

Escola Universitaria Politécnica



UNIVERSIDADE DA CORUÑA

Grado en Ingeniería Electrónica Industrial y Automática

TRABAJO DE FIN DE GRADO

TFG Nº: 770G01A189

TÍTULO: Diseño, montaje y control de un manipulador robótico.

AUTOR: ANTONIO REGUEIRA PEÑAMARÍA

**TUTOR: ESTEBAN JOVE PÉREZ
JOSE LUIS CALVO ROLLE**

FECHA: FEBRERO DE 2020

Fdo.: EL AUTOR

Fdo.: EL TUTOR

Agradecimientos

Este trabajo supone el cenit de mis estudios de grado, por tanto:

Quiero agradecer a mis tutores Esteban Jove Pérez y J.Luis Calvo Rolle su ayuda a la hora de llevar a cabo esta tarea.

Dar las gracias a todo el profesorado de la E.U.P. Siempre ha estado implicado con los alumnos y de un modo u otro han contribuido a formarme como ingeniero y persona.

Reconocer a mis amigos de siempre el apoyo recibido durante todo este tiempo y a los compañeros y amigos de la facultad su ayuda tanto académica como moral durante estos cinco años.

Por último dar las gracias a mi familia, que me ha apoyado en todo momento. Sin ellos no hubiera sido posible nada de esto. Gracias.

Antonio Regueira Peñamaría

Ferrol, Febrero 2020

TÍTULO: Diseño, montaje y control de un manipulador robótico.

ÍNDICE

PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: FEBRERO DE 2020

AUTOR: EL ALUMNO

Fdo.: ANTONIO REGUEIRA PEÑAMARÍA

I	ÍNDICE	5
	Contenidos del TFG	7
	Listado de figuras	11
	Listado de tablas	13
	Listado de códigos de programación	15
II	MEMORIA	17
	Índice del documento Memoria	19
1	OBJETO	21
2	ALCANCE	21
3	ANTECEDENTES	22
	3.1 Historia de la robótica	22
	3.2 Clasificación de los robots	23
	3.3 Robot manipulador	26
4	NORMAS Y REFERENCIAS	29
	4.1 Disposiciones legales y normas aplicadas	29
	4.2 Bibliografía	29
	4.3 Software utilizado	30
	4.4 Otras referencias	31
5	DEFINICIONES Y ABREVIATURAS	33
6	REQUISITOS DE DISEÑO	34
7	ANÁLISIS DE LAS SOLUCIONES	35
	7.1 Mecánica	35
	7.1.1 Morfología	35
	7.1.2 Tamaño del cuerpo	36
	7.1.3 Material	38
	7.1.4 Rodamientos	39
	7.1.5 Herramienta	40
	7.2 Electrónica	41
	7.2.1 Motor	41
	7.2.2 Reductora	42
	7.2.3 Sensorización	43
	7.2.4 Controlador	45
	7.3 Software	46
	7.3.1 Script	46
	7.3.2 Cinemática	46
8	RESULTADOS FINALES	48
	8.1 Mecánica	48
	8.1.1 Morfología	48
	8.1.2 Herramienta	53

8.2	Electrónica	54
8.2.1	Motorización	54
8.2.2	Drivers de motores	57
8.2.3	Sensorización	58
8.2.4	Controlador	59
8.3	Software	61
8.3.1	Librerías	61
8.3.2	Código	62
8.3.3	Cinemática	64
9	ORDEN DE PRIORIDAD ENTRE LOS DOCUMENTOS	66
III	ANEXOS	67
	Índice del documento Anexos	69
10	DOCUMENTACIÓN DE PARTIDA	71
11	Códigos de programación	74
11.1	Código Teensy	74
11.2	Código Python	82
12	ANEXOS EN FUNCIÓN DEL ÁMBITO DE APLICACIÓN DEL TFG	83
12.1	Seguridad	83
13	OTROS ANEXOS	84
13.1	Simulación y optimización de piezas	84
13.1.1	Simulación	84
13.1.2	Optimización	91
13.2	Ensamblaje del robot	94
13.2.1	Base	94
13.2.2	Cuerpo	94
13.2.3	Brazo	95
13.2.4	Antebrazo	95
13.2.5	Muñeca	96
13.3	Montaje y ensayo	97
13.3.1	Montaje	97
13.3.2	Ensayo	102
IV	PLANOS	105
	Índice del documento Planos	107
	BASE1	109
	BASE2	111
	BASE3	113
	BASE4	115
	BASE5	117
	CUERPO1	119

CUERPO2	121
CUERPO3	123
BRAZO1	125
TENSOR1	127
ANTEBRAZO1	129
ANTEBRAZO2	131
MUNECA1	133
BRIDA1	135
Soporte motor muñeca	137
Soporte encoder base	139
Soporte encoder hombro	141
Soporte encoder codo	143
Soporte encoder muñeca	145
Soporte imán hombro, codo y muñeca	147
Soporte imán base	149
V PLIEGO DE CONDICIONES	151
Índice del documento Pliego de condiciones	153
14 Especificaciones software	155
15 Especificaciones hardware	155
16 Condiciones almacenamiento	155
VI MEDICIONES	157
Índice del documento Mediciones	159
17 Materiales	161
18 Mano de obra	163
19 Software	164
20 Equipo	164
VII PRESUPUESTO	165
Índice del documento Presupuesto	167
21 Materiales	169
22 Mano de obra	171
23 Software	172
24 Equipo	172
25 Coste total	172

Listado de figuras

3.1	Robot Unimate	22
3.2	Robot Industrial	23
3.3	AGV	23
3.4	UAV Predator	24
3.5	Robot DaVinci	24
3.6	Coche Autónomo	25
3.7	Robot Cartesiano	26
3.8	Robot Esférico	26
3.9	Robot SCARA	27
3.10	Robot Articulado	27
3.11	Robot Delta	28
7.1	Robot IGUS	36
7.2	Alcance con diferentes tamaños de cuerpo	37
7.3	Fuerzas en un rodamiento	39
7.4	ejemplos gripper	40
7.5	Tipos de reductoras	42
7.6	Tipos de potenciómetros resistivos	43
7.7	Encoder magnético	44
7.8	Encoder óptico	44
7.9	Microcontroladores	45
7.10	Matriz DH	46
7.11	Posibles soluciones	47
8.1	Modelo Final Manipulador	48
8.2	Modelo Base Fija	49
8.3	Modelo Base Móvil	49
8.4	Modelo Cuerpo	50
8.5	Modelo Brazo	50
8.6	Detalle Tensor y Rodamiento	51
8.7	Modelo Antebrazo	51
8.8	Modelo Muñeca	52
8.9	Gripper flexible	53
8.10	Gripper flexible final	53
8.11	Momento	54

8.12	Detalle muelle	56
8.13	detalle contrapeso motor	57
8.14	Driver	57
8.15	Encoders AS5600	58
8.16	TCA9548A	59
8.17	Esquema conexiones	60
8.18	Máquina de estados	63
8.19	Área de trabajo del robot	65
13.1	Simulación base1	84
13.2	Simulación base2	85
13.3	Simulación base3	86
13.4	Simulación base4	86
13.5	Simulación base5	87
13.6	Simulación cuerpo1	87
13.7	Simulación cuerpo3	88
13.8	Simulación brazo1	89
13.9	Simulación tensor	89
13.10	Simulación antebrazo	90
13.11	Simulación Muñeca	90
13.12	Optimización brazo	91
13.13	Optimización antebrazo	92
13.14	Optimización muñeca	93
13.15	Ensamblaje base	94
13.16	Ensamblaje cuerpo	94
13.17	Ensamblaje brazo	95
13.18	Ensamblaje antebrazo	95
13.19	ensamblaje muñeca	96
13.20	Detalle del encoder de la base	97
13.21	Detalle del muelle y del encoder del hombro	97
13.22	Detalle del muelle estirado y del encoder del hombro	98
13.23	Detalle del montaje de drivers y breadboard	98
13.24	Detalle del tensor de la correa del antebrazo	99
13.25	Detalle del montaje de la rotación de la muñeca	99
13.26	Detalle de los motores del hombro, codo y muñeca	100
13.27	Detalle la articulación codo	100
13.28	Robot manipulador con pinza	101
13.29	Aproximación manual	103
13.30	Aproximación algoritmo	104

Listado de tablas

7.1	Comparación robots manipuladores	35
7.2	Propiedades materiales estructurales	38
7.3	Propiedades rodamientos	39
7.4	Comparación principales reductoras	40
7.5	Comparación principales reductoras	42
7.6	Comparación controladores	45
17.1	Poleas	161
17.2	Correas	161
17.3	Piezas genéricas hardware	162
17.4	Piezas en acero	163
17.5	Motores y drivers	163
18.1	Mano de obra	163
19.1	Software	164
20.1	Equipo	164
21.1	Poleas	169
21.2	Correas	169
21.3	Piezas genéricas hardware	170
21.4	Piezas en acero	171
21.5	Motores y drivers	171
22.1	Mano de obra	171
23.1	Software	172
24.1	Equipo	172
25.1	Coste total	172

Listado de códigos de programación

11.1	Código Teensy	74
11.2	Código Python	82

TÍTULO: Diseño, montaje y control de un manipulador robótico.

MEMORIA

PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: FEBRERO DE 2020

AUTOR: EL ALUMNO

Fdo.: ANTONIO REGUEIRA PEÑAMARÍA

Índice del documento MEMORIA

1 OBJETO	21
2 ALCANCE	21
3 ANTECEDENTES	22
3.1 Historia de la robótica	22
3.2 Clasificación de los robots	23
3.3 Robot manipulador	26
4 NORMAS Y REFERENCIAS	29
4.1 Disposiciones legales y normas aplicadas	29
4.2 Bibliografía	29
4.3 Software utilizado	30
4.4 Otras referencias	31
5 DEFINICIONES Y ABREVIATURAS	33
6 REQUISITOS DE DISEÑO	34
7 ANÁLISIS DE LAS SOLUCIONES	35
7.1 Mecánica	35
7.1.1 Morfología	35
7.1.2 Tamaño del cuerpo	36
7.1.3 Material	38
7.1.4 Rodamientos	39
7.1.5 Herramienta	40
7.2 Electrónica	41
7.2.1 Motor	41
7.2.1.1 Comparación	41
7.2.1.1.1 Economía:	41
7.2.1.1.2 Precisión:	41
7.2.1.1.3 Capacidad de fijar una posición:	41
7.2.1.1.4 Facilidad de control:	41
7.2.2 Reductora	42
7.2.3 Sensorización	43
7.2.3.1 Comparación	44
7.2.3.1.1 Economía:	44
7.2.3.1.2 Fiabilidad:	44
7.2.3.1.3 Facilidad de comunicación:	44
7.2.4 Controlador	45

7.3	Software	46
7.3.1	Script	46
7.3.2	Cinemática	46
8	RESULTADOS FINALES	48
8.1	Mecánica	48
8.1.1	Morfología	48
8.1.2	Herramienta	53
8.2	Electrónica	54
8.2.1	Motorización	54
8.2.2	Drivers de motores	57
8.2.3	Sensorización	58
8.2.4	Controlador	59
8.3	Software	61
8.3.1	Librerías	61
8.3.2	Código	62
8.3.3	Cinemática	64
9	ORDEN DE PRIORIDAD ENTRE LOS DOCUMENTOS	66

1 OBJETO

El objeto de este trabajo consiste en diseñar e implementar físicamente un manipulador robótico. Para ello se analizará el uso de diferentes morfologías, sensores y actuadores. Con el objetivo de manejar el manipulador, se elaborará un software de control que será testado sobre el montaje físico.

2 ALCANCE

El alcance comprende las siguientes tareas.

- Análisis de los diferentes tipos de manipuladores.
- Estudio de las posibles motorizaciones.
- Definición de carga máxima y del espacio de trabajo.
- Estudio de posibles herramientas, accionadas de manera eléctrica y neumática.
- Realimentación de la posición de cada una de las articulaciones.
- Evaluación de la cinemática del manipulador.
- Análisis de los resultados obtenidos y extracción de conclusiones.

3 ANTECEDENTES

3.1. Historia de la robótica

Según la RAE, robot se define como “Máquina o ingenio electrónico programable que es capaz de manipular objetos y realizar diversas operaciones.”

La palabra se dio a conocer cuando el escritor checo Karel Capek publicó su obra R.U.R (Robots Universales Rossum) en 1920, utilizando la palabra robot como derivado de robota, que significa trabajo o labor, conocida coloquialmente como servidumbre. A pesar de que existen casos documentados de autómatas programables antes del siglo XX, mediante vapor o sistemas mecánicos, es a partir del año 1961, año en el que se instaló el robot Unimate, cuando la robótica comenzó a despegar.

El Unimate se considera el primer robot industrial de la historia, instalado en una fábrica de coches, su objetivo era el de retirar piezas de un molde de fundición, un trabajo muy peligroso. Con memoria y diferentes servomecanismos era capaz de repetir la tarea con gran precisión. A partir de esta fecha la evolución ha sido constante y actualmente podemos encontrar robots en prácticamente todas las áreas de trabajo.

A finales del siglo XX se comenzó a introducir la visión artificial, que aunque en aquella época era poco sofisticada, conseguía guiar al robot. En la última década, con la integración de diferentes métodos de inteligencia artificial, aprendizaje automático y sensorización se ha conseguido realizar tareas que hace pocos años se consideraban inalcanzables.



Figura 3.1 – Robot Unimate

3.2. Clasificación de los robots

- **Industriales:** Son los más comunes, fueron los primeros robots en usarse y evolucionar, marcaron un hito en la manufactura, su principal uso, ya que permiten mover objetos, pintar, soldar o realizar cualquier tipo de movimiento repetitivo de manera fiable y muy precisa. Según la IFR [15], en enero del 2020 había más de 2.4 millones de robots industriales en el mundo, siendo instalados en el 2018 422000 unidades, casi 4 veces más que en el 2010 con 118000 robots industriales instalados.

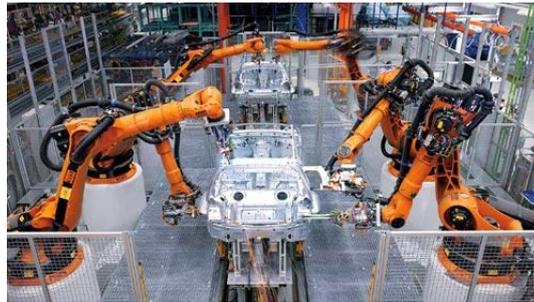


Figura 3.2 – Robot Industrial

- **Almacenaje:** Dedicados principalmente al transporte de palés o productos dentro de los almacenes, permiten un mayor aprovechamiento del volumen de almacenaje y una trazabilidad completa, con la tendencia al alza del “ecommerce” su uso es cada vez más extendido.



Figura 3.3 – AGV

- **Militares:** Introducidos durante la primera guerra mundial, como detonadores de minas, no tuvieron mucha repercusión hasta el 2002, donde se usó el robot Hermes para entrar en las cuevas de Afghanistan, también para detonación de minas. A partir del 2011 se han implantado en todos los escenarios, siendo los más habituales los robots aéreos de vigilancia.



Figura 3.4 – UAV Predator

- **Médicos:** Usados normalmente como asistentes de operaciones quirúrgicas, siendo el más famoso el robot DaVinci, creado en el año 1999. Consiguen una mayor precisión con menor invasión en el cuerpo, permitiendo al paciente recuperarse antes y reduciendo el riesgo de la operación.



Figura 3.5 – Robot DaVinci

- **Servicio:** Llevando la comida a las mesas en un restaurante o limpiando por las noches las oficinas y supermercados. Haciendo tareas que hasta hace pocos años eran solo propias de las películas, como vigilantes nocturnos o coches autónomos.



Figura 3.6 – Coche Autónomo

3.3. Robot manipulador

Un robot manipulador es aquel que tiene una serie de articulaciones que le permiten interactuar con objetos de una manera precisa, existen los siguientes tipos:

- **Cartesiano:** Es un robot en el que sus ejes son prismáticos, normalmente tiene 3, los cuales le permiten moverse en un espacio tridimensional, suelen usarse como CNC para mecanizado de materiales o como impresoras 3D. En esta categoría se pueden incluir los puente grúa, utilizados por ejemplo para almacenaje de contenedores o el movimiento de grandes cargas.

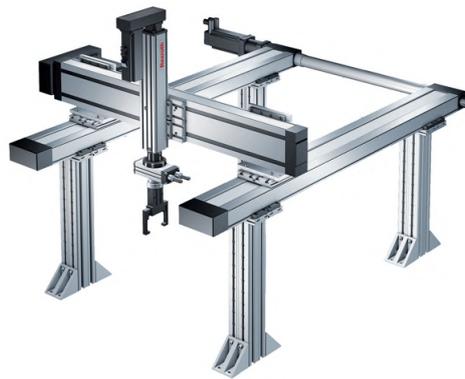


Figura 3.7 – Robot Cartesiano

- **Esférico:** También conocido como robot polar, actualmente en desuso, tienen 2 articulaciones rotacionales y una prismática, permitiendo un área de trabajo similar a una esfera, como ejemplo está el robot Unimate, del año 1961, conocido como el primer manipulador industrial.

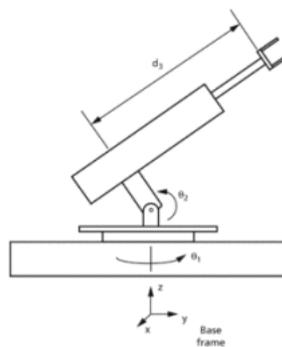


Figura 3.8 – Robot Esférico

- **SCARA:** Acrónimo por sus siglas en inglés “*Selective Compliant Assembly Robot Arm*” es un robot articulado que tiene la base en un plano Z de forma rígida. Moviéndose en el plano XY, la herramienta final si puede bajar en el plano Z, esto le permite grandes velocidades, es por ello que se suele usar en ensamblaje.



Figura 3.9 – Robot SCARA

- **Articulado:** El tipo más usado industria, son semejantes a un brazo humano. Normalmente tienen 6 grados de libertad siendo todas las articulaciones rotacionales, esto permite alcanzar puntos con determinada orientación que ningún otro robot puede.



Figura 3.10 – Robot Articulado

- **Delta:** Es un tipo de robot paralelo, esto es, que la herramienta siempre se desplaza de forma paralela a la base fija, se utiliza en empaquetado de objetos. Debido a que es el robot más rápido, también se puede ver como impresora 3D.



Figura 3.11 – Robot Delta

4 NORMAS Y REFERENCIAS

4.1. Disposiciones legales y normas aplicadas

- *UNE 1027* Dibujos Técnicos. Plegado de Planos.
- *UNE 1032* Dibujos Técnicos. Acotación.
- *UNE-EN ISO 3098-0* Documentación técnica de productos. Escritura. Requisitos generales.
- *UNE-EN ISO 5455* Dibujos Técnicos. Escalas.
- *UNE-EN ISO 5457* Documentación técnica de productos. Formatos y representación de los elementos gráficos de las hojas de dibujo.
- *UNE-EN ISO 7200* Documentación técnica de productos. Campos de datos en bloques de títulos y en cabeceras de documentos.

4.2. Bibliografía

- [1] OLLERO BATURONE, A; *Robótica Manipuladores y robots móviles*, 1ª ed, Barcelona, Marcombo, (2001).
- [2] REYES CORTÉS, F; *Robótica control de robots manipuladores*, 1ª ed, Barcelona, Marcombo, (2011).
- [3] ROLDÁN VILORIA, J; *Máquinas y herramientas procesos y cálculos mecánicos*, 1ª ed, Madrid, Ediciones Paraninfo, (2019).
- [4] *Gear efficiency — key to lower drive cost*, MACHINEDESIGN [Consulta 1 febrero 2020]. Disponible en: <https://www.machinedesign.com/motors-drives/article/21834720/gear-efficiency-key-to-lower-drive-cost>
- [5] *Choosing the right gear reducer*, DIRECT INDUSTRY [Consulta 1 febrero 2020]. Disponible en: <http://guide.directindustry.com/choosing-the-right-gear-reducer/>
- [6] *Types of Backlashes*, KHK GEARS [Consulta 1 febrero 2020]. Disponible en: https://khkgears.net/new/gear_knowledge/gear_technical_reference/gear_backlash.html
- [7] *¿Qué es un Rodamiento?*, NSK [Consulta 1 febrero 2020]. Disponible en: <https://www.nskeurope.es/es/products/what-s-a-bearing.html>
- [8] *What is a constant voltage drive for a stepper motor and when is it used?*, MOTION CONTROL TIPS [Consulta 1 febrero 2020]. Disponible en: <https://www.motioncontroltips.com/what-is-constant-voltage-drive-for-stepper-motor-and-when-is-it-used/>

- [9] *Teensy USB Development Board*, PJRC [Consulta 1 febrero 2020]. Disponible en: <https://www.pjrc.com/store/teensy32.html>
- [10] *Qué sensor de posición elegir*, DIRECT INDUSTRY [Consulta 1 febrero 2020]. Disponible en: <http://guide.directindustry.com/es/que-sensor-de-posicion-elegir/>
- [11] *Industrial Robots and Robot System Safety*, UNITED STATES DEPARTMENT OF LABOR [Consulta 1 febrero 2020]. Disponible en: https://www.osha.gov/dts/osta/otm/otm_iv/otm_iv_4.html
- [12] *Articulated robot*. Wikimedia Foundation. [Consulta 1 febrero 2020]. Disponible en: https://en.wikipedia.org/wiki/Articulated_robot
- [13] *Gear train*. Wikimedia Foundation. [Consulta 1 febrero 2020]. Disponible en: https://en.wikipedia.org/wiki/Gear_train
- [14] *Microstepping myths*. Machine Design. [Consulta 1 febrero 2020]. Disponible en: <https://www.machinedesign.com/archive/article/21812154/microstepping-myths>
- [15] *FACTS about ROBOTS – worldwide*, INTERNATIONAL FEDERATION OF ROBOTICS [Consulta 1 febrero 2020]. Disponible en: <https://ifr.org/ifr-press-releases/news/facts-about-robots-worldwide>
- [16] *Los 6 tipos de robots y sus características*, LIFEDER [Consulta 1 febrero 2020]. Disponible en: <https://www.lifeder.com/los-6-tipos-robots-principales/>
- [17] *Inverse Kinematics (IK) Solution of a Robotic Manipulator using PYTHON*, THE SCI PUB [Consulta 1 febrero 2020]. Disponible en: <https://thescipub.com/pdf/10.3844/jmrsp.2019.542.551>
- [18] *Momento*, XUNTA DE GALICIA [Consulta 1 febrero 2020]. Disponible en: https://www.edu.xunta.es/espazoAbalar/sites/espazoAbalar/files/datos/1464947489/contido/3_momento.html

4.3. Software utilizado

- Visual Studio Code 1.39.2
- Autodesk Inventor Profesional 2019
- Cura 2.3.0
- Thonny 3.1.2
- Fritzing 0.9.4

4.4. Otras referencias

- [19] *A Project to Make Japan's First Domestically Manufactured Industrial Robot Begins*, KAWASAKI ROBOTICS [Consulta 1 febrero 2020]. Disponible en: https://robotics.kawasaki.com/en1/anniversary/history/history_02.html
- [20] *Pneumatic robot gripper*, ALIBABA [Consulta 1 febrero 2020]. Disponible en: https://www.alibaba.com/product-detail/Pneumatic-robot-gripper-sheet-metal-clamps_60069069270.html
- [21] *2F-85 and 2F-140 Grippers*, ROBOTIQ [Consulta 1 febrero 2020]. Disponible en: <https://robotiq.com/products/2f85-140-adaptive-robot-gripper>
- [22] *RobotDigg robot gripper*, CULTS [Consulta 1 febrero 2020]. Disponible en: <https://cults3d.com/en/3d-model/gadget/robotdigg-robot-gripper>
- [23] *Digital Stepper Driver*, STEPPER ONLINE [Consulta 1 febrero 2020]. Disponible en: <https://www.omc-stepperonline.com/digital-stepper-driver-10-42a-20-50vdc-for-nema-17-23-24-stepper-motor-dm542t.html?search=dm542t>
- [24] *Robot industrial*, WIKIMEDIA FOUNDATION [Consulta 1 febrero 2020]. Disponible en: https://es.wikipedia.org/wiki/Robot_industrial
- [25] *AGV forklifts optimize logistics at large warehouse*, ORWAK [Consulta 1 febrero 2020]. Disponible en: <https://orwak.com/agv-forklifts-optimize-logistics-at-large-warehouse/>
- [26] *General Atomics MQ-1 Predator*, WIKIMEDIA FOUNDATION [Consulta 1 febrero 2020]. Disponible en: https://es.wikipedia.org/wiki/General_Atomics_MQ-1_Predator
- [27] *De 22 a 45 millones de facturación en un año: los robots Da Vinci crecen un 40*, MERCA2 [Consulta 1 febrero 2020]. Disponible en: <https://www.merca2.es/facturacion-robots-da-vinci/>
- [28] *Impact of Autonomous vehicles on the economy*, USE OF TECHNOLOGY [Consulta 1 febrero 2020]. Disponible en: <https://www.useoftechnology.com/autonomous-vehicles-impact-on-economy/>
- [29] *What is a Robotic Manipulator?*, AZO ROBOTICS [Consulta 1 febrero 2020]. Disponible en: <https://www.azorobotics.com/Article.aspx?ArticleID=138>
- [30] *spherical-coordinate robot (polar-coordinate robot, spherical-configuration robot, spherical-polar robot)*, CREALECTICS [Consulta 1 febrero 2020]. Disponible en: <https://crealectics.com/2017/08/30/articulation-a-dictionary-of-mechanical-engineering/>

- [31] *Large type SCARA robots YK-XG*, YAMAHA [Consulta 1 febrero 2020]. Disponible en: <https://global.yamaha-motor.com/business/robot/lineup/ykxg/large/>
- [32] *Delta Robot DR*, WEISS [Consulta 1 febrero 2020]. Disponible en: <https://www.weiss-gmbh.de/Delta-Robot-DR.4188.0.html?&L=1>
- [33] *TeensyStep*, LUNI64 [Consulta 1 febrero 2020]. Disponible en: <https://luni64.github.io/TeensyStep/index>
- [34] *AS5600 Position Sensor*, AMS [Consulta 1 febrero 2020]. Disponible en: <https://ams.com/as5600>
- [35] *roboLink® RL-DC*, IGUS [Consulta 1 febrero 2020]. Disponible en: <https://www.igus.com/product/14869>
- [36] *What is radial load and axial load?*, QUORA [Consulta 1 febrero 2020]. Disponible en: <https://www.quora.com/What-is-radial-load-and-axial-load>
- [37] *Rotary Potentiometer*, FARNELL [Consulta 1 febrero 2020]. Disponible en: <https://uk.farnell.com/bourns/3590s-2-5021/potentiometer-5k/dp/1700942>
- [38] *Pole Rotary Encoder Magnets*, MAGCAM [Consulta 1 febrero 2020]. Disponible en: <https://www.magcam.com/application/2-pole-rotary-encoder-magnets>
- [39] *What is an optical encoder?*, CELERA MOTION [Consulta 1 febrero 2020]. Disponible en: <https://www.celeramotion.com/microe/what-is-an-optical-encoder/>
- [40] *Misumi*, MISUMI [Consulta 1 febrero 2020]. Disponible en: <https://uk.misumi-ec.com/>
- [41] *Iglidur J4*, IGUS [Consulta 1 febrero 2020]. Disponible en: <https://www.igus.es/product/?artnr=PRT-02-20-AL>
- [42] *Define a Tool (TCP)*, ROBODK [Consulta 1 febrero 2020]. Disponible en: <https://robodk.com/doc/en/General.html>
- [43] *Adafruit TCA9548A*, ADAFRUIT [Consulta 1 febrero 2020]. Disponible en: <https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout/overview>
- [44] *Parametrización Denavit-Hartenberg para robots: teoría, vídeo y nueva aplicación libre*, CIENCIA EXPLICADA [Consulta 1 febrero 2020]. Disponible en: <https://www.ciencia-explicada.com/2013/02/parametrizacion-denavit-hartenberg-para.html>
- [45] *Beetle - The Smallest Arduino*, DFRobot [Consulta 1 febrero 2020]. Disponible en: <https://www.dfrobot.com/product-1075.html>
- [46] *Teensy 4.0 Development Board*, PJRC [Consulta 1 febrero 2020]. Disponible en: <https://www.pjrc.com/store/teensy40.html>

5 DEFINICIONES Y ABREVIATURAS

AIISI: American Iron and Steel Institute.

Análisis elementos finitos: Simulación computacional de esfuerzos mecánicos que se utiliza para hacer análisis y mejorar la calidad de los productos.

Comunicación serie: Proceso de comunicación en el que se envían los bits de uno en uno, a través de un bus.

Driver: Controlador de motores, incorpora la electrónica de potencia necesaria para que, al recibir cierta señal lógica, varíe la potencia eléctrica entregada a un motor.

FDM: fused deposition modeling.

Grados de libertad: Término usado en robótica para designar una articulación de cualquier tipo, cuantos más grados de libertad tenga un robot, mayor capacidad de maniobra.

I2C: Protocolo de comunicación que utiliza 2 cables, uno para la señal de reloj y otro para los datos, comunicación serie multi esclavo y multi maestro.

Multiplexor: Circuito combinacional con varias entradas y una salida, permite elegir el paso de información de una entrada en concreto a la salida.

PLA: Polylactic acid.

PPR: Puntos Por Revolución.

PWM: Pulse Width Modulation.

RAM: Random Acces Memory.

TCP: Tool Center Point.

6 REQUISITOS DE DISEÑO

Los requisitos de diseño de este robot se pueden resumir en los siguientes puntos:

- Diseñar la estructura del manipulador, de 5 grados de libertad, de forma que sea fácilmente intercambiable y lo más sencilla posible.
- Conseguir, como mínimo, un alcance de 750mm.
- Analizar posibles herramientas con forma de pinza que tengan actuador propio y sirvan para agarrar objetos sin deformarlos.
- Motorizar las diferentes articulaciones de la forma más óptima posible, pudiendo manejar un objeto de 1 kg en su peor posición.
- Implementar sensores de posición para tener una realimentación completa de todas articulaciones.
- Diseñar un script para poder controlar el manipulador.
- Analizar la cinemática del robot.

