

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Diseño y construcción de brazo robótico delta de operación manual
para el Laboratorio de Diseño de Procesos de la Universidad del Valle
de Guatemala

Trabajo de Graduación en modalidad de Trabajo Profesional
presentado por Mariana Quirós Amado para optar al grado académico
de Licenciada en Ingeniería Mecatrónica

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Diseño y construcción de brazo robótico delta de operación manual
para el Laboratorio de Diseño de Procesos de la Universidad del Valle
de Guatemala

Trabajo de Graduación en modalidad de Trabajo Profesional
presentado por Mariana Quirós Amado para optar al grado académico
de Licenciada en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.

Firma: 
Inga. Dulce María Chacón Muñoz

Firma: 
Ing. Juan Diego Castillo Amaya

Fecha de aprobación:

Guatemala, 20 de noviembre de 2025

CONTENIDO

LISTA DE CUADROS.....	i
LISTA DE FIGURAS.....	ii
Resumen.....	iii
Abstract.....	iv
1. Introducción.....	1
2. Objetivos.....	2
2.1. General.....	2
2.2. Específicos.....	2
3. Alcances y limitaciones.....	3
3.1. Alcances.....	3
3.2. Limitaciones.....	3
4. Justificación.....	4
5. Marco teórico.....	5
5.1. Robótica.....	5
5.1.1. Robot paralelo.....	5
5.2. Robot tipo Delta.....	5
5.3. Cinemática inversa robots paralelos Delta.....	6
5.3.1. Fundamentos de cinemática inversa.....	6
5.3.2. Análisis geométrico de la cinemática inversa.....	7
5.3.3. Análisis diferencial de la cinemática inversa.....	8
5.4. Impresión 3D.....	9
5.4.1. Rectilíneo.....	9
5.4.2. Panal.....	10
5.4.3. Triangular.....	10
5.4.4. Rejilla.....	10
5.4.5. Giroide.....	11
6. Metodología.....	12
7. Resultados y discusión.....	14
7.1. Diseño de piezas.....	14
7.2. Diseño estructural y análisis.....	19

7.3.	Impresión de piezas 3D.....	20
7.4.	Fabricación y ensamblaje de estructura	21
7.5.	Ensamblaje físico completo del sistema mecánico	22
7.6.	Circuito eléctrico y control manual.....	24
7.7.	Precisión, exactitud y tiempo de respuesta	28
7.7.1.	Precisión.....	28
7.7.2.	Exactitud	30
7.7.3.	Tiempo de respuesta	31
8.	Conclusiones.....	32
9.	Recomendaciones	33
10.	Bibliografía	34
11.	Anexos	36
11.1.	Anexo A: Código cinemática inversa en MATLAB.....	36
11.2.	Anexo B: Código principal Arduino IDE.	39
11.3.	Anexo C: Planos técnicos	42

LISTA DE CUADROS

Tabla 1. Parámetros establecidos para impresiones 3D.....	21
Tabla 2. Medidas y cantidades de perfiles T 2020.....	22
Tabla 3. Componentes impresos 3D	23
Tabla 4. Componentes adicionales	23
Tabla 5. Componentes eléctricos	25
Tabla 6. Variables de entrada.....	26
Tabla 7. Variables de salida	26
Tabla 8. Resultados de la verificación de precisión.....	29
Tabla 9. Resultados exactitud	31

LISTA DE FIGURAS

Figura 1. Estructura de las uniones entre eslabones en un Robot Delta. Enfoque geométrico..	7
Figura 2. Estructura de las uniones entre eslabones en un Robot Delta. Enfoque diferencial...	8
Figura 3. Relleno rectilíneo.....	9
Figura 4. Relleno panal	10
Figura 5. Relleno triangular	10
Figura 6. Relleno rejilla	11
Figura 7. Relleno giroide	11
Figura 8. Base servomotores.....	14
Figura 9. Tapadera base	15
Figura 10. Eslabón 1, brazo corto Delta	15
Figura 11. Eslabón 2, brazo largo Delta	15
Figura 12. Base móvil.....	16
Figura 13. Efecto final, garra.....	16
Figura 14. Anclaje estructura.....	17
Figura 15. Conector perfil L	17
Figura 16. Conector perfil T	17
Figura 17. Acople servomotor	18
Figura 18. Ensamblaje del diseño de piezas.	18
Figura 19. Estructura.....	19
Figura 20. Cálculo de esfuerzo de estructura.....	20
Figura 21. Proceso impresión 3D.....	20
Figura 22. Proceso de corte perfiles T 2020	21
Figura 23. Proceso de rectificado.....	22
Figura 24. Montaje final de estructura	22
Figura 25. Ensamblaje físico de Robot Delta	23
Figura 26. Circuito eléctrico	24
Figura 27. Control manual físico	27
Figura 28. Implementación circuito eléctrico y control manual a robot tipo Delta.....	28
Figura 29. Resultados pruebas serial Arduino	29
Figura 30. Resultados pruebas exactitud MATLAB.....	30

Resumen

En el presente trabajo se desarrolló el diseño y construcción de un brazo robótico tipo Delta con modalidad de operación manual, orientado al fortalecimiento de las capacidades prácticas del Laboratorio de Diseño de Procesos de la Universidad del Valle de Guatemala. Este proyecto tuvo como propósito principal la manipulación de objetos livianos, mediante el control directo del brazo, lo cual permitió realizar tareas simples, como levantar un objeto desde una posición inicial y trasladarlo a una posición final definida, lo cual simuló aplicaciones industriales.

El proyecto se justificó por la necesidad de contar con sistemas reales que permitieran la enseñanza práctica de principios fundamentales de robótica aplicada, como el diseño mecánico, la cinemática inversa, el control de servomotores y la integración de sistemas electromecánicos. Este diseño y construcción buscó mantener un nivel de complejidad académico para futuras implementaciones o modificaciones, orientadas a una posible experimentación e integración en una línea automatizada en los próximos años. Por todo esto se eligió un brazo Delta por sus ventajas en precisión, ligereza y velocidad y por representar una arquitectura robótica paralela de alto interés académico. A diferencia de otros manipuladores, el brazo robótico Delta permitió una construcción más compacta, movimientos repetitivos más estables y una estructura más eficiente para operaciones tipo “pick and place”.

El desarrollo del sistema abarcó el diseño mecánico del brazo mediante CAD Inventor, la selección de sus componentes eléctricos o mecánicos compatibles con el sistema de control integrado Arduino Portenta, y la implementación de un modo manual basado en un control físico local para su operación. La plataforma de Arduino fue seleccionada por su versatilidad y potencia dentro de sistemas de control embebido. Por otro lado, se diseñó un código de análisis en MATLAB que implementó cinemática inversa con el objetivo de calcular los ángulos de las articulaciones, lo cual permitió analizar y evaluar la precisión posicional y la fiabilidad de las mediciones del sistema.

Si bien el proyecto no contempló la automatización ni clasificación de productos, se buscó que el mismo contribuyera a la futura optimización de los procesos dentro del laboratorio y a su vez, sirviera como base para futuras investigaciones y el desarrollo continuo del modelo e integración de nuevas funcionalidades. Asimismo, se esperó que los resultados contribuyeran al fortalecimiento de la formación de futuros ingenieros, brindando un entorno y sistema para la aplicación de sistemas controlados y robótica aplicada.

Abstract

This project involved the design and construction of a manually operated Delta-type robotic arm, aimed at strengthening the practical capabilities of the Process Design Laboratory at the Universidad del Valle de Guatemala. The main purpose of this project was to manipulate light objects through direct control of the arm, which allowed simple tasks to be performed, such as lifting an object from an initial position and moving it to a defined final position, simulating industrial applications.

The project was justified by the need for real systems that would allow for the practical teaching of fundamental principles of applied robotics, such as mechanical design, inverse kinematics, servomotor control, and the integration of electromechanical systems. This design and construction sought to maintain a level of academic complexity for future implementations or modifications, aimed at possible experimentation and integration into an automated line in the coming years. For this reason, a Delta arm was chosen for its advantages in precision, lightness, and speed, and because it represents a parallel robotic architecture of high academic interest. Unlike other manipulators, the Delta robotic arm allowed for a more compact construction, more stable repetitive movements, and a more efficient structure for pick-and-place operations.

The development of the system included the mechanical design of the arm using CAD Inventor, the selection of electrical and mechanical components compatible with the Arduino Portenta integrated control system, and the implementation of a manual mode based on local physical controls for its operation. The Arduino platform was selected for its versatility and power within embedded control systems. In addition, an analysis code was designed in MATLAB that implemented inverse kinematics to calculate joint angles, which allowed for the analysis and evaluation of the system's positional accuracy and measurement reliability.

Although the project did not include automation or product classification, it was hoped that this work would contribute to the future optimization of processes within the laboratory and, in turn, serve as a basis for future research and the continuous development of the model and integration of new functionalities. Likewise, it was hoped that the results would contribute to strengthening the training of future engineers, providing an environment and system for the application of controlled systems and applied robotics.

1. Introducción

El desarrollo de la tecnología, a lo largo de la historia, impulsó avances en la automatización industrial en procesos de manufactura y empaquetado. Estos avances permitieron la mejora de la eficiencia y precisión de la producción en tecnologías avanzadas de entornos educativos y de investigación. En Guatemala, la automatización ha presentado un crecimiento constante, jugando un papel importante en la optimización de procesos, reducción de costos y aumento de la competitividad dentro de la industria.

El presente trabajo se enfocó en el diseño y construcción de un brazo robótico Delta con modalidad de operación manual, orientado a la mejora de las capacidades prácticas del Laboratorio de Diseño de Procesos en la Universidad del Valle de Guatemala. El sistema que se desarrolló permitió la manipulación de objetos livianos mediante controles locales físicos, realizando tareas básicas de colocación y recolección, típicas de aplicaciones industriales del tipo "pick and place". Los robots Delta son ampliamente reconocidos por su alta velocidad, precisión y eficiencia en operaciones repetitivas, lo que los hace ideales para su uso académico en prácticas de robótica aplicada.

Este proyecto da un enfoque introductorio y controlado, adecuado para la formación de estudiantes en áreas como diseño mecánico, programación embebida y control de sistemas, a diferencia de sistemas totalmente automatizados. El objetivo principal fue desarrollar una plataforma funcional que permitiera realizar movimientos controlados, integrando conocimientos en diseño CAD, selección de componentes mecatrónicos, control mediante microcontroladores y validación a través de herramientas de análisis.

El desarrollo del trabajo abarcó desde la idea del modelo, hasta su implementación y validación experimental. En la fase de pruebas se analizaron parámetros clave como la precisión en la posición del efector final (base móvil), la repetición de los movimientos y el tiempo de respuesta del sistema. Los datos permitieron verificar el desempeño del brazo bajo operación manual, dando las bases para futuras ampliaciones del sistema, que pudieran incluir automatización progresiva, visión artificial o integración en líneas de producción.

Finalmente, este documento presenta la documentación técnica del proyecto: Incluyendo planos, especificaciones, listas de materiales y código fuente. El alcance del proyecto es contribuir en el avance de la infraestructura educativa del laboratorio. Complementando el aprendizaje práctico de los estudiantes de ingeniería mecatrónica, mejorando la comprensión de sistemas robóticos paralelos y su aplicación en entornos reales.

2. Objetivos

2.1. General

Diseñar y construir un brazo robótico tipo Delta con modalidad de operación manual básica, para permitir la manipulación sencilla de objetos, mediante un sistema de controles físicos integrados, en el Laboratorio de Diseño de Procesos de la Universidad del Valle de Guatemala.

2.2. Específicos

1. Diseñar la estructura mecánica del brazo robótico tipo Delta, para garantizar estabilidad y precisión al manipular objetos livianos, mediante herramientas de modelado CAD.
2. Implementar un sistema de control manual básico, para permitir el movimiento del brazo en tareas simples de manipulación, mediante controles físicos integrados.
3. Realizar pruebas de validación funcional del sistema, para asegurar su precisión y repetibilidad en la manipulación de objetos, mediante la medición de parámetros como precisión posicional, tiempo de respuesta y consistencia del movimiento.
4. Documentar el proceso de diseño, construcción y programación del sistema, para facilitar futuras modificaciones o mejoras, mediante planos técnicos, especificaciones y código fuente.

3. Alcances y limitaciones

El siguiente trabajo tuvo como alcance la realización y construcción de un brazo robótico tipo Delta con modalidad de operación manual, con el objetivo de ser utilizado como un recurso didáctico en el Laboratorio de Diseño de Procesos de la UVG. El diseño y creación realizados con la finalidad de manipular objetos livianos en un entorno controlado.

3.1. Alcances

Entre sus principales alcances se enmarcaron:

- Desarrollo de un sistema funcional, que ejecutará movimientos controlados, mediante un sistema de control manual, con componentes electrónicos físicos.
- La creación de un modelo tridimensional y completo del brazo Delta en un software de diseño.
- Integración de componentes mecánicos y electrónicos.
- Validación funcional del proyecto por medio de pruebas de repetibilidad y precisión posicional.
- Documentación técnica del proceso, incluyendo todo el material generado y creado para una futura modificación o replicación.

Durante el desarrollo de este se utilizaron herramientas de Diseño y construcción, tales como Autodesk Inventor, MATLAB, Arduino IDE y componentes físicos para su creación. Cada uno de estos, aseguró la viabilidad del prototipo. Asimismo, el desarrollo de este se llevó a cabo con acceso autorizado a las instalaciones y recursos del laboratorio de Diseño de Procesos. La totalidad de información técnica utilizada para la creación del brazo robótico se obtuvo de bibliografías académicas, artículos y documentaciones especializadas en el tema.

3.2. Limitaciones

Por otro lado, el desarrollo del proyecto presentó algunas limitaciones:

- Inexistencia de retroalimentación sensorial, ya que sólo fue operado de forma manual por un operario.
- El diseño se limitó en base a un calendario académico, lo cual reduce la posibilidad de un trabajo o tareas extensas.
- No se contemplaron análisis térmicos de los componentes mecánicos y electrónicos.

Por último, este trabajo se basó en tres áreas clave de excelencia: automatización, diseño mecánico y electrónico e integración de sistemas, conforme a la carrera de Ingeniería Mecatrónica de la Universidad del Valle de Guatemala.

4. Justificación

El desarrollo de un brazo robótico Delta, con modalidad manual, resolvió la necesidad de fortalecer la automatización en entornos experimentales y académicos, facilitó el aprendizaje práctico en robótica y control de procesos. A la fecha de elaboración del proyecto, la línea de empaque automatizada del Laboratorio de Diseño de Procesos de la Universidad del Valle de Guatemala, no cuenta con un sistema de clasificación de productos, lo cual limita su funcionalidad como plataforma de enseñanza e investigación.

Este proyecto permitirá, en un futuro, a los estudiantes, aplicar conocimientos en diseño mecánico, electrónica y control automático, mejorando su comprensión de sistemas robóticos utilizados en la industria. Además, el desarrollo de un sistema operativo con un modo manual posibilitará su integración progresiva en un entorno automatizado sin tener dependencia completa de algoritmos autónomos desde el inicio. Esto es fundamental en un contexto educativo, donde es esencial garantizar el control del sistema en todo momento, para fines didácticos y de seguridad.

El uso de un robot tipo Delta resulta adecuado en el laboratorio por su alta velocidad, precisión y eficiencia en tareas de “pick and place”. Estas características lo convierten en una herramienta óptima para procesos de clasificación dentro de líneas de producción, incluso en aplicaciones a pequeña escala. No omitiendo, su estructura ligera y compacta que permitió realizar movimientos rápidos y repetitivos, optimizando los tiempos de ejecución, sin comprometer la estabilidad del sistema.

La implementación del proyecto implicará una mejora significativa en la infraestructura académica, proporcionando a los estudiantes una herramienta de experimentación física que les permitirá observar y comprender de manera tangible los principios de la robótica aplicada, la cinemática inversa y el control de motores.

Del mismo modo, este documento establece bases para futuras ampliaciones del sistema, tales como la implementación de visión artificial, desarrollo de algoritmos, clasificación automática y/ o su integración a sistemas más complejos de producción. De esta forma, se promoverá un enfoque de mejora continua en el laboratorio, fomentando la innovación, la investigación y el aprendizaje autónomo en el campo de la ingeniería mecatrónica.

5. Marco teórico

5.1. Robótica

La robótica es una disciplina de la ingeniería que mezcla las destrezas de mecánica, electrónica, control y computación. Se enfoca en el diseño, construcción, programación y control de sistemas, enfocado en realizar tareas de manera automática o semi automática. Con el paso del tiempo, la robótica ha avanzado significativamente, desde sus inicios, en aspectos industriales en 1960, expandiéndose a campos más avanzados como la medicina, agricultura, minería, educación, entre otros. En la actualidad es clave para la automatización de procesos, ya sean industriales o en otros campos, la optimización de una tarea o la reducción en errores medibles de los humanos.

En el sector industrial, los robots se utilizan y adaptan en tareas repetitivas, operaciones que requieren de alta precisión, o en la inspección de procesos. En el ámbito educativo, es de suma importancia para reforzar el aprendizaje práctico de conceptos adquiridos a lo largo de la educación, nutriendo innumerables habilidades para futuras aplicaciones.

Los grados de libertad (GDL, nombrado por siglas) son esenciales para describir y mostrar el comportamiento de sistemas mecánicos. Un GDL indica una forma independiente de movimiento en un espacio de tres dimensiones. En robótica, hace referencia a los ejes en los cuales un robot puede realizar un movimiento o rotación. Un robot que cuenta con tres grados de libertad se puede desplazar en los ejes X, Y y Z, mientras que uno que cuente con una cantidad mayor también podrá realizar movimientos de rotación sobre sus ejes.

El número de grados de libertad con los que cuente un robot, hacen referencia directamente a su versatilidad y complejidad. Por lo cual, los robots con más GDL pueden alcanzar un mayor número de posiciones y orientaciones, siempre y cuando se tengan los algoritmos adecuados para poder realizar dichos movimientos.

5.1.1. Robot paralelo

Un robot paralelo es un manipulador, donde el efector final (garra) se conecta a la base mediante múltiples cadenas cinemáticas para poder realizar el movimiento deseado. Estas cadenas, ayudan con el control de la posición y orientación de un efector final. Por otro lado, si se compara con los robots seriales, se tienen los actuadores distribuidos a lo largo del brazo, mientras que en los paralelos se tiene en la base, lo cual es de gran ayuda a una reducción de fricción entre elementos. Las ventajas de los robots paralelos incluyen: Alta rigidez, mayor capacidad de carga, velocidad y precisión. Sin embargo, su área de trabajo es más limitada y su control cinemático más complejo. Algunos robots Delta son:

- Tipo Delta,
- Tipo Hexápodos.
- Ciertas máquinas CNC.

5.2. Robot tipo Delta

El robot tipo Delta fue introducido a mediados de la década de los 80 por el ingeniero Reymond Clavel en la École Polytechnique Fédérale de Lausanne. Este diseño fue creado con base en la

necesidad de realizar tareas de forma rápida, eficiente y precisas. Este tipo de robot emplea una estructura paralela que se construye por múltiples brazos actuadores, generalmente tres, los cuales están conectados a una plataforma móvil a través de eslabones, que conforman estructuras de tipo paralelogramo.

Cada brazo del robot Delta está compuesto por dos eslabones articulados que unidos generan una configuración cinemática permitiendo el movimiento independiente de cada uno. Asimismo, esta estructura permite que la plataforma móvil tenga una orientación establecida a lo largo de su desplazamiento, lo cual proporciona una mejor estabilidad y precisión en el sistema al momento de realizar movimientos repetitivos. Por otro lado, la estructura de paralelogramo influye para que, el actuador conectado en la base móvil del robot reduzca la inercia de los elementos, proporcionando una velocidad mayor.

Es por todo esto, por lo cual el robot tipo Delta se destaca y proporciona como sus principales ventajas:

- Alta velocidad.
- Baja inercia.
- Alta productividad.
- Confiabilidad.
- Gran repetibilidad de tareas.

Estas características hacen que los robots Delta sean la opción óptima para la industria en tareas de automatización, como “pick and place”, empaquetado, inspecciones, entre otras que requieren de rapidez y precisión. También, su diseño compacto facilita su integración en espacios compactos y en líneas de producción automatizadas.

5.3. Cinemática inversa robots paralelos Delta

La cinemática es una rama de la robótica encargada de estudiar el movimiento de los manipuladores, sin considerar las fuerzas que lo generan. Este se divide en dos áreas: La cinemática directa, que determina la posición y orientación del efector final a partir de las variables otorgadas a las articulaciones del robot y la cinemática inversa, que se dedica a obtener las variables necesarias angulares para encontrar la posición y orientación deseada del efector final.

5.3.1. Fundamentos de cinemática inversa

En este caso específico, en los robots paralelos tipo Delta, la cinemática inversa es de suma importancia, ya que se define mediante la estructura del mecanismo y su similitud de geometría entre eslabones. A diferencia de los robots seriales, los cuales requieren múltiples soluciones o métodos numéricos complejos para encontrar posiciones mediante ángulos u otros datos, los robots Delta ofrecen la ventaja de ser más sencillos a la forma de analizarlos matemáticamente. Por lo cual, la cinemática inversa es la forma más directa para realizar cálculos sobre sus efectores o manipuladores.

Esto se debe a la simetría que tiene la arquitectura del robot. Este robot consiste en tres brazos paralelos unidos a una base fija y a otra móvil. Dicha configuración permite que el robot se

movilice en tres dimensiones. La cinemática inversa permite que se calcule con base a una posición deseada del efector final de coordenadas (x, y, z) , los ángulos que cada uno de los brazos obtendrá $(\theta_1, \theta_2, \theta_3)$. Con esta posición se puede colocar el efector final en la posición adecuada y deseada. Este proceso de cálculo se fundamenta en un enfoque diferencial y/o geométrico.

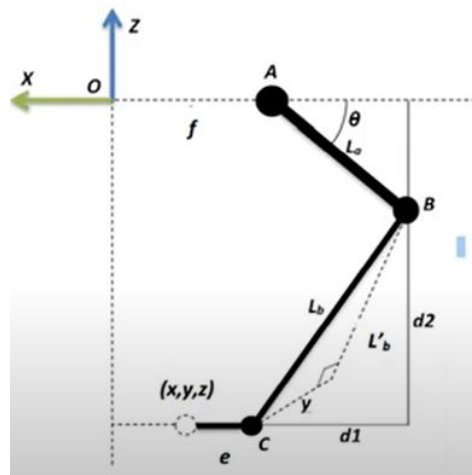
El enfoque diferencial se establece mediante la relación entre la velocidad del efector final y las velocidades de las articulaciones en los brazos. Dicha relación se obtiene y mide mediante la matriz Jacobiana. Este método se utiliza para algoritmos de control en tiempo real, es decir, cuando se requiere aplicar controladores cinemáticos que permitan incorporar controles de errores, compatibilidad con algoritmos de convergencia o cuando se vaya a utilizar controladores PID o LQI. (2022, Zenteno Lara).

Por otro lado, el enfoque geométrico se basa en trigonometría y relaciones espaciales entre la estructura y arquitectura del robot. Se utiliza ampliamente en el análisis de los robots Delta gracias a su simplicidad de diseño y simetría estructural. Este se basa en analizar los triángulos que se forman entre el punto fijo de la base, el extremo del eslabón superior y el punto de unión de la plataforma móvil, correspondiente al punto deseado del efector final.

5.3.2. Análisis geométrico de la cinemática inversa

El análisis geométrico de la cinemática inversa de un robot Delta se fundamenta en las relaciones trigonométricas que tiene y permite calcular los ángulos articulares del punto A, el cual se observa en la Figura 1, para posicionar el efector final en la posición deseada. Este método no requiere derivadas, ni matrices Jacobianas, por lo cual se basa exclusivamente en sus relaciones espaciales y trigonométricas directamente entre los eslabones y puntos de unión del sistema mecánico del robot paralelo.

Figura 1. Estructura de las uniones entre eslabones en un Robot Delta. Enfoque geométrico



Fuente: A Alashgar, 2013.

La geometría básica de un robot paralelo delta se basa en:

- Base fija donde se ubica el motor, punto “A”.
- Eslabón “AB”, cuyo ángulo se obtiene con respecto a la horizontal.
- Unión de eslabones, punto “B”.

