

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y  
AUTOMÁTICA INDUSTRIAL



Biblioteca

DISEÑO MECÁNICO Y ELECTRÓNICO  
PARA LA MOTORIZACIÓN E IMPLEMENTACIÓN  
DE UNA HERRAMIENTA LAPAROSCÓPICA  
EN UN ROBOT COLABORATIVO

TRABAJO FIN DE GRADO

Julio - 2024

**AUTORA:** Angelina Zvereva

**DIRECTOR:** Jose María Sabater Navarro

**TUTORA:** Juliana Manrique Cordoba



## Resumen

En este proyecto, se ha llevado a cabo el diseño mecánico y electrónico de una herramienta laparoscópica motorizada, diseñada para su integración en un robot colaborativo. El proceso de diseño incluyó la creación de la placa electrónica utilizando KiCad, el revelado y la soldadura de los componentes electrónicos. Además, se realizaron los diseños mecánicos con Inventor AutoCAD, los cuales se materializaron mediante impresión 3D y corte en metacrilato. El código que hace funcionar la motorización de la herramienta fue desarrollado tanto en Arduino como en Python.

El principal objetivo de este proyecto fue el aprendizaje en varias áreas clave de la ingeniería. En primer lugar, se profundizó en el diseño y la fabricación de circuitos electrónicos, desde la creación de la PCB hasta la soldadura y el montaje de los componentes. En segundo lugar, se adquirieron habilidades avanzadas en diseño mecánico, incluyendo el modelado 3D y la fabricación de prototipos mediante impresión 3D y corte en metacrilato. También, se desarrollaron competencias en programación y control de sistemas electrónicos, implementando código tanto en Arduino para el control de hardware como en Python para la integración y automatización de la herramienta.

Este proyecto no solo refuerza los conocimientos adquiridos durante el grado en ingeniería, sino que también promueve el desarrollo de habilidades prácticas en el diseño y la implementación de tecnologías avanzadas. Además, fomenta la capacidad de resolver problemas complejos, integrar diversas disciplinas y utilizar herramientas tecnológicas modernas, lo cual es crucial para la innovación en el campo de la cirugía mínimamente invasiva.

### Palabras clave:

Diseño, desarrollo, herramienta laparoscópica, EndoStitch, programación, Arduino, Python, Inventor AutoCAD, diseño 3D.



# Índice general

Índice de figuras . . . . .	6
<b>1. Introducción</b>	<b>8</b>
1.1. Presentación del proyecto . . . . .	8
1.2. Objetivos del trabajo . . . . .	9
1.3. Metodología utilizada . . . . .	9
1.4. Descripción de tecnologías colaborativas . . . . .	10
1.5. Análisis de herramientas similares y desarrollos existentes . . . . .	11
<b>2. Marco Teórico</b>	<b>14</b>
2.1. Principios de la laparoscopia y la necesidad de herramientas motorizadas . . . . .	14
2.2. Conceptos clave de robótica colaborativa . . . . .	16
2.3. Descripción detallada de la herramienta Endo Stitch . . . . .	17
2.4. Fundamentos teóricos de la programación en Arduino y Python . . . . .	18
2.4.1. Arduino . . . . .	19
2.4.2. Python . . . . .	20
2.4.3. Integración entre Arduino y Python . . . . .	20
2.5. Revisión de otras herramientas . . . . .	21
<b>3. Diseño Mecánico</b>	<b>22</b>
3.1. Especificaciones y requisitos de diseño . . . . .	22
3.2. Modelado 3D de la herramienta laparoscópica . . . . .	22
3.2.1. Caja rodamiento . . . . .	23
3.2.2. Soporte motor paso a paso . . . . .	24
3.2.3. Soporte cilindro . . . . .	25
3.2.4. Mariposa . . . . .	25

3.2.5. Soporte palanca . . . . .	26
3.2.6. Base . . . . .	26
3.2.7. Tapa cilindro . . . . .	27
3.2.8. Tapa palanca . . . . .	27
3.2.9. Enganche robot . . . . .	28
3.3. Análisis de prototipos y mejoras implementadas . . . . .	29
<b>4. Diseño Electrónico</b>	<b>32</b>
4.1. Desarrollo del circuito en KiCad . . . . .	33
4.2. Impresión de la PCB (Placa de Circuito Impreso) . . . . .	35
4.3. Integración del sistema electrónico con la herramienta . . . . .	37
<b>5. Implementación y Programación</b>	<b>40</b>
5.1. Descripción del proceso de programación en Arduino y Python . . . . .	41
5.1.1. Python . . . . .	41
5.1.2. Teensy . . . . .	42
5.2. Integración del software con el hardware . . . . .	44
5.3. Pruebas y ajustes del sistema . . . . .	44
5.3.1. Resultado final . . . . .	46
<b>6. Conclusiones</b>	<b>48</b>
6.1. Presentación de los resultados obtenidos . . . . .	48
6.2. Discusión sobre las limitaciones y posibles mejoras . . . . .	49
6.2.1. Limitaciones . . . . .	49
6.2.2. Posibles mejoras . . . . .	49
6.3. Resumen y reflexión de los logros alcanzados . . . . .	49
<b>7. Bibliografía</b>	<b>51</b>
<b>8. Anexos</b>	<b>55</b>
8.1. Código Teensy . . . . .	55
8.2. Código Python . . . . .	58
8.3. Datasheets . . . . .	62
8.4. Planos . . . . .	67

# Índice de figuras

1.1. Herramientas Endowrist del robot DaVinci	11
2.1. Laparoscopia abdominal	15
2.2. Herramienta Endo Stitch	17
2.3. Placas de Arduino	19
3.1. Herramienta final	23
3.2. Caja rodamiento	24
3.3. Soporte motor paso a paso	24
3.4. Soporte cilindro	25
3.5. Mariposa	25
3.6. Soporte palanca	26
3.7. Base	27
3.8. Tapa cilindro	27
3.9. Tapa palanca	28
3.10. Enganche robot	28
3.11. Extremo del brazo robótico	29
3.12. Impresora 3D MakerBot imprimiendo una pieza	30
3.13. Dos piezas de metacrilato listas para ser pegadas	31
4.1. Teensy 2.0	32
4.2. Diseño inicial de las conexiones	33
4.3. PCB con Kicad	34
4.4. Esquemático	34
4.5. Plancha, diseño en acetato y placa de cobre	36
4.6. Placa terminada	37

4.7. Placa soldada	38
4.8. Conexiones de la placa con resto de componentes	39
4.9. Herramienta montada y conectada	39
5.1. Diseño de la interfaz y las condiciones planteadas	40
5.2. GIU final	46
5.3. Movimiento de la aguja en la herramienta	47
5.4. Movimiento de la palanca de la herramienta	47





# Capítulo 1

## Introducción

En este capítulo, se expone el estado de la cirugía mínimamente invasiva robotizada en la actualidad, analizando qué recursos y herramientas disponibles para este tipo de intervenciones.



### 1.1. Presentación del proyecto

En las últimas décadas, hemos presenciado notables avances tecnológicos, algunos de los cuales han tenido un impacto significativo en la medicina y la cirugía. El objetivo principal siempre ha sido que estas intervenciones sean lo menos invasivas posibles para los pacientes, lo que conlleva a un tiempo de recuperación mínimo. Aquí es donde la ciencia de la medicina y la ingeniería se unen para impulsar el progreso y desarrollar herramientas que pueden tener un impacto significativo en nuestras vidas.

Este trabajo se centra en explorar cómo la tecnología transforma la cirugía y la mejora, especialmente a través de la robótica quirúrgica. Se examinarán las técnicas e instrumentos innovadores que están llevando las intervenciones quirúrgicas a un nuevo nivel, con un enfoque particular en la cirugía mínimamente invasiva. Entender y aprovechar estas innovaciones puede tener un impacto significativo en la mejora de los procedimientos quirúrgicos y la calidad de vida de los pacientes, como es su pronta recuperación.

## 1.2. Objetivos del trabajo

El propósito de este trabajo se centra en la integración de una herramienta comercial de sutura laparoscópica en un robot colaborativo de 6 grados de libertad, con la finalidad de controlar la herramienta desde un ordenador.

Actualmente, la herramienta con la cual se ha estado desarrollando el proyecto, ha de ser manipulada por un cirujano especializado, utilizando una única mano para accionar dicha herramienta y con la otra mano se sujetan unas pinzas quirúrgicas que ofrecen de ayuda a la hora de poder sujetar la otra parte de la zona a tratar. Esto supone una dificultad para el cirujano, además de que hace falta mucha práctica para manejar la herramienta, lo cual se transforma en que el usuario se ha de formar durante un tiempo muy prologado. Por lo tanto, la idea de poder crear un soporte para la herramienta que se pueda acoplar a un robot, hará que se disminuyan los posibles errores humanos y se pueda conseguir automatizar el proceso.

## 1.3. Metodología utilizada

La fase inicial de un proyecto es la investigación, donde se analiza y se investiga métodos actuales, con la idea de poder mejorar ciertos aspectos. Para ello hizo falta analizar herramientas ya existentes en el mercado, especialmente el Endostitch de Covidien que es la herramienta utilizada, para comprender su funcionamiento y características de la misma.

Por otro lado, también hizo falta investigar acerca de las herramientas más utilizadas en la actualidad como por ejemplo el Arduino, Inventor, Kicad y Python para determinar su idoneidad para el proyecto. Al igual que identificar cuales serían los requisitos a la hora de diseñar el soporte de la herramienta y la elección de los distintos motores, estableciendo especificaciones técnicas para el diseño mecánico y electrónico.

Para el diseño mecánico de la herramienta laparoscópica se utilizó el software de diseño Inventor, para poder diseñar modelos detallados en 3D tanto de la herramienta, como de

las piezas para su ajuste al robot. Teniendo en mente, que la pieza final no fuera excesivamente grande.

Para el diseño electrónico, se optó por la realización de un diseño de una PCB (Placa de Circuito Impreso) para la integración del circuito electrónico con la placa de Arduino. Y para programar el microcontrolador de Arduino, se utilizó el lenguaje de C++ para controlar los motores. Para desarrollar el script y la interfaz se utilizó Python, junto con la librería de Kicad para crear una interfaz que conecte el robot colaborativo con el usuario.

Finalmente, se construyó un prototipo funcional para la herramienta motorizada utilizando los diseños electrónicos y mecánicos planteados. Y se realizaron pruebas para verificar su funcionamiento y el cumplimiento con las especificaciones establecidas.



## 1.4. Descripción de tecnologías colaborativas

La evolución en estas décadas está marcando muchos cambios significativos en el sector industrial, cada día más entidades y empresas se apoyan en las nuevas tecnologías como pueden ser *Big Data*, *Machine Learning*, digitalización de procesos o los robots colaborativos. Con todas estas herramientas, se intenta aumentar el control de nuestro entorno y optimizar muchos de los procesos que se llevan a cabo. Este proyecto tiene su enfoque en los robots colaborativos.

Los robots colaborativos son sistemas capaces de trabajar junto a un humano sin necesidad de incorporar barreras u obstáculos para proteger al usuario del robot. Y se caracterizan por ser sencillos a la hora de programar y/o controlar. Muchos de los robots colaborativos implementan sensores o cámaras que pueden evitar mediante la programación la colisión con un obstáculo, ya sea estático o móvil.

## 1.5. Análisis de herramientas similares y desarrollos existentes

El robot da Vinci(5), fue uno de los robots pioneros en el campo de la robótica quirúrgica, y a su vez es el más utilizado, y ha promovido la evolución de la cirugía mínimamente invasiva, aprobado desde el año 2000 y ha supuesto un antes y un después. Con este robot, se realizan cirugías cardíacas, pediátricas y torácicas entre otros. Este robot, cuenta con una visión 3D de alta definición y es manejado directamente por el cirujano, que acciona las articulaciones del robot, y las herramientas Endowrist.



Figura 1.1: Herramientas Endowrist del robot DaVinci

Este robot de 7 grados de libertad, y está equipado con cuatro brazos con herramientas, y estas se introducen dentro del paciente, y simulan el movimiento de muñeca y dedos que haría un cirujano durante la cirugía. El cirujano recibe la información de lo que sucede en tiempo real gracias a una cámara laparoscópica, y las ordenes se ejecutan a su vez, a tiempo real. A su vez, el cirujano tiene a disposición unos pedales con los cuales controla las herramientas de manera fluida.

Otros robots que se utilizan en la actualidad aparte del da Vinci, como por ejemplo:

- **Senhance Surgical Robotic System.** (3) Robot que se maneja a través del movimiento de los ojos del cirujano, así ofreciendo una interfaz de control muy innovadora. Esta tecnología puede mejorar la precisión y reducir la fatiga.
- **Flex Robotics.** (4) Un robot que se mueve como una serpiente, capaz de adaptarse a la curvatura del cuerpo humano, y así minimizar el daño provocado al paciente.

Su flexibilidad lo hace ideal para intervenciones en áreas anatómicamente complejas o de difícil acceso.

- **Medtronic Hugo.** (29) Es un robot quirúrgico modular y adaptable que utiliza instrumentos laparoscópicos estándar. Ofrece visualización en 3D de alta definición y capacidades de análisis de datos en tiempo real, mejorando la precisión y control en las cirugías. Su diseño flexible permite su uso en diversas especialidades quirúrgicas, reduciendo costos operativos a largo plazo.
- **Versius de CMR Surgical.** (30) Un robot quirúrgico con diseño compacto y modular que simula los movimientos del brazo humano, proporcionando una gran precisión y facilidad de uso. Su interfaz es intuitiva y su capacidad de adaptación a diferentes configuraciones quirúrgicas permite que sea una cirugía menos invasiva y además una recuperación más rápida para los pacientes.
- **BROCA, el primer Proyecto Español de Robot Quirúrgico.** (15) Un robot similar al da Vinci pero con un diseño modular, y que además incorpora la sensación del tacto, es más pequeño, y a su vez consigue abaratar costes. La capacidad de percibir la fuerza y la textura a través de los instrumentos, mejora significativamente la precisión de la operación como la seguridad del paciente durante la cirugía.

Estas herramientas presentan numerosos beneficios sin embargo, al mismo tiempo, se enfrentan a varios retos importantes. En primer lugar, muchos robots son voluminosos y muy costosos, lo que significa que la mayoría de los hospitales no tienen la capacidad de adquirirlos. En cuanto al tamaño de los robots, podrían hacerse más compactos y modulares, ocupando así un pequeño espacio del hospital. Al mejorar los diversos componentes electrónicos y mecánicos, los robots serían más pequeños.

En segundo lugar, las considerables inversiones en la compra, mantenimiento y almacenamiento de sistemas disuaden a los hospitales de obtener dicha tecnología, lo que limita esta tecnología a los hospitales con mejores recursos. La importancia de los robots quirúrgicos consiste en reducir costes sin perder calidad. Esto abarca desde la adquisición de robots hasta el mantenimiento y la actualización de software rentable.

Por otro lado, la facilidad de su uso. Las interfaces de sitio quirúrgico más intuitivas reducirán drásticamente el tiempo de entrenamiento. La integración de tecnologías de realidad aumentada e inteligencia artificial puede ayudar en operaciones en tiempo real, aumentando así la precisión y dando mejores decisiones.

El resultado deseado, es un robot más accesible con manipuladores quirúrgicos que son fáciles de manejar y almacenar a un costo reducido para permitir que más hospitales utilicen la atención médica de mejor calidad proporcionada por el uso de robots quirúrgicos. Esto haría que todo el proceso fuera más seguro, al tiempo que permitiría una menor intrusión en los pacientes que enfrentan un momento difícil con la mayoría de las cirugías abiertas.



# Capítulo 2

## Marco Teórico

### 2.1. Principios de la laparoscopia y la necesidad de herramientas motorizadas

La laparoscopia (14) es el proceso de realizar una cirugía dentro del abdomen o la pelvis del cuerpo humano mediante el uso de unas pequeñas incisiones, que generalmente son de medio a un centímetro y medio; por lo tanto, también se conoce como una cirugía mínimamente invasiva. El proceso se contrasta con la cirugía abierta en la que se utilizan grandes cortes en el paciente para realizar la cirugía. La laparoscopia se usa en las cirugías de apendicectomías, colecistectomías y algunos procedimientos quirúrgicos ginecológicos y urológicos.

Los principios básicos de la laparoscopia incluyen los siguientes pasos:

1. Incisión: En lugar de una incisión grande en el abdomen o la pelvis, se hacen varias incisiones pequeñas, de alrededor de 0,5 a 1,5 cm, a través de las cuales se insertan los instrumentos laparoscópicos: pinzas, tijeras, electrocauterios, endoscopios
2. Uso de un laparoscopio: Este dispositivo se inserta en una de las incisiones y transmite constantemente imágenes a un monitor. Es el proceso a través del cual el cirujano puede ver el interior del cuerpo sin tener que hacer una gran incisión.
3. Insuflación de gas: Como resultado, el gas de dióxido de carbono se canaliza para aumentar la visibilidad de todo el proceso. Esto separa las paredes del abdomen de

los órganos y, por lo tanto, se crea un espacio a través del cual el cirujano obtiene espacio para operar.