7 ANÁLISIS DE LAS SOLUCIONES

En esta sección se analizará de manera detallada las posibles soluciones relacionadas con la mecánica, electrónica y software empleado.

7.1. Mecánica

7.1.1. Morfología

El primer requisito es el de diseñar la estructura del manipulador, además tenemos el condicionante del segundo requisito, el robot debe tener un alcance de 750mm.

Como robot manipulador es necesario tener la máxima maniobrabilidad posible, es por ello que se descarta el robot delta y el esférico, ya que, o no permiten la rotación del objeto manipulado o su morfología hace que el espacio de trabajo no pueda tener obstáculos.

En la tabla 7.1 se comparan las 3 posibles soluciones restantes tabuladas.

Tipo	Scara	Articulado	Cartesiano
Velocidad	Alta	Media	Media
Capacidad de carga	media	media	Alta
Maniobrabilidad	Baja	Alta	Media
Potencia de los motores	Baja	Alta	Baja
Complejidad mecánica	Media	Media	Baja

Tabla 7.1 – Comparación robots manipuladores

Teniendo en cuenta que el objetivo es poder manipular un objeto, se evalúa de forma preferente la característica de maniobrabilidad, la velocidad y la capacidad de carga son factores relativamente fáciles de escalar, añadiendo motores más potentes o una estructura más sólida. Sin embargo, la agilidad con la que un robot puede moverse en su entorno es una característica intrínseca a la morfología del mismo.

Por esta razón se escoge el robot articulado como modelo a seguir.

Se seguirá una morfología tradicional, dividiendo así el robot en base, cuerpo, brazo, antebrazo, muñeca y herramienta, como se muestra en la figura 7.1.

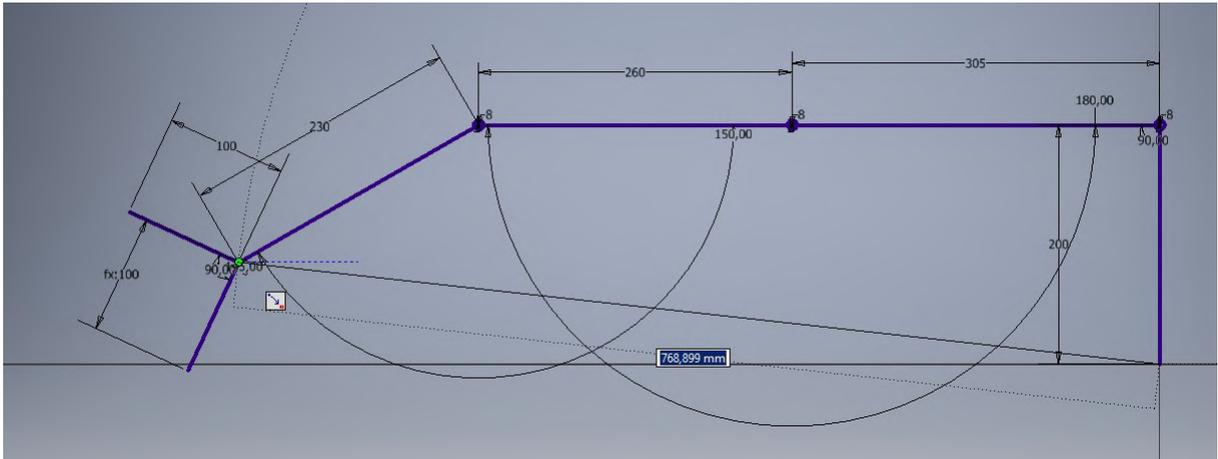


Figura 7.1 – Robot IGUS

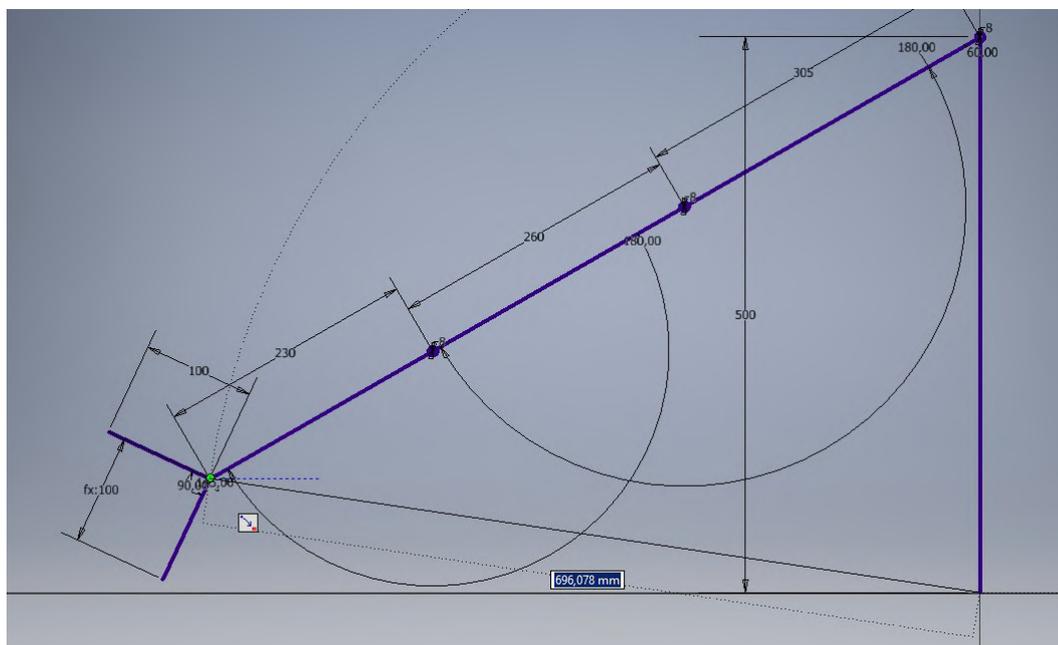
7.1.2. Tamaño del cuerpo

En cuanto a las proporciones de las diferentes partes se intentará seguir a la forma humana, haciendo que las secciones antebrazo y brazo sean parecidas y la muñeca ligeramente más corta.

Para la Base y el cuerpo existen 2 opciones, hacer un cuerpo largo, elevando la articulación hombro a lo largo del eje Z o acercar el hombro lo máximo a la base. La longitud del cuerpo elevará el área de trabajo del robot a lo largo del eje Z. Como se quiere manipular objetos que se encuentren en el mismo plano que la base, se opta por un cuerpo reducido, esto además hace que sea más sencillo el diseño, ya que se reduce el par soportado por los rodamientos de la base. Las diferencias entre los alcances en el plano de la base se ven en la figura 7.2 a y b.



(a) Alcance con cuerpo corto = 780mm



(b) Alcance con cuerpo largo = 696mm

Figura 7.2 – Alcance con diferentes tamaños de cuerpo

En estas figuras también se puede observar unas dimensiones de referencia para cada sección, aunque podrán cambiar para adaptarse a las poleas, correas y motores existentes en el mercado.

7.1.3. Material

Antes de diseñar la estructura se debe analizar el material con el que será construida. Esto implicará diferencias en las formas de los eslabones y las capacidades de carga.

Los 3 materiales actuales más comunes en la construcción de manipuladores, tanto a nivel aficionado como profesional, teniendo como requisitos que tengan propiedades anticorrosivas y que sean de fácil fabricación son:

- **Acero:** El acero es una aleación de hierro-carbono, con un porcentaje de este último que varía entre el 0.008 % y el 2%. Existen multitud de aleaciones, cada cual con sus propiedades mecánicas. Destacan aquellos comúnmente llamados inoxidables, siendo el más usado el denominado AISI 304, debido a su alto poder anticorrosivo. evitando de este modo tener que planificar futuros mantenimientos.
- **Aluminio:** El aluminio es tercer elemento mas común en la corteza terrestre. Desde mediados del siglo XX, debido a nuevas tecnologías para su procesamiento, ha ido incrementando su uso como elemento de construcción. Existen multitud de aleaciones, siendo los más populares, por su capacidad anticorrosiva, los de la serie 6000.
- **PLA:** El PLA es un polímero derivado del ácido láctico, es el principal material utilizado en las impresoras 3D basadas en tecnología FDM, biodegradable y fácilmente extruible.

Material	Densidad(g/cm ³)	Módulo elasticidad(MPa)	Anticorrosión
Acero 304	7.93	190-210	Muy alta
Aluminio 6061	2.7	70	Alta
PLA	1.3	37	Muy alta

Tabla 7.2 – Propiedades materiales estructurales

El PLA queda descartado como elemento estructural, se debe poder manipular un objeto de 1 kg a 750mm de distancia de la base, esto haría que la estructura fuera demasiado compleja y voluminosa.

Debido a la diferencia en el módulo de elasticidad, que influye de manera determinante en la flexión de un cuerpo sólido que esté en voladizo, se opta por el acero, esto ayudará a que la estructura sea más pequeña que si fuera en aluminio, como desventaja pesará más.

7.1.4. Rodamientos

Como análisis de la parte estructural solo queda mencionar los rodamientos, estos son una parte esencial en los ejes de rotación, permiten comunicar la energía mecánica de los motores a las articulaciones, maximizando el rendimiento energético reduciendo el rozamiento, tienen las siguientes características principales:

- **Dirección de la carga:** Axial y radial, en función de su dirección de montaje y la aplicación de las cargas que soportará.
- **Lubricante:** Hasta hace pocos años todos necesitaban lubricación, añadiendo costes de mantenimiento. Actualmente existen rodamientos fabricados con polímeros con bajo coeficiente de rozamiento que hace innecesaria esta ayuda.
- **Vida útil:** En función normalmente de la calidad del rodamiento, durará más o menos revoluciones.
- **Tipo:** Los hay de bolas, agujas, cónicos, cilindros etc, cada uno con diferentes usos y características, dependiendo principalmente de la dirección de la carga.

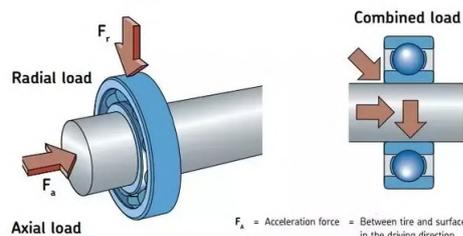


Figura 7.3 – Fuerzas en un rodamiento

Tipo	Bolas	Rodillos	Polímero
Soporte de F radial	Alta	Muy alta	Media
Soporte de F axial	Alta	Alta	Baja
Peso	Medio	Medio	Bajo
Soporta desalineado del eje	Alta	Media	Baja
Precio	Bajo	Bajo	Alto

Tabla 7.3 – Propiedades rodamientos

En función del uso y ubicación del rodamiento se colocarán unos u otros, los de polímero destacan por su ligereza y ausencia de mantenimiento y los de bolas/rodillos por su precio y variedad de modelos.

7.1.5. Herramienta

Como último requisito relacionado con la mecánica se encuentra el análisis de las posibles herramientas, estas se dividen en dos grupos principales, las accionadas de manera neumática y las accionadas de manera eléctrica, la mejor forma de compararlas es mediante una tabla.

7.4

Tipo	Neumática	eléctrica
Precisión	Baja	Alta
Fuerza	Alta	Baja
Simplicidad mecánica	Alta	Media
Peso	Bajo	Alto
Facilidad de actuación	Alta	Baja
Coste	Bajo	Alto

Tabla 7.4 – Comparación principales reductoras

Además las neumáticas pueden ser copas de succión, que permiten coger todo tipo de objetos, desde pequeñas cajas o bolsas hasta cristales de parabrisas.

Como contra, estas necesitan una bomba de vacío o presión, un regulador de esa presión para no dañar la herramienta y un sistema de tubos para llevar el aire.

Sin embargo, para controlar una herramienta eléctrica, solo es necesario conectar los cables de alimentación del motor hasta la misma, pudiendo regular la fuerza ejercida.

En resumen, si el movimiento a realizar es poco preciso o siempre el mismo se puede utilizar una herramienta neumática, abaratando la mayoría de costes, si se necesita cierta precisión o la herramienta es una pinza o similar y va a interactuar con objetos de diferentes tamaños es recomendable tener una herramienta eléctrica.



Figura 7.4 – ejemplos gripper

7.2. Electrónica

7.2.1. Motor

El tercer requisito es una motorización óptima, para ello se debe introducir primero los 3 tipos de motores que más se utilizan.

- **Motor DC:** El más clásico, de corriente continua y con escobillas, para controlarlo se suele utilizar modulación en ancho de pulso (PWM), necesita mantenimiento por el desgaste de escobillas y tiene baja relación torque-peso.
- **Motor BLDC:** Como el anterior pero sin escobillas, por tanto el mantenimiento se reduce drásticamente, tiene la mayor relación torque-peso pero necesita un driver más complejo, ya que se necesita generar una onda senoidal para excitar las bobinas del motor.
- **Motor paso a paso:** Es actualmente el motor más usado cuando se necesita precisión y repetitividad, convierte una serie de impulsos eléctricos en rotación de un valor angular fijo, normalmente 1.8 grados por paso. Para poder aprovechar todo su potencial se necesitan drivers con control por corriente constante, siendo más caros que los de control por tensión.

7.2.1.1. Comparación

A continuación se realiza la comparación en 4 áreas principales:

7.2.1.1.1. Economía: En este apartado se incluye el precio de un driver, los motores más económicos son los DC, los hay de todas las formas y potencias, con y sin reductoras.

7.2.1.1.2. Precisión: Los motores paso a paso son los más precisos, debido a su construcción, suelen tener 1,8 grados por paso, por tanto para 360 grados se usan 200 pasos, los drivers normalmente incorporan la función de microstepping, mediante este método se puede dividir cada paso inicial del motor en una serie de micro pasos, normalmente hasta 256, incrementando la precisión a costa del torque [14].

7.2.1.1.3. Capacidad de fijar una posición: Sin tener en cuenta el uso de frenos o reductoras autoblocantes, tanto los paso a paso como los BLDC pueden fijar su posición, energizando las mismas bobinas el rotor no se moverá de su sitio hasta que se ejerza un par superior al torque de mantenimiento.

7.2.1.1.4. Facilidad de control: Entendiendo como control la facilidad de hacer que el motor gire con resultado previsible, el más sencillo es el motor paso a paso, sin entrar en detalle de los diferentes tipos de drivers, debido a su construcción se puede esperar que el motor llegue a cierta posición siempre y cuando se dimensione correctamente.

7.2.2. Reductora

Es la segunda parte del bloque motor, reducen la velocidad y aumentan el torque o par, disminuyendo el tamaño del motor necesario y aumentando la precisión. En contraprestación tienen un rendimiento, de forma que se pierde cierta energía mecánica en forma de calor, debido a la fricción entre los engranajes. Existen principalmente los siguientes tipos:

- **Tornillo sin fin:** El más sencillo, consta de un tornillo y una corona, tiene una alta capacidad de reducción pero el rendimiento se reduce de un 93 % hasta aproximadamente un 50 % con una reducción de 300:1. La salida está girada 90 grados con respecto al eje del motor.
- **Tren de engranajes:** Pueden ser de 2 o más engranajes con dientes rectos o helicoidales, los rectos son más ruidosos, tienen más backlash pero aguantan más torque con el mismo tamaño que los helicoidales. Pueden tener la salida en el mismo sentido o girada 90 grados con respecto al eje del motor.
- **Planetarios:** Son más complejos, disponen de una corona, un piñón central (también llamado sol) y unos planetas, ya que tienen muchos dientes en contacto el torque se reparte mejor, aumentando su capacidad: Tienen la salida en la misma dirección que el eje del motor.

Tipo	Capacidad de reducción	η con baja reducción(%)	η con alta reducción(%)	Capacidad de torque
Tornillo sin fin	alta	90	49	alta
Tren engranajes(helicoidal)	media	95	98	media
Tren engranajes(rectos)	media	94	97	media
Planetario	alta	97	97	alta

Tabla 7.5 – Comparación principales reductoras

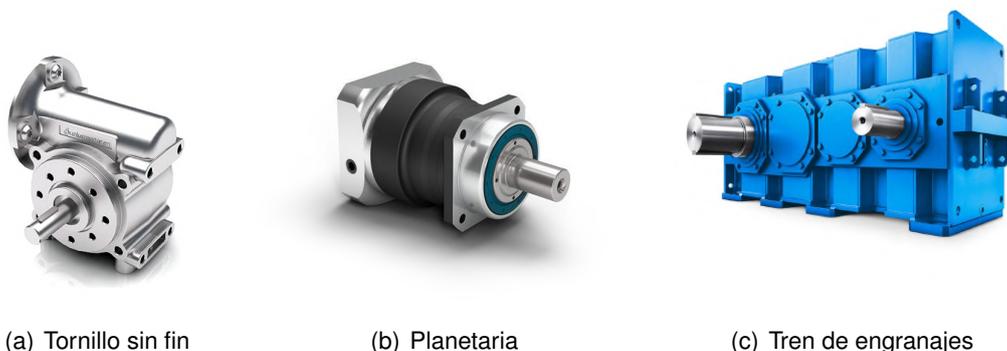


Figura 7.5 – Tipos de reductoras

7.2.3. Sensorización

Para conocer la posición de los diferentes eslabones se necesitan sensores, pueden ir integrados en el propio actuador o situarse en la articulación, si están incorporados en el propio actuador no se obtiene el ángulo real de las articulación si no del eje del actuador. Esto puede dar lugar a errores, debido, principalmente, al conocido backlash [6] u holgura en castellano, que es el juego que existe entre diferentes piezas móviles, como entre engranajes de una reductora, en la unión entre un eje y la chaveta o una correa con su polea. Este juego es imprescindible para que exista movimiento pero afecta a la precisión del sistema, por ello es siempre preferible tener el sensor en la articulación final, dejando que el software corrija cualquier diferencia entre el valor deseado y el real. Los tipos principales de sensores son los 3 siguiente:

- **Resistivos:** Miden la resistencia de una pista conductora, entre un punto de referencia y un cursor conectado a una pieza móvil, suelen ser los más económicos, sufren de desgaste, debido al rozamiento y de alteraciones en la medida por efectos como la temperatura o la humedad. Tienen resolución infinita, ya que son analógicos, los más comunes son de pista de carbón aunque también existen bobinados. Suelen necesitar una electrónica de instrumentación para poder funcionar correctamente.

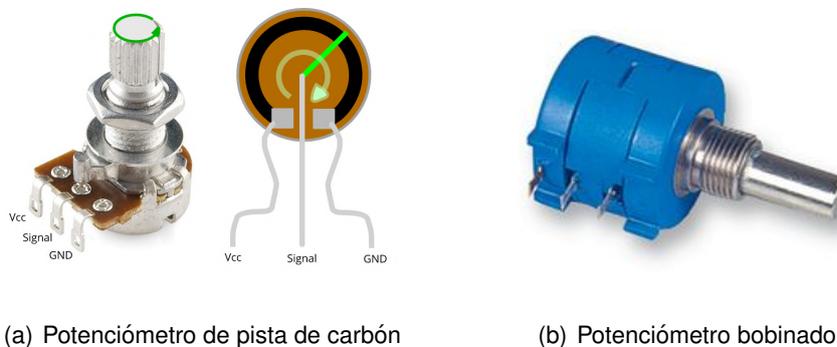


Figura 7.6 – Potenciómetros *resistivos*

- **Magnéticos:** También llamados de efecto hall, miden el campo magnético de un imán, no tienen partes en rozamiento de forma que se puede aislar la parte fija de la móvil. Pueden tener errores de medida debidos a la posición del imán o a interferencias electromagnéticas debidas a cables o motores. Suelen traer integrado un microcontrolador que transforma la lectura de campo magnético en un valor digital, normalmente transmitido por I2C, SPI o PWM, pueden ser incrementales o absolutos, los incrementales indican un desplazamiento a partir de un origen cada vez que reciben alimentación y los absolutos mantienen ese origen aunque dejen de recibir alimentación. El rango de resolución por vuelta suele estar entre 8 y 14 bits.

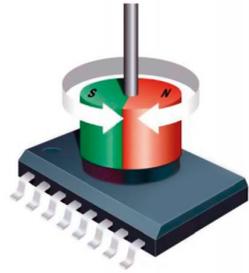


Figura 7.7 – Encoder magnético

- Ópticos:** Emiten luz mediante un led que atraviesa un disco perforado, en el otro lado del disco se ubica un sensor de luz, de forma que cuando el disco gira el sensor recibe pulsos, pueden ser incrementales o absolutos, siendo los incrementales los más comunes. El rango de resolución por vuelta suele estar entre 1000 y 8000 ppr. La salida suele ser en pulso de cuadratura, por lo que es necesario cierta electrónica para analizar la señal.

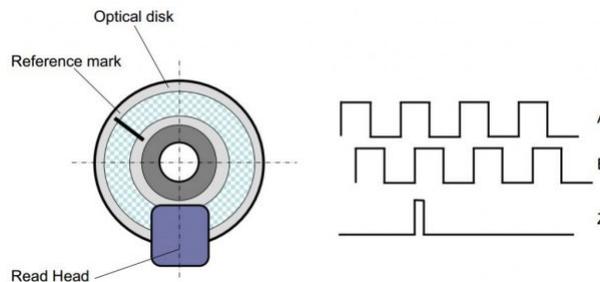


Figura 7.8 – Encoder óptico

7.2.3.1. Comparación

A continuación se comparan en 3 áreas principales:

7.2.3.1.1. Economía: Los resistivos son los más económicos, especialmente los que usan una película de carbón, los hay con diferentes valores, siendo preferible escoger la resistencia más alta posible, así la intensidad que circula por el sensor será mínima y tanto la vida útil, como la precisión, serán las máximas posibles.

7.2.3.1.2. Fiabilidad: Los ópticos son los más estables, no se puede alterar la señal percibida por el sensor, ya que estos pueden ir aislados del exterior y no se ven afectados por inducciones electromagnéticas.

7.2.3.1.3. Facilidad de comunicación: Los magnéticos suelen traer su propio circuito integrado, analizando la señal magnética y enviándola a través de un protocolo estándar de comunicación, es por ello que son los más cómodos de usar, además, debido a que el imán y el sensor están distanciados, suelen tener mayor tolerancia a desalineaciones debidas al montaje.

7.2.4. Controlador

Para controlar el robot hay 2 caminos principales, directamente desde un ordenador o a través de un periférico, se puede ver la comparación en la tabla 7.6.

El principal inconveniente del ordenador es su implementación, para utilizar los sensores y los drivers de motores se debe adquirir algún tipo de pasarela física entre los puertos de comunicación y el driver/sensor. Sin embargo, los periféricos como Arduino y similares facilitan la implementación de todo tipo de proyectos enfocados en la electrónica. Siendo suficientes para controlar y leer casi todo tipo de señales.

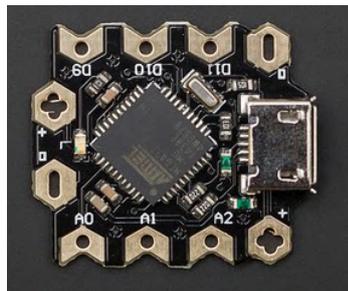
Tipo	Ordenador	Periférico
Potencia cálculo	Alta	Baja
Implementación	Difícil	Fácil
Precio	Alto	Bajo

Tabla 7.6 – Comparación controladores

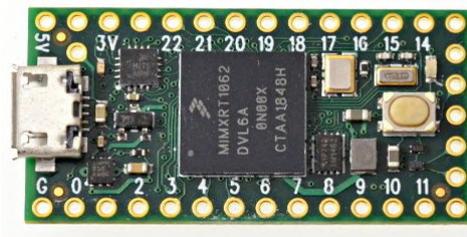
Por otra parte, el ordenador ofrece generalmente una seguridad que los periféricos no han alcanzado todavía, eso sumado a la potencia de cálculo hace que en instalaciones complejas se siga utilizando este tipo de controlador.

En este trabajo, debido a la baja complejidad y optimización de precio se optará por el uso de periféricos. Además, este tipo de productos cuentan con el apoyo de una comunidad de usuarios que desarrolla librerías y módulos para incrementar las capacidades y usos.

Existen numerosos modelos en función de lo que se quiera realizar, desde el modelo actual más pequeño, el Beetle, de tan solo 20x22mm y un reloj de 16MHz hasta el Teensy 4.0 con 600MHz.



(a) Beetle



(b) Teensy 4.0

Figura 7.9 – Microcontroladores

7.3. Software

7.3.1. Script

El último requisito es diseñar un script para poder controlar el manipulador, este se analizará en detalle en el apartado resultados finales pero se pueden sentar las bases para que después sea más sencillo.

- El encargado de gestionar los drivers de los motores y los sensores será una placa de desarrollo Teensy. Debido a la mayor potencia hardware.
- El script se hará en Visual Studio Code, ya que es más cómodo de usar que el IDE de Arduino y tiene una extensión dedicada a la programación de sistemas embebidos llamada PlatformIO.
- El cálculo de los ángulos necesario para cada eje se hará en un ordenador, a partir de ahí se comunicará mediante comunicación serial al Teensy, este canal de comunicación estará siempre activo, informando de los ángulos del manipulador.
- El script tendrá la opción de poder mover los motores en cadena abierta, en un espacio tridimensional suministrando los ángulos para cada articulación o a través de un camino previamente guardado, pudiendo almacenar una serie de movimientos.

7.3.2. Cinemática

Para hacer un correcto control de un manipulador se necesita estudiar la cinemática, esta depende de la morfología que tiene cada eslabón del robot.

Principalmente existen 2 estudios, la directa e inversa, la directa permite calcular el punto y orientación del TCP en un espacio tridimensional a partir de los ángulos y longitudes de los diferentes eslabones del robot, por el contrario, la inversa, permite el cálculo de los ángulos a partir de un punto y orientación en el espacio tridimensional.

Para realizar estos cálculos se suelen utilizar los parámetros de Denavit-Hartenberg [44], que permiten describir mediante una matriz esta geometría.

Para calcularla se han de multiplicar una serie de matrices como la de la figura 7.10 para cada eslabón.

$$\mathbf{A}_i = \left(\begin{array}{ccc|c} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

Figura 7.10 – Matriz DH

Siendo:

θ_1 : El ángulo desde X_{i-1} hasta X_i girando alrededor de Z_i .

α_1 : El ángulo que hay que rotar Z_{i-1} hasta Z_i a lo largo de Z_{i-1} .

a_i : La longitud de la normal común.

d_i : La distancia desde el sistema XYZ_{i-1} hasta la intersección de las normales común de Z_{i-1} hacia Z_i a lo largo de Z_{i-1} .

En este trabajo, debido a las articulaciones que tiene el robot manipulador, se puede optar por otro camino, el de implementar un script de Python realizado por R. Venkata y R. Sreenivasulu. [17]

Este código es capaz de calcular los ángulos para un manipulador de 2 y 3 grados de libertad de revolución teniendo como entradas la coordenada XY más la orientación de la pinza. Obteniendo, si existen, las 2 posibles soluciones, con el codo hacia abajo o hacia arriba.

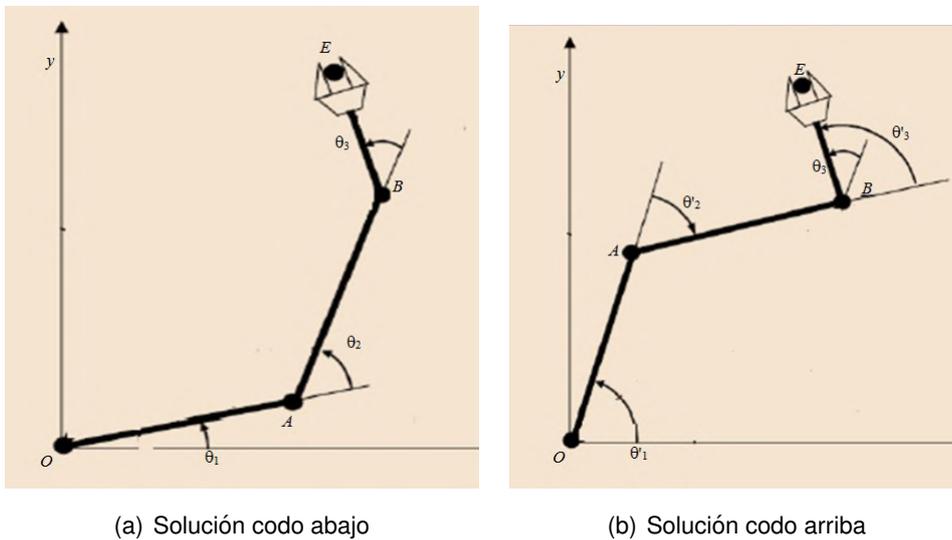


Figura 7.11 – Posibles soluciones

8 RESULTADOS FINALES

En este apartado se mostrarán los resultados a las posibles soluciones analizadas en el apartado anterior, divididas en 3 grupos:

- **Mecánica:** Donde se describirá la forma de cada una de las piezas, los rodamientos, las poleas y correas.
- **Electrónica:** Con división entre la parte de motorización y sensorización.
- **Software:** Donde se explicará grosso modo el funcionamiento del script, las librerías utilizadas y el algoritmo de cinemática inversa.

8.1. Mecánica

Usando el programa de diseño CAD 3D Autodesk Inventor Profesional, que tiene licencia gratuita para estudiantes, se puede diseñar cualquier tipo de pieza, además, ya que este programa tiene integrado un simulador de elementos finitos, se puede analizar la respuesta de una pieza ante determinados esfuerzos. Esto ayuda a optimizar la forma y poder ver la respuesta en función del material escogido.

