



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

“DISPOSITIVO MECATRÓNICO PARA EL APRENDIZAJE DE LAS OPERACIONES BÁSICAS DE MATEMÁTICAS EN BRAILLE”

Trabajo de grado previo a la obtención de título de Ingeniero en Mecatrónica

Línea de investigación: Producción industrial y tecnología sostenible.

Autor:

Bryan Alejandro Chiguano Moreta

Director:

MSc. Fernando Vinicio Valencia Aguirre

Asesor:

PhD. David Alberto Ojeda Peña

Ibarra, 04 diciembre 2024



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003738018		
APELLIDOS Y NOMBRES:	Chiguano Moreta Bryan Alejandro		
DIRECCIÓN:	La Esperanza, Ibarra		
EMAIL:	bachiguanom@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	+593986141374

DATOS DE LA OBRA	
TÍTULO:	“DISPOSITIVO MECATRÓNICO PARA EL APRENDIZAJE DE LAS OPERACIONES BÁSICAS DE MATEMÁTICAS EN BRAILLE”
AUTOR:	Bryan Alejandro Chiguano Moreta
FECHA:	04 diciembre de 2024
SOLO PARA TRABAJOS DE TITULACIÓN	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
DIRECTOR:	<i>MSc. Valencia Aguirre Fernando Vinicio</i>

2. CONSTANCIAS

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 04 días del mes de diciembre de 2024

EL AUTOR:

(Firma).....

Nombre: Chiguano Moreta Bryan Alejandro



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR

Ibarra, 04 de diciembre de 2024

MSc. Fernando Vinicio Valencia Aguirre

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del Trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

MSc. Fernando Vinicio Valencia Aguirre

C.C.: 1003188669



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “DISPOSITIVO MECATRÓNICO PARA EL APRENDIZAJE DE LAS OPERACIONES BÁSICAS DE MATEMÁTICAS EN BRAILLE” elaborado por Chiguano Moreta Bryan Alejandro, previo a la obtención del título de INGENIERO EN MECATRÓNICA, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:

MSc. Fernando Vinicio Valencia Aguirre

C.C.: 1003188669

PhD. David Alberto Ojeda Peña

C.C.: 1757898489

DEDICATORIA

A mis padres, Verónica Moreta y Jaime Chiguano, por su constante esfuerzo, sacrificio y dedicación, que han sido el pilar fundamental para alcanzar esta meta tan significativa en mi vida. Su apoyo incondicional y las valiosas experiencias que me han transmitido han sido mi guía a lo largo de mi carrera universitaria.

A mi hermana, Melany Chiguano, quien con su cariño y aliento diario ha sido una inspiración constante para superar los desafíos y seguir adelante. Espero ser un ejemplo en el que ella pueda reflejarse y encontrar motivación.

De manera especial, dedico este trabajo a mi querida prima, Arantza Mayte Montalvo Moreta, quien desde el cielo ha iluminado mi camino hacia este logro. Este título también es tuyo, como parte de la promesa que una vez compartimos.

Finalmente, agradezco profundamente a Teresa Tuquerres, cuya constante exigencia y orientación me han ayudado a crecer tanto en el ámbito académico como en el personal, encauzándome hacia la responsabilidad y el compromiso necesarios para alcanzar este sueño.

AGRADECIMIENTO

Agradezco profundamente la guía y el apoyo brindado por el Msc. Fernando Valencia y el PhD. David Ojeda, quienes, con su vasta experiencia y conocimiento, hicieron posible la culminación de este trabajo de grado. Su orientación fue clave en cada etapa de este proceso.

Mi más sincero agradecimiento a mis padres, Verónica Moreta y Jaime Chiguano, cuyo sacrificio, esfuerzo y amor incondicional han sido la base que me permitió llegar a este día. Su apoyo constante y su cariño han hecho de este camino un recorrido más llevadero y significativo. A mi hermana, Melany Chiguano, por su apoyo incondicional, siempre alentándome a seguir adelante. Extiendo mi gratitud a mis tíos, David Calderón y Arturo Moreta, quienes estuvieron presentes en cada momento que los necesité, brindándome su respaldo incondicional a lo largo de este arduo proceso.

A mi novia, la Ing. Teresa Tuquerres, por ser un pilar fundamental en mi vida y en mi formación académica. Su apoyo constante, motivación y confianza en mí fueron determinantes para alcanzar esta meta; sin su ayuda incondicional, esto no habría sido posible.

A mis amigos, por las alegrías compartidas y las experiencias vividas a lo largo de mi vida universitaria. Gracias por estar presentes en los momentos buenos y en los difíciles, y por las lecciones aprendidas que quedarán siempre grabadas en mi memoria.

Finalmente, al Ing. Danny Orbes, por su valiosa guía y asesoría en el área del aprendizaje Braille, cuyo conocimiento y apoyo fueron esenciales para el desarrollo de este proyecto de grado.

RESUMEN EJECUTIVO

El proyecto se centra en el desarrollo de un dispositivo mecatrónico destinado a facilitar el aprendizaje de las operaciones básicas de matemáticas en Braille para estudiantes con discapacidad visual. Este dispositivo busca superar las barreras que enfrentan estos estudiantes al aprender conceptos matemáticos abstractos, proporcionando una herramienta interactiva y accesible. El estudio abarca la determinación de los requisitos específicos del dispositivo, el diseño de un sistema interactivo, la construcción de un prototipo funcional y la validación de este. El objetivo final es crear una solución efectiva que mejore la educación inclusiva y contribuya a una mejor comprensión de las matemáticas para los estudiantes con discapacidad visual.

Palabras clave: Dispositivo mecatrónico, Braille, educación inclusiva, discapacidad visual, matemáticas, operaciones básicas, interacción, validación.

ABSTRACT

The project focuses on the development of a mechatronic device aimed at facilitating the learning of basic mathematical operations in Braille for visually impaired students. This device seeks to overcome the barriers faced by these students when learning abstract mathematical concepts by providing an interactive and accessible tool. The study encompasses the determination of the specific requirements of the device, the design of an interactive system, the construction of a functional prototype and its validation. The goal is to create an effective solution that enhances inclusive education and contributes to a better understanding of mathematics for visually impaired students.

Keywords: Mechatronic device, Braille, inclusive education, visual impairment, mathematics, basic operations, interaction, validation.

ÍNDICE DE CONTENIDOS

1. IDENTIFICACIÓN DE LA OBRA	2
2. CONSTANCIAS	3
CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN	
CURRICULAR	4
APROBACIÓN DEL COMITÉ CALIFICADOR	5
DEDICATORIA	6
AGRADECIMIENTO	7
RESUMEN EJECUTIVO.....	8
ABSTRACT.....	9
ÍNDICE DE CONTENIDOS	10
ÍNDICE DE FIGURAS	15
ÍNDICE DE TABLAS	17
ÍNDICE DE ECUACIONES	18
CAPÍTULO I	19
INTRODUCCIÓN	19
1.1. Planteamiento del Problema	19
1.2. Objetivos.....	20
1.2.1. Objetivo General	20
1.2.2. Objetivos Específicos.....	20

1.3. Justificación	21
1.4. Alcance	23
CAPÍTULO II: MARCO TEÓRICO	24
2.1. Antecedentes.....	24
2.2. Fundamentos Teóricos.....	29
2.2.1. Origen y Evolución del sistema Braille.	29
2.2.2. Dispositivos de aprendizaje de matemática Braille.	29
2.2.3. Importancia de la educación inclusiva.....	32
2.2.3.1. Desafíos en la educación de personas con capacidad visual....	32
2.2.3.2. Metodologías actuales para las enseñanzas de matemáticas en personas con discapacidad visual.....	33
2.2.4. Dispositivos Mecatrónicos comerciales para la educación inclusiva.	
34	
CAPÍTULO III: MATERIALES Y MÉTODOS	36
3.1. Modelo de Investigación.....	36
3.2. Diseño de Investigación.....	37
3.2.1. Fase 1: Determinación de los requerimientos del dispositivo para el uso de estudiantes con discapacidad visual.....	37
3.2.2. Fase 2: Diseño de un mecanismo que permita interactuar con los usuarios.	37

3.2.3.	Fase 3: Construcción del prototipo funcional del dispositivo mecatrónico	38
3.2.4.	Fase 4: Validación del prototipo mediante pruebas de funcionamiento	39
CAPÍTULO IV: RESULTADOS Y DISCUSIÓN		40
4.1.	Especificaciones del sistema a diseñar	40
4.2.	Elementos requeridos.....	42
4.2.1.	Entrada de datos	42
4.2.2.	Microcontrolador.....	43
4.2.3.	Sistema mecánico.....	44
4.2.4.	Comunicación Inalámbrica: Bluetooth HC-06.....	46
4.3.	Planteamiento de alternativas de solución para el sistema de respuesta.	46
4.3.1.	Propuesta de solución número uno para el sistema de respuesta	47
4.3.2.	Propuesta de solución número dos para el sistema de respuesta	49
4.4.	Estudio comparativo entre propuesta 1 y propuesta 2	51
4.5.	Mecanismo de solución seleccionada	57
4.6.	Sistema de alimentación	58
4.7.	Diseño de propuesta seleccionada	60
4.7.1.	Sistema Botonera	61
4.7.2.	Sistema Corrección de Respuesta	62

4.7.2.1.	Estudio tornillo sin fin.....	64
4.7.2.2.	Transmisión de movimiento rotacional a lineal	65
4.8.	Diagrama eléctrico	66
4.9.	Algoritmo de programación.....	67
4.9.1.	Ingreso de datos.....	67
4.9.2.	Corrección de respuesta	79
4.10.	Construcción y ensamble del dispositivo	88
4.10.1.	Sistema estructural	88
4.10.1.1.	Carcasa MDF	88
4.10.1.2.	Estructura de soporte electrónico	89
4.10.2.	Sistema mecánico.....	92
4.11.	Ensamble del dispositivo.....	94
4.12.	Pruebas	95
4.12.1.	Prueba Nro. 1	96
4.12.2.	Prueba Nro. 2	97
4.12.3.	Prueba Nro. 3	98
4.12.4.	Prueba Nro. 4	100
4.12.5.	Prueba Nro. 5	102
	CONCLUSIONES	106
	RECOMENDACIONES.....	107

REFERENCIAS.....	108
ANEXOS	116
Anexo A.....	116
Diagrama eléctrico.....	116
Anexo B	117
Algoritmo de ingreso de datos.....	117
Anexo C	118
Algoritmo de corrección.....	118
Anexo D.....	119
Planos.....	119
Anexo E	120
Manual de Usuario.....	120
Anexo F.....	121
Manual de Mantenimiento	121

ÍNDICE DE FIGURAS

Fig. 2.1 Louis Braille [19].	29
Fig. 2.2 Ábaco Cranmer [20].	30
Fig. 2.3 Regletas de cálculo [21].	30
Fig. 2.4 Regleta y punzón Braille [22].	31
Fig. 2.5 Orión TI-84 [23].	31
Fig. 2.6 Duxbury Braille Translator [24].	31
Fig. 2.7 BrailleNote Touch Plus [26].	34
Fig. 2.8 Focus Braille Display [27].	35
Fig. 2.9 TactileView [28].	35
Fig. 4.1 Dispositivos de entrada de datos (a) DS-428 [34], (b) Encoder rotativo [35].	43
Fig. 4.2 Microcontroladores (a) Arduino UNO [37], (b) Arduino MEGA [38].	44
Fig. 4.3 Sistema mecánico (a) SG90 [40], (b) MG90s [41].	46
Fig. 4.4 Bluetooth HC-06 [38].	46
Fig. 4.5 Ensamble propuesta número uno.	47
Fig. 4.6 Piezas de la propuesta número uno. (a) Rueda dentada, (b) Cremallera de desplazamiento, (c) Cremalleras, (d) Matriz, (e) Husillos, (f) Acoples.	48
Fig. 4.7 Ensamble propuesta número dos.	49
Fig. 4.8 Piezas de la propuesta dos. (a) Acople para el servomotor, (b) Tornillo, (c) Cilindro, (d) Guía del mecanismo.	50
Fig. 4.9 Comparación de puntuaciones entre propuesta 1 y 2.	56
Fig. 4.10 Propuesta dos.	57

Fig. 4.11 Sistema de alimentación. (a) S-60-12 [43], (b) Step down xl4015e1 [44].	59
Fig. 4.12 Diseño de la propuesta seleccionada en vista explosionada.	60
Fig. 4.13 Sistema de botoneras.....	61
Fig. 4.14 Sistema de corrección de respuesta.	62
Fig. 4.15 Sección 3.	63
Fig. 4.16 Prueba experimental de tornillo sin fin.	65
Fig. 4.17 Diagrama de Flujo de entrada de datos.	68
Fig. 4.18 Diagrama de flujo de control operacional.	80
Fig. 4.19 Carcasa en MDF.....	88
Fig. 4.20 Estructura de soporte electrónico. (a) Case regulador, (b) Case Arduino Mega, (c) Case Bluetooth, (d) Case driver, (e) Case encoder.	91
Fig. 4.21 Sistema mecánico.....	94
Fig. 4.22 Ensamble del dispositivo.....	94
Fig. 4.23 Prueba Nro.1.....	96
Fig. 4.24 Prueba Nro.2.....	98
Fig. 4.25 Prueba Nro.3.....	100
Fig. 4.26 Prueba Nro.4.....	102
Fig. 4.27 Prueba Nro.5.....	104
Fig. 4.28 Porcentaje de éxito en el dispositivo.	105

ÍNDICE DE TABLAS

Tabla 2.1 Dispositivos electrónicos Braille.	30
Tabla 2.2 Dispositivos Mecatrónicos Braille.....	34
Tabla 4.1 Comparativa entre Arduino UNO y MEGA [34].	44
Tabla 4.2 Comparativa entre servomotor SG90 y MG90s [39].	45
Tabla 4.3 Cantidad de elementos de la propuesta número uno.....	49
Tabla 4.4 Cantidad de elementos de la propuesta número dos.	50
Tabla 4.5 Ponderación porcentual de los criterios de selección.	56
Tabla 4.6 Tabla de consumo eléctrico.	58
Tabla 4.7 Costo de fabricación de la estructura en MDF.	89
Tabla 4.8 Costos de fabricación estructura impresión 3D.....	92
Tabla 4.9 Costo de fabricación sistema mecánico.	93
Tabla 4.10 Resultados de la prueba Nro. 1	96
Tabla 4.11 Porcentaje de éxito de la prueba Nro.1	96
Tabla 4.12 Resultados de la prueba Nro.2.....	97
Tabla 4.13 Porcentaje de éxito en la prueba Nro.2	98
Tabla 4.14 Resultados de la prueba Nro.3	99
Tabla 4.15 Porcentaje de éxito en la prueba Nro.3	99
Tabla 4.16 Resultados de prueba Nro.4.....	100
Tabla 4.17 Porcentaje de éxito en la prueba Nro.4.....	101
Tabla 4.18 Resultados de la prueba Nro.4.....	103
Tabla 4.19 Porcentaje de éxito en la prueba Nro.5	104

ÍNDICE DE ECUACIONES

Ecuación 4.1 Criterio cuantitativo [42].....	55
Ecuación 4.2 Paso de la Rosca [45].....	63

CAPÍTULO I

INTRODUCCIÓN

En la parte introductoria de este trabajo se presenta la problemática y la propuesta de solución de manera clara y precisa que se desarrollará en el transcurso del proyecto.

1.1.Planteamiento del Problema

La educación inclusiva para estudiantes con discapacidad visual en el contexto del Ecuador presenta desafíos significativos, especialmente en el ámbito de las matemáticas. Actualmente, existe una falta de recursos educativos adaptados al Braille que aborden de manera específica las operaciones básicas de matemáticas, lo cual dificulta el acceso equitativo y la participación de estos estudiantes en el aprendizaje y la práctica de dichas operaciones [1].

Estos estudiantes se enfrentan a barreras en su proceso de aprendizaje, ya que los recursos y materiales disponibles para el estudio de las matemáticas no están adecuadamente adaptados a sus necesidades. El Braille es un sistema de lectura y escritura táctil que permite a las personas con discapacidad visual acceder a información escrita, pero la falta de recursos específicos para el aprendizaje de las operaciones básicas de matemáticas en Braille limita su progreso en esta área fundamental [2].

Además, la ausencia de herramientas interactivas y adaptadas a la realidad local dificulta la adquisición de habilidades matemáticas, lo que impacta negativamente en el desarrollo académico y personal de los estudiantes con discapacidad visual. La falta de dispositivos electrónicos diseñados específicamente para el aprendizaje de operaciones

básicas de matemáticas en Braille dificulta aún más el acceso a una educación inclusiva y de calidad [3].

Por lo tanto, es necesario abordar esta problemática y desarrollar un dispositivo de aprendizaje de operaciones básicas de matemáticas en Braille que proporcione a los estudiantes con discapacidad visual en Ecuador una herramienta efectiva y adaptada a sus necesidades. Este dispositivo ayudará a superar las barreras existentes, permitiendo que los estudiantes adquieran habilidades matemáticas fundamentales y promoviendo su inclusión en el ámbito educativo [4].

1.2.Objetivos

1.2.1. Objetivo General

Desarrollar un dispositivo mecatrónico que facilite el aprendizaje de las operaciones básicas de matemáticas en Braille para estudiantes con discapacidad visual.

1.2.2. Objetivos Específicos

- Determinar los requerimientos del dispositivo para el uso de estudiantes con discapacidad visual.
- Diseñar un sistema que permita interactuar con los usuarios.
- Construir un prototipo funcional del dispositivo mecatrónico.
- Validar el prototipo.

1.3. Justificación

En el contexto del Ecuador, existe una necesidad urgente y creciente de desarrollar recursos y herramientas educativas inclusivas que promuevan la igualdad de oportunidades para los estudiantes con discapacidad visual. La educación es un pilar fundamental para el desarrollo personal, social y profesional de cualquier individuo, y el acceso a una educación de calidad en matemáticas es particularmente importante, ya que esta área fomenta el pensamiento lógico, la resolución de problemas y las habilidades analíticas. Sin embargo, los estudiantes con discapacidad visual enfrentan barreras significativas debido a la escasez de recursos específicos diseñados para facilitar el aprendizaje de las operaciones básicas de matemáticas en Braille. Esta limitación crea una brecha en su desarrollo académico, afectando directamente su participación y rendimiento en esta disciplina crucial [5].

El sistema Braille, reconocido a nivel mundial como un medio eficaz de lectura y escritura para personas con discapacidad visual, desempeña un papel esencial en la educación inclusiva. A pesar de su importancia, la disponibilidad de herramientas educativas que integren el Braille en el aprendizaje de las matemáticas sigue siendo muy limitada, tanto a nivel global como en el Ecuador. Esta carencia representa un desafío considerable para los estudiantes con discapacidad visual, quienes a menudo enfrentan dificultades para acceder a materiales adaptados que les permitan aprender y practicar habilidades matemáticas básicas de manera autónoma y efectiva [6].

La implementación de un dispositivo innovador diseñado específicamente para el aprendizaje de operaciones matemáticas básicas en Braille tiene el potencial de transformar la experiencia educativa de los estudiantes con discapacidad visual en el Ecuador. Este dispositivo, concebido como una herramienta interactiva y personalizada, no solo aborda las

limitaciones actuales, sino que también fomenta un entorno de aprendizaje más inclusivo. Al proporcionar una solución tecnológica accesible y adecuada, se les brinda a estos estudiantes la oportunidad de adquirir habilidades fundamentales que son esenciales para su desarrollo académico, cognitivo y personal. Además, este enfoque refuerza su autonomía, permitiéndoles participar de manera más activa y equitativa en el proceso educativo [7].

En comparación con las calculadoras Braille comerciales, este proyecto se diferencia al no incluir la funcionalidad de reproducción de audio. Esto responde a un análisis contextual que revela que el uso de audio puede generar distracciones significativas en aulas con múltiples estudiantes, dificultando la concentración tanto del usuario como de sus compañeros. Por esta razón, la solución propuesta en esta tesis prioriza la creación de una calculadora silenciosa y autónoma, enfocada en las necesidades específicas del estudiante durante evaluaciones o actividades en grupo. Esta característica distintiva asegura un aprendizaje más eficiente y un ambiente educativo más equilibrado [8].

En resumen, el desarrollo de un dispositivo para el aprendizaje de operaciones matemáticas básicas en Braille dirigido a estudiantes con discapacidad visual en el Ecuador responde a una necesidad urgente y prioritaria de brindar oportunidades educativas equitativas y fomentar la inclusión. Este dispositivo no solo facilitará el acceso a las matemáticas, sino que también fortalecerá la participación activa de estos estudiantes en su proceso de aprendizaje. De esta manera, contribuirá al desarrollo de sus capacidades académicas, a la mejora de su calidad de vida y a su plena integración en la sociedad, promoviendo un futuro más inclusivo y accesible para todos.

1.4. Alcance

El dispositivo está diseñado específicamente para estudiantes con discapacidad visual que hayan cursado o estén cursando el segundo nivel de aprendizaje en Braille, pertenecientes al Centro de Desarrollo Social y Económico “Santo Domingo” (Asociación de No Videntes de Imbabura). Su objetivo principal es facilitar el aprendizaje de las operaciones matemáticas básicas, enfocándose en la suma y la resta. Esta herramienta permitirá trabajar con dos números enteros de hasta dos dígitos, generando resultados que serán números enteros de hasta tres dígitos.

El equipo trabajará evaluando la respuesta proporcionada por el estudiante. Si la respuesta es correcta, el mecanismo determinará su veracidad. En el caso de una respuesta incorrecta, el mecanismo corregirá la respuesta, eliminándola y reemplazándola por la respuesta correcta. Esto se llevará a cabo mediante un mecanismo de matriz de relieve autónomo.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes

A continuación, se presentan diferentes artículos que abordan el desarrollo y ejecución de un módulo interactivo para la enseñanza de pre-Braille. El dispositivo permite al usuario formar letras y números mediante piezas extraíbles que representan los puntos en relieve del sistema Braille. Cada símbolo puede ser reproducido de forma auditiva a través de botones con texturas y frases específicas. El dispositivo está basado en hardware y software libre, desarrollado con Python y una Raspberry Pi. Al completarse, el dispositivo identifica y valida los símbolos del sistema Braille al 100%, esto se validado por un equipo pedagógico [5].

Asimismo, para facilitar el aprendizaje de letras y caracteres a personas con discapacidad visual, el sistema Braille utiliza relieves en superficies que representan dichos símbolos. Sin embargo, enseñar Braille a niños con discapacidad visual presenta desafíos que dificultan su aprendizaje. En respuesta, se ha desarrollado un dispositivo interactivo que les permite aprender de manera lúdica sin la necesidad constante de un tutor. Con un planteamiento cualitativo-interpretativo, plantea la creación de un dispositivo que combina hardware libre y un reproductor de audio, integrados en una maqueta de MDF (Medium Density Fibreboard). Los estudiantes colocan barras en las perforaciones de la maqueta, y el software, en conjunto con el hardware, reconoce la letra ingresada mediante codificación, facilitando su avance a través de la maqueta [6].

Adicionalmente, las herramientas educativas accesibles para personas con discapacidad visual son escasas, fundamentalmente debido a su elevado costo y la falta de

distribución global, particularmente en países latinoamericanos. Esta situación contribuye a la elevada tasa de analfabetismo y desempleo entre esta población. El proyecto a continuación tiene como objetivo otorgar un dispositivo con retroalimentación auditiva, diseñado para facilitar tanto el aprendizaje como la enseñanza del sistema Braille a personas con discapacidad visual. Para ello, se implementó diferentes etapas como la revisión bibliográfica, delimitación de parámetros, diseño y el montaje del artefacto, desarrollo de una plataforma virtual, y comprobación del sistema completo. Todos estos resultados se envían a la plataforma virtual mientras se conecta un dispositivo que contenga el signo generador Braille que al ser activado produce la realimentación auditiva integrada en la plataforma virtual [7].

También, se presenta el prototipo de un dispositivo electrónico que permite a las personas con discapacidad visual relacionarse con su entorno sin el requerimiento de apoyo externo. El dispositivo está diseñado para asistir a los profesores en la enseñanza de Braille mediante una aplicación móvil Android, que facilita el aprendizaje secuencial de los caracteres y permite su práctica. Además, incluye una base de datos para el seguimiento del progreso y un simulador de caracteres Braille. Este simulador fue fabricado a través de impresión 3D y se conecta de forma inalámbrica al dispositivo móvil. Para registrar y monitorear el uso, se implementó una base de datos accesible a través de la web, conectada externamente fuera de la red LAN mediante una red privada virtual (VPN) [8].

