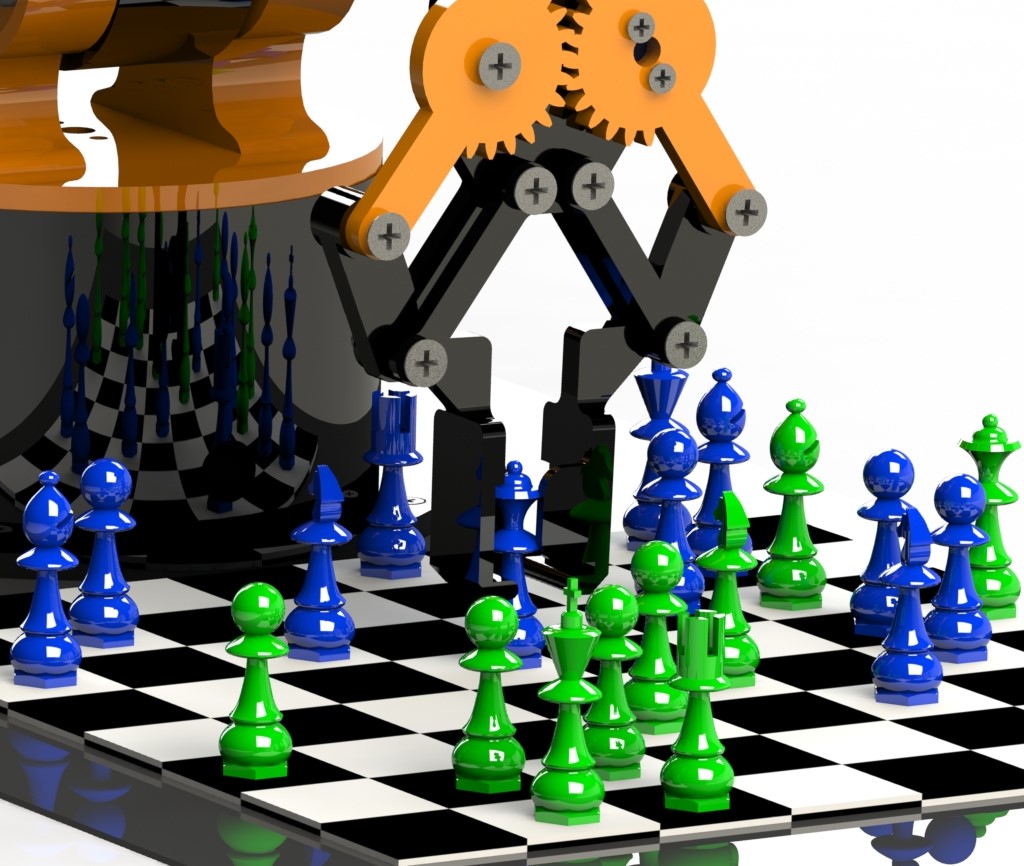


***Universidad Nacional de Lomas de Zamora Facultad de Ingeniería***

**PROYECTO EN INGENIERIA MECATRONICA**

**"MecaChess Robotic Arm"**



**INTEGRANTES - DNI:**

Gastón Alejandro Díaz - 43308937

Guillermo Daniel Duarte – 40888363

**DOCENTES:**

Cristian Leandro Lukaszewicz

Ezequiel Blanca

Juan Ignacio Szombach

**FECHA DE ENTREGA:** -/-/24

Índice

[1.Introducción 3](#_Toc181103495)

[1.1. Diseño Mecánico 3](#_Toc181103496)

[1.2. Programación 3](#_Toc181103497)

[1.3. Electrónica 3](#_Toc181103498)

[1.4. Análisis y Cálculos 4](#_Toc181103499)

[1.5. Materiales y Costos 4](#_Toc181103500)

[1.6. Integración y Pruebas 4](#_Toc181103501)

[2.Diseño y Construcción del Brazo Robótico 5](#_Toc181103502)

[2.1. Desarrollo del Diseño en SolidWorks 5](#_Toc181103503)

[2.1.1. Base 5](#_Toc181103504)

[2.1.2. Hombro 6](#_Toc181103505)

[2.1.3. Brazo 8](#_Toc181103506)

[2.1.4. Antebrazo 9](#_Toc181103507)

[2.1.5. Muñeca 10](#_Toc181103508)

[2.1.6. Mano 11](#_Toc181103509)

[2.1.7. Union de pinza parte 1 13](#_Toc181103510)

[2.1.8. Union de pinza parte 2 14](#_Toc181103511)

[2.1.9. Unión de pinza parte 3 14](#_Toc181103512)

[2.1.10. Dedos 15](#_Toc181103513)

[3.Diseño del esquemático eléctrico 18](#_Toc181103514)

[3.1. Diseño del Esquemático Eléctrico en Wokwi 18](#_Toc181103515)

[4.Cinemática directa 20](#_Toc181103516)

[4.1. Cinemática directa por método de Denavit-Hartenberg 20](#_Toc181103517)

[5.Cinemática inversa 23](#_Toc181103518)

[5.1. Introducción 23](#_Toc181103519)

[5.2. Método geométrico 24](#_Toc181103520)

[5.3. Método analítico 30](#_Toc181103521)

[6.Jacobiano analítico 37](#_Toc181103522)

[7.Algoritmos de Control 40](#_Toc181103523)

[7.1. Detección y Procesamiento del tablero 40](#_Toc181103524)

[7.1.1. Inicializacion de la Camara 40](#_Toc181103525)

[7.1.2. Reconocimiento del Tablero 40](#_Toc181103526)

[7.2. Detección de las Piezas 41](#_Toc181103527)

[7.3. Integración del motor de ajedrez Stockfish 41](#_Toc181103528)

[7.4. Comunicación Bidireccional entre Phyton y Arduino 41](#_Toc181103529)

[7.4.1. Transmision de Comandos de Movimiento a Arduino: 42](#_Toc181103530)

[7.4.2. Recepcion de Señales desde Arduino: 42](#_Toc181103531)

[7.5. Validación de Jugadas Utilizando la Biblioteca Chess de Phyton 42](#_Toc181103532)

[7.5.1. Verificacion de Legalidad de Jugadas: 42](#_Toc181103533)

[7.5.2. Funcionalidad Actual: 42](#_Toc181103534)

[7.5.3. Potencial para Expansion: 42](#_Toc181103535)

[7.6. Administración de Coordenadas y Configuración del Tablero en el Brazo Robótico 43](#_Toc181103536)

[7.7. Funcionalidad Pick-and-Place para Preparación del Tablero 43](#_Toc181103537)

[8. Diagrama de Flujo del Proceso de Funcionamiento del Sistema de Ajedrez Automatizado 43](#_Toc181103538)

[9.Costos del Prototipo y Presupuesto 45](#_Toc181103539)

[10. Comparación del Desempeño del Brazo Robot: Estado Previo y Posterior a las Mejoras 49](#_Toc181103540)

# 1.Introducción

El presente proyecto denominado "MecaChess Robotic Arm" tiene como finalidad la creación de un brazo robótico antropomórfico de 4 grados de libertad, diseñado para jugar al ajedrez de manera autónoma. Además, se le incorporaron funciones adicionales que permiten acomodar las piezas al inicio de una partida, jugar a través de la aplicación Chess.com, moverse por comandos de voz e identificar jugadas ilegales, entre otras. Para hacerlo más accesible, se emplearon materiales económicos y reciclados, logrando un balance óptimo entre costo y precisión sin comprometer la funcionalidad del brazo. Por último, se continúa trabajando en la incorporación de nuevas funcionalidades, como la implementación de inteligencia artificial y mejoras en el diseño, entre otros aspectos.

1.1. Diseño Mecánico

El diseño mecánico del brazo robot se centró en la optimización estructural para asegurar precisión y estabilidad durante el juego de ajedrez. Se implementaron materiales ligeros y resistentes, como los proporcionados por la tecnología de impresión 3D, y se evaluaron diversas configuraciones cinemáticas para alcanzar un equilibrio entre complejidad y eficiencia. Finalmente, se optó por un diseño de manipulador antropomórfico con 4 GDL, conocido por su versatilidad y precisión en aplicaciones de manipulación. El diseño fue realizado en SolidWorks, permitiendo una visualización detallada y un análisis estructural del brazo.

1.2. Programación

En el ámbito de la programación, se desarrollaron algoritmos avanzados para el control del movimiento del brazo, incluyendo la planificación de trayectorias. Respecto a la planificación de trayectorias, se programó en Arduino utilizando una técnica de tabulación de coordenadas en el tablero de ajedrez. Además, se implementaron funciones en Arduino para las cinemáticas directas e inversas, proporcionando flexibilidad en el manejo de los movimientos del brazo robot. Entonces Arduino nos ayudó para programar la parte de la lógica y ejecución de los movimientos del brazo robot, integrando así todas las funcionalidades mencionadas anteriormente.

Python fue utilizado para desarrollar los algoritmos de reconocimiento de patrones que permiten identificar las piezas de ajedrez y sus posiciones en el tablero, utilizando como método de visión OpenCV. En este mismo contexto, se empleó PyQt para generar una interfaz gráfica que facilitó la interacción con el sistema.

1.3. Electrónica

La electrónica del brazo robótico es sencilla pero efectiva. Se utiliza una pequeña placa en la que se han soldado pines para una mejor organización de los cables de alimentación de 6V y tierra (GND) provenientes de la fuente regulable. Cada servomotor del brazo consta de tres cables: dos de estos cables se conectan a la placa para gestionar la alimentación, mientras que el tercer cable se dirige al Arduino, el cual envía las señales para controlar el movimiento del brazo robótico.

Además, se realizó una simulación en Proteus para verificar y garantizar el correcto funcionamiento del sistema electrónico.

1.4. Análisis y Cálculos

Se realizaron los cálculos necesarios para garantizar el correcto funcionamiento del brazo robótico. Esto incluyó la cinemática directa, la cinemática inversa y como adicional el cálculo del jacobiano. Estos cálculos permitieron modelar y controlar con precisión los movimientos del brazo, asegurando que pudiera alcanzar y manipular correctamente las piezas de ajedrez en el tablero.

1.5. Materiales y Costos

Se hizo un esfuerzo por utilizar materiales accesibles, económicos y reciclados. La elección de componentes y materiales se basó en un compromiso entre costo y funcionalidad, asegurando que el brazo robot fuera no solo eficiente, sino también asequible y replicable.

1.6. Integración y Pruebas

Finalmente, se realizaron pruebas exhaustivas para evaluar el rendimiento del brazo robot en condiciones reales de juego. Estas pruebas incluyeron la calibración de los sistemas, la verificación de la precisión en la manipulación de las piezas y la evaluación de la velocidad de respuesta. Los resultados obtenidos permitieron ajustes finales para garantizar el cumplimiento de los objetivos planteados en el proyecto. Es importante destacar que el brazo robot sigue en constante mejora y actualización, con el fin de perfeccionar su desempeño y adaptarse a nuevas exigencias.

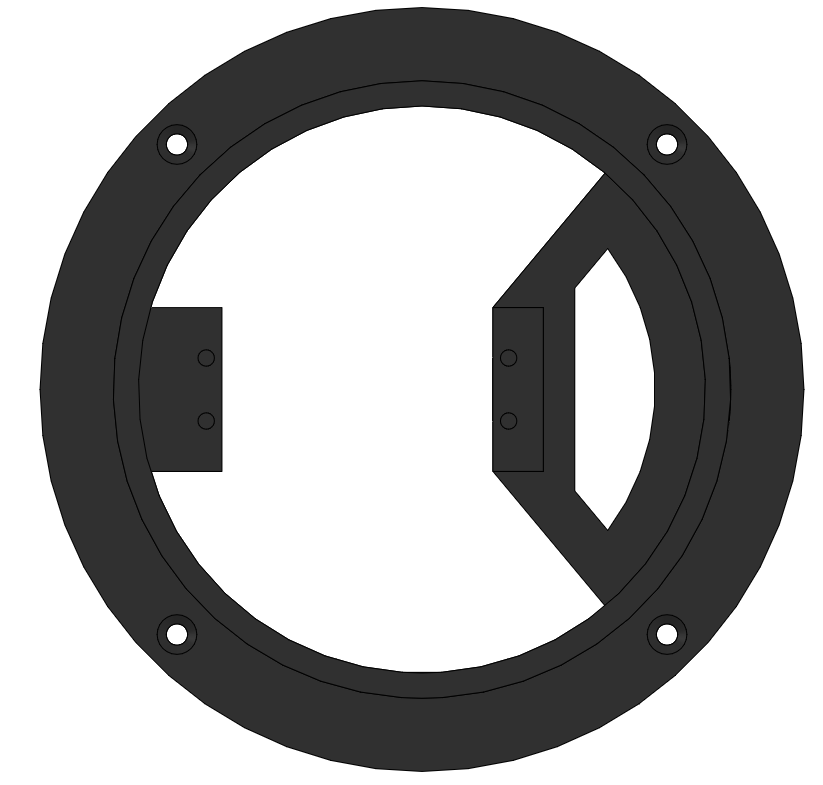
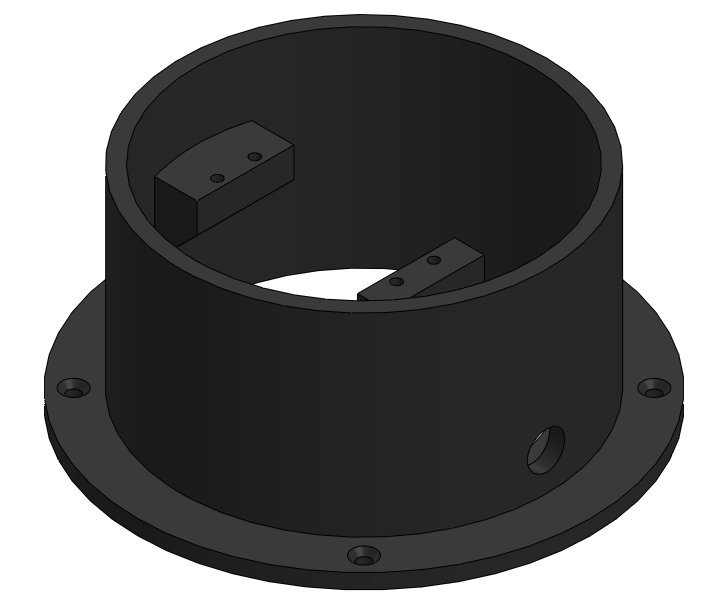
# 2.Diseño y Construcción del Brazo Robótico

2.1. Desarrollo del Diseño en SolidWorks

2.1.1. Base

El diseño de la base del brazo robótico consta de tres zonas principales (que aparecen enumeradas en las imágenes), cada una con funciones específicas para asegurar la estabilidad, integración del servomotor y la gestión de cables.

1



3

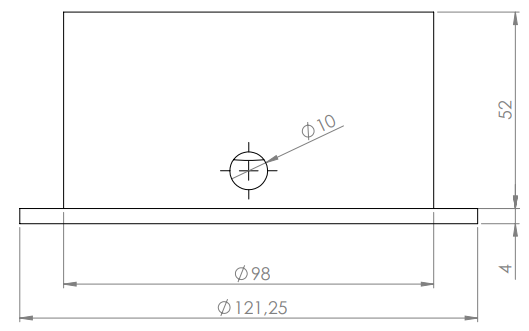
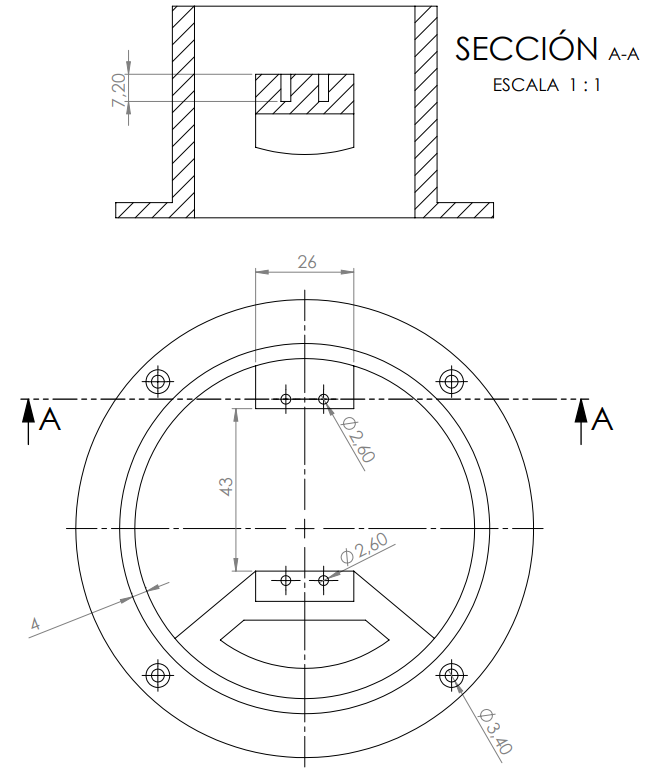
2

1- *Zona de fijación:* En esta área se encuentran cuatro agujeros estratégicamente ubicados para la fijación de la base a una superficie de madera, utilizando tornillos de 3.4 mm de diámetro y más de 4 mm de longitud. Estos tornillos garantizan una sujeción robusta, proporcionando estabilidad al conjunto del brazo robótico.