La mayoría de piezas se comprarán en la tienda online Misumi, un proveedor japonés de alta calidad que dispone de infinidad de piezas para automatización. [40]

8.1.1. Morfología

El modelo final es el siguiente:

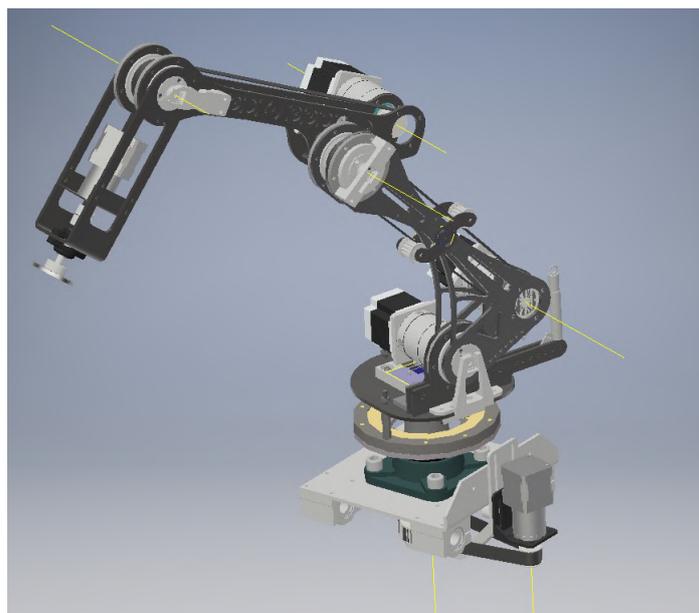


Figura 8.1 – Modelo Final Manipulador

■ Base fija

Servirá de base del manipulador, permitiendo el giro horizontal de la base móvil con el menor esfuerzo posible. Dispondrá de 2 rodamientos de bolas, que irán integrados en unos soportes cuadrados realizados en hierro de fundición, estos rodamientos serán el modelo CHDF30.

Como posible accesorio futuro se colocarán 4 rodamientos lineales SC16UU que permitirán disponer el manipulador sobre unas guías, añadiendo gran capacidad de maniobra al permitir moverse de manera prismática.

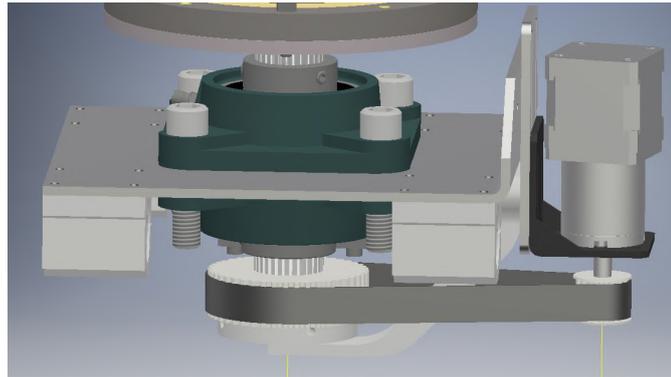


Figura 8.2 – Modelo Base Fija

■ Base

Nexo entre la base fija y el cuerpo, tendrá un eje de 30mm de diámetro donde irá acoplada una polea, esta irá unida con una polea al motor encargado del giro horizontal.

La pieza que une el eje con la base es CSTA30-190. Se aprovechará el uso de poleas para realizar una reducción de velocidad, esto permitirá tener más precisión y aumentará el torque disponible en la base.

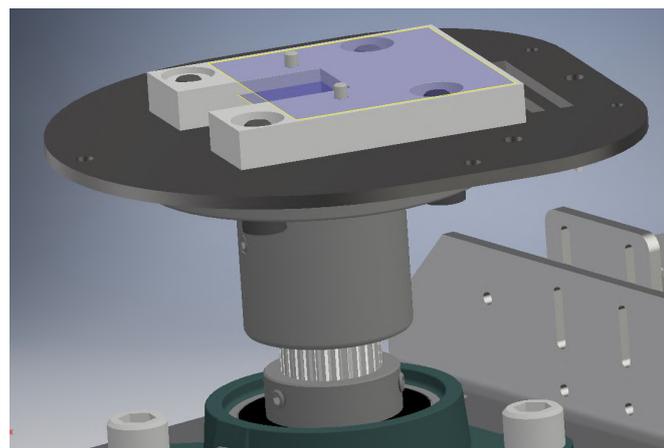


Figura 8.3 – Modelo Base Móvil

■ Cuerpo

Encargado de la unión del brazo con la base, tendrá un rodamiento IGUS PRT-02-20-AL-1 [41]. Este tipo de rodamiento no lleva lubricación, está fabricado en aluminio y un polímero, Iglidur J4, que tiene bajo coeficiente de rozamiento, permitiendo que una vez instalado no requiera mantenimiento.

Además esta pieza tendrá un soporte para un muelle que servirá para contrarrestar el momento generado por el brazo cuando se extiende hacia delante, esta pieza se detallará más adelante en el apartado electrónica.

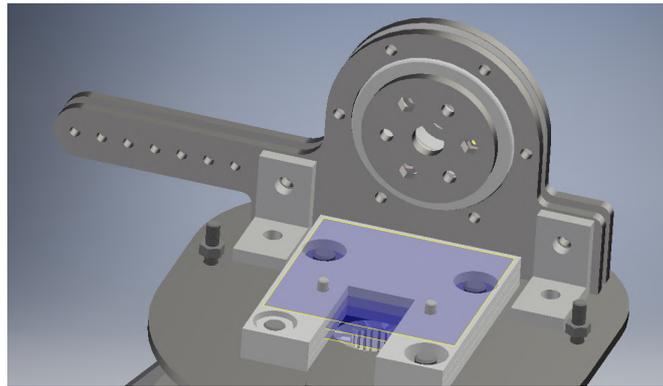


Figura 8.4 – Modelo Cuerpo

■ Brazo

Será la unión entre el cuerpo y el antebrazo, será la primera pieza a optimizar, debido a que cualquier reducción de peso mejorará el control y reducirá costes. Este proceso se detalla en la sección anexos 13.1.2. Para intentar acercar el centro de gravedad lo máximo posible a la base, se desplaza el motor encargado de mover el antebrazo como se puede ver en la figura:

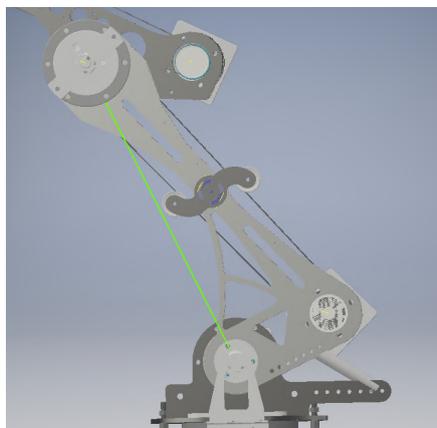


Figura 8.5 – Modelo Brazo

La articulación que sirve de soporte al antebrazo estará constituida por 2 rodamientos IGUS PRT-02-20-AL-1. El motivo de llevar 2 es que ayuda a que el eje de esa articulación sea más estable.

El brazo dispondrá de un tensor para regular la correa, una correcta tensión es necesaria para tener un buen control del movimiento. Ambos detalles se pueden observar marcados en azul en la siguiente figura:

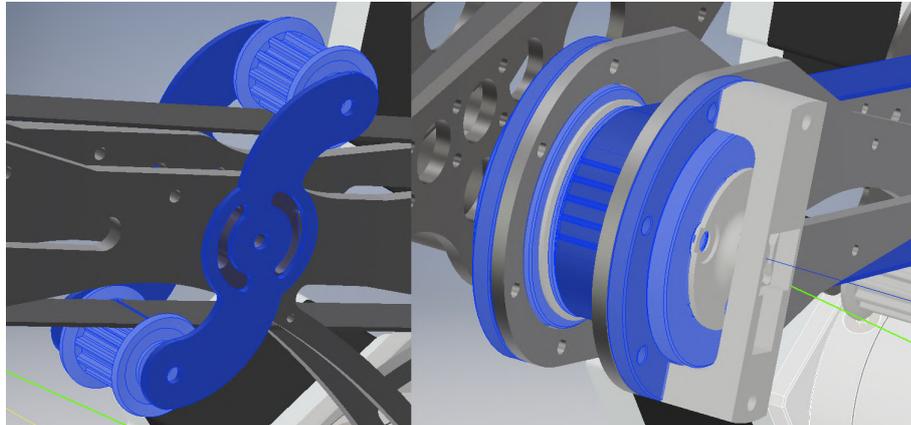


Figura 8.6 – Detalle Tensor y Rodamiento

Se aprovechará el uso de poleas para realizar una reducción de velocidad, esto permitirá tener más precisión y aumentará el torque disponible en el antebrazo.

■ **Antebrazo**

Similar estructuralmente al brazo, es el nexo entre la muñeca y el brazo.

Se retrasa la posición del motor encargado de girar la muñeca para realizar función de contrapeso, ayudando al control del movimiento, se obtendrá una reducción en el movimiento usando las poleas adecuadas.

Se optimizará el peso para hacerlo lo más ligero posible.

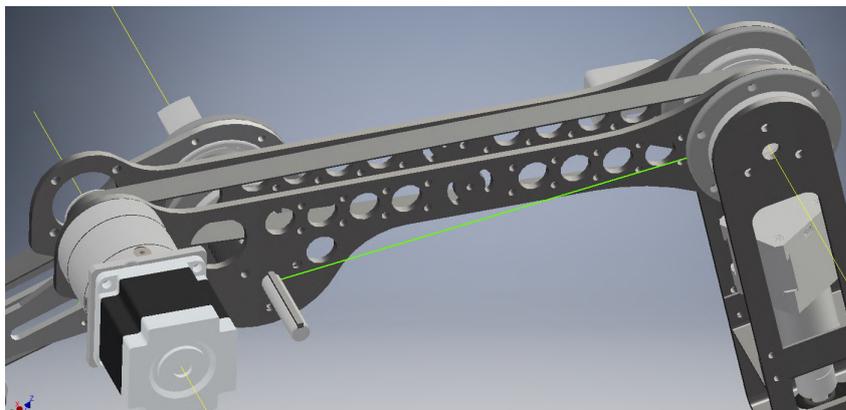


Figura 8.7 – Modelo Antebrazo

■ Muñeca

Servirá como soporte de la herramienta, siendo capaz de rotar de manera continua tanto en sentido horario como antihorario, tendrá 2 cojinetes EFOM-12-HT asegurando la estabilidad del eje que une el motor con la brida de la herramienta.

Se optimizándose su forma para que pese lo menos posible.

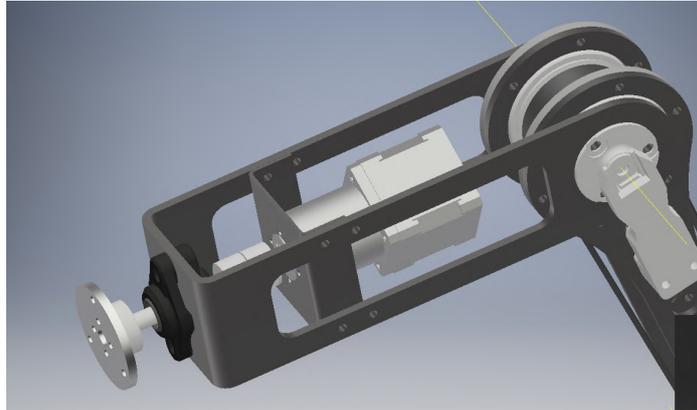


Figura 8.8 – Modelo Muñeca

Debido al uso de acero se debe optimizar el peso lo máximo posible, para ello se utiliza el simulador de esfuerzos del Autodesk Inventor, esta técnica solo es necesaria en las piezas brazo, antebrazo y muñeca, se detalla el procedimiento en el anexo de optimización y simulaciones 13.1:

8.1.2. Herramienta

Habiendo analizado las diferencias entre herramientas accionadas de manera neumática y eléctrica se opta por las eléctricas, el motivo es la simplificación del diseño.

La herramienta neumática requiere de un compresor, con su driver y sensor de presión para no dañar la herramienta, además no permiten un control del grado de apertura de la pinza.

En cuanto a herramientas eléctricas, en adelante pinzas, ya que su función será la de coger objetos, existen 2 variantes principales, rígidas y flexibles, la principal diferencia, como su nombre indica, es la capacidad de deformarse y amoldarse al objeto que se quiere manipular, mientras que en las rígidas debemos tener especial cuidado en el grado de apertura y cierre, en las flexibles tenemos grandes márgenes de seguridad, esto se entiende de manera más sencilla en la siguiente figura:

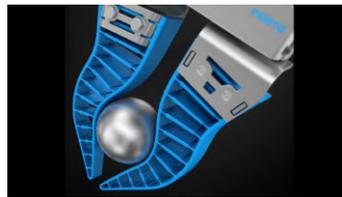


Figura 8.9 – Gripper flexible

Permitiendo agarrar diferentes objetos sin tener en cuenta su forma ni tamaño.

De todas las disponibles en el mercado se ha optado por el suministrador por la empresa robotdigg, ya que tiene 3 terminales y un motor paso a paso integrado en la pinza, de forma que la instalación es la más sencilla posible.



Figura 8.10 – Gripper flexible final

8.2. Electrónica

8.2.1. Motorización

Con lo analizado anteriormente y viendo como los motores paso a paso siguen siendo los más usados en robótica se decide que estos sean los que controlen el manipulador.

De todas formas cabe destacar que, desde comienzos del año 2018, se está incrementando el uso de motores BLDC, con sensores de posición integrados, ofertando drivers que tienen de serie control en bucle cerrado, haciendo que sean altamente precisos, aprovechándose de su gran relación potencia-peso, la decisión de escoger los motores paso a paso frente a los BLDC es debida a la abundancia de modelos en el mercado y al coste de los mismos, ya que, a misma potencia, los BLDC suelen ser el doble de caros.

Se van a utilizar motores paso a paso del fabricante stepper motor [23], se utilizará motores Nema23 y Nema17, ya que son los más comunes, los Nema 17 irán para las articulaciones base y rotación de muñeca, mientras que los Nema23 irán en hombro, codo y elevación de muñeca.

Para poder elegir las reductoras pertinentes se debe hacer una estimación del peso que tendrá que mover cada uno, midiendo la distancia desde el eje hasta el final de la articulación. Así se decide el par necesario en la salida de la reductora, como el manipulador está modelado en CAD se puede saber el peso exacto de cada pieza. El orden correcto es empezar desde el motor de rotación de la muñeca hasta la rotación de la base, así se tiene en cuenta el incremento de peso debido a los propios motores. Para hacer los cálculos se usa la ecuación del momento, que dice lo siguiente:

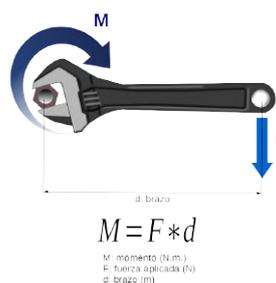


Figura 8.11 – Momento

- **Motor rotación de la muñeca**, este motor no tendrá que soportar mucho par, ya que el movimiento es en su mismo eje y lo que tendrá que mover es la herramienta más el objeto, debido a que la pinza tiene un diámetro máximo de apertura de 85 mm, el peso de la misma es 750g y el peso del objeto puede ser de hasta 1000g, el momento generado es prácticamente 0.

Esto es debido a que se supone que el peso de la pinza y del objeto esté uniformemente

repartido, como esto no se cumple, pero sería muy costoso de calcular, se supone la menor reductora disponible. Viendo el catálogo la menor es 5, por ello el modelo, que ya tiene ensamblada la reductora es 17HS19-1684S-PG5. Con un par de salida máximo de 1.55Nm.

- **Motor muñeca**, tendrá que soportar el peso del conjunto anterior más la estructura de la muñeca, siendo la suma 3kg, la distancia desde la articulación al centro de gravedad de este conjunto está a 0.23 metros, por ello el par generado es de:

$$29,43N \times 0,23m = 6,77Nm$$

A esto tenemos que añadir las pérdidas por rozamiento de los rodamientos IGUS y las poleas, en este caso casi despreciables, ya que en los datos de los rodamientos no aparecen cargas tan pequeñas y se considera que las poleas tienen un rendimiento normal de 0.95. Por cuestiones de seguridad se multiplica por 1.25, este factor de seguridad, hallado de manera empírica, asegura que el motor trabaje de forma cómoda sin estar excesivamente sobredimensionado.

Por ello se escoge un Nema 23 con reductora de 15, ofreciendo un par de salida de 9Nm, a esto se le añade una pequeña reductora utilizando la relación de polea-motor y polea-articulación de 18/32, consiguiendo así un par en la muñeca de 16Nm.

- **Motor antebrazo**, tendrá que soportar el peso del conjunto anterior más la estructura del antebrazo, siendo la suma 4.3kg, la distancia desde la articulación al centro de gravedad de este conjunto está a 0.3 metros, por ello el par generado es de:

$$42,18N \times 0,35m = 14,76Nm$$

Como en el caso anterior, las pérdidas por rodamientos y poleas son prácticamente despreciables, multiplicando por 1.25 del coeficiente de seguridad se obtiene 18.45 Nm.

Por ello se escoge un Nema 23 con reductora de 47, ofreciendo un par de salida de 25Nm, a esto se le añade una pequeña reductora utilizando la relación de polea-motor y polea-articulación de 25/32, consiguiendo así un par en la muñeca de 32Nm.

- **Motor brazo**, soportará el peso del conjunto anterior más la estructura del brazo, siendo la suma 8kg, la distancia desde la articulación al centro de gravedad de este conjunto está a 0.45 metros, por ello el par generado es de:

$$78,48N \times 0,45m = 35,32Nm$$

Con las pérdidas en rodamientos y poleas prácticamente despreciables, multiplicando por 1.25 del coeficiente de seguridad se obtiene 44.14 Nm.

Este par necesario sobrepasa el máximo disponible con el motor Nema23 más potente disponible, que ofrece 25Nm, por ello se recurre a la solución de contrarrestar ese momento generado, en este caso mediante el uso de muelles. En el apartado de mecánica

se comentaba que en la base se había añadido un saliente para sujetar el muelle, la estructura quedaría así:

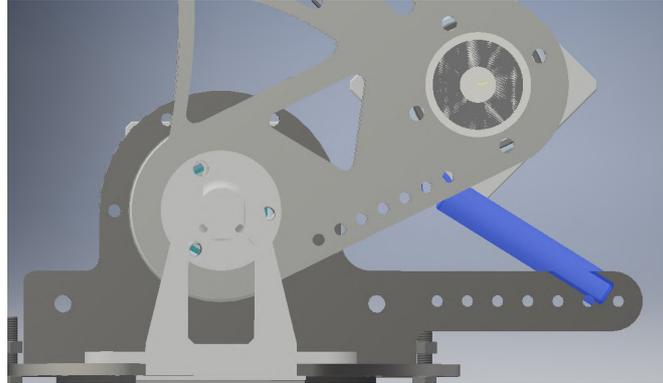


Figura 8.12 – Detalle muelle

Para el cálculo del muelle se tiene la ecuación característica del muelle:

$$F = KxX$$

Siendo:

F, la fuerza en Newtons

K, la contante del muelle, propia de cada tipo

X, el desplazamiento del muelle

Como la distancia entre el soporte del muelle y la articulación es de 70mm el muelle debería tener una fuerza al estar estirado de:

$$20Nm = F_{muelle} \times 0,07m \Rightarrow 285N$$

El uso de esta ecuación no es del todo correcto, debido a que el movimiento del muelle no es lineal, de todas formas sirve como buena aproximación al resultado, haciendo mucho más sencillo el cálculo.

En este caso, no se encontraron muelles en Misumi que ofrezcan esa fuerza con un tamaño reducido, por ello se colocarán 2 en paralelo, siendo el modelo elegido el AUT12-80, ofreciendo cada muelle 120N, por tanto una suma de 240N, para poder alcanzar el par necesario se coloca el motor que moverá el antebrazo por detrás de la articulación del hombro. Llegando así al par necesario.

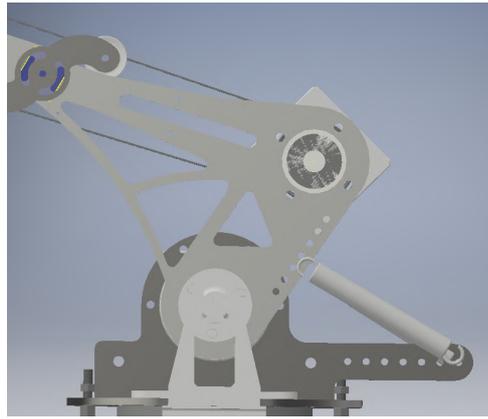


Figura 8.13 – detalle contrapeso motor

- **Motor base**, debido al uso de los rodamientos CHDF30 prácticamente no existen pérdidas, el par necesario es casi despreciable, por tanto con un Nema17 con baja reductora sería más que capaz, debido a que es la articulación más alejada del TCP [42] se necesita que sea muy preciso, ya que un mínimo movimiento en la base equivale a un gran movimiento de la pinza, por ello se recurre a una reductora de 19, además se utilizará una reductora hecha con las poleas de 15/44.

Como resulta más económico comprar 2 motores iguales se aumenta la reductora de la rotación de la pinza a 19, ya que el peso es prácticamente el mismo.

8.2.2. Drivers de motores

En cuanto a los drivers se utilizarán los recomendados por el fabricante stepperonline, el modelo DM542T, que funciona por control de corriente, aprovechando todo el potencial del motor paso a paso, además tiene una configuración mediante interruptores para elegir la intensidad de cada motor y una función de microstepping, en nuestro caso innecesaria ya que el uso de esta técnica reduce de manera drástica el torque del motor.



Figura 8.14 – Driver

8.2.3. Sensorización

Repasando el análisis de soluciones y teniendo en cuenta que se desea utilizar sensores absolutos, debido a la sencillez del montaje, se escogen los sensores magnéticos, el motivo es la mezcla entre economía, precisión y facilidad de comunicación.

Son hasta 10 veces más baratos que los ópticos, sobretodo cuando se entra en la categoría de sensores absolutos, suelen ser programables y más precisos.

Dentro de los sensores magnéticos destacan los del fabricante AMS, buscando en su catálogo entre 12 y 14 bits de resolución se encuentran los AS5600, se pueden encontrar hasta por 3€. Estos sensores son de 12 bits, lo que dan un rango de 4096 posiciones, equivalente a 0.087 grados en un círculo de 360.

Se pueden programar indicando una posición de origen y una final, siempre que esta sea mayor de 15 grados, obteniendo los 4096 valores en ese tramo, de forma que aumenta la resolución de los 0.087 grados iniciales.

Se comunican por PWM o por I2C, debido a que estarán a una distancia máxima de 1.5m del microcontrolador se opta por I2C, ya que ofrece una señal más estable.

Para la transmisión de las señales se elige manguera apantallada de 6 cables de $0,08mm^2$. La conexión con el encoder se realiza con unos conectores micro jst de 6 pines, siendo el resultado final el siguiente:

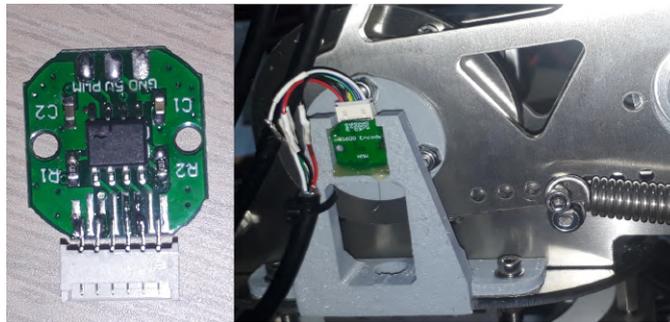


Figura 8.15 – Encoders AS5600

Para hacer la lectura de la posición existe una librería para Arduino que facilita su uso, es una simple compilación del manual que se encuentra en la web de AMS en la que se describen las funciones que puede realizar, dentro de la librería estas funciones se ponen en contacto con el encoder vía I2C y acceden a los registros pertinentes donde se guarda la información. Un dato importante es que la programación de valores iniciales y finales solamente se puede realizar 3 veces, una vez llegado a este límite el integrado impide una nueva configuración.

A pesar de tener comunicación I2C la dirección en estos encoders es única e invariable, haciendo imposible el uso de un bus único para todos los sensores, ante esto existen 2 opcio-

nes, añadir microcontroladores en cada encoder y crear entre ello un bus de comunicación o utilizar un multiplexor donde conectar todos los encoders, en este caso se ha decidido por el multiplexor, ya que simplifica el diseño.

Este multiplexor es el modelo TCA9548A, permite la conexión de hasta 8 dispositivos I2C, se ha escogido el modelo que viene ya soldado a su propia placa, pudiendo conectarse con el Teensy en una placa de prototipado.

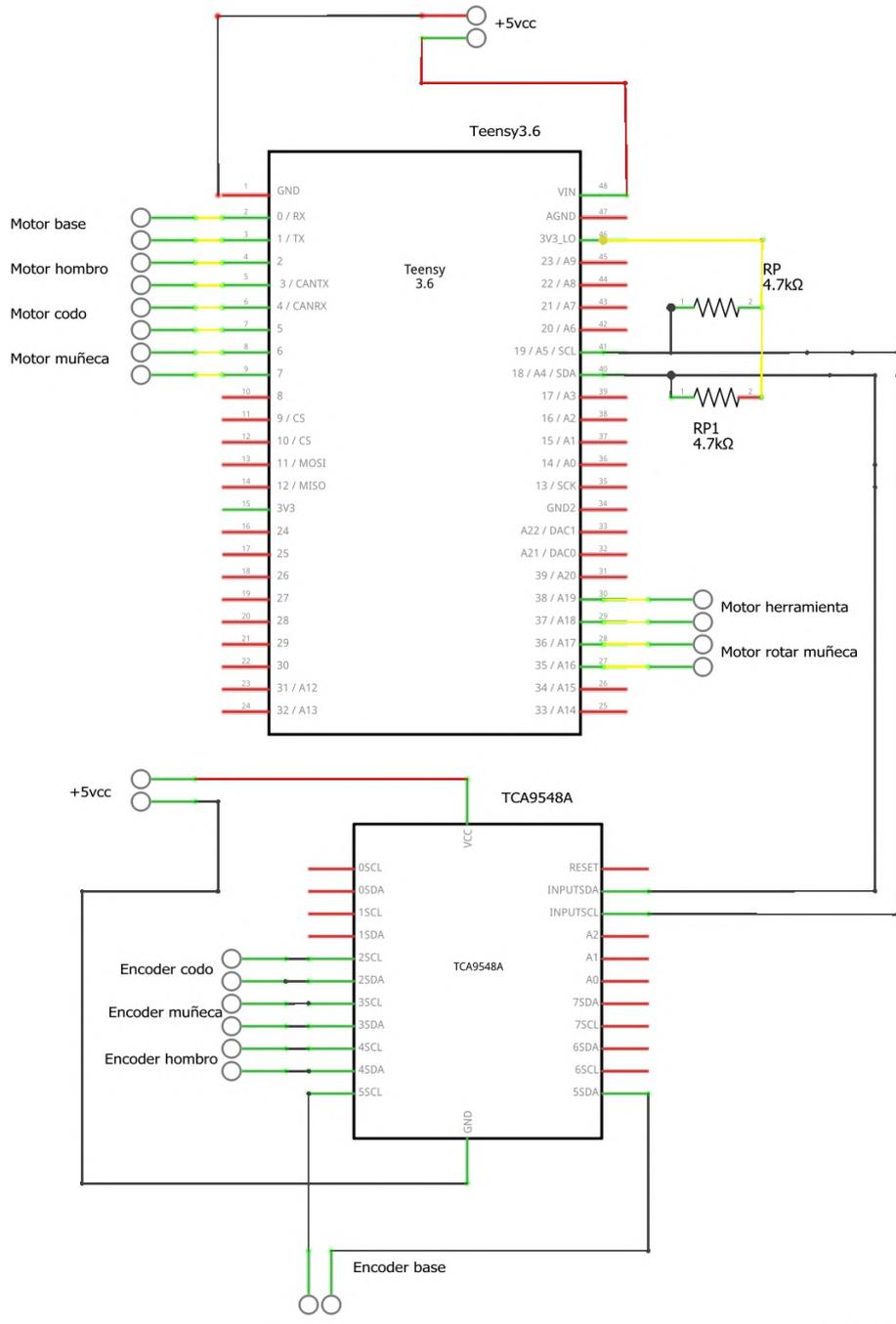


Figura 8.16 – TCA9548A

8.2.4. Controlador

Este script funcionará en Teensy, es una placa de desarrollo similar a la conocida Arduino, con unas especificaciones superiores. Se puede programar en C, Python, LabVIEW o como un Arduino más, haciendo muy fácil el uso de librerías. En este proyecto se usa el modelo 3.6, teniendo como características principales un reloj de 180MHz, 25 entradas analógicas con convertidor A/D de 13 bits, 22 salidas PWM, 2 salidas analógicas de 12 bits y 1M de memoria flash.

El esquema de conexiones se detalla en la figura 8.17. Para la alimentación de los encoders existen 2 opciones, 5 o 3v, de serie vienen preparados para 5v. Es importante recordar conectar las masas de la alimentación del teensy con la de los encoders, si esto no se hiciera podrían aparecer errores en las lecturas.



fritzing

Figura 8.17 – Esquema conexiones

8.3. Software

8.3.1. Librerías

Para realizar el script que controle el manipulador se han utilizado las siguientes librerías:

- **Wire.h**, que permite el uso de los puertos I2C con una serie de funciones básicas.
- **Arduino.h**, necesaria en PlatformIO para programar tarjetas tipo Arduino o, como en este caso, Teensy, aceptando funciones propias del IDE de Arduino.
- **AMS5600.h**, facilita el uso de los encoders magnéticos evitando que se tengan que indicar que registros del encoder hay que leer/escribir en cada momento.
- **TeensyStep.h**, permite el control de motores paso a paso, incluyendo las siguiente propiedades:
 1. Algoritmo de Bresenham para calcular la línea recta entre dos puntos de un plano, de forma que simplifica enormemente el cálculo de trayectorias.
 2. Posibilidad de fijar posiciones objetivo tanto relativas como absolutas.
 3. Modificación en todo momento de las curvas de aceleración y velocidad de cada motor.
 4. Posibilidad de crear agrupaciones de motores, de forma que la librería ya calcula como de rápido debe moverse cada motor para que todos alcancen el objetivo al mismo tiempo.
 5. Capacidad de realizar movimientos asíncronos, esto es, que puede recibir la orden de moverse a una posición y en pleno movimiento recibir otra que sobrescribirá la anterior, permitiendo mucha agilidad.
 6. Conexión con el driver en 2 pines, uno para el control de los pasos y otro para el control de la dirección, trabajando de manera excelente con los drivers de steppeonline.

8.3.2. Código

A la hora de diseñar la forma del script 11.1 se ha pensado en comunicación serie desde el USB de un ordenador, pero podría ser a través de cualquier otro medio, como ethernet, bluetooth o wifi, en este caso se le envía la orden mediante un string, que contiene el tipo de orden a ejecutar y el objetivo de cada motor. Estos 3 tipos de orden son los siguientes:

1. **Manual**, indicando el número de pasos que cada uno de los 6 motores tiene que dar, este valor de pasos es absoluto, por tanto la librería compara la posición actual, el objetivo y ya ejecuta la orden correspondiente.