- Eslabón “BC”, conectado a la plataforma móvil.
- Punto de conexión de efector final (base móvil) con eslabón “BC”, punto “C”.
- Efector final, proyección “e”.

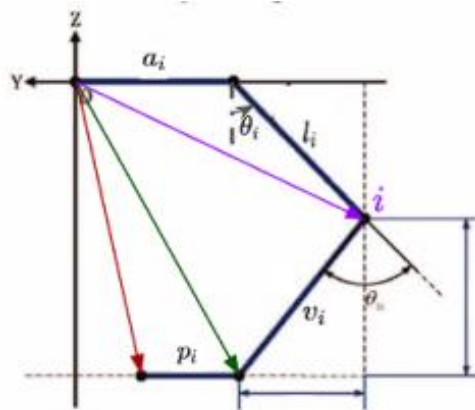
Todos estos elementos permiten abordar la cinemática inversa como un análisis de triángulos, donde se conoce la posición final deseada y se busca el ángulo con respecto a la horizontal, buscando siempre respetar las longitudes de los eslabones y los ángulos formados entre ellos.

Este método se utiliza por múltiples razones, entre ellas su fácil implementación, resolviendo cinemática inversa para cada uno de sus brazos; también, se requiere de operaciones geométricas simples y trigonometría sencilla. Por otro lado, la implementación en análisis mediante computadora es sencillo, generando que al momento de implementarlo en sistemas como microcontroladores o sistemas de control embebidos sea sencillo. Por último, es útil para diseño mecánico o simulaciones.

5.3.3. Análisis diferencial de la cinemática inversa

La relación entre las articulaciones y el efector final se basan en los elementos que unifican cada cadena cinemática de un robot paralelo. (2022, Zenteno Lara). Para lo cual, se establecen relaciones entre las velocidades de las articulaciones y del efector final, como se menciona anteriormente. Es por esto, por lo que se utiliza el método diferencial y matriz Jacobiana. Este enfoque se usa particularmente donde se requieren controles en tiempo real, seguimiento de trayectorias o implementación de algoritmos con retroalimentación de datos, que a futuro puede verificar el correcto funcionamiento de un robot paralelo tipo Delta.

Figura 2. Estructura de las uniones entre eslabones en un Robot Delta. Enfoque diferencial



Fuente: zDynamics, 2021.

En los robots Delta se tienen tres actuadores, los cuales son rotacionales y se derivan como “ $\theta_1, \theta_2, \theta_3$ ” y una plataforma móvil que cuenta con tres grados de libertad en los ejes “x, y, z”. La relación entre estos dos, se representa por medio de la matriz Jacobiana:

$$\dot{r} = J(\theta) \cdot \dot{\theta}$$

Donde

- \dot{r} , vector de velocidad lineal del efector final.

- $\dot{\theta}$, vector de velocidades angulares de los motores.
- $J(\theta)$, matriz Jacobiana directa, relación entre los dos conjuntos de variables.

5.4. Impresión 3D

La impresión 3D es un proceso en el cual se fabrican objetos tridimensionales por medio de una adición de material por capas. Siendo distinto a los procesos de fabricación de piezas manufacturadas en las cuales se retira material de una pieza en bruto. Este proceso permite el uso de diversos materiales, como los plásticos, resinas, entre otros. Esto dependerá del tipo de impresora que se utilice y las propiedades que requiera la pieza.

En la actualidad, la impresión 3D es cada vez más utilizada en la fabricación de piezas y componentes mecánicos. Esto gracias a que se logran crear piezas complejas sin la necesidad de un proceso extenso de maquinado. Las impresoras de esta tecnología son capaces de convertir diseños digitales en objetos físicos. Es por esto por lo que se pueden utilizar diversos materiales los cuales poseen distintas propiedades y dependerá del uso de la pieza y la resistencia mecánica y durabilidad que deba tener.

Uno de los materiales más utilizados en este proceso es el PLA, por su bajo costo y su facilidad de impresión. Asimismo, este es un plástico “resistente, flexible y tiene una baja inflamabilidad” (SICNOVA, 2022). Convirtiéndolo en un material adecuado para aplicaciones mecánicas.

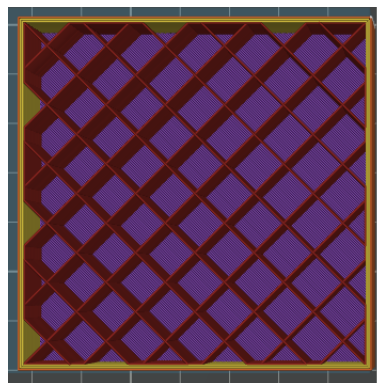
Para que se obtenga y realice una impresión de calidad, es crucial que se seleccione el tipo de relleno interno pertinente, el cual define la estructura interna de la pieza. Este relleno no solo influye en la estructura interna de la pieza, también repercute en la resistencia y rigidez de esta. Existen múltiples tipos de rellenos los cuales favorecen la pieza dependiendo del uso que se le vaya a dar.

Estos son algunos de los rellenos en la fabricación de piezas realizadas por impresión 3D con el material de plástico PLA:

5.4.1. Rectilíneo

El relleno interno rectilíneo está conformado por varias líneas paralelas, por cada una de sus capas. Estas líneas cruzan a la capa anterior en un ángulo de 90° . Esto con el fin de reforzar la pieza en dos dimensiones. Es uno de los patrones de relleno más rápidos de impresión.

Figura 3. Relleno rectilíneo

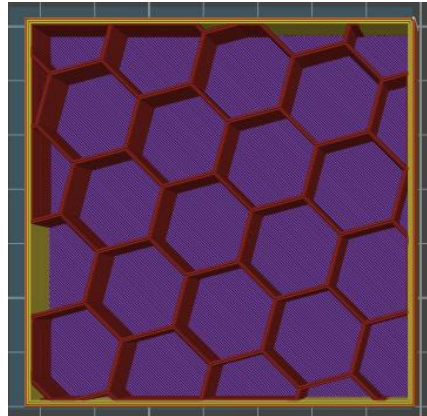


Fuente: Elaboración propia.

5.4.2. Panal

El relleno interno con forma de panal está conformado por múltiples hexágonos paralelas por cada una de sus capas. Esta estructura interna tiene mayor resistencia mecánica y caminos que no se cruzan en la impresión. Sin embargo, este patrón consume aproximadamente entre un 25% a un 30% más material que los demás patrones. Generalmente, este se utiliza cuando se quiere reducir peso en la pieza.

Figura 4. Relleno panal

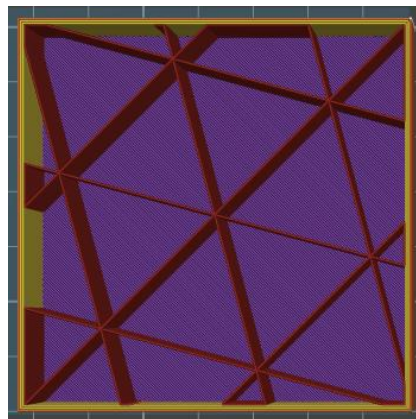


Fuente: Elaboración propia.

5.4.3. Triangular

El relleno triangular es un relleno similar a una cuadrícula, con la diferencia que los caminos de las capas se cruzan en tres direcciones, formando una estructura interna de triángulos. Esto proporciona una mayor resistencia a la pieza, en piezas con superficies grandes y planas.

Figura 5. Relleno triangular

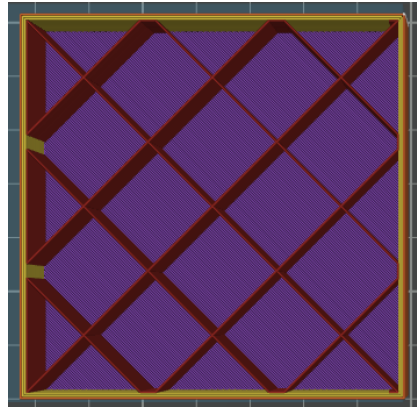


Fuente: Elaboración propia

5.4.4. Rejilla

El relleno de rejilla es muy similar al patrón rectilínea, con la diferencia que en el diseño de estructura interna se va colocando capa por capa y se cruza entre sí en la misma, no espera a la siguiente capa para cruzar las líneas a 90°. Al tener el cruce del material en la misma capa, alcanza que la estructura de la pieza sea más sólida.

Figura 6. Relleno rejilla

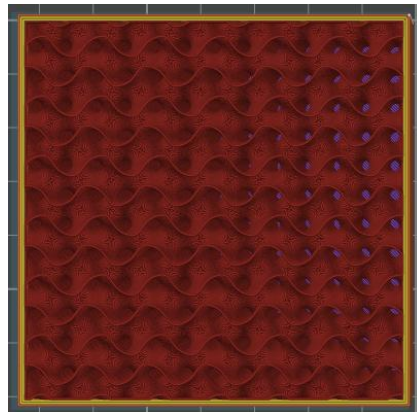


Fuente: Elaboración propia.

5.4.5. Giroide

El relleno giroide capa por capa va creando líneas onduladas alternadamente. Este diseño permite que la estructura interna tenga un equilibrio entre su resistencia, uso de material y es usada comúnmente cuando la fuerza e integridad de la estructura interna de la pieza es de suma importancia. Asimismo, este patrón permite que las propiedades mecánicas en todos sus ejes sean muy similares, logrando que la carga en todas las direcciones sea mayor.

Figura 7. Relleno giroide



Fuente: Elaboración propia.

6. Metodología

La metodología propuesta para el desarrollo del brazo robótico, tipo Delta, constó de siete fases principales, alineadas con los objetivos específicos del proyecto: Diseño mecánico de piezas, diseño estructural y análisis, impresión de piezas 3D, fabricación y ensamblaje de estructura, ensamblaje físico completo del sistema mecánico, realización del circuito eléctrico con control manual y la validación funcional del sistema y análisis de resultados. Este trabajo se enmarcó en una investigación de tipo experimental, ya que se construyó un prototipo físico que será sometido a pruebas en laboratorio.

La primera fase, como punto de partida del proyecto, se basó en el desarrollo de un código en MATLAB que implementó la cinemática inversa del robot Delta. Este análisis permitió determinar los rangos de movimiento del efector final en función de distintas longitudes de eslabones. Con base en estos resultados, se definieron los tamaños óptimos de los eslabones para cumplir con los requisitos de alcance horizontal y vertical, dentro del entorno del laboratorio.

A partir de esto, se diseñó la estructura mecánica y se realizó el modelado tridimensional del brazo robótico tipo Delta, utilizando el software Autodesk Inventor. Durante esta etapa, se definieron las dimensiones de los eslabones, plataforma móvil y elementos estructurales, tomando en cuenta criterios de rigidez, peso, estabilidad y precisión en el movimiento. Se priorizó la facilidad de ensamblaje y fabricación, manteniendo la geometría típica de un sistema Delta.

Posteriormente, durante la segunda fase, se diseñó la estructura de soporte de los componentes mecánicos del robot. Para el correcto diseño de esta, se utilizaron factores como la rigidez, estabilidad, peso total del brazo Delta y que fuera sencillo de ensamblar. En Autodesk Inventor se realizó un análisis de esfuerzo, mediante su simulación, aplicando las cargas del peso del brazo, el cual permitió validar la capacidad estructural del diseño antes de su fabricación.

Luego, durante la tercera fase, las piezas diseñadas fueron fabricadas mediante impresión 3D. Los parámetros de impresión se fueron ajustando mediante varias impresiones, verificando la compatibilidad de piezas, ligereza y acabado. Esta etapa permitió obtener componentes ligeros y de bajo costo, ideales para un prototipo. Después de esto, para la cuarta fase se fabricó la estructura principal, por medio del corte y ensamblaje de perfiles de aluminio extruido 2020. Se emplearon conectores de 90° y piezas impresas en 3D, garantizando uniones firmes y una base sobre la cuál se montaría el resto del sistema robótico tipo Delta.

Al momento de contar con todas las piezas individuales y soporte del brazo, se realizó el ensamblaje del sistema mecánico completo. Durante esta fase (quinta) se verificó la libertad de movimiento en las articulaciones que conforman el sistema y se concluyó con el prototipo físico. Luego, se comenzó con la selección de componentes eléctricos para lo cual, se seleccionaron elementos electrónicos necesarios para el funcionamiento del sistema. Los componentes se eligieron tomando en cuenta los criterios de compatibilidad con la plataforma

Arduino Portenta H7, requerimientos de torque, peso, precisión y disponibilidad. También se diseñó la integración física de estos elementos en la estructura previamente modelada.

La implementación del sistema de control manual fue la sexta fase, en la cual se diseñó el circuito eléctrico y se desarrolló el sistema de control manual básico utilizando el entorno de programación Arduino IDE. Luego, se programó la lógica para permitir el accionamiento individual de los servomotores, mediante controles físicos integrados, como potenciómetros y botones. Esta fase permitió realizar desplazamientos controlados del brazo, replicando tareas de manipulación simples, en condiciones de operación local.

Una vez construido el prototipo para la última fase, se utilizó el entorno MATLAB para verificar la correspondencia entre los ángulos articulares calculados, mediante cinemática inversa y los ángulos reales generados por los servomotores durante la operación del brazo. Esta comparación permitió evaluar si el sistema responde adecuadamente a las posiciones objetivo-establecidas. La validación incluyó análisis de repetibilidad, exactitud y verificación de movimiento. Este proceso condujo a concluir el desempeño correcto del sistema bajo condiciones de operación manual básica.

Finalmente, se documentó todo el proceso mediante planos técnicos, listas de materiales, circuito eléctrico y código fuente, con el objetivo de facilitar futuras mejoras, implementación de nuevas funcionalidades o replicación del proyecto en otros contextos educativos.

7. Resultados y discusión

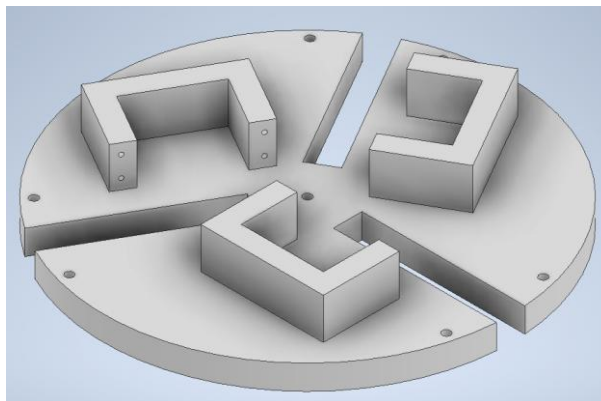
En la presente sección se muestran los resultados del proyecto “Diseño y construcción de brazo robótico delta de operación manual para el Laboratorio de Diseño de Procesos de la Universidad del Valle de Guatemala”. El análisis de resultados y discusión se encuentran subapartados del proceso de diseño, fabricación y validación, con el propósito de cumplir los objetivos planteados en la sección cuatro.

7.1. Diseño de piezas

Para la construcción del brazo Delta se diseñaron y modelaron diversas piezas utilizando el software Autodesk Inventor, definiendo en cada una de ellas las dimensiones y características correspondientes al diseño mecánico del robot. Estas piezas fueron desarrolladas con el propósito de obtener modelos que cumplieran con los requerimientos geométricos del mecanismo delta y funcionales del sistema, sirviendo como base para su posterior fabricación mediante impresión 3D. El diseño de cada una de las piezas elaboradas fue creado siguiendo las geometrías identificadas en la sección de Marco Teórico para un robot paralelo tipo delta.

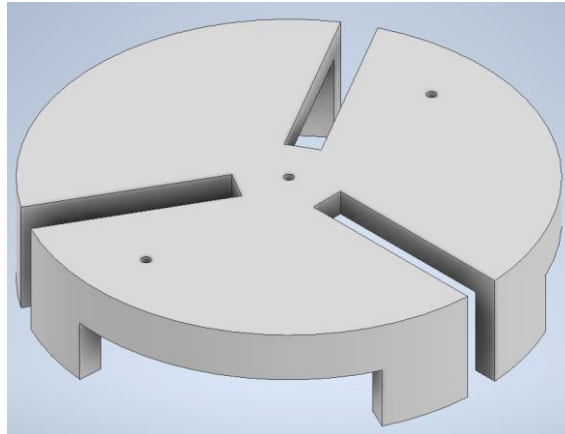
El primer componente que se creó fue la base fija (Figura 8), la cual fue diseñada con una geometría triangular entre los ejes de los servomotores. Asimismo, se consideró una separación angular entre cada uno de estos ejes de 120° . Además, se implementó una estructura interna que reforzó la sujeción de los motores. Posteriormente, se creó la tapadera de esta base (Figura 9), que proporciona estabilidad, facilita el anclaje con la estructura principal y proporciona ventilación a los servomotores.

Figura 8. Base servomotores



Fuente: Elaboración propia.

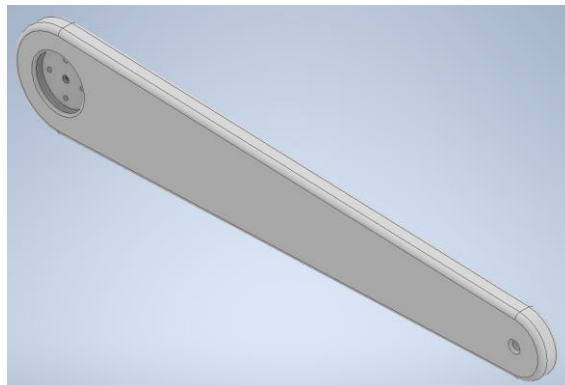
Figura 9. Tapadera base



Fuente: Elaboración propia.

También se diseñó un modelo para los brazos superiores e inferiores, encargados de transmitir el movimiento al efector final. Los brazos superiores tienen una longitud entre ejes de unión de 200mm. Mientras que los brazos inferiores tienen una longitud de 350mm entre los ejes de sujeción y en sus extremos tienen articulaciones tipo rótula, para permitir la libertad de movimiento que requiere la plataforma móvil. Por otro lado, el diseño del eslabón dos, que se muestra en la Figura 11 no es impreso 3D, este es una varilla de 3mm de diámetro y es de acero inoxidable.

Figura 10. Eslabón 1, brazo corto Delta



Fuente: Elaboración propia.

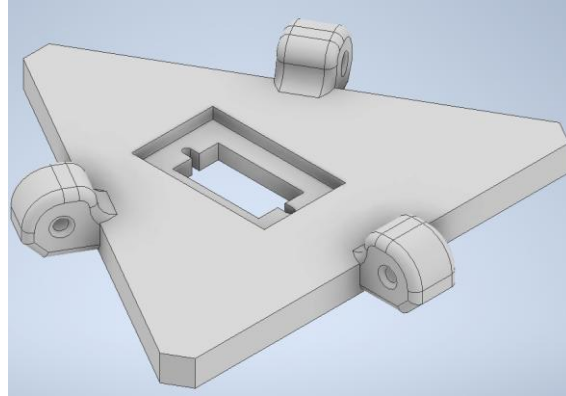
Figura 11. Eslabón 2, brazo largo Delta



Fuente: Elaboración propia.

Para la plataforma móvil, se diseñó una pieza triangular con una separación de 120° entre sus vértices, los cuales coinciden con las dimensiones establecidas en la base estática. Asimismo, como se puede observar en la Figura 12, en el centro de la pieza se incorporó un agujero del tamaño de un servomotor, destinado al anclaje del efector final.

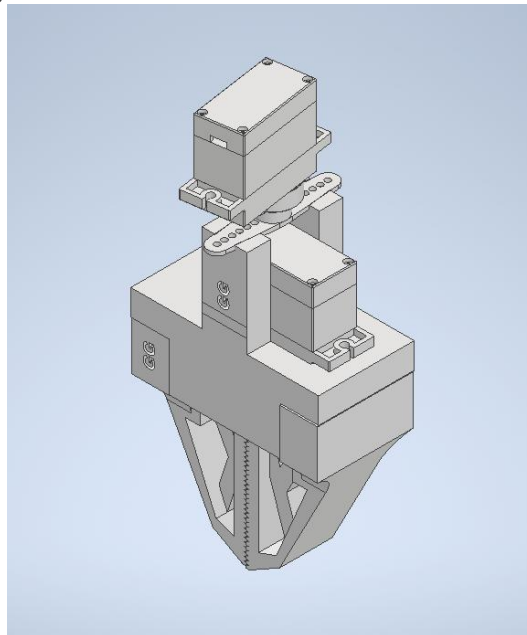
Figura 12. Base móvil



Fuente: Elaboración propia.

Para el diseño del efector final, se adaptó un modelo abierto publicado en la plataforma Cults3D, con una garra integrada en su extremo inferior, el cual facilitó la sujeción de piezas pequeñas y ligeras.

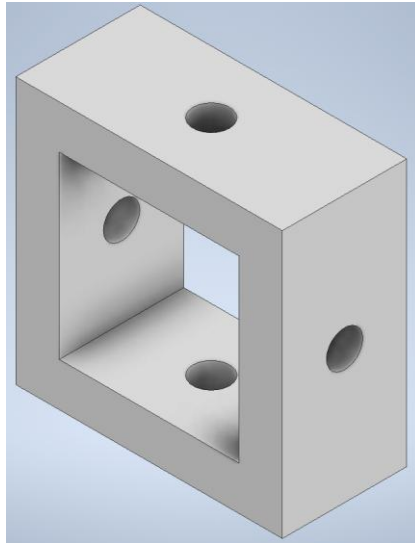
Figura 13. Efector final, garra



Fuente: Elaboración propia.

Adicionalmente, se diseñó una pieza de anclaje estructural, lo cual permitió unir la base fija a la estructura del brazo, proporcionando estabilidad al robot y asegurando una correcta sujeción de las piezas.

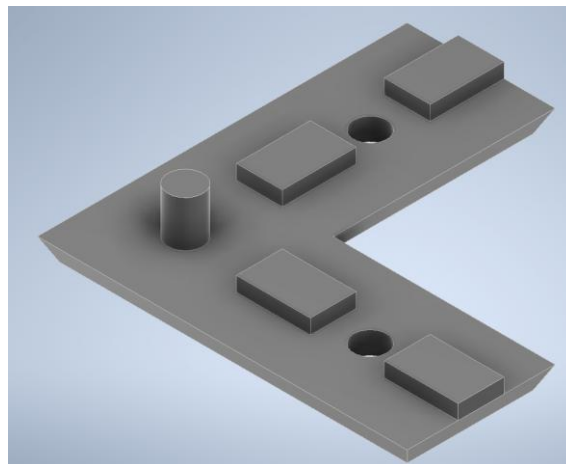
Figura 14. Anclaje estructura



Fuente: Elaboración propia.

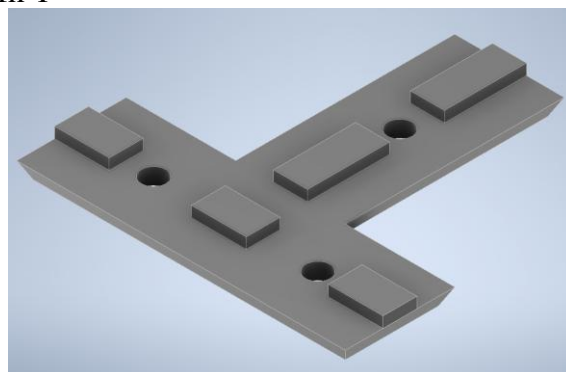
Otro de los diseños creados fueron componentes de anclaje estructural, los cuales dieron refuerzo a las uniones de la estructura principal. Del mismo modo, estos permitieron asegurar con mayor rigidez las uniones de los perfiles que conforman esta estructura.

Figura 15. Conector perfil L



Fuente: Elaboración propia.

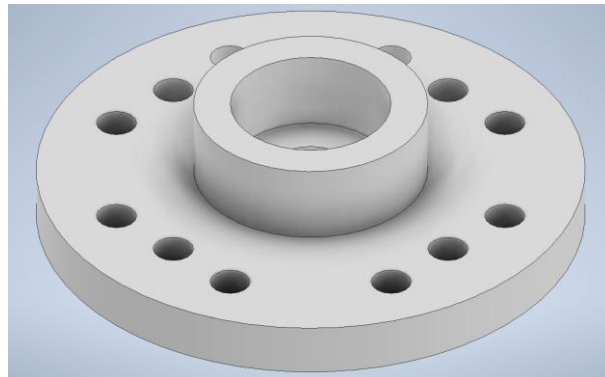
Figura 16. Conector perfil T



Fuente: Elaboración propia.