2- *Zona de montaje del servomotor*: Aquí se dispone de una superficie específicamente diseñada para el montaje del servomotor MG996R. Cuatro agujeros, con un diámetro de 2.6 mm y una longitud de 7.2 mm, permiten fijar el servomotor de manera segura. La precisión en las medidas asegura un ajuste perfecto del servomotor.

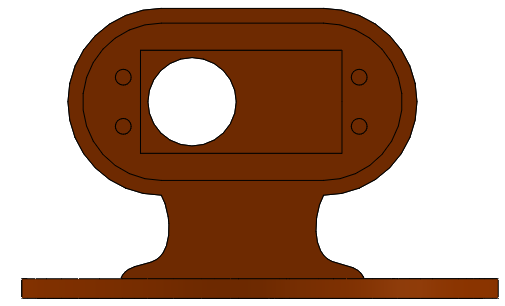
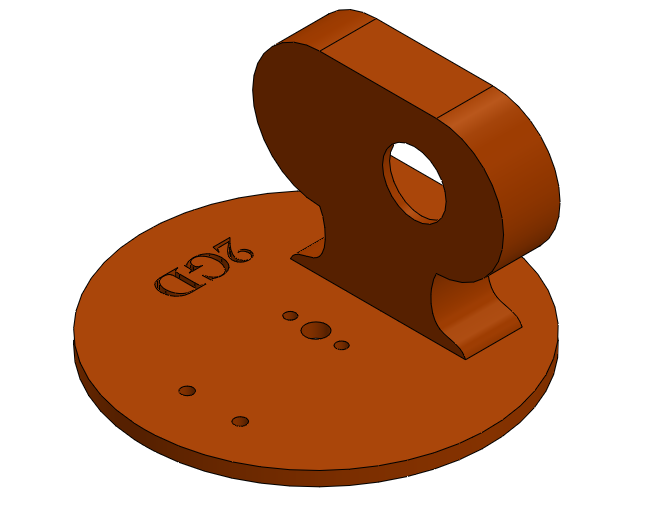
3- *Salida de cables:* Esta zona consiste en un orificio estratégicamente ubicado en el cilindro de la base para permitir el paso de los cables jumper. Dado que las partes superior e inferior de la base estarán cerradas, esta salida de cables es crucial para gestionar los cables que se conectarán a la placa y serán controlados por el Arduino UNO.

A continuación, se detalla el dibujo técnico de la base del brazo robótico. Todas las unidades representadas están en milímetros (mm).



2.1.2. Hombro

El diseño del hombro del brazo robótico presenta cuatro zonas destacadas, cada una con características específicas que contribuyen a su funcionamiento y rendimiento.



4

3

2

1

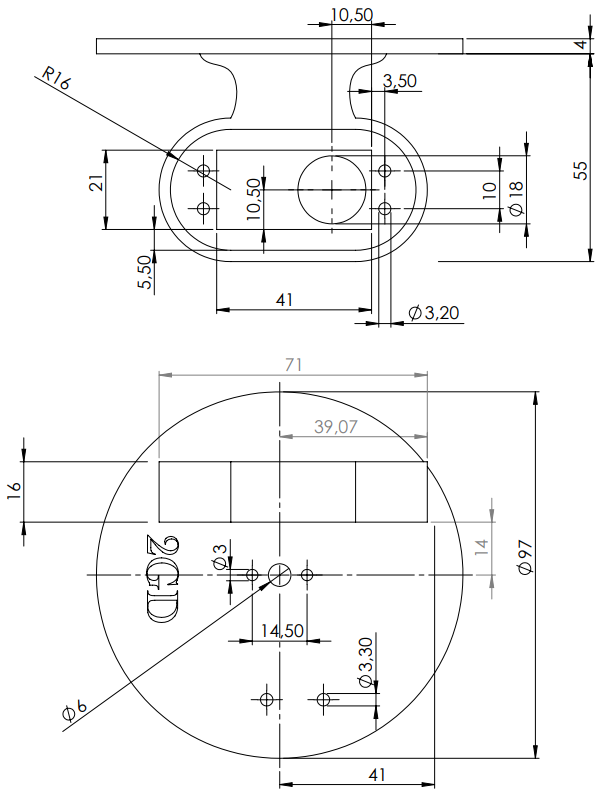
1- *Identificación de los integrantes del proyecto:* En esta zona se encuentran las iniciales de los integrantes del proyecto, Gastón Díaz y Guillermo Duarte. Para simplificar la representación, se utiliza la expresión matemática "2GD", que combina las iniciales de ambos colaboradores.

2- *Conexión con la pieza anterior (base):* En esta zona se establece la conexión con la pieza anterior, la base del brazo robótico. Aquí se encuentra el punto de unión donde se conecta el servomotor MG996R anteriormente mencionado, que, al activarse, permite el movimiento de rotación de toda la estructura del hombro.

3- *Montaje del segundo servomotor 2.1*: Esta área está destinada al montaje del segundo servomotor MG996R que cooresponde a la articulacion , el cual también se sujeta con cuatro tornillos. Además, presenta una forma de encaje rectangular que garantiza un ajuste firme y seguro en esta sección del hombro.

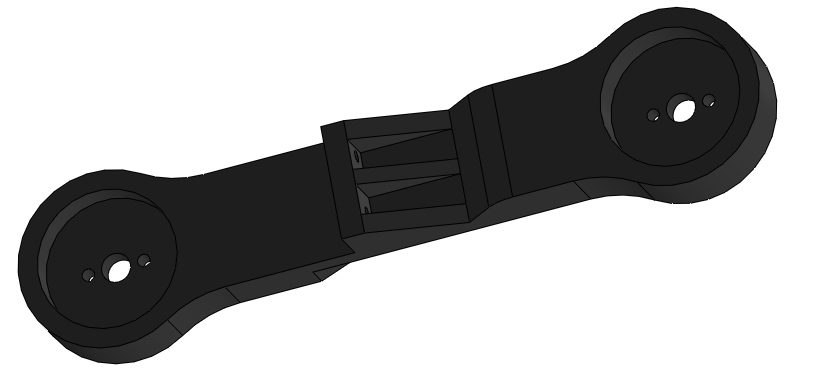
4- *Montaje del segundo servomotor 2.2*: En esta sección del diseño se han integrado dos perforaciones específicas para la instalación del segundo servomotor MG996R, que está destinado a la articulación . Este servomotor estará alineado colinealmente con el servomotor 2.1, aunque con una dirección de giro opuesta. Tal disposición permite que ambos servomotores funcionen de manera sincronizada, lo cual es esencial para mejorar la rigidez del sistema y compensar la deficiencia de par motor del servomotor 2.1. Adicionalmente, esta configuración optimiza la obtención de mayores velocidades y otros beneficios operativos, incrementando la eficiencia global del mecanismo.

A continuación, se detalla el dibujo técnico del hombro del brazo robótico. Todas las unidades representadas están en milímetros (mm).



2.1.3. Brazo

El diseño del brazo del brazo robótico presenta cuatro zonas distintivas.



4

3

2

1

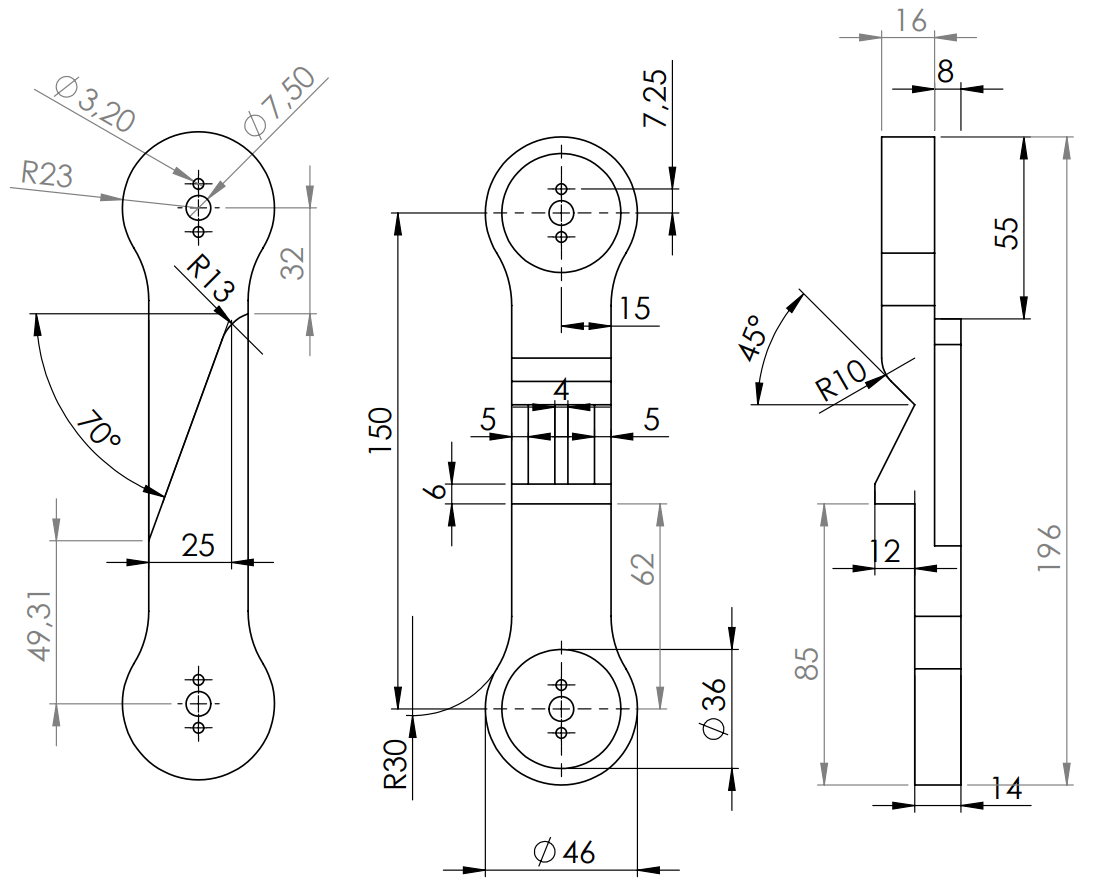
1- ***Unión con la pieza anterior*:** Esta zona es crucial ya que aquí se encuentra el eje del servomotor número 2 proveniente de la pieza anterior. Este eje permite el movimiento articulado de este eslabón específico del brazo.

2- ***Unión con la pieza siguiente:*** En esta zona se establece la conexión con la pieza siguiente del brazo robótico. Lo que garantiza el movimiento de rotación de esta articulación adicional.

3- ***Chaflán con redondeo para ampliar el rango de movimiento:*** Esta área presenta un chaflán con redondeo diseñado específicamente para permitir movimientos más cercanos a la base del robot. Este detalle es crucial para asegurar que el brazo robótico tenga un amplio rango de movimiento en el espacio de trabajo necesario.

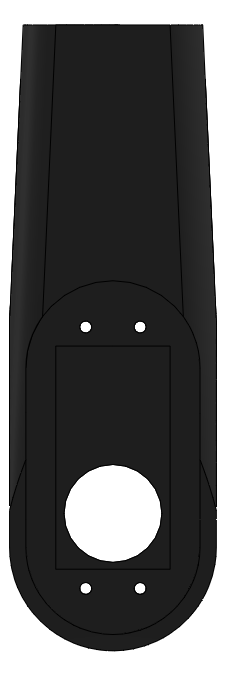
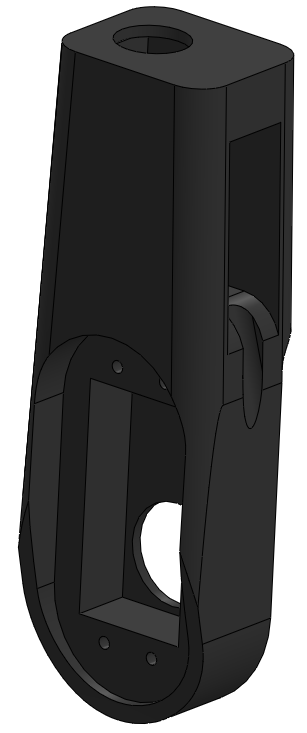
4- ***Preparación para una posible segunda pieza y un segundo servomotor opcional:*** Aquí se encuentran dos agujeros destinados para atornillar una pieza que podría unir este eslabón con el hombro, lo cual se podría utilizar para el caso de usar un segundo servomotor en el hombro.

A continuación, se detalla el dibujo técnico del brazo del brazo robótico. Todas las unidades representadas están en milímetros (mm).



2.1.4. Antebrazo

El antebrazo del brazo robótico, presenta dos zonas destacadas que contribuyen a su funcionamiento y estructura



2

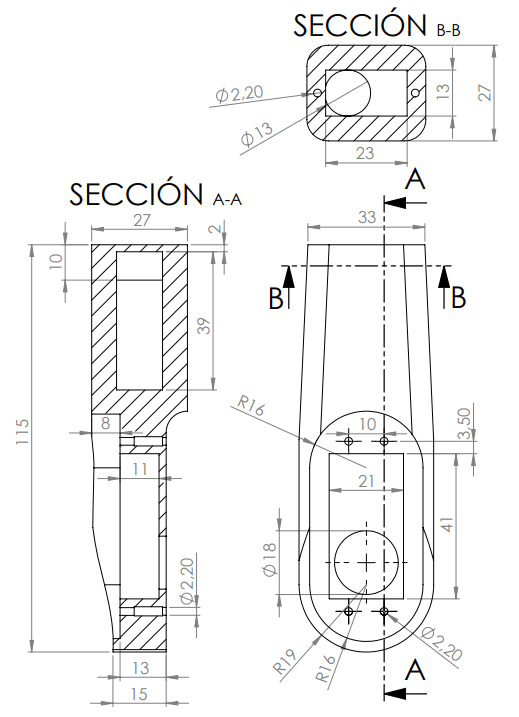
1

1- *Zona de montaje del servomotor MG996R*: Esta área está diseñada para alojar y fijar nuestro penultimo servomotor MG996R, el cual está vinculado con la pieza anterior, el brazo (el mismo servomotor que se encuentra en la zona 2 del brazo se monta aquí para proporcionar movimiento al antebrazo).

2- ***Zona de montaje de mini servos SG90:*** Aquí se encuentra el espacio destinado para el mini servo SG90. Aunque inicialmente se contempló la instalación de este servo para agregar un grado de libertad adicional al brazo robótico, en este caso particular no se están utilizando. En su lugar, esta parte del antebrazo se unirá directamente con la muñeca utilizando dos tornillos, lo cual es suficiente para los movimientos requeridos y elimina la necesidad de usar un mini servo SG90.

A continuación, se detalla el dibujo técnico del antebrazo del brazo robótico.

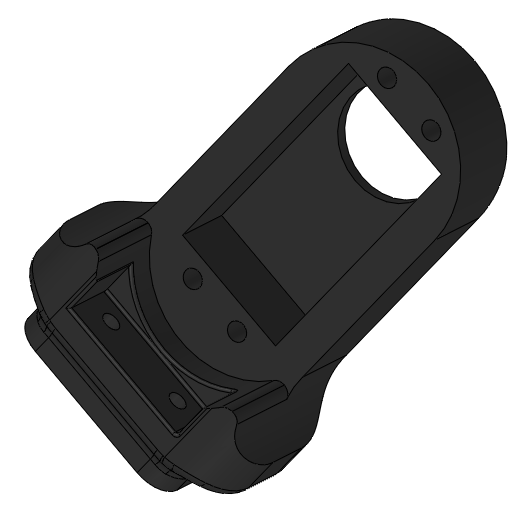
Se destaca que se han realizado cortes en las secciones A-A y B-B para permitir una visualización detallada de las medidas de algunos agujeros y para mejorar la comprensión de la parte donde estaría ubicado el servomotor SG90.