Ej: "Manual,1000,500,200,1500,4000,500"

2. **Manual**, indicando el ángulo objetivo para las articulaciones de la base, hombro, codo y muñeca, los 2 motores restantes, rotación de la muñeca y pinza, solo se pueden manipular mediante pasos objetivo, debido a que no disponen de encoder, esto es porque no es condición necesaria para poder controlar el manipulador de manera precisa.

Ej: "Angulo,180,75.45,50,45.8,2000,500"

3. **Automático**, mediante un camino guardado a la hora de cargar el programa, se le indica una posición de inicio, para hacer más sencillo el control y después irá recorriendo las posiciones automáticamente, entrando en un bucle para cada posición en la que el robot se asegura que está en la posición adecuada antes de moverse al siguiente punto, leyendo la información de los encoders y comparándola con la objetivo.

Ej: "Path, 230,80,50,64,0,0"

La máquina de estados para este programa sería la siguiente:

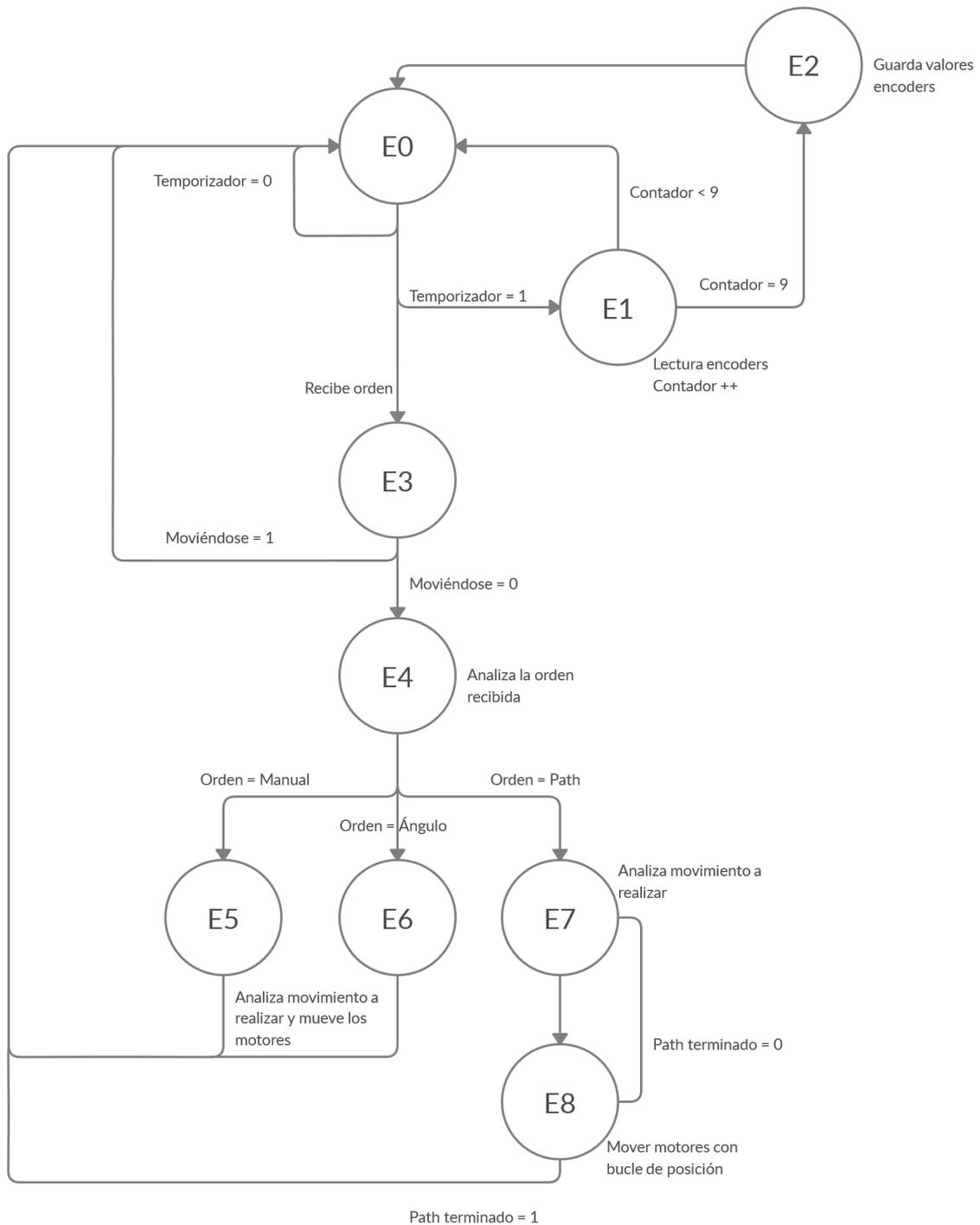


Figura 8.18 – Máquina de estados

Siendo:

- **E0** : Estado inicial del sistema, en bucle permanente hasta que se recibe una orden o salta el temporizador para leer los encoders.

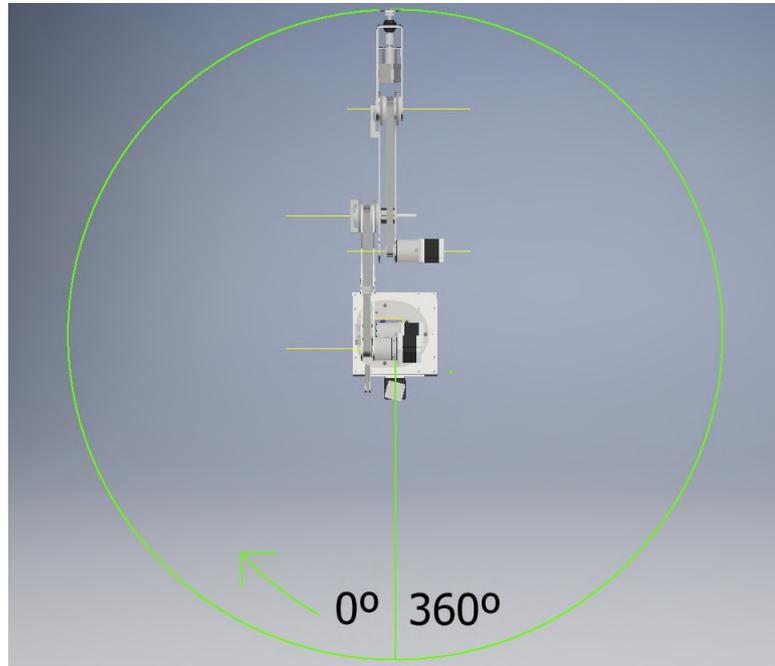
- **E1** : Momento en el que salta el temporizador, cada 10ms, se hace una lectura de los encoders, debido a la Histéresis en el sensor se realiza una media de 10 lecturas, por ello si el contador no está en 9 vuelve al E0.
- **E2** : Si el contador es 9 entonces se calcula la media y se actualizan las variables estables de los encoders, que serán con las que trabaje el programa.
- **E3** : Analiza si actualmente se está moviendo o si no, si está parado procede a ejecutar el comando, si está en movimiento avisa imprimiendo un mensaje en pantalla. Esta opción podría eliminarse y que siempre obedezca, permitiendo el cambio de trayectoria en movimiento.
- **E4** : Analiza la orden recibida, en función de si es manual suministrando el número de pasos que debe dar cada motor, manual suministrando los ángulos a los que debe ir cada articulación o automática suministrando los ángulos de las articulaciones, las primeras son para control en lazo abierto y la tercera para control en lazo cerrado.
- **E5** : Analiza la cantidad de pasos recibida por serial y mueve los motores.
- **E6** : Calcula la diferencia entre el ángulo actual y el recibido por serial y mueve los motores.
- **E7** : Mueve los motores calculando la diferencia entre el ángulo actual y el recibido por serial, además entra en bucle para asegurarse que llega a la posición.
- **E8** : Bucle cerrado para confirmar que ha llegado a la posición correcta, los márgenes de seguridad se pueden configurar en cualquier momento, permitiendo movimientos mas o menos precisos.

8.3.3. Cinemática

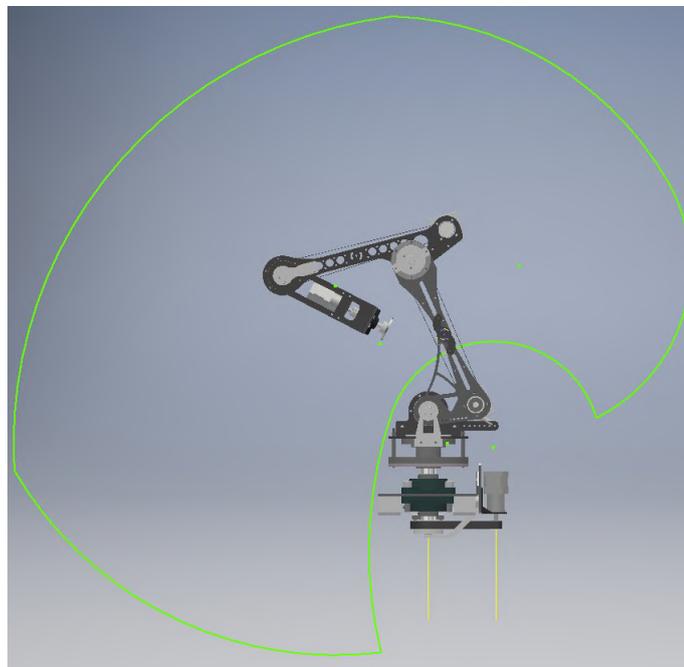
La solución que se plantea en este trabajo es usar el script de Python 11.2, el motivo es la simplicidad, ya que no se tiene que calcular ninguna matriz ni utilizar software específico como MATLAB o similar, además la comunicación entre Python y Teensy es muy sencilla, pudiendo en un futuro comunicar ambos y controlar el manipulador completamente desde Python.

Para poder calcular la solución se debe dividir el volumen de trabajo del robot en 2 planos, Z y XY 8.19, utilizando el script mencionado para resolver el posicionamiento de las articulaciones hombro, codo y muñeca. Para la solución del plano Z, que solo plantea la rotación del

motor de la base, se indica el ángulo con respecto al origen, ya que el robot puede girar 360 grados.



(a) PlanoZ



(b) PlanoXY

Figura 8.19 – Área de trabajo del robot

9 ORDEN DE PRIORIDAD ENTRE LOS DOCUMENTOS

Frente a posibles discrepancias, se debe establecer el orden de prioridad de los documentos del TFG. Si no se especifica otra cosa, el orden de prioridad debe ser el siguiente:

- 1 Planos
- 2 Pliego de Condiciones
- 3 Presupuesto
- 4 Memoria
- 5 Anexos

TÍTULO: Diseño, montaje y control de un manipulador robótico.

ANEXOS

PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: FEBRERO DE 2020

AUTOR: EL ALUMNO

Fdo.: ANTONIO REGUEIRA PEÑAMARÍA

Índice del documento ANEXOS

10 DOCUMENTACIÓN DE PARTIDA	71
11 Códigos de programación	74
11.1 Código Teensy	74
11.2 Código Python	82
12 ANEXOS EN FUNCIÓN DEL ÁMBITO DE APLICACIÓN DEL TFG	83
12.1 Seguridad	83
13 OTROS ANEXOS	84
13.1 Simulación y optimización de piezas	84
13.1.1 Simulación	84
13.1.2 Optimización	91
13.1.2.1 Brazo	91
13.1.2.2 Antebrazo	92
13.1.2.3 Muñeca	93
13.2 Ensamblaje del robot	94
13.2.1 Base	94
13.2.2 Cuerpo	94
13.2.3 Brazo	95
13.2.4 Antebrazo	95
13.2.5 Muñeca	96
13.3 Montaje y ensayo	97
13.3.1 Montaje	97
13.3.2 Ensayo	102

10 DOCUMENTACIÓN DE PARTIDA



ESCUELA UNIVERSITARIA POLITÉCNICA

ASIGNACIÓN DE TRABAJO FIN DE GRADO

En virtud de la solicitud efectuada por:

En virtud da solicitude efectuada por:

APELLIDOS, NOMBRE: Regueira Peñamaría, Antonio

APELIDOS E NOME:

Fecha de Solicitud: OCT2019

Fecha de Solicitude:

Alumno de esta escuela en la titulación de Grado en Ingeniería en Electrónica Industrial y Automática, se le comunica que la Comisión de Proyectos ha decidido asignarle el siguiente Trabajo Fin de Grado:

O alumno de esta escola na titulación de Grado en Enxeñería en Electrónica Industrial e Automática, comunícaselle que a Comisión de Proxectos ha decidido asignarlle o seguinte Traballo Fin de Grado:

Título T.F.G: Diseño, montaje y control de un manipulador robótico.

Número TFG: 770G01A189

TUTOR:(Titor) Jove Pérez, Esteban

COTUTOR/CODIRECTOR: José Luis Calvo Rolle

La descripción y objetivos del Trabajo son los que figuran en el reverso de este documento:

A descrición e obxectivos do proxecto son os que figuran no reverso deste documento.

Ferrol a Domingo, 2 de Febrero del 2020

Retirei o meu Traballo Fin de Grado o día _____ de _____ do ano _____

Fdo: Regueira Peñamaría, Antonio

DESCRIPCIÓN Y OBJETIVO:OBJETO:

El objeto de este Trabajo Final de Grado es el de diseñar e implementar físicamente un manipulador robótico. Para ello se analizará el uso de diferentes morfologías, sensores y actuadores. Con el objetivo de manejar el manipulador, se elaborará un software de control que será testado sobre el montaje físico.

ALCANCE:

Análisis de los diferentes tipos de manipuladores.

Estudio de las posibles motorizaciones.

Definición de carga máxima y del espacio de trabajo.

Estudio de posibles herramientas, accionadas de manera eléctrica y neumática.

Realimentación de la posición de cada una de las articulaciones.

Evaluación de la cinemática del manipulador.

Análisis de los resultados obtenidos y extracción de conclusiones.

11 Códigos de programación

11.1. Código Teensy

Código 11.1: Código Teensy

```
#include <Arduino.h> // libreria necesaria en PlatformIO

#include <Wire.h> // libreria I2C
#include "AMS5600.h" // libreria Encoders

#include <stdlib.h>
#include <stdio.h>
#include <string.h> // Libreria para trabajar con string
#include <TeensyStep.h> //Libreria para controlar los motores paso a paso
//https://luni64.github.io/TeensyStep/documentation/

//direcciones I2C del encoder
const int ams5600Address =0x36;

// direcciones de los registros de configuracion del as5600
const int hi-conf = 0x07;
const int lo-conf = 0x08;
// dirigirse a la figura 22 del datasheet del ams5600
int loconf = 00;
int hiconf = 12;

#define TCAADDR 0x70 // dirección I2C del multiplexor

AMS/_5600 ams5600; // definir sensor as5600

// variables para guardar los valores del encoder
float EncCodo=0;
float Encmuneca=0;
float EncHombro=0;
float EncBase=0;

int Contador=0; // contador lecturas encoder
```

```
// variables para guardar los datos estables de los encoders
float estableBase=0, estableHombro=0, estableCodo=0, estableMuneca=0, estableMunecaRot;

String incoming = ""; // string que recibe el dato por Serial
int flagNuevoDato=0; // flag nuevo dato

int flag =0; // flag que esta activo cuando algun motor se esta moviendo
int flagE =0; // flag para hacer la lectura de los encoders
int flagBucle = 0; // flag que indica que la accion a realizar es Automatica

//Motores, indicando los pines del Teensy donde se conectan en el orden: Step y Dir
Stepper Base(2,3);
Stepper Hombro(4,5);
Stepper Codo(6,7);
Stepper Muneca(8,9);
Stepper MunecaRot(35,36);
Stepper grip(37,38);
StepControl Scara; //Controlador de motores

//Variables de velocidades y aceleraciones
int BaseVelocity=3000;
int BaseAcceleration=2000;
int HombroVelocity=3000;
int HombroAcceleration=2000;
int CodoVelocity=3000;
int CodoAcceleration=2100;
int MunecaVelocity=1000;
int MunecaAcceleration=500;
int MunecaRotVelocity=10000;
int MunecaRotAcceleration=5000;
int gripVelocity=1200;
int gripAcceleration=2000;

int contadorBucle =0; // contador para ir avanzando en las posiciones guardadas

// función propia del multiplexor para acceder a los diferentes encoders
void tcselect(uint8_t i) {
    if (i > 7) return;
    Wire.beginTransmission(TCAADDR);
    Wire.write(1 << i);
    Wire.endTransmission();
}

// Creacion de un timer para la lectura de encoders
IntervalTimer myTimer;

// Creacion de un timer para el parpadeo de un LED
IntervalTimer timerLed;
int ledState=0;
```

```

//Constantes proporcion Pasos motor/grados de cada articulacion
float relBase=66.3472, relHombro=52.2222, relCodo=66.8444, relMuneca=29.6277;

/*****
/* Function: convertRawAngleToDegrees
/* In: angle data from AMS5600::getRawAngle
/* Out: human readable degrees as float
/* Description: takes the raw angle and calculates
/* float value in degrees.
*****/
float convertRawAngleToDegrees(word newAngle)
{
  /* Raw data reports 0 – 4095 segments, which is 0.087 of a degree */
  float retVal = newAngle * 0.087;
  return retVal;
}

void parpadeo(){
  ledState =! ledState;
  digitalWrite(13, ledState); // enciende y apaga el LED integrado en el Teensy
}

// Activa el flag para que se lean los encoders y comprueba si algun motor se esta moviendo
void checkTimer(){
  flagE=1;
  if (!Scara.isRunning() && flag ==1)
  {
    flag = 0;
  }
}

//Función para hacer split del string que recibe el teensy
String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  int maxIndex = data.length() - 1;

  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }

  return found >index ? data.substring(strIndex[0], strIndex[1]) : ;
}

```

```
void setup()
Serial.begin(9600); // Se activa la comunicación Serial
Wire.begin(); // Se activa la comunicación I2C
//Configura los registros del encoder de la muñeca para no tener problemas por ruido
tcselect(3);
Wire.beginTransmission(ams5600Address);
Wire.write(lo-conf);
Wire.write(loconf);
Wire.endTransmission();
tcselect(3);
Wire.beginTransmission(ams5600Address);
Wire.write(hi-conf);
Wire.write(hiconf);
Wire.endTransmission();

pinMode(13, OUTPUT);

// Asignación velocidades maximas y aceleraciones a los motores
Base.setMaxSpeed(BaseVelocity);
Base.setAcceleration(BaseAceleration);
Hombro.setMaxSpeed(HombroVelocity);
Hombro.setAcceleration(HombroAceleration);
Codo.setMaxSpeed(CodoVelocity);
Codo.setAcceleration(CodoAceleration);
Muneca.setMaxSpeed(MunecaVelocity);
Muneca.setAcceleration(MunecaAceleration);
MunecaRot.setMaxSpeed(MunecaRotVelocity);
MunecaRot.setAcceleration(MunecaRotAceleration);
grip.setMaxSpeed(gripVelocity);
grip.setAcceleration(gripAceleration);

myTimer.priority(100); // prioridad de la interrupcion del timer 0=max 255=min
myTimer.begin(checkTimer, 10000); // cada 10ms

timerLed.priority(101);
timerLed.begin(parpadeo, 1000000); // cada 1s

// Mueve los motores los pasos indicado y activa el flag de movimiento
void move(int BPos, int HPos, int CPos, int MPos, int MRPos, int GPos)

int BasePose = BPos;
int HombroPose = HPos;
int CodoPose = CPos;
int MunecaPose = MPos;
int MunecaRotPose = MRPos;
```

```
int gripPose = GPos;

Base.setTargetAbs(BasePose);
Hombro.setTargetAbs(HombroPose);
Codo.setTargetAbs(CodoPose);
Muneca.setTargetAbs(MunecaPose);
MunecaRot.setTargetAbs(MunecaRotPose);
grip.setTargetAbs(gripPose);
Scara.moveAsync(Base, Hombro, Codo, Muneca, MunecaRot);
flag=1;

// Calcula cuanto tiene que moverse segun los angulos recibidos y activa el flag de movimiento
void moveAngulo(float BPos, float HPos, float CPos, float MPos, float MRPos, float GPos)

int BasePose = -relBase*(estableBase - BPos);
int HombroPose = relHombro*(estableHombro - HPos);
int CodoPose = relCodo*(estableCodo - CPos);
int MunecaPose = relMuneca*(estableMuneca - MPos);
int MunecaRotPose = MRPos;
int gripPose = GPos;

Base.setTargetRel(BasePose);
Hombro.setTargetRel(HombroPose);
Codo.setTargetRel(-CodoPose);
Muneca.setTargetRel(MunecaPose);
MunecaRot.setTargetAbs(MunecaRotPose);
grip.setTargetRel(gripPose);
Scara.moveAsync(Base, Hombro, Codo, Muneca, MunecaRot, grip);
flag=1;

// Comprueba si los motores estan dentro de la ventana de error y si alguno esta fuera
lo corrige.
void motorBucle(float BPos, float HPos, float CPos, float MPos, float MRPos, float GPos)

if(((fabs(BPos-estableBase)>0.05 || fabs(HPos-estableHombro)>0.25 || fabs(CPos-estableCodo)>0.25
|| fabs(MPos-estableMuneca)>0.5)))
Serial.print(fabs(BPos-estableBase), 2);
Serial.print();
Serial.print(fabs(HPos-estableHombro), 2);
Serial.print();
Serial.print(fabs(CPos-estableCodo), 2);
Serial.print();
Serial.print(fabs(MPos-estableMuneca), 2);
Serial.println();
moveAngulo(BPos, HPos, CPos, MPos, MRPos, GPos);

else
contadorBucle++; // si todos estan dentro de los motores suma contador para ir a la siguiente
```

posicion

void loop()

```
if(flagBucle==1 flag==0) // ejecuta los comando en orden, solo si se indico la acción automática
```

```
if(contadorBucle ==0)
```

```
motorBucle(180.00, 85.00, 0, 0, 0, 0);
```

```
if(contadorBucle ==1)
```

```
motorBucle(230, 67.868, 34.535, -123.334, -1800, 0);
```

```
if(contadorBucle ==2)
```

```
motorBucle(230, 72.44, 50.01, -112.425, -1800, 0);
```

```
if(contadorBucle ==4)
```

```
motorBucle(180, 85, 0, 0, 0, 0);
```

```
if(contadorBucle ==5)
```

```
motorBucle(230, 67.868, 34.534, -123.334, -1800, 0);
```

```
if(contadorBucle ==6)
```

```
motorBucle(230, 72.44, 50.01, -112.425, -1800, 0);
```

```
if(contadorBucle ==8)
```

```
motorBucle(230, 57.4, 7.47, -139.93, -1800, 0);
```

```
if(contadorBucle ==9)
```

```
motorBucle(180.00, 85.00, 0, 0, 0, 0);
```

```
flagBucle=0;
```

```
if(contadorBucle ==3)
```

```
grip.setTargetRel(-2100);
```

```
Scara.moveAsync(grip);
```

```
flag=1;
```

```
contadorBucle++;
```

```
if(contadorBucle ==7)
```

```
grip.setTargetRel(2100);
```

```
Scara.moveAsync(grip);
```

```
flag=1;
```

```
contadorBucle++;
```

```
if (Serial.available() >0) // comprueba si ha recibido algun dato
```

```
incoming = Serial.readString(); // guarda en un string el dato en bruto
// Separa el string en diferentes variables según el separador indicado ", "
String accion = getValue(incoming, ',', 0);
String SBase = getValue(incoming, ',', 1);
String SHombro = getValue(incoming, ',', 2);
String SCodo = getValue(incoming, ',', 3);
String SMuneca = getValue(incoming, ',', 4);
String SMunecaRot = getValue(incoming, ',', 5);
String SGrip = getValue(incoming, ',', 6);

if(flag == 0) // Si no esta en movimiento comprueba el valor accion y ejecuta la orden
correspondiente
if (accion == "Manual")
move(SBase.toInt(), SHombro.toInt(), SCodo.toInt(), SMuneca.toInt(), SMunecaRot.toInt(),
SGrip.toInt());

else if (accion == ".Angulo")
moveAngulo(SBase.toFloat(), SHombro.toFloat(), SCodo.toFloat(), SMuneca.toFloat(),
SMunecaRot.toFloat(), SGrip.toFloat());

else if(accion == "Path")
moveAngulo(SBase.toFloat(), SHombro.toFloat(), SCodo.toFloat(), SMuneca.toFloat(),
SMunecaRot.toFloat(), SGrip.toFloat());
flagBucle=1;

contadorBucle=0;

else // si se esta moviendo omite la orden recibida e imprime un mensaje en pantalla
Serial.println("No puedo, me estoy moviendo");

// accion de leer los encoders
if(flagE==1)
Contador++;
tcaselect(2);
EncCodo=EncCodo+convertRawAngleToDegrees(ams5600.getScaledAngle());
tcaselect(3);
Encmuneca=Encmuneca+convertRawAngleToDegrees(ams5600.getRawAngle());
tcaselect(4);
EncHombro=EncHombro+convertRawAngleToDegrees(ams5600.getScaledAngle());
tcaselect(5);
EncBase=EncBase+convertRawAngleToDegrees(ams5600.getScaledAngle());

// se aplica una media por si hubiera algun tipo de ruido entre un valor y otro
if(Contador==9)

EncCodo=EncCodo/10;
```

```
Encmuneca=Encmuneca/10;
EncHombro=EncHombro/10;
EncBase=EncBase/10;

// Se mapean los ángulos para formatearlos según indica el código de Python
estableBase=map(EncBase, 0, 360, 360, 0);
estableHombro=map(EncHombro, 115.57, 320 , 0, 85);
estableCodo=map(EncCodo, 12.84, 320.64, 154.35, -130.8);
estableMuneca=map(Encmuneca, 15.11, 248.59, -138.35, 126.2);

// Imprime una serie de datos constantemente por pantalla, así se puede visualizar
// en tiempo real los datos con los que trabaja el Teensy

Serial.print(flag);
Serial.print();
Serial.print(contadorBucle);
Serial.print();
Serial.print(estableBase);
Serial.print();
Serial.print(estableHombro);
Serial.print();
Serial.print(estableCodo);
Serial.print();
Serial.print(estableMuneca);
Serial.print();
Serial.print(Base.getPosition());
Serial.print();
Serial.print(Hombro.getPosition());
Serial.print();
Serial.print(Codo.getPosition());
Serial.print();
Serial.print(Muneca.getPosition());
Serial.print();
Serial.print(MunecaRot.getPosition());
Serial.print();
Serial.println(grip.getPosition());
Contador=0;
EncCodo=0;
Encmuneca=0;
EncHombro=0;
EncBase=0;
```

11.2. Código Python

Código 11.2: Código Python

```

import math

class Kine3:
    def cosCalc(self, a, b, c):
        angle = math.acos((a*a + b*b -c*c)/(2*a*b))
        return angle

    def inputCalc(self):
        # Introducción de las coordenadas X, Y y ángulo de la herramienta
        xe = float(input("xe: "))
        ye = float(input("ye: "))
        phie = math.radians(float(input("phie in degrees: ")))
        print (phie)
        # Longitud de los 3 segmentos en mm
        l1 = 305 # Brazo
        l2 = 257.45 # Antebrazo
        l3 = 424 # Muñeca

        xw = xe -l3 * math.cos(phie)
        yw = ye -l3 * math.sin(phie)
        r =math.sqrt(xw**2 + yw**2)
        gamma = self.cosCalc(r, l1, l2)
        theta2 = math.pi -self.cosCalc(l1, l2, r)
        theta1 = math.atan2(yw, xw) -gamma
        theta3 = phie -theta1 -theta2
        #imprime en pantalla los ángulos
        print("theta1: {} ".format(math.degrees(theta1 + 2 *gamma)))
        print("theta2: {} ".format(-math.degrees(theta2 * -1)))
        print("theta3: {} ".format(math.degrees(theta3 + 2 * (theta2 -gamma))))

# Ejecutar script
# Ejecutar kine3 = Kine3()
# Ejecutar kine3.inputCalc()
# Introducir coordenada X, Y y ángulo
# Si es posible da la solución de los ángulo en orden hombro, Codo, Muñeca
# EJ:
#>>> Run IK2.py
#>>> kine3 = Kine3()
#>>> kine3.inputCalc()
#xe: 330
#ye: 10
#phie in degrees: -90

```

#-1.5707963267948966

#theta1: 65.79365650704413

#theta2: 28.548053892782708

#theta3: -127.24560261426143

12 ANEXOS EN FUNCIÓN DEL ÁMBITO DE APLICACIÓN DEL TFG

12.1. Seguridad

No aplica debido a que el objetivo de este trabajo es ser un prototipo e investigación.

13 OTROS ANEXOS

13.1. Simulación y optimización de piezas

13.1.1. Simulación

En ese anexo se detalla el análisis de estrés de cada pieza estructural realizada en acero, sometiéndola a los esfuerzos que soportará una vez ensamblada.

Base1

Fijando los soportes donde irán los rodamientos y colocando una fuerza de 200 N en el centro de la base.