Directamente, el proceso de aprendizaje para personas con discapacidad se optimiza al combinar metodologías pedagógicas y herramientas tecnológicas. Este estudio tiene como objetivo implementar un modelo de software para la educación - formación del sistema Braille en niños con discapacidad visual. Para el desarrollo del software, se utilizó la

metodología en cascada, logrando un sistema que ha resultado satisfactorio tanto para estudiantes como para docentes, mejorando la memoria y las habilidades motoras de los alumnos. No obstante, este prototipo representa solo el inicio del desarrollo del sistema, con planes de agregar más funciones para cubrir completamente los contenidos de lengua y literatura. Esta investigación expone la viabilidad de integrar tecnología en la educación inclusiva [9].

A lo largo del periodo de aislamiento por la pandemia de COVID-19, se demostró la problemática en el ámbito de la educación inclusiva. Aunque se han desarrollado enfoques efectivos, estos asumían condiciones ideales, sin considerar escenarios de peligro. Para abordar este desafío, se está desarrollando un prototipo que involucre la interconexión de un maestro y el estudiante con discapacidad visual, permitiendo continuar con la enseñanza de matemáticas de manera remota. El desarrollo del prototipo se basa en un servidor virtual privado (VPS), que facilitará la enseñanza a través de una página web manejada por el maestro. Esta interfaz se usa como intermediario, traduciendo los símbolos matemáticos al lenguaje Braille mediante un prototipo electrónico. Para valorar la eficacia del dispositivo, se implementará un método de enseñanza para operaciones matemáticas básicas, y se analizarán las interacciones de los estudiantes con el prototipo para validar el enfoque de aprendizaje [10].

A continuación, se presenta un prototipo semiautomático para la enseñanza de lenguaje Braille, implica en un sistema mecatrónico desarrollado para asistir a niños con discapacidad visual en las operaciones de sumas y restas de un solo dígito, facilitando un aprendizaje más rápido. Además, incorpora juegos interactivos para hacerlo más amigable y

atractivo. La investigación educativa realizada destacó la importancia de este tipo de dispositivos para fomentar la inclusión social y mejorar el entorno educativo [11].

Por otra parte, se proponen cómo tecnologías de impresión 3D y el uso de herramientas CAD facilitan el modelado y producción de un prototipo de enseñanza destinado a capacitar matemáticas avanzadas y notación científica a estudiantes con discapacidad visual. El Instituto Tecnológico de La Laguna, implementa un programa de Vinculación Comunitaria donde estudiantes de Maestría y Licenciatura colaboran con sectores marginados de la sociedad. Actualmente, se está desarrollando el cumplimiento de los requisitos de los estudiantes, determinando el perfil del recurso educativo, que abarca la información de especificaciones y requisitos para asegurar su correcto funcionamiento y reproducibilidad. Para integrar a más estudiantes con discapacidad visual en el sistema educativo, es esencial que alcancen logros significativos no solo en el ámbito académico, sino también en su desarrollo emocional y social [12].

Se demostró la importancia de la Tiflotecnología en la instrucción de nuevos licenciados en matemáticas de la Universidad del Atlántico, así como en las acciones de instrucción y formación para estudiantes con discapacidad visual. El análisis se fomentó a través de un estudio de caso cualitativo, que incluyó etapas de organización, análisis e interpretación de datos. Los participantes aprendieron sobre el funcionamiento de sistemas autónomos y no autónomos en software y hardware, accediendo a dispositivos móviles y de cómputo, identificando la relevancia de estas herramientas para su formación profesional. No obstante, la formación de los docentes en matemáticas es deficiente en temas de inclusión, lo que limita su capacidad para trabajar eficazmente con esta población [13].

En definitiva, la principal barrera que enfrentan las personas con discapacidad visual es la falta de acceso a la información y a dispositivos específicamente diseñados para sus necesidades. La investigación que aborda esta problemática mediante el desarrollo de un prototipo de calculadora Braille, se encuentra orientado a mejorar el aprendizaje de usuarios no videntes. La calculadora está equipada con una unidad de procesamiento de datos, una base de datos con archivos de audio y una interfaz que incorpora símbolos matemáticos. El dispositivo permite realizar operaciones matemáticas básicas y reproduce automáticamente tanto los valores ingresados como los resultados, elaborada para cumplir las necesidades de personas con discapacidad visual [14].

2.2. Fundamentos Teóricos

A continuación, se presentan los fundamentos teóricos que sustentan el desarrollo de este proyecto, proporcionando el marco conceptual necesario para comprender los principios y bases que lo respaldan.

2.2.1. *Origen y Evolución del sistema Braille.*

El sistema Braille fue diseñado en el siglo XIX por Louis Braille (véase la Fig. 2.1), quien perdió la vista a los tres años. Inspirado en la "escritura nocturna" de Charles Barbier, un sistema táctil militar, Braille redujo la complejidad original de 12 a 6 puntos, optimizando así la lectura táctil para personas ciegas. Su diseño permitió representar letras, números y signos de puntuación de manera eficiente. Aunque inicialmente no fue aceptado, la adopción gradual en instituciones educativas y su posterior reconocimiento en 1854 impulsaron su expansión global. La adaptabilidad del sistema a diferentes idiomas, notación matemática y musical, y su integración con tecnologías modernas ha permitido que millones de personas ciegas accedan a la educación y la vida diaria con independencia.




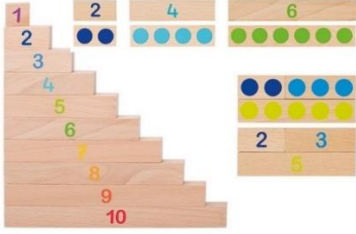
Fig. 2.1 Louis Braille [19].

2.2.2. *Dispositivos de aprendizaje de matemática Braille.*

Con la llegada de la tecnología, el sistema Braille ha evolucionado aún más. Se han desarrollado dispositivos electrónicos de Braille (observar Tabla 2.1), como las líneas braille y los teclados Braille, que permiten a las personas ciegas interactuar con computadoras y

otros dispositivos digitales. Las impresoras Braille modernas pueden producir textos en Braille de manera rápida y eficiente, aumentando el acceso a materiales escritos para personas ciegas.

Tabla 2.1 Dispositivos electrónicos Braille.

<i>Descripción</i>	Gráfico	Ventajas
<p>Ábacos Adaptados: El ábaco Cranmer, adaptado para personas con discapacidad visual, es una herramienta tradicional para enseñar y realizar cálculos matemáticos básicos.</p>		<p>Su diseño portátil como se observa en la Fig. 2.2, permite mover cuentas a lo largo de varillas para representar números y realizar operaciones aritméticas.</p>
<p>Regletas de Cálculo: Estas regletas son barras táctiles que representan números y operaciones matemáticas. Se usan para enseñar conceptos como suma, resta, multiplicación y división.</p>		<p>Las regletas pueden ser manipuladas físicamente, lo que ayuda a los estudiantes a comprender mejor las operaciones matemáticas (ver Fig. 2.3).</p>

Regleta y punzón de Braille: Los tableros de Braille y punzones se utilizan para escribir en Braille, incluyendo notación matemática.



Fig. 2.4 Regleta y punzón Braille [22].

Practicar la escritura y lectura de números, símbolos matemáticos y cálculos en Braille (observar Fig. 2.4).

Calculadoras Braille: Estas calculadoras, diseñadas para personas con discapacidad visual, combinan funciones gráficas con interfaces auditivas y táctiles.



Fig. 2.5 Orión TI-84 [23].

En la Fig. 2.5 se muestra una calculadora que permiten realizar cálculos avanzados y acceder a gráficos y funciones matemáticas.

Software Matemático Adaptado: Duxbury convierte texto y contenido matemático a Braille, facilitando la creación de materiales educativos accesibles.

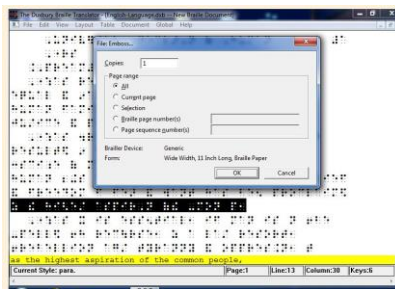


Fig. 2.6 Duxbury Braille Translator [24].

Es útiles para estudiantes con discapacidad visual al combinarse con lectores de pantalla y líneas Braille como se muestra en la Fig. 2.6.

2.2.3. *Importancia de la educación inclusiva*

La educación inclusiva garantiza el acceso de todos los estudiantes a una enseñanza de calidad en un entorno que valore la diversidad. El sistema Braille, fundamental para la alfabetización y autonomía de las personas ciegas, facilita su participación en áreas como literatura, ciencia y tecnología. Este enfoque fomenta la aceptación y bienestar emocional de los estudiantes con discapacidades, mientras que las estrategias inclusivas mejoran la calidad educativa para todos [25].

2.2.3.1. Desafíos en la educación de personas con capacidad visual.

La educación de estudiantes con discapacidad visual presenta retos específicos que deben abordarse para asegurar una experiencia educativa inclusiva y equitativa. Estos retos incluyen desde la accesibilidad de los materiales de estudio hasta la capacitación adecuada de los docentes y la integración de los estudiantes en la vida social y académica.

Escasez de materiales: La disponibilidad de libros de texto y otros recursos educativos en Braille o formatos accesibles suele ser limitada, dificultando el acceso a la información para estos estudiantes. La conversión de materiales al formato adecuado puede ser un proceso prolongado, lo que provoca retrasos en la disponibilidad de dichos recursos.

Tecnología asistiva: Herramientas tecnológicas, como líneas Braille electrónicas, impresoras Braille y software de lectura de pantalla, suelen ser costosas y no están disponibles en todas las instituciones educativas. Además, la tecnología asistiva requiere actualizaciones y mantenimiento frecuentes, lo cual representa un desafío para aquellas escuelas con presupuestos restringidos.

Falta de formación especializada: Muchos docentes carecen de capacitación específica para enseñar a estudiantes con discapacidad visual, lo que limita su capacidad para adaptar sus métodos y materiales educativos. Además, la falta de acceso a oportunidades de desarrollo profesional continuo en educación inclusiva puede impactar negativamente la calidad de la enseñanza ofrecida a estos estudiantes.

2.2.3.2. Metodologías actuales para las enseñanzas de matemáticas en personas con discapacidad visual.

La enseñanza de matemáticas a personas con discapacidad visual requiere metodologías y herramientas específicas que hagan que los conceptos abstractos y simbólicos sean accesibles y comprensibles a través de medios táctiles y auditivos.

- **Enfoque Multisensorial:** Utiliza múltiples sentidos (tacto, audición, y a veces el olfato y el gusto) para enseñar conceptos matemáticos. Este enfoque ayuda a reforzar el aprendizaje a través de diferentes canales sensoriales. Se emplean objetos táctiles, sonidos, y tecnologías que emiten señales auditivas para representar conceptos matemáticos como números, operaciones y formas geométricas
- **Aprendizaje Basado en la Manipulación:** Involucra el uso de materiales manipulativos que los estudiantes pueden tocar y mover para comprender mejor los conceptos matemáticos. El uso de regletas de cálculo, ábacos adaptados y bloques matemáticos permite a los estudiantes explorar y resolver problemas de manera tangible
- **Enseñanza Directa y Sistemática:** Un enfoque estructurado y paso a paso que incluye instrucción explícita, práctica guiada y retroalimentación

inmediata. Los maestros proporcionan instrucciones claras y detalladas, seguidas de oportunidades de práctica con apoyo y luego práctica independiente

- **Uso de Narrativas y Contextos Reales:** Integra problemas matemáticos dentro de historias o contextos de la vida real para hacer los conceptos más significativos y relevantes. Se utilizan situaciones cotidianas o problemas basados en la vida real que los estudiantes pueden relacionar y entender mejor.

2.2.4. *Dispositivos Mecatrónicos comerciales para la educación inclusiva.*

Los dispositivos mecatrónicos combinan elementos mecánicos, electrónicos y computacionales para crear herramientas avanzadas que pueden ser utilizadas en la educación inclusiva. Los dispositivos que se muestran en la Tabla 2.2 están diseñados para mejorar el acceso a la información y facilitar el aprendizaje de estudiantes con discapacidades.

Tabla 2.2 Dispositivos Mecatrónicos Braille.


Dispositivo	Gráfico	Función
<p>BrailleNote Touch Plus:</p> <p>Es un dispositivo de toma de notas en Braille que combina una línea Braille con una pantalla táctil y funciones de Android (ver Fig. 2.7).</p>		<p>Permite a los estudiantes tomar notas, leer libros electrónicos, navegar por internet y acceder a aplicaciones educativas en formato accesible, y opciones de entrada táctil o teclado Braille.</p>

Fig. 2.7 BrailleNote Touch Plus [26].

Focus Braille Display:

Una línea Braille compacta y robusta que se conecta a dispositivos móviles y computadoras (véase Fig. 2.8).



Fig. 2.8 Focus Braille Display [27].

Facilita la lectura y escritura en Braille, compatible con varios lectores de pantalla y dispositivos móviles, conectividad Bluetooth y USB, teclas de navegación intuitivas, y diseño portátil.

TactileView:

El software y dispositivo de impresión que se aprecia en la Fig. 2.9 crea gráficos táctiles para personas con discapacidad visual

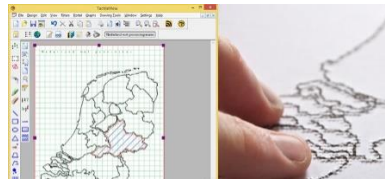


Fig. 2.9 TactileView [28].

Permite la creación de diagramas táctiles, gráficos matemáticos y otros materiales visuales en formato accesible, impresoras térmicas o en relieve, software de diseño intuitivo, y soporte para diversos formatos de archivo.

CAPÍTULO III: MATERIALES Y MÉTODOS

Este capítulo proporciona una descripción detallada de la metodología empleada para desarrollar el dispositivo mecatrónico para el aprendizaje de las operaciones básicas de matemáticas en Braille, Además, se explica el procedimiento para verificar los objetivos específicos establecidos en este documento.

3.1. Modelo de Investigación

El proyecto aborda investigación aplicada, debido a que esta genera una solución a un problema planteado [15], que trata del desarrollo de un dispositivo mecatrónico de aprendizaje de operaciones básicas de matemáticas Braille.

Así mismo, es de tipo documental ya que se reúne información de documentos como revistas, libros, artículos científicos, entre otros, [16]. De igual manera, consta con investigación de campo debido a que es de suma importancia la visita a la asociación de no videntes Imbabura “Santo Domingo” para recaudar información en base a encuestas y entrevistas [17].

Así también, la investigación es de tipo descriptivo, siendo en esta donde se especifique las variables que se puedan dar dentro del objeto de estudio [18]. Y para culminar, es de tipo experimental ya que a través de la acción del investigador verifica el correcto funcionamiento del dispositivo a través de pruebas de ejecución [19].

3.2. Diseño de Investigación

Considerando lo mencionado anteriormente, se pueden estructurar las etapas de acuerdo con los objetivos planteados y las actividades asociadas a cada proceso.

3.2.1. Fase 1: Determinación de los requerimientos del dispositivo para el uso de estudiantes con discapacidad visual.

Actividad 1: Búsqueda de información acerca de la normativa Braille.

Se indaga en la página web de la CBE (Comisión Braille Española); puesto que es el máximo ente de regulación en España. El cual consolida las normas de uso y desarrollo del lenguaje Braille.

Actividad 2: Indagación del lenguaje matemático Braille.

Se busca información acerca de la simbología Braille aplicada al área de las matemáticas, como son signos e identificadores de números.

3.2.2. Fase 2: Diseño de un mecanismo que permita interactuar con los usuarios.

Actividad 1: Evaluar de los diferentes mecanismos de respuesta.

Se investiga los mecanismos que existen para la elaboración del sistema de respuesta con el fin de determinar su eficiencia, ventajas o limitaciones ante los requerimientos planteados, y así analizar cuál de ellos es el más adecuado o efectivo.

Actividad 2: Seleccionar del mecanismo apropiado.

Se efectúa un análisis de alternativas para seleccionar la opción más adecuada considerando los criterios y restricciones del diseño a realizarse.

Actividad 3: Adquirir de materiales.

Se formulan diferentes opciones de los materiales que podrían formar parte del dispositivo, analizando la disponibilidad en el mercado y sus costos. Obteniendo los elementos más adecuados para su posterior adquisición.

Actividad 4: Construir del mecanismo de respuesta.

Se construye el mecanismo para recrear el modelo de retroalimentación del dispositivo, el cual consiste en una matriz que ejecutara la respuesta correcta de la operación en el caso de que el usuario falle en la digitación de la solución.

3.2.3. Fase 3: Construcción del prototipo funcional del dispositivo mecatrónico

Actividad 1: Elaborar el sistema de comunicación.

Se realiza las conexiones de cada matriz hacia la placa seleccionada y de la cual se conectará las salidas al sistema de respuesta del dispositivo.

Actividad 2: Elaborar la programación de conocimientos numéricos.

Se elabora la programación, donde se recibe las señales de entrada y se procese la operación matemática donde emita una señal de salida y se envíe al mecanismo de respuesta.

Actividad 3: Ensamblar el dispositivo.

Una vez construido el sistema de comunicación y corroborado el correcto funcionamiento de la programación se procede a realizar el ensamble del dispositivo con todos los elementos.

3.2.4. Fase 4: Validación del prototipo mediante pruebas de funcionamiento

Actividad 1: Planteamiento de prueba.

Se realiza un documento con varios ejercicios a desarrollar en lenguaje Braille el cual será entregado al estudiante que maneje el dispositivo.

Actividad 2: Evaluar la funcionalidad del dispositivo con personal de ANVI.

Se observa el correcto funcionamiento del dispositivo al momento de que el usuario realiza su evaluación.

Actividad 3: Presentación del documento escrito y el módulo en funcionamiento.

Se presenta la documentación final y se valida el dispositivo con los docentes pertinentes.

CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

La matemática Braille es un lenguaje que se debe de entender a través de procesos asociados que brinden fácil entendimiento y adaptación a las diferentes edades, esto se logra mediante propuestas a la fabricación de prototipos con relieve, el desarrollo de este capítulo se concentra en la descripción de una propuesta específica que facilite el manejo y el entendimiento de dicho lenguaje, se estudia los diferentes elementos y las funciones que cumplen en conjunto con la programación planteada.

4.1. Especificaciones del sistema a diseñar

En la elaboración de proyectos para el entendimiento del lenguaje braille, se debe implementar diferentes pasos para brindar una solución válida e inclusiva, teniendo un orden a seguir donde se conforma el proceso de diseño y como consiguiente el desarrollo del dispositivo, que tiene como objetivo garantizar el entendimiento de matemática braille, para lo cual se presentan alternativas de solución que será evaluada en un estudio comparativo.

En el desarrollo de un proyecto, se intenta alcanzar la perfección en todos los aspectos. Sin embargo, no siempre es posible lograrlo. Es fundamental reconocer que hay ciertos requisitos que son más importantes que otros, y es en estos donde se debe centrar la atención para satisfacer las necesidades esenciales y dirigir los recursos de manera efectiva. Por ello, se proponen características que facilitan la toma de decisiones sobre el diseño a implementar.

Función. – La funcionalidad debe ser una de las principales características en el dispositivo, para ello la correcta ejecución de la matriz de corrección garantizará el aprendizaje de los usuarios con discapacidad visual

Precisión. - Al trabajar con personas con discapacidad visual es de gran importancia que los caracteres brailles plasmados mediante impresión 3D no superen los 3mm de altura, ya que al tener una dimensión sobrevalorada los usuarios pierden la capacidad de interpretar el lenguaje matemático Braille.

Eficiencia. - El dispositivo mecatrónico está diseñado para contribuir en la enseñanza de la matemática braille, de esta forma debe ofrecer fiabilidad al usuario en los resultados arrojados por la matriz de corrección, así como también siendo de gran importancia que el dispositivo genere el mínimo ruido posible, debido a las condiciones de los usuarios.

Estético. - El dispositivo debe contar con medidas proporcionalmente distribuidas haciendo del equipo lo más compacto y hermético.

Peso. - Se prioriza no sobrepasar de 1kg garantizando la movilidad del dispositivo por los usuarios.

Mantenimiento. - El dispositivo debe ser construido con elementos disponibles en el mercado nacional garantizando su reemplazo en caso de ser necesario.

Seguridad. - Al ser un dispositivo dirigido a personas con discapacidad visual se debe asegurar que todas sus partes electrónicas y eléctricas estén correctamente cubiertas y su estructura no debe contar con aristas.

Costo. - El dispositivo debe ser accesible, ya que está direccionado a entidades de apoyo a personas no videntes.

A continuación, se describen los requerimientos del dispositivo y los elementos que se utilizará basándose en un análisis con los criterios antes mencionados.

4.2. Elementos requeridos

A continuación, se presenta el listado de elementos necesarios para la fabricación y ensamblaje del dispositivo. Cada componente fue seleccionado cuidadosamente para cumplir con los requerimientos funcionales y técnicos del proyecto, asegurando la eficiencia y precisión en el funcionamiento del sistema.

4.2.1. Entrada de datos

Para el ingreso de datos se realizó un análisis conjunto con el usuario, debido a que se requiere considerar una similitud al lenguaje Braille. En base a ello se descarta cualquier tipo de dispositivo que no sea de forma circular o tenga una gran dimensión.

Los botones de enclavamiento permiten la apreciación de dos diferentes estados de posición sea ON (Permite en paso de la corriente) u OFF (Cierra el paso a la corriente), a diferencia de los botones momentáneos que cierran el circuito solo mientras el botón esta presionado físicamente contando así con un solo estado de posición. Este aspecto es fundamental debido a que cada botón representará el carácter Braille dentro de la celda y se desea que se pueda palpar el número insertado.

Debido a esto los botones de enclavamiento DS-428 que se muestran en la Fig. 4.1 (a), son los ideales para plasmar una dimensión palpable de 3 mm para el usuario. Y al ser fabricados de un material plástico contribuyen en la reducción de peso siendo el dispositivo más transportable. Así también su costo de 0.75 USD poro unidad lo hace accesible para el proyecto.

Además, el encoder rotativo que me aprecia en la Fig. 4.1 (b), proporciona una entrada más continua y precisa, lo que es ideal para ajustar los parámetros del sistema de forma incremental, como el control de parámetros o la selección de modos de operación.



Fig. 4.1 Dispositivos de entrada de datos (a) DS-428 [34], (b) Encoder rotativo [35].

4.2.2. Microcontrolador

En el dispositivo se empleará 46 botones de enclavamiento los cuales requieren una entrada digital, y una tierra común. Además de ello se requiere un procesamiento de datos con mayor velocidad debido a la demanda existente en el sistema de respuesta.

El Arduino Mega destaca por su capacidad para manejar múltiples entradas y salidas, con 54 pines digitales y 16 entradas analógicas (véase Fig. 4.2 (a)). En comparación con otros microcontroladores más reducidos, como el Arduino Uno (ver Fig. 4.2 (b)), el Mega otorga una mayor capacidad de memoria y pines de expansión, lo que posibilita una mayor flexibilidad y escalabilidad en sistemas complejos, siendo este el más acertado en base a las necesidades planteadas, se muestra mayor detalle de sus características en la Tabla 4.1.

Su costo es de aproximadamente 15 USD, y es de fácil acceso en el mercado.

Tabla 4.1 Comparativa entre Arduino UNO y MEGA [34].

Características	Arduino UNO	Arduino MEGA
Microcontrolador	ATmega328P	ATmega2560
Memoria Flash	32 KB	256 KB
SRAM	2 KB	8 KB
EEPROM	1 KB	4 KB
Pines digitales	14 (6 con PWM)	54 (15 con PWM)
Entradas analógicas	6	16
Puertos serie (UART)	1	4
Dimensiones	68.6 mm x 53.4 mm	101.52 mm x 53.3 mm
Voltaje de Operación	5V	5V



(a)



(b)

Fig. 4.2 Microcontroladores (a) Arduino UNO [37], (b) Arduino MEGA [38].

4.2.3. Sistema mecánico

Para el sistema de respuesta se requiere de 12 actuadores debido a ellos se considera el uso de servomotores, analizando sus dimensiones, peso y funcionalidad.

Los servomotores son componentes cruciales para generar el movimiento. En la Tabla 4.2 se muestra una comparativa de los dos servomotores estudiados en este caso. Debido a que se utilizan 12 actuadores, los servomotores SG90 que se muestra en la Fig. 4.3 (a), tienen

un bajo consumo de corriente, en comparación con otros servos más grandes como el MG90S que se observa en la Fig. 4.3 (b), los convierte en una opción eficiente para proyectos donde se requiere precisión en movimientos sin consumir grandes cantidades de energía, además en este proyecto no se requiere una gran capacidad de torque debido a que la carga de peso no es demandante ya que el mecanismo será fabricado en impresión 3D. Su costo comercial es de 3,50 USD lo cual los hace bastante accesibles.

Además, la conexión de estos actuadores a través del módulo PCA9685 permite un control individual de múltiples servos, reduciendo la carga sobre el microcontrolador y facilitando el control mediante el bus I2C. Su precio comercial es de 8,00 USD.