2.1.5. Muñeca

La muñeca del brazo robótico, presenta tres zonas distintivas que son cruciales para su funcionamiento.

2



3

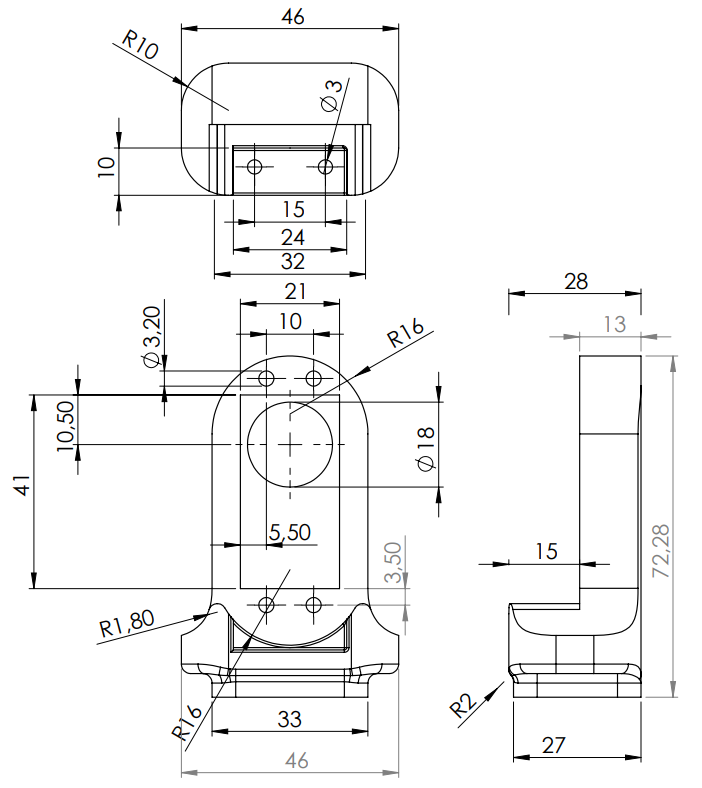
1

1- ***Zona de conexión con la pieza anterior:*** Esta zona es fundamental ya que aquí se conecta firmemente con la pieza anterior, el antebrazo. Ambas piezas están unidas con dos tornillos que aseguran una conexión sólida y estable, permitiendo que el antebrazo y la muñeca funcionen como un único eslabón.

2- ***Zona de montaje del último servo MG996R:***Aquí se encuentra el espacio designado para el montaje del último servo MG996R. Esta zona está especialmente diseñada con una cavidad rectangular que permite un ajuste preciso del servo. Además, se asegura con cuatro tornillos para garantizar una fijación segura y estable del servo en la muñeca.

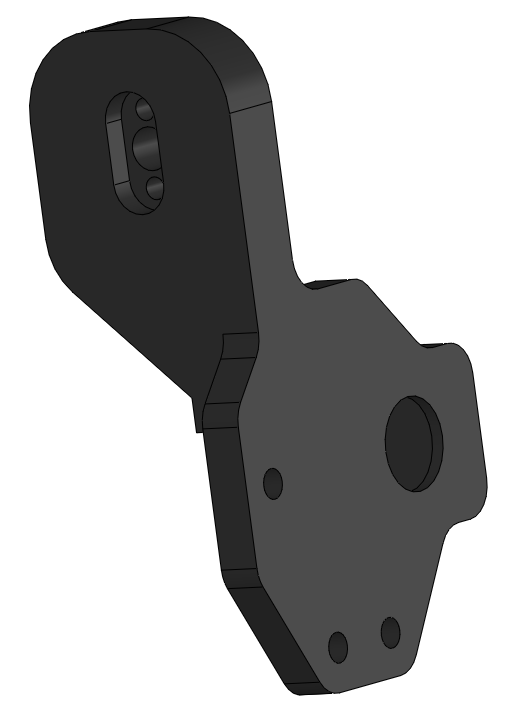
3- ***Zona con redondeo para movimiento libre de la mano:*** Esta zona presenta un redondeo a medida que facilita un movimiento completamente libre para la mano, que es la siguiente pieza en la secuencia del brazo robótico. Este diseño sin restricciones permite que la mano tenga un rango completo de movimiento necesario para ejecutar las tareas asignadas.

Tambien se proporciono un dibujo técnico detallado que incluirá todas las medidas en milímetros (mm).



2.1.6. Mano

La mano del brazo robótico, presenta cuatro zonas cruciales para su funcionamiento y complejidad.



3

4

2

1

1- *Zona de montaje del utlimo servo MG996R:* En esta área se encuentra el espacio designado para el montaje del ultimo servo MG996R. Esta zona está conectada con la pieza anterior, la muñeca, (específicamente con la zona 2 de la muñeca). Este servo proporciona control y movimiento a la mano.

2- *Orificio para la vinculación con otra pieza*: Se destaca un orificio de 3.5 mm de diámetro en esta zona. Este orificio está diseñado para la inserción de bulones con tuercas que se utilizarán para vincular esta pieza con otra que se verá más adelante en el ensamblaje. Esta pieza adicional tiene una especie de engranaje que se conectará con la mano para añadir funcionalidades específicas al sistema.

3- *Zona de vinculación con otra pieza mediante bulones y tuercas M4*: En esta área se encuentran puntos de fijación para bulones y tuercas M4 que servirán para vincular la mano con otra pieza más pequeña. A diferencia de la pieza mencionada en la zona 2, esta no presenta una forma de engranaje, sino que se trata de una conexión más simple pero igualmente importante para el funcionamiento de la mano.

4- *Zona de montaje de un servo SG90*: Aquí se ubica el espacio destinado para el montaje del servo SG90. Este servo se ajusta en esta parte mediante dos tornillos y una cavidad diseñada para que encaje perfectamente. Se destaca que también está vinculada con otra pieza que tiene forma de engranaje, pero en este caso, se vincula directamente con el servo SG90.

Este servo es justamente el que abre y cierra el gripper de nuestro brazo robotico.

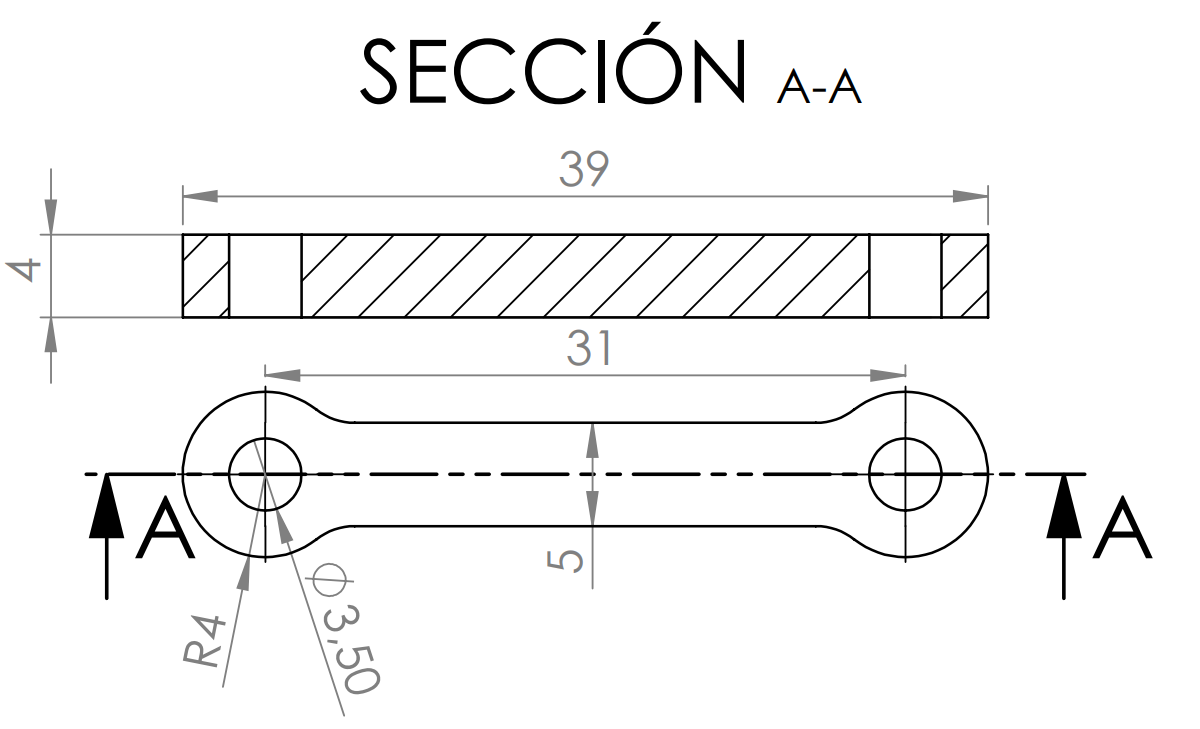
A continuación, se detalla el dibujo técnico de la mano del brazo robótico. Todas las unidades representadas están en milímetros (mm).

Se destaca que se ha realizado un corte en la sección A-A para permitir una visualización detallada de las medidas de algunos agujeros que de otra manera no serían visibles. Este corte proporciona una vista interna que facilita el análisis y la comprensión del diseño.



2.1.7. Union de pinza parte 1

La séptima pieza, de la cual se hicieron cuatro idénticas, consiste en una pieza pequeña y sencilla con dos partes destacadas



2

1

1- *Primera zona*: Se encuentra un agujero de 3.5 mm de diámetro en esta zona. Este agujero se vincula con la pieza de la mano (específicamente con la zona 3 de la mano donde van los bulones).

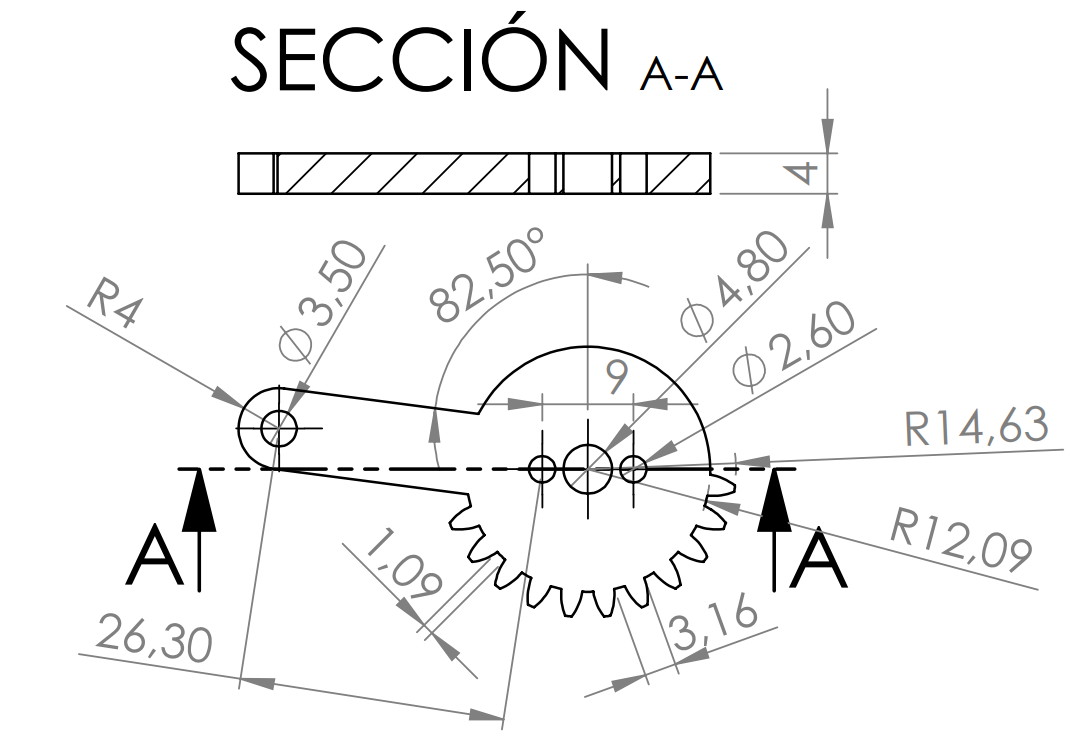
2- *Segunda zona:* En el otro extremo de la pieza se encuentra otro agujero de 3.5 mm de diámetro. Este agujero se vincula con otra pieza en forma de engranaje que se verá más adelante en el ensamblaje. Esta conexión permite la integración de esta pieza en el sistema mecánico del brazo robótico, agregando funcionalidades específicas al conjunto.

Tambien se puede observar el dibujo tecnico, para poder ver las dimensiones de la misma en milimetros (mm).

2.1.8. Union de pinza parte 2

La octava pieza es una pieza pequeña, pero algo más compleja debido a que presenta una forma de engranaje con once dientes. En el dibujo técnico, donde se muestran las medidas en milímetros de este engranaje, se destacan los siguientes detalles: el diámetro externo es de 14.63 mm, el diámetro interno es de 12.09 mm, el paso del engranaje es de 3.16 mm y el hueco entre dientes es de 1.09 mm.

Ahora bien, esta pieza presenta tres zonas importantes.



1

3

2

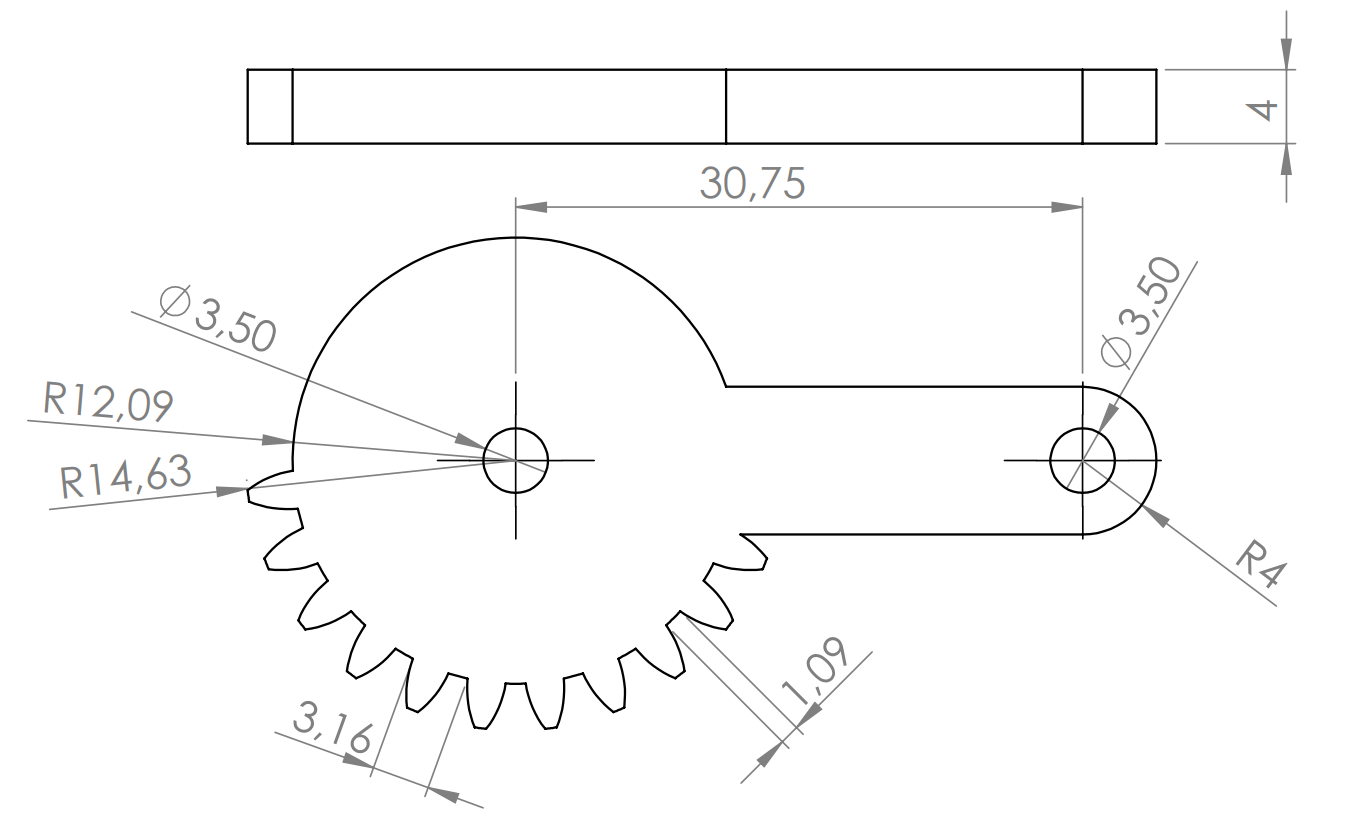
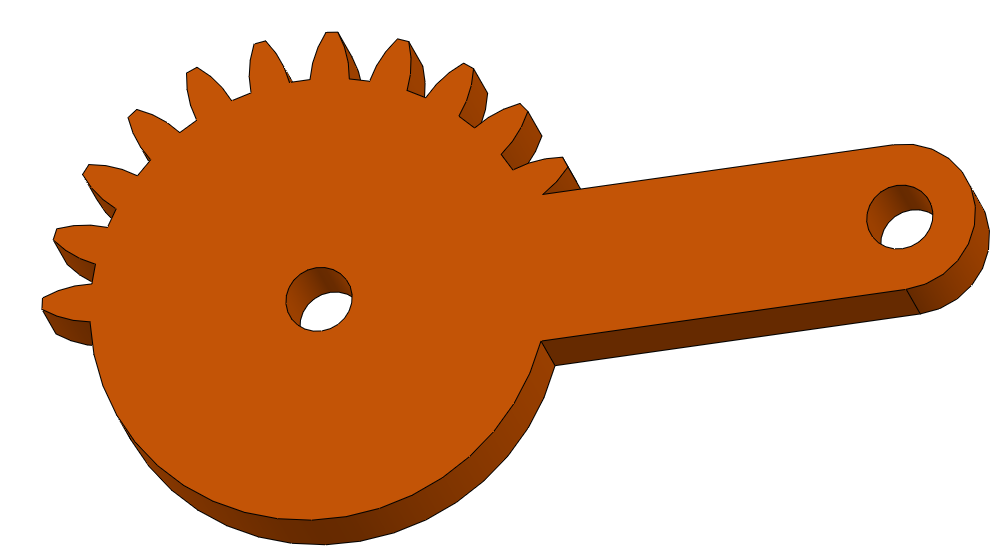
1- *Conexión con el servo SG90*: Esta zona está diseñada para ser vinculada con el último servo SG90 (ubicado en la zona 4 de la mano). De esta manera, el movimiento proporcionado por este servo permite el movimiento del engranaje, lo que a su vez hace que se abran o cierren los dedos de la mano.