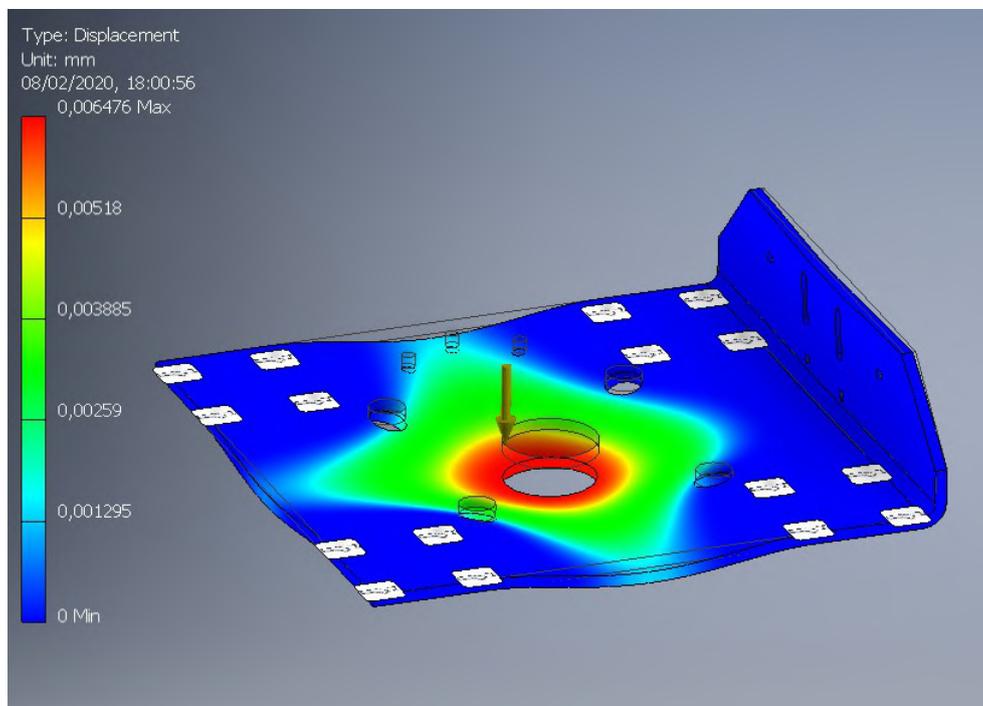


Figura 13.1 – Simulación base1

Base2

Fijando los tornillos que unen la pieza a base1 y simulando la fuerza que puede generar la tensión de la correa de la base.

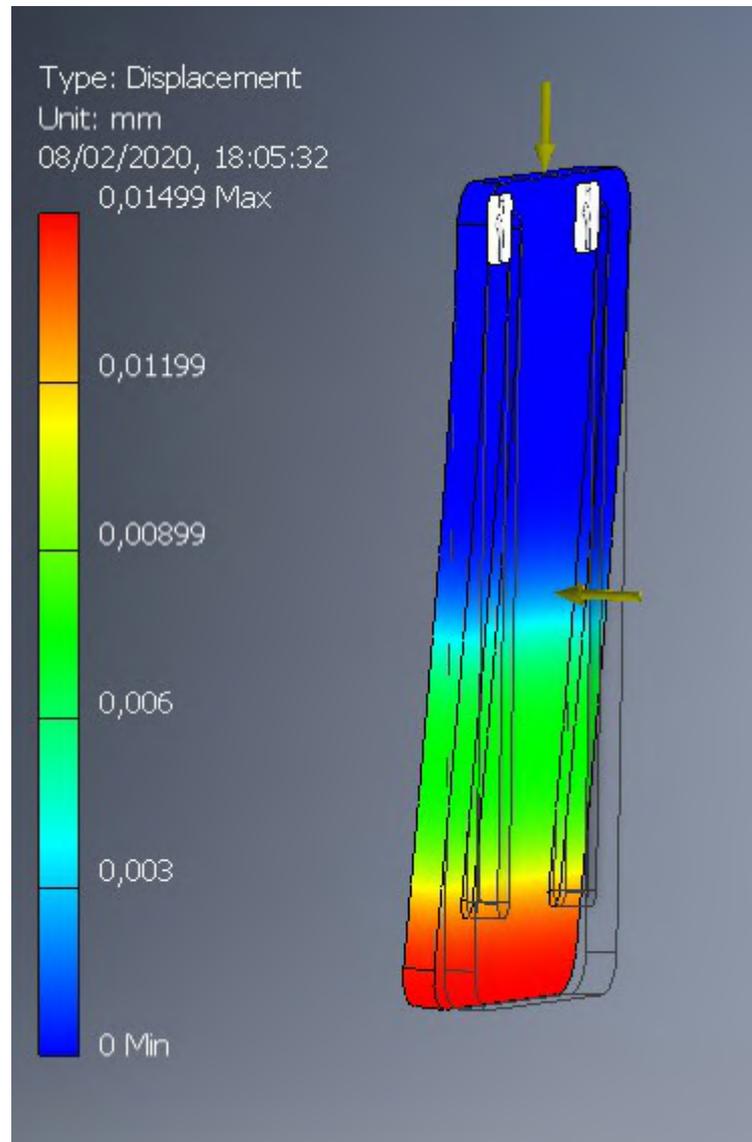


Figura 13.2 – Simulación base2

Base3

Fijando la unión con la pieza y simulado una fuerza vertical de 150N y un par correspondiente a mover 1 kg con el brazo estirado en horizontal.

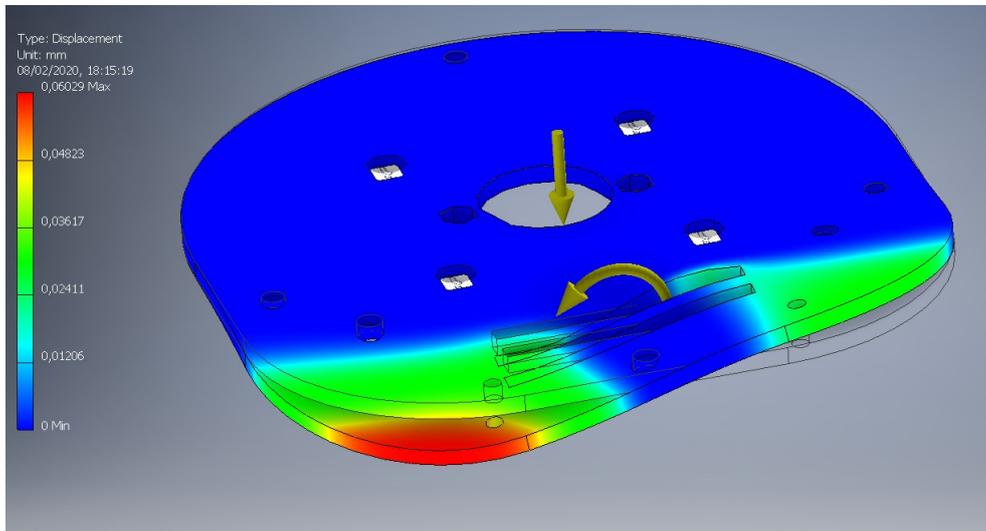


Figura 13.3 – Simulación base3

Base4

Fijando los tornillos y simulando una fuerza vertical equivalente al motor que soporta(15N) y el par que suministra ese motor.

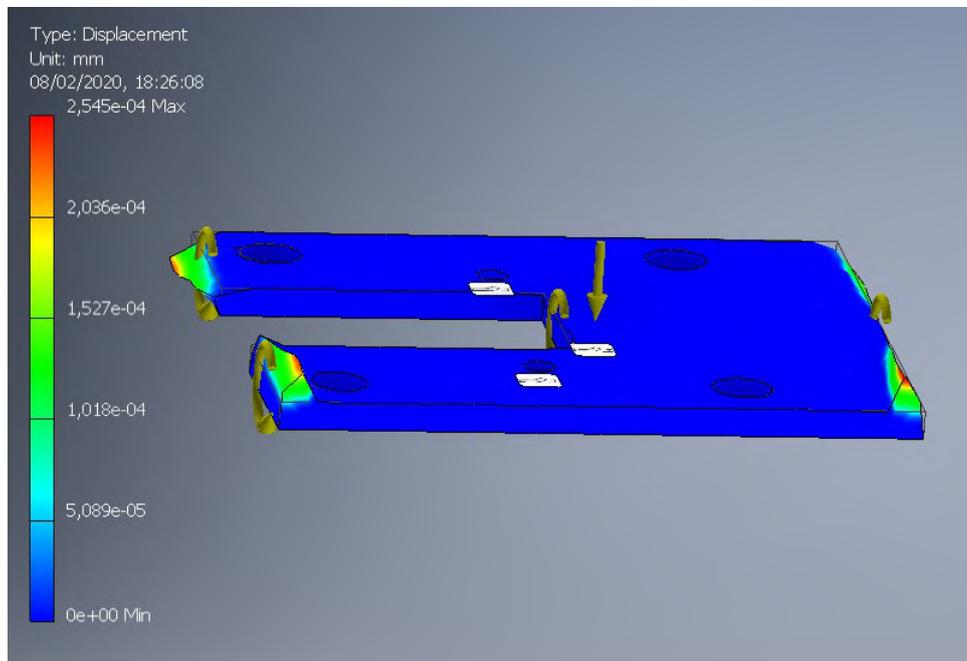


Figura 13.4 – Simulación base4

Base5

Mismas condiciones que la pieza Base4.

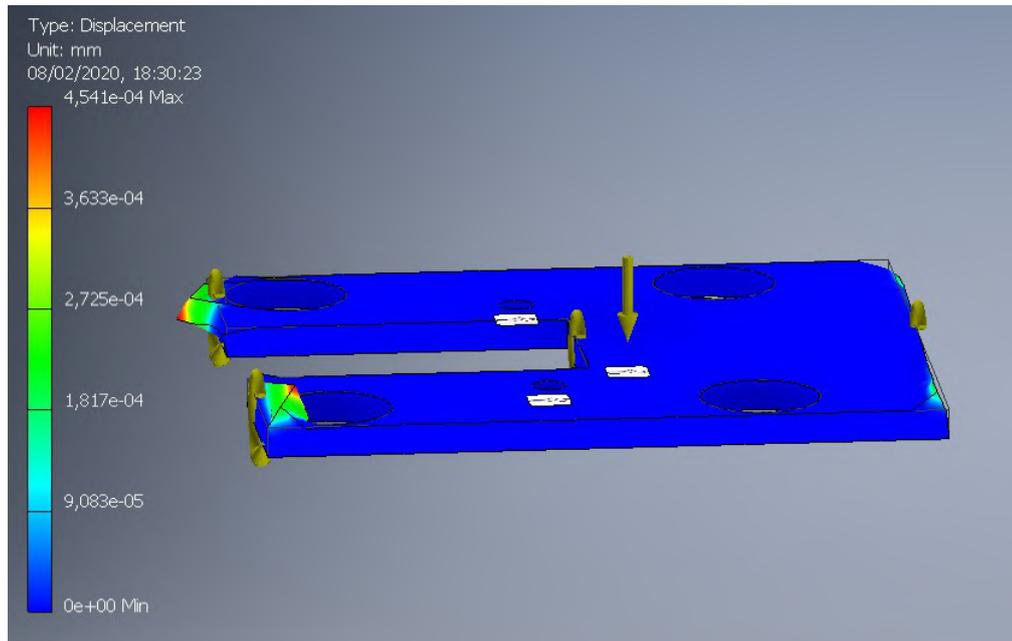


Figura 13.5 – Simulación base5

Cuerpo1

Fijando los tornillos y simulando una fuerza vertical de 140N y el par generado por el muelle de contrapeso.

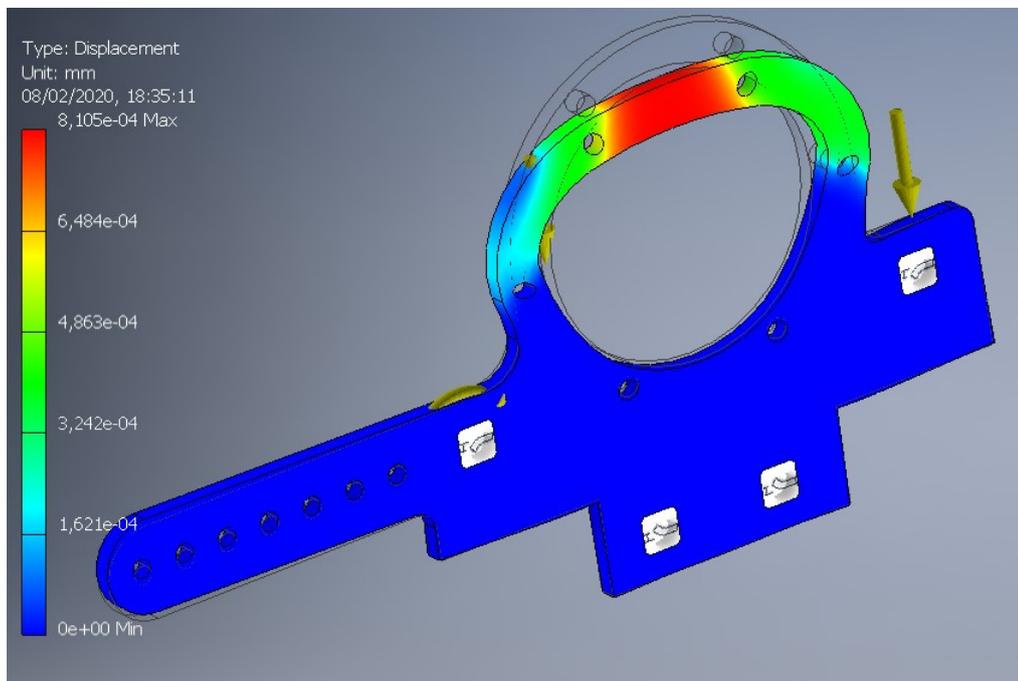


Figura 13.6 – Simulación cuerpo1

Cuerpo2

Esta pieza no soporta ningún tipo de esfuerzo, ya que es un separador entre el rodamiento del hombro y las piezas brazo1.

Cuerpo3

Fijando los tornillos y simulando el par transmitido por la pieza Base3.

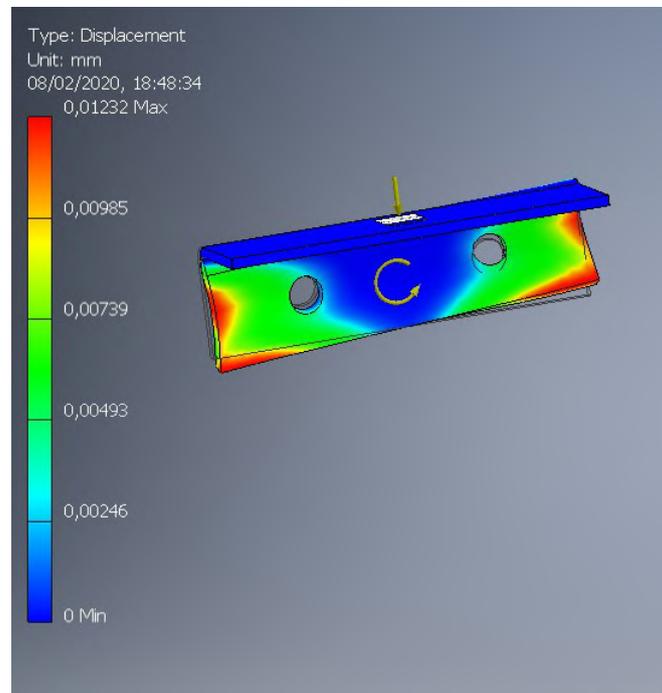


Figura 13.7 – Simulación cuerpo3

Brazo1

Fijando la unión con el rodamiento el motor del hombro, aplicando la fuerza en el codo de 50N debido a que el conjunto soportado pesa 9Kg pero se reparte entre 2 piezas Brazo1.

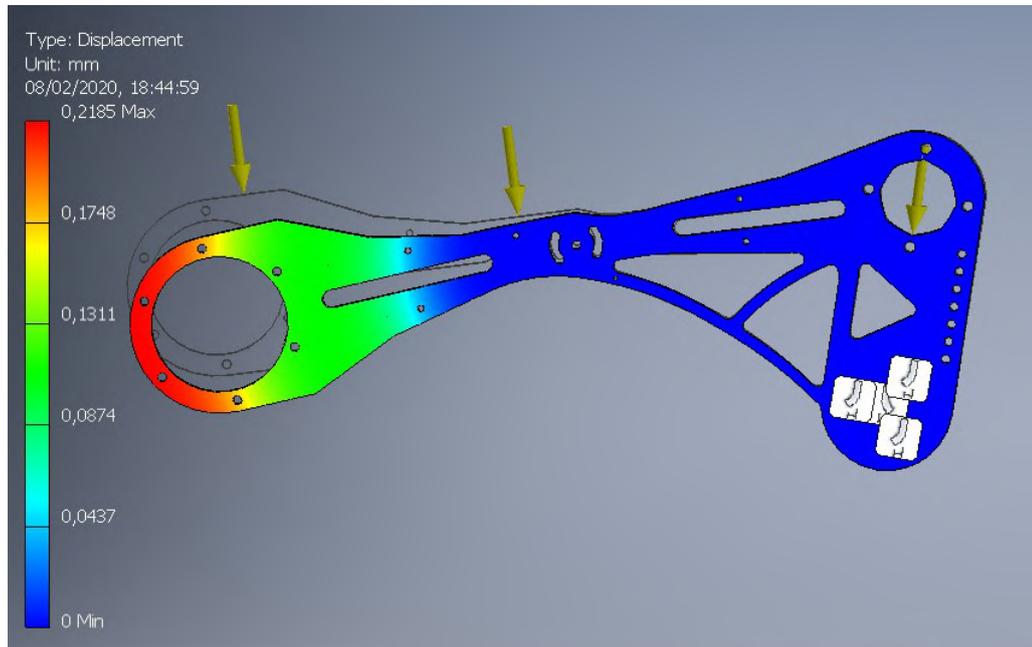


Figura 13.8 – Simulación brazo1

Tensor

Fijando la unión con el brazo y simulando un par proporcional a la tensión de la correa.

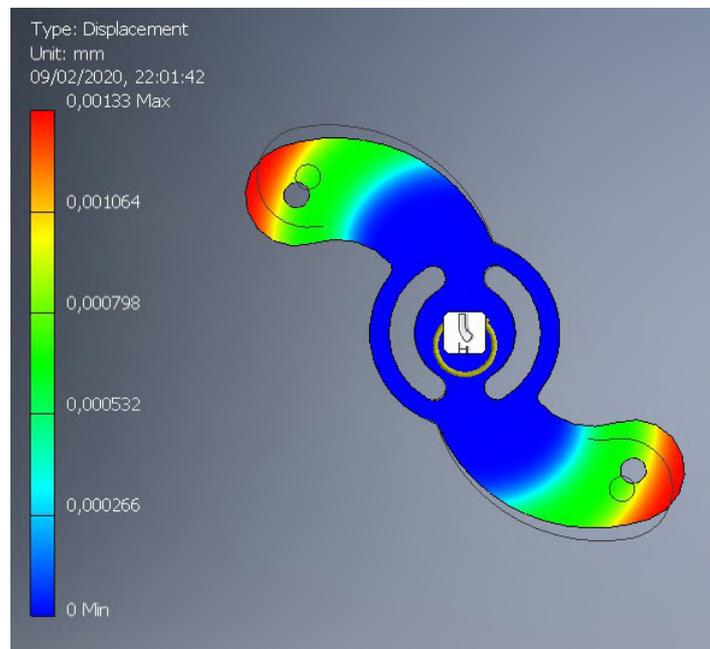


Figura 13.9 – Simulación tensor

Antebrazo1

Fijando la unión con el brazo y aplicando por una parte 25N debido a que la fuerza se reparte entre 2 piezas antebrazo1

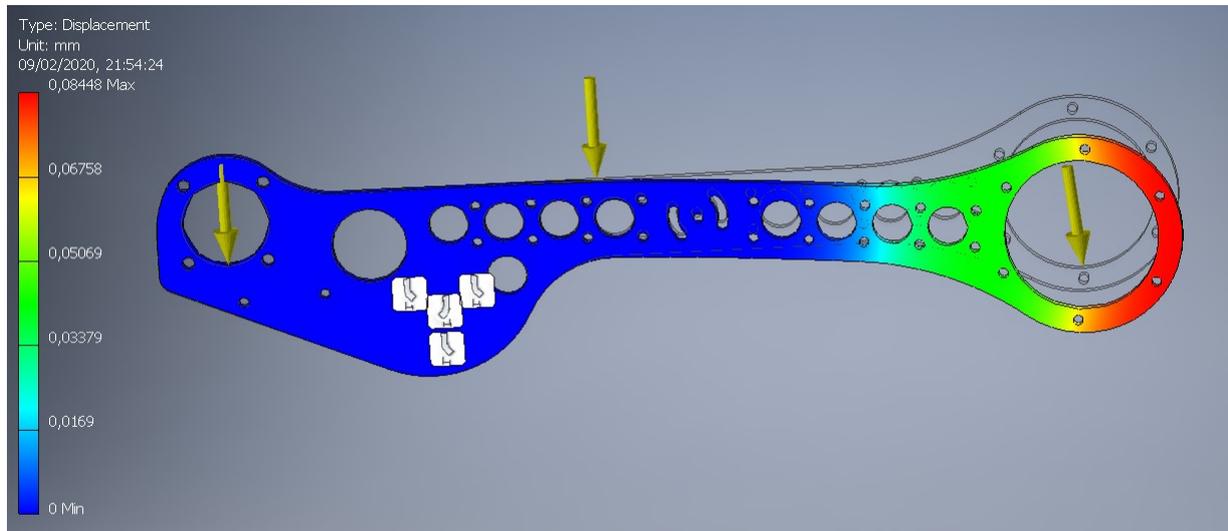


Figura 13.10 – Simulación antebrazo

Antebrazo2

Esta pieza no necesita simulación ya que es un separador entre las 2 piezas del antebrazo.

Muñeca

Fijando la unión con el antebrazo y aplicando una fuerza de 25N en la unión con el eje de la brida

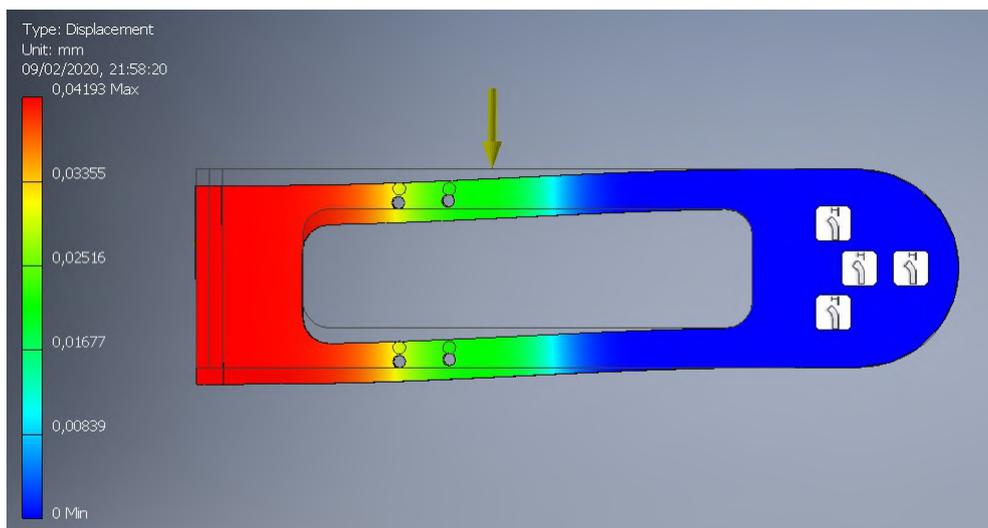
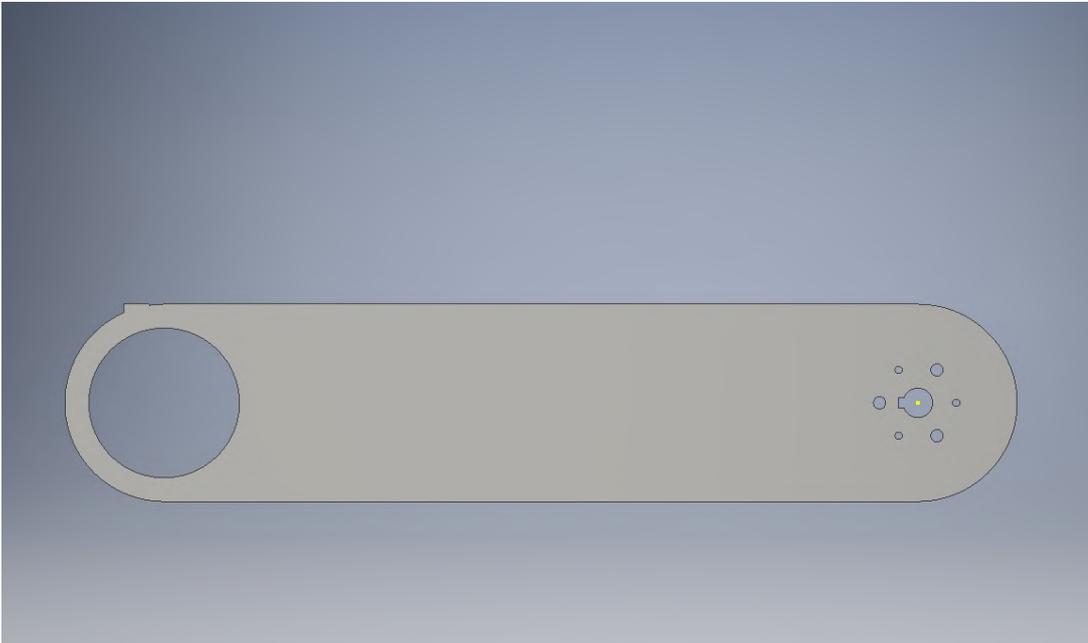


Figura 13.11 – Simulación Muñeca

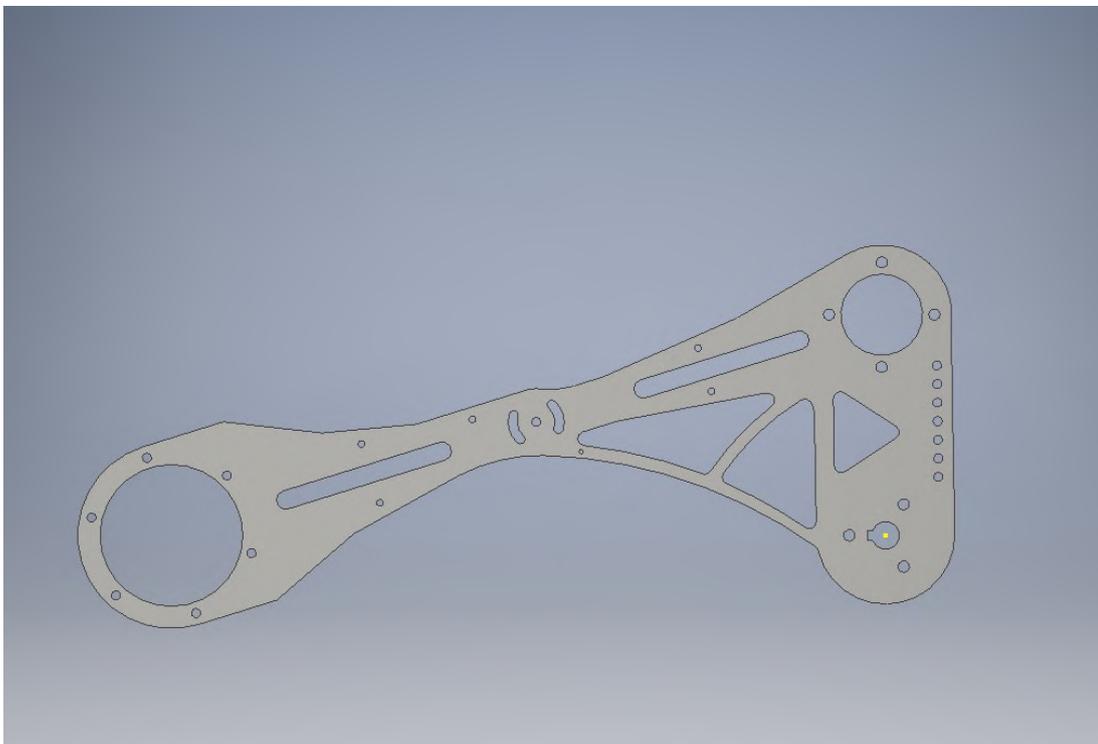
13.1.2. Optimización

La optimización solo se aplica a las piezas brazo1, antebrazo1 y muñeca.