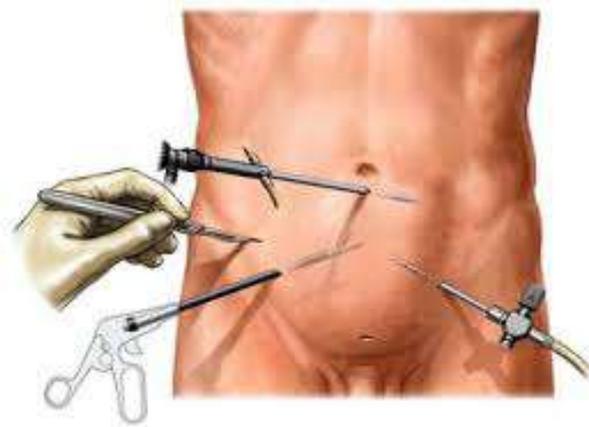


Figura 2.1: Laparoscopia abdominal

Aunque la laparoscopia ha revolucionado, la idea de reducir el trauma y acelerar la recuperación de un paciente, todavía tiene algunos defectos que las herramientas motorizadas están ayudando a superar.

Las herramientas laparoscópicas clásicas son operadas por el cirujano, una característica que no garantiza mucha precisión ni control, especialmente cuando existen casos difíciles y complicaciones durante la cirugía. Las herramientas motorizadas permiten movimientos mucho más diminutos y precisos, mejorando así la precisión, ya que estas además pueden ser diseñadas para imitar los movimientos naturales de la mano humana, permitiendo un acceso más fácil y eficiente a zonas difíciles.

Por otro lado, las operaciones en laparoscopia son un proceso largo y exhaustivo para el cirujano, en la que deben mantener una posición incómoda durante mucho tiempo. Las herramientas motorizadas pueden proporcionar beneficios ergonómicos adecuados y a su vez, reducir la fatiga física al automatizar movimientos repetitivos.

Con la incorporación de cámaras de alta definición y capacidades de imágenes avanzadas, las herramientas motorizadas mejoran la visibilidad dentro del cuerpo que se está evaluando, permitiendo así una mejor ejecución de los procedimientos que se están realizando.

Los sistemas robóticos, como por ejemplo, el robot da Vinci, trabajan con herramientas motorizadas bajo el control de un cirujano por una consola. Este tipo de sistema ofrece una precisión es muy alta, junto con estabilidad y mucha más calidad en los movimientos, de esta manera permitiendo la realización de cirugías con una mayor seguridad y eficacia.

La laparoscopia ha evolucionado enormemente desde su invención y ha transformado la forma en la que se realizan muchas de las cirugías hoy en día. La necesidad de herramientas motorizadas surge el hecho de que falta precisión, control y ergonomía en los instrumentos manuales. Estas innovaciones están mejorando los resultados quirúrgicos y la experiencia del paciente, marcando un importante progreso en la medicina moderna.



## 2.2. Conceptos clave de robótica colaborativa

La seguridad es clave en la robótica colaborativa. Estos robots están diseñados para detectar la presencia de los humanos y responder de manera segura evitando colisiones o daños hacia la persona. Ya que estos están diseñados para interactuar de manera natural con los seres humanos. Algunos de ellos pueden llegar a estar programados para que entiendan comandos por voz, o gestos que mejora y facilita la comunicación entre humano y robot.

Los robots colaborativos son muy flexibles a la hora de poder adaptarlos a distintas tareas o entornos. Pueden ser programados y reconfigurados por especialistas para que desempeñen una nueva tarea o adaptarse a cambios en la tarea actual. Esta capacidad de adaptación los convierte en valiosos recursos para entornos de fabricación y logística. Ya que además estos ayudan a mejorar la productividad, la eficiencia y hasta pueden liberar a las personas de realizar tareas muy repetitivas o peligrosas para la salud.

## 2.3. Descripción detallada de la herramienta Endo Stitch

La herramienta que se utiliza en este trabajo es el Endo Stitch de la marca Covidien, Medtronic(11) (17) (20). Este es un dispositivo quirúrgico diseñado especialmente para realizar suturas en cirugías laparoscópicas y garantizar que esta sea mínimamente invasiva para el paciente. Esta herramienta facilita mucho el labor del cirujano para realizar suturas en partes del cuerpo donde su acceso es muy limitado y requiere de una gran precisión.



(a) Herramienta

(b) Vista del mango

Figura 2.2: Herramienta Endo Stitch

El Endo Stitch es una herramienta muy versátil ya que es adecuada para una gran variedad de procedimientos laparoscópicos, que pueden ser del ámbito ginecológico; utilizando procedimientos como histerectomías y reparaciones de prolapso donde la precisión y la invasión mínima son cruciales. Y a su vez gastrointestinal y urológico; ya que facilita la sutura de anastomosis y la reparación de órganos internos mejorando su eficacia. Esta diseñada de tal manera, que tiene un mango muy ergonómico que permite el control de manera precisa por parte del cirujano, reduciendo su fatiga durante intervenciones muy largas. La herramienta está diseñada para que pueda ser utilizada con una sola mano,

dejando en libertad la otra mano del cirujano para la utilización de otras herramientas necesarias durante la intervención. Esto es muy útil en intervenciones donde el espacio de maniobra es muy reducido.

La herramienta tiene unos cartuchos donde se encuentra la aguja y el hilo para realizar las suturas, y estos cartuchos son reemplazables, cosa que mejora la eficiencia y reduce el tiempo en la intervención. El Endo Stitch utiliza un sistema de sutura que es bidireccional, que permite al cirujano realizar suturas sin tener que interrumpirlas. Ya que utilizando la pequeña palanca azul que se encuentra en el mango, como se puede observar en la figura [2.2b](#), la cual puede mover la aguja y anclarla en un lado u otro para asegurar que la sutura se realiza con precisión y rapidez.

El Endo Stitch supone muchos beneficios clínicos, entre los cuales destaca una mayor precisión a la hora de realizar suturas en espacios limitados, lo que minimiza notablemente los posibles errores y complicaciones en el postoperatorio. Por otro lado, lo que se describió antes acerca de los cartuchos de sutura reemplazables, afecta a la eficiencia y la reducción del tiempo en el quirófano. Esto, no solo beneficia al paciente, sino que además al reducir el tiempo en quirófano se optimiza su uso y se pueden realizar más intervenciones durante el día si eso es necesario. En cuanto al paciente, esta herramienta ayuda a reducir el trauma quirúrgico y acelerar la recuperación del paciente, ya que los tejidos que se han de curar son incisiones muy pequeñas. Las cuales contribuyen a menos dolor, y cicatrices más pequeñas para el paciente. No todo son ventajas para el paciente, el cirujano se beneficia de la simplificación del trabajo y de esa manera la reducción de la fatiga y el estrés que causan las intervenciones largas y complejas.

## 2.4. Fundamentos teóricos de la programación en Arduino y Python

En este proyecto he utilizado dos herramientas muy populares de la programación y la electrónica: Arduino y Python. Son herramientas que cada una tiene sus propias fortalezas que las hacen atractivas para utilizarlas en la creación de nuevos proyectos.

### 2.4.1. Arduino

Arduino (16) es una plataforma de hardware abierto basada hardware y software fáciles de usar que incorporan sensores y actuadores a través de los cuales es posible crear proyectos que pueden interactuar con el entorno físico. Arduino cuenta con una amplia biblioteca de programación que facilita la interacción entre los distintos componentes proporcionando funciones ya predefinidas.

El hardware del Arduino consiste en una serie de placas que se pueden adquirir, como Arduino Uno, Teensy, Mega, etc. Estas placas incluyen un microcontrolador y un conjunto de pines tanto de entrada como de salida que permiten la conexión de distintos dispositivos electrónicos a la placa. Estos dispositivos pueden ser sensores, luces LED y motores entre otros. Su entorno de desarrollo se basa en un lenguaje de programación similar a C++ y permite escribir diversos sketches, los programas que controlan la placa. Estos programas una vez creados se cargan en la placa de Arduino para su futura utilización.



Figura 2.3: Placas de Arduino

Con los pines que se encuentran en la placa, se puede lograr leer datos del entorno recogidos gracias a las entradas analógicas y digitales, como los sensores, y mediante las salidas, como LEDs o motores, responder a ellas.

### 2.4.2. Python

Python (21) es un lenguaje de programación de alto nivel que se caracteriza por su sintaxis sencilla y versátil que agrada a muchos usuarios.

La sintaxis de Python es clara y sencilla, lo cual facilita la escritura del código. Este cuenta con una extensa variedad de bibliotecas y frameworks que cubren casi la totalidad de las necesidades de los desarrolladores de la actualidad, desde análisis de datos, como NumPy o pandas, hasta inteligencia artificial, como PyTorch. Gracias a esto, Python es un lenguaje de programación muy versátil con el cual se pueden programar una gran variedad de aplicaciones con la utilización de scripts simples para crear endpoint HTTP o para la automatización.

Un endpoint HTTP es una URL que permite a un cliente interactuar con un servidor a través de peticiones, como podría ser un GET, un POST, etc. Es una parte fundamental a la hora de utilizar interfaces de programación de aplicaciones (API), ya que el endpoint define la operación que se puede realizar sobre él. Los métodos de un endpoint pueden ser GET, que recupera información, POST, que envía datos, PUT, que actualiza un recurso existente y DELETE, que elimina un recurso.

### 2.4.3. Integración entre Arduino y Python

En este proyecto se ha optado por la integración de ambas herramientas para así poder combinar las positivas características de cada uno. Como la capacidad de interactuar con el entorno exterior gracias al hardware de Arduino y la simplicidad de procesamiento que tiene Python para esos datos.

El Arduino puede enviar y recibir datos a través de USB con la integración de la comunicación Serial, y Python puede leer y escribir esos datos, utilizando la librería *pySerial* (28). Además, Python puede controlar el Arduino a través de scripts que se le envían mediante la comunicación serie. Otra ventaja de utilizar estas dos herramientas, es que en el caso de que fuera necesario, Python puede interactuar con servicios http y bases de datos, dando así la opción a recopilar y almacenar los datos recibidos a través del Arduino, y

posteriormente procesador o visualizados en una aplicación web. Estos datos, pueden ser procesados y analizados en Python utilizando las potentes bibliotecas de análisis de datos y machine learning.

## 2.5. Revisión de otras herramientas

A parte de las herramientas ya mencionadas en el apartado anterior, Arduino y Python, en el proyecto se han utilizado otras herramientas muy versátiles, como son el Autodesk Inventor y el KiCad.

El Autodesk Inventor es un software de diseño asistido que permite modelar diseños en 3D con gran detalle, y poder ensamblar distintas piezas creando objetos reales. El Inventor tiene una gran capacidad de crear modelos muy precisos y complejos con mucha facilidad, y esta herramienta es crucial a la hora de desarrollar nuevos productos o la simulación de diseños antes de lanzarlos a su fabricación. Además, permite realizar análisis de esfuerzo y simulaciones de movimiento si fuera necesario, característica que ayuda a prevenir errores en el diseño que pueden acarrear grandes pérdidas económicas. Y por último, cabe destacar la versatilidad a la hora de exportar los archivos, ya que soporta las extensiones para poder mandar dichos diseños a imprimir en 3D.

Y por otro lado tenemos el KiCad, un software libre que permite crear esquemas eléctricos y diseños de placas de circuito impreso (PCB) con mucha facilidad y precisión. Kicad ofrece las herramientas necesarias para poder convertir diseños de circuitos electrónicos en PCB físicas, gracias su software de diseño y la amplia variedad de bibliotecas de componentes electrónicos que tiene. Y por último y no menos importante, permite la simulación del circuito para verificar su correcto funcionamiento y evitar errores innecesarios antes de fabricar la placa.

# Capítulo 3

## Diseño Mecánico

### 3.1. Especificaciones y requisitos de diseño

La herramienta que se ha planteado en este proyecto está pensada para acoplarse a un robot de Universal Robots, por lo que el acople que se ha modelado ha sido con el propósito de que se enganche a su extremo. Por otro lado, se ha mantenido en mente la idea de hacer la herramienta lo mas ligera y simple posible para no sobrecargar el brazo robótico y que este siga siendo ágil y maniobrable.

Todos los diseños han sido modelados con el programa Inventor de AutoCAD, y posteriormente, algunas han sido impresas en una impresora 3D, MakerBot, y otras piezas han sido recortada sobre metacrilato.

### 3.2. Modelado 3D de la herramienta laparoscópica

En la siguiente figura [3.1](#), se puede ver el modelo en 3D de la herramienta final:

El producto final se ha modelado acorde a la herramienta EndoStitch a escala real, y consta de varias partes:

- La herramienta EndoStitch.
- La base sobre la cual van conectados todos los componentes
- La sujecion de la herramienta con el robot

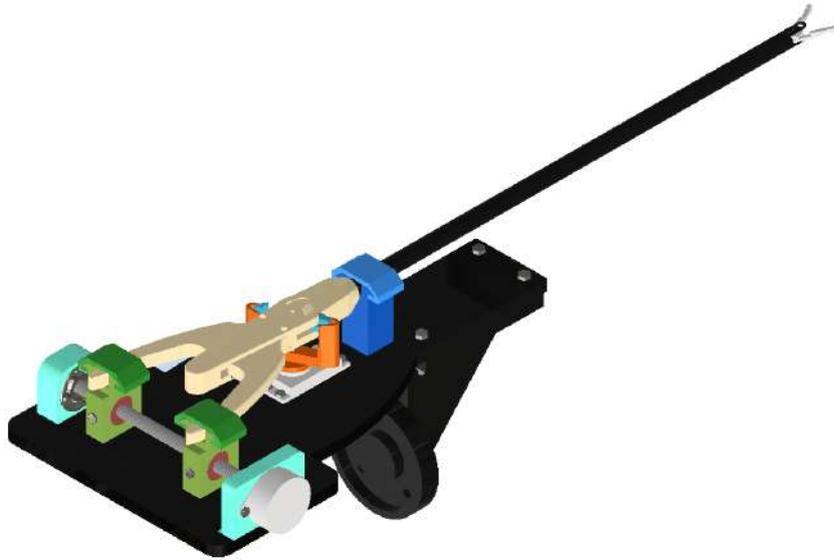


Figura 3.1: Herramienta final

- La caja que sujeta el rodamiento
- El soporte donde se sujeta el motor paso a paso
- Los dos motores, tanto el paso a paso como el servomotor
- La mariposa, que es lo que acciona el movimiento de la aguja en el EndoStitch
- Los dos soportes de las palancas del EndoStitch, que son las encargadas de sujetar las palancas que se encargan del pinzamiento de la herramienta
- El soporte que sujeta el cilindro de la herramienta
- La tapa que sujeta el soporte de la palanca
- La tapa que sujeta el soporte del cilindro para inmovilizar la herramienta
- Y por último, todos los componentes que ayudan al accionamiento y sujeción de las piezas; como son los tornillos y tuercas, el husillo, el rodamiento y la tuerca cilíndrica para el husillo

Todos los planos se pueden ver en los anexos.

### 3.2.1. Caja rodamiento

La caja rodamiento se ha diseñado con el propósito de contener el rodamiento con el cual girara el husillo. El agujero central es pasante debido a que el husillo llega a atravesar



Figura 3.2: Caja rodamiento

la pieza, y tiene tope para poder fijar mejor el rodamiento.

### 3.2.2. Soporte motor paso a paso



Figura 3.3: Soporte motor paso a paso

Este soporte es el encargado de sujetar el motor paso a paso en el lateral de la herramienta final. Tiene un escalón, debido a que el motor tiene que estar colocado a ras de la base, y para mayor sujeción y colocación de la pieza encima de la base se hizo un agujero a una distancia.

### 3.2.3. Soporte cilindro

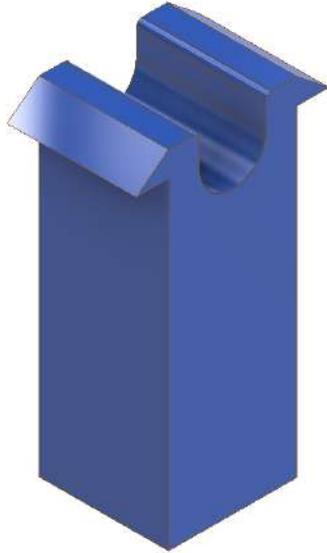


Figura 3.4: Soporte cilindro

Este soporte se encarga de inmovilizar el cilindro de la herramienta EndoStitch, para que en el momento en el que se accionan los motores y se abre y se cierra el extremo, evitar vibraciones que puedan entorpecer la cirugía.

### 3.2.4. Mariposa

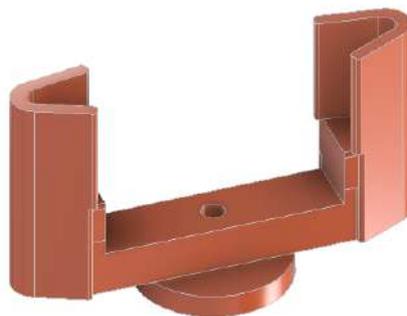


Figura 3.5: Mariposa

La mariposa es la encargada de hacer girar la pequeña palanca azul que se puede observar en la figura [2.2b](#). Esta pieza sufrió diversos cambios durante su creación, debido a que esta se localiza muy cerca del mango de la herramienta y la primera versión rozaba y no podía girar bien. Por esa razón, como se puede observar en la figura [3.5](#), hay un escalón para poder separar bien las piezas.

### 3.2.5. Soporte palanca

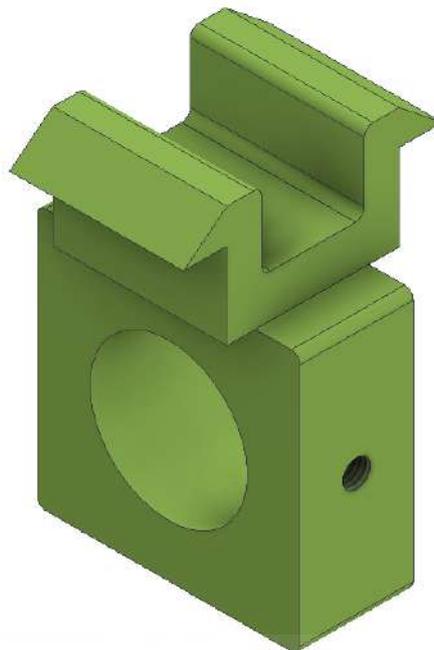


Figura 3.6: Soporte palanca

Este soporte tiene dos orificios pasantes, el grande es donde va una tuerca encargada de mover el husillo cilíndrico comprada en *Igus*. Esto es lo que permitirá que la pieza pueda realizar el movimiento de pinzamiento durante la cirugía. Y por otro lado, el otro orificio pasante más pequeño es para poder sujetar mejor la tuerca del interior y que no resbale cuando el husillo se mueva.