2- *Dientes del engranaje*: Aquí se encuentran los once dientes del engranaje de media circunferencia. Estos dientes permiten el acoplamiento y la transmisión de movimiento con otra pieza, proporcionando un control preciso sobre el movimiento de los dedos del brazo robótico.

3- *Vinculación con los dedos*: En esta área se coloca un bulón y una tuerca. Este bulón y tuerca se vinculan con otra pieza que forma parte de los dedos, la cual se detallará más adelante en el ensamblaje. Esta conexión permite la integración y el funcionamiento coordinado de las partes de la mano del brazo robótico.

2.1.9. Unión de pinza parte 3

La novena pieza es similar a la anterior, pero con algunas diferencias en las zonas de conexión



3

1

2

1- *Vinculación con bulón y tuerca*: Esta zona está diseñada para ser vinculada con un bulón y una tuerca. Esta conexión proporciona estabilidad y permite la integración adecuada de esta pieza en el mecanismo del brazo robótico.

2- *Dientes del engranaje*: Al igual que en la pieza anterior, esta zona presenta dientes de engranaje que se acoplan con la zona correspondiente de la otra pieza con engranajes (Específicamente la zona 2 de la unión de pinza parte 2). Esta conexión permite la transmisión de movimiento entre las dos piezas, facilitando el control coordinado de los movimientos de los dedos.

3- *Vinculación con el dedo opuesto:* En esta zona se coloca un bulón y una tuerca que se vinculan con el dedo opuesto de la pieza anterior. Mientras que uno de los engranajes maneja el dedo izquierdo, este controla el dedo derecho, permitiendo así una manipulación equilibrada y coordinada de los dedos.

2.1.10. Dedos

Los dedos son la última pieza que consta de cuatro zonas importantes y se hicieron 2 piezas idénticas para representar el dedo izquierdo y el dedo derecho, aquí está la descripción de las zonas importantes junto con las medidas detalladas del dibujo técnico.



1

2

4

3

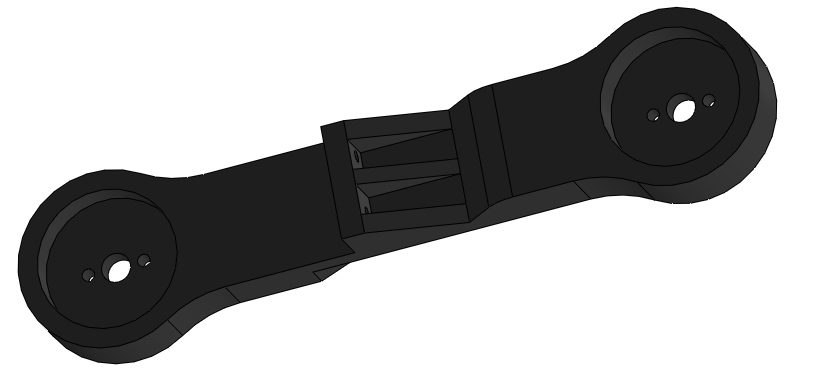
1- ***Zona de vinculación con bulón y unión de pinza (parte 3 o parte 2):*** Esta zona presenta un orificio para la vinculación con un bulón y se conecta con la parte 3 o parte 2 de la unión de pinza, dependiendo de si es el dedo izquierdo o derecho respectivamente. Esta área es ligeramente más gruesa para asegurar que las partes 2 y 3 de la unión de pinza queden completamente paralelas entre sí, mejorando así el movimiento de apertura y cierre del gripper.

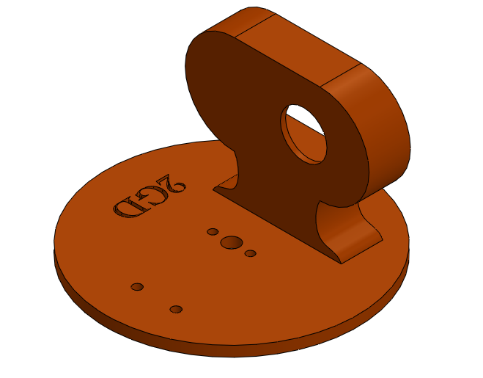
2- *Zona de vinculación con bulón y unión de pinza (parte 1):* Similar a la anterior, esta zona también está diseñada para conectarse con un bulón y una tuerca. En este caso, se vincula con la parte 1 de la unión de pinza (específicamente con la zona 2 de esa pieza). Esta conexión asegura la integración adecuada del dedo con el mecanismo de la unión de pinza, permitiendo un movimiento coordinado.

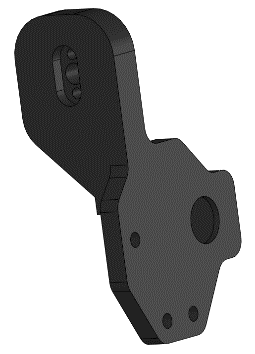
3- ***Zona con pequeña pared para estabilidad del gripper:*** En esta zona, se encuentra una pequeña pared que juega un papel crucial cuando el gripper se cierra sobre una pieza de ajedrez. Esta pared evita que la pieza se caiga hacia adelante o hacia atrás durante ciertos movimientos, proporcionando estabilidad adicional al agarre.

4- ***Zona cilíndrica para agarre seguro de las piezas de ajedrez:*** En el extremo del gripper, esta zona cilíndrica está diseñada para encajar perfectamente con el cuello de las piezas de ajedrez. Esto asegura un agarre firme y preciso de las piezas, evitando que se resbalen o se caigan durante las operaciones de agarre y movimiento.

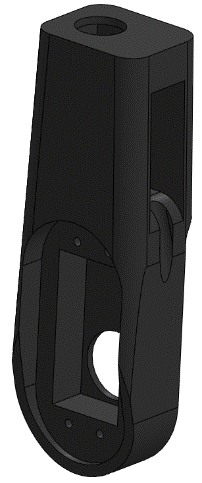
A modo Genérico y recopilando todas las piezas que conforman el brazo robot, se presenta a continuación una tabla que resume la cantidad y descripción de cada componente, facilitando asi su visualización:

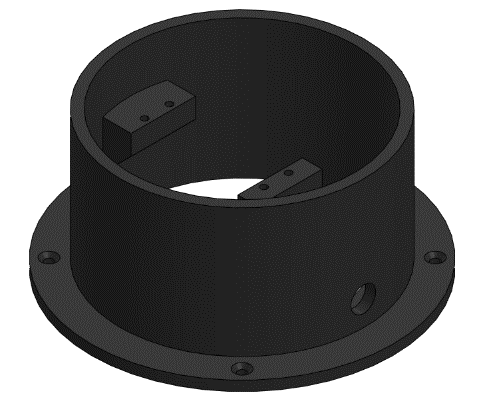


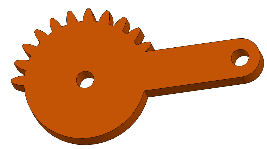












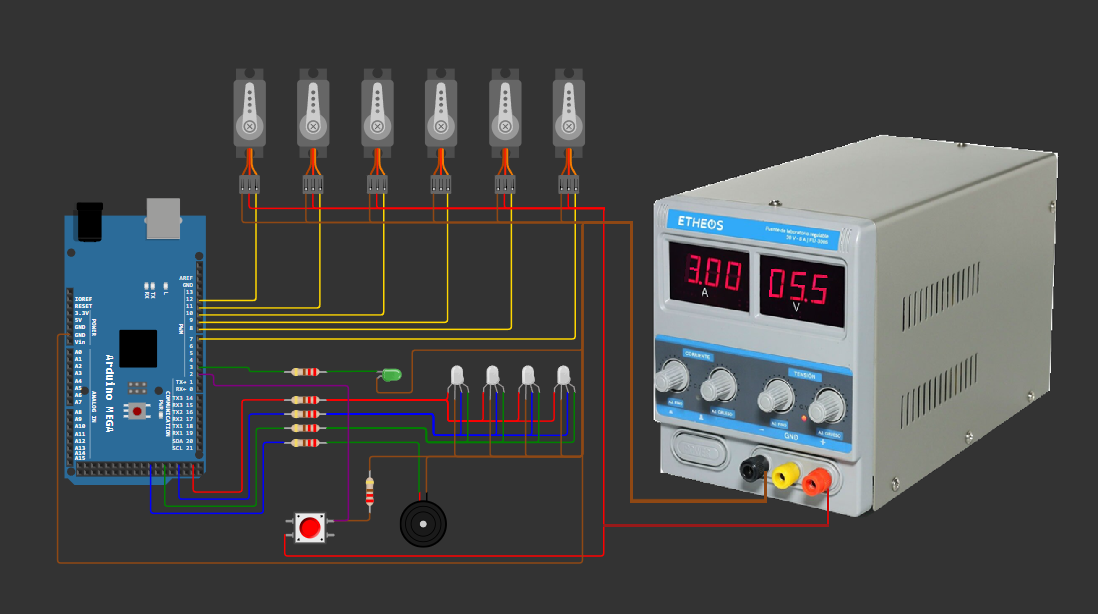


|  |  |  |  |
| --- | --- | --- | --- |
| **Numero** | **Componente** | **Descripción** | **Cantidad** |
| **1** | **Base** | **Soporte estable que sostiene todo el sistema.** | **1** |
| **2** | **Hombro** | **Permite el movimiento del brazo en diferentes ángulos.** | **1** |
| **3** | **Brazo** | **Conecta el hombro con el antebrazo, facilitando el movimiento.** | **1** |
| **4** | **Antebrazo** | **Parte que conecta el brazo con la muñeca.** | **1** |
| **5** | **Muñeca** | **Proporciona flexibilidad y movimiento del gripper.** | **1** |
| **6** | **Mano** | **El efector final que interactua con el tablero.** | **1** |
| **7** | **Unión de Pinza parte 1** | **Primera parte de la pinza que permite el agarre.** | **1** |
| **8** | **Unión de Pinza parte 2** | **Segunda parte que complementa el mecanismo de agarre.** | **1** |
| **9** | **Unión de Pinza parte 3** | **Tercera parte que complementa el mecanismo de agarre.** | **4** |
| **10** | **Dedos** | **Elemento que permite un agarre preciso y controlado de las piezas.** | **2** |

# 3.Diseño del esquemático eléctrico

3.1. Diseño del Esquemático Eléctrico en Wokwi

El circuito se diseñó utilizando Wokwi, una plataforma en línea que permite simular proyectos de electrónica y programación de microcontroladores, como Arduino, de manera interactiva. Wokwi facilita la creación de circuitos virtuales, ofreciendo una amplia biblioteca de componentes que se pueden conectar fácilmente. Esta herramienta fue fundamental para probar y optimizar el funcionamiento del circuito antes de implementarlo físicamente, asegurando una mejor planificación y evitando errores en la etapa de construcción.



Q2.2

Q1

Q2.1

Q3

Q4

Gripper

220 Ω

GND

**Componentes del Circuito Electrónico:**

* **Microcontrolador Arduino Mega:** Controla todos los componentes del circuito y gestiona la lógica del proyecto.
* **Servomotores MG996R:** Utilizados en las articulaciones Qn, las salidas de los pines son 7, 8, 9, 10 y 11.
* **Servomotor SG90:** Utilizado para el gripper, permitiendo abrir y cerrar, salida en pin 13.
* **Resistencias:** 6 Resistencias de 10 kΩ utilizadas para limitar la corriente en el circuito y proteger los componentes.
* **LEDs RGB:** Indicadores de estado o señales visuales, las salidas de los pines son 22, 26 y 30.
* **LED Verde:** Indicador de funcionamiento, salida en el pin 3.
* **Pulsador:** Un Pulsador de 4 patas, permite la interacción del usuario y el control manual del circuito, salida en el pin 2.
* **Buzzer:** Permite emitir sonidos para alertas o señales, controlado a través del pin 34.
* **Fuente de Laboratorio:** Fuente regulable de 5.5 Volt CC, proporciona la alimentación necesaria para el circuito.