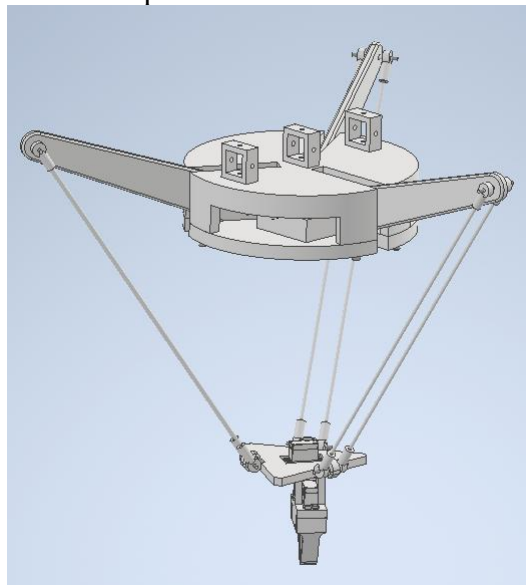
Por último, se diseñó el acople del servomotor, con el fin de facilitar la unión entre el eslabón uno y los servomotores de la base fija. Esta pieza es la encargada de transmitir el movimiento de forma estable, por lo cual se diseñaron múltiples agujeros de sujeción, dándole mayor estabilidad a la unión.

Figura 17. Acople servomotor



Fuente: Elaboración propia.

Figura 18. Ensamblaje del diseño de piezas.



Fuente: Elaboración propia.

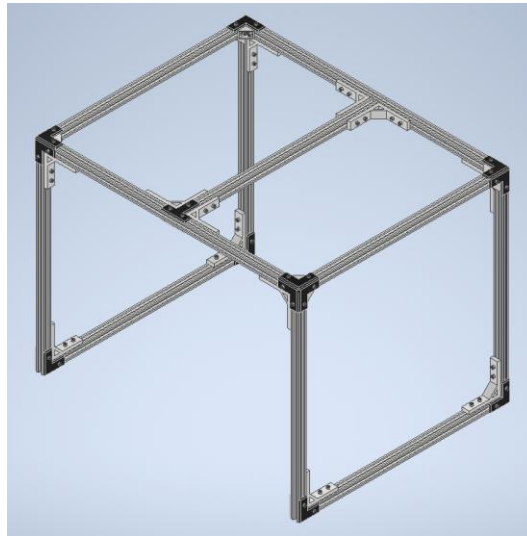
El diseño y modelado de las piezas mecánicas dentro del software Inventor, permitió validar el ensamblaje y evaluar la viabilidad del diseño antes de la etapa de fabricación. Con esto se eliminaba el riesgo de una existencia de colisiones entre piezas. Cada componente fue diseñado tomando en cuenta la geometría de un robot paralelo tipo delta. Las formas y dimensiones establecidas en el diseño, proporcionan estabilidad y distribución de movimiento. Esta fase de diseño constituyó la base fundamental para la fabricación de componentes, mediante impresión 3D o manufactura.

7.2. Diseño estructural y análisis

El diseño de la estructura del brazo Delta se realizó y modeló en Autodesk Inventor, usando perfiles de aluminio tipo 2020 por su bajo peso, rigidez y facilidad de ensamblaje. Este tipo de perfil permitió desarrollar y diseñar una estructura modular, resistente, de fácil armado y desarmado, requerimiento preciso para su debida operación. Asimismo, se establecieron las medidas adecuadas para su espacio predeterminado en el Laboratorio de Procesos de la Universidad del Valle de Guatemala.

En el diseño se colocó la concentración de peso en el centro de la base, simulando el punto de mayor carga correspondiente al anclaje del brazo tipo Delta diseñado en el apartado 9.1.

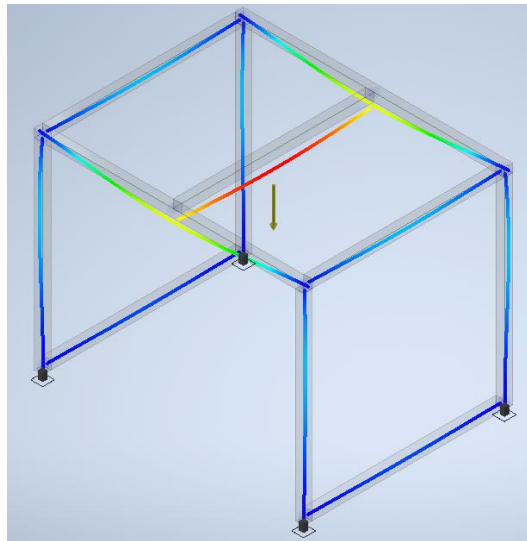
Figura 19. Estructura



Fuente: Elaboración propia.

El análisis mecánico de cargas se realizó mediante el módulo de simulación estática del mismo software donde se diseñaron las piezas. En el sistema Autodesk Inventor se aplicaron las condiciones y parámetros de restricciones correspondientes al montaje real del brazo Delta. Se fijó la base de la estructura, la cual es la que colinda con la mesa y se colocó el peso del brazo (Figura 18). Como se puede observar en la Figura 20 los cálculos estructurales muestran que las concentraciones mayores se encuentran en las uniones de los perfiles superiores paralelos a la base fija. Este análisis muestra que la deformación se mantiene dentro de los rangos admisibles para los perfiles de aluminio 2020.

Figura 20. Cálculo de esfuerzo de estructura



Fuente: Elaboración propia.

El análisis estructural permitió establecer que la estructura diseñada soporta de forma aceptable y adecuada las condiciones de peso y funcionalidad del sistema, antes de su fabricación. Asimismo, se demostró que los perfiles tipo 2020 ofrecen condiciones aceptables entre su peso y resistencia, lo cual asegura la rigidez de la estructura. Además, en la simulación, se establecieron los puntos críticos de unión, tal cual se diseñaron en el apartado 9.1. con soportes en forma L (Figura 15) y T (Figura 16), los cuales refuerzan las uniones críticas y brindan mayor estabilidad a estas, previniendo deformaciones y cumpliendo con los requisitos mecánicos y geométricos necesarios para conformar la estructura del brazo robótico tipo Delta, diseñado anteriormente.

7.3. Impresión de piezas 3D

Para la impresión de las piezas diseñadas en Autodesk Inventor, se utilizó una impresora 3D con material PLA, esto debido a su facilidad de impresión y rigidez estructural. Este material dio como resultado piezas ligeras y resistentes. Las piezas diseñadas anteriormente se imprimieron con una densidad del 20% de relleno. Asimismo, se imprimieron con un patrón giroide, seleccionado por su resistencia mecánica en todas las direcciones de sus ejes, así como por su capacidad de absorber vibraciones, como se menciona en el Marco Teórico en el apartado 7.4.4.

Figura 21. Proceso impresión 3D



Fuente: Elaboración propia.

Los parámetros utilizados para la impresión de estas piezas fueron los siguientes:

Tabla 1. Parámetros establecidos para impresiones 3D

Parámetro	Valor
Material	PLA – 1.75mm
Altura de capa	0.2mm
Densidad de relleno	20%
Patrón de relleno	Giroide
Soportes	Requeridos en ángulos mayores a 35° o agujeros perpendiculares a la base de impresión.

Fuente: Elaboración propia.

Además, las impresiones requirieron parámetros extras como, temperatura de extrusión, de cama, velocidad de impresión y otras, las cuales dependen y varían según el modelo y marca de la impresora utilizada. En este caso, se utilizaron los estándares que proporcionan el Software de la impresora utilizada, la cual fue una Elegoo Neptune 4 Pro, asegurando la correcta adhesión entre capas y la calidad uniforme de las piezas.

El proceso de impresión permitió fabricar con precisión todas las piezas diseñadas en el software Autodesk Inventor, logrando que la superficie de las piezas fuera la esperada y la adhesión de las capas, buena. El uso del patrón giroide para el relleno mostró ser la elección adecuada y eficiente, ya que ofreció una resistencia adecuada en sus ejes y a su vez, el peso de las piezas es reducido. Por otro lado, la combinación del material utilizado y la densidad del relleno también contribuyó a obtener piezas livianas reduciendo el esfuerzo sobre los servomotores.

7.4. Fabricación y ensamblaje de estructura

Para la fabricación de la estructura del brazo se realizó un marco principal el cual se conforma por perfiles de aluminio tipo T 2020 encargado de sostener la base fija, donde se encuentran los servomotores y el conjunto del robot. El proceso inició con el corte y rectificación, mediante mecanizado de los perfiles. Se cortaron con una sierra manual y se rectificaron con un taladro fresador. En el proceso de maquinado se ajustaron las longitudes de los perfiles cortados, coincidiendo así con las dimensiones establecidas en el diseño 3D.

Figura 22. Proceso de corte perfiles T 2020



Fuente: Elaboración propia.

Figura 23. Proceso de rectificado



Fuente: Elaboración propia.

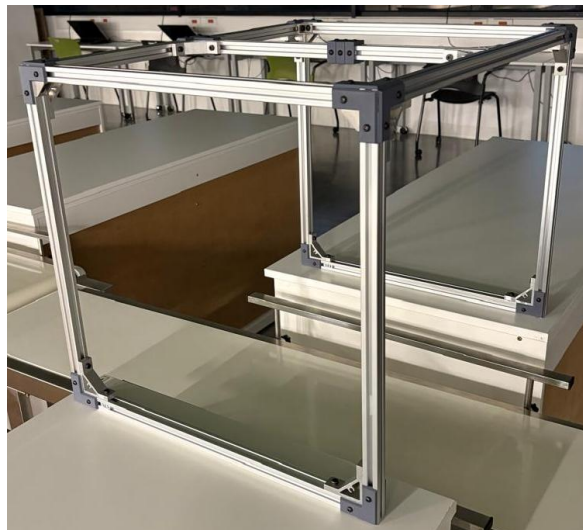
Tabla 2. Medidas y cantidades de perfiles T 2020

Longitud de la pieza (mm)	Cantidad de perfiles (unidades)
650	2
480	9

Fuente: Elaboración propia.

Luego de tener los perfiles cortados y rectificados, se prepararon los elementos de la estructura para comenzar con las conexiones, por medio de conectores de aluminio en 90° y tornillería. Después, se reforzaron las uniones utilizando escuadras de sujeción tipo L y T, junto con tornillos y tuercas tipo T-slot, asegurando la estabilidad, firmeza y ajustabilidad de la estructura.

Figura 24. Montaje final de estructura



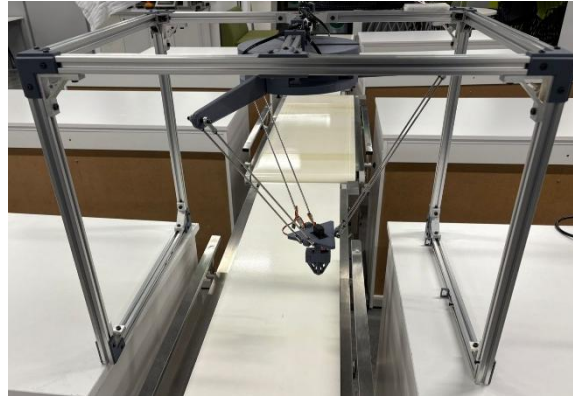
Fuente: Elaboración propia.

7.5. Ensamblaje físico completo del sistema mecánico

Al tener finalizado el proceso de impresión y fabricación de las piezas y de la estructura, se procedió con el ensamblaje del sistema del brazo robótico tipo Delta. El montaje se hizo

uniendo las piezas impresas 3D con el material PLA (bioplástico), compuesta por perfiles 2020, siguiendo el modelo diseñado en Autodesk Inventor, completando así el ensamblaje del brazo Delta.

Figura 25. Ensamblaje físico de Robot Delta



Fuente: Elaboración propia.

Tabla 3. Componentes impresos 3D

Componente	Descripción	Cantidad
Acople servomotor	Transmitir el movimiento del servomotor a los brazos.	3
Anclaje estructura	Permite la unión del brazo Delta con la estructura metálica.	3
Base móvil	Efector final, destinado a soportar la garra.	1
Base servomotores	Base del mecanismo, destinada a la sujeción de servomotores.	1
Conector perfil L	Proporciona soporte y unión a los perfiles de la estructura metálica.	20
Conector perfil T	Proporciona soporte y unión a los perfiles de la estructura metálica.	3
Eslabón 1	Transmite movimiento al eslabón 2.	3
Garra	Sujeción y liberación de piezas ligeras.	1
Tapadera base	Permite la unión de la base de servomotores con los anclajes de la estructura.	1

Fuente: Elaboración propia.

Tabla 4. Componentes adicionales

Componente	Cantidad
Perfil T 2020 480mm	9
Perfil T 2020 650mm	2
Soporte Angular 90 20x20x50mm	12
Soporte Angular 90 60x60mm	8
Tornillo M3x40	6
Tornillo M4x20 cabeza Allen de botón	6
Tornillo M4x65 cabeza Allen cilíndrico	1
Tornillo M5x10 cabeza Allen de botón	16
Tornillo M5x12 cabeza Allen cilíndrico	24
Tornillo M4x10 cabeza Allen de botón	61

Componentes	Cantidad
Tuerca hexagonal M3	42
Tuerca ojo de pescado M3 hembra	12
Tuerca T EU M4	67
Tuerca T M5	40
Varilla 3mm 320mm	6

Fuente: Elaboración propia.

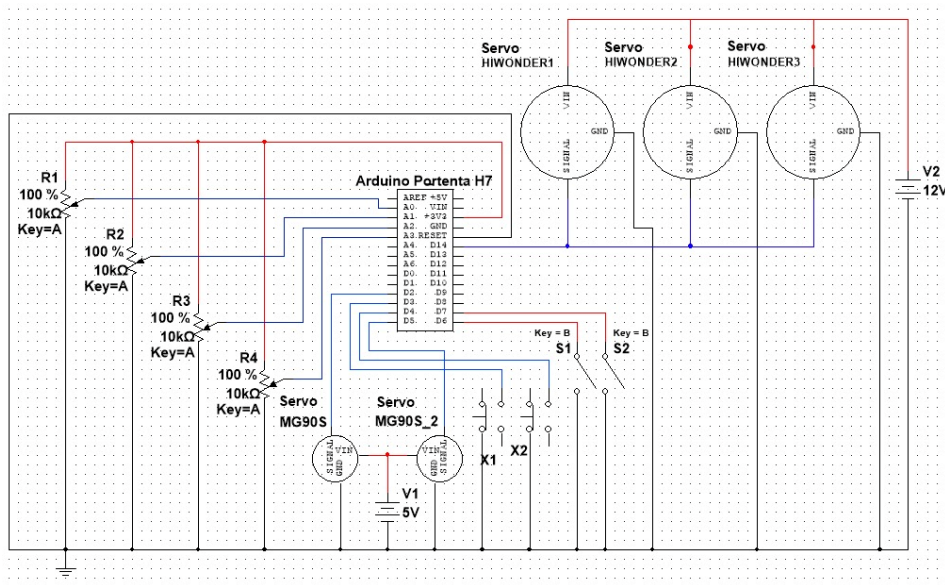
El proceso de ensamblaje permitió validar la compatibilidad de las piezas impresas y los componentes de la estructura, confirmando que las dimensiones diseñadas en Autodesk Inventor se ajustaron correctamente. Además, se comprobó que la unión entre las piezas impresas y la estructura metálica de perfiles tiene la rigidez necesaria para soportar los componentes y el funcionamiento del mecanismo sin tener flexiones, ni deformaciones en ningún componente. Para finalizar, este proceso de ensamblaje validó el sistema mecánico diseñado anteriormente.

7.6. Circuito eléctrico y control manual

Con la finalidad de poder controlar el brazo robótico tipo Delta, mediante la cinemática inversa, armado y descrito en los apartados anteriores, se implementó un sistema de control manual, basado en un circuito eléctrico y código fuente de programación, desarrollado para una placa Arduino Portenta H7. Para este código se usó como referencia “Arduino Controlled Delta Robot” (Arduino Project Hub, 2018), el cuál se modificó y optimizó para adaptarse al hardware utilizado y a los servomotores empleados. Mediante este código se logra accionar el robot, a través de componentes analógicos, los cuales permiten posicionar angularmente cada brazo.

El circuito eléctrico se compone de distintos componentes que trabajan en conjunto para generar las señales necesarias, ya sea de control del robot paralelo tipo Delta, o de su correcta alimentación. En la Figura 26 se muestra el circuito general que se utilizó y se pueden observar las conexiones entre la placa Portenta H7, los componentes y las fuentes de alimentación.

Figura 26. Circuito eléctrico



Fuente: Elaboración propia.

Los componentes eléctricos utilizados para el control manual fueron:

Tabla 5. Componentes eléctricos

Componente	Fabricante	Modelo	Descripción	Cantidad
Botón	-	-	Encargados de abrir o cerrar la garra.	2
Fuente de poder	LUMION	12V150W	Transformación de voltaje 120V a 12VDC.	1
Módulo de alimentación para protoboard	-	MB102	Transformación de 12V a 3.3-5V.	1
Portenta	Arduino	H7	Unidad de control, lectura de señales analógicas y digitales de los componentes.	1
Potenciómetro lineal	-	10kOhm	Transformación de voltaje para transmitir movimiento a los servomotores.	4
Protoboard	-	Galleta completa	Distribución de señales.	1
Servomotor	Befkkip	MG90S	Encargados del movimiento de la garra.	2
Servomotor	HiWonder	HTS-35H	Encargados del movimiento de los brazos del robot tipo Delta.	3
Switch	-	3 pines	Cambio de modalidades en la programación.	1

Fuente: Elaboración propia.

El código implementado que se encuentra en el Anexo B tiene tres modalidades, las cuales se manejan mediante el control manual que fue diseñado y cada elemento eléctrico tiene su funcionalidad específica.

Las modalidades son las siguientes:

- Modalidad 1: Ajuste de posiciones sobre ejes del robot.
Mediante tres potenciómetros se puede ajustar las medidas del efector final en los ejes X, Y, Z. El primer potenciómetro modifica la posición en el eje “X”, el segundo sobre el eje “Y” y el tercero sobre su eje “Z”. Esta modalidad permite que las entradas analógicas de estos componentes se conviertan en señales y así modifiquen los ángulos de los tres servomotores. Esta función permite que la base móvil se desplace dentro del área de trabajo del robot.
- Modalidad 2: Ajuste de ángulos directos.
Mediante tres potenciómetros se pueden controlar los valores de estos, los ángulos directamente de cada uno de los servomotores, modalidad que permite ajustar los

ángulos de forma individual en un rango de 0° a 240°. Esta función permite el movimiento de cada servomotor, para ajustes de ángulos

- Modalidad 3: Envío de coordenadas por comunicación serial.

Mediante el serial del software Arduino IDE y su serial, se pueden enviar coordenadas dentro del espacio de movimiento (X, Y, Z). Con este dato, el código realiza el cálculo de los ángulos necesarios para posicionar los tres servomotores de la base fija, en la posición final de la base móvil.

Además de estas modalidades, al momento de iniciar el robot o cambiar de modo, los tres servomotores se colocan automáticamente en una posición de inicio establecida. También, cada modalidad controla la garra, la cual se acciona y mueve mediante un potenciómetro para su rotación y dos botones para la acción de abrir y cerrar. En las tablas 6 y 7, se pueden observar las variables de entrada y salida utilizadas en el código.

Tabla 6. Variables de entrada

Nombre	Tipo de señal	Tipo de dato	Valores	Descripción
POT_X	Analógica	int	0-1023	Modo 1: control eje X. Modo 2: control ángulo S1.
POT_Y	Analógica	int	0-1023	Modo 1: control eje Y. Modo 2: control ángulo S2.
POT_Z	Analógica	int	0-1023	Modo 1: control eje Z. Modo 2: control ángulo S3.
POT_GARRA	Analógica	int	0-1023	Control rotación del servomotor principal de la garra.
SW1	Digital	int	HIGH/LOW	Controla de la modalidad del código.
SW2	Digital	int	HIGH/LOW	Controla de la modalidad del código.
BTN_ABRIR	Digital	int	HIGH/LOW	Controla el comando para abrir la garra.
BTN_CERRAR	Digital	int	HIGH/LOW	Controla el comando para cerrar la garra.
Serial	Comunicación serial	String	X Y Z	Modo 3: ingreso de coordenadas para mover los 3 servomotores.

Fuente: Elaboración propia.

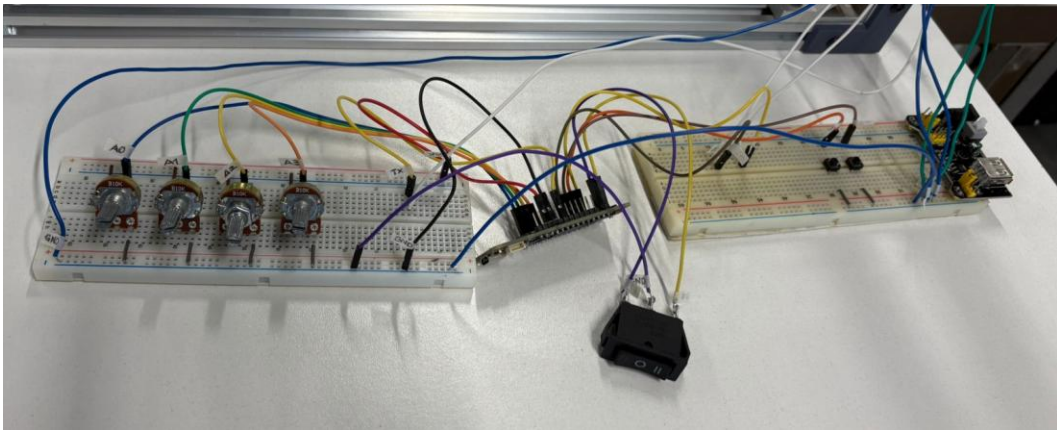
Tabla 7. Variables de salida

Nombre	Tipo de señal	Tipo de dato	Valores	Descripción
Serial 1	Comunicación serial	uint8_t[]	0-1000	Envía los comandos a los servomotores HiWonder HTS-35H.
PWM1	PWM	Pulso us	500-2500 us	Movimiento del servomotor principal de la garra.

Nombre	Tipo de señal	Tipo de dato	Valores	Descripción
PWM2	PWM	Pulso us	500-2500 us	Movimiento de la garra: abrir/cerrar.
Serial	Comunicación serial	Texto	Texto	Muestra el estado de la modalidad, cinemática inversa, ángulos directos o ángulos por coordenadas.

Fuente: Elaboración propia.

Figura 27. Control manual físico



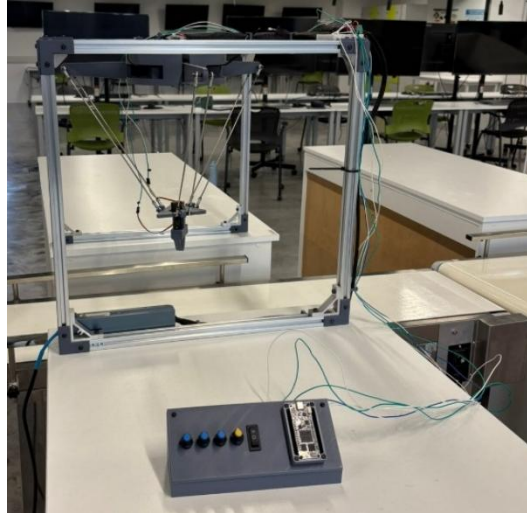
Fuente: Elaboración propia.

El circuito y control manual permitieron comprobar experimentalmente la funcionalidad del brazo tipo Delta, mediante el uso de potenciómetros, botones y comunicación serial. A través del circuito eléctrico, se logró controlar el movimiento del brazo, por medio de sus tres servomotores, ubicados en la base estática. Durante las pruebas, el circuito mostró un comportamiento estable, sin ninguna suspensión en la comunicación entre el circuito y la placa madre (Arduino Portenta H7). Esto garantizó una operación continua del sistema de robot.

Asimismo, la implementación de tres modalidades dentro del control manual, permitió ajustar las posiciones de los servomotores por diferentes métodos, los cuales aportaron ventajas al sistema operativo del robot Delta.