Tabla 4.2 Comparativa entre servomotor SG90 y MG90s [39].

Características	SG90	MG90s
Engranajes	Plástico	Metálicos
Peso	9g	13.4g
Torque	1.8 kg.cm a 4.8V	2.2 kg.cm a 4.8V
Velocidad	0.12s/60°	0.10s/60°
Angulo de rotación	180°	180°
Voltaje de operación	4.8V – 6.0V	4.8V – 6.0V
Material del cuerpo	Plástico	Plástico-metálico
Durabilidad	Menor	Mayor
Ruido	Moderado	Moderado
Costo	Generalmente más bajo	Ligeramente más caro



Fig. 4.3 Sistema mecánico (a) SG90 [40], (b) MG90s [41].

4.2.4. Comunicación Inalámbrica: Bluetooth HC-06

Se requiere una comunicación entre un dispositivo móvil hacia el microcontrolador, para ello la integración del módulo Bluetooth HC-06 que se muestra en la Fig. 4.4, permite la comunicación inalámbrica. A diferencia de otros módulos de comunicación, el HC-06 es sencillo de usar y consume relativamente poca energía, lo que lo hace ideal para proyectos que requieren interactividad sin cables y sin un consumo elevado de corriente. Su valor comercial es de 9,00 USD.



Fig. 4.4 Bluetooth HC-06 [38].

4.3.Planteamiento de alternativas de solución para el sistema de respuesta.

En esta sección se detallan dos propuestas para el sistema de respuesta del dispositivo. Estas propuestas, denominadas Propuesta 1 y Propuesta 2, se han diseñado con el fin de optimizar el rendimiento general del dispositivo. Es importante señalar que, aunque el

mecanismo de respuesta es diferente en cada propuesta, el ingreso de datos y el resto de los componentes seleccionados previamente para el proyecto, tales como los Arduino Mega, los botones momentáneos, los servomotores SG90, y el módulo Bluetooth HC-06, permanecen sin cambios en ambas configuraciones. Esto garantiza la coherencia y fiabilidad en la captura y procesamiento de la información, facilitando una comparación directa entre las dos alternativas planteadas para el sistema de respuesta.

4.3.1. Propuesta de solución número uno para el sistema de respuesta

En la Fig. 4.5, se observa la alternativa uno que empleará el mecanismo conocido como piñón (5) y cremallera (2), que tendrá la función de elevar los husillos (3) hacia la parte superior de la matriz (1), que representará la simbología matemática braille, todos estos elementos se mantendrán posicionados gracias a la estructura de soporte (4); el peso de este mecanismo es de: 128 gramos.

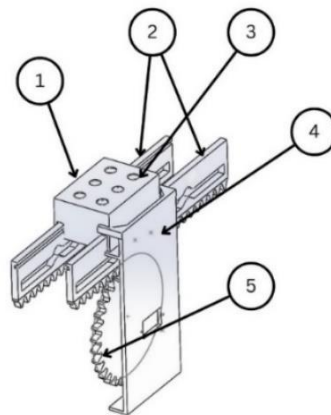


Fig. 4.5 Ensamble propuesta número uno.

En la Fig. 4.6 (a), se puede observar el piñón (rueda dentada), que cuenta con un diámetro de 90,75mm con un espesor de 5mm y está formada de 33 dientes a su alrededor. En la Fig. 4.6 (b), se aprecia la cremallera que se empleará para elevar los husillos en

diferentes posiciones, sus dimensiones son 200mm de largo, 30mm de ancho con un espesor de 5mm; en la parte inferior se acopla al piñon gracias a la presencia de 23 dientes los cuales servirán para el desplazamiento total de la pieza, en este sistema se emplean 2 cremalleras, una para cada lado (ver Fig. 4.6 (c)). El relieve se denotará desde la superficie de la matriz que se muestra en la Fig. 4.6 (d), y sus dimensiones son 65mm de largo, 45mm de ancho y una altura de 30mm. Para la representación de los caracteres Braille se emplearán 6 husillos (véase Fig. 4.6 (e)), los cuales tienen una altura de 42mm, y un diametro de 7mm. En la Fig. 4.6 (f), se aprecia los cilindros que encajan en la ranura de la cremallera.

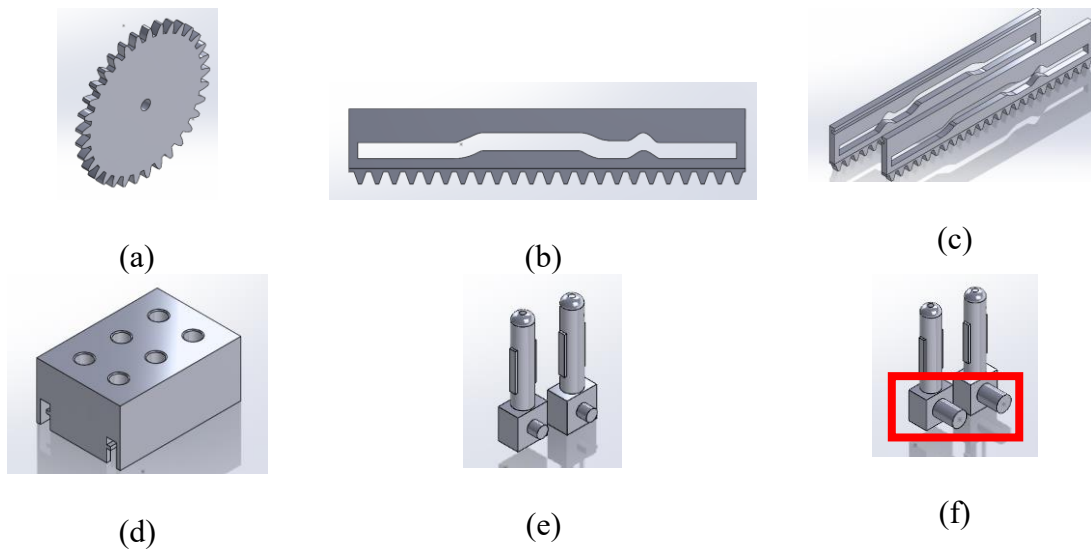


Fig. 4.6 Piezas de la propuesta número uno. (a) Rueda dentada, (b) Cremallera de desplazamiento, (c) Cremalleras, (d) Matriz, (e) Husillos, (f) Acoples.

Todos estos elementos serán fabricados mediante impresión 3D en material PLA. En la Tabla 4.3, se aprecia la cantidad de elementos a emplear en el sistema de respuesta.

Tabla 4.3 Cantidad de elementos de la propuesta número uno.

N.º	Nombre de pieza	Cantidad
1	Matriz	3
2	Cremallera	6
3	Husillo	18
4	Soporte de mecanismo	6
5	Piñón	6

4.3.2. *Propuesta de solución número dos para el sistema de respuesta*

En la Fig. 4.7 se aprecia la alternativa dos donde se implementa un mecanismo semejante el tornillo sin fin. El servomotor transmite su movimiento a través del acople (4), el cual proporciona movilidad al tornillo (3), generando la elevación de los cilindros (1) dentro de la guía (2), permitiendo reconocer el relieve de los diferentes caracteres numéricos brailles, este mecanismo tiene un peso de 17 gramos.

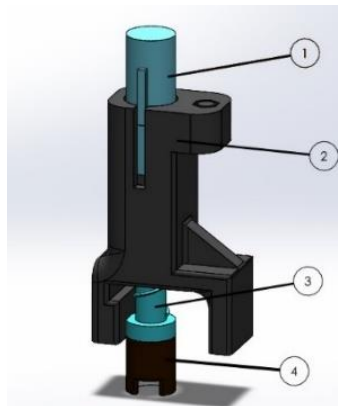


Fig. 4.7 Ensamble propuesta número dos.

En la Fig. 4.8 (a), se muestra el acople que cuenta con un diámetro inferior de 3.65 mm que se acopla perfectamente con el servomotor; en la parte superior cuenta con una perforación en forma de hexágono en donde se acopla el tornillo (ver Fig. 4.8 (b)), el cual cuenta con un largo de 23mm y un diámetro de 7,5mm. El tornillo se inserta dentro del cilindro que se observa en la Fig. 4.8 (c), el cual tiene una perforación interna de 8,3mm y

un largo de 34mm. Todos estos elementos se acoplan dentro de la guía que se aprecia en la Fig. 4.8 (d), la cual esta con las perforaciones necesarias para insertar los elementos y tiene una altura de 60mm.

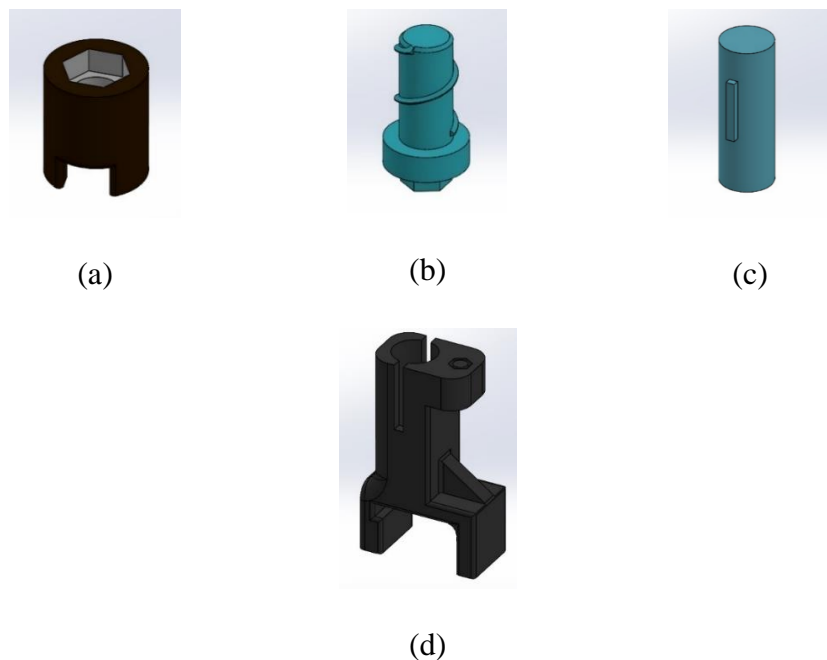


Fig. 4.8 Piezas de la propuesta dos. (a) Acople para el servomotor, (b)Tornillo, (c) Cilindro, (d) Guía del mecanismo.

Estos elementos se fabrican en impresión 3D, en la Tabla 4.4, se indica la cantidad de piezas que se emplea en el sistema de respuesta.

Tabla 4.4 Cantidad de elementos de la propuesta número dos.

N.º	Nombre de pieza	Cantidad
1	Cilindro	12
2	Guía	12
3	Tornillo	12
4	Acople	12

4.4. Estudio comparativo entre propuesta 1 y propuesta 2

El objetivo de este estudio es analizar y comparar dos posibles soluciones mecánicas para su implementación en el sistema de respuesta, ambas propuestas ampliamente utilizadas en aplicaciones de transmisión de movimiento lineal y rotatorio.

Este análisis se llevará a cabo considerando varios factores técnicos, económicos y de rendimiento que resultan críticos para el éxito del proyecto. Entre los aspectos clave a evaluar se incluyen la eficiencia mecánica, precisión, capacidad de carga, velocidad de operación, durabilidad, costo de implementación y espacio requerido. Estos criterios han sido seleccionados con el fin de identificar cuál de las soluciones ofrece el mayor grado de eficiencia y adecuación para las necesidades específicas del dispositivo.

A través de esta comparación se pretende determinar la solución que permita maximizar el rendimiento del sistema, minimizando fallos y optimizando la operación del dispositivo en términos de precisión y control del movimiento. Asimismo, se valorará el costo-beneficio de cada alternativa, teniendo en cuenta tanto la inversión inicial como el mantenimiento a largo plazo, para garantizar que el sistema sea económicamente viable sin comprometer su eficacia técnica.

1. Eficiencia mecánica

- **Tornillo sin fin:** Este mecanismo ofrece un autobloqueo, es decir que no se puede revertir el movimiento una vez que en tornillo este elevado, a menos que se aplique fuerza directa al tornillo. Esto es útil para mecanismos que requieren alta seguridad y precisión en la posición final.

- ***Piñón y cremallera:*** Conocido por ser un mecanismo de transmisión lineal de alta eficiencia, con una conversión casi directa de movimiento rotatorio en movimiento lineal. La eficiencia mecánica es mayor en comparación con el tornillo sin fin, pero el sistema puede ser reversible, lo que puede requerir frenos adicionales para mantener la posición.

2. Precisión y control

- ***Tornillo sin fin:*** Ofrece gran precisión, ya que su relación de reducción de velocidad es alta, lo que permite movimientos pequeños y controlados. Es ideal para aplicaciones que requieren movimientos finos o precisos.
- ***Piñón y cremallera:*** Aunque es eficiente en cuanto a la conversión del movimiento, su precisión puede ser limitada si se compara con el tornillo sin fin, especialmente en movimientos finos, debido a la menor relación de reducción.

3. Carga soporta

- ***Tornillo sin fin:*** Tiene una gran capacidad para soportar cargas, lo que lo convierte en la mejor opción cuando el sistema debe mover o soportar objetos pesados.
- ***Piñón y cremallera:*** Si bien también puede soportar cargas considerables, su diseño es más adecuado para movimientos rápidos y situaciones en las que la carga no es excesiva.

4. Velocidad de operación

- ***Tornillo sin fin:*** Generalmente ofrece movimientos más lentos, lo que es ventajoso para aplicaciones donde se requiere precisión en el

posicionamiento, pero no es la mejor opción si el sistema necesita rapidez. En el dispositivo cualquiera de los caracteres Braille requeridos para plasmar la respuesta se posicionan en 2 segundos.

- ***Piñón y cremallera:*** Permite alcanzar velocidades más altas, lo que lo hace adecuado para aplicaciones donde se requiere rapidez en el movimiento lineal. En este dispositivo para plasmar el resultado, tomando en cuenta que a lo largo de la cremallera sus diferentes posiciones elevan los husillos, el tiempo estimado que recorre toda la cremallera es de 12 segundos.

5. Durabilidad y mantenimiento

- ***Tornillo sin fin:*** Es un mecanismo más robusto y requiere menos mantenimiento debido a su diseño cerrado, pero la fricción puede generar desgaste a largo plazo, por lo que puede necesitar lubricación periódica.
- ***Piñón y cremallera:*** Tiene un diseño más expuesto, lo que puede hacer que sea más susceptible a la suciedad o el polvo en ambientes industriales. Esto puede requerir un mantenimiento más frecuente, aunque en condiciones adecuadas su durabilidad también es alta.

6. Costo de implementación

- ***Tornillo sin fin:*** En este caso en especial debido a los materiales empleados (PLA) y su modo de fabricación (Impresión 3D), y al diseño del mecanismo su costo es de relativamente bajo; 2,22 UDS por unidad, como se emplean 12 mecanismos el precio total de implementación es de 26,64 USD.
- ***Piñón y cremallera:*** En este mecanismo se emplea el material conocido como PLA y su fabricación es impresión 3D, debido a eso se tiene un costo por

unidad de mecanismo de 11,00 USD, se requiere de 3 mecanismos por lo cual el costo de fabricación total es de 33,00 USD.

7. Espacio requerido

- **Tornillo sin fin:** Puede ser más compacto en términos de espacio ocupado por el mecanismo, lo que es útil cuando se dispone de espacio limitado. El espacio requerido para la implementación de este sistema es de 149mm de largo* 72 mm de ancho y una altura de 69mm.
- **Piñón y cremallera:** Requiere más espacio, especialmente para permitir el desplazamiento lineal completo de la cremallera. Para la implementación de este mecanismo se requiere un espacio de 142mm de largo *200mm de ancho y una altura de 148mm.

8. Aplicabilidad

- **Tornillo sin fin:** Si el proyecto requiere precisión, seguridad y un movimiento controlado, el tornillo sin fin es la opción más eficiente.
- **Piñón y cremallera:** Si el dispositivo necesita rapidez, eficiencia y una operación económica, el piñón y cremallera podría ser una mejor opción.

Si el proyecto prioriza precisión en el control del movimiento como es el caso de este trabajo, la capacidad de soportar cargas elevadas y autobloqueo para evitar que el sistema se desplace sin control, el tornillo sin fin sería la mejor opción. Sin embargo, si la velocidad de operación, costo y un sistema más simple de transmisión lineal son factores más importantes, entonces el piñón y cremallera puede ofrecer una solución más eficiente.

De esta manera es necesario transformar los criterios cualitativos en valores numéricos mediante la asignación de escalas. Este enfoque facilita la comparación directa

entre opciones, al proporcionar una medida concreta para cada criterio, según su grado de cumplimiento o desempeño.

Cada criterio será evaluado en una escala del 1 al 5, donde:

- 1 indica el nivel más bajo o desfavorable
- 5 representa el nivel más alto o favorable

Se propone una ponderación que asigna un porcentaje o valor numérico a cada criterio, reflejando un nivel de relevancia en el conjunto de factores evaluados. Así se asegura que los factores más relevantes tengan un mayor peso en la evaluación final.

El criterio cuantitativo se calcula a partir de (ver Ecuación 4.1), donde al multiplicar cada criterio con el peso porcentual correspondiente se obtiene un puntaje ponderado que permite comparar de manera objetiva las diferentes alternativas.

Ejemplo:

*Criterio Cuantitativo = peso porcentual * ponderacion de criterio*

$$\text{Criterio Cuantitativo (Función)} = 0,20 * 4$$

$$\text{Criterio Cuantitativo (Función)} = 0,80$$

Ecuación 4.1 Criterio cuantitativo [42].

Tabla 4.5 Ponderación porcentual de los criterios de selección.

Criterio	Peso porcentual	Mecanismo 1	Mecanismo 2
Función	25%	5 (1.25)	5 (1.25)
Precisión	15%	3 (0.45)	4 (0.6)
Eficiencia	15%	3 (0.45)	5 (0.75)
Estética	10%	4 (0.4)	4 (0.4)
Peso	10%	4 (0.4)	4 (0.4)
Mantenimiento	10%	3 (0.3)	4 (0.4)
Seguridad	10%	5 (0.5)	5 (0.5)
Costo	5%	3 (0.15)	4 (0.2)
Total	100%	3.75	4.55

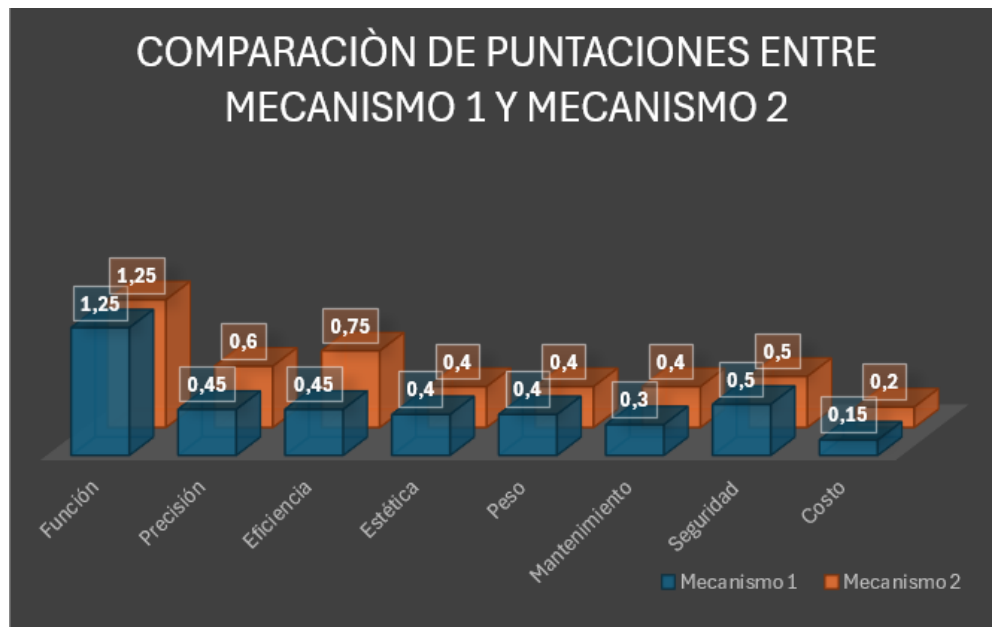


Fig. 4.9 Comparación de puntuaciones entre propuesta 1 y 2.

4.5. Mecanismo de solución seleccionada

En el desarrollo de sistemas mecánicos, la elección del mecanismo de transmisión es fundamental para garantizar la eficiencia y fiabilidad del proyecto. El tornillo sin fin (véase Fig. 4.10), destaca por su capacidad de generar una alta reducción de velocidad junto con un aumento significativo del par (torque), lo cual es esencial para aplicaciones que requieren un control preciso a bajas velocidades y con cargas considerables. Este mecanismo es ampliamente utilizado en sistemas donde es necesario un autobloqueo, es decir, evitar que el movimiento retroceda bajo carga, lo cual es ideal donde el sistema debe mantener su posición sin requerir energía adicional.

Es así como el tornillo sin fin prevalece como la opción más adecuada para este proyecto debido a su capacidad de gestionar y mantener la posición sin retroceso, lo que asegura una operación fiable en el movimiento. Por otro lado, se opta por el mecanismo de tornillo sin fin debido a su bajo costo de implementación, lo que lo convierte en una solución accesible y eficiente para este proyecto, sin comprometer la precisión y la funcionalidad requeridas. En base a todos los criterios analizados que se muestran en la Tabla 4.5, este componente se convierte en la elección ideal en base a la comparativa entre las dos propuestas (ver Fig. 4.9), para sistemas donde la precisión es fundamental.

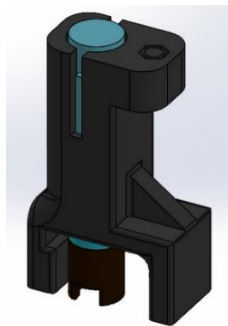


Fig. 4.10 Propuesta dos.

4.6. Sistema de alimentación

Una vez seleccionado el mecanismo, se procede a realizar una tabla de consumo eléctrico (ver Tabla 4.6), con el fin de evaluar las necesidades del sistema. Esta tabla permite determinar los requerimientos de potencia de los distintos componentes, lo que facilita la elección adecuada de la fuente de alimentación y del regulador de voltaje, asegurando un funcionamiento eficiente y estable del dispositivo.

Tabla 4.6 Tabla de consumo eléctrico.

Componente	Voltaje	Corriente en	Conversión a	Notas
	Operativo	5V (A)	12V (A)	
Arduino Mega x2	5V	0,2 A (c/u)	0,17 A (c/u)	Usando regulador de 12V a 5V
Botones (46 Ud.)	5V	Insignificante	Insignificante	Sin impacto en la fuente
Bluetooth HC-06	5V	0,03 A	0,025 A	Necesita regulador a 5V
Servomotores SG90 (x12)	5V	0,5 A (c/u)	0,208 A (c/u)	0,5 A (por servo a 6V) usando regulador
Encoder rotativo	5V	0,01 A	0,008 A	-
Módulo PCA9685	5V	0,01 A	0,008 A	-
Modulo STEP DOWN XL4015E1	12V	-	-	Regulador para convertir 12V a 5V

- **Alimentación y Regulador de Voltaje STEP DOWN XL4015E1**

La fuente de alimentación del sistema es manejada eficientemente por el módulo regulador de voltaje XL4015E1 que se aprecia en la Fig. 4.11 (b), permite convertir un voltaje de entrada de hasta 36V a un nivel adecuado para los distintos componentes del sistema, como los servos o el Arduino Mega. Su capacidad de entrega de corriente (hasta 5A) es clave para mantener la estabilidad de los dispositivos sin sobrecalentamientos ni interrupciones, a pesar de la demanda de corriente por parte de los servos. Su valor comercial es de 5.00 USD.

Total, máximo estimado de corriente

- Arduino Mega (x2): 0,34 A
- Botones y Bluetooth: ~0,025 A
- Servomotores SG90 (x12): 2,5 A
- Encoder rotativo y PCA9685: ~0,016 A

El consumo estimado en 12V en condiciones promedio con carga moderada en los servos es 2,88 A lo que significa que la fuente S-60-12 que se observa en la Fig. 4.11 (a), es suficiente para la alimentación de todo el dispositivo a pesar del funcionamiento de todos los servos simultáneamente. Su valor comercial es de 15,00 USD.



(a)



(b)

Fig. 4.11 Sistema de alimentación. (a) S-60-12 [43], (b) Step down xl4015e1 [44].