La zona superior de la pieza es donde irán colocadas e inmovilizadas las palancas que accionan el movimiento del pinzamiento. Además se ha modelado una tapa para poder sujetarlas mejor.

### 3.2.6. Base

Esta pieza es la que sujeta todos los componentes de la herramienta final. Tiene varios huecos, que es donde irán incrustadas el resto de piezas, los motores, los soportes de la herramienta, y en el extremo delantero el enganche de la pieza al robot. Al igual, que están incluidos los orificios de los tornillos para las piezas que irán atornilladas a la base, que son el enganche del robot, y el servomotor.



Figura 3.7: Base

### 3.2.7. Tapa cilindro

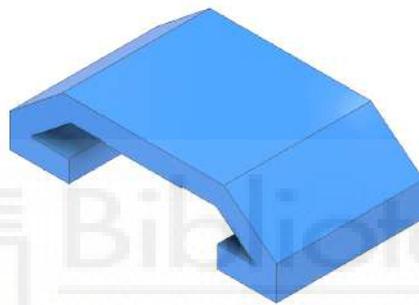


Figura 3.8: Tapa cilindro

De las dos tapas que hay en este proyecto, esta es la que sujeta el cilindro de la herramienta, que se engancha deslizándose sobre la pieza [3.4](#), se ha de insertar por uno de los lados y colocarlo centrado por encima del soporte. Tiene una característica distinta respecto a la tapa que se coloca encima del soporte de la palanca, y es que tiene un pequeño saliente que sirve para poder fijar mejor el cilindro para evitar que se mueva. Este saliente se puede observar en el plano 01.07 de los anexos.

### 3.2.8. Tapa palanca

Esta tapa es muy parecida a la que sujeta el cilindro exceptuando que no tiene el saliente que sujeta con más fuerza la pieza. Esta pieza lo que fija son las palancas que accionan el movimiento de pinzamiento en el EndoStitch, y se coloca deslizándose por el lateral, y se coloca sobre el soporte del cilindro para que esté bien alineado (figura [3.6](#)). Esta pieza no tiene ese saliente debido a que se ha considerado que no hace falta que esté tan restringido en el movimiento como en el cilindro, ya que al fin y al cabo la palanca si

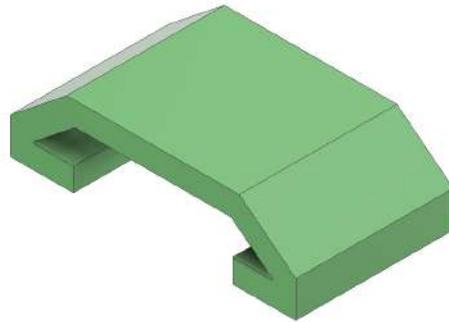


Figura 3.9: Tapa palanca

que se va a estar moviendo, y lo importante es que el extremo sea lo más fijo posible de cara a la operación quirúrgica y a la seguridad del paciente.

#### 3.2.9. Enganche robot



Figura 3.10: Enganche robot

Esta pieza como bien indica su nombre es la unión entre el conjunto de la herramienta ya ensamblada y el brazo robótico al que vamos a colocar dicha herramienta. La pieza consiste en una sujeción rectangular que va unida con tornillos a la base (figura 3.7), y una extrusión circular para poder atornillarse al extremo del brazo robótico, en la figura 3.11 se puede observar cómo es el diseño y la cantidad de tornillos que tiene para poder atornillarlo. Además cuenta con un nervio que une las dos zonas para poder distribuir mejor el peso de la herramienta total.



Figura 3.11: Extremo del brazo robótico

### 3.3. Análisis de prototipos y mejoras implementadas

Cómo ya se mencionó anteriormente, algunas de las piezas se hicieron con impresora 3D y otras en metacrilato.

Las piezas que se hicieron con la impresora 3D fueron las siguientes:

- Mariposa
- Soporte palanca
- Tapa cilindro
- Tapa palanca
- Enganche robot

Y las piezas que se cortaron en metacrilato fueron:

- Caja rodamiento
- Soporte motor paso a paso
- Soporte cilindro
- Base

Las piezas que se imprimieron con la impresora 3D tuvieron que pasar una prueba previa debido a que tenía que calcular el margen de error de la impresora y cómo iba a

reaccionar el filamento de ácido poliláctico (PLA) tras su enfriamiento. Por eso algunas de las piezas que se habían impreso quedaban muy justas para enganchar con otras, y la solución que se tomaba era imprimir de nuevo teniendo en cuenta la tolerancia. Por otro lado, habían ciertas piezas que para imprimirlas se requería imprimir una estructura para que sujete bien y se imprima bien, detalle que a veces entorpecía el deslizamiento de la tapa con el soporte y tocaba limar algunas de las piezas. A pesar de ser una forma muy versátil de crear piezas, esos fueron algunos de los problemas que me encontré a la hora de crearlas.

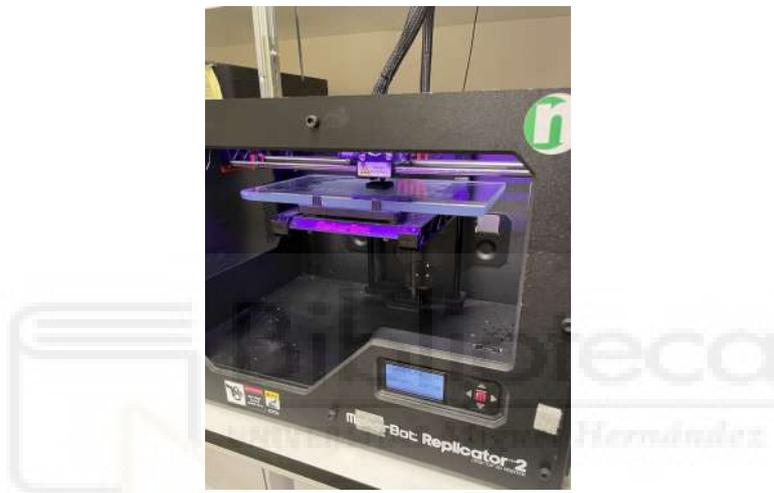


Figura 3.12: Impresora 3D MakerBot imprimiendo una pieza

En cuanto a las piezas de metacrilato, resultaron ser más precisas que las de impresión 3D, debido a que el margen de error es menor, no tiene tanta tolerancia como la impresora 3D y la materia no es tan impredecible como puede ser el PLA. Sin embargo, el metacrilato que había disponible era demasiado fino para la herramienta que había en mente inicialmente, por lo que la solución que opté fue cortar de cada pieza su pareja para luego poder juntarlas y así dar lugar una pieza más resistente. Las piezas se pegaron con *Loctite*. Figura [3.13](#)



Figura 3.13: Dos piezas de metacrilato listas para ser pegadas

# Capítulo 4

## Diseño Electrónico

En este capítulo se hablará acerca del proceso que se ha llevado a cabo para crear la placa de circuito impreso, PCB. Inicialmente, teniendo en cuenta las necesidades del circuito y los componentes que iba a llevar, se ideó un diseño de las conexiones que harían falta, y donde debería ir colocado el Teensy y las entradas y salidas del resto de componentes.

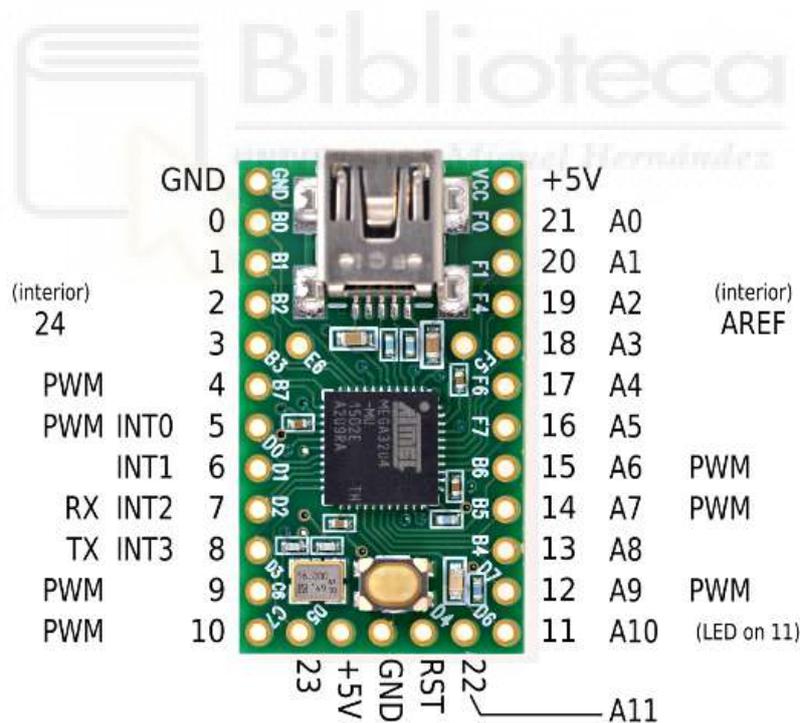


Figura 4.1: Teensy 2.0

Con la asignación de pines que se puede observar en la figura 4.1 y teniendo en mente, que necesitaba poder conectar dos motores y en caso de hiciera falta, una entrada y tierra para poder conectarlo a una alimentación externa, si no era suficiente con la proporcionada al Teensy con el cable.

Este diseño se hizo inicialmente a mano alzada para ver que conexiones había, y cuales era necesario conectar y a donde. Y el diseño quedó tal cual se puede observar en la figura

4.2

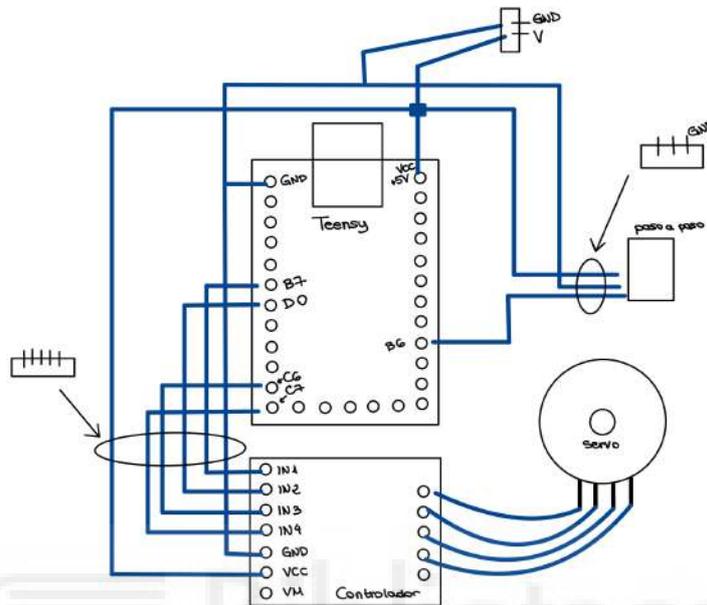


Figura 4.2: Diseño inicial de las conexiones

## 4.1. Desarrollo del circuito en KiCad

Para crear un diseño de un circuito en KiCad hay que crear un nuevo proyecto dentro del programa, donde este ya contendrá los archivos necesarios para diseñar el circuito.

Gracias a las herramientas que nos proporciona KiCad podemos colocar los componentes del circuito y moverlos a nuestro antojo. Una vez colocados los componentes, podemos con una herramienta que se llama *Place Wire* se traza las conexiones entre los pines y componentes colocados.

Una vez trazadas las conexiones se genera un *Netlist* para que se pueda crear luego el diseño de la PCB. Una vez hecho esto, y cargado el *Netlist* en el Pcbnew, y trazamos las rutas que van a seguir los componentes. Es muy importante dejar el espacio necesario entre los alambres de conexión para que cuando se haga la placa y se revele con ácido

haya menos posibilidad de error de que se hayan juntado. Una cosa que tuve en cuenta a la hora de diseñar la placa, es dejar unos huecos para poder taladrarlos con un taladro fresador y luego atornillarlo a la herramienta final.

Una vez hecho todo eso, el diseño de la placa quedó como se puede observar en la figura 4.3

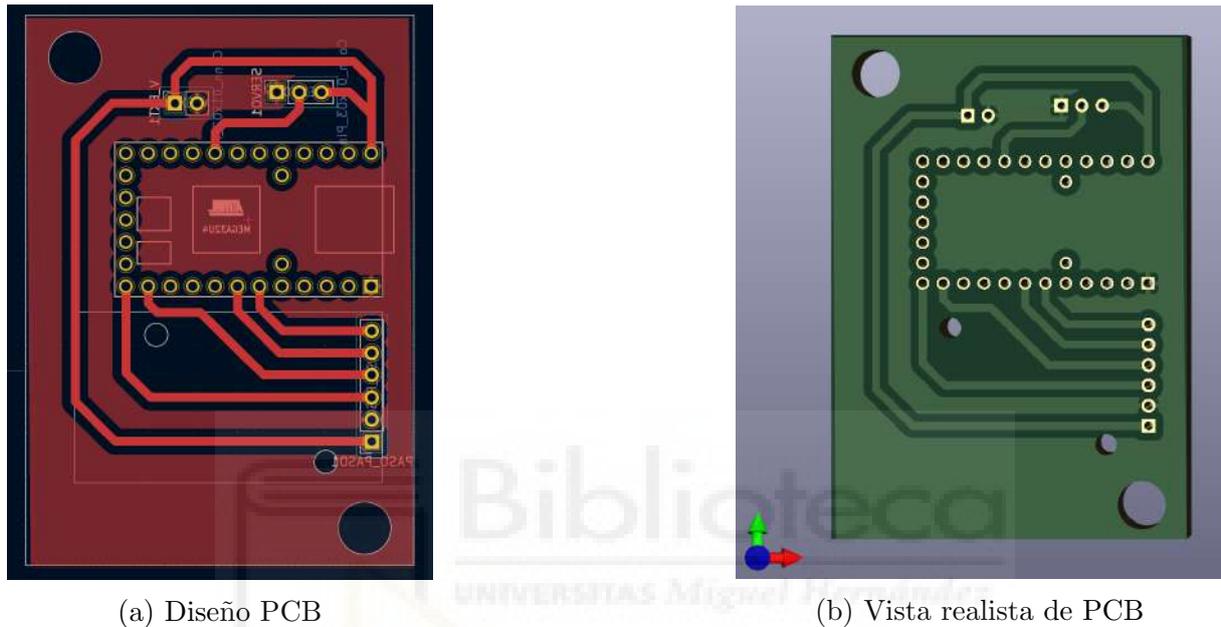


Figura 4.3: PCB con Kicad

Adjunto además el esquemático de la placa, figura 4.4, donde se pueden observar las conexiones entre el Arduino y los dos robots, y a su vez la posibilidad de conectarlo a una alimentación externa.

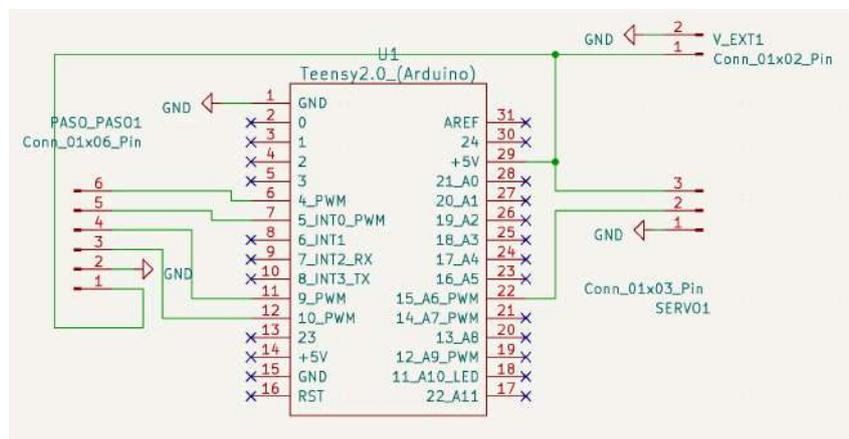


Figura 4.4: Esquemático

## 4.2. Impresión de la PCB (Placa de Circuito Impreso)

La impresión de una PCB con ácido es un método muy común pero que se deben tomar ciertas precauciones a la hora de realizarlo, ya que implica pasos como la preparación de la placa, el revelado, y el grabado con ácido clorhídrico y peróxido de hidrógeno. Los ingredientes que fueron utilizados en el proceso fueron los siguientes:

- Placa de cobre positiva
- Ácido clorhídrico, conocido comúnmente como sulfumán, con una concentración aproximada de 30
- Peróxido de hidrógeno, con una concentración de 30
- Revelador de placa positiva: Solución de hidróxido de sodio
- Impresora láser para poder imprimir el diseño del circuito en acetato
- Plancha, para transferir el diseño del circuito a la placa de cobre
- Guantes de seguridad
- Recipientes de plástico para mezclar y contener las soluciones químicas

El revelado de la placa se hizo en el laboratorio de tecnología electrónica de la Universidad Miguel Hernández.