13.1.2.1. Brazo



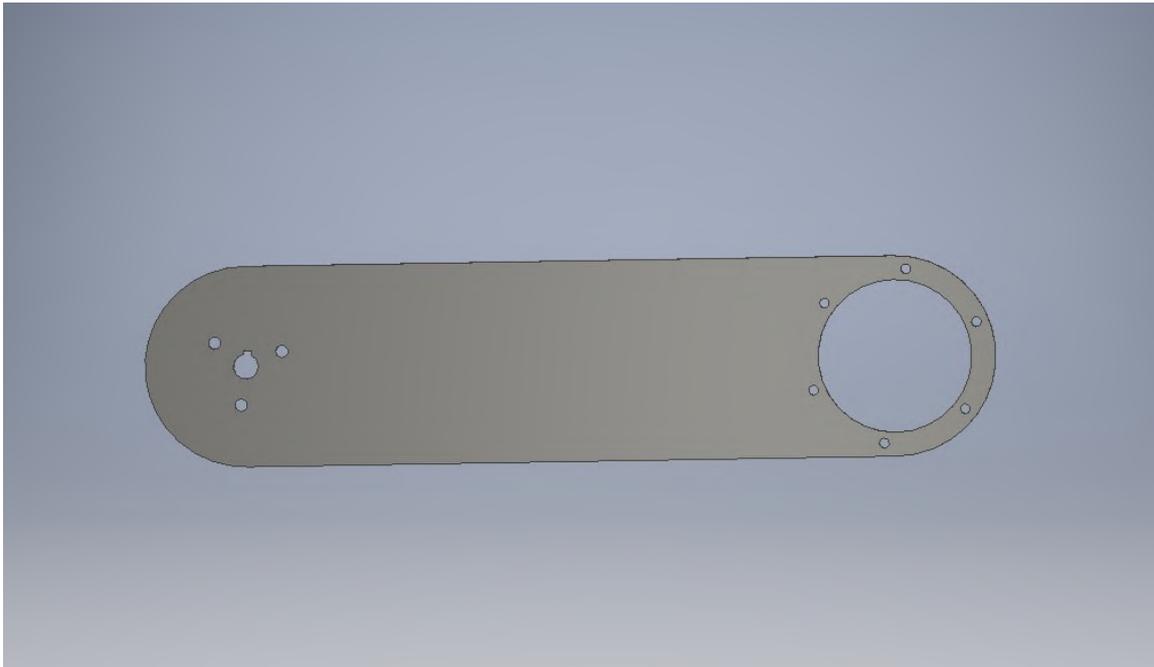
(a) Pieza original, peso = 850g



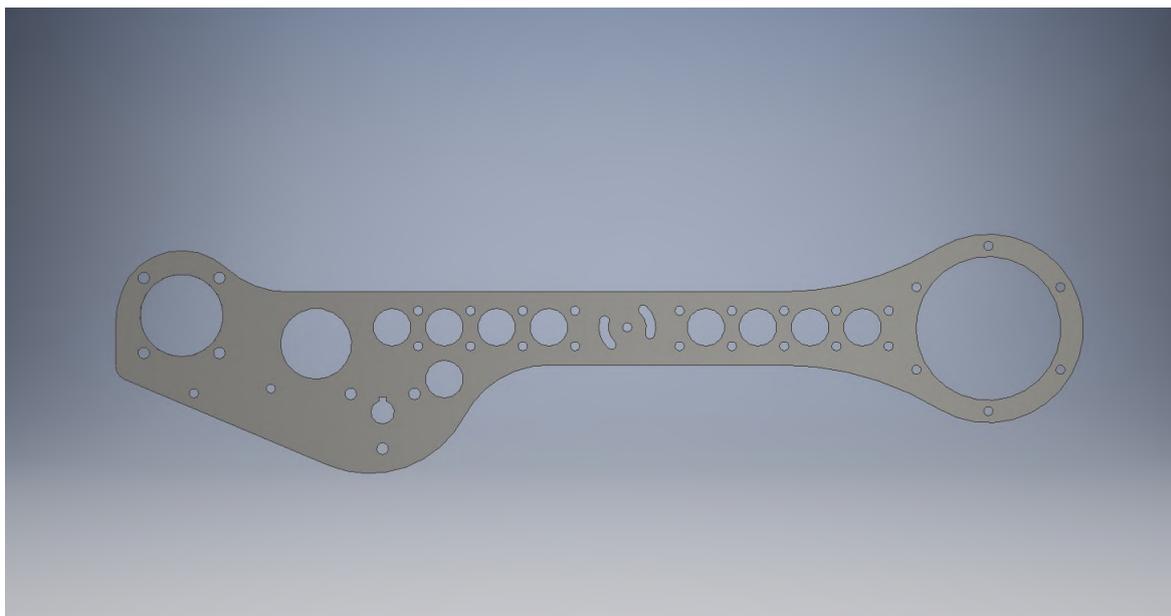
(b) Pieza optimizada, peso = 550g

Figura 13.12 – Optimización brazo

13.1.2.2. Antebrazo



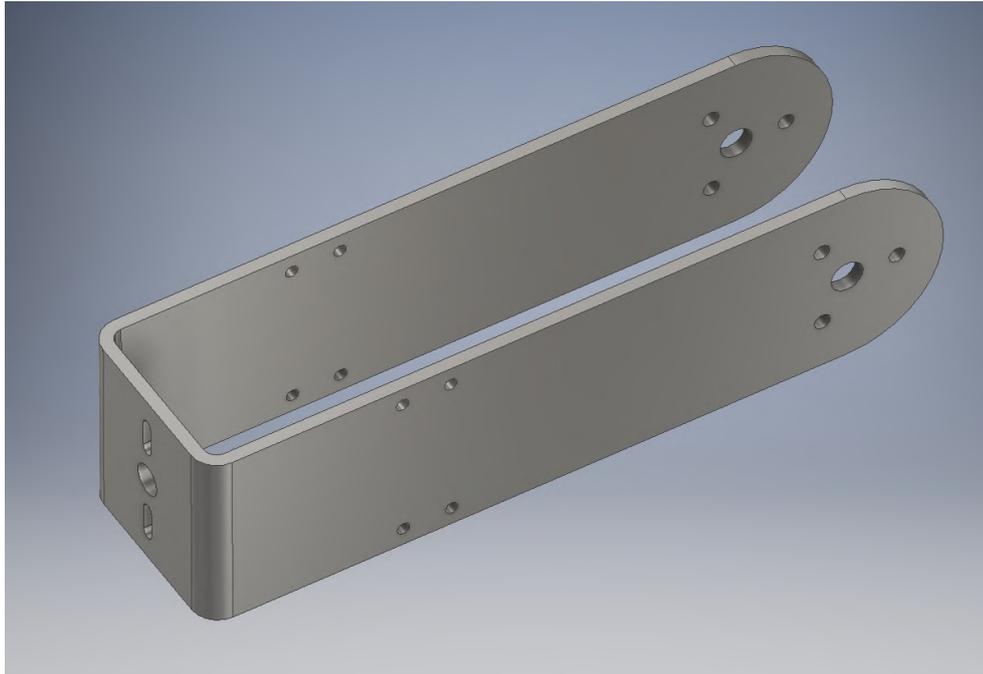
(a) Pieza original, peso = 720g



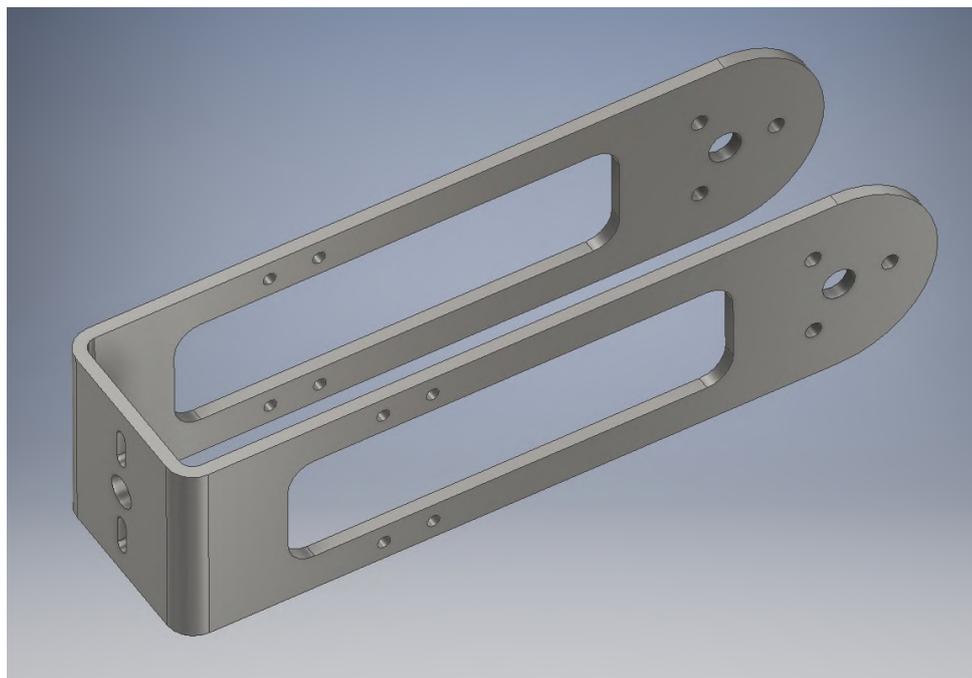
(b) Pieza optimizada, peso = 445g

Figura 13.13 – Optimización antebrazo

13.1.2.3. Muñeca



(a) Pieza original, peso = 930g



(b) Pieza optimizada, peso = 620g

Figura 13.14 – Optimización muñeca

13.2. Ensamblaje del robot

En este anexo se detalla de forma simplificada el montaje del robot a través de figuras sacadas del CAD.

13.2.1. Base

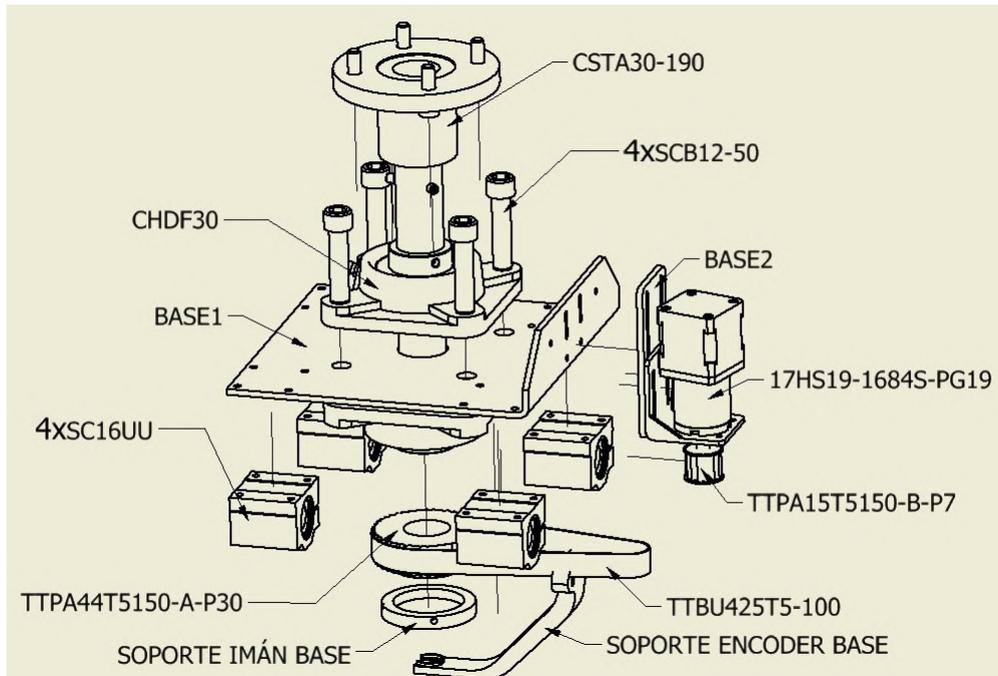


Figura 13.15 – Ensamblaje base

13.2.2. Cuerpo

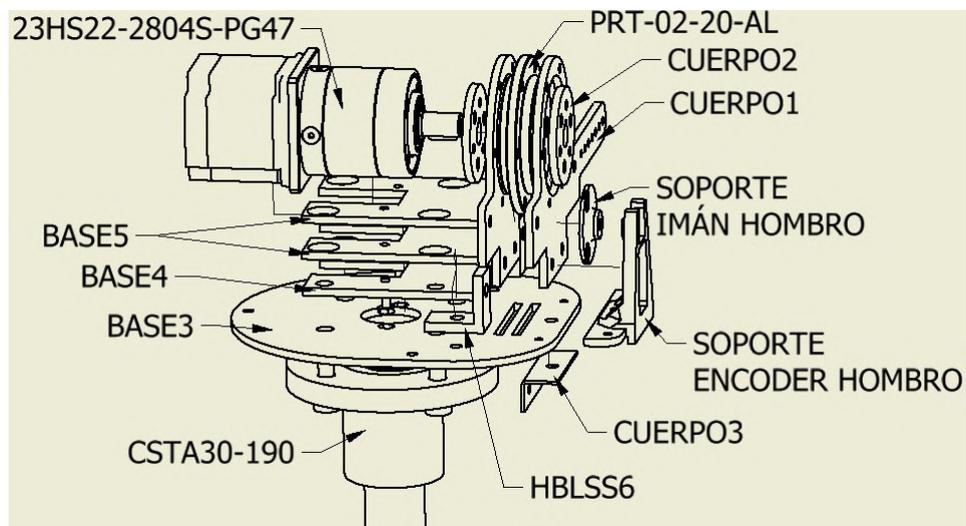


Figura 13.16 – Ensamblaje cuerpo

13.2.3. Brazo

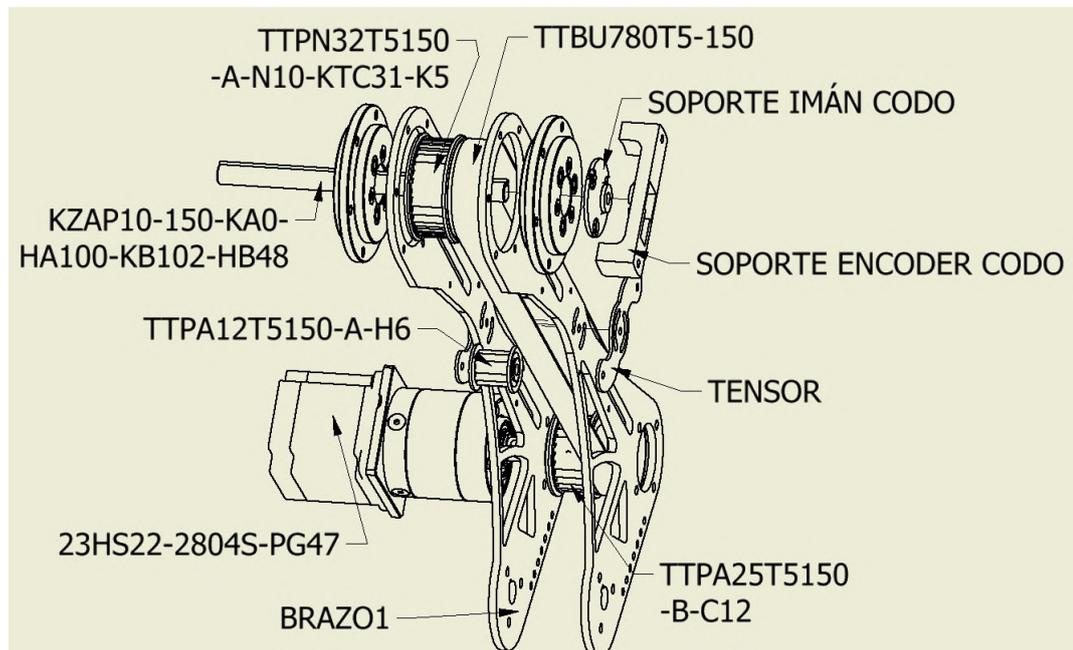


Figura 13.17 – Ensamblaje brazo

13.2.4. Antebrazo

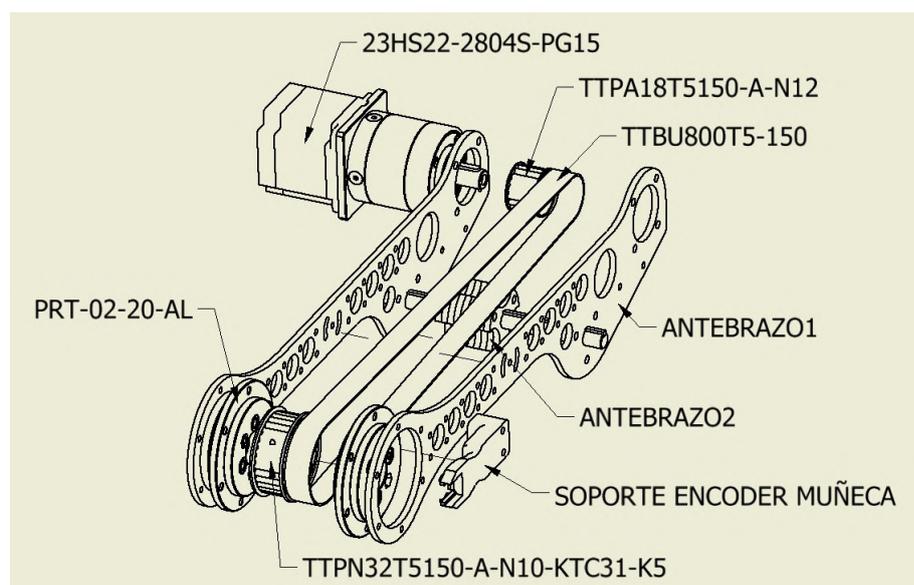


Figura 13.18 – Ensamblaje antebrazo

13.2.5. Muñeca

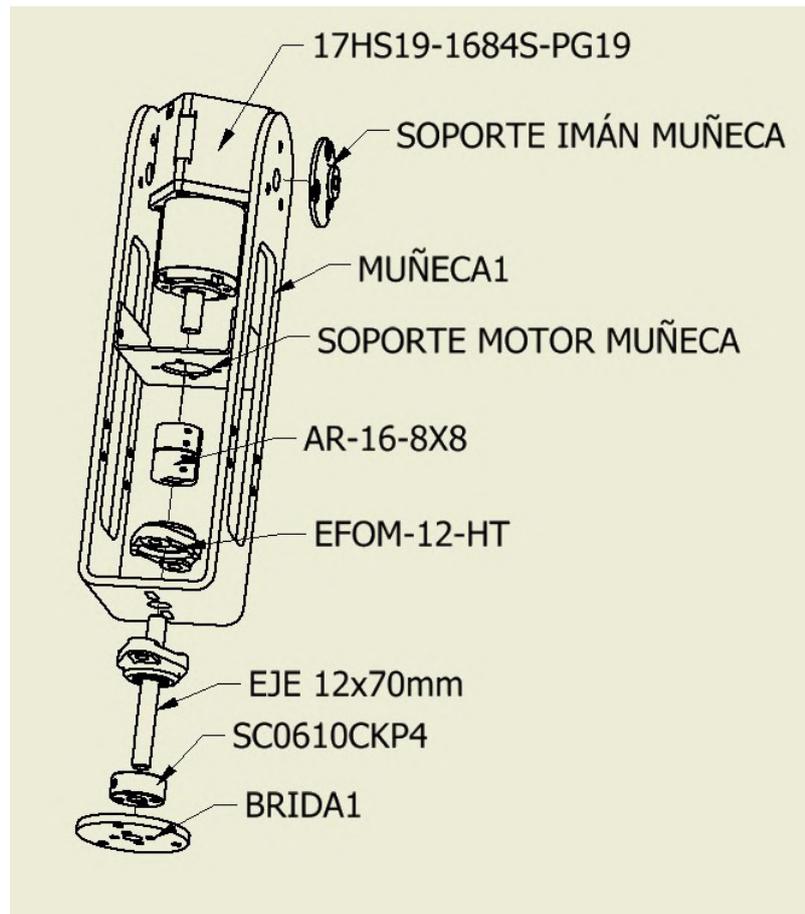


Figura 13.19 – ensamble muñeca

13.3. Montaje y ensayo

En este anexo se muestran detalles del montaje y un pequeño análisis sobre el control del robot.

13.3.1. Montaje

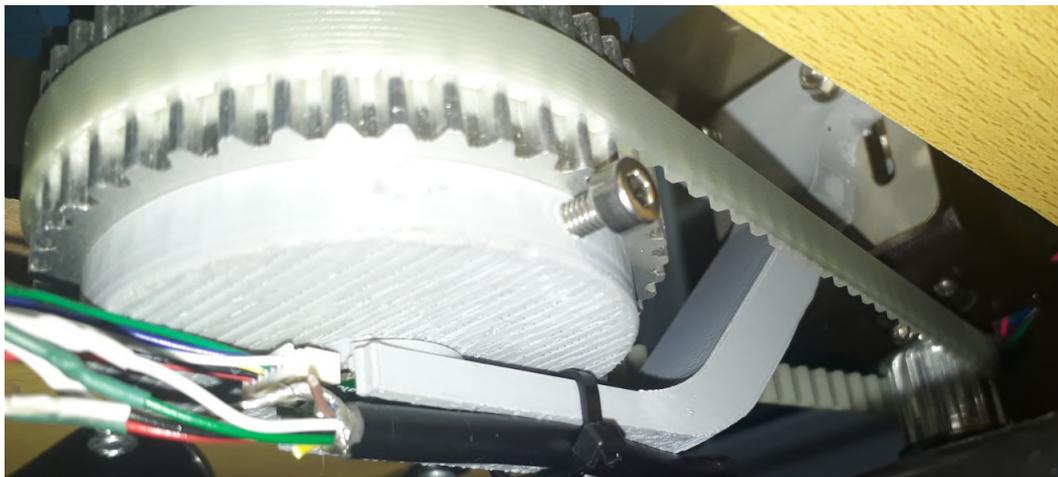


Figura 13.20 – Detalle del encoder de la base



Figura 13.21 – Detalle del muelle y del encoder del hombro

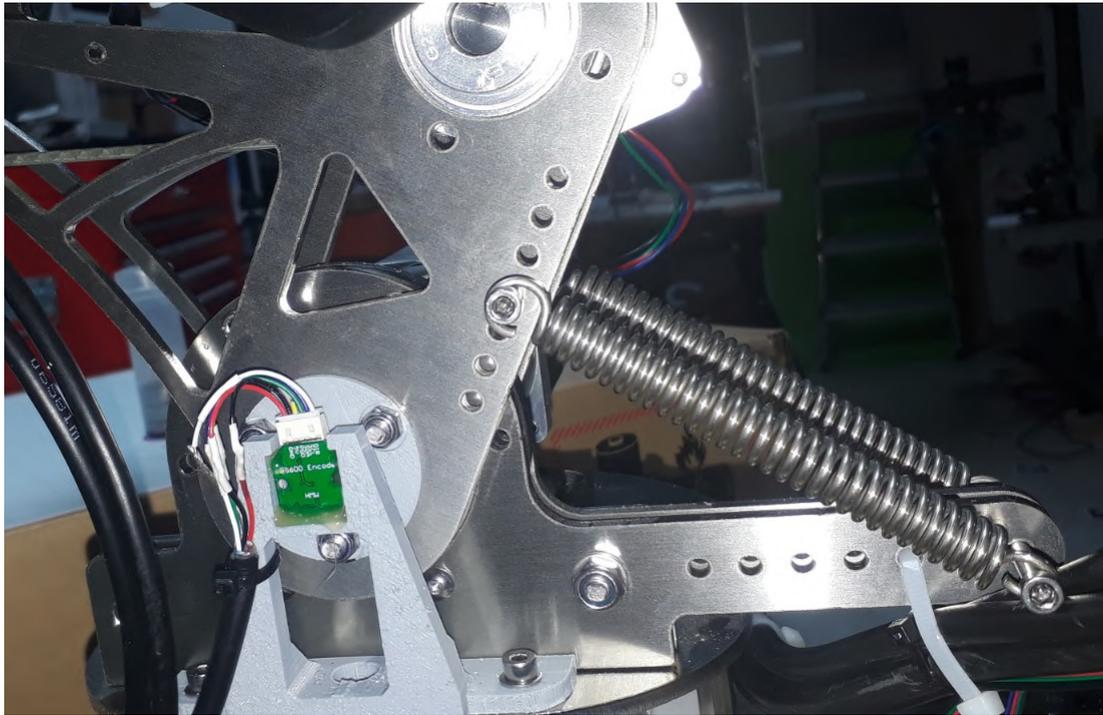


Figura 13.22 – Detalle del muelle estirado y del encoder del hombro

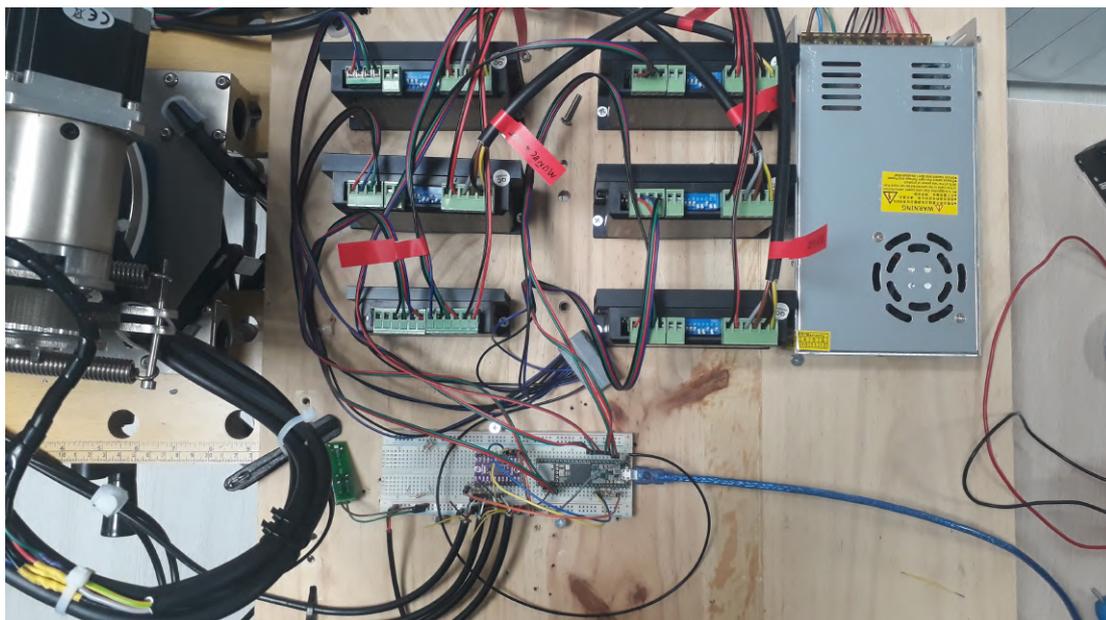


Figura 13.23 – Detalle del montaje de drivers y breadboard



Figura 13.24 – Detalle del tensor de la correa del antebrazo



Figura 13.25 – Detalle del montaje de la rotación de la muñeca

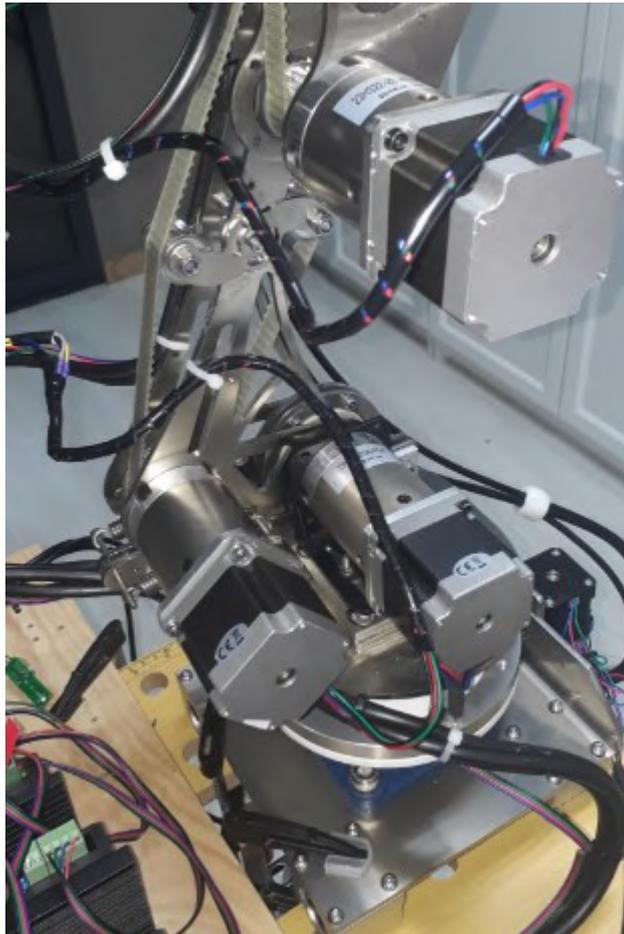


Figura 13.26 – Detalle de los motores del hombro, codo y muñeca

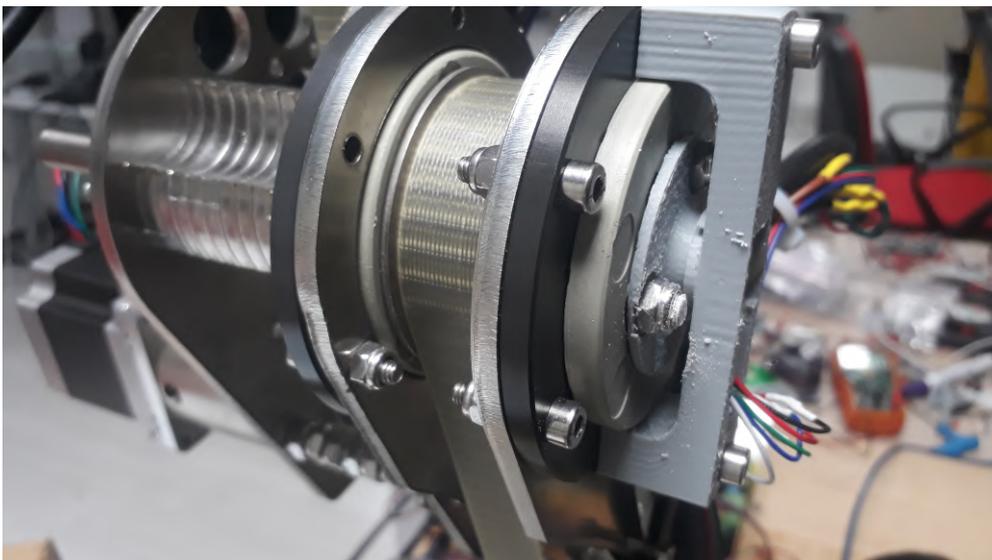


Figura 13.27 – Detalle la articulación codo



Figura 13.28 – Robot manipulador con pinza

13.3.2. Ensayo

En este apartado se muestra el robot realizando la aproximación a un objeto ubicado en las coordenadas $X=330\text{mm}$, $Y=20\text{mm}$ con ángulo sobre el plano $Z=230^\circ$. Haciendo una parada en el punto $X=330\text{mm}$, $Y=80\text{mm}$ para facilitar el agarre del objeto.

Esta prueba se realiza en el modo manual indicando los ángulos previamente calculados en Python y parte con el brazo totalmente estirado con orientación $Z=180^\circ$.

Para alcanzar el primer punto necesita 3 intentos. 13.29(c)

Para alcanzar el segundo punto necesita 4 intentos, aquí se puede apreciar sobreoscilación del sistema. 13.29(h)

Vuelve a la posición inicial con el objeto agarrado.