4.7. Diseño de propuesta seleccionada

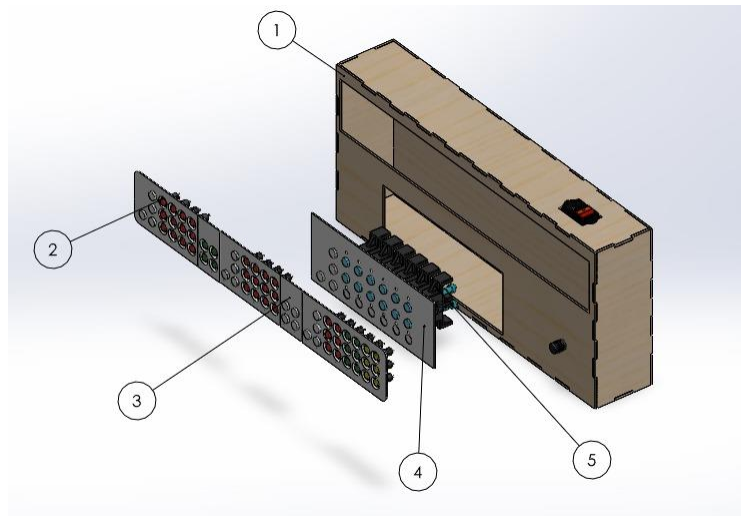


Fig. 4.12 Diseño de la propuesta seleccionada en vista explosionada.

Se selecciono la propuesta 2 para el mecanismo de respuesta debido a que cumple con los criterios establecidos para este proyecto como se muestra en la Tabla 4.5, la cual cuenta con un sistema de botoneras (3), donde su elemento principal para la representación del lenguaje Braille son botones de enclavamiento (2). Consiguiente al almacenamiento de datos y análisis correspondientes ejecutados por dos microcontroladores (Arduino Mega) el sistema de respuesta (4), plasma la respuesta correcta en el caso de ser necesaria la corrección con ayuda de 12 servomotores y su modulo controlador, este relieve se ejecuta mediante el accionamiento del mecanismo tronillo sin fin (5) adicionalmente se cuenta con diversos complementos como son: Módulo bluetooth, regulador de voltaje y módulo encoder, los cuales son alimentados a través de la fuente de voltaje. Todo esto se encuentra protegido mediante una carcasa en MDF (1). En la Fig. 4.12 se muestra un esquema de la solución propuesta seleccionada.

4.7.1. Sistema Botonera

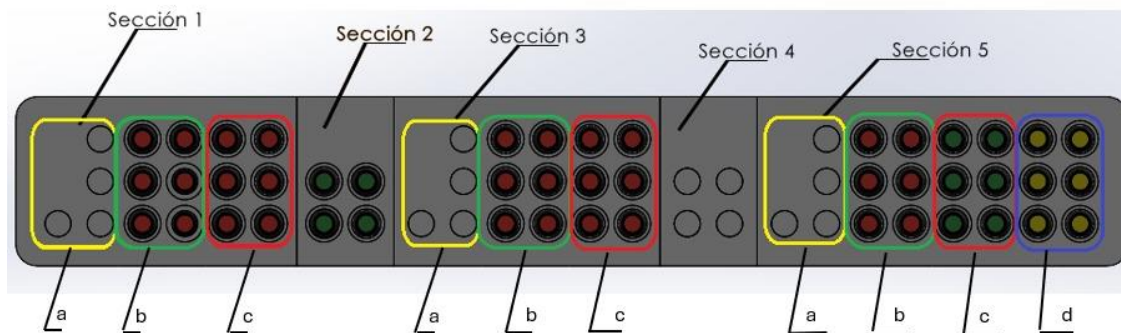


Fig. 4.13 Sistema de botoneras.

Sección 1 y 3: Están conformados por tres celdas Braille (3x2), de las cuales la primera celda (a) es plasmada por impresión 3D representando el signo numeral Braille; las dos siguientes se encuentran estructuradas por seis botones individualmente, estas permitirán la entrada del número (c) representa la unidad y (b) representa la decena.

Sección 2: Está conformada por una celda 2x2 en la cual se incorporan cuatro botones, lo que permitirá la entrada del signo que represente a la operación matemática.

Sección 4: Se encuentra una impresión plasmada en 3D de una celda de 2x2 que representa el signo igual en lenguaje Braille. Por último.

Sección 5: Formada por cuatro celdas Braille, donde (a) se encuentra plasmada en impresa 3D formando el signo numeral Braille, las tres celdas restantes constituidas de seis botones cada una sirven para la entrada de la respuesta de la operación seleccionada donde (d) representa la unidad (c) la decena y (b) la centena respectivamente.

En la Fig. 4.13 se muestra el sistema de botoneras en donde los caracteres Braille plasmados en 3D junto con los botones cuentan con un relieve de 4 mm sobre la base, lo cual facilita su identificación táctil. Para obtener un mayor nivel de detalle sobre las dimensiones

y medidas específicas de cada sección, se pueden consultar los planos de construcción disponibles en el Anexo D.

4.7.2. Sistema Corrección de Respuesta

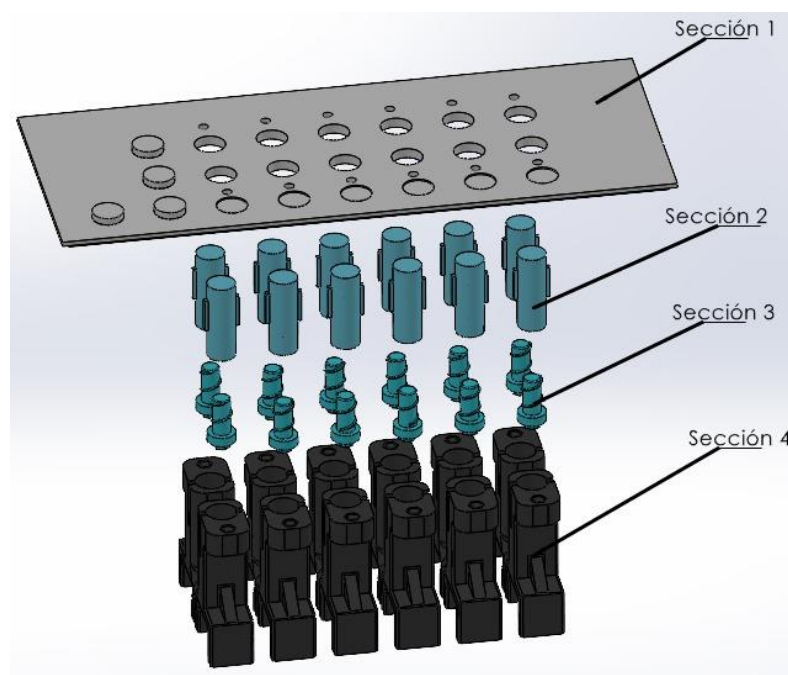


Fig. 4.14 Sistema de corrección de respuesta.

Sección 1: Esta sección está conformada por una impresión plasmada en 3D de 257,57 mm * 102,51 mm, en la cual se evidencia el signo numeral Braille con un relieve de 4 mm, su función en el sistema es recubrir los elementos internos y trabajar como base de la cual se denotará el relieve de los caracteres Braille.

Sección 2: Está conformada por doce cilindros realizados en impresión 3D con los cuales se representan los caracteres Braille, cuentan con una altura de 34 mm, un diámetro externo de 12 mm, en la parte exterior cuenta con un par de aristas rectangulares de 1,58 mm que sirven para encajar en la guía, el diámetro interno es de 8,30 mm donde a lo largo de 27 mm se presentan los canales donde encajaran los tornillos.

Sección 3: Esta sección está conformada por dos elementos diseñados para acoplarse en uno con el otro (Elemento 2-Elemento 3), los cuales son movilizados por medio del servomotor (Elemento 1) como se evidencia en la Fig. 4.15.

El elemento tres se encuentra diseñado con un diámetro de 8,75 mm, un paso de rosca (M8*1.25mm) y una longitud de 16mm.

El paso de la rosca se calcula a partir de (ver Ecuación 4.2)

- Fórmula para calcular el paso de rosca

Paso = Longitud de la rosca / número de rosca

$$Paso = \frac{16}{2} = 8mm$$

Ecuación 4.2 Paso de la Rosca [45].

- Con el motor en 180 grados la elevación del cilindro sobre la base es de 4 mm para lo cual el número de vueltas necesarias a lo largo del tornillo son 2.

Adicionalmente en la parte inferior se observa la presencia de un hexágono de 3,58 mm por lado que servirá para encajar en el elemento 2 en cual cuenta con un diámetro de 12 mm y una altura de 13,80 mm en cual se une directamente al servomotor.

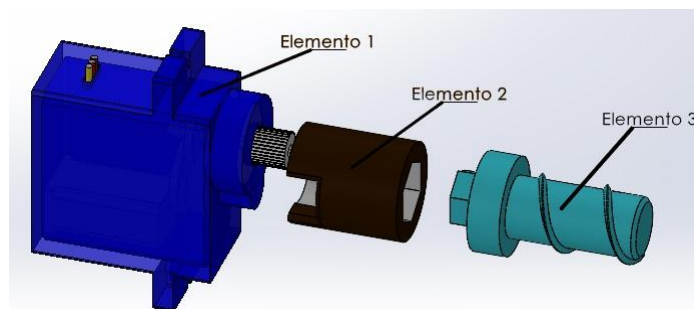


Fig. 4.15 Sección 3.

Sección 4: En esta sección se encuentra plasmadas en 3D las guías, las cuales sirven de soporte al resto de elementos que conforman el sistema de respuesta, en estas piezas se empotran los servomotores mediante tornillos. Cuenta con una altura de 60 mm, para mejorar la interpretación de todas las secciones mencionadas se observa la Fig. 4.14.

Para obtener un mayor nivel de detalle sobre las dimensiones y medidas específicas de cada sección, se pueden consultar los planos de construcción disponibles en el Anexo D.

4.7.2.1. Estudio tornillo sin fin

- *Prueba Experimental*



(a)



(b)



(c)



(e)



(f)



(g)

Fig. 4.16 Prueba experimental de tornillo sin fin.

Se realizó un análisis experimental para evaluar la capacidad de carga del tornillo fabricado en material PLA. Este análisis consistió en la aplicación progresiva de pesas sobre el tornillo como se muestra en la Fig. 4.16 (a)-(f), incrementando la carga de manera controlada hasta observar el punto de ruptura de la pieza. Los resultados del estudio mostraron que el peso máximo soportado por el tornillo antes de su falla estructural es de 130 libras (ver Fig. 4.16 (g)). Este valor representa el límite de carga máxima del diseño y permite validar su comportamiento en condiciones extremas, confirmando su resistencia dentro de las especificaciones materiales del PLA.

4.7.2.2. Transmisión de movimiento rotacional a lineal

La transformación de movimiento rotacional a lineal se realiza mediante el uso de un tornillo sin fin, o tornillo de avance, que convierte la rotación en desplazamiento lineal. A continuación, se explicó cómo funciona este mecanismo en el contexto del sistema de respuesta, donde el objetivo es permitir que el usuario palpe relieves.

- **Movimiento rotacional del motor:** En el sistema el servomotor proporciona el movimiento rotacional de 0 a 180 grados el cual debe transformarse en un movimiento lineal.
- **Engranaje con el tornillo sin fin:** El eje del motor está acoplado a un tornillo sin fin mediante el elemento 2. Este tornillo sin fin está diseñado para enroscarse en una pieza móvil que actúa como una tuerca (Sección 2). A medida que el tornillo gira, el cilindro es impulsado en dirección longitudinal a lo largo del eje del tornillo.
- **Transformación a movimiento lineal:** Cuando el tornillo sin fin rota, la fricción y el diseño de las roscas hacen que el cilindro no gire, sino que se desplace linealmente a lo largo del tornillo. Este desplazamiento lineal se utiliza para elevar o bajar el cilindro el cual se encarga de plasmar los relieves.
- **Retroalimentación táctil para el usuario:** Los cilindros que plasman los diferentes caracteres Braille se eleva o desciende de acuerdo con la información de entrada procesada en el sistema. Esto permite que el usuario pueda palpar y sentir el relieve. La precisión del tornillo sin fin permite un control fino de la posición sobre la base, asegurando que los relieves se alineen con las posiciones correctas para que el usuario pueda tocarlos y percibir el resultado de la operación.

4.8. Diagrama eléctrico

Se presenta el plano eléctrico del dispositivo en el Anexo A, el cual muestra todas las conexiones entre los componentes. Este diagrama incluye etiquetas detalladas que corresponden a las identificaciones de los cables en el dispositivo físico, facilitando así la

comprensión de la distribución y función de cada conexión. Las etiquetas permiten una correlación directa entre el diagrama y el ensamblaje real, asegurando que cada cable esté correctamente posicionado según el diseño del sistema.

4.9. Algoritmo de programación

A continuación, se presenta el código de programación del dispositivo, organizado en dos secciones principales. La primera sección corresponde al sistema de ingreso y almacenamiento de datos, donde se describe el proceso mediante el cual el dispositivo recibe, interpreta y almacena la información ingresada. La segunda sección detalla el funcionamiento del sistema de respuesta, que explica cómo el dispositivo procesa los datos almacenados para generar las respuestas correspondientes. Esta estructura permite una comprensión clara y ordenada del funcionamiento completo del dispositivo.

4.9.1. Ingreso de datos

En el siguiente diagrama de flujo se muestra el funcionamiento del sistema de ingreso de datos, acompañado de las principales líneas de código y su respectiva función en el proceso. Este diagrama proporciona una visión general del flujo de operaciones y la lógica que guía el sistema, permitiendo comprender el proceso de ingreso y procesamiento de datos. En el anexo B se incluye el código completo para este sistema, proporcionando detalles adicionales y permitiendo una revisión exhaustiva de cada segmento del programa.

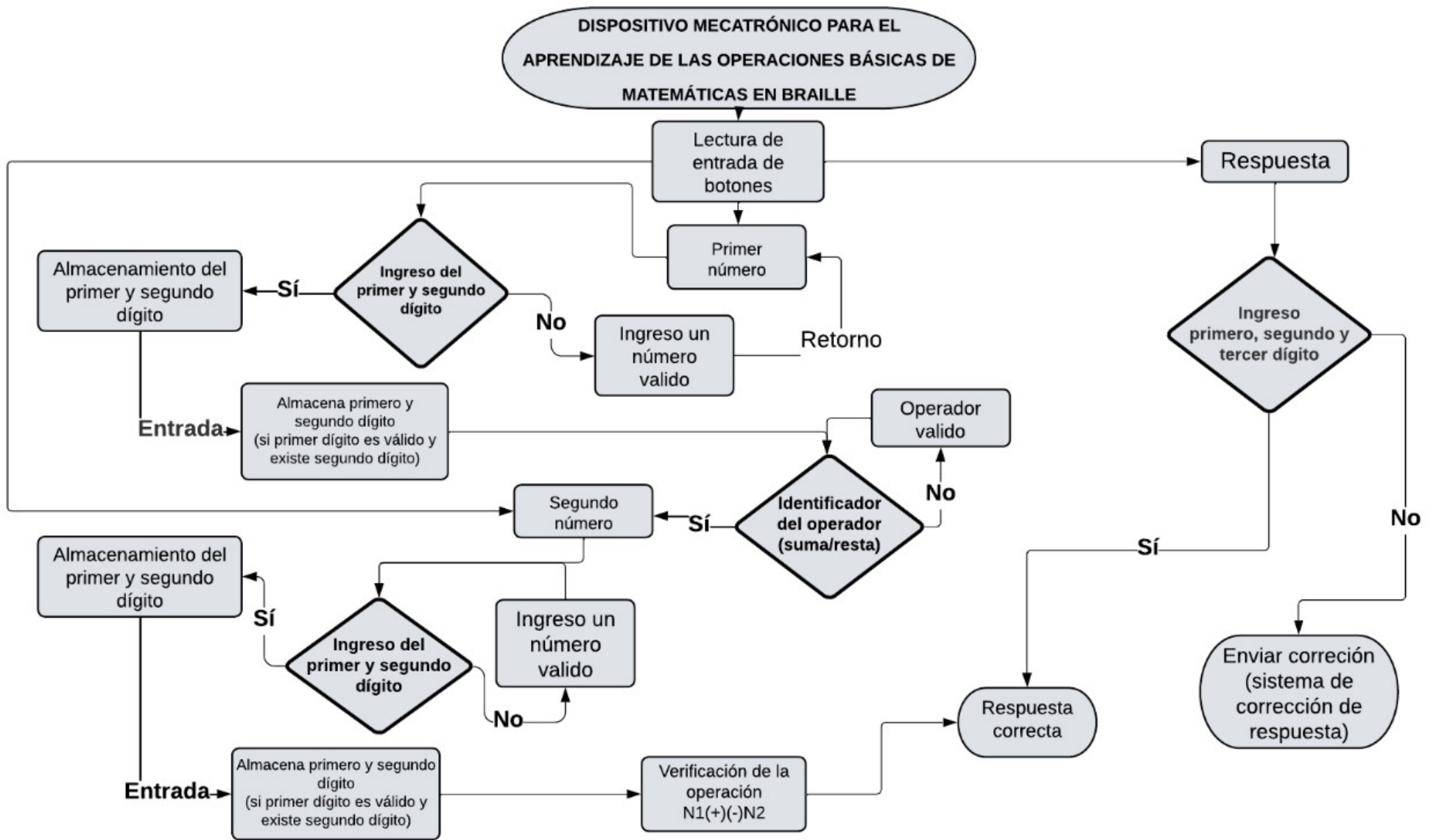


Fig. 4.17 Diagrama de Flujo de entrada de datos.

Entrada de información de datos numéricos brailles: Representa un dígito en un sistema de seis posiciones, cada número del 0 al 9 tiene una configuración específica para que no exista confusión.

```
int num0[6] = {0,1,1,1,0,0};
int num1[6] = {1,0,0,0,0,0};
int num2[6] = {1,0,1,0,0,0};
int num3[6] = {1,1,0,0,0,0};
int num4[6] = {1,1,0,1,0,0};
int num5[6] = {1,0,0,1,0,0};
int num6[6] = {1,1,1,0,0,0};
int num7[6] = {1,1,1,1,0,0};
int num8[6] = {1,0,1,1,0,0};
int num9[6] = {0,1,1,0,0,0};
```

Ingreso de las operaciones aritméticas básicas en lenguaje Braille: En este caso la suma o la resta, representada en un tamaño de 4.

```
int vsuma[4]={1,1,1,0};
int vresta[4]={0,0,1,1};
```

Los siguientes datos no se encuentran con valores iniciales declarados, esto ayuda al almacenamiento de valores generados a lo largo de la programación.

- *dig1A, dig1B:* sugiere la configuración de los dos dígitos del primer número.
- *ope:* Guarda la configuración de las diferentes operaciones ya sea la suma o resta.
- *dig2A, dig2B:* Almacena la configuración de los dos dígitos del segundo número.
- *res1, res2, res3:* Por último, se obtiene la configuración de almacenamiento del resultado que se ingresara.

Configuración de números y operaciones: El siguiente código inicializa y define varias configuraciones de números y operaciones en un sistema binario de 6 bits para números y 4 bits para operaciones. También define varios *Arrays* y variables para almacenar

y manejar operandos, operaciones, resultados, y estados de verificación, que se usan en una aplicación relacionada con cálculos o procesamiento de datos binarios.

```
int dig1A[6];
int dig1B[6];
int ope[4];
int dig2A[6];
int dig2B[6];
int res1[6];
int res2[6];
int res3[6];
```

Inicia la comunicación serial a una velocidad de 9600 baudios, permitiendo que el microcontrolador se comunique con una computadora u otros dispositivos a través del puerto serial. Es útil para depurar el programa y monitorear los datos que se están procesando.

- *for (i=2; i<14; i++) { pinMode(i, INPUT_PULLUP); } Configura los pines digitales del 2 al 13 como entradas con una resistencia interna pull-up.*
- *for (c=22; c<70; c++) { pinMode(c, INPUT_PULLUP); } Configura los pines digitales del 22 al 69 como entradas con resistencia pull-up.*

Detección del estado del botón: Esto permite que los pines detecten si un botón está presionado (*LOW*) o no (*HIGH*) sin necesidad de resistencias externas. Inicia la comunicación serial para monitorear y depurar el funcionamiento del dispositivo.

```
void setup() {
  Serial.begin(9600);
  for (i=2;i<14;i++){
    pinMode(i,INPUT_PULLUP);
  }
  for (c=22;c<70;c++){
    pinMode(c,INPUT_PULLUP);
  }
}
```

Lectura de los botones: Se utiliza múltiples bucles y estructuras para leer el estado de diferentes botones conectados a los pines del microcontrolador, verificando si cada botón está presionado (*HIGH*) o no (*LOW*) y almacena el estado en arrays específicos como *dig1A*, *dig1B*, *dig2A*, *dig2B*, *ope*, *res1*, *res2*, y *res3*. Estos arrays representan los dígitos y operaciones ingresadas por el usuario a través de los botones, para luego procesar cálculos o generar salidas Braille.

```

void loop() {
///////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////////

for (i=8,des=0;i<14;i++,des++){
  if(digitalRead(i)==HIGH){
    dig1A[des]=1;
  } else {
    dig1A[des]=0;
  }
}

for (i=2,des=0;i<8;i++,des++){
  if(digitalRead(i)==HIGH){
    dig1B[des]=1;
  } else {
    dig1B[des]=0;
  }
}

if(digitalRead(62)==HIGH){
  dig2A[0]=1;
} else {
  dig2A[0]=0;
}

if(digitalRead(63)==HIGH){
  dig2A[1]=1;
} else {
  dig2A[1]=0;
}

```

Verificación de dig1A y dig1B: Se compara el estado almacenado en *dig1A* y *dig1B* con las configuraciones predefinidas (*num0 a num9*) para determinar qué número se ha ingresado. Identificando qué dígito representa cada conjunto de botones presionados y asigna el valor correspondiente a variables como *digito1a* y *digito1b*. Por otro lado, verifica si los dígitos están "vacíos" (todos los estados son 0) para manejar casos en los que no se ha ingresado ningún número.

```

////////////////////////////////////
////////////////////////////////////DIGITO1A////////////////////////////////////
////////////////////////////////////
if(dig1A[0]==0 && dig1A[1]==0 && dig1A[2]==0 && dig1A[3]==0 && dig1A[4]==0 && dig1A[5]==0) {
    vac1A=1;
    ver1A=0;
}

else if(dig1A[0]==num1[0] && dig1A[1]==num1[1] && dig1A[2]==num1[2] && dig1A[3]==num1[3] &&
dig1A[4]==num1[4] && dig1A[5]==num1[5]){
    ver1A=1;
    digito1a=1;
    vac1A=0;
}

else if(dig1A[0]==num2[0] && dig1A[1]==num2[1] && dig1A[2]==num2[2] && dig1A[3]==num2[3] &&
dig1A[4]==num2[4] && dig1A[5]==num2[5]){
    ver1A=1;
    digito1a=2;
    vac1A=0;
}
////////////////////////////////////
////////////////////////////////////DIGITO1B////////////////////////////////////
////////////////////////////////////

if(dig1B[0]==0 && dig1B[1]==0 && dig1B[2]==0 && dig1B[3]==0 && dig1B[4]==0 && dig1B[5]==0) {
    vac1B=1;
    ver1B=0;
}

else if(dig1B[0]==num1[0] && dig1B[1]==num1[1] && dig1B[2]==num1[2] && dig1B[3]==num1[3] &&
dig1B[4]==num1[4] && dig1B[5]==num1[5]){
    digito1b=1;
}

```

```

    ver1B=1;
    vac1B=0;
}

else if(dig1B[0]==num2[0] && dig1B[1]==num2[1] && dig1B[2]==num2[2] && dig1B[3]==num2[3] &&
dig1B[4]==num2[4] && dig1B[5]==num2[5]){
    ver1B=1;
    digito1b=2;
    vac1B=0;
}

```

Verificación de dig2A y dig2B: Similar que el *dig1A* y *dig1B*, compara el estado almacenado en *dig2A* y *dig2B* con las configuraciones predefinidas (*num0 a num9*) para determinar qué número se ha ingresado. Permite identificar múltiples números ingresados simultáneamente.

```

////////////////////////////////////
////////////////////////////////////DIGITO 2A////////////////////////////////////
////////////////////////////////////

if(dig2A[0]==0 && dig2A[1]==0 && dig2A[2]==0 && dig2A[3]==0 && dig2A[4]==0 && dig2A[5]==0) {
    vac2A=1;
    ver2A=0;
}

else if(dig2A[0]==num1[0] && dig2A[1]==num1[1] && dig2A[2]==num1[2] && dig2A[3]==num1[3] &&
dig2A[4]==num1[4] && dig2A[5]==num1[5]){
    ver2A=1;
    digito2a=1;
    vac2A=0;
}

else if(dig2A[0]==num2[0] && dig2A[1]==num2[1] && dig2A[2]==num2[2] && dig2A[3]==num2[3] &&
dig2A[4]==num2[4] && dig2A[5]==num2[5]){
    ver2A=1;
    digito2a=2;
    vac2A=0;
}