El modo por ejes, accionado por potenciómetros, permitió explorar el espacio de trabajo. El modo por ángulos directos a cada servomotor permitió la calibración de estos y visualizar el comportamiento de cada brazo, mientras que el modo por coordenadas seriales, comprobó la funcionalidad de la cinemática inversa implementada en el código.

Figura 28. Implementación circuito eléctrico y control manual a robot tipo Delta



Fuente: Elaboración propia.

7.7. Precisión, exactitud y tiempo de respuesta

Para obtener los ángulos de cada articulación del brazo Delta, se realizó un código en MATLAB que calcula la cinemática inversa de este, con el enfoque geométrico que se menciona en el apartado 7.3.2. En este código, el cual se encuentra detallado en el Anexo A, se ingresan las coordenadas deseadas de la garra y, a partir de la geometría de los eslabones del robot, se obtienen los ángulos de los servomotores. El objetivo de esto era analizar el movimiento en los tres ejes de la estructura por parte del brazo y verificar que las dimensiones de los eslabones cumplieran con el alcance dentro de este rango del espacio de trabajo.

Por otro lado, con los resultados obtenidos, se llevó a cabo una verificación experimental por medio del código implementado en Arduino IDE, el cual se encuentra en el Anexo B, donde se compararon los datos experimentales para verificar la precisión de ángulos ejecutados por los servomotores. De la misma forma, la exactitud del sistema, comparando los valores teóricos con los valores reales obtenidos de forma experimental, mediante el funcionamiento del brazo.

7.7.1. Precisión

La precisión del sistema se evaluó mediante un estudio de repetividad dentro de la modalidad 3 del programa, estableciendo las coordenadas (X, Y, Z) por medio del serial. Dichas coordenadas permitieron el movimiento experimental del sistema a distintas posiciones. Para esto, se definieron cinco coordenadas distintas, a las cuales el efector final se desplazó repetidamente. En cada una de las posiciones, se registraron los ángulos de los tres servomotores, durante las cinco pruebas.

Figura 29. Resultados pruebas serial Arduino

```

Output Serial Monitor X
Message (Enter to send message to 'Arduino Portenta H7' on 'COM12')

Robot DELTA Portenta H7 | 3 modalidades
Izq: CI (potes X Y Z) | Centro: Angulos directos | Der: Coordenadas (x y z serial IDE)
Modo: 3) Coordenadas (x y z serial IDE)
Coordenadas OK X = 0.0mm Y = 0.0mm Z = -310.0mm | S1 = 123° S2 = 123° S3 = 123°
Coordenadas OK X = 20.0mm Y = 150.0mm Z = -330.0mm | S1 = 85° S2 = 124° S3 = 118°
Coordenadas OK X = -50.0mm Y = -100.0mm Z = -300.0mm | S1 = 139° S2 = 103° S3 = 119°
Coordenadas OK X = 150.0mm Y = 0.0mm Z = -330.0mm | S1 = 108° S2 = 131° S3 = 88°
Coordenadas OK X = -100.0mm Y = -50.0mm Z = -320.0mm | S1 = 124° S2 = 96° S3 = 126°
Coordenadas OK X = 0.0mm Y = 0.0mm Z = -310.0mm | S1 = 123° S2 = 123° S3 = 123°
Coordenadas OK X = 20.0mm Y = 150.0mm Z = -330.0mm | S1 = 85° S2 = 124° S3 = 118°
Coordenadas OK X = -50.0mm Y = -100.0mm Z = -300.0mm | S1 = 139° S2 = 103° S3 = 119°
Coordenadas OK X = 150.0mm Y = 0.0mm Z = -330.0mm | S1 = 108° S2 = 131° S3 = 88°
Coordenadas OK X = -100.0mm Y = -50.0mm Z = -320.0mm | S1 = 124° S2 = 96° S3 = 126°
Coordenadas OK X = 0.0mm Y = 0.0mm Z = -310.0mm | S1 = 123° S2 = 123° S3 = 123°
Coordenadas OK X = 20.0mm Y = 150.0mm Z = -330.0mm | S1 = 85° S2 = 124° S3 = 118°
Coordenadas OK X = -50.0mm Y = -100.0mm Z = -300.0mm | S1 = 139° S2 = 103° S3 = 119°
Coordenadas OK X = 150.0mm Y = 0.0mm Z = -330.0mm | S1 = 108° S2 = 131° S3 = 88°
Coordenadas OK X = -100.0mm Y = -50.0mm Z = -320.0mm | S1 = 124° S2 = 96° S3 = 126°
Coordenadas OK X = 0.0mm Y = 0.0mm Z = -310.0mm | S1 = 123° S2 = 123° S3 = 123°
Coordenadas OK X = 20.0mm Y = 150.0mm Z = -330.0mm | S1 = 85° S2 = 124° S3 = 118°
Coordenadas OK X = -50.0mm Y = -100.0mm Z = -300.0mm | S1 = 139° S2 = 103° S3 = 119°
Coordenadas OK X = 150.0mm Y = 0.0mm Z = -330.0mm | S1 = 108° S2 = 131° S3 = 88°
Coordenadas OK X = -100.0mm Y = -50.0mm Z = -320.0mm | S1 = 124° S2 = 96° S3 = 126°
Coordenadas OK X = 0.0mm Y = 0.0mm Z = -310.0mm | S1 = 123° S2 = 123° S3 = 123°
Coordenadas OK X = 20.0mm Y = 150.0mm Z = -330.0mm | S1 = 85° S2 = 124° S3 = 118°
Coordenadas OK X = -50.0mm Y = -100.0mm Z = -300.0mm | S1 = 139° S2 = 103° S3 = 119°
Coordenadas OK X = 150.0mm Y = 0.0mm Z = -330.0mm | S1 = 108° S2 = 131° S3 = 88°
Coordenadas OK X = -100.0mm Y = -50.0mm Z = -320.0mm | S1 = 124° S2 = 96° S3 = 126°

```

Fuente: Elaboración propia.

Tabla 8. Resultados de la verificación de precisión

Posición	Coordenadas ingresadas ArduinoIDE			Datos pruebas ArduinoIDE		
	Coordenada X	Coordenada Y	Coordenada Z	Ángulo S1	Ángulo S2	Ángulo S3
P1	0	0	-310	123	123	123
	0	0	-310	123	123	123
	0	0	-310	123	123	123
	0	0	-310	123	123	123
	0	0	-310	123	123	123
P2	20	150	-330	85	124	118
	20	150	-330	85	124	118
	20	150	-330	85	124	118
	20	150	-330	85	124	118
	20	150	-330	85	124	118
P3	-50	-100	-300	139	103	119
	-50	-100	-300	139	103	119
	-50	-100	-300	139	103	119
	-50	-100	-300	139	103	119
	-50	-100	-300	139	103	119
P4	150	0	-330	108	131	88

P4	150	0	-330	108	131	88
	150	0	-330	108	131	88
	150	0	-330	108	131	88
	150	0	-330	108	131	88
Posición	Coordenadas ingresadas ArduinoIDE			Datos pruebas ArduinoIDE		
	Coordenada X	Coordenada Y	Coordenada Z	Ángulo S1	Ángulo S2	Ángulo S3
P5	-100	-50	-320	124	96	126
	-100	-50	-320	124	96	126
	-100	-50	-320	124	96	126
	-100	-50	-320	124	96	126
	-100	-50	-320	124	96	126

Fuente: Elaboración propia.

Como se puede observar en la Tabla 8, los resultados obtenidos en la verificación de precisión muestran que no existe ninguna variación entre las repeticiones para cada servomotor. Esto indica que el sistema posee una alta repetitividad y que los servomotores HiWonder responden de manera estable a las señales que se generaron por la placa madre Arduino Portenta H7. Este comportamiento demuestra que el brazo tipo Delta diseñado y construido, mantiene una precisión aceptable en aplicaciones de manipulación básica.

7.7.2. Exactitud

La exactitud del sistema se determinó por medio de una comparación entre los ángulos calculados en MATLAB y el sistema teórico, con los ángulos ejecutados en el entorno Arduino IDE, modelo experimental. Las pruebas se realizaron con las mismas coordenadas para las cinco pruebas de precisión que se utilizaron en el apartado 7.7.1.

Figura 30. Resultados pruebas exactitud MATLAB

```

Command Window
Prueba Cinematica Inversa Matlab

Prueba 1
Datos Válidos:
X = 0mm Y = 0mm Z = -310mm
S1 =123° S2 = 123° S3 = 123°

Prueba 2
Datos Válidos:
X = 20mm Y = 150mm Z = -330mm
S1 =85° S2 = 124° S3 = 118°

Prueba 3
Datos Válidos:
X = -50mm Y = -100mm Z = -300mm
S1 =139° S2 = 103° S3 = 119°

Prueba 4
Datos Válidos:
X = 150mm Y = 0mm Z = -330mm
S1 =108° S2 = 131° S3 = 88°

Prueba 5
Datos Válidos:
X = -100mm Y = -50mm Z = -320mm
S1 =124° S2 = 96° S3 = 126°

```

Fuente: Elaboración propia.

Tabla 9. Resultados exactitud

Posición	Coordenadas ingresadas ArduinoIDE			Datos experimentales ArduinoIDE			Datos teóricos MATLAB		
	Coor. X	Coor. Y	Coor. Z	Ángulo S1	Ángulo S2	Ángulo S3	Ángulo S1	Ángulo S2	Ángulo S3
P1	0	0	-310	123	123	123	123	123	123
P2	20	150	-330	85	124	118	85	124	118
P3	-50	-100	-300	139	103	119	139	103	119
P4	150	0	-330	108	131	88	108	131	88
P5	-100	-50	-320	124	96	126	124	96	126

Fuente: Elaboración propia.

Como se puede observar en la Tabla 9, se tiene un promedio de los resultados experimentales y los resultados teóricos obtenidos mediante el código de cinemática inversa por el método geométrico implementado en el Software Matlab. Y al analizar los datos, se determinó que no existe discrepancia alguna entre los datos teóricos y experimentales. Existe una alta concordancia entre el modelo geométrico y el sistema físico. Lo cual validó que el código de cinemática inversa implementado describe de manera precisa y exacta el comportamiento del robot Delta.

7.7.3. Tiempo de respuesta

El tiempo de respuesta se determinó en el código, el cual se encuentra detallado en el Anexo B del documento. En este se puede observar que el tiempo de respuesta del sistema se estableció en 0.8 segundos por movimiento completo, es decir el tiempo establecido y estimado en que los servomotores de la base estática alcancen su posición objetivo. Durante las pruebas de precisión y exactitud, se pudo observar que los movimientos del brazo Delta se ejecutaron de forma fluida, rápida y alcanzaron su posición esperada. Por este medio, se pudo afirmar que el sistema presenta un desempeño estable y adecuado para para su modalidad de control manual en sus tres modos.

8. Conclusiones

1. Se logró diseñar una estructura mecánica del brazo robótico tipo Delta, la cual cumplió con los requerimientos geométricos del mecanismo, garantizando su estabilidad y precisión. Los análisis de esfuerzo y desplazamiento demostraron que la estructura y diseño realizado y fabricado soporta las condiciones de trabajo de este.
2. Se implementó un sistema de control manual básico, mediante un circuito eléctrico compuesto por potenciómetros y botones, controlados por la placa de Arduino Portenta H7, permitiendo el correcto control y funcionamiento de los servomotores de los brazos y garra. El código implementa múltiples funcionalidades, las cuales comprobaron la funcionalidad del brazo robótico tipo Delta.
3. Mediante pruebas experimentales, se validó por repeticiones una alta precisión en la repetitividad del sistema. También, por medio de una comparación teórica y experimental aplicando cinemática inversa, se comprobó que el sistema presenta una exactitud y estabilidad apropiadas, estableciendo una base adecuada para futuras mejoras y optimización del sistema.
4. Durante el proceso de realización del brazo robótico tipo Delta, se hizo la documentación completa del proyecto, incluyendo planos técnicos, las configuraciones de impresión 3D, códigos de control y el algoritmo de implementación de la cinemática inversa. Este material se establece como referencia técnica que permitirá futuras mejoras o replicas del sistema en el Laboratorio de Diseño de Procesos.

9. Recomendaciones

1. En caso de rupturas o deformaciones en las piezas impresas en 3D, se recomienda utilizar como mínimo el mismo material y parámetros definidos en la fabricación en este proyecto. Garantizando el óptimo funcionamiento de las sus propiedades mecánicas y funcionamiento ya establecido.
2. El circuito eléctrico y cableado del sistema es adecuado para las pruebas establecidas en este trabajo, sin embargo, se recomienda diseñar un sistema de cableado modular, mediante conexiones rápidas que faciliten el mantenimiento y armado de los componentes eléctricos a futuro.
3. Para una futura mejora del sistema, se recomienda realizar un estudio del consumo de corriente eléctrica de cada componente que conforma el sistema y el control manual, con el fin de incorporar un sistema de control inalámbrico que permite reducir interferencias y proporcione una mejora en la estabilidad de comunicación entre componentes.
4. Incorporar un sistema de automatización mediante reconocimiento de imágenes, por el cual el robot pueda detectar diversos objetos, realizar tareas de reconocimiento y posicionamiento automático de ángulos dependiendo de este. Lo cual, implementaría mejoras significativas al sistema de operación del robot tipo Delta por medio aplicaciones más avanzadas de clasificación y operación en un futuro.
5. Con el objetivo de aumentar la precisión de la posición del efector final, se recomienda implementar un código de cinemática inversa más avanzado, comparando su desempeño con el modelo actual. Garantizando una operación más segura, mayor estabilidad y precisión.

10. Bibliografía

¿Qué es la impresión 3D y cómo funciona? (2024). REPSOL.

<https://www.repsol.com/es/energia-avanzar/innovacion/impresion-3d/index.cshtml>

Alashgar, E. H. A. (2013). *Modeling and High Precision Motion Control of 3 DOF Parallel*

Delta Robot Manipulator. [https://www.semanticscholar.org/paper/Modeling-and-High-Precision-Motion-Control-of-3-DOF-](https://www.semanticscholar.org/paper/Modeling-and-High-Precision-Motion-Control-of-3-DOF-Alashgar/1ec519af04d1b62e23c184c83cc4fa2e3000343e)

[Alashgar/1ec519af04d1b62e23c184c83cc4fa2e3000343e](https://www.semanticscholar.org/paper/Modeling-and-High-Precision-Motion-Control-of-3-DOF-Alashgar/1ec519af04d1b62e23c184c83cc4fa2e3000343e)

Arduino controlled delta robot. (2018). Arduino Project Hub.

<https://projecthub.arduino.cc/geordag/arduino-controlled-delta-robot-417f96>

Díaz-Rodríguez, M., Carretero, J. A., Bautista-Quintero, R., & Chacón, R. (s. f.). *Cinemática y dinámica inversa de un robot paralelo 3-PRS*.

<https://www.redalyc.org/journal/5075/507554974009/html/>

Díaz-Rodríguez, M., Carretero, J. A., Bautista-Quintero, R., & Chacón, R. (s. f.). *Cinemática y dinámica inversa de un robot paralelo 3-PRS*.

<https://www.redalyc.org/journal/5075/507554974009/html/>

GG_CADworks. (s.f.). *Gripper SG90 (2DOF)* [Archivo STL]. Cults3D.

<https://cults3d.com/es/modelo-3d/variado/gripper-sg90-2dof>

Impresoras3d.com. (2024). *El mejor patrón de relleno para Impresión 3D*.

impresoras3d.com. <https://www.impresoras3d.com/el-mejor-patron-de-relleno-para-impresion-3d>

M, A. (2023). *Los patrones de relleno en la impresión 3D, ¿qué tipos hay y para qué se*

utilizan? 3Dnatives. <https://www.3dnatives.com/es/guia-patrones-relleno-impresion-3d-170720232/#!>

Qué es la robótica y cuáles son sus principales usos - Universidad ORT Uruguay. (s. f.).

<https://fi.ort.edu.uy/blog/que-es-la-robotica-y-cuales-son-sus-usos>

RdR, & RdR. (2023). *Robots Delta. Guía comparativa de marcas con sus características.*

REVISTA DE ROBOTS. <https://revistaderobots.com/robots-y-robotica/robot-delta-aplicaciones-y-precios/?cn-reloaded=1>

Relleno en la impresión 3D | Xometry Pro. (2024, 5 enero). Xometry Pro.

<https://xometry.pro/es/articulos/impresion-3d-relleno>

Robots Paralelos Delta | Automatización Rápida de igus. (s. f.).

[https://www.igus.es/roboLink/delta-](https://www.igus.es/roboLink/delta-robot#:~:text=Un%20robot%20paralelo%2C%20tambi%C3%A9n%20conocido,configuraci%C3%B3n%20en%20forma%20de%20paralelogramo.)

[robot#:~:text=Un%20robot%20paralelo%2C%20tambi%C3%A9n%20conocido,configuraci%C3%B3n%20en%20forma%20de%20paralelogramo.](https://www.igus.es/roboLink/delta-robot#:~:text=Un%20robot%20paralelo%2C%20tambi%C3%A9n%20conocido,configuraci%C3%B3n%20en%20forma%20de%20paralelogramo.)

Rodríguez, M. (2024). *¿Sabes en robótica la diferencia entre grados de libertad y grados de*

movilidad? Canal Gestión Integrada. <https://www.inesem.es/revistadigital/gestion-integrada/diferencia-robotica-gradolibertad-movilidad-3/>

Systemes, D. (2022). *Impresión 3D.* Dassault Systemes.

<https://www.3ds.com/es/make/guide/process/3d-printing>

zDynamics. (2021). *CINEMÁTICA DIRECTA DE #ROBOTS PARALELOS (I) | zDynamics*

[Vídeo]. YouTube. <https://www.youtube.com/watch?v=L3dSf3Se45o>

Zenteno Lara, D. (2022). *Análisis Cinemático y Dinámico de Robots Tipo Delta.* zDynamics.

<https://www.zdynamics.org/docs/ParalelosDelta.pdf>

11. Anexos

11.1. Anexo A: Código cinemática inversa en MATLAB.

```
clear; clc;

% Parámetros específicos diseño robot Delta
a = 200.0;
b = 350.0;
R = 79.204;
r = 69.859;

% Angulo Servomotores para que los eslabones "a" esten alineados
% con la horizontal
offset_S1 = 130;
offset_S2 = 130;
offset_S3 = 130;

% Limite para bajar del brazo
Limite_Z = -337.7;

function angulo_servo = AjusteAngular(Angulo_t, Angulo_offset)
    angulo_servo = round(Angulo_offset - Angulo_t);
    % Se limita a los angulos de los servos
    angulo_servo = max(0, min(240, angulo_servo));
end

% Prueba 1
X_P1 = 0;
Y_P1 = 0;
Z_P1 = -310;
[ver_P1, t1_P1, t2_P1, t3_P1] = CI_RobotDelta(X_P1, Y_P1, ...
    max(Z_P1, Limite_Z), r, R, b, a);

% Prueba 2
X_P2 = 20;
Y_P2 = 150;
Z_P2 = -330;
[ver_P2, t1_P2, t2_P2, t3_P2] = CI_RobotDelta(X_P2, Y_P2, ...
    max(Z_P2, Limite_Z), r, R, b, a);

% Prueba 3
X_P3 = -50;
Y_P3 = -100;
Z_P3 = -300;
[ver_P3, t1_P3, t2_P3, t3_P3] = CI_RobotDelta(X_P3, Y_P3, ...
    max(Z_P3, Limite_Z), r, R, b, a);
```

```

% Prueba 4
X_P4 = 150;
Y_P4 = 0;
Z_P4 = -330;
[ver_P4, t1_P4, t2_P4, t3_P4] = CI_RobotDelta(X_P4, Y_P4, ...
    max(Z_P4,Limite_Z), r, R, b, a);

% Prueba 5
X_P5 = -100;
Y_P5 = -50;
Z_P5 = -320;
[ver_P5, t1_P5, t2_P5, t3_P5] = CI_RobotDelta(X_P5, Y_P5, ...
    max(Z_P5,Limite_Z), r, R, b, a);

fprintf('Prueba Cinematica Inversa Matlab')

% Verificacion P1
if ver_P1 == 0
    S1_P1 = AjusteAngular(t1_P1, offset_S1);
    S2_P1 = AjusteAngular(t2_P1, offset_S2);
    S3_P1 = AjusteAngular(t3_P1, offset_S3);
    fprintf('\n \nPrueba 1 \n');
    fprintf(['Datos Válidos: \nX = %dmm Y = %dmm Z = %dmm \nS1 =%d° ' ...
        ' S2 = %d° S3 = %d°\n'],X_P1,Y_P1,Z_P1,S1_P1,S2_P1,S3_P1);
else
    fprintf(['\nEl dato ingresado no es un punto dentro de los rangos ' ...
        ' aceptados por la geometría del robot Delta. \n ' ...
        'X = %dmm Y = %dmm Z = %dmm\n'],X_P1,Y_P1,Z_P1);
end

% Verificacion P2
if ver_P2 == 0
    S1_P2 = AjusteAngular(t1_P2, offset_S1);
    S2_P2 = AjusteAngular(t2_P2, offset_S2);
    S3_P2 = AjusteAngular(t3_P2, offset_S3);
    fprintf('\nPrueba 2 \n');
    fprintf(['Datos Válidos: \nX = %dmm Y = %dmm Z = %dmm \nS1 =%d° ' ...
        ' S2 = %d° S3 = %d°\n'],X_P2,Y_P2,Z_P2,S1_P2,S2_P2,S3_P2);
else
    fprintf(['\nEl dato ingresado no es un punto dentro de los rangos ' ...
        ' aceptados por la geometría del robot Delta. \n ' ...
        'X = %dmm Y = %dmm Z = %dmm\n'],X_P2,Y_P2,Z_P2);
end

```

```

% Verificacion P3
if ver_P3 == 0
    S1_P3 = AjusteAngular(t1_P3, offset_S1);
    S2_P3 = AjusteAngular(t2_P3, offset_S2);
    S3_P3 = AjusteAngular(t3_P3, offset_S3);
    fprintf('\nPrueba 3 \n');
    fprintf(['Datos Válidos: \nX = %dmm Y = %dmm Z = %dmm \nS1 =%d° ' ...
            ' S2 = %d° S3 = %d°\n'],X_P3,Y_P3,Z_P3,S1_P3,S2_P3,S3_P3);
else
    fprintf(['\nEl dato ingresado no es un punto dentro de los rangos ' ...
            ' aceptados por la geometría del robot Delta. \n ' ...
            ' X = %dmm Y = %dmm Z = %dmm\n'],X_P3,Y_P3,Z_P3);
end

% Verificacion P4
if ver_P4 == 0
    S1_P4 = AjusteAngular(t1_P4, offset_S1);
    S2_P4 = AjusteAngular(t2_P4, offset_S2);
    S3_P4 = AjusteAngular(t3_P4, offset_S3);
    fprintf('\nPrueba 4 \n');
    fprintf(['Datos Válidos: \nX = %dmm Y = %dmm Z = %dmm \nS1 =%d° ' ...
            ' S2 = %d° S3 = %d°\n'],X_P4,Y_P4,Z_P4,S1_P4,S2_P4,S3_P4);
else
    fprintf(['\nEl dato ingresado no es un punto dentro de los rangos ' ...
            ' aceptados por la geometría del robot Delta. \n ' ...
            ' X = %dmm Y = %dmm Z = %dmm\n'],X_P4,Y_P4,Z_P4);
end

% Verificacion P5
if ver_P5 == 0
    S1_P5 = AjusteAngular(t1_P5, offset_S1);
    S2_P5 = AjusteAngular(t2_P5, offset_S2);
    S3_P5 = AjusteAngular(t3_P5, offset_S3);
    fprintf('\nPrueba 5 \n');
    fprintf(['Datos Válidos: \nX = %dmm Y = %dmm Z = %dmm \nS1 =%d° ' ...
            ' S2 = %d° S3 = %d°\n'],X_P5,Y_P5,Z_P5,S1_P5,S2_P5,S3_P5);
else
    fprintf(['\nEl dato ingresado no es un punto dentro de los rangos ' ...
            ' aceptados por la geometría del robot Delta. \n ' ...
            ' X = %dmm Y = %dmm Z = %dmm\n'],X_P5,Y_P5,Z_P5);
end

function [verificacion, t1, t2, t3] = CI_RobotDelta(x, y, z, r, R, b, a)
% Cinematica Inversa

sqrt3 = sqrt(3.0);
sin120 = sqrt3/2.0;
cos120 = -0.5;

[s1, t1] = AnguloPlanoYZ(x, y, z, r, R, b, a);
[s2, t2] = AnguloPlanoYZ(x*cos120 + y*sin120, y*cos120 - x*sin120, ...
    z, r, R, b, a);
[s3, t3] = AnguloPlanoYZ(x*cos120 - y*sin120, y*cos120 + x*sin120, ...
    z, r, R, b, a);

verificacion = s1 + s2 + s3; % Si la suma da 0, los
                            % valores son válidos

end

function [stat, theta] = AnguloPlanoYZ(x0, y0, z0, radio_r, ...
    radio_R, eslabon_b, eslabon_a)

tan30 = 0.57735;

y1 = -0.5 * tan30 * radio_R;
y0 = y0 - 0.5 * tan30 * radio_r;

a = (x0*x0 + y0*y0 + z0*z0 + eslabon_a*eslabon_a - ...
    eslabon_b*eslabon_b - y1*y1) / (2.0*z0);
b = (y1 - y0) / z0;

d = -(a + b*y1)^2 + eslabon_a * (b*b*eslabon_a + eslabon_a);
if d < 0
    stat = -1; theta = 0;
    return;
end

yj = (y1 - a*b - sqrt(d)) / (b*b + 1.0);
zj = a + b*yj;

theta = atan2d(-zj, (y1 - yj));
if (yj > y1), theta = theta + 180.0; end

if theta < -180 || theta > 180
    stat = -1;
else
    stat = 0;
end
end
end