Para la impresión de la PCB, hizo falta sacar una máscara del circuito. La máscara es una transparencia que tiene dibujado el circuito impreso de manera que lo que queda en color negro, es lo que quedará con pistas de cobre. Esta fase es muy importante, ya que tiene que ser de alta calidad para que los trazos se puedan ver nítidos.

Para transferir el diseño a la placa de cobre, lo primero que hay que hacer es limpiar bien la placa de cobre con alcohol para poder eliminar cualquier impureza que tenga. Luego colocar el acetato con el diseño de la placa sobre la placa de cobre y meterlo en una plancha para transferir el diseño. El calor hay que aplicarlo de manera uniforme y

durante unos minutos.



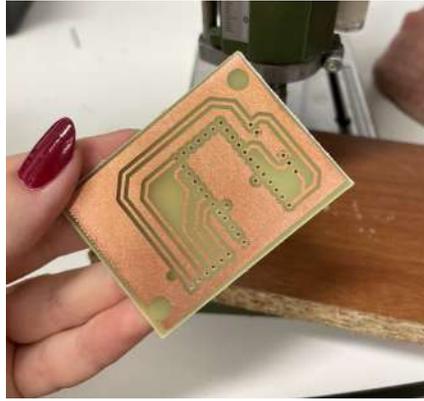
Figura 4.5: Plancha, diseño en acetato y placa de cobre

Una vez hecho los pasos preparativos, fue necesario disolver el revelador de la placa positiva, el hidróxido de sodio en agua. Se sumergió la placa en la solución e hizo falta agitar ligeramente para ayudar a que la capa fotosensible se disuelva, revelando las áreas de cobre que no estaban protegidas por el diseño que se transfirió con la placa. Una vez hecho eso, se enjuagó la placa con agua para eliminar los restos de revelador y frenar el proceso.

Luego se mezcló ácido clorhídrico con ácido de hidrógeno en una proporción de 2:1. Este paso fue necesario hacerlo bajo la extracción de la campana de gases ya que son sustancias tóxicas. Se sumergió la placa en la solución, y se esperó a que hiciera la reacción química que disolvió el resto del cobre que no estaba protegido. Una vez el cobre deseado se haya disuelto, se enjuagó nuevamente la placa.

Una vez terminado la revelación de la placa, con acetona se limpió el tóner de la placa que protegía las pistas de cobre, y se comprobó que las pistas estaban completas y no habían puentes no deseados.

Por último, con una taladro fresador se hicieron los pequeños agujeros para poder insertar los componentes para su futura soldadura, y también los dos agujeros para poder sujetar la placa a la base con unos tornillos.



(a) Placa revelada y taladrada



(b) Taladro fresadora

Figura 4.6: Placa terminada

### 4.3. Integración del sistema electrónico con la herramienta

Una vez fabricada la placa, había que soldar todos los componentes electrónicos usando las herramientas adecuadas. Los componentes necesarios para soldar en la placa:

- Placa Teensy
- Controlados para el motor paso a paso
- Pines macho de 2 enlaces

- Pines macho de 3 enlaces

Para realizar la soldadura, sólo fue necesario un soldador y filamento de estaño. Y como último paso, un multímetro para poder verificar que las conexiones funcionen correctamente.

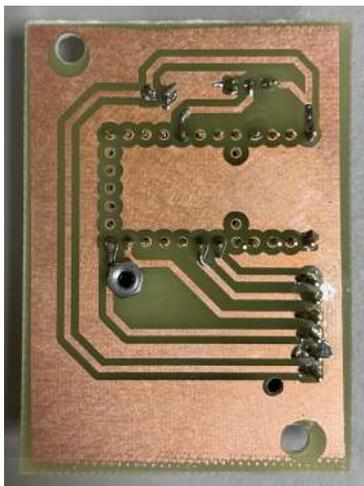


Figura 4.7: Placa soldada

Con todo soldado, se realiza la conexión de todos los componentes necesarios faltantes, como los motores y el USB que conecta el Arduino con el ordenador, y se hace una prueba para ver que todos los elementos funcionaban por separado y todos juntos. Hizo falta ponerle un pequeño cuadrado de gomaespuma debajo del controlador del motor, debido a que el componente tras su soldadura había quedado sólo sujeto por los pines, y el tornillo en el otro extremo y quedaba a una altura que podía dar problemas en el futuro, como rotura de la soldadura o que se dañase la placa. Y por otro lado, la tuerca que sujeta el controlador estaba muy cerca de algunas vías de cobre y hacía contacto, detalle que daba problemas a la hora de intentar hacer funcionar el circuito. Por lo que la solución fue cubrir la tuerca con laca transparente y además colocar una fina capa de gomaespuma adhesiva para que no haga ningún contacto innecesario.

Una vez soldado y comprobado todo, con las piezas ya impresas con la impresora 3D y las otras cortadas en metacrilato, se realiza el montaje de la herramienta.

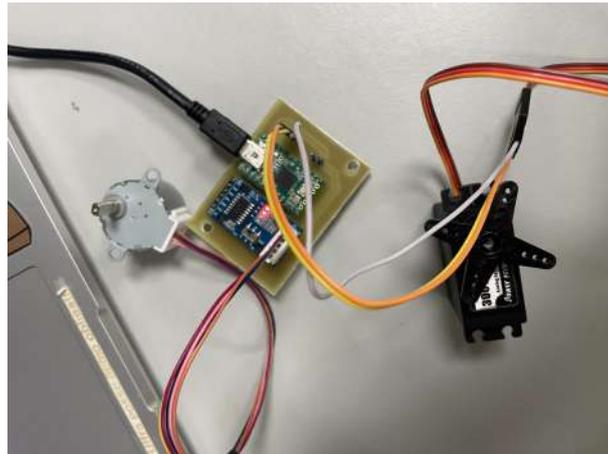


Figura 4.8: Conexiones de la placa con resto de componentes



Figura 4.9: Herramienta montada y conectada

# Capítulo 5

## Implementación y Programación

Durante el diseño de la interfaz, se tuvieron en cuenta varios diseños. Si los botones iban a estar en una zona u en otra, como se iba a visualizar las acciones realizadas o cuales iban a ser las condiciones para la correcta ejecución.

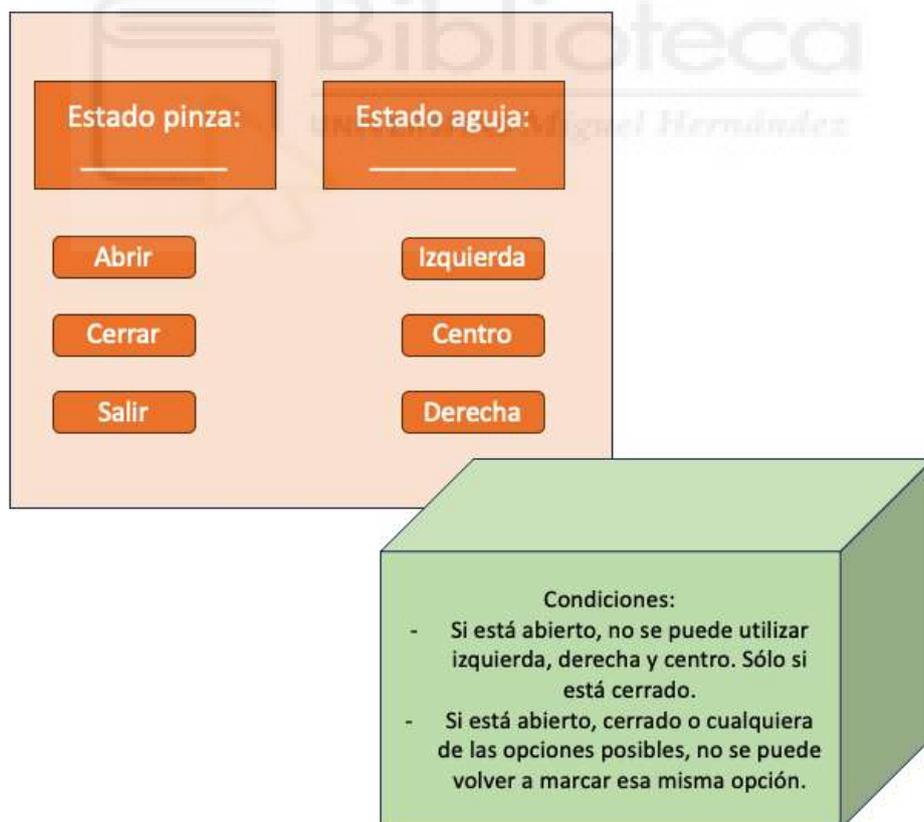


Figura 5.1: Diseño de la interfaz y las condiciones planteadas

Una vez pensado una idea general, era hora de programar en Python.

## 5.1. Descripción del proceso de programación en Arduino y Python

### 5.1.1. Python

En el código de Python, utilizó la biblioteca *tkinter* para poder crear una interfaz gráfica de usuario (GUI). La idea es que esta permita a un cirujano controlar la herramienta ya motorizada EndoStitch a través de comandos que se le envían al Teensy.

La configuración de la conexión serial que se establece con el Teensy es mediante el paquete *serial.Serial*.

```
1 import serial
2 import time
3 arduino = serial.Serial('/dev/cu.usbmodem123451', 9600)
4 time.sleep(2)
```

Por otro lado, la GUI incluye los botones para abrir, cerrar, y mover la aguja tanto para el lado izquierdo como para el derecho, y el botón centro que es el que permitirá cambiar de aguja. También hay un botón de reinicio para volver a la posición inicial. Cada botón está vinculado a una función que envía un comando específico al Teensy.

```
1 root = tk.Tk()
2 root.title("Control Endostitch")
3 root.config(width=400, height=300)
4 root.resizable(False, False)
5
6 # Botones
7 botonAbrir = ttk.Button(root, text="Abrir", style="Raton.TButton",
8   command=Abrir, state="normal")
9 botonAbrir.place(x=50, y=150)
```

Las funciones de *Abrir*, *Cerrar*, *Centro*, *Izquierda* y *Derecha* envían comandos específicos al Teensy para así controlar los motores. La función *Desactivar*, desactiva temporalmente los botones para evitar acciones simultáneas y de esa manera evitar posibles errores durante la intervención quirúrgica. Por otro lado la función *Activar*, los activa tras un intervalo de tiempo que he definido.

```
1 def Abrir():
2     arduino.write(b'1')
3     Desactivar()
4     write_msg_p("\nPinzas Abiertas\n\n\n", T)
```

Se definen botones en la GIU para que cuando estos sean pulsados llamen a la función que corresponda

```
1 botonAbrir = ttk.Button(root, text="Abrir", command=Abrir)
2 botonCerrar = ttk.Button(root, text="Cerrar", command=Cerrar)
3 botonCentro = ttk.Button(root, text="Centro", command=Centro)
4 botonIzq = ttk.Button(root, text="Izquierda", command=Izquierda)
5 botonDer = ttk.Button(root, text="Derecha", command=Derecha)
```

### 5.1.2. Teensy

El Teensy controla un servomotor y un motor paso a paso a través de los comandos recibidos desde la GIU de Python. Utilizando la biblioteca *Servo*, controla el servomotor y con la biblioteca *Stepper* maneja el motor paso a paso.

El código se estructura en dos partes distintas, *setup* y *loop*.

La función *setup* se encarga de inicializar los motores y establece la comunicación serial a 9600 baudios para poder recibir los comandos enviados desde Python.

```
1 #include <Servo.h>
2 #include <Stepper.h>
3
4 Servo servo;
5 Stepper myStepper(2038, 4, 9, 5, 10);
6 int angle = 90;
7 const int stepsPerRevolution = 2038;
8 const int pinLED = 11;
9
10 void setup() {
11     servo.attach(15);
12     servo.write(angle);
13     myStepper.setSpeed(5);
14     Serial.begin(9600);
15     pinMode(pinLED, OUTPUT);
16 }
```

Por otro lado, la función *loop*, se encarga de verificar si hay datos disponibles en el puerto serial, y dependiendo del comando que reciba realiza unas acciones u otras.

```
1 void loop()
2 {
3     if (Serial.available() > 0)
4     {
5         char var = Serial.read();
6
7         if (var == '1') {
8             servo.write(90);
9         }
10        else if (var == '2') {
11            servo.write(65);
12        }
13        // Otros comandos
14    }
15 }
```

## 5.2. Integración del software con el hardware

La integración de Python con el hardware controlado por el Teensy se consigue gracias a la comunicación bidireccional serial. Esto resulta ser que cuando Python envía comando al Teensy, este actúa sobre los motores, actuadores físicos y le devuelven el estado a Python para su procesamiento y posterior visualización en GUI.

El proceso se ve tal que así:

1. El puerto del serial se configura a 9600 baudios en ambos códigos. De esta manera conseguimos tener una comunicación sincronizada entre ambos códigos. Python le envía al Teensy unos bytes específicos para que este último los interprete y ejecute lo que sea conveniente.
2. El Teensy una vez recibido el código, traduce el comando y ejecuta un movimiento físico, ya sea del motor paso a paso o el servomotor.
3. Mientras fluye el intercambio de información, la GUI de Python se actualiza para reflejar el estado actual de la herramienta. De esta manera el usuario puede observar las acciones que se han realizado.

## 5.3. Pruebas y ajustes del sistema

Las pruebas y los ajustes del sistema son fases cruciales en un proyecto para poder asegurar que este funciona correctamente y de la manera más eficiente. Y el proceso consta de distintas pruebas:

- **Pruebas individuales.** Se comprueba individualmente el servomotor y el motor paso a paso para asegurarme de que responden correctamente a los comandos enviados desde el Teensy.

```
1  servo.write(90); // Prueba de posición inicial del servomotor
2  myStepper.step(stepsPerRevolution); // Prueba de rotación
   completa del motor paso a paso
3
```

- **Pruebas de comunicación Serial.** Se verifica que el Teensy fuera capaz de recibir los comandos que le enviaba Python y poder devolver una respuesta.

```
1  arduino.write(b'1') # Enviar comando para mover el servomotor
2
```

- **Pruebas de integración.** Se comprueba que cada componente que se ha programado funciona y ejecuta la acción deseada. Se conectan todos los componentes y se prueba la GIU completa para asegurarme que todos los botones controlan los accionadores adecuados.

```
1  botonAbrir = ttk.Button(root, text="Abrir", command=Abrir)
2
```

- **Pruebas funcionales.** En este paso se simula un escenario real para comprobar y ver como responde la herramienta y si se ejecuta de manera eficiente y precisa.

- **Ajustes de temporización.** Tras realizar las pruebas, se notó que no terminaban de satisfacer los tiempos de *delay* escogidos. Por lo que se ajustaron los tiempos de espera del Teensy, y el tiempo en el que los botones se encuentran desactivados tras haber pulsado ya uno en Python.

```
1  delay(1500); // Ajuste del tiempo de espera para el movimiento
   del servomotor
2
```

- **Corrección de errores y optimización.** Durante las pruebas, se detectan errores de comandos incorrectos o problemas en la sincronización entre el envío y la recepción de la información, y son cosas que se fueron depurando. Y una vez hechas las pruebas y basadas en los resultados de las mismas, se realizan los ajustes necesarios para asegurar el mayor rendimiento. Esto es por ejemplo, mejorar la disposición de los componentes en la GIU y refinar la lógica para evitar conflictos.

### 5.3.1. Resultado final

Tras todas las pruebas y depuraciones de código, así fue cómo quedó la interfaz

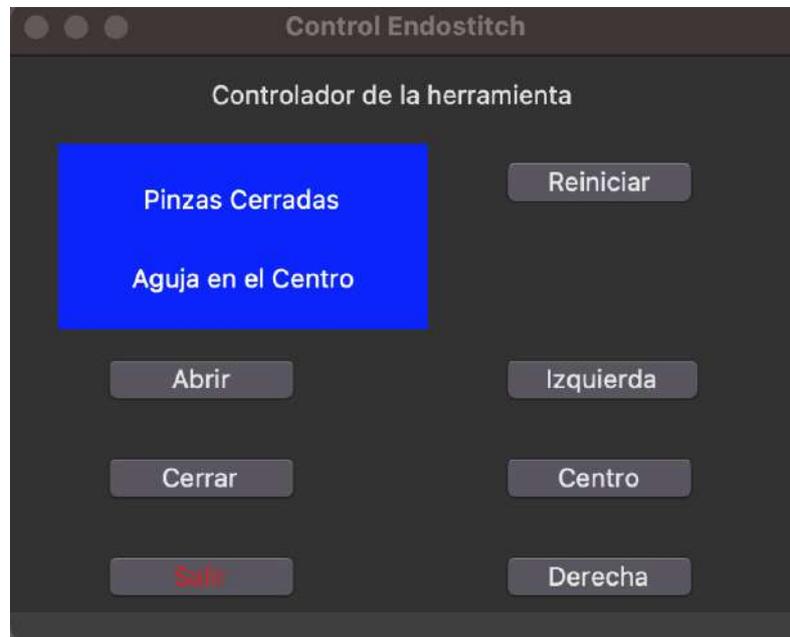
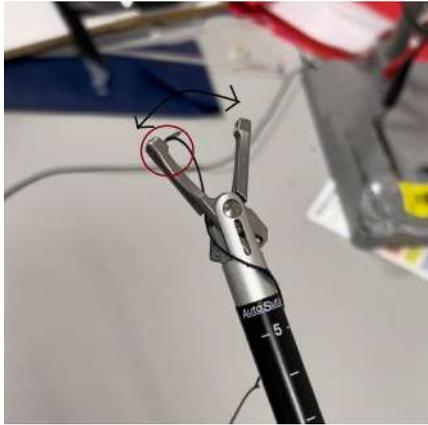


Figura 5.2: GIU final

Al pulsar cada uno de los botones, estos mandan una señal que se ejecuta. Por un lado está la aguja, que se mueve de un lado a otro desplazando la aguja de izquierda a derecha y viceversa como se puede observar en la figura [5.3](#).