```

```

////////////////////////////////////
////////////////////////////////////DIGITO 2B////////////////////////////////////
////////////////////////////////////

if(dig2B[0]==num1[0] && dig2B[1]==num1[1] && dig2B[2]==num1[2] && dig2B[3]==num1[3] &&
dig2B[4]==num1[4] && dig2B[5]==num1[5]){
    ver2B=1;
    digito2b=1;
    vac2B=0;
}

else if(dig2B[0]==num2[0] && dig2B[1]==num2[1] && dig2B[2]==num2[2] && dig2B[3]==num2[3] &&
dig2B[4]==num2[4] && dig2B[5]==num2[5]){
    ver2B=1;
    digito2b=2;
    vac2B=0;
}

```

Verificación de Resultados (*res1*, *res2*, *res3*): Compara los estados almacenados en *res1*, *res2* y *res3* con las configuraciones predefinidas para determinar los resultados de las operaciones. Identificando el resultado de una operación matemática realizada por el usuario.

```

////////////////////////////////////
////////////////////////////////////RESULTADO RA////////////////////////////////////
////////////////////////////////////

if(res1[0]==0 && res1[1]==0 && res1[2]==0 && res1[3]==0 && res1[4]==0 && res1[5]==0) {
    verr1=0;
    vacr1=1;
}

else if(res1[0]==num1[0] && res1[1]==num1[1] && res1[2]==num1[2] && res1[3]==num1[3] &&
res1[4]==num1[4] && res1[5]==res1[5]){
    verr1=1;
    resul1=1;
    vacr1=0;
}

else if(res1[0]==num2[0] && res1[1]==num2[1] && res1[2]==num2[2] && res1[3]==num2[3] && res1[4]==num2[4]
&& res1[5]==num2[5]){
    verr1=1;
}

```

```

resul1=2;
vacr1=0;
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////RESULTADO RB/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

if(res2[0]==0 && res2[1]==0 && res2[2]==0 && res2[3]==0 && res2[4]==0 && res2[5]==0) {
verr2=0;
vacr2=1;
}

else if(res2[0]==num1[0] && res2[1]==num1[1] && res2[2]==num1[2] && res2[3]==num1[3] && res2[4]==num1[4]
&& res2[5]==num1[5]){
verr2=1;
resul2=1;
vacr2=0;
}

else if(res2[0]==num2[0] && res2[1]==num2[1] && res2[2]==num2[2] && res2[3]==num2[3] && res2[4]==num2[4]
&& res2[5]==num2[5]){
verr2=1;
resul2=2;
vacr2=0;
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////RESULTADO RC/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

if(res3[0]==0 && res3[1]==0 && res3[2]==0 && res3[3]==0 && res3[4]==0) {
verr3=0;
vacr3=1;
}

else if(res3[0]==num1[0] && res3[1]==num1[1] && res3[2]==num1[2] && res3[3]==num1[3] &&
res3[4]==num1[4]){
verr3=1;
resul3=1;
vacr3=0;
}

else if(res3[0]==num2[0] && res3[1]==num2[1] && res3[2]==num2[2] && res3[3]==num2[3] &&
res3[4]==num2[4]){
verr3=1;

```

```

resul3=2;
vacr3=0;
}

```

Verificación de Operadores (*ope*): Se compara el estado de los botones de operación con configuraciones predefinidas para identificar si el operador es *suma (+)* o *resta (-)*. Determinando qué tipo de operación matemática que se utilizará en la operación. Dependiendo de la comparación de *ope* con *vsuma* o *vresta*, se asigna el carácter correspondiente ('+', '-', o 'N' para ninguno) a *finop [0]*.

```

////////////////////////////////////
////////////////////////////////////OPERADOR////////////////////////////////////
////////////////////////////////////

if(ope[0]==vsuma[0] && ope[1]==vsuma[1] && ope[2]==vsuma[2] && ope[3]==vsuma[3]){
  finop[0]='+';
}

else if(ope[0]==vresta[0] && ope[1]==vresta[1] && ope[2]==vresta[2] && ope[3]==vresta[3]){
  finop[0]='-';
}

else {
  finop[0]='N';
}

```

Lee el primer carácter disponible y comprueba si es 'X', esto actúa como una señal de inicio para procesar los datos y realizar los cálculos.

Por otro lado, verifica si los dos dígitos (*ver1A* y *ver1B*) son válidos y calcula el primer número (*ing1*) combinando *digito1a* y *digito1b*. Permitiendo ingresar un número de dos dígitos. Por ejemplo, si *digito1a* es 3 y *digito1b* es 7, entonces *ing1* será 37. Si solo se ingresó un dígito, se toma *digito1a* como el valor de *ing1*, permitiendo ingresar un número de un solo dígito. Si ninguna de las condiciones anteriores es válida, imprime 'N' para indicar que no se

ha ingresado un número válido, informando que la entrada es inválida o que falta información.

```
////////////////////////////////////  
////////////////////////////////////OPERACIONES////////////////////////////////////  
////////////////////////////////////  
  
if (Serial.available()){  
  
    if (Serial.read()=='X'){  
  
        if (ver1B==1 && ver1A==1)  
        {  
            ing1=(digito1a*10)+digito1b;  
            Serial.print(ing1);  
        }  
        else if (vac1B==1 && ver1A==1){  
            ing1=digito1a;  
            Serial.print(ing1);  
        }  
        else if (vac1A==1){  
            ver1B=0;  
            Serial.print('N');  
        }  
        else if (ver1A==0){  
            ver1B=0;  
            Serial.print('N');  
        }  
        else {  
            Serial.print('N');  
        }  
    }  
}
```

El operador \cdot representa la operación matemática seleccionada (*por ejemplo, + o -*), y las comas sirven como separadores en la salida serial.

```
Serial.print(',');  
Serial.print(finop[0]);  
Serial.print(',');
```

Similar a *ing1*, verifica si ambos dígitos (*ver2A* y *ver2B*) son válidos y junta *digito2a* y *digito2b* para formar un número de dos dígitos (*ing2*). Permitiendo ingresar otro número de dos dígitos para usar en la operación. Si solo se ingresó un dígito, se toma *digito2a* como el valor de *ing2*. Si no hay un valor válido, imprime 'N'. Indica que no se ha ingresado un número válido en el segundo valor.

```
    if (ver2B==1 && ver2A==1)
    {
    ing2=(digito2a*10)+digito2b;
    Serial.print(ing2);
    }
    else if (ver2A==1 && vac2B==1){
    ing2=digito2a;
    Serial.print(ing2);
    }
    else if (vac2A==1){
    ver2B=0;
    Serial.print('N');
    }
    else if (ver2A==0){
    ver2B=0;
    Serial.print('N');
    }
```

Se verifica si los tres dígitos (*verr3*, *verr2*, *verr1*) son válidos y calcula el resultado (*sol*) como un número de tres dígitos, Permite mostrar un resultado de hasta tres dígitos. Si solo hay dos dígitos válidos, calcula *sol* como un número de dos dígitos. Si solo hay un dígito válido, imprime *sol* como un solo dígito. Si no hay un valor válido, imprime 'N', indicando que el resultado es inválido o falta información.

Este código representa los cálculos básicos, donde el usuario ingresa valores y un operador, el sistema muestra el resultado de manera clara y organizada en la salida serial.

```

if (verr3==1 && verr2==1 && verr1==1)
{
sol=(resul1*100)+(resul2*10)+resul3;
Serial.println(sol);
}
else if (vacr3==1 && verr2==1 && verr1==1){
sol=(resul1*10)+resul2;
Serial.println(sol);
}
else if (vacr3==1 && vacr2==1 && verr1==1){
sol=resul1;
Serial.println(sol);
}
else{
Serial.println('N');
}
}
}
}
}
}

```

4.9.2. Corrección de respuesta

En el siguiente diagrama de flujo se presenta el funcionamiento del sistema de respuesta, junto con las principales líneas de código y sus respectivas funciones. Este diagrama permite visualizar la secuencia lógica y las operaciones que lleva a cabo el sistema para generar respuestas, proporcionando una comprensión clara de su comportamiento. En el anexo C se incluye el código completo de este sistema, permitiendo una revisión detallada de cada función implementada.

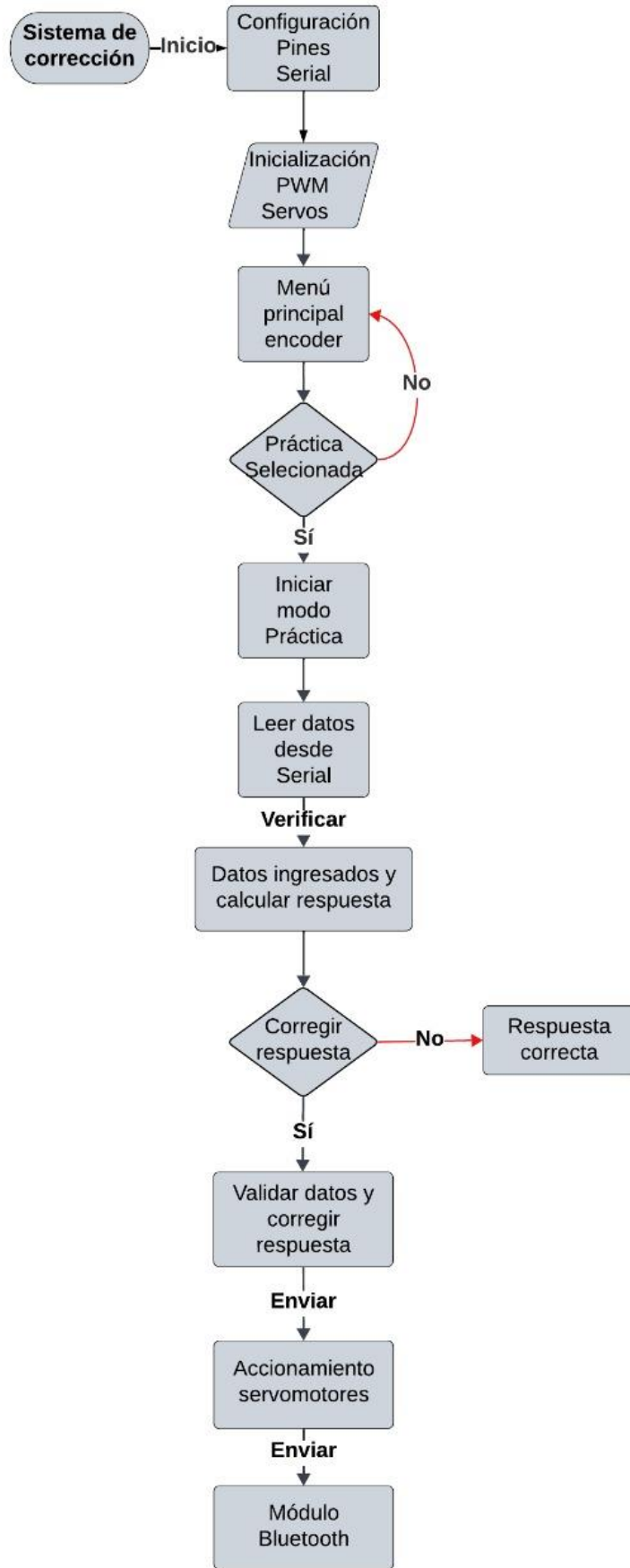


Fig. 4.18 Diagrama de flujo de control operacional.

Incorporación de Librerías y Estructuración de Pines: Se importan las bibliotecas necesarias para el funcionamiento del controlador *PWM*, el encoder rotativo y la comunicación con los módulos seriales y Bluetooth. Además, se definen los pines asociados al encoder rotativo (*CLK*, *DT*, y *SW*).

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <Encoder.h>

// Definir los pines del encoder rotativo
#define CLK 2
#define DT 3
#define SW 4
```

Inicialización de Variables Globales y Objetos: Se inicializan los objetos *myEnc* y *pwm* para el manejo del encoder y los servos, respectivamente. Así mismo se configuran las variables para el menú principal, el submenú de práctica y las variables para los datos de práctica y resultados esperados. *SERVOMIN* y *SERVOMAX* definen los límites de los servos.

```
// Inicializar el encoder
Encoder myEnc(CLK, DT);

// Crear una instancia del controlador PWM
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

#define SERVOMIN 600 // Valor mínimo del pulso del servo
#define SERVOMAX 150 // Valor máximo del pulso del servo

// Variables del menú
int menuIndex = 0;
String menuItems[] = { "Practica", "Entretenimiento" };
const int numMenuItems = 2;
bool buttonPressed = false;
int previousMenuIndex = -1; // Para rastrear cambios en el menú

// Variables para los datos recibidos
String cifra1, signo, cifra2, respuesta;
```

```

int expectedRes; // Variable para la respuesta correcta

// Variables para el submenú de práctica
int practicaMenuIndex = 0;
String practicaMenuItems[] = { "Validar", "Salir" };
const int numPracticaMenuItems = 2;

int digits[3] = { 0, 0, 0 };

```

Configuración Inicial: En esta sección se configuran los pines, los puertos seriales, y el controlador *PWM*. Además, se establece una frecuencia de *60 Hz* para los servos, y se imprime un mensaje de inicialización del menú. La función *encero ()* limpia las posiciones iniciales de los servos.

```

void setup() {
  pinMode(SW, INPUT_PULLUP); // Configurar el botón del encoder como entrada con resistencia pull-up
  Serial.begin(9600);        // Serial para depuración
  Serial1.begin(9600);       // Serial1 para comunicación con el otro Arduino
  Serial2.begin(9600);       // Serial2 para el módulo Bluetooth

  pwm.begin();
  pwm.setPWMFreq(60); // Frecuencia de 60 Hz para los servos

  Serial.println("Menu Inicializado");
  displayMenu();
  encero();
}

```

Bucle Principal: Este bucle principal detecta los cambios en la posición del encoder para navegar por el menú y selecciona el elemento correspondiente cuando se presiona el botón. Envía el mensaje del menú actual al módulo Bluetooth y evita múltiples lecturas erráticas con un retardo.

```

void loop() {
  // Leer el valor del encoder
  long newPosition = myEnc.read();
  static long oldPosition = -999;

  if (newPosition != oldPosition) {
    oldPosition = newPosition;
    // Calcular el índice del menú en función de la posición del encoder
    menuIndex = (newPosition / 4) % numMenuItems; // Ajusta el divisor según la sensibilidad que desees
    if (menuIndex < 0) {
      menuIndex += numMenuItems;
    }
    if (menuIndex != previousMenuIndex) {
      displayMenu();
      sendMenuAudio(); // Enviar el mensaje correspondiente al módulo Bluetooth
      previousMenuIndex = menuIndex;
    }
  }

  // Leer el estado del botón del encoder
  if (digitalRead(SW) == LOW) {
    if (!buttonPressed) {
      buttonPressed = true;
      selectMenuItem();
    }
  } else {
    buttonPressed = false;
  }

  delay(100); // Pequeño retardo para evitar lecturas erráticas
}

```

Funciones de Visualización y Envío de Menú: *displayMenu ()* muestra el menú actual en el monitor serial, mientras que *sendMenuAudio ()* envía el elemento del menú seleccionado al módulo Bluetooth

```

void displayMenu() {
  Serial.println("----- MENU -----");
  for (int i = 0; i < numMenuItems; i++) {
    if (i == menuIndex) {
      Serial.print("> ");
    } else {

```

```

    Serial.print(" ");
  }
  Serial.println(menuItems[i]);
}
}

void sendMenuAudio() {
  String message = menuItems[menuIndex];
  Serial2.println(message);
  Serial.print("Enviando por Bluetooth: ");
  Serial.println(message);
}

```

Recepción y Análisis de Datos: Recibe una cadena de datos, que se espera que contenga dos números y un operador matemático. La función divide los datos en partes (*cifra1, signo, cifra2, respuesta*) y los muestra en la consola para revisión.

```

void parseData(String data) {
  int firstComma = data.indexOf(',');
  int secondComma = data.indexOf(',', firstComma + 1);
  int thirdComma = data.indexOf(',', secondComma + 1);

  cifra1 = data.substring(0, firstComma);
  signo = data.substring(firstComma + 1, secondComma);
  cifra2 = data.substring(secondComma + 1, thirdComma);
  respuesta = data.substring(thirdComma + 1);

  Serial.println("Datos recibidos:");
  Serial.println("Cifra 1: " + cifra1);
  Serial.println("Signo: " + signo);
  Serial.println("Cifra 2: " + cifra2);
  Serial.println("Respuesta: " + respuesta);

  //Mover los servos basados en los datos recibidos
  //moveServos();
}

```


Validación de Respuestas: Calcula la respuesta esperada usando el operador (+ o -) y compara la respuesta recibida. Si la respuesta es incorrecta, mueve los servos para mostrar el valor correcto en Braille y luego restablece los valores.

```
void validateData() {
  bool error = false;

  int num1 = cifra1.toInt();
  int num2 = cifra2.toInt();
  int res = respuesta.toInt();
  expectedRes = (signo == "+") ? (num1 + num2) : (num1 - num2);

  if (res != expectedRes) {
    error = true;
  }

  if (error) {
    // Mover los servos para mostrar la respuesta correcta en Braille
    moveServosToCorrectAnswer();
    delay(15000); // Esperar 7 segundos
    encero(); // Limpiar la respuesta
  }
}
```

Control de Servos:

- *moveServos ():* Mueve los servos según los valores de las cifras y la respuesta recibida.
- *moveServosToCorrectAnswer ():* Mueve los servos para representar la respuesta correcta en Braille si hay un error.

```
void moveServos() {
  // Ejemplo simple: mover servos basados en los valores de cifra1, cifra2 y respuesta
  // Aquí puedes agregar la lógica específica para mover tus servos

  int servoPos1 = map(cifra1.toInt(), 0, 9, SERVOMIN, SERVOMAX); // Asumiendo cifra1 es un dígito del 0 al 9
  int servoPos2 = map(cifra2.toInt(), 0, 9, SERVOMIN, SERVOMAX); // Asumiendo cifra2 es un dígito del 0 al 9
  int servoPosRes = map(respuesta.toInt(), 0, 999, SERVOMIN, SERVOMAX); // Asumiendo respuesta es un número de
  0 a 999

  pwm.setPWM(0, 0, servoPos1); // Mover el primer servo
  pwm.setPWM(1, 0, servoPos2); // Mover el segundo servo
  pwm.setPWM(2, 0, servoPosRes); // Mover el tercer servo
}
```

```

void moveServosToCorrectAnswer() {
  // Mover los servos para mostrar la respuesta correcta en Braille
  if(expectedRes < 0)
  {
    expectedRes=expectedRes*-1;
  }
  if (expectedRes < 10) {
    digits[0] = expectedRes;
    matriz0();
  } else if (expectedRes < 100) {
    digits[0] = expectedRes / 10; // Decenas
    digits[1] = expectedRes % 10; // Unidades
    matriz0();
    matriz1();
  } else {
    digits[0] = expectedRes / 100;    // Centenas
    digits[1] = (expectedRes / 10) % 10; // Decenas
    digits[2] = expectedRes % 10;    // Unidades
    matriz0();
    matriz1();
    matriz2();
  }
}

```

Las funciones *matriz0()*, *matriz1()*, y *matriz2()* mueven los servos a posiciones específicas que representan cada dígito en Braille, lo cual facilita que el usuario vea la respuesta en ese sistema.

```

void matriz0() {
  if (digits[0] == 1) {
    pwm.setPWM(0, 0, SERVOMAX);
    pwm.setPWM(1, 0, SERVOMIN);
    pwm.setPWM(2, 0, SERVOMIN);
    pwm.setPWM(3, 0, SERVOMIN);
  } else if (digits[0] == 2) {
    pwm.setPWM(0, 0, SERVOMAX);
    pwm.setPWM(1, 0, SERVOMAX);
    pwm.setPWM(2, 0, SERVOMIN);
    pwm.setPWM(3, 0, SERVOMIN);
  } else if (digits[0] == 3) {
    pwm.setPWM(0, 0, SERVOMAX);
    pwm.setPWM(1, 0, SERVOMIN);
    pwm.setPWM(2, 0, SERVOMAX);
    pwm.setPWM(3, 0, SERVOMIN);
  } else if (digits[0] == 4) {
    pwm.setPWM(0, 0, SERVOMAX);
    pwm.setPWM(1, 0, SERVOMIN);
    pwm.setPWM(2, 0, SERVOMAX);
    pwm.setPWM(3, 0, SERVOMAX);
  }
}

void matriz1() {
  if (digits[1] == 1) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMIN);
  }
}

```

```

    pwm.setPWM(6, 0, SERVOMIN);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 2) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMIN);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 3) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 4) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMAX);

void matriz2() {
    if (digits[2] == 1) {
        pwm.setPWM(8, 0, SERVOMAX);
        pwm.setPWM(9, 0, SERVOMIN);
        pwm.setPWM(10, 0, SERVOMIN);
        pwm.setPWM(11, 0, SERVOMIN);
    } else if (digits[2] == 2) {
        pwm.setPWM(8, 0, SERVOMAX);
        pwm.setPWM(9, 0, SERVOMAX);
        pwm.setPWM(10, 0, SERVOMIN);
        pwm.setPWM(11, 0, SERVOMIN);
    } else if (digits[2] == 3) {
        pwm.setPWM(8, 0, SERVOMAX);
        pwm.setPWM(9, 0, SERVOMIN);
        pwm.setPWM(10, 0, SERVOMAX);
        pwm.setPWM(11, 0, SERVOMIN);
    } else if (digits[2] == 4) {
        pwm.setPWM(8, 0, SERVOMAX);
        pwm.setPWM(9, 0, SERVOMIN);
        pwm.setPWM(10, 0, SERVOMAX);
        pwm.setPWM(11, 0, SERVOMAX);
    }
}

```

Función de Reinicio: La función *encero ()* ajusta todos los servos a su posición mínima como una forma de restablecer la configuración.