```

11.2. Anexo B: Código principal Arduino IDE.

```
// Librerías principales
#include <Arduino.h>
#include "mbed.h"

// UART servomotores
HardwareSerial& BUS = Serial1;
const uint32_t BUS_BAUD = 115200;

// IDs y tiempo
const uint8_t ID_Servo1 = 1, ID_Servo2 = 2, ID_Servo3 = 3;
const uint16_t MOVE_TIME_MS = 800;

// Potenciómetros para movimiento es coordenadas X, Y, Z. Modo 1
const int POT_X = A0;
const int POT_Y = A1;
const int POT_Z = A2;

// Switch para cambiar de modos
const int SW1 = D6;
const int SW2 = D7;
const unsigned long HOLD_MS = 150;

enum { Modo_INDF = 0, Modo_CI = 1, Modo_ANG = 2, Modo_COOR = 3 };

// Ajuste ejes (Medidas para poner ejes en posición correcta)
const bool InterCambiar_XY = false;
const int Cambiar_X = -1;
const int Cambiar_Y = -1;
const float Rango_XY = 200.0f;

// Geometría Delta (mm)
const float e = 69.859f; // r
const float f = 79.204f; // R
const float re = 350.0f; // b
const float rf = 200.0f; // a

// Constantes trigonométricas para cálculos CI
const float Raiz_3 = 1.732050807f;
const float pi = 3.141592653f;
const float seno_120 = Raiz_3/2.0f;
const float coseno_120 = -0.5f;

// Offset de servomotores, para colocarlos a la horizontal al modo de inicio
// y colocar los 0 en su posición
const int offset_1 = 130;
const int offset_2 = 130;
const int offset_3 = 130;

// Límite eje Z
const float Nivel_mesa = -360.0f; //337
float Limite_Z = Nivel_mesa;

// Variables garra
const int GARRA_S1 = D2;
const int GARRA_S2 = D5;
const int POT_GARRA = A3;
const int BTN_ABRIR = D3; // Input Pullup
const int BTN_CERRAR = D4; // Input Pullup

mbed::PwmOut pwm1(digitalPinToPinName(GARRA_S1));
mbed::PwmOut pwm2(digitalPinToPinName(GARRA_S2));

const int PERIODO_us = 20000;
const int MIN_us = 500;
const int MAX_us = 2500;

// Angulos
const int ANG_MIN = 10;
const int ANG_MAX = 180;
const int ANG_CENTRO = 90;
const int ANG_ABIERTO = 90;
const int ANG_CERRADO = 180;

const float ALPHA_POT = 0.20f; // Filtro IIR
const int STEP_DEG = 3; // Rampa

float FiltroPOT = 512.0f;
int Angulo1_centro = ANG_CENTRO;
int Angulo1_buscado = ANG_CENTRO;
int Angulo2_cerrado = ANG_CERRADO;

unsigned long t_ultAbierto=0, t_ultCerrado=0;
const unsigned long DEBOUNCE_MS = 30;
bool Ult_Abierto = HIGH, Ult_Cerrado = HIGH;

// Protocolo Hiwonder envio - Sale de la página de Hiwonder
static uint8_t checksum(uint8_t id, uint8_t len, uint8_t cmd, const uint8_t *p,
uint8_t n) {
    uint16_t s = id + len + cmd;
    for (uint8_t i=0; i<n; i++) s += p[i];
    return (uint8_t)(~(s & 0xFF));
}
```

```

static void sendPacket(uint8_t id, uint8_t cmd, const uint8_t *params, uint8_t n) {
    uint8_t len = (uint8_t)(n + 3);
    uint8_t csum = checksum(id, len, cmd, params, n);
    BUS.write(0x55); BUS.write(0x55);
    BUS.write(id); BUS.write(len); BUS.write(cmd);
    if (n) BUS.write(params, n);
    BUS.write(csum);
    BUS.flush();
}

// Conversión de ángulos 0°-240° -> 0-1000
static uint16_t degToPos(float deg) {
    if (deg < 0) deg = 0; if (deg > 240) deg = 240;
    return (uint16_t)lroundf(deg * (1000.0f/240.0f));
}

static void moveTimeWrite(uint8_t id, uint16_t pos, uint16_t time_ms) {
    uint8_t p[4] = {
        (uint8_t)(pos & 0xFF), (uint8_t)(pos >> 8),
        (uint8_t)(time_ms & 0xFF), (uint8_t)(time_ms >> 8)
    };
    sendPacket(id, 0x01, p, 4); // SERVO_MOVE_TIME_WRITE
}

// Cinemática Inversa plano YZ
int calc_Ang_YZ(float x0, float y0, float z0, float &theta) {
    float y1 = -0.5f * 0.57735f * f;
    y0 -= 0.5f * 0.57735f * e;
    float a = (x0*x0 + y0*y0 + z0*z0 + rf*rf - re*re - y1*y1) / (2.0f*z0);
    float b = (y1 - y0) / z0;
    float d = -(a + b*y1)*(a + b*y1) + rf*(b*b*rf + rf);
    if (d < 0) return -1;
    float yj = (y1 - a*b - sqrtf(d)) / (b*b + 1.0f);
    float zj = a + b*yj;
    theta = 180.0f * atanf(-zj / (y1 - yj)) / pi + ((yj > y1) ? 180.0f : 0.0f);
    if (theta < -180.0f || theta > 180.0f) return -1;
    return 0;
}

// Calculo Cinematica Inversa
int calc_CI(float x0, float y0, float z0, float &t1, float &t2, float &t3) {
    t1 = t2 = t3 = 0.0f;
    int s1 = calc_Ang_YZ(x0, y0, z0, t1);
    int s2 = calc_Ang_YZ(x0*coseno_120 + y0*seno_120, y0*coseno_120 - x0*seno_120, z0, t2);
    int s3 = calc_Ang_YZ(x0*coseno_120 - y0*seno_120, y0*coseno_120 + x0*seno_120, z0, t3);
    return s1 + s2 + s3;
}

static int servoDegFromTheta(float thetaDeg, int offsetDeg) {
    int ang = offsetDeg - (int)lroundf(thetaDeg);
    if (ang < 0) ang = 0;
    if (ang > 240) ang = 240;
    return ang;
}

// Diferentes modalidades del control
static int readSwitchModeStable() {
    static int lastMode = Modo_INDF;
    static unsigned long tEdge = 0;

    int SW_D6 = digitalRead(SW1);
    int SW_D7 = digitalRead(SW2);

    int rawMode = Modo_INDF;
    if (SW_D6 == LOW && SW_D7 == HIGH) rawMode = Modo_CI;
    else if (SW_D6 == HIGH && SW_D7 == HIGH) rawMode = Modo_ANG;
    else if (SW_D6 == HIGH && SW_D7 == LOW) rawMode = Modo_COOR;
    else rawMode = Modo_INDF;

    if (rawMode != lastMode && (millis() - tEdge) >= HOLD_MS) {
        lastMode = rawMode;
        tEdge = millis();
        if (Serial) {
            Serial.print("Modo: ");
            if (lastMode == Modo_CI) Serial.println("Modo 1 = CI (potes X Y Z)");
            else if (lastMode == Modo_ANG) Serial.println("Modo 2 = Angulos directos");
            else if (lastMode == Modo_COOR) Serial.println("Modo 3 = Coordenadas por Serial (X Y Z)");
            else Serial.println("Indefinido");
        }
    }
    return lastMode;
}

// Modo 3: Coordenadas enviadas desde el serial
static bool handleConsoleMove() {
    if (!Serial.available()) return false;

    String line = Serial.readStringUntil('\n');
    line.trim();
    if (line.length() == 0) return false;

    float xin, yin, zin;
    int matched = 0;
}

```

```

{
  const char* c = line.c_str();
  float a,b,c3;
  if (sscanf(c, "%f %f %f", &a, &b, &c3) == 3) { xin=a; yin=b; zin=c3; matched=1; }
}
if (!matched) {
  const char* c = line.c_str();
  float a,b,c3;
  if (sscanf(c, "x=%f y=%f z=%f", &a,&b,&c3) == 3) { xin=a; yin=b; zin=c3; matched=1; }
}

if (!matched) { Serial.println("Formato invalido. Usa: x y z o x= y= z="); return false; }

if (zin < Limite_Z) zin = Limite_Z;

float t1,t2,t3;
int ret = calc_CI(xin, yin, zin, t1, t2, t3);
if (ret != 0) {
  Serial.print("CI invalida para X = "); Serial.print(xin,1);
  Serial.print(" Y = "); Serial.print(yin,1);
  Serial.print(" Z = "); Serial.println(zin,1);
  return true;
}

int s1 = servoDegFromTheta(t1, offset_1);
int s2 = servoDegFromTheta(t2, offset_2);
int s3 = servoDegFromTheta(t3, offset_3);

moveTimeWrite(ID_Servo1, degToPos(s1), MOVE_TIME_MS);
moveTimeWrite(ID_Servo2, degToPos(s2), MOVE_TIME_MS);
moveTimeWrite(ID_Servo3, degToPos(s3), MOVE_TIME_MS);

Serial.print("OK X = "); Serial.print(xin,1); Serial.print(" Y = ");
Serial.print(yin,1); Serial.print(" Z = "); Serial.print(zin,1);
Serial.print(" | S1 = "); Serial.print(s1);
Serial.print(" S2 = "); Serial.print(s2);
Serial.print(" S3 = "); Serial.println(s3);
return true;
}

// Funciones auxiliares de la garra
static inline int clampInt(int v, int lo, int hi) {
  return (v < lo) ? lo : (v > hi) ? hi : v;
}
static inline int angleToUs(int ang) {
  ang = clampInt(ang, 0, 180);
  long span = (long)(MAX_uS - MIN_uS);
  return MIN_uS + (int)(span * (long)ang / 180L);
}
static inline void writeServoUs(mbed::PwmOut &pwm, int us) {
  pwm.pulsewidth_us(clampInt(us, MIN_uS, MAX_uS));
}
int readOversampledAnalog(uint8_t pin, uint8_t samples = 8) {
  long acc = 0;
  for (uint8_t i=0; i<samples; i++) { acc += analogRead(pin); delayMicroseconds(200); }
  return (int)(acc / samples);
}

// Setup
void setup() {
  // Entradas Delta
  pinMode(SW1, INPUT_PULLUP);
  pinMode(SW2, INPUT_PULLUP);

  // UART HiWonder
  BUS.begin(BUS_BAUD);

  // IDE Arduino
  Serial.begin(115200);
  unsigned long t0 = millis();
  while (!Serial && (millis() - t0) < 3000) {}
  Serial.println("Sistema combinado: Delta (HTS-35H) + Servos PWM (pot + 2 botones)");

  // PWM servos garra
  analogReadResolution(10);
  pinMode(BTN_ABRRIR, INPUT_PULLUP);
  pinMode(BTN_CERRAR, INPUT_PULLUP);
  pwm1.period_us(PERIODO_uS);
  pwm2.period_us(PERIODO_uS);

  // Mover Servo 180 garra
  int potRaw = readOversampledAnalog(POT_GARRA, 16);
  FiltroPOT = (float)potRaw;
  Angulo1_buscado = map(potRaw, 0, 1023, ANG_MIN, ANG_MAX);
  Angulo1_buscado = clampInt(Angulo1_buscado, ANG_MIN, ANG_MAX);
  Angulo1_centro = Angulo1_buscado;
  writeServoUs(pwm1, angleToUs(Angulo1_centro));
  Angulo2_cerrado = ANG_CERRADO;
  writeServoUs(pwm2, angleToUs(Angulo2_cerrado));
}

```

```

// Loop
void loop() {
  // Cambio de modos
  int mode = readSwitchModeStable();

  if (mode == Modo_COOR) {
    handleConsoleMove();
  } else {
    static float x=0, y=0, z=-337.0f;
    static int a1=0, a2=0, a3=0;

    if (mode == Modo_CI) {
      float x_in = (float)analogRead(POT_X) * (2.0f*Rango_XY) / 1023.0f - Rango_XY;
      float y_in = (float)analogRead(POT_Y) * (2.0f*Rango_XY) / 1023.0f - Rango_XY;

      float x_corr = x_in, y_corr = y_in;
      if (InterCambiar_XY) { float t = x_corr; x_corr = y_corr; y_corr = t; }
      x = Cambiar_X * x_corr;
      y = Cambiar_Y * y_corr;

      z = (float)analogRead(POT_Z) * 110.0f / 1023.0f - 400.0f;
      if (z < Limite_Z) z = Limite_Z;

      float t1,t2,t3;
      int ret = calc_CI(x, y, z, t1, t2, t3);
      if (ret == 0) {
        a1 = servoDegFromTheta(t1, offset_1);
        a2 = servoDegFromTheta(t2, offset_2);
        a3 = servoDegFromTheta(t3, offset_3);

        moveTimeWrite(ID_Servo1, degToPos(a1), MOVE_TIME_MS);
        moveTimeWrite(ID_Servo2, degToPos(a2), MOVE_TIME_MS);
        moveTimeWrite(ID_Servo3, degToPos(a3), MOVE_TIME_MS);

        if (Serial) {
          Serial.print("CI X = "); Serial.print(x,1); Serial.print(" Y = ");
          Serial.print(y,1); Serial.print(" Z = "); Serial.print(z,1);
          Serial.print(" | S1 = "); Serial.print(a1);
          Serial.print(" S2 = "); Serial.print(a2);
          Serial.print(" S3 = "); Serial.println(a3);
        }
      }
    }

    // Modo 2: movimiento por ángulos
    else if (mode == Modo_ANG) {
      a1 = (int)(analogRead(POT_X) * 240.0f / 1023.0f);
      a2 = (int)(analogRead(POT_Y) * 240.0f / 1023.0f);
      a3 = (int)(analogRead(POT_Z) * 240.0f / 1023.0f);

      moveTimeWrite(ID_Servo1, degToPos(a1), MOVE_TIME_MS);
      moveTimeWrite(ID_Servo2, degToPos(a2), MOVE_TIME_MS);
      moveTimeWrite(ID_Servo3, degToPos(a3), MOVE_TIME_MS);

      if (Serial) {
        Serial.print("ANG S1 = "); Serial.print(a1);
        Serial.print(" S2 = "); Serial.print(a2);
        Serial.print(" S3 = "); Serial.println(a3);
      }
    }
  }

  // Garra Rotación - Abrir y Cerrar
  int potRaw = readOversampledAnalog(POT_GARRA, 8);
  FiltroPOT = ALPHA_POT * (float)potRaw + (1.0f - ALPHA_POT) * FiltroPOT;
  Angulo1_buscado = map((int)roundf(FiltroPOT), 0, 1023, ANG_MIN, ANG_MAX);
  Angulo1_buscado = clampInt(Angulo1_buscado, ANG_MIN, ANG_MAX);

  if (Angulo1_centro < Angulo1_buscado) Angulo1_centro =
  min(Angulo1_centro + STEP_DEG, Angulo1_buscado);
  else if (Angulo1_centro > Angulo1_buscado) Angulo1_centro =
  max(Angulo1_centro - STEP_DEG, Angulo1_buscado);
  writeServos(pwm1, angleToUs(Angulo1_centro));

  // Abrir y cerrar
  bool Abrir = digitalRead(BTN_ABRIR);
  bool Cerrar = digitalRead(BTN_CERRAR);
  unsigned long t = millis();

  if (Ult_Abierto == HIGH && Abrir == LOW && (t - t_ultAbierto) > DEBOUNCE_MS) {
    Angulo2_cerrado = ANG_ABIERTO;
    writeServos(pwm2, angleToUs(Angulo2_cerrado));
    t_ultAbierto = t;
  }
  Ult_Abierto = Abrir;

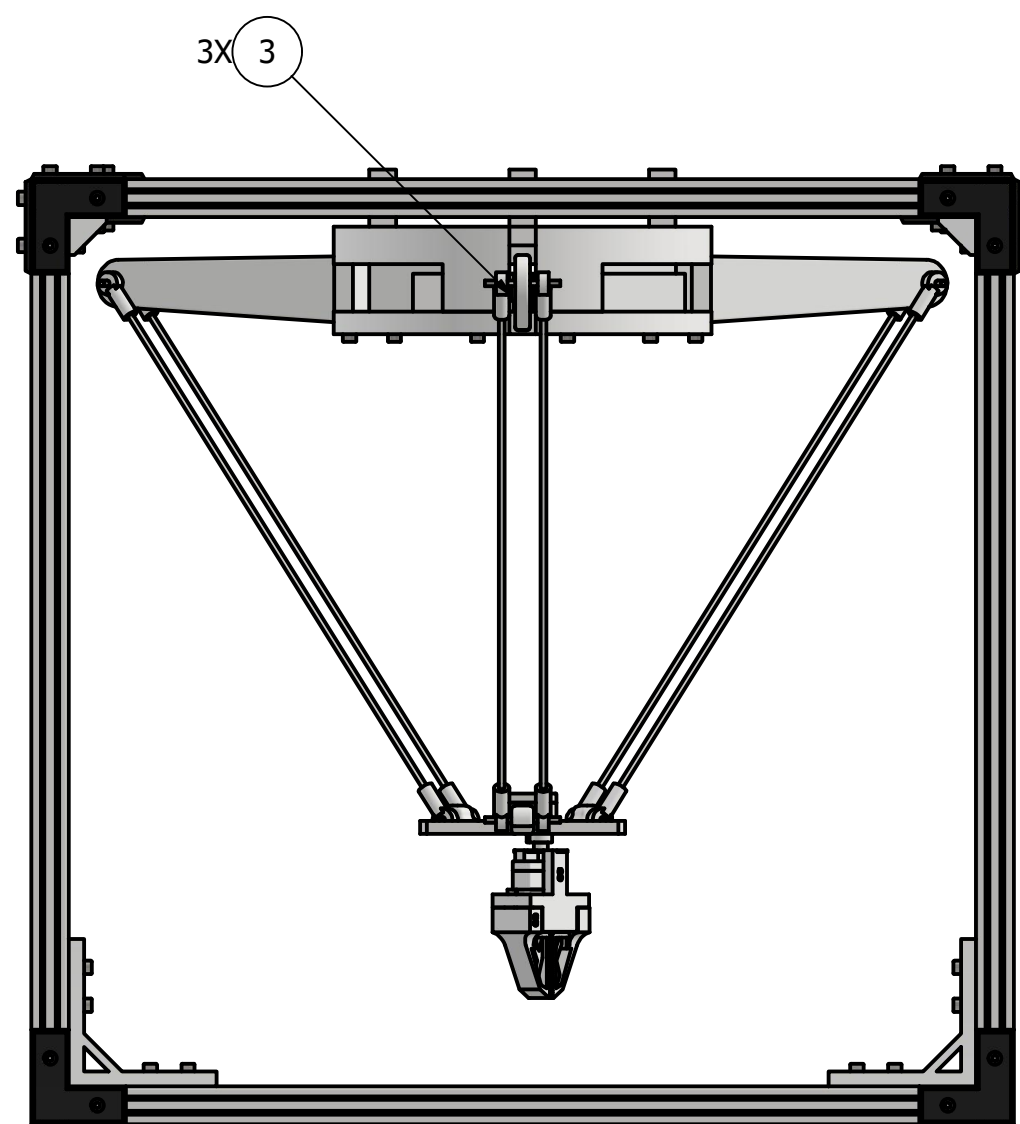
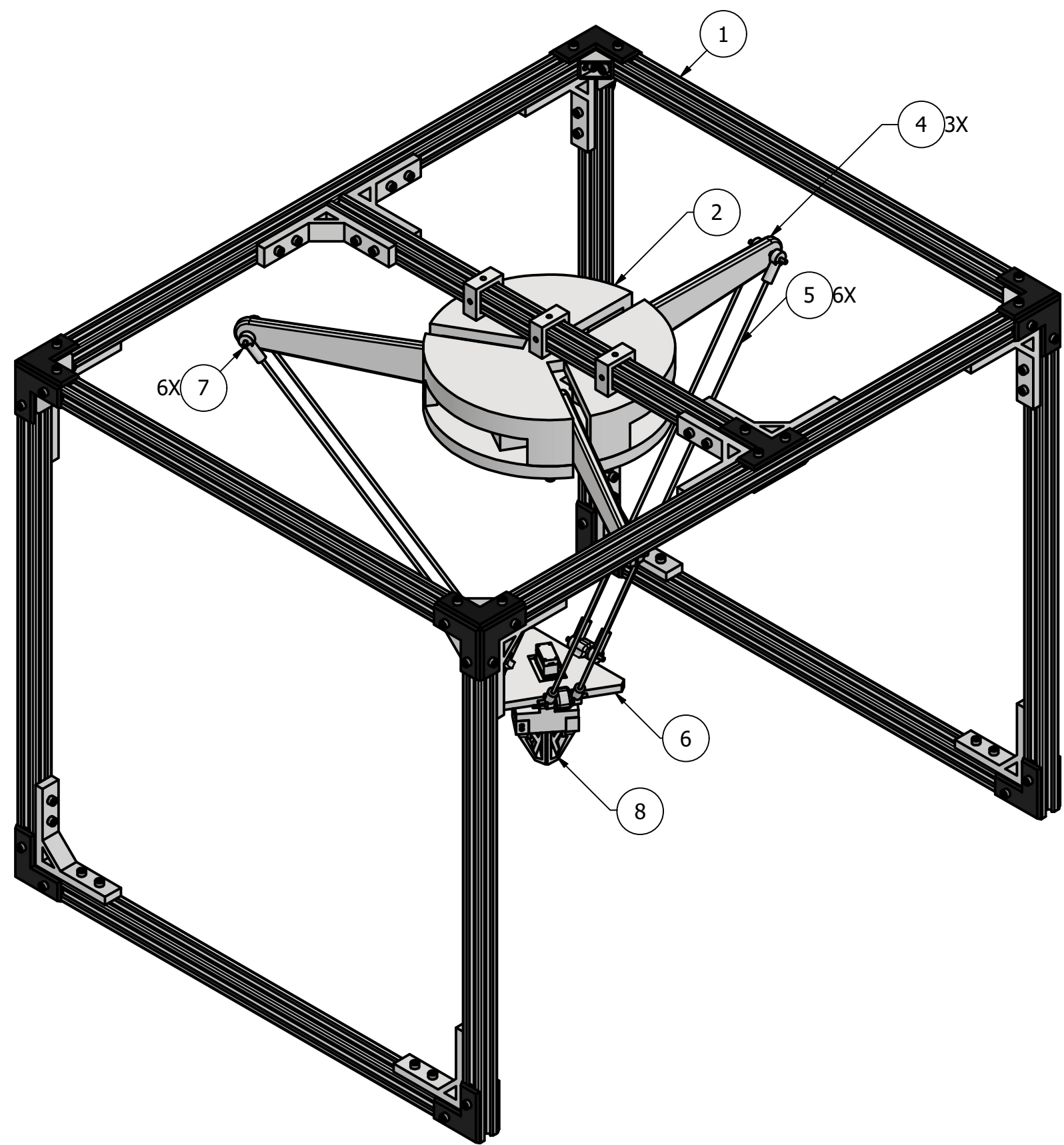
  if (Ult_Cerrado == HIGH && Cerrar == LOW && (t - t_ultCerrado) > DEBOUNCE_MS) {
    Angulo2_cerrado = ANG_CERRADO;
    writeServos(pwm2, angleToUs(Angulo2_cerrado));
    t_ultCerrado = t;
  }
  Ult_Cerrado = Cerrar;

  delay(300);
}

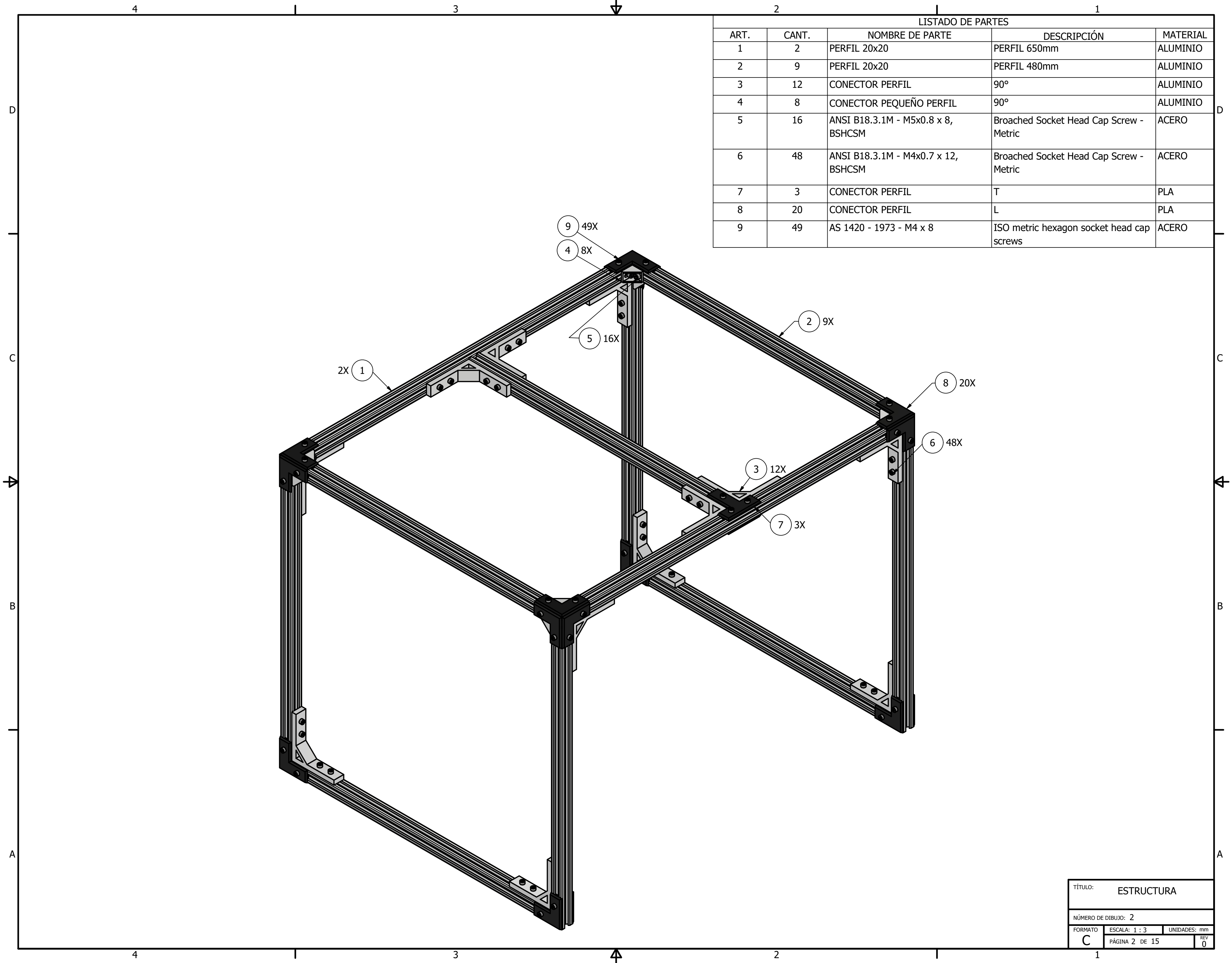
```