Y por otro lado, la señal de accionar la palanca que abre o cierra la herramienta como se observa en la figura [5.4](#).

Se adjunta un enlace para poder ver el [vídeo del funcionamiento de la herramienta.](#)



(a) Aguja abierta en el lado izquierdo



(b) Aguja cerrada



(c) Aguja abierta en el lado derecho

Figura 5.3: Movimiento de la aguja en la herramienta



(a) Herramienta abierta



(b) Herramienta cerrada

Figura 5.4: Movimiento de la palanca de la herramienta

# Capítulo 6

## Conclusiones

### 6.1. Presentación de los resultados obtenidos

El proyecto de la herramienta motorizada ha terminado con la creación de una pieza funcional que integra hardware y software de una forma eficiente. Este es capaz de controlar de una manera bastante precisa un servomotor y un motor paso a paso.

La herramienta es capaz de abrir y cerrar las pinzas, y de igual modo tiene la capacidad de mover la aguja en distintas posiciones; a la izquierda, a la derecha y al centro para poder reemplazarla por otra nueva. Y todo esto se ha logrado con la ayuda de Python y su librería *tkinter*. Con esta librería he creado de una interfaz gráfica muy intuitiva y que es muy fácil de manejar, y que permite al usuario operar con ella sin complicaciones.

La integración entre Python y Teensy fue implementada con éxito gracias a la conexión serial, lo que ofrece una respuesta rápida y precisa, cosa que es muy necesaria a la hora de realizar una intervención quirúrgica. Y tras largas y exhaustivas pruebas y depuraciones de código, el sistema de ha vuelto estable y fiable para ser el primer prototipo.

## **6.2. Discusión sobre las limitaciones y posibles mejoras**

### **6.2.1. Limitaciones**

La velocidad de respuesta puede verse afectada y no se contempla una solución para ello. Además, la interfaz creada quizás sea poco intuitiva, y se podría desarrollar algo con un diseño más avanzado y con más opciones de visualización y no que solo sea funcional. Y por último, el sistema que he diseñado está muy limitado en cuanto a los componentes que puede utilizar.

### **6.2.2. Posibles mejoras**

Como todos los proyectos, siempre hay posibles mejoras que plantear y realizar. Y para este proyecto, que requiere mucha sensibilidad en el accionamiento de la herramienta, mejoraría el uso de motores con una mayor resolución o la implementación de encoders para obtener una feedback más preciso. Por otro lado, sería conveniente optimizar el código para poder conseguir una mayor velocidad de respuesta, y como se ha mencionado antes, desarrollar una interfaz con funcionalidades más avanzadas.

Algo que no se ha planteado y que es muy importante, sería implementar módulos de seguridad para asegurar que la herramienta no causa ningún tipo de daño al paciente. Esto podría ser que detuviese el movimiento en caso de encontrarse con alguna anomalía o error. Para ellos harían falta una variedad de sensores, y programar un rearme en caso de que se de la situación.

## **6.3. Resumen y reflexión de los logros alcanzados**

Este proyecto en el cual se ha diseñado y se ha desarrollado la herramienta motorizada ha sido una experiencia llena de aprendizaje, desafíos y logros. Se han puesto en uso los conocimientos de electrónica, programación y diseño que me ha proporcionado el grado tras tanto tiempo de aprendizaje, lo cual ha sido crucial para el desarrollo exitoso del proyecto. Las pruebas y la corrección de errores fueron un proceso que se perpetuó duran-

te todo el proyecto, pero con el cual he podido crecer tanto yo misma, como el proyecto en si. Ya que este me ha obligado a adaptarme a las situaciones y encontrar soluciones creativas a los problemas que surgían.

Se puede decir que la herramienta motorizada cumple con los requisitos iniciales del proyecto, realiza la función planteada y es muy fácil de utilizarla. La integración tanto entre los lenguajes de programación como la del software y el hardware fueron exitosas, y plenamente funcionales. Y tras muchas pruebas, a pesar de poder mejorarlo en muchos aspectos, es una sistema estable y fiable para que pueda ser probado con seguridad en piel sintética.



# Capítulo 7

## Bibliografía



# Bibliografía

- [1] Gagner, M., & Pomp, A. (1994). Laparoscopic Fundoplication in 10 Steps. *Surgical Endoscopy*, 8(1), 6-10.
- [2] McMahon, A. J., Baxter, J. N., & O'Dwyer, P. J. (1994). Laparoscopic vs. Mini-laparotomy Cholecystectomy: A Randomized Trial. *Lancet*, 343(8890), 135-138.
- [3] U.S. Food and Drug Administration. (2023). FDA Clears New Robotically-Assisted Surgical Device for Adult Patients. Recuperado de <https://www.fda.gov/news-events/press-announcements/fda-clears-new-robotically-assisted-surgical-device-adult-patients>
- [4] The Coding Train. (2023). Introduction to Machine Learning with JavaScript. Recuperado de <https://www.youtube.com/watch?v=MFTJpnDeuKM>
- [5] Morente, L. M. (2017). Todo lo que debes saber sobre Da Vinci, el robot quirúrgico. *Expansión Tecnología*.
- [6] Schauer, P. R., & Ikramuddin, S. (2001). Laparoscopic Surgery: Current Practices and Future Directions. *Journal of the American College of Surgeons*, 193(4), 394-408.
- [7] Albert, M. J. (2014). 'Broca', el robot quirúrgico con tacto. Recuperado de [https://elpais.com/sociedad/2014/04/14/actualidad/1397501475\\_830646.html#](https://elpais.com/sociedad/2014/04/14/actualidad/1397501475_830646.html#)
- [8] Leung, E., & Law, W. L. (2006). Laparoscopic Colorectal Surgery: Current Status and Future Directions. *Asian Journal of Surgery*, 29(3), 138-142.
- [9] Albu-Schäffer, A., Haddadin, S., Ott, C., & Stemmer, A. (2007). The DLR Lightweight Robot: Design and Control Concepts for Robots in Human Environments. *Industrial Robot: An International Journal*, 34(5), 376-385.

- [10] Ali, M. R., & Chung, R. S. (2003). The Endo Stitch device for laparoscopic suturing. *Journal of the Society of Laparoendoscopic Surgeons*, 7(3), 221-225.
- [11] Lee, S. W., & Nomura, E. (2009). Clinical applications of laparoscopic suturing and knot tying with the Endo Stitch™. *Surgical Endoscopy*, 23(4), 932-936.
- [12] Banzi, M., & Shiloh, M. (2014). *Make: Getting Started with Arduino*. O'Reilly Media.
- [13] Medoff, N. (2016). *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education.
- [14] Peters, J. H., & Ellison, E. C. (2001). *Principles of Laparoscopic Surgery*. Springer-Verlag.
- [15] Albert, M. J. (2014). 'Broca', el robot quirúrgico con tacto. Recuperado de [https://elpais.com/sociedad/2014/04/14/actualidad/1397501475\\_830646.html#](https://elpais.com/sociedad/2014/04/14/actualidad/1397501475_830646.html#)
- [16] Arduino. "Documentación de Arduino". Recuperado de <https://www.arduino.cc/en/Guide/Introduction>
- [17] COVIDIEN. (2014). Endo Stitch™ 10 mm Single Use Suturing Device: Instructions for Use. Recuperado de <https://www.covidien.com/endo-stitch>
- [18] Intuitive Surgical. (2023). da Vinci Surgical System. Recuperado de <https://www.intuitivesurgical.com>
- [19] KiCad. "Documentación de KiCad". Recuperado de <https://kicad.org/documentation/>
- [20] Medtronic. (2023). Endo Stitch™ Suturing Device. Recuperado de <https://www.medtronic.com/us-en/healthcare-professionals/products/surgical/advanced-suturing/endo-stitch-suturing-device.html>
- [21] Python. "Documentación oficial de Python". Recuperado de <https://docs.python.org/3/>
- [22] Rethink Robotics. (2021). Collaborative Robots: What Are They and How Do They Work? Recuperado de <https://www.rethinkrobotics.com/collaborative-robots/>

- [23] Universal Robots. (2020). What Is Collaborative Robot (Cobot) Technology? Recuperado de <https://www.universal-robots.com/what-is-a-cobots/>
- [24] Comparación entre placas Arduino”. Recuperado de <https://robots-argentina.com.ar/didactica/comparacion-entre-placas-arduino/>
- [25] Hernando, J. (2011). Cómo hacer una PCB en casa”. Recuperado de <https://es.wikihow.com/hacer-una-PCB-en-casa>
- [26] ”Documentación sobre PCBs”. Recuperado de [https://www.pcpaudio.com/pcpfiles/doc\\_amplificadores/PCBs/PCBs.php](https://www.pcpaudio.com/pcpfiles/doc_amplificadores/PCBs/PCBs.php)
- [27] ISO/TS 15066:2016. Robots and robotic devices – Collaborative robots. International Organization for Standardization.
- [28] ”Documentación de pySerial”. Recuperado de <https://pyserial.readthedocs.io/en/latest/>
- [29] ”Hugo RAS System”. Recuperado de <https://www.medtronic.com/covidien/en-us/products/robotic-assisted-surgery/hugo-ras-system.html>
- [30] ”Versius Surgical Robotic System”. Recuperado de <https://cmrsurgical.com/versius-surgical-robotic-system/>

# Capítulo 8

## Anexos

### 8.1. Código Teensy

```
1 //Variable de entrada
2 char var;
3
4 //Variables de Servomotor
5 #include <Servo.h>
6 Servo servo;
7 int angle = 90;
8
9 //Variables motor paso a paso
10 #include <Stepper.h>
11 Stepper myStepper(2038, 4, 9, 5, 10); //sequence IN1-IN3-IN2-IN4
12
13 const int stepsPerRevolution = 2038;
14 const int pinLED = 11; //Quiero que se encienda el LED para mostrar que
    funciona
15
16 void setup()
17 {
18     //SERVO
19     servo.attach(15); // vincula el servo al pin digital 15
20     servo.write(angle);
21
22     //PASO A PASO
23     myStepper.setSpeed(5);
```

```
24
25 Serial.begin(9600);
26 pinMode(pinLED, OUTPUT);
27 }
28 void loop()
29 {
30
31   digitalWrite(pinLED, HIGH); //ilumina el LED
32   delay(1000);
33   digitalWrite(pinLED, LOW);
34   delay(2000);
35
36   if (Serial.available() > 0)
37   {
38     var = Serial.read(); // read the incoming byte:
39
40     if(var == '1')
41     {
42       Serial.println("1: Servomotor posicion inicial 90 grados");
43       angle = 90;
44       servo.write(angle);
45       delay(1500);
46     }
47     else if(var == '2')
48     {
49       Serial.println("2: Servomotor posicion 65 grados");
50       //servo 100% CW (equivalente a angulo 180 grados)
51       angle = 65;
52       servo.write(angle);
53       delay(1500);
54     }
55     else if(var == '3')
56     {
57       Serial.println("3: Servomotor posicion 115 grados");
58       //servo 100% CW (equivalente a angulo 90 grados)
59       angle = 115;
60       servo.write(angle);
61       delay(1500);
62     }

```

```
63     else if(var == '4')
64     {
65         Serial.println("4: El motor paso a paso gira sentido horario (1
vuelta)");
66         myStepper.setSpeed(10); //Velocidad 10 RPM
67         myStepper.step(stepsPerRevolution);
68         delay(1000);
69     }
70     else if(var == '5')
71     {
72         Serial.println("5: El motor paso a paso gira sentido antihorario
(1 vuelta)");
73         myStepper.setSpeed(10); //Velocidad 10 RPM
74         myStepper.step(-stepsPerRevolution);
75         delay(1000);
76     }else
77         Serial.println("Error -> Escriba el valor adecuado");
78 }
79
80 }
```



## 8.2. Código Python

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 import serial
4 import time
5
6 arduino = serial.Serial('/dev/cu.usbmodem123451', 9600)
7 time.sleep(2)
8
9 ultima_accion = None # Variable para almacenar la ultima accion
   realizada
10
11 def write_msg_p(pinzas, b):
12     b.configure(text=pinzas)
13
14 def write_msg_a(aguja, c):
15     c.configure(text=aguja)
16
17 def Activar():
18     botonCerrar.config(state="normal") # Estado del boton es activado
19     botonAbrir.config(state="normal")
20     botonIzq.config(state="normal")
21     botonDer.config(state="normal")
22     botonCentro.config(state="normal")
23
24 def Desactivar():
25     botonCerrar.config(state="disabled") # Estado del boton es
   desactivado
26     botonAbrir.config(state="disabled")
27     botonIzq.config(state="disabled")
28     botonDer.config(state="disabled")
29     botonCentro.config(state="disabled")
30     root.after(3000, Activar) # Reactivar despues de 3 segundos (3000
   milisegundos)
31
32 def MostrarError():
33     messagebox.showerror("Error", "No se puede realizar esta accion
```

```
despues de Abrir")
34
35 def Abrir():
36     global ultima_accion
37     arduino.write(b'1') #4
38     Desactivar()
39     write_msg_p("\nPinzas Abiertas\n\n\n", T)
40     ultima_accion = "Abrir"
41
42 def Cerrar():
43     global ultima_accion
44     arduino.write(b'2') #5
45     Desactivar()
46     write_msg_p("\nPinzas Cerradas\n\n\n", T)
47     ultima_accion = "Cerrar"
48
49 def Centro():
50     global ultima_accion
51     if ultima_accion != "Abrir":
52         arduino.write(b'4') #1
53         write_msg_a("\nAguja en el Centro\n", P)
54     else:
55         MostrarError()
56
57 def Izquierda():
58     global ultima_accion
59     if ultima_accion != "Abrir":
60         arduino.write(b'5') #3
61         write_msg_a("\nAguja a la Izquierda\n", P)
62     else:
63         MostrarError()
64
65 def Derecha():
66     global ultima_accion
67     if ultima_accion != "Abrir":
68         arduino.write(b'3') #2
69         write_msg_a("\nAguja a la Derecha\n", P)
70     else:
71         MostrarError()
```

```
72
73 def msg(texto, box):
74     box.configure(text=texto)
75
76 def Reiniciar():
77     botonCerrar.config(state="normal")
78     Cerrar()
79     root.after(5000, Centro)
80
81 root = tk.Tk()
82 root.title("Control Endostitch")
83 root.config(width=400, height=300)
84 root.resizable(False, False)
85
86 titulo = ttk.Label(root, text="Controlador de la herramienta")
87 titulo.place(x=100, y=10)
88
89 T = tk.Label(root, height=5, width=20, bg="blue")
90 T.place(x=25, y=45)
91 P = tk.Label(root, height=3, width=20, bg="blue")
92 P.place(x=25, y=85)
93
94 s = ttk.Style()
95 s.configure("Salir.TButton", foreground="#C62828")
96 s.map("Salir.TButton", foreground=[("active", "#FFA500")])
97 s.map("Raton.TButton", foreground=[("active", "#008F39")])
98
99 botonAbrir = ttk.Button(root, text="Abrir", style="Raton.TButton",
    command=Abrir, state="normal")
100 botonAbrir.place(x=50, y=150)
101
102 botonCerrar = ttk.Button(root, text="Cerrar", style="Raton.TButton",
    command=Cerrar, state="normal")
103 botonCerrar.place(x=50, y=200)
104
105 botonSalir = ttk.Button(root, text="Salir", style="Salir.TButton",
    command=root.destroy, state="normal")
106 botonSalir.place(x=50, y=250)
107
```

