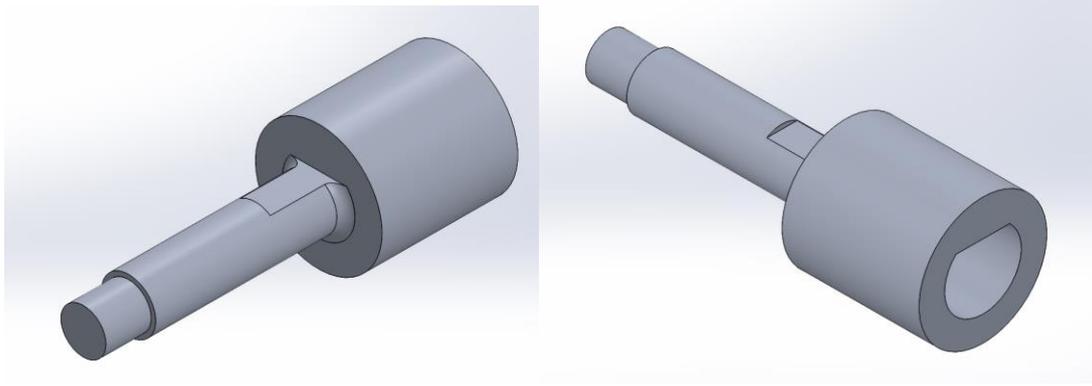


Figura 15. Eje adaptado para la polea HTD de 16 dientes con acople para el motor NEMA 17 con tren planetario 5:1. Fuente: elaboración propia.

Las dimensiones generales del conjunto del hombro son 168x121x91mm (sin tener en cuenta la longitud del eje) y una masa total aproximada de 626g (contabilizando el motor y la tornillería). Si se desean conocer más detalles acerca de los componentes o el montaje del ensamblaje, consultar el **Documento Pliego de Condiciones**.

5.3 BRAZO

El brazo debe ser un conjunto esbelto, para aumentar el rango articular del robot, pero también debe generar el par suficiente como para levantar el conjunto del antebrazo y la pinza, por tanto, debe ser resistente y contar con un motor de potencia suficiente para realizar su esta tarea.

Para cumplir con los requisitos, nuevamente se ha diseñado un conjunto centrado en una pieza central, base del brazo, a la que irán unidas el resto de las piezas y el motor. Además, se ha desplazado el motor del eje de la articulación del codo a la zona central del brazo, de forma que, una vez montado, el centro de masas del robot se mantenga lo más centrado posible y no afecte en gran medida a la estabilidad del modelo. Todas las piezas diseñadas en este conjunto se han impreso en 3D con poliestireno de alto impacto (Z-HIPS) y relleno del 30%, a excepción del Separador Polea Brazo, fabricado con metacrilato de extrusión (PMMA-XT) mediante cortadura láser. Puede observarse el diseño final en la **Figura 16**.

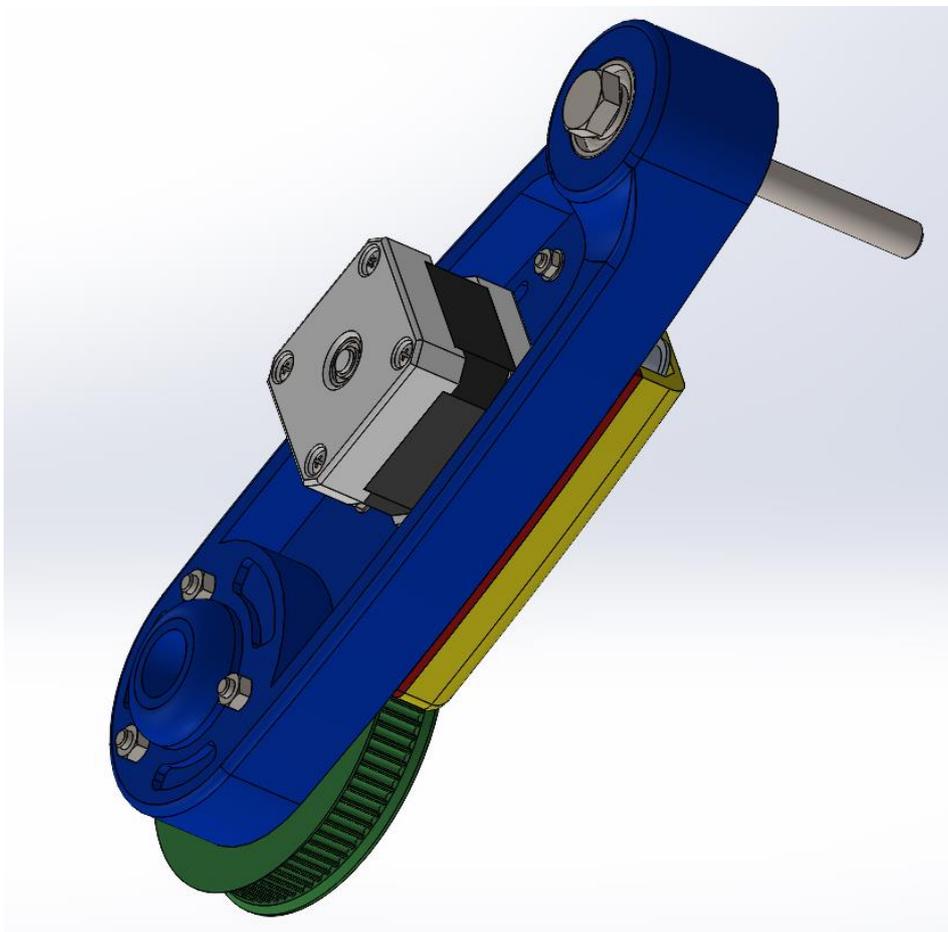


Figura 16. Conjunto brazo diferenciando sus 4 piezas principales, base del brazo (azul), cobertura del brazo (amarillo), polea brazo (verde) y separador (rojo), así como el posicionamiento del motor. Fuente: elaboración propia.

Para mantener alineada la articulación del codo se ha empleado un tornillo M8x80, según la norma ISO 4014, y un par de rodamientos rígidos de bolas obturados rodantes de 8x22x7mm (modelo 608-2RSH) para facilitar el giro de la articulación.

El motor empleado para el movimiento de la articulación del codo es un NEMA 17 de 45Ncm, bipolar de 2A y dimensiones 42x42x40mm, el mismo motor que el empleado en la base

(modelo 17HS16-2004S1). Fue seleccionado debido a sus prestaciones, su relación potencia - peso y con un par rotativo superior al necesario para evitar problemas futuros durante la realización de las tareas programadas y no limitar el rango articular. Su hoja de especificaciones técnicas y su curva de comportamiento de par de rotación en función de la velocidad y frecuencia aparecen en el **ANEXO I**.

Para la transmisión del movimiento a la articulación, se ha empleado un sistema correa – ruedas dentadas, adaptado para una correa cerrada GT2 de 6mm de ancho y 280mm de longitud. Las ruedas dentadas empleadas son una polea GT2 de 16 dientes ajustable al eje del NEMA 17 y una rueda de 100 dientes y 63mm de diámetro (también adaptada a la correa GT2) unida al subensamblaje del antebrazo.

Para facilitar la transmisión, se ha hecho uso de dos poleas GT2 sin dientes de 3mm de diámetro interno, encargadas de reducir el ángulo de agarre de la polea menor, tal como se muestra en la **Figura 17**. Además, se ha diseñado el separador (pieza roja) para mantener alineadas las poleas en todo momento y evitar el contacto directo de la correa con la pieza base del brazo.

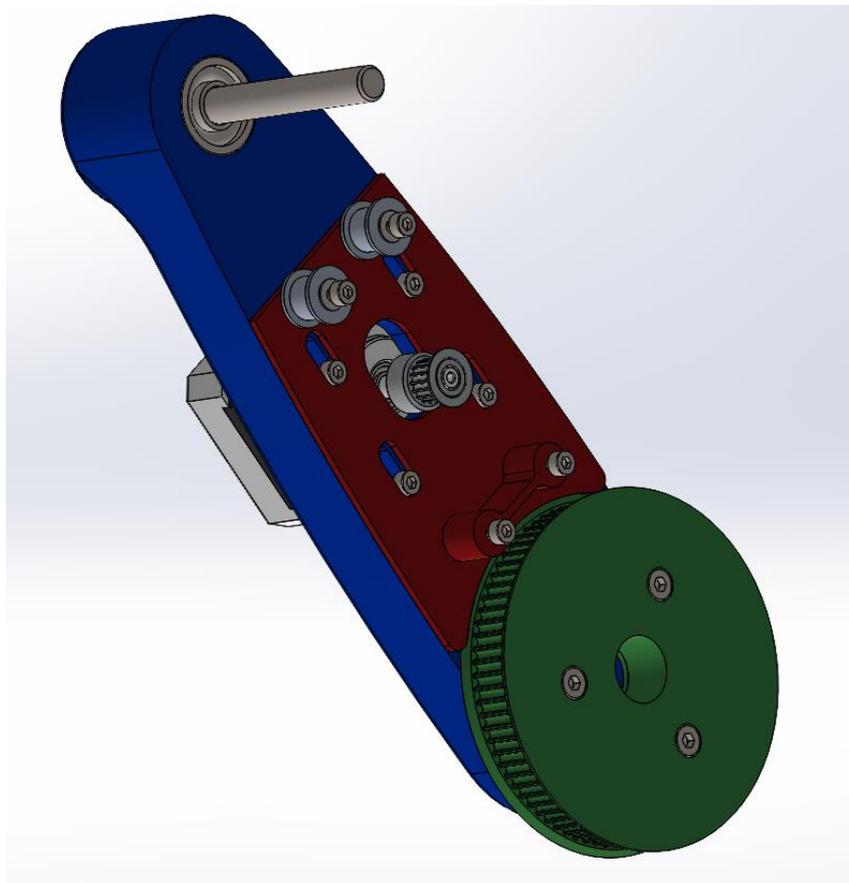


Figura 17. Vista trasera del conjunto del brazo, sin la cobertura, permitiendo ver los componentes del sistema de transmisión. Fuente: Elaboración propia.

Las dimensiones generales del conjunto del brazo son 218x42x80mm (sin tener en cuenta la longitud del eje) y una masa total aproximada de 397g (contabilizando el motor y la tornillería). Si se desean conocer más detalles acerca de los componentes o el montaje del ensamblaje, consultar el **Documento Pliego de Condiciones**.

5.4 ANTEBRAZO

El antebrazo es el conjunto encargado de mover la muñeca, la única articulación con dos grados de movimiento, uno de rotación respecto al eje longitudinal del antebrazo, y otro respecto al eje rotacional de la propia muñeca. Por tanto, debe permitir el acople de dos motores, ser ligero, para evitar sobrecargar las articulaciones del hombro y el codo, y tener una longitud adecuada para aumentar el rango articular del robot.

Para cumplir los requisitos, se ha diseñado el antebrazo en dos piezas fundamentales, además de la ruda dentada que permitirá el movimiento del codo. Todas las piezas diseñadas en este conjunto se han impreso en 3D con poliestireno de alto impacto (Z-HIPS) y relleno del 30%. La primera pieza, el antebrazo principal, que llevará en su interior una varilla roscada M4 que hará las funciones de eje para completar la primera rotación de la muñeca, y la segunda pieza, la conexión con la muñeca, que será la pieza encargada de transmitir el giro del eje y efectuar la segunda rotación de la muñeca. Puede observarse el conjunto en la **Figura 18**.

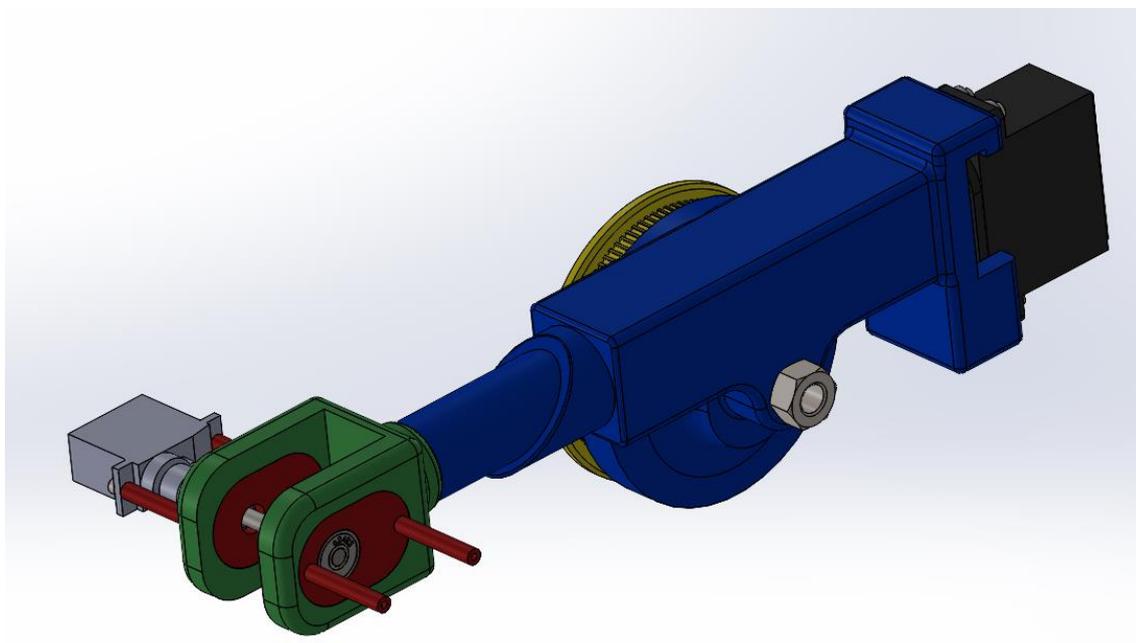


Figura 18. Conjunto antebrazo diferenciando sus 4 piezas, antebrazo principal (azul), rueda dentada (amarillo), conexión muñeca (verde) y adaptadores motores (rojo), así como el posicionamiento de los motores. Fuente: elaboración propia.

El primer motor empleado en ese subensamblaje, encargado de la rotación de la varilla roscada (situado a la izquierda en la Figura 18), es un servomotor modelo DS04-NFC, de rotación continua de 360°, par de torsión de 54Ncm a 4,8V y dimensiones 40,8x20x39,5mm. Nuevamente se ha tratado de colocar el motor lo más centrado posible, respecto al ensamblaje general, para mantener el centro de masas centrado y evitar sobrecargar al resto de motores.

En la **Figura 19** se muestra la conexión entre el motor, la varilla roscada y la conexión a la pinza.

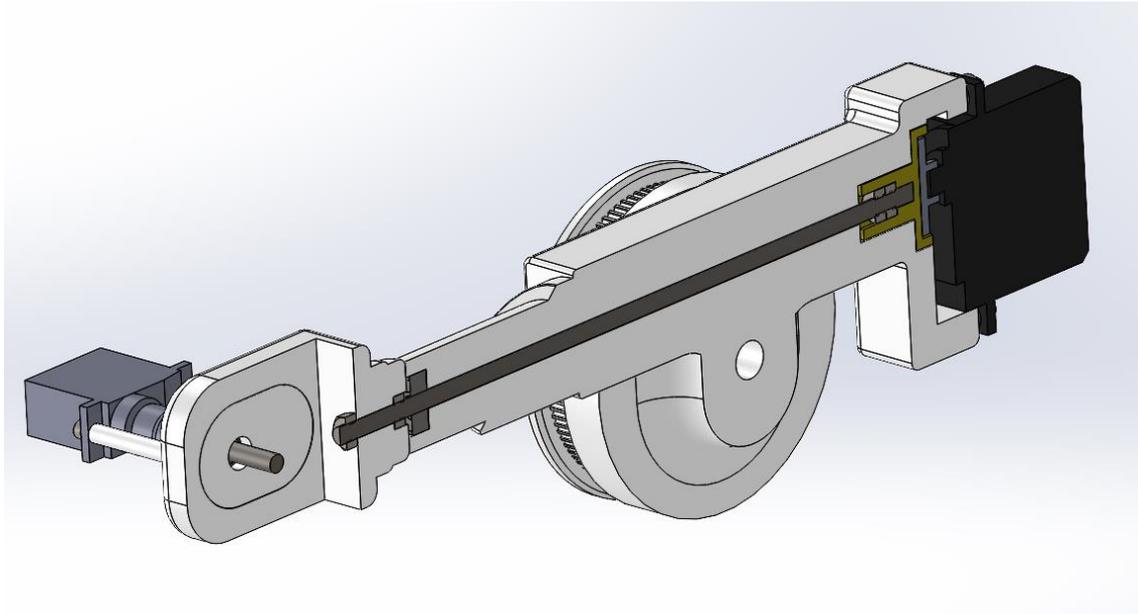


Figura 19. Sección del antebrazo, muestra la varilla roscada y su unión al servomotor (amarillo) y a la conexión con la muñeca. Fuente: Elaboración propia.

La varilla está unida al servo mediante la pieza amarilla de la Figura 19 y dos tuercas M4 a modo de fijación. También se observa cómo, para facilitar el giro de la varilla, en el extremo del antebrazo principal, hay un rodamiento rígido de bolas de dimensiones 4x13x5mm (modelo 624-2RS1). La unión entre la varilla y la conexión a la pinza se realiza mediante dos tuercas M4.

El segundo motor es un servomotor modelo MG90D, de rotación continua de 180°, par de torsión 20Ncm a 4,8V, y dimensiones 22,8x12,2x28,5mm. Está situado a uno de los lados de la conexión a la muñeca y transmite el movimiento mediante una varilla roscada de M4. En la **Figura 20** puede observarse el modo de unión entre la varilla y el servo.

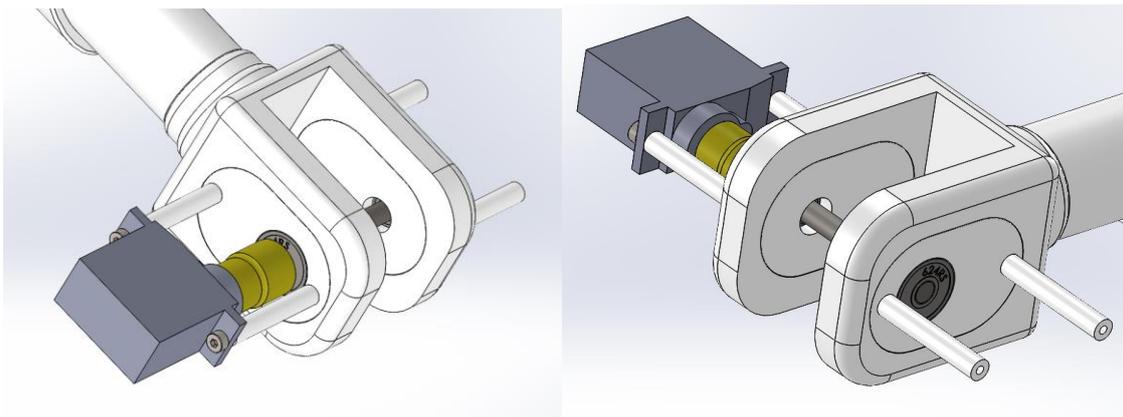


Figura 20. Conexión muñeca y sistema de unión entre servomotor y varilla roscada (pieza amarilla). Fuente: Elaboración propia.

La varilla está unida al servomotor mediante la pieza amarilla y un par de tuercas M4. Para asegurar la correcta rotación del eje, se han empleado dos rodamientos rígidos de bolas de dimensiones 4x13x5mm (modelo 624-2RS1).

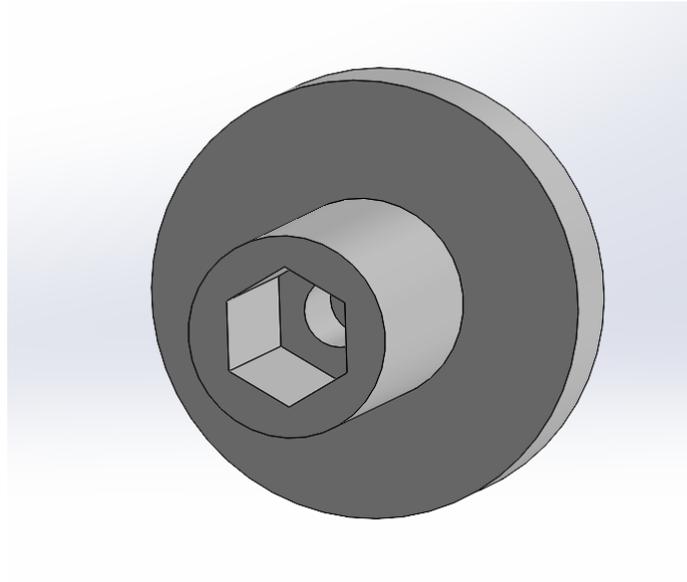


Figura 21. Pieza de acople de la varilla roscada con el servomotor DS04-NFC. Fuente: Elaboración propia.

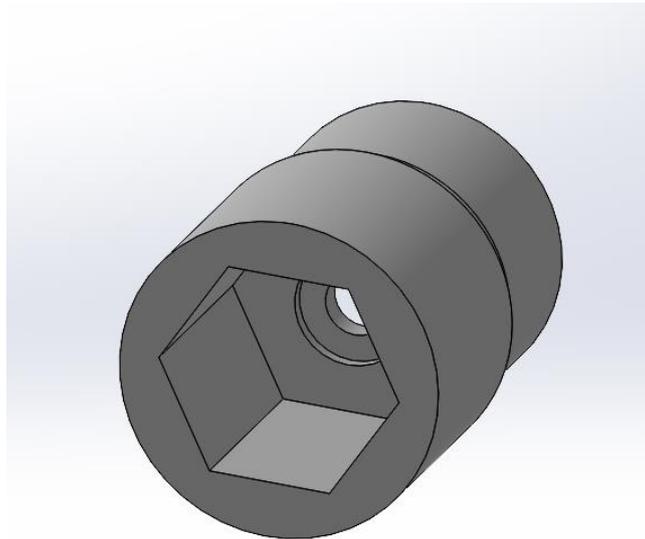


Figura 22. Pieza de acople de la varilla roscada con el servomotor MG90D. Fuente: Elaboración propia.

Las piezas diseñadas para este conjunto y los motores seleccionados son menos pesadas que en el resto de subensamblajes para cumplir con los criterios de diseño planteados. Las dimensiones generales del conjunto del antebrazo son 255x84x76mm y una masa total aproximada de 166g (contabilizando motores y tornillería). Si se desean conocer más detalles acerca de los componentes o el montaje del ensamblaje, consultar el **Documento Pliego de Condiciones**.

5.5 PINZA

El conjunto de la pinza ha sido diseñado para cumplir la tarea de agarre y manipulación de objetos. Para cumplir con las especificaciones generales impuestas para el robot, debe ser ligero, para evitar sobrecargar en exceso el resto las articulaciones, y fácilmente acoplable a la pieza de conexión del antebrazo. En la **Figura 23** se muestra el diseño de la pinza.

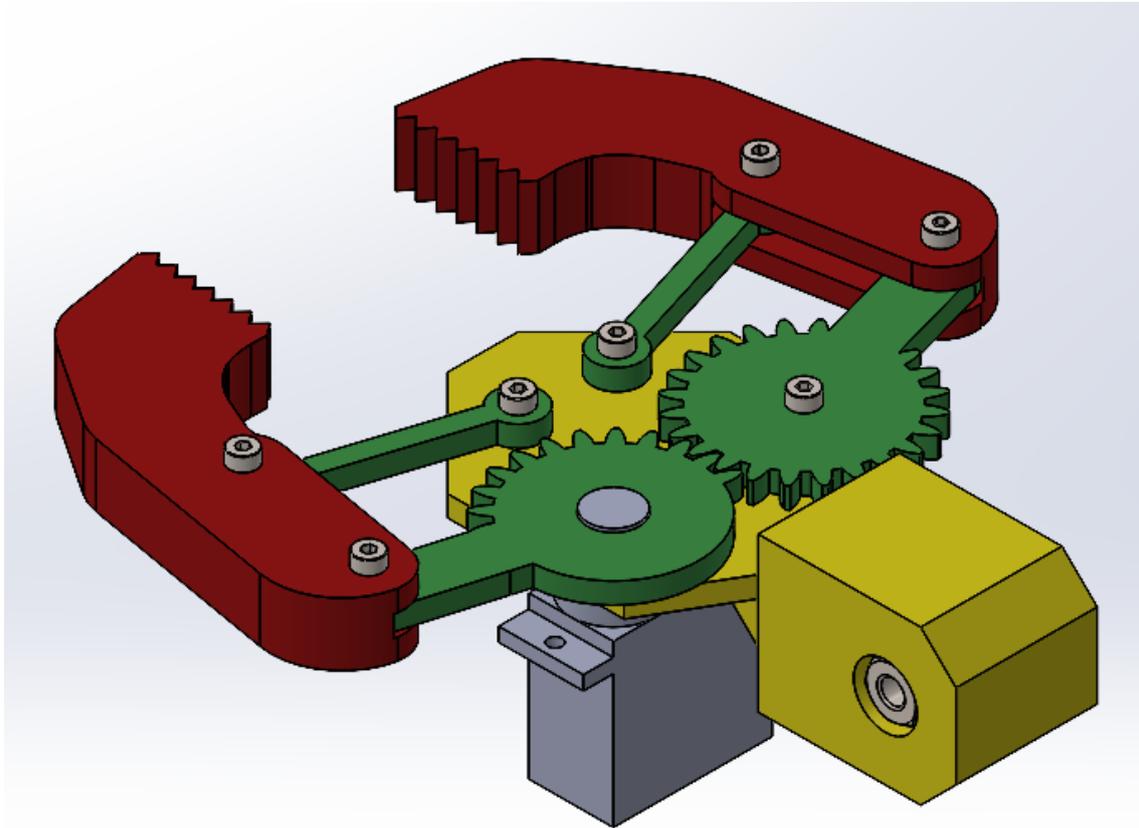


Figura 23. Conjunto pinza diferenciando sus 3 partes principales, las pinzas (rojo), el sistema de transmisión de movimiento (verde) y el soporte de la pinza (amarillo), así como el posicionamiento del motor. Fuente: Elaboración propia.

El motor empleado para la apertura y cierre de la pinza, situado en la parte inferior, es un servomotor MG90D, el mismo que el empleado en la articulación de la muñeca, de rotación continua de 180°, torsión 20Ncm a 4.8V, y dimensiones 22.8x12.2x28.5mm. El sistema de transmisión de movimiento por engranajes fue adquirido del modelo 'Robotic Arm Gripper' (Baasith Syifaa, 2020) de la web *GrabCAD Community.*, y fue adaptado al diseño de la pinza de este proyecto.

Las piezas diseñadas para este conjunto se han impreso en 3D con poliestireno de alto impacto (Z-HIPS) y relleno del 30%, a excepción del sistema de transmisión de movimiento, piezas verdes de la **Figura 23**, fabricadas con metacrilato de extrusión (PMMA-XT) mediante cortadura láser. Las dimensiones generales de la pinza son 108x37x99mm y una masa total aproximada de 40g (contabilizando motor y tornillería). Si se desean conocer más detalles acerca de los componentes o el montaje del ensamblaje, consultar el **Documento Pliego de Condiciones**.

5.6 TRASMISIONES POR CORREA Y RUDEAS DENTADAS

En las articulaciones de la base, el hombro y el codo se han empleado transmisiones por correa y ruedas dentadas entre el motor y la articulación para reducir la potencia necesaria a aplicar por los motores y para abrir la posibilidad de situar los motores en zonas más próximas al centro del robot, mejorando su estabilidad.

Estas tres articulaciones se mueven mediante los tres motores paso a paso, los más pesados de los empleados en el diseño, y por ello únicamente se ha empleado este sistema de transmisión en ellos, mientras que los servomotores empleados en los movimientos de la muñeca y la pinza se han colocado directamente sobre el eje de la articulación.

Las ruedas dentadas empleadas en las articulaciones se diseñaron en función del motor empleado en la articulación, la polea escogida para dicho motor y el tipo de correa empleada.

En el caso de las articulaciones de la base y el codo, al tener el mismo motor (NEMA 17 modelo 17HS16-2004S1), se seleccionó una polea de diámetro de eje 5mm, adaptable para estos motores, de 16 dientes y compatible con correas GT2 de 6mm de anchura. Estas correas presentan un paso de 2mm, una altura de diente de 0,75mm y una altura total de 1,38mm. Pueden verse las dimensiones de la correa en la **Figura 24**.

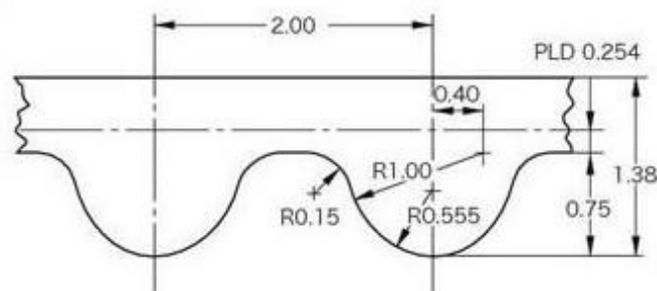


Figura 24. Dimensiones de la correa GT2. Fuente: Brico Geek, Correa dentada GT2 – 6mm.

Para el diseño de la rueda dentada situada sobre los ejes de las articulaciones de la base y el codo (polea mayor), se decidió realizar dos poleas de 100 dientes de forma que cumpliesen con los requisitos necesarios para el diseño de transmisiones por correas (que se verificarán más adelante). Teniendo en cuenta las características de la correa empleada, se calculó el diámetro específico.

$$\left\{ \begin{array}{l} D = \frac{L}{\pi} = \frac{200}{\pi} = 63.66mm \\ L = Z * p = 100 * 2mm = 200mm \end{array} \right.$$

Siendo L la longitud de la circunferencia de la rueda dentada, Z el número de dientes, y p el paso entre los dientes.

Para verificar la correcta elección del número de dientes, se comprobará la relación de transmisión entre poleas, teniendo en cuenta los siguientes datos:

$$\begin{aligned}
 \text{Polea menor} &\rightarrow \begin{cases} d = 9.6\text{mm} \\ z = 16 \end{cases} \\
 \text{Polea mayor} &\rightarrow \begin{cases} D = 63.66\text{mm} \\ Z = 100 \end{cases}
 \end{aligned}$$

La relación de transmisión equivale a la relación entre número de dientes y diámetros, por lo que debe cumplirse que:

$$i = \frac{Z}{z} \approx \frac{D}{d}$$

$$\begin{cases} \frac{Z}{z} = \frac{100}{16} = 6.25 \\ \frac{D}{d} = \frac{63.66}{9.6} = 6.63 \end{cases}$$

Para reducir la diferencia entre ambos resultados, se redujo el diámetro de la polea mayor lo máximo posible hasta que cumpliera con la tolerancia de la correa GT2 de $\pm 0,5\text{mm}$. Por tanto, el diámetro final seleccionado para la polea mayor fue de $63,2\text{mm}$, obteniendo un paso entre dientes de $1,97\text{mm}$ (aceptable para la tolerancia).

$$\begin{cases} \frac{Z}{z} = \frac{100}{16} = 6.25 \\ \frac{D}{d} = \frac{63,2}{9,6} = 6.58 \end{cases}$$

Estos valores cumplen con el requisito $i \leq 10$ impuesto para las transmisiones por correa.

Una vez verificada la correcta elección de las características de la polea mayor, se debe comprobar la distancia entre ejes, la longitud de la correa y el ángulo de agarre en los casos particulares de la base y el codo.

La distancia entre ejes (a) debe cumplir, según la norma DIN, la siguiente inecuación:

$$0,7 \cdot (D + d) < a < 2 \cdot (D + d)$$

Para el caso particular de las articulaciones de la base y el codo:

$$0,7 \cdot (63,2 + 9,6) < a < 2 \cdot (63,2 + 9,6)$$

$$51,0\text{mm} < a < 145,6\text{mm}$$

Para el caso de la base, según el diseño empleado, la distancia entre ejes de las dos ruedas dentadas es de $72,5\text{mm}$, como se muestra en la **Figura 25**, además de un juego en el enclave del motor de 15mm para realizar el tensado de la correa.

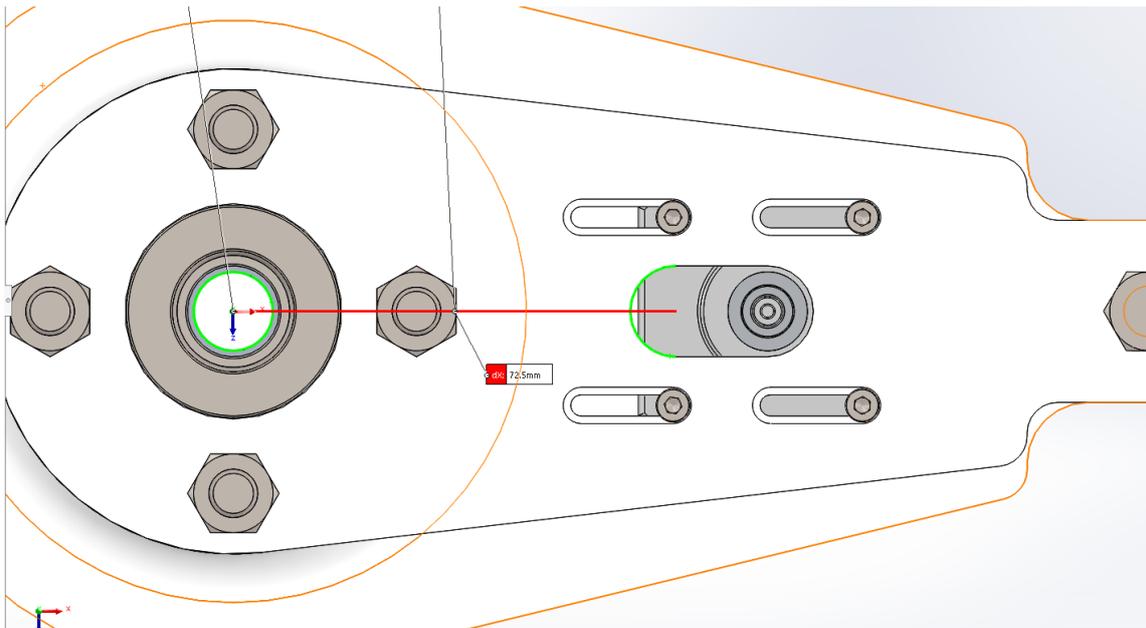


Figura 25. Distancia horizontal entre ejes de las poleas de la base. Fuente: elaboración propia.

Para el caso del codo, según el diseño empleado, la distancia entre las dos ruedas dentadas es de 70,3mm, como se muestra en la **Figura 26**, además de un juego en el enclave del motor de 8mm para realizar el tensado de la correa.

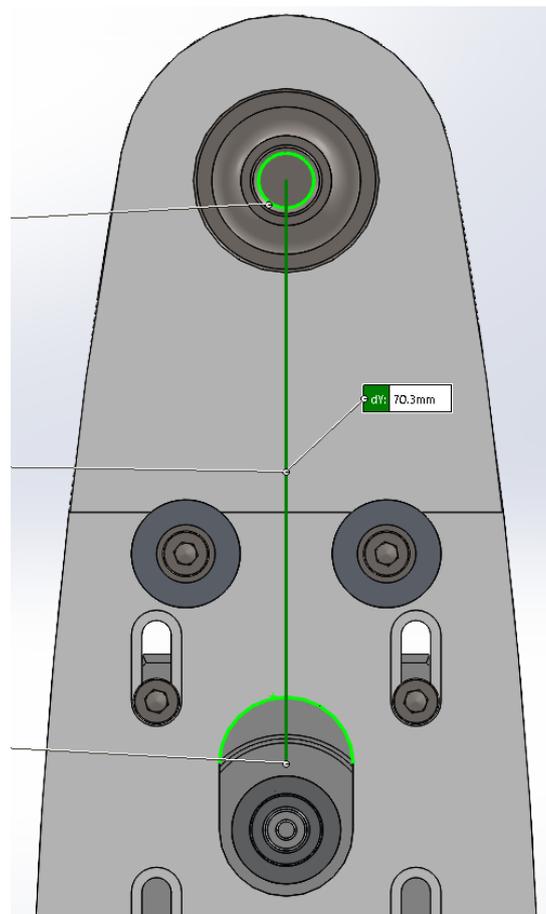


Figura 26. Distancia vertical entre ejes de las poleas del codo. Fuente: elaboración propia.

Respecto al ángulo de agarre, para la base se ha calculado mediante la siguiente ecuación trigonométrica:

$$\beta = 2 \cdot \cos^{-1} \left(\frac{D - d}{2a} \right) = 2 \cdot \cos^{-1} \left(\frac{63.2 - 9.6}{2 \cdot 72.5} \right) = 136.6^\circ$$

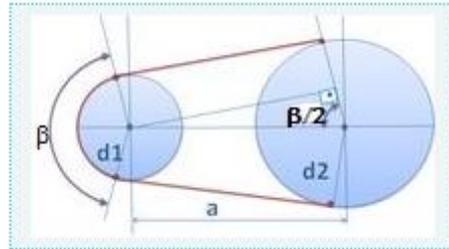


Figura 27. Esquema que permite la deducción de la ecuación trigonométrica. Fuente: BIRT LH, Correas trapeciales [5].

Para el codo, debido a que es la articulación encargada de mover el antebrazo y cargar con su peso, se tomó la decisión de reducir el ángulo de agarre de la polea menor para evitar al máximo posible el deslizamiento. Para ello se emplearon dos poleas GT2 sin dientes situadas entre los ejes de las ruedas dentadas, como puede observarse en la **Figura 28**.

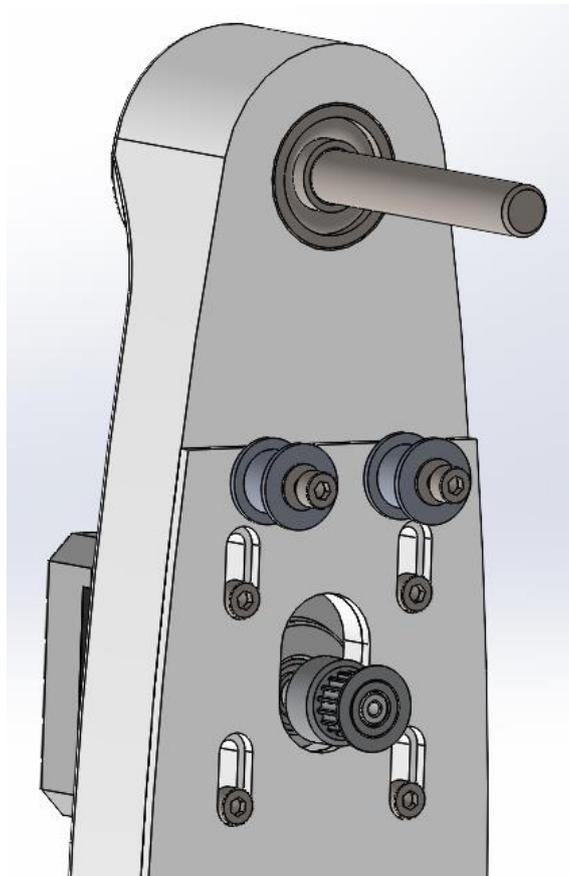


Figura 28. Trasmisión del codo, polea menor y poleas encargadas de aumentar el ángulo de agarre. Fuente: Elaboración propia.

Para obtener este ángulo, así como las longitudes necesarias para las correas del codo y la base, se ha realizado una representación gráfica de los sistemas de transmisión en el programa Fusion360.

La transmisión del codo se representa en la **Figura 29**.

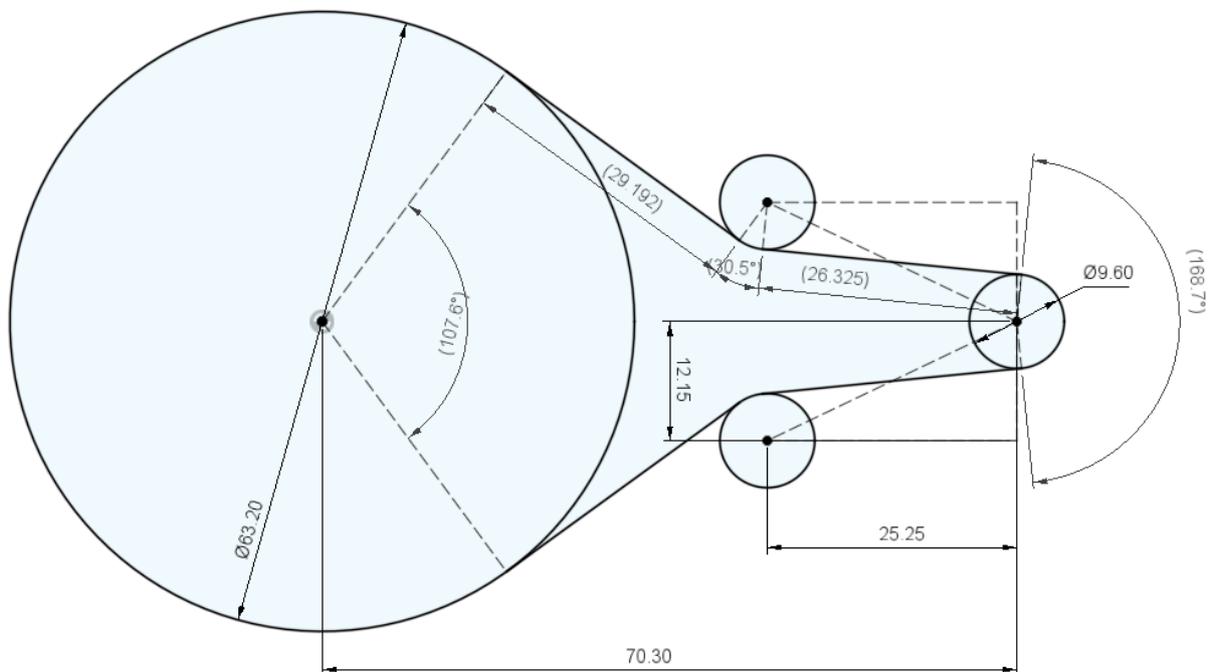


Figura 29. Esquema gráfico de la transmisión del codo junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.

El ángulo de agarre logrado para la polea menor es de 168,7°, más próximo a 180° (el ángulo de agarre ideal para una polea menor), reduciendo así en gran medida el posible deslizamiento de la correa.

La longitud mínima de la correa necesaria para la transmisión del codo puede obtenerse de las medias expresadas en la **Figura 29**.

$$L = \frac{63.2}{2} \cdot (360 - 107.6) \cdot \frac{\pi}{180} + 2 \cdot \left(29.2 + 26.3 + \frac{9.6}{2} \cdot 30.5 \cdot \frac{\pi}{180} \right) + \frac{9.6}{2} \cdot 168.7 \cdot \frac{\pi}{180}$$

$$= 269.45 \text{ mm}$$

La longitud mínima de la correa necesaria para la transmisión de la base puede deducirse de las medidas de la **Figura 30**.

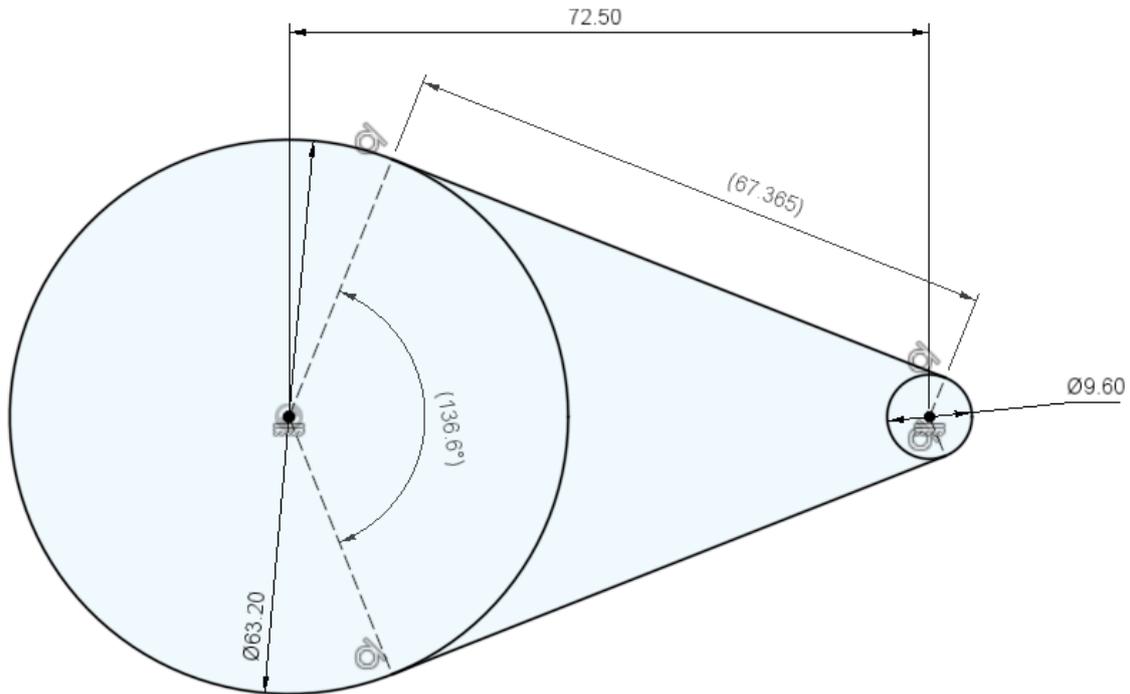


Figura 30. Esquema gráfico de la transmisión de la base junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.

$$L = \frac{63.2}{2} \cdot (360 - 136.6) \cdot \frac{\pi}{180} + 2 \cdot 67.4 + \frac{9.6}{2} \cdot 136.6 \cdot \frac{\pi}{180} = 269.45 \text{ mm}$$

Sin embargo, la longitud comercial de correas GT2 de 6mm de ancho más próxima a la calculada, y la que se empleó para el diseño, es de 280mm para ambas articulaciones.

En el caso de la articulación del hombro, con el motor (NEMA 17 con tren planetario modelo 17HS15-1684S-PG5), se seleccionó una patea de diámetro de eje 6mm, de 16 dientes y compatible con correas HTD 3M de hasta 11mm de anchura. Estas correas presentan un paso de 3mm, una altura de diente de 1.2mm y una altura total de 2.4mm. Pueden verse las dimensiones de la correa en la **Figura 31**.

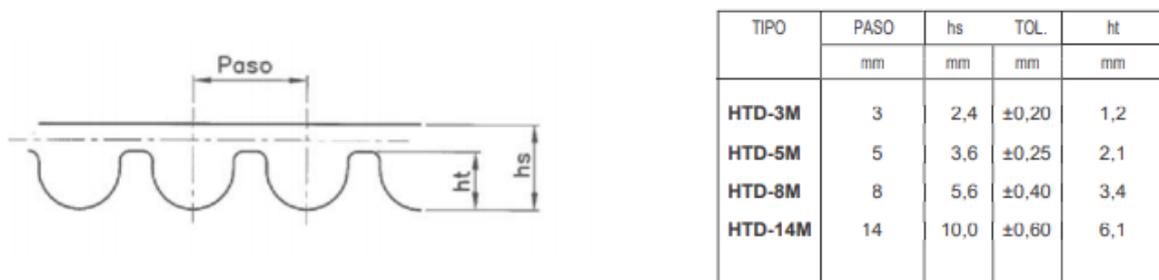


Figura 31. Dimensiones de las correas HTD (caso particular HTD M3). Fuente: Talleres auxiliares del motor, Selección del paso de la correa dentada [20].

Para el diseño de la rueda dentada situada sobre los ejes de la articulación del hombro (patea mayor), se decidió realizar una patea de 80 dientes de forma que cumpliera con los requisitos necesarios para el diseño de transmisiones por correas (que se verificarán más adelante). Teniendo en cuenta las características de la correa empleada, se calculó el diámetro específico.

$$\begin{cases} D = \frac{L}{\pi} = \frac{240}{\pi} = 76.4mm \\ L = Z \cdot p = 80 \cdot 3mm = 240mm \end{cases}$$

Siendo L la longitud de la circunferencia de la rueda dentada, Z el número de dientes, y p el paso entre los dientes.

Para verificar la correcta elección del número de dientes, se comprobará la relación de transmisión entre poleas, teniendo en cuenta los siguientes datos:

$$\begin{aligned} \text{Polea menor} &\rightarrow \begin{cases} d = 14.5mm \\ z = 16 \end{cases} \\ \text{Polea mayor} &\rightarrow \begin{cases} D = 76.4mm \\ Z = 80 \end{cases} \end{aligned}$$

La relación de transmisión equivale a la relación entre número de dientes y diámetros, por lo que debe cumplirse que:

$$i = \frac{Z}{z} \approx \frac{D}{d}$$

$$\begin{cases} \frac{Z}{z} = \frac{80}{16} = 5 \\ \frac{D}{d} = \frac{76.4}{14.5} = 5,27 \end{cases}$$

Para reducir la diferencia entre ambos resultados, se redujo el diámetro de la polea mayor lo máximo posible hasta que cumpliera con la tolerancia de la correa HTD 3M de $\pm 0.5mm$. Por tanto, el diámetro final seleccionado para la polea mayor fue de 75.6mm, obteniendo un paso entre dientes de 2.96mm (aceptable para la tolerancia).

$$\begin{cases} \frac{Z}{z} = \frac{80}{16} = 5 \\ \frac{D}{d} = \frac{75.6}{14.5} = 5,21 \end{cases}$$

Estos valores cumplen con el requisito $i \leq 10$ impuesto para las transmisiones por correa.

Una vez verificada la correcta elección de las características de la polea mayor, se debe comprobar la distancia entre ejes, la longitud de la correa y el ángulo de agarre.

La distancia entre ejes (a) debe cumplir, según la norma DIN, la siguiente inecuación:

$$0.7 \cdot (D + d) < a < 2 \cdot (D + d)$$

Para el caso particular del hombro:

$$0.7 \cdot (75.6 + 14.5) < a < 2 \cdot (75.6 + 14.5)$$

$$63.1mm < a < 180.2mm$$

Según el diseño empleado, la distancia entre ejes de las dos ruedas dentadas es de 80mm, como se muestra en la **Figura 32**, además de un juego en el enclave del motor de 8mm para realizar el tensado de la correa.

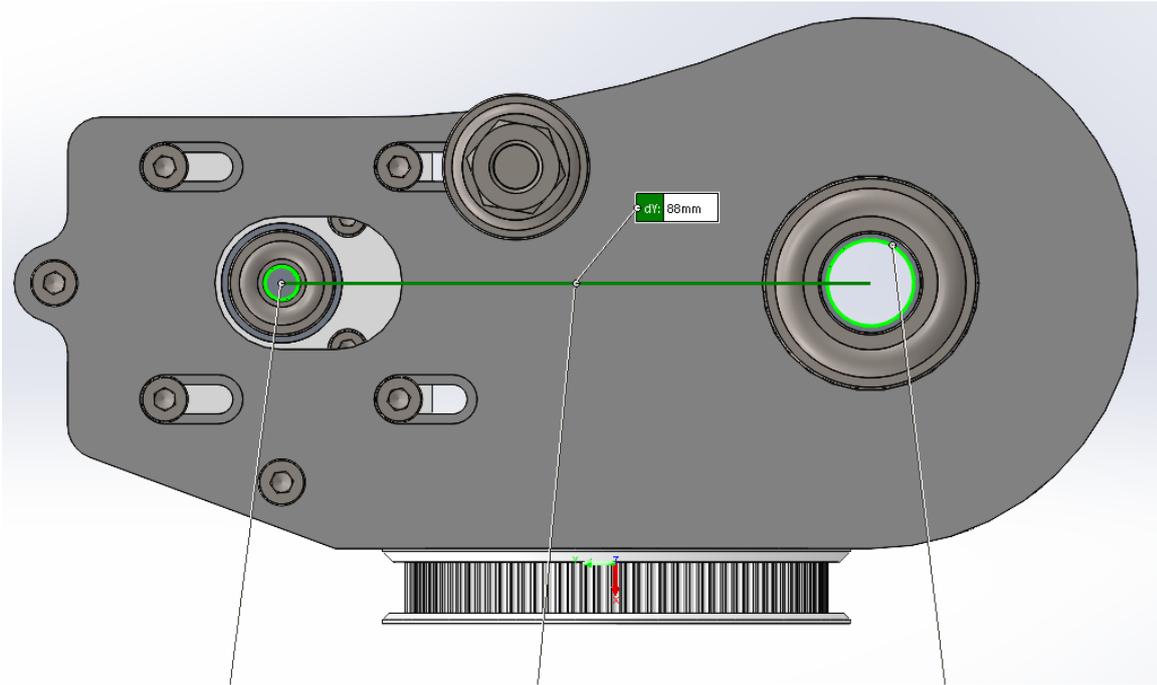


Figura 32. Distancia horizontal entre ejes de las poleas del hombro. Fuente: elaboración propia.

Para el hombro, debido a que es la articulación encargada de mover el brazo y cargar con su peso, se tomó la decisión de reducir el ángulo de agarre de la polea menor para evitar al máximo posible el deslizamiento. Para ello se emplearon dos rodamientos rígidos de bolas modelo 608-2RSH situados entre los ejes de las ruedas dentadas, como puede observarse en la **Figura 33**.

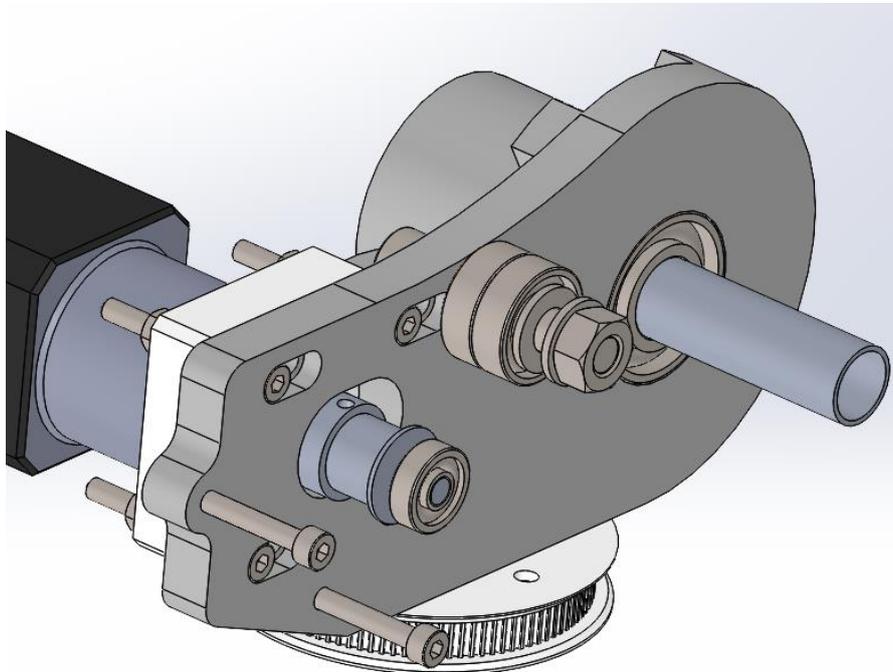


Figura 33. Trasmisión del hombro, polea menor y rodamientos encargados de aumentar el ángulo de agarre. Fuente: Elaboración propia.

Para obtener el ángulo de agarre, así como la longitud necesaria para la correa del hombro, se ha realizado una representación gráfica del sistema de transmisión en el programa Fusion360.