```

void encero() {
    pwm.setPWM(0, 0, SERVOMIN);
    pwm.setPWM(1, 0, SERVOMIN);
    pwm.setPWM(2, 0, SERVOMIN);
    pwm.setPWM(3, 0, SERVOMIN);
    delay(1000);
    pwm.setPWM(4, 0, SERVOMIN);
    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMIN);
    pwm.setPWM(7, 0, SERVOMIN);
    delay(1000);
    pwm.setPWM(8, 0, SERVOMIN);
    pwm.setPWM(9, 0, SERVOMIN);
    pwm.setPWM(10, 0, SERVOMIN);
    pwm.setPWM(11, 0, SERVOMIN);
    delay(1000);
}

```

4.10. Construcción y ensamble del dispositivo

Esta sección ofrece una visión integral de los recursos necesarios para la fabricación, permitiendo comprender tanto el diseño como la optimización en la producción.

4.10.1. Sistema estructural

A continuación, se explicará el proceso de construcción del sistema estructural del dispositivo, el cual está compuesto por dos elementos principales: la carcasa fabricada en MDF y la estructura interna que alberga los componentes electrónicos

4.10.1.1. Carcasa MDF

Para la fabricación de la carcasa se implementa el uso del corte láser. A continuación, se detalla el proceso de fabricación y los costos asociados al uso de MDF de 3 mm como material principal. Para el corte se utilizó una cortadora láser de CO2 de 150W, logrando un tiempo aproximado de 5 minutos de corte para las piezas (1- 6) que conforman la carcasa las cuales se pueden apreciar en la Fig. 4.19.

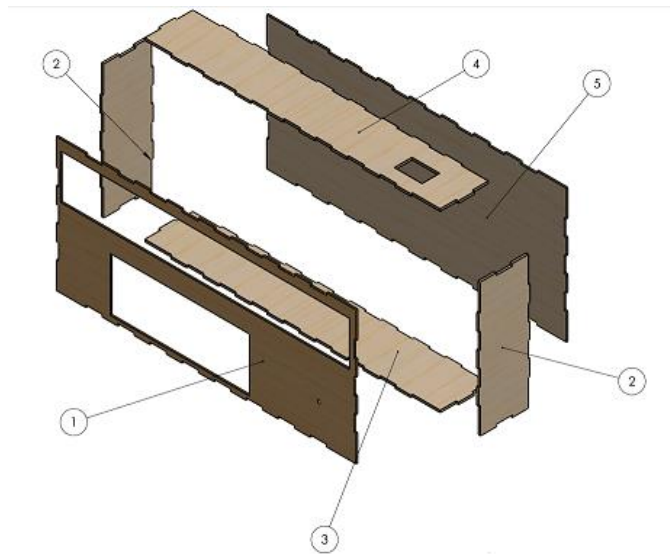


Fig. 4.19 Carcasa en MDF.

En la Tabla 4.7 se detalla el costo de corte láser de cada una de las piezas que conforman la carcasa del dispositivo, incluyendo los precios individuales y el costo total del proceso de fabricación.

- **Costo de fabricación**

Tabla 4.7 Costo de fabricación de la estructura en MDF.

Pieza	Costo por unidad (USD)	Cantidad	Costo total (USD)
Laterales	1,00	2	2,00
Inferior	3,00	1	3,00
Superior	3,00	1	3,00
Frontal	3,00	1	3,00
Trasera	4,00	1	4,00

- Dando así un costo de fabricación de 15USD en toda la estructura.

4.10.1.2. Estructura de soporte electrónico

Para la fabricación de los componentes estructurales de soporte electrónico, se utilizó una impresora 3D Artillery Sidewinder, reconocida por su precisión y eficiencia en la producción de piezas detalladas.

El material seleccionado fue PLA (Ácido Poliláctico), un termoplástico popular en la impresión 3D debido a su facilidad de uso y acabado de alta calidad. Se estableció una

densidad de relleno del 10% con un patrón de relleno tipo rejilla, lo que proporciona un equilibrio óptimo entre resistencia estructural y reducción de material utilizado. La temperatura de impresión se configuró en 220 grados Celsius, adecuada para garantizar una adhesión uniforme y minimizar deformaciones durante el proceso de extrusión.

Para asegurar la estabilidad de las impresiones, se habilitaron soportes en todos los sitios, permitiendo una correcta formación de estructuras complejas y evitando el colapso de partes sobresalientes. Además, se optó por no utilizar ningún tipo de adherencia en la placa de impresión, confiando en la adhesión mecánica del PLA a la superficie de la impresora para facilitar la remoción de las piezas una vez finalizada la impresión.

Se trabaja con una tolerancia de ± 0.3 mm, lo que permite una ligera desviación entre el modelo CAD o el dibujo técnico y la pieza final fabricada. Esta tolerancia es esencial para mantener un margen de precisión adecuado. Con esta tolerancia, se asegura que las piezas cumplan con las especificaciones del diseño sin comprometer su funcionalidad, permitiendo además que los componentes encajen correctamente en el ensamblaje final.

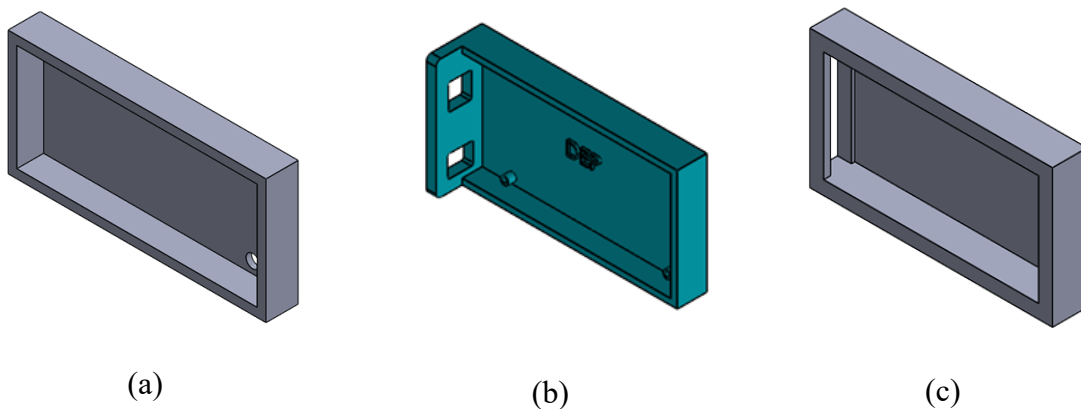




Fig. 4.20 Estructura de soporte electrónico. (a) Case regulador, (b) Case Arduino Mega, (c) Case Bluetooth, (d) Case driver, (e) Case encoder.

- **Costos de Impresión 3D**

El análisis de costos de impresión se realizó considerando tanto el material utilizado como el tiempo empleado en cada componente (ver Tabla 4.8). A continuación, se detalla el desglose de costos por impresión:

Case Arduino: Se imprimieron 2 unidades, cada una con un peso de 30 gramos y un tiempo de impresión de 3 horas y 5 minutos. El costo total por ambas unidades fue de **7,00 USD** (3,50 USD por unidad) (véase Fig. 4.20 (b)).

Case Bluetooth: Este componente fue impreso con un peso de 1 gramo y un tiempo de impresión de 9 minutos, resultando en un costo de **0,30 USD** (ver Fig. 4.20 (c)).

Case Driver: Con un peso de 5 gramos y un tiempo de impresión de 29 minutos, el costo asociado fue de **1,50 USD** (véase Fig. 4.20 (d)).

Case Encoder: Este elemento requirió 4 gramos de material y 30 minutos de impresión, con un costo de **1,30 USD** (observar Fig. 4.20 (e)).

Case Regulador: Finalmente, el regulador fue impreso utilizando 5 gramos de PLA en 27 minutos, con un costo total de **1,40 USD** (ver Fig. 4.20 (a)).

- *Resumen de Costos*

Tabla 4.8 Costos de fabricación estructura impresión 3D

Unidades	Componente	Peso (g)	Tiempo de Impresión	Costo (USD)
2	Case Arduino	30*2=60	3 h 5 minutos	3,50 *2 = 7,00
1	Case Bluetooth	1	9 minutos	0,30
1	Case Driver	5	29 minutos	1,50
1	Case Encoder	4	30 minutos	1,30
1	Case Regulador	5	27 minutos	1,40
	Total, General	75	7 h 40 minutos	11,50 USD

Este desglose permite una evaluación precisa de los costos involucrados en la fabricación de cada componente, facilitando la planificación financiera del proyecto y la optimización del uso de recursos.

4.10.2. Sistema mecánico

Para la fabricación de los componentes que se muestra en al Fig. 4.21, se ha utilizado material PLA, que ofrece buena resistencia y facilidad de impresión. La densidad de relleno empleada es del 100%, utilizando un patrón de rejilla para asegurar la solidez de las piezas. La temperatura de impresión es de 220 grados, y se ha configurado soporte en todas las áreas necesarias para asegurar la estabilidad de las impresiones. Para una mejor adhesión durante el proceso, se utiliza una adherencia tipo balsa en la placa de impresión. Además, la tolerancia

de fabricación es de $\pm 0,03$ mm, lo que garantiza precisión en la correspondencia entre el modelo CAD y las piezas impresas.

En cuanto a los costos, se muestra la Tabla 4.9 en donde se calculan los precios en función del peso del material y el tiempo de impresión para cada componente. Los detalles de cada pieza son los siguientes:

- Cilindro: 2 gramos, 37 minutos de impresión, con un costo de \$0,32 USD.
- Guía: 13 gramos, 1 hora y 49 minutos de impresión, con un costo de \$1,50 USD.
- Tornillo: 1 gramo, 21 minutos de impresión, con un costo de \$0,25 USD.
- Acople: 1 gramo, 15 minutos de impresión, con un costo de \$0,15 USD.

- ***Resumen de costos***

Tabla 4.9 Costo de fabricación sistema mecánico.

Componente	Peso (g)	Tiempo de Impresión	Costo por Unidad (USD)	Cantidad Total	Costo Total (USD)
Cilindro	2 g	37 min	0,32	12	3,84
Guía	13 g	1 hr 49 min	1,50	12	18,00
Tornillo	1 g	21 min	0,25	12	3,00
Acople	1 g	15 min	0,15	12	1,80

Total, Costo de Impresión: \$26,64 USD

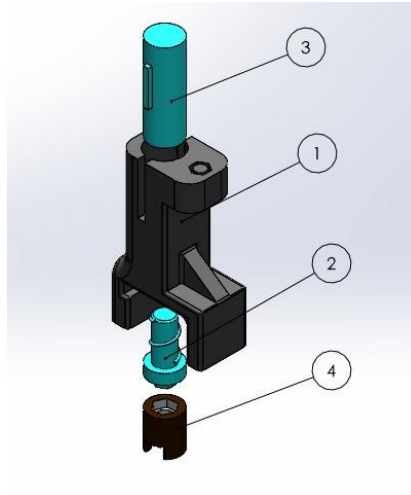


Fig. 4.21 Sistema mecánico.

4.11. Ensamble del dispositivo

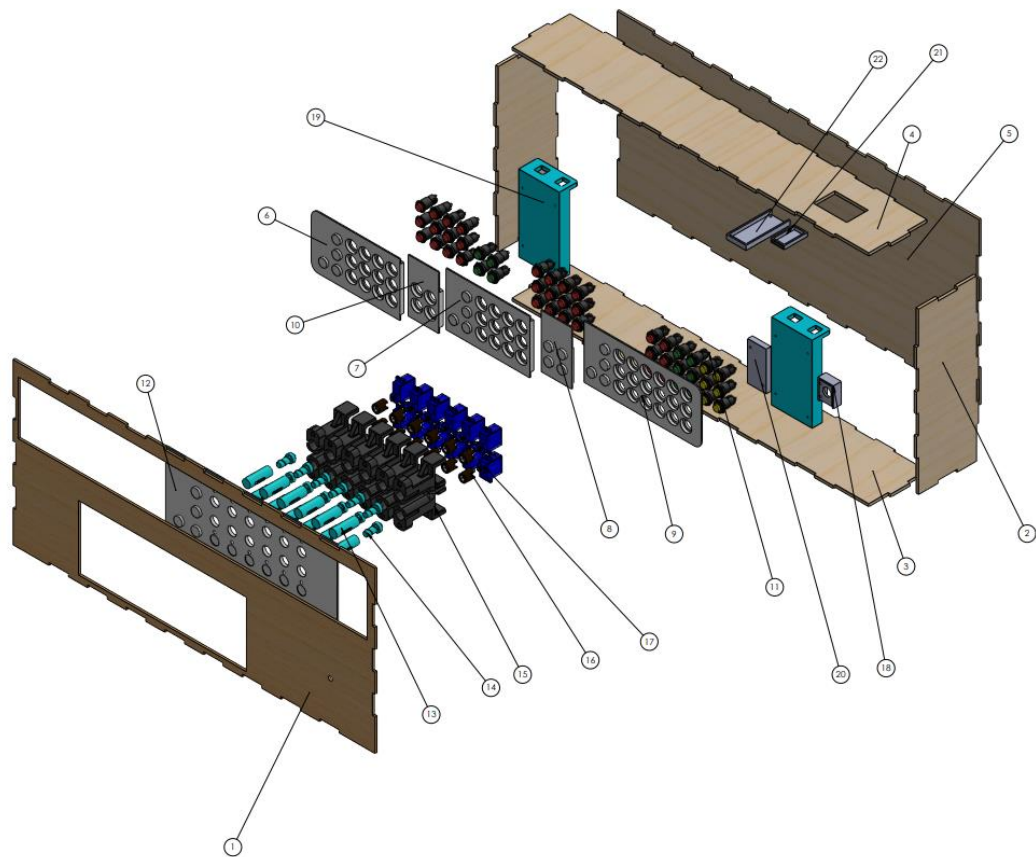


Fig. 4.22 Ensamble del dispositivo.

1. Se sueldan los botones ON/OFF (11) de manera que formen matrices 2x3 (6) (7) (9), compartiendo tierra común.
2. Se introdujo los tornillos sin fin (14), al riel de desplazamiento (13), para posterior unir al acople del servo motor (16).
3. Una vez armado los servomotores con los tornillos sin fin se acoplarán a las guías (15).
4. Colocamos las guías con los servos en la matriz diseñada para la sujeción de cada mecanismo (12) y su conexión.
5. Se sujeta la estructura tanto de los dispositivos electrónicos como la estructura externa.
6. Finalmente se obtiene acoplada la matriz de la botonera, matriz de servomotores y botón de verificación de respuesta de las operaciones completando así el ensamblaje del dispositivo. En la Fig. 4.22 se muestra el dispositivo en vista explosionada para mejor comprensión del proceso de ensamblaje.

4.12. Pruebas

Para evaluar la eficacia y confiabilidad del dispositivo desarrollado, se llevaron a cabo una serie de pruebas destinadas a verificar su desempeño en diferentes condiciones de operación.

El objetivo principal de estas pruebas fue medir la capacidad del sistema para responder correctamente a comandos específicos y analizar cualquier posible falla en la matriz de corrección. La metodología empleada consistió en realizar pruebas en grupos de 20, 40, 60, 80 y 100 repeticiones, cada uno de los cuales permitió observar la tasa de éxito y fallos en condiciones controladas y bajo diferentes números de iteraciones.

4.12.1. Prueba Nro. 1

Este conjunto inicial de 20 pruebas que se muestra en la Tabla 4.10, se diseñó para evaluar la funcionalidad básica del dispositivo y detectar cualquier error temprano en el sistema de respuesta. La prueba se enfocó en identificar problemas de conectividad y precisión en la matriz de corrección, proporcionando un primer diagnóstico de las áreas que requerían ajustes.

Tabla 4.10 Resultados de la prueba Nro. 1

Prueba	Resultado	7	Fallo	14	Fallo
1	Correcto	8	Fallo	15	Correcto
2	Fallo	9	Correcto	16	Fallo
3	Correcto	10	Correcto	17	Fallo
4	Fallo	11	Correcto	18	Correcto
5	Correcto	12	Fallo	19	Correcto
6	Correcto	13	Correcto	20	Correcto

Tabla 4.11 Porcentaje de éxito de la prueba Nro.1

Total pruebas	Pruebas correctas	Pruebas fallidas	Porcentaje de éxito (%)	Porcentaje Fallo (%)
20	12	8	60	40

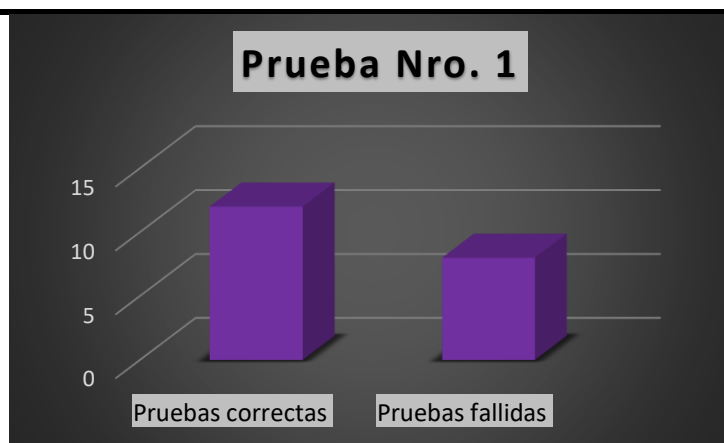


Fig. 4.23 Prueba Nro.1.

En esta primera fase de pruebas, el dispositivo presentó una tasa de fallos del 40% (ver Tabla 4.11), reflejada en 8 errores de respuesta en la matriz de corrección como se puede apreciar en la Fig. 4.23. Las fallas principales observadas fueron problemas en la comunicación entre el microcontrolador y el módulo de control de servos, lo cual causaba movimientos imprecisos y fallos en el posicionamiento de los actuadores. Además, algunos botones de enclave no respondían adecuadamente, lo que sugería posibles problemas de conexión o variabilidad en el voltaje de entrada.

4.12.2. Prueba Nro. 2

Con los ajustes preliminares implementados, esta fase buscó observar el comportamiento del dispositivo en condiciones de mayor demanda, realizando 40 pruebas consecutivas como se muestra en la Tabla 4.12. Se prestó especial atención a la sincronización entre los componentes y la estabilidad de la comunicación, elementos clave para mejorar la precisión y eficiencia del sistema.

Tabla 4.12 Resultados de la prueba Nro.2

Prueba	Resultado	14	Correcto	28	Correcto
1	Correcto	15	Fallo	29	Correcto
2	Correcto	16	Correcto	30	Correcto
3	Correcto	17	Correcto	31	Fallo
4	Fallo	18	Correcto	32	Correcto
5	Correcto	19	Fallo	33	Correcto
6	Correcto	20	Correcto	34	Correcto
7	Correcto	21	Fallo	35	Fallo
8	Fallo	22	Correcto	36	Correcto
9	Correcto	23	Correcto	37	Correcto
10	Correcto	24	Correcto	38	Fallo
11	Correcto	25	Fallo	39	Correcto
12	Fallo	26	Correcto	40	Correcto
13	Correcto	27	Correcto		

Tabla 4.13 Porcentaje de éxito en la prueba Nro.2

Total pruebas	Pruebas correctas	Pruebas fallidas	Porcentaje de éxito (%)	Porcentaje Fallo (%)
40	30	10	75	25

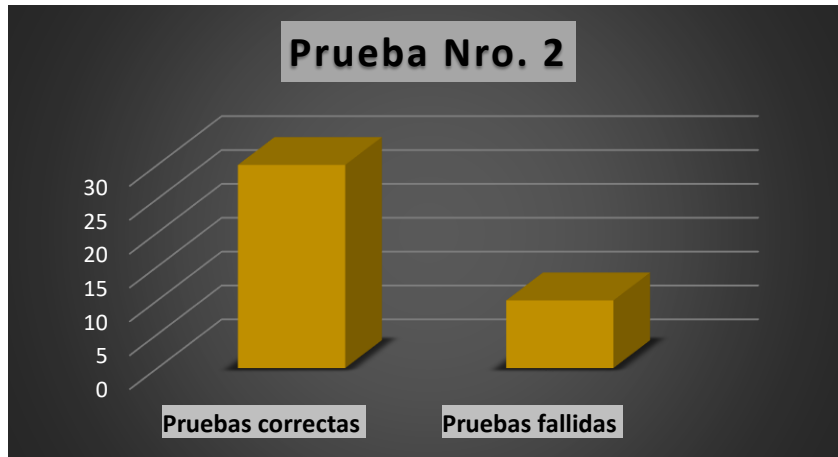


Fig. 4.24 Prueba Nro.2.

Durante esta serie de pruebas, la tasa de fallos se redujo al 25% como se detalla en la Tabla 4.13, con 10 errores en el sistema (ver Fig. 4.24). Los principales problemas identificados se relacionaron con la sincronización entre los diferentes servos, lo cual afectaba la precisión y velocidad de respuesta del dispositivo. Este comportamiento indicó la necesidad de implementar ajustes en la programación del controlador de servos, así como en el regulador de voltaje, para estabilizar la alimentación de cada componente durante el funcionamiento.

4.12.3. Prueba Nro. 3

Las pruebas de 60 repeticiones que se observa en la Tabla 4.14, permitieron continuar evaluando la estabilidad del sistema en escenarios más prolongados. Esta fase se centró en

optimizar la integración de los componentes y reducir el margen de error en el encoder rotativo y los actuadores, especialmente durante la transmisión de comandos.

Tabla 4.14 Resultados de la prueba Nro.3

Prueba	Resultado	21	Correcto	42	Correcto
1	Correcto	22	Correcto	43	Correcto
2	Fallo	23	Correcto	44	Correcto
3	Correcto	24	Correcto	45	Fallo
4	Correcto	25	Correcto	46	Correcto
5	Correcto	26	Correcto	47	Correcto
6	Fallo	27	Correcto	48	Correcto
7	Correcto	28	Fallo	49	Correcto
8	Correcto	29	Correcto	50	Correcto
9	Correcto	30	Correcto	51	Correcto
10	Correcto	31	Correcto	52	Fallo
11	Correcto	32	Correcto	53	Correcto
12	Correcto	33	Correcto	54	Correcto
13	Fallo	34	Correcto	55	Correcto
14	Correcto	35	Correcto	56	Correcto
15	Correcto	36	Fallo	57	Fallo
16	Correcto	37	Correcto	58	Correcto
17	Correcto	38	Correcto	59	Correcto
18	Correcto	39	Correcto	60	Correcto
19	Fallo	40	Correcto		
20	Correcto	41	Correcto		

Tabla 4.15 Porcentaje de éxito en la prueba Nro.3

Total pruebas	Pruebas correctas	Pruebas fallidas	Porcentaje de éxito (%)	Porcentaje Fallo (%)
60	51	9	85	15

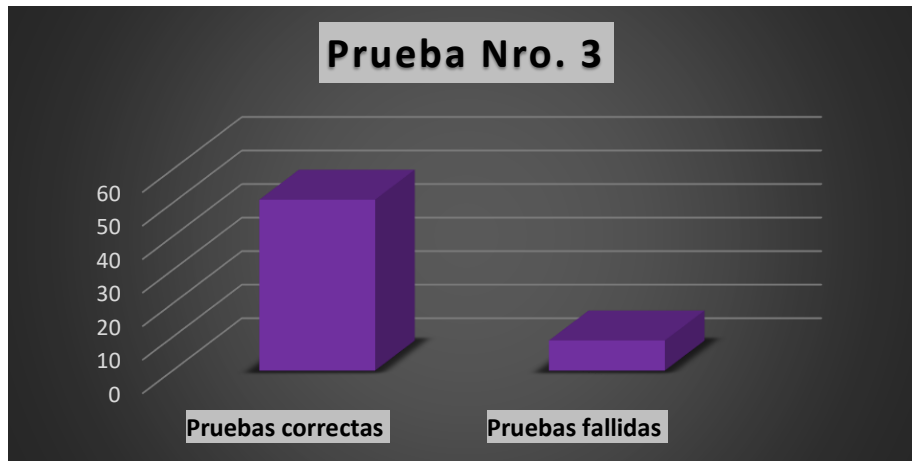


Fig. 4.25 Prueba Nro.3.

A medida que el sistema fue optimizado, la tasa de fallos se redujo al 15% (véase Tabla 4.15), lo cual representa una mejora considerable. Los errores observados en la Fig. 4.25, se debieron principalmente a pequeños desajustes en el encoder rotativo, afectando el registro preciso de la posición y el control de los servos. Además, se observaron problemas ocasionales en la conectividad del módulo Bluetooth, lo cual interrumpía la comunicación remota en algunas pruebas.

4.12.4. Prueba Nro. 4

En esta fase, el enfoque se dirigió a asegurar la consistencia del rendimiento del dispositivo bajo una carga considerable de pruebas, con 80 repeticiones que se puede apreciar en la Tabla 4.16. La finalidad era identificar y minimizar las fallas restantes, evaluando la capacidad del dispositivo para responder de forma ágil y precisa incluso en operaciones extendidas.

Tabla 4.16 Resultados de prueba Nro.4

Prueba	Resultado	3	Correcto	6	Correcto
1	Correcto	4	Fallo	7	Correcto
2	Correcto	5	Correcto	8	Correcto

9	Correcto	33	Correcto	57	Correcto
10	Correcto	34	Correcto	58	Correcto
11	Correcto	35	Fallo	59	Fallo
12	Correcto	36	Correcto	60	Correcto
13	Correcto	37	Correcto	61	Correcto
14	Correcto	38	Correcto	62	Correcto
15	Correcto	39	Correcto	63	Correcto
16	Correcto	40	Correcto	64	Correcto
17	Fallo	41	Correcto	65	Correcto
18	Correcto	42	Correcto	66	Correcto
19	Correcto	43	Correcto	67	Correcto
20	Correcto	44	Correcto	68	Correcto
21	Correcto	45	Correcto	69	Correcto
22	Correcto	46	Correcto	70	Correcto
23	Correcto	47	Correcto	71	Correcto
24	Correcto	48	Correcto	72	Correcto
25	Correcto	49	Correcto	73	Fallo
26	Correcto	50	Fallo	74	Correcto
27	Fallo	51	Correcto	75	Correcto
28	Correcto	52	Correcto	76	Correcto
29	Correcto	53	Correcto	77	Correcto
30	Correcto	54	Correcto	78	Correcto
31	Correcto	55	Correcto	79	Correcto
32	Correcto	56	Correcto	80	Correcto

Tabla 4.17 Porcentaje de éxito en la prueba Nro.4

Total, pruebas	Pruebas correctas	Pruebas fallidas	Porcentaje de éxito (%)	Porcentaje Fallo (%)
80	73	7	91.25	8.75

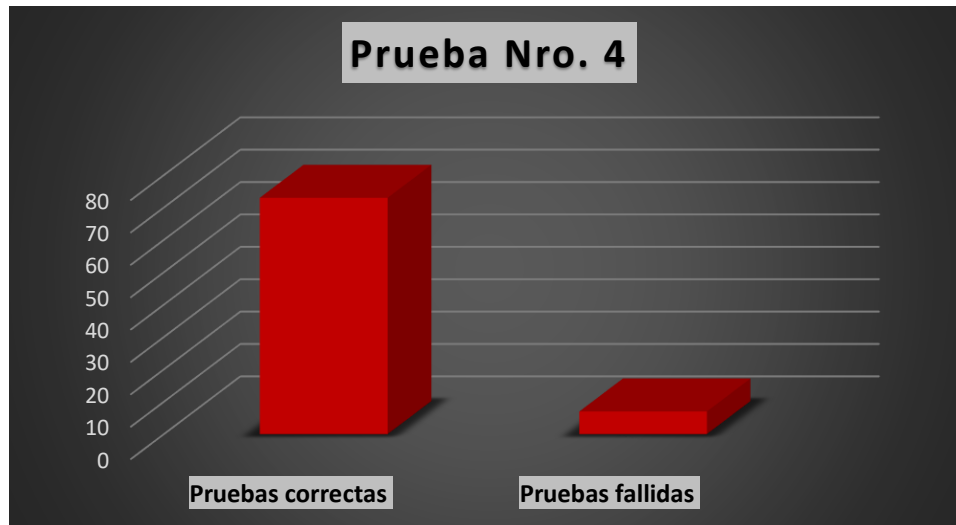


Fig. 4.26 Prueba Nro.4.

En esta fase, el sistema alcanzó un 91.25% de éxito que se muestra en la Tabla 4.17, con solo 7 fallos plasmados en la Fig. 4.26. Los errores residuales se relacionaron principalmente con una respuesta ligeramente lenta en el sistema de corrección de posición, especialmente cuando el dispositivo operaba bajo cargas más altas o en condiciones de mayor demanda de respuesta. Estos fallos se resolvieron mediante ajustes en el código de control y mejoras en la distribución de la carga del sistema.

4.12.5. Prueba Nro. 5