```
108 botonIzq = ttk.Button(root, text="Izquierda", style="Raton.TButton",
    command=Izquierda, state="normal")
109 botonIzq.place(x=250, y=150)
110
111 botonCentro = ttk.Button(root, text="Centro", style="Raton.TButton",
    command=Centro, state="normal")
112 botonCentro.place(x=250, y=200)
113
114 botonDer = ttk.Button(root, text="Derecha", style="Raton.TButton",
    command=Derecha, state="normal")
115 botonDer.place(x=250, y=250)
116
117 botreiniciar = ttk.Button(root, text="Reiniciar", style="Raton.TButton",
    command=Reiniciar, state="normal")
118 botreiniciar.place(x=250, y=50)
119
120 root.mainloop()
```



## 8.3. Datasheets



### 1. 使用環境條件

Apply Environmental Condition :

No.	項目 item	規格 standard
1-1	保存溫度 Storage Temperature Range	-20°C ~ 60°C
1-2	操作溫度 Operating Temperature Range	-10°C ~ 50°C
1-3	操作電壓 Operating Voltage Range	4.8~6.0V

### 2. 測試環境

Standard Test Environment :

2-1	測試環境 Standard Test Environment	<p>每一个检查必须是正常的温度和湿度进行测量，温度 <math>25 \pm 5^\circ\text{C}</math>，相对湿度 <math>65 \pm 10\%</math>，在按照本规范的标准测试条件下判断特征。</p> <p>Every characteristic of the inspect must be normal temperature and humidity carry out the test , temperature <math>25 \pm 5^\circ\text{C}</math> and relative humidity <math>65 \pm 10\%</math> of judgment made in accordance with this specification standard testing conditions.</p>
-----	-----------------------------------	---

### 3. 外觀檢查

Appearance Inspection :

No.	項目 item	規格 standard
3-1	外觀尺寸 Outline Drawing	尺寸见附件 Dimension see the attachment
3-2	外觀 Appearance	无损坏，不允许影响功能 No damage which affects functions allowed

#### 4. 電氣特性

Electrical Specification (Function of the Performance) :

No.	項目 item	4.8V	6.0V
4-1	空載轉速 Operating speed (at no load)	0.14 sec/60°	0.12 sec/60°
4-2	空載電流 Running current (at no load)	250 mA	300 mA
4-3	停止扭力 Stall torque (at locked)	3.5 kg-cm	4.4 kg-cm
4-4	停止電流 Stall current (at locked)	800 mA	900 mA
4-5	待機電流 Idle current (at stopped)	4 mA	5 mA

注：項目 4-2 定义平均值时，伺服器无负荷运行

Note: Item 4-2 definition is average value when the servo running with no load

#### 5. 機械特性

Mechanical Specification :

No.	項目 item	規格 standard
5-1	外觀尺寸 Overall Dimensions	见附件 See the drawing
5-2	機構極限角度 Limit angle	180° ± 10°
5-3	重量 Weight	43 ± 1g
5-4	導線規格 Connector wire gauge	# 28 PVC
5-5	導線長度 Connector wire length	300 ± 5 mm
5-6	舵片規格 Horn gear spline	25T/φ 5.80
5-7	舵片種類 Horn type	十字，圓盤，星型，條型 Cross , Disk , Star, Double
5-8	減速比 Reduction ratio	1/240



Product Name  
模拟伺服器 Analog Servo

Model No.  
3001HB

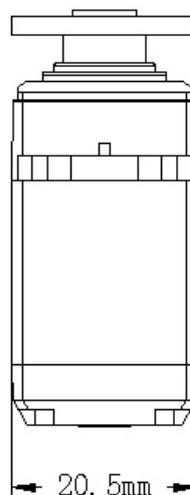
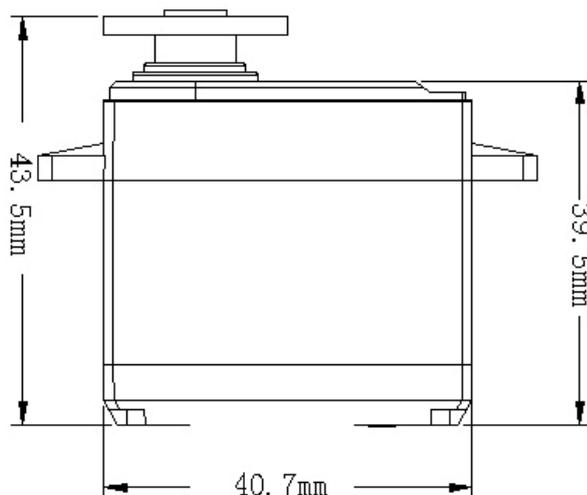
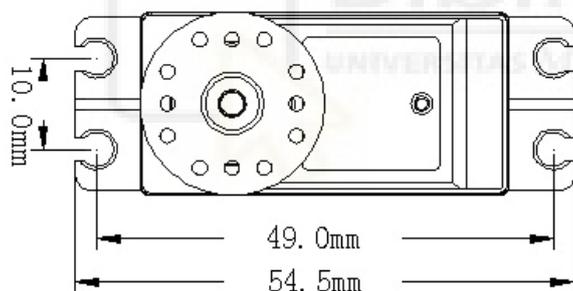
Version  
V1

Page  
2/3

## 6. 控制特性

Control Specification :

No.	項目	規格
6-1	控制系統 Control system	改變脈沖寬度 Pulse Width Modification
