

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y  
AUTOMÁTICA INDUSTRIAL



**UNIVERSITAS**  
*Miguel Hernández*



“DESARROLLO DE UN SISTEMA DE  
FOTOGRAFIADO AUTOMATIZADO EN  
CÁMARAS DE CULTIVO”

**TRABAJO FIN DE GRADO**

Diciembre – 2025

AUTOR: David Olivares Climent

DIRECTOR: Adrián Peidró Vidal

## Índice de contenidos

1	Introducción a la memoria.....	4
1.1	Antecedentes .....	4
1.2	Objetivo.....	6
1.3	Estructura de la memoria.....	6
2	Diseño y desarrollo del sistema.....	8
2.1	Componentes.....	8
2.1.1	Raspberry Pi Modelo 4 B.....	12
2.1.2	Fuente de alimentación.....	12
2.1.3	Motores paso a paso.....	13
2.1.4	Controladores de los motores paso a paso .....	13
2.1.5	Cámara USB.....	14
2.1.6	Plataforma .....	14
2.1.7	Carro portador.....	15
2.1.8	Poleas .....	16
2.1.9	Rodamientos .....	16
2.1.10	Acoples dentados GT2.....	17
2.1.11	Correas dentadas GT2.....	17
2.1.12	Actuadores lineales .....	18
2.1.13	Diodos rectificadores 1n4007 .....	19
2.1.14	Regulador Buck DC-DC LM2596.....	19
2.1.15	Módulo de relés .....	20
2.1.16	Sensores Hall e imanes .....	20
2.1.17	Acelerómetro.....	21
2.1.18	Impresora 3D.....	21
2.1.19	Tambor .....	21
2.1.20	Cilindro-chaveta .....	22
2.1.21	Soportes de los actuadores lineales .....	22
2.1.22	Ejes de los tambores .....	23
2.1.23	Anillas de retención.....	23
2.1.24	Soportes auxiliares para rodamientos.....	24
2.1.25	Ejes secundarios.....	24

2.1.26	Guías para el hilo .....	25
2.1.27	Ruedas del carro portador.....	25
2.2	Metodología de montaje funcional.....	26
2.2.1	Montaje del mecanismo de transmisión primario .....	26
2.2.2	Montaje del mecanismo de transmisión secundario .....	33
2.2.3	Sistema de movimiento del carro portador .....	37
2.2.4	Trazado de los hilos .....	39
3	Funcionamiento general del sistema .....	42
3.1	Funcionamiento mecánico .....	42
3.1.1	Recorrido físico de la plataforma .....	42
3.1.2	Retorno de posición de la plataforma .....	44
3.1.3	Movimiento horizontal del carro portador .....	46
3.1.4	Control de inclinación de la plataforma .....	48
3.2	Funcionamiento lógico de programación.....	52
3.2.1	La función de toma de fotografías.....	54
3.2.2	Movimiento vertical del sistema .....	54
3.2.3	Movimiento horizontal del sistema .....	56
3.2.4	Código principal.....	58
3.2.4.1	Inclinación de la plataforma .....	60
3.2.4.2	Ausencia de detección de códigos QR.....	64
3.2.4.3	Detección de códigos QR.....	66
3.2.4.4	Retorno de la plataforma a la posición inicial.....	68
4	Conexiones eléctricas de los componentes .....	69
4.1	Conexión de la fuente de alimentación.....	69
4.2	Conexión de los actuadores lineales en el módulo de relés.....	70
4.3	Conexión de los pines de control de los relés .....	71
4.4	Conexión de los controladores de los motores.....	72
4.5	Conexión de los sensores Hall y acelerómetro.....	73
4.6	Conexiones de los pines en la Raspberry Pi .....	74
5	Resultados .....	75
6	Conclusiones .....	80
7	Trabajos futuros.....	81
8	Referencias bibliográficas.....	82

9	Anexo I: Planos de conjunto y de despiece.....	84
10	Anexo II: Código del sistema .....	94



## Índice de figuras

Figura 1. Trazado del movimiento en zigzag a través de la estantería. ....	4
Figura 2. Imágenes de la cámara de cultivo: a) La estantería, b) Disposición de las placas en la balda. ....	5
Figura 3. Plano general del mecanismo de transmisión completo. ....	9
Figura 4. Trayectoria de explosión de la transmisión primaria. ....	10
Figura 5. Trayectoria de explosión de la transmisión secundaria. ....	10
Figura 6. Montaje de la transmisión primaria. ....	11
Figura 7. Montaje de la transmisión secundaria. ....	11
Figura 8. Raspberry Pi Modelo 4B. ....	12
Figura 9. Fuente de alimentación. ....	13
Figura 10. Motor paso a paso Nema 17. ....	13
Figura 11. Controladores de motores paso a paso TB6600. ....	14
Figura 12. Cámara USB. ....	14
Figura 13. Vistas de la plataforma. ....	15
Figura 14. Vista frontal del carro portador. ....	15
Figura 15. Ubicaciones de las poleas: a) En el carro portador, b) En la plataforma. ....	16
Figura 16. Vista de un rodamiento. ....	16
Figura 17. Vistas del acople dentado GT2. ....	17
Figura 18. Correa GT2 abierta unida a los distintos elementos: a) Al carro portador, b) Engranada con el acople GT2 del motor, c) Engranada con la polea dentada del soporte auxiliar. ....	18
Figura 19. Correa dentada GT2 cerrada engranando los acoples dentados de las transmisiones. ....	18
Figura 20. Actuador lineal 1039B, 1.5-35N, 12mm, 24V. ....	19
Figura 21. Características del actuador lineal 1039B. ....	19
Figura 22. Diodo rectificador 1n4007. ....	19
Figura 23. Regulador Buck LM2596. ....	20
Figura 24. Módulo de 4 relés con optoacopladores. ....	20
Figura 25. Sensor Hall OH137. ....	21
Figura 26. Acelerómetro adxl345. ....	21
Figura 27. Vista general del tambor. ....	22
Figura 28. Vista isométrica del cilindro-chaveta. ....	22
Figura 29. Vistas del soporte del actuador lineal. ....	23
Figura 30. Vista isométrica del eje del tambor. ....	23
Figura 31. Vista isométrica de la anilla de retención. ....	23
Figura 32. Vista isométrica del soporte auxiliar. ....	24
Figura 33. Vista isométrica del eje secundario. ....	25
Figura 34. Guías ubicadas en: a) Frente a los tambores, b) El carro portador, c) La plataforma. ....	25
Figura 35. Ruedas del carro portador: a) Bombeadas, b) Cónicas. ....	25
Figura 36. Ranura en forma de 'D' en el tambor. ....	27
Figura 37. Agujero pasante del tambor. ....	27

Figura 38. Distintas vistas del tambor.....	27
Figura 39. a) Inserción del eje del tambor en el rodamiento, b) Inserción del eje del tambor en el rodamiento junto a la anilla de retención, c) Montaje final de los elementos anteriores en el tambor. ....	28
Figura 40. Muesca del agujero pasante del tambor.....	28
Figura 41. Vistas del actuador. a) En su soporte, b) Anclaje de los extremos mediante el alambre. ....	29
Figura 42. Vista de los extremos rectangulares de los ejes del actuador y del tambor. ....	29
Figura 43. a) Apoyo del tambor sobre el cilindro, b) Distintas posiciones del cilindro-chaveta con respecto al tambor. ....	30
Figura 44. Elementos acoplados al eje del motor. ....	31
Figura 45. Posiciones del actuador: a) Actuador no alimentado. Muelle extendido con el tambor en su posición natural, b) Actuador alimentado. Muelle comprimido con el tambor desplazado.....	32
Figura 46. Vista general del mecanismo apoyado sobre la tabla fija. ....	33
Figura 47. a) Eje secundario, b) Eje secundario insertado en el rodamiento del soporte auxiliar.....	34
Figura 48. Acople de los ejes secundarios.....	35
Figura 49. Vistas del ajuste de los elementos de la segunda transmisión montada. ....	35
Figura 50. Bloque de transmisión conformado por la transmisión primaria y secundaria, acopladas mediante la correa GT2 cerrada. ....	36
Figura 51. Perfil de la viga principal y su anclaje mediante las escuadras metálicas. ....	37
Figura 52. Disposición de las diferentes ruedas del carro portador alrededor de la viga principal.....	38
Figura 53. Disposición de las ruedas horizontales en el carro portador. ....	38
Figura 54. Guía para el hilo frente al tambor. ....	39
Figura 55. Vistas de las guías del carro portador ubicadas antes de las poleas. ....	39
Figura 56. Guías de la plataforma junto a las poleas.....	40
Figura 57. Esquema físico del mecanismo de doble polea.....	41
Figura 58. Recorrido de la plataforma a través de las placas durante el funcionamiento.....	42
Figura 59. Vista de un montaje experimental con el sensor Hall vertical: a) Posición del sensor en el carro portador, b) Proximidad entre el imán y el sensor. ....	44
Figura 60. Transmisión del carro portador con la correa GT2: a) Polea dentada auxiliar ubicada en el extremo izquierdo, b) Anclaje de la correa GT2 al carro portador, c) Unión con el acople GT2 del motor horizontal.....	46
Figura 61. Vistas del cable USB colgando del sistema: a) Catenaria horizontal con anclaje al carro portador, b) Cable colgando según la altura de la plataforma. ....	47
Figura 62. Sistema experimental de cables colgantes similar a los puentes grúa: a) Catenarias ideales, b) Flexión de los cables por su propia rigidez. ....	47
Figura 63. a) Engrane del tambor, b) Engrane del soporte del actuador, c) Acople entre ambos engranes.....	49
Figura 64. Posiciones de la chaveta respecto al tambor: a) Enclavijado de la chaveta en una ranura del eje estriado del tambor, b) Chaveta en contacto con la pared del tambor.....	49
Figura 65. Redondeo de la chaveta. ....	50

Figura 66. Vista desde diferentes ángulos de los redondeamientos de las aristas del eje nervado del tambor. ....	50
Figura 67. Función para crear códigos QR. ....	52
Figura 68. Posible ubicación de los QR en la estantería. ....	53
Figura 69. Código de la función de tomar fotografía. ....	54
Figura 70. Funciones del movimiento vertical. ....	55
Figura 71. Funciones del movimiento horizontal. ....	57
Figura 72. Diagrama de flujo del funcionamiento general del sistema. ....	59
Figura 73. Sistemas de referencia: a) De los ejes del acelerómetro, b) Sobre la plataforma. ....	60
Figura 74. Disposición y nomenclatura de los tambores, actuadores y motores. ....	61
Figura 75. Inclinación de la plataforma sobre el eje X. ....	61
Figura 76. Inclinación de la plataforma en el eje Y. ....	62
Figura 77. Esquema de desplazamientos del movimiento en espiral. ....	65
Figura 78. Detección de códigos QR y sus centroides. ....	67
Figura 79. Código del retorno de la plataforma a su posición inicial. ....	68
Figura 80. Diagrama general de conexiones de los elementos del sistema junto sus voltajes de operación. ....	69
Figura 81. Alimentación eléctrica: a) Conexión de la fuente de alimentación, b) Conexión del regulador Buck. ....	70
Figura 82. Conexiones de: a) El módulo de relés, b) Los diodos en antiparalelo a los actuadores. ....	71
Figura 83. Conexión de los pines de control y alimentación en el módulo de relés. ....	71
Figura 84. Conexión de los pines de control en los controladores de los motores. ....	72
Figura 85. Hoja de características de las conexiones de los motores paso a paso. ....	72
Figura 86. Divisor de voltaje para los sensores Hall: a) Placa de circuito, b) Diagrama eléctrico. ....	73
Figura 87. Tabla de conexiones de los pines de la Raspberry a los diversos dispositivos. ....	74
Figura 88. Vistas en diferentes posiciones de la plataforma pendida por los hilos con la cámara anclada. ....	75
Figura 89. Vista del sistema de hilos y poleas: a) En el carro portador, b) Plataforma pendida sobre los hilos que bajan del carro portador. ....	76
Figura 90. Montaje completo del sistema mecánico de transmisión. ....	76
Figura 91. Posiciones de los tambores, tanto en sus posiciones naturales como desacoplados. ....	77
Figura 92. Posición experimental del: a) Sensor Hall vertical, b) Imán en la plataforma y su proximidad entre ambos elementos. ....	77
Figura 93. Detección del QR mediante la cámara anclada a la plataforma suspendida por los hilos. ....	78
Figura 94. Diferentes posiciones del centroide del QR respecto al centro de la imagen, mostrada con un cuadrado verde. ....	79

# 1 Introducción a la memoria

## 1.1 Antecedentes

El presente proyecto deriva de un TFM previo (Cardona-Marín, 2023), cuyo objetivo es la ayuda al equipo del instituto de bioingeniería de la UMH en su trabajo diario. Este último está relacionado con el estudio de raíces de cultivos de tomate, ubicados dentro de una cámara de cultivo. A estos cultivos se les necesita realizar fotografías periódicamente durante el día, estudiando así su adaptabilidad en diferentes condiciones ambientales según el crecimiento de sus raíces.

El sistema instalado en la cámara consiste en un primer montaje experimental, cuyo logro ha consistido en realizar el movimiento de una plataforma que, con una cámara incorporada, se desplaza a lo largo de la estantería allí ubicada. La cámara se posiciona enfrente de cada una de las placas donde se encuentran los cultivos a estudiar, realizando su fotografía, mientras describe un movimiento de zigzag en su recorrido.

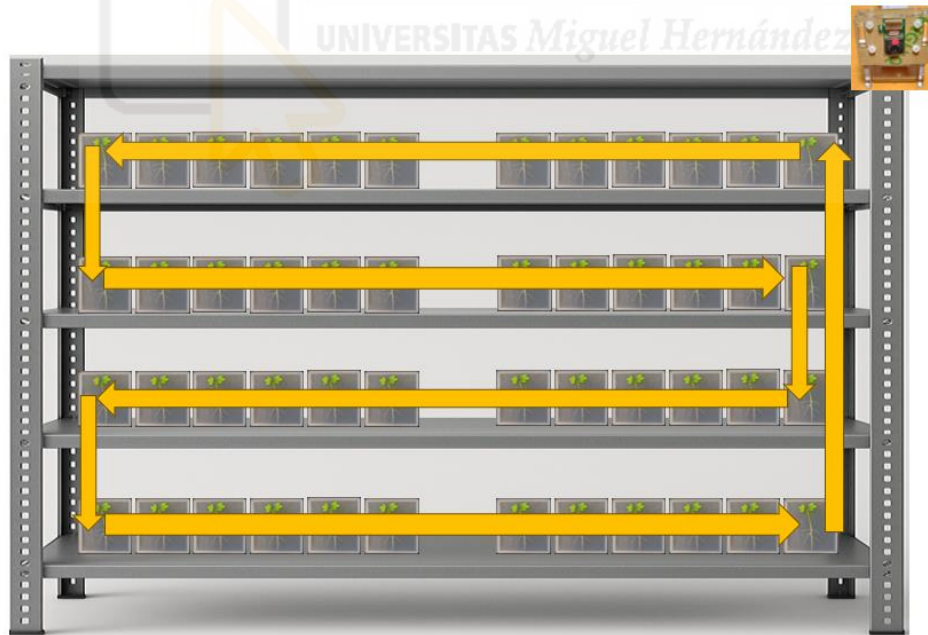


Figura 1. Trazado del movimiento en zigzag a través de la estantería.

La cámara de cultivo donde está instalado el sistema tiene unas dimensiones de 2.5 x 1.8 x 2 metros, siendo estas medidas el largo, alto y ancho, respectivamente. Dispone de dos estanterías

y de un pasillo central donde poder moverse, sin embargo, solo se utiliza una de ellas para el funcionamiento del sistema. La estantería consta de 4 baldas, en las que en cada una de ellas pueden colocarse hasta 18 placas de cultivo dispuestas en grupos de 6, por lo que la estantería completa tiene una capacidad máxima de 72 placas.



Figura 2. Imágenes de la cámara de cultivo: a) La estantería, b) Disposición de las placas en la balda.

La razón de la elaboración de este proyecto se debe a que el sistema allí instalado, además de ocupar un excesivo volumen dentro de la cámara, muestra una baja rigidez en su estructura, la cual provoca oscilaciones durante su movimiento y dificulta la correcta toma de imágenes.

En este proyecto se ha diseñado un sistema basado en hilos y poleas, donde la plataforma con la cámara anclada queda suspendida desde lo alto, cuyo objetivo es el de reducir en gran medida el espacio ocupado dentro de la cámara de cultivo, además de intentar mejorar la precisión del sistema en la toma de imágenes.

El sistema se montó inicialmente de manera provisional y experimental en un laboratorio por razones de comodidad y accesibilidad, y posteriormente trasladado parte del sistema a la cámara de cultivo. Por ese motivo, es desde ese laboratorio de donde son tomadas algunas de las imágenes mostradas en esta memoria.

## 1.2 Objetivo

El objetivo del proyecto consiste en diseñar un sistema en el que una plataforma, sujeta por varios hilos, pueda tanto descender como ascender mediante un mecanismo de poleas. Los hilos se encontrarán enrollados en unos tambores que rotarán gracias a la acción de unos motores, provocando de esta manera el desplazamiento de la plataforma.

Para la construcción del sistema, se ha montado el mecanismo físico empleando, además de componentes comerciales, una serie de piezas fabricadas en 3D específicas para este sistema.

Junto a ese montaje físico, se ha necesitado realizar un software de control que, además de dirigir el movimiento del sistema, disponga de la capacidad de corregir posibles inconvenientes durante su funcionamiento.

## 1.3 Estructura de la memoria

La memoria se encuentra organizada de manera que en el Capítulo 2.1 Componentes se explicarán los componentes utilizados en el proyecto, tanto el equipo adquirido en el mercado, como las piezas diseñadas y fabricadas para el proyecto en cuestión.

Posteriormente, en el Capítulo 2.2 Metodología de montaje funcional, se detallará el proceso de montaje de cada uno de los subconjuntos del sistema, junto a una explicación de su funcionamiento.

En el Capítulo 3.1 Funcionamiento mecánico, se explicará el funcionamiento mecánico del sistema en todo su conjunto, haciendo uso de los subconjuntos explicados anteriormente. Se describirá, además del movimiento que realiza el sistema, varios métodos de corrección ante posibles inconvenientes durante su funcionamiento.

En el Capítulo 3.2 Funcionamiento lógico de programación se explicará la lógica de programación correspondiente al movimiento mecánico, así como el control de los métodos de corrección.

En el Capítulo 4 Conexiones eléctricas de los componentes, se mostrarán las conexiones eléctricas y electrónicas realizadas entre los diversos componentes del sistema, comentando datos técnicos sobre ellos.

En el Capítulo 5 Resultados se mostrarán los resultados obtenidos tras el diseño y montaje del sistema, en el Capítulo 6 Conclusiones se comentarán las conclusiones del proyecto y, finalmente, en el Capítulo 7 Trabajos futuros se enumerarán algunas ideas propuestas para trabajo futuro.



## 2 Diseño y desarrollo del sistema

El sistema basa su control en un movimiento horizontal y vertical mediante el uso de 3 motores en total. Respecto al primer movimiento, este se realiza mediante uno de los motores, desplazando el carro portador a izquierda o derecha. En cuanto al movimiento vertical, este se basa en un mecanismo de poleas, donde la plataforma queda pendida por 4 hilos que la sujetan por sus esquinas, y cuyos se tratan de uno de pesca de 0.4mm de grosor.

Mediante la rotación de cuatro tambores independientes, en donde cada uno de los hilos se encuentra enrollado, se logra elevar o descender la plataforma recogiendo o soltando hilo en función del sentido en el que giren. Además, al disponer únicamente de 2 motores para el movimiento vertical, se ha diseñado un mecanismo para que cada uno de ellos pueda hacer rotar 2 de los tambores al unísono.

Por otro lado, el análisis de imágenes para la detección de códigos QR, que se presenta posteriormente en el Capítulo 3.2 *Funcionamiento lógico de programación*, ha sido puesto a prueba de manera separada al mecanismo, por lo que no se encuentra incorporado de forma definitiva en el sistema final.

### 2.1 Componentes

A continuación, se enumerarán y describirán brevemente los componentes utilizados y las piezas diseñadas para el proyecto. En las Figura 4 y Figura 5 se pueden observar las trayectorias de explosión del orden de montaje de los mecanismos físicos diseñados, junto a la enumeración de los diversos elementos que los componen. En las Figura 6 y Figura 7 se pueden visualizar los montajes definitivos de estos mecanismos.

Los dos montajes mostrados en las Figura 6 y Figura 7 corresponden a la transmisión primaria y secundaria, respectivamente. Es decir, la transmisión que contiene el motor corresponde a la primaria, mientras que la que no contiene el motor corresponde a la secundaria.

El mecanismo completo consta de 2 transmisiones primarias y 2 transmisiones secundarias idénticas y alineadas entre sí. Respecto las transmisiones primarias, su orientación se ha realizado en modo espejo, es decir, las 2 transmisiones están orientadas en sentido contrario, quedando los 2 motores próximos entre sí. De forma análoga se han montado las transmisiones secundarias, quedando los soportes auxiliares próximos entre sí.

Las correas GT2 cerradas conectan cada par de transmisión primaria–secundaria para lograr rotar 2 tambores al mismo tiempo. Para ello, los acoples dentados de ambas transmisiones quedan alineados a la misma altura. A cada una de estas parejas de transmisiones primaria-secundaria las denominaremos bloques de transmisión (véase Figura 3).

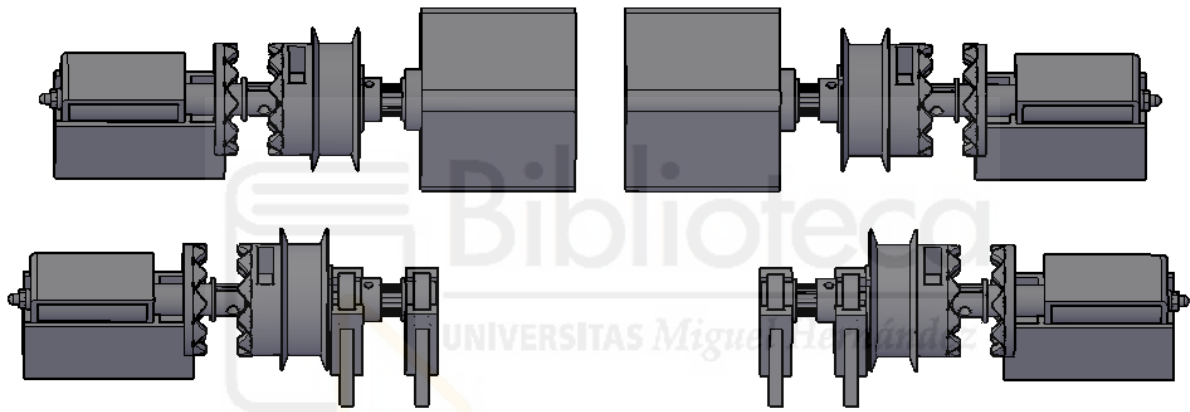


Figura 3. Plano general del mecanismo de transmisión completo.

Para el diseño de las piezas en 3D se ha utilizado el software AutoCAD. Es mediante este software de donde se han realizado capturas de pantalla de algunas de las piezas en 3D mostradas en la memoria, ya que de esta manera es posible visualizar mejor sus detalles.

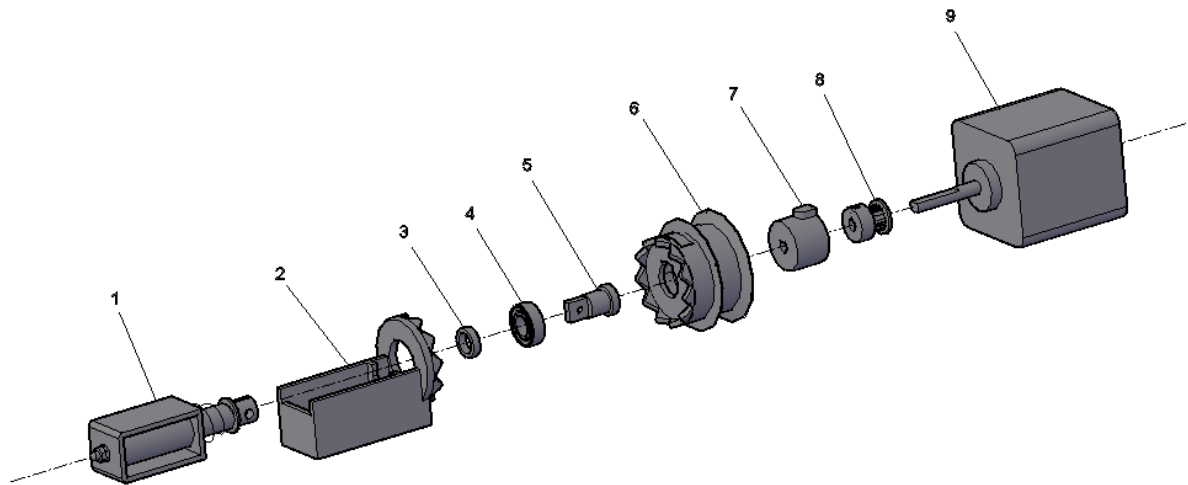


Figura 4. Trayectoria de explosión de la transmisión primaria.

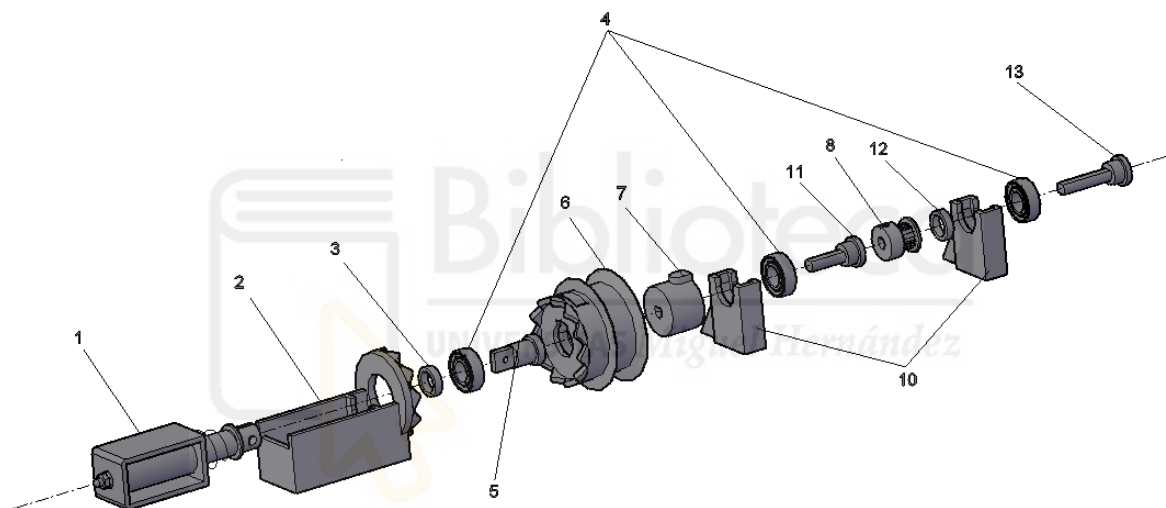


Figura 5. Trayectoria de explosión de la transmisión secundaria.