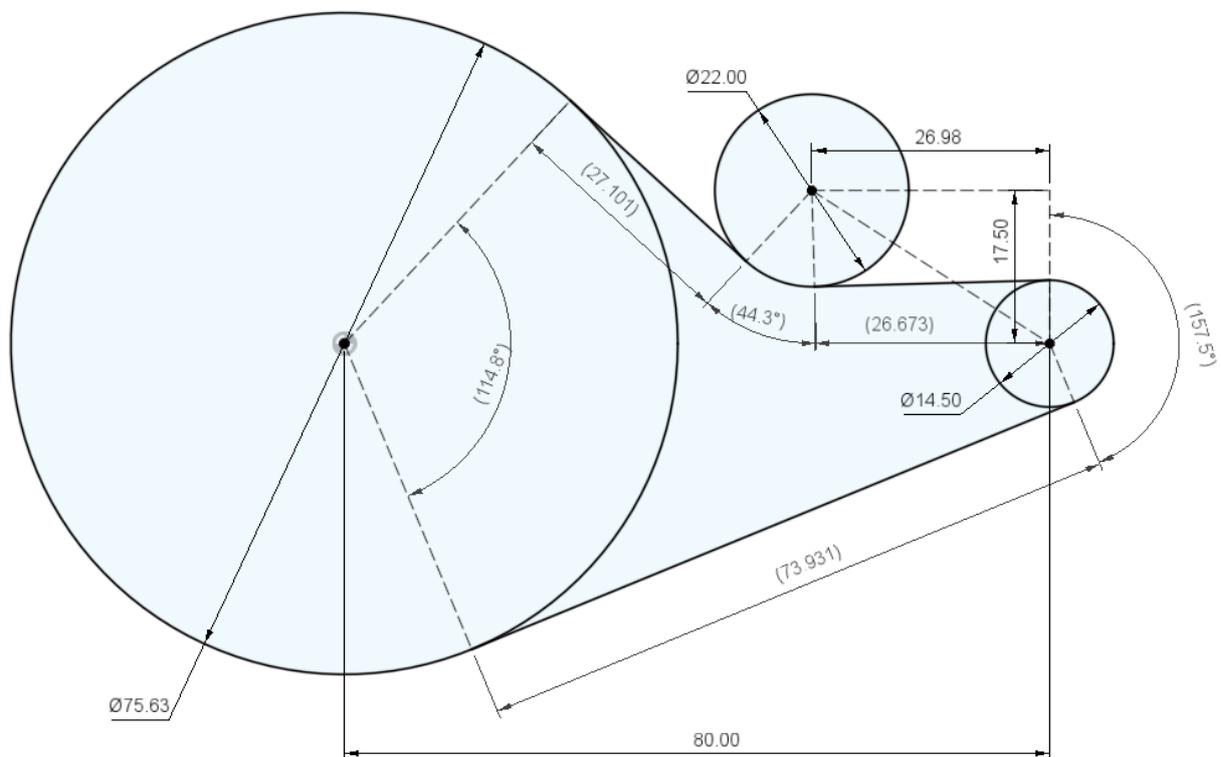


Figura 34. Esquema gráfico de la transmisión del hombro junto con las medidas de los tramos y ángulos del dibujo a escala 1:1. Fuente: Elaboración propia.

El ángulo de agarre logrado para la polea menor es de 157.4°, más próximo a 180° (el ángulo de agarre ideal para una polea menor).

La longitud mínima de la correa necesaria para la transmisión del codo puede obtenerse de las medias expresadas en la **Figura 34**.

$$L = \frac{75.6}{2} \cdot (360 - 114.8) \cdot \frac{\pi}{180} + \left(27.1 + 26.7 + \frac{22}{2} \cdot 44.3 \cdot \frac{\pi}{180} \right) + 73.9 + \frac{14.5}{2} \cdot 157.5 \cdot \frac{\pi}{180} = 317.9 \text{ mm}$$

Sin embargo, la longitud comercial de correas HTD 3M de 10mm de ancho más próxima a la calculada, y la que se empleó para el diseño, es de 330mm.

En la **Tabla 2** se muestran los resultados del diseño de las transmisiones de las articulaciones base, hombro y codo.

	Polea menor		Polea mayor		Correa	
	Dientes	Diámetro (mm)	Dientes	Diámetro (mm)	Tipo	Ancho (mm)
Base	16	9.6	100	63.2	GT2	6
Hombro	16	14.5	80	75.6	HTD 3M	10
Codo	16	9.6	100	63.2	GT2	6

Tabla 2. Características de las transmisiones base, hombro y codo. Fuente: Elaboración propia.

6 ELECTRÓNICA

Para la electrónica del ensamblaje, a parte de los motores empleados ya descritos anteriormente, se han empleado 3 dispositivos L298n para facilitar el control de los motores paso a paso, un microcontrolador Arduino UNO R3, una fuente de alimentación FAC-363B, una placa base para realizar las conexiones pertinentes y cables de conexión macho-macho y macho-hembra. A continuación, se muestra el conexionado de los distintos componentes, así como su diagrama electrónico esquemático. Estos diagramas se han realizado mediante la aplicación Fritzing, un programa empleado para el diseño de circuitos electrónicos de todo tipo en el que diversas compañías del sector de la electrónica ofrecen varios de sus productos para facilitar su representación y con un soporte que permite a usuarios crear plantillas de dispositivos electrónicos que no se encuentran en el programa básico, definiendo sus conexiones internas, sus funciones y diseñar una interfaz gráfica para el mismo. El conexionado se representa en la **Figura 35**.

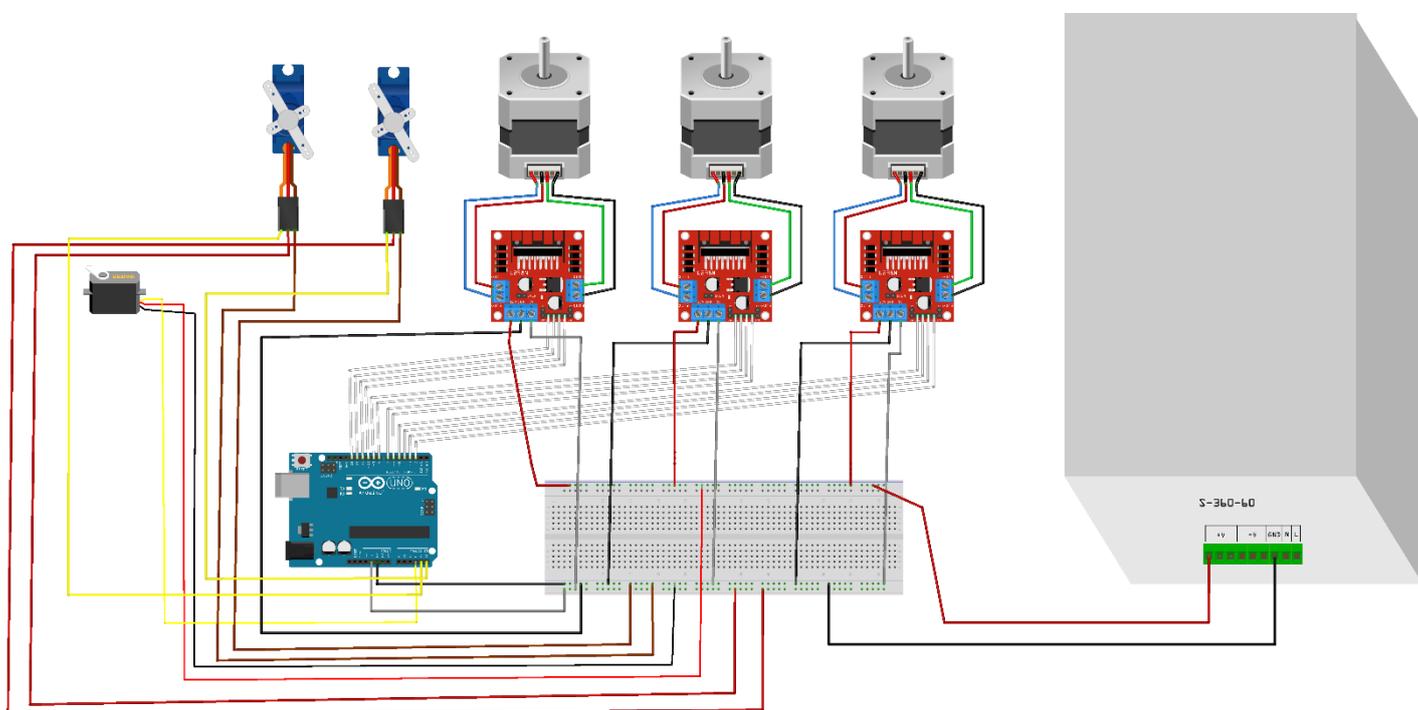


Figura 35. Conexionado y componentes electrónicos del brazo robot. Fuente: Elaboración propia.

El servomotor negro hace referencia al servo de giro continuo de 360° empleado para la rotación de la muñeca, mientras que los servos azules representan los servos de 180° empleados para el giro de la muñeca y la apertura y cierre de la pinza. En el caso de los motores paso a paso, se ha empleado el mismo modelo de NEMA 17 encontrado en la librería de Fritzing, a causa de que no se pudo conseguir un modelo del NEMA 17 con tren planetario de engranajes empleado en el hombro. Para representar la fuente de alimentación se empleó un modelo S-360-60 importado diseñado por un usuario en Github de uso libre.

Para el control de los motores paso a paso se emplearon controladores L298n, los cuales, junto con el servomotor de 360°, se conectaron directamente a la fuente de alimentación mediante la placa de circuitos, mientras que los dos servomotores de 180° se conectaron a la alimentación de 5V de la placa Arduino, para evitar errores de sobretensión y sobrecargas

(podría ocurrir si se conectan directamente a la fuente de alimentación por la elevada corriente suministrada por la misma).

Los controladores L298n deben conectarse tanto a los motores como a la placa Arduino y a la fuente de alimentación. Los dos pares de conexiones a los lados del mismo sirven para controlar el motor paso a paso correspondiente. Estas 4 conexiones están unidas a los 4 pines que se conectarán a los pines de salida de Arduino, por los que llegará la información del programa para mover los motores. De las tres conexiones restan, la primera empezando por la izquierda debe conectarse a la fuente de alimentación, ya que es la encargada de alimentar el motor. La segunda se conectará a la tierra de la fuente de alimentación y a la de Arduino. La última conexión se emplea para alimentar el propio controlador e irá conectado a la alimentación de 5V de la placa Arduino.

En el caso de los servomotores, con únicamente 3 conexiones, la primera de ellas (amarilla) se conectarán a los pines de Arduino para recibir información, la segunda es la toma de tierra y la tercera la alimentación.

La configuración de la fuente de alimentación se realizará en función del programa introducido y la cantidad de motores que deben moverse. Por lo general, el voltaje de entrada deberá estar entre 4.5V y 6V, mientras que la corriente deberá ser como mínimo de 1.5A, si únicamente se pretende mover un motor a la vez y el robot no está cargado.

7 CINEMÁTICA

7.1 MODELO CINEMÁTICO DIRECTO

En primer lugar, para la resolución de la cinemática de un robot, se debe identificar tanto los eslabones o partes que conforman el robot como los ejes de las articulaciones que sirven de unión entre estos. En el caso de este brazo, al ser un robot de 5 grados de libertad, se tiene un conjunto de 7 partes, contando la base como sólido 0 (tal y como establece la sistemática de Denavit y Hartenberg), y un total de 5 articulaciones. La numeración y distribución aparece representada en la **Figura 36**.

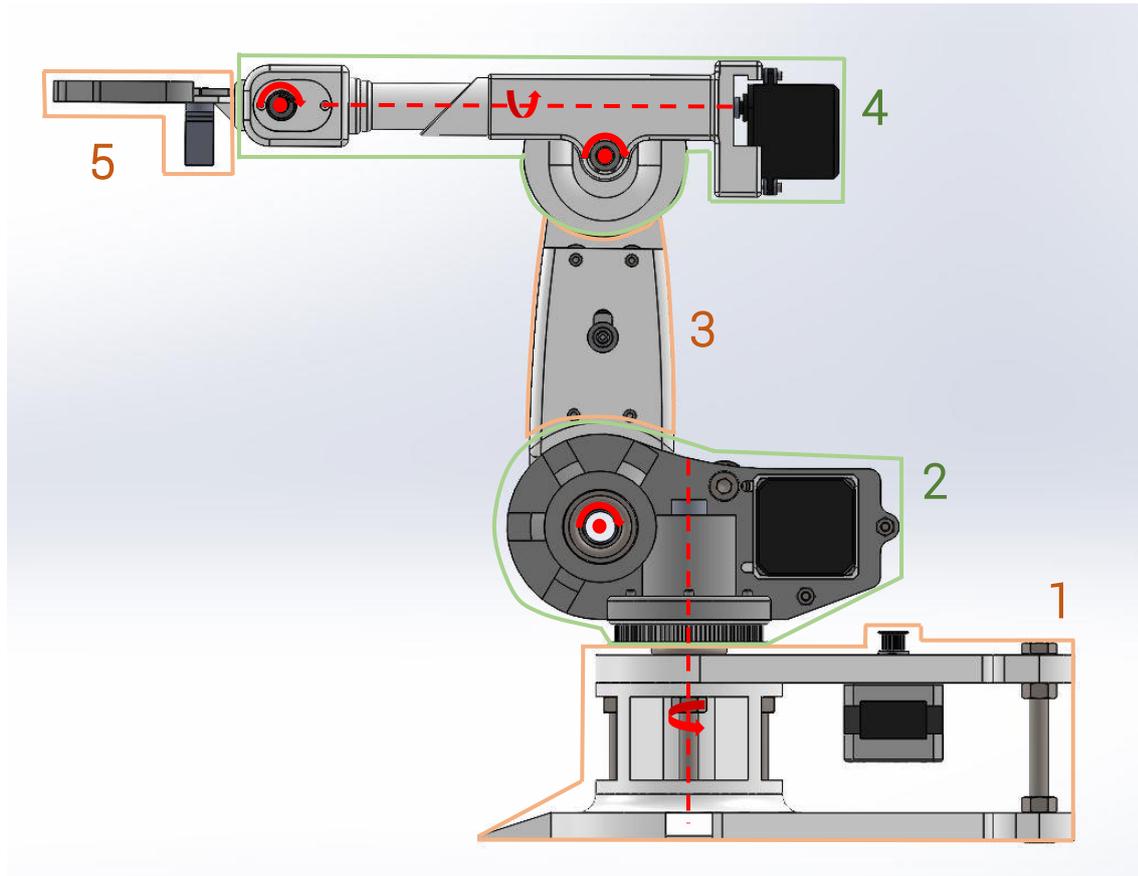


Figura 36. Representación de ejes articulares (rojo), eslabones del brazo (naranja y verde) y sentido de giro de las articulaciones. Fuente: Elaboración propia.

Tras la numeración, deben establecerse los ejes de las articulaciones, que más adelante servirán para definir la tabla Denavit-Hartenberg y obtener las matrices de paso para la resolución del modelo cinemático directo. La definición de los sistemas de coordenadas se representa en la **Figura 37**.

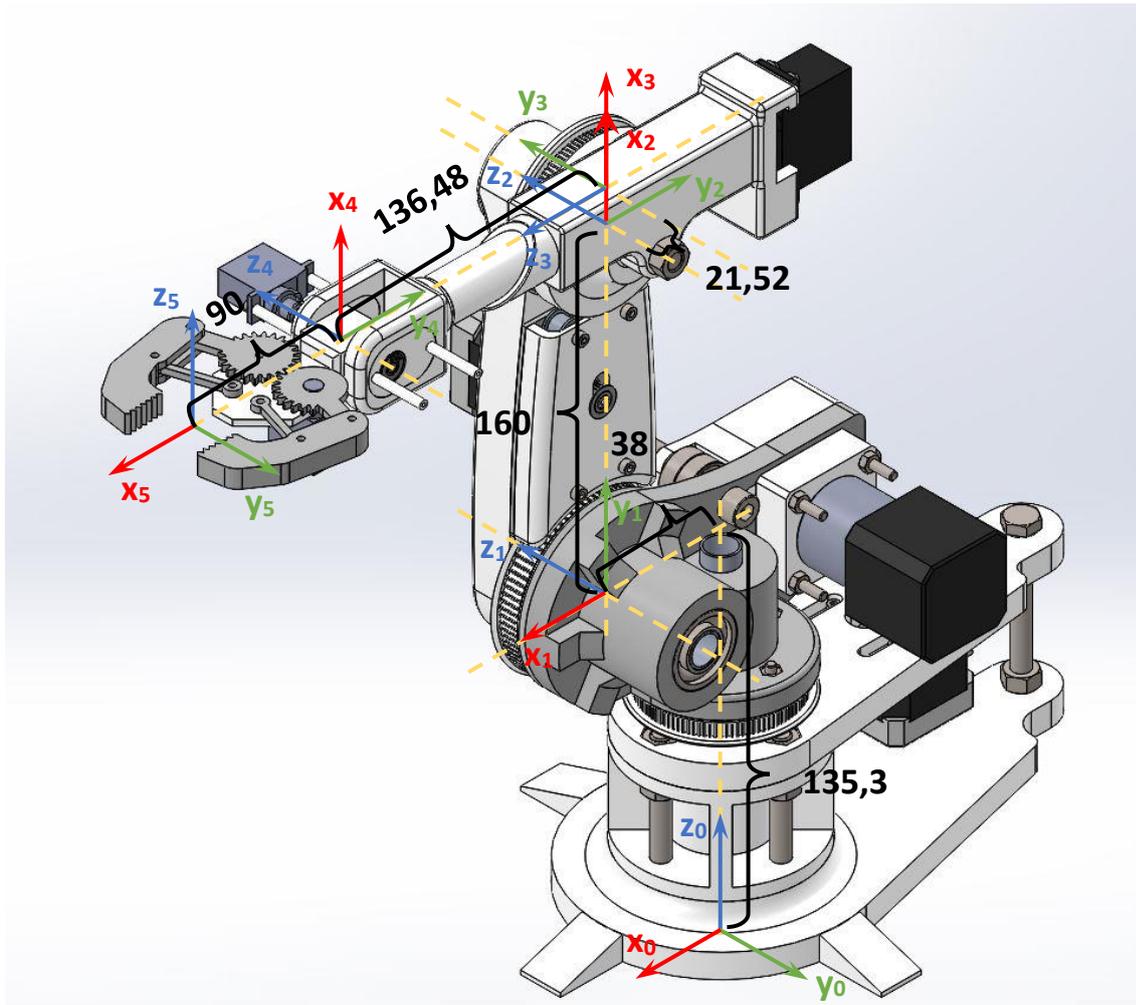


Figura 37. Representación de ejes locales y distancias articulares para la obtención de la tabla D-H.
Fuente: Elaboración propia.

Una vez definidos todos los sistemas de coordenadas, tomando una configuración de reposo para el robot, se podrá realizar la tabla Denavit-Hartenberg mediante la cual se podrán obtener las relaciones entre todas y cada una de las articulaciones del robot y comprender su posicionamiento y orientación relativas a cualquiera de las demás. Para el caso particular de este proyecto, la posición de reposo elegida coincide con la configuración que tiene el robot en la Figura 37. Por tanto, la tabla D-H resultante es la que aparece en la **Tabla 3**.

	θ_i (deg)	d_i (mm)	a_i (mm)	α_i (deg)
0-1	θ_1	135.30	38.00	90
1-2	$\theta_2 + 90$	0	160.00	0
2-3	θ_3	0	21,52	90
3-4	θ_4	136.48	0	-90
4-5	$\theta_5 - 90$	0	90.00	-90

Tabla 3. Parámetros de la matriz D-H.

A partir de la tabla D-H, multiplicando las matrices de transformación correspondientes a los giros en Z, las traslaciones en Z, las traslaciones en X y los giros en X, se pueden obtener las matrices de paso correspondientes al modelo cinemático directo.

$$T_{0-1} = \begin{bmatrix} \cos(q1) & -\sin(q1) & 0 & 0 \\ \sin(q1) & \cos(q1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 135,3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 38 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{0-1} = \begin{bmatrix} \cos(q1) & 0 & \sin(q1) & 38 \cdot \cos(q1) \\ \sin(q1) & 0 & -\cos(q1) & 38 \cdot \sin(q1) \\ 0 & 1 & 0 & 135,3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1-2} = \begin{bmatrix} \cos(q2 + 90) & -\sin(q2 + 90) & 0 & 0 \\ \sin(q2 + 90) & \cos(q2 + 90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 160 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{1-2} = \begin{bmatrix} \cos(q2 + 90) & -\sin(q2 + 90) & 0 & 160 \cdot \cos(q2 + 90) \\ \sin(q2 + 90) & \cos(q2 + 90) & 0 & 160 \cdot \sin(q2 + 90) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2-3} = \begin{bmatrix} \cos(q3) & -\sin(q3) & 0 & 0 \\ \sin(q3) & \cos(q3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 21,52 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{2-3} = \begin{bmatrix} \cos(q3) & 0 & \sin(q3) & 21,52 \cdot \cos(q3) \\ \sin(q3) & 0 & -\cos(q3) & 21,52 \cdot \sin(q3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{3-4} = \begin{bmatrix} \cos(q4) & -\sin(q4) & 0 & 0 \\ \sin(q4) & \cos(q4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 136,48 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{3-4} = \begin{bmatrix} \cos(q4) & 0 & -\sin(q4) & 0 \\ \sin(q4) & 0 & \cos(q4) & 0 \\ 0 & -1 & 0 & 136,48 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{4-5} = \begin{bmatrix} \cos(q5 - 90) & -\sin(q5 - 90) & 0 & 0 \\ \sin(q5 - 90) & \cos(q5 - 90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 90 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{4-5} = \begin{bmatrix} \cos(q5 - 90) & 0 & -\sin(q5 - 90) & 90 \cdot \cos(q3) \\ \sin(q5 - 90) & 0 & \cos(q5 - 90) & 90 \cdot \sin(q3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplicando las seis matrices de paso, se obtiene la matriz de transformación T_{0-5} , que corresponde a la matriz que relaciona el órgano terminal del robot con el sistema de referencia de la base. Esta matriz de transformación servirá para la obtención del modelo cinemático inverso.

$$T_{0-5} = T_{0-1} \cdot T_{1-2} \cdot T_{2-3} \cdot T_{3-4} \cdot T_{4-5}$$

La matriz T_{0-5} en función de las variables q no se ha incluido en la memoria debido a su extensión.

Con el fin de verificar estos cálculos, se decidió emplear la aplicación RobotScene [7].

RobotScene consiste en una aplicación desarrollada para el modelado sencillo e intuitivo de robots en un espacio virtual tridimensional. Con ella pueden crearse objetos de distintas morfologías y tamaños que, mediante el ejecutable de construcción de robots, los usuarios pueden ensamblar y colocar siguiendo las relaciones de una tabla de Denavit-Hartenberg.

La rapidez y simplicidad de la aplicación permiten a los usuarios hacer uso de esta herramienta sin necesidad de conocimientos previos de diseño tridimensional. Únicamente con ciertos conocimientos básicos de robótica industrial y programación (en caso de querer emplear las herramientas de cinemática inversa del programa) se pueden modelar y estudiar robots de todo tipo. Es por ello por lo que esta aplicación es de gran utilidad para proyectos de este estilo o para la docencia de asignaturas relacionadas con la robótica.

Haciendo uso de esta aplicación, se diseñó una simplificación de los eslabones del modelo, de forma que estos estuviesen orientados y posicionados correctamente en función de su disposición en el robot.

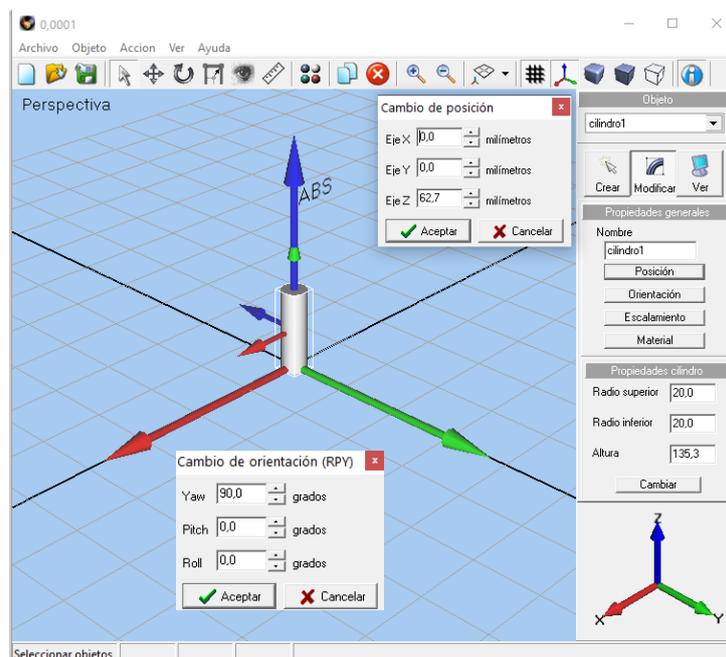


Figura 38. Base diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.

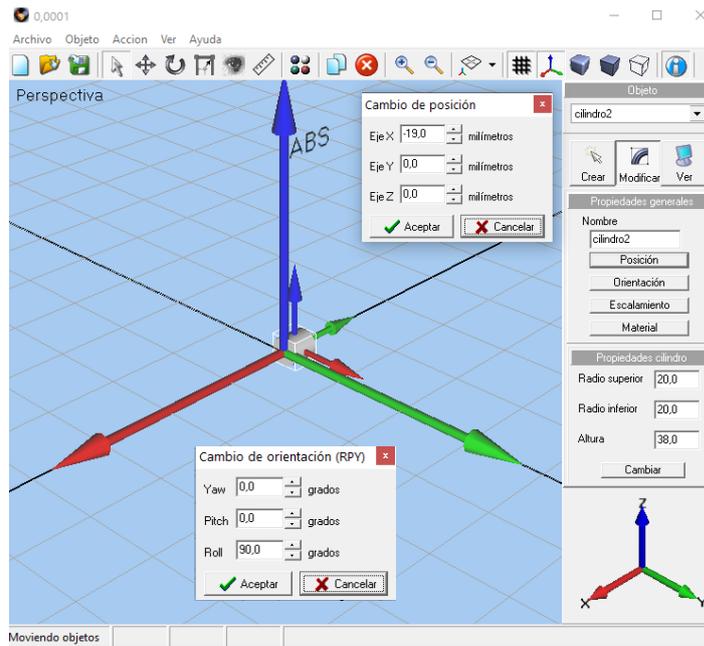


Figura 39. Hombro diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.

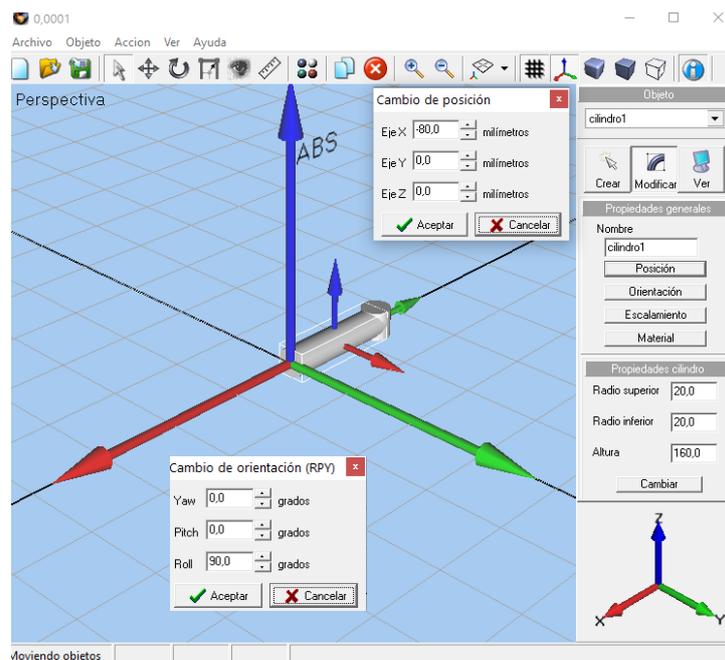


Figura 40. Brazo diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. FueAnte: Elaboración propia.

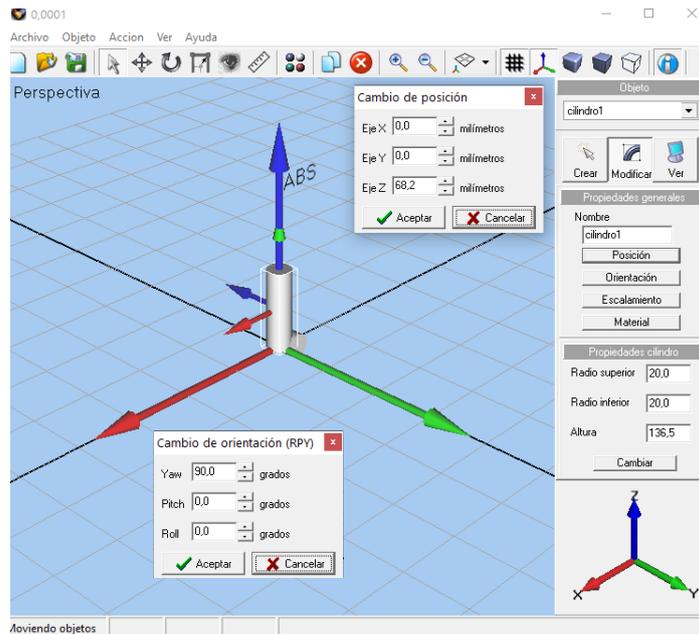


Figura 41. Antebrazo diseñado en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.

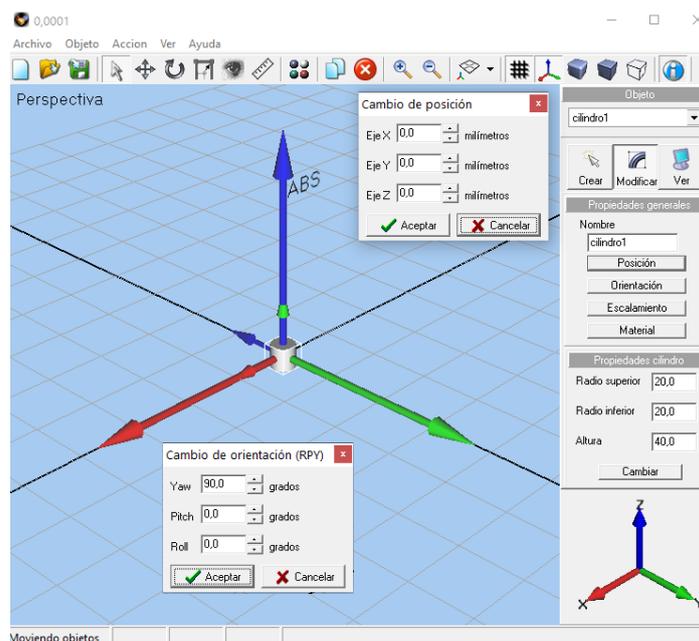


Figura 42. Muñeca diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.

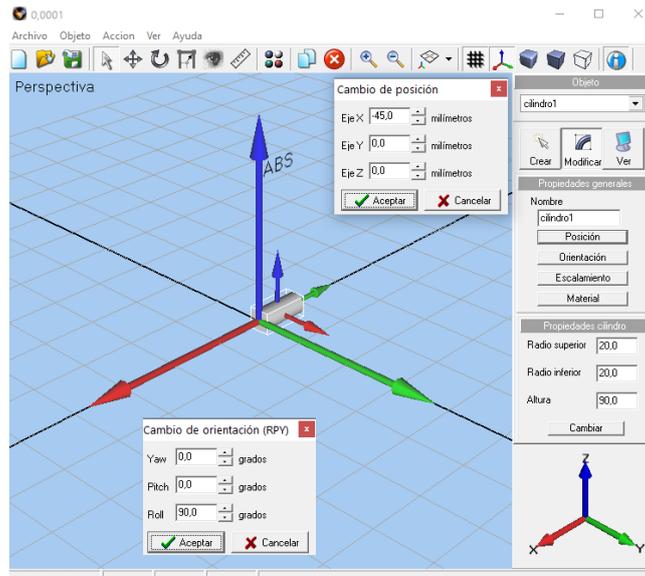


Figura 43. Pinza diseñada en RobotScene y su posición y orientación respecto al sistema de coordenadas absoluto. Fuente: Elaboración propia.

Una vez realizados los objetos, se introdujeron en el constructor de robots y se configuraron siguiendo la tabla de Denavit-Hartenberg calculada. Los sistemas de referencia del resultado obtenido coincidieron con los del esquema realizados en primera instancia. Esto permite verificar la cinemática directa del modelo.

Como puede observarse en la **Figura 44**, la tabla de parámetros introducida coincide con la calculada, así como los sistemas de referencia de las articulaciones y, por tanto, esto valida el resultado obtenido, **Figura 45**.

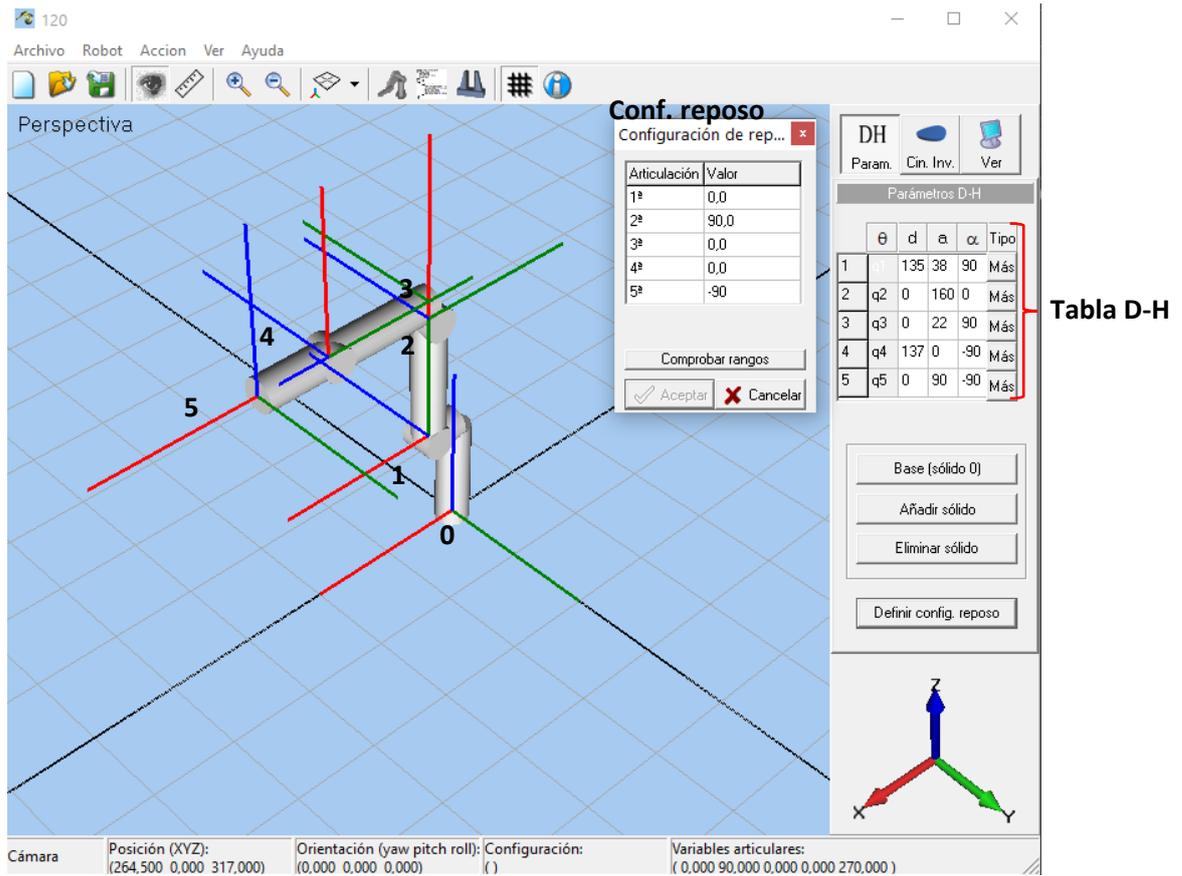


Figura 44. Ensamblaje del robot en RobotScene con sus sistemas de referencia, la tabla D-H y su posición de reposo. Fuente: Elaboración propia.

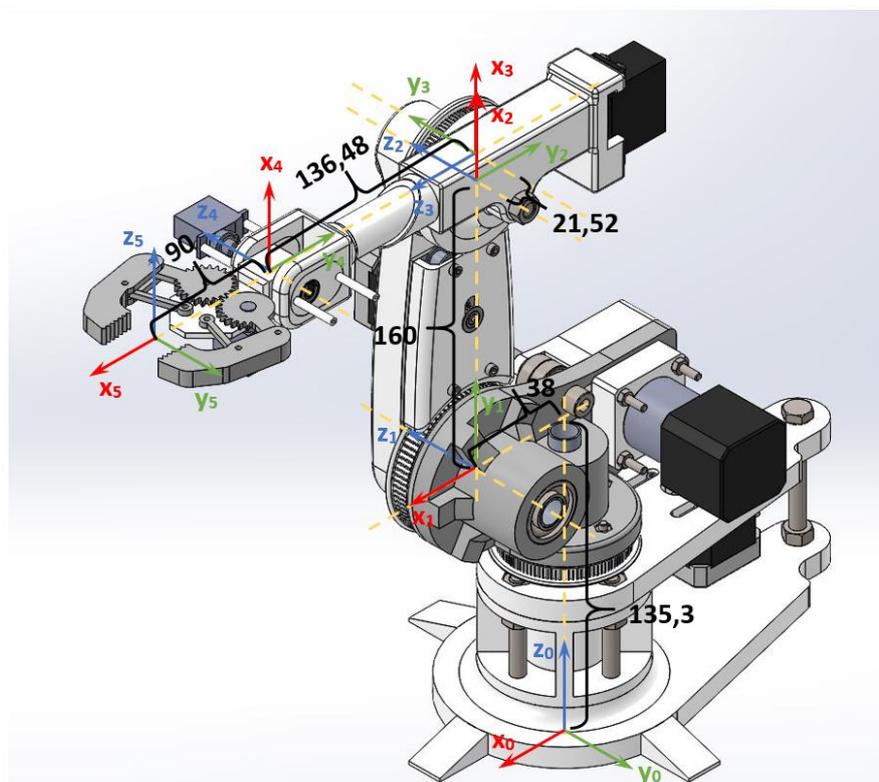


Figura 45. Comparativa de los sistemas de referencia obtenidos. Fuente: Elaboración propia.

7.2 MODELO CINEMÁTICO INVERSO

El modelo cinemático inverso permite obtener el valor de los parámetros articulares del robot a partir de la posición y orientación deseada dentro del espacio de trabajo del robot

Para la resolución de este robot, se empleará un desacople de las articulaciones de la muñeca (encargadas de la orientación) facilitando en gran medida los cálculos y el análisis.

Teniendo en cuenta que los tres ejes de las articulaciones orientativas (q_4 y q_5) están alineados entre sí (en el eje del antebrazo), se podrá aplicar el desacople del órgano terminal. Esto permite calcular la traslación hasta el punto deseado en función de los ángulos q_1 , q_2 y q_3 . Una vez resuelto el problema de traslación, podrá resolverse el desacople para obtener la orientación que debe tomar la muñeca y así obtener la solución completa del problema cinemático inverso.

El cálculo de la traslación se realiza para posicionar la muñeca, mediante la cual se obtendrá el componente de orientación (calculando q_4 y q_5). Sin embargo, al tener un órgano terminal, este añadirá una cierta distancia extra en nuestro eje X (d_5). Para poder resolver correctamente el problema de la traslación respecto a la posición de la muñeca, deberemos restar esta distancia a la matriz de posición final (T_{0-5}) y emplear un cambio de variable, el cual deberá deshacerse una vez realizados todos los cálculos correspondientes para obtener la solución final.

$$T_{muñeca} = T_{0-5} \cdot transl(-d_5, 0, 0) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & -d_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x - d_5 n_x \\ n_y & o_y & a_y & p_y - d_5 n_y \\ n_z & o_z & a_z & p_z - d_5 n_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} ? & ? & ? & m_x \\ ? & ? & ? & m_y \\ ? & ? & ? & m_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para resolver el problema de traslación, pueden emplearse diversos métodos, como métodos numéricos, el método de la transformación inversa, el método algebraico despejando directamente, o el método geométrico. En este último, analizando la geometría del robot, se puede obtener una solución de forma directa.

Durante la realización del proyecto se intentó resolver el problema de la traslación empleando el método analítico despejando directamente, sin embargo, debido al alto nivel computacional requerido para realizar dichos cálculos, se terminó optando por la última opción nombrada.

En primer lugar, para la resolución por el método geométrico, se estudiarán las distintas articulaciones, correspondientes a los ángulos q_1 , q_2 y q_3 que presenta el robot. El problema de posicionamiento se resolverá logrando posicionar la muñeca en el punto al cual se desea llegar con el robot (y más adelante, deshaciendo el cambio de variable para tener en cuenta la longitud del órgano terminal). Para esto, se expresarán los ángulos q_1 , q_2 y q_3 en función de las variables m_x , m_y y m_z definidas anteriormente.

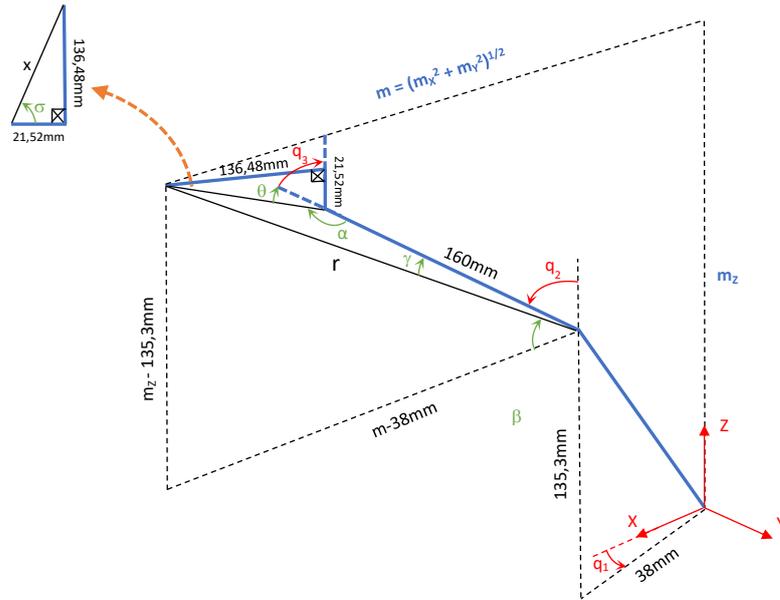


Figura 46. Representación del modelo geométrico del robot junto a sus incógnitas (azul), parámetros D-H (rojo) y elementos auxiliares (verde). Fuente: Elaboración propia.

Nota adicional:

$$m = \sqrt{m_x^2 + m_y^2}$$

$$r = \sqrt{(m_z - 135.5)^2 + (m - 38)^2}$$

$$x = \sqrt{136.48^2 + 21.52^2} = 138.17 \text{ mm}$$

$$\sigma = \cos^{-1}\left(\frac{21.52}{138.17}\right) = 1.41 \text{ rad}$$

A partir del esquema de la **Figura 46**, empleando trigonometría, se obtienen los valores de q_1 , q_2 y q_3 . Cabe destacar que la solución obtenida está definida en 4 configuraciones distintas para el brazo, las posiciones codo arriba y codo abajo para los ángulos q_2 y q_3 , y las orientaciones q_1 y $q_1 + 180^\circ$, mediante las cuales la muñeca llegará al mismo punto del espacio de trabajo.

$$\left\{ \begin{array}{l} q_1 = \tan^{-1}\left(\frac{m_y}{m_x}\right) \\ q_1 = \tan^{-1}\left(\frac{-m_y}{-m_x}\right) = \tan^{-1}\left(\frac{m_y}{m_x}\right) + 180^\circ \end{array} \right.$$

$$\left\{ \begin{array}{l} \beta = \tan^{-1}\left(\frac{m_z - 135.5}{m - 38}\right) = \tan^{-1}\left(\frac{m_z - 135.5}{\sqrt{m_x^2 + m_y^2} - 38}\right) \\ 138.17^2 = r^2 + 160^2 - 2 \cdot r \cdot 160 \cdot \cos(\gamma) \rightarrow \gamma = \cos^{-1}\left(\frac{138.17^2 - r^2 - 160^2}{-2 \cdot r \cdot 160}\right) \end{array} \right.$$

$$\begin{cases} \text{Codo Arriba} \rightarrow \mathbf{q}_2 = -\frac{\pi}{2} + (\beta + \gamma) \\ \text{Codo Abajo} \rightarrow \mathbf{q}_2 = -\frac{\pi}{2} + (\beta - \gamma) \end{cases}$$

$$r^2 = 160^2 + 138.17^2 - 2 \cdot 160 \cdot 138.17 \cdot \cos(\alpha)$$

$$\alpha = \cos^{-1}\left(\frac{r^2 - 160^2 - 138.17^2}{-2 \cdot 160 \cdot 138.17}\right)$$

$$\theta = \pi - \alpha$$

$$\begin{cases} \text{Codo Arriba} \rightarrow \mathbf{q}_3 = 1.41 - \theta \\ \text{Codo Abajo} \rightarrow \mathbf{q}_3 = \theta - 1.41 \end{cases}$$

Una vez obtenidos los valores de q_1 , q_2 y q_3 en función de las variables esquemáticas de la **Figura 46**, se deberán sustituir sus valores para expresarlos en función de m_x , m_y y m_z y por último deshacer el cambio de variable para obtener q_1 , q_2 y q_3 dependientes de los elementos de la matriz del punto deseado. Sin embargo, dado que las ecuaciones obtenidas serían extremadamente extensas, se ha optado por mantener las expresiones de los ángulos como aparecen en los cálculos anteriores (más fácilmente interpretables).

En segundo lugar, se deberá resolver el problema del desacople para así obtener q_4 y q_5 , aportando el resultado referente a la orientación del órgano terminal. La forma de obtener estos dos ángulos es a partir de la matriz de rotación R_{3-5} , la cual estará expresada en función de estos. Para su obtención, se emplearán propiedades básicas de las matrices.

$$[R_{0-3}] \cdot [R_{3-5}] = [R_{0-5}] \rightarrow [R_{3-5}] = [R_{0-3}]^{-1} \cdot [R_{0-5}]$$

Teniendo en cuenta que conocemos el punto de destino, la matriz R_{0-5} será conocida, y como ya hemos expresado las variables de traslación en función del punto de destino, también será conocida R_{0-3} . Ahora únicamente se deberá expresar q_4 y q_5 en función de los parámetros del punto deseado resolviendo la siguiente ecuación e igualándola a las anteriores.

$$\begin{aligned} [R_{3-5}] &= [R_{3-4}][R_{4-5}] = \\ &\begin{bmatrix} \cos(q_4) & 0 & -\sin(q_4) \\ \sin(q_4) & 0 & \cos(q_4) \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos(q_5 - 90) & 0 & -\sin(q_5 - 90) \\ \sin(q_5 - 90) & 0 & \cos(q_5 - 90) \\ 0 & -1 & 0 \end{bmatrix} = \\ &\begin{bmatrix} \cos(q_4)\cos(q_5 - 90) & \sin(q_4) & -\cos(q_4)\sin(q_5 - 90) \\ \sin(q_4)\cos(q_5 - 90) & -\cos(q_4) & -\sin(q_4)\sin(q_5 - 90) \\ -\sin(q_5 - 90) & 0 & -\cos(q_5 - 90) \end{bmatrix} \end{aligned}$$

Dado que la multiplicación de las matrices R_{0-3} inversa y la matriz de rotación R_{0-5} en función de los parámetros del punto deseado son muy extensas, se ha optado por expresarlas en función de unas variables r , mientras que el resultado obtenido en Matlab no se ha incluido en la memoria.

$$[R_{0-3}]^{-1} \cdot [R_{0-5}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

De esta forma, la ecuación queda:

$$\begin{bmatrix} \cos(q_4) \cos(q_5 - 90) & \sin(q_4) & -\cos(q_4) \sin(q_5 - 90) \\ \sin(q_4) \cos(q_5 - 90) & -\cos(q_4) & -\sin(q_4) \sin(q_5 - 90) \\ -\sin(q_5 - 90) & 0 & -\cos(q_5 - 90) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Seleccionamos ahora las ecuaciones correspondientes a los elementos (1,1), (2,1), (3,3) para despejar las variables q_4 y q_5 en función de las r .

$$\begin{cases} r_{11} = \cos(q_4) \cos(q_5 - 90) \\ r_{21} = \sin(q_4) \cos(q_5 - 90) \\ r_{33} = -\cos(q_5 - 90) \end{cases}$$

Dividiendo la segunda ecuación entre la primera, obtenemos:

$$q_4 = \tan^{-1} \left(\frac{r_{21}}{r_{11}} \right)$$

Despejando de la tercera obtenemos:

$$q_5 = \cos^{-1}(-r_{33}) - 90$$

A partir de los resultados obtenidos en los cálculos de traslación y orientación, la cinemática inversa puede agruparse en 4 distintas configuraciones, como puede verse en la **Tabla 4**.

Configuración	q1	q2	q3	q4	q5
Conf.1	Frente	Codo Arriba	Codo Arriba	q4	q5
Conf.2	Frente	Codo Abajo	Codo Abajo	q4	q5
Conf.3	Espalda	Codo Arriba	Codo Arriba	q4	q5
Conf4	Espalda	Codo Abajo	Codo Abajo	q4	q5

Tabla 4. Posibles configuraciones de las articulaciones.

$$\left\{ \begin{array}{l} q_1 \left\{ \begin{array}{l} \text{Frente} \rightarrow q_1 = \tan^{-1} \left(\frac{m_y}{m_x} \right) \\ \text{Espalda} \rightarrow q_1 = \tan^{-1} \left(\frac{-m_y}{-m_x} \right) = \tan^{-1} \left(\frac{m_y}{m_x} \right) + 180^\circ \end{array} \right. \\ q_2 \left\{ \begin{array}{l} \text{Codo Arriba} \rightarrow q_2 = -\frac{\pi}{2} + (\beta + \gamma) \\ \text{Codo Abajo} \rightarrow q_2 = -\frac{\pi}{2} + (\beta - \gamma) \end{array} \right. \\ q_3 \left\{ \begin{array}{l} \text{Codo Arriba} \rightarrow q_3 = 1.41 - \theta \\ \text{Codo Abajo} \rightarrow q_3 = \theta - 1.41 \end{array} \right. \\ q_4 = \tan^{-1} \left(\frac{r_{21}}{r_{11}} \right) \\ q_5 = \cos^{-1}(-r_{33}) - 90 \end{array} \right.$$

Con el fin de verificar los cálculos realizados para la cinemática inversa, se tomaron unos valores determinados de q_1 , q_2 , q_3 , q_4 y q_5 . Con dichos valores, se calculó la matriz de transformación T_{0-5} que indica la posición y orientación del punto deseado.

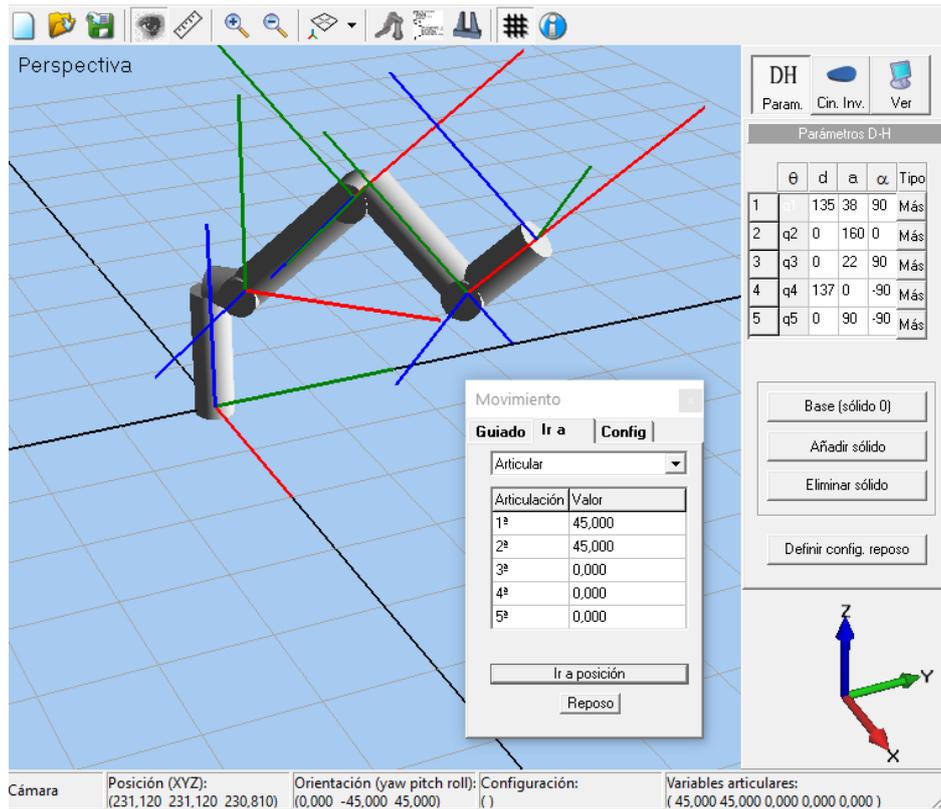


Figura 47. Valores y posición de las articulaciones del robot seleccionados para la verificación. Fuente: Elaboración propia.

Los valores dados a q_1 , q_2 , q_3 , q_4 y q_5 fueron 45° , -45° , 0° , 0° y 90° , respectivamente. Añadiendo a estos ángulos los iniciales de la posición de reposo del robot, da como resultados los ángulos de la Figura 47. La matriz T_{0-5} obtenida fue:

$$T_{0-5} = \begin{bmatrix} 0.500 & 0.7071 & 0.500 & 230.8701 \\ 0.500 & -0.7071 & 0.500 & 230.8701 \\ 0.7071 & 0 & -0.7071 & 230.7877 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A partir de esta matriz de transformación se realizó una comprobación inductiva de los cálculos realizados. Se calcularon los valores de las q mediante la matriz T_{0-5} obtenida y las ecuaciones calculadas.

En primer lugar, para el cálculo de los ángulos q_1 , q_2 y q_3 , restando la longitud del órgano terminal, se obtuvieron los siguientes valores de m_x , m_y y m_z :

$$T_{muñeca} = T_{0-5} \cdot transl(0,0,-90) = \begin{bmatrix} 0.50 & 0.71 & 0.50 & 230.870 \\ 0.50 & -0.71 & 0.50 & 230.870 \\ 0.71 & 0 & -0.71 & 230.788 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -90 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.50 & 0.71 & 0.50 & 185.870 \\ 0.50 & -0.71 & 0.50 & 185.870 \\ 0.71 & 0 & -0.71 & 167.148 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{cases} m_x = 185.870mm \\ m_y = 185.870mm \\ m_z = 167.148mm \end{cases}$$

Sustituyendo los valores de m en las ecuaciones obtenidas para q_1 , q_2 y q_3 (para la primera configuración), se obtuvieron los siguientes valores:

$$m = \sqrt{m_x^2 + m_y^2} = 262.86 \text{ mm}$$

$$r = \sqrt{(m_z - 135.5)^2 + (m - 38)^2} = 227.076 \text{ mm}$$

$$x = \sqrt{136.48^2 + 21.52^2} = 138.17 \text{ mm}$$

$$\sigma = \cos^{-1}\left(\frac{21.52}{138.17}\right) = 1.41 \text{ rad}$$

$$\beta = \tan^{-1}\left(\frac{m_z - 135.5}{\sqrt{m_x^2 + m_y^2} - 38}\right) = 0.1407 \text{ rad}$$

$$\gamma = \cos^{-1}\left(\frac{138.17^2 - r^2 - 160^2}{-2 \cdot r \cdot 160}\right) = 0.6447 \text{ rad}$$

$$\alpha = \cos^{-1}\left(\frac{r^2 - 160^2 - 138.17^2}{-2 \cdot 160 \cdot 138.17}\right) = 1.7272 \text{ rad}$$

$$\theta = \pi - \alpha = 1.4144 \text{ rad}$$

$$\begin{cases} \mathbf{q}_1 = \tan^{-1}\left(\frac{m_y}{m_x}\right) = 0.7854 \text{ rad} = 45^\circ \\ \mathbf{q}_2 = -\frac{\pi}{2} + (\beta + \gamma) = -\frac{\pi}{2} + 0.7854 \text{ rad} = -45^\circ \\ \mathbf{q}_3 = 1.41 - \theta = 0 \text{ rad} = 0^\circ \end{cases}$$

Que coinciden con los valores esperados para la configuración de frente y codo arriba.

Para la obtención de q_4 y q_5 , se calcularon las matrices R_{0-3} inversa y R_{0-5} . Tomando los elementos (1,1), (2,1), (1,3) y (2,3) de la matriz resultante de multiplicar estas dos, se pudieron realizar las ecuaciones obtenidas.

$$\begin{cases} r_{11} = 1 \\ r_{21} = 0 \\ r_{33} = 1 \end{cases}$$

$$\begin{cases} \mathbf{q}_4 = \tan^{-1}\left(\frac{r_{21}}{r_{11}}\right) = 0^\circ \\ \mathbf{q}_5 = \cos^{-1}(-r_{33}) - 90 = 180 - 90 = 90^\circ \end{cases}$$

Tras analizar los resultados obtenidos, puede verificarse el correcto cálculo de la cinemática inversa.