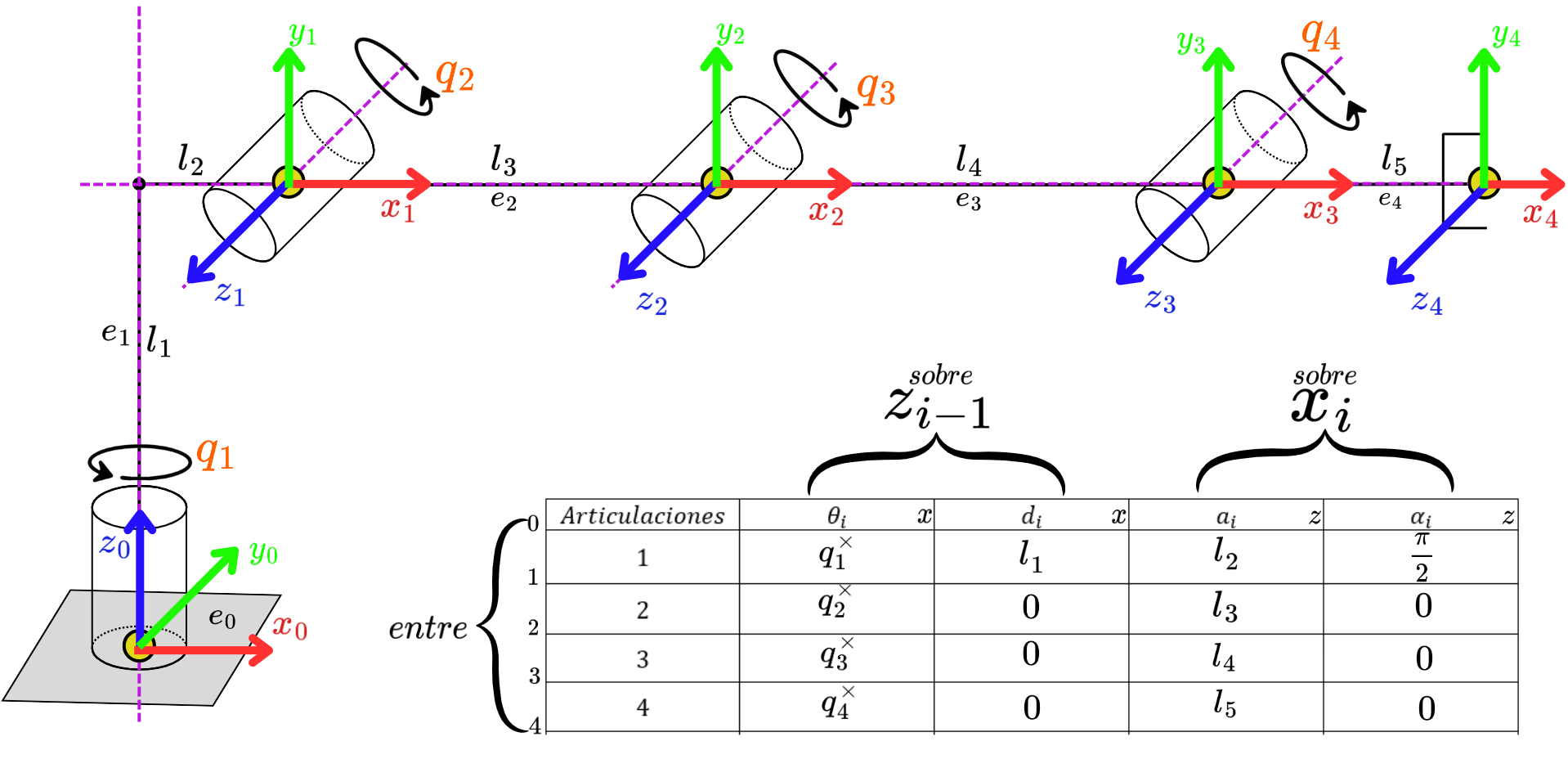
El circuito se llevó a cabo en una plaqueta de cobre perforada, debido a que el diseño del circuito es bastante simple. Este tipo de plaqueta permite realizar conexiones de manera eficiente y rápida, sin la necesidad de recurrir a un diseño en software o a una placa de circuito impreso (PCB) más compleja. La simplicidad del conexionado facilita la construcción y modificación del circuito según se vaya actualizando.

# 4.Cinemática directa

Nuestro robot de 4 GDL (RRRR):



4.1. Cinemática directa por método de Denavit-Hartenberg



Obtenemos todas las matrices homogéneas con la ayuda de la tabla de parámetros de Denavit-Hartenberg.

Recordando que:

Reemplazamos para

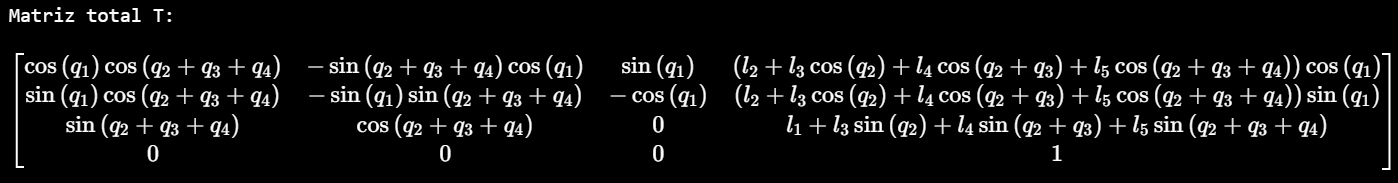
Reemplazamos para

Reemplazamos para

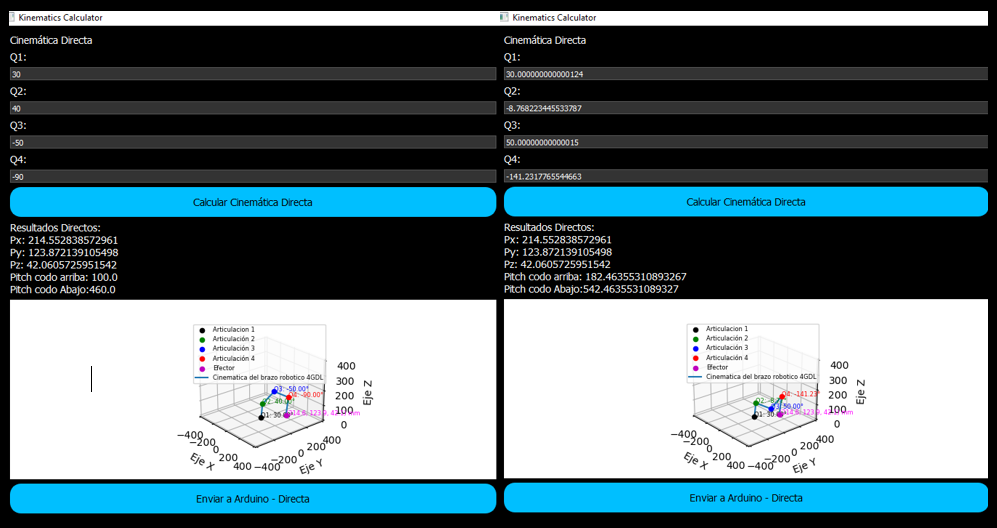
Reemplazamos para

La matriz T es:

Matriz de obtenida con ayuda de Python



El siguiente programa, desarrollado en Python, permite simular diversas configuraciones de articulaciones que alcanzan una misma posición del extremo del manipulador robótico, mostrando así las diferentes alternativas disponibles. En la imagen de la derecha, se puede observar que q3​ tiene la configuración "codo arriba", lo que hace que q2 deba quedar hacia abajo, ya que su cálculo depende de q3​. Por otro lado, en la imagen de la izquierda, q3 presenta la configuración "codo abajo", lo que implica que q2 debe posicionarse hacia arriba, de acuerdo con la relación mencionada entre ambos ángulos. De este modo, se puede observar cómo ambas configuraciones logran alcanzar la misma posición y orientación del extremo del manipulador.



# 5.Cinemática inversa

5.1. Introducción

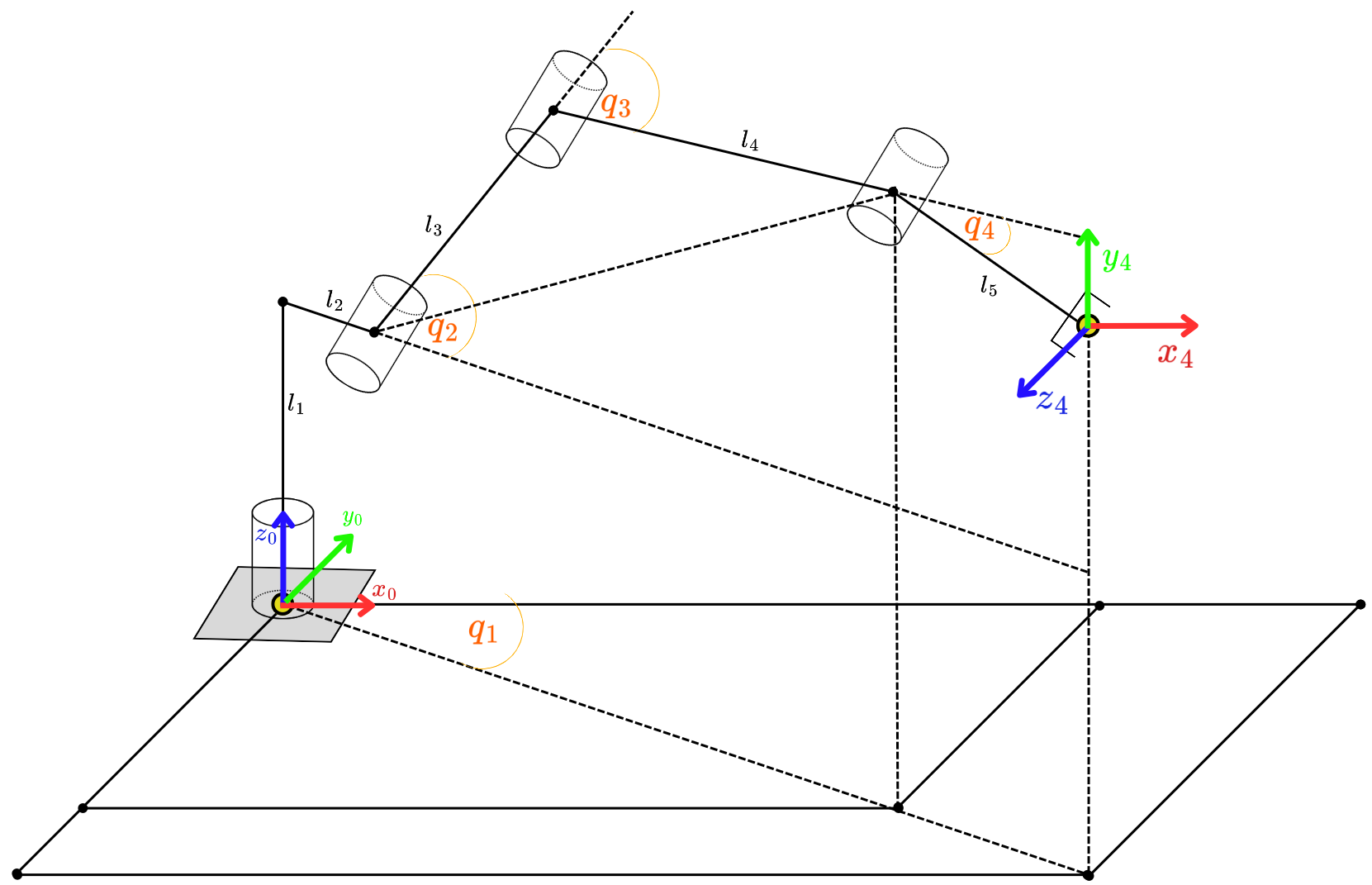
La cinemática inversa es una técnica fundamental en robótica que se utiliza para determinar las posiciones de las articulaciones de un robot necesarias para alcanzar una posición deseada en el espacio. A pesar de que el objetivo es el mismo, existen múltiples métodos para resolver este problema y varias configuraciones posibles para cada brazo robótico, lo que permite alcanzar una posición deseada mediante diferentes combinaciones de ángulos de las articulaciones.

En este informe, se abordará la cinemática inversa utilizando tanto método geométrico como analítico. Comenzaremos con el análisis geométrico, que involucra la representación gráfica del sistema en una perspectiva isométrica para facilitar la comprensión del proceso paso a paso. Esto permitirá interpretar mejor los resultados y los dibujos incluidos en el informe.

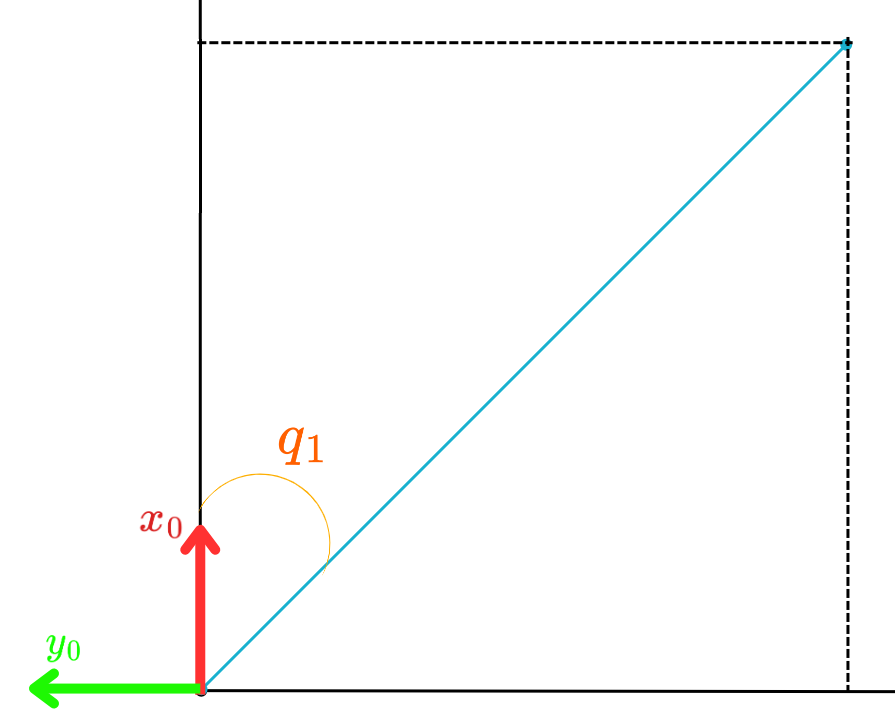
Posteriormente, aplicaremos el método analítico para calcular las coordenadas articulares necesarias. Dados los valores de las coordenadas y la orientación del brazo, determinaremos las configuraciones articulares que permitirán al brazo robótico alcanzar la posición y orientación deseadas.

A través de estos métodos, se explorarán las distintas soluciones posibles para la cinemática inversa y se evaluarán las implicaciones de cada enfoque, proporcionando una visión integral de cómo se puede lograr el control preciso de los brazos robóticos.

5.2. Método geométrico



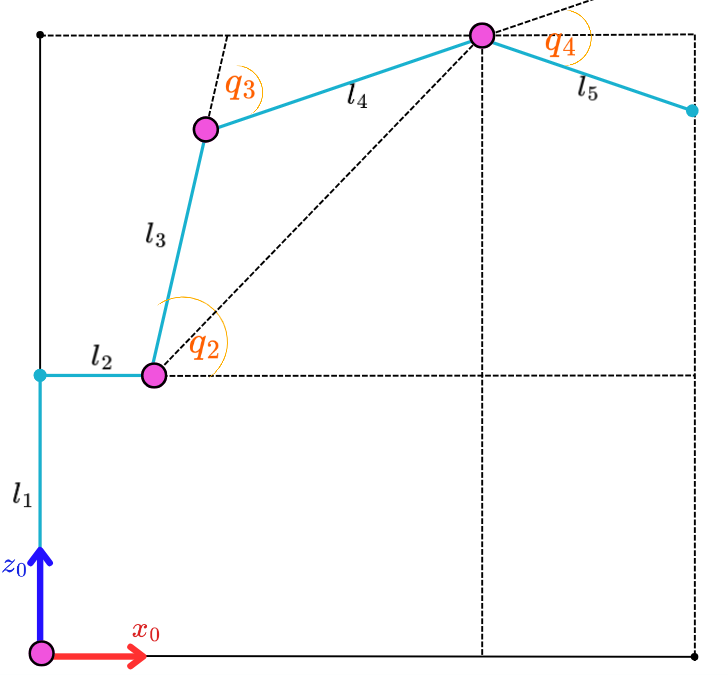
Buscamos la variable articular y analizamos el plano



Como podemos observar, podemos aplicar Pitágoras para obtener la hipotenusa formada en ese triangulo rectángulo, que representa el trayecto desde la base hasta el efector del brazo.

Por razones trigonométricas obtenemos la variable articular (SOHCAHTOA)

Buscamos las variables articulares en el plano



El ángulo es el “ángulo de cabeceo” o “pitch”, el cual se mide respecto al eje horizontal (este es un dato que conoce el usuario, así como también los valores de )

Ahora obtendremos los valores de y

Primeramente, buscaremos la hipotenusa usando el teorema de Pitágoras.

Ahora intentaremos buscar los ángulos incognitos, en este caso primeramente , con el método de razones trigonométricas (SOHCAHTOA).

Ahora hacemos uso del teorema del coseno y algunas propiedades trigonométricas para obtener y a su vez obtener

Ahora también se sabemos que es igual a lo siguiente.