Elemento nº1: Actuador lineal

Elemento nº2: Soporte para el actuador lineal

Elemento nº3: Anilla de retención para el eje del tambor

Elemento nº4: Rodamientos de bolas rígidos

Elemento nº5: Eje del tambor

Elemento nº6: Tambor

Elemento nº7: Cilindro-chaveta

Elemento nº8: Acople dentado GT2

Elemento nº9: Motor paso a paso

Elemento nº10: Soporte auxiliar para rodamientos

Elemento n°11: Eje secundario hueco

Elemento n°12: Anilla de separación

Elemento n°13: Eje secundario

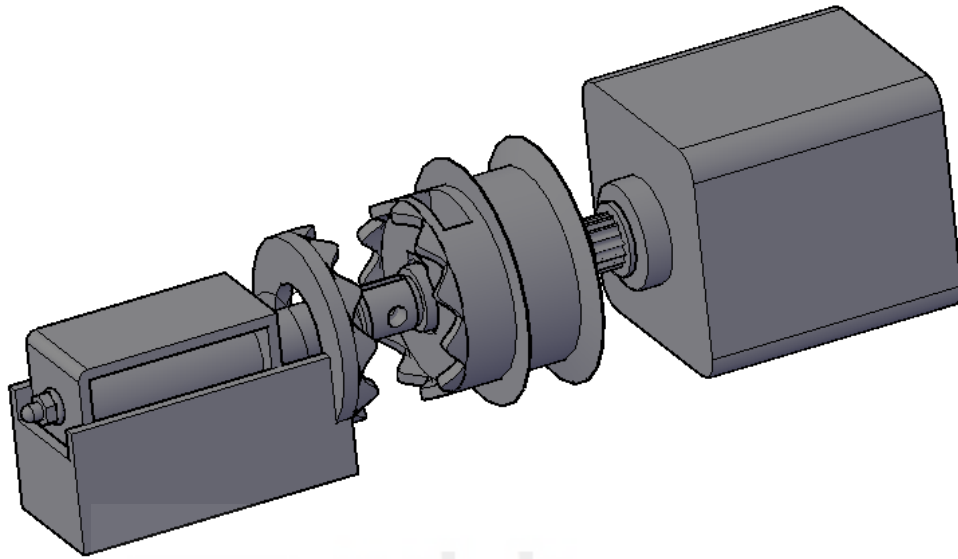


Figura 6. Montaje de la transmisión primaria.

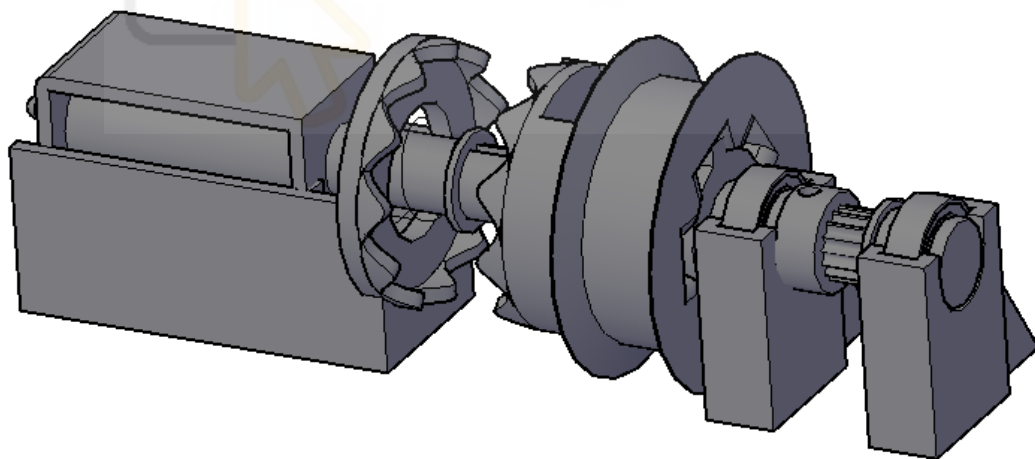


Figura 7. Montaje de la transmisión secundaria.

### 2.1.1 Raspberry Pi Modelo 4 B

Para el control del sistema se dispone de una Raspberry Pi 4 modelo B, pudiendo capturar vídeo en tiempo real y realizar cálculos sobre esos fotogramas.



Figura 8. Raspberry Pi Modelo 4B.

### 2.1.2 Fuente de alimentación

Se ha utilizado una fuente de alimentación de 240W con una salida a 24 voltios. Alimenta tanto a los controladores de los motores como a los actuadores lineales.

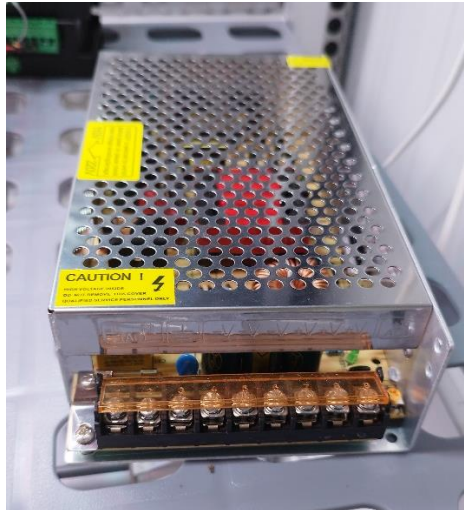


Figura 9. Fuente de alimentación.

### 2.1.3 Motores paso a paso

Se han utilizado 3 motores paso a paso del modelo NEMA 17 para desplazar de forma precisa la cámara durante su recorrido.

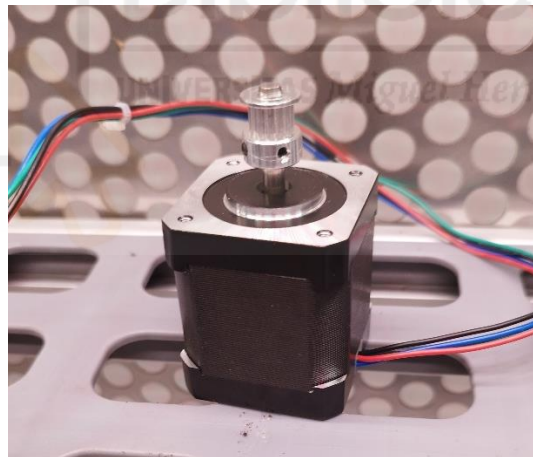


Figura 10. Motor paso a paso Nema 17.

### 2.1.4 Controladores de los motores paso a paso

Se han adquirido 3 controladores para los motores paso a paso del modelo TB6600.

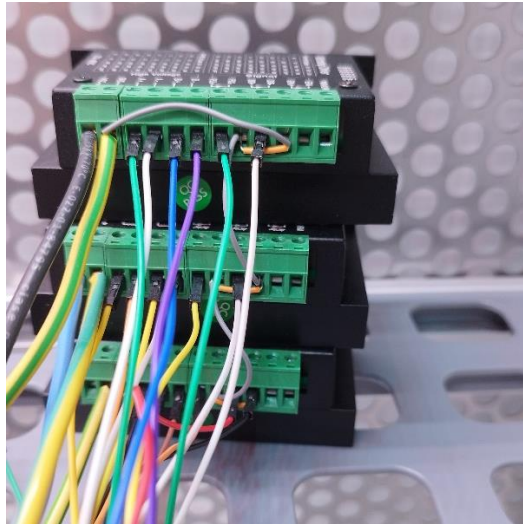


Figura 11. Controladores de motores paso a paso TB6600.

### 2.1.5 Cámara USB

Para la toma de imágenes se utilizará una cámara USB.

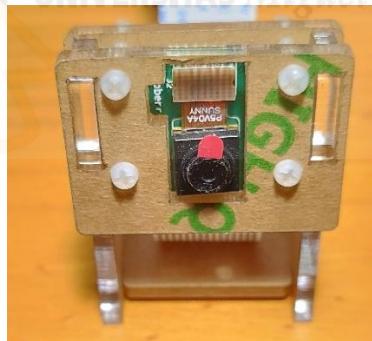


Figura 12. Cámara USB.

### 2.1.6 Plataforma

Consta de dos varillas metálicas de sección cuadrada y huecas, a las cuales se les ha hecho orificios paralelos por donde se han insertado dos varillas roscadas. Estas últimas se han fijado a las dos varillas cuadradas mediante tuercas de freno, formando así una estructura rígida.

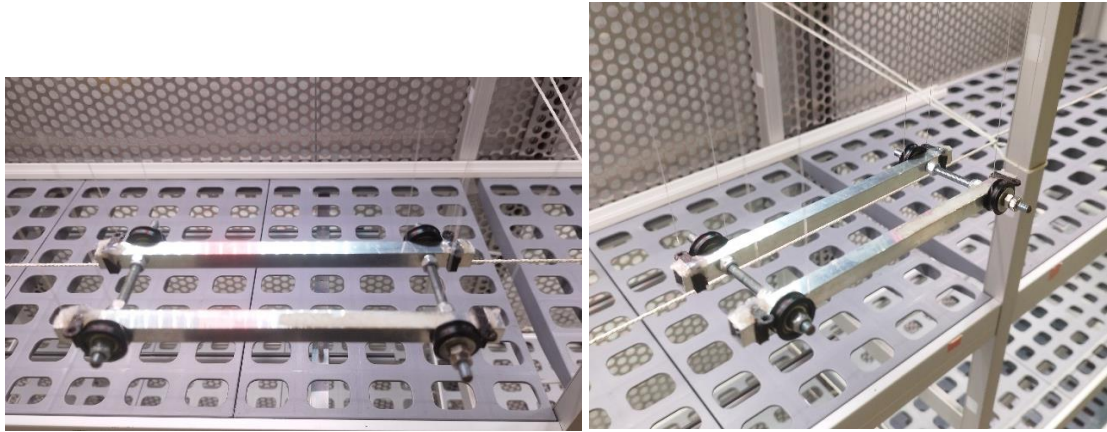


Figura 13. Vistas de la plataforma.

### 2.1.7 Carro portador

Es la estructura base que porta a la plataforma, soportando su peso y desplazándola horizontalmente. Está conformado por dos escuadras metálicas que, gracias a la multitud de agujeros que contienen, se han podido unir ambas mediante tornillos, tuercas de freno y dos tramos de varillas metálicas, idénticas a las utilizadas en la plataforma.



Figura 14. Vista frontal del carro portador.

### 2.1.8 Poleas

En el sistema se encuentran repartidas un total de 8 poleas. Cuatro de ellas están ubicadas en unas varillas roscadas, orientadas hacia la misma dirección y ancladas a pequeñas escuadras metálicas atornilladas al carro portador.

Las 4 poleas restantes se encuentran en la plataforma, en el saliente libre de las varillas roscadas de cada de sus esquinas.

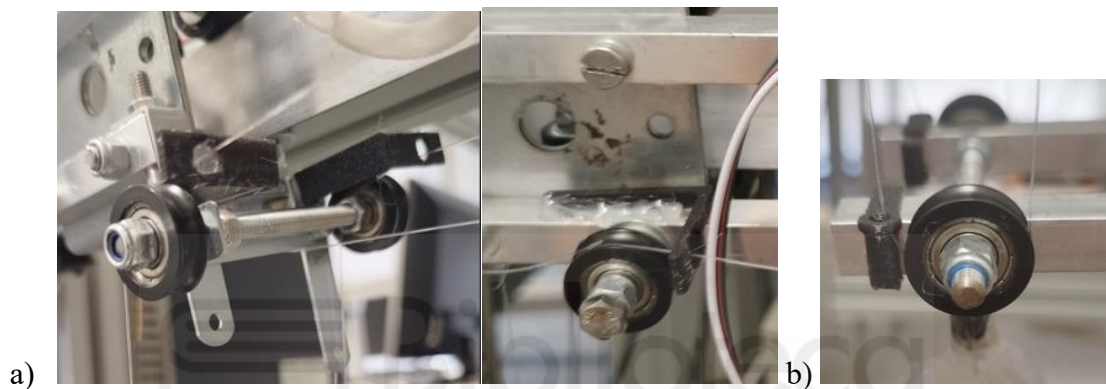


Figura 15. Ubicaciones de las poleas: a) En el carro portador, b) En la plataforma.

### 2.1.9 Rodamientos

Se han escogido rodamientos con un diámetro interior de 9mm y uno exterior de 17mm, con un ancho de 5mm.



Figura 16. Vista de un rodamiento.

### 2.1.10 Acoples dentados GT2

Estos acoples dentados, con 16 dientes y un ancho de 6mm por diente, están acoplados a los ejes de los motores de 5mm de diámetro. Su función es engranar con las correas dentadas GT2 de 6mm de ancho.



Figura 17. Vistas del acople dentado GT2.

### 2.1.11 Correas dentadas GT2

Se dispone de dos tipos de correas dentadas: una correa abierta extensa y dos cerradas cortas.

La primera se utiliza para el movimiento horizontal. Está fijada por ambos extremos al carro portador, completando así una cadena cerrada. Engrana con el acople GT2 del motor horizontal y con otra polea dentada auxiliar, ubicada en un soporte fijo en el extremo opuesto al motor. Las 2 correas cerradas restantes se utilizan en las transmisiones de los tambores.

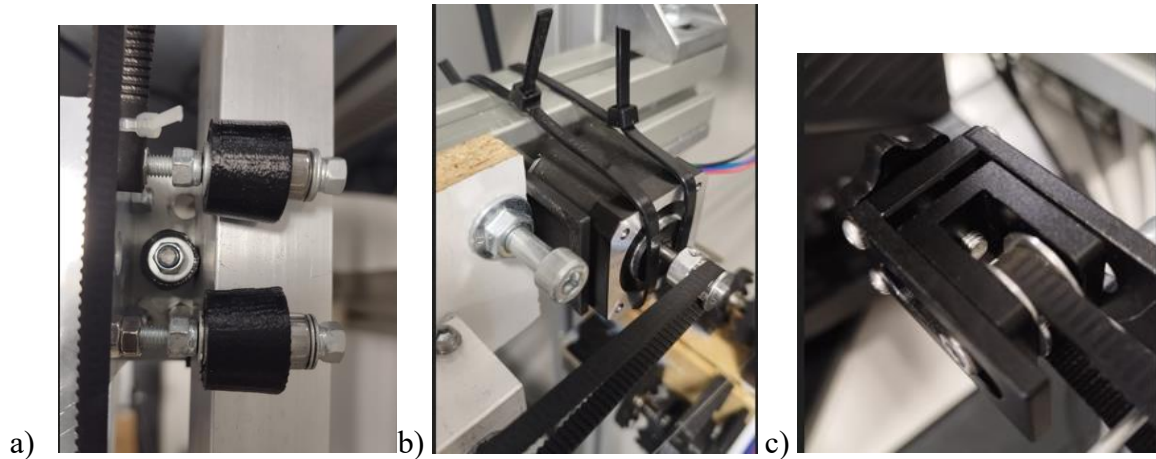


Figura 18. Correa GT2 abierta unida a los distintos elementos: a) Al carro portador, b) Engranada con el acople GT2 del motor, c) Engranada con la polea dentada del soporte auxiliar.

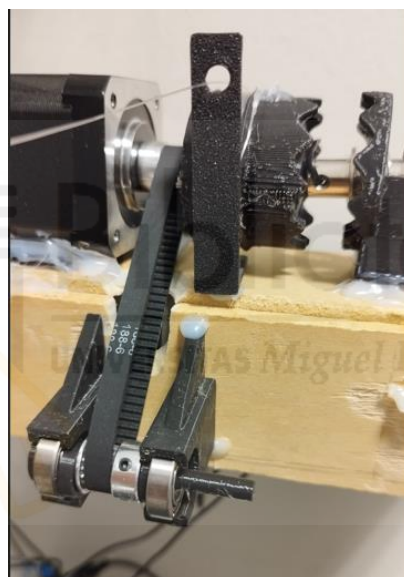


Figura 19. Correa dentada GT2 cerrada engranando los acoples dentados de las transmisiones.

### 2.1.12 Actuadores lineales

Estos actuadores corresponden al modelo 1039B, con una fuerza de atracción de 1.5-35N y una carrera del eje de 12mm.

La elección de este modelo de actuador se debe a que se necesitaba un muelle de retorno incorporado con gran fuerza, más que por la fuerza electromagnética del actuador en sí. Además, este actuador dispone de una gran disipación térmica, lo que permite alimentarlo durante bastante tiempo sin sobrecalentarlo en exceso.



Figura 20. Actuador lineal 1039B, 1.5-35N, 12mm, 24V.

Specifications	Voltage	current	trip	suction	Power on time
EML-1039B	12	600	12mm	initial 1.5N	≤30second
	24	300		keep 35N	

Figura 21. Características del actuador lineal 1039B.

### 2.1.13 Diodos rectificadores 1n4007

Se han utilizado diodos rectificadores de modelo 1n4007 para proteger los componentes electrónicos ante los picos de voltaje producidos por los actuadores lineales.



Figura 22. Diodo rectificador 1n4007.

### 2.1.14 Regulador Buck DC-DC LM2596

Se ha utilizado un regulador Buck DC-DC para reducir el voltaje de la fuente de alimentación de 24V a 18V. El regulador permite una entrada de 3 a 40V y una salida de 1.5 a 35V.

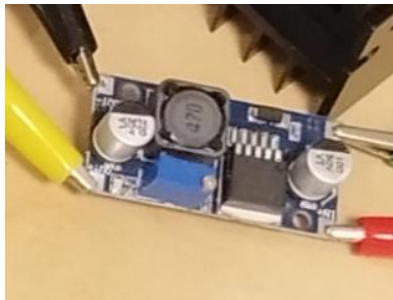


Figura 23. Regulador Buck LM2596.

### 2.1.15 Módulo de relés

Este módulo conformado por 4 relés está conectado a cada uno de los actuadores lineales. Lleva optoacopladores incorporados, permitiendo así aislar el circuito de potencia del de control.

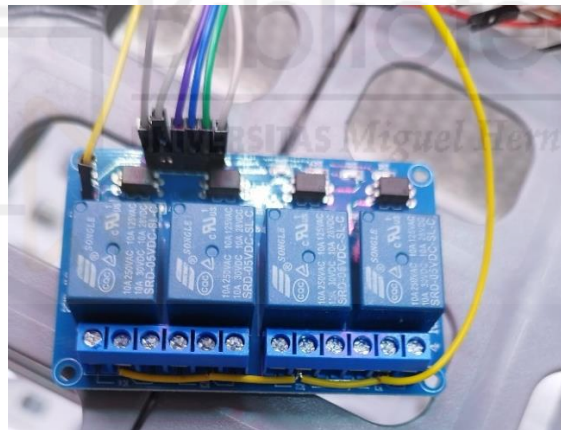


Figura 24. Módulo de 4 relés con optoacopladores.

### 2.1.16 Sensores Hall e imanes

En el sistema final se utilizarían sensores Hall para detectar las posiciones límite del sistema. El modelo de los sensores Hall es el OH137, el cual en presencia de un campo magnético se activa en un determinado umbral, pero se desactiva al cruzar un umbral menor al umbral de activación. Es decir, una vez el sensor es activado, para desactivarlo es necesario separar el imán más allá del punto en el que fue activado previamente.



Figura 25. Sensor Hall OH137.

### 2.1.17 Acelerómetro

Para leer la inclinación de la plataforma mediante la aceleración gravitacional en los 3 ejes cartesianos, se utilizaría un acelerómetro del modelo 'adx1345'.



Figura 26. Acelerómetro adxl345.

### 2.1.18 Impresora 3D

La impresora 3D ha sido un elemento imprescindible en el proyecto, permitiendo fabricar las piezas específicas requeridas. Las piezas enumeradas a continuación corresponden a las impresas en 3D.

### 2.1.19 Tambor

En él es donde se enrolla cada uno de los hilos. Dispone de aletas laterales cuya función es la de retener al cable dentro del carril de enrollamiento para evitar que descarrile.

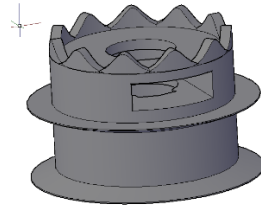


Figura 27. Vista general del tambor.

### 2.1.20 Cilindro-chaveta

Consta de un cilindro con una chaveta fija incorporada, cuya longitud es aproximadamente la mitad de la longitud del cilindro. El cilindro dispone de un agujero pasante con la misma sección que la de los ejes de los motores.



Figura 28. Vista isométrica del cilindro-chaveta.

### 2.1.21 Soportes de los actuadores lineales

La función de los soportes de los actuadores es enclavijar a estos últimos, manteniéndolos axialmente fijos a los ejes de los motores. La función del engrane de dientes se explicará en el Capítulo 3.1.4 Control de inclinación.

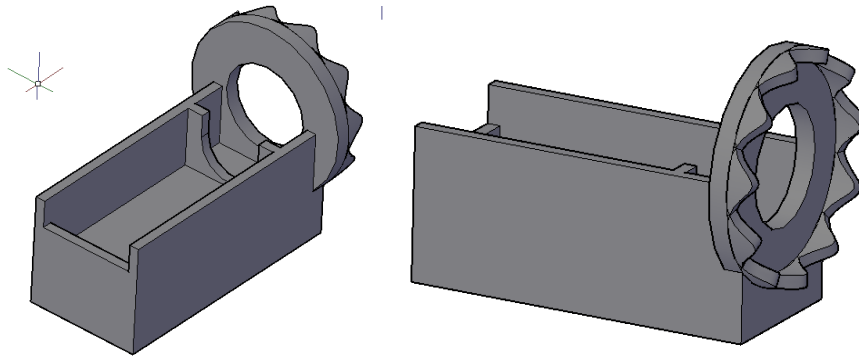


Figura 29. Vistas del soporte del actuador lineal.

### 2.1.22 Ejes de los tambores

Estos ejes se encuentran en cada uno de los rodamientos ubicados dentro del tambor. Se acoplan a los actuadores para realizar la función de pivote lateral del tambor.

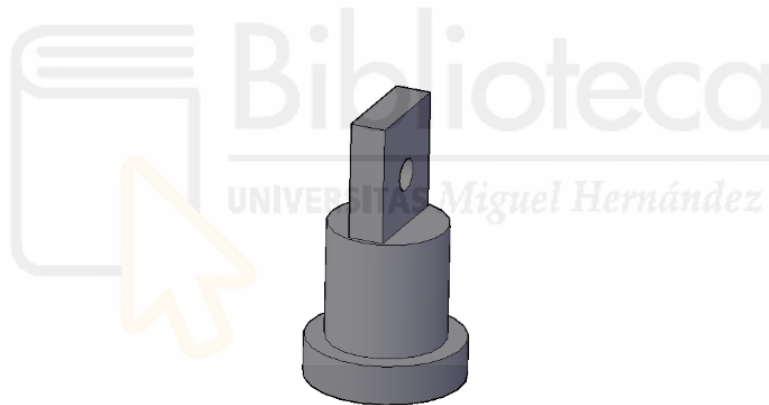


Figura 30. Vista isométrica del eje del tambor.

### 2.1.23 Anillas de retención

La función de estas anillas es mantener fijos a los ejes de los tambores.

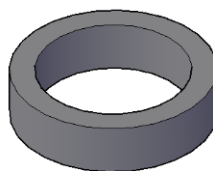


Figura 31. Vista isométrica de la anilla de retención.

### 2.1.24 Soportes auxiliares para rodamientos

En estos soportes se colocan rodamientos. La altura a la que se encuentra el centro del rodamiento coincide con la altura del eje de los motores.

Las medidas de la cavidad hueca son ligeramente superiores a las medidas del rodamiento, por lo que este último encaja a presión sin necesidad de adhesivos.

El saliente triangular tiene la función de aumentar el área de contacto para evitar el volcado del soporte debido a la fuerza de tensión en la transmisión.

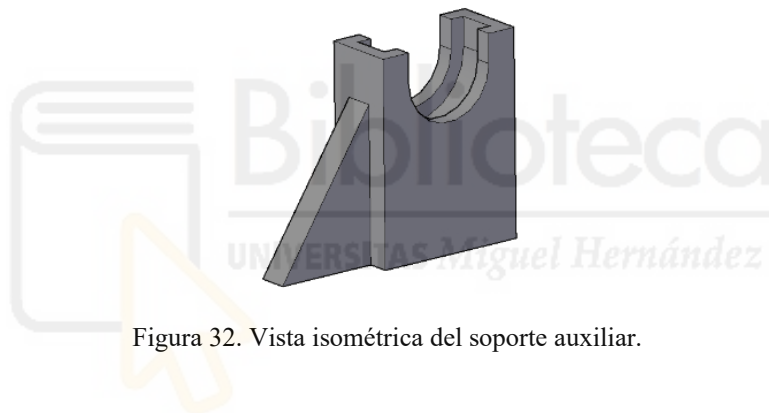


Figura 32. Vista isométrica del soporte auxiliar.

### 2.1.25 Ejes secundarios

Estos ejes están colocados en los rodamientos de los soportes auxiliares, y ejercen la misma función que el eje de los motores, es decir, ayudar a transmitir el par de rotación del cilindro-chaveta al tambor.

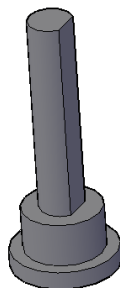


Figura 33. Vista isométrica del eje secundario.

### 2.1.26 Guías para el hilo

Contienen agujeros toroidales por donde pasan los hilos para reducir el rozamiento entre estos y la propia guía, evitando que puedan bloquearse en algún punto.

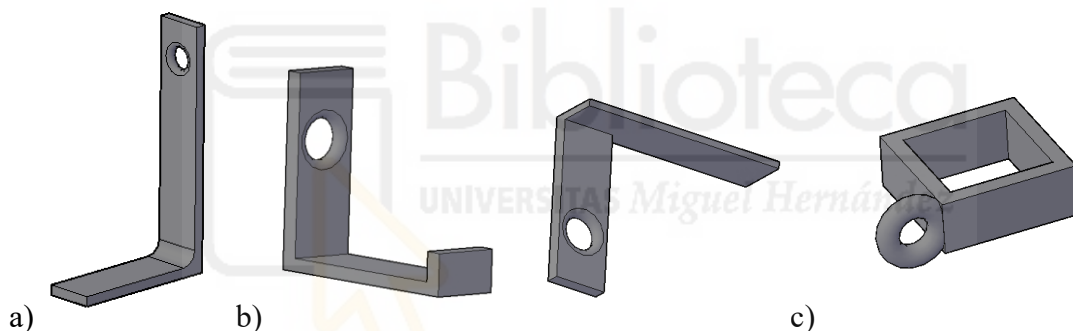


Figura 34. Guías ubicadas en: a) Frente a los tambores, b) El carro portador, c) La plataforma.

### 2.1.27 Ruedas del carro portador

Estas ruedas están fabricadas con filamento TPU, cuya flexibilidad permite que las ruedas se deformen ligeramente, adaptándose a la superficie de contacto.

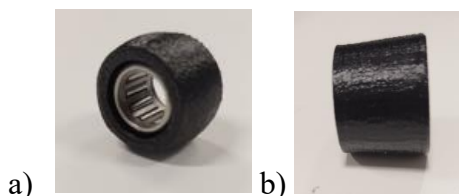


Figura 35. Ruedas del carro portador: a) Bombeadas, b) Cónicas.

## 2.2 Metodología de montaje funcional

En esta sección, explicaremos el montaje de los subconjuntos del mecanismo de los motores.

### 2.2.1 Montaje del mecanismo de transmisión primario

Para comenzar, explicaremos cómo están contruidos los tambores.

En los tambores, cerca de uno de sus laterales, se encuentra una cavidad en forma de 'D' radialmente al eje de giro (véase Figura 36) en la que se ha introducido un rodamiento. La cavidad posee unas dimensiones ligeramente mayores a las de este último, por lo que puede introducirse a presión y quedar bloqueado. Además, el rodamiento encaja perfectamente en el extremo final de la cavidad debido a que esta posee una curvatura de su mismo diámetro, por lo que la pista exterior descansa sobre esa pared curvada.

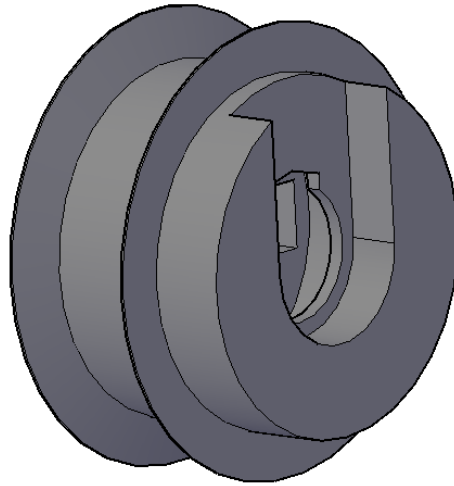


Figura 36. Ranura en forma de 'D' en el tambor.