8 PROGRAMACIÓN Y ANÁLISIS

Para poder programar el brazo robot con sistemas de posicionamiento del órgano termina, secuencias de agarre o cualquier otro tipo de programa complejo, primero se debe establecer los límites articulares del modelo. Para ello se realizaron 3 programas sencillos para analizar los destinos motores por separado y los rangos articulares de las distintas articulaciones.

El primer programa se empleó para los motores paso a paso, es decir, para las articulaciones de la base, el hombro y el codo.

Mediante este programa, conectando los motores paso a paso de uno en uno, se puede mover la articulación un cierto ángulo (o número de pasos) escribiendo en el Monitor Serial de Arduino IDE los comandos RH (giro amplio sentido horario), LH (giro corto sentido horario), RA (giro amplio sentido antihorario) y LA (giro corto sentido antihorario). Una variable entera (step) guardará la suma de los pasos realizados por el motor hasta alcanzar el máximo rango, de forma que cuando se considere que se ha alcanzado dicha posición, se deberá introducir el comando X en el Monitor Serial. Esto mostrará el total de pasos recorridos para alcanzar el máximo rango en la dirección elegida y moverá el motor de vuelta a la posición de reposo de la articulación.

Para facilitar la comprensión del programa, se introdujeron instrucciones que van apareciendo en la pantalla del Monitor Serial a lo largo de la ejecución del programa. Una vez comprobado el rango máximo en uno de los sentidos de la articulación pertinente, se podrá proceder directamente (sin reiniciar el programa) a analizar el sentido contrario.

El **Código 1**, muestra el código de programa realizado, así como los comentarios, línea a línea, de las distintas funciones y comandos empleados y su significado.

Para estudiar el máximo rango de la rotación de la muñeca (servomotor de rotación continua de 360°) se diseñó un programa similar al anterior, pero teniendo en cuenta las condiciones del servomotor.

El servomotor de rotación continua funciona mediante órdenes de avance y detección, no por posiciones. Por tanto, para estudiar su máximo rango, se estableció una variable de tiempo que indica el tiempo que debe girar el motor para alcanzar el rango deseado.

Nuevamente, para facilitar la comprensión del programa, se colocaron indicaciones en el Monitor Serial, y los comandos de movimiento son los mismos que en el programa anterior.

El **Código 2** muestra el código de programa realizado, así como los comentarios, línea a línea, de las distintas funciones y comandos empleados y su significado.

Para el máximo rango de los servos del giro de la muñeca y apertura y cierre de la pinza (servos de 180°), se diseñó un programa similar al anterior, pero teniendo en cuenta las condiciones de estos servomotores.

Los servomotores de 180° de giro funcionan mediante posicionamiento por ángulos. Por tanto, para estudiar su máximo rango, se estableció una variable de ángulo que indica el ángulo girado por el servomotor para alcanzar la posición deseada.

Nuevamente, para facilitar la comprensión del programa, se colocaron indicaciones en el Monitor Serial, y los comandos de movimiento son los mismos que en el programa anterior.

El **Código 3** muestra el código de programa realizado, así como los comentarios, línea a línea, de las distintas funciones y comandos empleados y su significado.

```
#include <Stepper.h> //Incluye la librería Stepper.h

const int stepRev = 1028; //Define la constante correspondeinte al número de pasos por vuelta del stepper
int steps, horario, antihorario; //Define las variables enteras
Stepper myStepper(stepRev, 6, 7, 8, 9); //Define el stepper y los pines de conexión
String a; //Define la variable String

void setup(){
  myStepper.setSpeed(80); //Establece la velocidad del stepper
  Serial.begin(9600); //Inicia el Monitor Serial
  horario = 0; //Establece valor 0 a horario
  antihorario = 0; //Establece valor 0 a antihorario
  Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
}

void loop(){
  if(Serial.available()){ //Se cumple si el Monitor Serial está disponible
    a = Serial.readString(); //Asigna el mensaje enviado por Monitor Serial a la variable a
    if(a.startsWith("RH")){ //Se cumple si el mensaje enviado comienza por RH
      Serial.println("Giro amplio sentido Horario"); //Imprime "Giro amplio sentido Horario" en el Monitor Serial
      steps = steps+514; //Suma 100 a la variable steps
      horario = 1; //Establece valor 1 a horario
      myStepper.step(514); //Mueve el stepper 100 pasos
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LH")){ //Se cumple si el mensaje enviado comienza por LH
      Serial.println("Giro corto sentido Horario"); //Imprime "Giro corto sentido Horario" en el Monitor Serial
      steps = steps+257; //Suma 50 a la variable steps
      horario = 1; //Establece valor 1 a horario
      myStepper.step(257); //Mueve el stepper 50 pasos
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("RA")){ //Se cumple si el mensaje enviado comienza por RA
      Serial.println("Giro amplio sentido Antihorario"); //Imprime "Giro amplio sentido Antihorario" en el Monitor Serial
      steps = steps+514; //Suma 100 a la variable steps
      antihorario = 1; //Establece valor 1 a antihorario
      myStepper.step(-514); //Mueve el stepper -100 pasos
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LA")){ //Se cumple si el mensaje enviado comienza por LA
      Serial.println("Giro corto sentido Antihorario"); //Imprime "Giro corto sentido Antihorario" en el Monitor Serial
      steps = steps+257; //Suma 50 a la variable steps
      antihorario = 1; //Establece valor 1 a antihorario
      myStepper.step(-257); //Mueve el stepper -50 pasos
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("X")){ //Se cumple si el mensaje enviado comienza por X
      Serial.println("Máximo alcanzado"); //Imprime "Máximo alcanzado" en el Monitor Serial
      Serial.print("Numero de pasos avanzado: "); //Imprime "Numero de pasos avanzado: " en el Monitor Serial
      Serial.println(steps); //Imprime la variable steps en el Monitor Serial
      if(horario==1){ //Se cumple si la variable horario vale 1
        myStepper.step(-steps); //Mueve el stpper -steps pasos
        horario = 0; //Establece valor 0 a horario
      }
      if(antihorario==1){ //Se cumple si la variable antihorario vale 1
        myStepper.step(steps); //Mueve el stepper steps pasos
        antihorario = 0; //Establece valor 0 a antihorario
      }
      delay(1000); //Espera de 1000ms
    }
  }
}
```

Código 1. Código comentado del programa de comprobación de máximo rango de los motores paso a paso. Fuente: Elaboración propia.

```

#include <Servo.h> //Incluye la librería Servo.h

int tiempo, horario, antihorario; //Define las variables enteras
Servo ServoMu; //Define el Servo
String a; //Define la variable String

void setup(){
  Serial.begin(9600); //Inicia el Monitor Serial
  pinMode(A1,OUTPUT); //Establece el pin A1 como OUTPUT
  ServoMu.attach(A1); //Establece el pin de conexión del ServoMu
  horario = 0; //Establece valor 0 a horario
  antihorario = 0; //Establece valor 0 a antihorario
  Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
}

void loop(){
  if(Serial.available()){ //Se cumple si el Monitor Serial está disponible
    a = Serial.readString(); //Asigna el mensaje enviado por Monitor Serial a la variable a
    if(a.startsWith("RH")){ //Se cumple si el mensaje enviado comienza por RH
      Serial.println("Giro amplio sentido Horario"); //Imprime "Giro amplio sentido Horario" en el Monitor Serial
      ServoMu.write(0); //Gira el servo en sentido horario a máxima velocidad
      delay(2000); //Espera de 2000ms
      ServoMu.write(90); //Detiene el giro
      tiempo = tiempo+2000; //Suma 2000 a la variable tiempo
      horario = 1; //Establece valor 1 a horario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LH")){ //Se cumple si el mensaje enviado comienza por LH
      Serial.println("Giro corto sentido Horario"); //Imprime "Giro corto sentido Horario" en el Monitor Serial
      ServoMu.write(0); //Gira el servo en sentido horario a máxima velocidad
      delay(200); //Espera de 200ms
      ServoMu.write(90); //Detiene el giro
      tiempo = tiempo+200; //Suma 200 a la variable tiempo
      horario = 1; //Establece valor 1 a horario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("RA")){ //Se cumple si el mensaje enviado comienza por RA
      Serial.println("Giro amplio sentido Antihorario"); //Imprime "Giro amplio sentido Antihorario" en el Monitor Serial
      ServoMu.write(180); //Gira el servo en sentido antihorario a máxima velocidad
      delay(2000); //Espera de 2000ms
      ServoMu.write(90); //Detiene el giro
      tiempo = tiempo+2000; //Suma 2000 a la variable tiempo
      antihorario = 1; //Establece valor 1 a antihorario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LA")){ //Se cumple si el mensaje enviado comienza por LA
      Serial.println("Giro corto sentido Antihorario"); //Imprime "Giro corto sentido Antihorario" en el Monitor Serial
      ServoMu.write(180); //Gira el servo en sentido antihorario a máxima velocidad
      delay(200); //Espera de 200ms
      ServoMu.write(90); //Detiene el giro
      tiempo = tiempo+200; //Suma 200 a la variable tiempo
      antihorario = 1; //Establece valor 1 a antihorario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("X")){ //Se cumple si el mensaje enviado comienza por X
      Serial.println("Máximo alcanzado"); //Imprime "Máximo alcanzado" en el Monitor Serial
      Serial.print("Tiempo de avance: "); //Imprime "Tiempo de avance: " en el Monitor Serial
      Serial.println(tiempo); //Imprime la variable tiempo en el Monitor Serial
      if(horario==1){ //Se cumple si la variable horario vale 1
        ServoMu.write(180); //Gira el servo en sentido antihorario a máxima velocidad
        delay(tiempo); //Espera de tiempo ms
        ServoMu.write(90); //Detiene el giro
        horario = 0; //Establece valor 0 a horario
        Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
      }
      if(antihorario==1){ //Se cumple si la variable antihorario vale 1
        ServoMu.write(0); //Gira el servo en sentido horario a máxima velocidad
        delay(tiempo); //Espera de tiempo ms
        ServoMu.write(90); //Detiene el giro
        antihorario = 0; //Establece valor 0 a antihorario
        Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
      }
      delay(1000); //Espera de 1000ms
    }
  }
}

```

Código 2. Código comentado del programa de comprobación de máximo rango del servomotor de giro continuo de 360°. Fuente: Elaboración propia.

```

#include <Servo.h>           //Incluye la librería Servo.h

int angulo, horario, antihorario; //Define las variables enteras
Servo ServoMu;             //Define el Servo
String a;                  //Define la variable String

void setup(){
  Serial.begin(9600);      //Inicia el Monitor Serial
  pinMode(A2,OUTPUT);     //Establece el pin A2 como OUTOUT
  ServoMu.attach(A2);     //Establece el pin de conexión del ServoMu
  horario = 0;            //Establece valor 0 a horario
  antihorario = 0;       //Establece valor 0 a antihorario
  ServoMu.read();        //Establece la posición actual del servo
  Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
}

void loop(){
  if(Serial.available()){ //Se cumple si el Monitor Serial está disponible
    a = Serial.readString(); //Asigna el mensaje enviado por Monitor Serial a la variable a
    if(a.startsWith("RH")){ //Se cumple si el mensaje enviado comienza por RH
      Serial.println("Giro amplio sentido Horario"); //Imprime "Giro amplio sentido Horario" en el Monitor Serial
      angulo = angulo+15; //Suma 15 a la variable angulo
      horario = 1; //Establece valor 1 a horario
      ServoMu.write(15); //Gira el servo 15 grados en sentido horario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LH")){ //Se cumple si el mensaje enviado comienza por LH
      Serial.println("Giro corto sentido Horario"); //Imprime "Giro corto sentido Horario" en el Monitor Serial
      angulo = angulo+5; //Suma 5 a la variable angulo
      horario = 1; //Establece valor 1 a horario
      ServoMu.write(5); //Gira el servo 5 grados en sentido horario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("RA")){ //Se cumple si el mensaje enviado comienza por RA
      Serial.println("Giro amplio sentido Antihorario"); //Imprime "Giro amplio sentido Antihorario" en el Monitor Serial
      angulo = angulo-15; //Suma 15 a la variable angulo
      antihorario = 1; //Establece valor 1 a antihorario
      ServoMu.write(-15); //Gira el servo 15 grados en sentido antihorario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("LA")){ //Se cumple si el mensaje enviado comienza por LA
      Serial.println("Giro corto sentido Antihorario"); //Imprime "Giro corto sentido Antihorario" en el Monitor Serial
      angulo = angulo+5; //Suma 5 a la variable angulo
      antihorario = 1; //Establece valor 1 a antihorario
      ServoMu.write(-5); //Gira el servo 5 grados en sentido antihorario
      delay(500); //Espera de 500ms
      Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
    }
    if(a.startsWith("X")){ //Se cumple si el mensaje enviado comienza por X
      Serial.println("Máximo alcanzado"); //Imprime "Máximo alcanzado" en el Monitor Serial
      Serial.print("Angulo avanzado: "); //Imprime "Angulo avanzado: " en el Monitor Serial
      Serial.println(angulo); //Imprime la variable angulo en el Monitor Serial
      if(horario==1){ //Se cumple si la variable horario vale 1
        ServoMu.write(-angulo); //Gira el servo angulo en sentido antihorario
        horario = 0; //Establece valor 0 a horario
        Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
      }
      if(antihorario==1){ //Se cumple si la variable antihorario vale 1
        ServoMu.write(angulo); //Gira el servo angulo en sentido horario
        antihorario = 0; //Establece valor 0 a antihorario
        Serial.println("Cómo te quieres mover?: "); //Imprime "Cómo te quieres mover?: " en el Monitor Serial
      }
      delay(1000); //Espera de 1000ms
    }
  }
}
}
}

```

Código 3. Código comentado del programa de comprobación de máximo rango de servomotor de giro de 180°. Fuente: Elaboración propia.

Tras analizar todos los motores se obtuvieron los rangos articulares mostrados en la **Tabla 5**.

Articulación	Máximo rango horario (°)	Máximo rango antihorario (°)
Base	180	-180
Hombro	10	-60
Codo	70	-80
Rotación muñeca	180	-180
Giro muñeca	90	180
Apertura/Cierre pinza	90	170

Tabla 5. Rangos articulares. Fuente: Elaboración propia.

Con el objetivo de verificar y analizar los rangos articulares de las distintas articulaciones (diseñado especialmente para usuarios que emplean el robot por primera vez) se ha realizado un programa en el que el usuario es capaz de seleccionar una articulación haciendo uso de un menú a través del Monitor Serial de Arduino IDE y mover el motor correspondiente hasta sus dos rangos máximos y volver a la posición de reposo.

A diferencia de los programas anteriores, en este todos los motores estarán conectados siguiendo el conexionado descrito en el punto 7. *Electrónica*. En **Código 4 y 5** se muestra el código de programa realizado, así como los comentarios, línea a línea, de las distintas funciones y comandos empleados y su significado.

```
#include <Stepper.h>           //Incluye la librería Stepper.h
#include <Servo.h>             //Incluye la librería Servo.h

//DEFINICIÓN DE CONSTANTES

const int stepRev = 200;      //Define la constante correspondiente a los pasos por vuelta del stepper
int dir=0, menu=1, opc=0;    //Define las variables enteras
char c='\0';                  //Define la variable tipo char
String words;                 //Define la variable tipo String

Stepper StepperBase(stepRev, 10, 11, 12, 13); //Define el stepper de la Base y los pines de conexión
Stepper StepperShoulder(stepRev, 6, 7, 8, 9); //Define el stepper del Hombro y los pines de conexión
Stepper StepperElbow(stepRev, 2, 3, 4, 5);    //Define el stepper del Codo y los pines de conexión
Servo rotWrist;                //Define el servo de rotación de la Muñeca
Servo Wrist;                   //Define el servo de giro de la Muñeca
Servo Gripper;                 //Define el servo de apertura y cierre de la Pinza

//VOID SETUP

void setup(){
  StepperBase.setSpeed(90);    //Establece la velocidad del motor de la Base en 90rpm
  StepperShoulder.setSpeed(90); //Establece la velocidad del motor del Hombro en 90rpm
  StepperElbow.setSpeed(90);   //Establece la velocidad del motor del Codo en 90rpm
  pinMode(A1,OUTPUT);          //Establece el pin A1 como OUTPUT
  pinMode(A2,OUTPUT);          //Establece el pin A2 como OUTPUT
  pinMode(A3,OUTPUT);          //Establece el pin A3 como OUTPUT
  rotWrist.attach(A1);         //Asigna el servo de rotación de la muñeca al pin A1
  Wrist.attach(A2);            //Asigna el servo de giro de la muñeca al pin A2
  Gripper.attach(A3);          //Asigna el servo de la pinza al pin A3
  Serial.begin(9600);          //Inicia el Monitor Serial
  Serial.println("Movimiento Articular - Arduino"); //Muestra el inicio del programa en el Monitor
}
```

Código 4. Código comentado del programa de recorrido articular, sección de definición de variables y Void setup(). Fuente: Elaboración propia.

```

//PROGRAMA PRINCIPAL

void loop(){

// --- MENÚ ---

if(menu==1){
//Se cumple si la variable menú vale 1
Serial.println(" "); //Imprime una línea en blanco
Serial.println("----- ESCOGE UNA OPCIÓN ENTRE 1 Y 3 -----"); //Imprime "----- ESCOGE UNA OPCIÓN ENTRE 1 Y 3 -----"
Serial.println("1) Motor Base"); //Imprime "1) Motor Base"
Serial.println("2) Motor Hombro"); //Imprime "2) Motor Hombro"
Serial.println("3) Motor Codo"); //Imprime "3) Motor Codo"
Serial.println("4) Rotación Muñeca"); //Imprime "4) Rotación Muñeca"
Serial.println("5) Giro Muñeca"); //Imprime "5) Giro Muñeca"
Serial.println("6) Abrir y cerrar Pinza"); //Imprime "6) Abrir y cerrar Pinza"
menu=0; //Se cambia el valor de la variable menu a 0 para evitar que se vuelva a mostrar el mensaje
}
while (Serial.available()) { //Se realiza mientras el Monitor Serial esté disponible
delay(5); //Espera de 5ms
c = Serial.read(); //Lee los caracteres
words += c; //Convierte los caracteres en una variable String
}

opc=words.toInt(); //Convierte la selección en una variable entera
// --- MOVIMIENTO DE LOS MOTORES ---

if (opc!=0){ //Se cumple si el valor introducido es distinto de 0
if(opc>1 && opc<=6){ //Se cumple si el valor introducido está entre 1 y 6 se realiza el switch
switch(opc){ //Realiza la opción seleccionada
case 1: //Pone en marcha el motor de la Base, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Motor Base en marcha <<<<<<"); //Imprime ">>>>> Motor Base en marcha <<<<<<"
StepperBase.step(-625); //Gira de 180° en sentido horario moviendo 625 pasos el motor
delay(500); //Espera de 500ms
StepperBase.step(625); //Gira de 180° en el sentido antihorario moviendo -625 pasos el motor
break; //Termina el case
case 2: //Pone en marcha el motor del Hombro, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Motor Hombro en marcha <<<<<<"); //Imprime ">>>>> Motor Hombro en marcha <<<<<<"
StepperShoulder.step(117); //Gira 10° en sentido horario moviendo 117 pasos el motor
delay(500); //Espera de 500ms
StepperShoulder.step(-800); //Gira -70° en sentido antihorario moviendo -800 pasos el motor
delay(500); //Espera de 500ms
StepperShoulder.step(750); //Vuelve al reposo subiendo 64° moviendo 750 pasos el motor
break; //Termina el case
case 3: //Pone en marcha el motor del Codo, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Motor Codo en marcha <<<<<<"); //Imprime ">>>>> Motor Codo en marcha <<<<<<"
StepperElbow.step(-200); //Gira 70° en sentido horario moviendo -200 pasos el motor
delay(500); //Espera de 500ms
StepperElbow.step(440); //Gira 154° en sentido antihorario moviendo 440 pasos el motor
delay(500); //Espera de 500ms
StepperElbow.step(-240); //Vuelve al reposo subiendo 84° moviendo -240 pasis el motor
break; //Termina el case
case 4: //Pone en marcha el motor de rotación de la Muñeca, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Rotación Muñeca en marcha <<<<<<"); //Imprime ">>>>> Rotación Muñeca en marcha <<<<<<"
rotWrist.write(0); //Mueve el motor en sentido horario a velocidad máxima
delay(1600); //Espera de 1500ms, tiempo para completar una vuelta por el motor
rotWrist.write(90); //Detiene el motor
delay(500); //Espera de 500ms
rotWrist.write(180); //Mueve el motor en sentido antihorario a velocidad máxima
delay(1600); //Vuelve a la posición de reposo mediante una espera de 1500ms
rotWrist.write(90); //Detiene el motor
break; //Termina el case
case 5: //Pone en marcha el motor de giro de la Muñeca, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Giro Muñeca en marcha <<<<<<"); //Imprime ">>>>> Giro Muñeca en marcha <<<<<<"
Gripper.write(90); //Mueve el motor a la posición 90°
delay(500); //Espera de 500ms
Gripper.write(180); //Mueve el motor a la posición 180° (el reposo)
break; //Termina el case
case 6: //Pone en marcha el motor de apertura y cierre de la Pinza, recorriendo su rango articular y volviendo al reposo
Serial.println(">>>>> Motor Pinza en marcha <<<<<<"); //Imprime ">>>>> Motor Pinza en marcha <<<<<<"
Gripper.write(90); //Mueve el motor a la posición 90°
delay(500); //Espera de 500ms
Gripper.write(170); //Mueve el motor a la posición 170° (el reposo)
break; //Termina el case
}
}
else{ //Opción que corre si el número intorducido no está entre 1 y 6
Serial.println("OPCIÓN INVALIDA!!!! INTENTE NUEVAMENTE"); //Imprime "OPCIÓN INVALIDA!!!! INTENTE NUEVAMENTE"
}
menu=1; //Establece el valor de menu en 1 para volver a mosotrar el Menú
opc=0; //Reinicia la variable opc
}
words = ""; // Reinicia la variable word
}
}

```

Código 5. Código comentado del programa de recorrido articular, sección de Void loop(). Fuente: Elaboración propia.

Para este programa se emplearon las librerías *Stepper.h* y *Servo.h* para controlar los distintos motores del brazo. Con el objetivo de facilitar la comprensión del programa y su uso, se diseñó un menú que muestre las distintas opciones, es decir, que articulación se desea mover, el usuario puede introducir el número de la opción deseada en el Monitor Serial y mediante un *switch-case* el programa activa la opción seleccionada por el usuario. Tras mover el motor de la articulación seleccionada a lo largo de todo su rango, el menú vuelve a aparecer, permitiendo seleccionar la misma u otra de las opciones.

En el caso del movimiento del motor del hombro, se ha puesto un total de 750 pasos para volver a la posición de reposo, en lugar de los 683 pasos que debería haberse puesto, ya que durante los ensayos se probó que el motor necesitaba un 10% más de pasos para alcanzar el reposo desde el máximo rango antihorario.

Para lograr que el menú no reapareciese en el Monitor Serial durante cada *loop* del programa, así como las funciones del *switch*, se emplearon sistemas de variables de control, de forma que mediante funciones condicionales *if*, que verificaban los valores de las variables de control, permitían la ejecución de las funciones. Un ejemplo para analizar de este programa es el del menú y la variable de control *menu*.

```
if(menu==1) {
  Serial.println(" ");
  Serial.println("----- ESCOGE UNA OPCIÓN ENTRE 1 Y 3 -----");
  Serial.println("1) Motor Base");
  Serial.println("2) Motor Hombro");
  Serial.println("3) Motor Codo");
  Serial.println("4) Rotación Muñeca");
  Serial.println("5) Giro Muñeca");
  Serial.println("6) Abrir y cerrar Pinza");
  menu=0;
}
```

Código 6. Ejemplo del sistema de variables de control en el caso del menú principal. Fuente: Elaboración propia.

Como puede observarse, se le asigna valor 1 a *menu* durante su definición, de forma que cuando el programa llega al *if* de control, solo se imprimirá el menú si esta variable vale 1. Al terminar de imprimir el menú, se cambia el valor de la variable a 0, de forma que cuando el *loop* se repite, este *if* ya no se cumple y no vuelve a imprimirse todo de nuevo. El valor de *menu* volverá a establecerse como 1 cuando el usuario haya introducido algún valor en el Monitor Serial, bien seleccionando alguno de los *case* del *switch* o introduciendo un número inválido, de forma que el menú volverá a aparecer en el Monitor Serial y se podrá volver a seleccionar una de sus opciones.

Por último, se ha programado un sistema de posicionamiento del órgano terminal del robot, con el objetivo de verificar los cálculos cinemáticos calculados para las articulaciones de la base, el hombro y el codo.

Mediante este programa, únicamente se ha tratado de mandar órdenes al robot para posicionar el órgano terminal en varios puntos deseados por el usuario, sin embargo, no se ha tenido en cuenta en este caso la orientación de la herramienta ni su uso pertinente. A pesar de su correcto funcionamiento, como se pudo comprobar mediante el programa de rango articular, se ha optado por no emplear los motores de rotación y giro de la muñeca, ni de apertura y cierre de la pinza. Esto simplifica la programación y, a pesar de no emplear todos los motores del robot, también sirve para verificar su correcto funcionamiento, ya que los motores destinados a la

orientación de la herramienta únicamente son útiles a la hora de desempeñar tareas de pick&place, en los que la pinza o herramienta del robot debe estar en una orientación determinada para agarrar cierto objeto evitando que esta sufra daños o para lograr el movimiento más eficiente posible. En este caso, dado que únicamente se busca posicionar la herramienta en un punto preciso, no es realmente necesario el uso de estos motores.

Cabe destacar que de ser necesario, en futuros proyectos relativos al brazo robot, en lo que concierne a tareas del estilo de agarre de objetos, podrían ser empleados los tres motores restantes, los cuales cumplirían con su cometido, tanto teórica como prácticamente, tal como ha podido observarse durante los cálculos y verificación de la cinemática inversa del robot, del apartado **8.2 Inversa**, y durante la programación y comprobación del rango máximo de las articulaciones del robot mediante el programa de rango articular, del apartado **9 Programación y análisis**.

El programa diseñado para el posicionamiento del órgano terminal del robot consiste en una interfaz que permite al usuario introducir 3 puntos del espacio, en milímetros, a los cuales desea que el robot desplace su herramienta. Tras verificar que estos tres puntos se encuentran dentro del rango articular verificado con anterioridad, el programa procede a mover el robot a estos tres puntos, trazando un triángulo en el espacio, y volviendo finalmente a su posición de reposo, desde la cual pueden volver a introducir nuevos puntos de destino. En **Código 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 y 18** se muestra el código de programa realizado, así como los comentarios, línea a línea, de las distintas funciones y comandos empleados y su significado.

```

#include <Stepper.h> //Incluye librería Stepper.h

int menu=1, pix=0, piy=0, plz=0, p2x=0, p2y=0, p2z=0, p3x=0, p3y=0, p3z=0, opc=0; //Definición de variables enteras relativas a los puntos y al menú
int q1p1pasos, q2p1pasos, q3p1pasos, q1p2pasos, q2p2pasos, q3p2pasos, q1p3pasos, q2p3pasos; //Definición de variables enteras referentes a los pasos
int q3p3pasos, q1p3pasos, q2p3pasos, q3p3pasos; //Definición de variables enteras referentes a los pasos
int control1=1, control2=1, control3=1, control4=1, control5=1; //Definición de variables enteras referentes a las variables de control
int control6=1, control7=1, control8=1, control9=1; //Definición de variables enteras referentes a las variables de control
float m1, r1, q1p1, Beta1, Gamma1, q2p1, Alfa1, Tetta1, q3p1, q1p1float, q2p1float, q3p1float; //Definición de variables tipo float del punto 1
float m2, r2, q1p2, Beta2, Gamma2, q2p2, Alfa2, Tetta2, q3p2; //Definición de variables tipo float del punto 2
float q1p2NR, q2p2NR, q3p2NR, q1p2float, q2p2float, q3p2float; //Definición de variables tipo float del punto 2
float m3, r3, q1p3, Beta3, Gamma3, q2p3, Alfa3, Tetta3, q3p3, q1p3NR, q2p3NR, q3p3NR; //Definición de variables tipo float del punto 3
float q1p3float, q2p3float, q3p3float, q1p3reposito, q2p3reposito, q3p3reposito; //Definición de variables tipo float del punto 3
const float pi=3.1416; //Definición de constante (número pi)
char c='\0'; //Definición de variable tipo char
String words; //Definición de variable tipo String

Stepper StepperBase(200, 10, 11, 12, 13); //Definición del motor paso a paso de la Base y sus pines
Stepper StepperShoulder(200, 6, 7, 8, 9); //Definición del motor paso a paso del Hombro y sus pines
Stepper StepperElbow(200, 2, 3, 4, 5); //Definición del motor paso a paso del Codo y sus pines

void setup() {
  StepperBase.setSpeed(90); //Establece la velocidad del motor de la Base en 90rpm
  StepperShoulder.setSpeed(90); //Establece la velocidad del motor del Hombro en 90rpm
  StepperElbow.setSpeed(90); //Establece la velocidad del motor del Codo en 90rpm
  Serial.begin(9600); //Inicializar el Monitor Serial
  Serial.println("Movimiento Articular - Arduino"); //Imprime en el Monitor Serial el título del programa
}

void loop() {

  //***** MENÚ PUNTO 1 *****
  //*****

  if(menu==1){ //Se cumple el if solo si la variable menu vale 1
    Serial.println(" "); //Deja una línea en blanco
    Serial.println("----- TRAZADO DE TRIÁNGULO -----"); //Imprime la función del programa
    Serial.println("Introduce las coordenadas X, Y y Z del primer punto"); //Imprime la petición de introducir las coordenadas del primero punto al que llevar el robot
    menu=0; //Establece como 0 el valor de la variable menu para evitar que el if se vuelva a realizar
    //hasta que el todas las órdenes pertinentes se hayan realizado
  }

  if(control1==1){ //Se cumple el if solo si la variable control1 vale 1
    while(Serial.available()){ //Espera a que se introduzca algo en el Monitor Serial
      delay(5); //Tiempo de espera de 5ms
      c = Serial.read(); //Guarda en la variable c (tipo char) los distintos caracteres introducidos en el Monitor Serial
      words += c; //Une los distintos caracteres de c y los introduce en la variable words (tipo String)
    }
    pix=words.toInt(); //Transforma la variable word (tipo String) en pix (tipo int)
    //para tener la coordenada X del primer punto como un número con el que realizar los cálculos pertinentes
    delay(5); //Tiempo de espera de 5ms
    words = ""; //Limpia la variable words (tipo String) para poder emplearla para los próximos datos introducidos en el Monitor Serial
  }

  if(pix!=0){ //Se cumple el if solo si pix es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada X del punto

    if(control1==1){ //Se cumple el if solo si la variable control1 vale 1
      Serial.print("Coordenada X: "); //Imprime "Coordenada X: "
      Serial.println(pix); //Imprime el valor de la coordenada X introducido
    }
    control1=0; //Establece como 0 el valor de la variable control1 para evitar que se realice el anterior if de nuevo hasta que termine la función del programa
  }
}

```

Código 7. Código comentado del programa de Trazado de triángulo, sección de definición de variables, Void setup() y Menú del punto 1. Fuente: Elaboración propia.

```

if(control2==1){
    while(Serial.available()){
        delay(5);
        c = Serial.read();
        words += c;
    }
    ply=words.toInt();
    delay(5);
    words = "";
}

if(ply!=0){ //Se cumple el if solo ply es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Y del punto

    if(control2==1){
        Serial.print("Coordenada Y: ");
        Serial.println(ply);
    }
    control2=0;

    if(control3==1){
        while(Serial.available()){
            delay(5);
            c = Serial.read();
            words += c;
        }
        plz=words.toInt();
        delay(5);
        words="";
    }

    if(plz!=0){ //Se cumple el if solo plz es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Z del punto

        if(control3==1){
            Serial.print("Coordenada Z: ");
            Serial.println(plz);
            Serial.println(" ");
            Serial.println("Introduce las coordenadas X, Y y Z del segundo punto");
        }
        control3=0;

        //*****
        //***** MENÚ PUNTO 2 *****
        //*****

        if(control4==1){
            while(Serial.available()){
                delay(5);
                c = Serial.read();
                words += c;
            }
            px=words.toInt();
            delay(5);
            words = "";
        }

        if(px!=0){ //Se cumple el if solo px es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada X del punto

            if(control4==1){
                Serial.print("Coordenada X: ");
                Serial.println(px);
            }
            control4=0;

            if(control5==1){
                while(Serial.available()){
                    delay(5);
                    c = Serial.read();
                    words += c;
                }
                p2y=words.toInt();
                delay(5);
                words = "";
            }

            if(p2y!=0){ //Se cumple el if solo p2y es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Y del punto

                if(control5==1){
                    Serial.print("Coordenada Y: ");
                    Serial.println(p2y);
                }
                control5=0;

                if(control6==1){
                    while(Serial.available()){
                        delay(5);
                        c = Serial.read();
                        words += c;
                    }
                    p2z=words.toInt();
                    delay(5);
                    words="";
                }

                if(p2z!=0){ //Se cumple el if solo p2z es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Z del punto

```

Código 8. Código comentado del programa de recorrido articular, sección Menú del punto 1 y Menú del punto 2. Fuente: Elaboración propia.

```

if(control6==1){
    Serial.print("Coordenada Z: ");
    Serial.println(p2z);
    Serial.println(" ");
    Serial.println("Introduce las coordenadas X, Y y Z del tercer punto");
}
control6=0; //Establece como 0 el valor de la variable control3 para evitar que se realice el anterior if de nuevo hasta que termine la función del programa
//*****
//***** MENÚ PUNTO 3 *****
//*****

if(control7==1){
    while(Serial.available()){
        delay(5);
        c = Serial.read();
        words += c;
    }
    p3x=words.toInt();
    delay(5);
    words = "";
}

if(p3x!=0){ //Se cumple el if solo p3x es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada X del punto

    if(control7==1){
        Serial.print("Coordenada X: ");
        Serial.println(p3x);
    }
    control7=0; //Establece como 0 el valor de la variable control1 para evitar que se realice el anterior if de nuevo hasta que termine la función del programa

if(control8==1){
    while(Serial.available()){
        delay(5);
        c = Serial.read();
        words += c;
    }
    p3y=words.toInt();
    delay(5);
    words = "";
}

if(p3y!=0){ //Se cumple el if solo p3y es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Y del punto

    if(control8==1){
        Serial.print("Coordenada Y: ");
        Serial.println(p3y);
    }
    control8=0; //Establece como 0 el valor de la variable control2 para evitar que se realice el anterior if de nuevo hasta que termine la función del programa

if(control9==1){
    while(Serial.available()){
        delay(5);
        c = Serial.read();
        words += c;
    }
    p3z=words.toInt();
    delay(5);
    words="";
}

if(p3z!=0){ //Se cumple el if solo p3z es distinto de 0, es decir, solo se realiza el if y continua el programa si se ha introducido un valor para la coordenada Z del punto

if(control9==1){
    Serial.print("Coordenada Z: ");
    Serial.println(p3z);
    Serial.println(" ");
    Serial.println("Para realizar los cálculos pulse 1");
}
control9=0; //Establece como 0 el valor de la variable control3 para evitar que se realice el anterior if de nuevo hasta que termine la función del programa

// ---- CÁLCULOS ----

while(Serial.available()){
    delay(5);
    c = Serial.read();
    words += c;
}

opc=words.toInt();

if(opc!=0){
    if(opc==1){

```

Código 9. Código comentado del programa de recorrido articular, sección de Menú del punto 2, Menú del punto 3 y Cálculos. Fuente: Elaboración propia.

```

// --- PUNTO 1 ---

m1 = sqrt(pow(plx,2)+pow(ply,2)); //Cálculo de la variable m1
r1 = sqrt(pow((plz-135.5),2)+pow((m1-38),2)); //Cálculo de la variable r1
q1p1 = atan2(ply,plx); //Valor en radianes del ángulo que debe desplazarse la base
Beta1 = atan((plz-135.5)/(m1-38)); //Cálculo de la variable Beta1
Gamma1 = acos((pow(227.5,2)-pow(r1,2)-pow(160,2))/(-2*r1*160)); //Cálculo de la variable Gamma1
q2p1 = -pi/2+(Beta1+Gamma1); //Valor en radianes del ángulo que debe desplazarse el hombro
Alfa1 = acos((pow(r1,2)-pow(160,2)-pow(227.5,2))/(-2*160*227.5)); //Cálculo de la variable Alfa1
Tetta1 = pi-Alfa1; //Cálculo de la variable Tetta1
q3p1 = 1.48-Tetta1; //Valor en radianes del ángulo que debe desplazarse el codo

q1pfloat=q1p1*(180/pi)*3.472; //Cálculo para la obtención de los pasos del motor de la base. La constante 3.472 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/360)
q2pfloat=q2p1*(180/pi)*11.667; //Cálculo para la obtención de los pasos del motor del hombro. La constante 14.278 se obtiene de
//multiplicar la relación de transmisión (5) por el cambio de unidades de grados a pasos (700/300)
q3pfloat=q3p1*(180/pi)*2.857; //Cálculo para la obtención de los pasos del motor del codo. La constante 3.472 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/437.5)

q1ppasos = (int) q1pfloat; //Convierte los pasos en una variable entera para poder introducirse al motor
q2ppasos = (int) q2pfloat; //Convierte los pasos en una variable entera para poder introducirse al motor
q3ppasos = (int) q3pfloat; //Convierte los pasos en una variable entera para poder introducirse al motor

// --- PUNTO 2 ---

m2 = sqrt(pow(p2x,2)+pow(p2y,2)); //Cálculo de la variable m2
r2 = sqrt(pow((p2z-135.5),2)+pow((m2-38),2)); //Cálculo de la variable r2
q1p2 = atan2(p2y,p2x); //Valor en radianes del ángulo que debe desplazarse la base de la posición de reposo al punto 2
Beta2 = atan((p2z-135.5)/(m2-38)); //Cálculo de la variable Beta2
Gamma2 = acos((pow(227.5,2)-pow(r2,2)-pow(160,2))/(-2*r2*160)); //Cálculo de la variable Gamma2
q2p2 = -pi/2+(Beta2+Gamma2); //Valor en radianes del ángulo que debe desplazarse el hombro de la posición de reposo al punto 2
Alfa2 = acos((pow(r2,2)-pow(160,2)-pow(227.5,2))/(-2*160*227.5)); //Cálculo de la variable Alfa2
Tetta2 = pi-Alfa2; //Cálculo de la variable Tetta2
q3p2 = 1.48-Tetta2; //Valor en radianes del ángulo que debe desplazarse el codo de la posición de reposo al punto 2

q1p2NR = q1p2-q1p1; //Ángulo que debe desplazarse la base desde el punto 1 al punto 2
q2p2NR = q2p2-q2p1; //Ángulo que debe desplazarse el hombro desde el punto 1 al punto 2
q3p2NR = q3p2-q3p1; //Ángulo que debe desplazarse el codo desde el punto 1 al punto 2

q1p2float=q1p2NR*(180/pi)*3.472; //Cálculo para la obtención de los pasos del motor de la base. La constante 3.472 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/360)
q2p2float=q2p2NR*(180/pi)*11.667; //Cálculo para la obtención de los pasos del motor del hombro. La constante 11.667 se obtiene de
//multiplicar la relación de transmisión (5) por el cambio de unidades de grados a pasos (700/300)
q3p2float=q3p2NR*(180/pi)*2.857; //Cálculo para la obtención de los pasos del motor del codo. La constante 2.857 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/437.5)

q1p2pasos = (int) q1p2float; //Convierte los pasos en una variable entera para poder introducirse al motor
q2p2pasos = (int) q2p2float; //Convierte los pasos en una variable entera para poder introducirse al motor
q3p2pasos = (int) q3p2float; //Convierte los pasos en una variable entera para poder introducirse al motor

// --- PUNTO 3 ---

m3 = sqrt(pow(p3x,2)+pow(p3y,2)); //Cálculo de la variable m3
r3 = sqrt(pow((p3z-135.5),2)+pow((m3-38),2)); //Cálculo de la variable r3
q1p3 = atan2(p3y,p3x); //Valor en radianes del ángulo que debe desplazarse la base de la posición de reposo al punto 3
Beta3 = atan((p3z-135.5)/(m3-38)); //Cálculo de la variable Beta3
Gamma3 = acos((pow(227.5,2)-pow(r3,2)-pow(160,2))/(-2*r3*160)); //Cálculo de la variable Gamma3
q2p3 = -pi/2+(Beta3+Gamma3); //Valor en radianes del ángulo que debe desplazarse el hombro de la posición de reposo al punto 3
Alfa3 = acos((pow(r3,2)-pow(160,2)-pow(227.5,2))/(-2*160*227.5)); //Cálculo de la variable Alfa3
Tetta3 = pi-Alfa3; //Cálculo de la variable Tetta3
q3p3 = 1.48-Tetta3; //Valor en radianes del ángulo que debe desplazarse el codo de la posición de reposo al punto 3

q1p3NR = q1p3-q1p2; //Ángulo que debe desplazarse la base desde el punto 2 al punto 3
q2p3NR = q2p3-q2p2; //Ángulo que debe desplazarse el hombro desde el punto 2 al punto 3
q3p3NR = q3p3-q3p2; //Ángulo que debe desplazarse el codo desde el punto 2 al punto 3

q1p3float=q1p3NR*(180/pi)*3.472; //Cálculo para la obtención de los pasos del motor de la base. La constante 3.472 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/360)
q2p3float=q2p3NR*(180/pi)*11.667; //Cálculo para la obtención de los pasos del motor del hombro. La constante 14.278 se obtiene de
//multiplicar la relación de transmisión (5) por el cambio de unidades de grados a pasos (1028/360)
q3p3float=q3p3NR*(180/pi)*2.857; //Cálculo para la obtención de los pasos del motor del codo. La constante 3.472 se obtiene de
//multiplicar la relación de transmisión (6.25) por el cambio de unidades de grados a pasos (200/360)

q1p3pasos = (int) q1p3float; //Convierte los pasos en una variable entera para poder introducirse al motor
q2p3pasos = (int) q2p3float; //Convierte los pasos en una variable entera para poder introducirse al motor
q3p3pasos = (int) q3p3float; //Convierte los pasos en una variable entera para poder introducirse al motor

q1p3reposo=q1p3*(180/pi)*3.472; //Cálculo para la obtención de los pasos de la base desde el punto 3 hasta la posición de reposo
q2p3reposo=q2p3*(180/pi)*11.667; //Cálculo para la obtención de los pasos del hombro desde el punto 3 hasta la posición de reposo
q3p3reposo=q3p3*(180/pi)*2.857; //Cálculo para la obtención de los pasos del codo desde el punto 3 hasta la posición de reposo

q1p3pasrep = (int) q1p3reposo; //Convierte el valor de q1p3reposo a una variable entera
q2p3pasrep = (int) q2p3reposo; //Convierte el valor de q2p3reposo a una variable entera
q3p3pasrep = (int) q3p3reposo; //Convierte el valor de q3p3reposo a una variable entera

```

Código 10. Código comentado del programa de recorrido articular, sección de Cálculos. Fuente: Elaboración propia.

```

//***** MOVIMIENTO AL PUNTO 1 *****
//*****

if(q2p1>=-1.047 && q2p1<=0.174){ //Se cumple el if solo si el ángulo del hombro está entre esos dos valores, para evitar mover el motor fuera del rango articular
if(q3p1>=-1.466 && q3p1<=1.221){ //Se cumple el if solo si el ángulo del codo está entre esos dos valores, para evitar mover el motor fuera del rango articular
Serial.println(" "); //Imprime una línea en blanco
Serial.print("Los ángulos articulares del punto 1 son: ("); //Imprime "Los ángulos articulares son: ("
Serial.print(q1p1*(180/pi)); //Imprime el valor del ángulo articular de la base en grados
Serial.print(", "); //Imprime ", "
Serial.print(q2p1*(180/pi)); //Imprime el valor del ángulo articular del hombro en grados
Serial.print(", "); //Imprime ", "
Serial.print(q3p1*(180/pi)); //Imprime el valor del ángulo articular del codo en grados
Serial.println(" "); //Imprime ") "
Serial.println(" "); //Imprime una línea en blanco
Serial.print("Los pasos de los motores son: ("); //Imprime "Los pasos de los motores son: ("
Serial.print(q1pipasos); //Imprime los pasos que debe mover el motor de la base
Serial.print(", "); //Imprime ", "
Serial.print(q2pipasos); //Imprime los pasos que debe mover el motor del hombro
Serial.print(", "); //Imprime ", "
Serial.print(-q3pipasos); //Imprime los pasos que debe mover el motor del codo. Se ha cambiado el signode q3pipasos
//porque el sentido de giro del motor es el contrario
Serial.println(" "); //Imprime ") "
Serial.println(" "); //Imprime una línea en blanco
delay(2000); //Establece un tiempo de espera de 3s
}
if(q1pipasos>=120 || q1pipasos<=-120){ //Se cumple el if solo si el número de pasos de la base es mayor a 120 o menor a -120
if(q1pipasos>0){ //Se cumple el if solo si el número de pasos de la base es mayor a 0
StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
StepperBase.step(q1pipasos-120); //Mueve el motor de la base q1pipasos-120 pasos
StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
StepperBase.step(20); //Mueve el motor de la base 20 pasos
}
if(q1pipasos<0){ //Se cumple el if solo si el número de pasos de la base es menor a 0
StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
StepperBase.step(q1pipasos+120); //Mueve el motor de la base q1pipasos+120 pasos
StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
StepperBase.step(-20); //Mueve el motor de la base -20 pasos
}
}else{ //Se cumple si el número de pasos está entre 120 y -120
StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
StepperBase.step(q1pipasos); //Mueve el motor de la base q1pipasos pasos
}
if(q2pipasos>=120 || q2pipasos<=-120){ //Se cumple el if solo si el número de pasos del hombro es mayor a 120 o menor a -120
if(q2pipasos>0){ //Se cumple el if solo si el número de pasos del hombro es mayor a 0
StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
StepperShoulder.step(q2pipasos-120); //Mueve el motor del hombro q2pipasos-120 pasos
StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
}
}
}

```

Código 11. Código comentado del programa de recorrido articular, sección de Movimiento al punto 1.
Fuente: Elaboración propia.

```

}
if(q2p1pasos<0){ //Se cumple el if solo si el número de pasos del homrbo es menor a 0
  StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
  StepperShoulder.step(q2p1pasos+120); //Mueve el motor del hombro q2p1pasos+120 pasos
  StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
  StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
}
}else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(q2p1pasos); //Mueve el motor del hombro q1p1pasos pasos
}
if(q3p1pasos>=120 || q3p1pasos<=-120){ //Se cumple el if solo si el número de pasos del codo es mayor a 120 o menor a -120
  if(q3p1pasos>0){ //Se cumple el if solo si el número de pasos del codo es mayor a 0
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(-q3p1pasos+120); //Mueve el motor del hombro -q3p1pasos+120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
  }
  if(q3p1pasos<0){ //Se cumple el if solo si el número de pasos del codo es menor a 0
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(-q3p1pasos-120); //Mueve el motor del hombro -q3p1pasos-120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  }
}else{ //Se realiza el else si el ángulo del codo está fuera del rango estipulado
  Serial.println("PUNTO 1 FUERA DE RANGO"); //Imprime "PUNTO FUERA DE RANGO"
}
}else{ //Se realiza el else si el ángulo del hombro está fuera del rango estipulado
  Serial.println("PUNTO 1 FUERA DE RANGO"); //Imprime "PUNTO FUERA DE RANGO"
}
}
//***** MOVIMIENTO AL PUNTO 2 *****
//*****
if(q2p2>=-1.047 && q2p2<=0.174){ //Se cumple el if solo si el ángulo del hombro está entre esos dos valores, para evitar mover el motor fuera del rango articular
  if(q3p2>=-1.466 && q3p2<=1.221){ //Se cumple el if solo si el ángulo del codo está entre esos dos valores, para evitar mover el motor fuera del rango articular
    Serial.println(" "); //Imprime una línea en blanco
    Serial.print("Los ángulos articulares del punto 2 son: ("); //Imprime "Los ángulos articulares son: ("
    Serial.print(q1p2NR*(180/pi)); //Imprime el valor del ángulo articular de la base en grados
    Serial.print(", "); //Imprime ", "
    Serial.print(q2p2NR*(180/pi)); //Imprime el valor del ángulo articular del hombro en grados
    Serial.print(", "); //Imprime ", "
    Serial.print(q3p2NR*(180/pi)); //Imprime el valor del ángulo articular del codo en grados
    Serial.println(") "); //Imprime ") "
    Serial.println(" "); //Imprime una línea en blanco
    Serial.print("Los pasos de los motores son: ("); //Imprime "Los pasos de los motores son: ("
    Serial.print(q1p2pasos); //Imprime los pasos que debe mover el motor de la base
    Serial.print(", "); //Imprime ", "
    Serial.print(q2p2pasos); //Imprime los pasos que debe mover el motor del hombro
    Serial.print(", "); //Imprime ", "
    Serial.print(-q3p2pasos); //Imprime los pasos que debe mover el motor del codo. Se ha cambiado el signode q3p1pasos
    //porque el sentido de giro del motor es el contrario
    Serial.println(") "); //Imprime ") "
    Serial.println(" "); //Imprime una línea en blanco
    delay(2000); //Establece un tiempo de espera de 3s
  }
}

```

Código 12. Código comentado del programa de recorrido articular, sección de Movimiento al punto 1 y Movimiento al punto 2. Fuente: Elaboración propia.

```

if(q1p2pasos>=120 || q1p2pasos<=-120){ //Se cumple el if solo si el número de pasos de la base es mayor a 120 o menor a -120
  if (q1p2pasos>0) { //Se cumple el if solo si el número de pasos de la base es mayor a 0
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
    StepperBase.step(q1p2pasos-120); //Mueve el motor de la base q1p2pasos-120 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
  }
  if (q1p2pasos<0) { //Se cumple el if solo si el número de pasos de la base es menor a 0
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
    StepperBase.step(q1p2pasos+120); //Mueve el motor de la base q1p2pasos+120 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
  }
}
else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
  StepperBase.step(q1p2pasos); //Mueve el motor de la base q1p2pasos pasos
}
}
if(q2p2pasos>=120 || q2p2pasos<=-120){ //Se cumple el if solo si el número de pasos del hombro es mayor a 120 o menor a -120
  if (q2p2pasos>0) { //Se cumple el if solo si el número de pasos del hombro es mayor a 0
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
    StepperShoulder.step(q2p2pasos-120); //Mueve el motor del hombro q2p2pasos-120 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  }
  if (q2p2pasos<0) { //Se cumple el if solo si el número de pasos del hombro es menor a 0
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
    StepperShoulder.step(q2p2pasos+120); //Mueve el motor del hombro q2p2pasos+120 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  }
}
else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(q2p2pasos); //Mueve el motor del hombro q2p2pasos pasos
}
}

```

Código 13. Código comentado del programa de recorrido articular, sección de Movimiento al punto 2.
Fuente: Elaboración propia.

```

if(q3p2pasos>=120 || q3p2pasos<=-120){ //Se cumple el if solo si el número de pasos del codo es mayor a 120 o menor a -120
  if(q3p2pasos>0){ //Se cumple el if solo si el número de pasos del codo es mayor a 0
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(-q3p2pasos+120); //Mueve el motor del codo -q3p2pasos+120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
  }
  if(q3p2pasos<0){ //Se cumple el if solo si el número de pasos del codo es menor a 0
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(-q3p2pasos-120); //Mueve el motor del codo -q3p2pasos-120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  }
}
}else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
  StepperElbow.step(-q3p2pasos); //Mueve el motor del hombro -q3p2pasos pasos
}
}else{ //Se realiza el else si el ángulo del codo está fuera del rango estipulado
  Serial.println("PUNTO 2 FUERA DE RANGO"); //Imprime "PUNTO FUERA DE RANGO"
}
}else{ //Se realiza el else si el ángulo del hombro está fuera del rango estipulado
  Serial.println("PUNTO 2 FUERA DE RANGO"); //Imprime "PUNTO FUERA DE RANGO"
}
}

//***** MOVIMIENTO AL PUNTO 3 *****
//*****

if(q2p3>=-1.047 && q2p3<=0.174){ //Se cumple el if solo si el ángulo del hombro está entre esos dos valores, para evitar mover el motor fuera del rango articular
  if(q3p3>=-1.466 && q3p3<=1.221){ //Se cumple el if solo si el ángulo del codo está entre esos dos valores, para evitar mover el motor fuera del rango articular
    Serial.println(" "); //Imprime una línea en blanco
    Serial.print("Los ángulos articulares del punto 3 son: ("); //Imprime "Los ángulos articulares son: ("
    Serial.print(q1p3NR*(180/pi)); //Imprime el valor del ángulo articular de la base en grados
    Serial.print(", "); //Imprime ", "
    Serial.print(q2p3NR*(180/pi)); //Imprime el valor del ángulo articular del hombro en grados
    Serial.print(", "); //Imprime ", "
    Serial.print(q3p3NR*(180/pi)); //Imprime el valor del ángulo articular del codo en grados
    Serial.println(") "); //Imprime ") "
    Serial.println(" "); //Imprime una línea en blanco
    Serial.print("Los pasos de los motores son: ("); //Imprime "Los pasos de los motores son: ("
    Serial.print(q1p3pasos); //Imprime los pasos que debe mover el motor de la base
    Serial.print(", "); //Imprime ", "
    Serial.print(q2p3pasos); //Imprime los pasos que debe mover el motor del hombro
    Serial.print(", "); //Imprime ", "
    Serial.print(-q3p3pasos); //Imprime los pasos que debe mover el motor del codo. Se ha cambiado el signode q3p3pasos
    //porque el sentido de giro del motor es el contrario
    Serial.println(") "); //Imprime ") "
    Serial.println(" "); //Imprime una línea en blanco
    delay(2000); //Establece un tiempo de espera de 3s
  }
  if(q1p3pasos>=120 || q1p3pasos<=-120){ //Se cumple el if solo si el número de pasos de la base es mayor a 120 o menor a -120
    if(q1p3pasos>0){ //Se cumple el if solo si el número de pasos de la base es mayor a 0
      StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
      StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
      StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
      StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
      StepperBase.step(q1p3pasos-120); //Mueve el motor de la base q1p3pasos-120 pasos
      StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
      StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
      StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
      StepperBase.step(20); //Mueve el motor de la base 20 pasos
    }
  }
}
}

```

Código 14. Código comentado del programa de recorrido articular, sección de Movimiento al punto 2 y Movimiento al punto 3. Fuente: Elaboración propia.

```

if(q1p3pasos<0){
    StepperBase.setSpeed(50); //Se cumple el if solo si el número de pasos de la base es menor a 0
    StepperBase.step(-20); //Establece velocidad 50rpm al motor de la base
    StepperBase.setSpeed(60); //Mueve el motor de la base -20 pasos
    StepperBase.step(-20); //Establece velocidad 60rpm al motor de la base
    StepperBase.setSpeed(70); //Mueve el motor de la base -20 pasos
    StepperBase.step(-20); //Establece velocidad 70rpm al motor de la base
    StepperBase.setSpeed(90); //Mueve el motor de la base -20 pasos
    StepperBase.step(q1p3pasos+120); //Establece velocidad 90rpm al motor de la base
    StepperBase.setSpeed(70); //Mueve el motor de la base q1p3pasos+120 pasos
    StepperBase.step(-20); //Establece velocidad 70rpm al motor de la base
    StepperBase.setSpeed(60); //Mueve el motor de la base -20 pasos
    StepperBase.step(-20); //Establece velocidad 60rpm al motor de la base
    StepperBase.setSpeed(50); //Mueve el motor de la base -20 pasos
    StepperBase.step(-20); //Establece velocidad 50rpm al motor de la base
}
}else{ //Se cumple si el número de pasos está entre 120 y -120
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(q1p3pasos); //Mueve el motor de la base q1p3pasos pasos
}
if(q2p3pasos>=120 || q2p3pasos<=-120){ //Se cumple el if solo si el número de pasos del hombro es mayor a 120 o menor a -120
    if(q2p3pasos>0){ //Se cumple el if solo si el número de pasos del hombro es mayor a 0
        StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
        StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
        StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
        StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
        StepperShoulder.step(q2p3pasos-120); //Mueve el motor del hombro q2p3pasos-120 pasos
        StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
        StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
        StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
        StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
    }
    if(q2p3pasos<0){ //Se cumple el if solo si el número de pasos del hombro es menor a 0
        StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
        StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
        StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
        StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
        StepperShoulder.step(q2p3pasos+120); //Mueve el motor del hombro q2p3pasos+120 pasos
        StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
        StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
        StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
        StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    }
}else{ //Se cumple si el número de pasos está entre 120 y -120
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(q2p3pasos); //Mueve el motor del hombro q2p3pasos pasos
}
if(q3p3pasos>=120 || q3p3pasos<=-120){ //Se cumple el if solo si el número de pasos del codo es mayor a 120 o menor a -120
    if(q3p3pasos>0){ //Se cumple el if solo si el número de pasos del codo es mayor a 0
        StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
        StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
        StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
        StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
        StepperElbow.step(-q3p3pasos+120); //Mueve el motor del codo -q3p3pasos+120 pasos
        StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
        StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
        StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
        StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    }
}
}

```

Código 15. Código comentado del programa de recorrido articular, sección de Movimiento al punto 3.
Fuente: Elaboración propia.

```

if(q3p3pasos<0){ //Se cumple el if solo si el número de pasos del codo es menor a 0
  StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
  StepperElbow.step(-q3p3pasos-120); //Mueve el motor del codo -q3p3pasos-120 pasos
  StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
  StepperElbow.step(20); //Mueve el motor del codo 20 pasos
}
}else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
  StepperElbow.step(-q3p3pasos); //Mueve el motor del codo -q3p3pasos pasos
}

//***** MOVIMIENTO AL REPOSO *****
//*****

delay(2000); //Tiempo de espera de 2000ms
Serial.println(">>>> Volviendo al REPOSO <<<<<"); //Imprime ">>>> Volviendo al REPOSO <<<<<"
Serial.println(" "); //Imprime una línea en blanco

if(q1p3pasrep>=120 || q1p3pasrep<=-120){ //Se cumple el if solo si el número de pasos de la base es mayor a 120 o menor a -120
  if(q1p3pasrep>0){ //Se cumple el if solo si el número de pasos de la base es mayor a 0
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
    StepperBase.step(-q1p3pasrep+120); //Mueve el motor de la base -q1p3pasrep+120 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(-20); //Mueve el motor de la base -20 pasos
  }
  if(q1p3pasrep<0){ //Se cumple el if solo si el número de pasos de la base es menor a 0
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(90); //Establece velocidad 90rpm al motor de la base
    StepperBase.step(-q1p3pasrep-120); //Mueve el motor de la base -q1p3pasrep-120 pasos
    StepperBase.setSpeed(70); //Establece velocidad 70rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
    StepperBase.setSpeed(50); //Establece velocidad 50rpm al motor de la base
    StepperBase.step(20); //Mueve el motor de la base 20 pasos
  }
}else{ //Se cumple si el número de pasos está entre 120 y -120
  StepperBase.setSpeed(60); //Establece velocidad 60rpm al motor de la base
  StepperBase.step(-q1p3pasrep); //Mueve el motor de la base -q1p3pasrep pasos
}

if(q2p3pasrep>=120 || q2p3pasrep<=-120){ //Se cumple el if solo si el número de pasos del hombro es mayor a 120 o menor a -120
  if(q2p3pasrep>0){ //Se cumple el if solo si el número de pasos del hombro es mayor a 0
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
    StepperShoulder.step(-q2p3pasrep+120); //Mueve el motor del hombro -q2p3pasrep+120 pasos
    StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
    StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
    StepperShoulder.step(-20); //Mueve el motor del hombro -20 pasos
  }
}

```

Código 16. Código comentado del programa de recorrido articular, sección de Movimiento al punto 3 y Movimiento al reposo. Fuente: Elaboración propia.

```

if(q2p3pasrep<0){
  StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  StepperShoulder.setSpeed(90); //Establece velocidad 90rpm al motor del hombro
  StepperShoulder.step(-q2p3pasrep-120); //Mueve el motor del hombro -q2p3pasrep-120 pasos
  StepperShoulder.setSpeed(70); //Establece velocidad 70rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
  StepperShoulder.setSpeed(50); //Establece velocidad 50rpm al motor del hombro
  StepperShoulder.step(20); //Mueve el motor del hombro 20 pasos
}
}else{
  StepperShoulder.setSpeed(60); //Establece velocidad 60rpm al motor del hombro
  StepperShoulder.step(-q2p3pasrep); //Mueve el motor del hombro -q2p3pasrep pasos
}

if(q3p3pasrep>=120 || q3p3pasrep<=-120){
  if(q3p3pasrep>0){
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(q3p3pasrep-120); //Mueve el motor del codo q3p3pasrep-120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(20); //Mueve el motor del codo 20 pasos
  }
  if(q3p3pasrep<0){
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(90); //Establece velocidad 90rpm al motor del codo
    StepperElbow.step(q3p3pasrep+120); //Mueve el motor del codo q3p3pasrep+120 pasos
    StepperElbow.setSpeed(70); //Establece velocidad 70rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
    StepperElbow.setSpeed(50); //Establece velocidad 50rpm al motor del codo
    StepperElbow.step(-20); //Mueve el motor del codo -20 pasos
  }
}else{
  StepperElbow.setSpeed(60); //Establece velocidad 60rpm al motor del codo
  StepperElbow.step(q3p3pasrep); //Mueve el motor del codo q3p3pasrep pasos
}

delay(1000); //Tiempo de espera de 1000ms
}else{
  Serial.println("PUNTO 3 FUERA DE RANGO"); //Se realiza el else si el ángulo del codo está fuera del rango estipulado
  //Imprime "PUNTO FUERA DE RANGO"
}
}else{
  Serial.println("PUNTO 3 FUERA DE RANGO"); //Se realiza el else si el ángulo del hombro está fuera del rango estipulado
  //Imprime "PUNTO FUERA DE RANGO"
}
}else{
  Serial.println("OPCIÓN INVALIDA!!!! INTENTE NUEVAMENTE"); //Se realiza el else si el número introducido para realizar los cálculos no es un 1
  //Imprime "OPCIÓN INVALIDA!!!! INTENTE NUEVAMENTE"
}
}

```

Código 17. Código comentado del programa de recorrido articular, sección de Movimiento al reposo.
Fuente: Elaboración propia.