Según la siguiente propiedad del coseno, podemos simplificar aún más la expresión anterior

Entonces la expresión simplificada queda como:

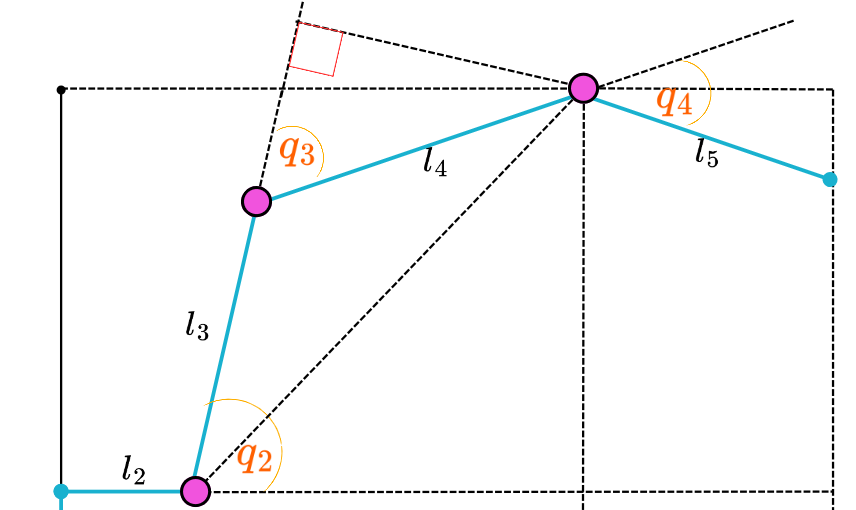
También conocemos la siguiente propiedad trigonométrica, la cual como es igual a 1, no afectaría la igualdad.

En nuestro caso podemos aprovechar esta propiedad para dejar expresado el en función de un arcotangente.

El indica que posee solución doble, y para expresar el en función de arcotangente hacemos lo siguiente:

Ahora podemos formar un triángulo rectángulo un poco difícil de ver.

Debido a que ya obtuvimos ya podríamos hacer uso del mismo.



Ahora buscaremos el otro Angulo incognito , mediante razones trigonométricas (SOHCAHTOA).

Entonces ahora mismo ya podemos obtener

Como bien sabemos, teóricamente la suma de todas las variables articulares nos dará como resultado el total del ángulo final. (Ver imagen para comprender mejor con un ejemplo)

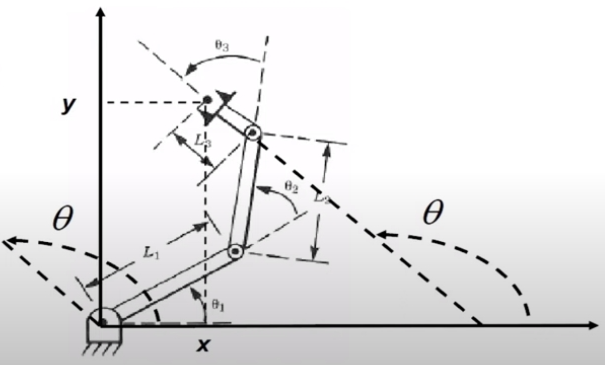
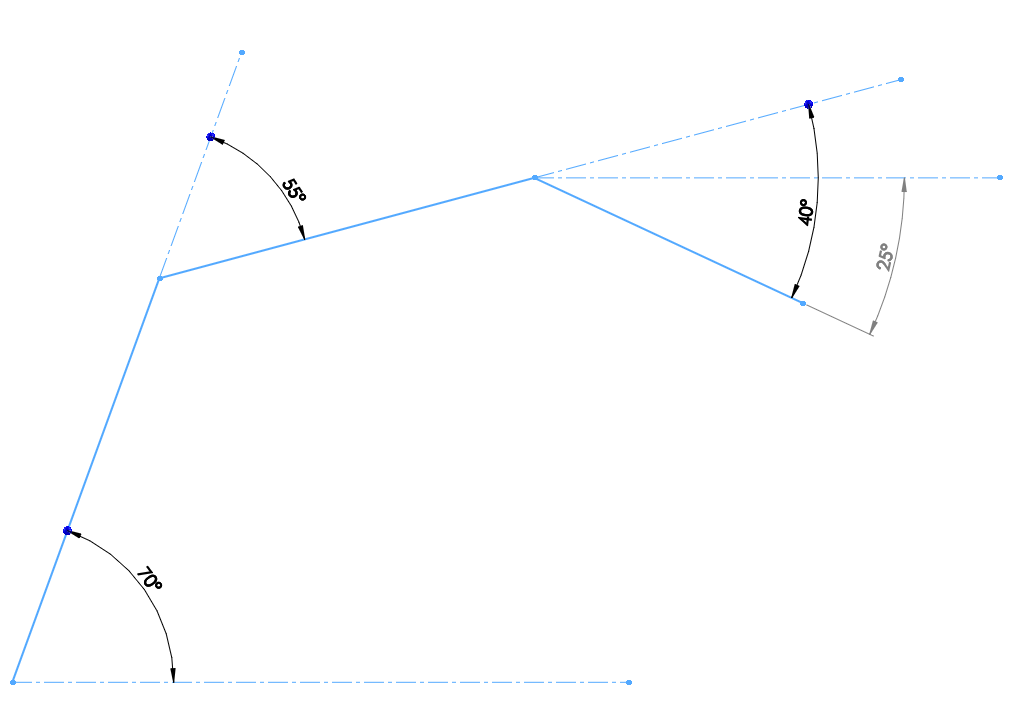


Imagen de ejemplo

Sabiendo esto, podemos expresar lo siguiente para la variable

Para verificar que la expresión de se cumple, decidimos poner valores aleatorios en SolidWorks, simulando las variables articulares, con la misma posición que empezamos a hacer el dibujo.



Verifica la expresión al cumplirse la igualdad, por lo tanto, la cinemática inversa, del robot del robot de 4GDL fue resuelta únicamente utilizando el método geométrico, y los datos de ()

Esta expresión siempre se cumplirá cuando este por debajo de la horizontal.

Si está por encima de la horizontal la expresión seria:

O también:

Si está por encima de la línea colineal a la expresión seria:

En el cual el valor de seria expresada en número negativo

5.3. Método analítico

Buscamos las matrices de transformación homogénea inversas que se obtuvieron por la cinemática directa.

Procedemos a obtener la primera variable articular:

Ahora conformaremos de la siguiente manera ecuaciones que nos ayudaran en el proceso de obtención más adelante:

Ahora buscamos la ecuación formada en (1,4):

Ahora conformaremos de la siguiente manera ecuaciones que nos ayudaran en el proceso de obtención más adelante:

Ahora buscamos la ecuación formada en (1,4):

Como siguiente paso busco obtener y de la matriz total.

Con de despejando *:*

Multiplico por a ambos lados de la igualdad:

Ahora con despejo :

Multiplico por a ambos lados de la igualdad:

Igualo ambos términos de obtenidos recientemente a modo de visualización:

Sumo ambas ecuaciones:

Sumar ambas ecuaciones es válido porque la identidad trigonométrica asegura que los términos adicionales se cancelan, simplificando la expresión de ​.

De la MTH total obtengo, además:

Recordando la expresión obtenida anteriormente:

Podemos usar esta misma expresión para encontrar las variables articulares sabiendo que:

Reemplazando obtenemos:

Como se pudo observar lo que se buscó mediante la MTH total obtenida de la cinemática directa fue encontrar los senos y cosenos del ángulo que se quiso hallar para posteriormente obtener el propio ángulo mediante:

Aplicaremos el mismo procedimiento para las demás variables articulares.

Se pudo haber aplicado el mismo procedimiento, pero partiendo de y :

Parto de despejando :

Multiplico por a ambos lados de la igualdad:

Parto de despejando :

Multiplico por a ambos lados de la igualdad:

Igualo ambos términos de a modo de visualización:

Sumo ambas ecuaciones:

De la MTH total obtengo, además:

Reemplazando obtenemos:

“Cabe resaltar que tanto como se podían haber obtenido directamente de la MTH total”, dado que:

“Se aplicó el procedimiento de la forma más larga para entender cómo obtener dichos valores en caso de requerirlo para especificaciones de robot con más grados de libertad el cual será de gran ayuda dado que no saldrían los resultados de forma tan directa”.

Prosiguiendo con los cálculos tenemos el vector de posición:

De pongo de un lado de la igualdad los términos que conozco y del otro los que no conozco:

De esta manera dejo expresado los términos que desconozco igualados a una constante:

Aplico el mismo procedimiento para :

De pongo de un lado de la igualdad los términos que conozco y del otro los que no conozco:

De esta manera dejo expresado los términos que desconozco igualados a otra constante:

En este caso tanto para como para me termina dando lo mismo, por consiguiente, busco en aplicando los mismos pasos.

De pongo de un lado de la igualdad los términos que conozco y del otro los que no conozco:

Por lo tanto, me queda conformado en base a una ecuación estándar conocida donde además podremos aplicar soluciones estándares:

“Cabe destacar que a la hora de reemplazar valores para obtener las variables articulares, pues se cuánto valen los términos los cuales desconozco sus variables articulares que deje a un lado de la igualdad, ya que están igualados a términos que si conozco y por ende a la hora de reemplazar valores no será de problema”:

Ecuaciones que utilizaremos:

Las soluciones para dichas ecuaciones son:

Aplicando las fórmulas mencionadas con anterioridad y reemplazando términos obtenemos:

Aplicando la solución:

Usaremos la solución complementaria que se amolda mejor, codo abajo que corresponde a:

Con es codo arriba por ende me quedara para abajo ya que depende de

Con es codo abajo por ende me quedara para arriba ya que depende de

La configuración de depende directamente de , dado que calculamos en función de los senos y cosenos de :

Por otro lado, ahora uso una de las fórmulas obtenidas en el principio dado que conozco las demás variables y despejo :

Para el cálculo del ángulo ​ en un brazo robótico con estructura RRRR, se observa que la solución directa usando la fórmula estándar de arcos no siempre se adapta correctamente debido a su rango limitado. La función arcos devuelve valores en el intervalo , lo cual no cubre todas las posibles orientaciones del end-effector.

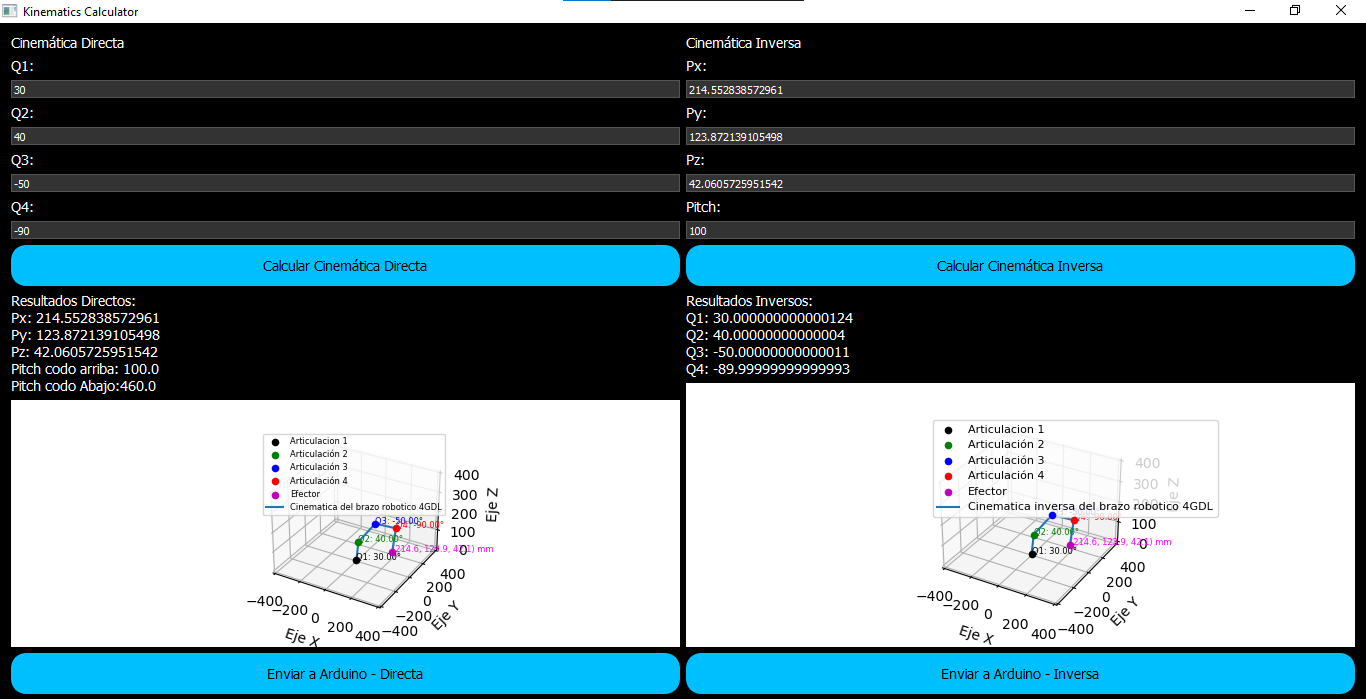
*Ajustes para la solución de :*

Dado que ​ puede tener dos soluciones válidas en el rango de , se ha encontrado que la fórmula ajustada ​ proporciona una solución adicional que se adapta mejor. Este ajuste permite considerar la configuración completa del brazo robótico y asegura que se cubran todas las orientaciones posibles para el end-effector.

Por consiguiente nos queda:

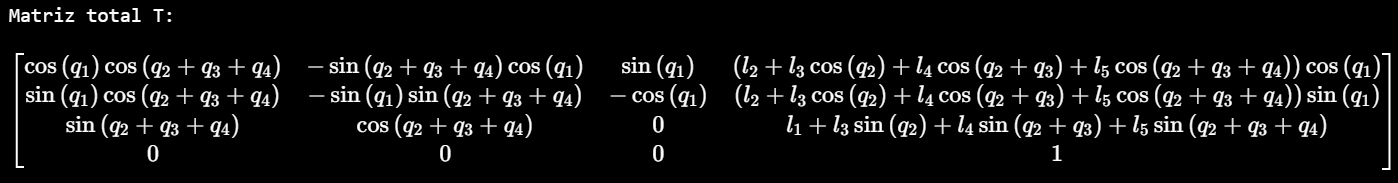
Además, para evitar que el motor gire y llegue al destino por el camino más largo, pues le restaremos 360º para que siga por el camino más corto:

A modo de corroboración, se puede apreciar que la posición y orientación proporcionadas por la cinemática directa, al ser ingresadas en la cinemática inversa, producen como resultado los ángulos de las articulaciones utilizados en la cinemática directa.



# 6.Jacobiano analítico

A partir de la MTH derivo sabiendo que:



Ángulos de Euler:

Igualo con la MTH total:

**(3;1):**

**(3;3):**

Al menos uno de los términos debe ser cero dado que el producto entre ambos términos es cero:

Caso 1:

Caso 2:

Para que el producto sea igual a cero es suficiente con

Por lo tanto:

Por ende, tenemos que:

Quedándonos los siguientes ángulos de Euler:

Una vez obtenido los parámetros necesarios derivo en función de cada variable articular:

**Derivo :**

**Derivo :**

**Derivo :**

Recordando:

**Derivo :**

**Derivo :**

**Derivo :**

Reemplazando los valores obtenemos:

Reemplazamos el jacobiano analítico:

# 7.Algoritmos de Control

En esta sección, se describe el diseño y la implementación de los algoritmos de control que permiten al brazo robot jugar ajedrez de manera autónoma. La programación se ha realizado en Python y Arduino, abarcando diversas funcionalidades críticas para el funcionamiento del sistema.

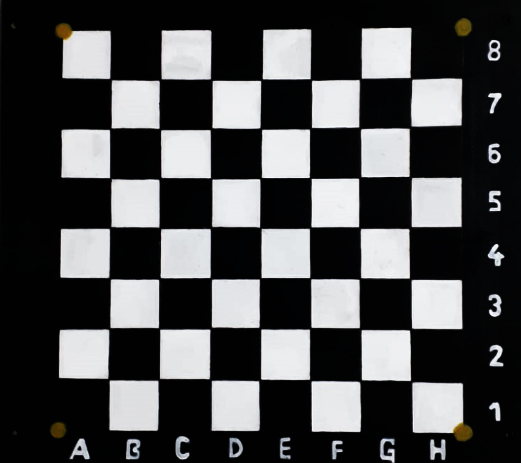
7.1. Detección y Procesamiento del tablero

7.1.1. Inicializacion de la Camara

Para la captura de imágenes del tablero de ajedrez, se utiliza un sistema de cámara vinculado a una computadora portátil mediante la aplicación **"Irium Webcam".** Esta aplicación facilita la conexión del dispositivo móvil, en este caso un celular, a la notebook a través de interfaces de comunicación USB o Wi-Fi. Posteriormente, mediante la ejecución de comandos en el entorno de programación Python, se activa la cámara del dispositivo móvil, permitiendo la adquisición de imágenes en tiempo real. Este procedimiento asegura la transmisión continua de datos visuales necesarios para el análisis y procesamiento subsiguientes del tablero de ajedrez.