La fase final de 100 repeticiones que se observa en la Tabla 4.18, se realizó para validar la eficiencia y confiabilidad total del dispositivo, verificando que alcanzara un rendimiento óptimo. Este conjunto de pruebas confirmó la capacidad del sistema para operar sin errores, consolidando los ajustes realizados en las etapas previas y demostrando que el dispositivo cumple con los estándares requeridos.

Tabla 4.18 Resultados de la prueba Nro.4

Prueba	Resultado				
1	Correcto	34	Correcto	68	Correcto
2	Correcto	35	Correcto	69	Correcto
3	Correcto	36	Correcto	70	Correcto
4	Correcto	37	Correcto	71	Correcto
5	Correcto	38	Correcto	72	Correcto
6	Correcto	39	Correcto	73	Correcto
7	Correcto	40	Correcto	74	Correcto
8	Correcto	41	Correcto	75	Correcto
9	Correcto	42	Correcto	76	Correcto
10	Correcto	43	Correcto	77	Correcto
11	Correcto	44	Correcto	78	Correcto
12	Correcto	45	Correcto	79	Correcto
13	Correcto	46	Correcto	80	Correcto
14	Correcto	47	Correcto	81	Correcto
15	Correcto	48	Correcto	82	Correcto
16	Correcto	49	Correcto	83	Correcto
17	Correcto	50	Correcto	84	Correcto
18	Correcto	51	Correcto	85	Correcto
19	Correcto	52	Correcto	86	Correcto
20	Correcto	53	Correcto	87	Correcto
21	Correcto	54	Correcto	88	Correcto
22	Correcto	55	Correcto	89	Correcto
23	Correcto	56	Correcto	90	Correcto
24	Correcto	57	Correcto	91	Correcto
25	Correcto	58	Correcto	92	Correcto
26	Correcto	59	Correcto	93	Correcto
27	Correcto	60	Correcto	94	Correcto
28	Correcto	61	Correcto	95	Correcto
29	Correcto	62	Correcto	96	Correcto
30	Correcto	63	Correcto	97	Correcto
31	Correcto	64	Correcto	98	Correcto
32	Correcto	65	Correcto	99	Correcto
33	Correcto	66	Correcto	100	Correcto
		67	Correcto		

Tabla 4.19 Porcentaje de éxito en la prueba Nro.5

Total, pruebas	Pruebas correctas	Pruebas fallidas	Porcentaje de éxito (%)	Porcentaje Fallo (%)
100	100	0	100	0

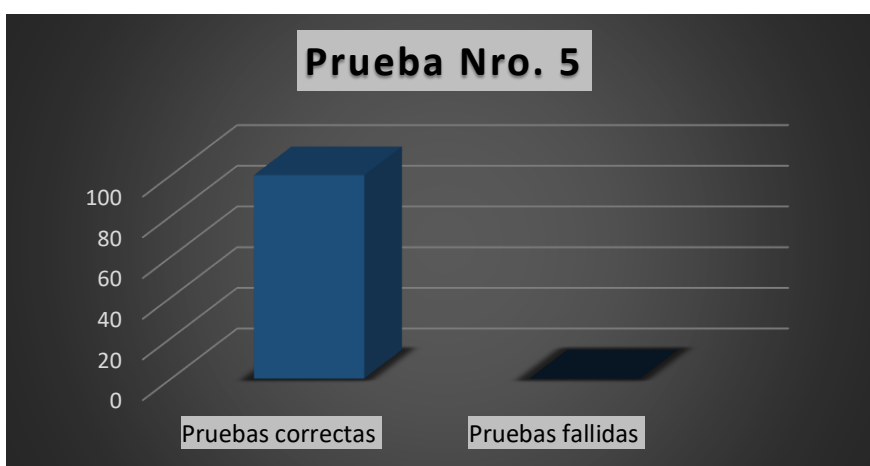


Fig. 4.27 Prueba Nro.5.

Para la prueba final, se alcanzó un rendimiento óptimo con un 100% de éxito como se muestra en la Tabla 4.18. La ausencia de fallos en esta fase (ver Fig.4.27), fue el resultado de los ajustes implementados a lo largo de las pruebas anteriores, incluyendo la optimización del sistema de comunicación y la calibración precisa de los componentes electrónicos y mecánicos. Esta última fase confirmó que el dispositivo puede operar con alta precisión y confiabilidad en condiciones reales, validando la efectividad del sistema para su propósito final.

Durante las primeras fases de prueba, se documentaron los errores y se identificaron áreas de mejora, permitiendo implementar ajustes en el sistema hasta lograr un rendimiento

óptimo. A lo largo de este proceso, se registraron tanto el porcentaje de aciertos como el de fallos en cada conjunto de pruebas, permitiendo un análisis comparativo de los avances logrados y la eficacia general del dispositivo al alcanzar el 100% de precisión en las pruebas finales como se aprecia en la Fig. 4.28.

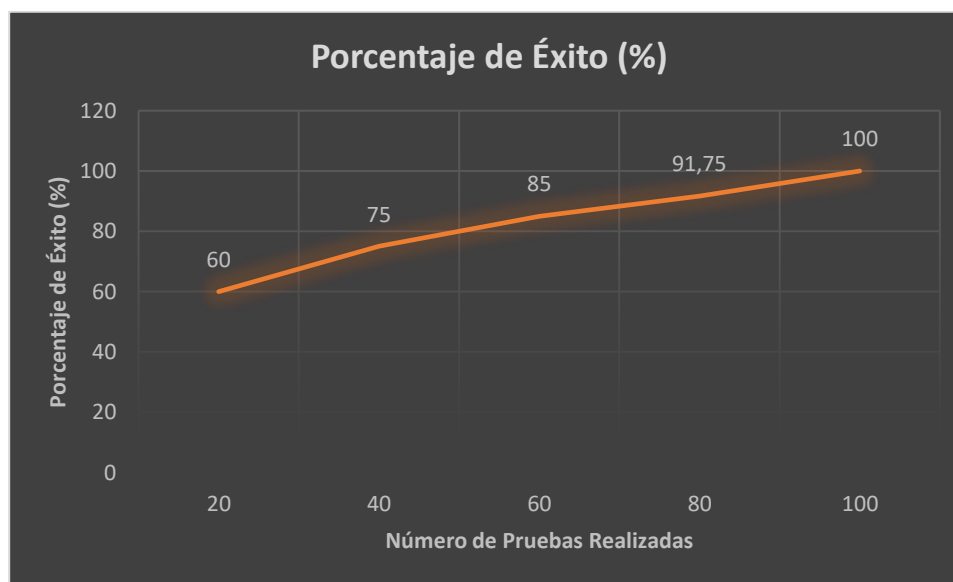


Fig. 4.28 Porcentaje de éxito en el dispositivo.

Estas pruebas son fundamentales para asegurar que el dispositivo cumple con los estándares de calidad y responde adecuadamente en condiciones operativas reales. A continuación, se detallan los resultados obtenidos en cada fase de pruebas, resaltando los aspectos técnicos y los ajustes implementados para mejorar el rendimiento del sistema.

CONCLUSIONES

Se identificaron los requerimientos del dispositivo, como la proporcionalidad de los caracteres Braille, su portabilidad y una estructura ergonómica, garantizando su uso por estudiantes con discapacidad visual. Se analizaron las necesidades educativas, características de accesibilidad y especificaciones técnicas, como materiales, elementos electrónicos y un diseño modular para facilitar el montaje y mantenimiento, asegurando funcionalidad, seguridad y adaptación al aprendizaje en Braille.

Se diseñó un sistema interactivo que permite a los usuarios con discapacidad visual realizar operaciones matemáticas básicas de manera autónoma y accesible. Este diseño consideró elementos táctiles en Braille para el ingreso de datos y la retroalimentación, cumpliendo con los principios de usabilidad e inclusión basados en la sencillez de manejo del dispositivo.

Se construyó un prototipo funcional del dispositivo mecatrónico, integrando componentes electrónicos y estructurales que cumplen con las especificaciones de diseño. El proceso de fabricación utilizó tecnologías como impresión 3D y corte láser, optimizando recursos y garantizando un producto accesible y eficiente, con un costo aproximado de 275,00 USD.

El prototipo fue validado mediante pruebas experimentales y análisis técnicos que confirmaron su funcionamiento y cumplimiento de los objetivos. Tras realizar ajustes específicos, el último banco de 100 pruebas alcanzó un éxito del 100%. Además, se comprobó su capacidad para soportar las cargas mecánicas necesarias y operar de manera óptima en el contexto educativo para el cual fue diseñado.

RECOMENDACIONES

Continuar investigando y desarrollando nuevas tecnologías para mejorar la precisión y la velocidad del dispositivo en la representación de símbolos matemáticos en Braille.

Explorar la adaptación del dispositivo para aplicaciones en diferentes niveles educativos y otras áreas del conocimiento que requieran la representación táctil de información compleja.

Ofrecer capacitación continua a educadores y personal educativo sobre el uso efectivo del dispositivo para maximizar su impacto en el aprendizaje de los estudiantes.

Incorporar la implementación de operaciones matemáticas adicionales, como multiplicación y división, así como aumentar el rango de cifras que el dispositivo puede procesar, mejorando su versatilidad y aplicabilidad en diferentes escenarios.

Reducir las dimensiones físicas del dispositivo mediante la optimización del diseño, manteniendo su funcionalidad actual.

REFERENCIAS

- [1] J. E. Bernal Suquinagua, . L. A. Narváez Angamarca y W. E. Alvarez Ocampo, «Repositorio Institucional de la Universidad Politécnica Salesiana,» 8 Marzo 2011. [En línea]. Available: <https://dspace.ups.edu.ec/handle/123456789/1404>. [Último acceso: 9 Abril 2024].
- [2] S. P. Ulloa Alvear y L. M. Ulloa Alvear, «Educación inclusiva y su incidencia en el proceso de enseñanza aprendizaje de niños de baja visión. Importancia del material didáctico,» *Revista San Gregorio*, nº 29, p. 34, 2019.
- [3] J. C. Ocampo, «Discapacidad, Inclusión y Educación Superior en Ecuador: El Caso de la Universidad Católica de Santiago de Guayaquil,» *Revista latinoamericana de educación inclusiva*, vol. 12, nº 2, p. 18, 2018.
- [4] D. M. Coronel Narváez y J. A. Mosquera Ospina, «Repositorio Digital de la Universidad Nacional de Educación UNAE,» Septiembre 2021. [En línea]. Available: <http://repositorio.unae.edu.ec/handle/56000/1903>. [Último acceso: 9 Abril 2024].
- [5] S. Llanos, I. D. Cobo Cruz, V. C. Gonzales Gomez, T. V. Montoya Gutierrez y L. C. Rojas Cruz, «Educación inclusiva desde la discapacidad visual: Estrategias orientadoras para el profesorado en formación de la primera infancia,» *Redipe*, vol. 13, nº 4, pp. 76-99, 2023.

- [6] S. I. Ruiz, «La fabricación digital como herramienta de educación inclusiva para personas con discapacidad visual,» *Limaq*, vol. 2, n° 10, pp. 1-8, 2022.
- [7] A. Hernandez Quirama y M. d. P. Oviedo Caceres, «La educación inclusiva para el colectivo docente es un reto que se asume en soledad,» *Logos Ciencia y Tecnología*, vol. 11, n° 2, pp. 113-125, 2019.
- [8] G. G. Nuria, «Universidad de Valladolid Instituto Universitario de Oftalmología Aplicada (IOBA),» 5 Julio 2023. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/62692/TFM-M678.pdf?sequence=1&isAllowed=y>. [Último acceso: 11 Agosto 2024].
- [9] . F. E. Villa Cajilima y C. Avendaño Mejía, «Repositorio institucional Universidad de Azuay,» 2019. [En línea]. Available: <http://dspace.uazuay.edu.ec/handle/datos/8920>. [Último acceso: 9 Abril 2024].
- [10] P. J. Valarezo Argudo, R. P. Pucha Quito y G. P. Hurtado Crespo, «Ciencia Digital,» 8 Noviembre 2019. [En línea]. Available: <https://doi.org/10.33262/cienciadigital.v3i4.1.978>. [Último acceso: 09 04 2024].
- [11] M. C. HERNÁNDEZ OSPINA, «Repositorio Institucional E-docUR,» 17 Diciembre 2020. [En línea]. Available: https://doi.org/10.48713/10336_30704. [Último acceso: 9 Abril 2024].

- [12] Y. C. Calderon, «Journal Boliviano de Ciencias,» 22 Diciembre 2023. [En línea]. Available: <https://doi.org/10.52428/20758944.v19i54.990>. [Último acceso: 9 Abril 2024].
- [13] S. A. Alulema-Asqui y M. Campoverde Molina, «Revista Multidisciplinaria Arbitrada de Investigación Científica,» 25 Octubre 2023. [En línea]. Available: <https://doi.org/10.56048/MQR20225.7.4.2023.1188-1206>. [Último acceso: 9 Abril 2024].
- [14] B. A. Rueda Diaz, «ResearchGate,» Noviembre 2020. [En línea]. Available: https://www.researchgate.net/publication/345311809_IMPLEMENTACION_DE_UN_DISPOSITIVO_DE_ENLACE_ENTRE_DOCENTES_Y_ESTUDIANTES_CON_DISCAPACIDAD_VISUAL_EN_EL_AREA_DE_MATEMATICAS. [Último acceso: 9 Abril 2024].
- [15] D. S. Santacruz Padilla, F. C. Gómez Meneses y E. Viveros Villada, Entrelazando formación, experiencias, escenarios y procesos vivenciales de investigación e innovación, San Juan de Pasto: Luz Elida Vera Hernández, Editorial UNIMAR, 2021.
- [16] E. Cuan Durón, E. Urquizo Barraza, M. A. Domínguez Zúñiga y E. Cerrillo Andrade, «Ciencia Básica y Aplicada, Universidad Autónoma de Baja California,» 23 Septiembre 2020. [En línea]. Available: http://revistaaristas.tij.uabc.mx/index.php/revista_aristas/article/view/30. [Último acceso: 9 Abril 2024].

- [17] E. A. González B, K. J. Villadiego.F y E. E. Rodríguez B, «Conocimiento, investigación y educación CIE,» 12 Octubre 2022. [En línea]. Available: <https://ojs.unipamplona.edu.co/index.php/cie/article/view/1504>. [Último acceso: 9 Abril 2024].
- [18] G. D. Brito Moncayo y C. F. Pérez Rivera, «Repositorio Digital. Universidad Técnica de Ambato,» Octubre 2016. [En línea]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/24019>. [Último acceso: 9 Abril 2024].
- [19] F. A. Iván, «Linkedin,» 1 Diciembre 2023. [En línea]. Available: https://www.linkedin.com/pulse/louis-braille-el-hombre-que-sac%C3%B3-los-invidentes-de-fern%C3%A1ndez-amil-ryfgf?trk=public_profile_article_view. [Último acceso: 08 Septiembre 2024].
- [20] T. C. Iván, «Tiflogía,» 14 Abril 2019. [En línea]. Available: <https://tiflogia.blogspot.com/2015/04/el-abaco.html>. [Último acceso: 12 Noviembre 2024].
- [21] Goki, «Jugarijugar,» [En línea]. Available: <https://www.jugarijugar.com/es/matematicas-manipulativas/2163-regletas-de-calculo-con-numerales.html>. [Último acceso: 14 Noviembre 2024].
- [22] «Inclusive Technology International,» [En línea]. Available: <https://www.inclusive.international/949-2/>. [Último acceso: 15 Noviembre 2024].

- [23] «ORBIT Research,» 2019. [En línea]. [Último acceso: 2024].
- [24] «Duxbury Braille Translator,» 3 Julio 2023. [En línea]. Available: <https://www.duxburysystems.com/dbt.asp>. [Último acceso: 15 Noviembre 2024].
- [25] M. E. Zuñiga , M. V. Rosas y H. J. Viano, «DISMINUCIÓN VISUAL Y EDUCACIÓN INCLUSIVA A TRAVÉS DE PRÁCTICAS EDUCATIVAS ABIERTAS,» *Docentes conectados*, vol. 2, nº 4, 2019.
- [26] Humanware, «Humanware,» 2019. [En línea]. Available: <https://store.humanware.com/hus/blindness-brailnote-touch-plus-32.html>. [Último acceso: 03 Noviembre 2024].
- [27] «New England Low Vision,» [En línea]. Available: <https://nelowvision.com/product/braille-display-focus-40-5g/>. [Último acceso: 20 Octubre 2024].
- [28] «BRIDGES,» [En línea]. Available: <https://www.bridges-canada.com/products/tactileview>. [Último acceso: 19 Octubre 2024].
- [29] J. J. Castro Maldonado, L. K. Gomez Macho y E. Camargo Casallas, «SciELO,» 07 Septiembre 2020. [En línea]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-921X2023000100140. [Último acceso: 20 Mayo 2024].

- [30] P. V. Tania, «SciELO,» 04 Octubre 2022. [En línea]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-09762022000300004. [Último acceso: 20 Mayo 2024].
- [31] S. F. E. Andres, «SciELO,» 28 Septiembre 2022. [En línea]. Available: https://ve.scielo.org/scielo.php?pid=S1315-00062022000300010&script=sci_abstract. [Último acceso: 20 Mayo 2024].
- [32] G. P. Guevara Alban, A. E. Verdesoto Arguello y N. E. Castro Molina, «Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción),» *reciMundo*, vol. 3, nº 4, pp. 163-173, 2020.
- [33] R. G. Carlos, «DISEÑOS DE INVESTIGACIÓN EXPERIMENTAL,» *CienciAmerica*, vol. 10, nº 1, p. 7, 2021.
- [34] «Shop,» [En línea]. Available: <https://www.shop.com.mm/products/ds-428-push-button-with-lock-ce-store-i109027587.html>. [Último acceso: 05 Septiembre 2024].
- [35] «Stereon,» [En línea]. Available: <https://www.steren.com.ec/encoder-rotativo.html>. [Último acceso: 21 Junio 2024].
- [36] M. Gudiño, «Newsletter de Arrow,» 22 Diciembre 2017. [En línea]. Available: <https://www.arrow.com/es-mx/research-and-events/articles/arduino-uno-vs-mega-vs-micro>. [Último acceso: 11 Octubre 2024].

- [37] «Ubuy,» [En línea]. Available: <https://www.ubuy.ec/es/product/3N9L2GK-arduino-uno-rev-3>. [Último acceso: 2 Junio 2024].
- [38] «MEGATRONICA,» [En línea]. Available: <https://megatronica.cc/producto/arduino-mega-2560-cable/>. [Último acceso: 2 Junio 2024].
- [39] V. Arrieta, «Arca Electrónica,» 15 Agosto 2019. [En línea]. Available: https://www.arcaelectronica.com/blogs/tutoriales/servomotores-arduino-sg90-mg90-mg995-mg996?srsIid=AfmBOopgn24Gv-K8ktQxzU5GItYpR_ISP2LxFC-of49iN1Bzyts2OIA. [Último acceso: 08 Septiembre 2024].
- [40] «ORELLANA,» [En línea]. Available: <https://orellanaelec.com/producto/servomotor-sg90-9gr-180/>. [Último acceso: 3 Junio 2024].
- [41] «ElectroStore,» 2019. [En línea]. Available: <https://grupeelectrostore.com/shop/motores/servomotores/servomotor-tower-pro-mg90s-25kg-cm-micro/>. [Último acceso: 1 Junio 2024].
- [42] G. Saenz, «Slideshare,» 25 Agosto 2015. [En línea]. Available: <https://es.slideshare.net/slideshow/matriz-de-ponderacion/52068866>. [Último acceso: 23 Julio 2024].
- [43] «Alibaba,» [En línea]. Available: <https://spanish.alibaba.com/product-detail/ac-dc-power-supply-module-S-60478964365.html>. [Último acceso: 12 Junio 2024].

- [44] «Arduino.cl,» [En línea]. Available: <https://arduino.cl/producto/fuente-regulable-dc-dc/>. [Último acceso: 11 Junio 2024].
- [45] S. L. Simonson, «TAMESON,» 23 Diciembre 2022. [En línea]. Available: <https://tameson.es/pages/medir-el-tamano-y-paso-de-rozca#:~:text=%C2%BFC%C3%B3mo%20calcular%20el%20paso%20de,paso%20es%20de%202%20mm..> [Último acceso: 5 Julio 2024].

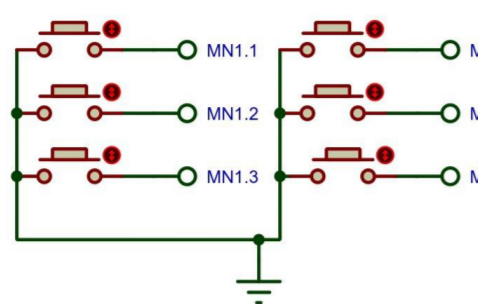
ANEXOS

Anexo A

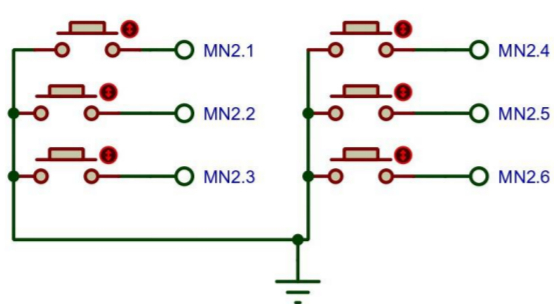
Diagrama eléctrico

Primer Número

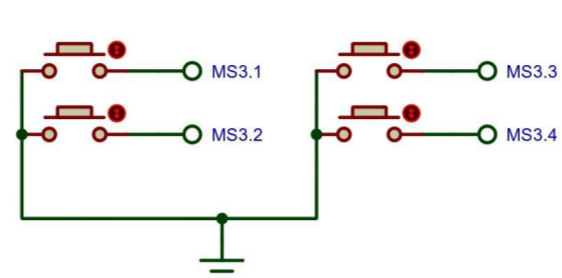
Unidad



Decena

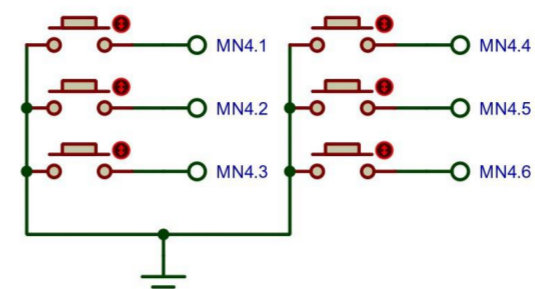


Operador Matemático Suma/Resta

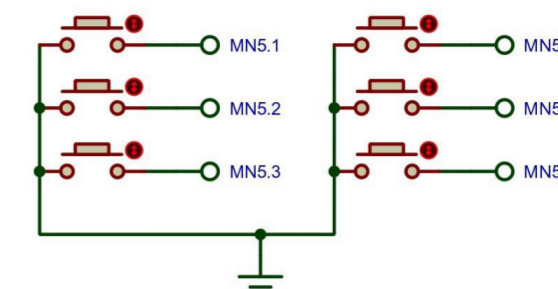


Segundo Número

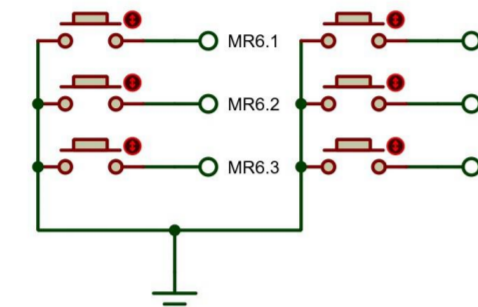
Unidad



Decena

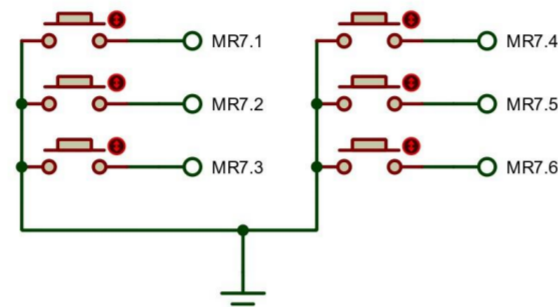


Unidad

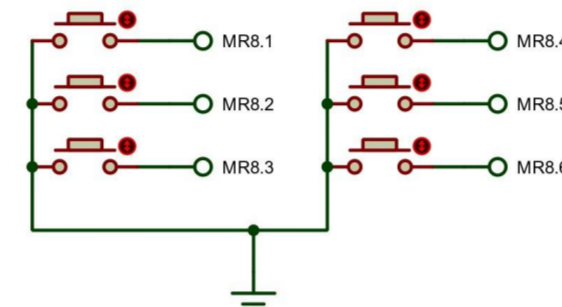


Respuesta

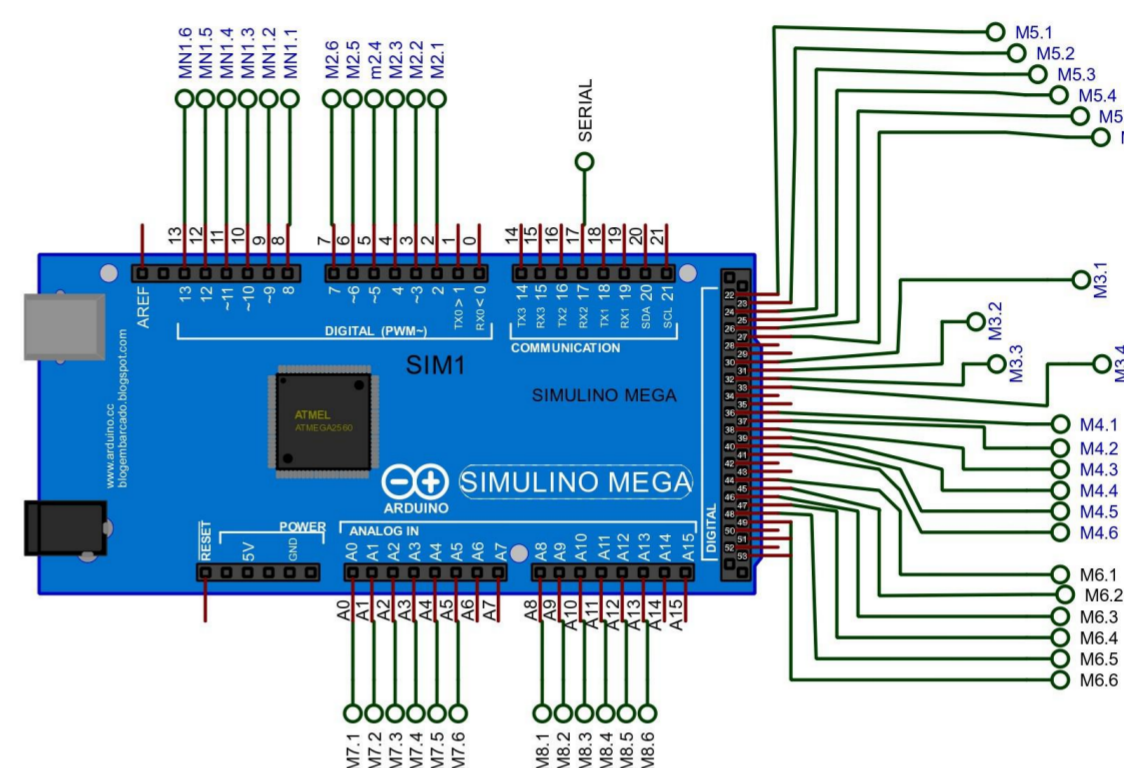
Decena



Centena

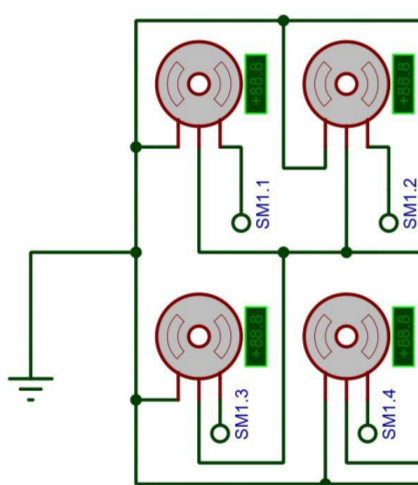


Controlador Ingreso de datos

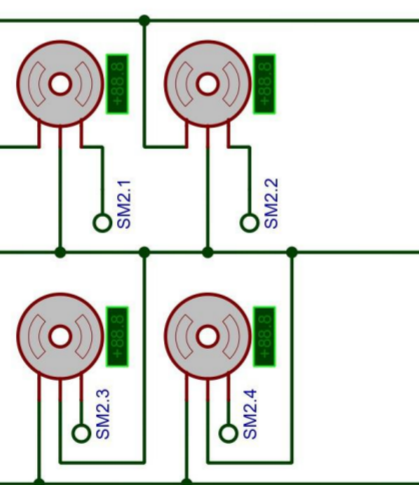


Corrección de Respuesta Servomotores SG90

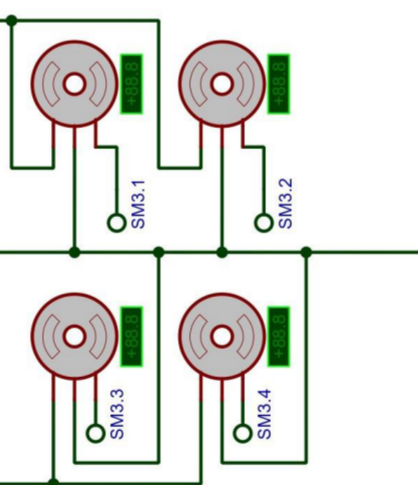
Unidad



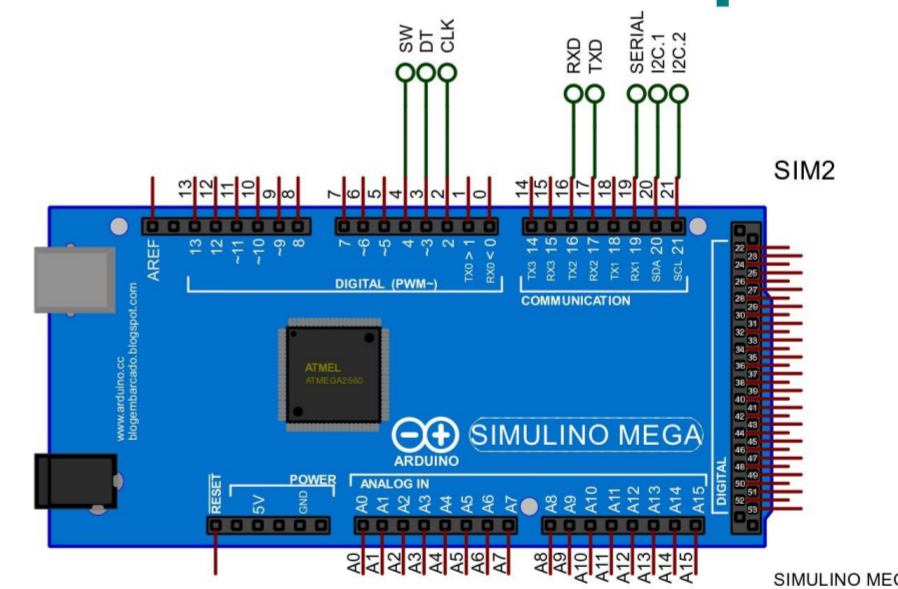
Decena



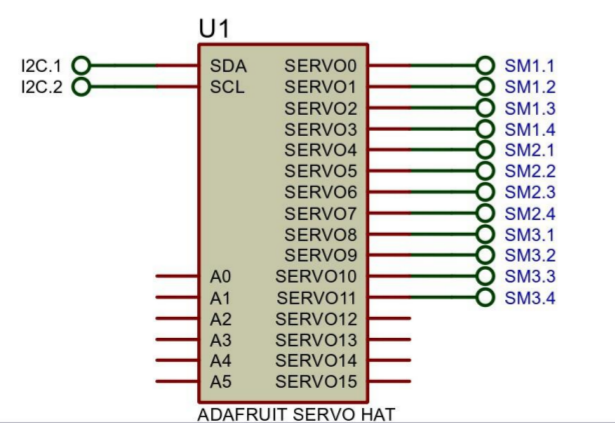
Centena



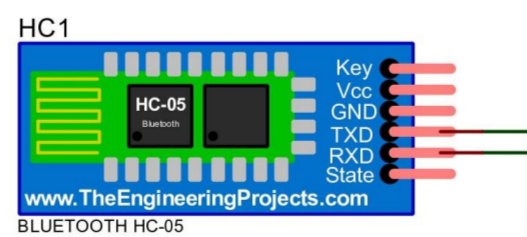
Controlador Corrección de Respuesta



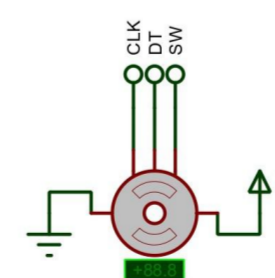
Driver Servomotores



Bluetooth



Encoder



UTN CIME

PROYECTO: DIAGRAMA ELÉCTRICO		ESCALA: 1:1	
PARTE: DIAGRAMA ELÉCTRICO		TOLERANCIA: ±	
CÓDIGO: DI-EL-CABR			
MATERIAL:	NINGUNO	DESIÑO:	CHIGUANO A. 6/1/2024
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A. 6/1/2024
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F. 17/7/2024
CANTIDAD:	1	APROBÓ:	VALENCIA F. 17/7/2024

Anexo B

Algoritmo de ingreso de datos