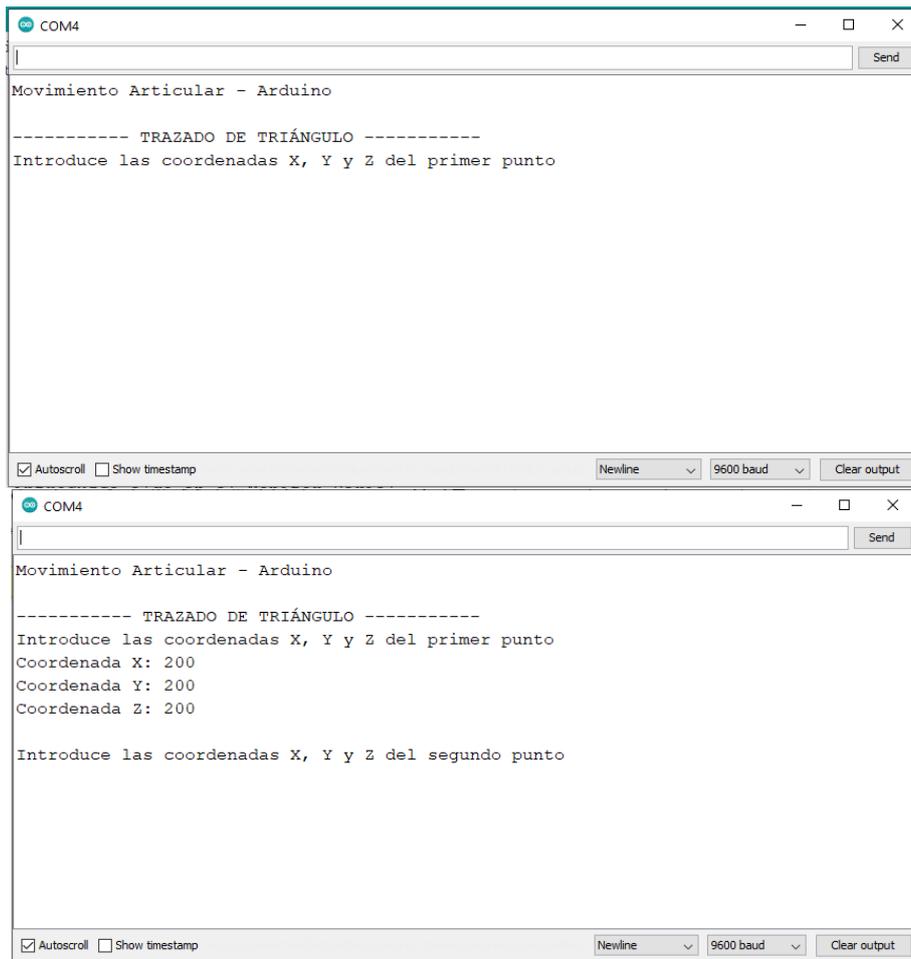


Figura 48. Inicio del programa de Trazado de triángulo en el Monitor Serial (arriba) y vista del programa tras la introducción de las coordenadas del punto 1 por parte del usuario (abajo). Fuente: Elaboración propia.

Como puede observarse, a medida que el usuario va introduciendo en el Monitor Serial los valores (en milímetros) de las coordenadas del punto 1, en este caso 200mm para los tres ejes, se ven mostrando los mensajes con la selección. Al terminar de introducir los últimos datos, el programa muestra la siguiente instrucción que debe seguir el usuario (introducir las coordenadas del punto 2 en este caso).

Para recibir y guardar los datos introducidos por el usuario, se empleó la función *Serial.read()*, asignando los distintos caracteres a la variable *char 'c'* y luego uniendo todos los caracteres en una variable tipo *String 'words'* mediante el comando += de Arduino IDE, que permite ir añadiendo datos a una variable sin eliminar los previos. Tras esto, se transforman los números (ahora en formato String) de la variable 'words' a números como variable int, permitiendo obtener una lista de variables con las coordenadas X, Y y Z de los 3 puntos a los que deberá moverse el robot. Se muestra un ejemplo de este sistema de guardado de datos en la **Código 19**.

```

if(control5==1) {
  while(Serial.available()) {
    delay(5);
    c = Serial.read();
    words += c;
  }
  p2y=words.toInt();

  delay(5);
  words = "";
}

```

Código 19. Sistema de lectura y guardado de datos introducidos por el usuario en el Monitor Serial.
Fuente: elaboración propia.

En tercer lugar, los cálculos, para los cuales se emplearon las funciones obtenidas mediante el estudio de la cinemática inversa del robot. Estos cálculos permiten al programa calcular los ángulos que deben girar las distintas articulaciones para posicionar el órgano terminal de robot desde la posición de reposo hasta los puntos introducidos por el usuario.

Además de esto, en los cálculos se realiza la conversión de radianes a pasos, ya que la función que permite mover los motores paso a paso requiere un input expresado en pasos.

Teniendo en cuenta de que el programa parte de la posición de reposo del robot, el primer punto no supone un problema ya que los ángulos obtenidos son los ángulos que deben moverse los motores. Sin embargo, para el segundo y tercer punto, si queremos que el robot vaya directamente a estos puntos sin pasar por la posición de reposo, habrá que realizar una suma vectorial entre el ángulo que deben girar las articulaciones para volver al reposo desde el punto anterior y los ángulos que deben girar las articulaciones para mover al robot desde el reposo hasta el nuevo punto. Esto significa, restar los ángulos obtenidos del nuevo punto menos los del anterior (teniendo en cuenta que los ángulos obtenidos cambiados de signo harán que el robot vaya del punto en cuestión a la posición de reposo).

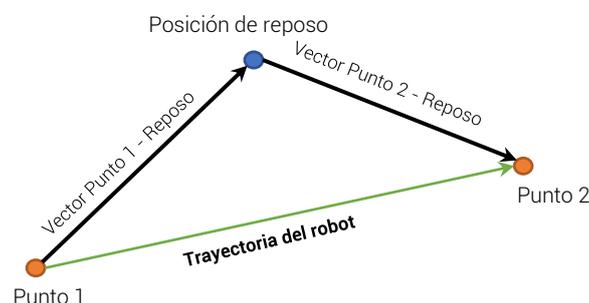


Figura 49. Definición gráfica del cálculo vectorial para la obtención de los movimientos articulares para realizar la trayectoria Punto 1 – Punto 2. Fuente: Elaboración propia.

La cuarta parte del programa consiste en las funciones de movimiento del brazo robot a los puntos deseados. Para esto, se han empleado los datos obtenidos mediante el programa de máximo rango articular, de forma que, si los cálculos dan lugar a un ángulo fuera del rango de alguna de las articulaciones, el programa pedirá al usuario que vuelva a introducir otro punto ya que el anterior estaba fuera de rango.

Para la realización de esta sección del programa, también se busca que los movimientos del robot sean lo más fluidos y suaves posibles, para evitar paradas bruscas al llegar a los puntos de destino y lograr un mejor desempeño. Para esto se ha de tener en cuenta los perfiles de velocidad del robot.

En primer lugar, están los perfiles de velocidad rectangulares, en los que el robot genera aceleraciones y deceleraciones bruscas que pueden llegar a comprometer la seguridad o provocar daños estructurales en el robot o en las piezas a transportar, y contribuir a un mayor desgaste mecánico.

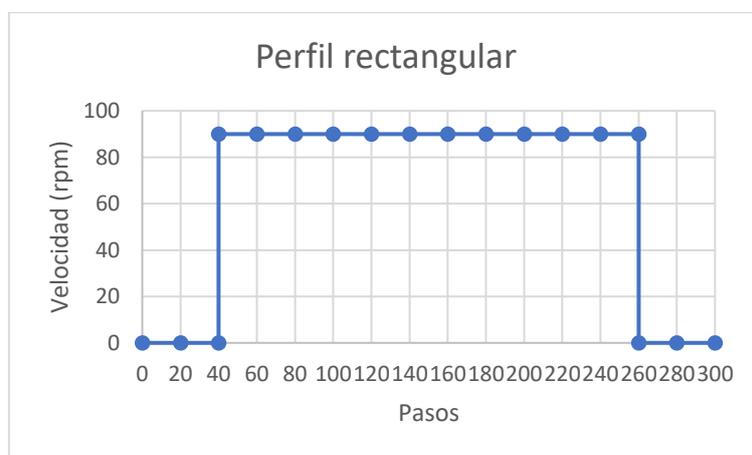


Figura 50. Ejemplo de perfil de velocidad rectangular. Fuente: elaboración propia.

En segundo lugar, están los perfiles de velocidad trapezoidales. Estos perfiles hacen que el robot vaya aumentando la velocidad durante un tiempo hasta alcanzar la velocidad máxima, al igual que van frenando a lo largo de un tiempo hasta detener el motor. Este tipo de perfiles de velocidad son más indicados, ya que hacen que el robot tenga movimientos más fluidos y de mejor precisión.

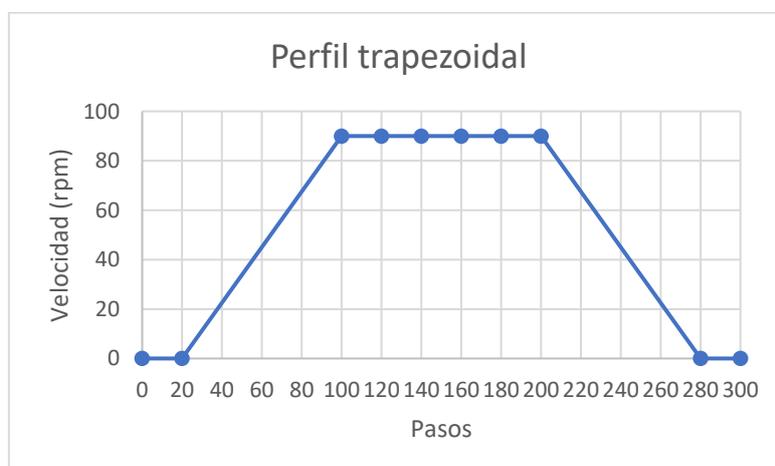


Figura 51. Ejemplo de perfil de velocidad trapezoidal. Fuente: elaboración propia.

En tercer lugar, están los perfiles de velocidad escalonados. Estos perfiles tratan de imitar los perfiles trapezoidales, buscan obtener movimientos más fluidos y precisos que los rectangulares y son más sencillos de generar y programar que un perfil trapezoidal.

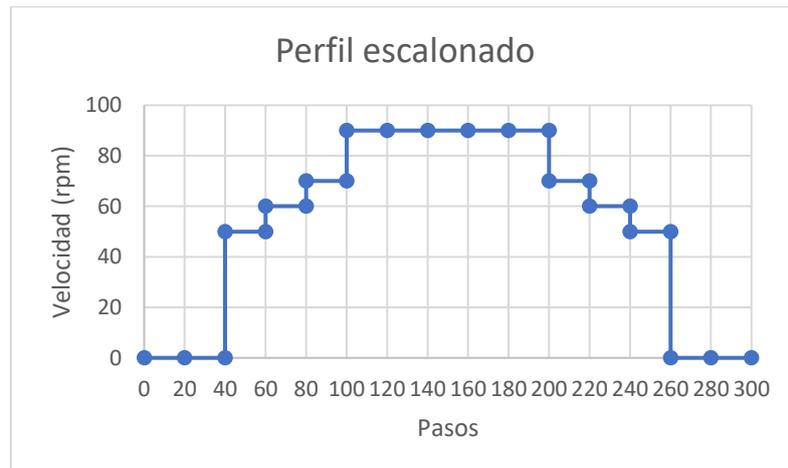


Figura 52. Ejemplo de perfil de velocidad escalonado (coincidente con el empleado en el programa de trazado de triángulo). Fuente: elaboración propia.

En este programa se emplearon los perfiles escalonados para movimientos de más de 120 pasos, en los que la duración del movimiento permitía aumentar la velocidad hasta los 90rpm, y los perfiles rectangulares, para aquellos movimientos menores a 120 pasos, para los cuales se empleaba una velocidad de 60rpm, más baja y que no provocaba arranques ni paradas demasiado bruscas, a pesar de tener el primer tipo de perfiles de velocidad.

Mediante los perfiles escalonados, las articulaciones se mueven 20 pasos a cada una de las 3 velocidades de transición, 50rpm (la menor velocidad admitida por el motor con la que se generaba suficiente par como para mover el brazo), 60rpm y 70rpm. Tras esos 60 pasos de transición, se establece la velocidad máxima, 90rpm, hasta que el robot se aproxima al punto deseado, realizando una reducción de velocidad inversa a la de aceleración. Puede observarse un ejemplo de este perfil de velocidades en la **Figura 52**.

Se decidieron emplear perfiles escalonados debido a que en un principio se realizó el programa mediante perfiles rectangulares para todos los movimientos, e incluir perfiles trapezoidales requería incluir una librería de control de motores paso a paso diferente a la empleada, lo cual habría causado una modificación completa del programa y la generación de líneas de código mucho más complejas. Además, tras analizar las diferencias entre un perfil escalonado y uno trapezoidal, se consideró que la diferencia no era lo suficientemente significativa como para invertir tanto tiempo del proyecto en modificar todo el código.

También se emplearon perfiles rectangulares para aquellos movimientos demasiado cortos como para poder aplicar el aumento escalonado de la velocidad. Con el objetivo de reducir la brusquedad de los arranques y frenados del robot durante estos movimientos cortos, se optó por reducir la velocidad máxima de movimiento de las articulaciones, de 90rpm a 60rpm.

El movimiento de los motores se realiza de forma secuencial, es decir, en primer lugar, se mueve el motor de la base, luego el hombro y por último el codo. Durante la realización del proyecto se trató de incluir una librería (DHT.h) con funciones que permitiesen realizar tareas simultáneas, sin embargo, la librería no se reconocía de forma correcta en el programa y las

funciones no trabajaban correctamente. Por esto, se terminó optando por mantener el movimiento secuencial.

Cabe destacar que el diseño del programa se realizó de forma que, tras haberse movido el robot por los puntos 1, 2 y 3, se envía el robot de nuevo a la posición de reposo para, al finalizar el programa, el usuario pueda introducir nuevos puntos. Esto se hace invirtiendo los ángulos obtenidos mediante la cinemática inversa para mover el órgano terminal desde el reposo al último punto.

Finalmente, en la quinta parte del programa, se realiza un reset de variables, las variables de control se les vuelve a asignar valor 1 y los valores de las coordenadas se les asigna valor 0. De esta forma el programa se reiniciará automáticamente sin tener que recargarlo en la placa Arduino.

9 RESULTADOS Y CONCLUSIONES

Se ha diseñado un brazo robot de 5 grados de libertad mediante la aplicación SolidWorks compuesto por 5 subensamblajes principales, la base, el hombro, el brazo, el antebrazo y la pinza.

Tras su ensamblaje en este programa, se ha realizado un estudio cinemático del modelo, logrando comprender el movimiento del robot en detalle. Mediante la cinemática directa se pudo estudiar las relaciones entre las distintas articulaciones y desarrollar la tabla de Denavit-Hartenberg que más adelante permitiría desarrollar la cinemática inversa. Esta permitió relacionar directamente los puntos del espacio de trabajo del robot con los movimientos articulares necesarios para alcanzar cualquiera de estos puntos.

Además, adquiriendo los componentes electrónicos necesarios de terceros e imprimiendo las piezas diseñadas en SolidWorks mediante una impresora 3D, se realizó el montaje del modelo, permitiendo realizar pruebas con él y verificar su correcto funcionamiento.

Por último, haciendo uso de la maqueta montada y del programa Arduino IDE, se pudo programar el brazo robot. En primer lugar, se programó una aplicación simple capaz de verificar los máximos rangos articulares de los diferentes motores del modelo uno por uno.

En segundo lugar, esta vez conectando todos los sistemas electrónicos del robot a la fuente de alimentación, se realizó un programa capaz de mostrar al usuario los resultados obtenidos con el programa anterior mediante un sencillo sistema de selección a través del Monitor Serial.

En tercer lugar, con el objetivo de aplicar las funciones desarrolladas mediante el estudio cinemático inverso, se programó un sistema de introducción de punto y generación de trayectorias para el brazo robot. Este programa se diseñó para que el usuario pudiese introducir tres puntos aleatorios del espacio de trabajo del robot y mediante los cálculos pertinentes, el robot realizase las trayectorias de posicionamiento del órgano terminal a estos tres puntos.

A continuación, se muestran 2 enlaces a los vídeos demostrativos de los dos últimos programas diseñados.

Movimiento Articular:

https://drive.google.com/file/d/1idYD1UQcrM9ioeQacFEsJdWFaLFZJN_Z/view?usp=sharing

Trazado Triángulo:

<https://drive.google.com/file/d/1USIH9ymgGMoK66mRSWzwrnCbT54Rzv6E/view?usp=sharing>

Con todo esto, podemos afirmar que se han cumplido los 4 objetivos impuestos al comienzo del proyecto y se han analizado todos los resultados, logrando una mayor comprensión en relación con los aspectos técnicos y teóricos detrás del desarrollo del mismo.

10 LÍNEAS FUTURAS DE INVESTIGACIÓN

- Diseñar un programa que implemente la orientación del órgano terminal para poder realizar tareas tipo pick&place.
- Modelar una caja o sistema de almacenamiento para los componentes electrónicos externos al robot que protejan las conexiones y aumente la seguridad de uso de la maqueta.
- Instalar sensores de proximidad o infrarrojos mediante los cuales obtener inputs que permitan al robot actuar en situaciones anómalas o facilitar su programación, gracias a la capacidad de recibir datos del entorno.
- Obtener fuentes de alimentación portátiles, independientes para cada uno de los motores, facilitando así el transporte del modelo y su uso en cualquier situación.

11 BIBLIOGRAFÍA

- [1] González Sorribes, Antonio (2021). *Tema 1: Introducción morfología del robot*. Departamento de Ingeniería de Sistemas y Automática, Universidad Politécnica de Valencia.
- [2] Anónimo. *Autonomía de los robots*. <http://community.fortunecity.ws/campus/essay/680/ANATOM%CDA.html>
- [3] Arduino. *What is Arduino?*. <https://www.arduino.cc/en/Guide/Introduction> . Última modificación: 5 de febrero 2018.
- [4] Automatización de procesos. *Motores pap explicados*. <http://www.productoscnc.es/motores-pap-explicados.html>
- [5] BIRT LH. *Correas trapeciales*. https://ikastaroak.ulhi.net/edu/es/DFM/DPM/DPM06/es_DFM_DPM06_Contenidos/CalculoCorreasTrapeciales.html
- [6] Electron Perdido. *Servos*. <https://electronperdido.com/categoria-producto/servos/>
- [7] Escuela de ingeniería y arquitectura, Universidad de Zaragoza. *RobotScene: simulación/explotación de escenarios robóticos con capacidad multirobot*. http://webdiis.unizar.es/~romeo/Manual_RobotScene
- [8] Grabcad Community. *Bearing 624RS*. <https://grabcad.com/library/bearing-624rs-1> . Última modificación: 11 de febrero 2016.
- [9] Grabcad Community. *Servomotor MG996R*. <https://grabcad.com/library/servomotor-mg996r-3> . Última modificación: 15 de mayo 2020

- [10] Grabcad Community. Turnigy TG9e 9g Servomotor. <https://grabcad.com/library/turnigy-tg9e-9g-servomotor-1> . Última modificación: 30 de abril de 2020.
- [11] Impresoras3D.com. Guía definitiva sobre tipos de filamentos 3D. <https://www.impresoras3d.com/la-guia-definitiva-sobre-los-distintos-filamentos-para-impresoras-3d/> . Última modificación: 4 de enero de 2021
- [12] Ingemechanica. Correas de transmisión Cálculo y diseño. <https://ingemechanica.com/tutorialsemanal/tutorialn121.html> Última modificación: 3 de agosto 2021.
- [13] Ingeniería de sistemas y automática. Tema 2 Morfología. <http://www.isa.uniovi.es/~alonsog/Robotica/02%20Morfologia.pdf>
- [14] Luxorparts. 360° Continuous Rotation Servo. https://www.kjell.com/globalassets/mediaassets/701903_90770_datasheet_en.pdf?ref=762940D624
- [15] Mootio components. Que es par motor y pautas para calcularlo correctamente. <http://www.mootio-components.com/blog/es/que-es-par-motor/> . Última modificación: 20 de octubre 2017.
- [16] ResearchGate. Representación de las dimensiones de un robot paralelo tipo delta. https://www.researchgate.net/figure/Figura-1-Representacion-de-las-dimensiones-de-un-robot-paralelo-tipo-delta_fig1_271908711
- [17] ResearchGate. Robot manipulador SCARA. https://www.researchgate.net/figure/Figura-1-Robot-manipulador-SCARA_fig1_335149942
- [18] StepperOnline. Nema 17 Bipolar 45Ncm (64oz.in) 2A 42x42x40mm 4 Wires w/ 1m Cable & Connector. <https://www.omc-stepperonline.com/nema-17-bipolar-45ncm-64oz-in-2a-42x42x40mm-4-wires-w-1m-cable-and-connector.html?search=17HS16-2004s1>
- [19] StepperOnline. Nema 17 Stepper Motor Bipolar L=38mm w/ Gear Ratio 5:1 Planetary Gearbox. <https://www.omc-stepperonline.com/nema-17-stepper-motor-bipolar-l38mm-w-gear-raio-51-planetary-gearbox-17hs15-1684s-pg5.html?search=nema%2017%20%25Gear%20Ratio%205%3A1>
- [20] Talleres auxiliares del motor, S.A. Selección del paso de la correa dentada. <http://www.mecapedia.uji.es/catalogos/correa/tam.2.pdf>
- [21] TowerPro Online Shop. MG90D. <https://www.towerpro.com.tw/product/mg90d-2/>
- [22] UdeSantiagoVirtual. Estructura de los robots. <http://www.udesantiagovirtual.cl/moodle2/mod/book/view.php?id=24911>
- [23] UdeSantiagoVirtual. Estructura de los robots. <http://www.udesantiagovirtual.cl/moodle2/mod/book/view.php?id=24911&chapterid=227>
- [24] Ultimaker. Ficha de datos técnicos PLA. <https://docs.ultimaker.com/7ade/0900766b81698006.pdf> . Última modificación: 16 de mayo 2017.

- [25] Universitat Jaume I. *Propiedades del PLA*.
<https://sites.google.com/view/poliacidolactico-coma/poli%C3%A1cido-l%C3%A1ctico/propiedades-del-pla>
- [26] Wikipedia. *Brazo robótico*. https://es.wikipedia.org/wiki/Brazo_rob%C3%B3tico . Última modificación: 1 de marzo 2021.
- [27] Wikipedia. *Robot*. <https://es.wikipedia.org/wiki/Robot> . Última modificación: 26 de junio 2021.

ANEXO I - Hojas técnicas de los motores



MG90D
 New Digital 13g Metal gear, double ball bearing servo.
 Excellent for RC F3A 3D flyer helicopter & airplane, alloy casing in the middle
 This is upgrade version of TowerPro MG90s.

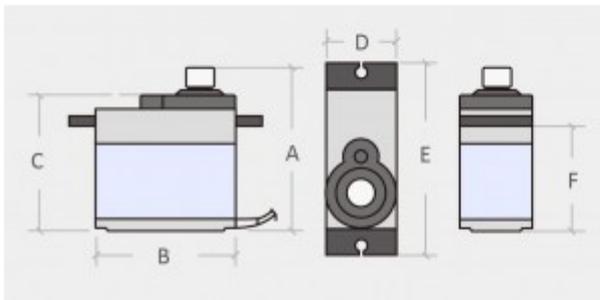
Specification:

Weight: 13g
 Dimension: 22.8×12.2×28.5mm
 Stall torque: 2.1kg/cm (4.8v) ; 2.4kg/cm (6.6V)
 Operating speed: 0.10sec/60degree (4.8V); 0.08sec/60degree (6.0V)
 Operating voltage: 4.8V- 6.6V
 Gear Type: Metal gear
 Servo case: Middle with alloy case
 Bearing: Double ball bearing
 Temperature range: 0- 55°C
 Dead band width: 1us
 servo wire length: 25cm
 servo arms & screws included
 Servo Plug: JR (Fits JR and Futaba)
 It's universal "S" type connector that fits most receivers, including Futaba, JR, Hitec ,GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum.
 CE &RoHS approved

We have upgraded our servo gear set and shaft to aluminum 6061-T6.
 It is stronger and lighter than copper.

Categories: Mini Servo 11-20g, Servos & Parts

Description



PRODUCT CONFIGURE TABLE

Weight(g)	13
Torque(kg)(4.8v)	2.1
Speed(sec/60deg)	0.10
A(mm)	32.5
B(mm)	22.6
C(mm)	28.5
D(mm)	12
E(mm)	31.5
F(mm)	19.8

360° Continuous Rotation Servo

Gear motor for Arduino and Raspberry-Pi robotics projects.

Typical use: Model aircraft, cars and robots.

A continuous servo rotates forward or backwards instead of moving to a given position.

Bidirectional rotation - pulse duration determines the speed and direction of rotation.



Item no: 90770

Model no: DS04-NFC

Weight: 38g.

Torque: 5,5 kg/cm (54 Ncm) (at 4,8 V).

Speed: 0,22 sec/60° ≈ 45 rpm (at 4,8 V).

Linear response to PWM (0-45 RPM) for easy ramping

Operating voltage: 4,8-6 V.

Operating temperature: -10 to 50 °C.

Current: < 1000 mA.

Cable length: 290 mm.

Connector type: JR / Futaba

Breadboard friendly connector 2,54 mm pitch.

Connector wire gauge: 28 AWG.

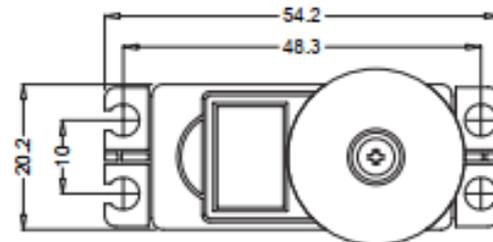
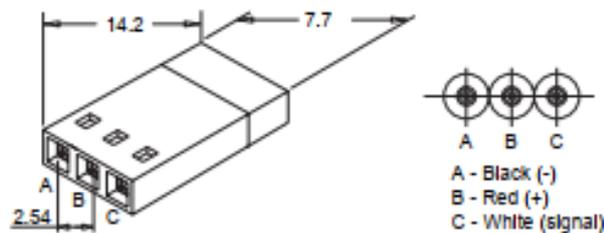
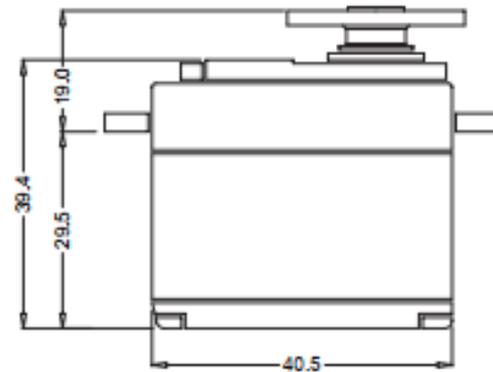
Control system: PWM (Pulse Width Modulation)

Counterclockwise rotation: 1-1,5 ms

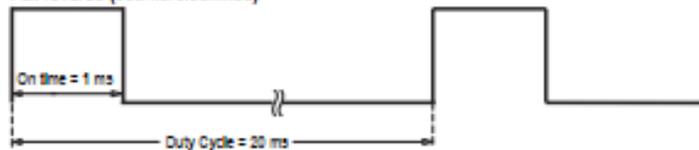
No rotation (stop): 1,5 ms

Clockwise rotation: 1,5-2 ms

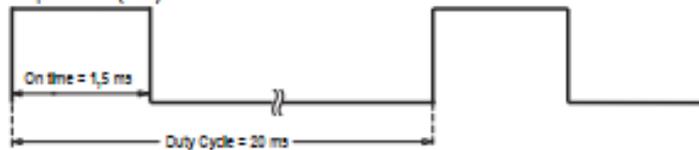
Pulse Frequency / Duty cycle: 50 Hz / 20 ms square wave

**360° servo PWM signal timings**

Full reverse (counterclockwise)

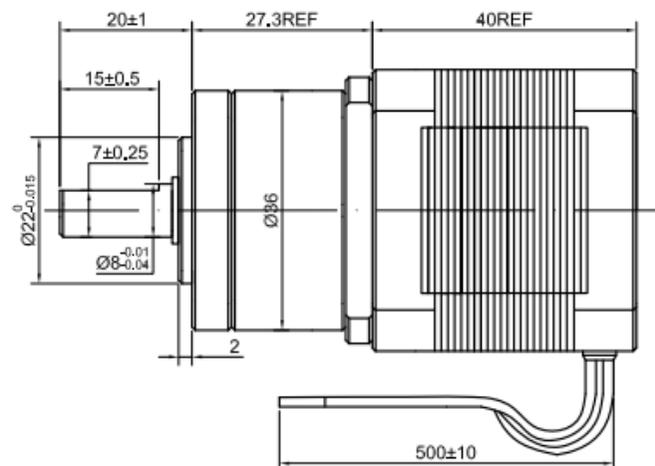
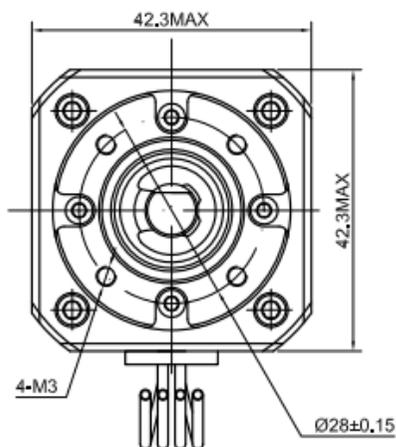


Stop rotation (stall)



Full forward (clockwise)



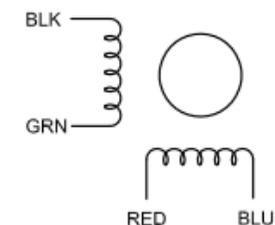


SPECIFICATION	CONNECTION	BIPOLAR
AMPS/PHASE		1.68
RESISTANCE/PHASE(Ohms)@25°C		1.65±10%
INDUCTANCE/PHASE(mH)@1KHz		3.20±20%
HOLDING TORQUE w/o GEARBOX(Nm)[lb-in]		0.36[3.19]
GEAR RATIO		5 $\frac{2}{11}$
EFFICIENCY		90.00%
STEP ANGLE w/o GEARBOX(°)		1.80
BACKLASH@NO-LOAD		<=1°
MAX.PERMISSIBLE TORQUE(Nm)		2.00
MOMENT PERMISSIBLE TORQUE(Nm)		4.00
SHAFT MAXIMUM AXIAL LOAD(N)		50.00
SHAFT MAXIMUM RADIAL LOAD(N)		100.00
WEIGHT(Kg)[lb]		0.55[1.21]
TEMPERATURE RISE:MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED)		
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]		
INSULATION CLASS B 130°C[266°F]		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A —	BLK	A
2	A\ —	GRN	A\
3	B —	RED	B
4	B\ —	BLU	B\

FULL STEP 2 PHASE-Ex. ,
WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\		
1	+	+	-	-	↓	CCW
2	-	+	+	-		↑
3	-	-	+	+		
4	+	-	-	+		

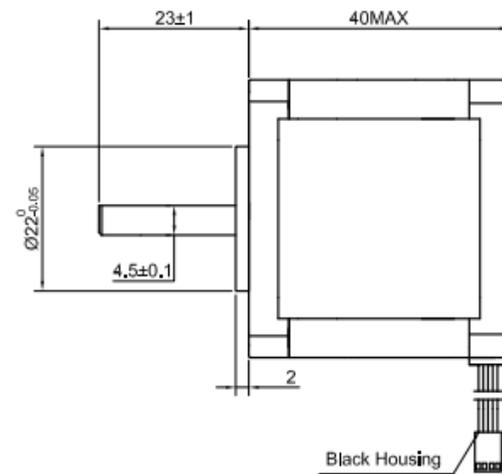
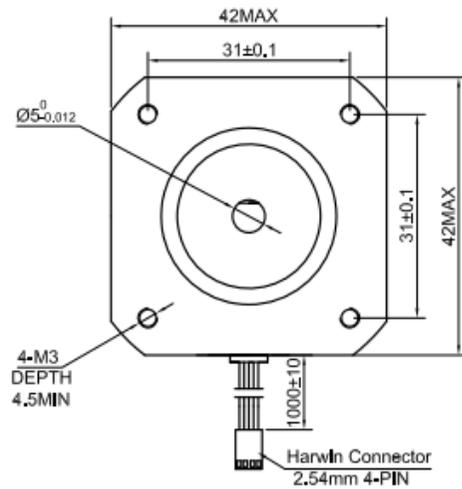


STEPPERONLINE®

APVD		8.18.2018
CHKD		
DRN		
SCALE	SIGNATURE	DATE

STEPPER MOTOR

17HS15-1684S-PG5

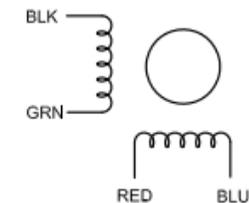


SPECIFICATION	CONNECTION	BIPOLAR
AMPS/PHASE		2.00
RESISTANCE/PHASE(Ohms)@25°C		1.10±10%
INDUCTANCE/PHASE(mH)@1KHz		2.60±20%
HOLDING TORQUE(Nm)[lb-in]		0.45[3.98]
STEP ANGLE(°)		1.80
STEP ACCURACY(NON-ACCUM)		±5.00%
ROTOR INERTIA(g-cm²)		54.00
WEIGHT(Kg)[lb]		0.30[0.66]
TEMPERATURE RISE:MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED)		
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]		
INSULATION RESISTANCE 100 Mohm (UNDER NORMAL TEMPERATURE AND HUMIDITY)		
INSULATION CLASS B 130°C[266°F]		
DIELECTRIC STRENGTH 500VAC FOR 1MIN.(BETWEEN THE MOTOR COILS AND THE MOTOR CASE)		
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A —	BLK	A A'
2	A\ —	GRN	
3	B —	RED	B B'
4	B\ —	BLU	

FULL STEP 2 PHASE-Ex. .
WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\	↓ CW	↑ CCW
1	+	+	-	-		
2	-	+	+	-		
3	-	-	+	+		
4	+	-	-	+		



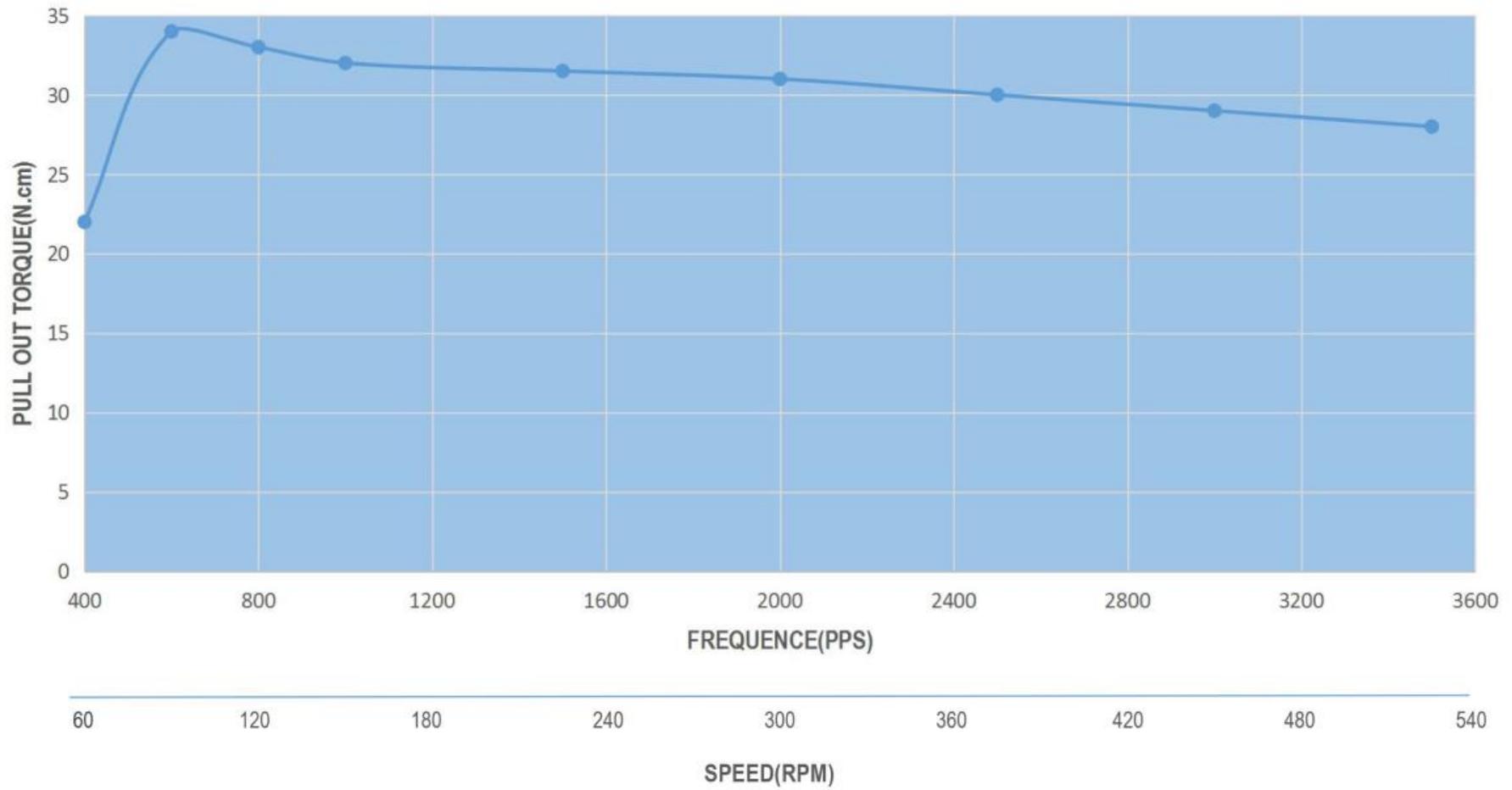
STEPPERONLINE®

APVD		8.18.2018
CHKD		
DRN		
SCALE	SIGNATURE	DATE

STEPPER MOTOR

17HS16-2004S1

17HS16-2004S1 PULL OUT TORQUE(2A, 24V HALF STEP)



ANEXO II – Obtención y cálculo de presupuestos

MANO DE OBRA

	TÉCNICO DE TALLER	TÉCNICO DE IMPRESIÓN 3D	INGENIERO
Salario	1,200.00€	2,200.00€	2,600.00€
Horas trabajadas mensuales	180	180	180
IRPF	4.0%	12.9%	12.9%
Contingencias comunes	23.6%	23.6%	23.6%
Contingencias profesionales	6.7%	3.1%	3.1%
Accidentes e incapacidades	2.0%	2.0%	2.0%
Desempleo (general)	5.5%	5.5%	5.5%
Proceso aprendizaje	0.6%	0.6%	0.6%
FOGASA	0.2%	0.2%	0.2%
Horas extra	12.0%	12.0%	12.0%
Coste extra para la empresa	511.20€	1,053.60€	1,245.10€
Total coste salarial	1,711.20€	3,253.60€	3845.10€
Total coste por hora (€/h)	9.51 €	18.08 €	21.36 €
Total coste por minuto (€/min)	0,158 €	0,301 €	0,356 €

PROCESOS

- Corte Láser

GASTOS MAQUINARIA	
COSTES COMPRA	
Cortadora láser	60,255.69€
Transporte	650.00€
Seguro riesgos laborales	975.00€
Cimentación	87.84€
Instalación eléctrica	72.00€
SUBTOTAL	61,952.59€
HERRAMIENTAS Y MANTENIMIENTO	
Mantenimiento general (anual)	550.00€

Este proceso requiere un ingeniero para la creación del archivo CAM de la pieza. Su coste dependerá del tiempo empleado en la preparación del archivo en minutos (T_{prog}), el cual variará según la complejidad de la pieza (número de caras y de operaciones). Su coste será:

$$C_{ingeniero} = MO1 \cdot t_{prog}$$

Un técnico preparará la máquina para el corte por láser y será responsable de su control durante el proceso de corte. Su coste dependerá del tiempo de preparación de la máquina (t_{prep}) y del tiempo de corte (ambos tiempos en minutos) (t_{mec}):

$$C_{técnico} = MO3 \cdot (T_{prep} + t_{mec})$$

El gasto en material depende del material utilizado (MAT 8 o MAT9) y el volumen del bruto (V_b).

$$C_{material} = V_b \cdot MAT4 \text{ (o MAT12)}$$

Tomaremos una amortización constante lo más larga posible por ley, 8 años; y asumiendo un uso mensual de 180 horas, obtenemos una amortización de 7.872€/h (MAQ1). Conociendo la potencia de la máquina (P) y el coste de la energía eléctrica (C_{el}), calcularemos los costes energéticos. Además, se estimarán los gastos de mantenimiento como 3.24€/h (C_{Man}). Los costes de la máquina serán:

$$C_{máquina} = t_{mec} \cdot (MAQ3 + C_{Man} + P \cdot C_{el})$$

Finalmente, el coste total del proceso de cortado por láser será:

$$C_{corte_láser} = C_{ingeniero} + C_{técnico} + C_{material} + C_{máquina}$$

- Impresión 3D

GASTOS MAQUINARIA	
COSTES COMPRA	
Impresora 3D Zortrax M200 Plus	2,528.90€
Transporte	5.30€
Instalación	25.00€
SUBTOTAL	2,559.20€
HERRAMIENTAS Y MANTENIMIENTO	
Mantenimiento general (anual)	120.00€

Este proceso requiere un técnico de impresión 3D para poner a punto la impresora y preparar los archivos CAM de las piezas para su impresión. Su coste dependerá del tiempo empleado en la preparación del archivo (T_{prep}), el cual variará en función de la complejidad de la pieza (número de orificios, caras y tamaño), y del tiempo de puesta a punto (t_{pap}). Su coste será:

$$C_{técnico} = MO2 \cdot (T_{prep} + t_{pap})$$

El gasto en material depende de la masa de material empleado, en este caso MAT36.

$$C_{material} = m_b \cdot MAT36$$

Se considerará una amortización constante para un total de 20,000 horas de impresión (vida promedio de una impresora 3D), obteniendo una amortización de 0.128€/h (MAQ 2). Conociendo la potencia de la máquina (P) y el coste de la energía eléctrica (C_{ele}), se calcularán los costes energéticos. Además, se estiman unos gastos de mantenimiento de 0,02€/h. Los costes serán:

$$C_{máquina} = t_{mec} \cdot (MAQ3 + C_{Man} + P \cdot C_{el})$$



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

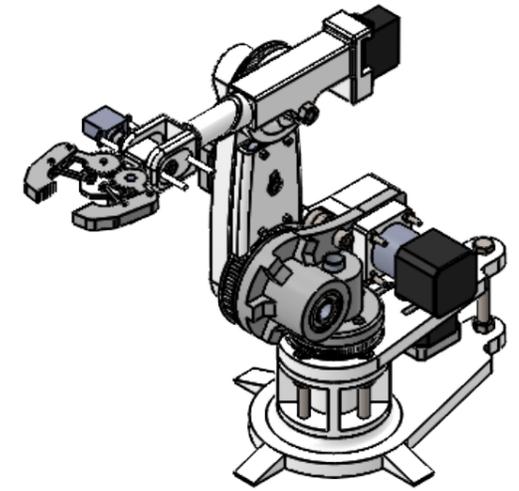
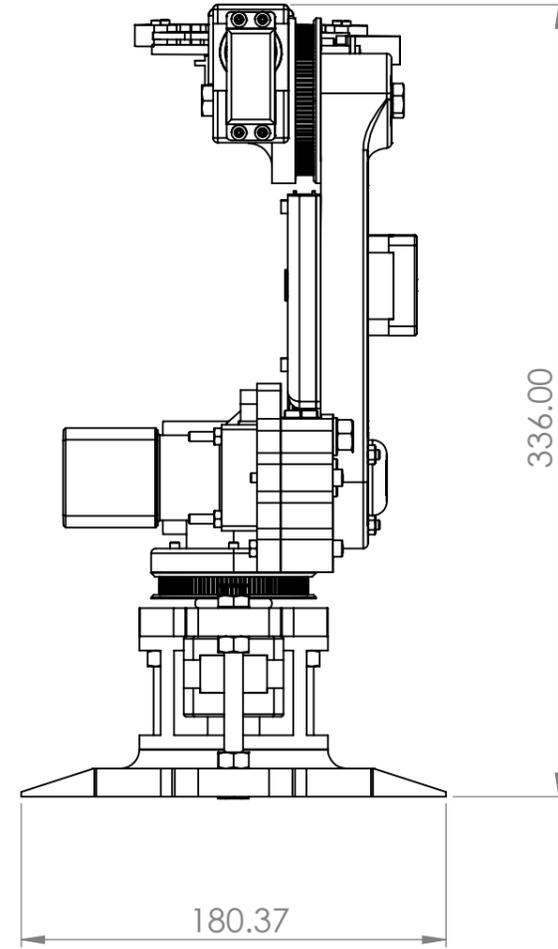
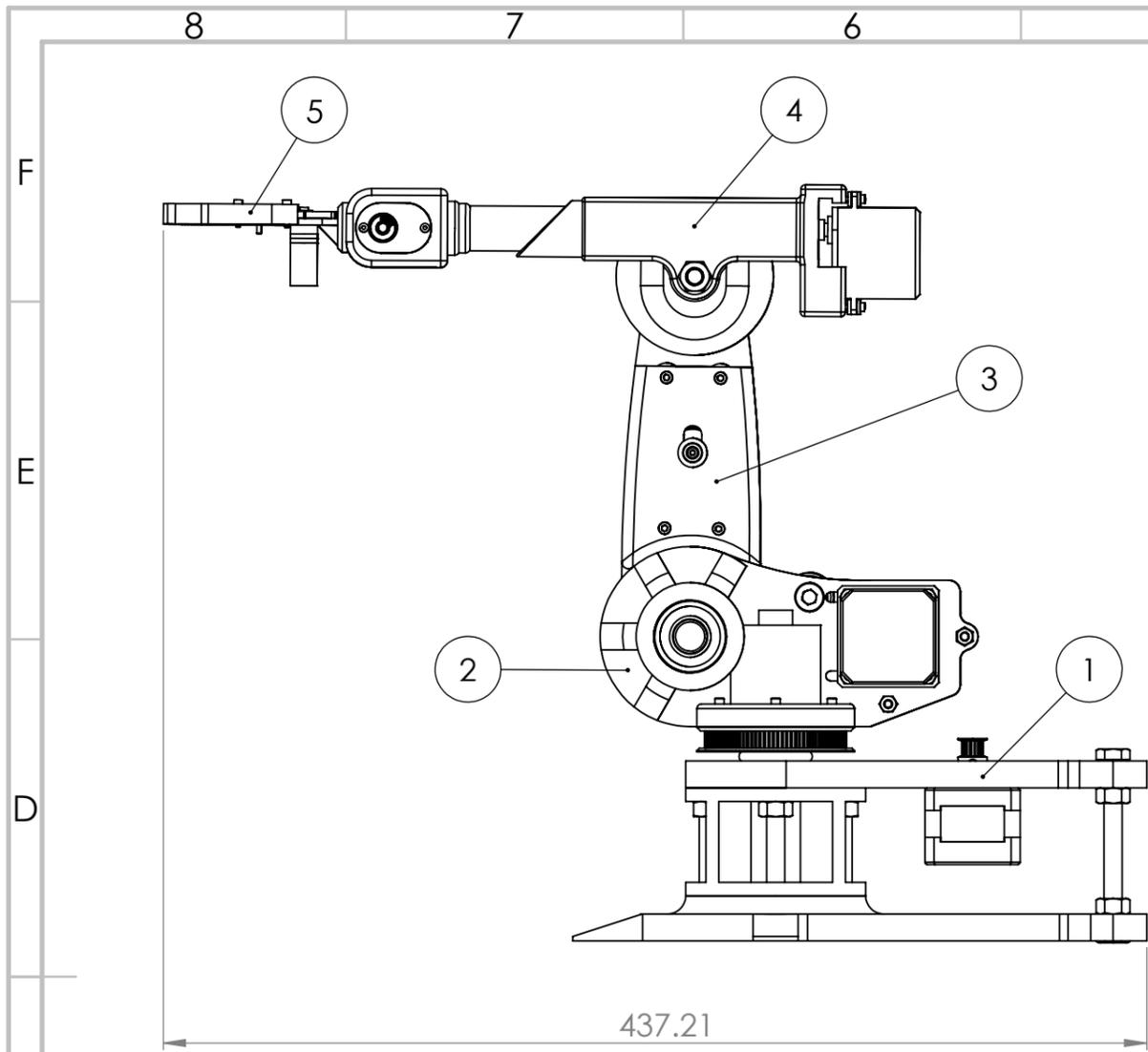


Escuela Técnica Superior de Ingeniería del Diseño
Universitat Politècnica de València