A la hora de volver a la primera posición necesita de nuevo 3 intentos. 13.29(m)

Para llegar a la posición donde dejar el objeto necesita 2 intentos. 13.29(o)



(a) Aproximación 1



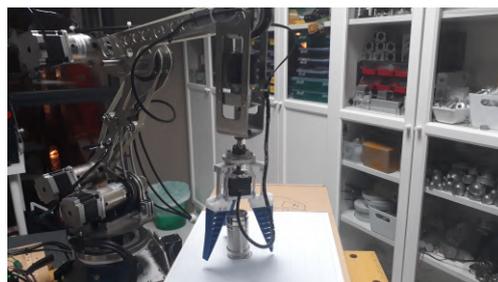
(b) Aproximación 2



(d) Aproximación 3



(e) Aproximación 4



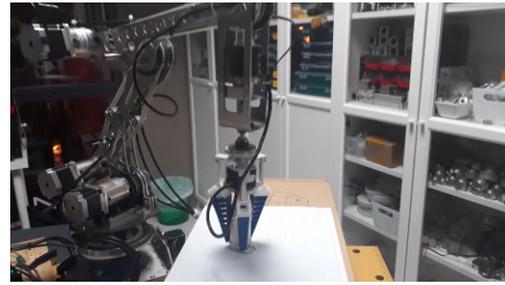
(f) Aproximación 5



(g) Aproximación 6



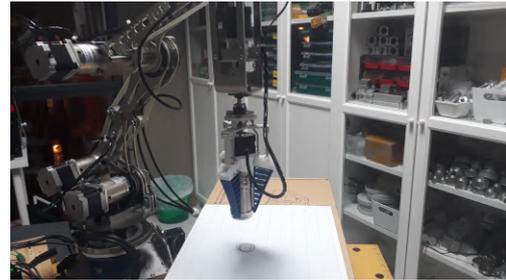
(i) Aproximación 7



(j) Aproximación 8



(k) Aproximación 9



(l) Aproximación 10



(n) Aproximación 11



(ñ) Aproximación 12



(p) Aproximación 13



(q) Aproximación 14

Figura 13.29 – Aproximación manual

Al integrar el algoritmo se realizan 4 intentos consecutivos, siendo el resultado de la ubicación del objeto el siguiente:

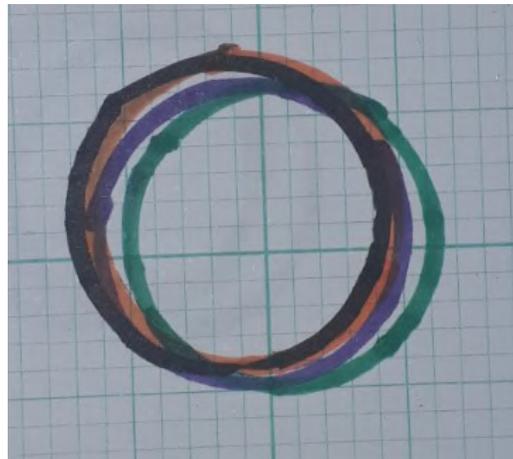


Figura 13.30 – Aproximación algoritmo

El orden de los resultados es 1ºnaranja, 2ºverde, 3ºmorado y 4ºnegro, Cada casilla de la hoja mide 2mm.

Por tanto se puede apreciar que el error máximo es aproximadamente 3mm, esto se puede corregir reduciendo la velocidad tras una primera aproximación o reduciendo los márgenes de precisión del bucle automático.

TÍTULO: Diseño, montaje y control de un manipulador robótico.

PLANOS

PETICIONARIO: ESCUELA UNIVERSITARIA POLITÉCNICA

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: FEBRERO DE 2020

AUTOR: EL ALUMNO

Fdo.: ANTONIO REGUEIRA PEÑAMARÍA

Índice de planos

1	BASE1	109
2	BASE2	111
3	BASE3	113
4	BASE4	115
5	BASE5	117
6	CUERPO1	119
7	CUERPO2	121
8	CUERPO3	123
9	BRAZO1	125
10	TENSOR1	127
11	ANTEBRAZO1	129
12	ANTEBRAZO2	131
13	MUÑECA1	133
14	BRIDA1	135
15	Soporte motor muñeca	137
16	Soporte encoder base	139
17	Soporte encoder hombro	141
18	Soporte encoder codo	143
19	Soporte encoder muñeca	145
20	Soporte imán hombro, codo y muñeca	147
21	Soporte imán base	149

1

2

3

4

A

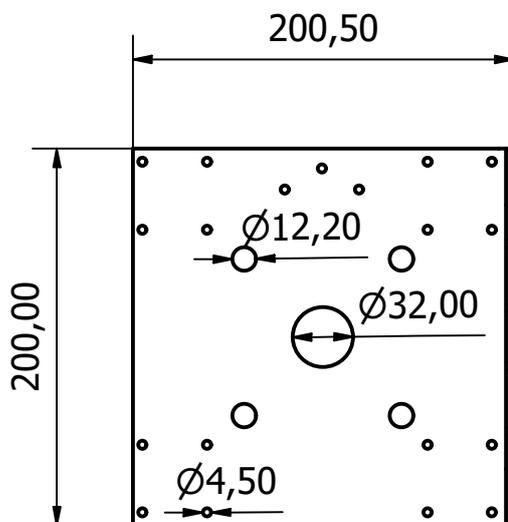
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BASE1

FECHA: FEBRERO-2020

ESCALA: 1:4

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 01

A4

1

2

3

4

A

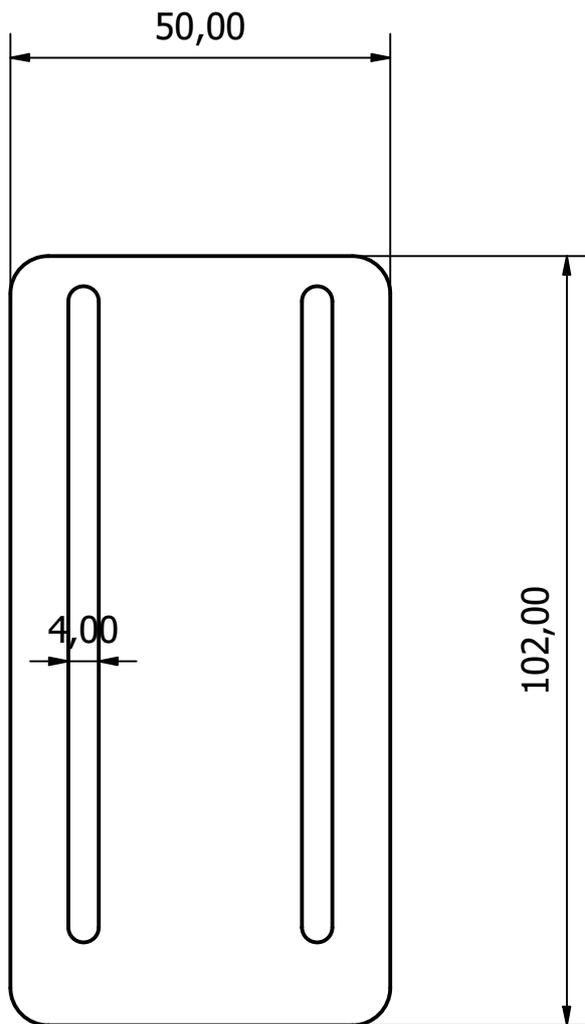
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BASE2

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 02

A4

1

2

3

4

A

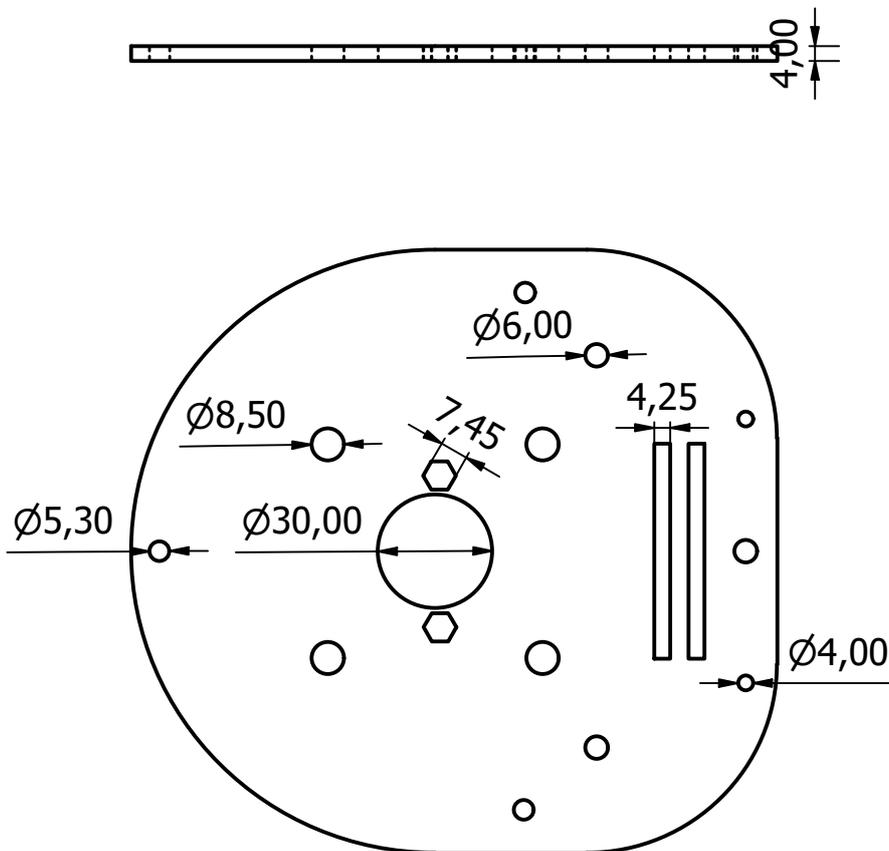
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BASE3

FECHA: FEBRERO-2020

ESCALA: 1:2

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 03

A4

1

2

3

4

A

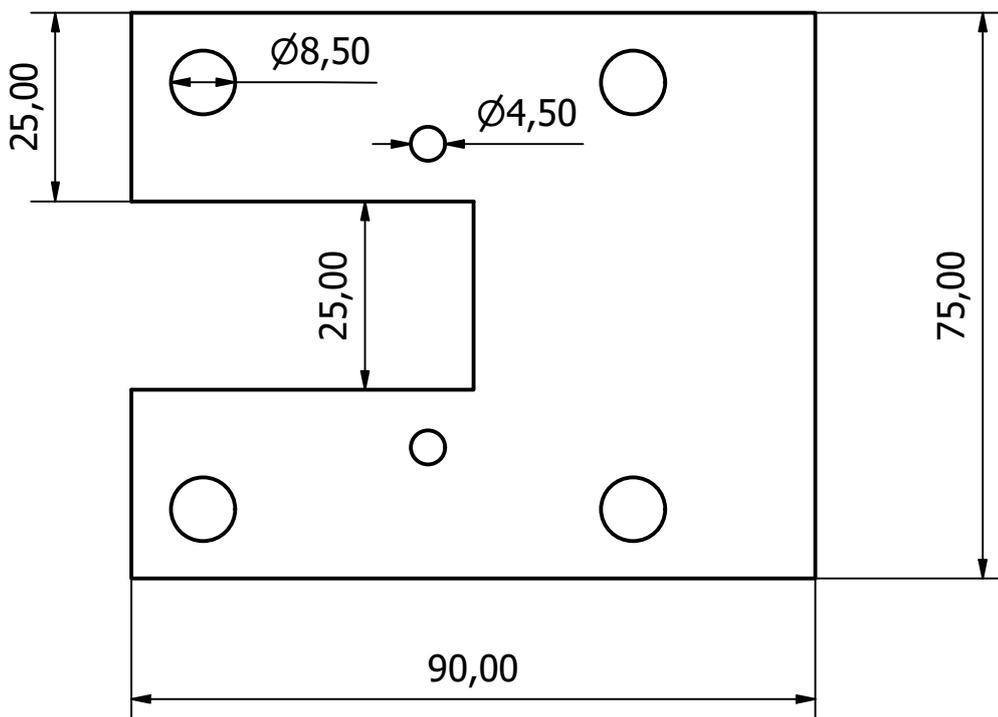
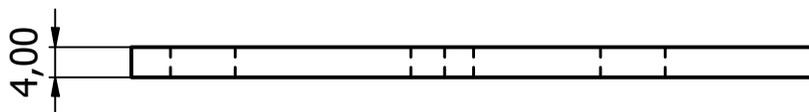
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BASE4

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 04

A4

1

2

3

4

A

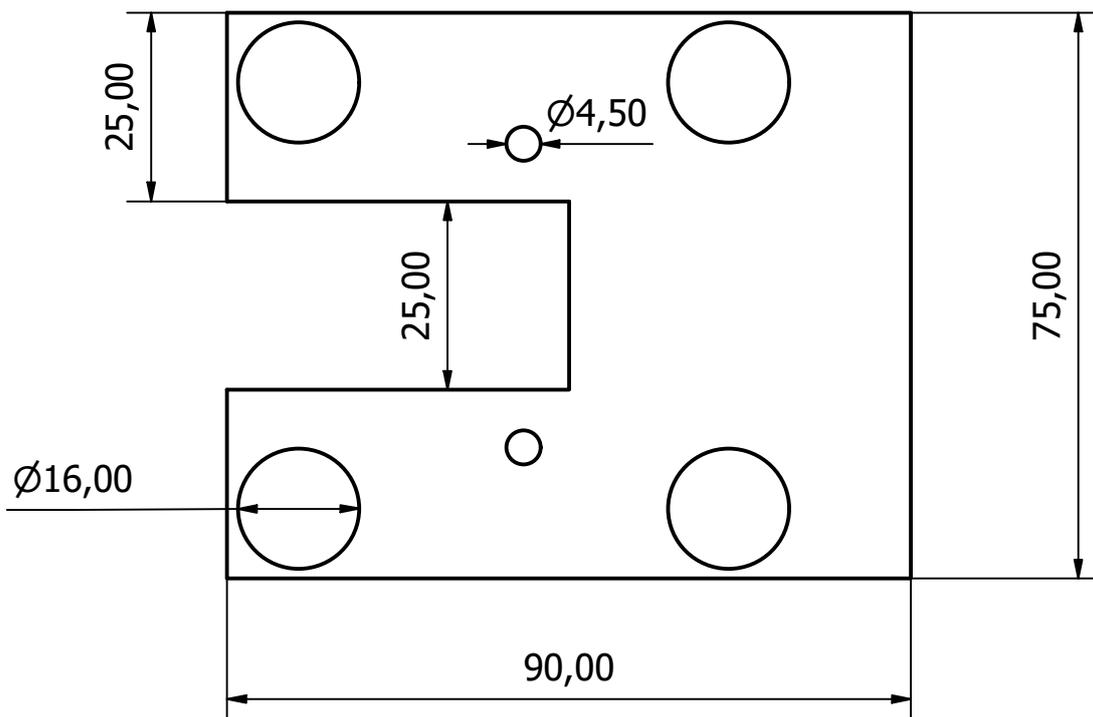
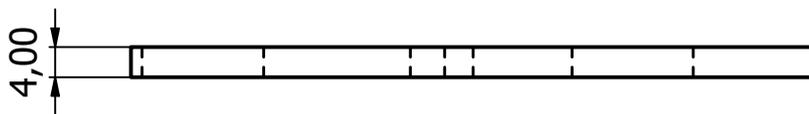
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BASE5

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 05

A4

1

2

3

4

A

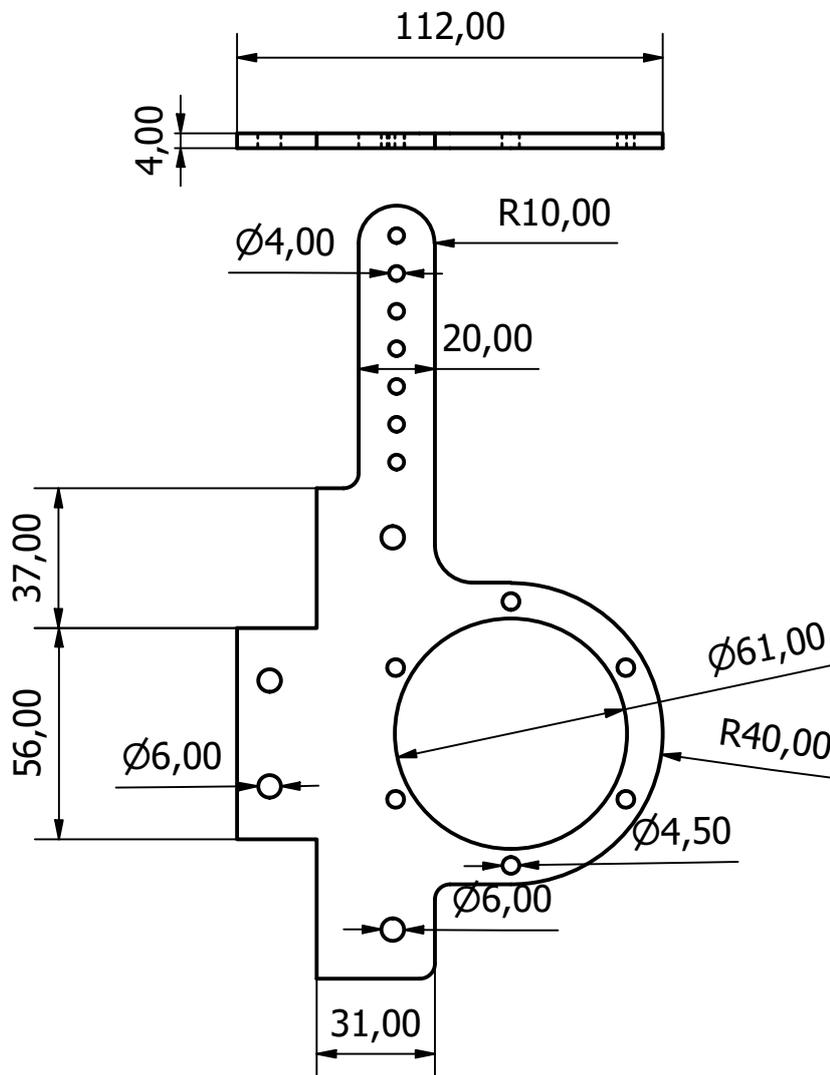
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

CUERPO1

FECHA: FEBRERO-2020

ESCALA: 1:2

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 06

A4

1

2

3

4

A

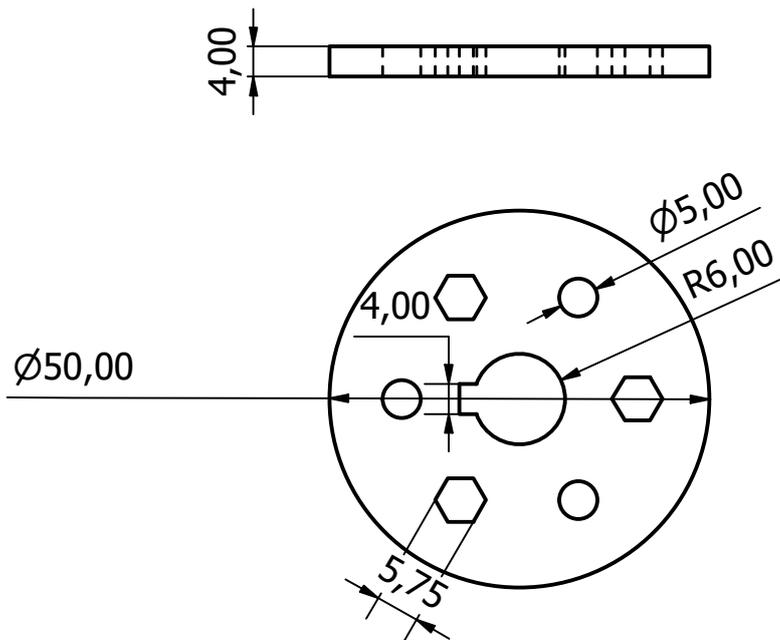
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

CUERPO2

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 07

A4

1

2

3

4

A

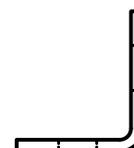
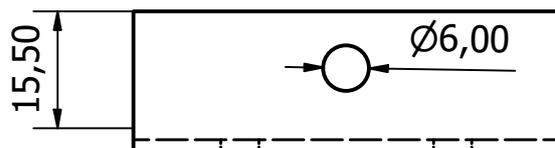
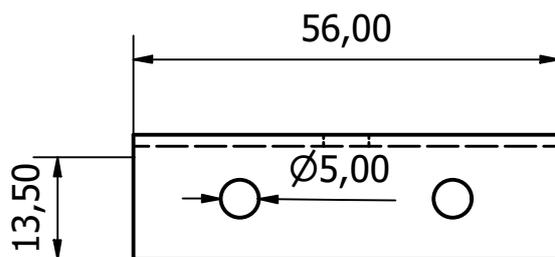
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

CUERPO3

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 08

A4

1

2

3

4

A

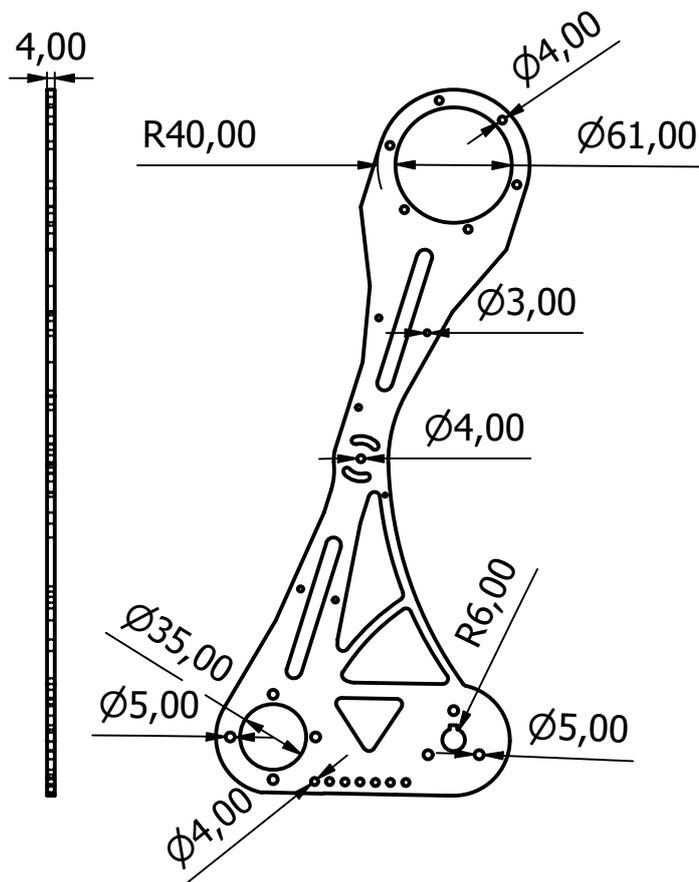
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BRAZO1

FECHA: FEBRERO-2020

ESCALA: 1:4

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 09

1

2

3

4

A

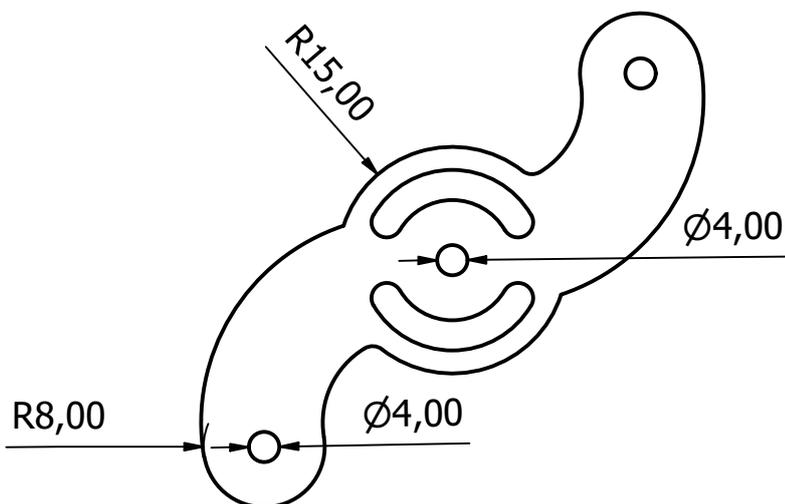
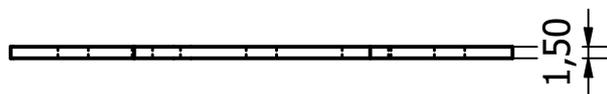
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SENSOR1

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 10

A4

1

2

3

4

A

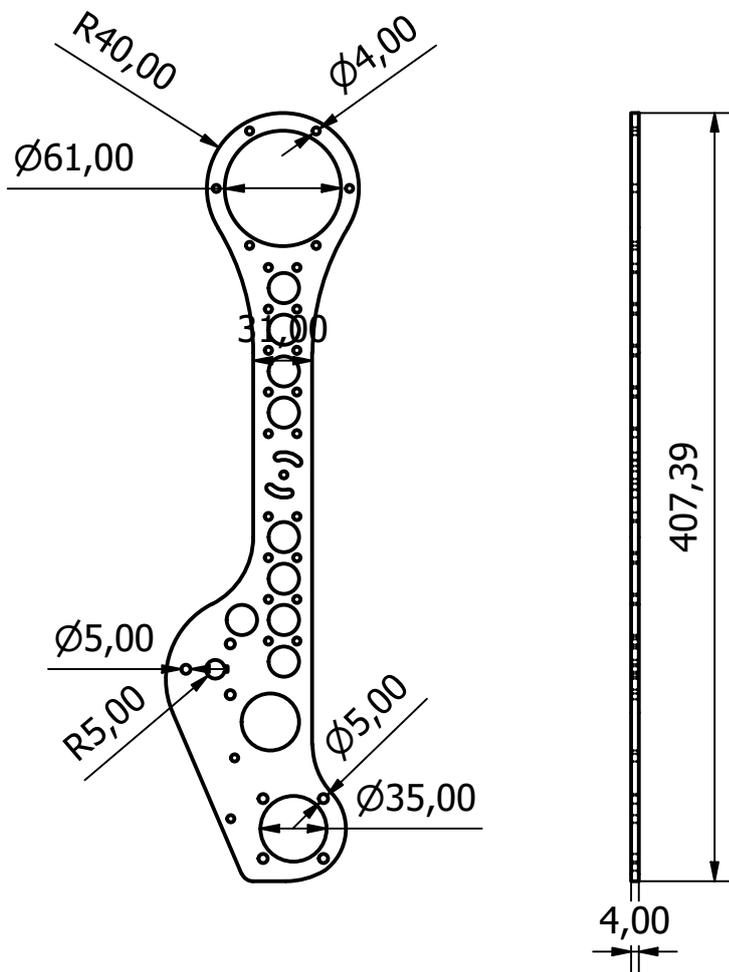
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

ANTEBRAZO1

FECHA: FEBRERO-2020

ESCALA: 1:4

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 11

A4

1

2

3

4

A

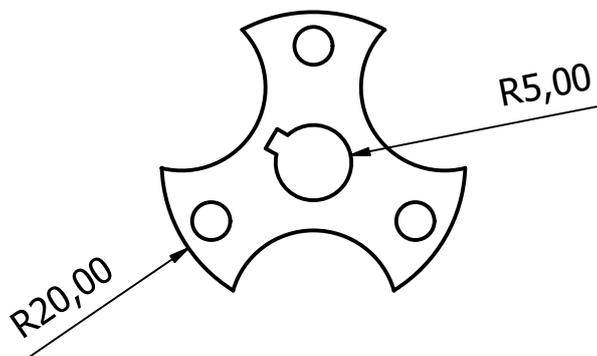
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

ANTEBRAZO2

FECHA: FEBRERO-2020

ESCALA: 1:1

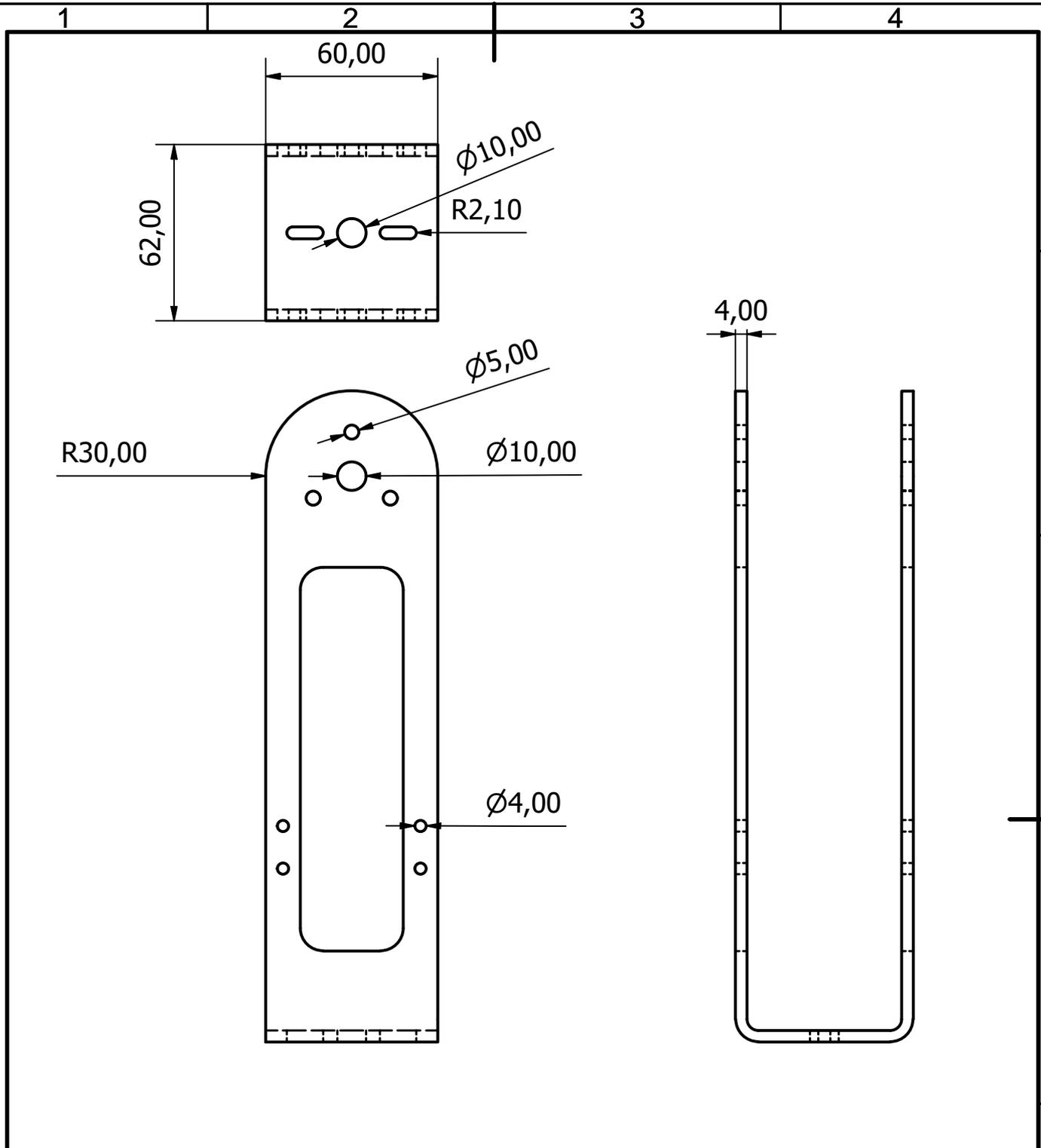
AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 12