Por otro lado, el tambor dispone de un agujero pasante por su eje de giro, cuyo diámetro es ligeramente inferior al diámetro exterior del rodamiento, por lo que este último no puede salirse por el agujero (véase Figura 37).

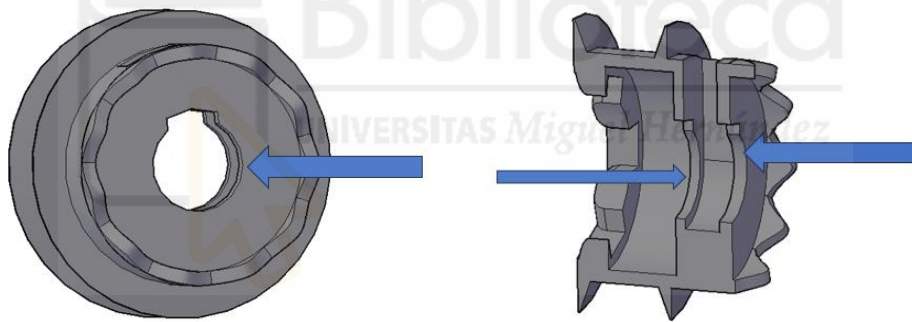


Figura 37. Agujero pasante del tambor.

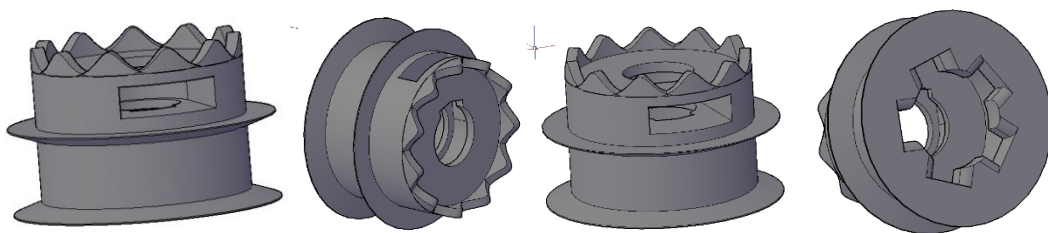


Figura 38. Distintas vistas del tambor.

A continuación, el eje del tambor se ha insertado de manera que el extremo circular queda dentro del tambor, mientras que el extremo rectangular sobresale de él. Por ese extremo saliente

se ha añadido una anilla de retención, deslizándola hasta contactar contra el rodamiento (véase Figura 39).

Debido a que esta anilla posee un diámetro interior ligeramente superior al del eje, y de la misma manera que en el caso del rodamiento, la fricción que se genera entre ambas piezas es suficiente como para prescindir del uso de pegamentos. Además, el diámetro exterior de la anilla es algo mayor que el diámetro interior del rodamiento, por lo que la pista interior de este último queda en contacto con ella.

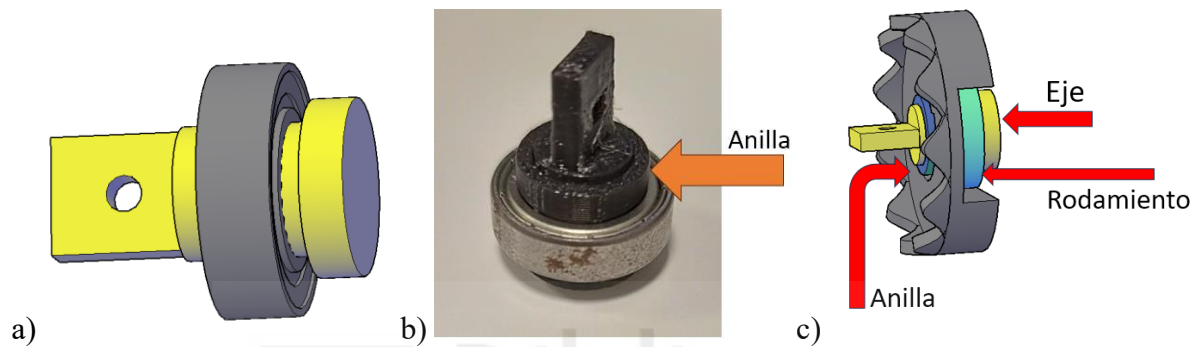


Figura 39. a) Inserción del eje del tambor en el rodamiento, b) Inserción del eje del tambor en el rodamiento junto a la anilla de retención, c) Montaje final de los elementos anteriores en el tambor.

La muesca que contiene el agujero pasante del tambor simplemente se debe a que, a causa de las pequeñas imperfecciones de las piezas, el extremo circular del eje rozaba con esa zona del tambor al rotar (véase Figura 40).

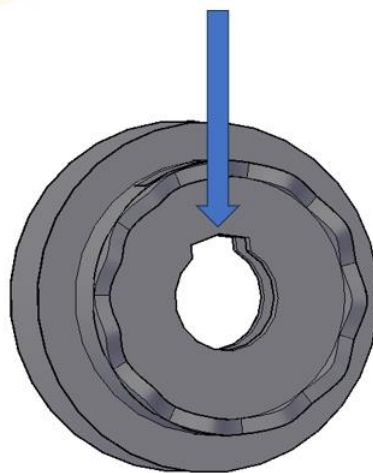


Figura 40. Muesca del agujero pasante del tambor.

A continuación, explicaremos como está montado el pivote lateral del tambor, formado por su eje y por el eje del actuador lineal.

El actuador lineal se ha colocado dentro de su soporte, y mediante un alambre que actúa como pasador, se ha unido el extremo de su eje con el extremo rectangular del eje del tambor (véase Figura 41b). Como resultado, este lateral del tambor queda sujeto por la unión de su eje y el eje del actuador, formando así un pivote que permite su rotación.

En pruebas anteriores se imprimieron pasadores en 3D, pero estos se soltaban al cabo del tiempo debido a los impactos del actuador. Además, si era necesario retirarlos resultaba complicado, por lo que un nudo con alambres permite retirarlos fácilmente en caso de ser necesario, además de continuar siendo funcional.

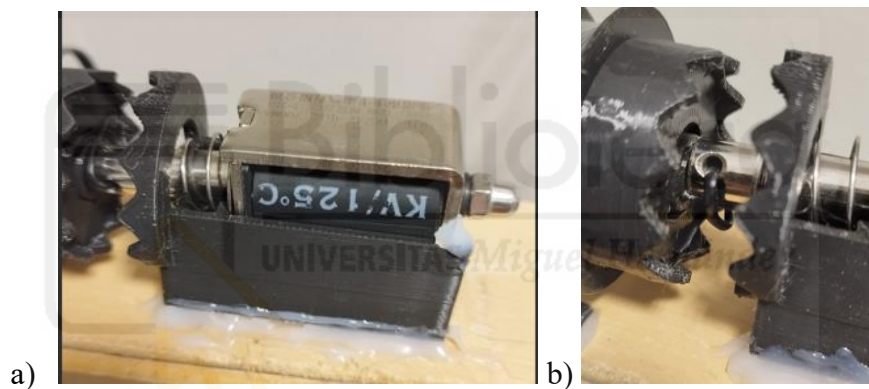


Figura 41. Vistas del actuador. a) En su soporte, b) Anclaje de los extremos mediante el alambre.

La unión de los dos ejes se consigue gracias a que la forma rectangular del extremo del eje del tambor acopla con la ranura del eje del actuador, encajando y quedando alineados ambos elementos (véase Figura 42).

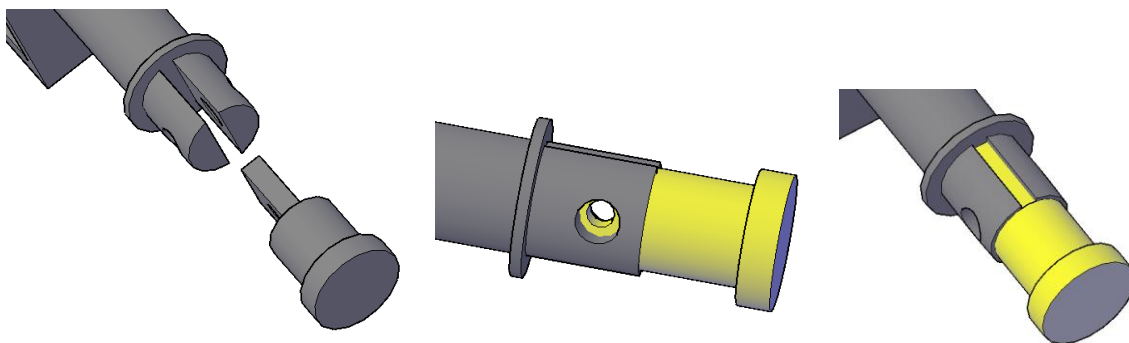


Figura 42. Vista de los extremos rectangulares de los ejes del actuador y del tambor.

A continuación, se explicará el montaje del pivote lateral opuesto del tambor, formado entre él y el cilindro-chaveta.

Cuando el tambor se encuentra en su posición natural, el cilindro-chaveta se encuentra en su interior ya que el tambor está hueco. Cuando el tambor es desplazado de su posición natural, la distancia desplazada (los 12mm que el actuador se desplaza al activarse) es mayor que la longitud de la chaveta, por lo que esta última queda fuera de las ranuras del eje nervado, ubicado en el otro lateral del tambor.

De esta manera, el tambor queda apoyado sobre el tramo libre circular del cilindro-chaveta, pudiendo el tambor girar libremente respecto a este último, ya que su diámetro es el mismo que el diámetro inscrito del eje nervado (véase Figura 43).

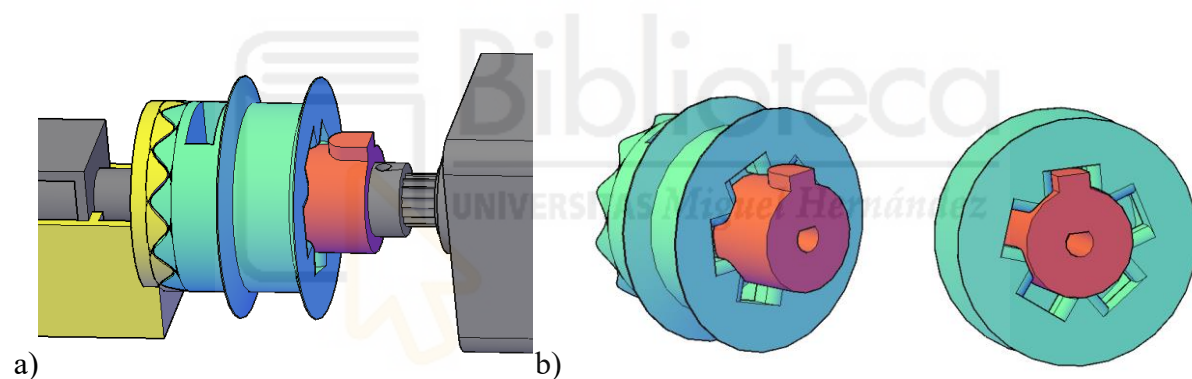


Figura 43. a) Apoyo del tambor sobre el cilindro, b) Distintas posiciones del cilindro-chaveta con respecto al tambor.

Finalmente, se explicará el funcionamiento de rotación de los tambores gracias al cilindro-chaveta.

En cada uno de los ejes de los motores se ha insertado un acople dentado y un cilindro-chaveta. Este último cuenta con un eje pasante que acopla en los ejes de los motores, permitiendo aplicar el torque de estos a la chaveta.

El lateral opuesto al de la cavidad del rodamiento en los tambores dispone de un eje nervado con varias ranuras axiales, en los que la chaveta puede acoplar. La función de estas ranuras es transferir el par de rotación del motor al tambor cuando la chaveta se encuentra en una de ellas.

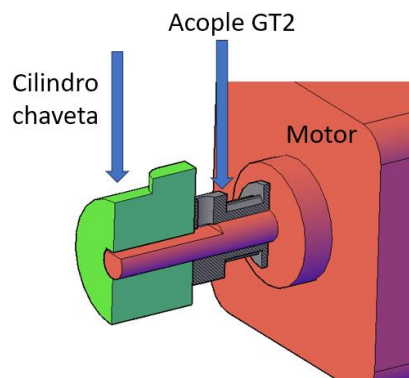


Figura 44. Elementos acoplados al eje del motor.

Antes de concluir esta sección, explicaremos la utilidad respecto al ensamblaje del rodamiento en el tambor, cuya aplicación se explicará en el Capítulo 3.1.4 Control de inclinación de la plataforma.

Cuando el actuador lineal no se encuentra alimentado, el muelle incorporado en él empuja al eje del tambor, el cual se ve forzado a desplazarse. Debido a que la anilla de fijación del eje del tambor se encuentra en contacto con la pista interior del rodamiento, se consigue que este último sea desplazado también. Como consecuencia, el tambor es desplazado ya que el rodamiento se encuentra inmovilizado en su interior.

En el caso opuesto, cuando el actuador se encuentra alimentado, este tracciona del eje del tambor gracias al alambre que actúa como pasador. En este movimiento, es el extremo circular del eje el que está en contacto con la pista interior del rodamiento, por lo que, de manera análoga al de la anilla, se consigue empujarlo en el otro sentido, desplazando también al tambor (véase Figura 39).

En resumen, este mecanismo funciona gracias a la propia resistencia del ensamblaje del rodamiento. La pista exterior está boqueada por la cavidad del tambor, mientras que la pista interior está en contacto por ambos laterales, tanto por el extremo circular del eje como por la anilla. Como resultado, al empujar o tirar de la pista interior del rodamiento, este es desplazado hacia un lado o hacia el otro.



a)



b)

Figura 45. Posiciones del actuador: a) Actuador no alimentado. Muelle extendido con el tambor en su posición natural, b) Actuador alimentado. Muelle comprimido con el tambor desplazado.

## 2.2.2 Montaje del mecanismo de transmisión secundario

En esta sección se explicará el montaje de la transmisión secundaria de los tambores, la cual es similar a la primera, pero sin los motores.

Antes de comenzar la explicación, cabe mencionar que los mecanismos de transmisión secundarios se encuentran en un plano perpendicular a los mecanismos de transmisión primarios (véase Figura 46). Estos últimos se han fijado a una tabla de madera con silicona caliente, la cual a su vez se encuentra anclada sobre otra tabla fija de madera, dispuesta horizontalmente y anclada a la estantería.

En cuanto a los mecanismos secundarios, estos se encuentran fijados también a una tabla de madera, sin embargo, esta se encuentra a 90 grados de la otra, apoyada sobre el lateral de la tabla fija.

Debido al gran peso de los motores, estos se han colocado en la tabla de arriba, mientras que el peso de la otra tabla y los mecanismos secundarios es soportado por las correas GT2 que unen ambos bloques de transmisión.



Figura 46. Vista general del mecanismo apoyado sobre la tabla fija.

En cuanto a la transmisión secundaria en sí, esta realiza la misma función que la transmisión primaria, pero de manera pasiva sustituyendo los motores por dos soportes auxiliares. En estos

últimos se encuentran rodamientos insertados a presión, de la misma manera que los de los tambores.

Comenzando desde el soporte exterior, en su rodamiento hay insertado un eje de su mismo diámetro, y que además dispone de una extensión similar al eje del motor (esto es, en forma de 'D'). Adicionalmente, posee un extremo circular con un diámetro mayor al del interior del rodamiento, por lo que de igual manera que en el caso del eje del tambor, este hace contacto con la pista interior.

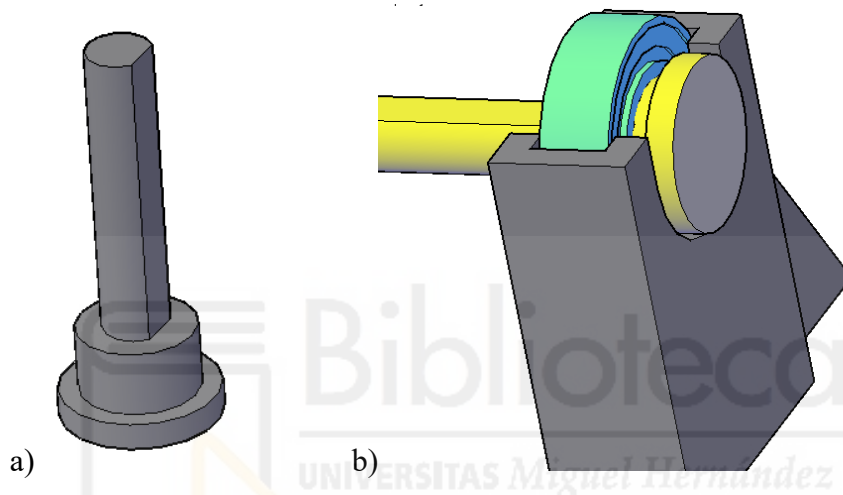


Figura 47. a) Eje secundario, b) Eje secundario insertado en el rodamiento del soporte auxiliar.

Seguidamente, se ha insertado un acople GT2 en el eje, y se ha colocado una anilla de separación (mostrada en color rojo en la Figura 49) entre este y el soporte para evitar que ambos elementos rocen entre sí.

Por otro lado, se ha diseñado una variación de ese eje secundario, el cual se encuentra en otro soporte auxiliar de igual manera. Este posee una cavidad en forma de 'D' idéntica al de su extremo, pudiendo así encajar ahí el otro eje.

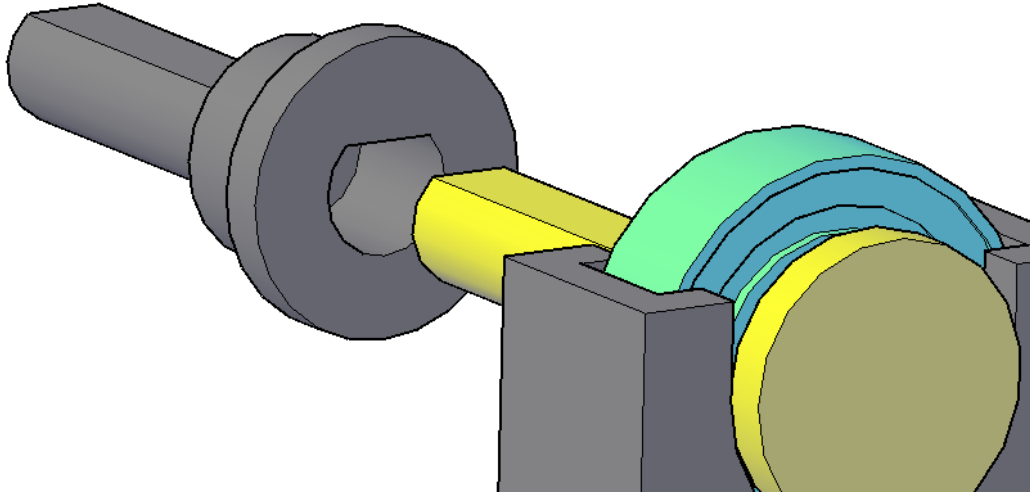


Figura 48. Acople de los ejes secundarios.

Por último, en el extremo del eje hueco se ha colocado un cilindro-chaveta, completando así la analogía del mecanismo primario, y permitiendo así poder transmitir la rotación desde el acople GT2 al cilindro-chaveta, y de este al tambor. El resto de los elementos, es decir, el tambor y su eje y el actuador con su soporte, se han colocado de igual forma que en la transmisión primaria.

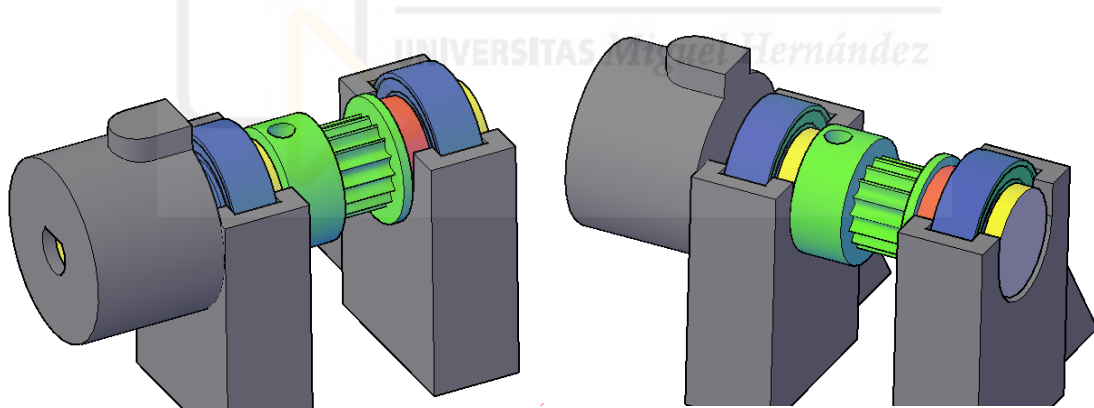


Figura 49. Vistas del ajuste de los elementos de la segunda transmisión montada.

Adicionalmente, para conseguir rotar tanto el tambor primario como el tambor secundario al unísono, se han insertado las correas GT2 cerradas entre los dos acoples dentados de ambas transmisiones, quedando así engranadas. De esta manera, cuando el motor rota, transmite el par a la correa, que a su vez transmite el par al acople dentado de la segunda transmisión, provocando el giro del tambor secundario (véase Figura 50).

La tensión que ejerce la correa sobre ambas transmisiones se consigue gracias a la gravedad. El peso de la segunda tabla de madera junto los componentes de la segunda transmisión fijados en ella, es suficiente para conseguir esa tensión. A pesar de ello, para aumentar la seguridad y fiabilidad del mecanismo, se atornillarán al menos un par de escuadras metálicas entre ambas tablas, manteniendo a su vez la tensión requerida en las correas.

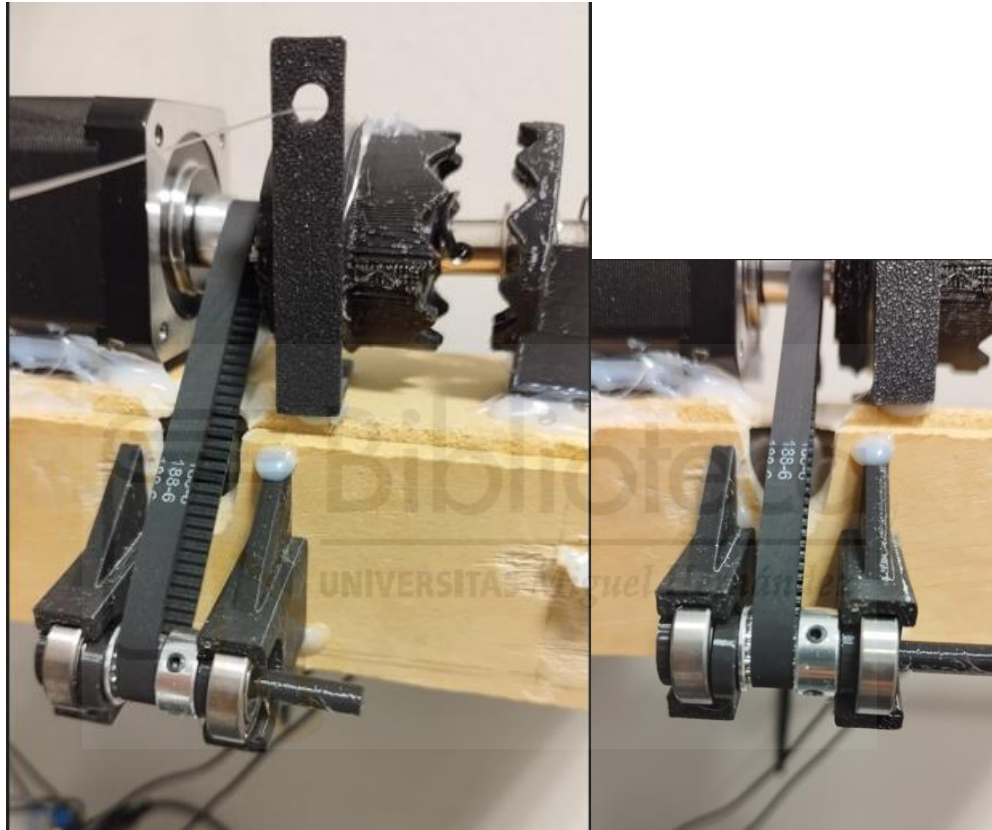


Figura 50. Bloque de transmisión conformado por la transmisión primaria y secundaria, acopladas mediante la correa GT2 cerrada.

### 2.2.3 Sistema de movimiento del carro portador

Respecto al movimiento horizontal, este se consigue gracias al desplazamiento del carro portador. Este último es libre de moverse a lo largo de una viga principal, la cual está sujeta a la estructura fija de la estantería mediante dos pequeñas escuadras metálicas, mostradas en la Figura 51. Esta viga posee una sección en 'L', orientada de forma que el ángulo recto que conforman las dos alas está dirigido hacia el observador, situado frente al sistema.



Figura 51. Perfil de la viga principal y su anclaje mediante las escuadras metálicas.

La estructura del carro portador se basa en dos grandes escuadras metálicas. Estas contienen diversos agujeros que se han utilizado para insertar varillas roscadas, en las cuales se han insertado las ruedas sobre las que el carro portador puede desplazarse.

En los agujeros de las alas horizontales se han fijado verticalmente 3 varillas roscadas en cada una, mientras que en las alas verticales se han fijado horizontalmente 2 varillas en cada una. En todas ellas se han insertado las ruedas, fabricadas con TPU y montadas sobre unos rodamientos, y se han fijado mediante arandelas y tuercas de freno para bloquearlas en su posición.

Las 3 ruedas verticales de cada escuadra se encuentran dispuestas triangularmente, de manera que 2 de ellas quedan en contacto con un lateral del ala vertical de la viga, mientras que la rueda restante en el otro lateral. Las 2 ruedas horizontales de cada escuadra soportan el peso

del propio carro y de la plataforma, y descansan sobre el ala horizontal de la viga (véase Figura 52).



Figura 52. Disposición de las diferentes ruedas del carro portador alrededor de la viga principal.

Se tuvo en cuenta que, el peso de la plataforma y los demás elementos, podrían forzar al carro a volcar debido a no estar el peso alineado completamente con el punto de apoyo vertical. Por ello, las ruedas horizontales se han diseñado con una forma cónica con el fin de adaptarse a esa inclinación, mientras que las verticales se han diseñado abombadas para que contacten en todo momento, sin holguras, con la viga.

La disposición de las ruedas permite que el carro portador solo pueda abandonar la viga por uno de sus extremos, limitando su movimiento exclusivamente a uno horizontal.



Figura 53. Disposición de las ruedas horizontales en el carro portador.

## 2.2.4 Trazado de los hilos

En esta sección se explicará el recorrido de los hilos, tanto por las poleas como por las guías ubicadas en distintos puntos.

El extremo libre de los hilos enrollados en los tambores pasa a través de una guía ubicada justo en frente de estos (véase Figura 54). Estas guías permiten que los hilos puedan enrollarse y desenrollarse radialmente al eje de giro de los tambores, evitando que descarrilen.

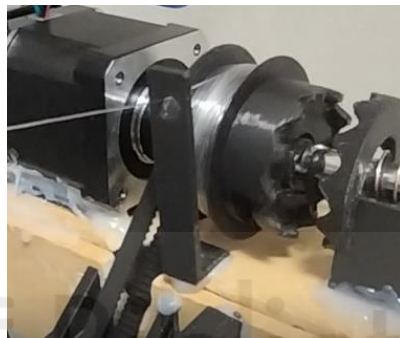


Figura 54. Guía para el hilo frente al tambor.