11.3. Anexo C: Planos técnicos

LISTADO DE PARTES				
ART.	CANT.	NOMBRE DE PARTE	DESCRIPCIÓN	MATERIAL
1	1	ESTRUCTURA	PERFIL 20X20	ALUMINIO
2	1	BASE ESTÁTICA		PLA
3	3	ACOPLER SERVOMOTOR		PLA
4	3	ESLABÓN 1		PLA
5	6	ESLABÓN 2	VARILLA 3mm	ACERO INOXIDABLE
6	1	BASE MÓVIL		PLA
7	6	PIN ESLABÓN 2	VARILLA 3mm	ACERO INOXIDABLE
8	1	GARRA		PLA

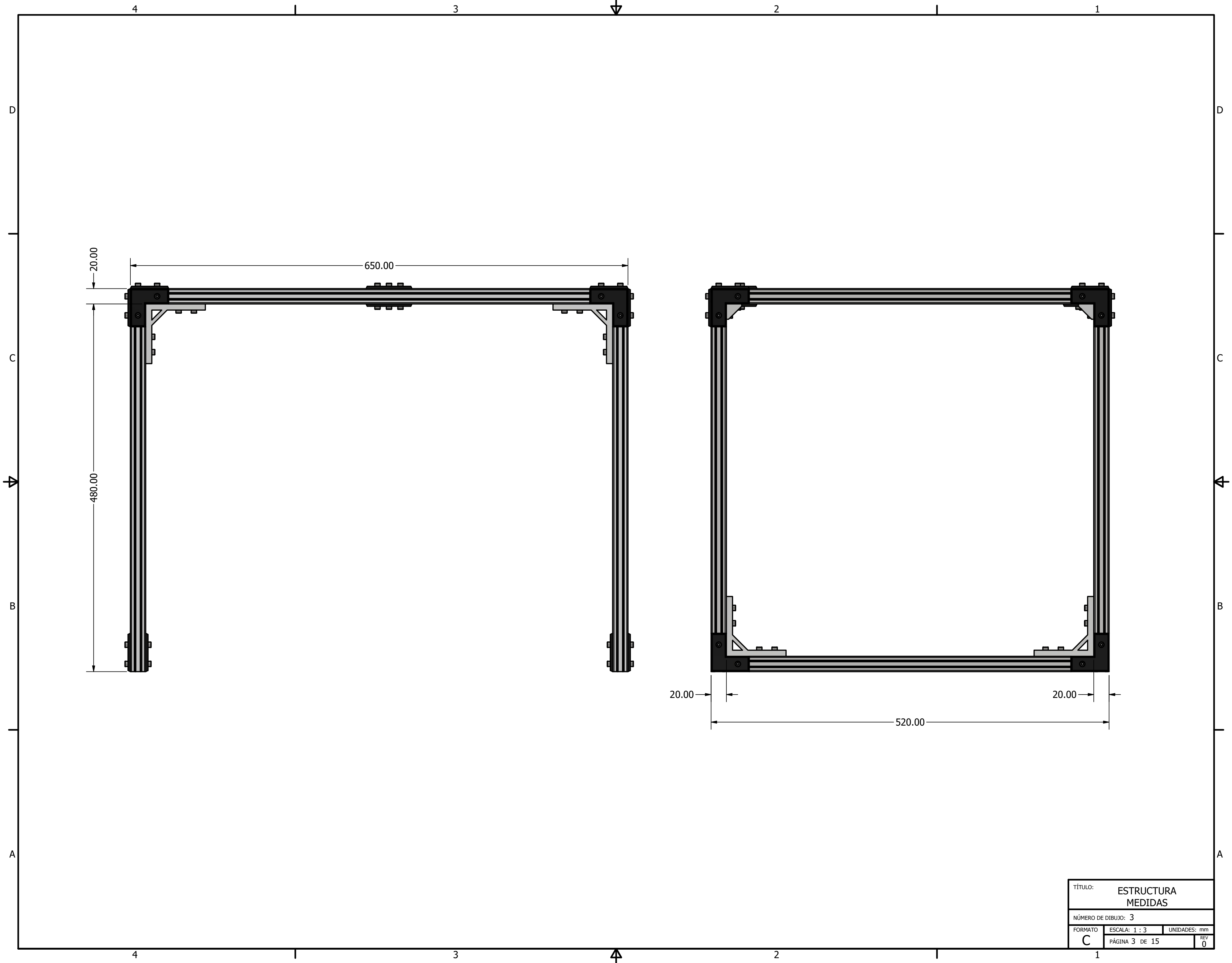


<p>TODAS LAS MEDIDAS EN MILÍMETROS SALVO SE INDIQUE LO CONTRARIO. INTERPRETAR SEGÚN ASME Y14.5 - 2009.</p> <p>TOLERANCIAS GENERALES</p> <p>LINEAL .X ± 0.1 .XX ± 0.05 .XXX ± 0.010</p> <p>ANGULAR ± 1°</p> <p>FRACCIONES ± "</p> <p>RUGOSIDAD SUPERFICIAL ✓</p>	<p>DIBUJADO POR MARIANA QUIRÓS</p> <p>FECHA 2025</p>	<p>UNIVERSIDAD DEL VALLE DE GUATEMALA</p> <p>18 avenida, 11-95 zona 15, Vista Hermosa III Guatemala, Guatemala 01015 PBX: (502) 2634-0336 / 40 info@uvv.edu.gt</p>	
	<p>DISEÑADO POR MARIANA QUIRÓS</p> <p>FECHA 2025</p>		<p>REVISADO POR DULCE CHACÓN</p> <p>FECHA 2025</p>
	<p>APROBADO POR DULCE CHACÓN</p> <p>FECHA 2025</p>		<p>TÍTULO: BRAZO DELTA</p>
	<p>NOMBRE / NÚMERO DE PROYECTO 1</p>		<p>NÚMERO DE DIBUJO: 1</p>
<p>MATERIAL VER LP</p> <p>TRATAMIENTO VER LP</p> <p>MASA: N/A</p>	<p>TERCER ÁNGULO DE PROYECCIÓN</p>	<p>TODA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO ES PROPIEDAD DE LA UNIVERSIDAD DEL VALLE DE GUATEMALA. SU REPRODUCCIÓN TOTAL O PARCIAL QUEDA PROHIBIDA SALVO PREVIA AUTORIZACIÓN DE LA INSTITUCIÓN.</p>	<p>FORMATO C</p> <p>ESCALA: 1 : 4</p> <p>UNIDADES: mm</p> <p>PÁGINA 1 DE 15</p> <p>REV 0</p>

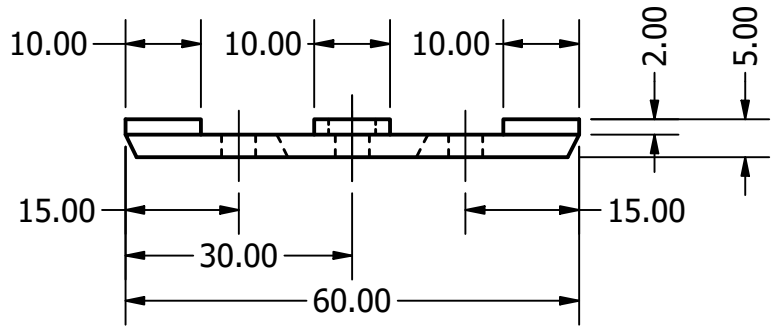
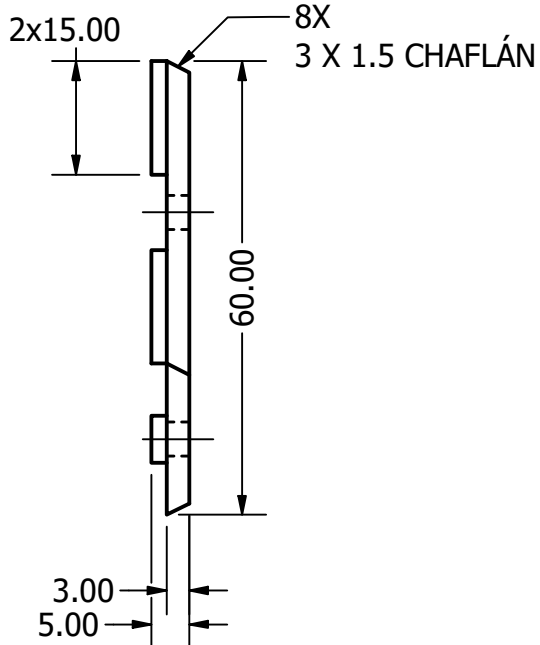
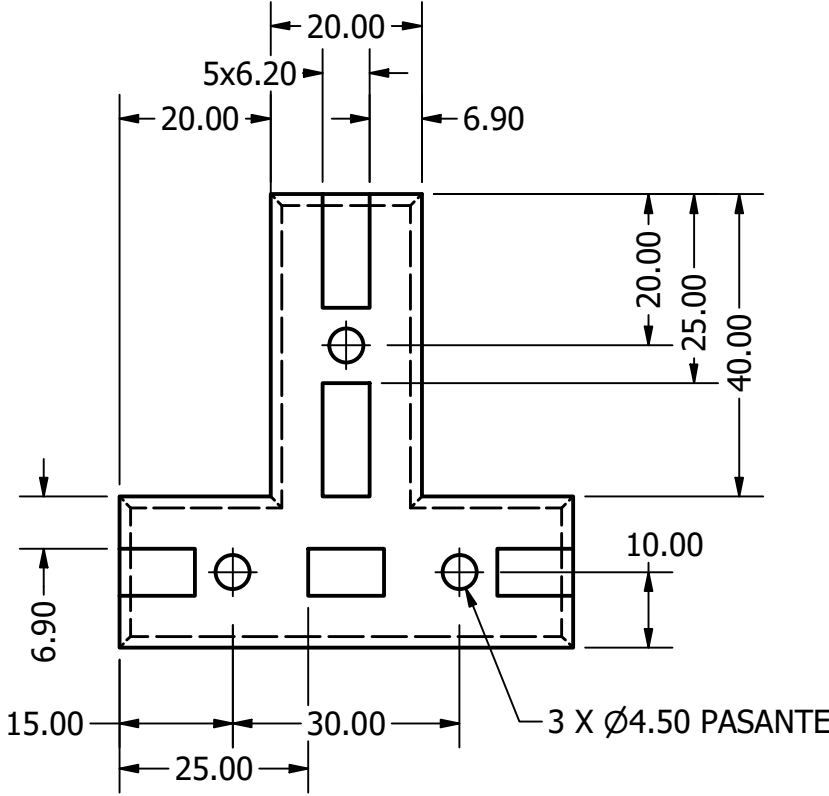
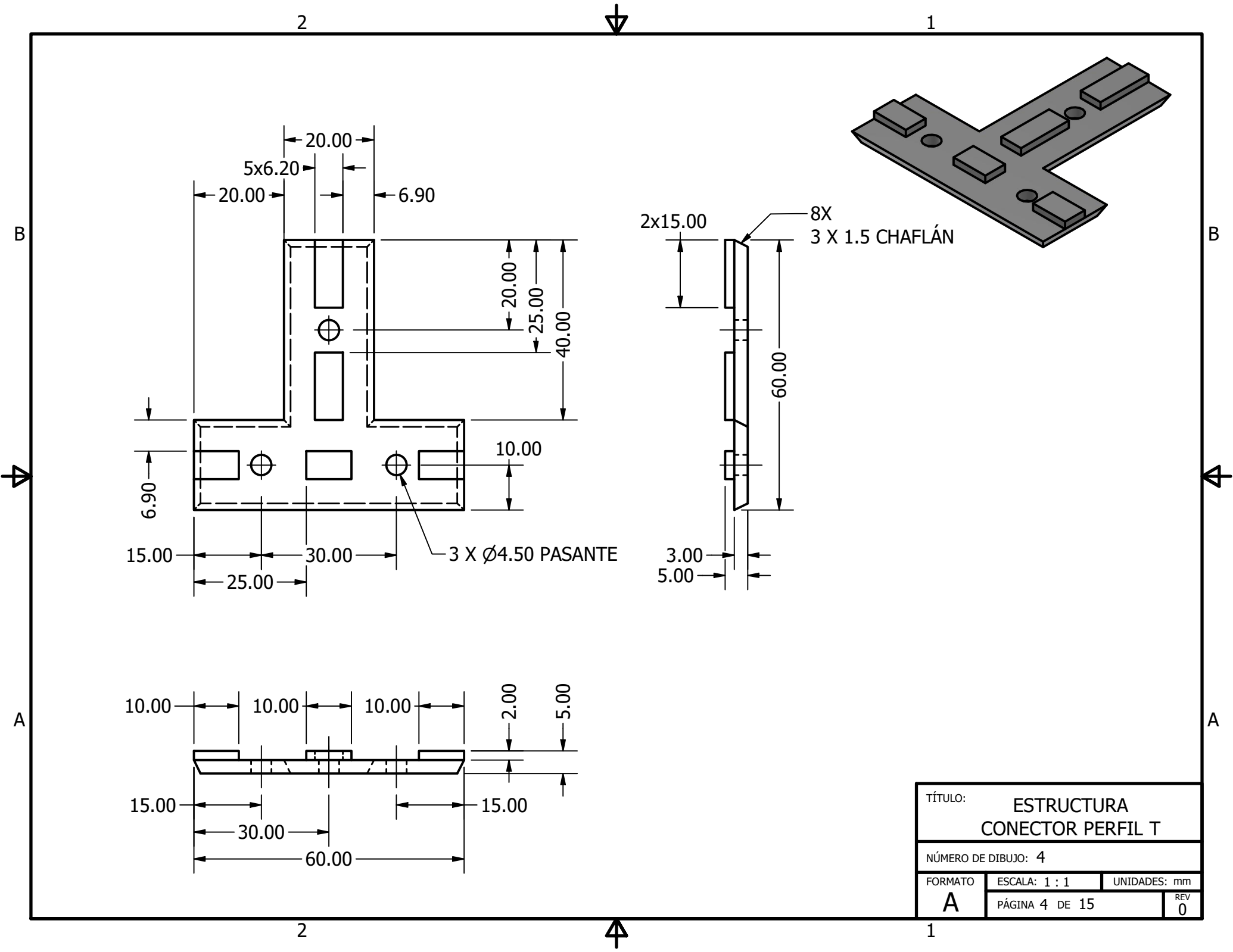


LISTADO DE PARTES				
ART.	CANT.	NOMBRE DE PARTE	DESCRIPCIÓN	MATERIAL
1	2	PERFIL 20x20	PERFIL 650mm	ALUMINIO
2	9	PERFIL 20x20	PERFIL 480mm	ALUMINIO
3	12	CONECTOR PERFIL	90°	ALUMINIO
4	8	CONECTOR PEQUEÑO PERFIL	90°	ALUMINIO
5	16	ANSI B18.3.1M - M5x0.8 x 8, BSHCSM	Broached Socket Head Cap Screw - Metric	ACERO
6	48	ANSI B18.3.1M - M4x0.7 x 12, BSHCSM	Broached Socket Head Cap Screw - Metric	ACERO
7	3	CONECTOR PERFIL	T	PLA
8	20	CONECTOR PERFIL	L	PLA
9	49	AS 1420 - 1973 - M4 x 8	ISO metric hexagon socket head cap screws	ACERO

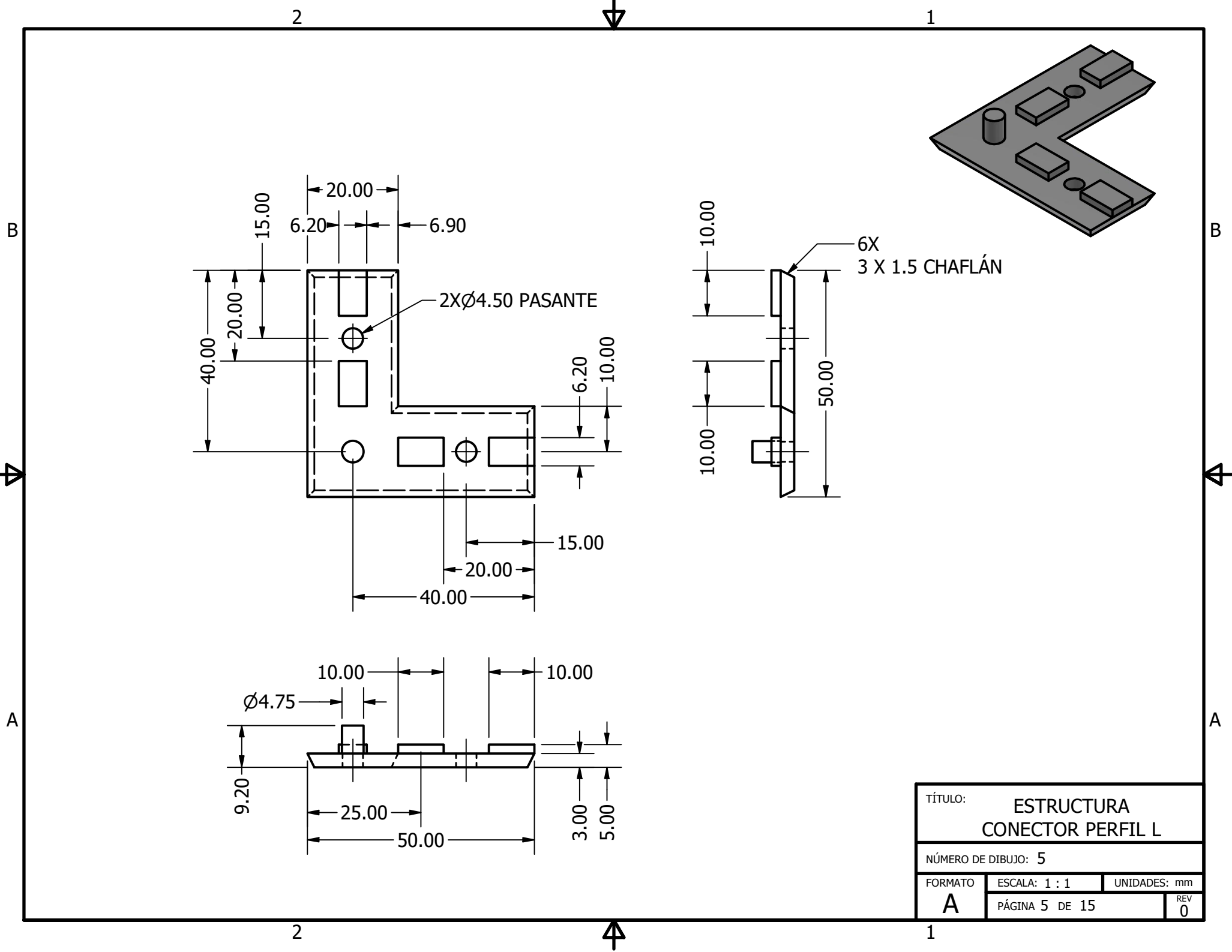
TÍTULO: ESTRUCTURA		
NÚMERO DE DIBUJO: 2		
FORMATO: C	ESCALA: 1 : 3	UNIDADES: mm
PÁGINA 2 DE 15		REV 0