6-2	放大器種類 Amplifier type	模擬控制器 Analog Controller
6-3	操作角度 Operating travel	90° (在 1000→2000 $\mu$ sec)
6-4	中立位置 Neutral position	1500 $\mu$ sec
6-5	脈波訊號虛位 Dead band width	4 $\mu$ sec
6-6	旋轉方向 Rotating direction	順時針 (在 1500→2000 $\mu$ sec) Counterclockwise (when 1500→2000 $\mu$ sec)
6-7	脈波寬度範圍 Pulse width range	800→2200 $\mu$ sec
6-8	可作動角度範圍 Maximum travel	大約 165° (在 800→2200 $\mu$ sec) Approx 165° (when 800→2200 $\mu$ sec)



Product Name  
模擬伺服器 Analog Servo

Model No.  
3001HB

Version  
V1

Page  
3/3


 PM Motor  
 PM Type

## Reference Characteristics

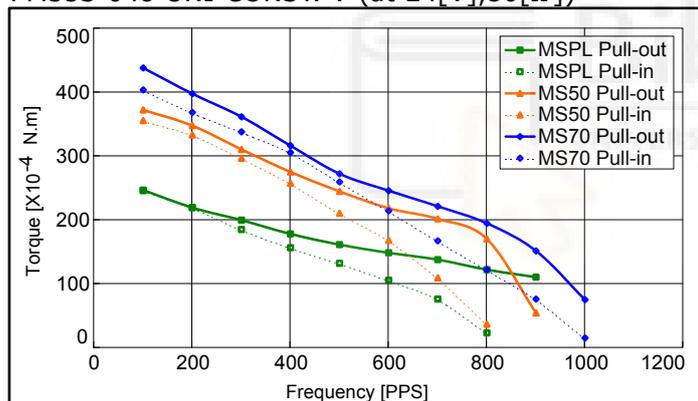
Motor Size	PM35S-048	
Number of Steps per Rotation	48(7.5°/Step)	
Drive Method	2-2 PHASE	
Drive Circuit	UNIPOLAR CONST. VOLT.	BIPOLAR CHOPPER
Drive Voltage	24[V]	24[V]
Current/Phase	600[mA]	
Coil Resistance/Phase	50[Ω]	6.6[Ω]
Drive IC	2SC3346	UDN2916B-V
Magnet Material	Ferrite plastic magnet (MSPL) Polar anisotropy ferrite sintered magnet (MS50) Nd-Fe-B bonded magnet (MS70)	
Insulation Resistance	100M[Ω] MIN	
Dielectric Strength	AC 500[V] 1[min]	
Class of Insulation	CLASS E	
Operating Temp.	-10[°C] ~ 50[°C]	
Storage Temp.	-30[°C] ~ 80[°C]	
Operating Hum.	20[%] RH ~ 90[%] RH	

## Applications

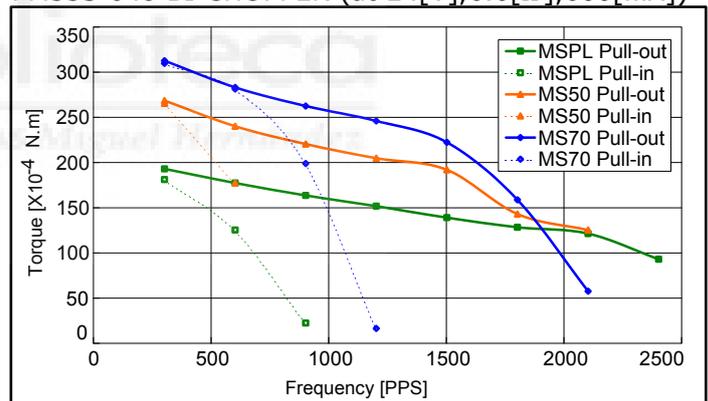
OA Equipment : Printers / Scanners  
 Industrial equipment : Flow control valves  
 Automotive : Meters / Optic axis control device  
 Home automation appliances : Sewing machines

## Torque Characteristics

PM35S-048 UNI-CONST. V (at 24[V],50[Ω])

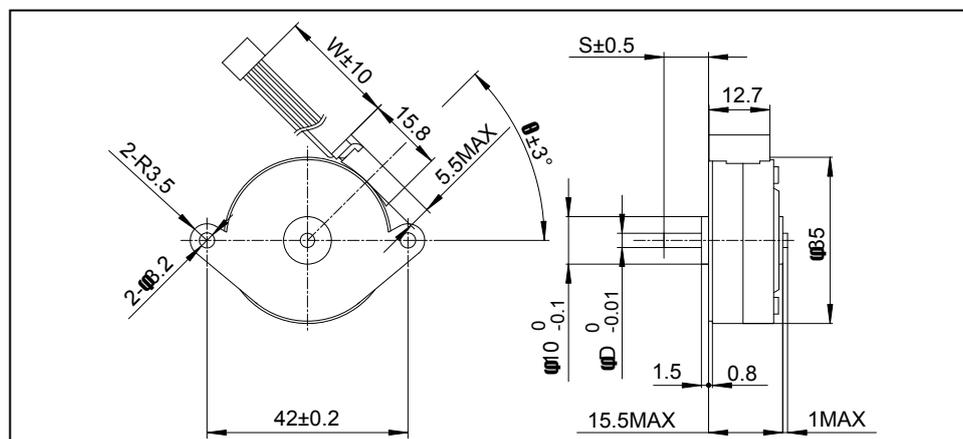


PM35S-048 BI-CHOPPER (at 24[V],6.6[Ω],600[mA])



These torque values are reference only. Heat radiation conditions and temperature rise effect by duty are different on each equipment, therefore please select motors after considering the heat conditions in the actual equipment.

## Dimensions

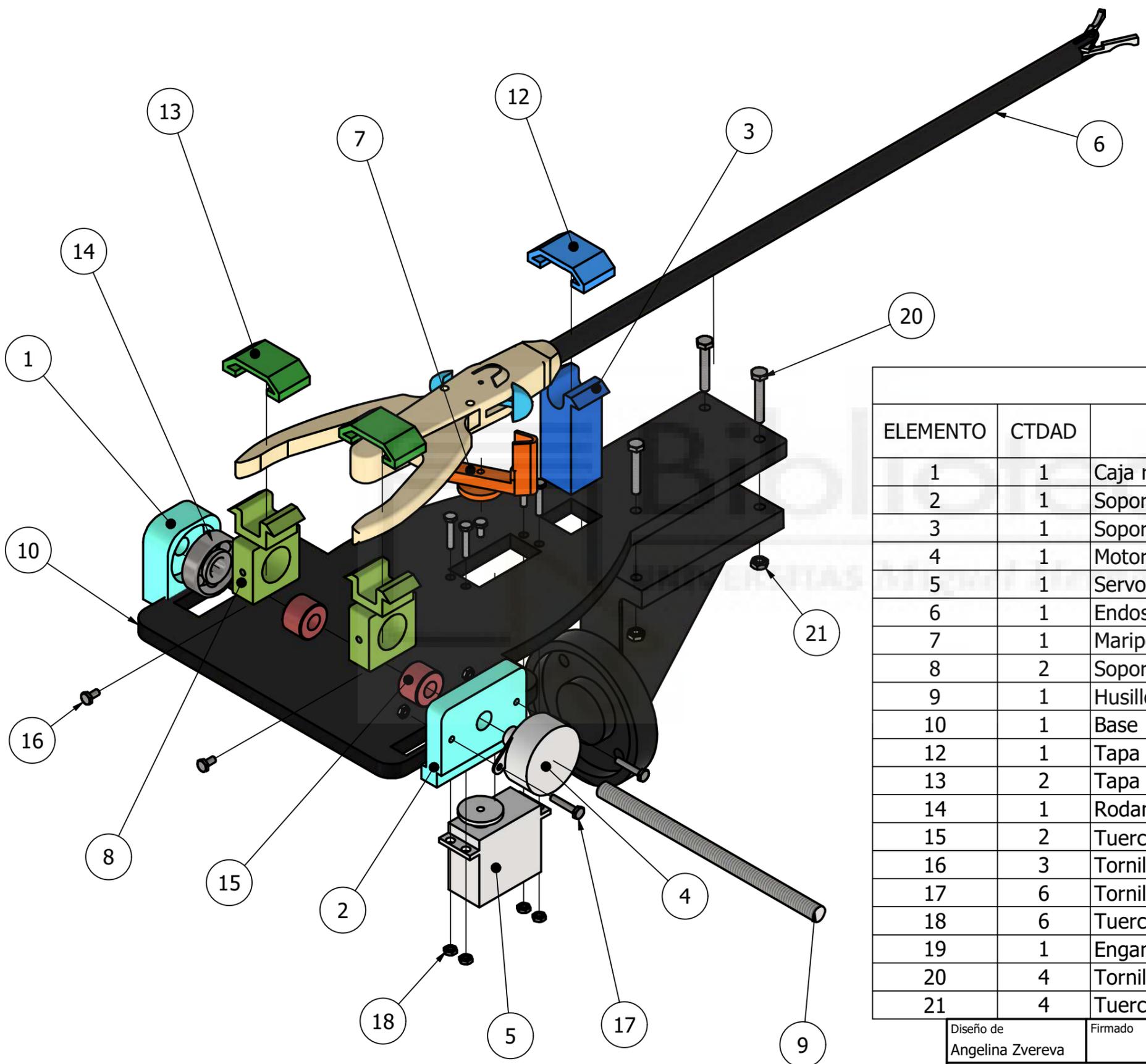


If you would like to know this Dimensions(D,S,W,θ), Please see Standard Dimensions in our Home Page.

www.DataSheet4U.com

## 8.4. Planos





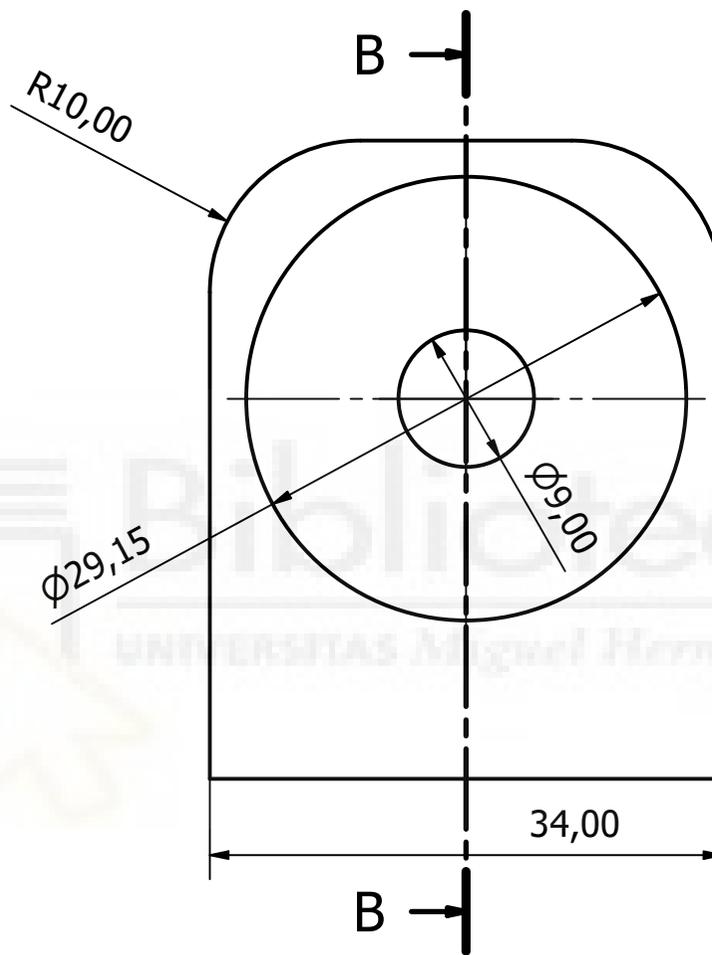
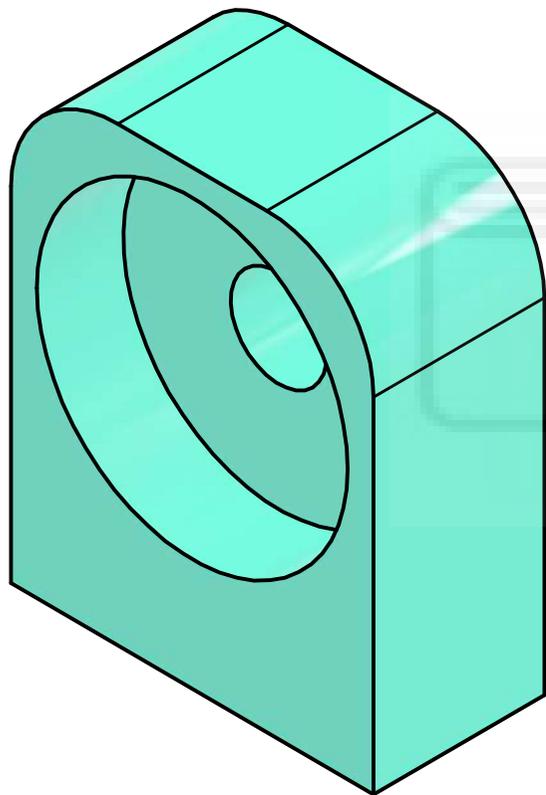
LISTA DE PIEZAS

ELEMENTO	CTDAD	Nº DE PIEZA	Nº DE PLANO
1	1	Caja rodamiento	01.01
2	1	Soporte motor paso a paso	01.02
3	1	Soporte cilindro	01.03
4	1	Motor paso a paso	
5	1	Servo 3001HB	
6	1	Endostitch	
7	1	Mariposa	01.04
8	2	Soporte palanca	01.05
9	1	Husillo D = 8 x 140	
10	1	Base	01.06
12	1	Tapa cilindro	01.07
13	2	Tapa palanca	01.08
14	1	Rodamiento de bolas Dint = 10	
15	2	Tuerca para husillo cilíndrica WSRM-1812TR8X1.5	
16	3	Tornillo - M3 x 6	
17	6	Tornillo - M3 x 16	
18	6	Tuerca - M3	
19	1	Enganche robot	01.09
20	4	Tornillo - M4 x 25	
21	4	Tuerca - M4	

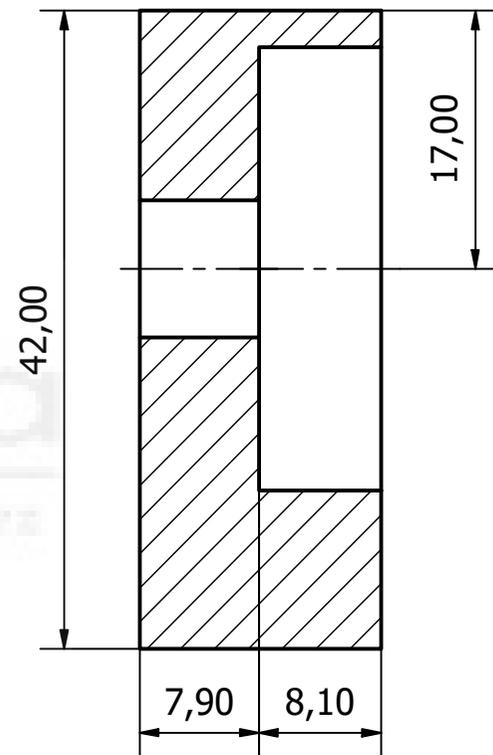