A continuación, los hilos pasan por las guías ubicadas en el carro portador, permitiendo que se alineen completamente con el sentido de giro de las poleas (véase Figura 55).



Figura 55. Vistas de las guías del carro portador ubicadas antes de las poleas.

Desde el carro portador, los hilos descienden hacia la plataforma, donde se encuentran unas guías en los extremos libres de las varillas cuadradas (véase Figura 56). Estas guías tienen el propósito de, además de alinear el hilo con la polea, detener la caída de la plataforma en caso de que hubiera algún inconveniente. De otro modo, si los hilos se salieran de las poleas de la plataforma, esta caería al suelo y los hilos podrían enrollarse sobre sí mismos, dificultando reestablecer el sistema al tener que volver a enrollarlos en su lugar.



Figura 56. Guías de la plataforma junto a las poleas.

Finalmente, el extremo final de cada hilo se ha fijado al carro portador, completando así el recorrido de los hilos.

El resultado es, en resumen, un sistema de doble polea como el de la Figura 57, donde la mitad del peso de la plataforma queda sostenido por el carro portador, mientras que la otra mitad es repartida entre los 4 hilos.

Por otro lado, debido a la propia rigidez del hilo de pesca, ha sido necesario colocar pesos en la plataforma para tensar los hilos, con la finalidad de que estos no se enreden y hagan buen contacto con las poleas. El propio peso de la plataforma es tan reducido que no es suficiente para conseguir tensarlos.

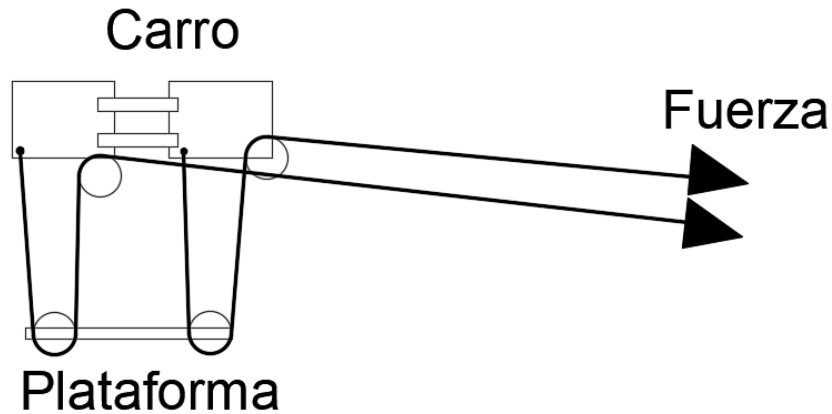


Figura 57. Esquema físico del mecanismo de doble polea.

El mecanismo de doble polea resulta interesante en este sistema, teniendo una ventaja extra. Debido a que los motores consumen corriente aun sin realizar ningún movimiento, ejerciendo así un par resistente para mantener los ejes en sus posiciones angulares, mientras estén alimentados pueden soportar cierto peso de la plataforma.

Sin embargo, en caso de que se produjera un corte de luz, los motores dejarían de estar alimentados, y por tanto dejarían de ejercer ese par resistente sobre los tambores. Gracias a que el peso de la plataforma se reparte entre el carro portador y los 4 hilos, y la tensión de estos 4 hilos entre los 2 motores verticales, el par equivalente en cada uno de estos últimos no es suficiente para hacerlos girar, dando como resultado que la plataforma permanezca sin caerse.

## 3 Funcionamiento general del sistema

### 3.1 Funcionamiento mecánico

En esta sección se explicará el funcionamiento general del sistema, describiendo el trazado y los movimientos de la plataforma que realizaría durante su recorrido, basándonos en los montajes descritos anteriormente.

#### 3.1.1 Recorrido físico de la plataforma

El proceso iniciaría desde el punto de partida, situado en la esquina superior derecha. Desde ahí, la plataforma comenzaría su recorrido describiendo un movimiento en zigzag, de derecha a izquierda y de arriba a abajo, deteniéndose frente a cada una de las placas para realizar sus fotografías (véase Figura 58).



Figura 58. Recorrido de la plataforma a través de las placas durante el funcionamiento.

Inmediatamente después de tomar la fotografía de la primera placa, ubicada al inicio del recorrido, la plataforma se desplazaría a la izquierda hasta situarse frente a la placa adyacente. En ese momento, la plataforma se detendría, tomaría la fotografía de esa nueva placa y se desplazaría hacia la siguiente. Cuando se posicionara frente a la nueva placa, se detendría de nuevo y tomaría la nueva fotografía. Este proceso continuaría hasta alcanzar la última placa de esta primera balda de la estantería.

Cuando la plataforma termine de recorrer toda la primera balda, descendería a la inferior y se posicionaría en esa última placa de la balda. A continuación, desde esa nueva posición, la plataforma realizaría el recorrido inverso al descrito hasta ahora, es decir, un desplazamiento a derechas deteniéndose frente a cada una de las placas y tomando su fotografía.

Finalmente, cuando alcance la primera placa de esta segunda balda y realice su fotografía, la plataforma descendería de nuevo hacia la balda inferior para repetir de nuevo el proceso. Este ciclo de desplazamientos en zigzag se repetiría hasta abarcar las 4 baldas de la estantería.

Por último, una vez realizada la fotografía de la última placa del recorrido, ubicada en la esquina inferior derecha del plano de movimiento, la plataforma retornaría a su posición de partida, manteniéndose a la espera del siguiente ciclo programado.

### 3.1.2 Retorno de posición de la plataforma

Respecto al retorno al punto inicial de la plataforma, se pensó en desplazar a esta última una distancia absoluta equivalente a la recorrida hasta el momento, tanto horizontal como verticalmente, pero en sentido opuesto.

Sin embargo, se cuestionó en la posibilidad de que se produjera un corte de luz, donde la plataforma quedara interrumpida a mitad del recorrido. La consecuencia sería la imposibilidad de determinar cuánta distancia se recorrió hasta ese momento, por lo que no habría opción a retornarla a su posición inicial de manera fiable.

La solución planteada ante ese posible inconveniente ha consistido en colocar unos sensores Hall, que se utilizarían para poder determinar los límites del espacio de movimiento de la plataforma. De esta manera, independientemente de la posición en la que se encuentre la plataforma, en el proceso de retorno esta procedería a desplazarse vertical y horizontalmente hasta detectar los sensores, manteniendo así la plataforma dentro de su rango de movimiento.

Uno de estos sensores se integraría en la plataforma, mientras que el otro se ubicaría cerca del motor horizontal. Sus posiciones corresponderían a las posiciones extremas vertical y horizontal del sistema, respectivamente. Con respecto a sus activaciones, se utilizarían un par de imanes que se ubicarían en el carro portador (véase Figura 59).

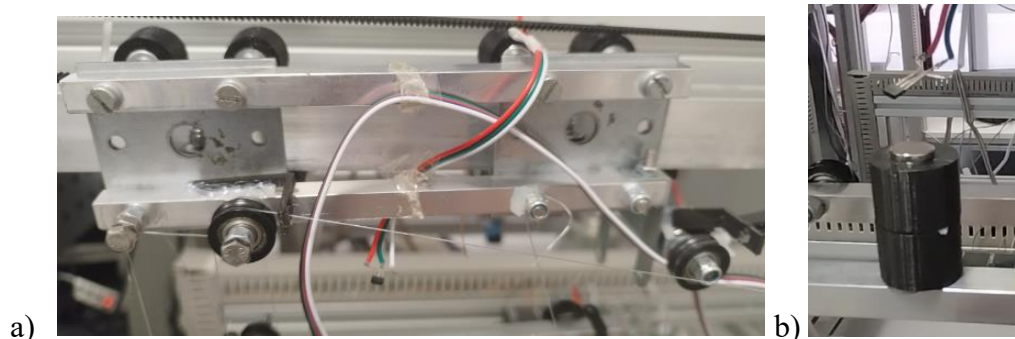


Figura 59. Vista de un montaje experimental con el sensor Hall vertical: a) Posición del sensor en el carro portador, b) Proximidad entre el imán y el sensor.

En resumen, el proceso de retorno consistiría en un ascenso continuo de la plataforma, hasta que el sensor Hall ubicado en ella detectara el campo magnético del imán del carro portador, alcanzando así el límite vertical. De manera análoga, se produciría un desplazamiento

horizontal a derechas, deteniéndose cuando el imán del carro portador interaccionara con el sensor Hall ubicado cerca del motor horizontal. Finalmente, una vez completado este último movimiento, la plataforma debería encontrarse en la posición inicial, ubicada en la esquina superior derecha de la estantería.



### 3.1.3 Movimiento horizontal del carro portador

El movimiento horizontal del carro portador se consigue gracias a la correa GT2 de extremos abiertos. Uno de estos extremos se encuentra unido a un punto del carro portador. A su vez, el otro extremo de la correa se ha pasado alrededor del acople dentado del motor horizontal, y luego por la polea dentada del soporte fijo, ubicado en el extremo opuesto al motor.

Finalmente, este extremo libre se ha atado en otro punto del carro portador, tensando la correa en el proceso. De esta manera se ha conseguido una cadena cerrada, donde la rotación del motor se transforma en un movimiento lineal de la correa (véase Figura 60).

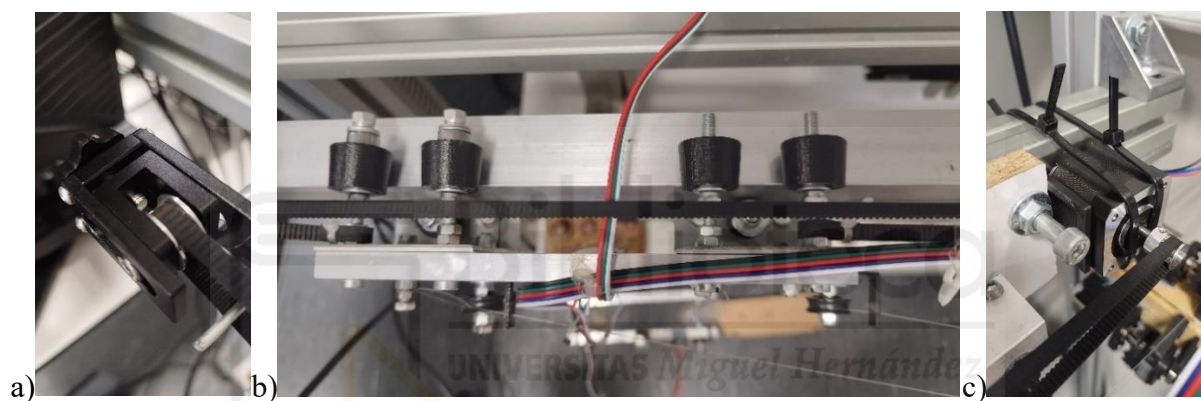


Figura 60. Transmisión del carro portador con la correa GT2: a) Polea dentada auxiliar ubicada en el extremo izquierdo, b) Anclaje de la correa GT2 al carro portador, c) Unión con el acople GT2 del motor horizontal.

Por otro lado, la cámara anclada a la plataforma debe moverse por todo el plano de la estantería, por lo que su cable de alimentación debe poder extenderse desde un extremo al otro del recorrido. Para ello, en las pruebas se utilizó un alargador USB sujeto de un punto en un lateral del carro portador, formando así una catenaria para evitar que arrastrase por el suelo (véase Figura 61).

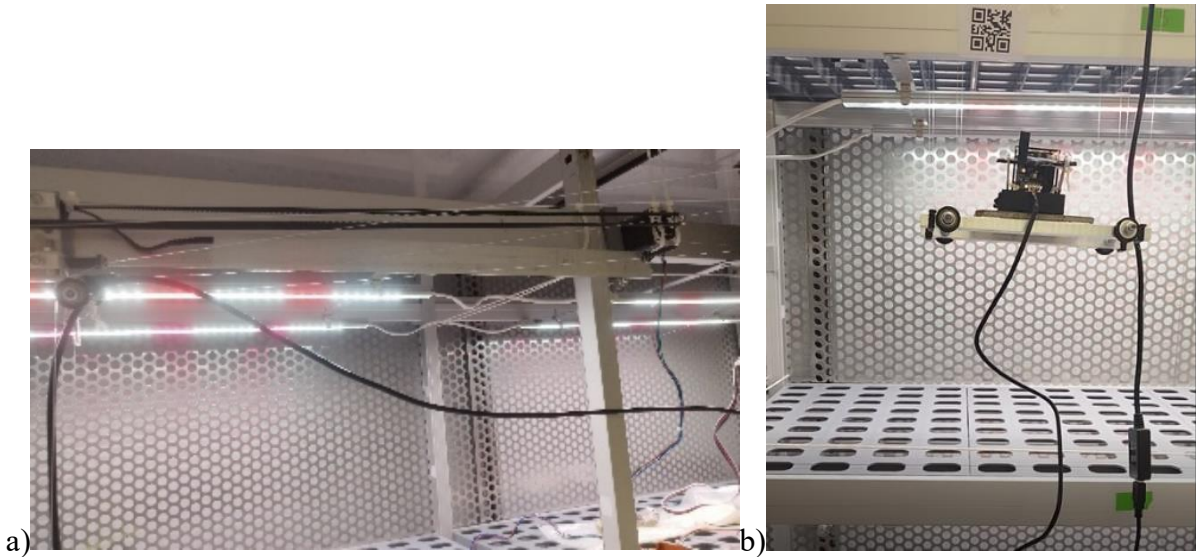


Figura 61. Vistas del cable USB colgando del sistema: a) Catenaria horizontal con anclaje al carro portador, b) Cable colgando según la altura de la plataforma.

Inicialmente, se consideró introducir un mecanismo similar al de los puentes grúa, donde abrazaderas sujetan en distintos puntos el cable, de manera que este último cuelga como una catenaria en diversos tramos del recorrido (véase Figura 62). Sin embargo, debido a la propia rigidez de los cables y a sus reducidas masas, estos no eran capaces de flexionarse adecuadamente para formar las catenarias. Esto afectaba negativamente a la estabilidad del sistema, por lo que se descartó.

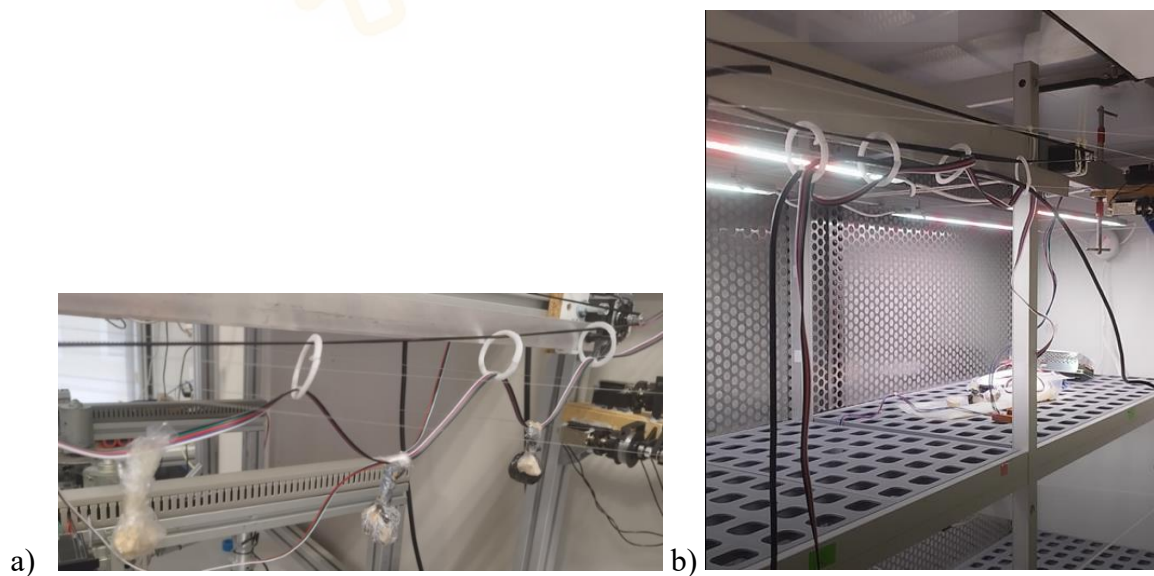


Figura 62. Sistema experimental de cables colgantes similar a los puentes grúa: a) Catenarias ideales, b) Flexión de los cables por su propia rigidez.

### 3.1.4 Control de inclinación de la plataforma

Durante el funcionamiento del sistema, se ha planteado la posibilidad de que se produzcan varios inconvenientes. Estos podrían ser, por ejemplo, un desfase en la sincronización de los motores, un error de tensado en las correas GT2 de las transmisiones, o incluso una variación del diámetro útil en alguno de los tambores, debido a la superposición del propio hilo cuando este se enrolla. Estas causas podrían provocar una inclinación indeseada en la plataforma, por lo que sería necesario realizar correcciones para nivelarla y asegurar un correcto funcionamiento.

Para poder realizar estas correcciones, se haría uso de los actuadores lineales y del mecanismo de transmisión. Como se comentó previamente en el Capítulo 2.2.1 Montaje del mecanismo de transmisión primario, cuando un actuador lineal se activa, desplaza al tambor de su posición natural, quedando la chaveta por fuera de las ranuras y permitiendo que el tambor pivote sobre el cilindro.

Para conseguir que el tambor no gire en ese instante, el lateral con dientes curvos que contiene contacta con el engrane del soporte del actuador. Ambos engranes tienen el mismo número y tipo de dientes, excepto que los del soporte del actuador son ligeramente más anchos para asegurar que los del tambor contacten completamente con ellos (véase Figura 63).

Cuando ambos engranes acoplan entre sí, tanto el intercalado de los dientes como la fuerza que ejerce el actuador traccionando sobre el tambor evitan, gracias a la fricción que se genera, que este se desenrolle indeseadamente.

Una vez realizado el proceso de nivelación de la plataforma, el tambor debe volver a su posición, por lo que el actuador deja de ser alimentado y el muelle incorporado en él empuja al tambor, retornándolo a su lugar. Es aquí donde se hace uso del diseño redondeado de los dientes, permitiendo que deslicen fácilmente entre ellos y favoreciendo ese retorno del tambor.

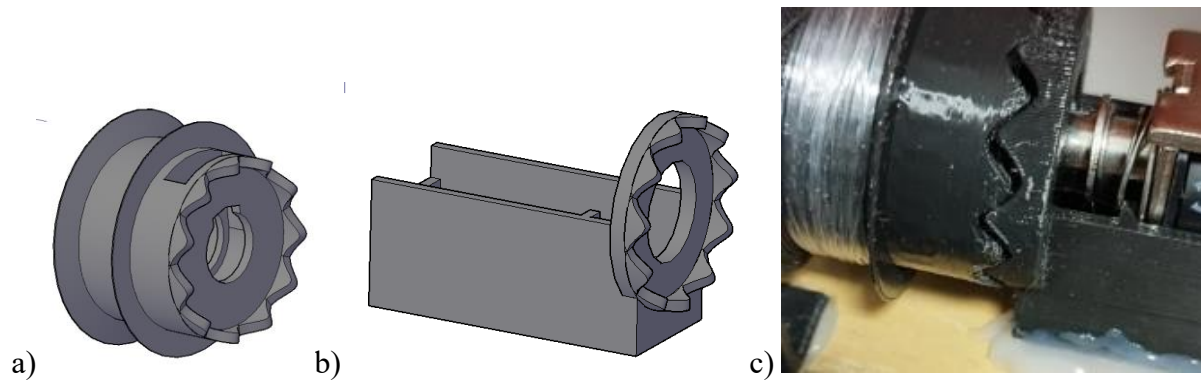


Figura 63. a) Engrane del tambor, b) Engrane del soporte del actuador, c) Acople entre ambos engranes.

A continuación, durante ese movimiento de traslación del tambor, podría suceder que una de las ranuras del eje nervado del tambor coincidiera directamente con la chaveta, acoplándose ambos elementos directamente (véase Figura 64a).

Sin embargo, podría suceder que, debido a que el tambor es rotado un determinado ángulo durante el proceso de nivelación, la chaveta no coincida directamente con la ranura en el momento de acople. En este caso, la chaveta colisiona con la pared de ese lateral del tambor (véase Figura 64b). El procedimiento aquí consiste simplemente en continuar rotando el tambor mediante el funcionamiento normal del sistema, ya que se ha observado que la chaveta acabará introduciéndose al cabo del tiempo en una ranura adyacente, por lo que no hay inconveniente en que la chaveta quede por fuera.

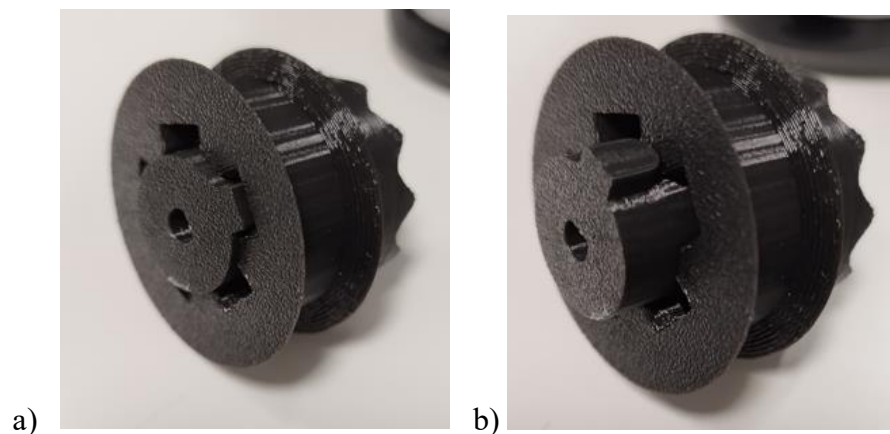


Figura 64. Posiciones de la chaveta respecto al tambor: a) Enclavijado de la chaveta en una ranura del eje estriado del tambor, b) Chaveta en contacto con la pared del tambor.

Para facilitar que la chaveta consiga entrar en la ranura, esta debe poder deslizarse fácilmente sobre la cara plana de ese lateral del tambor, mientras que el muelle del actuador ejerce presión sobre ambas piezas. Para ello, la chaveta dispone de un tamaño ligeramente inferior al de las ranuras, y además contiene un redondeo semicircular en el extremo de contacto, cuyo fin es el de reducir la fricción entre ambas piezas (véase Figura 65).

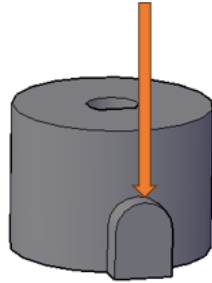


Figura 65. Redondeo de la chaveta.

Adicionalmente, cada una de las aristas de las ranuras poseen un redondeo para favorecer que la chaveta entre en ellas (Figura 66).

Estas ideas surgieron de las primeras pruebas experimentales, en las que se fueron limando las piezas poco a poco, hasta encontrar la forma y tamaño aproximados que debían tener para que deslizaran y encajaran correctamente.

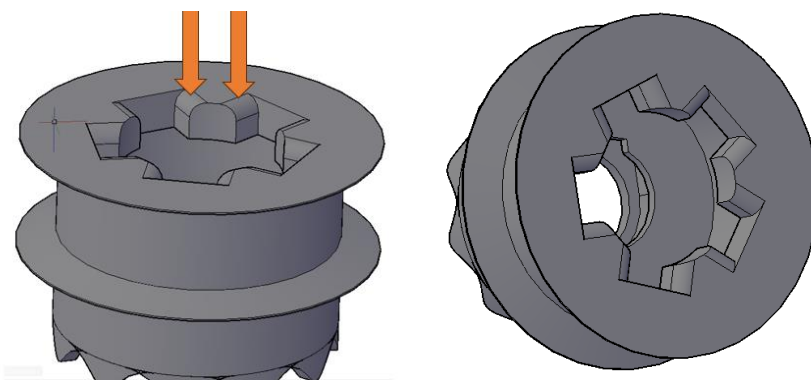


Figura 66. Vista desde diferentes ángulos de los redondeamientos de las aristas del eje nervado del tambor.

Sin embargo, debido a la circunstancia mencionada en que la chaveta colisiona con la pared del tambor, este mecanismo tiene unas limitaciones, que se dan debido a ese pequeño giro requerido para introducir la chaveta en la ranura. Esto supone que el tambor se desfase respecto

a la posición angular en la que fue empujado, causando que, aunque la plataforma se nivele perfectamente, esta vuelva a desnivelarse ligeramente debido a esa imperfección.

Este error por desfase depende del número de ranuras. En este caso, el eje nervado del tambor está conformado por 5, por lo que el desfase tiene un valor máximo de  $72^\circ$ . Esto se traduce en que, dependiendo del diámetro del tambor desfasado y de las dimensiones de la plataforma, la inclinación de esta sea mayor o menor.

Para reducir ese desfase angular, una solución podría consistir en aumentar el número de ranuras, diseñándolas con un menor tamaño o bien colocándolas más próximas entre sí. Sin embargo, debido a las limitaciones físicas, se ha priorizado la fiabilidad del mecanismo antes que una gran precisión, por lo que 5 ranuras se ha considerado un número razonable para el pequeño diámetro del tambor.

Por otro lado, como consecuencia de estas limitaciones, corregir una pequeña inclinación en la plataforma podría causar que esta se desajustara incluso más. Por ello, este proceso de calibración únicamente se ejecuta en casos excepcionales donde la plataforma se encuentre en ángulos críticos, dejando un margen de error en el cual el sistema pueda funcionar adecuadamente.

## 3.2 Funcionamiento lógico de programación

En cuanto al código del sistema, este se ha probado de manera separada al mecanismo físico, como se comentó anteriormente. Mediante la realización de las pruebas aisladas se ha observado un funcionamiento correcto, por lo que se prevé que la integración con el mecanismo en la cámara de cultivo pueda ser satisfactoria.

El código se ha realizado haciendo uso del lenguaje Python, gracias a su versatilidad y compatibilidades al usar la Raspberry Pi y OpenCV. Además, para mejorar la legibilidad, el código se ha organizado en 2 secciones: un script con el código principal y otro script con las funciones utilizadas.

Mediante las funciones que incorpora esta librería de OpenCV mencionada, es posible detectar y leer códigos QR, generados mediante la librería adicional qrcode, cuyo uso se muestra en la función de la Figura 67.

The image shows a snippet of Python code for generating QR codes. The code is displayed in a light-colored box with a faint watermark in the background that reads 'Biblioteca UNIVERSITAS Miguel Hernández'. The code includes an import statement for 'qrcode', a loop to generate QR codes for a range of numbers, and the creation and saving of each QR code as a black-on-white PNG image.

```
import qrcode

i=0
for _ in range(numero_de_placas):
    i = i + 1
    texto = str(i)

    qr = qrcode.QRCode(
        version=1,
        error_correction=qrcode.constants.ERROR_CORRECT_L,
        box_size=10,
        border=4,
    )

    qr.add_data(texto)
    qr.make(fit=True)
    img = qr.make_image(fill_color="black", back_color="white")

    img.save(f"QR{i}.png")
```

Figura 67. Función para crear códigos QR.

Con el fin de aprovechar las funcionalidades de OpenCV, bajo cada una de las placas de cultivo se colocarían unos pequeños códigos QR, los cuales se utilizarían tanto para la identificación de cada una de las placas como para la orientación del sistema durante el recorrido (véase Figura 68).

El contenido del QR, con el que se identificaría cada una de las placas, consistiría en un número correspondiente al número de placa. Es decir, el identificador de la primera placa sería un 1, el de la placa número 2 sería un 2, etc. Con esto se pretende que el QR generado contenga una baja cantidad de píxeles, por lo que su tamaño físico puede ser reducido, manteniendo una adecuada nitidez para poder detectarlo sin grandes dificultades.

Mediante la detección de los QR, se prevé que el sistema pueda orientarse correctamente, buscando el QR de la placa a fotografiar para asegurar que realmente se encuentre frente a ella. Además, con esto se podría conseguir un mayor control al realizar las fotografías, pudiendo centrar la cámara para capturar la placa en su totalidad.