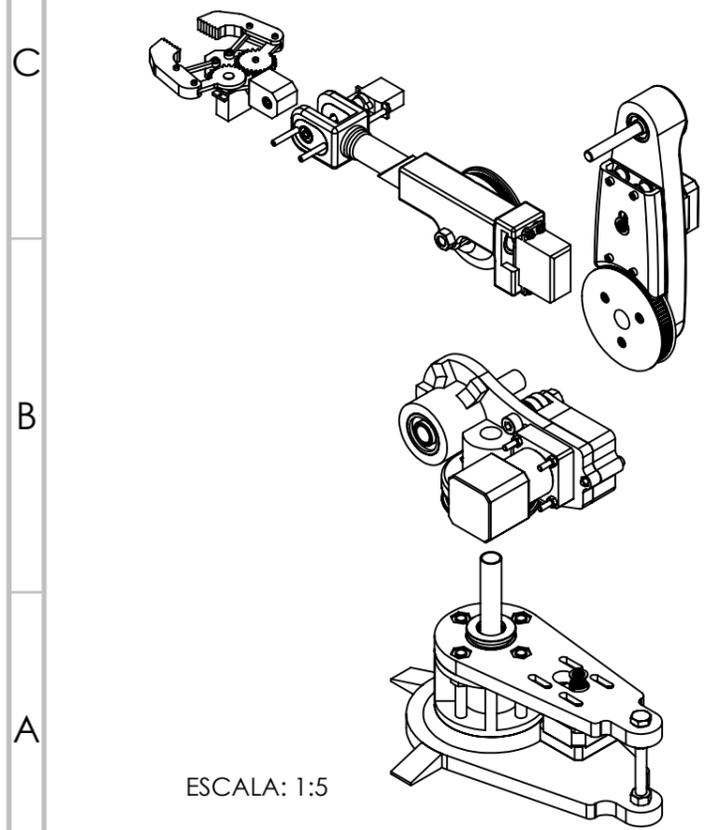
DOCUMENTO PLANOS

Índice documento planos

1.	ENSAMBLAJE ROBOT	101
1.1.	ENSAMBLAJE BASE	102
1.1.1.	BASE INFERIOR	103
1.1.2.	BASE SUPERIOR	104
1.1.3.	TROCO DE LA BASE	105
1.2.	ENSAMBLAJE HOMBRO.....	106
1.2.1.	POLEA HOMBRO.....	107
1.2.2.	SEPARADOR POLEA HOBRO.....	108
1.2.3.	BASE HOMBRO	109
1.2.4.	COBERTURA HOMBRO.....	110
1.2.5.	SOPORTE MOTOR.....	111
1.2.6.	ADAPTADOR POLEA	112
1.3.	ENSAMBLAJE BRAZO	113
1.3.1.	POLEA BRAZO	114
1.3.2.	SEPARADOR POLEA BRAZO.....	115
1.3.3.	BASE BRAZO.....	116
1.3.4.	SEPARADOR	117
1.3.5.	COBERTURA BRAZO	118
1.4.	ENSAMBLAJE ANTEBRAZO.....	119
1.4.1.	ANTEBRAZO PRINCIPAL.....	120
1.4.2.	ACOPLE EJE PRINCIPAL	122
1.4.3.	ADAPTADOR MOTOR.....	123
1.4.4.	CONEXIÓN MUÑECA.....	124
1.4.5.	POLEA ANTEBRAZO	125
1.4.6.	ACOPLE EJE SECUNDARIO	126
1.5.	ENSAMBLAJE PINZA.....	127
1.5.1.	SOPORTE PINZA.....	128
1.5.2.	PINZA	129
1.5.3.	ENGRANAJE 1	130
1.5.4.	ENGRANAJE 2	131
1.5.5.	FIJADOR PINZA.....	132



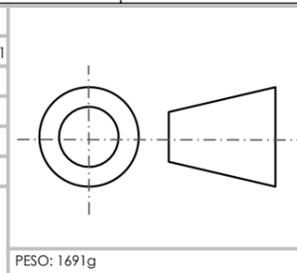
ESCALA: 1:5



ESCALA: 1:5

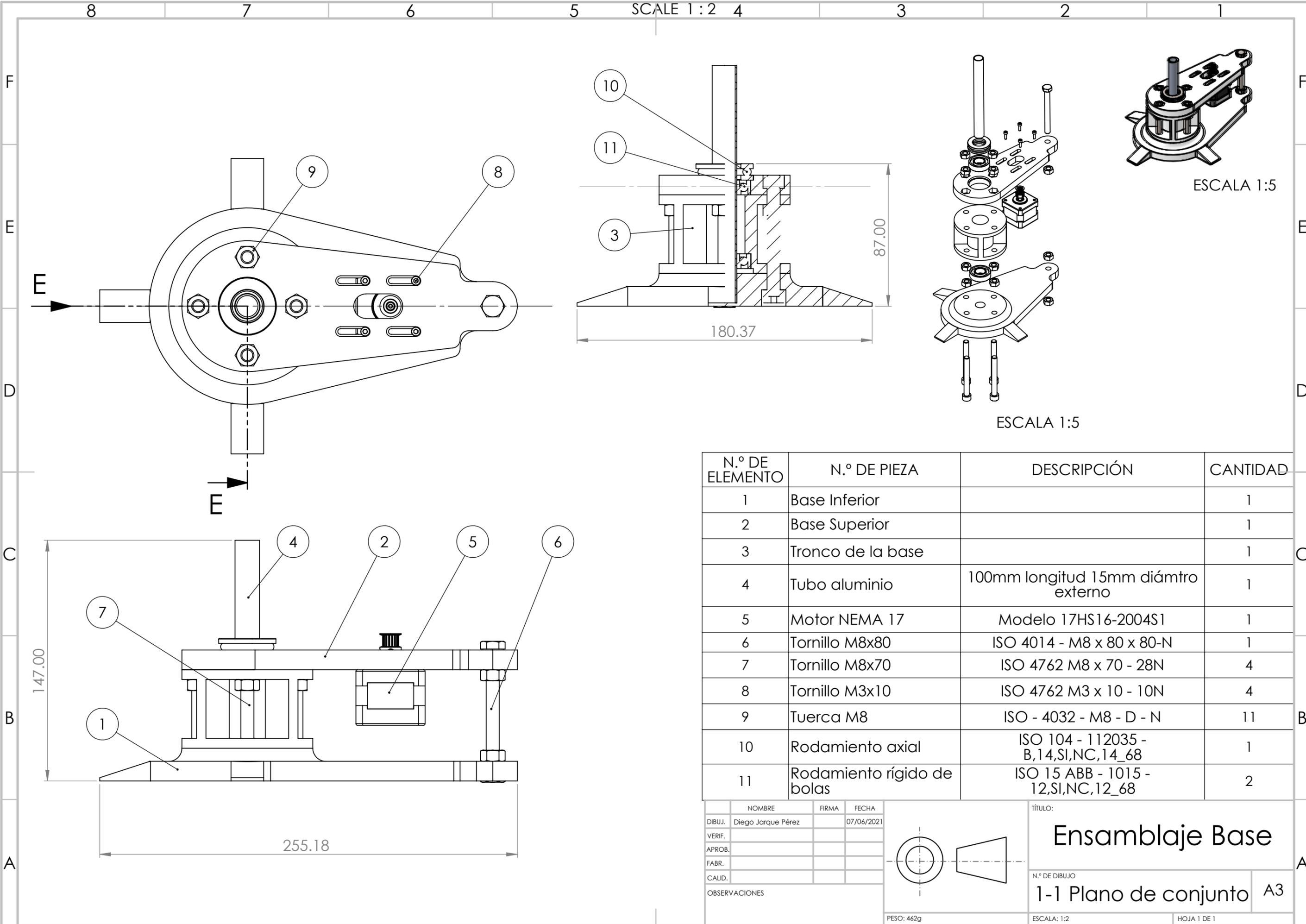
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Ensamblaje Base	Piezas, tornillería y motores del conjunto	1
2	Ensamblaje Hombro	Piezas, tornillería y motores del conjunto	1
3	Ensamblaje Brazo	Piezas, tornillería y motores del conjunto	1
4	Ensamblaje Antebrazo	Piezas, tornillería y motores del conjunto	1
5	Ensamblaje Pinza	Piezas, tornillería y motores del conjunto	1

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		07/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES		



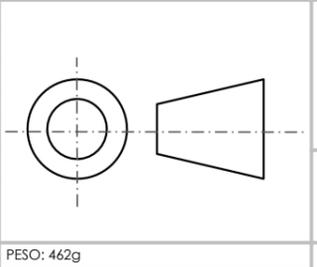
PESO: 1691g

TÍTULO:	
Ensamblaje robot	
N.º DE DIBUJO	A3
1. Plano de conjunto	
ESCALA: 1:3	HOJA 1 DE 1



N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Base Inferior		1
2	Base Superior		1
3	Tronco de la base		1
4	Tubo aluminio	100mm longitud 15mm diámetro externo	1
5	Motor NEMA 17	Modelo 17HS16-2004S1	1
6	Tornillo M8x80	ISO 4014 - M8 x 80 x 80-N	1
7	Tornillo M8x70	ISO 4762 M8 x 70 - 28N	4
8	Tornillo M3x10	ISO 4762 M3 x 10 - 10N	4
9	Tuerca M8	ISO - 4032 - M8 - D - N	11
10	Rodamiento axial	ISO 104 - 112035 - B,14,SI,NC,14_68	1
11	Rodamiento rígido de bolas	ISO 15 ABB - 1015 - 12,SI,NC,12_68	2

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		07/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES		



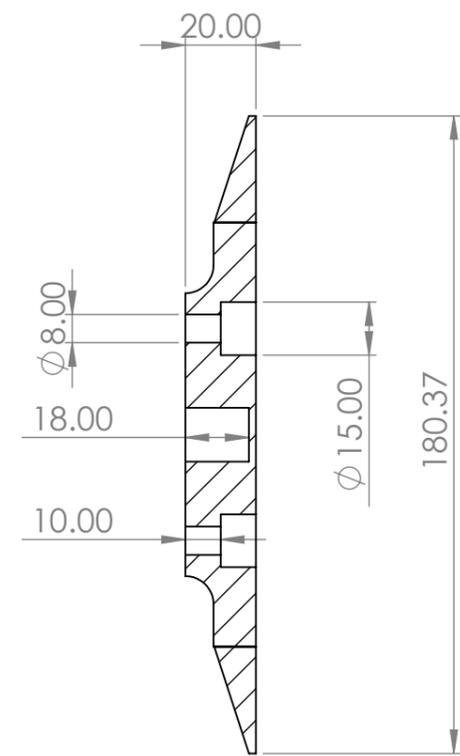
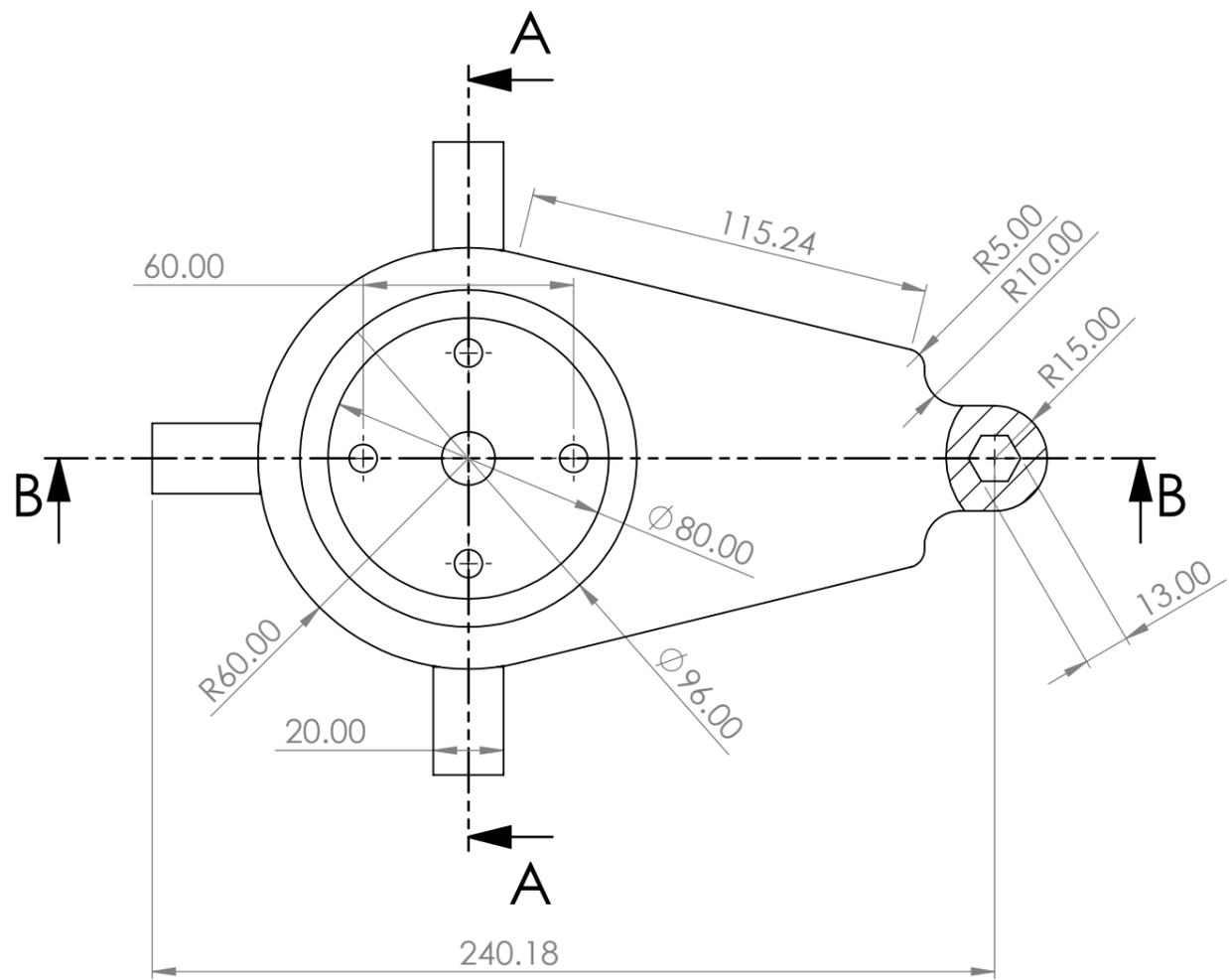
TÍTULO:
Ensamblaje Base

N.º DE DIBUJO:
1-1 Plano de conjunto

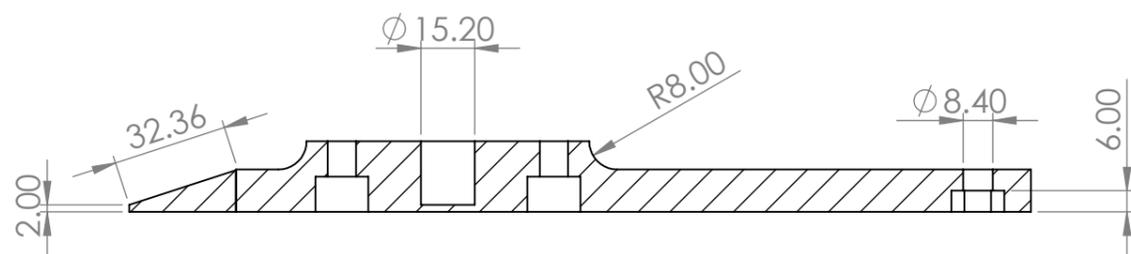
ESCALA: 1:2

HOJA 1 DE 1

PESO: 462g

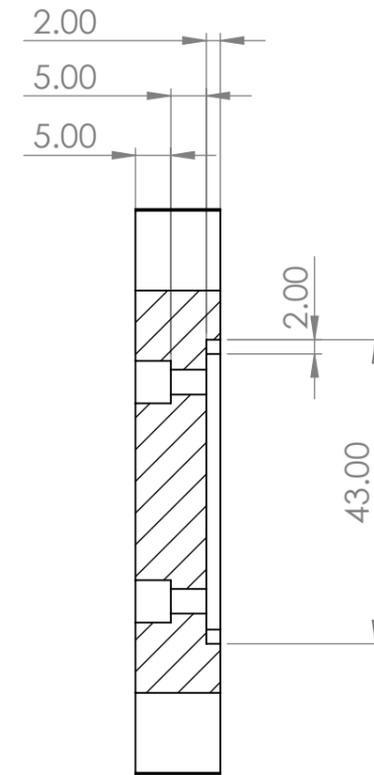
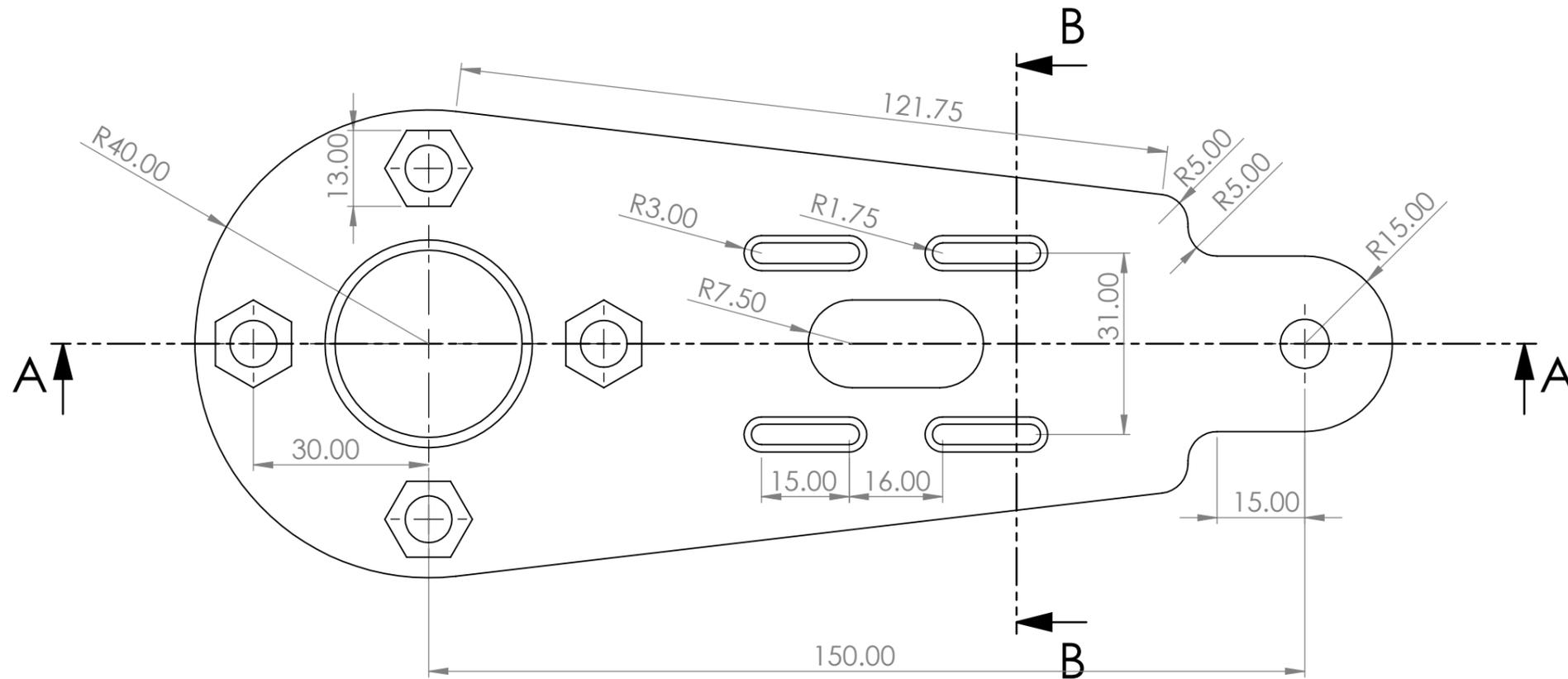


SECCIÓN A-A

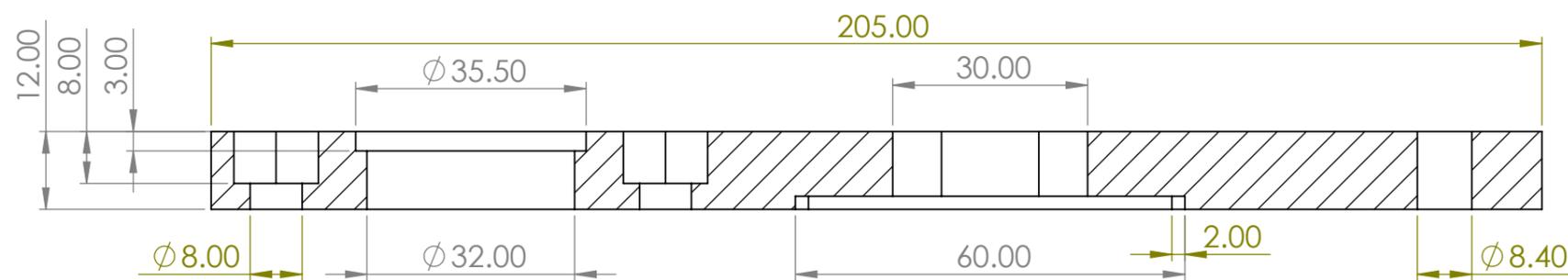


SECCIÓN B-B

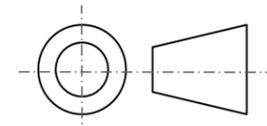
SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: $\pm 0,1$ mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:		Base Inferior	
DIBUJ. Diego Jarque Pérez				31/05/2021		N.º DE DIBUJO		1-1.1	
VERIF.						MATERIAL:		Z-HIPS	
APROB.						PESO: 62g		ESCALA: 1:2	
FABR.						HOJA 1 DE 1		A3	
CALID.									
OBSERVACIONES									

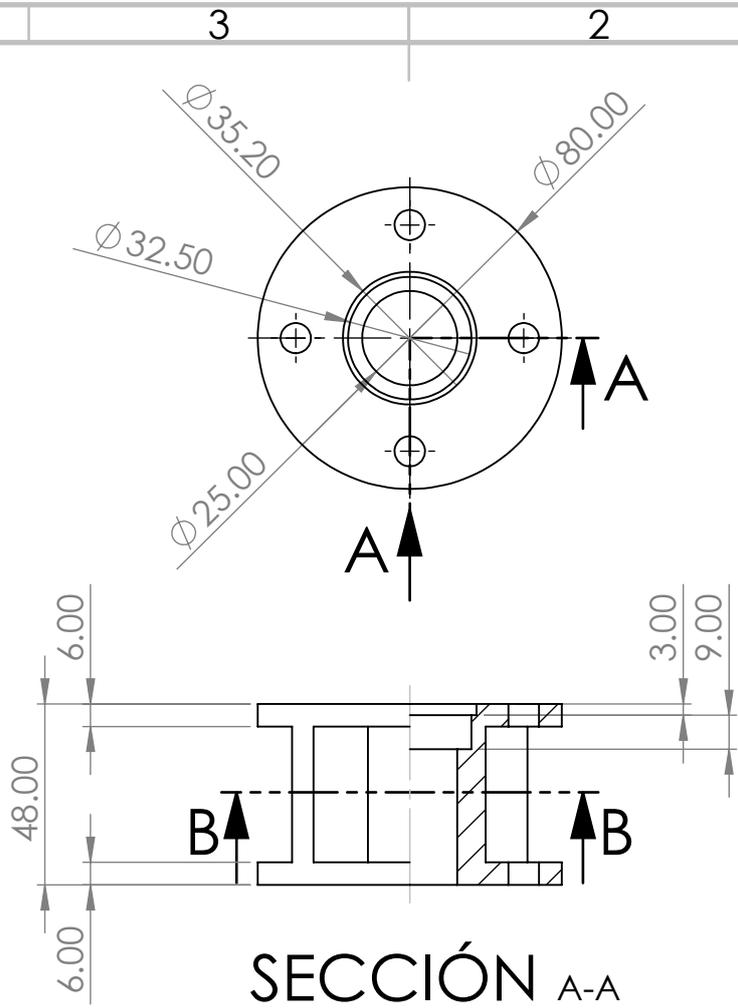


SECCIÓN B-B
ESCALA 1:1

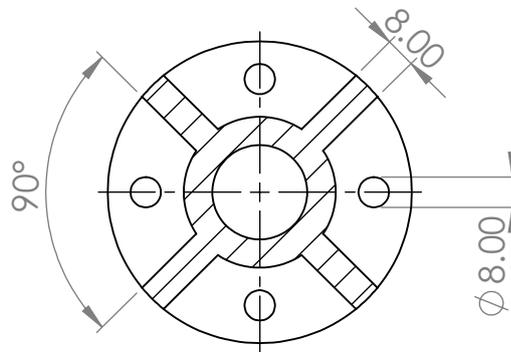


SECCIÓN A-A
ESCALA 1:1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: $\pm 0,1$ mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				31/05/2021		 <p style="text-align: center; font-size: 24pt; font-weight: bold;">Base Superior</p>			
VERIF.									
APROB.									
FABR.									
CALID.						MATERIAL:		N.º DE DIBUJO	
OBSERVACIONES						Z-HIPS		1-1.2	
						PESO: 27g		ESCALA: 1:1	
								HOJA 1 DE 1	

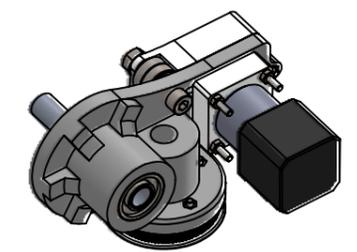
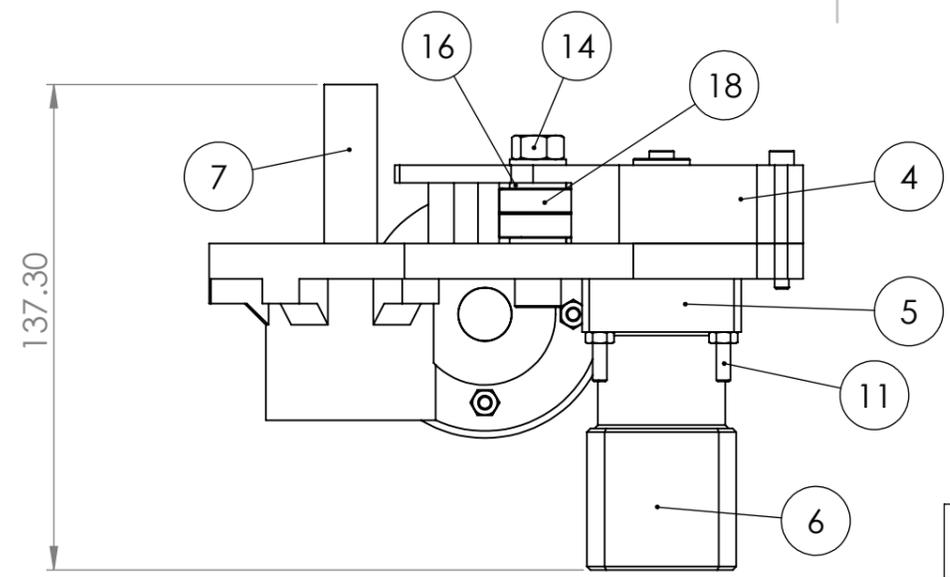


SECCIÓN A-A
ESCALA 1 : 2



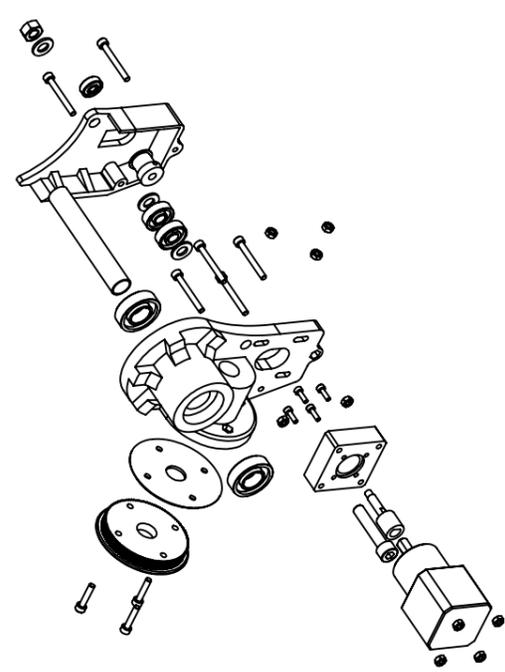
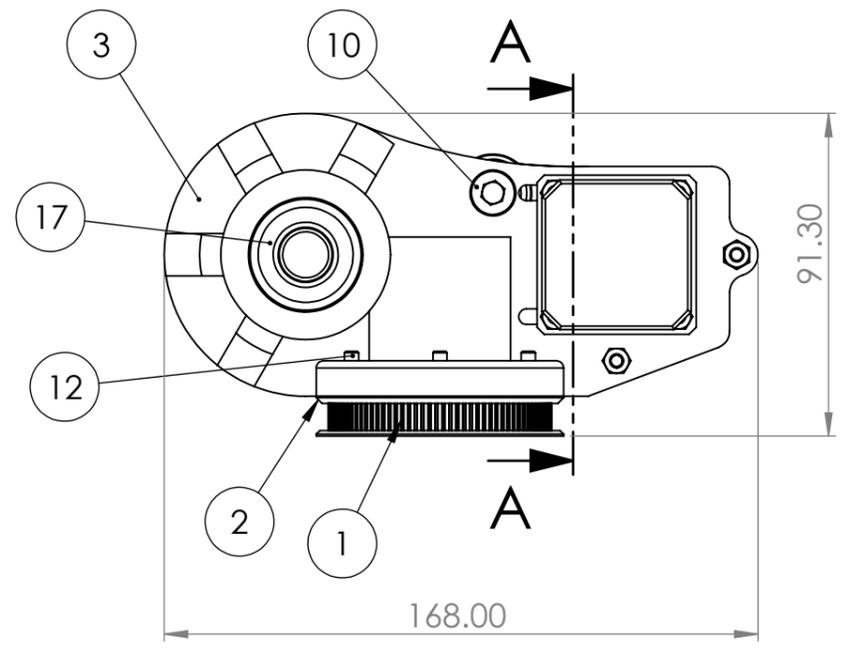
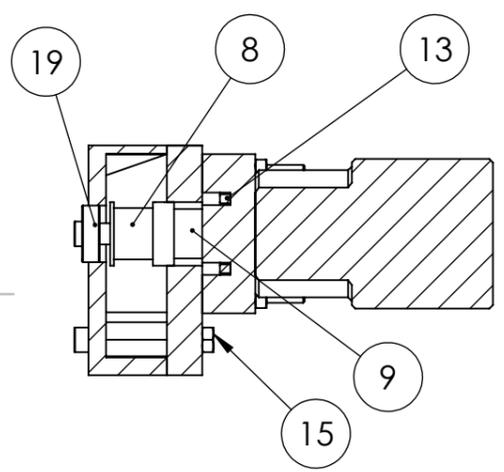
SECCIÓN B-B
ESCALA 1 : 2

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0,1 mm · CANTIDAD: 1		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
NOMBRE		FIRMA	FECHA	TÍTULO:	
DIBUJ.	Diego Jarque Pérez		31/05/2021	Tronco de la Base	
VERIF.					
APROB.					
FABR.					
CALID.				N.º DE DIBUJO	
OBSERVACIONES:			MATERIAL:	1-1.3	A4
			Z-HIPS	ESCALA: 1:2	HOJA 1 DE 1
			PESO: 23g		



ESCALA 1:4

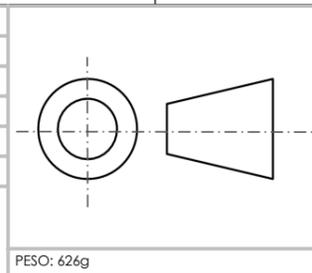
SECCIÓN A-A
ESCALA 1:2



ESCALA 1:5

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Polea Hombro		1
2	Separador Polea Hombro		1
3	Base Hombro		1
4	Cobertura Hombro		1
5	Soporte Motor		1
6	Motor Nema 17 planetary 5:1 gearbox	Modelo 17HS15-1684S-PG5	1
7	Tubo aluminio	90mm longitud 15mm diámetro externo	1
8	Polea 3M HTD	HDT 16 Dientes 6mm ancho	1
9	Adaptador polea		1
10	Tornillo M8 x 40	ISO 4762 M8 x 40 - 28N	1
11	Tornillo M4 x 35	ISO 4762 M4 x 35 - 20N	6
12	Tornillo M4 x 20	ISO 4762 M4 x 20 - 20N	3
13	Tornillo M3 x 10	SO 4762 M3 x 10 - 10N	4
14	Tuerca M8	ISO - 4032 - M8 - D - N	1
15	Tuerca M4	ISO - 4032 - M4 - D - N	9
16	Arandela 8mm	ISO 7089 - 8	3
17	Rodamiento rígido de bolas	ISO 15 ABB - 1015 - 12,SI,NC,12_68	2
18	Rodamiento rígido de bolas	ISO 15 ABB - 108 - 8,SI,NC,8_68	2
19	Rodamiento rígido de bolas	ISO 15 ABB - 025 - 8,SI,NC,8_68	1

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		10/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES		



TÍTULO: **Ensamblaje Hombro**

N.º DE DIBUJO: 1-2 Plano de conjunto

ESCALA: 1:2

HOJA 1 DE 1

PESO: 626g

4 3 2 1

F

F

E

E

D

D

C

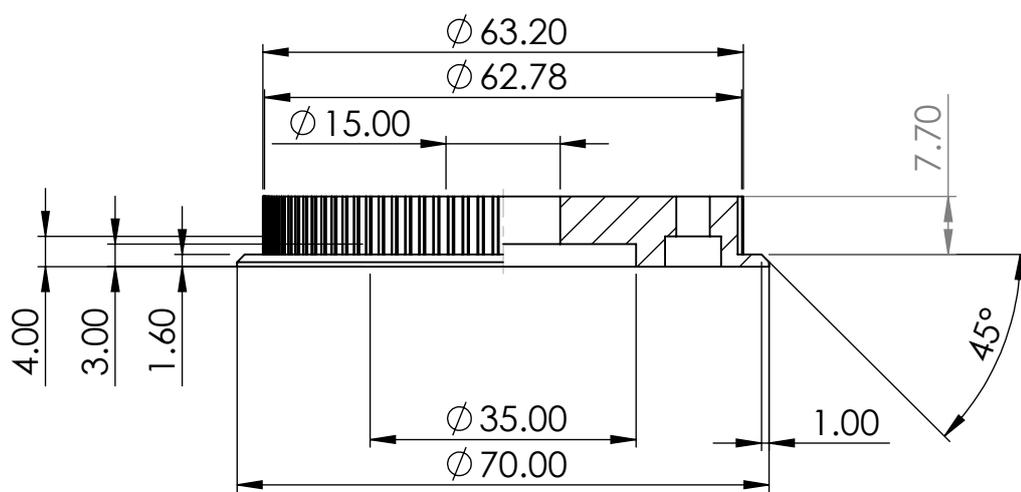
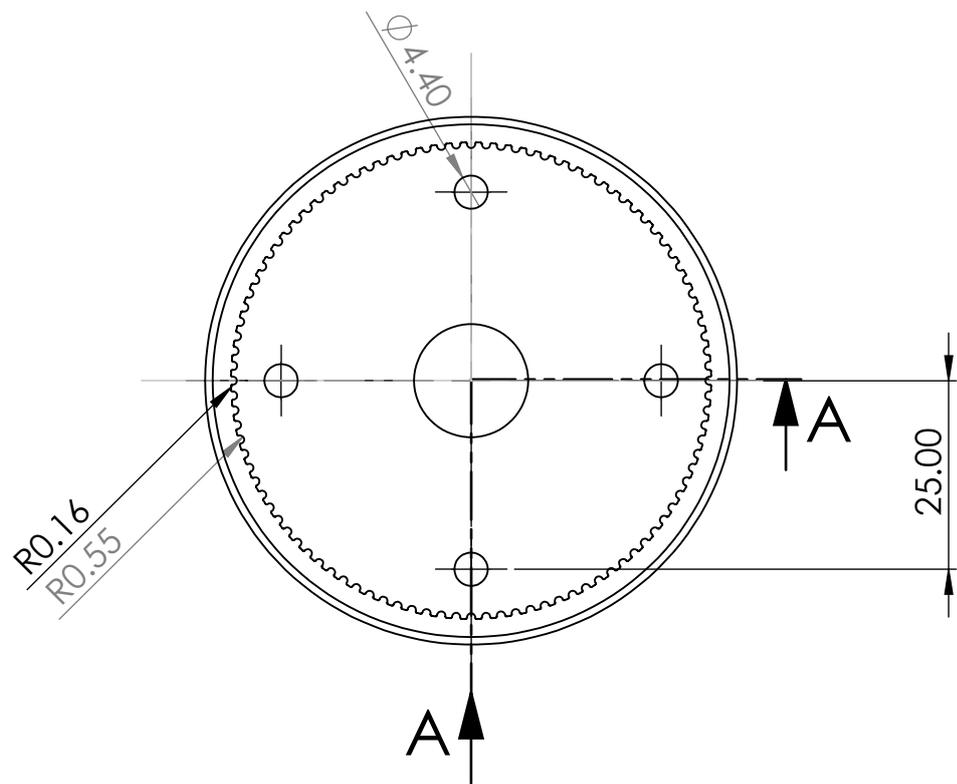
C

B

B

A

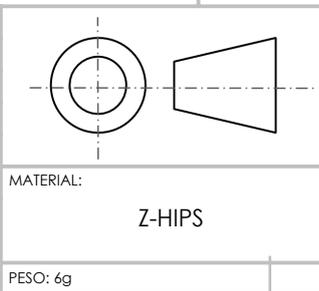
A



SECCIÓN A-A

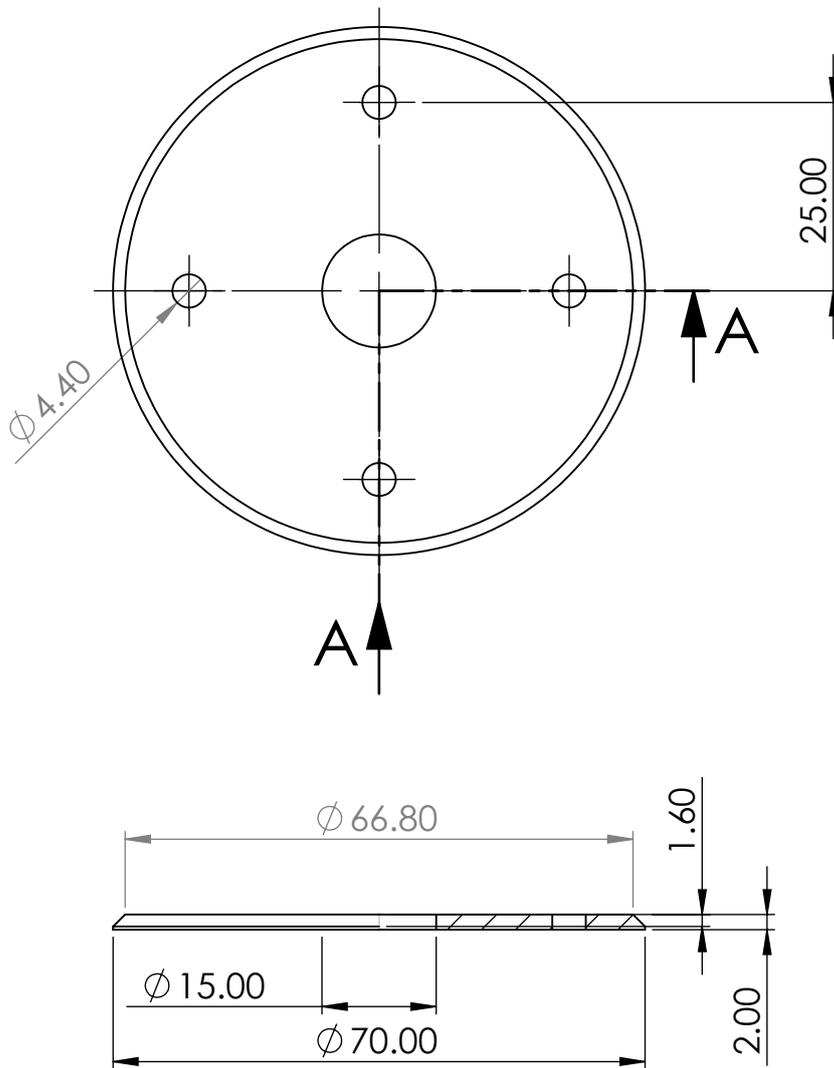
SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0,1 mm · CANTIDAD: 1	ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
---	----------	---------------------------------	---------------------	----------

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		14/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES:		



TÍTULO: <h1>Polea Hombro</h1>	N.º DE DIBUJO <h2>1-2.1</h2>	A4
MATERIAL: Z-HIPS	PESO: 6g	ESCALA: 1:1
		HOJA 1 DE 1

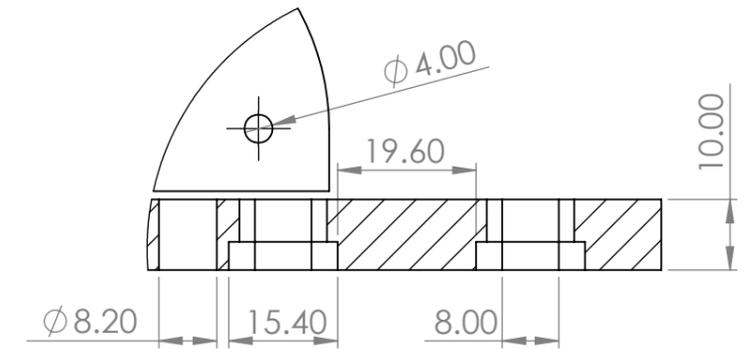
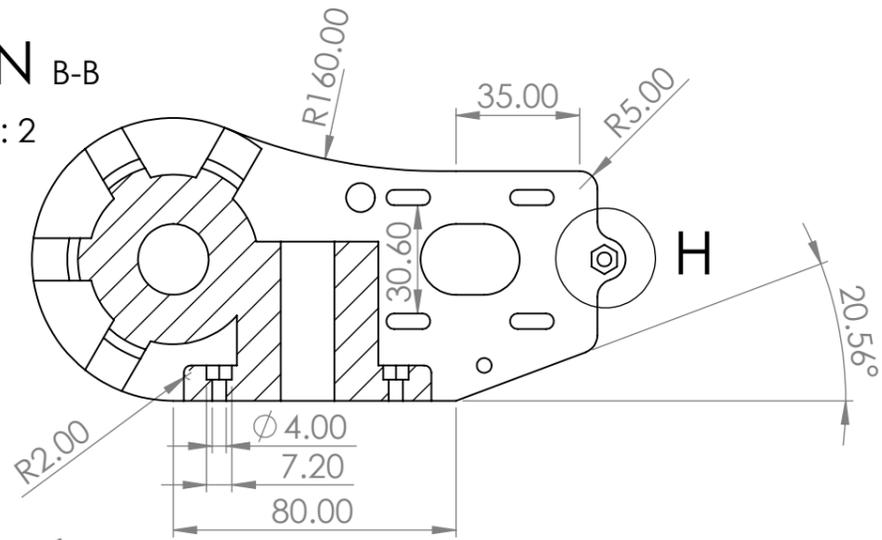
4 3 2 1



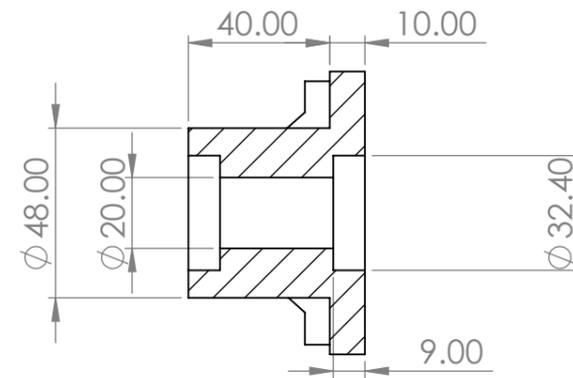
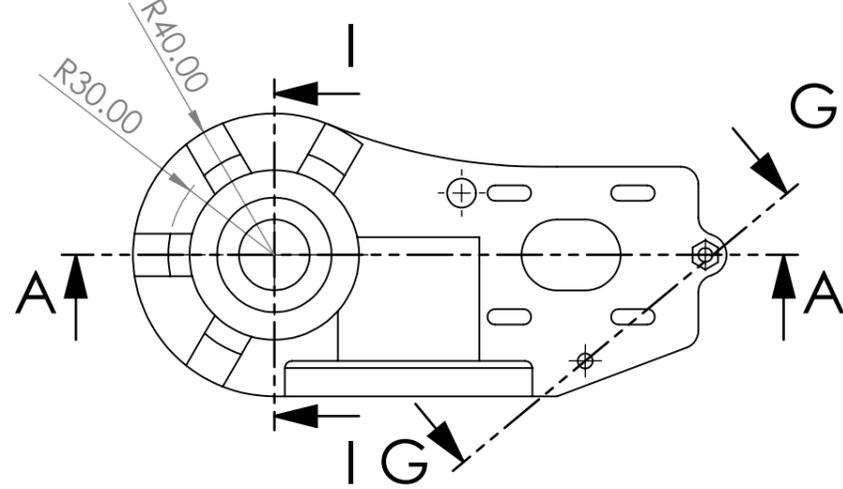
SECCIÓN A-A
ESCALA 1 : 1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0,1 mm · CANTIDAD: 1			ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
NOMBRE			FIRMA	FECHA	TÍTULO: Separador Polea Hombro	
DIBUJ. Diego Jarque Pérez				14/06/2021	N.º DE DIBUJO	
VERIF.					1-2.2	
APROB.					A4	
FABR.					ESCALA: 1:1	
CALID.					HOJA 1 DE 1	
OBSERVACIONES:			MATERIAL: PMMA-XT		PESO: 1,6g	

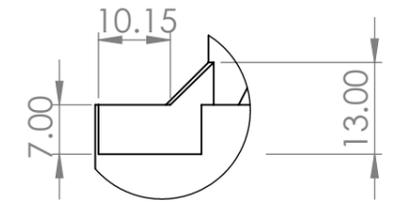
SECCIÓN B-B
ESCALA 1:2



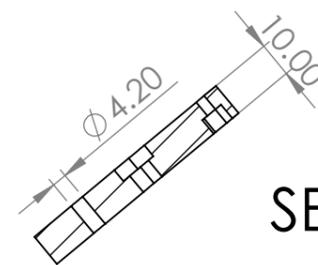
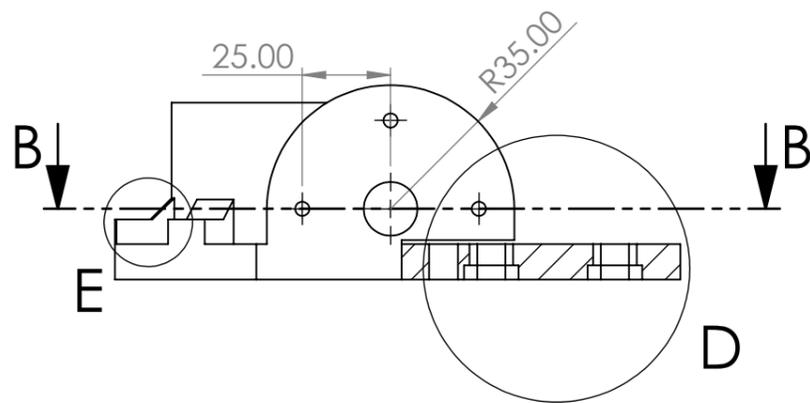
DETALLE D



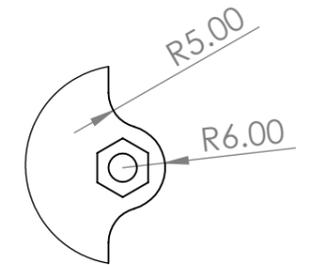
SECCIÓN H-H
ESCALA 1:2



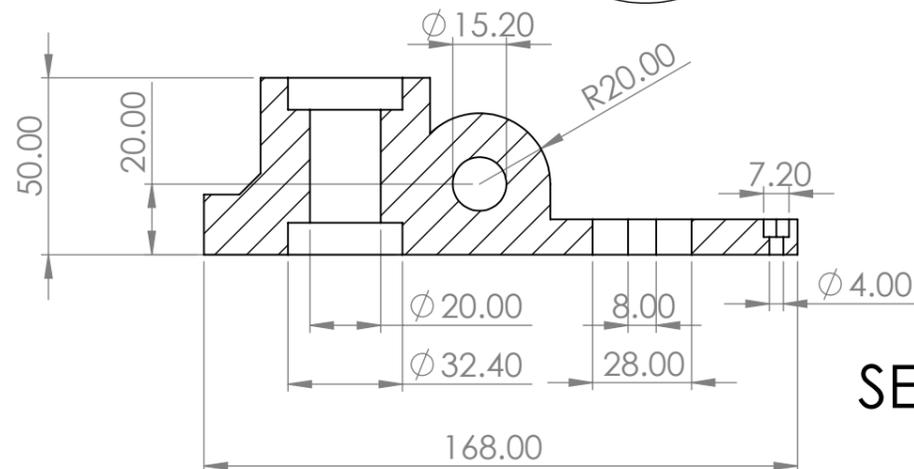
DETALLE E



SECCIÓN G-G
ESCALA 1:2

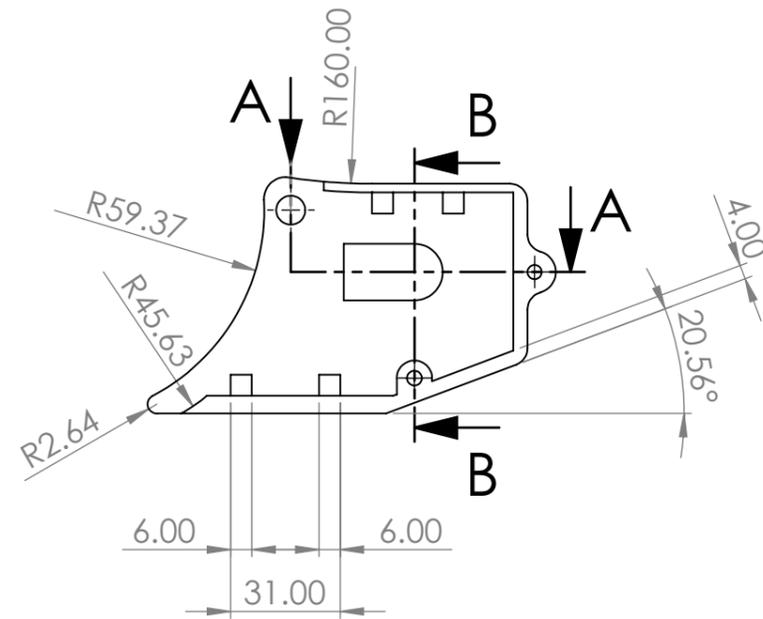
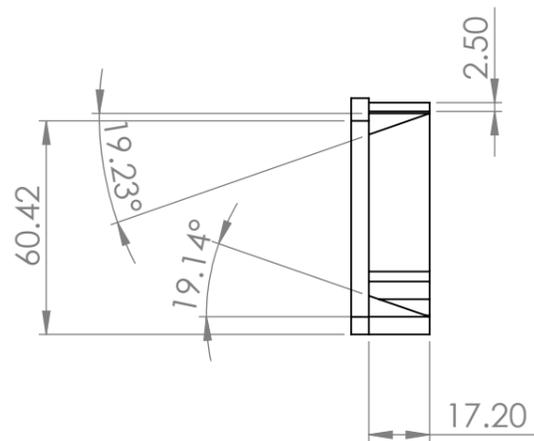


DETALLE H

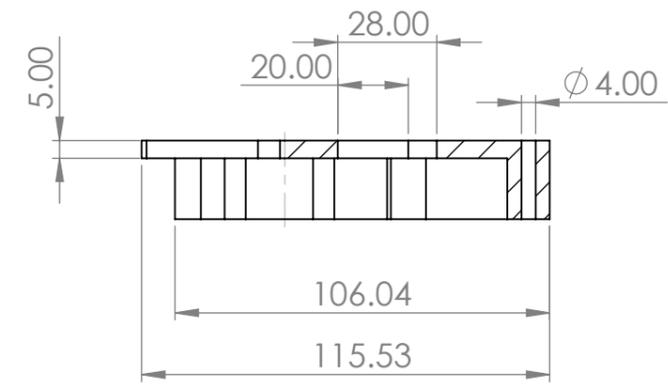
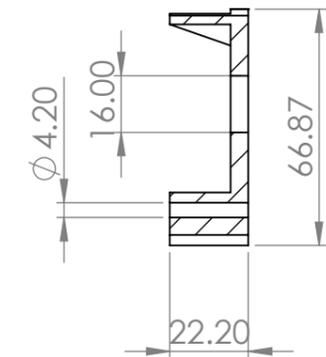


SECCIÓN A-A
ESCALA 1:2

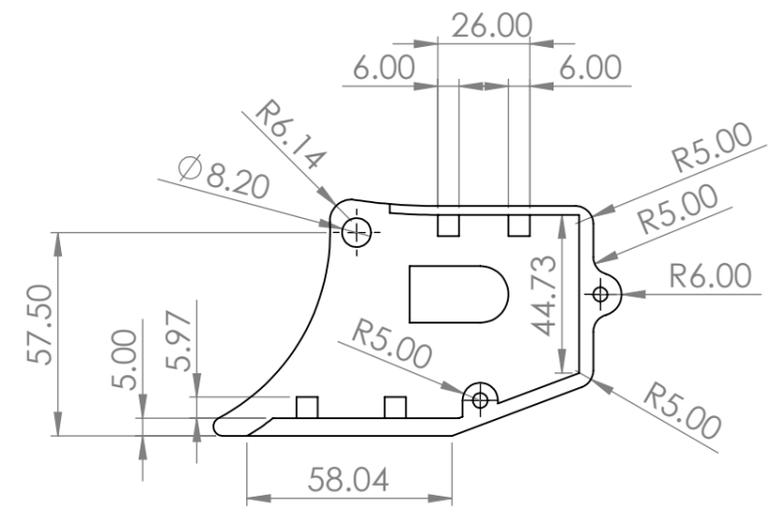
SI NO SE INDICA LO CONTRARIO: - LAS COTAS SE EXPRESAN EN MM - TOLERANCIAS: ± 0,1 mm - CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				14/06/2021		Base Hombro			
VERIF.						N.º DE DIBUJO			
APROB.						MATERIAL:		1-2.3	
FABR.						Z-HIPS		A3	
CALID.						PESO: 47g		ESCALA: 1:2	
OBSERVACIONES								HOJA 1 DE 1	