7.1.2. Reconocimiento del Tablero

El reconocimiento del tablero de ajedrez se realiza mediante detección por color. Se configuran los parámetros de detección utilizando el método de goteo para ajustar el color específico a detectar. En este caso, se utilizan cuatro círculos amarillos pintados en las esquinas del tablero, donde luego mediante algoritmos en Python identifican las posiciones de los círculos en las esquinas del tablero otorgándome asi las coordenadas en x e y.



Con las coordenadas de estos cuatro círculos amarillos, se determina la geometría del tablero, dividiéndolo en una malla de 8x8 casillas. Para aumentar la precisión y eliminar datos irrelevantes, se recorta la imagen ajustándola al área delimitada por las esquinas detectadas, excluyendo así el contenido externo al tablero.

7.2. Detección de las Piezas

Para identificar el tipo de pieza en el tablero de ajedrez, se realiza un mapeo basado en la posición inicial de cada pieza y su movimiento a lo largo del juego. Aunque la detección de color permite identificar la pieza como verde o azul, no proporciona información específica sobre el tipo de pieza.

Para resolver esto, se utiliza el conocimiento de las posiciones iniciales de las piezas en el ajedrez. A partir de esta información, se realiza un seguimiento de cada pieza durante las jugadas. Se construye una matriz que asigna letras a cada tipo de pieza y su ubicación en el tablero.

A lo largo del juego, se actualizan constantemente estas matrices para reflejar la posición actual de las piezas. Mediante la comparación de la matriz actual con la matriz anterior, y utilizando funciones específicas, se detectan cambios tales como capturas de piezas y movimientos. Este proceso permite mantener un registro preciso del estado del juego y facilita la identificación de las piezas en cualquier momento dado.

7.3. Integración del motor de ajedrez Stockfish

Para la toma de decisiones y la generación de jugadas en el sistema de ajedrez automatizado, se utiliza el motor de ajedrez Stockfish, uno de los motores más reconocidos y utilizados en la comunidad ajedrecística. Este motor se integra con el entorno de programación en Python, permitiendo que todas las decisiones relacionadas con las jugadas sean gestionadas por Stockfish.

La integración se realiza mediante una interfaz que conecta Python con Stockfish, facilitando la comunicación entre ambos sistemas. El motor Stockfish se encarga de evaluar la posición actual del tablero, analizar las posibles jugadas y seleccionar las mejores estrategias basadas en algoritmos avanzados de cálculo y heurística.

Stockfish proporciona recomendaciones de jugadas óptimas y maneja la lógica de decisión en función del estado del juego, mientras que Python orquesta la interacción con el motor y coordina el procesamiento de las decisiones en el contexto del sistema de ajedrez automatizado. Esta integración asegura que el sistema pueda realizar jugadas estratégicas y precisas, optimizando el desempeño en partidas de ajedrez.

7.4. Comunicación Bidireccional entre Phyton y Arduino

7.4.1. Transmision de Comandos de Movimiento a Arduino:

Tras el procesamiento de las jugadas en Python, se envía a Arduino un vector en formato cadena (string) que especifica el movimiento a realizar. Por ejemplo, el vector ['H7', 'H5', '0'] indica un movimiento desde la casilla H7 a H5, con el último dígito 0 señalando que solo se ha realizado un desplazamiento sin captura de pieza.

Python transmite estos comandos a Arduino, que interpreta la información y controla los servomotores del brazo robótico para efectuar el movimiento correspondiente sobre el tablero de ajedrez.

7.4.2. Recepcion de Señales desde Arduino:

Arduino está equipado con un pulsador que el usuario debe accionar tras realizar un movimiento en el tablero. Este pulsador envía una señal a Python para iniciar la captura de una imagen y el análisis de la jugada.

Arduino también maneja LEDs para la detección del tablero y otras funciones del juego. Python envía señales a Arduino para controlar el encendido y apagado de estos LEDs, facilitando la visualización y la detección del tablero.

7.5. Validación de Jugadas Utilizando la Biblioteca Chess de Phyton

#### El sistema de ajedrez automatizado incorpora una función de validación de jugadas mediante la biblioteca chess de Python. Esta biblioteca permite verificar la legalidad de las jugadas realizadas durante el juego, proporcionando una capa adicional de precisión y control.

7.5.1. Verificacion de Legalidad de Jugadas:

La biblioteca chess se utiliza para evaluar si una jugada realizada en el tablero es válida conforme a las reglas del ajedrez. Esta evaluación se basa en las reglas oficiales del juego y asegura que cada movimiento propuesto o ejecutado cumpla con los requisitos legales establecidos.

7.5.2. Funcionalidad Actual:

Actualmente, el sistema está diseñado para informar si una jugada es válida o no, pero no toma acciones automáticas en respuesta a la validez de la jugada. Esta implementación proporciona un mecanismo de verificación que garantiza que las jugadas cumplan con las normas del juego.

7.5.3. Potencial para Expansion:

Aunque la función actual se limita a la validación de jugadas, la integración de la biblioteca chess abre posibilidades para futuras mejoras. Estas pueden incluir la implementación de medidas adicionales para detectar y gestionar jugadas irregulares o acciones de trampa, tomando decisiones en relacion al mismo, ofreciendo así una mayor robustez y fiabilidad al sistema de ajedrez automatizado.

7.6. Administración de Coordenadas y Configuración del Tablero en el Brazo Robótico

El brazo robótico opera en un entorno tridimensional utilizando un array de coordenadas que define posiciones y orientaciones específicas. Cada entrada del array contiene coordenadas X, Y, Z y el ángulo de pitch, los cuales permiten al brazo alcanzar posiciones exactas y mantener una orientación adecuada para la ejecución de tareas.

El tablero está conformado por estas coordenadas almacenadas en el array. Cada casilla del tablero está representada por un conjunto de coordenadas que define su posición y orientación en el espacio tridimensional. Estas coordenadas son fundamentales, ya que se utilizan al llamar a las funciones correspondientes que permiten al brazo realizar los movimientos según las jugadas o acciones requeridas.

El proceso de ejecución de movimientos comienza con la extracción de las coordenadas necesarias desde el array para las posiciones inicial y final. Esta información es crucial para la planificación del movimiento del brazo. Posteriormente, se calcula el desplazamiento requerido, y se ajusta el ángulo de pitch para asegurar que el brazo mantenga la orientación correcta durante la ejecución de la tarea.

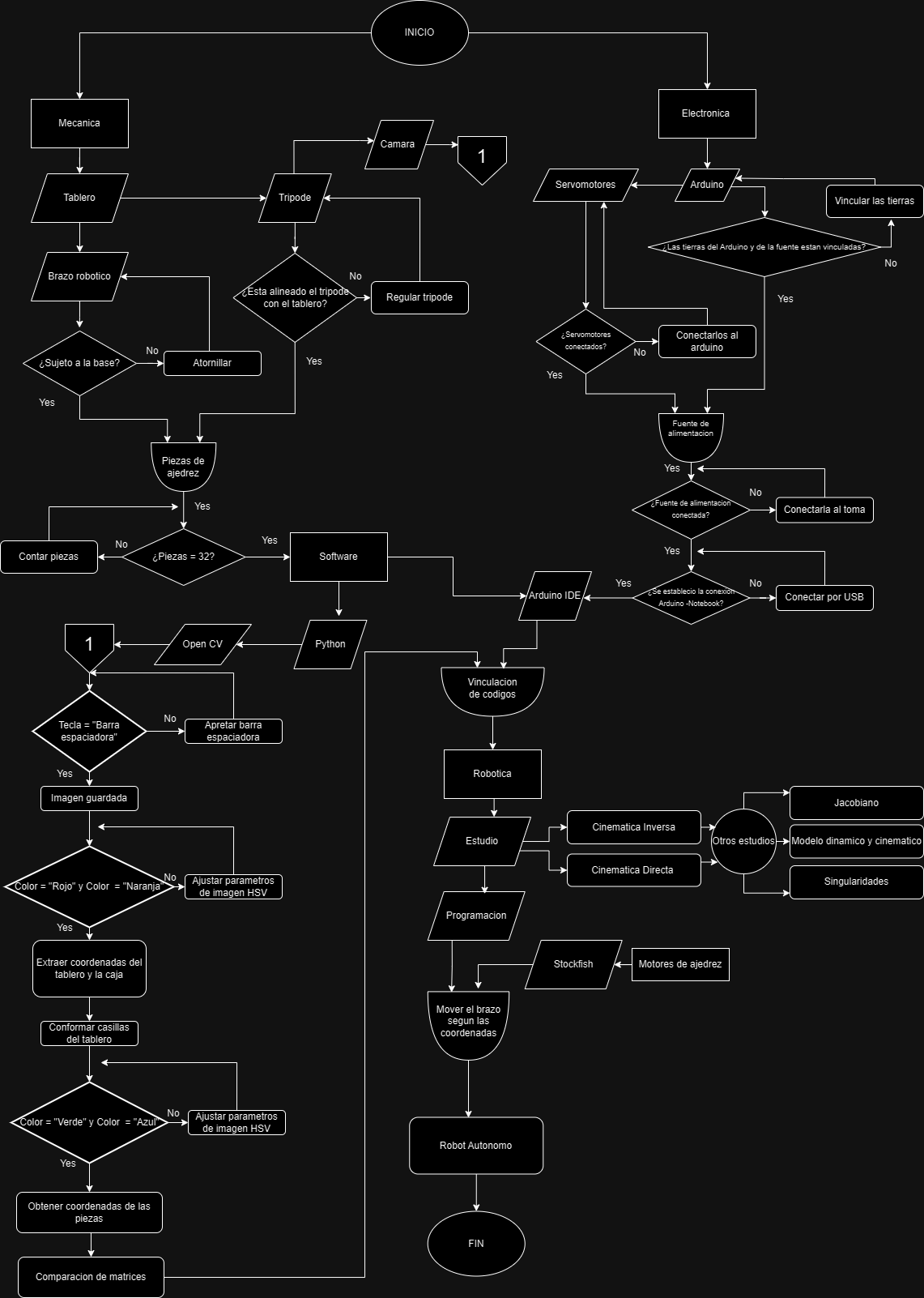
Este enfoque asegura que el brazo robótico pueda posicionarse con precisión en el tablero y ejecutar movimientos complejos de manera efectiva, manteniendo la exactitud y eficiencia requeridas en un entorno tridimensional.

7.7. Funcionalidad Pick-and-Place para Preparación del Tablero

El brazo robótico incorpora una función de pick-and-place para trasladar piezas desde una caja al tablero de juego. Utilizando coordenadas específicas, el brazo recoge las piezas de la caja y las coloca en sus posiciones iniciales en el tablero. Este proceso asegura que las piezas se ubiquen correctamente y estén listas para comenzar el juego, ajustando con precisión las coordenadas X, Y, Z y el ángulo de pitch para una correcta reubicación.

# 8. Diagrama de Flujo del Proceso de Funcionamiento del Sistema de Ajedrez Automatizado

Este diagrama de flujo proporciona una representación visual general del proceso de funcionamiento del sistema de ajedrez automatizado. Abarca desde la captura inicial de imágenes del tablero hasta la ejecución de las jugadas. El diagrama ilustra la secuencia de pasos y la interacción entre los diferentes componentes del sistema, incluyendo la comunicación entre Python y Arduino, el uso del motor de ajedrez Stockfish para la toma de decisiones, entre otros. Esta representación gráfica facilita la comprensión integral del flujo de operaciones y las interacciones clave dentro del sistema.



# 9.Costos del Prototipo y Presupuesto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Imagen | Elemento | Cantidad | Precio unitario US$ | Subtotal US$ |
|  | Arduino MEGA 2560 | 1 | $ 35.99 | $ 35.99 |
|  | Cable USB de Arduino | 1 | $ 4.21 | $ 4.21 |
|  | Gotita | 1 | $ 2.50 | $ 2.50 |
|  | Tuerca | 68 | $ 0.13 | $8.84 |
|  | Base de madera (casera)  (Largo: 60 cm, Ancho: 14 cm, Espesor: 2 cm) | 3 | $ 2.33 | $ 6.99 |
|  | Bisagras | 3 | $ 1.15 | $ 3.45 |
|  | Tornillos para las bisagras | 30 | $ 0.01 | $ 0.30 |
|  | Trípode para celular (casero) | 1 | - | - |
|  | Brazo impreso (Todas las piezas) | 1 | $ 2.87 | $ 2.87 |
|  | Servomotores MG966R  (Voltaje requerido:5V a 7V, Torque: 9.4 a 11 kgf.cm, Angulo: X°) | 5 | $ 12.52 | $ 37.57 |
|  | Servomotores SG90  (Voltaje requerido:5V, Torque: 1.8 kgf.cm, Angulo: X°) | 1 | $ 2.58 | $ 2.58 |
|  | Tornillos centrales de servomotores MG996R | 3 | (Viene con el servo) | (Viene con el servo) |
|  | Tornillos centrales de servomotores SG90 | 3 | (Viene con el servo) | (Viene con el servo) |
|  | Accesorio de servomotor MG996R | 3 | (Viene con el servo) | (Viene con el servo) |
|  | Accesorio de servomotor SG90 | 3 | (Viene con el servo) | (Viene con el servo) |
|  | Tornillos de sujeción de servomotores SG90 | 6 | (Viene con el servo) | (Viene con el servo) |
|  | Tablero de ajedrez pintado a mano | 1 | - | - |
|  | Bulones philips (25 mm) M4 con tuercas hexagonales | 10 | $ 0.12 | $ 1.25 |
|  | Tornillos de sujeción de servomotores MG996R | 12 | (Viene con el servo) | (Viene con el servo) |
|  | Casquillos | 12 | (Viene con el servo) | (Viene con el servo) |
|  | Gomitas | 12 | (Viene con el servo) | (Viene con el servo) |
|  | Tornillos (7 mm) M3 | 30 | $ 0.02 | $ 0.67 |
|  | Piezas de ajedrez y cajita | 35 | $ 129.99 | $ 129.99 |
|  | Pulsador | 1 | $ 0,20 | $ 0,20 |
|  | Corrugado | 1 | $ 15,18 | $ 15,18 |
|  | Madera Fibro Facil | 1 | $ 5,06 | $ 5,06 |
|  | Termocontraible (5 metros) | 1 | $ 7,08 | $ 7,08 |
|  | Cable | 1 | $ 30,36 | $ 30,36 |
|  | Estaño | 1 | $ 20,24 | $ 20,24 |
|  | Cinta aisladora | 5 | $ 0,91 | $ 4,55 |
|  | Cinta de Pintor | 1 | $ 4,05 | $ 4,05 |
|  | Led verde | 1 | $ 1,01 | $ 1,01 |
|  | Resistencia de 220 ohms | 5 | $ 0,25 | $ 1,01 |
|  | Led RGB | 4 | $ 0,38 | $ 1,52 |
| Precio total | | | | 327,47 $ |

**Informe sobre el Gasto del Proyecto del brazo robot**

En el marco del proyecto del brazo robot, hemos observado que el gasto total realizado supera significativamente las estimaciones iniciales presentadas en la tabla de costos. Esta discrepancia se debe principalmente a la implementación de múltiples prototipos a lo largo del proceso de desarrollo.

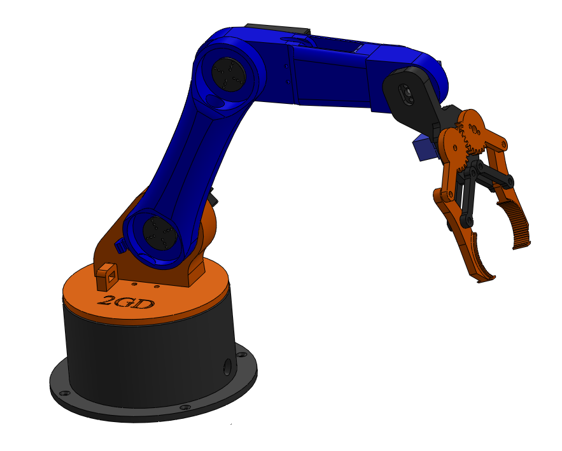
Cada uno de estos prototipos ha requerido ajustes y modificaciones que han impactado directamente en los costos. Las adaptaciones necesarias para mejorar el diseño y la funcionalidad del producto han conllevado inversiones adicionales en materiales, recursos humanos y tiempo. Estas decisiones, aunque no previstas inicialmente, han sido fundamentales para asegurar la calidad y viabilidad del brazo robot.