Figura 68. Posible ubicación de los QR en la estantería.

### 3.2.1 La función de toma de fotografías

La fotografía que se realice se guardaría con un nombre de archivo correspondiente a la hora y fecha en la que esta fue tomada. Con ello, cada nombre sería distinto (véase Figura 69).

```
def take_photo(data, frame):  
    timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H-%M-%S')  
    num_placa = data  
    filename = f"{num_placa}-{timestamp}.jpg"  
    cv2.imwrite(filename, frame)
```

Figura 69. Código de la función de tomar fotografía.

### 3.2.2 Movimiento vertical del sistema

En cuanto a la rotación de los motores, es necesario enviar trenes de pulsos a los controladores, cuya frecuencia determina sus velocidades. En este caso, la frecuencia se ha establecido a 1000Hz.

Respecto a los controladores TB6600, estos disponen de varios interruptores mediante los cuales se pueden configurar distintas combinaciones de micro pasos. Mediante una relación 1/1 sería necesario enviar 200 pulsos para completar una revolución completa en los motores. Sin embargo, con esa relación, los motores generarían grandes vibraciones que podrían causar que alguna pieza se desprendiera de su posición. Además, desplazaría la plataforma a gran velocidad, produciéndose un balanceo al detenerse que dificultaría la toma de imágenes.

Por ese motivo, se utilizaría una relación de 1/16 en los 3 controladores, lo que significa que se deberían enviar 3200 pulsos para completar una revolución completa en un motor. Esto permitiría un control más preciso sin comprometer a la estabilidad.

A continuación, en la Figura 70 se muestran las funciones que determinarían la rotación de los motores verticales.

```

delay = 0.001
steps_ver = 17320

#_Motores verticales_
#Pines de dirección de los motores verticales
DIR_1 = 17
DIR_2 = 22
#Pines de paso de los motores verticales
STEP_1 = 27
STEP_2 = 23

GPIO.setmode(GPIO.BCM)

#Se configuran todos los pines como salidas
GPIO.setup(DIR_1,GPIO.OUT)
GPIO.setup(STEP_1,GPIO.OUT)
GPIO.setup(DIR_2,GPIO.OUT)
GPIO.setup(STEP_2,GPIO.OUT)
GPIO.setup(DIR_3,GPIO.OUT)
GPIO.setup(STEP_3,GPIO.OUT)

#Función de movimiento ascendente
def move_up(delay=delay,steps_ver=steps_ver):
    GPIO.output(DIR_1, GPIO.HIGH)
    GPIO.output(DIR_2, GPIO.LOW)
    for _ in range(steps_ver):
        GPIO.output(STEP_1, GPIO.HIGH)
        GPIO.output(STEP_2, GPIO.HIGH)
        time.sleep(delay)
        GPIO.output(STEP_1, GPIO.LOW)
        GPIO.output(STEP_2, GPIO.LOW)
        time.sleep(delay)

#Función de movimiento descendente
def move_down(delay=delay,steps_ver=steps_ver):
    GPIO.output(DIR_1, GPIO.LOW)
    GPIO.output(DIR_2, GPIO.HIGH)
    for _ in range(steps_ver):
        GPIO.output(STEP_1, GPIO.HIGH)
        GPIO.output(STEP_2, GPIO.HIGH)
        time.sleep(delay)
        GPIO.output(STEP_1, GPIO.LOW)
        GPIO.output(STEP_2, GPIO.LOW)
        time.sleep(delay)

```

Figura 70. Funciones del movimiento vertical.

### 3.2.3 Movimiento horizontal del sistema

En cuanto al movimiento horizontal, este sistema suspendido por hilos debe mantener la altura de la plataforma invariable durante el movimiento. En caso de que se desplazara el carro portador pero los tambores permanecieran estáticos, la plataforma descendería o ascendería según se moviera el carro a derecha o izquierdas, respectivamente.

Por esa razón, es necesario utilizar todos los motores al unísono. Mientras que el motor horizontal desplaza el carro portador, los motores verticales recogen o sueltan hilo de los tambores en función del sentido horizontal. Si el sentido es a derechas, los tambores recogen cable como si de elevar la plataforma se tratara, mientras que, si el sentido es a izquierdas, los tambores sueltan cable como si descendieran la plataforma.

Para conseguir ejecutar ambas operaciones al mismo tiempo es necesario utilizar la librería Threading. Dentro de las funciones del movimiento horizontal se han definido otras nuevas que determinan el movimiento vertical correspondiente. Así, junto al movimiento horizontal se ejecuta el movimiento vertical en un hilo paralelo.

A diferencia del movimiento vertical puro, este debe tener una frecuencia menor en sus pulsos. La razón se debe a que el diámetro del acople GT2 del motor horizontal es de 10mm, mientras que el diámetro de los tambores es de 36mm. Esto provoca que, para neutralizar el desplazamiento vertical producido por el movimiento horizontal, la velocidad de ese movimiento vertical adicional deba ser 3.6 veces menor al horizontal. Por ello, el valor de la pausa temporal es 3.6 veces superior a la pausa temporal del motor horizontal. Dicho de otra forma, la frecuencia será 3.6 veces menor a la del movimiento vertical puro.

Sin embargo, no basta con solo reducir la frecuencia. También es necesario reducir en 3.6 veces la cantidad de pulsos verticales respecto a los horizontales. Esto se debe a que los tambores deben completar una menor cantidad de revoluciones para recorrer la misma distancia lineal que se realiza en el movimiento horizontal (véase Figura 71).

```

delay = 0.001
steps_hor = 4780

#_Motores verticales_
#Pines de dirección de los motores verticales
DIR_1 = 17
DIR_2 = 22
#Pines de paso de los motores verticales
STEP_1 = 27
STEP_2 = 23

#_Motor horizontal_
#Pin de dirección del motor horizontal
DIR_3 = 24
#Pin de paso del motor horizontal
STEP_3 = 25

GPIO.setmode(GPIO.BCM)

#Se configuran todos los pines como salidas
GPIO.setup(DIR_1,GPIO.OUT)
GPIO.setup(STEP_1,GPIO.OUT)
GPIO.setup(DIR_2,GPIO.OUT)
GPIO.setup(STEP_2,GPIO.OUT)
GPIO.setup(DIR_3,GPIO.OUT)
GPIO.setup(STEP_3,GPIO.OUT)

#Función para el desplazamiento a izquierdas
def move_left(delay=delay, steps_hor=steps_hor):

#Función para mover los motores verticales en un hilo paralelo
def move_left_thread(delay=delay, steps_hor=steps_hor):
    delay2 = delay*3.6
    GPIO.output(DIR_1, GPIO.LOW)
    GPIO.output(DIR_2, GPIO.HIGH)
    for _ in range(steps_hor//3.6):
        GPIO.output(STEP_1, GPIO.HIGH)
        GPIO.output(STEP_2, GPIO.HIGH)
        time.sleep(delay2)
        GPIO.output(STEP_1, GPIO.LOW)
        GPIO.output(STEP_2, GPIO.LOW)
        time.sleep(delay2)

#Iniciar la función anterior en otro hilo paralelo
hilo_left = threading.Thread(target=move_left_thread)
hilo_left.start()

#Mover motor horizontal
GPIO.output(DIR_3, GPIO.HIGH)
for _ in range(steps_hor):
    GPIO.output(STEP_3, GPIO.HIGH)
    time.sleep(delay)
    GPIO.output(STEP_3, GPIO.LOW)
    time.sleep(delay)

#Función para el desplazamiento a derechas
def move_right(delay=delay,steps_hor=steps_hor, steps_ver=0):

#Función para mover los motores verticales en un hilo paralelo
def move_right_thread(dealy=delay, steps_hor=steps_hor):
    delay2 = delay*3.6
    GPIO.output(DIR_1, GPIO.HIGH)
    GPIO.output(DIR_2, GPIO.LOW)
    for _ in range(steps_hor//3.6):
        GPIO.output(STEP_1, GPIO.HIGH)
        GPIO.output(STEP_2, GPIO.HIGH)
        time.sleep(delay2)
        GPIO.output(STEP_1, GPIO.LOW)
        GPIO.output(STEP_2, GPIO.LOW)
        time.sleep(delay2)

#Iniciar la función anterior en otro hilo paralelo
hilo_right = threading.Thread(target=move_right_thread)
hilo_right.start()

#Mover motor horizontal
GPIO.output(DIR_3, GPIO.LOW)
for _ in range(steps_hor):
    GPIO.output(STEP_3, GPIO.HIGH)
    time.sleep(delay)
    GPIO.output(STEP_3, GPIO.LOW)
    time.sleep(delay)

```

Figura 71. Funciones del movimiento horizontal.

### 3.2.4 Código principal

La ejecución del sistema se realizaría diariamente a distintas horas del día, específicamente a las 00:00, a las 06:00, a las 12:00 y a las 18:00 horas.

Al inicio de cada ciclo, se retornaría la plataforma a su posición de partida para asegurarse de que se encuentre en ella, de la manera explicada en el Capítulo 3.1.2 Retorno de posición de la plataforma.

A continuación, la plataforma se desplazaría hacia la primera placa ubicada cerca de la posición inicial, mediante un desplazamiento fijo de aproximación. Al realizar ese primer desplazamiento se comprobaría, y en caso de que fuera necesario, se corregiría la nivelación de la plataforma antes de realizar cualquier otro movimiento.

Una vez la plataforma se considere nivelada, se realizaría la búsqueda del QR correspondiente a la placa en cuestión. Si no se detectara ningún QR, se realizarían desplazamientos para encontrarlo en los alrededores.

En caso de que sí se detectara un QR, se comprobaría si este se encuentra centrado en la imagen de la cámara, realizando ajustes para lograr centrarlo si fuera necesario. Finalmente, se realizaría la fotografía cuando el QR estuviera centrado, esperando de esta manera que se capturara la totalidad de la placa.

Posteriormente a la fotografía, el sistema desplazaría de nuevo la plataforma una distancia fija de aproximación hacia la siguiente placa, y el proceso se repetiría de nuevo. Esto es, la nivelación de la plataforma, búsqueda del QR, centrado del QR y fotografía de la placa. Finalmente, una vez terminado todo el recorrido en zigzag, se retornaría la plataforma a su posición inicial.

Como ayuda visual a esta explicación, en la Figura 72 se puede observar el diagrama de flujo del código principal de manera simplificada:

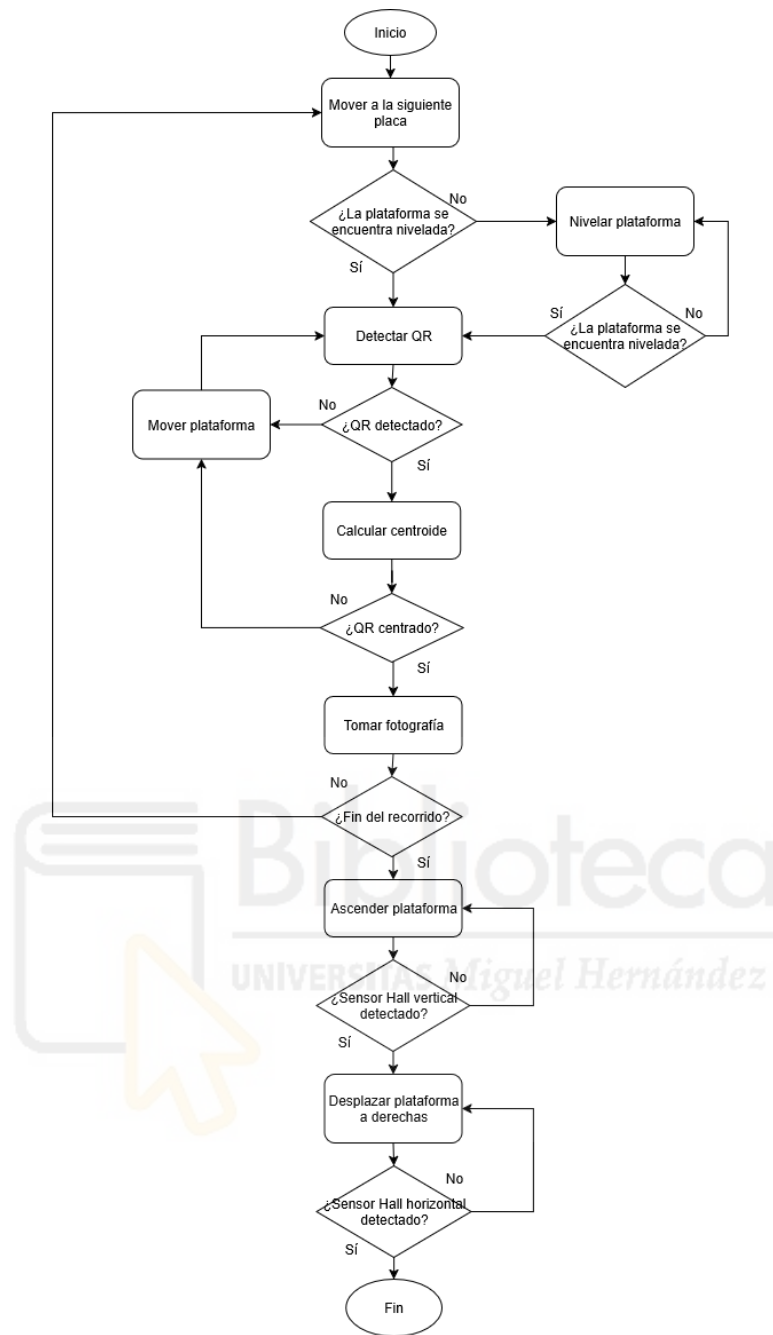


Figura 72. Diagrama de flujo del funcionamiento general del sistema.

### 3.2.4.1 Inclinação de la plataforma

Respecto al control de la inclinación de la plataforma, este se realizaría mediante los actuadores lineales y un acelerómetro anclado a la plataforma, cuyo elemento indica los valores de aceleración en los 3 ejes cartesianos. De esta manera, esta especie de control en bucle cerrado actuaría como medida preventiva ante un posible fallo del sistema, como por ejemplo el descarrilamiento de los hilos de las poleas.

Para lograr esto, se utilizaría el desacople de tambores explicado anteriormente. Para ello, seguiremos la nomenclatura mostrada en las Figura 73a y Figura 73b sobre el acelerómetro y el sistema de referencia del sistema, donde el eje X se correspondería con la dirección del movimiento horizontal, el eje Z al movimiento vertical y el eje Y al eje perpendicular entre los dos anteriores. Sin embargo, realmente nos interesarían únicamente los ejes X e Y, ya son sobre ellos en los que la plataforma puede inclinarse.

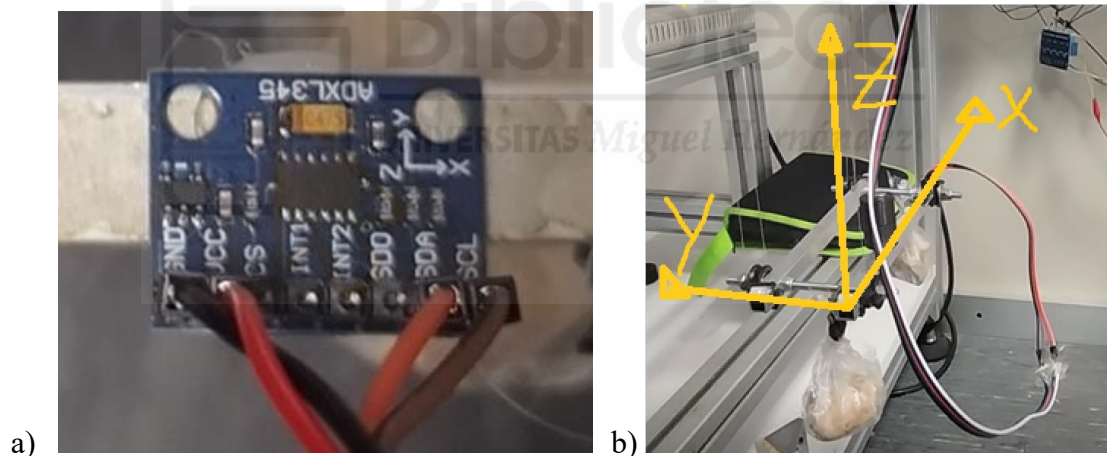


Figura 73. Sistemas de referencia: a) De los ejes del acelerómetro, b) Sobre la plataforma.

El procedimiento planteado para el control de la inclinación consistiría en nivelar uno de los 4 laterales de la plataforma, basándonos en la nomenclatura de la Figura 76, cuyo número indicado en cada una de las esquinas corresponde con el número del tambor que controla cada una de ellas.

En la Figura 74 se han enumerado en color amarillo los motores y en rojo los tambores/actuadores. Para rotar un tambor es necesario alimentar el actuador lineal opuesto del

bloque de transmisión correspondiente. Es decir, para rotar únicamente el tambor número 2, es necesario activar el actuador número 1 para desacoplar así el tambor número 1.

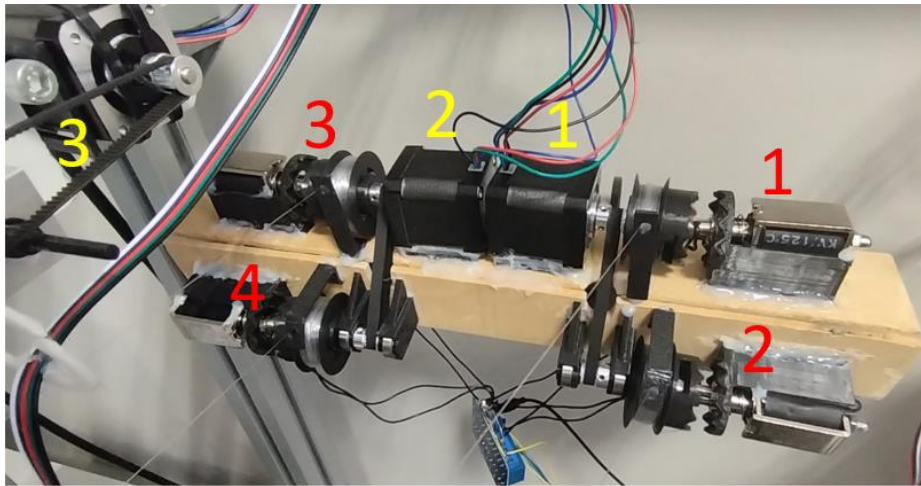


Figura 74. Disposición y nomenclatura de los tambores, actuadores y motores.

Para el caso en que la plataforma se inclinara respecto al eje X, es decir, la aceleración en el eje Y variaría (véase Figura 75), simplemente sería necesario activar un motor u otro para conseguir equilibrarla. Debido a que uno de los motores puede rotar los tambores 1 y 2, y el otro motor puede rotar el 3 y 4, cada uno de ellos elevaría o descendería su respectivo lateral de la plataforma.

Por ejemplo, si la aceleración del eje Y fuera positiva, el sistema pondría en marcha el motor 1 para ascender la plataforma de las esquinas 1 y 2. Si la aceleración fuera negativa, se ascenderían las esquinas 3 y 4 mediante el motor 2.

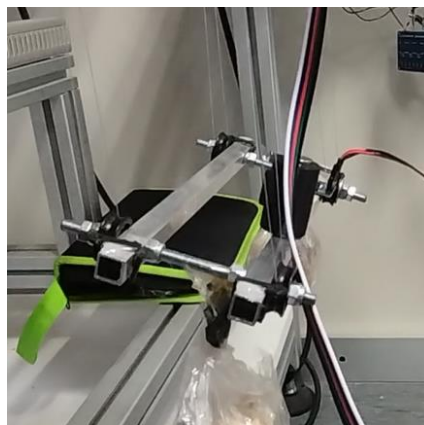


Figura 75. Inclinación de la plataforma sobre el eje X.

Otra variante consistiría en que la plataforma se inclinara respecto el eje Y (Figura 76). En este caso, se necesitaría rotar los tambores superiores o inferiores de cada uno de los bloques de transmisión, utilizando para ello el desacople de tambores. Por ejemplo, si la aceleración en el eje X fuera positiva, se rotarían los tambores 2 y 4, elevando así la plataforma de las esquinas 2 y 4.

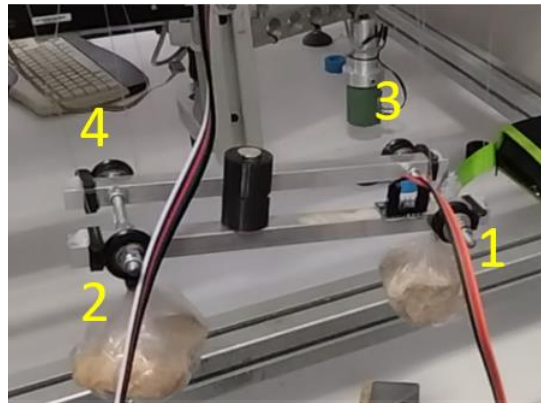


Figura 76. Inclinación de la plataforma en el eje Y.

En caso de que los valores de aceleración tanto del eje X como del Y se encontraran fuera de rango, primero se ajustaría la plataforma en base a la inclinación en el eje X, y posteriormente en el eje Y. Esta elección es arbitraria, pensando en evitar, si fuera posible, el uso del desacople de los tambores.

Este proceso de nivelación se activaría cuando en el eje X o el Y sobrepasaran valores de inclinación superiores a  $20^\circ$ , cuyo valor es escogido experimentalmente de forma aproximada según lo observado durante las pruebas. El incremento en la aceleración presenta un comportamiento senoidal, ya que para un valor de inclinación de  $0^\circ$ , la aceleración en los ejes X e Y es  $0g$ . Por tanto, para una inclinación de  $20^\circ$ , la magnitud de la aceleración es  $3.3g$ , calculada como  $9,81 \cdot \sin(20^\circ)$ .

Por otro lado, el sistema detendría la nivelación una vez que los valores de aceleración en los dos ejes cruzaran un umbral inferior a determinar, dejando así una tolerancia mínima de funcionamiento.

La importancia de nivelar la plataforma radica en evitar que los hilos se salgan de las poleas, evitando así que la plataforma caiga y los hilos se enreden, y no en mejorar la detección de los códigos QR.

Una vez se nivele la plataforma, se procedería a leer fotograma a fotograma buscando el QR correspondiente. En este proceso podrían darse dos posibles resultados: que sí se detectara o que no se detectara un QR.



### 3.2.4.2 Ausencia de detección de códigos QR

En cada fotograma captado por la cámara, se procedería a analizarlo en busca del QR. En caso de que no se detectara uno, la plataforma comenzaría a describir, con movimientos rectilíneos, un trazado en espiral como el mostrado en la Figura 77, cuya finalidad sería la de detectar el área circundante para encontrar el QR.

En este trazado de espiral se realizaría en primer lugar un desplazamiento ascendente. A continuación, se volvería a leer un nuevo fotograma para observar si gracias a ese desplazamiento se hubiera conseguido captar un QR.

Si de nuevo no se lograra detectar un QR, el próximo desplazamiento sería a izquierdas. Y si hubiera que realizar otro desplazamiento, este sería descendente, mientras que el siguiente sería a derechas. Este ciclo se repetiría hasta detectar un QR, realizando desplazamientos cada vez mayores, intentando detectar toda el área circundante.

En la Figura 77 se muestran los desplazamientos que se realizarían en la espiral, basados en unidades, donde cada una correspondería a 100 pulsos enviados del controlador al motor. Este valor de 100 pulsos es provisional, pretendiendo que los desplazamientos sean suaves para evitar balanceos y que la cámara pase de largo del QR.

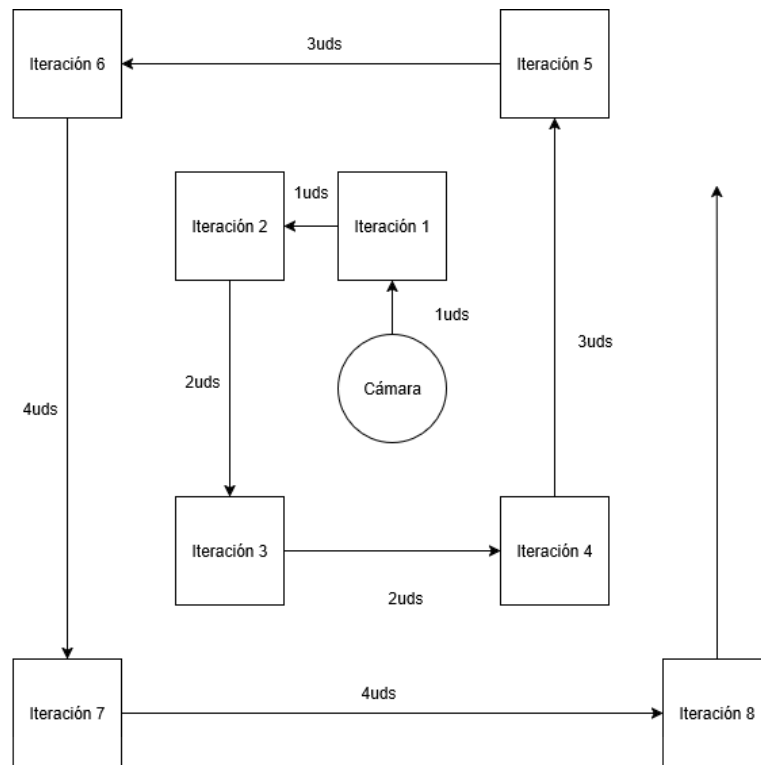


Figura 77. Esquema de desplazamientos del movimiento en espiral.

Esta espiral de movimiento se ha diseñado con la finalidad de poder corregir posibles desviaciones en los desplazamientos entre placas, aunque en la práctica se espera que esas desviaciones, después de realizar los últimos ajustes en el sistema final, sean casi imperceptibles como para detectar los QR de forma inmediata.

### 3.2.4.3 Detección de códigos QR

Cuando se logre detectar un código QR en alguno de los fotogramas capturados, sería necesario que en 5 fotogramas consecutivos se detectara la presencia de alguno para validar realmente su detección, además de validar que su contenido no se encuentre vacío.

Mencionar que en el caso anterior de ausencia de QR se aplicaría de igual forma, esto es, que en 5 fotogramas capturados no se detectara ninguno o que su contenido estuviera vacío. El motivo de ese procedimiento sería el de, además de evitar detectar posibles falsos positivos causados por ruido o sombras en la imagen, permitir que la plataforma amortigüe su posible oscilación pendular producida por su movimiento de traslación. Adicionalmente, con la pausa temporal entre lecturas de códigos se espera que favorezca a esto último.

Con ello, una vez se detecte un QR, se procedería a calcular la posición de su centroide respecto a la imagen. Este es posible calcularlo gracias a que la función de detección de códigos de OpenCV devuelve los puntos de los vértices de ese código detectado.

A continuación, la posición del centroide se compararía con la posición del centro de la imagen de la cámara. Con esto se determinaría la posición relativa del centroide respecto un cuadrado virtual ubicado en el centro de la imagen, cuyo valor provisional corresponde a 60 píxeles por lado (Figura 78).

Si la posición del centroide se encontrara dentro de los límites del cuadrado, la fotografía sería finalmente tomada, y el sistema continuaría su camino hacia la siguiente placa.

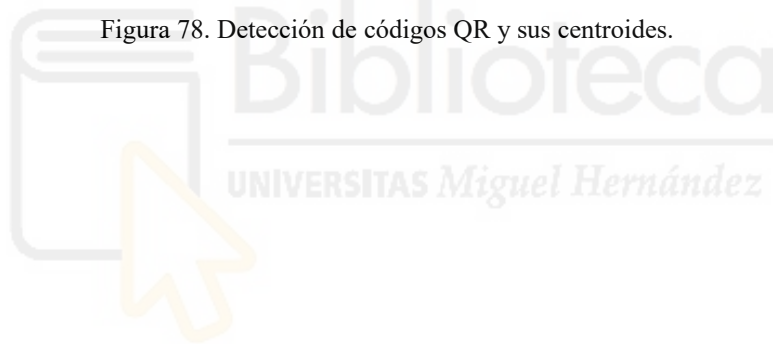
Si, por el contrario, el centroide se encontrara fuera de ese cuadrado, se desplazaría la plataforma en una dirección u otra (vertical u horizontal) para conseguir centrarlos.