TÍTULO:		ESTRUCTURA MEDIDAS	
NÚMERO DE DIBUJO: 3			
FORMATO	ESCALA: 1 : 3	UNIDADES: mm	
C	PÁGINA 3 DE 15	REV	0

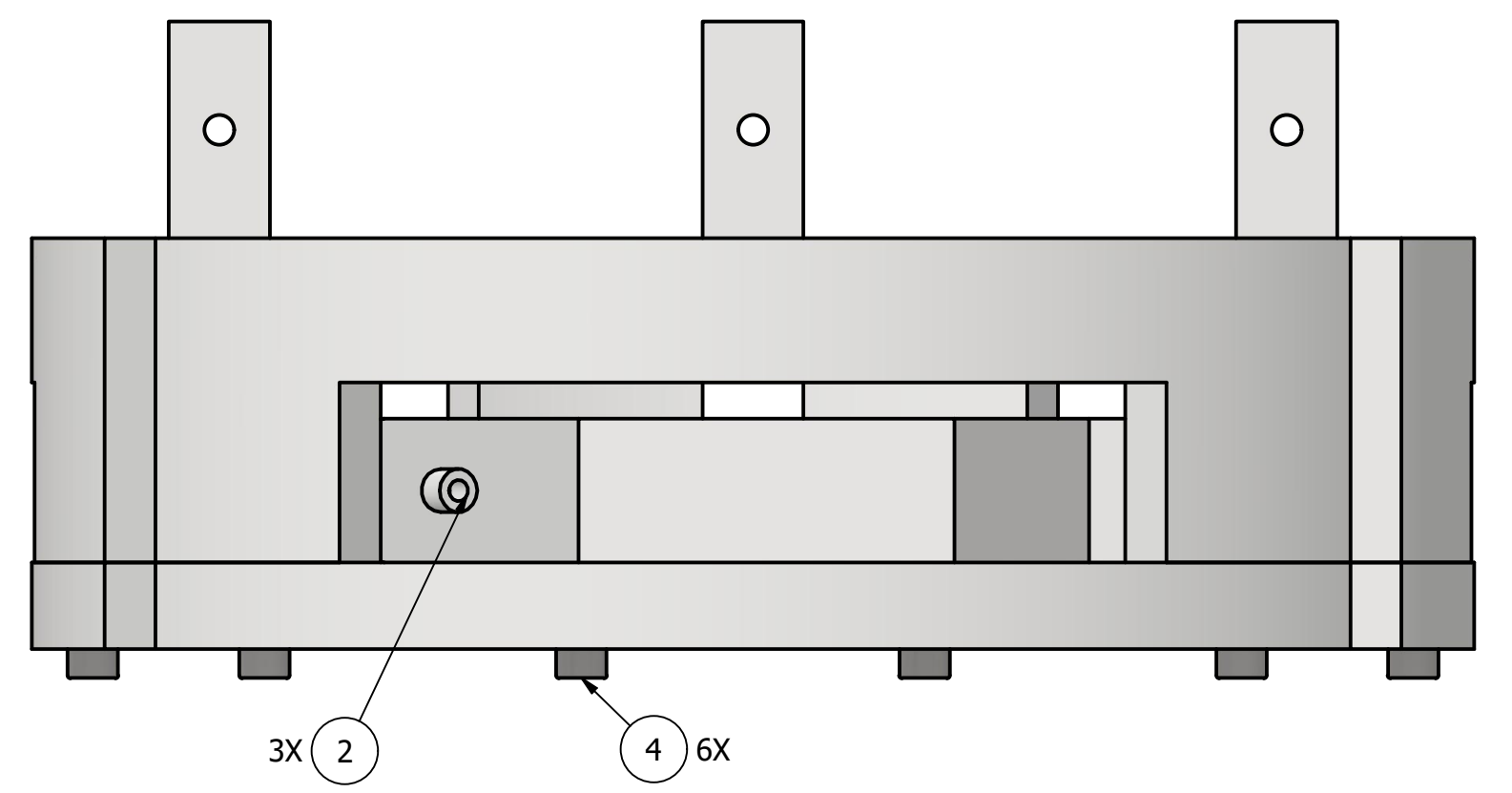
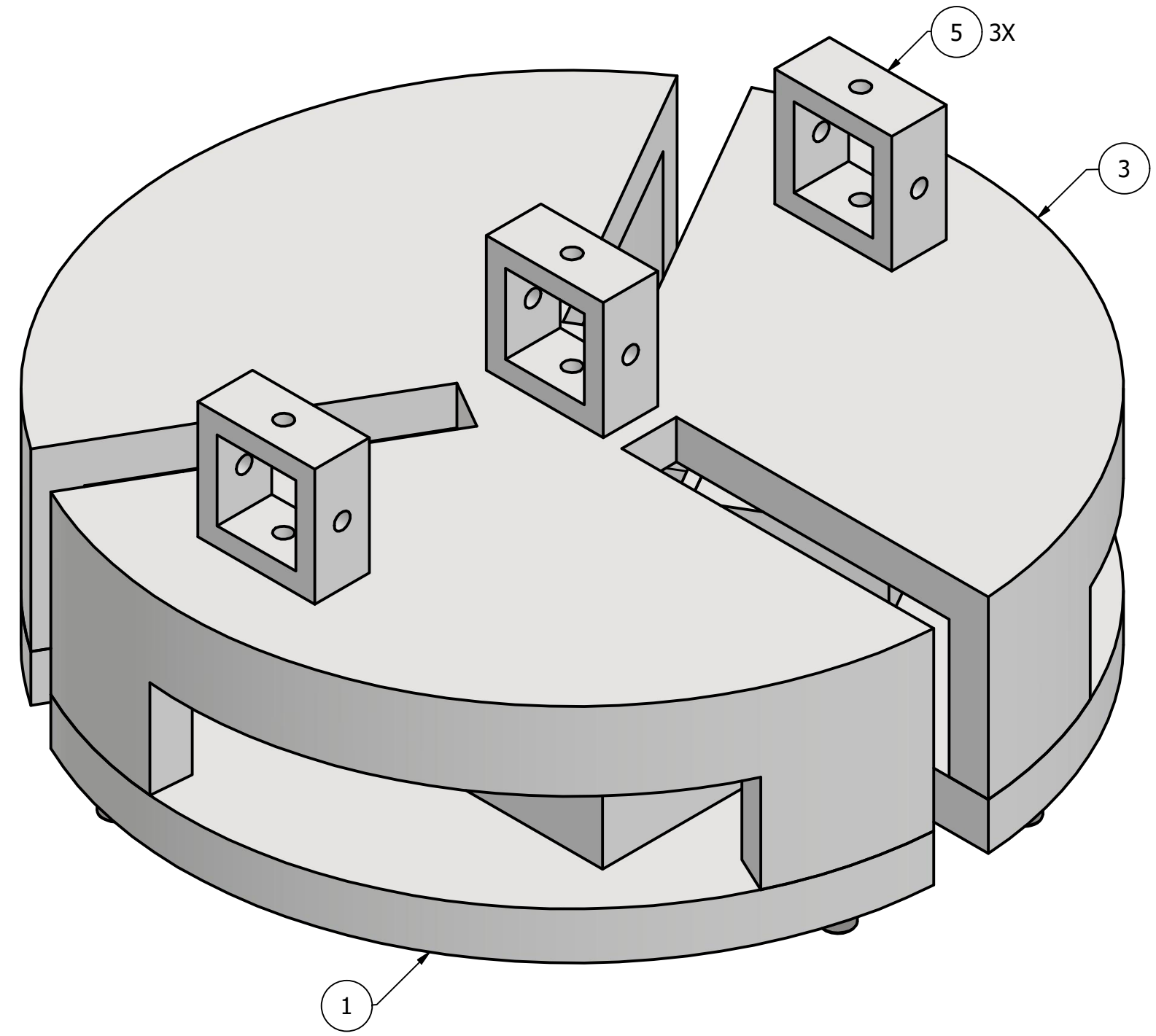


TÍTULO:			ESTRUCTURA CONECTOR PERFIL T		
NÚMERO DE DIBUJO: 4					
FORMATO	ESCALA: 1 : 1	UNIDADES: mm			
A	PÁGINA 4 DE 15		REV		0

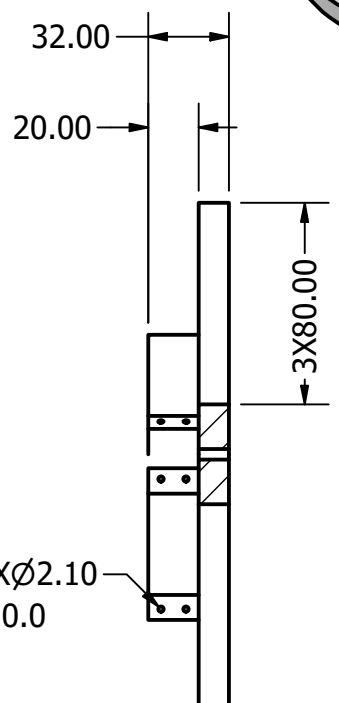
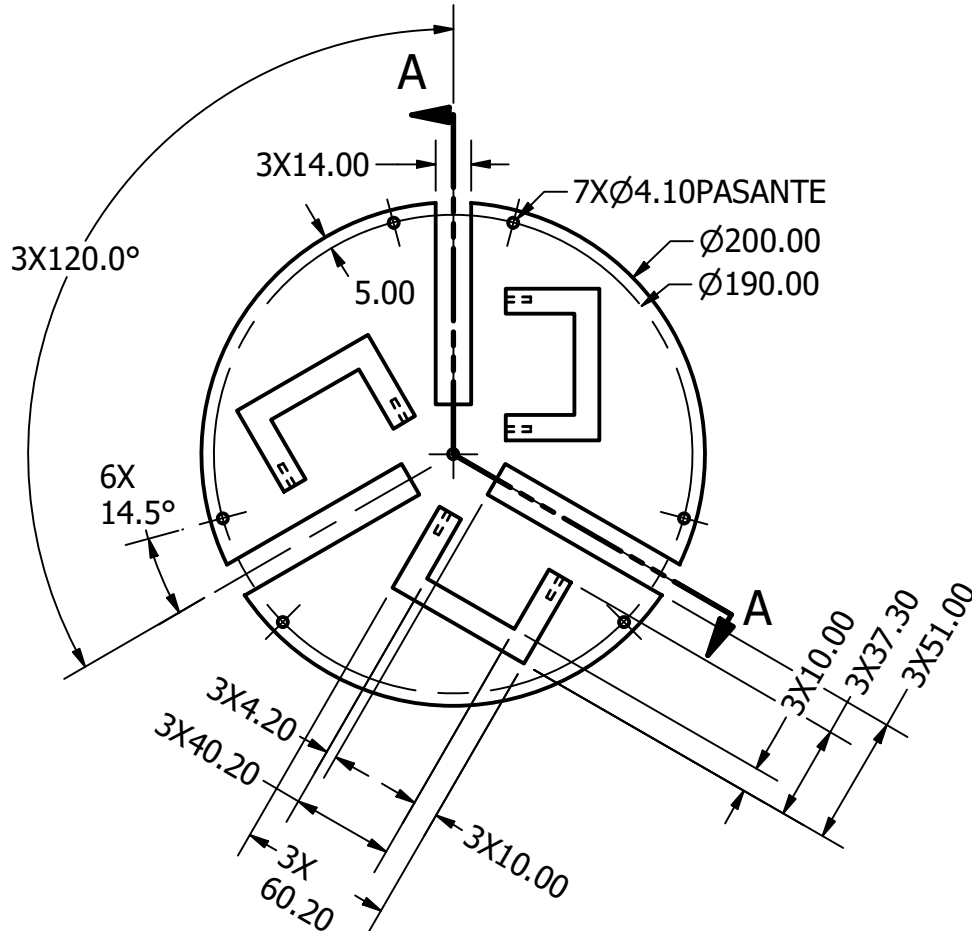
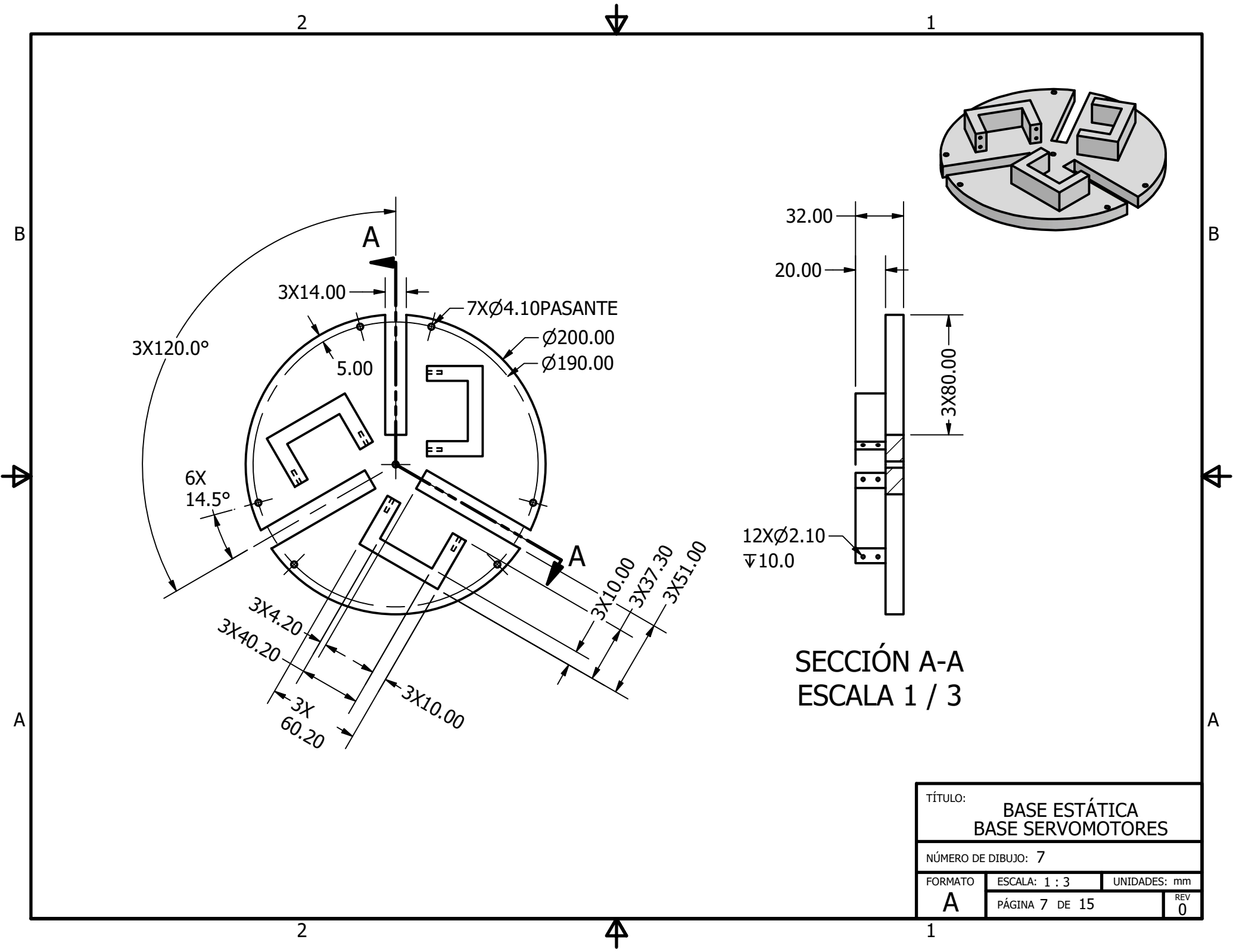


TÍTULO:			ESTRUCTURA CONECTOR PERFIL L		
NÚMERO DE DIBUJO: 5					
FORMATO	ESCALA: 1 : 1	UNIDADES: mm			
A	PÁGINA 5 DE 15			REV	0

LISTADO DE PARTES				
ART.	CANT.	NOMBRE DE PARTE	DESCRIPCIÓN	MATERIAL
1	1	BASE SERVOMOTORES		PLA
2	3	SERVOMOTOR HiWONDER	35 kg	-
3	1	TAPADERA BASE		PLA
4	6	AS 1420 - 1973 - M4 x 30	ISO metric hexagon socket head cap screws	ACERO
5	3	ANCLAJE ESTRUCTURA		PLA

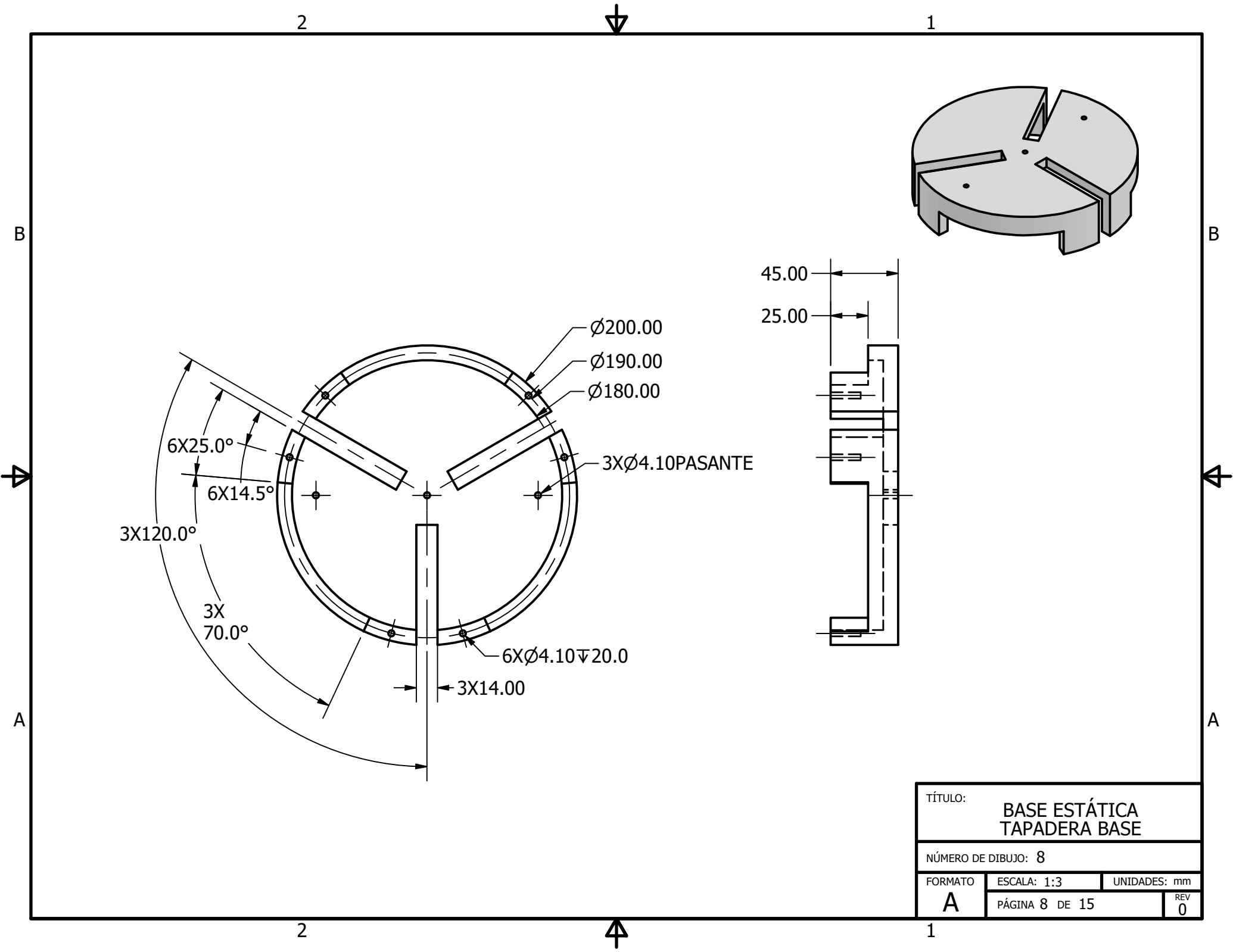


TÍTULO: BASE ESTÁTICA		
NÚMERO DE DIBUJO: 6		
FORMATO: C	ESCALA: 1 : 1	UNIDADES: mm
PÁGINA 6 DE 15		REV 0

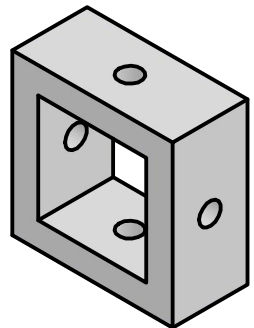
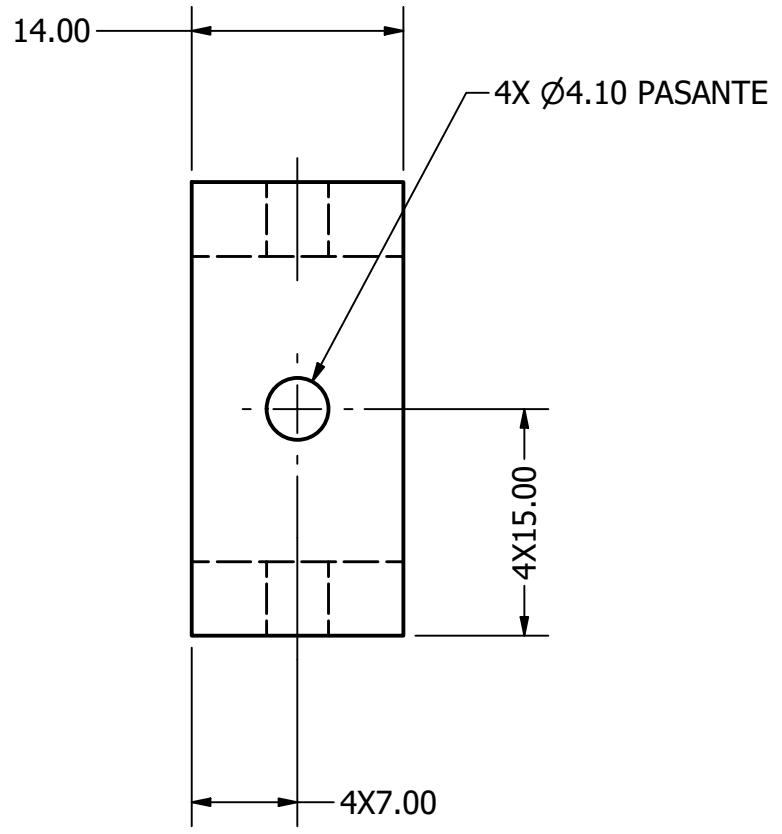
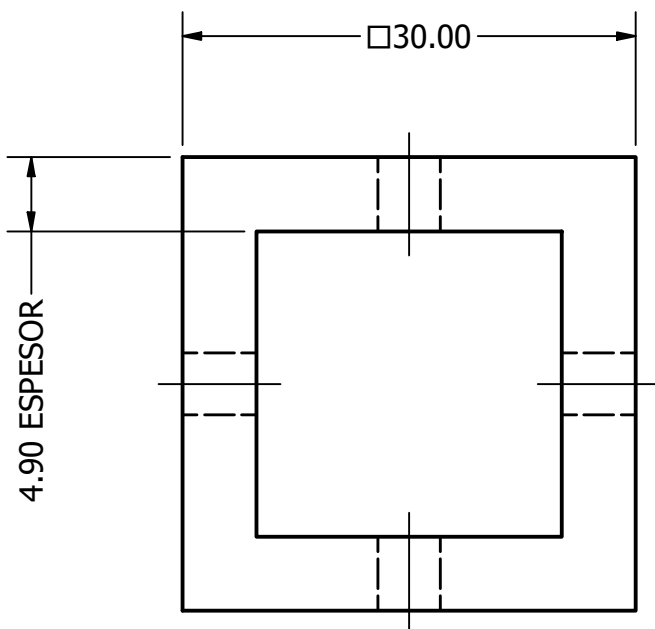
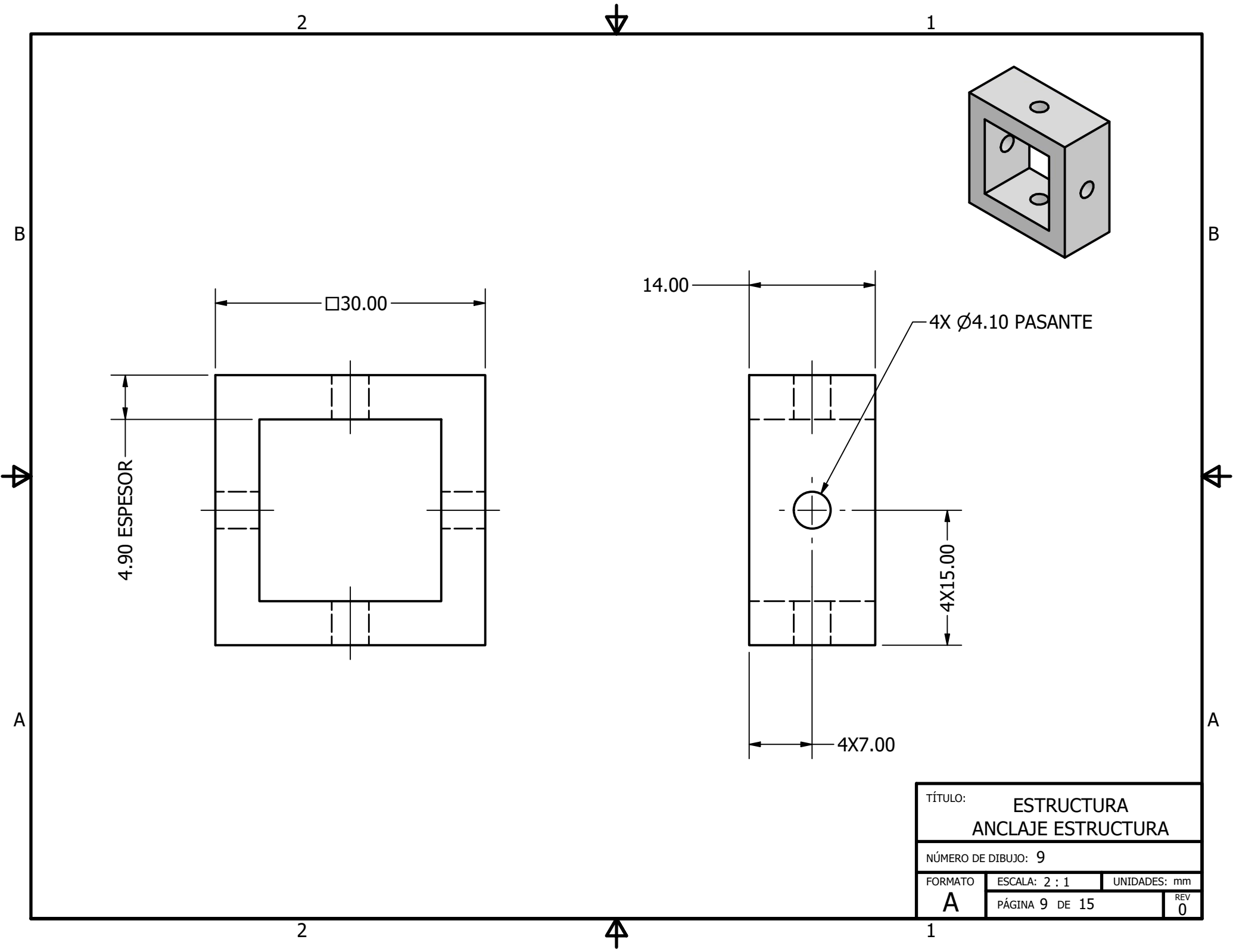