A4



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

MUÑECA1

FECHA: FEBRERO-2020

ESCALA: 1:2

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 13

1

2

3

4

A

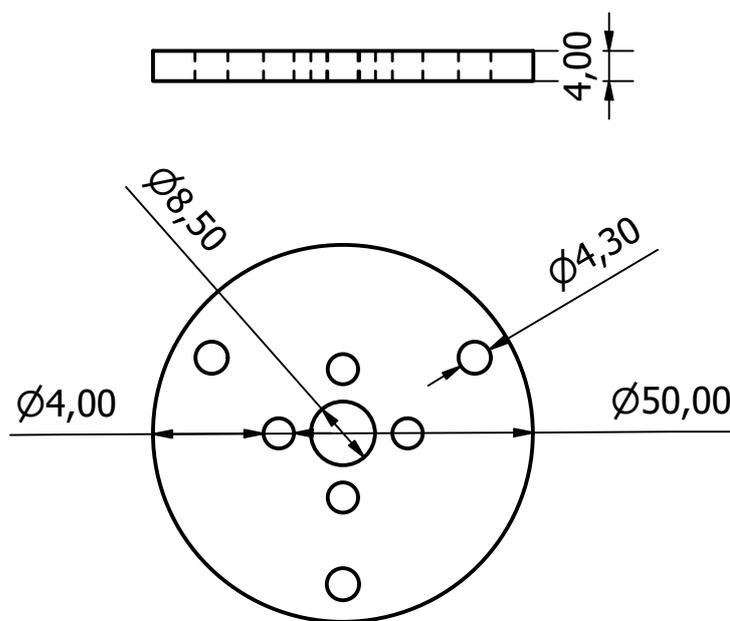
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

BRIDA1

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 14

A4

1

2

3

4

A

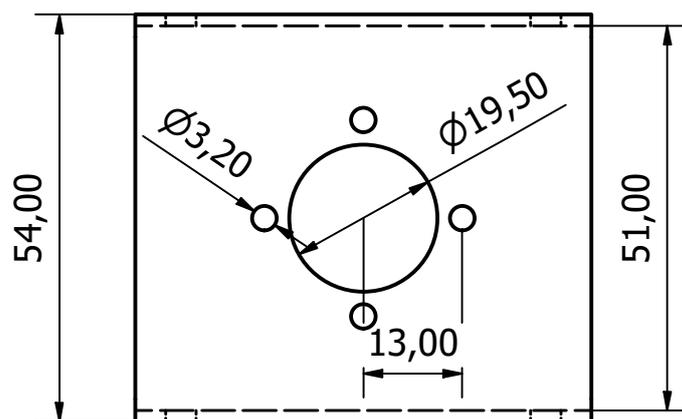
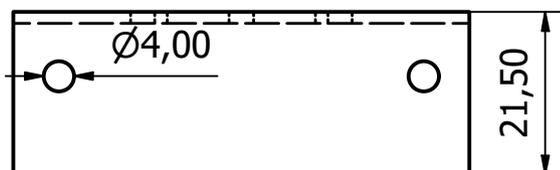
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE MOTOR MUÑECA

FECHA: FEBRERO-2020

ESCALA: 1:1

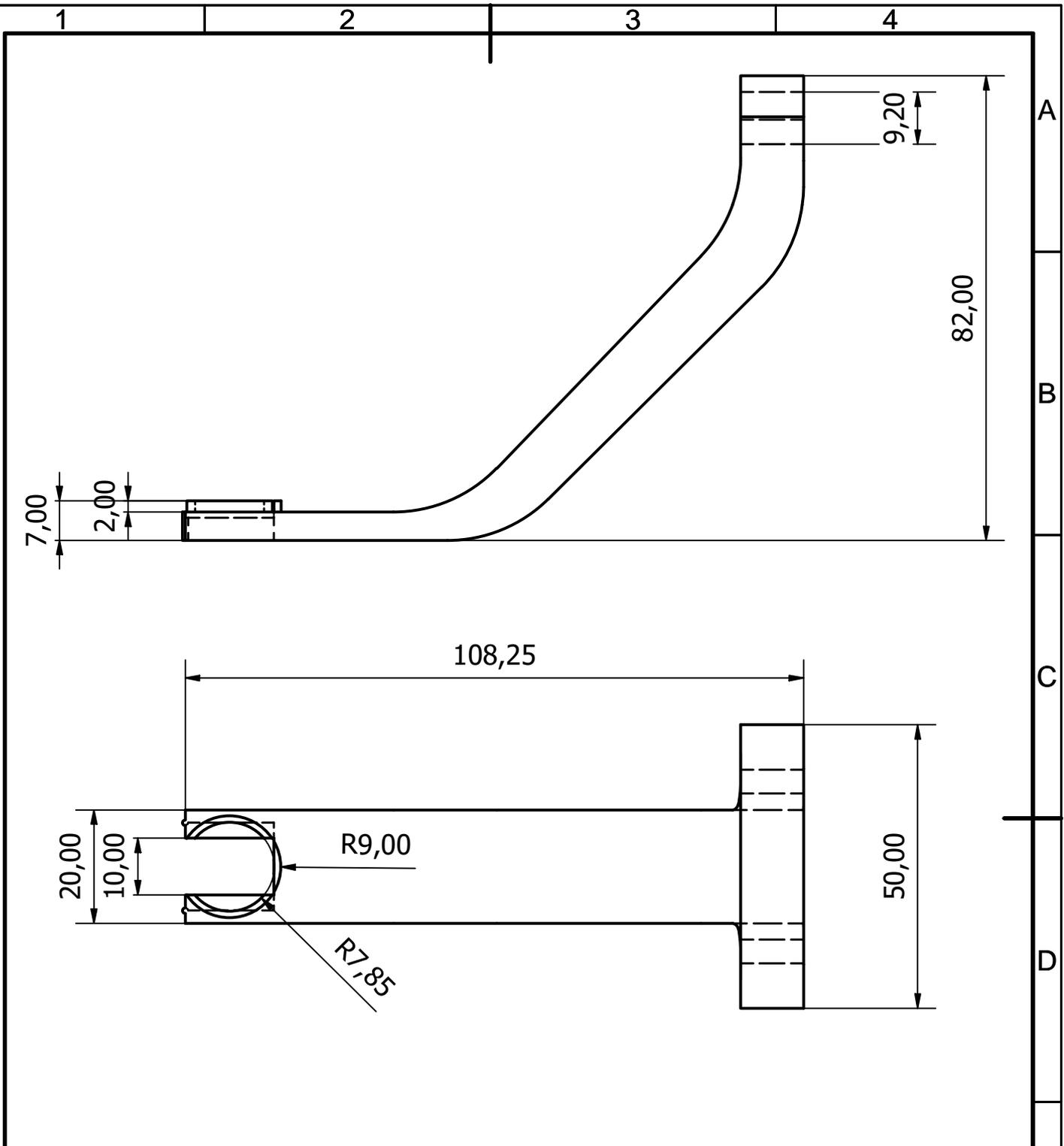
AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 15

A4



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE ENCODER BASE

FECHA: FEBRERO-2020

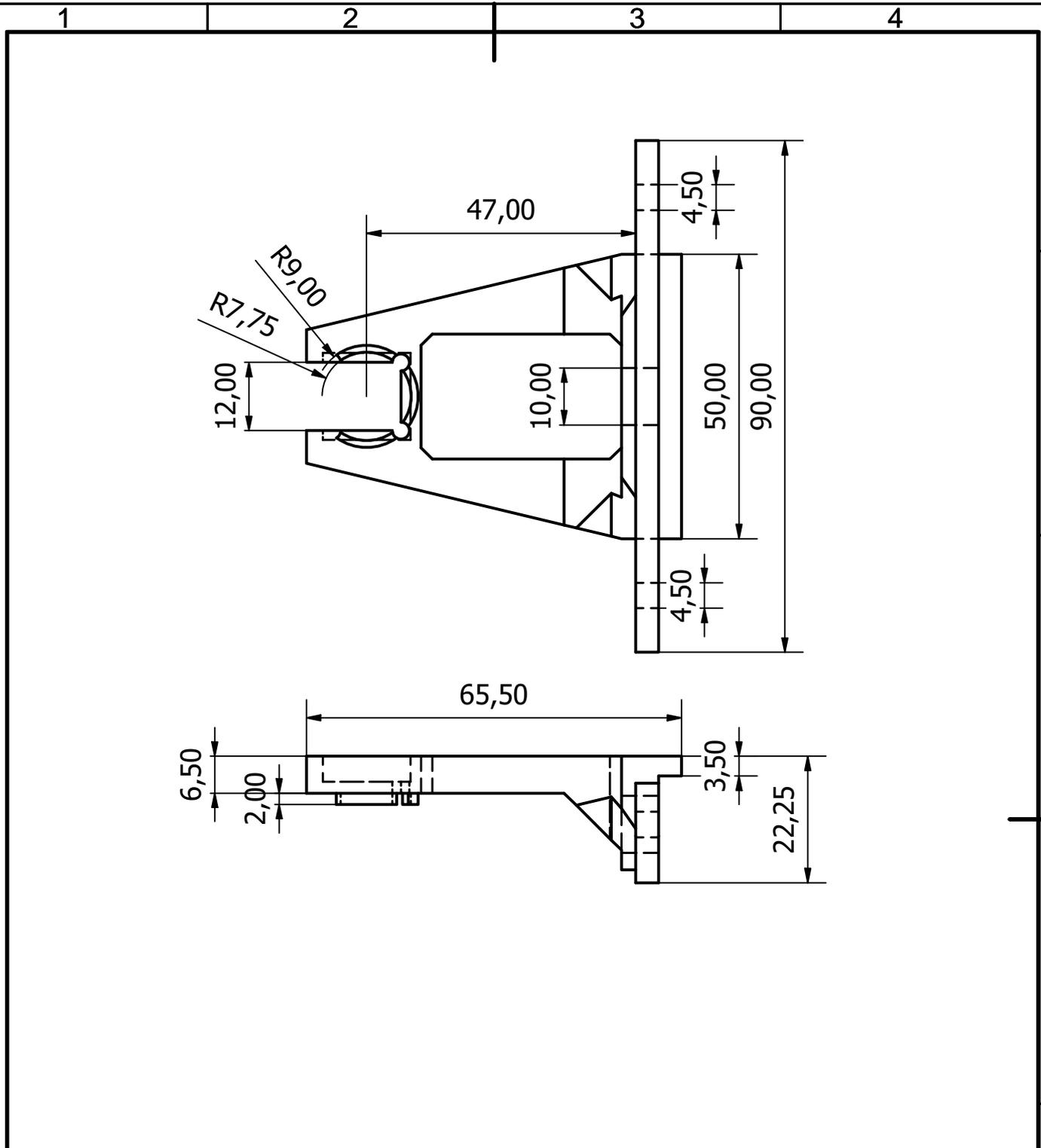
ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 16



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE ENCODER HOMBRO

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 17

1

2

3

4

A

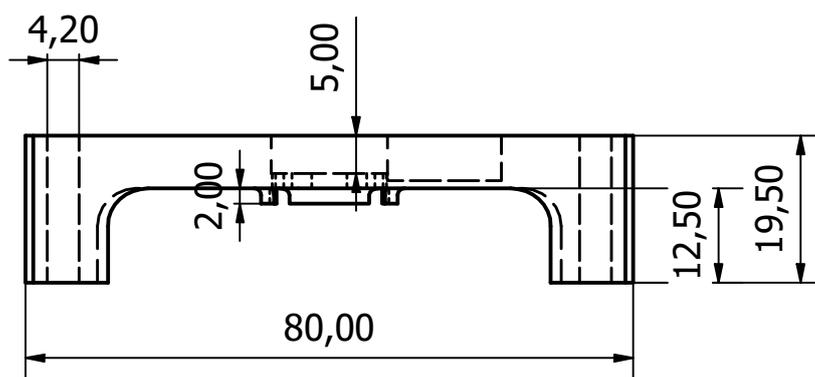
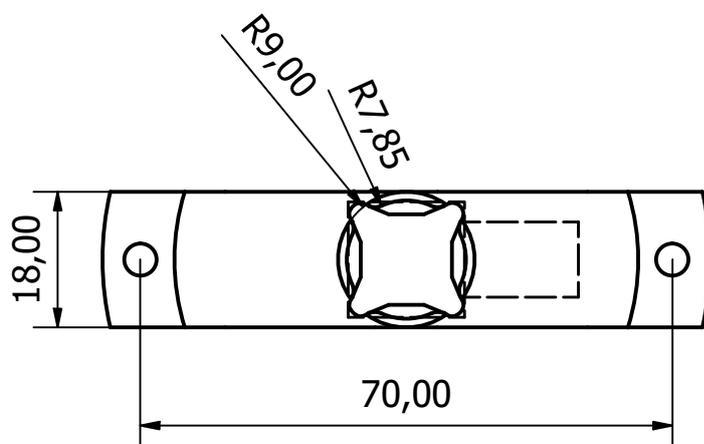
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE ENCODER CODO

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 18

A4

1

2

3

4

A

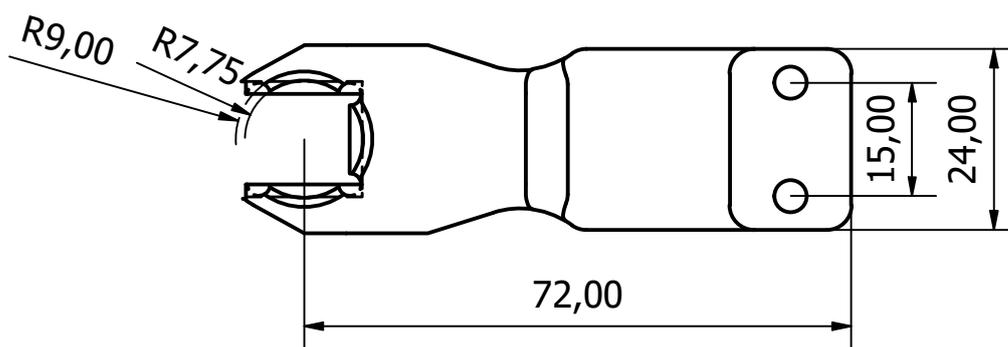
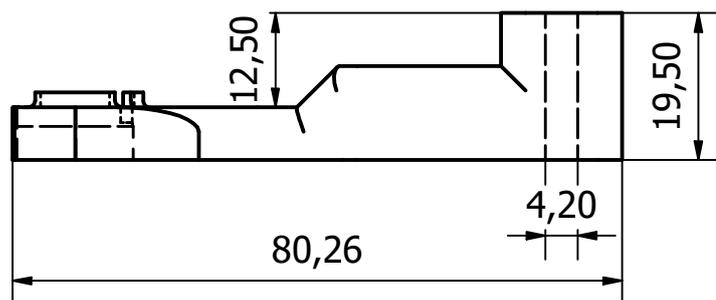
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE ENCODER MUÑECA

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 19

A4

1

2

3

4

A

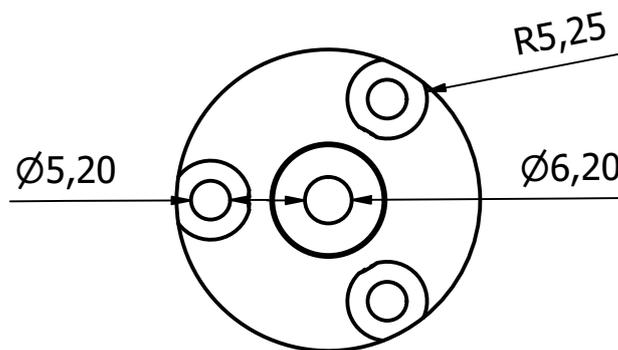
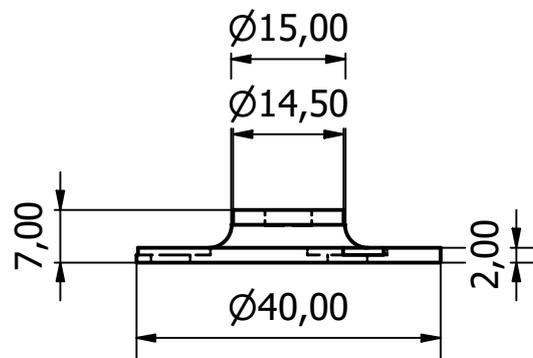
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE IMÁN BRAZO, CODO Y MUÑECA

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 20

A4

1

2

3

4

A

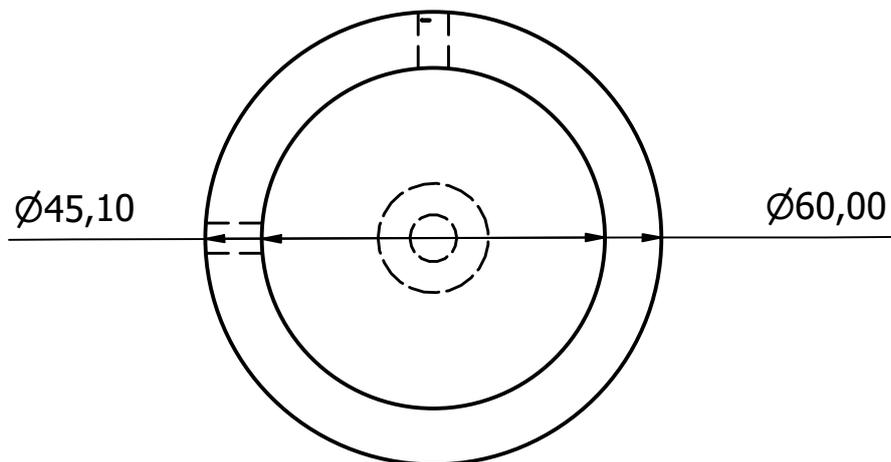
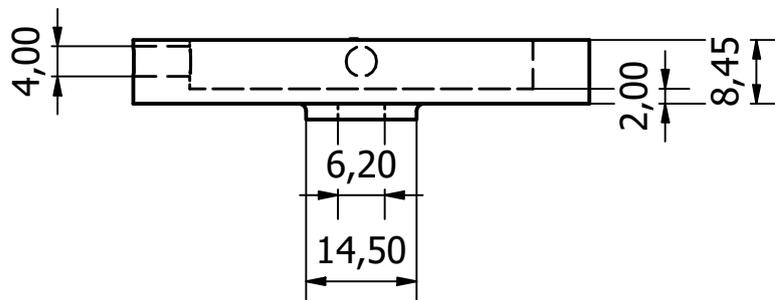
B

C

D

E

F



UNIVERSIDADE DA CORUÑA

ESCUELA UNIVERSITARIA POLITÉCNICA

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

TFG Nº: 770G01A189

TÍTULO DEL TFG:

DISEÑO, MONTAJE Y CONTROL DE UN MANIPULADOR ROBÓTICO

TÍTULO DEL PLANO:

SOPORTE IMÁN BASE

FECHA: FEBRERO-2020

ESCALA: 1:1

AUTOR:

ANTONIO REGUEIRA PEÑAMARIA

FIRMA:

PLANO Nº: 21

A4

TÍTULO: **Diseño, montaje y control de un manipulador robótico.**

PLIEGO DE CONDICIONES

PETICIONARIO: **ESCUELA UNIVERSITARIA POLITÉCNICA**

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: **FEBRERO DE 2020**

AUTOR: **EL ALUMNO**

Fdo.: **ANTONIO REGUEIRA PEÑAMARÍA**

Índice del documento PLIEGO DE CONDICIONES

14 Especificaciones software	155
15 Especificaciones hardware	155
16 Condiciones almacenamiento	155

14 Especificaciones software

- La librería TeensyStep.h puede dar problemas si no se usa en una placa Teensy debido a que hace uso de los timers y estos tienen diferente configuración en las placas Arduino.
- La librería I2C puede dar errores de lectura o inestabilidad debido a la distancia a la que se encuentra el sensor de la muñeca y codo, se soluciona reduciendo la velocidad del reloj I2C en la propia librería o ajustando las resistencias pull-up de los cables SDA y SCL.

15 Especificaciones hardware

- Se recomienda el uso de cables apantallado para la transmisión de señales, de lo contrario pueden aparecer ruidos en las lecturas de sensores y comunicaciones.
- Se debe sobredimensionar ligeramente los cables que transmiten la potencia a los motores, esto evitará posibles pérdidas de energía.
- Se ha usado PLA como material de impresión, si el robot está ubicado en alguna zona con luz solar directa se recomienda usar ABS u otro polímero más resistente.
- La tornillería, al igual que la estructura debe ser de acero inoxidable, esto evita mantenimiento y posibles problemas debido a la corrosión.

16 Condiciones almacenamiento

Las condiciones de almacenamiento deben ser las más restrictivas de entre todas las piezas compradas a terceros. De manera general se recomienda que no se superen los 45°C, que no haya atmósfera saturada de polvo ni humedad relativa mayor de 80 %.

TÍTULO: **Diseño, montaje y control de un manipulador robótico.**

MEDICIONES

PETICIONARIO: **ESCUELA UNIVERSITARIA POLITÉCNICA**

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: **FEBRERO DE 2020**

AUTOR: **EL ALUMNO**

Fdo.: **ANTONIO REGUEIRA PEÑAMARÍA**

Índice del documento MEDICIONES

17 Materiales	161
18 Mano de obra	163
19 Software	164
20 Equipo	164

17 Materiales

Referencia	Descripción	Fabricante	UD
TTPA44T5150 -A-P30	Polea base	Misumi	1
TTPA15T5150 -B-P7	Polea motor base	Misumi	1
TTPA25T5150 -B-C12	Polea motor antebrazo	Misumi	1
TTPN32T5150 -A-N10-KTC31-K5	Polea antebrazo y muñeca	Misumi	2
TTPA18T5150 -A-N12	Polea motor muñeca	Misumi	1
TTPA12T5150 -A-H6	Polea tensor	Misumi	2

Tabla 17.1 – Poleas

Referencia	Descripción	Fabricante	UD
TTBU425T5-100	Correa base	Misumi	1
TTBU780T5-150	Correa antebrazo	Misumi	1
TTBU800T5-150	Correa muñeca	Misumi	1

Tabla 17.2 – Correas

Referencia	Descripción	Fabricante	UD
CSTA30-190	Soporte + eje 30mm base	Misumi	1
HBLSS6	Escuadra soporte base-cuerpo	Misumi	2
AUT12-80	Muelle contrapeso brazo	Misumi	2
KZAP10-150-KA0- HA100-KB102-HB48	Eje codo	Misumi	1
AR-16-8X8	Acople eje motor rotación muñeca	Misumi	1
SC0610CKP4	Brida soporte herramienta	Misumi	1
SCB12-50	Tornillo rodamientos base	Misumi	4
SM-EG3-RDG	Pinza eléctrica	Robotdigg	1
AS5600	Encoder magnético	Aliexpress	4
Cable 6x28AWG	Cable señal encoder	Aliexpress	4
RVK 4G1	cable potencia	Suministrador local	7
Teensy 3.6	Microcontrolador	Amazon	1
Tca9548a	Multiplexor	Amazon	1
Breadboard 830	Placa prototipos	Amazon	1
Cables señal 1.17mm	Lógica de drivers	Amazon	1
Fuente 5v	Alimentación encoders	Amazon	1
fuentes 36V 16.7A	Alimentación motores	Aliexpress	1
Cable microUSB	Alimentación Teensy	Amazon	1
PLA gris	1.75mm 1kg	Amazon	1
Jumpers 20cm	Conexión breadboard	Amazon	1

Tabla 17.3 – Piezas genéricas hardware

Referencia	UD
Base1	1
Base2	2
Base3	2
Base4	1
Base5	2
Cuerpo1	2
Cuerpo2	2
Cuerpo3	1
Brazo1	2
Tensor1	2
Antebrazo1	2
Antebrazo2	11
Muñeca1	1
Brida1	1
Soporte motor muñeca	1

Tabla 17.4 – Piezas en acero

Referencia	Fabricante	UD
17HS19-1684S-PG19	Stepperonline	2
23HS22-2804S-PG47	Stepperonline	2
23HS22-2804S-PG15	Stepperonline	1
DM542T	Stepperonline	5
TB6600	Aliexpress	1

Tabla 17.5 – Motores y drivers

18 Mano de obra

Tipo de mano de obra	UD(horas)
Graduado en ingeniería electrónica industrial y automática	600

Tabla 18.1 – Mano de obra

19 Software

Programa	UD(meses)
Autodesk Inventor	3
Visual Studio Code	1
Thonny	1

Tabla 19.1 – Software

20 Equipo

Equipo	Tiempo usado (meses)
Ordenador	6
Impresora 3D	3

Tabla 20.1 – Equipo

TÍTULO: **Diseño, montaje y control de un manipulador robótico.**

PRESUPUESTO

PETICIONARIO: **ESCUELA UNIVERSITARIA POLITÉCNICA**

AVDA. 19 DE FEBREIRO, S/N

15405 - FERROL

FECHA: **FEBRERO DE 2020**

AUTOR: **EL ALUMNO**

Fdo.: **ANTONIO REGUEIRA PEÑAMARÍA**

Índice del documento PRESUPUESTO

21 Materiales	169
22 Mano de obra	171
23 Software	172
24 Equipo	172
25 Coste total	172

21 Materiales

Referencia	Descripción	Fabricante	UD	Precio UD (€)	Precio total (€)
TTPA44T5150 -A-P30	Polea base	Misumi	1	40.53	40.53
TTPA15T5150 -B-P7	Polea motor base	Misumi	1	22.39	22.39
TTPA25T5150 -B-C12	Polea motor antebrazo	Misumi	1	34.98	34.98
TTPN32T5150 -A-N10-KTC31-K5	Polea antebrazo y muñeca	Misumi	2	48.42	96.84
TTPA18T5150 -A-N12	Polea motor muñeca	Misumi	1	23.03	23.03
TTPA12T5150 -A-H6	Polea tensor	Misumi	2	14.15	28.30
				TOTAL	246.07

Tabla 21.1 – Poleas

Referencia	Descripción	Fabricante	UD	Precio UD (€)	Precio total (€)
TTBU425T5-100	Correa base	Misumi	1	7.30	7.30
TTBU780T5-150	Correa antebrazo	Misumi	1	14.50	14.50
TTBU800T5-150	Correa muñeca	Misumi	1	14.50	14.50
				TOTAL	36.30

Tabla 21.2 – Correas

Referencia	Descripción	Fabricante	UD	Precio UD (€)	Precio total (€)
CSTA30-190	Soporte + eje 30mm base	Misumi	1	25	25
HBLSS6	Escuadra soporte base-cuerpo	Misumi	2	0.78	1.86
AUT12-80	Muelle contrapeso brazo	Misumi	2	4.70	9.40
KZAP10-150-KA0- HA100-KB102-HB48	Eje codo	Misumi	1	14.40	14.40
AR-16-8X8	Acople eje motor rotación muñeca	Misumi	1	11.09	11.09
SC0610CKP4	Brida soporte herramienta	Misumi	1	6.01	6.01
SCB12-50	Tornillo rodamientos base	Misumi	4	3	12
SM-EG3-RDG	Pinza eléctrica	Robotdigg	1	189	189
AS5600	Encoder magnético	Aliexpress	4	3.38	13.52
Cable 6x28AWG	Cable señal encoder	Aliexpress	4	0.58	2.32
RVK 4G1	cable potencia	Suministrador local	7	0.52	3.64
Teensy 3.6	Microcontrolador	Amazon	1	47.99	47.99
Tca9548a	Multiplexor	Amazon	1	2.43	2.43
Breadboard 830	Placa prototipos	Amazon	1	9.59	9.59
Cables señal 1.17mm	Lógica de drivers	Amazon	1	7.88	7.88
Fuente 5v	Alimentación encoders	Amazon	1	8.99	8.99
fuentes 36V 16.7A	Alimentación motores	Aliexpress	1	26	26
Cable microUSB	Alimentación Teensy	Amazon	1	6.99	6.99
PLA gris	1.75mm 1kg	Amazon	1	21.99	21.99
Jumpers 20cm	Conexión breadboard	Amazon	1	3.99	3.99
				TOTAL	424.01

Tabla 21.3 – Piezas genéricas hardware

Referencia	UD	Precio UD(€)	Precio total(€)
Base1	1	19.46	19.46
Base2	1	1.81	1.81
Base3	1	5.80	5.80
Base4	1	2.16	2.16
Base5	2	2.25	4.50
Cuerpo1	2	5.15	10.30
Cuerpo2	2	1.47	2.94
Cuerpo3	1	10.75	10.75
Brazo1	2	11.82	23.64
Tensor1	2	0.79	1.58
Antebrazo1	2	10.44	20.88
Antebrazo2	11	1.06	11.66
Muñeca1	1	18.03	18.03
Brida1	1	1.51	1.51
Soporte motor muñeca	1	11.66	11.66
		TOTAL	146.79

Tabla 21.4 – Piezas en acero

Referencia	Fabricante	UD	Precio UD(€)	Precio total(€)
17HS19-1684S-PG19	Stepperonline	2	27.34	54.68
23HS22-2804S-PG47	Stepperonline	2	51.90	103.80
23HS22-2804S-PG15	Stepperonline	1	51.90	51.90
DM542T	Stepperonline	5	32.29	161.45
TB6600	Aliexpress	1	5.78	5.78
			TOTAL	328.61

Tabla 21.5 – Motores y drivers

22 Mano de obra

Tipo de mano de obra	UD(horas)	Honorarios(€/h)	Precio total(€)
Graduado en ingeniería electrónica industrial y automática	600	40	24000

Tabla 22.1 – Mano de obra

23 Software

Programa	UD(meses)	Precio UD(€)	Coste total(€)
Autodesk Inventor	3	327	981
Visual Studio Code	1	0	0
Thonny	1	0	0

Tabla 23.1 – Software

24 Equipo

Equipo	Total(€)	Tiempo amortización(meses)	Tiempo usado (meses)	Coste total(€)
Ordenador	1000	24	6	250
Impresora 3D	400	24	3	50

Tabla 24.1 – Equipo

25 Coste total

Concepto	Precio total(€)
Materiales	1546,24
Mano de obra	24000
Software	981
Equipo	250
TOTAL	26.777,24
TOTAL(+21 % IVA)	32.400,46

Tabla 25.1 – Coste total

El presupuesto total asciende a la cantidad de: **treinta y dos mil cuatrocientos con cuarenta y seis euros (32.400,46€)**