Dado el caso en el que el centroide se encontrara por encima del cuadrado, se realizaría un ascenso en la plataforma; si se encontrara por debajo, se realizaría un descenso; si se encontrara a la derecha, se desplazaría hacia la derecha y si se encontrara a la izquierda se desplazaría hacia la izquierda.

La plataforma se desplazaría hasta hacer coincidir las posiciones tanto horizontales como verticales de ambos, mediante intervalos de 200 pulsos para evitar excederse. En este caso, se ha elegido arbitrariamente centrar la plataforma horizontalmente en primer lugar.



Figura 78. Detección de códigos QR y sus centroides.



### 3.2.4.4 Retorno de la plataforma a la posición inicial

Finalmente, pasemos al código de retorno de la plataforma a su posición inicial, el cual se muestra en la Figura 79. En este se realizaría, en primer lugar, un ascenso continuo hasta activarse el sensor Hall, que se ubicaría en la plataforma. En ese instante, la plataforma se detendría y descendería hasta dejar de detectar el imán del carro portador, cuyo fin consistiría en poder detectar un nuevo evento en el sensor.

A continuación, se realizaría un movimiento horizontal a derechas que finalizaría al activarse el sensor Hall horizontal. Durante este desplazamiento, se realizarían comprobaciones en cada iteración sobre el estado del sensor Hall vertical. La razón de esto sería evitar que la plataforma ascendiera indeseadamente debido a pequeñas desincronizaciones de los motores. Si se llegara a detectar el sensor Hall vertical durante este desplazamiento, se descendería la plataforma hasta desactivar el sensor, reanudando posteriormente el desplazamiento horizontal.

Una vez alcanzado el límite horizontal, se separaría la plataforma hasta que el sensor Hall horizontal se desactivara. Finalmente, se realizaría de nuevo un ajuste vertical para corregir un posible descenso indeseado debido a desincronizaciones. De esta manera se espera que la plataforma pudiera alcanzar el punto inicial del sistema.

```
#Reinicio inicial para asegurar que la plataforma se encuentre en el punto inicial

#La plataforma se asciende hasta detectar el imán ubicada en esta
while GPIO.input(8) == 1:
    move_up(steps_ver=100)

#La plataforma se desciende un poco hasta dejar de detectar el imán
while GPIO.input(8) == 0:
    move_down(steps_ver=100)

#La plataforma se desplaza a la derecha hasta detectar el imán del carro portador
while GPIO.input(7) == 1:
    if GPIO.input(8) == 0:
        while GPIO.input(8) == 0:
            move_down(steps_ver=100)
    else:
        move_right(steps_hor=100)

#La plataforma se desplaza un poco a la izquierda hasta dejar de detectar el imán
while GPIO.input(7) == 0:
    move_left(steps_hor=100)

#Se vuelve a confirmar que la plataforma se encuentra en el punto límite superior
while GPIO.input(8) == 1:
    move_up(steps_ver=100)
while GPIO.input(8) == 0:
    move_down(steps_ver=100)
```

Figura 79. Código del retorno de la plataforma a su posición inicial.

## 4 Conexiones eléctricas de los componentes

En la Figura 80 se muestra la conexión general entre los dispositivos utilizados en el proyecto, junto con los valores de voltaje a los que están conectados.

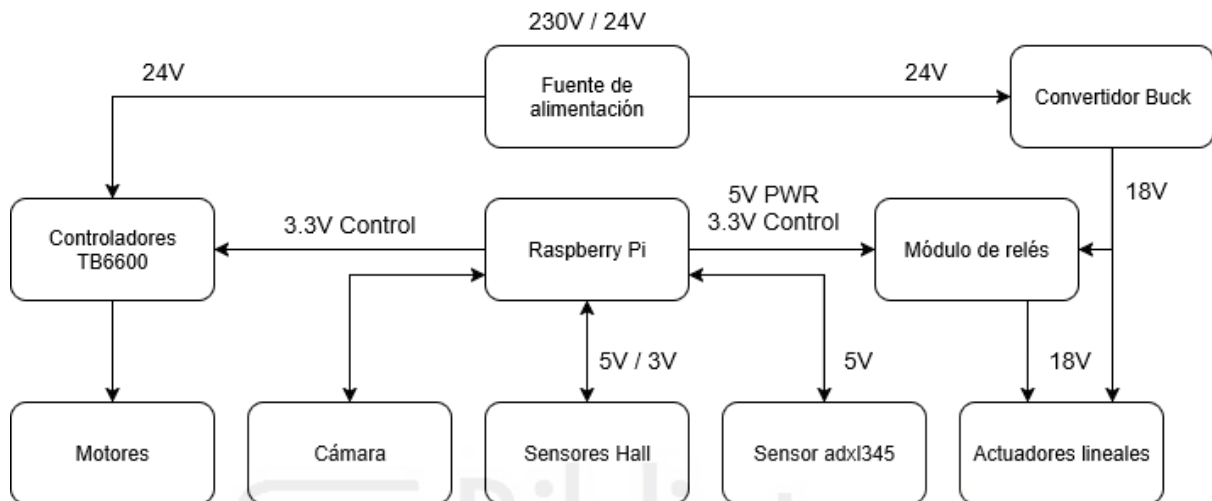


Figura 80. Diagrama general de conexiones de los elementos del sistema junto sus voltajes de operación.

A continuación, se explicarán las conexiones eléctricas y electrónicas en cada uno de estos elementos.

### 4.1 Conexión de la fuente de alimentación

La fuente de alimentación alimenta a los controladores de los motores y al regulador Buck, ambos a 24V (Figura 81).

El uso del regulador Buck se debe a que al alimentar los actuadores lineales a 24V, se ha observado que la fuerza de atracción es tan fuerte y el movimiento tan rápido que del impacto generado puede llegar a desprender algunas de las piezas del mecanismo. Por ello, se han reducido los 24V de la fuente a 18V mediante el regulador, reduciendo así ese impacto generado en gran medida, mientras que la fuerza de atracción se mantiene relativamente alta para el correcto funcionamiento.

Además, la ventaja adicional que se obtiene al reducir el voltaje es la de reducir el sobrecalentamiento de los actuadores, permitiendo que puedan ser activados durante un mayor período de tiempo.

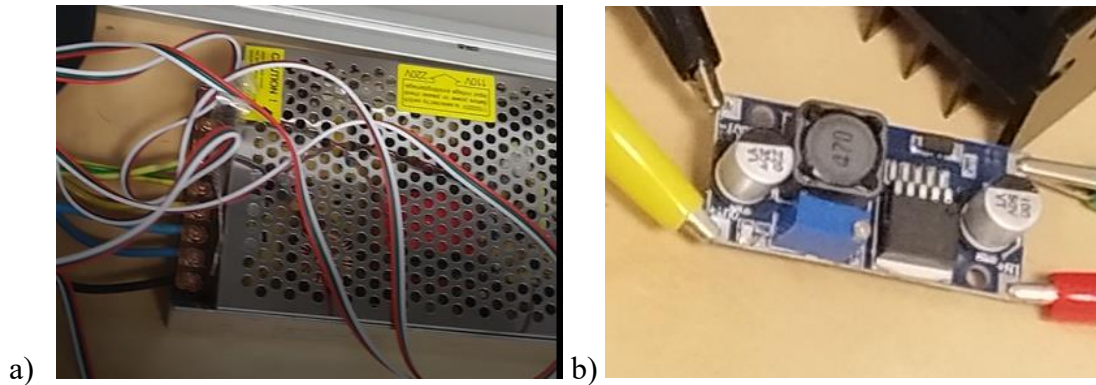


Figura 81. Alimentación eléctrica: a) Conexión de la fuente de alimentación, b) Conexión del regulador Buck.

## 4.2 Conexión de los actuadores lineales en el módulo de relés

Los actuadores lineales están conectados en un punto común que corresponde al polo negativo de alimentación, conectado al negativo del regulador Buck. Los cables restantes en cada uno de los actuadores están conectados a cada uno de los relés del módulo, en el terminal normalmente abierto (NA). Los terminales comunes de los relés (los terminales centrales) se encuentran conectados entre sí, y estos están conectados al borne positivo de alimentación, es decir, la salida positiva del regulador Buck (Figura 82a).

Por otro lado, al desconectar abruptamente cargas inductivas de una fuente de alimentación, se producen altos picos de voltaje inductivos inversos. Por ese motivo, para proteger a la Raspberry de esos picos de voltaje de los actuadores cuando son desconectados, se han soldado una serie de diodos (en este caso diodos rectificadores 1n4007) en antiparalelo en cada uno de los actuadores. Es decir, los ánodos de los diodos se encuentran conectados en el punto común de los actuadores (polo negativo), mientras que los cátodos de los diodos se encuentran conectados en los cables positivos de cada uno de los actuadores (Figura 82b).

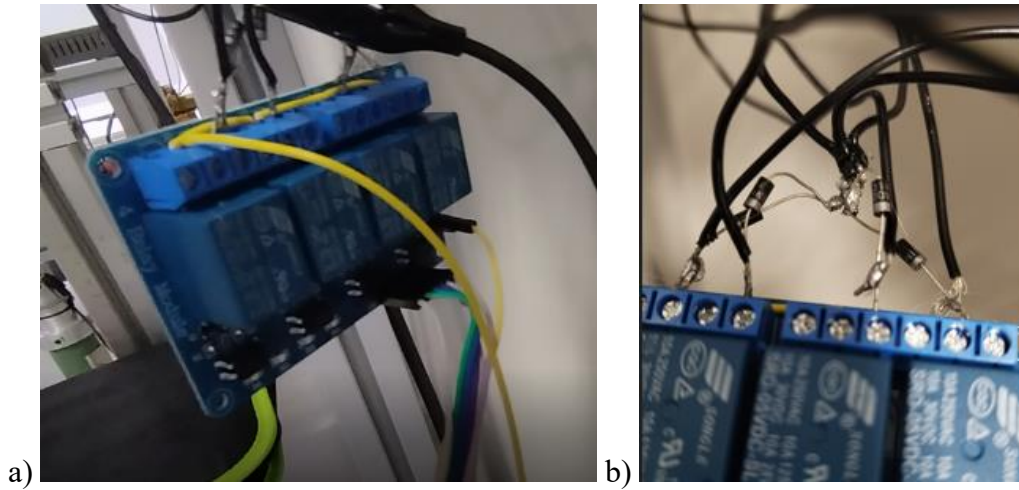


Figura 82. Conexiones de: a) El módulo de relés, b) Los diodos en antiparalelo a los actuadores.

### 4.3 Conexión de los pines de control de los relés

En el módulo de relés, los pines de control se encuentran conectados directamente a los pines de la Raspberry, ya que el módulo contiene optoacopladores integrados que evitan añadir componentes de seguridad.

Este módulo funciona con lógica inversa, es decir, para activar uno de los relés se necesita que el pin de control de ese relé en cuestión posea un estado 0 lógico.

Las bobinas de los relés se alimentan de los 5V de la Raspberry, mientras que la lógica de control del módulo hace uso del pin de 3.3V (Figura 83).

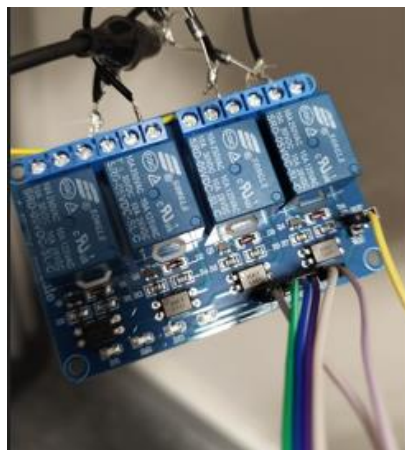


Figura 83. Conexión de los pines de control y alimentación en el módulo de relés.

## 4.4 Conexión de los controladores de los motores

Respecto a las entradas de control, los controladores de los motores TB6600 disponen de 3: el pin de dirección del motor, la entrada de pulsos y la entrada de habilitación (la cual no se ha utilizado al no haberse considerado necesaria).

Los pines GPIO encargados del control de los motores están conectados en los terminales positivos de los controladores, permitiendo utilizarlos mediante lógica positiva. Los bornes negativos están conectados al polo negativo (Figura 84).

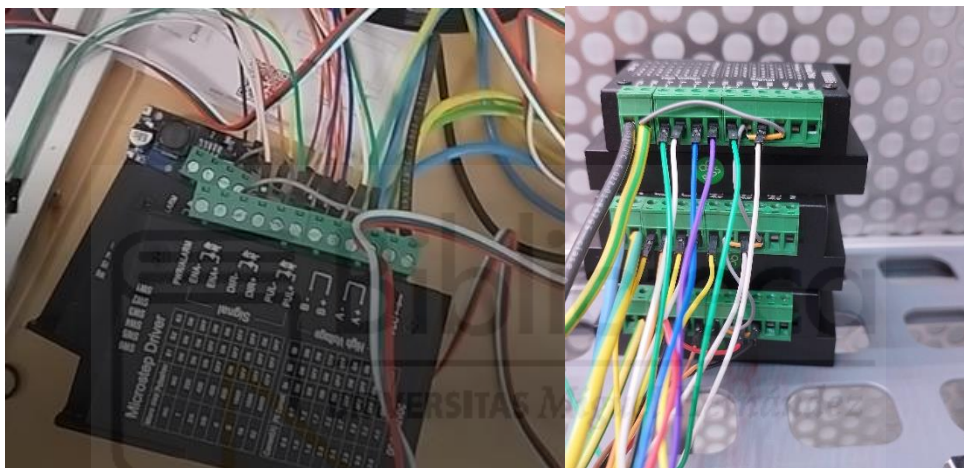


Figura 84. Conexión de los pines de control en los controladores de los motores.

Respecto a la alimentación de los motores, los bornes de salida de los controladores se han conectado a los motores según el orden que se indica en la hoja de especificaciones de los motores (Figura 85).

Type of connection / Art der Verbindung		Motor / Motor			
Pin NO / Pin-Nr	Bipolar / Bipolar	Leads / Leitung		Winding / Wicklung	
1	A+ —	BLK / Schwarz		A+	
2	A- —	GRN / Grün		A-	
3	B+ —	RED / Rot		B+	
4	B- —	BLU / Blau		B-	
Step / Schritt	A+	B+	A-	B-	CCW ↓ ↑ CW
1	+	+	-	-	
2	-	+	+	-	
3	-	-	+	+	
4	+	-	-	+	

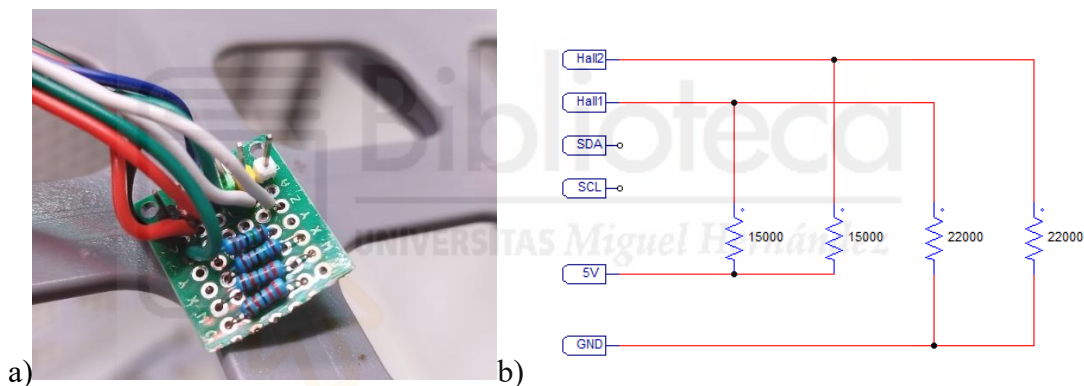
**STEPPERONLINE®**

Figura 85. Hoja de características de las conexiones de los motores paso a paso.

## 4.5 Conexión de los sensores Hall y acelerómetro

Según la hoja de características de los sensores Hall OH137, su funcionamiento sigue una lógica inversa, por lo que es necesario conectarles resistencias ‘pull-up’ (Figura 86a). Es decir, cuando no se detecta un campo magnético, el voltaje en la salida corresponde a VCC, mientras que cuando se detecta campo magnético el voltaje corresponde a GND.

Los pines GPIO de la Raspberry Pi no deben recibir un voltaje superior a 3.3V, pero los sensores Hall están alimentados con 5V. Por ello ha sido necesario configurar un divisor de voltaje para reducir el voltaje de salida de los sensores de 5V a unos 3V. La combinación de los valores de las resistencias elegidos son 22k ohmios y 15k ohmios, conectando la resistencia de 22k ohmios entre la señal del sensor y GND, según se muestra en la Figura 86b.



Para las conexiones se utilizarían 2 tipos de cables planos con diferentes números de conductores, uno conformado por 5 y otro conformado por 3.

El de 3 conductores se utilizaría para conectar el sensor Hall horizontal, mientras que el de 5 conductores se utilizaría para conectar el acelerómetro y el sensor Hall vertical. Dos de estos últimos conductores corresponderían a la alimentación, la cual compartirían ambos elementos. Otros 2 conductores corresponderían a la comunicación I2C con el acelerómetro, mientras que el conductor restante correspondería a la salida del sensor Hall. Debido a que el acelerómetro y el sensor vertical se encontrarían en la plataforma, se conectarían ambos elementos al mismo cable para utilizar el menor número de ellos.

Según el esquema de la Figura 86b, el cable plano de 5 conductores se conectaría a los pines GND, 5V, SDA, SCL y Hall1. Adyacente a este cable plano, se conectaría el de 3 conductores en los pines GND, 5V y Hall2.

## 4.6 Conexiones de los pines en la Raspberry Pi

En la Figura 87 se muestra la distribución de los 40 pines de la Raspberry Pi y la conexión a los diferentes dispositivos del circuito.

Conexión	Nombre	Pin		Nombre	Conexión
PWR control del módulo relés	3V3 PWR	1	2	5V PWR	PWR sensores Hall y adx1345
SDA adx1345	GPIO 2 (SDA)	3	4	5V PWR	PWR relés
SCL adx1345	GPIO 3 (SCL)	5	6	GND	GND sensores Hall Y adx1345
	GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)	
DIR1 / STEP1 (-)	GND	9	10	GPIO 15 (RXD)	
DIR 1 (+)	GPIO 17	11	12	GPIO 18 (PCM_CLK)	
STEP 1 (+)	GPIO 27	13	14	GND	DIR2 / STEP2 (-)
DIR 2 (+)	GPIO 22	15	16	GPIO 23	STEP 2 (+)
	3V3 PWR	17	18	GPIO 24	DIR 3 (+)
	GPIO 10 (OSI)	19	20	GND	DIR3 / STEP3 (-)
	GPIO 9 (ISO)	21	22	GPIO 25	STEP 3 (+)
	GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)	Hall Vertical
GND módulo relés	GND	25	26	GPIO 7 (CE1)	Hall Horizontal
	GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)	
Actuador lineal 3	GPIO 5	29	30	GND	
Actuador lineal 2	GPIO 6	31	32	GPIO 12 (PWM0)	
	GPIO 13 (PWM1)	33	34	GND	
	GPIO 19 (PCM_FS)	35	36	GPIO 16	Actuador lineal 4
Actuador lineal 1	GPIO 26	37	38	GPIO 20 (PCM_DIN)	
	GND	39	40	GPIO 21 (PCM_DOUT)	

Figura 87. Tabla de conexiones de los pines de la Raspberry a los diversos dispositivos.

## 5 Resultados

En este capítulo se presentan los resultados obtenidos tras el montaje del mecanismo y la validación de las pruebas realizadas en el análisis de códigos QR.

Respecto al mecanismo físico del carro portador y la plataforma, este funcionó de manera estable durante su movimiento, tanto vertical como horizontalmente (Figura 88 y Figura 89). No obstante, se presentó una ligera desviación en la altura de la plataforma a causa del movimiento horizontal. Este comportamiento no se considera un problema grave, ya que previsiblemente podría ser corregido ajustando a nivel unitario la frecuencia de los pulsos de los motores verticales.

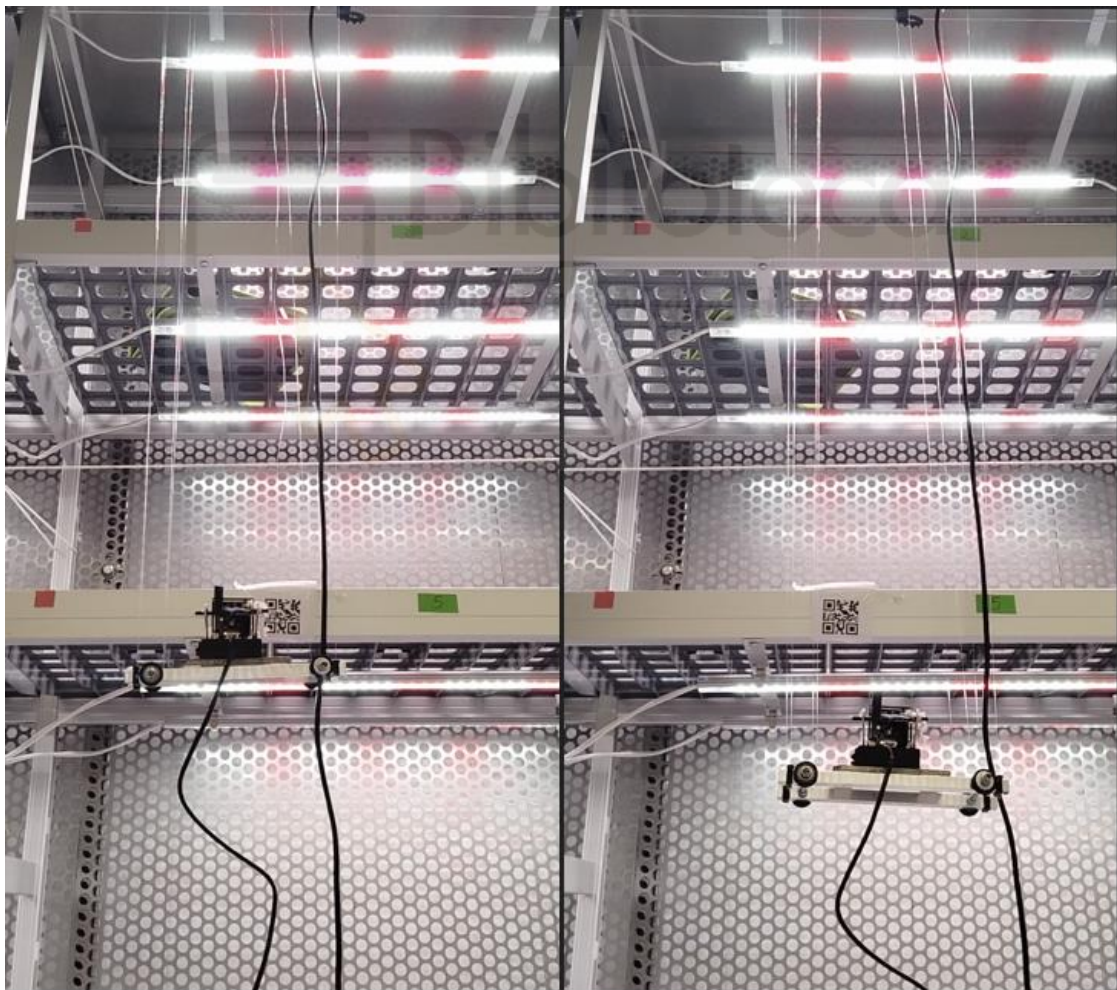


Figura 88. Vistas en diferentes posiciones de la plataforma pendida por los hilos con la cámara anclada.



Figura 89. Vista del sistema de hilos y poleas: a) En el carro portador, b) Plataforma pendida sobre los hilos que bajan del carro portador.

En cuanto al mecanismo de transmisión, estos se han presentado con robustez, engranando los tambores de ambos bloques y enrollando los hilos correctamente (Figura 90 y Figura 91). Además, se ha observado que los tambores acoplan y desacoplan de manera estable, siendo su única limitación la que se comentó en el Capítulo 3.1.4 Control de inclinación de la plataforma a causa del número de ranuras.

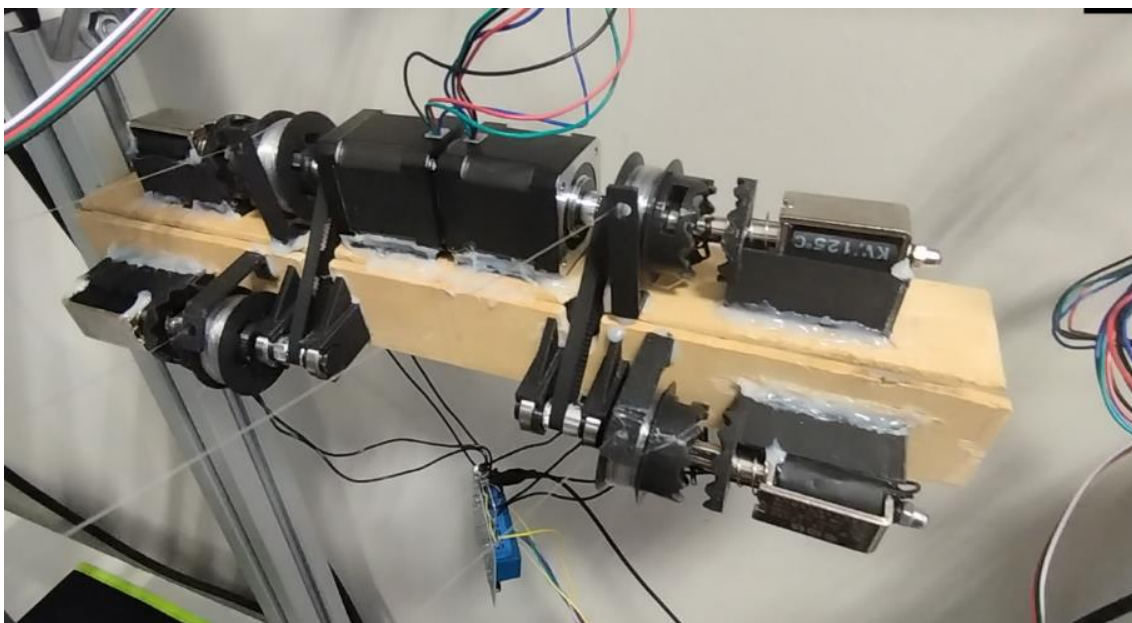


Figura 90. Montaje completo del sistema mecánico de transmisión.

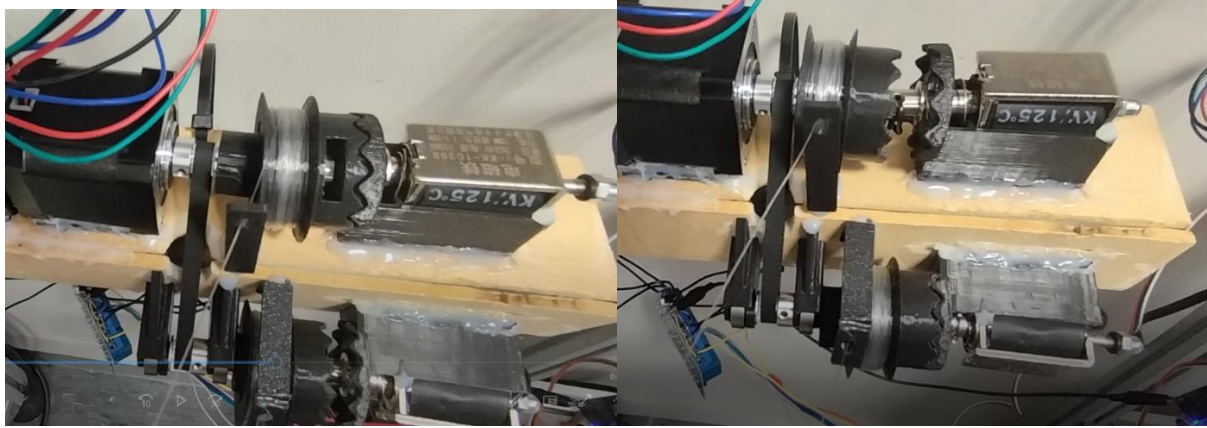


Figura 91. Posiciones de los tambores, tanto en sus posiciones naturales como desacoplados.