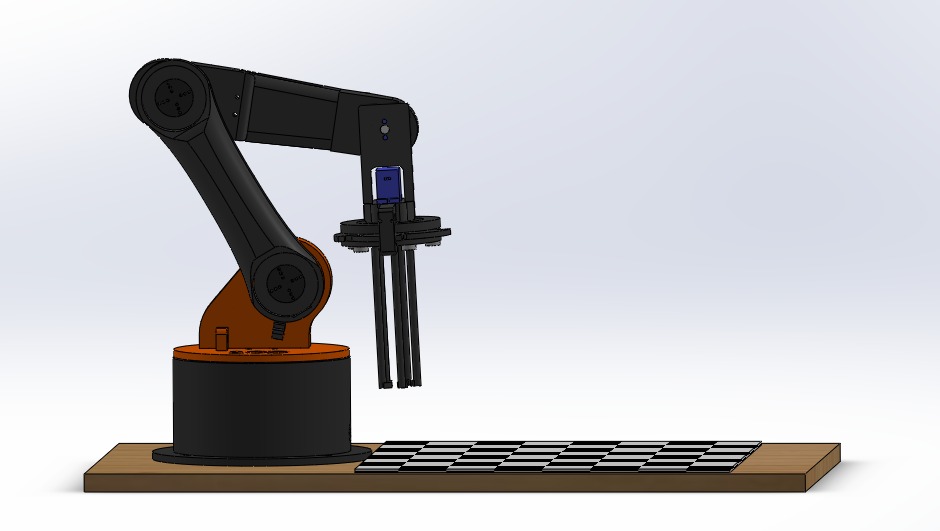
Es esencial considerar que estas inversiones adicionales son una parte crucial del proceso de innovación, permitiéndonos optimizar el producto final y alinearlo mejor con las necesidades del mercado y de los usuarios.

# 10. Comparación del Desempeño del Brazo Robot: Estado Previo y Posterior a las Mejoras

Esta sección presenta una comparación detallada del desempeño del brazo robot antes y después de las modificaciones implementadas. El análisis abarca las diferencias en precisión, funcionalidad, y eficiencia operativa, proporcionando una evaluación del impacto de las mejoras realizadas en el sistema. El diagrama ilustra las principales características y capacidades del brazo robot en ambos estados, destacando las mejoras en el diseño, la funcionalidad de los componentes, y el rendimiento general del sistema.

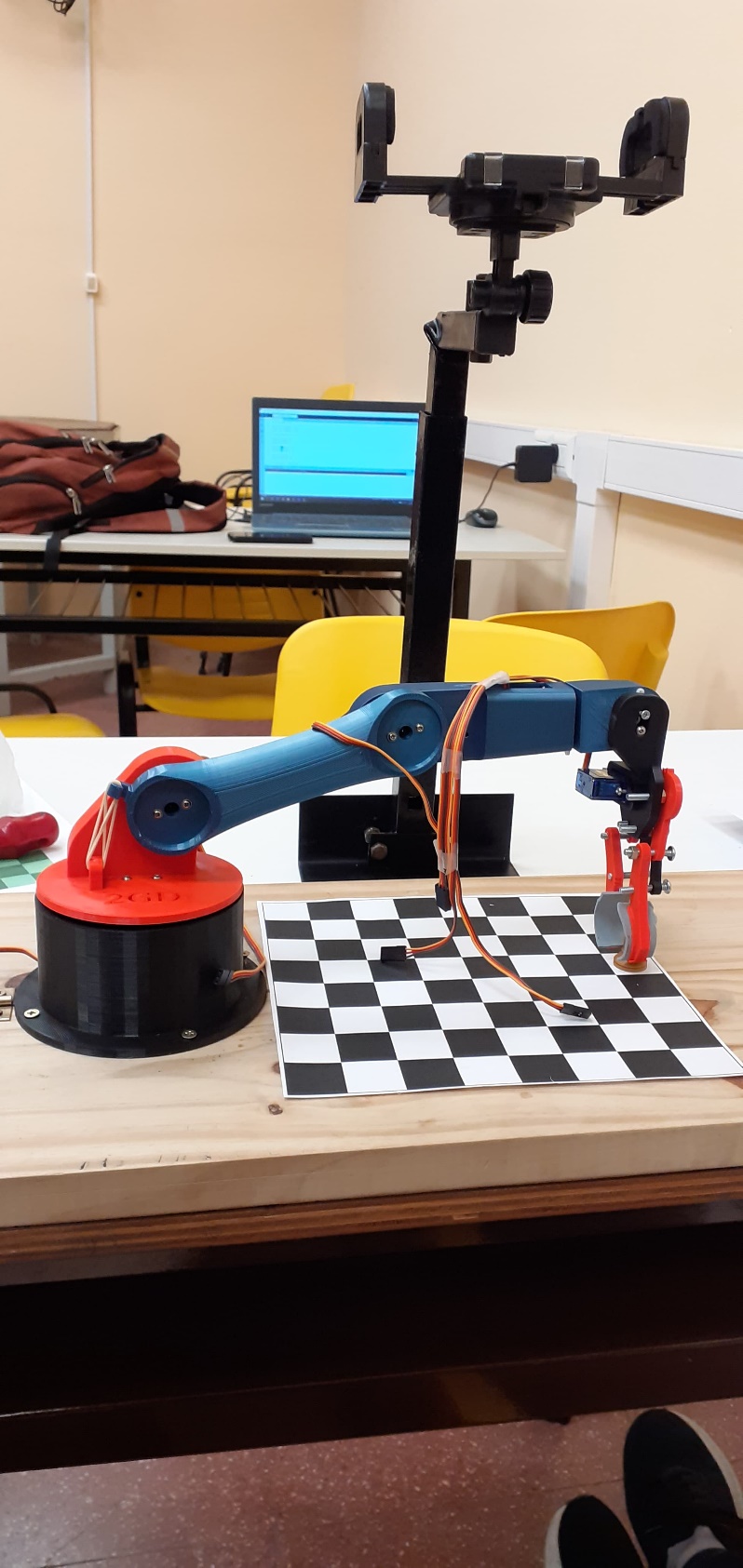
**DISEÑOS POSTERIORES AL FINAL**





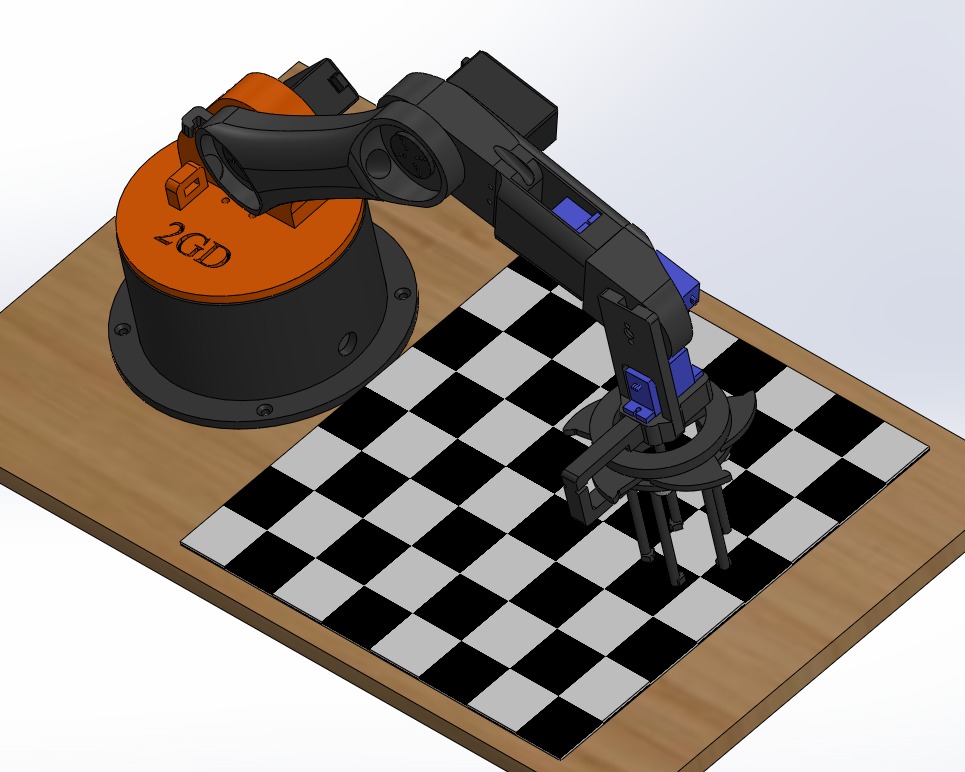
**Primer Prototipo**

En nuestra fase inicial, partimos de un modelo de brazo robot obtenido de Internet, lo que nos permitió agilizar los tiempos de desarrollo. Realizamos ligeras modificaciones en el gripper, se anuló la articulación correspondiente a la muñeca que permitía rotar la misma, esta venia del diseño obtenido el cual su función fue innecesaria para la función del brazo, facilitando los cálculos, luego se adaptaron las piezas, el tablero, las cinemáticas y los cálculos necesarios para adecuarlos al modelo. Tras llevar a cabo las pruebas pertinentes, evaluamos su rendimiento inicial.



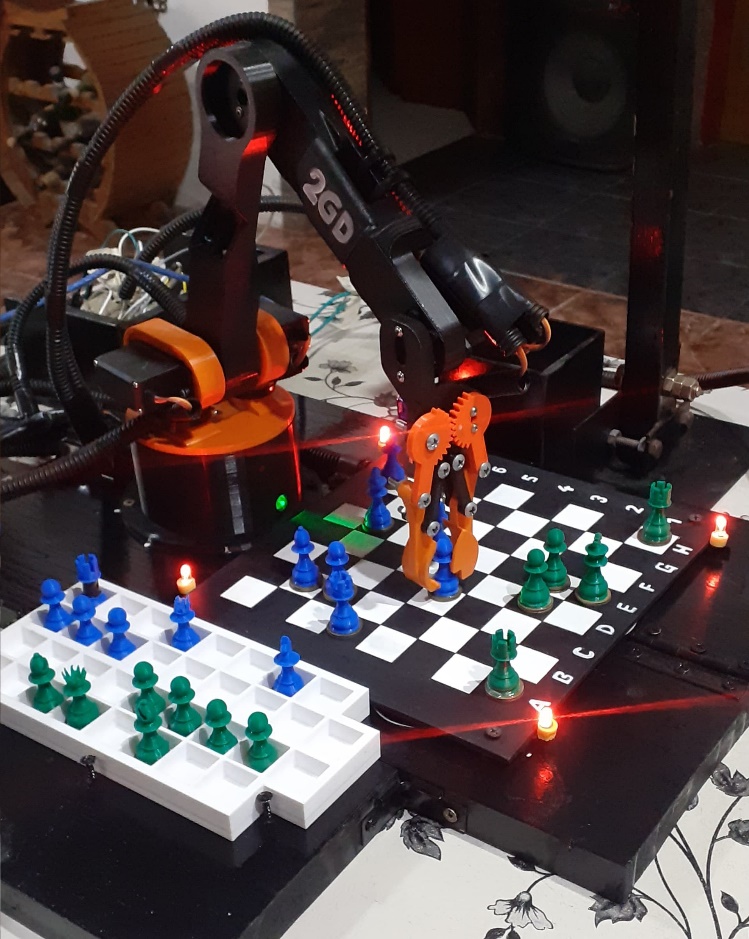
**Segundo Prototipo**

A partir de los resultados del primer prototipo, identificamos varios problemas que requerían ajustes. El gripper fue rediseñado para incorporar un mecanismo tipo iris, que buscaba facilitar el agarre de las piezas. Sin embargo, al realizar pruebas con este prototipo, surgieron problemas de torque en la articulación 2, debido a que el nuevo servomotor generaba un momento mayor que no podía ser compensado adecuadamente. Además, la fricción del mecanismo tipo iris y la fragilidad de sus piezas impidieron su funcionamiento eficaz, lo que nos llevó a considerar un nuevo diseño. Finalmente, se pintaron de negro las piezas antiguas del brazo robot y las nuevas piezas se imprimieron también en negro. Esta decisión se tomó para evitar interferencias en la detección de las piezas azules.



**Tercer Prototipo**

El tercer prototipo incluyó una nueva base para la articulación 2, modificada para montar dos servomotores alineados colinealmente. Esta configuración permitió que ambos servos funcionaran en conjunto para compensar el momento generado en las posiciones más alejadas del tablero. Se rediseño el último eslabón para poder instalar un servo MG996R, ya que el servo SG90 no proporcionaba el torque suficiente. Además, extendimos este eslabón para aumentar el alcance del brazo. En cuanto al gripper, volvimos al mecanismo del primer modelo, pero con mejoras en su diseño. También rediseñamos las piezas y ampliamos el tablero en comparación con los modelos anteriores.



En la primera imagen, se puede observar el brazo robot sin el segundo servo alineado colinealmente a la articulación dos, a pesar de que la base estaba diseñada para soportarlo. En contraste, la segunda imagen muestra una notable mejora tanto estética como funcional. Se ha instalado el segundo servo en Q2 y se han utilizado tubos corrugados para gestionar los cables, lo que contribuye a una presentación más ordenada.

Además, se pintó la madera de negro para optimizar la detección, y se cambió el color del tablero. También se añadió una caja para almacenar las piezas, así como una nueva compartimentación para la parte electrónica que alberga el Arduino y la placa con los cables. Por último, se mejoró el trípode, lo que refuerza la estabilidad del conjunto.

**Cuarto Prototipo**

En el cuarto y último diseño, refinamos aún más el brazo robot. Después de realizar cientos de pruebas, comprobamos que el rendimiento en las posiciones más alejadas del brazo seguía comprometido. Por lo tanto, rediseñamos el tablero, reduciendo su tamaño en una casilla por lado. Esto también requirió un rediseño de las piezas y mejoras en el gripper, llevándolo a su máxima eficiencia en términos de diseño.

Este proceso iterativo nos ha permitido avanzar en la creación de un brazo robot más robusto y eficiente, superando los desafíos presentados en cada etapa.



# 11.Funcionalidades del Proyecto

Este proyecto presenta una innovadora experiencia de juego de ajedrez, incorporando múltiples funcionalidades que mejoran la interactividad y el control. A continuación, se detallan las principales características:

11.1. Control por Comando de Voz

El sistema permite mover las piezas y capturar oponentes utilizando comandos de voz. Simplemente indicándoles las casillas a las que deseas mover las piezas o las que quieres capturar, el brazo robótico ejecutará la acción correspondiente.

11.2. Integración con Chess.com

Los usuarios pueden jugar en línea contra otros jugadores en la plataforma Chess.com. El brazo robótico replica las jugadas realizadas por los oponentes en tiempo real, brindando una experiencia de juego fluida y dinámica.

11.3. Simulación de las Cinemáticas

Se ha desarrollado un programa en Python que simula las cinemáticas, tanto directas como inversas, del brazo robótico. Esta simulación se realiza en un plano tridimensional, permitiendo visualizar y planificar los movimientos antes de ejecutarlos. Una vez calculadas, las órdenes se envían al microcontrolador para que el brazo las replique con precisión.

11.4. **Interfaz Gráfica de Usuario y Juego Autónomo**

La interfaz gráfica del proyecto muestra información clave durante el juego, como la validez de cada jugada, las jugadas realizadas y el turno del jugador. Además, el sistema cuenta con la capacidad de jugar ajedrez de manera autónoma, evaluando las jugadas posibles y tomando decisiones estratégicas. Esto permite a los usuarios disfrutar de una partida sin intervención manual, haciendo la experiencia más inmersiva.

Este proyecto combina tecnología avanzada y diversión, ofreciendo una experiencia de juego de ajedrez única y envolvente.