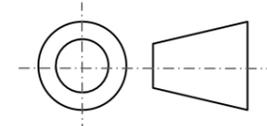


SECCIÓN B-B
ESCALA 1 : 2



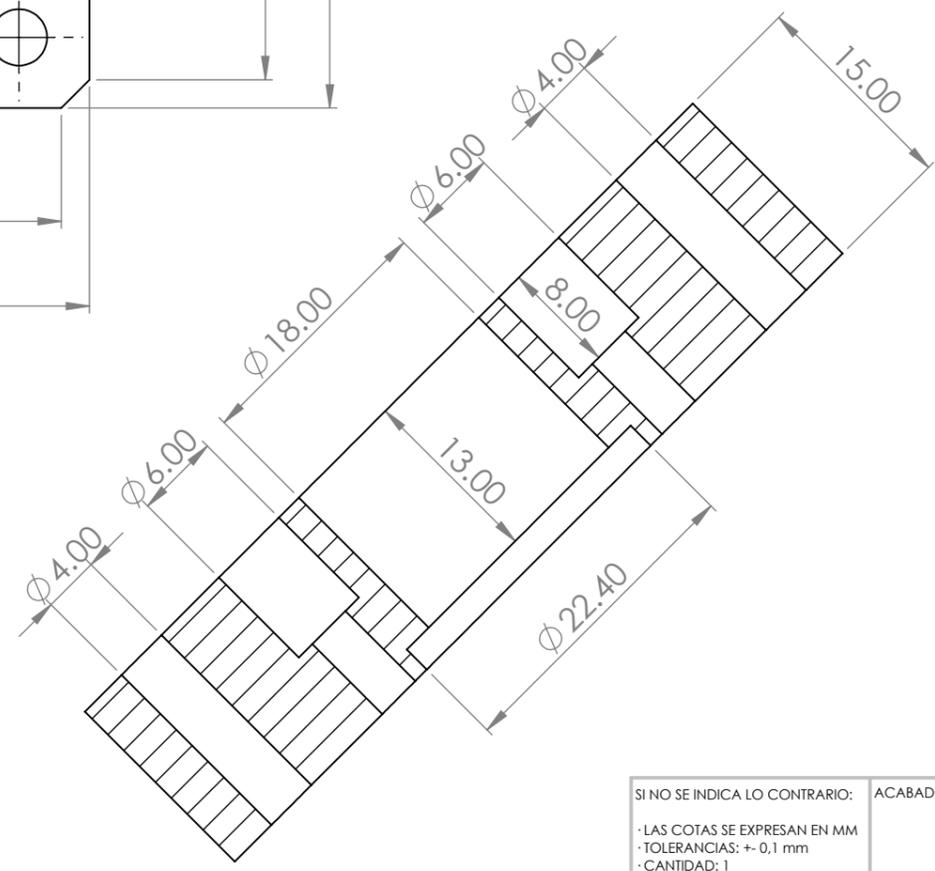
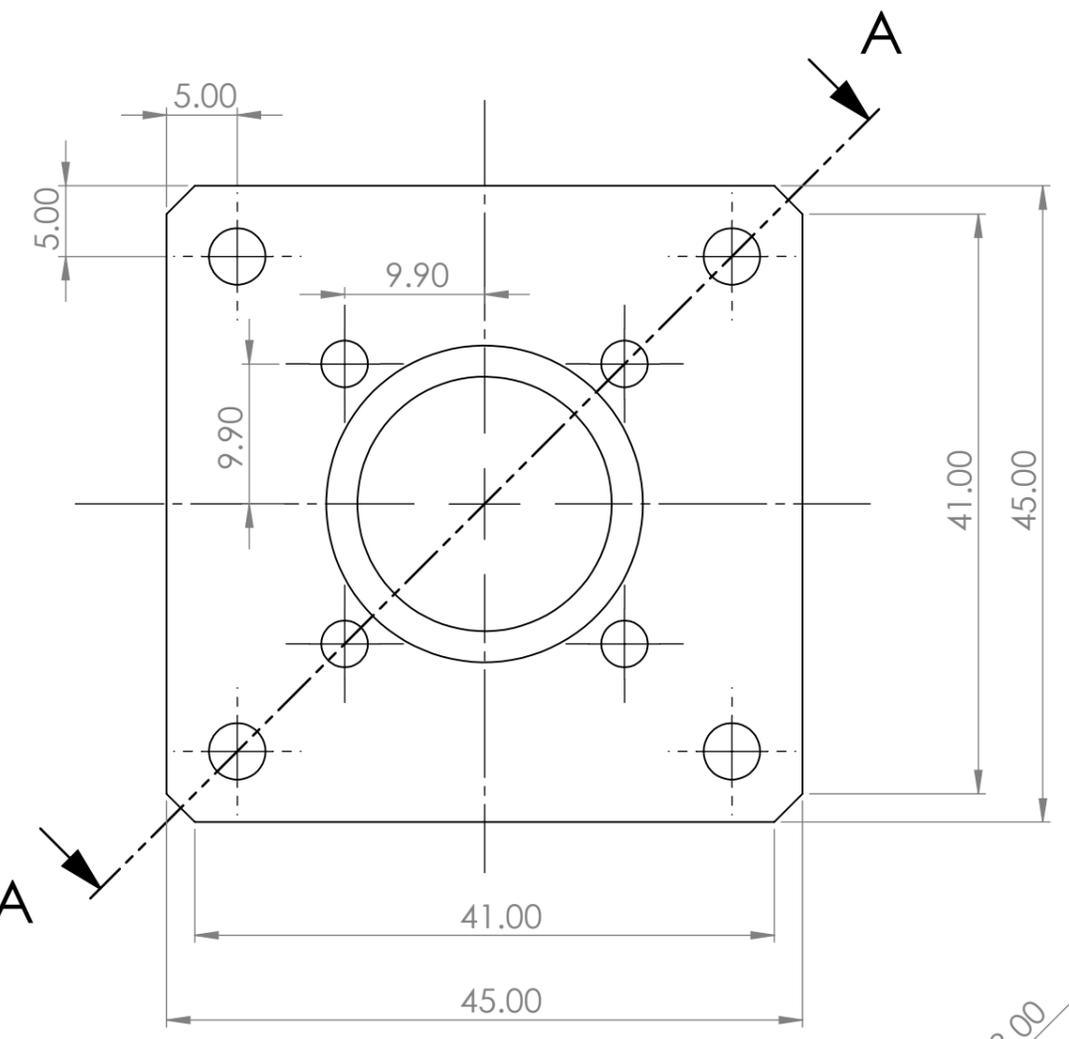
SECCIÓN A-A
ESCALA 1 : 2



SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				15/06/2021		 <p>Cobertura Hombro</p>			
VERIF.									
APROB.									
FABR.									
CALID.									
OBSERVACIONES				MATERIAL:		N.º DE DIBUJO		A3	
				Z-HIPS		1-2.4			
				PESO: 9,4g		ESCALA: 1:2		HOJA 1 DE 1	

8 7 6 5 4 3 2 1

F
E
D
C
B
A

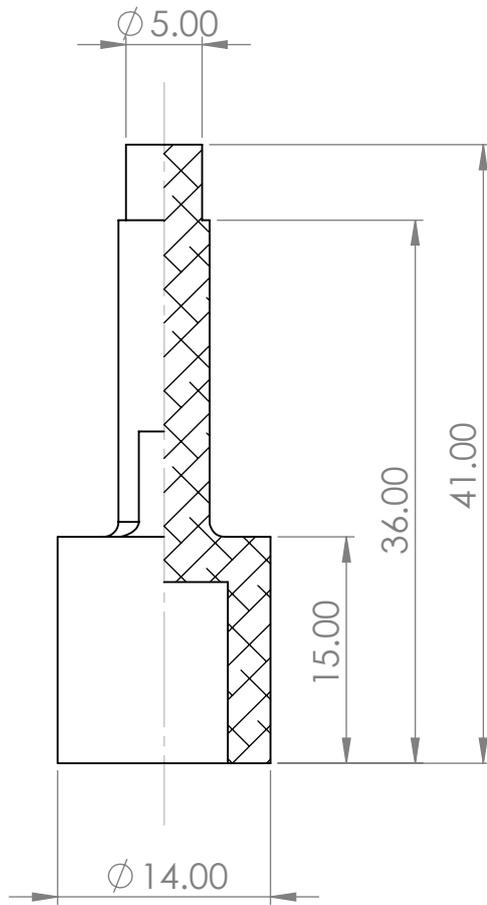
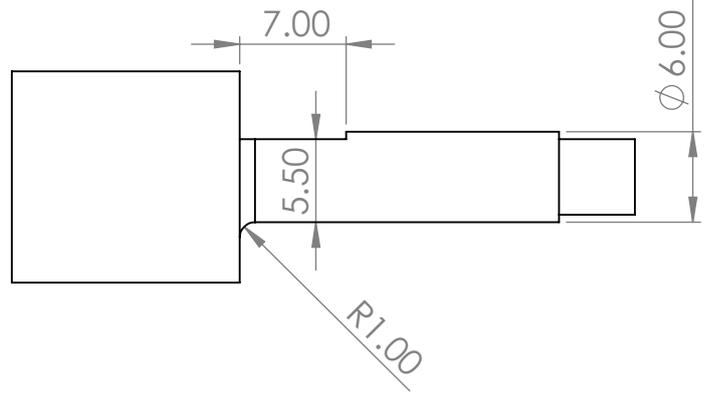
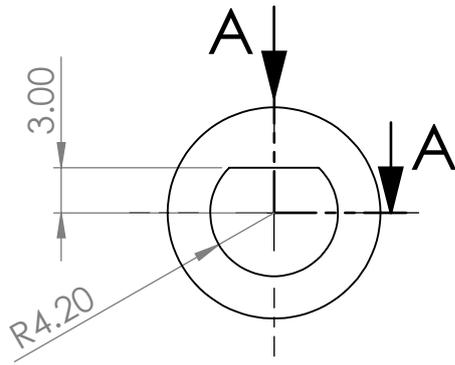


SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
DIBUJ. Diego Jarque Pérez		FIRMA		FECHA 15/06/2021		TÍTULO: Soporte Motor			
VERIF.						N.º DE DIBUJO 1-2.5		A3	
APROB.						MATERIAL: Z-HIPS		ESCALA: 2:1	
FABR.						PESO: 5,6g		HOJA 1 DE 1	
CALID.									
OBSERVACIONES									

F
E
D
C
B
A

8 7 6 5 4 3 2 1



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO:
 · LAS COTAS SE EXPRESAN EN MM
 · TOLERANCIAS: +0,1 mm
 · CANTIDAD: 1

ACABADO:

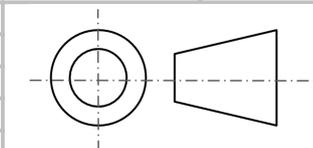
REBARBAR Y ROMPER ARISTAS VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		15/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			

OBSERVACIONES:



MATERIAL:

Z-HIPS

TÍTULO:

Adaptador Polea

N.º DE DIBUJO

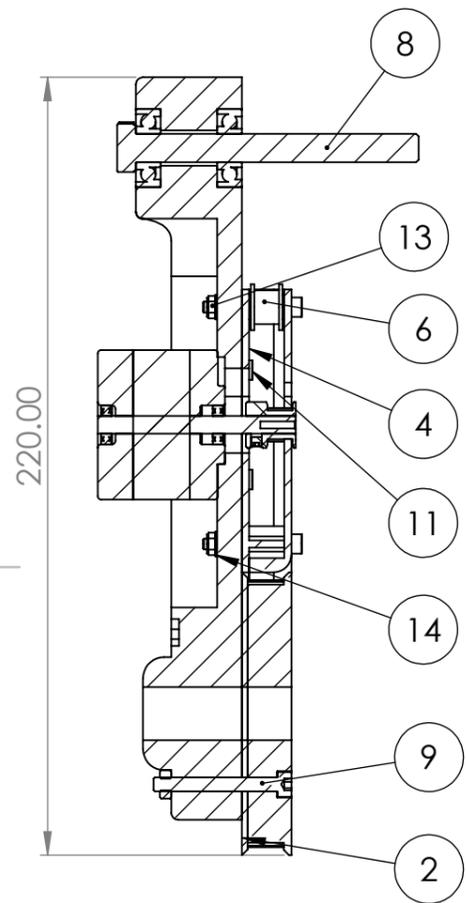
1-2.9

A4

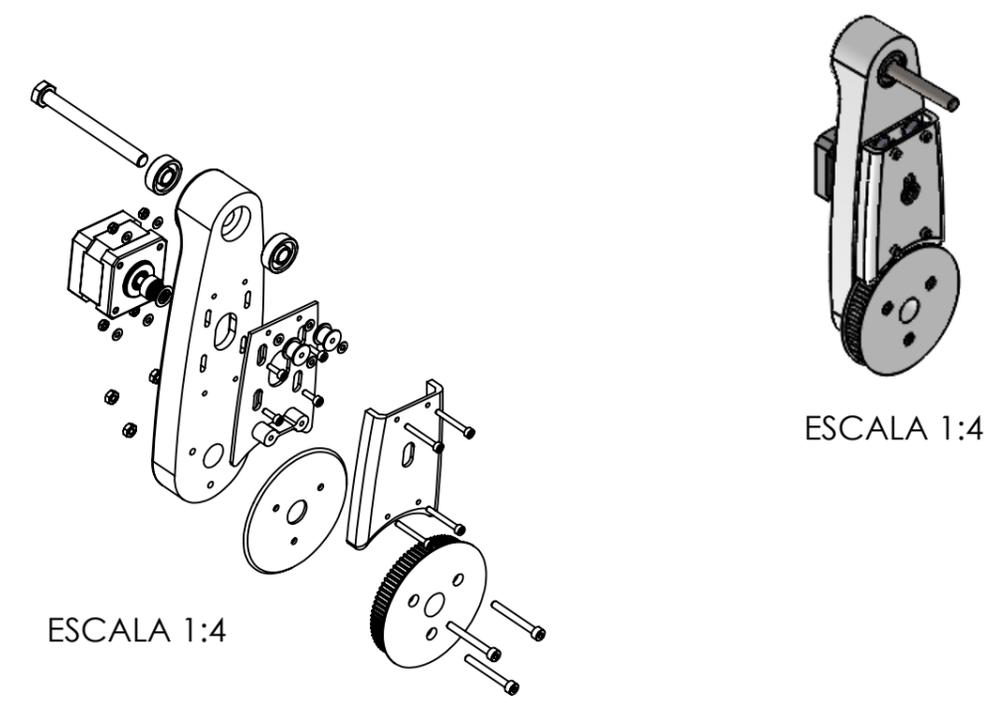
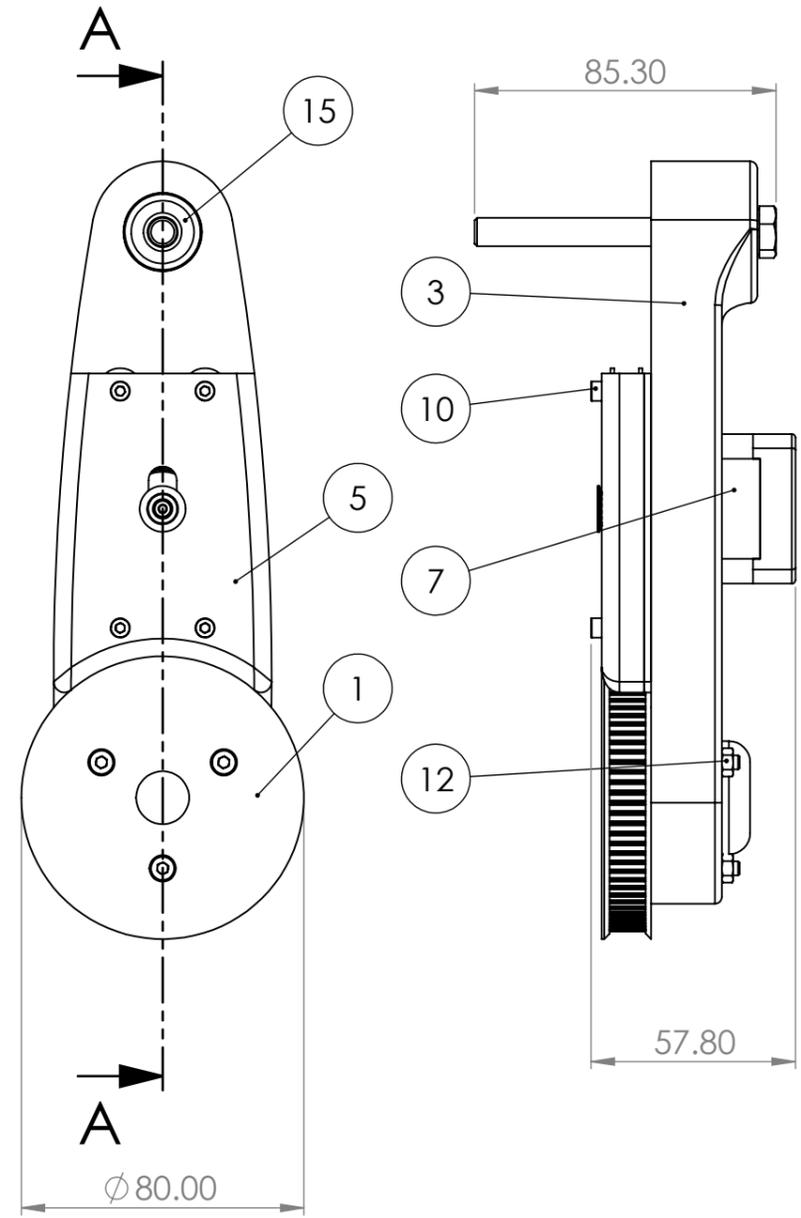
PESO: 0,6g

ESCALA:2:1

HOJA 1 DE 1

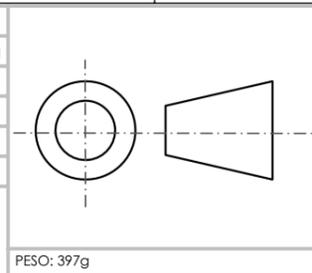


SECCIÓN A-A
ESCALA 1:2



N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Polea Brazo		1
2	Separador Polea Brazo		1
3	Base Brazo		1
4	Separador		1
5	Cobertura Brazo		1
6	Polea no dentada	Polea no dentada 16T GT2	2
7	Motor NEMA 17	Modelo 17HS16-2004S1	1
8	Tornillo M8 x 80	ISO 4014 - M8 x 80 x 80-N	1
9	Tornillo M4 x 35	ISO 4762 M4 x 35 - 20N	3
10	Tornillo M3 x 25	ISO 4762 M3 x 25 - 18N	4
11	Tornillo M3 x 10	ISO 4762 M3 x 10 - 10N	4
12	Tuerca M4	ISO - 4032 - M4 - D - N	3
13	Tuerca M3	ISO - 4032 - M3 - W - N	4
14	Arandela 3mm	Washer ISO 7089 - 3	8
15	Rodamiento rígido de bolas	ISO 15 ABB - 108 - 8,SI,NC,8_68	2

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		15/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES		



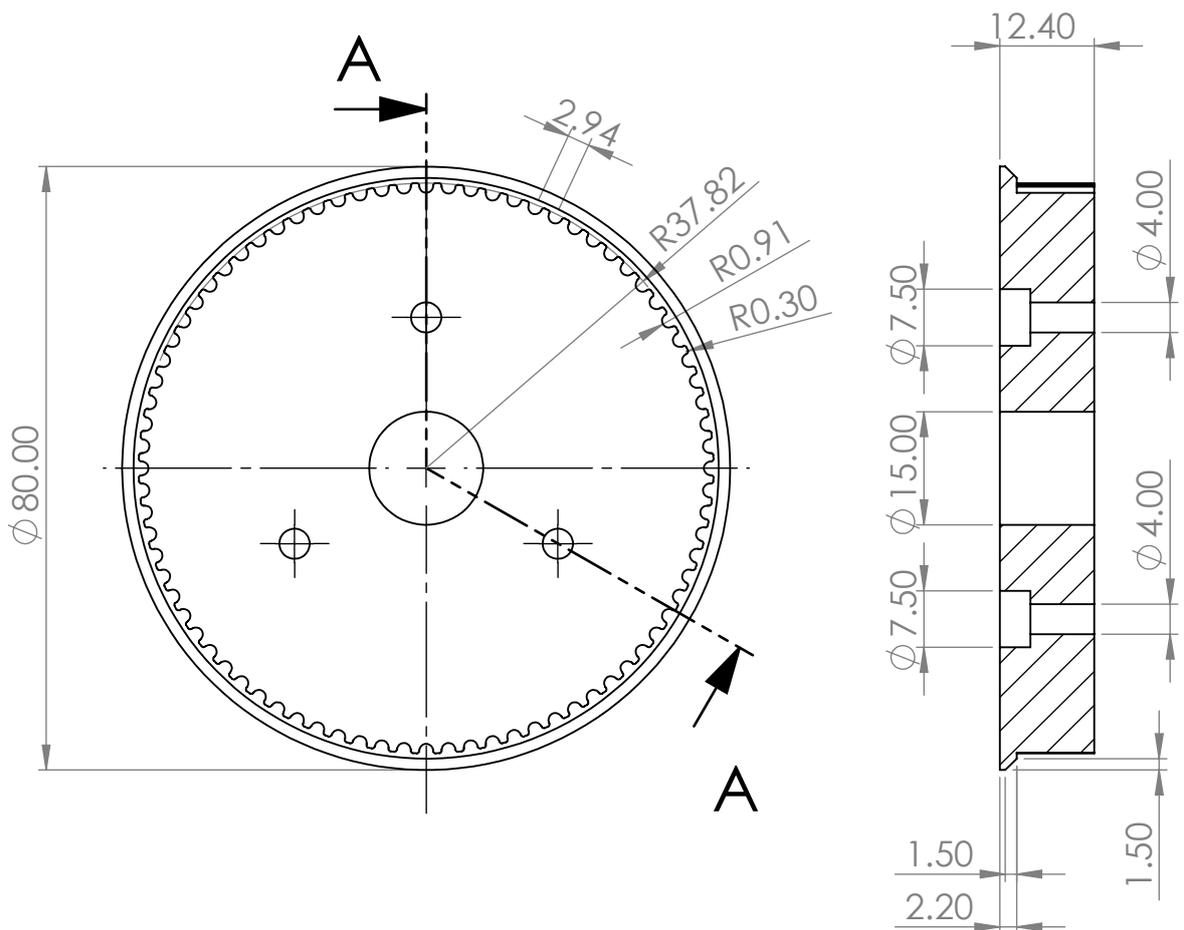
TÍTULO:
Ensamblaje Brazo

N.º DE DIBUJO
1-3 Plano de conjunto

ESCALA:1:2

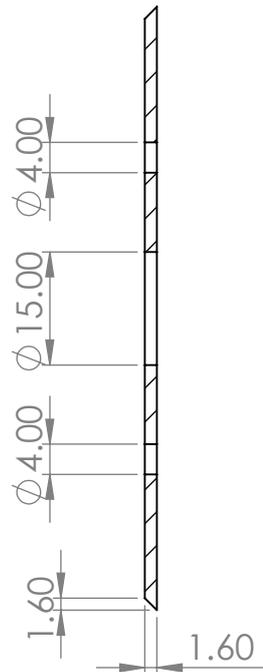
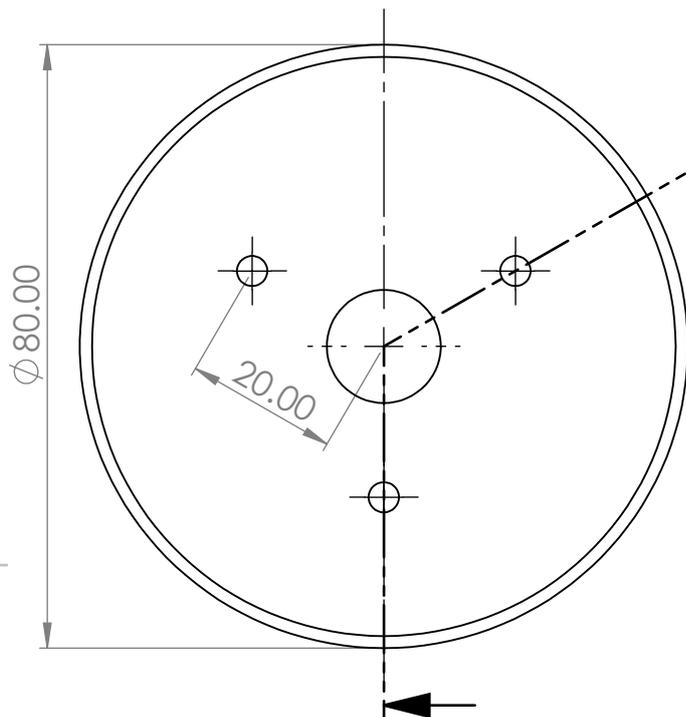
HOJA 1 DE 1

PESO: 397g



SECCIÓN A-A

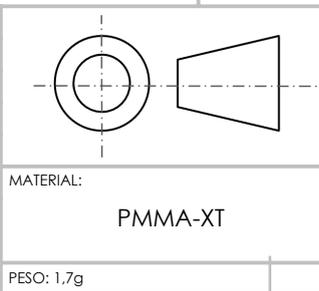
SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0,1 mm · CANTIDAD: 1		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
NOMBRE	FIRMA	FECHA		TÍTULO: Polea Brazo	
DIBUJ.	Diego Jarque Pérez	15/06/2021		N.º DE DIBUJO 1-3.1	
VERIF.				A4	
APROB.				ESCALA: 1:1	
FABR.				HOJA 1 DE 1	
CALID.			MATERIAL: Z-HIPS	PESO: 12g	
OBSERVACIONES:					



SECCIÓN A-A
ESCALA 1 : 1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0.05 mm · CANTIDAD: 1	ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
--	----------	---------------------------------	---------------------	----------

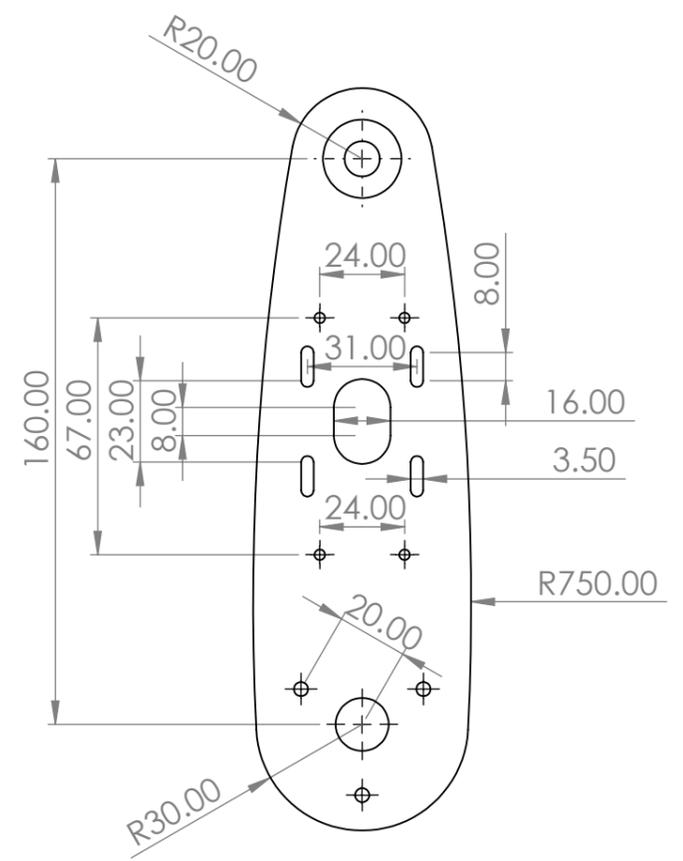
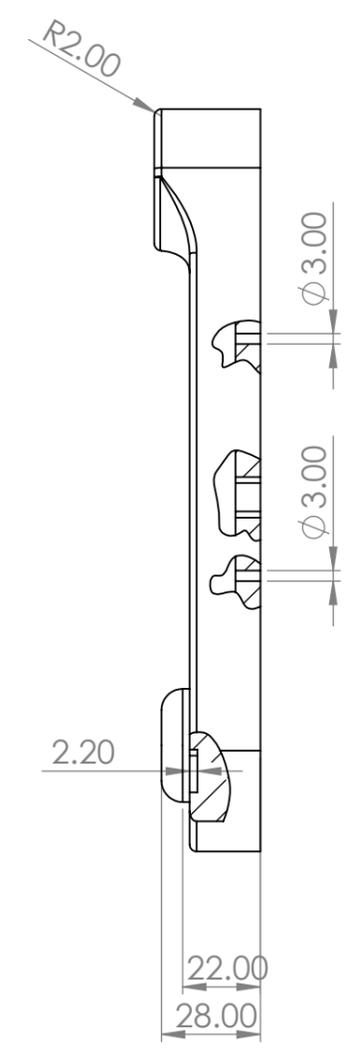
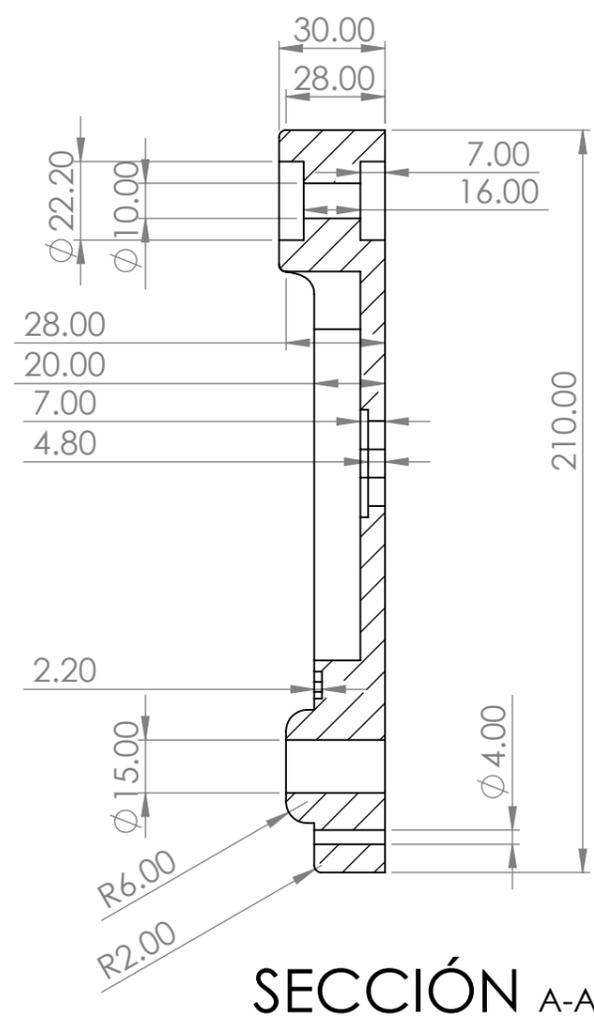
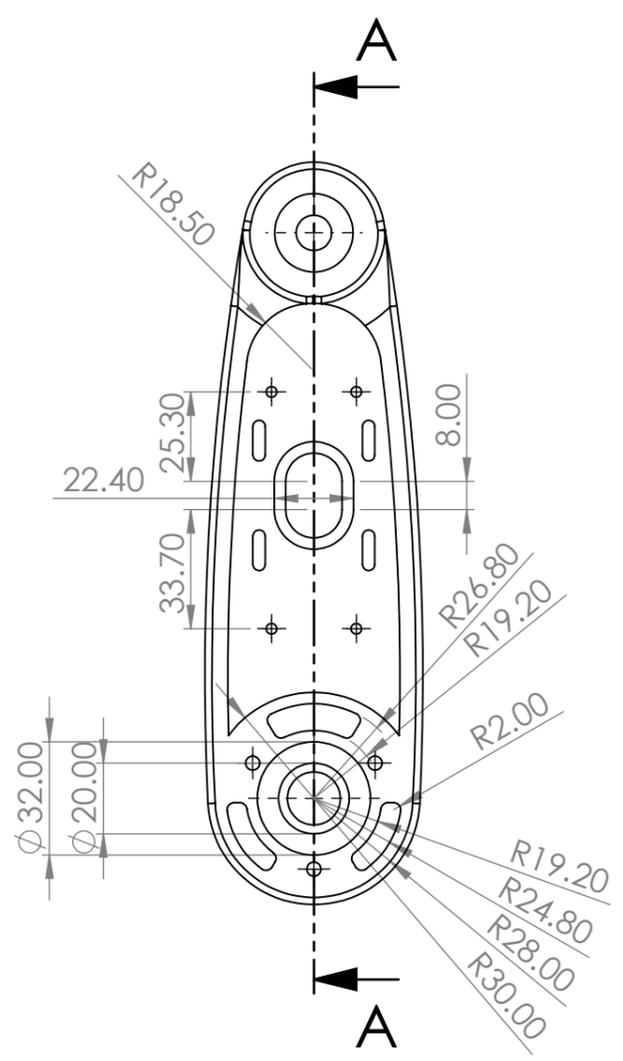
	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		15/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			
OBSERVACIONES:			



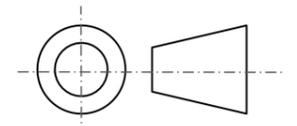
TÍTULO:	Separador Polea Brazo	
N.º DE DIBUJO	1-3.2	A4
PESO: 1.7g	ESCALA: 1:1	HOJA 1 DE 1

8 7 6 5 4 3 2 1

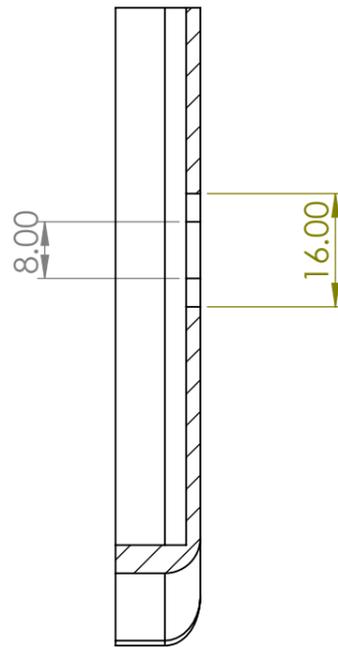
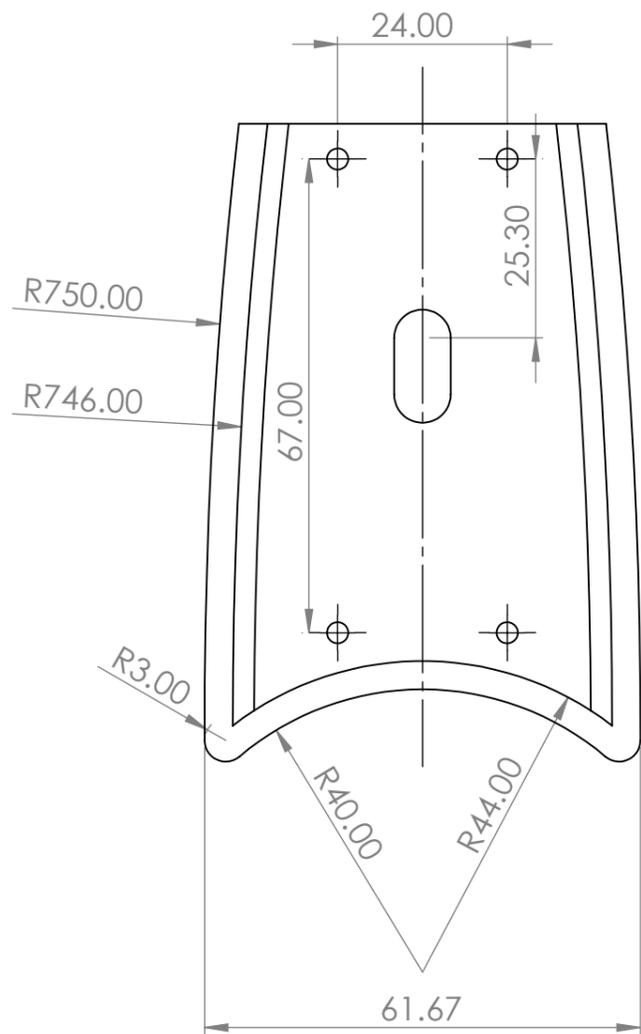
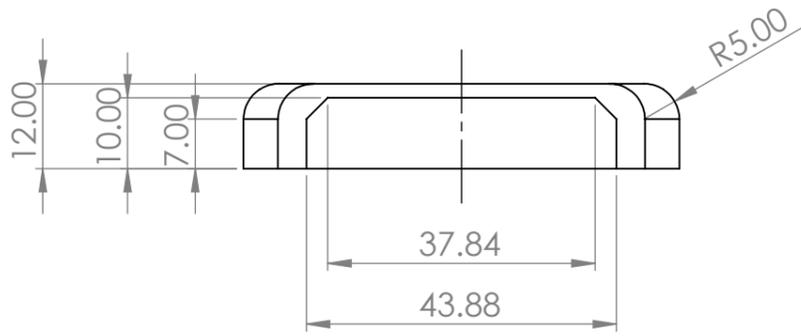
F E D C B A



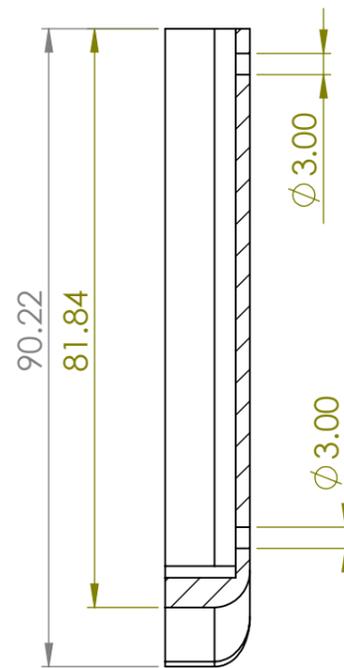
A B C D E F

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				15/06/2021		 <h1 style="text-align: center;">Base Brazo</h1>			
VERIF.									
APROB.									
FABR.									
CALID.						MATERIAL:		N.º DE DIBUJO	
OBSERVACIONES						Z-HIPS		1-3.3	
						PESO: 37g		ESCALA: 1:2	
								HOJA 1 DE 1	

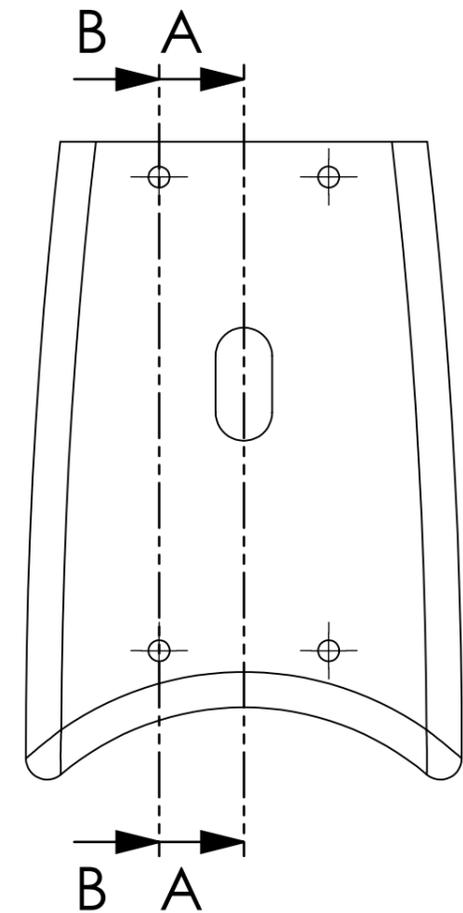
8 7 6 5 4 3 2 1



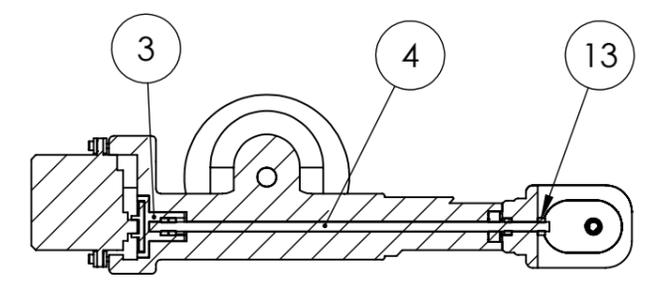
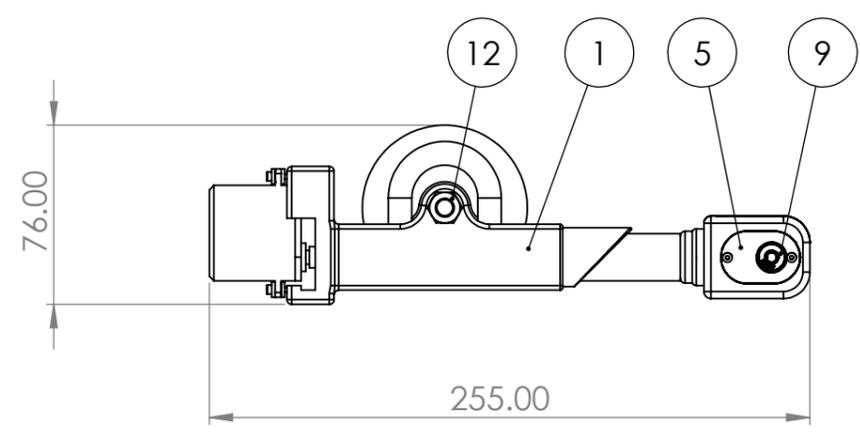
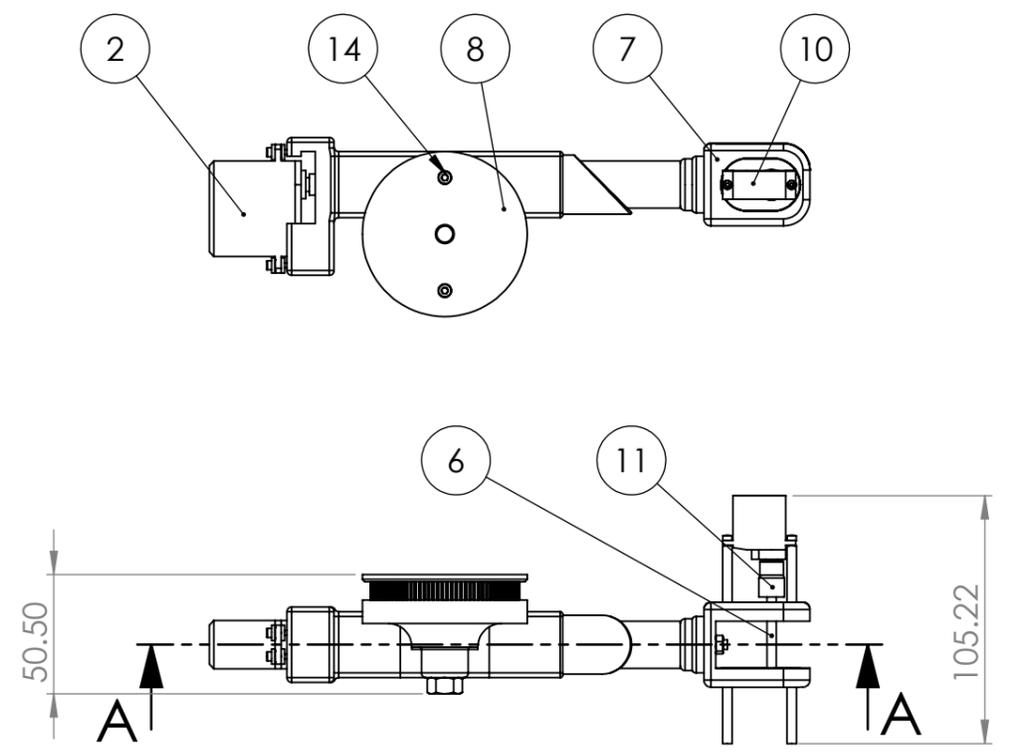
SECCIÓN A-A



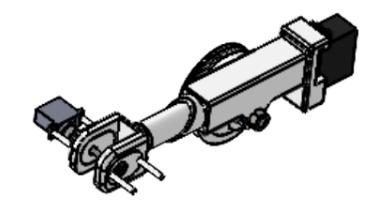
SECCIÓN B-B



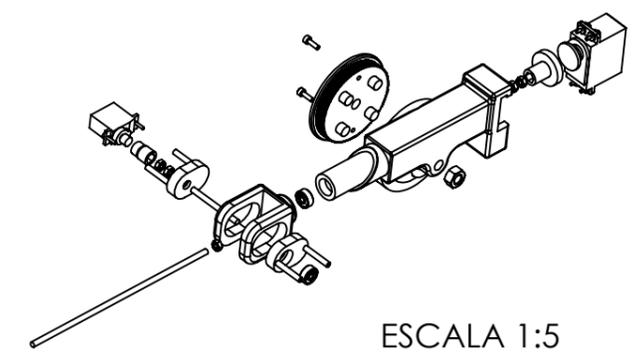
SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: $\pm 0,1$ mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				15/06/2021		Cobertura Brazo			
VERIF.						N.º DE DIBUJO			
APROB.						1-3.5			
FABR.						MATERIAL:		A3	
CALID.						Z-HIPS		ESCALA:1:1	
OBSERVACIONES						PESO: 4.3g		HOJA 1 DE 1	



SECCIÓN A-A
ESCALA 1 : 3



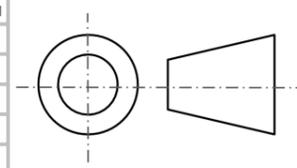
ESCALA 1:5



ESCALA 1:5

N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Antebrazo Principal		1
2	Servomotor	Modelo DS04-NFC	1
3	Acople Eje Principal		1
4	Eje Principal	Varilla roscada M4 170mm	1
5	Adaptador Motor		2
6	Eje Secundario	Varilla roscada M4 50mm	1
7	Conexión Muñeca		1
8	Polea Antebrazo		1
9	Rodamiento rígido de bolas	Modelo 624RS	3
10	Microservomotor	Modelo MG90D	1
11	Acople Eje Secundario		1
12	Tuerca M8	ISO - 4032 - M8 - D - N	1
13	Tuerca M4	ISO - 4032 - M4 - D - N	6
14	Tornillo M3 x 10	ISO 4762 M3 x 10 - 10N	2

NOMBRE	FIRMA	FECHA
DIBUJ. Diego Jarque Pérez		17/06/2021
VERIF.		
APROB.		
FABR.		
CALID.		
OBSERVACIONES		



PESO: 166g

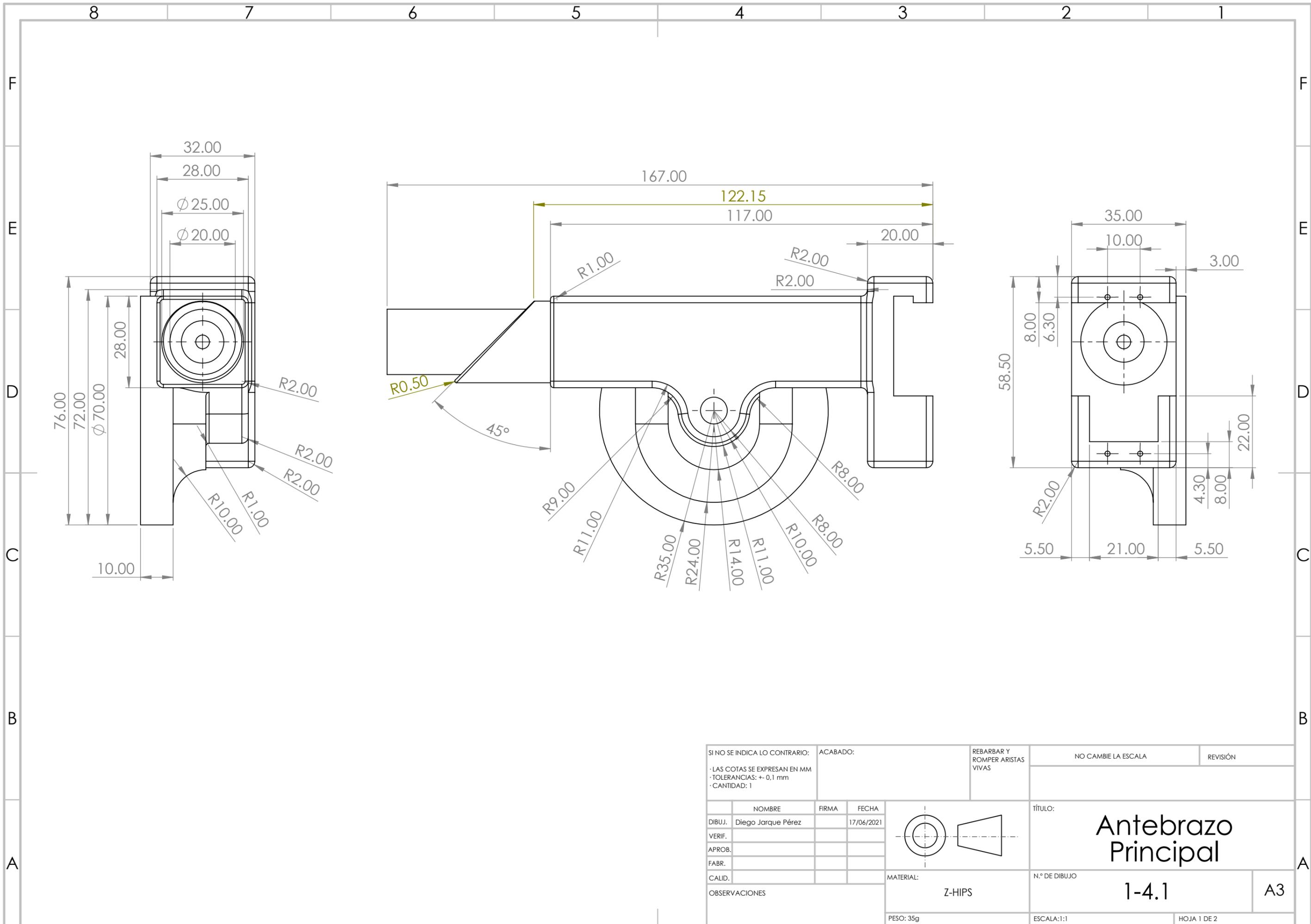
TÍTULO: **Ensamblaje Antebrazo**

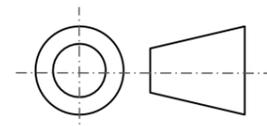
N.º DE DIBUJO: **1-4 Plano de conjunto**

ESCALA: 1:3

HOJA 1 DE 1

A3

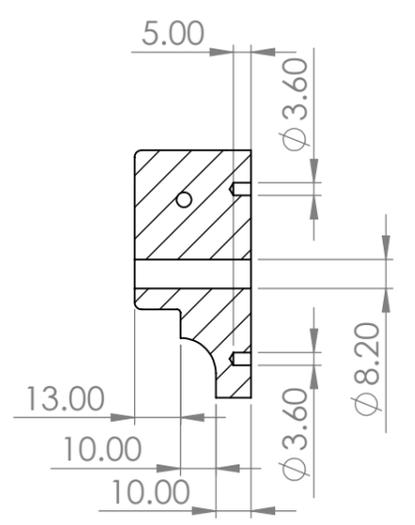


SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: $\pm 0,1$ mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
DIBUJ. Diego Jarque Pérez		FIRMA		FECHA 17/06/2021		 <p>TÍTULO: Antebrazo Principal</p>			
VERIF.									
APROB.									
FABR.									
CALID.									
OBSERVACIONES				MATERIAL: Z-HIPS		N.º DE DIBUJO: 1-4.1		A3	
				PESO: 35g		ESCALA: 1:1		HOJA 1 DE 2	

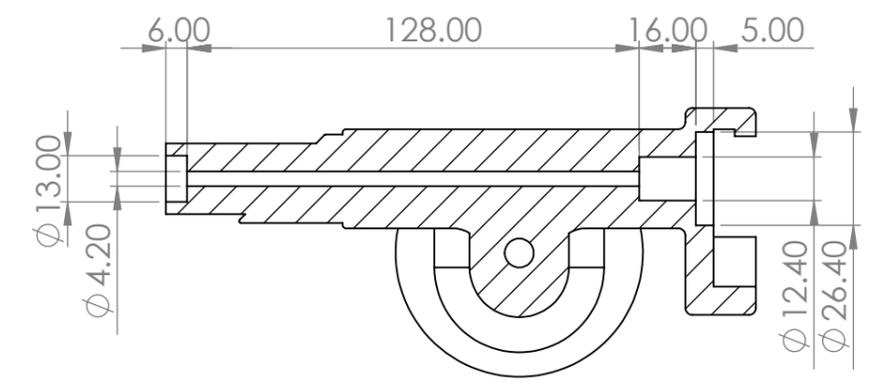
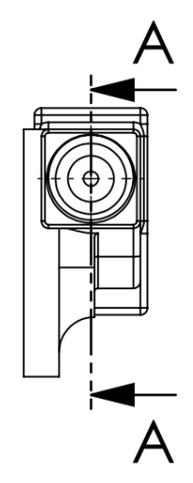
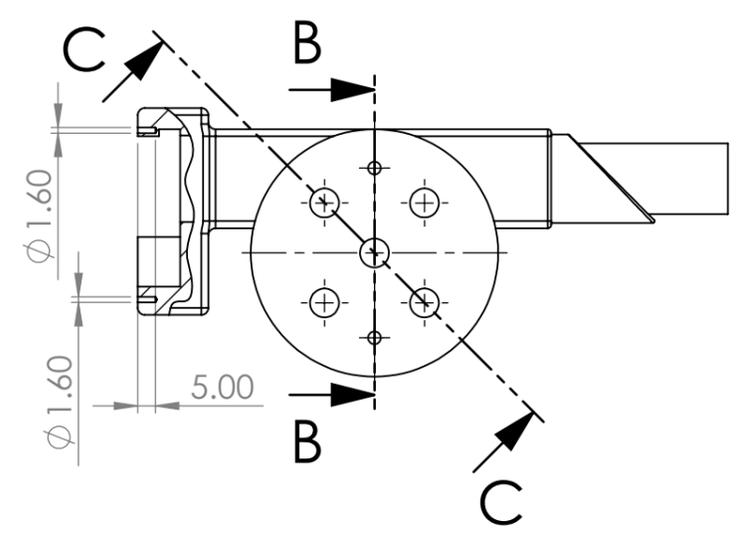
8 7 6 5 4 3 2 1

F
E
D
C
B
A

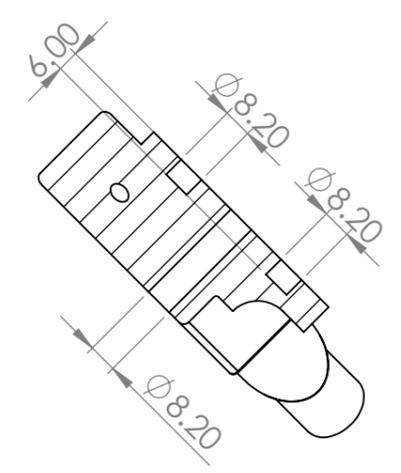
F
E
D
C
B
A



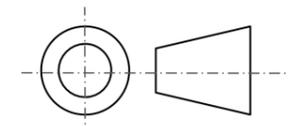
SECCIÓN B-B



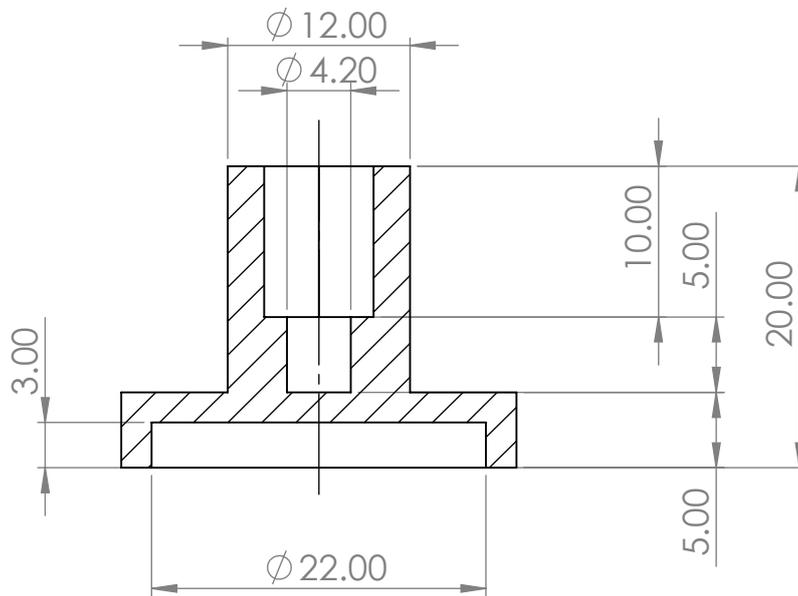
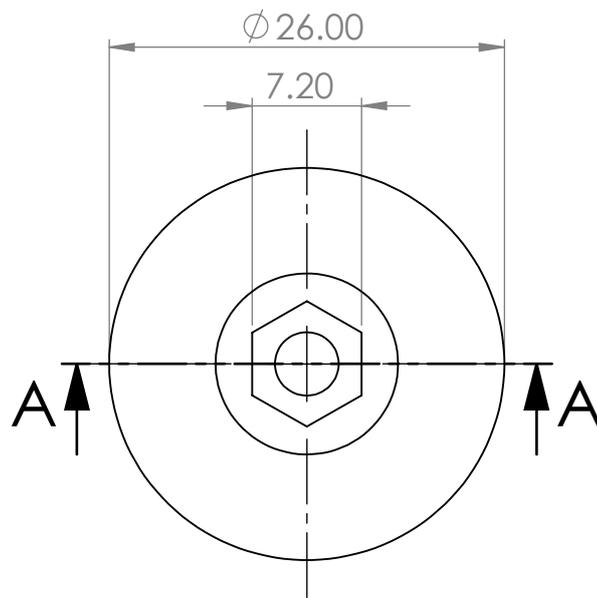
SECCIÓN A-A



SECCIÓN C-C

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
DIBUJ. Diego Jarque Pérez		FIRMA		FECHA 17/06/2021		 <p>TÍTULO: Antebrazo Principal</p>			
VERIF.									
APROB.									
FABR.									
CALID.									
OBSERVACIONES				MATERIAL: Z-HIPS		N.º DE DIBUJO: 1-4.1		A3	
				PESO: 35g		ESCALA: 1:2		HOJA 2 DE 2	

8 7 6 5 4 3 2 1



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO:
 · LAS COTAS SE EXPRESAN EN MM
 · TOLERANCIAS: +0,1 mm
 · CANTIDAD: 1

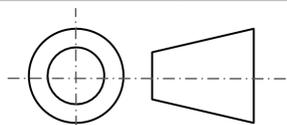
ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		17/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			



TÍTULO:

Acople Eje Principal

OBSERVACIONES:

MATERIAL:

Z-HIPS

N.º DE DIBUJO

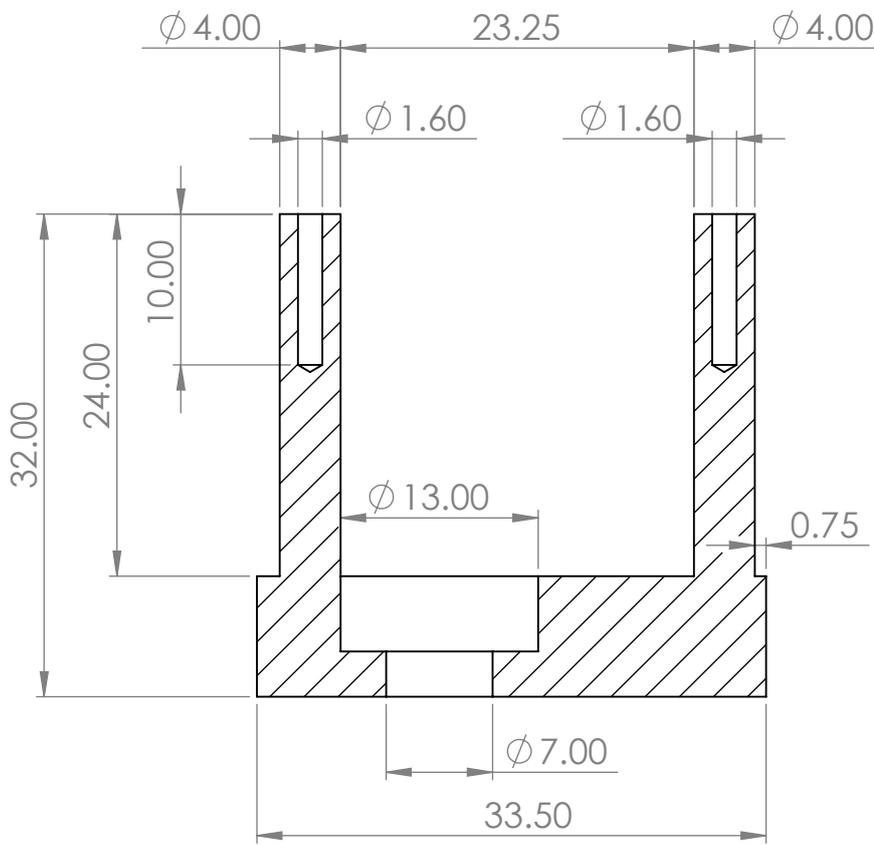
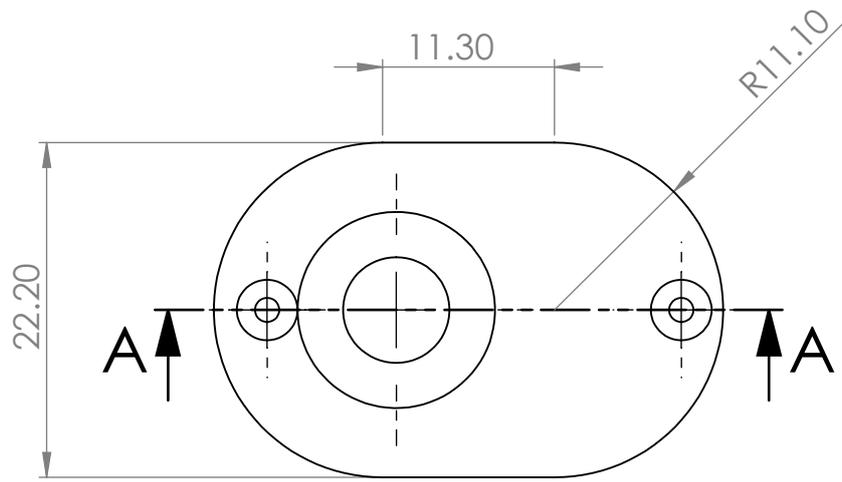
1-4.3

A4

PESO: 0,6g

ESCALA:2:1

HOJA 1 DE 1



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO:
 · LAS COTAS SE EXPRESAN EN MM
 · TOLERANCIAS: +0,1 mm
 · CANTIDAD: 2

ACABADO:

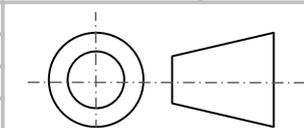
REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		17/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			

OBSERVACIONES:



MATERIAL:

Z-HIPS

PESO: 1,1g

TÍTULO:

Adaptador Motor

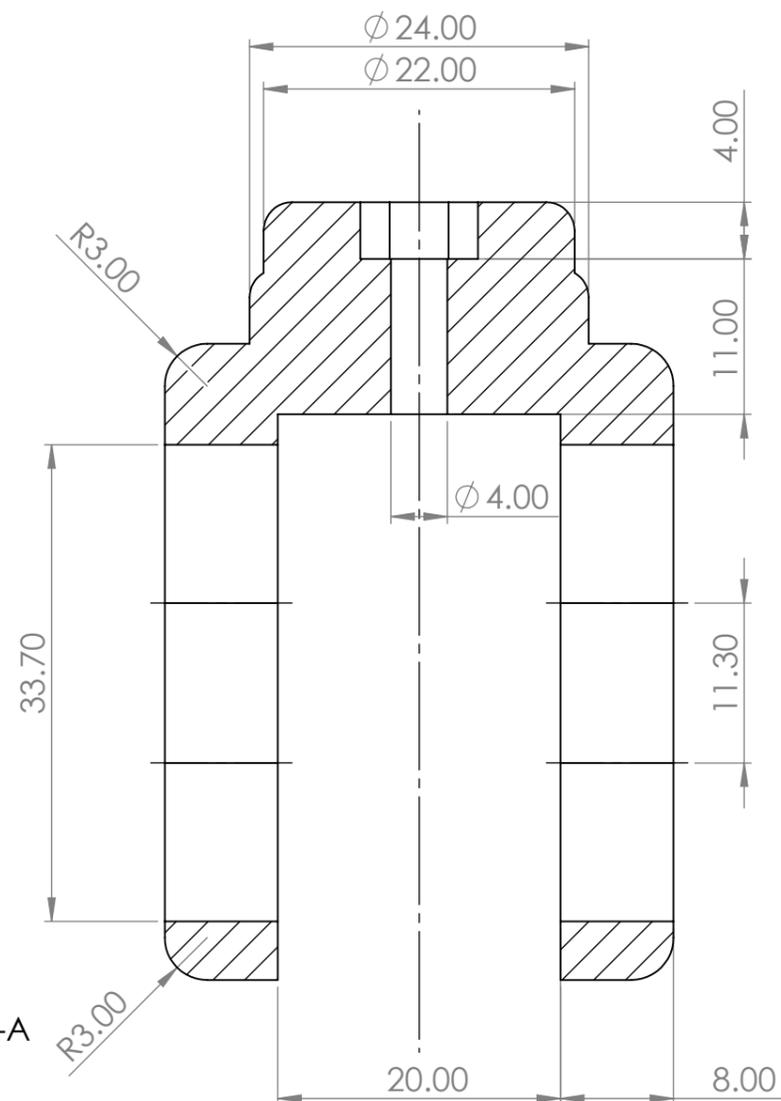
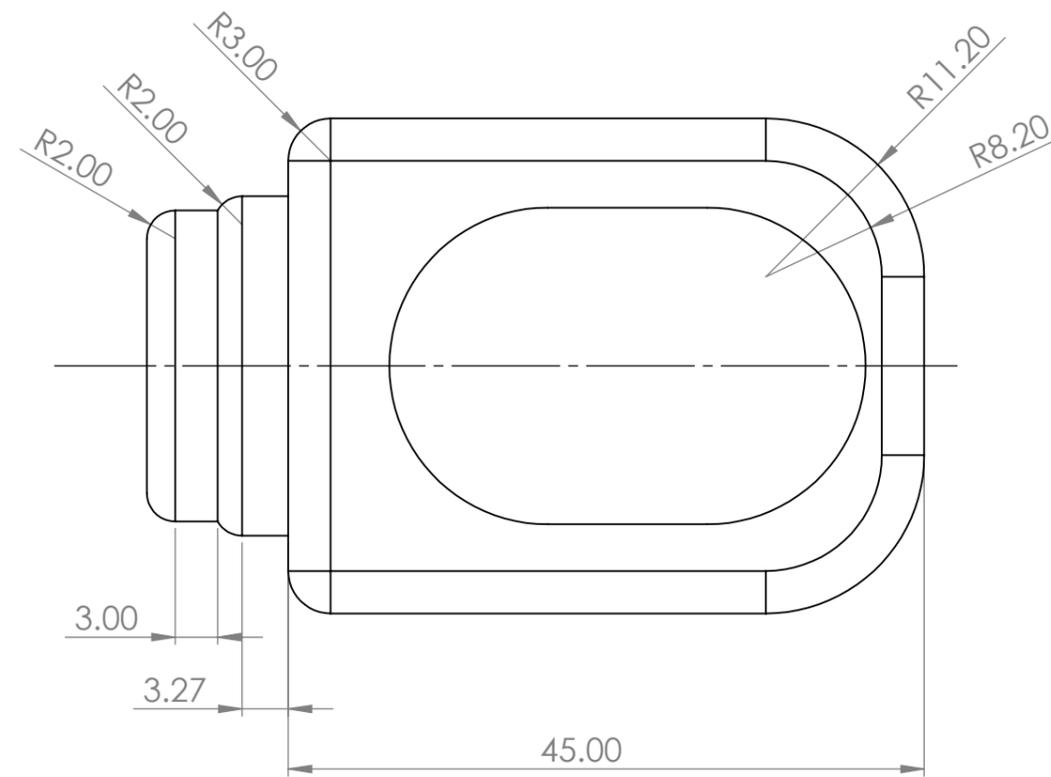
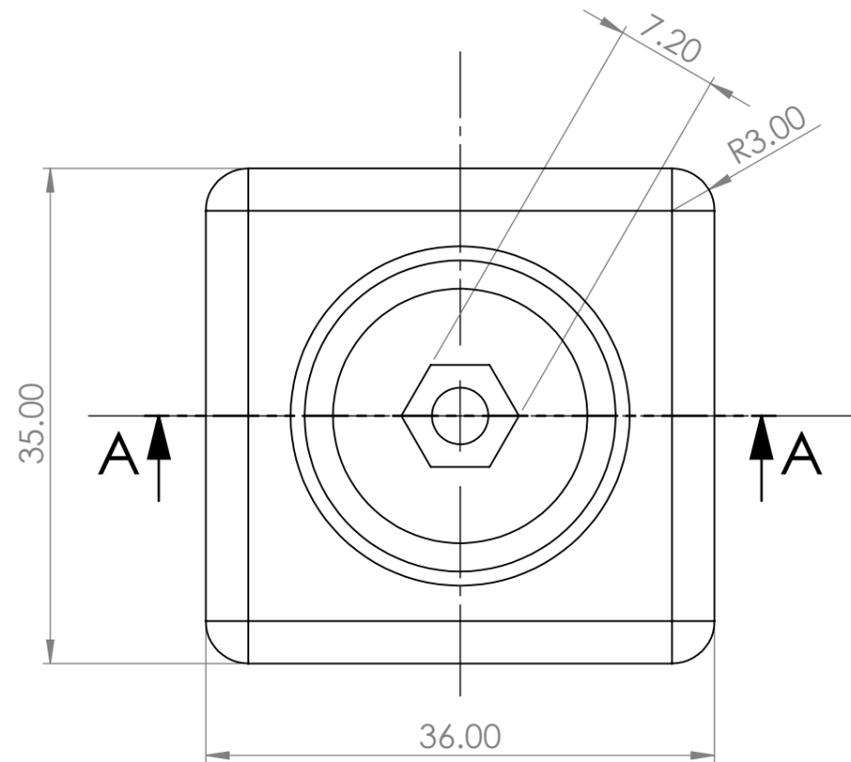
N.º DE DIBUJO

1-4.5

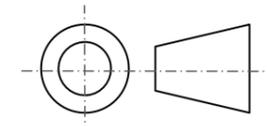
A4

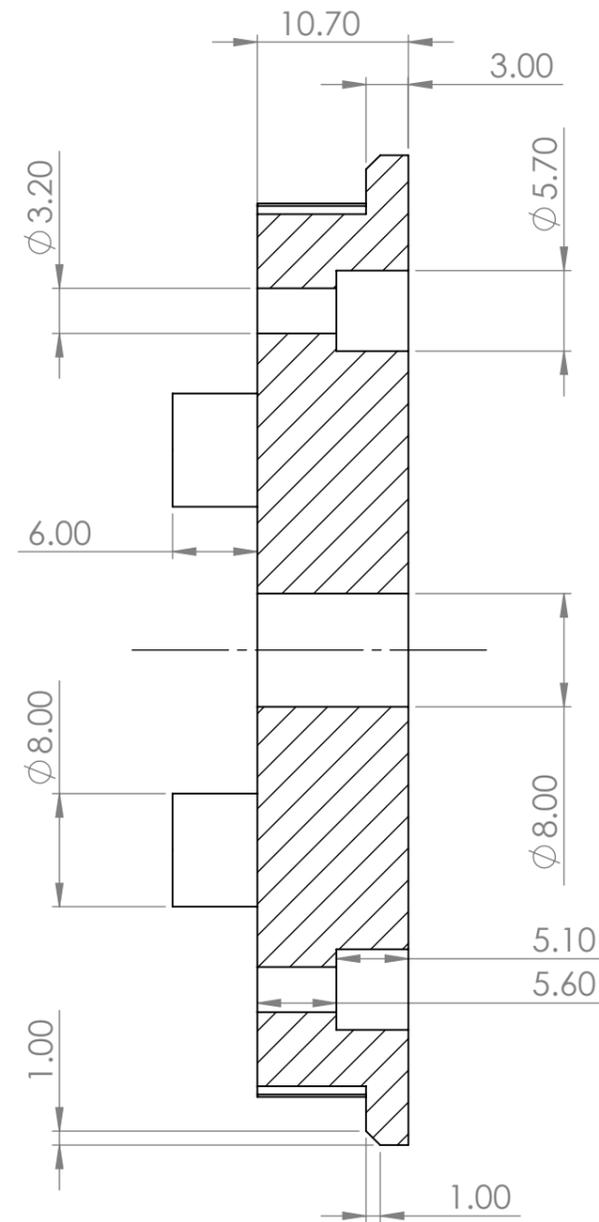
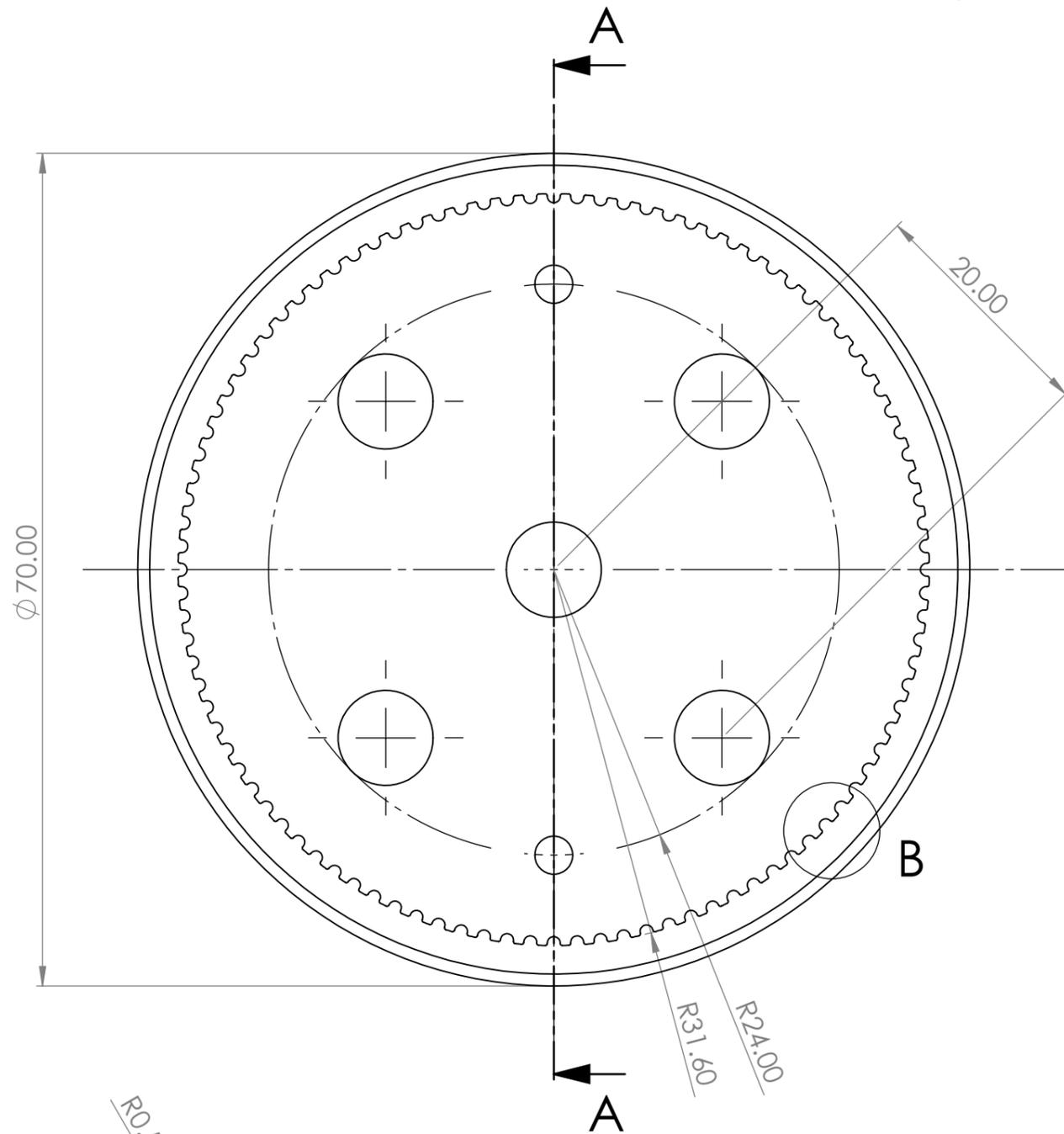
ESCALA:2:1

HOJA 1 DE 1

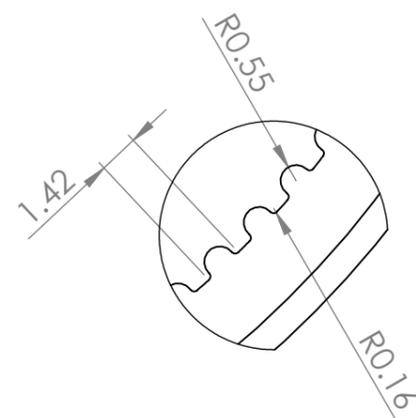


SECCIÓN A-A
ESCALA 2 : 1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				17/06/2021		 <p>Conexión Muñeca</p>			
VERIF.									
APROB.									
FABR.									
CALID.									
OBSERVACIONES				MATERIAL: Z-HIPS		N.º DE DIBUJO: 1-4.7		A3	
				PESO: 4.8g		ESCALA: 2:1		HOJA 1 DE 1	

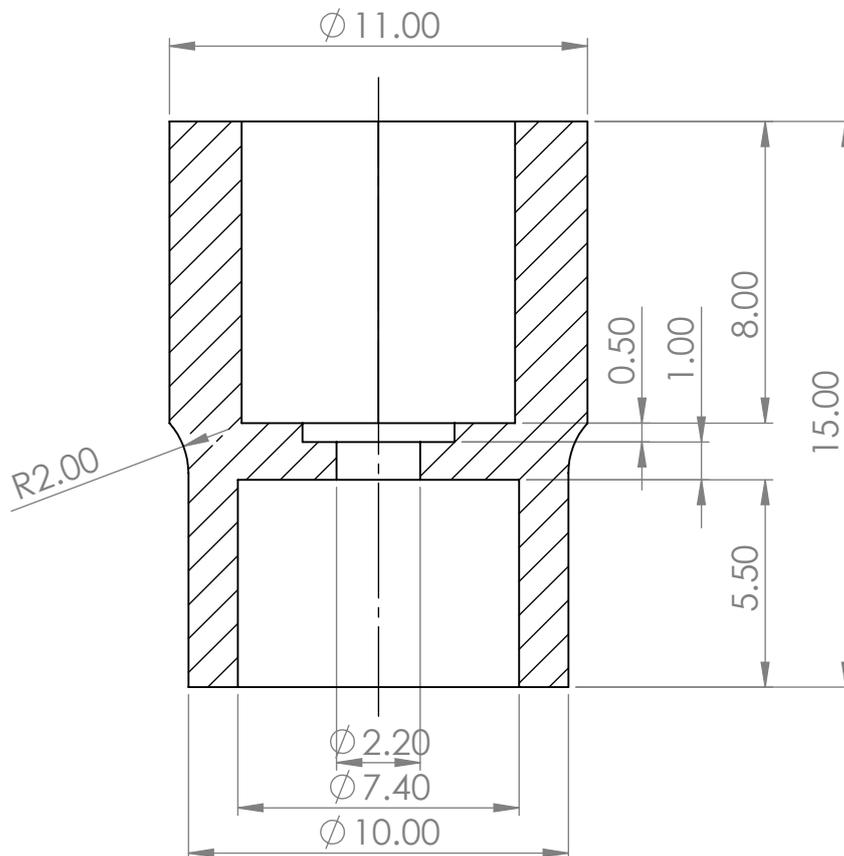
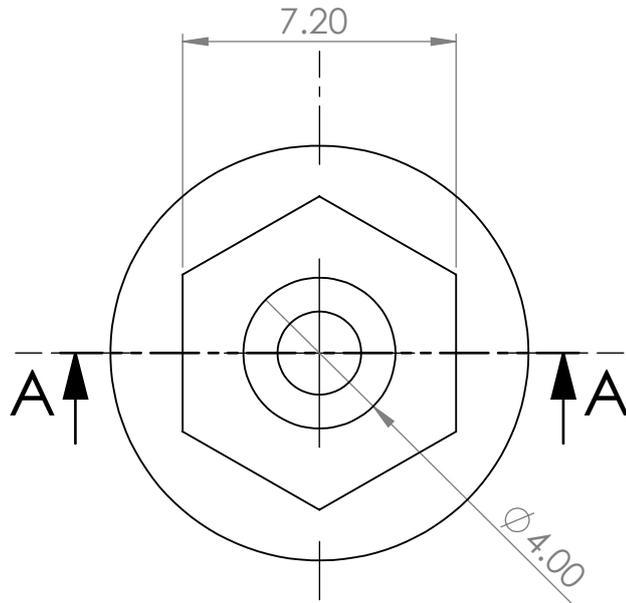


SECCIÓN A-A
ESCALA 2 : 1



DETALLE B
ESCALA 4 : 1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
DIBUJ. Diego Jarque Pérez		FIRMA		FECHA 17/06/2021		TÍTULO: Polea Antebrazo			
VERIF.						N.º DE DIBUJO 1-4.8		A3	
APROB.						MATERIAL: Z-HIPS		ESCALA:2:1	
FABR.						PESO: 8.2g		HOJA 1 DE 1	
CALID.									
OBSERVACIONES									



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO:
 · LAS COTAS SE EXPRESAN EN MM
 · TOLERANCIAS: +0,1 mm
 · CANTIDAD: 1

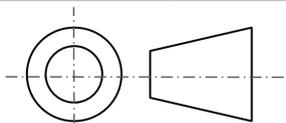
ACABADO:

REBARBAR Y ROMPER ARISTAS VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		17/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			
OBSERVACIONES:			



TÍTULO:

Acople Eje Secundario

MATERIAL:

Z-HIPS

N.º DE DIBUJO

1-4.11

A4

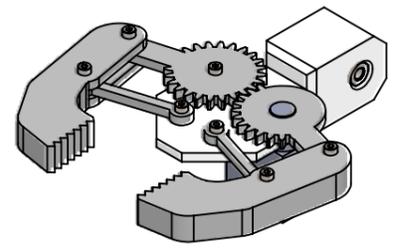
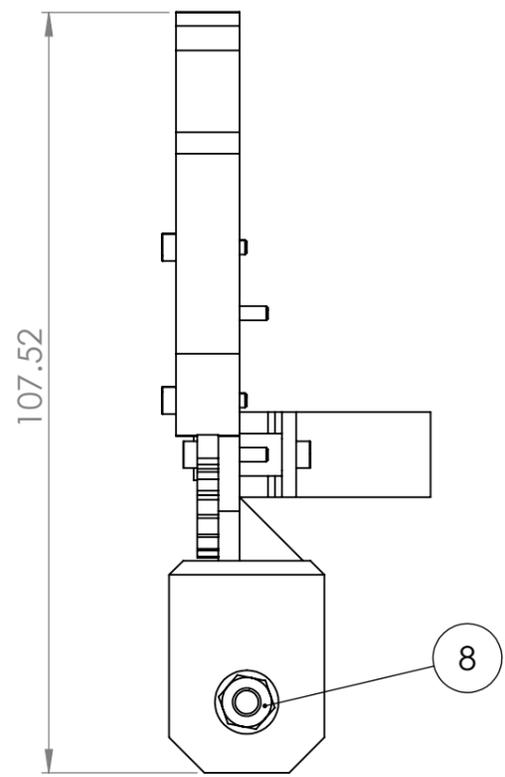
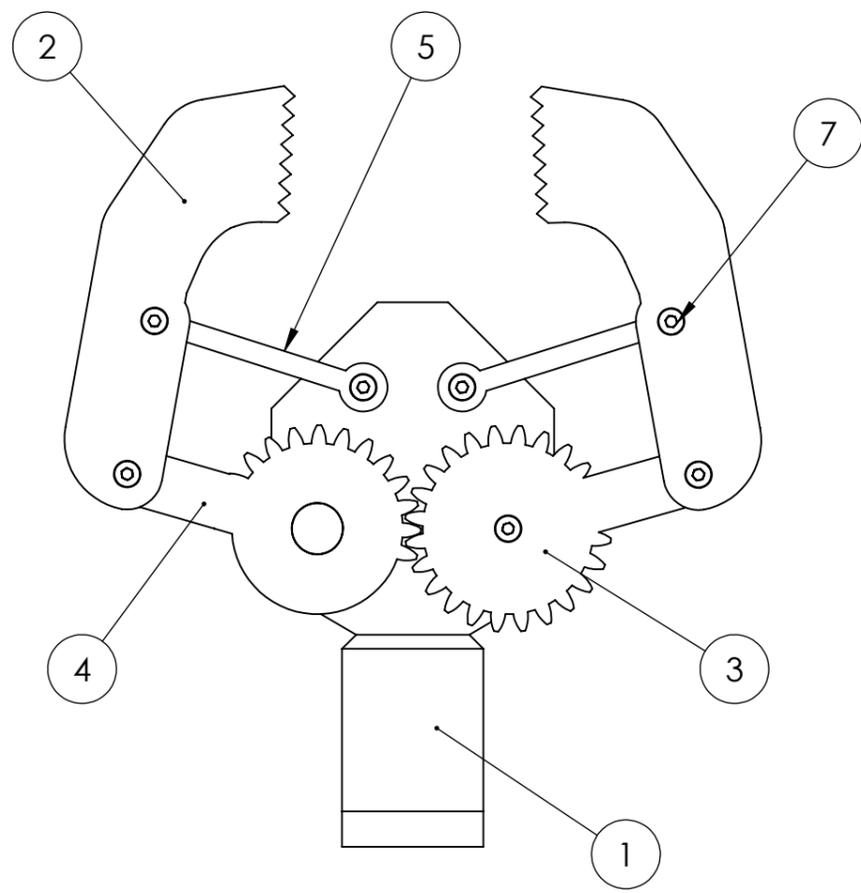
PESO: 0,2g

ESCALA:5:1

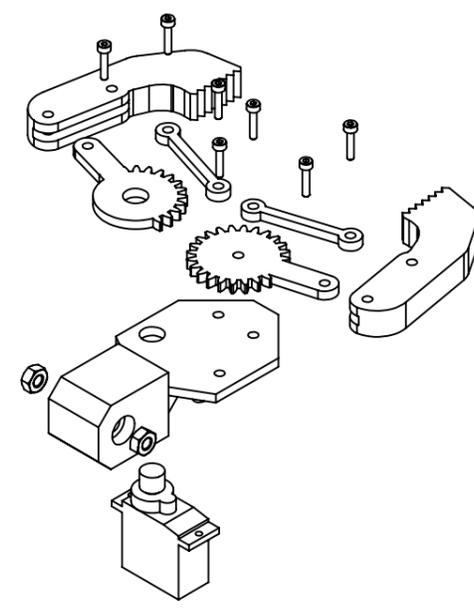
HOJA 1 DE 1

8 7 6 5 4 3 2 1

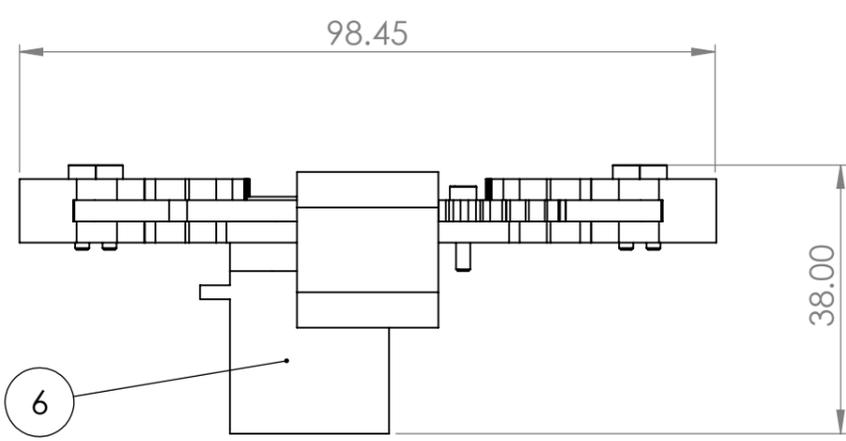
F
E
D
C
B
A



ESCALA 1:2

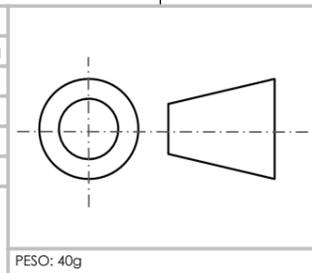


ESCALA 1:2



N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Soporte Pinza		1
2	Pinza		2
3	Engranaje 1	Engranaje conducido	1
4	Engranaje 2	Engranaje conductor	1
5	Fijador Pinza		2
6	Microservomotor	Modelo MG90D	1
7	Tornillo M2 x 10	ISO 4762 M2 x 10 - 10N	8
8	Tuerca M4	ISO - 4032 - M4 - D - N	2

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		15/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			
OBSERVACIONES			



TÍTULO:
Ensamblaje Pinza

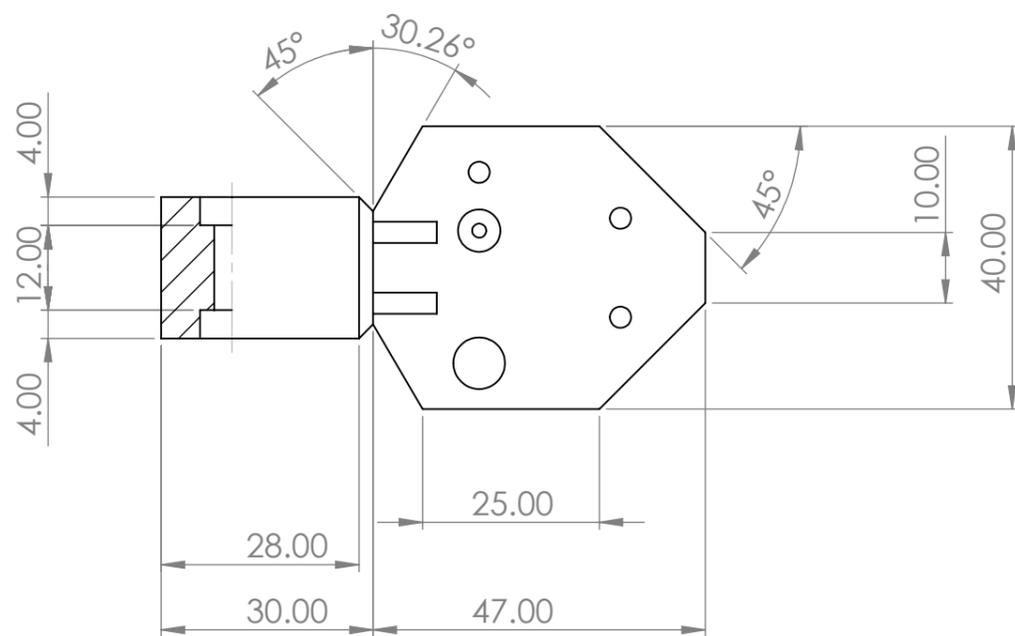
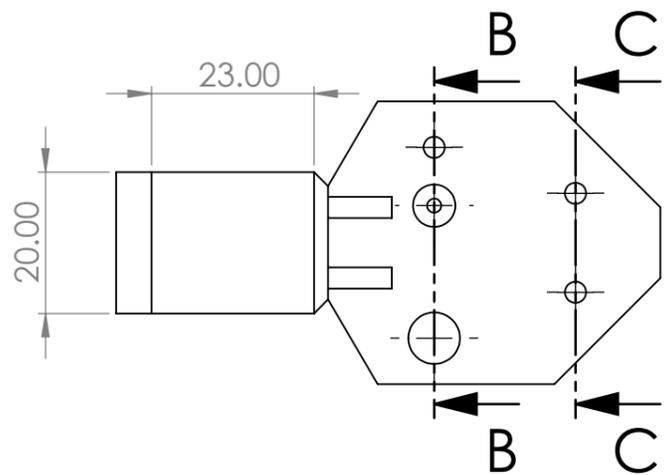
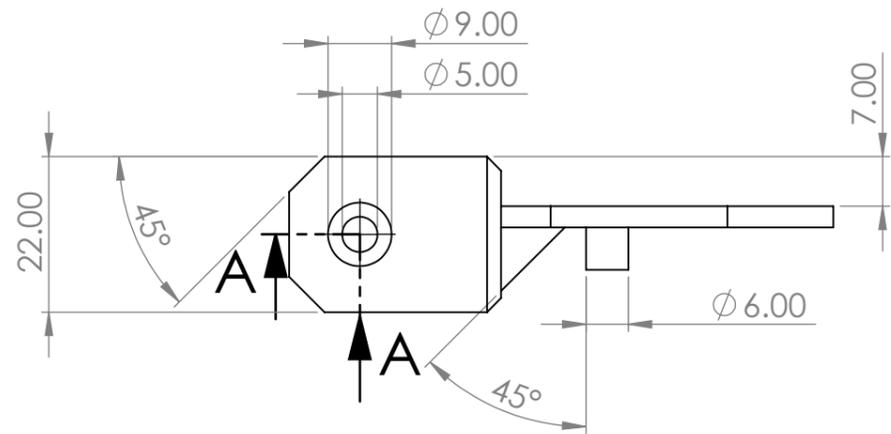
N.º DE DIBUJO
1-5 Plano de conjunto

ESCALA:1:1

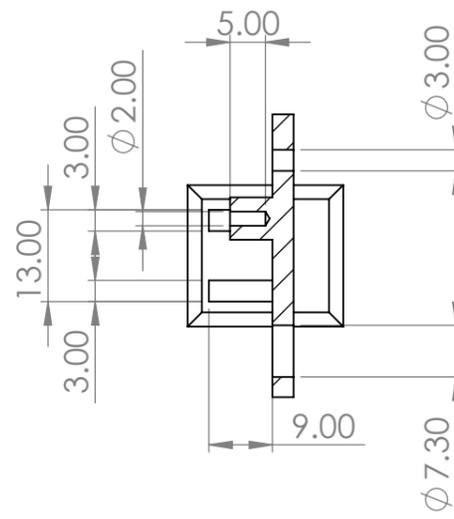
HOJA 1 DE 1

PESO: 40g

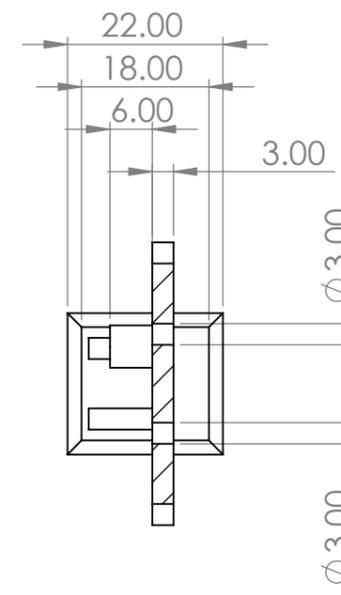
8 7 6 5 4 3 2 1



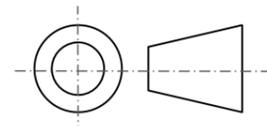
SECCIÓN A-A

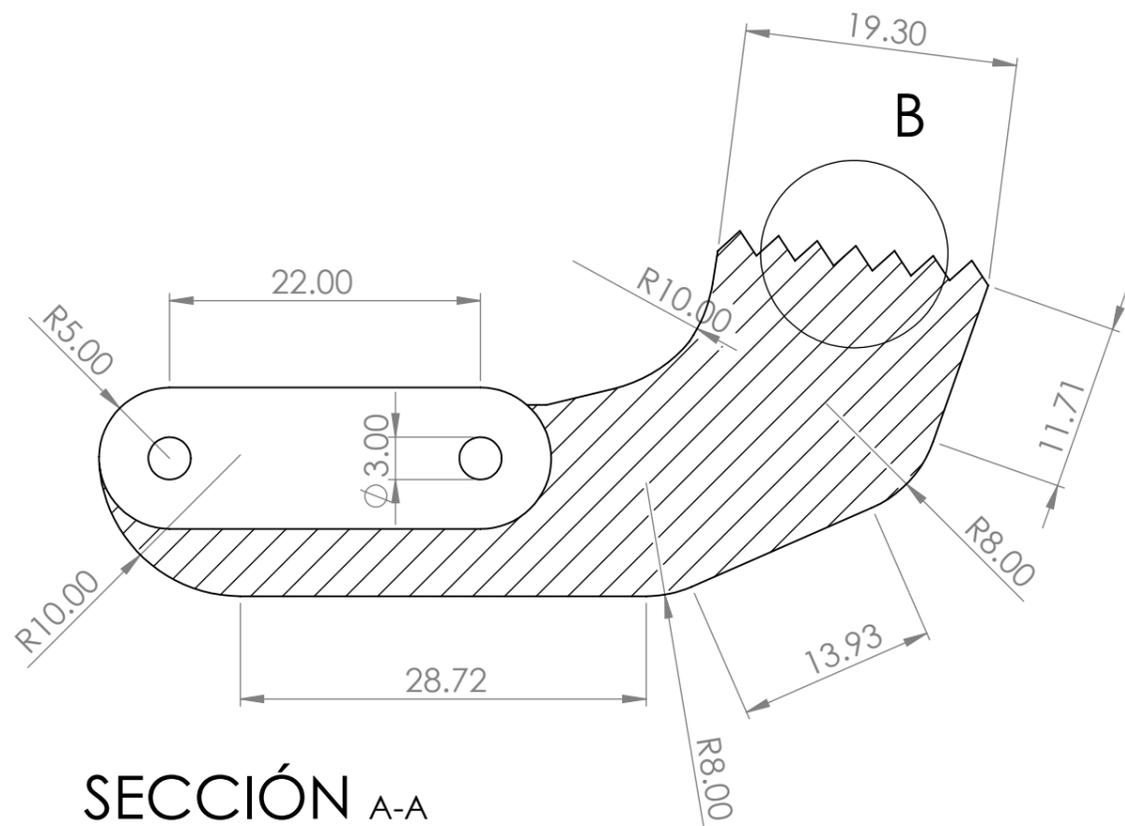
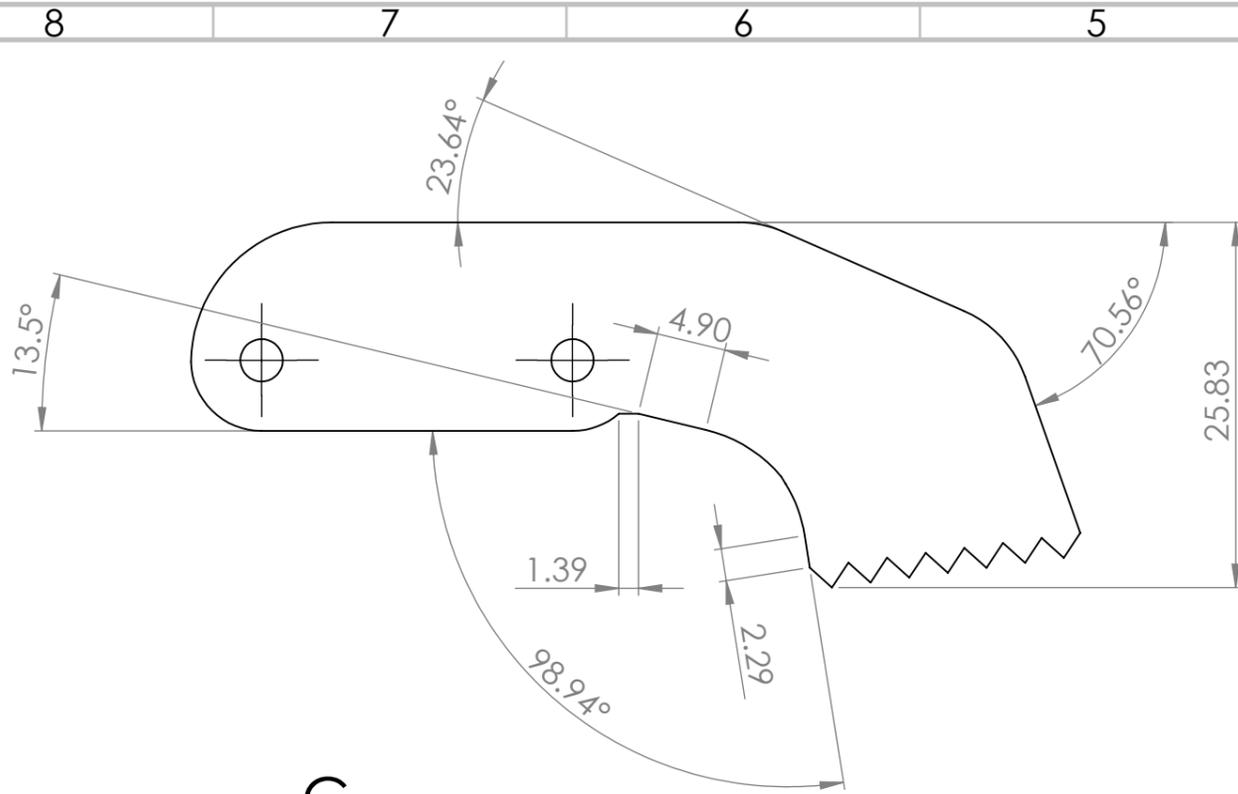


SECCIÓN B-B

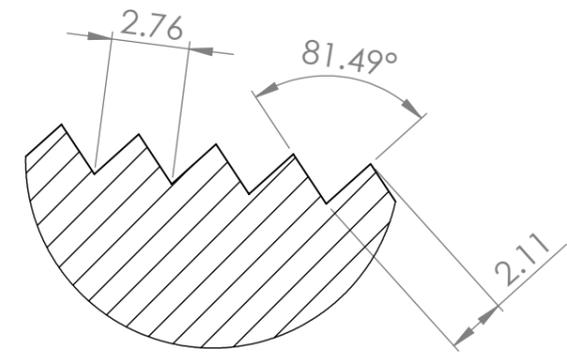


SECCIÓN C-C

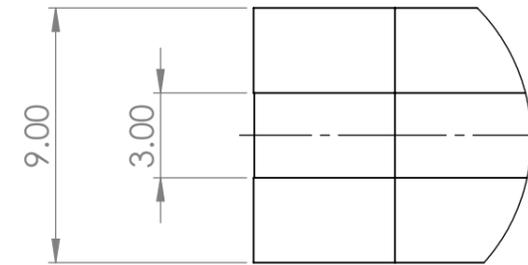
SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				15/06/2021		 <h1 style="text-align: center;">Soporte Pinza</h1>			
VERIF.									
APROB.									
FABR.									
CALID.									
OBSERVACIONES				MATERIAL: Z-HIPS		N.º DE DIBUJO: 1-5.1		A3	
				PESO: 3.9		ESCALA: 1:1		HOJA 1 DE 1	



SECCIÓN A-A

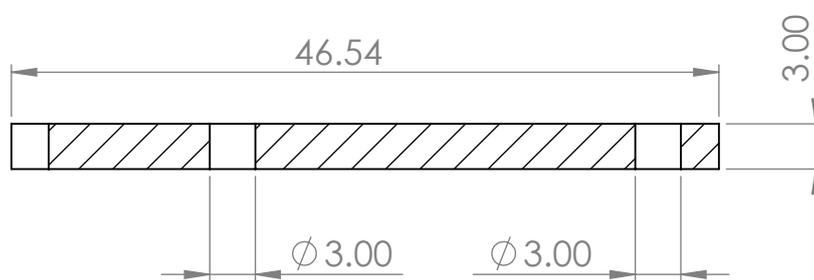
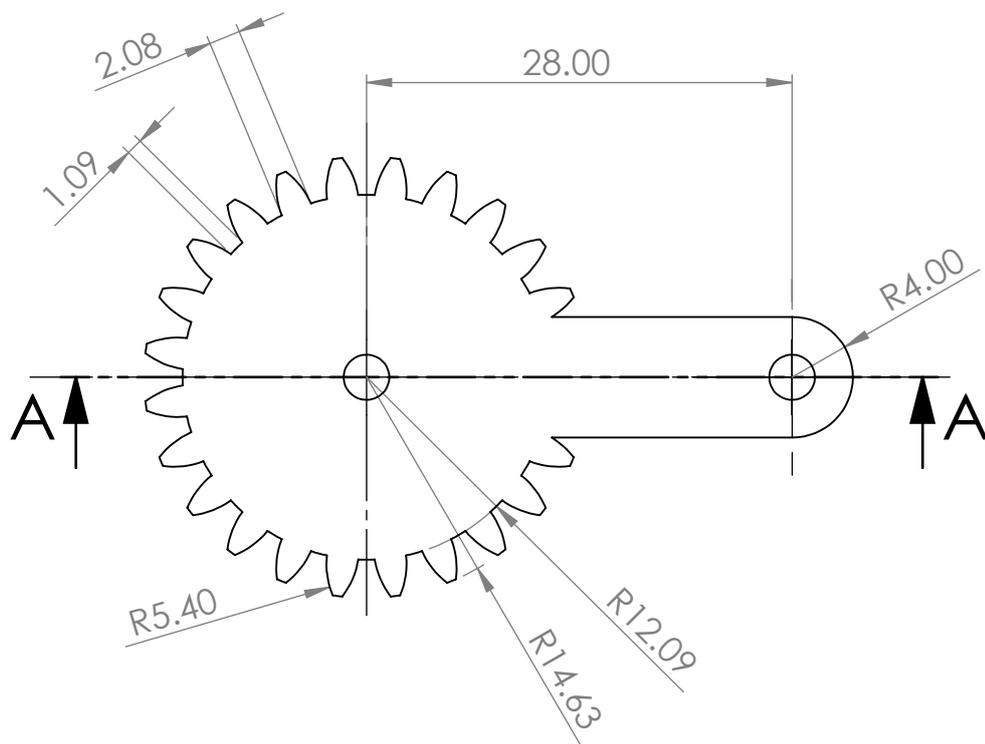


DETALLE B
ESCALA 4:1



DETALLE C
ESCALA 4:1

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: ± 0,1 mm · CANTIDAD: 2		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				17/06/2021		Pinza			
VERIF.						N.º DE DIBUJO		A3	
APROB.						MATERIAL:		ESCALA:2:1	
FABR.						Z-HIPS		HOJA 1 DE 1	
CALID.						PESO: 1,8g			
OBSERVACIONES									



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO:
 · LAS COTAS SE EXPRESAN EN MM
 · TOLERANCIAS: +0.05 mm
 · CANTIDAD: 1

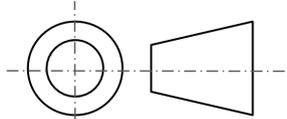
ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.	Diego Jarque Pérez		17/06/2021
VERIF.			
APROB.			
FABR.			
CALID.			



TÍTULO:

Engranaje 1

OBSERVACIONES:

MATERIAL:

PMMA-XT

N.º DE DIBUJO

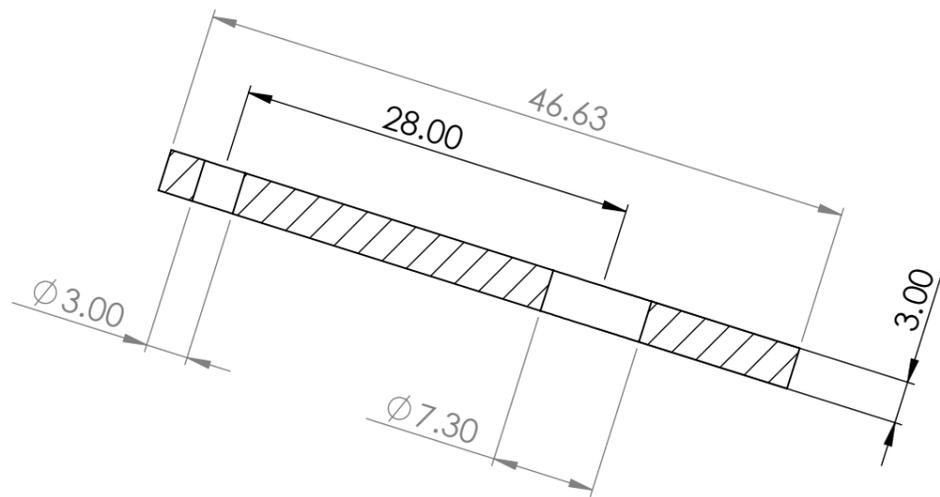
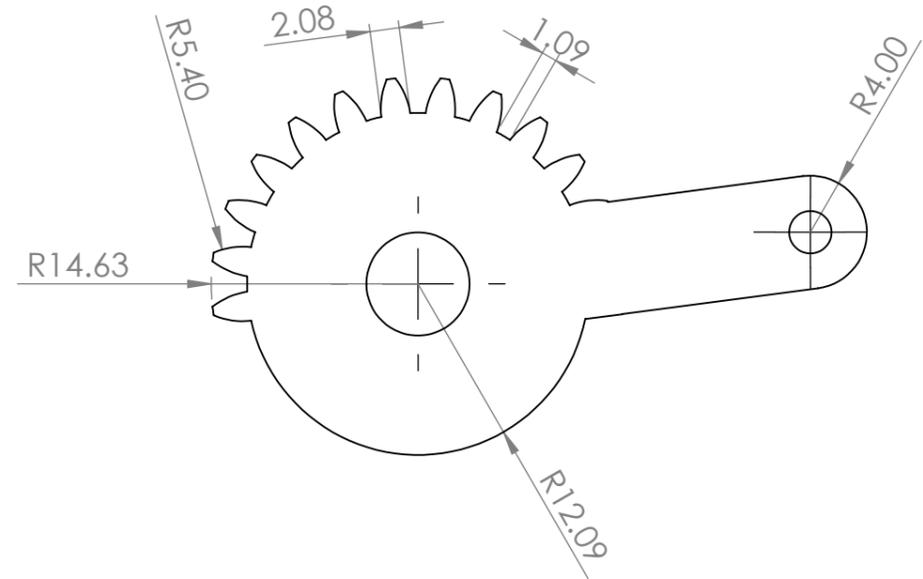
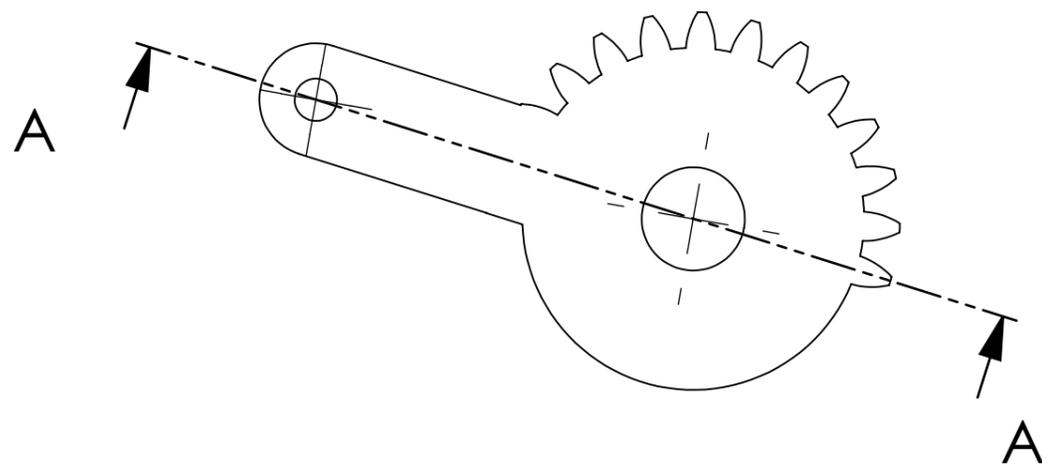
1-5.3

A4

PESO: 2,3g

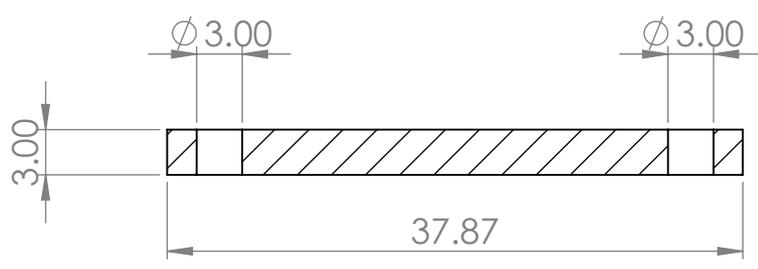
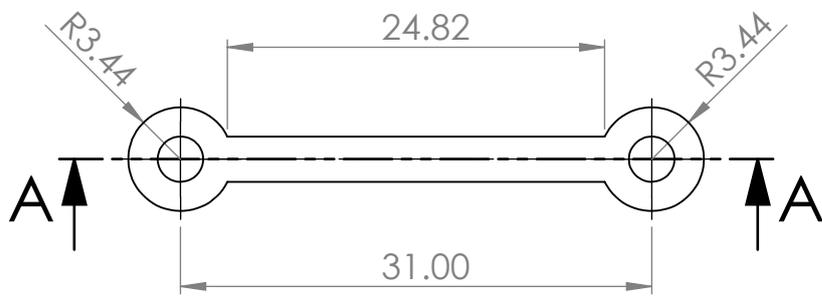
ESCALA:2:1

HOJA 1 DE 1

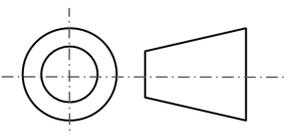


SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: $\pm 0,05$ mm · CANTIDAD: 1		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				17/06/2021		<h1 style="text-align: center;">Engranaje 2</h1>			
VERIF.									
APROB.									
FABR.									
CALID.						MATERIAL:		N.º DE DIBUJO	
OBSERVACIONES						PMMA-XT		1-5.4	
						PESO: 2,1g		ESCALA:2:1	
								HOJA 1 DE 1	



SECCIÓN A-A

SI NO SE INDICA LO CONTRARIO: · LAS COTAS SE EXPRESAN EN MM · TOLERANCIAS: +0.05 mm · CANTIDAD: 2		ACABADO:		REBARBAR Y ROMPER ARISTAS VIVAS		NO CAMBIE LA ESCALA		REVISIÓN	
NOMBRE		FIRMA		FECHA		TÍTULO:			
DIBUJ. Diego Jarque Pérez				17/06/2021		 <p>Fijador Pinza</p>			
VERIF.									
APROB.									
FABR.									
CALID.						MATERIAL:		N.º DE DIBUJO	
OBSERVACIONES:						PMMA-XT		1-5.5	
						PESO: 0,5g		ESCALA:2:1	
								HOJA 1 DE 1	



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño
Universitat Politècnica de València

DOCUMENTO PLIEGO DE CONDICIONES TÉCNICAS

Índice documento pliego de condiciones técnicas

1	OBJETO	135
2	CONDICIONES DE LOS MATERIALES.....	135
2.1	ENSAMBLAJE BASE	135
2.2	ENSAMBLAJE HOMBRO.....	136
2.3	ENSAMBLAJE BRAZO	137
2.4	ENSAMBLAJE ANTEBRAZO.....	137
2.5	ENSAMBLAJE PINZA.....	138
3	CONDICIONES DE EJECUCIÓN.....	139
3.1	ENSAMBLAJE BASE	139
3.2	ENSAMBLAJE HOMBRO.....	139
3.3	ENSAMBLAJE BRAZO	140
3.4	ENSAMBLAJE ANTEBRAZO.....	140
3.5	ENSAMBLAJE PINZA.....	141
3.6	MONTAJE	141
4	PRUEBAS Y AJUSTES FINALES DE SERVICIO	145

1 OBJETO

El objeto de este pliego de condiciones se refiere a la fabricación y montaje del ensamblaje de del modelo de brazo robot de 5gdl diseñado por Diego Jarque Pérez para el Trabajo de Final de Grado del Grado en Ingeniería Mecánica en la Universidad Politécnica de Valencia.