Respecto a los sensores Hall, únicamente se probó uno de ellos de manera experimental en el laboratorio, en el ascenso vertical. Se tuvo éxito en su funcionamiento, detectando el límite vertical de la plataforma (Figura 92). No obstante, su integración en el sistema final no se ha llevado a cabo debido a una necesidad, tanto de organización del cableado como sujeción de los componentes, para garantizar un funcionamiento estable.

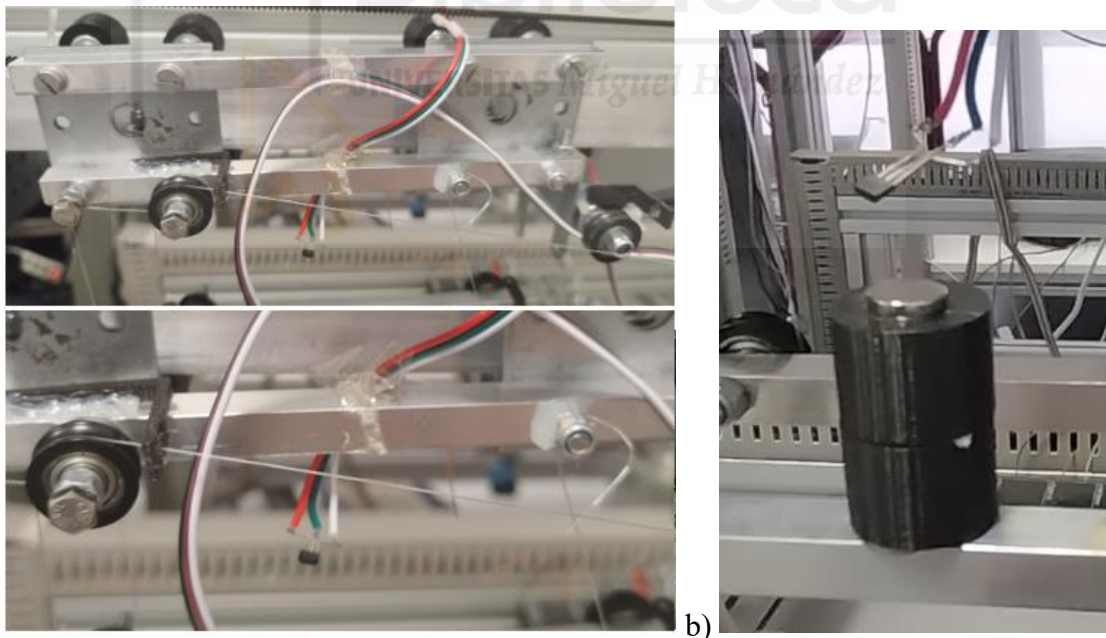


Figura 92. Posición experimental del: a) Sensor Hall vertical, b) Imán en la plataforma y su proximidad entre ambos elementos.

Finalmente, en cuanto a la detección de los códigos QR, esta ha sido llevado a cabo con éxito en las pruebas experimentales. El sistema puede detectar la presencia de los códigos y

determinar la posición de su centroide, decidiendo mover la plataforma en una dirección u otra (Figura 93 y Figura 94). Sin embargo, debido a la proximidad entre la cámara y los códigos, estos últimos deberían imprimirse con un tamaño menor, ya que el tamaño del QR de prueba ocupaba casi toda el área de la imagen, por lo que al moverse la plataforma una ligera distancia, el QR eventualmente desaparecía de la imagen.

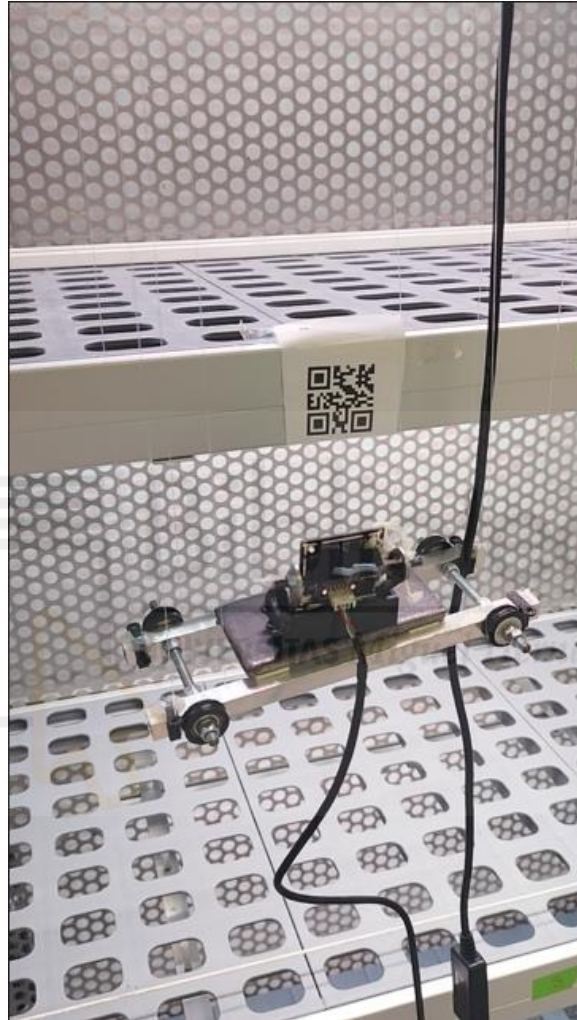


Figura 93. Detección del QR mediante la cámara anclada a la plataforma suspendida por los hilos.

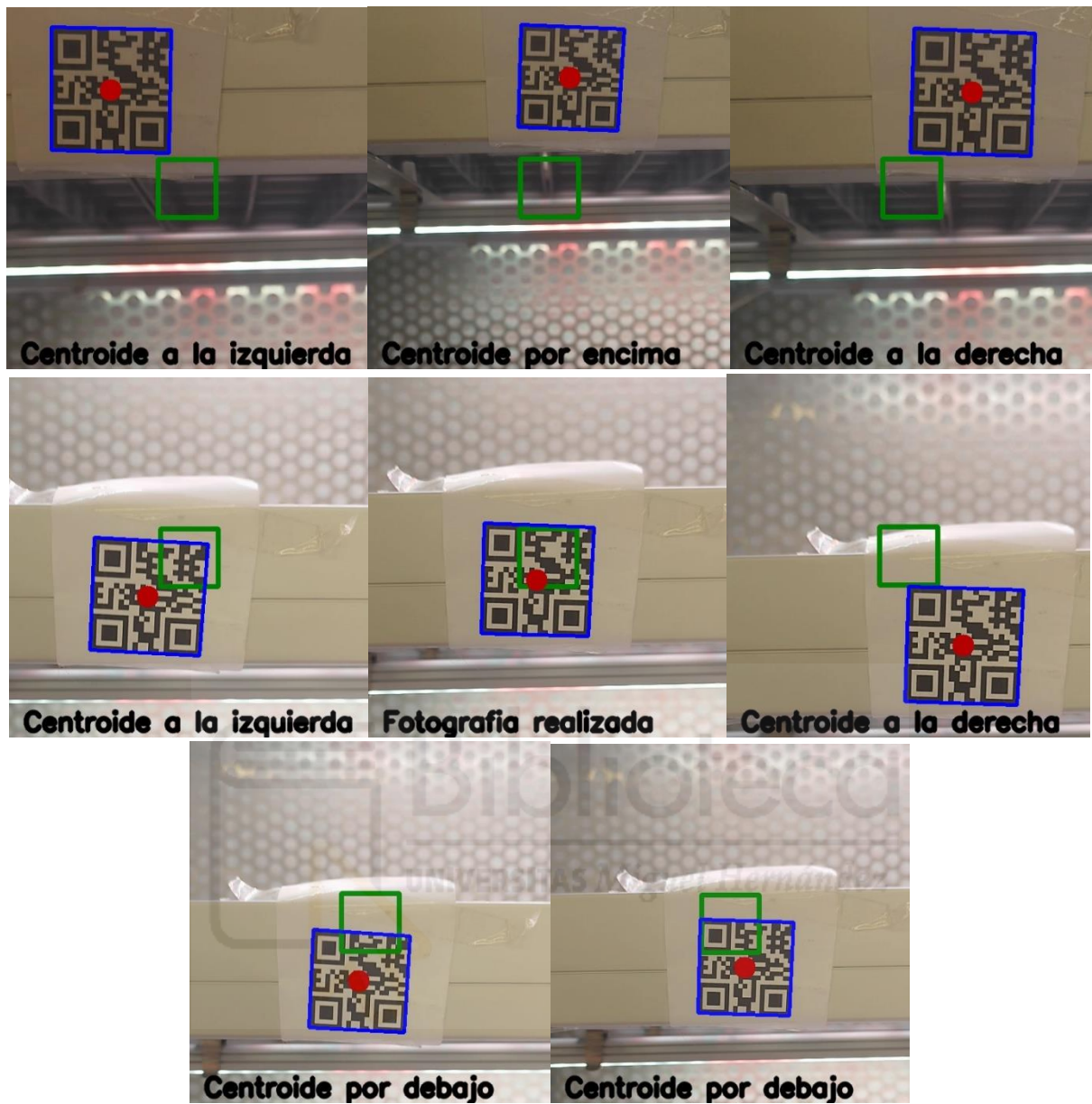


Figura 94. Diferentes posiciones del centroide del QR respecto al centro de la imagen, mostrada con un cuadrado verde.

## 6 Conclusiones

Para finalizar, el proyecto realizado ha consistido en un trabajo sobre sistemas mecánicos, centrándose mayoritariamente en el área de hardware más que en la de software. La intención consistía en mejorar el antiguo sistema automático instalado en la cámara, tanto en el espacio físico ocupado como en la precisión de su funcionamiento.

Por un lado, se ha conseguido reducir el espacio ocupado dentro de la cámara. Se ha pasado a ocupar únicamente una sección de la parte superior de la estantería, en lugar de ocupar el equivalente a la mitad de la sección del pasillo de la cámara. Sin embargo, con la posible integración de los sensores, la cantidad de cables dentro de la cámara se vería aumentada.

Por otro lado, se han planteado posibles mejoras a realizar respecto al anterior sistema de control. El sistema anterior constaba únicamente del desplazamiento fijo establecido entre placas, sin posibilidad a verificar si la fotografía se estaba realizando correctamente. En este proyecto se ha ampliado esa lógica, añadiendo un control posterior en tiempo real sobre la posición de la cámara mediante la lectura de códigos QR.

Adicionalmente, se ha conseguido poder rotar 4 tambores independientes mediante el mecanismo de transmisión, utilizando para ello únicamente dos motores con el fin de reducir costos, espacio y material innecesario, lo cual suponía un desafío en diseño.

Aunque no se probó la integración completa de todos los elementos en el entorno definitivo, los ensayos independientes mediante los sensores Hall y los códigos QR permiten prever que, posterior a las últimas conexiones y correcciones en el sistema, este funcione adecuadamente en su conjunto. Además, el montaje actual podría servir de base para posibles mejoras en un futuro, proponiendo algunos ejemplos en el siguiente capítulo.

## 7 Trabajos futuros

Debido a las limitaciones físicas del montaje experimental y a los limitados materiales a utilizar, la precisión de este sistema no es perfecta.

Sin embargo, se plantean diversas posibles soluciones que podrían ayudar a mejorar el sistema, aumentando su estabilidad y precisión. Estas son:

- Probar a diseñar tambores con un mayor número de ranuras o ranuras más estrechas para así reducir el error angular.
- Diseñar un embrague basado en acoples electromagnéticos tales que puedan acoplar en cualquier posición, eliminando así el error producido por el número de ranuras y por los dientes de los engranes.
- En caso de que se deseara aumentar la velocidad de desplazamiento de este sistema, podría ser interesante diseñar un controlador PID que controlara el balanceo de la plataforma para estabilizarla en tiempo real.
- El cable USB de la cámara y el de 5 conductores colgarían del sistema, pudiendo desequilibrar la plataforma debido a la propia masa de los cables. Podría ser interesante agregar un muelle de torsión que recogiera los cables según la plataforma ascendiera, logrando además un entorno más ordenado.
- Una propuesta ampliada respecto a la anterior, consistiría en añadir un dispositivo, sujeto en la propia plataforma y alimentado por una batería, que pudiera detectar los códigos QR y leer las señales de los sensores. Este se comunicaría por vía inalámbrica con otro dispositivo que, ubicado en otro lugar de la cámara, se encargara exclusivamente del movimiento de los motores. Respecto al dispositivo conectado a la batería, este podría consistir en uno con bajo consumo, como por ejemplo el ESP32-CAM. De esta manera, se simplificaría en gran medida el uso de cables en el sistema.

## 8 Referencias bibliográficas

[1] C. Cardona Marín, “Diseño e implementación de un sistema semiautomatizado de fenotipado de la arquitectura radicular en tomate”, Trabajo Fin de Máster, Universidad Miguel Hernández, 2023

[2] Raspberry Pi Foundation, “GPIO and the 40-pin header”, Raspberry Pi Documentation. [En línea]. Disponible en: <https://www.raspberrypi.com/documentation/computers/images/GPIO-Pinout-Diagram-2.png?hash=df7d7847c57a1ca6d5b2617695de6d46>. [Último acceso: ago. 2025]

[3] Luis Llamas, “Cómo usar OpenCV en Raspberry Pi”, luisllamas.es. [En línea]. Disponible en: <https://www.luisllamas.es/como-usar-opencv-en-raspberry-pi/>. [Último acceso: ago. 2025]

[4] [nota.nkmk.me](https://nota.nkmk.me), “Obtener el tamaño de la imagen (ancho, alto) con Python, OpenCV, Pillow (PIL)“. [En línea]. Disponible en: [https://note-nkmk-me.translate.google.com/en/python-opencv-pillow-image-size/?x\\_tr\\_sl=en&x\\_tr\\_tl=es&x\\_tr\\_hl=es&x\\_tr\\_pto=rq](https://note-nkmk-me.translate.google.com/en/python-opencv-pillow-image-size/?x_tr_sl=en&x_tr_tl=es&x_tr_hl=es&x_tr_pto=rq). [Último acceso: ago. 2025]

[5] programacionpython80889555, “Generando códigos «QR» en python”, El programador chapuzas. [En línea]. Disponible en: <https://programacionpython80889555.wordpress.com/2020/02/04/generando-codigos-qr-en-python/>. [Último acceso: ago. 2025]

[6] programacionpython80889555, “Lectura de códigos «QR» en pthon”, usando «opencv», El programador chapuzas. [En línea]. Disponible en: <https://programacionpython80889555.wordpress.com/2020/02/18/lectura-de-codigos-qr-en-ptthon-usando-opencv/>. [Último acceso: ago. 2025]

[7] pimoroni, “adx1345-python”, GitHub repository. [En línea]. Disponible en: <https://github.com/pimoroni/adx1345-python>. [Último acceso: ago.2025]

[8] Juan (Código Pitón), “Cómo Usar Hilos (Threads) en Python”, Fundamentos de Python. [En línea]. Disponible en: <https://www.codigopiton.com/como-usar-hilos-o-threads-en-python/>. [Último acceso: ago. 2025]

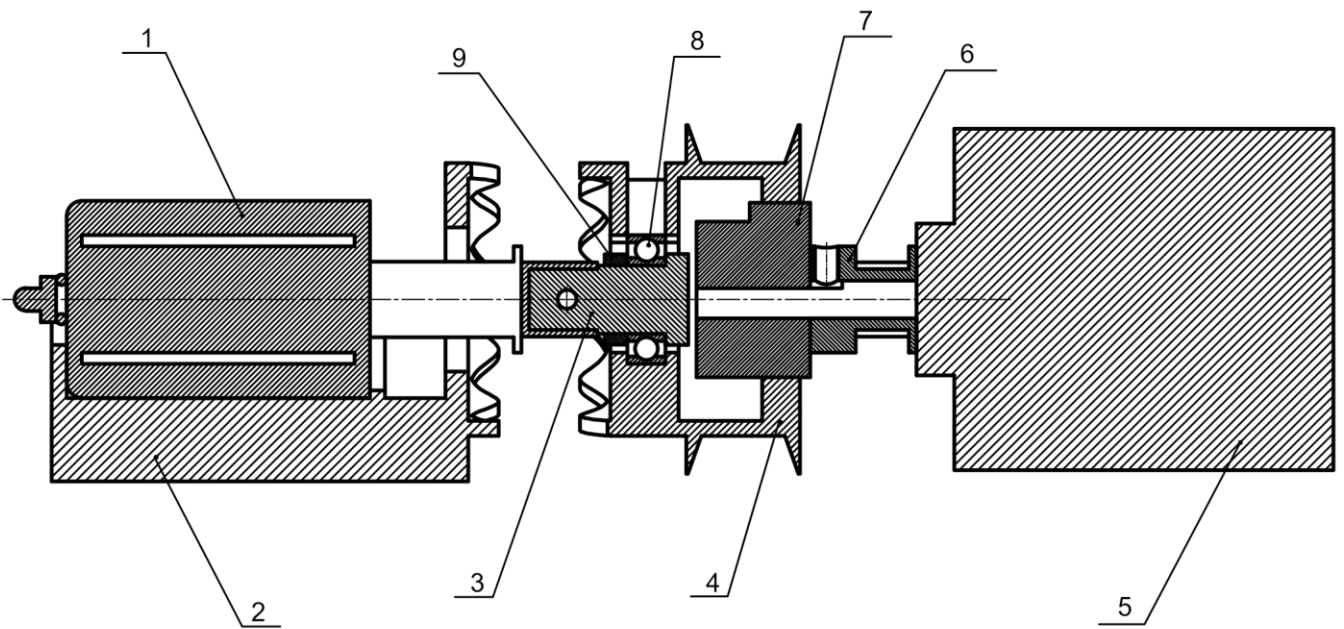
[9] Craft Mechanics, “Mechanical Principle - Jaw Clutch Mechanism”, YouTube, Nov. 6, 2024. [En línea]. Disponible en: <https://www.youtube.com/watch?v=WBjEtMhhnWE>. [Último acceso: ago. 2025]

[10] Nanjing Ouzhuo Technology Co., Ltd., “OH137 Datasheet (PDF)”, Alldatasheet.es. [En línea]. Disponible en: <https://www.alldatasheet.es/datasheet-pdf/pdf/547704/OHHALLSENSOR/OH137.html>. [Último acceso: ago. 2025]



## 9 Anexo I: Planos de conjunto y de despiece





Biblioteca

9	ANILLA DE RETENCIÓN	1	PLA	9X12X3
8	RODAMIENTO DE BOLAS	1	ACERO	9X17X5
7	CILINDRO CHAVETA	1	PLA	
6	ACOPLE DENTADO	1	PLA	16T W6 B5
5	MOTOR	1	PLA	
4	TAMBOR	1	PLA	
3	EJE TAMBOR	1	PLA	
2	SOPORTE ACTUADOR	1	PLA	
1	ACTUADOR LINEAL	1	ACERO	1039B
MARCA	DESCRIPCIÓN	Nº PIEZAS	MATERIAL	MODELO

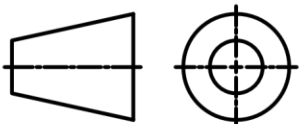
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

1:1

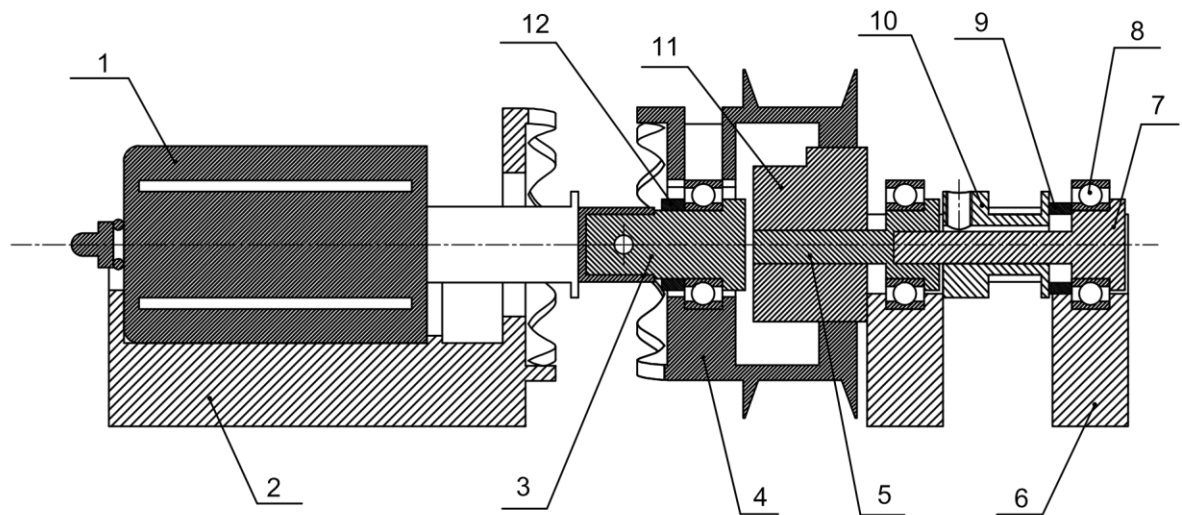
Nº DE PLANO

1.1

HOJA

1/1

MECANISMO DE TRANSMISIÓN PRIMARIO



12	ANILLA DE RETENCIÓN	1	PLA	9X12X3
11	CILINDRO CHAVETA	1	PLA	
10	ACOPLE DENTADO	1	ALUMINIO	16T W6 B5
9	ANILLA DE SEPARACIÓN	1	PLA	9X12X3
8	RODAMIENTO DE BOLAS	3	ACERO	9X17X5
7	EJE SECUNDARIO	1	PLA	
6	SOPORTE RODAMIENTO	2	PLA	
5	EJE HUECO SECUNDARIO	1	PLA	
4	TAMBOR	1	PLA	
3	EJE TAMBOR	1	PLA	
2	SOPORTE ACTUADOR	1	PLA	
1	ACTUADOR LINEAL	1	ACERO	1039B
MARCA	DESCRIPCIÓN	Nº PIEZAS	MATERIAL	MODELO

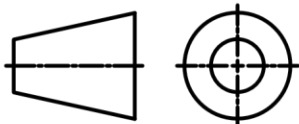
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

1:1

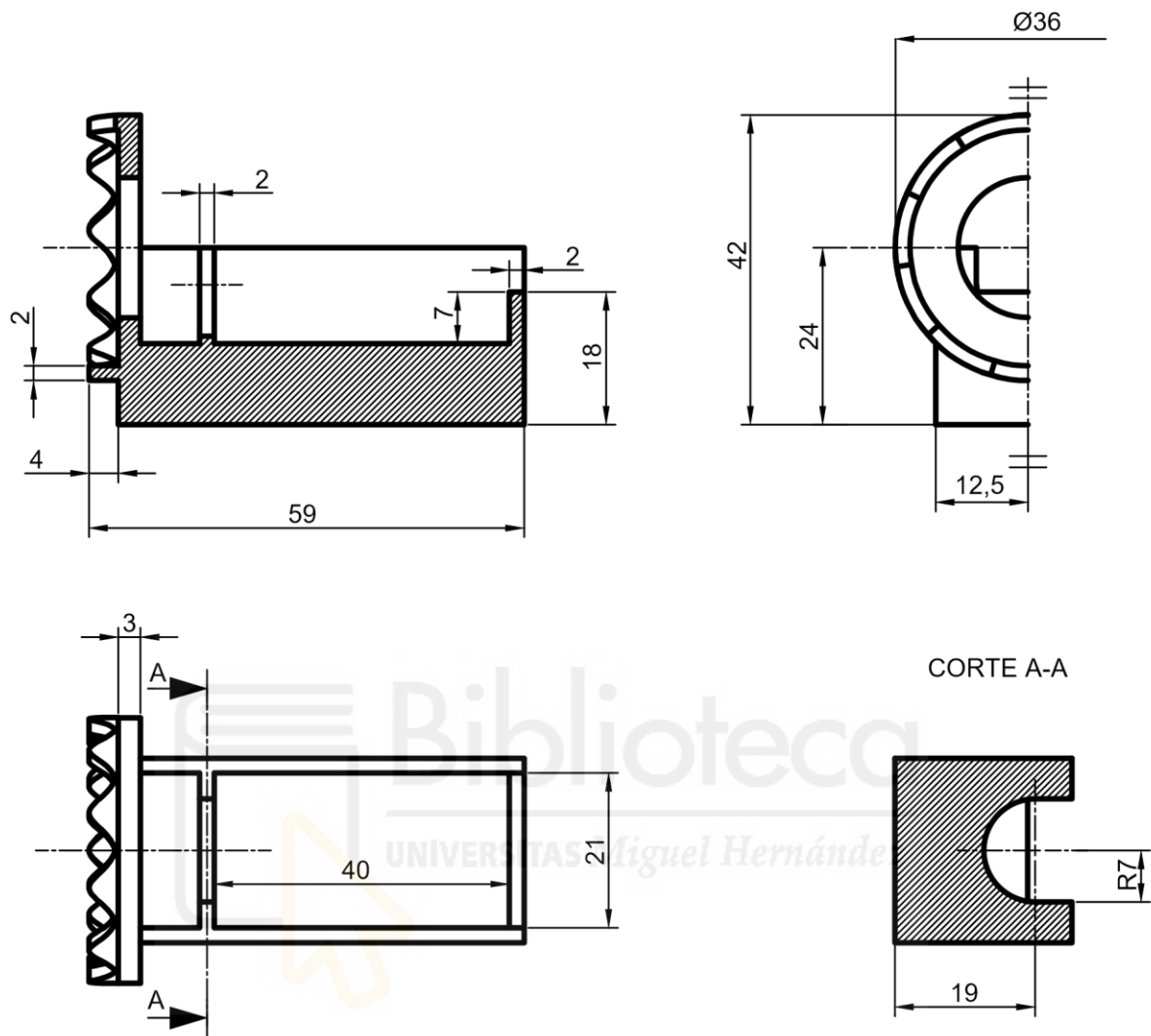
Nº DE PLANO

1.2

HOJA

1/1

MECANISMO DE TRANSMISIÓN SECUNDARIO



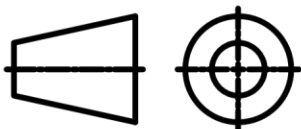
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