Diseño de Angelina Zvereva    Firmado *[Firma]*    Escala 1:2    Fecha 19/05/2024    Unidades mm



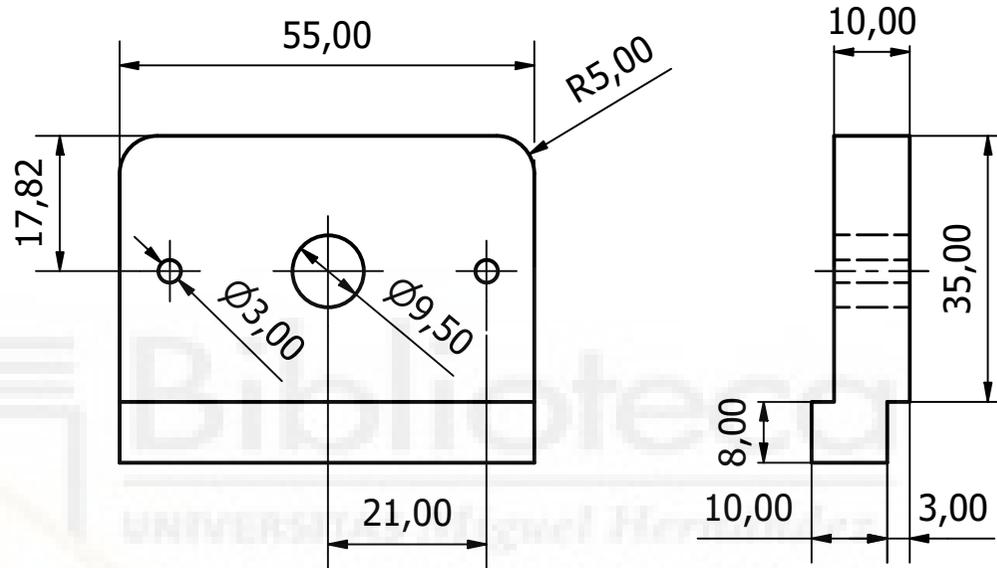
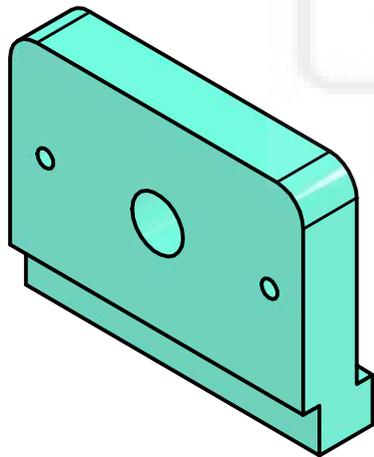
Documento: Planos  
Explosionado    Nº de Plano 01.00

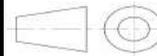


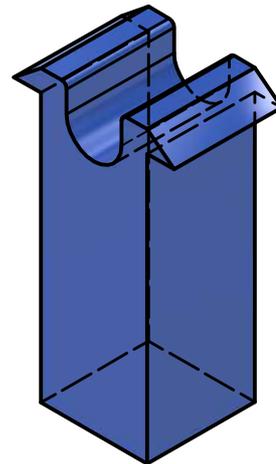
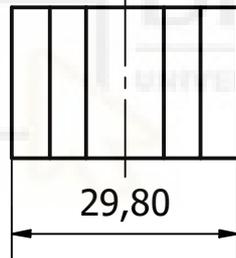
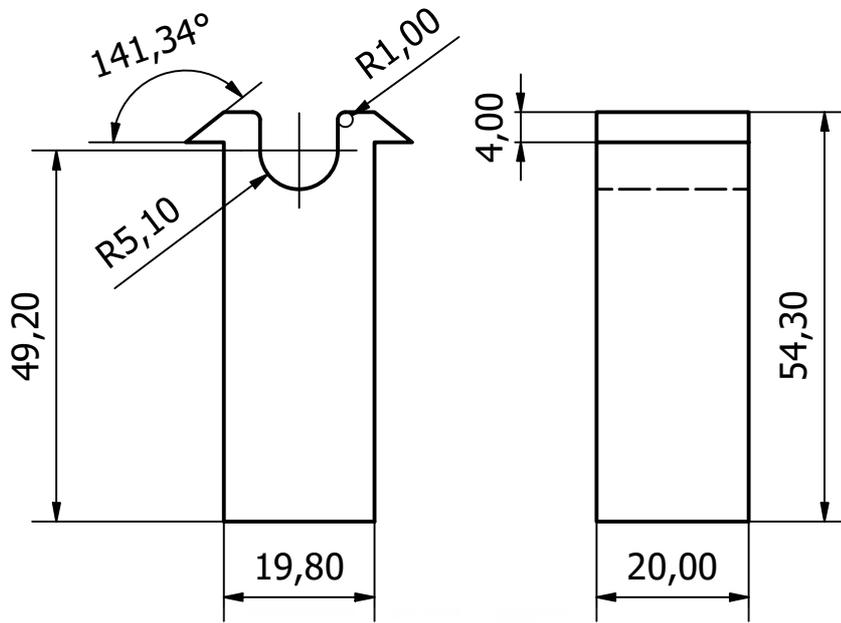
B-B ( 2 : 1 )



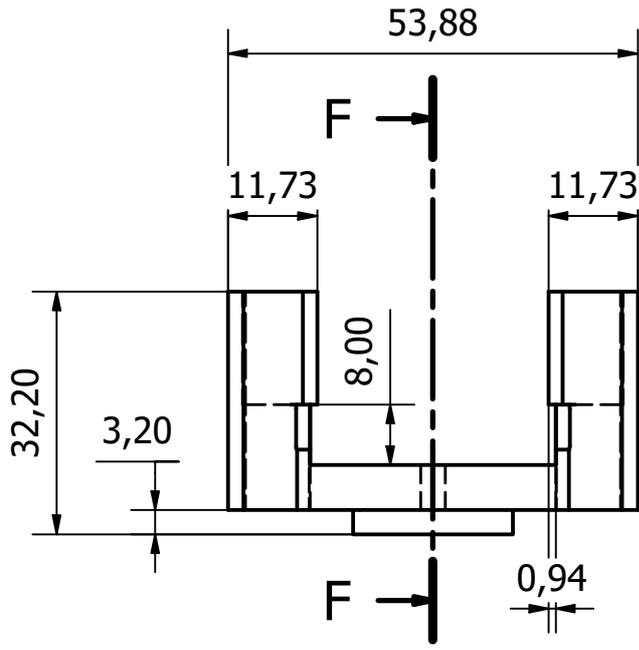
Diseño de Angelina Zvereva	Firmado 	Escala 2:1		Fecha 19/05/2024	Unidades mm
 UNIVERSITAS Miguel Hernández		Documento: Planos			
		Caja rodamiento		Nº de Plano 01.01	



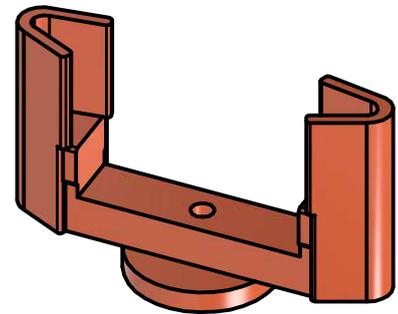
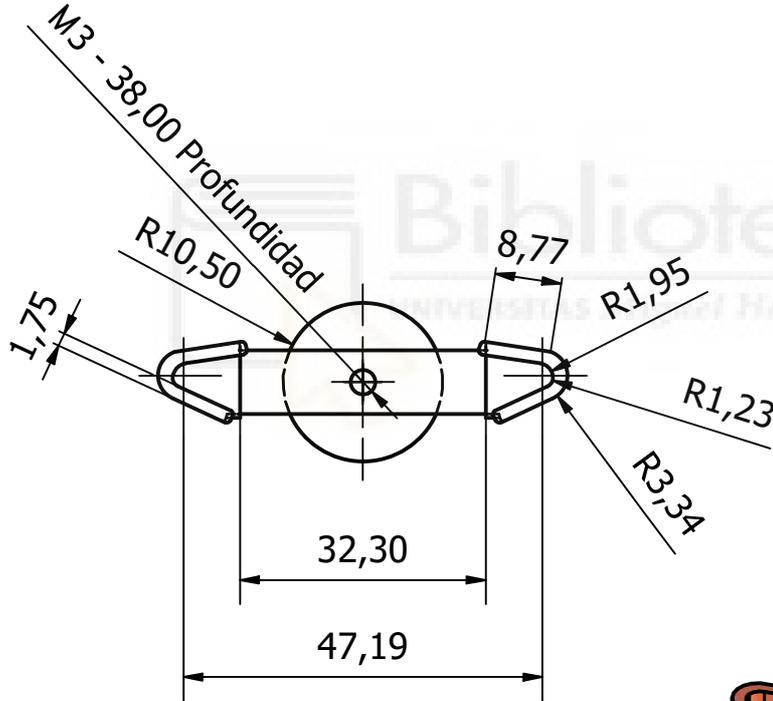
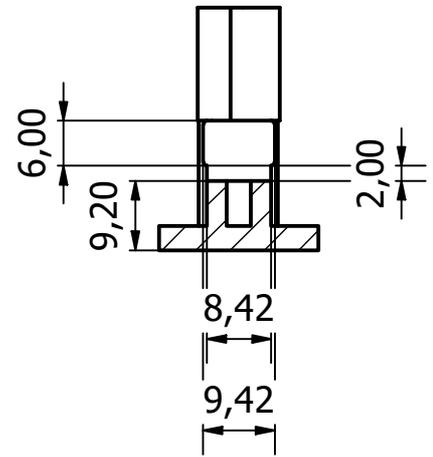
Diseño de Angelina Zvereva	Firmado 	Escala 1:1		Fecha 19/05/2024	Unidades mm
 <b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos			
		Soporte motor paso a paso	Nº de Plano 01.02		



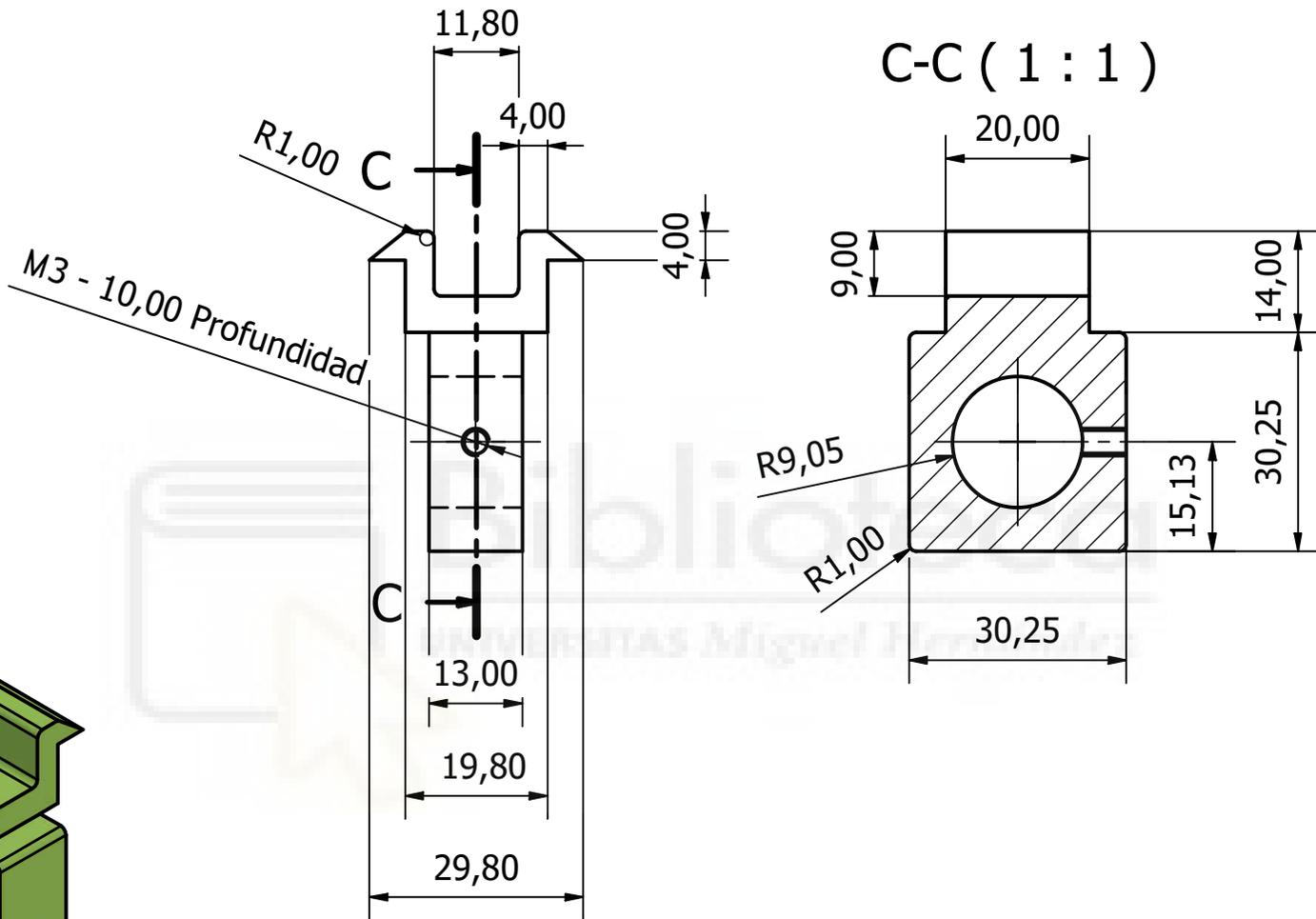
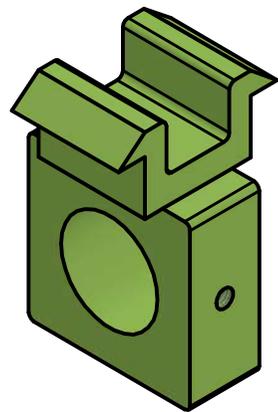
Diseño de Angelina Zvereva	Firmado <i>Angelina</i>	Escala 1:1		Fecha 19/05/2024	Unidades mm
		Documento: Planos			
		Soporte cilindro		Nº de Plano 01.03	

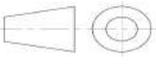


F-F ( 1 : 1 )

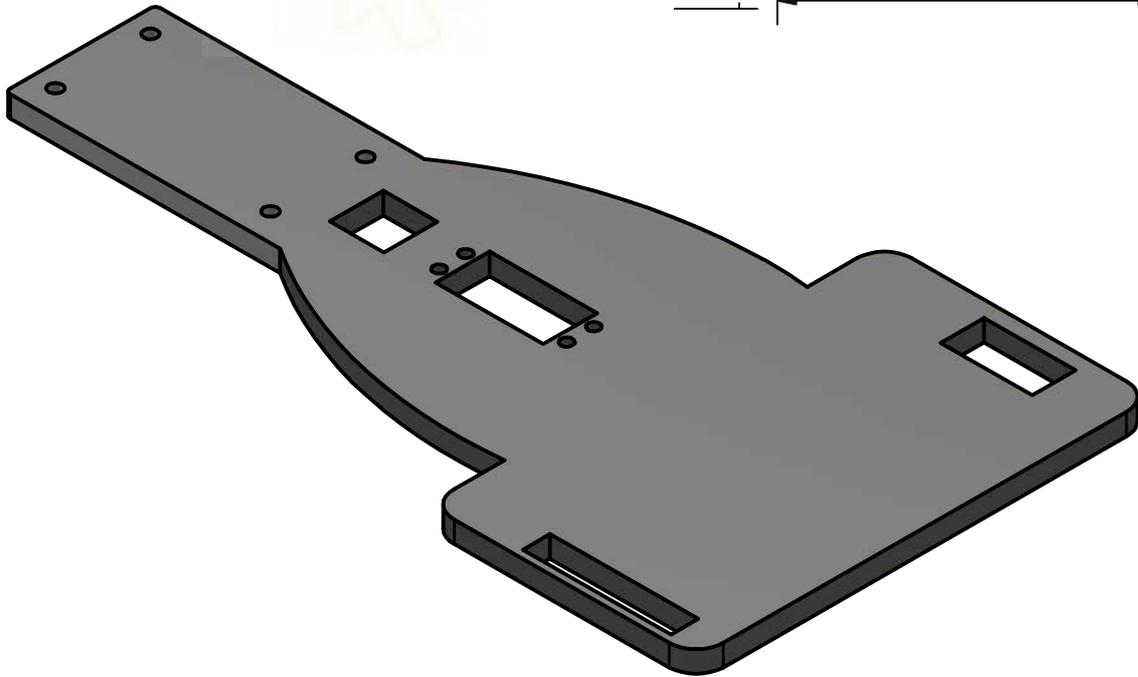
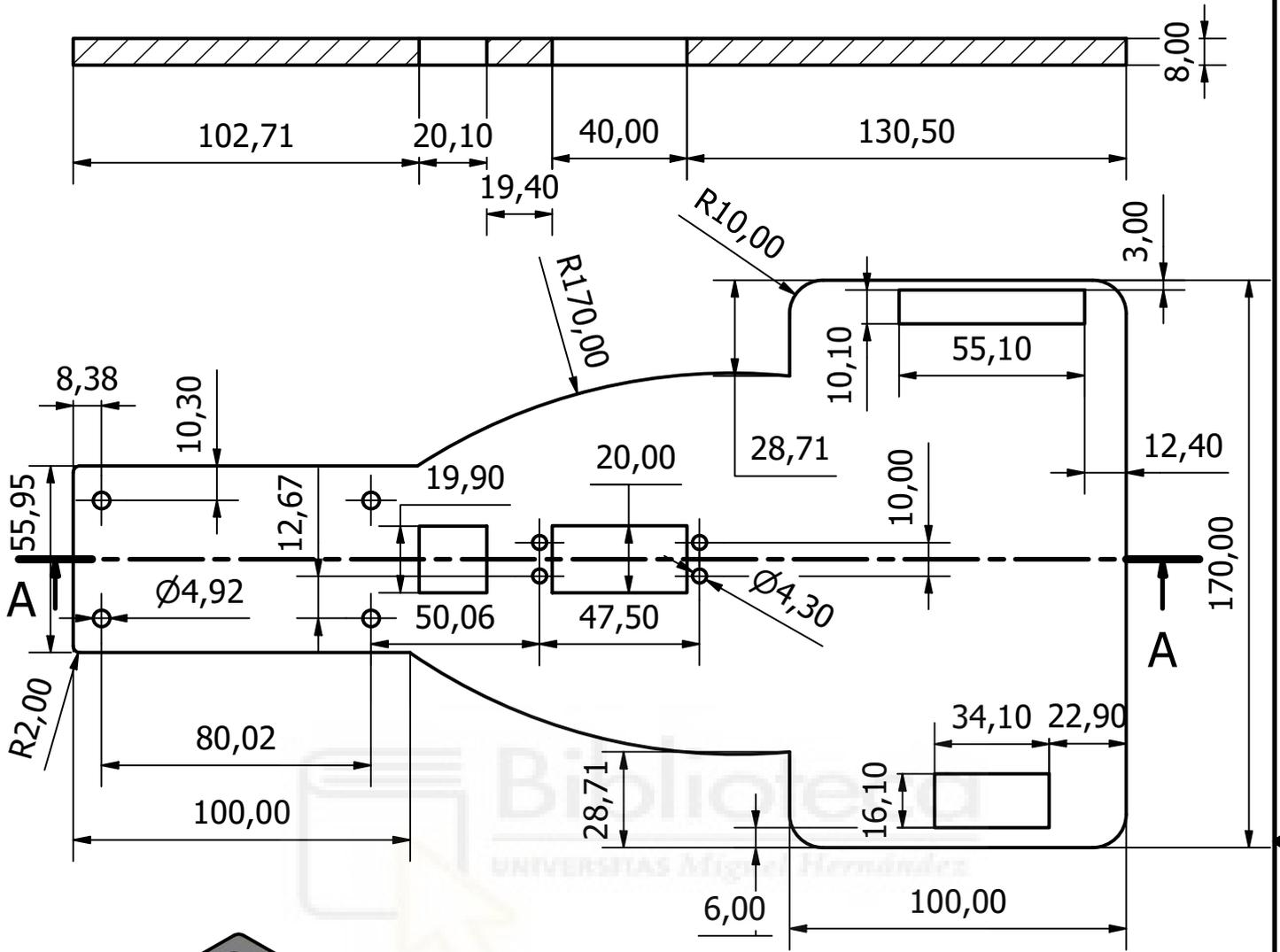


Diseño de Angelina Zvereva	Firmado <i>Angelina</i>	Escala 1:1	Fecha 19/05/2024	Unidades mm
		Documento: Planos		
		Mariposa	Nº de Plano 01.04	

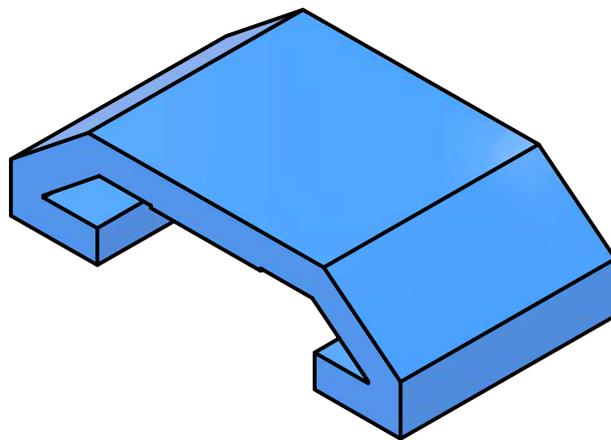
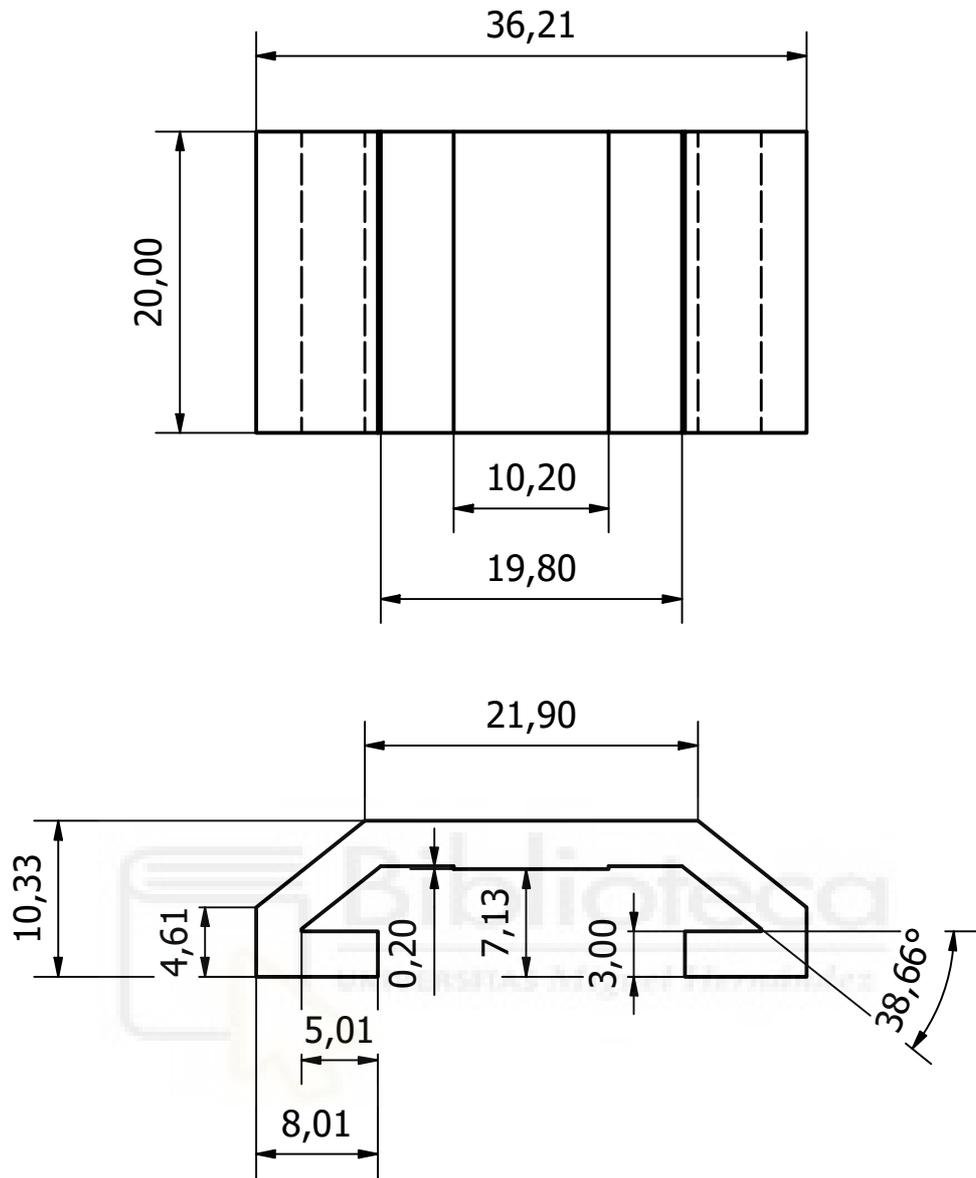


Diseño de Angelina Zvereva	Firmado <i>Angelina</i>	Escala 1:1		Fecha 19/05/2024	Unidades mm
 <b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos			
		Soporte palanca		Nº de Plano 01.05	

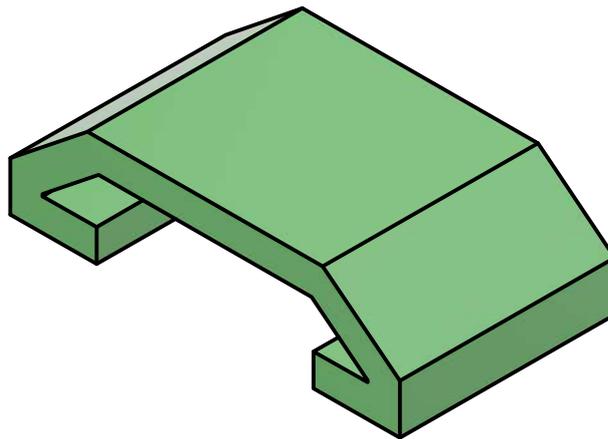
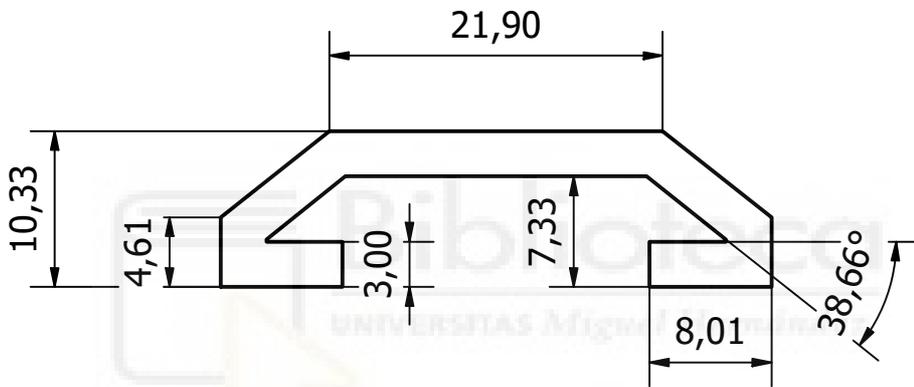
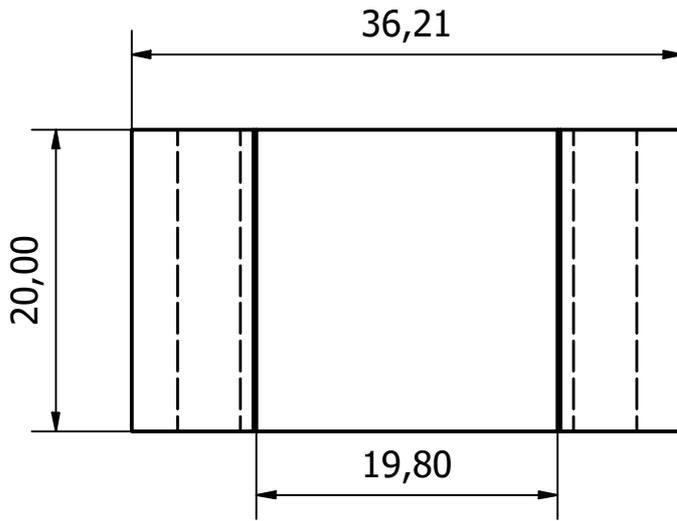
# A-A ( 1 : 2 )



Diseño de Angelina Zvereva	Firmado <i>Angelina</i>	Escala 1:2	Fecha 19/05/2024	Unidades mm
 <b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos		
		Base	Nº de Plano 01.06	

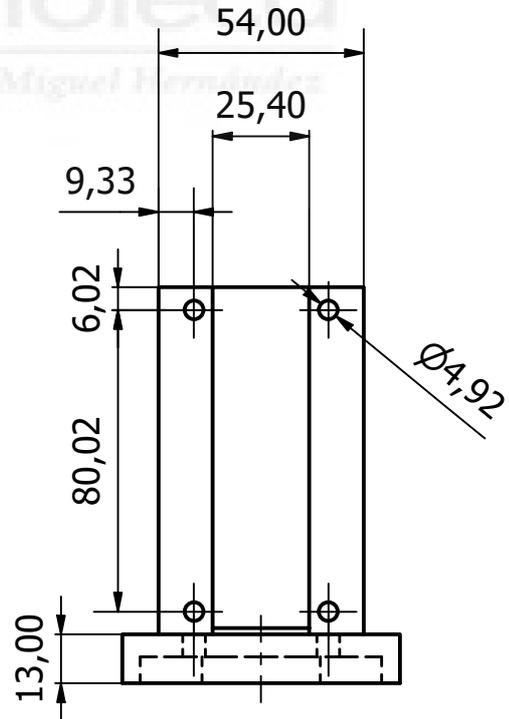
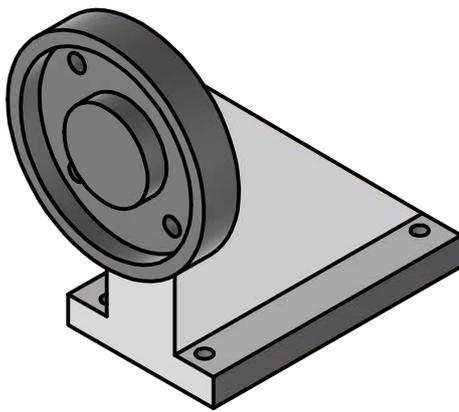
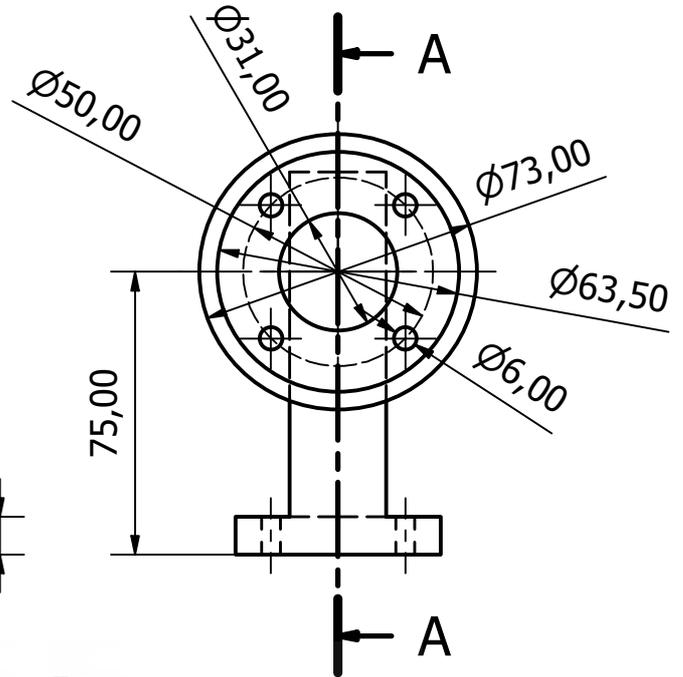
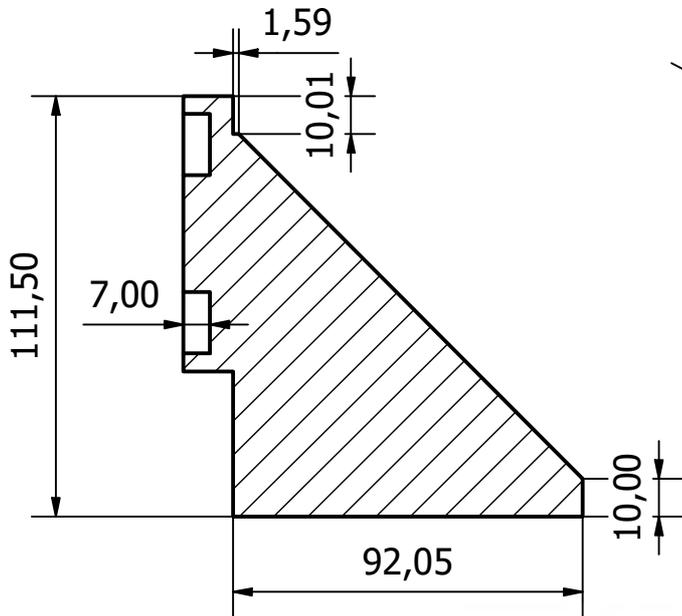


Diseño de Angelina Zvereva	Firmado 	Escala 2:1		Fecha 19/05/2024	Unidades mm
<b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos			
		Tapa cilindro	Nº de Plano 01.07		



Diseño de Angelina Zvereva	Firmado 	Escala 2:1		Fecha 19/05/2024	Unidades mm
<b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos			
		Tapa palanca	Nº de Plano 01.08		

# A-A ( 1 : 2 )



Diseño de Angelina Zvereva	Firmado 	Escala 1:2		Fecha 19/05/2024	Unidades mm
<b>UNIVERSITAS</b> Miguel Hernández		Documento: Planos			
		Enganche robot	Nº de Plano 01.09		