El ámbito de aplicación de este documento se extiende a la fabricación o compra de todos los sistemas mecánicos, electrónicos y piezas necesarias para completar el ensamblaje del brazo robot. En determinados supuestos se podrán adoptar, por la propia naturaleza del mismo o del desarrollo tecnológico, soluciones diferentes a las exigidas en este documento, siempre que quede suficientemente justificada su necesidad y que no impliquen una disminución de las exigencias mínimas de diseño especificadas en el mismo.

2 CONDICIONES DE LOS MATERIALES

2.1 ENSAMBLAJE BASE

Descripción

Conformado por Base Inferior, Base Superior y Tronco de la Base.

La Base Inferior tiene unas dimensiones generales de 240x180x20mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 62g.

La Base Superior tiene unas dimensiones generales de 205x80x12mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 27g.

El Tronco de la Base tiene unas dimensiones generales de Ø80x48mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 23g.

Las dimensiones específicas de estas piezas se encuentran detalladas en los planos 1-1.1, 1-1.2 y 1-1.3 del **Documento Planos**.

La Base cuenta con un motor NEMA 17, modelo 17HS16-2004S1, de dimensiones generales 42x42x40mm y Ø5x23mm de eje, 310g de peso y alimentación de 2,2V. Su par de rotación es de 45Ncm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

Al motor estará unida una polea dentada GT2 de 6mm de ancho y 5mm de diámetro de eje.

Control de calidad

Se realizará una medición general de las piezas una vez fabricadas, haciendo especial detenimiento en orificios para tornillería o encaje de piezas. En caso de que la pieza no cumpla con la tolerancia especificada en el plano, debe descartarse y fabricarse de nuevo. En el caso en el que alguno de los orificios no cumpla con las dimensiones se lijará hasta que lo haga, siempre y cuando esto no comprometa la estructura o el funcionamiento correcto de la pieza. También se realizará una inspección visual en busca de grietas o marcas que puedan comprometer el funcionamiento del ensamblaje.

2.2 ENSAMBLAJE HOMBRO

Descripción

Conformado por Base Hombro, Polea Hombro, Separador Polea Hombro, Cobertura Hombro, Soporte Motor y Adaptador Polea.

La Base Hombro tiene unas dimensiones generales de 168x50x80mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 47g.

La Polea Hombro tiene unas dimensiones generales de $\varnothing 70 \times 10$ mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 6g.

El Separador Polea Hombro tiene unas dimensiones generales de $\varnothing 70 \times 2$ mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 1,6g.

La Cobertura Hombro tiene unas dimensiones generales de 116x22x60mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 9,4g.

El Soporte Motor tiene unas dimensiones generales de 45x45x15mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 5,6g.

El Adaptador Polea tiene unas dimensiones generales de $\varnothing 14 \times 41$ mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 0,6g.

Las dimensiones específicas de estas piezas se encuentran detalladas en los planos 1-2.1, 1-2.2, 1-2.3, 1-2.4, 1-2.5 y 1-2.9 del **Documento Planos**.

El Hombro cuenta con un motor NEMA 17 con tren de engranajes 5:1, modelo 17HS15-1684S-PG5, de dimensiones generales 42x42x40mm, $\varnothing 28 \times 27$ mm de tren de engranajes y $\varnothing 8 \times 20$ mm de eje, 460g de peso y alimentación de 2,8V. Su par de rotación es de 4Nm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

Al Adaptador Polea estará unida una polea HTD 3M16T de 16 dientes, 11mm de ancho y 6mm de diámetro de eje.

Control de calidad

Se realizará una medición general de las piezas una vez fabricadas, haciendo especial detenimiento en orificios para tornillería o encaje de piezas. En caso de que la pieza no cumpla con la tolerancia especificada en el plano, debe descartarse y fabricarse de nuevo. En el caso en el que alguno de los orificios no cumpla con las dimensiones se lijará hasta que lo haga, siempre y cuando esto no comprometa la estructura o el funcionamiento correcto de la pieza. También se realizará una inspección visual en busca de grietas o marcas que puedan comprometer el funcionamiento del ensamblaje

2.3 ENSAMBLAJE BRAZO

Descripción

Conformado por Base Brazo, Polea Brazo, Separador Polea Brazo, Cobertura Brazo y Separador.

La Base Brazo tiene unas dimensiones generales de 210x60x30mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 37g.

La Polea Brazo tiene unas dimensiones generales de Ø80x12mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 12g.

El Separador Polea Brazo tiene unas dimensiones generales de Ø80x2mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 1,7g.

La Cobertura Brazo tiene unas dimensiones generales de 90x62x12mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 4,3g.

El Separador tiene unas dimensiones generales de 90x62x2mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 2,4g.

Las dimensiones específicas de estas piezas se encuentran detalladas en los planos 1-3.1, 1-3.2, 1-3.3, 1-3.4, 1-3.5 del **Documento Planos**.

El Brazo cuenta con un motor NEMA 17, modelo 17HS16-2004S1, de dimensiones generales 42x42x40mm y Ø5x23mm de eje, 310g de peso y alimentación de 2,2V. Su par de rotación es de 45Ncm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

Al motor estará unida una polea dentada GT2 de 6mm de ancho y 5mm de diámetro de eje.

Para la transmisión se emplearán dos poleas 16T GT2 no dentadas de 6mm de ancho y 3mm de diámetro de eje.

Control de calidad

Se realizará una medición general de las piezas una vez fabricadas, haciendo especial detenimiento en orificios para tornillería o encaje de piezas. En caso de que la pieza no cumpla con la tolerancia especificada en el plano, debe descartarse y fabricarse de nuevo. En el caso en el que alguno de los orificios no cumpla con las dimensiones se lijará hasta que lo haga, siempre y cuando esto no comprometa la estructura o el funcionamiento correcto de la pieza. También se realizará una inspección visual en busca de grietas o marcas que puedan comprometer el funcionamiento del ensamblaje

2.4 ENSAMBLAJE ANTEBRAZO

Descripción

Conformado por Antebrazo Principal, Acople Eje Principal, Adaptador Motor, Conexión Muñeca, Polea Antebrazo, Acople eje Secundario.

El Antebrazo Principal tiene unas dimensiones generales de 167x35x76mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 35g.

El Acople Eje Principal tiene unas dimensiones generales de Ø26x20mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 0,6g.

El Adaptador Motor tiene unas dimensiones generales de 34x22x32mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 1,1g.

La Conexión Muñeca tiene unas dimensiones generales de 55x35x36mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 4,8g.

La Polea Antebrazo tiene unas dimensiones generales de Ø70x11mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 8,2g.

El Acople Eje Secundario tiene unas dimensiones generales de Ø11x15mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 0,2g.

Las dimensiones específicas de estas piezas se encuentran detalladas en los planos 1-4.1, 1-4.3, 1-4.5, 1-4.7, 1-4.8 y 1-4.11 del **Documento Planos**.

El Antebrazo cuenta con un servomotor modelo DS04-NFC, de dimensiones generales 54x50x20mm, 50g de peso y alimentación de 4,8V. Su par de rotación es de 54Ncm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

También cuenta con un servomotor modelo MG90D, de dimensiones generales 23x12x29mm, 13g de peso y alimentación 4,8V. Su par de rotación es de 21Ncm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

Control de calidad

Se realizará una medición general de las piezas una vez fabricadas, haciendo especial detenimiento en orificios para tornillería o encaje de piezas. En caso de que la pieza no cumpla con la tolerancia especificada en el plano, debe descartarse y fabricarse de nuevo. En el caso en el que alguno de los orificios no cumpla con las dimensiones se lijará hasta que lo haga, siempre y cuando esto no comprometa la estructura o el funcionamiento correcto de la pieza. También se realizará una inspección visual en busca de grietas o marcas que puedan comprometer el funcionamiento del ensamblaje

2.5 ENSAMBLAJE PINZA

Descripción

Conformado por Soporte Pinza, Pinza, Engranaje 1, Engranaje 2 y Fijador Pinza.

El Soporte Pinza tiene unas dimensiones generales de 77x40x22mm, fabricado mediante poliestireno de alto impacto Z-HIPS y con un peso de 3,9g.

La Pinza tiene unas dimensiones generales de 63x26x9mm, fabricada mediante poliestireno de alto impacto Z-HIPS y con un peso de 1,8g.

El Engranaje 1 tiene unas dimensiones generales de 47x29x3mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 2,3g.

El Engranaje 2 tiene unas dimensiones generales de 47x29x3mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 2,1g.

El Fijador Pinza tiene unas dimensiones generales de 38x7x3mm, fabricado mediante metacrilato de extrusión PMMA-XT y con un peso de 0,5g.

Las dimensiones específicas de estas piezas se encuentran detalladas en los planos 1-5.1, 1-5.2, 1-5.3, 1-5.4 y 1-5.5 del **Documento Planos**.

La Pinza cuenta con un servomotor modelo MG90D, de dimensiones generales 23x12x29mm, 13g de peso y alimentación 4,8V. Su par de rotación es de 21Ncm. Este motor puede ser sustituido por un modelo similar siempre y cuando cumpla con las dimensiones y potencia especificadas.

Control de calidad

Se realizará una medición general de las piezas una vez fabricadas, haciendo especial detenimiento en orificios para tornillería o encaje de piezas. En caso de que la pieza no cumpla con la tolerancia especificada en el plano, debe descartarse y fabricarse de nuevo. En el caso en el que alguno de los orificios no cumpla con las dimensiones se lijará hasta que lo haga, siempre y cuando esto no comprometa la estructura o el funcionamiento correcto de la pieza. También se realizará una inspección visual en busca de grietas o marcas que puedan comprometer el funcionamiento del ensamblaje

3 CONDICIONES DE EJECUCIÓN

3.1 ENSAMBLAJE BASE

Se fabricarán mediante impresión 3D, y con un 30% de relleno, 1 Base Inferior, 1 Base Superior y 1 Tronco de la Base. La tolerancia de impresión no debe superar los 0,1mm y ha de realizarse la impresión en un lugar con temperatura media estable y sin corriente, para evitar posibles anomalías en las piezas durante la impresión.

El ensamblaje se realizará mediante tornillos Allen M8x70, un tornillo de cabeza hexagonal M8x80 y tuercas M8. El motor se unirá a la Base Superior mediante tornillos Allen M3x10. Antes de realizar el ensamblaje, se han de colocar 2 rodamientos rígidos de bolas obturados, modelo 6002-2RSH, y un rodamiento axial, modelo 51104 en las posiciones indicadas en el plano 1-1 Plano de Conjunto del **Documento Planos**.

3.2 ENSAMBLAJE HOMBRO

Se fabricarán mediante impresión 3D, y con un 30% de relleno, 1 Base Hombro, 1 Polea Hombro, 1 Cobertura Hombro y 1 Soporte Motor, y con un 0% de relleno, 1 Adaptador Polea. La tolerancia de impresión no debe superar los 0,1mm y ha de realizarse la impresión en un lugar con temperatura media estable y sin corriente, para evitar posibles anomalías en las piezas durante la impresión. Se fabricará mediante cortado láser 1 Separador Polea Hombro, con una tolerancia de 0,05mm.

El ensamblaje se realizará mediante tornillos Allen M4x35 y M4x20, un tornillo Allen M8x40, tuercas M4 y M8 y arandelas M8. El motor se unirá al Soporte Motor mediante tornillos Allen M3x10 antes de unir el Soporte Motor al resto del ensamblaje. Antes de unir la Cobertura Hombro al resto del ensamblaje, se deberá colocar la correa HTD 3M de 330mm de longitud en la polea dentada HTD 3M16T, así como dos rodamientos rígidos de bolas obturados, modelo 608-2RSH en la posición indicada en el plano 1-2 Plano de Conjunto del **Documento Planos**. La correa debe quedar por debajo de los rodamientos, de forma que la cara lisa de la correa coincida con la superficie de los rodamientos.

3.3 ENSAMBLAJE BRAZO

Se fabricarán mediante impresión 3D, y 30% de relleno, 1 Base Brazo, 1 Polea Brazo y 1 Cobertura Brazo. La tolerancia de impresión no debe superar los 0,1mm y ha de realizarse la impresión en un lugar con temperatura media estable y sin corriente, para evitar posibles anomalías en las piezas durante la impresión. Se fabricarán mediante corte láser 1 Separador Polea Brazo y 1 Separador, con una tolerancia de 0,05mm.

El ensamblaje se realizará mediante tornillos Allen M4x35 y M3x25, tuercas M4 y M3 y arandelas M3. El motor se unirá a la Base Brazo y al Separador mediante tornillos Allen M3x10. Antes de unir la Cobertura Brazo, se deberá colocar la correa GT2 de 280mm de longitud sobre la polea dentada GT2 16T, así como las dos poleas no dentadas GT2 en la posición indicada en el plano 1-3 Plano de Conjunto del **Documento Planos**. La correa debe pasar entre ambas poleas no dentadas, de forma que la cara lisa de la correa coincida con la superficie de las poleas. No se debe montar la Polea Brazo hasta realizar el ensamblaje general.

3.4 ENSAMBLAJE ANTEBRAZO

Se fabricarán mediante impresión 3D, y 30% de relleno, 1 Antebrazo Principal, 1 Acople Eje Principal, 2 Adaptador Motor, 1 Conexión Muñeca, 1 Polea Antebrazo, 1 Acople Eje Secundario. La tolerancia de impresión no debe superar los 0,1mm y ha de realizarse la impresión en un lugar con temperatura media estable y sin corriente, para evitar posibles anomalías en las piezas durante la impresión.

El ensamblaje se realizará mediante tornillos Allen M3x10, tuercas M4 y M3 y una varilla roscada M4. El servomotor modelo DS04-NFC irá unido al Antebrazo Principal mediante tornillos M2x10 (incluidos en el pack de compra), sin embargo, antes de montarlo, se deberá pegar mediante algún tipo de adhesivo fuerte para plásticos, la rueda del eje del servomotor con el Acople Eje Principal. Para unir el Acople Eje Principal a la varilla roscada de M4, se emplearán un par de tuercas M4, de forma que el auto apriete evite el deslizamiento del eje y giren ambos solidariamente. Para unir la Conexión Muñeca a la varilla roscada de M4, se emplearán 2 tuercas M4, posicionadas según se indica en el plano 1-4 Plano de Conjunto del **Documento Planos**, de forma que también roten simultáneamente. Antes de unir la Conexión Muñeca, se debe colocar un rodamiento rígido de bolas obturado, modelo 624-2RS1, en la posición indicada en el plano 1-4 Plano de Conjunto del **Documento Planos**.

El servomotor modelo MG90D también irá unido a una varilla roscada M4 con el Acople Eje Secundario mediante 2 tuercas M4 para lograr una rotación solidaria. No se debe montar el eje secundario hasta realizar el ensamblaje general.

3.5 ENSAMBLAJE PINZA

Se fabricarán mediante impresión 3D, y con un 30% de relleno, 1 Soporte Pinza y 2 Pinza. La tolerancia de impresión no debe superar los 0,1mm y ha de realizarse la impresión en un lugar con temperatura media estable y sin corriente, para evitar posibles anomalías en las piezas durante la impresión. Se fabricarán mediante corte láser 1 Engranaje 1, 1 Engranaje 2 y 2 Fijador Pinza, con una tolerancia de 0,05mm.

El ensamblaje se realizará mediante tornillos Allen M2x10 y tuercas M4. El servomotor modelo MG90D se unirá al Soporte Pinza mediante un tornillo M2x10 (incluido en el pack de compra). Tras montar el servomotor, se pegará mediante un adhesivo fuerte para plásticos al Engranaje 2, de forma que roten solidariamente (importante no pegar con el Soporte Pinza). Al montar el Engranaje 1, asegurarse de que ambos engranajes forman el mismo ángulo con el Soporte Pinza para el correcto funcionamiento de la pinza.

3.6 MONTAJE

Una vez montados los 5 subensamblajes, se procederá al montaje del ensamblaje general, empezando desde la Base y terminando con la Pinza. Para ello se introducirá un tubo de aluminio de 15mm de diámetro externo, 1mm de espesor y longitud 100mm en el orificio correspondiente de la Base, tal como se muestra en el plano 1-1 Plano de Conjunto del **Documento Planos**, y posteriormente se colocará la correa GT2 de 280mm de longitud sobre la Polea Hombro y se introducirá el Hombro en el tubo de aluminio, de forma que el conjunto quede como muestra el plano 1 Plano de Conjunto del **Documento Planos**. Tras esto, se alinearé la polea GT2 16T del NEMA 17 de la base con la Pola Hombro y se colocará la correa envolviendo a ambas. Seguidamente se tensará la correa moviendo el motor de la Base a una distancia adecuada, haciendo uso juego del enclave. No será necesario asegurar el tubo de aluminio debido a que el ajuste de apriete con los rodamientos 6002-2RSH evitará que este se desplace.

En segundo lugar, se montará el Brazo. Para ello se empleará un tubo de aluminio de 15mm de diámetro externo, 1mm de espesor y longitud 90mm. Se introducirá el tubo en el orificio del hombro, como se muestra en el plano 1-2 Plano de Conjunto del **Documento Planos**. Colocar la Polea Brazo en el rubo junto con el Separador Polea Brazo (aun sin ensamblar) en el tubo y pasar la correa HTD 3M por ella, de forma que ya estará conectada a ambas ruedas dentadas. A continuación, introducir el Brazo en el tubo y atornillar la Polea Brazo y el Separador Polea Brazo como muestra el plano 1-2 Plano de conjunto del **Documento Planos**. Por último, tensar la correa desplazando el Soporte Motor haciendo uso del juego del enclave. No será necesario asegurar el tubo de aluminio debido a que el ajuste de apriete con los rodamientos 6002-2RSH evitará que este se desplace.

En tercer lugar, se montará el Antebrazo. Para ello se introducirá un tornillo de cabeza hexagonal M8x80 en el orificio del Antebrazo, como se muestra en el plano 1-4 Plano de Conjunto del **Documento Planos**, y seguidamente se colocará el Antebrazo en posición como se muestra en el plano 1 Plano de Conjunto del **Documento Planos**. Al hacerlo se deberá hacer pasar la correa GT2 de 280mm por la Polea Antebrazo, de forma que ya estará unida a ambas rudas dentadas. Por último, tensar la correa haciendo uso del juego del enclave del motor del Brazo.

En cuarto lugar, colocar la Pinza haciendo coincidir el orificio del Soporte Base con los de los Adaptadores de los Motores. Tras esto, introducir el Eje Secundario por el orificio y atornillar el servomotor MG90D al Adaptador Motor mediante dos tornillos M2x10 (incluidos en el pack de compra).

Por último, se realizará el conexionado del ensamblaje con los componentes electrónicos siguiendo el esquema eléctrico del apartado 7 *Electrónica* del **documento Memoria**.

Para ello se comenzará conectando los motores paso a paso con los módulos L298n. Observando las fichas técnicas de los motores se deben identificar los cables correspondientes a las entradas A y A-, y B y B-, de forma que dichas parejas vayan conectadas en el mismo par de puertos a los costados del módulo como se muestra en la **Figura 53**.

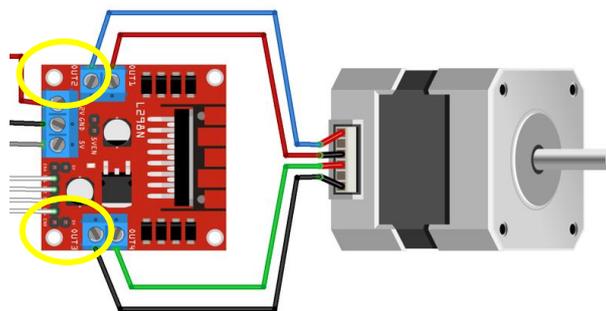


Figura 53. Conexión de un motor paso a paso con el módulo L298n, lugar de conexión redondeado en amarillo. Fuente: Elaboración propia.

Una vez conectados los motores paso a paso se procederá a conectar los módulos L298n con el microcontrolador Arduino UNO R3. En este caso debe tenerse en cuenta que los pines de conexión de la placa Arduino y los módulos coincidan con los pines definidos en el programa con el que se quiera trabajar, de forma que los motores reciban las órdenes de forma correcta.

En este caso se conectarán los 4 pines situados en la parte inferior del módulo con los pines I/O de la placa Arduino, y se repetirá este proceso para cada uno de los 3 módulos L298n. Debe prestarse atención al orden de conexión de los pines, ya que cada uno de estos está a su vez conectado a los puertos de conexión del motor, de forma que si el orden de conexión es incorrecto podría provocar que el motor no girase de la forma esperada por el usuario. La **Figura 54** muestra la conexión entre los L298n y el microcontrolador.

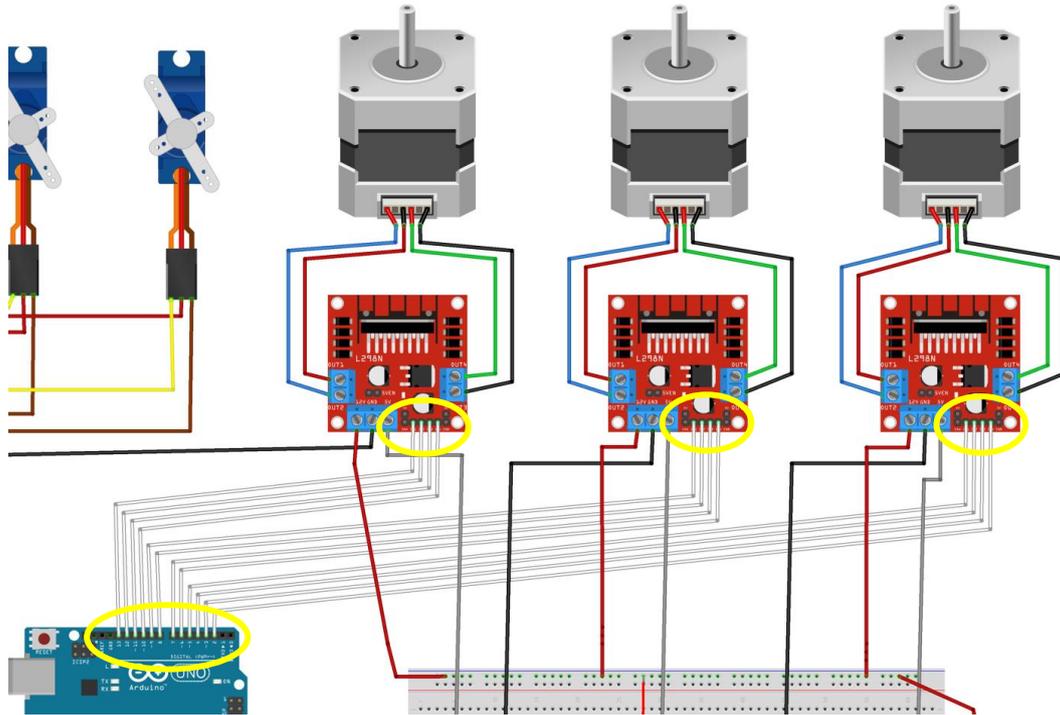


Figura 54. Esquema de conexionado entre los módulos L298n (dispositivos rojos) y la placa Arduino UNO (dispositivo azul), puntos de conexión redondeados en amarillo. Fuente: Elaboración propia.

Tras esto, se conectarán los servomotores al Arduino UNO. Esto podrá hacerse de forma directa, sin necesidad de módulos, consultando la ficha técnica del servomotor y verificando el cable de transmisión de datos y conectándolo al pin I/O del microcontrolador verificando nuevamente que el programa y el conexionado tengan el mismo pin. Puede observarse esto último en la **Figura 55**.

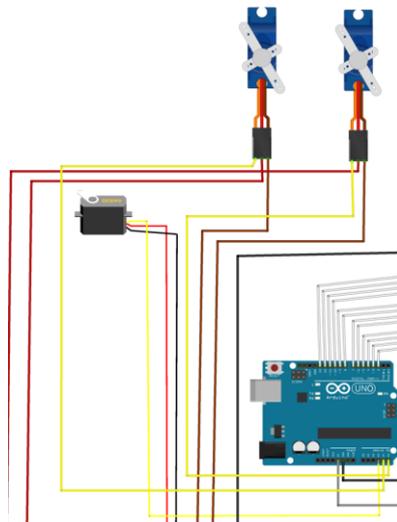


Figura 55. Conexión entre servomotores y Arduino UNO (cable amarillo). Fuente: Elaboración propia.

Seguidamente se conectarán los módulos L298 y los microservos (muñeca y pinza) a la alimentación de 5V de la placa Arduino. Para facilitar esto, se hará uso de una placa de conexiones, tal como se muestra en la **Figura 56**.

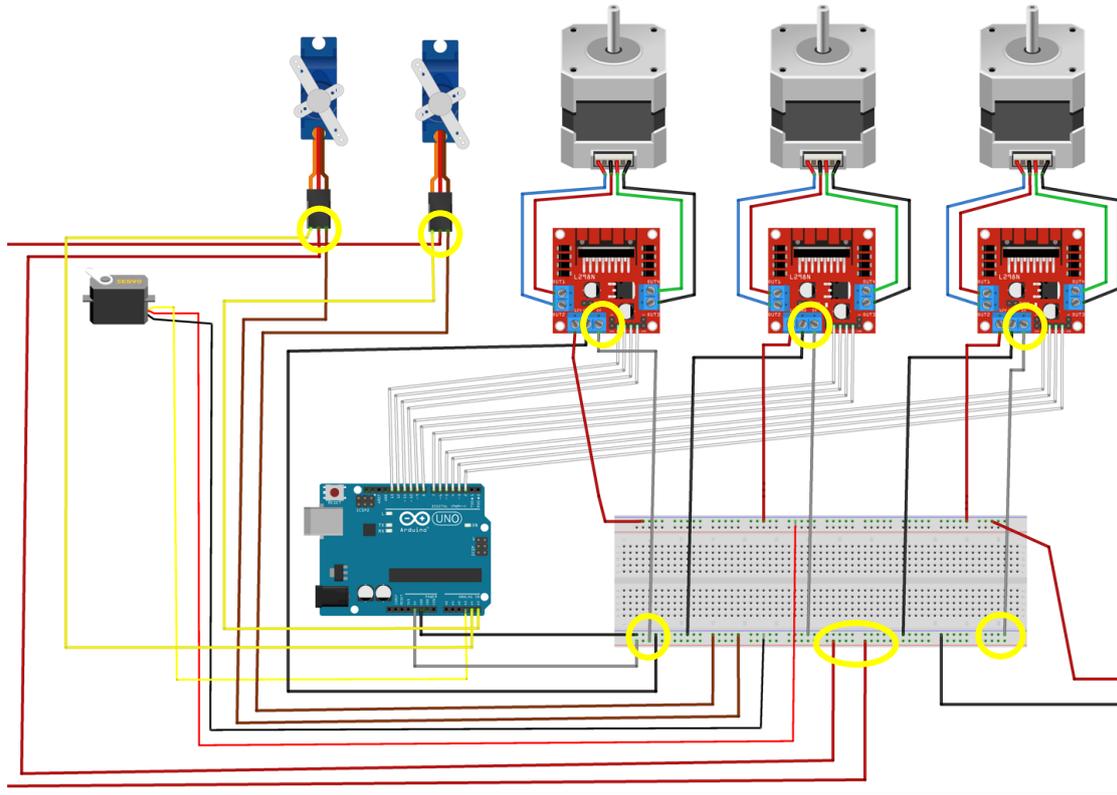


Figura 56. Conexiones entre alimentación de 5V de Arduino (cable gris), microservos (cables rojos) y módulos L298n (cables grises), puntos de conexión redondeados en amarillo. Fuente: Elaboración propia.

Finalmente, se conectarán las conexiones a tierra de los servomotores, los módulos L298n, la fuente de alimentación y Arduino UNO R3, y la alimentación de la fuente de alimentación con los módulos L298n y el servomotor de rotación continua. Se muestra esto en la **Figura 57**.

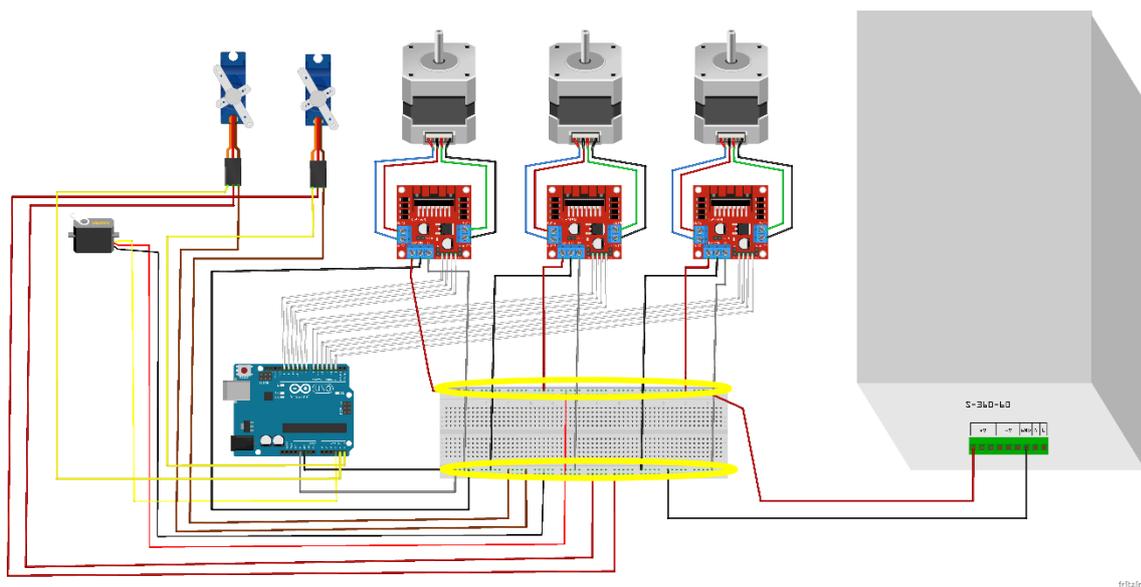


Figura 57. Esquema de conexionado de tierra representado con los cables negros y conexionado de la alimentación de los motores con los cables rojos. Zonas de conexión con la placa de conexionado redondeadas en amarillo. Fuente: Elaboración propia.

4 PRUEBAS Y AJUSTES FINALES DE SERVICIO

Una vez finalizado el ensamblaje del brazo robot y el conexionado necesario, se realizarán una serie de pruebas para verificar el correcto funcionamiento del sistema.

Para la verificación de la estructura del ensamblaje se realizarán las comprobaciones definidas en la **Tabla 6**.

COMPROBACIONES ESTRUCTURALES
Confirmar apriete de los tornillos y tuercas del ensamblaje
Revisar los planos y el inventario de materiales y verificar que todos los componentes necesarios han sido colocados
Inspección visual del ensamblaje en busca de grietas o defectos superficiales que puedan provocar fallos durante el uso del brazo
Verificar el correcto tensado de los sistemas de transmisión por correas para evitar deslizamientos
Colocar el cableado de los motores de forma que no interfiera con el giro de las articulaciones

Tabla 6. Lista de comprobaciones previas al uso del brazo robot, referentes a la estructura. Fuente: *Elaboración propia.*

Con respecto a la verificación de la electrónica, en primer lugar, se comprobará que todo el cableado está correctamente unido, se repasará el conexionado y se repasarán todas las uniones entre cables y dispositivos para comprobar que son firmes y no se separarán durante el uso.

En segundo lugar, se realizarán pruebas individuales a cada uno de los motores para verificar que funcionan correctamente y que su rango articular esté bien definido. Una vez hecho esto se procederá a calibrar todos los motores, dejándolos a punto para la realización de cualquier tarea programada para el brazo robot.

Al finalizar cualquier prueba o tarea con el brazo, deberá comprobarse de nuevo los puntos de la **Tabla 6** para asegurarse de que el robot seguirá trabajando de forma correcta durante los siguientes ensayos.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

DOCUMENTO PRESUPUESTO

Índice documento presupuesto

1	MATERIALES.....	148
2	MANO DE OBRA.....	149
3	EQUIPO INDUSTRIAL.....	149
4	ENSAMBLAJE BASE.....	150
4.1	BASE INFERIOR.....	150
4.2	BASE SUPERIOR.....	151
4.3	TRONCO BASE.....	151
5	ENSAMBLAJE HOMBRO.....	152
5.1	POLEA HOMBRO.....	153
5.2	SEPARADOR POLEA HOMBRO.....	153
5.3	BASE HOMBRO.....	153
5.4	COBERTURA HOMBRO.....	154
5.5	SOPORTE MOTOR.....	154
5.6	ADAPTADOR POLEA.....	154
6	ENSAMBLAJE BRAZO.....	155
6.1	POLEA BRAZO.....	156
6.2	SEPARADOR POLEA BRAZO.....	156
6.3	BASE BRAZO.....	156
6.4	SEPARADOR.....	157
6.5	COBERTURA BRAZO.....	157
7	ENSAMBLAJE ANTEBRAZO.....	158
7.1	ANTEBRAZO PRINCIPAL.....	159
7.2	ACOPLE EJE PRINCIPAL.....	159
7.3	ADAPTADOR MOTOR.....	159
7.4	CONEXIÓN MUÑECA.....	160
7.5	POLEA ANTEBRAZO.....	160
7.6	ACOPLE EJE SECUNDARIO.....	160
8	ENSAMBLAJE PINZA.....	161
8.1	SOPORTE PINZA.....	161
8.2	PINZA.....	162
8.3	ENGRANAJE 1.....	162
8.4	ENGRANAJE 2.....	162
8.5	FIJADOR PINZA.....	163
9	COSTE TOTAL.....	164

1 MATERIALES

Referencia	Unidad	Descripción	Precio [€]
Materiales			
MAT1	Ud	Arduino UNO R3	19.118 €
MAT2	Ud	Arandela D4 ISO 7089 D4	0.008 €
MAT3	Ud	Arandela D8 ISO 7089	0.016 €
MAT4	Ud	Cable USB de 0,3 m SW USB macho tipo A a B para Arduino UNO	2.828 €
MAT5	Ud	Cables puente macho a macho 20mm	0.075 €
MAT6	Ud	Driver L298n para control de motores paso a paso	3.950 €
MAT7	Ud	GT2 correa cerrada de 6mm de ancho y 280mm de longitud	1.161 €
MAT8	Ud	HTD 3M correa cerrada de 10mm de ancho y 330mm de longitud	5.684 €
MAT9	m2	Metacrilato de extrusión PMMA-XT para corte láser 2mm de espesor	23.778 €
MAT10	m2	Metacrilato de extrusión PMMA-XT para corte láser 3mm de espesor	35.667 €
MAT11	Ud	Microservo de rotación 180°, modelo MG90D	6.281 €
MAT12	Ud	Nema 17 (45Ncm) Bipolar de 2A, modelo 17HS16-2004S1	9.567 €
MAT13	Ud	Nema 17 Bipolar L=38mm w/ Planetary Gearbox 5:1, modelo 17HS15-1684S-PG5	25.051 €
MAT14	Ud	Polea GT2 de 16 dientes y 5mm de diámetro para correa de 6mm	1.256 €
MAT15	Ud	Polea GT2 sin dientes y 3mm de diámetro para correa de 6mm	1.651 €
MAT16	Ud	Polea HTD 3M de 16 dientes y 6mm de diámetro para correa de 11mm	2.441 €
MAT17	Ud	Protoboard MB-102 de 830 contactos	3.942 €
MAT18	Ud	Rodamiento axial de bolas 51104 20x35x10mm	9.510 €
MAT19	Ud	Rodamiento de bola rígido obturado rodante 6002-2RSH, 15x32x9mm	2.917 €
MAT20	Ud	Rodamiento de bola rígido obturado rodante 608-2RSH, 8x22x7mm	2.312 €
MAT21	Ud	Rodamiento de bola rígido obturado rodante 624-2RS1, 4x13x5mm	4.916 €
MAT22	Ud	Rodamiento de bola rígido obturado rodante 625-2RS1, 5x16x5mm	2.305 €
MAT23	Ud	Servomotor de rotación continua 360°, modelo DS04-NFC	8.524 €
MAT24	Ud	Tornillo M2,5x10mm ISO 1207	0.158 €
MAT25	Ud	Tornillo M3x10mm ISO 4762	0.158 €
MAT26	Ud	Tornillo M3x25mm ISO 4762	0.158 €
MAT27	Ud	Tornillo M4x20mm ISO 4762	0.047 €
MAT28	Ud	Tornillo M4x35mm ISO 4762	0.063 €
MAT29	Ud	Tornillo M8x40mm ISO 4762	0.174 €
MAT30	Ud	Tornillo M8x70mm ISO 4762	0.474 €

Referencia	Unidad	Descripción	Precio [€]
MAT31	Ud	Tornillo M8x80mm ISO 4014	0.395 €
MAT32	m	Tubo de aluminio de 15mm de diámetro y 1mm de espesor	3.318 €
MAT33	Ud	Tuerca M3 ISO 4032	0.008 €
MAT34	Ud	Tuerca M4 ISO 4032	0.008 €
MAT35	Ud	Tuerca M8 ISO 4032	0.032 €
MAT36	m	Varilla roscada de acero cincado M4	0.474 €
MAT37	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €

2 MANO DE OBRA

Referencia	Unidad	Descripción	Precio [€]
Mano de obra			
MO1	h	Ingeniero mecánico	21.363 €
MO2	h	Técnico de impresión 3D	10.645 €
MO3	h	Técnico de taller	9.507 €

3 EQUIPO INDUSTRIAL

Referencia	Ud	Descripción	Precio [€]
Maquinaria			
MAQ1	h	Cortadora láser	7.872 €
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €

La justificación de los costes de mano de obra y equipo industrial se encuentra detallada en el ANEXO II del Documento Memoria.

4 ENSAMBLAJE BASE

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Base	Ud	Ensamblaje Base. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller			
MATERIALES					
B1	Ud	Base Inferior del ensamblaje Base, excluido coste de ensamblaje	6.971 €	1	6.971 €
B2	Ud	Base Superior del ensamblaje Base, excluido coste de ensamblaje	5.16 €	1	5.157 €
B3	Ud	Tronco Base del ensamblaje Base, excluido coste de ensamblaje	4.957 €	1	4.957 €
MAT31	m	Tubo de aluminio de 15mm de diámetro y 1mm de espesor	3.318 €	0.1	0.332 €
MAT11	Ud	Nema 17 (45Ncm) Bipolar de 2A, modelo 17HS16-2004S1	9.567 €	1	9.567 €
MAT13	Ud	Polea GT2 de 16 dientes y 5mm de diámetro para correa de 6mm	1.256 €	1	1.256 €
MAT30	Ud	Tornillo M8x80mm ISO 4014	0.395 €	1	0.395 €
MAT29	Ud	Tornillo M8x70mm ISO 4762	0.474 €	4	1.896 €
MAT24	Ud	Tornillo M3x10mm ISO 4762	0.158 €	4	0.632 €
MAT34	Ud	Tuerca M8 ISO 4032	0.032 €	11	0.348 €
MAT17	Ud	Rodamiento axial de bolas 51104 20x35x10mm	9.510 €	1	9.510 €
MAT18	Ud	Rodamiento de bola rígido obturado rodante 6002-2RSH, 15x32x9mm	2.917 €	2	5.834 €
MANO DE OBRA					
MO3	h	Técnico de taller	9.507 €	2	19.014 €
MEDIOS AUXILIARES					
	%	Medios auxiliares sobre costes directos	65.869 €	10%	6.587 €
Total importe ejecución material					72.456 €

4.1 BASE INFERIOR

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
B1	Ud	Base Inferior del ensamblaje Base, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	62	3.100 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	2	0.256 €
Total importe ejecución material					6.971 €

4.2 BASE SUPERIOR

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
B2	Ud	Base Superior del ensamblaje Base, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	27	1.350 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1.5	0.192 €
Total importe ejecución material					5.157 €

4.3 TRONCO BASE

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
B3	Ud	Tronco Base del ensamblaje Base, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	23	1.150 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1.5	0.192 €
Total importe ejecución material					4.957 €

5 ENSAMBLAJE HOMBRO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Hombro	Ud	Ensamblaje Hombro. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller			
MATERIALES					
H1	Ud	Pola Hombro del ensamblaje Hombro, excluido coste de ensamblaje	2.236 €	1	2.236 €
H2	Ud	Separador Pola Hombro del ensamblaje Hombro, excluido coste de ensamblaje	2.50 €	1	2.503 €
H3	Ud	Base Hombro del ensamblaje Hombro, excluido coste de ensamblaje	6.349 €	1	6.349 €
H4	Ud	Cobertura Hombro del ensamblaje Hombro, excluido coste de ensamblaje	4.277 €	1	4.277 €
H5	Ud	Soporte Motor del ensamblaje Hombro, excluido coste de ensamblaje	2.216 €	1	2.216 €
H6	Ud	Adaptador Polea del ensamblaje Hombro, excluido coste de ensamblaje	0.998 €	1	0.998 €
MAT12	Ud	Nema 17 Bipolar L=38mm w/ Planetary Gearbox 5:1, modelo 17HS15-1684S-PG5	25.051 €	1	25.051 €
MAT31	m	Tubo de aluminio de 15mm de diámetro y 1mm de espesor	3.318 €	0.09	0.299 €
MAT15	Ud	Polea HTD 3M de 16 dientes y 6mm de diámetro para correa de 11mm	2.441 €	1	2.441 €
MAT28	Ud	Tornillo M8x40mm ISO 4762	0.174 €	1	0.174 €
MAT27	Ud	Tornillo M4x35mm ISO 4762	0.063 €	6	0.379 €
MAT26	Ud	Tornillo M4x20mm ISO 4762	0.047 €	3	0.142 €
MAT24	Ud	Tornillo M3x10mm ISO 4762	0.158 €	4	0.632 €
MAT33	Ud	Tuerca M4 ISO 4032	0.008 €	9	0.071 €
MAT34	Ud	Tuerca M8 ISO 4032	0.032 €	1	0.032 €
MAT3	Ud	Arandela D8 ISO 7089	0.016 €	3	0.047 €
MAT18	Ud	Rodamiento de bola rígido obturado rodante 6002-2RSH, 15x32x9mm	2.917 €	2	5.834 €
MAT19	Ud	Rodamiento de bola rígido obturado rodante 608-2RSH, 8x22x7mm	2.312 €	2	4.624 €
MAT21	Ud	Rodamiento de bola rígido obturado rodante 625-2RS1, 5x16x5mm	2.305 €	1	2.305 €
MANO DE OBRA					
MO3	h	Técnico de taller	9.507 €	3	28.521 €
MEDIOS AUXILIARES					
	%	Medios auxiliares sobre costes directos	89.130 €	10%	8.913 €
Total importe ejecución material					98.043 €

5.1 POLEA HOMBRO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H1	Ud	Pola Hombro del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	6	0.300 €
MANO DE OBRA					
M02	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					2.236 €

5.2 SEPARADOR POLEA HOMBRO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H2	Ud	Separador Pola Hombro del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT8	m2	Metacrilato de extrusión PMMA-XT para corte láser 2mm de espesor	23.778 €	0.0038	0.090 €
MANO DE OBRA					
M01	h	Ingeniero mecánico	21.363 €	0.05	1.068 €
M03	h	Técnico de taller	9.507 €	0.1	0.951 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.05	0.394 €
Total importe ejecución material					2.503 €

5.3 BASE HOMBRO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H3	Ud	Base Hombro del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	47	2.350 €
MANO DE OBRA					
M02	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	3	0.384 €
Total importe ejecución material					6.349 €

5.4 COBERTURA HOMBRO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H4	Ud	Cobertura Hombro del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	9.4	0.470 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1.5	0.192 €
Total importe ejecución material					4.277 €

5.5 SOPORTE MOTOR

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H5	Ud	Soporte Motor del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	5.6	0.280 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					2.216 €

5.6 ADAPTADOR POLEA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
H6	Ud	Adaptador Polea del ensamblaje Hombro, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	0.6	0.030 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.05	0.904 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	0.5	0.064 €
Total importe ejecución material					0.998 €

6 ENSAMBLAJE BRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Brazo	Ud	Ensamblaje Brazo. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller			
MATERIALES					
Br1	Ud	Pola Brazo del ensamblaje Brazo, excluido coste de ensamblaje	2.536 €	1	2.536 €
Br2	Ud	Separador Pola Brazo del ensamblaje Brazo, excluido coste de ensamblaje	2.53 €	1	2.531 €
Br3	Ud	Base Brazo del ensamblaje Brazo, excluido coste de ensamblaje	11.272 €	1	11.272 €
Br4	Ud	Separador del ensamblaje Brazo, excluido coste de ensamblaje	3.614 €	1	3.614 €
Br5	Ud	Cobertura Brazo del ensamblaje Brazo, excluido coste de ensamblaje	4.086 €	1	4.086 €
MAT13	Ud	Polea GT2 de 16 dientes y 5mm de diámetro para correa de 6mm	1.256 €	1	1.256 €
MAT14	Ud	Polea GT2 sin dientes y 3mm de diámetro para correa de 6mm	1.651 €	2	3.302 €
MAT11	Ud	Nema 17 (45Ncm) Bipolar de 2A, modelo 17HS16-2004S1	9.567 €	1	9.567 €
MAT30	Ud	Tornillo M8x80mm ISO 4014	0.395 €	1	0.395 €
MAT27	Ud	Tornillo M4x35mm ISO 4762	0.063 €	3	0.190 €
MAT25	Ud	Tornillo M3x25mm ISO 4762	0.158 €	4	0.632 €
MAT24	Ud	Tornillo M3x10mm ISO 4762	0.158 €	4	0.632 €
MAT33	Ud	Tuerca M4 ISO 4032	0.008 €	3	0.024 €
MAT34	Ud	Tuerca M3 ISO 4032	0.008 €	4	0.032 €
MAT2	Ud	Arandela D4 ISO 7089 D4	0.008 €	8	0.063 €
MAT19	Ud	Rodamiento de bola rígido obturado rodante 608-2RSH, 8x22x7mm	2.312 €	2	4.624 €
MANO DE OBRA					
MO3	h	Técnico de taller	9.507 €	2	19.014 €
MEDIOS AUXILIARES					
	%	Medios auxiliares sobre costes directos	63.769 €	10%	6.377 €
Total importe ejecución material					70.146 €

6.1 POLEA BRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Br1	Ud	Pola Brazo del ensamblaje Brazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	12	0.600 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					2.536 €

6.2 SEPARADOR POLEA BRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Br2	Ud	Separador Pola Brazo del ensamblaje Brazo, excluido coste de ensamblaje			
MATERIALES					
MAT8	m2	Metacrilato de extrusión PMMA-XT para corte láser 2mm de espesor	23.778 €	0.005	0.119 €
MANO DE OBRA					
MO1	h	Ingeniero mecánico	21.363 €	0.05	1.068 €
MO3	h	Técnico de taller	9.507 €	0.1	0.951 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.05	0.394 €
Total importe ejecución material					2.531 €

6.3 BASE BRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Br3	Ud	Base Brazo del ensamblaje Brazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	37	1.850 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.5	9.038 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	3	0.384 €
Total importe ejecución material					11.272 €

6.4 SEPARADOR

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Br4	Ud	Separador del ensamblaje Brazo, excluido coste de ensamblaje			
MATERIALES					
MAT8	m2	Metacrilato de extrusión PMMA-XT para corte láser 2mm de espesor	23.778 €	0.0056	0.133 €
MANO DE OBRA					
MO1	h	Ingeniero mecánico	21.363 €	0.1	2.136 €
MO3	h	Técnico de taller	9.507 €	0.1	0.951 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.05	0.394 €
Total importe ejecución material					3.614 €

6.5 COBERTURA BRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Br5	Ud	Cobertura Brazo del ensamblaje Brazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	4.3	0.215 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	2	0.256 €
Total importe ejecución material					4.086 €

7 ENSAMBLAJE ANTEBRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Antebrazo	Ud	Ensamblaje Antebrazo. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller			
MATERIALES					
A1	Ud	Antebrazo Principal del ensamblaje Antebrazo, excluido coste de ensamblaje	11.172 €	1	11.172 €
A2	Ud	Acople eje Principal del ensamblaje Antebrazo, excluido coste de ensamblaje	1.863 €	1	1.863 €
A3	Ud	Adaptador Motor del ensamblaje Antebrazo, excluido coste de ensamblaje	1.888 €	2	3.776 €
A4	Ud	Conexión Muñeca del ensamblaje Antebrazo, excluido coste de ensamblaje	9.406 €	1	9.406 €
A5	Ud	Polea Antebrazo del ensamblaje Antebrazo, excluido coste de ensamblaje	2.346 €	1	2.346 €
A6	Ud	Acople eje Secundario del ensamblaje Antebrazo, excluido coste de ensamblaje	1.843 €	1	1.843 €
MAT22	Ud	Servomotor de rotación continua 360°, modelo DS04-NFC	8.524 €	1	8.524 €
MAT35	m	Varilla roscada de acero zincado M4	0.474 €	0.17	0.081 €
MAT35	m	Varilla roscada de acero zincado M4	0.474 €	0.05	0.024 €
MAT20	Ud	Rodamiento de bola rígido obturado rodante 624-2RS1, 4x13x5mm	4.916 €	3	14.748 €
MAT10	Ud	Microservo de rotación 180°, modelo MG90D	6.281 €	1	6.281 €
MAT33	Ud	Tuerca M4 ISO 4032	0.008 €	6	0.047 €
MAT34	Ud	Tuerca M8 ISO 4032	0.032 €	1	0.032 €
MAT24	Ud	Tornillo M3x10mm ISO 4762	0.158 €	2	0.316 €
MANO DE OBRA					
MO3	h	Técnico de taller	9.507 €	3	28.521 €
MEDIOS AUXILIARES					
	%	Medios auxiliares sobre costes directos	88.979 €	10%	8.898 €
Total importe ejecución material					97.877 €

7.1 ANTEBRAZO PRINCIPAL

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A1	Ud	Antebrazo Principal del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	35	1.750 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.5	9.038 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	3	0.384 €
Total importe ejecución material					11.172 €

7.2 ACOPLA EJE PRINCIPAL

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A2	Ud	Acople eje Principal del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	0.6	0.030 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	0.2	0.026 €
Total importe ejecución material					1.863 €

7.3 ADAPTADOR MOTOR

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A3	Ud	Adaptador Motor del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	1.1	0.055 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	0.2	0.026 €
Total importe ejecución material					1.888 €

7.4 CONEXIÓN MUÑECA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A4	Ud	Conexión Muñeca del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	4.8	0.240 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.5	9.038 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					9.406 €

7.5 POLEA ANTEBRAZO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A5	Ud	Polea Antebrazo del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	8.2	0.410 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					2.346 €

7.6 ACOPLE EJE SECUNDARIO

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
A6	Ud	Acople eje Secundario del ensamblaje Antebrazo, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	0.2	0.010 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.1	1.808 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	0.2	0.026 €
Total importe ejecución material					1.843 €

8 ENSAMBLAJE PINZA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Pinza	Ud	Ensamblaje Pinza. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller			
MATERIALES					
P1	Ud	Soporte Pinza del ensamblaje Pinza, excluido coste de ensamblaje	3.938 €	1	3.938 €
P2	Ud	Pinza del ensamblaje Pinza, excluido coste de ensamblaje	3.833 €	2	7.666 €
P3	Ud	Engranaje 1 del ensamblaje Pinza, excluido coste de ensamblaje	3.785 €	1	3.785 €
P4	Ud	Engranaje 2 del ensamblaje Pinza, excluido coste de ensamblaje	3.785 €	1	3.785 €
P5	Ud	Fijador Pinza del ensamblaje Pinza, excluido coste de ensamblaje	2.414 €	2	4.827 €
MAT10	Ud	Microservo de rotación 180°, modelo MG90D	6.281 €	1	6.281 €
MAT23	Ud	Tornillo M2,5x10mm ISO 1207	0.158 €	8	1.264 €
MAT33	Ud	Tuerca M4 ISO 4032	0.008 €	2	0.016 €
MANO DE OBRA					
MO3	h	Técnico de taller	9.507 €	1.5	14.261 €
MEDIOS AUXILIARES					
	%	Medios auxiliares sobre costes directos	45.823 €	10%	4.582 €
Total importe ejecución material					50.405 €

8.1 SOPORTE PINZA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
P1	Ud	Soporte Pinza del ensamblaje Pinza, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	3.9	0.195 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					3.938 €

8.2 PINZA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
P2	Ud	Pinza del ensamblaje Pinza, excluido coste de ensamblaje			
MATERIALES					
MAT36	g	Zortrax Z-HIPS M200 filamento para 3D-impresora gris	0.050 €	1.8	0.090 €
MANO DE OBRA					
MO2	h	Técnico de impresión 3D	18.076 €	0.2	3.615 €
MAQUINARIA					
MAQ2	h	Impresora 3D Zortrax M200 Plus	0.128 €	1	0.128 €
Total importe ejecución material					3.833 €

8.3 ENGRANAJE 1

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
P3	Ud	Engranaje 1 del ensamblaje Pinza, excluido coste de ensamblaje			
MATERIALES					
MAT9	m2	Metacrilato de extrusión PMMA-XT para corte láser 3mm de espesor	35.667 €	0.0008	0.029 €
MANO DE OBRA					
MO1	h	Ingeniero mecánico	21.363 €	0.05	1.068 €
MO3	h	Técnico de taller	9.507 €	0.2	1.901 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.1	0.787 €
Total importe ejecución material					3.785 €

8.4 ENGRANAJE 2

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
P4	Ud	Engranaje 2 del ensamblaje Pinza, excluido coste de ensamblaje			
MATERIALES					
MAT9	m2	Metacrilato de extrusión PMMA-XT para corte láser 3mm de espesor	35.667 €	0.0008	0.029 €
MANO DE OBRA					
MO1	h	Ingeniero mecánico	21.363 €	0.05	1.068 €
MO3	h	Técnico de taller	9.507 €	0.2	1.901 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.1	0.787 €
Total importe ejecución material					3.785 €

8.5 FIJADOR PINZA

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
P5	Ud	Fijador Pinza del ensamblaje Pinza, excluido coste de ensamblaje			
MATERIALES					
MAT9	m2	Metacrilato de extrusión PMMA-XT para corte láser 3mm de espesor	35.667 €	0.00003	0.001 €
MANO DE OBRA					
M01	h	Ingeniero mecánico	21.363 €	0.05	1.068 €
M03	h	Técnico de taller	9.507 €	0.1	0.951 €
MAQUINARIA					
MAQ1	h	Cortadora láser	7.872 €	0.05	0.394 €
Total importe ejecución material					2.414 €

9 COSTE TOTAL

Referencia	Unidad	Descripción	Importe	Cantidad	Parcial
Base	Ud	Ensamblaje Base. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller	72.456 €	1	72.456 €
Hombro	Ud	Ensamblaje Hombro. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller	98.043 €	1	98.043 €
Brazo	Ud	Ensamblaje Brazo. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller	70.146 €	1	70.146 €
Antebrazo	Ud	Ensamblaje Antebrazo. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller	97.877 €	1	97.877 €
Pinza	Ud	Ensamblaje Pinza. Unión mediante tornillería de todos los componentes del ensamblaje por un técnico de taller	50.405 €	1	50.405 €
MAT1	Ud	Aduino UNO R3	19.118 €	1	19.118 €
MAT4	Ud	Cable USB de 0,3 m SW USB macho tipo A a B para Arduino UNO	2.828 €	1	2.828 €
MAT5	Ud	Cables puente macho a macho 20mm	0.075 €	40	3.010 €
MAT6	Ud	Driver L298n para control de motores paso a paso	3.950 €	3	11.850 €
MAT7	Ud	GT2 correa cerrada de 6mm de ancho y 280mm de longitud	1.161 €	2	2.323 €
MAT8	Ud	HTD 3M correa cerrada de 10mm de ancho y 330mm de longitud	5.684 €	1	5.684 €
MAT17	Ud	Protoboard MB-102 de 830 contactos	3.942 €	1	3.942 €
Presupuesto de ejecución material					437.682 €
13% de gastos generales					56.899 €
6% de beneficio industrial					26.261 €
Suma					520.842 €
21% IVA					109.377 €
Presupuesto de ejecución por contrata					630.219 €

Asciende el presupuesto de ejecución por contrata a la expresada cantidad de SEISCIENTOS TREINTA EUROS CON VEINTIDOS CÉNTIMOS.