1:1

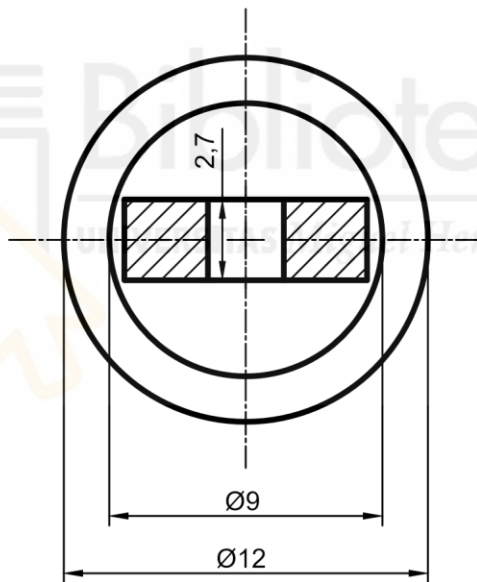
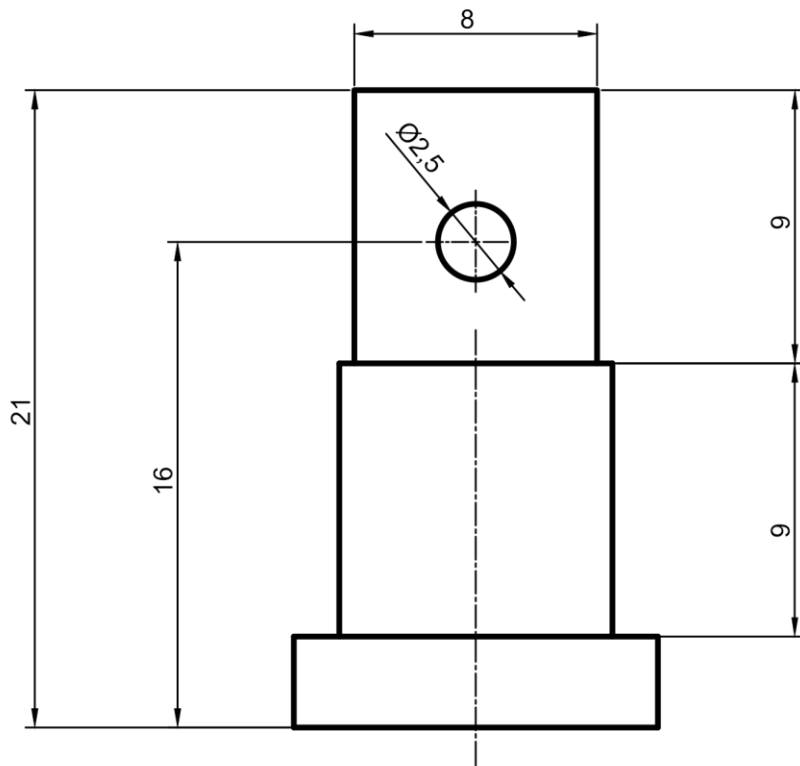
Nº DE PIEZA

1

HOJA

1/1

SOPORTE DEL ACTUADOR LINEAL



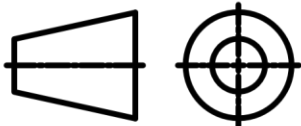
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

4:1

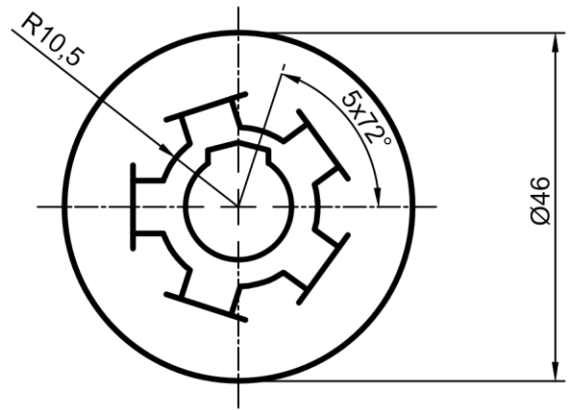
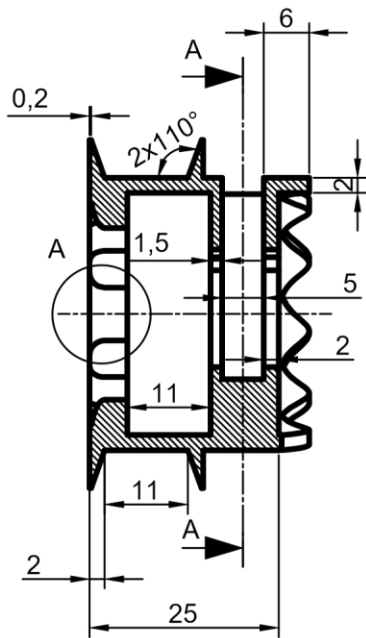
Nº DE PIEZA

2

HOJA

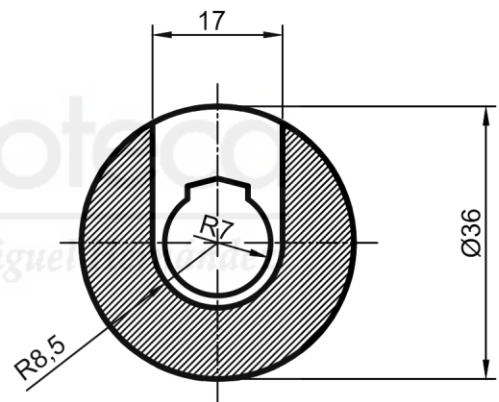
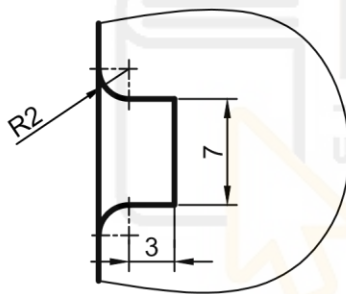
1/1

EJE DEL TAMBOR



CORTE A-A

DETALLE 'A'



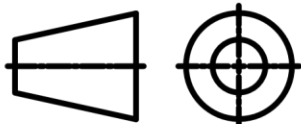
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

1:1

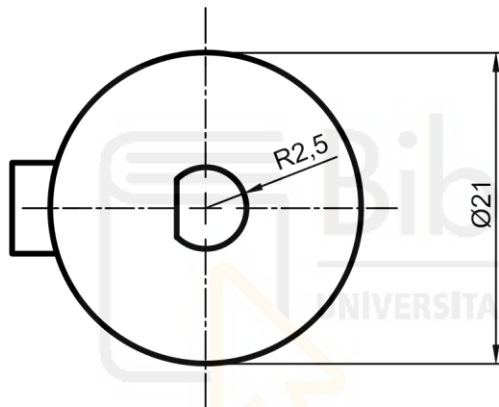
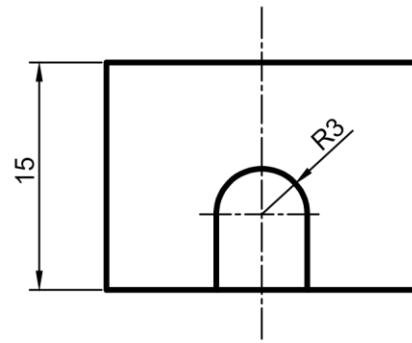
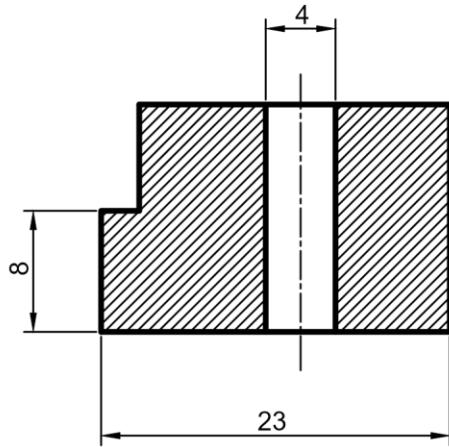
Nº DE PIEZA

3

HOJA

1/1

TAMBOR



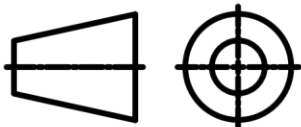
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

2:1

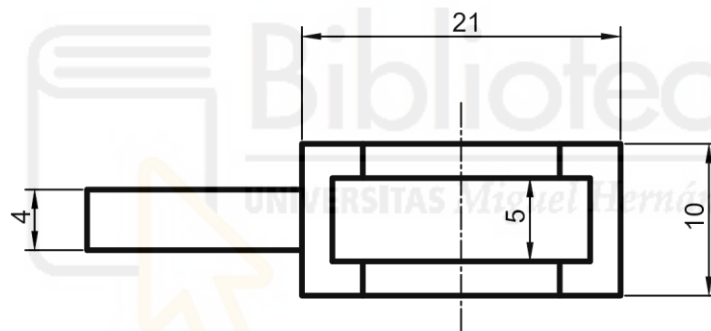
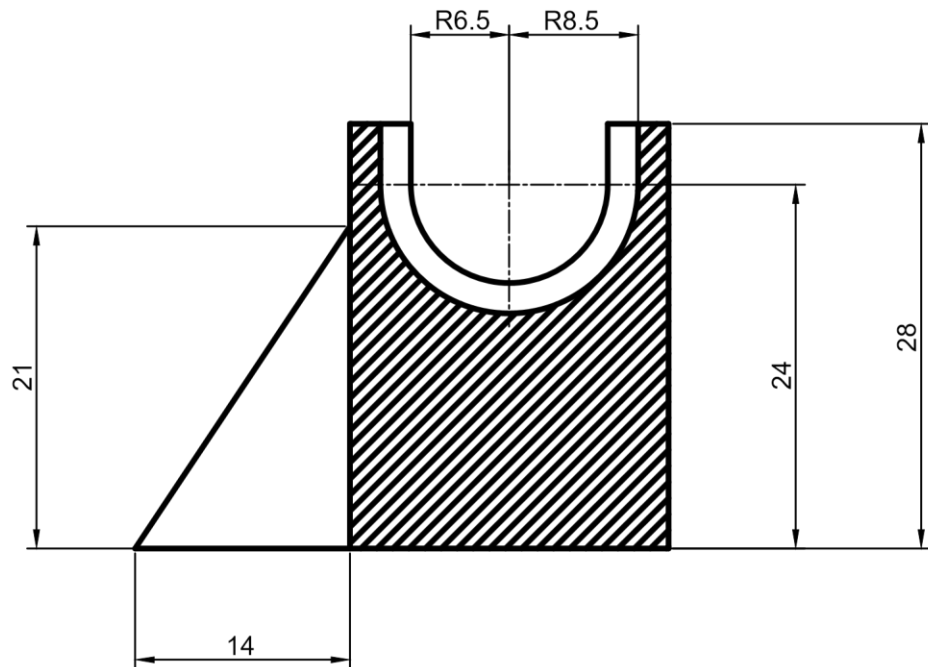
Nº DE PIEZA

4

HOJA

1/1

CILINDRO CHAVETA



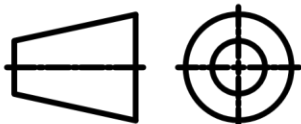
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

2:1

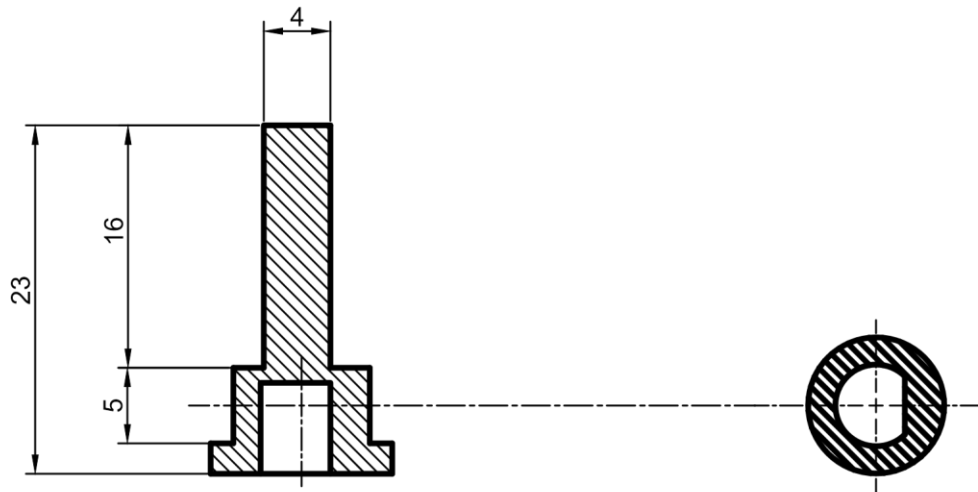
Nº DE PIEZA

5

HOJA

1/1

SOPORTE AUXILIAR PARA RODAMIENTO



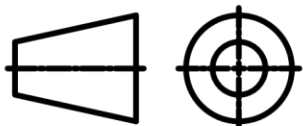
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

2:1

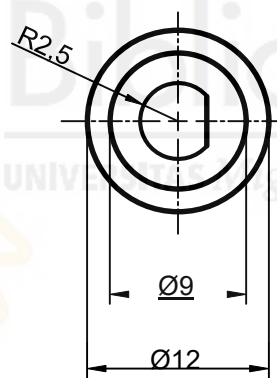
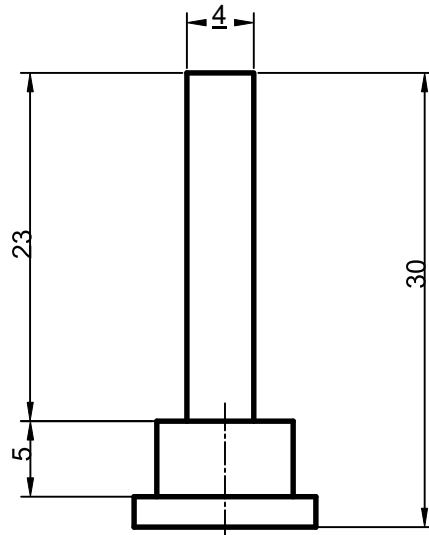
Nº DE PIEZA

6

HOJA

1/1

EJE DE TRANSMISIÓN SECUNDARIO HUECO



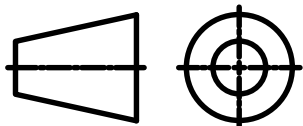
NOMBRE

DAVID OLIVARES CLIMENT

TRABAJO FIN DE GRADO

TITULACIÓN

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL



ESCALA

2:1

Nº DE PIEZA

7

HOJA

1/1

EJE DE TRANSMISIÓN SECUNDARIO MACIZO

## 10 Anexo II: Código del sistema

### FUNCIONES

```
1 import RPi.GPIO as GPIO
2 import threading
3 import datetime
4 import qrcode
5 import time
6 import cv2
7 import numpy as np
8
9 # __Motores verticales__
10 #Pines de dirección de los motores verticales
11 DIR_1 = 17
12 DIR_2 = 22
13 #Pines de paso de los motores verticales
14 STEP_1 = 27
15 STEP_2 = 23
16
17 # __Motor horizontal__
18 #Pin de dirección del motor horizontal
19 DIR_3 = 24
20 #Pin de paso del motor horizontal
21 STEP_3 = 25
22
23 GPIO.setmode(GPIO.BCM)
24
25 #Se configuran los pines de los motores como salidas
26 GPIO.setup(DIR_1,GPIO.OUT)
27 GPIO.setup(STEP_1,GPIO.OUT)
28 GPIO.setup(DIR_2,GPIO.OUT)
29 GPIO.setup(STEP_2,GPIO.OUT)
```

```

30  GPIO.setup(DIR_3,GPIO.OUT)
31  GPIO.setup(STEP_3,GPIO.OUT)
32
33  #Se configuran los pines de los actuadores lineales como salidas
34  GPIO.setup(26,GPIO.OUT) #Actuador 1
35  GPIO.setup(6,GPIO.OUT) #Actuador 2
36  GPIO.setup(5,GPIO.OUT) #Actuador 3
37  GPIO.setup(16,GPIO.OUT) #Actuador 4
38
39  delay = 0.001
40  steps_hor = 4780
41  steps_ver = 17320
42
43  def detectorQR(frame):
44
45      # Crear un detector de códigos QR
46      detector = cv2.QRCodeDetector()
47
48      # Detectar y decodificar el código QR
49      data, bbox, _ = detector.detectAndDecode(frame)
50
51      if bbox is not None:
52          bbox = np.int32(bbox[0])
53
54      return bbox,data
55
56  def take_photo(data,frame):
57      timestamp = datetime.datetime.now().strftime('%Y-%m-%d %H-%M-%S')
58      num_placa = data
59      filename = f'{num_placa}-{timestamp}.jpg'
60      cv2.imwrite(filename, frame)
61
62
63

```

```

64 #Función para el desplazamiento a izquierdas
65 def move_left(delay=delay, steps_hor=steps_hor):
66
67     #Función para mover los motores verticales en un hilo paralelo
68     def move_left_thread(delay=delay, steps_hor=steps_hor):
69         delay2 = delay*3.6
70         GPIO.output(DIR_1, GPIO.LOW)
71         GPIO.output(DIR_2, GPIO.HIGH)
72         for _ in range(int(steps_hor//3.6)):
73             GPIO.output(STEP_1, GPIO.HIGH)
74             GPIO.output(STEP_2, GPIO.HIGH)
75             time.sleep(delay2)
76             GPIO.output(STEP_1, GPIO.LOW)
77             GPIO.output(STEP_2, GPIO.LOW)
78             time.sleep(delay2)
79
80     #Iniciar la función anterior en otro hilo paralelo
81     hilo_left = threading.Thread(target=move_left_thread)
82     hilo_left.start()
83
84     #Mover motor horizontal
85     GPIO.output(DIR_3, GPIO.LOW)
86     for _ in range(steps_hor):
87         GPIO.output(STEP_3, GPIO.HIGH)
88         time.sleep(delay)
89         GPIO.output(STEP_3, GPIO.LOW)
90         time.sleep(delay)
91
92
93 #Función para el desplazamiento a derechas
94 def move_right(delay=delay, steps_hor=steps_hor, steps_ver=0):
95
96     #Función para mover los motores verticales en un hilo paralelo
97     def move_right_thread(dealy=delay, steps_hor=steps_hor):

```

```
98     delay2 = delay*3.6
99     GPIO.output(DIR_1, GPIO.HIGH)
100    GPIO.output(DIR_2, GPIO.LOW)
101    for _ in range(int(steps_hor//3.6)):
102        GPIO.output(STEP_1, GPIO.HIGH)
103        GPIO.output(STEP_2, GPIO.HIGH)
104        time.sleep(delay2)
105        GPIO.output(STEP_1, GPIO.LOW)
106        GPIO.output(STEP_2, GPIO.LOW)
107        time.sleep(delay2)
108
109    #Iniciar la función anterior en otro hilo paralelo
110    hilo_right = threading.Thread(target=move_right_thread)
111    hilo_right.start()
112
113    #Mover motor horizontal
114    GPIO.output(DIR_3, GPIO.HIGH)
115    for _ in range(steps_hor):
116        GPIO.output(STEP_3, GPIO.HIGH)
117        time.sleep(delay)
118        GPIO.output(STEP_3, GPIO.LOW)
119        time.sleep(delay)
120
121    #Función de movimiento ascendente
122    def move_up(delay=delay,steps_ver=steps_ver):
123        GPIO.output(DIR_1, GPIO.HIGH)
124        GPIO.output(DIR_2, GPIO.LOW)
125        for _ in range(steps_ver):
126            GPIO.output(STEP_1, GPIO.HIGH)
127            GPIO.output(STEP_2, GPIO.HIGH)
128            time.sleep(delay)
129            GPIO.output(STEP_1, GPIO.LOW)
130            GPIO.output(STEP_2, GPIO.LOW)
131            time.sleep(delay)
```



```
132
133 #Función de movimiento descendente
134 def move_down(delay=delay,steps_ver=steps_ver):
135     GPIO.output(DIR_1, GPIO.LOW)
136     GPIO.output(DIR_2, GPIO.HIGH)
137     for _ in range(steps_ver):
138         GPIO.output(STEP_1, GPIO.HIGH)
139         GPIO.output(STEP_2, GPIO.HIGH)
140         time.sleep(delay)
141         GPIO.output(STEP_1, GPIO.LOW)
142         GPIO.output(STEP_2, GPIO.LOW)
143         time.sleep(delay)
144
145 def motor1_up(delay=delay,steps=50):
146     GPIO.output(DIR_1, GPIO.HIGH)
147     for _ in range(steps):
148         GPIO.output(STEP_1, GPIO.HIGH)
149         time.sleep(delay)
150         GPIO.output(STEP_1, GPIO.LOW)
151         time.sleep(delay)
152
153
154 def motor2_up(delay=delay,steps=50):
155     GPIO.output(DIR_2, GPIO.LOW)
156     for _ in range(steps):
157         GPIO.output(STEP_2, GPIO.HIGH)
158         time.sleep(delay)
159         GPIO.output(STEP_2, GPIO.LOW)
160         time.sleep(delay)
161
162
163
164
165
```

```
166 def adxl_control():
167     axes = adxl345.getAxes(True)
168
169     if axes['x'] > 3.3:
170
171         #Se rotan los tambores 1 y 2
172         if axes['y'] > 3.3:
173             while axes['y'] > 0.4:
174                 motor1_up()
175
176         #Se rotan los tambores 3 y 4
177         elif axes['y'] < -3.3:
178             while axes['y'] < -0.4:
179                 motor2_up()
180
181     else:
182         pass
183
184     #Se rotan los tambores 2 y 4 activando los actuadores 1 y 3
185     GPIO.output(26,GPIO.LOW)
186     GPIO.output(5,GPIO.LOW)
187     time.sleep(0.5)
188     while axes['x'] > 0.4:
189         move_up(steps_ver=50)
190     GPIO.output(26,GPIO.HIGH)
191     GPIO.output(5,GPIO.HIGH)
192
193     elif axes['x'] < -3.3:
194
195         #Se rotan los tambores 1 y 2
196         if axes['y'] > 3.3:
197             while axes['y'] > 0.4:
198                 motor1_up()
199
```



```
200     #Se rotan los tambores 3 y 4
201     elif axes['y'] < -3.3:
202         while axes['y'] < -0.4:
203             motor2_up()
204
205     else:
206         pass
207
208     #Se rotan los tambores 1 y 3 activando los actuadores 2 y 4
209     GPIO.output(6,GPIO.LOW)
210     GPIO.output(16,GPIO.LOW)
211     time.sleep(0.5)
212     while axes['x'] < -0.4:
213         move_up(steps_ver=50)
214         GPIO.output(6,GPIO.HIGH)
215         GPIO.output(16,GPIO.HIGH)
216
217     #Si el eje X está nivelado
218     else:
219
220         #Se rotan los tambores 1 y 2
221         if axes['y'] > 3.3:
222             while axes['y'] > 0.4:
223                 motor1_up()
224
225         #Se rotan los tambores 3 y 4
226         elif axes['y'] < -3.3:
227             while axes['y'] < -0.4:
228                 motor2_up()
229
230         #Si los dos ejes están nivelados se continúa
231         else:
232             pass
```

## CÓDIGO PRINCIPAL

```
1  from Funciones import *
2  import schedule
3
4  margen = 60
5  camera = cv2.VideoCapture(0)
6
7  def main():
8
9      def ciclo(dire):
10
11         for _ in range(18): #Las 18 placas de cada balda
12
13             #Comprobación de la nivelación de la plataforma tras cada movimiento
14             adxl_control()
15
16             contador = 0
17             contador_inv = 0
18             contador_mov = 1
19             contador_increm = 0
20             while True:
21                 time.sleep(0.1)
22                 if cv2.waitKey(1) == ord('q'):
23                     break
24                 ret, frame = camera.read()
25                 bbox, data = detectorQR(frame)
26
27                 #Comprueba que el QR no tenga contenido vacío
28                 if bbox is not None and data != "":
29                     contador = contador + 1
30                     contador_inv = 0
31                     if contador == 5:
32                         contador = 0
```

```
33
34     cx1 = (bbox[0][0] + bbox[1][0]) / 2
35     cx2 = (bbox[2][0] + bbox[3][0]) / 2
36     cx = (cx1 + cx2) / 2
37
38     cy1 = (bbox[0][1] + bbox[3][1]) / 2
39     cy2 = (bbox[1][1] + bbox[2][1]) / 2
40     cy = (cy1 + cy2) / 2
41
42     #Comparación con el punto central del eje X de la imagen
43     if frame.shape[1]//2 - margen < cx < frame.shape[1]//2 + margen:
44
45         #Comparación con el punto central del eje Y de la imagen
46         if frame.shape[0]//2 - margen < cy < frame.shape[0]//2 + margen:
47             take_photo(data,frame)
48             time.sleep(2)
49             break
50
51         else:
52             #Si el centroide se encuentra por debajo del cuadrado
53             #Ya que en el eje Y el crecimiento es hacia abajo
54             if cy > frame.shape[0]//2 + margen:
55                 move_down(steps_ver=200)
56
57             #Si el centroide se encuentra por encima del cuadrado
58             #Ya que en el eje Y el crecimiento es hacia abajo
59             elif cy < frame.shape[0]//2 - margen:
60                 move_up(steps_ver=200)
61
62         else:
63             #Si el centroide se encuentra a la derecha del cuadrado
64             if cx > frame.shape[1]//2 + margen:
65                 move_right(steps_hor=200)
66
```

```

67         #Si el centroide se encuentra a la izquierda del cuadrado
68         elif cx < frame.shape[1]/2 - margen:
69             move_left(steps_hor=200)
70
71     else:
72         contador_inv = contador_inv + 1
73         contador = 0
74         if contador_inv == 5:
75             contador_inv = 0
76
77         if contador_mov == 1:
78             move_up(steps_ver=200 + 200*contador_increm)
79             contador_mov = 2
80
81         elif contador_mov == 2:
82             move_left(steps_hor=200 + 200*contador_increm)
83             contador_mov = 3
84
85         elif contador_mov == 3:
86             move_down(steps_ver=400 + 200*contador_increm)
87             contador_mov = 4
88
89         elif contador_mov == 4:
90             move_right(steps_hor=400 + 200*contador_increm)
91             contador_mov = 1
92
93         #Se incrementa la distancia para las siguientes iteraciones
94         contador_increm = contador_increm + 1
95
96
97     if dire == "left":
98         move_left()
99     else:
100        move_right()

```

```
101
102
103 #Reinicio inicial para asegurar que la plataforma se encuentre en el punto inicial
104
105 #La plataforma se asciende hasta detectar el imán ubicada en esta
106 while GPIO.input(8) == 1:
107     move_up(steps_ver=100)
108
109 #La plataforma se desciende un poco hasta dejar de detectar el imán
110 while GPIO.input(8) == 0:
111     move_down(steps_ver=100)
112
113 #La plataforma se desplaza a la derecha hasta detectar el imán del carro portador
114 while GPIO.input(7) == 1:
115     if GPIO.input(8) == 0:
116         while GPIO.input(8) == 0:
117             move_down(steps_ver=100)
118     else:
119         move_right(steps_hor=100)
120
121 #La plataforma se desplaza un poco a la izquierda hasta dejar de detectar el imán
122 while GPIO.input(7) == 0:
123     move_left(steps_hor=100)
124
125 #Se confirma de nuevo que la plataforma se encuentre en el punto límite superior
126 while GPIO.input(8) == 1:
127     move_up(steps_ver=100)
128 while GPIO.input(8) == 0:
129     move_down(steps_ver=100)
130
131 #Desplazamiento inicial para situarse frente a la primera placa
132 move_left(steps_hor=4500)
133 move_down(steps_ver=10000)
134
```

```
135     #Proceso principal
136     ciclo(dire="left")
137     move_down()
138     ciclo(dire="right")
139     move_down()
140     ciclo(dire="left")
141     move_down()
142     ciclo(dire="right")
143
144     #Se retorna la plataforma a su posición inicial para despejar la zona de trabajo
145     while GPIO.input(8) == 1:
146         move_up(steps_ver=100)
147     while GPIO.input(8) == 0:
148         move_down(steps_ver=100)
149     while GPIO.input(7) == 1:
150         move_right(steps_hor=100)
151     while GPIO.input(7) == 0:
152         move_left(steps_hor=100)
153
154     schedule.every().day.at("00:00").do(main)
155     schedule.every().day.at("06:00").do(main)
156     schedule.every().day.at("12:00").do(main)
157     schedule.every().day.at("18:00").do(main)
158
159     try:
160         while True:
161             schedule.run_pending()
162             time.sleep(30)
163
164     finally:
165         GPIO.cleanup()
166         camera.release()
167         cv2.destroyAllWindows()
168         schedule.clear()
```