SECCIÓN A-A
ESCALA 1 / 3

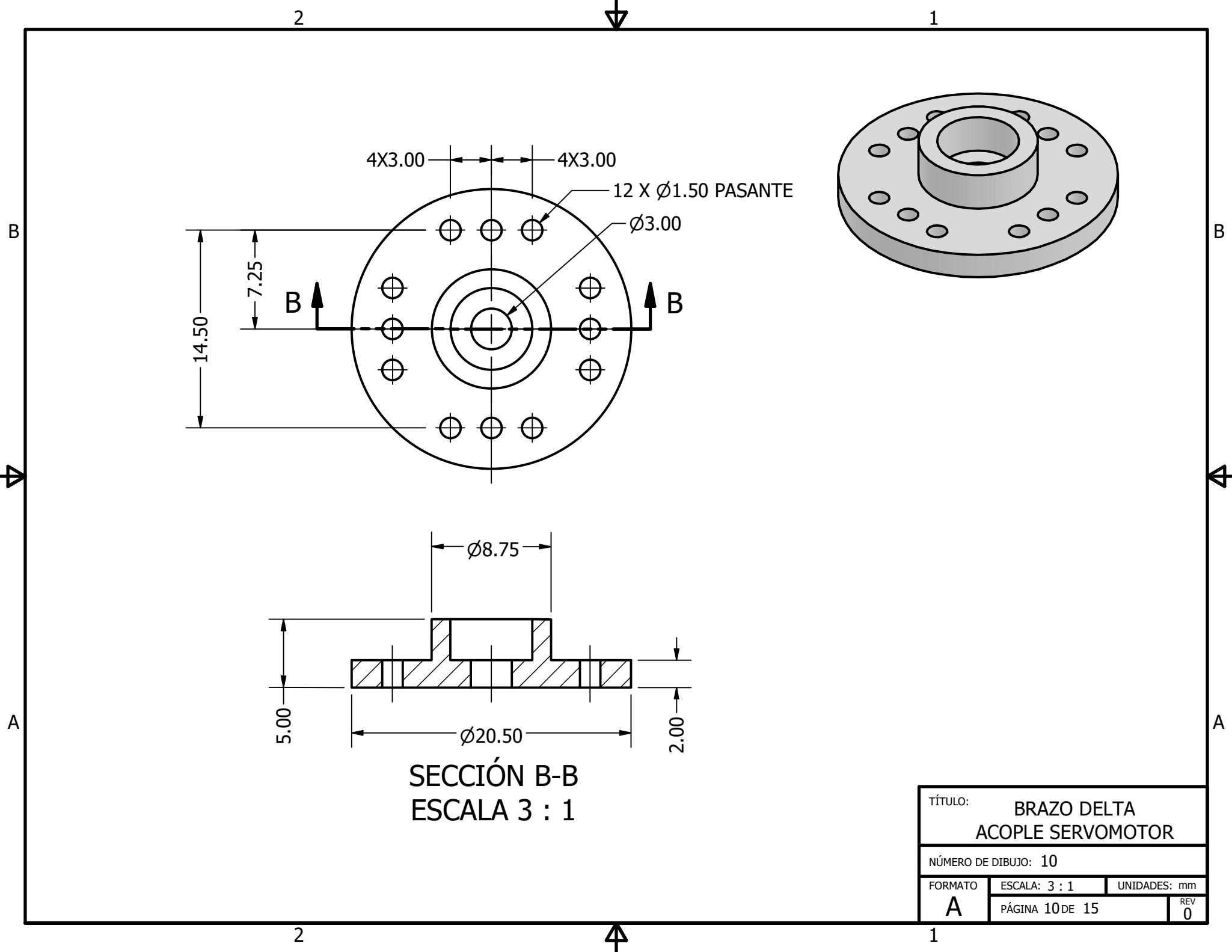
TÍTULO:		
BASE ESTÁTICA BASE SERVOMOTORES		
NÚMERO DE DIBUJO: 7		
FORMATO	ESCALA: 1 : 3	UNIDADES: mm
A	PÁGINA 7 DE 15	REV 0



TÍTULO:		
BASE ESTÁTICA TAPADERA BASE		
NÚMERO DE DIBUJO: 8		
FORMATO	ESCALA: 1:3	UNIDADES: mm
A	PÁGINA 8 DE 15	REV 0



TÍTULO:		
ESTRUCTURA ANCLAJE ESTRUCTURA		
NÚMERO DE DIBUJO: 9		
FORMATO	ESCALA: 2 : 1	UNIDADES: mm
A	PÁGINA 9 DE 15	REV 0



2

1

4X3.00

4X3.00

12 X Ø1.50 PASANTE

Ø3.00

14.50

7.25

B

B

Ø8.75

5.00

Ø20.50

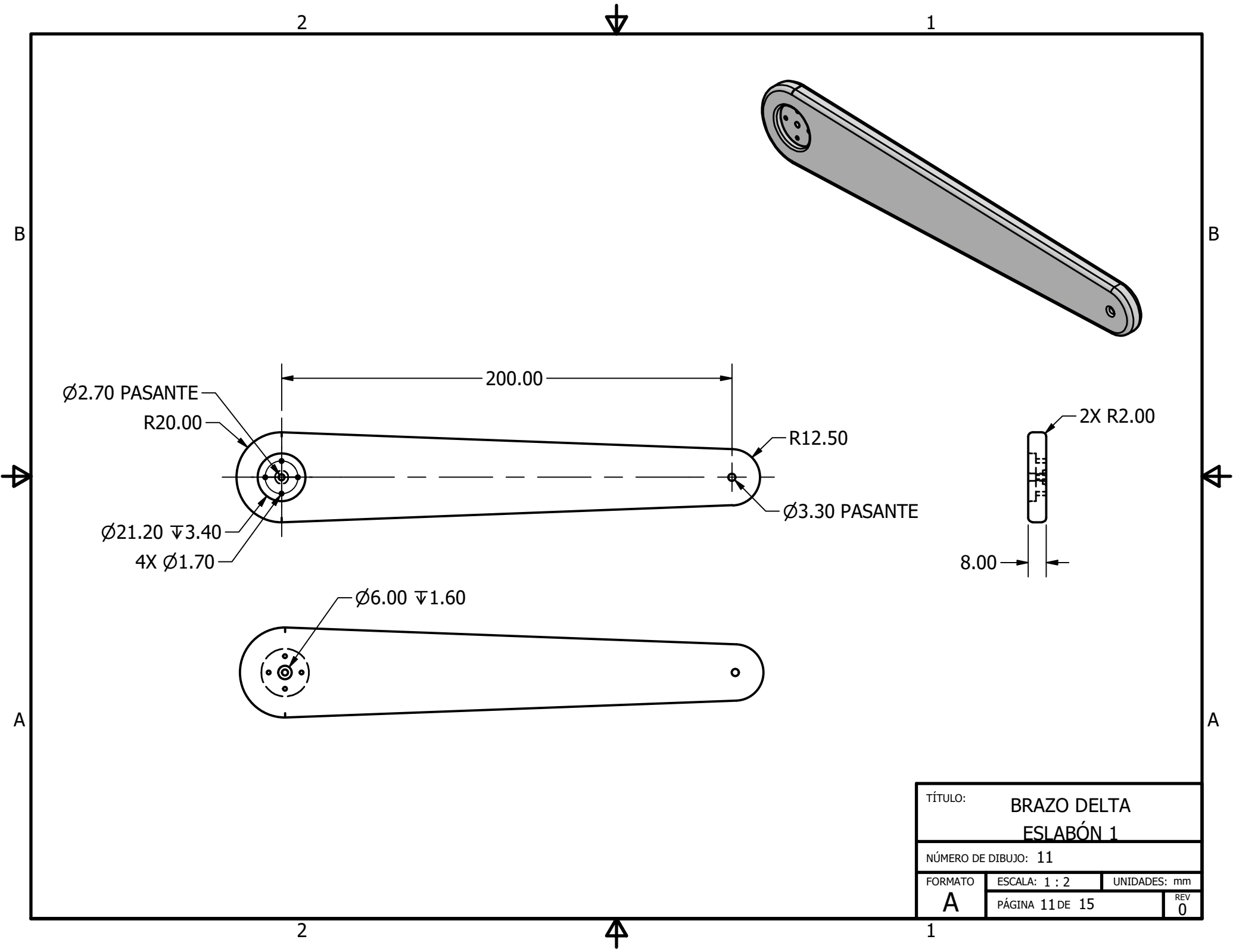
2.00

SECCIÓN B-B
ESCALA 3 : 1

TÍTULO:			BRAZO DELTA ACOPLE SERVOMOTOR		
NÚMERO DE DIBUJO: 10					
FORMATO	ESCALA: 3 : 1	UNIDADES: mm			
A	PÁGINA 10 DE 15		REV	0	

2

1



Ø2.70 PASANTE

R20.00

Ø21.20 ▽3.40

4X Ø1.70

200.00

R12.50

Ø3.30 PASANTE

Ø6.00 ▽1.60

2X R2.00

8.00

TÍTULO:			BRAZO DELTA		
			ESLABÓN 1		
NÚMERO DE DIBUJO: 11					
FORMATO	ESCALA: 1 : 2	UNIDADES: mm			
A	PÁGINA 11 DE 15		REV	0	

2



1

LISTADO DE PARTES

ART.	CANT.	NOMBRE DE PARTE	DESCRIPCIÓN	MATERIAL
1	1	Barra	Barra roscada en los extremos	Acero Inoxidable
2	2	Rodamiento	Rodamiento ojo de pescado 3mm	-

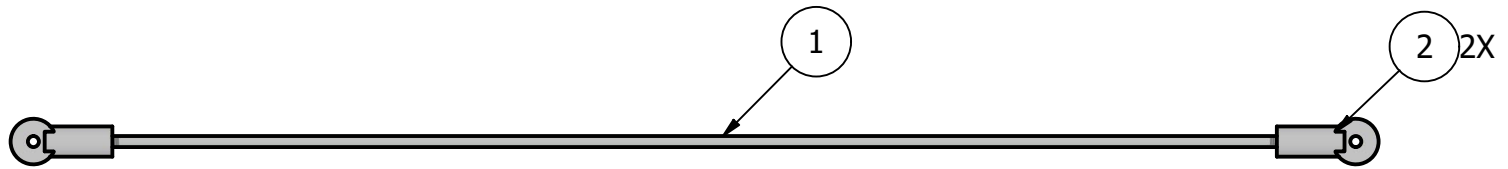
B

B



A

A

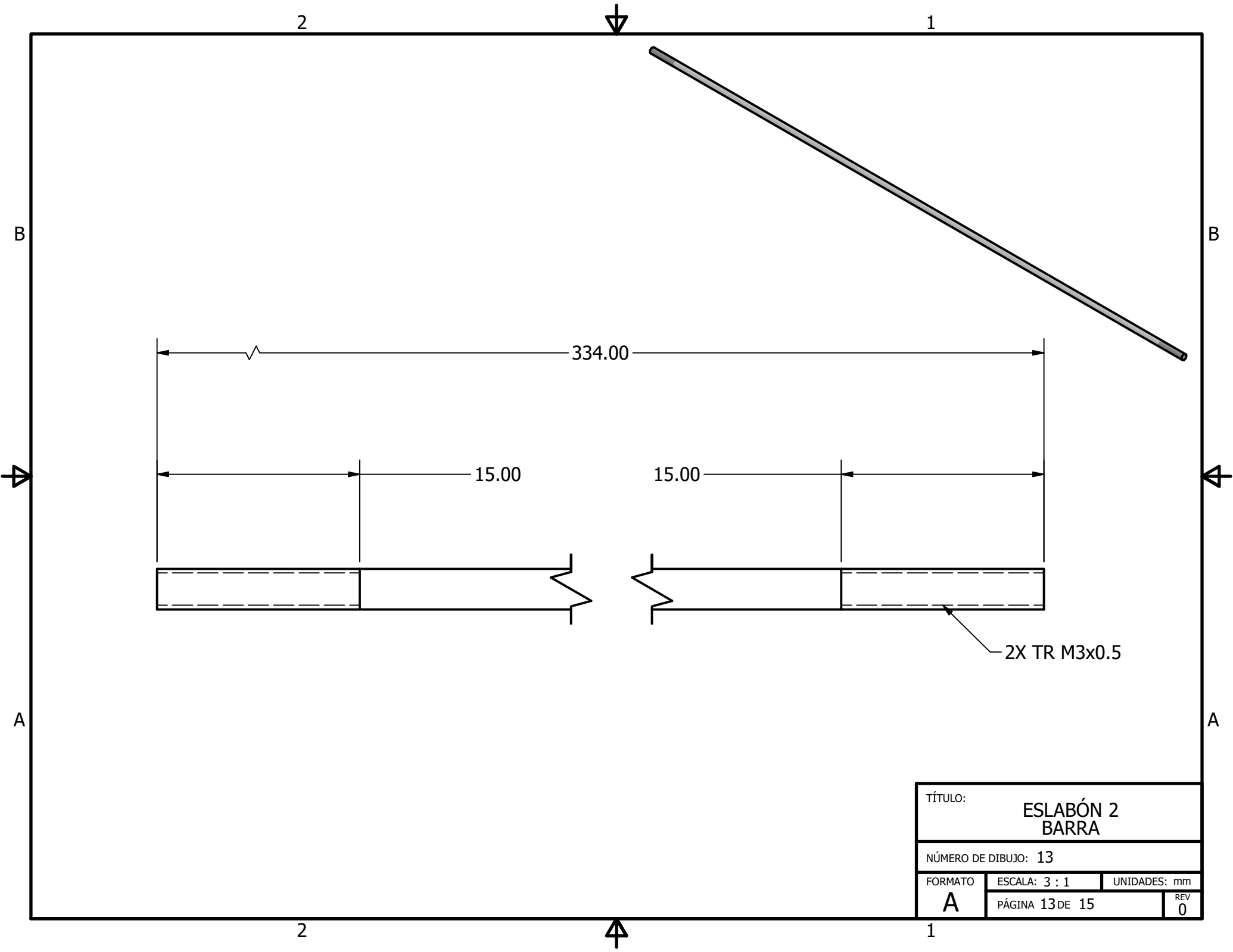


TÍTULO: BRAZO DELTA ESLABÓN 2			
NÚMERO DE DIBUJO: 12			
FORMATO	ESCALA: 1 : 1	UNIDADES: mm	
A	PÁGINA 12 DE 15	REV	0

2



1



2

1

B

B

A

A

334.00

15.00

15.00

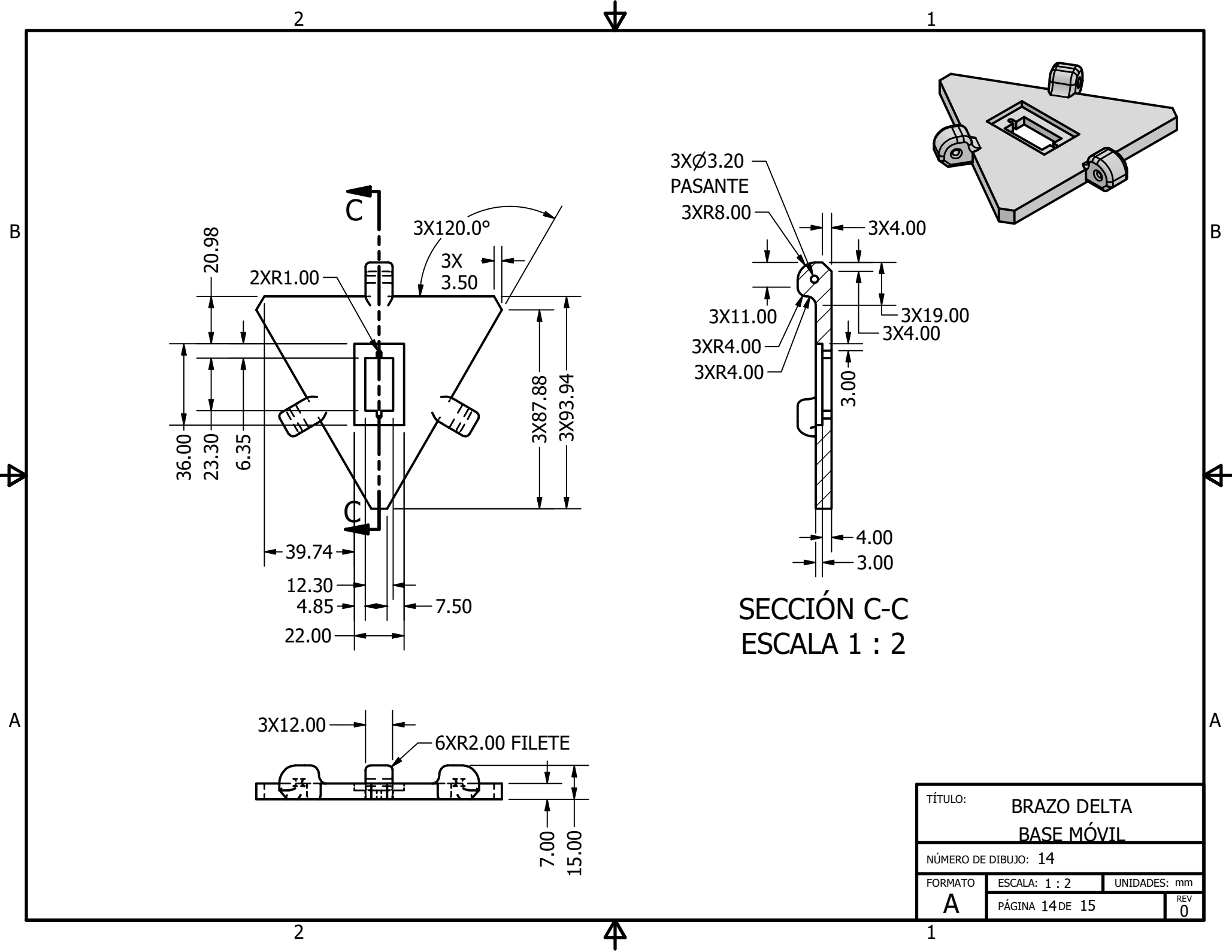
2X TR M3x0.5

TÍTULO:			
ESLABÓN 2 BARRA			
NÚMERO DE DIBUJO: 13			
FORMATO	ESCALA: 3 : 1	UNIDADES: mm	
A	PÁGINA 13 DE 15		REV 0

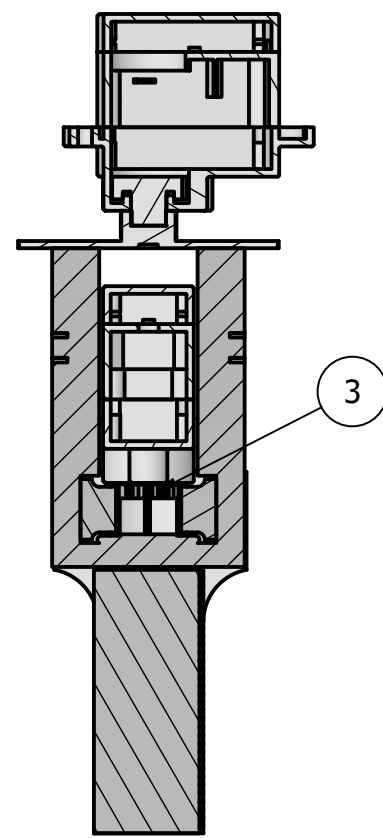
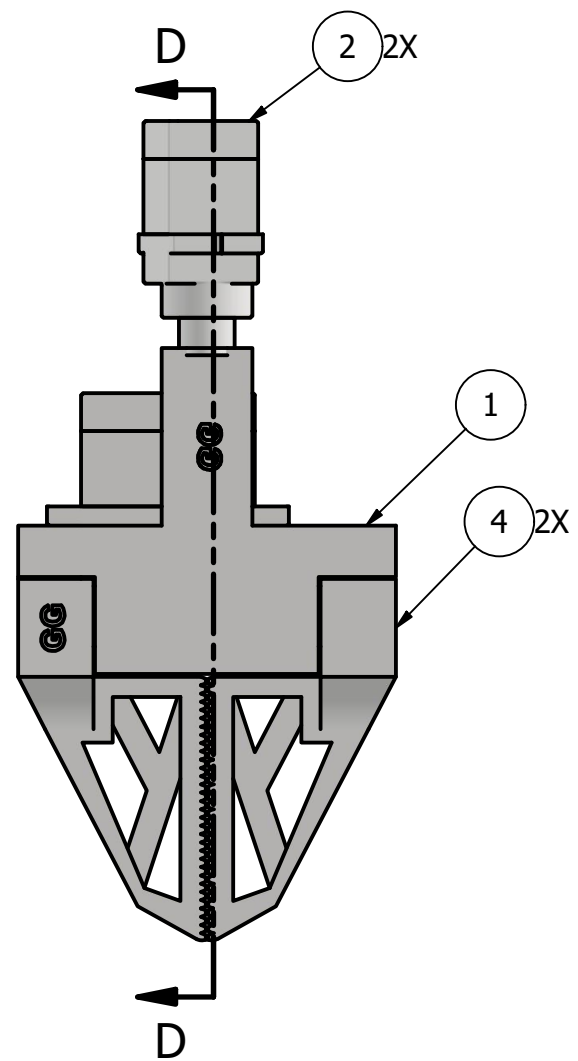
2

1

↑



LISTADO DE PARTES				
ART.	CANT.	NOMBRE DE PARTE	DESCRIPCIÓN	MATERIAL
1	1	Base Servomotor		PLA
2	2	Servomotor	MG90S	-
3	1	Engranaje Pinza		PLA
4	2	Gripper Pinza		PLA



SECCIÓN D-D
ESCALA 1 : 1

TÍTULO: BRAZO DELTA GARRA		
NÚMERO DE DIBUJO: 15		
FORMATO B	ESCALA: 1 : 1	UNIDADES: mm
	PÁGINA 15 DE 15	REV 0