```
int num0[6] = {0,1,1,1,0,0};
int num1[6] = {1,0,0,0,0,0};
int num2[6] = {1,0,1,0,0,0};
int num3[6] = {1,1,0,0,0,0};
int num4[6] = {1,1,0,1,0,0};
int num5[6] = {1,0,0,1,0,0};
int num6[6] = {1,1,1,0,0,0};
int num7[6] = {1,1,1,1,0,0};
int num8[6] = {1,0,1,1,0,0};
int num9[6] = {0,1,1,0,0,0};
int vsuma[4]= {1,1,1,0};
int vresta[4]={0,0,1,1};
int dig1A[6];
int dig1B[6];
int ope[4];
int dig2A[6];
int dig2B[6];
int res1[6];
int res2[6];
int res3[6];
int i,c,des;
intver1A,ver1B,ver2A,ver2B,verr1,verr2,verr3,verop;
intvac1A,vac1B,vac2A,vac2B,vacr1,vacr2,vacr3;
intdigitola,digitolb,digito2a,digito2b,resul1,resul2,resul3;
int ing1,ing2,sol;
char finop[1];
char act;
```

```
void setup() {
Serial.begin(9600);
for (i=2;i<14;i++){
pinMode(i,INPUT_PULLUP);
}
for (c=22;c<70;c++){
pinMode(c,INPUT_PULLUP);
}
}
```

```
void loop() {
////////////////////////////////////
////////////////////////////////////Botones////////////////////////////////////
////////////////////////////////////

for (i=8,des=0;i<14;i++,des++){
if(digitalRead(i)==HIGH){
```

```
dig1A[des]=1;
} else {
    dig1A[des]=0;
}
}
for (i=2,des=0;i<8;i++,des++){
if(digitalRead(i)==HIGH) {
dig1B[des]=1;
} else {
    dig1B[des]=0;
}
}

if(digitalRead(62)==HIGH) {
dig2A[0]=1;
} else {
    dig2A[0]=0;
}
if(digitalRead(63)==HIGH) {
dig2A[1]=1;
} else {
    dig2A[1]=0;
}

if(digitalRead(64)==HIGH) {
dig2A[2]=1;
} else {
    dig2A[2]=0;
}

if(digitalRead(53)==HIGH) {
dig2A[3]=1;
} else {
    dig2A[3]=0;
}

    if(digitalRead(66)==HIGH) {
dig2A[4]=1;
} else {
    dig2A[4]=0;
}

    if(digitalRead(67)==HIGH) {
dig2A[5]=1;
} else {
    dig2A[5]=0;
}
```

```
}

for (i=22,des=0;i<28;i++,des++){
if(digitalRead(i)==HIGH){
dig2B[des]=1;
} else {
dig2B[des]=0;
}
}
```

```
for (i=30,des=0;i<34;i++,des++){
if(digitalRead(i)==HIGH){
ope[des]=1;
} else {
ope[des]=0;
}
}
```

```
for (i=54,des=0;i<60;i++,des++){
if(digitalRead(i)==HIGH){
res1[des]=1;
} else {
res1[des]=0;
}
}
```

```
for (i=36,des=0;i<42;i++,des++){
if(digitalRead(i)==HIGH){
res2[des]=1;
} else {
res2[des]=0;
}
}
```

```
for (i=44,des=0;i<50;i++,des++){
if(digitalRead(i)==HIGH){
res3[des]=1;
} else {
res3[des]=0;
}
}
```

```
delay(1000);
```

```
////////////////////////////////////  
    if(dig1A[0]==0 && dig1A[1]==0 && dig1A[2]==0 && dig1A[3]==0 && dig1A[4]==0 &&  
dig1A[5]==0) {  
    vac1A=1;  
    ver1A=0;  
    }  
  
    else if(dig1A[0]==num1[0] && dig1A[1]==num1[1] && dig1A[2]==num1[2] &&  
dig1A[3]==num1[3] && dig1A[4]==num1[4] && dig1A[5]==num1[5]){  
    ver1A=1;  
    digitola=1;  
    vac1A=0;  
//    Serial.print(dig1A[0]);  
//    Serial.print(dig1A[1]);  
//    Serial.print(dig1A[2]);  
//    Serial.print(dig1A[3]);  
//    Serial.print(dig1A[4]);  
//    Serial.println(dig1A[5]);  
//    Serial.println("Digito 1A");  
//    Serial.println(digitola);  
    }  
  
    else if(dig1A[0]==num2[0] && dig1A[1]==num2[1] && dig1A[2]==num2[2] &&  
dig1A[3]==num2[3] && dig1A[4]==num2[4] && dig1A[5]==num2[5]){  
    ver1A=1;  
    digitola=2;  
    vac1A=0;  
    }  
  
    else if(dig1A[0]==num3[0] && dig1A[1]==num3[1] && dig1A[2]==num3[2] &&  
dig1A[3]==num3[3] && dig1A[4]==num3[4] && dig1A[5]==num3[5]){  
    ver1A=1;  
    digitola=3;  
    vac1A=0;  
    }  
  
    else if(dig1A[0]==num4[0] && dig1A[1]==num4[1] && dig1A[2]==num4[2] &&  
dig1A[3]==num4[3] && dig1A[4]==num4[4] && dig1A[5]==num4[5]){  
    ver1A=1;  
    digitola=4;  
    vac1A=0;  
    }  
  
    else if(dig1A[0]==num5[0] && dig1A[1]==num5[1] && dig1A[2]==num5[2] &&  
dig1A[3]==num5[3] && dig1A[4]==num5[4] && dig1A[5]==num5[5]){  
    ver1A=1;
```

```
digitola=5;
vac1A=0;
}

else if(dig1A[0]==num6[0] && dig1A[1]==num6[1] && dig1A[2]==num6[2] &&
dig1A[3]==num6[3] && dig1A[4]==num6[4] && dig1A[5]==num6[5]){
    ver1A=1;
    digitola=6;
    vac1A=0;
}

else if(dig1A[0]==num7[0] && dig1A[1]==num7[1] && dig1A[2]==num7[2] &&
dig1A[3]==num7[3] && dig1A[4]==num7[4] && dig1A[5]==num7[5]){
    ver1A=1;
    digitola=7;
    vac1A=0;
}

else if(dig1A[0]==num8[0] && dig1A[1]==num8[1] && dig1A[2]==num8[2] &&
dig1A[3]==num8[3] && dig1A[4]==num8[4] && dig1A[5]==num8[5]){
    ver1A=1;
    digitola=8;
    vac1A=0;
}

else if(dig1A[0]==num9[0] && dig1A[1]==num9[1] && dig1A[2]==num9[2] &&
dig1A[3]==num9[3] && dig1A[4]==num9[4] && dig1A[5]==num9[5]){
    ver1A=1;
    digitola=9;
    vac1A=0;
}

else if(dig1A[0]==num0[0] && dig1A[1]==num0[1] && dig1A[2]==num0[2] &&
dig1A[3]==num0[3] && dig1A[4]==num0[4] && dig1A[5]==num0[5]){
    ver1A=1;
    digitola=0;
    vac1A=0;
}
else{
    ver1A=0;
}

////////////////////////////////////
/
////////////////////////////////////DIGITO1B////////////////////////////////////
/
```

```
////////////////////////////////////  
/  
  
    if(dig1B[0]==0 && dig1B[1]==0 && dig1B[2]==0 && dig1B[3]==0 && dig1B[4]==0 &&  
dig1B[5]==0) {  
    vac1B=1;  
    ver1B=0;  
    }  
  
    else if(dig1B[0]==num1[0] && dig1B[1]==num1[1] && dig1B[2]==num1[2] &&  
dig1B[3]==num1[3] && dig1B[4]==num1[4] && dig1B[5]==num1[5]){  
    digito1b=1;  
    ver1B=1;  
    vac1B=0;  
  
//    Serial.print(dig1B[0]);  
//    Serial.print(dig1B[1]);  
//    Serial.print(dig1B[2]);  
//    Serial.print(dig1B[3]);  
//    Serial.print(dig1B[4]);  
//    Serial.println(dig1B[5]);  
//    Serial.println("Digito 1B");  
//    Serial.println(digito1b);  
    }  
  
    else if(dig1B[0]==num2[0] && dig1B[1]==num2[1] && dig1B[2]==num2[2] &&  
dig1B[3]==num2[3] && dig1B[4]==num2[4] && dig1B[5]==num2[5]) {  
    ver1B=1;  
    digito1b=2;  
    vac1B=0;  
    }  
  
    else if(dig1B[0]==num3[0] && dig1B[1]==num3[1] && dig1B[2]==num3[2] &&  
dig1B[3]==num3[3] && dig1B[4]==num3[4] && dig1B[5]==num3[5]) {  
    ver1B=1;  
    digito1b=3;  
    vac1B=0;  
    }  
  
    else if(dig1B[0]==num4[0] && dig1B[1]==num4[1] && dig1B[2]==num4[2] &&  
dig1B[3]==num4[3] && dig1B[4]==num4[4] && dig1B[5]==num4[5]) {  
    ver1B=1;  
    digito1b=4;  
    vac1B=0;  
    }  
  
    else if(dig1B[0]==num5[0] && dig1B[1]==num5[1] && dig1B[2]==num5[2] &&
```



```
dig1B[3]==num5[3] && dig1B[4]==num5[4] && dig1B[5]==num5[5]) {
    ver1B=1;
    digitolb=5;
    vac1B=0;
}

else if(dig1B[0]==num6[0] && dig1B[1]==num6[1] && dig1B[2]==num6[2] &&
dig1B[3]==num6[3] && dig1B[4]==num6[4] && dig1B[5]==num6[5]) {
    ver1B=1;
    digitolb=6;
    vac1B=0;
}

else if(dig1B[0]==num7[0] && dig1B[1]==num7[1] && dig1B[2]==num7[2] &&
dig1B[3]==num7[3] && dig1B[4]==num7[4] && dig1B[5]==num7[5]) {
    ver1B=1;
    digitolb=7;
    vac1B=0;
}

else if(dig1B[0]==num8[0] && dig1B[1]==num8[1] && dig1B[2]==num8[2] &&
dig1B[3]==num8[3] && dig1B[4]==num8[4] && dig1B[5]==num8[5]) {
    ver1B=1;
    digitolb=8;
    vac1B=0;
}

else if(dig1B[0]==num9[0] && dig1B[1]==num9[1] && dig1B[2]==num9[2] &&
dig1B[3]==num9[3] && dig1B[4]==num9[4] && dig1B[5]==num9[5]) {
    ver1B=1;
    digitolb=9;
    vac1B=0;
}

else if(dig1B[0]==num0[0] && dig1B[1]==num0[1] && dig1B[2]==num0[2] &&
dig1B[3]==num0[3] && dig1B[4]==num0[4] && dig1B[5]==num0[5]) {
    ver1B=1;
    digitolb=0;
    vac1B=0;
}
else{
    ver1B=0;
}
```

////////////////////////////////////
////////////////////////////////////

```
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////DIGITO
2A////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
```

```
    if(dig2A[0]==0 && dig2A[1]==0 && dig2A[2]==0 && dig2A[3]==0 && dig2A[4]==0 &&
dig2A[5]==0) {
    vac2A=1;
    ver2A=0;
    }
```

```
    else if(dig2A[0]==num1[0] && dig2A[1]==num1[1] && dig2A[2]==num1[2] &&
dig2A[3]==num1[3] && dig2A[4]==num1[4] && dig2A[5]==num1[5]){
```

```
        ver2A=1;
        digito2a=1;
        vac2A=0;
```

```
// Serial.print(dig1A[0]);
// Serial.print(dig1A[1]);
// Serial.print(dig1A[2]);
// Serial.print(dig1A[3]);
// Serial.print(dig1A[4]);
// Serial.println(dig1A[5]);
// Serial.println("Digito 1A");
// Serial.println(digito1a);
}
```

```
    else if(dig2A[0]==num2[0] && dig2A[1]==num2[1] && dig2A[2]==num2[2] &&
dig2A[3]==num2[3] && dig2A[4]==num2[4] && dig2A[5]==num2[5]){
```

```
        ver2A=1;
        digito2a=2;
        vac2A=0;
    }
```

```
    else if(dig2A[0]==num3[0] && dig2A[1]==num3[1] && dig2A[2]==num3[2] &&
dig2A[3]==num3[3] && dig2A[4]==num3[4] && dig2A[5]==num3[5]){
```

```
        ver2A=1;
        digito2a=3;
        vac2A=0;
    }
```

```
    else if(dig2A[0]==num4[0] && dig2A[1]==num4[1] && dig2A[2]==num4[2] &&
dig2A[3]==num4[3] && dig2A[4]==num4[4] && dig2A[5]==num4[5]){
```

```
        ver2A=1;
```

```
digito2a=4;
vac2A=0;
}

else if(dig2A[0]==num5[0] && dig2A[1]==num5[1] && dig2A[2]==num5[2] &&
dig2A[3]==num5[3] && dig2A[4]==num5[4] && dig2A[5]==num5[5]){
ver2A=1;
digito2a=5;
vac2A=0;
}

else if(dig2A[0]==num6[0] && dig2A[1]==num6[1] && dig2A[2]==num6[2] &&
dig2A[3]==num6[3] && dig2A[4]==num6[4] && dig2A[5]==num6[5]){
ver2A=1;
digito2a=6;
vac2A=0;
}

else if(dig2A[0]==num7[0] && dig2A[1]==num7[1] && dig2A[2]==num7[2] &&
dig2A[3]==num7[3] && dig2A[4]==num7[4] && dig2A[5]==num7[5]){
ver2A=1;
digito2a=7;
vac2A=0;
}

else if(dig2A[0]==num8[0] && dig2A[1]==num8[1] && dig2A[2]==num8[2] &&
dig2A[3]==num8[3] && dig2A[4]==num8[4] && dig2A[5]==num8[5]){
ver2A=1;
digito2a=8;
vac2A=0;
}

else if(dig2A[0]==num9[0] && dig2A[1]==num9[1] && dig2A[2]==num9[2] &&
dig2A[3]==num9[3] && dig2A[4]==num9[4] && dig2A[5]==num9[5]){
ver2A=1;
digito2a=9;
vac2A=0;
}

else if(dig2A[0]==num0[0] && dig2A[1]==num0[1] && dig2A[2]==num0[2] &&
dig2A[3]==num0[3] && dig2A[4]==num0[4] && dig2A[5]==num0[5]){
ver2A=1;
digito2a=0;
vac2A=0;
}
else{
```

```
ver2A=0;
}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////DIGITO
2B////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

if(dig2B[0]==num1[0] && dig2B[1]==num1[1] && dig2B[2]==num1[2] &&
dig2B[3]==num1[3] && dig2B[4]==num1[4] && dig2B[5]==num1[5]){
    ver2B=1;
    digito2b=1;
    vac2B=0;
// Serial.print(dig1B[0]);
// Serial.print(dig1B[1]);
// Serial.print(dig1B[2]);
// Serial.print(dig1B[3]);
// Serial.print(dig1B[4]);
// Serial.println(dig1B[5]);
// Serial.println("Digito 2B");
// Serial.println(digito2b);
}

else if(dig2B[0]==num2[0] && dig2B[1]==num2[1] && dig2B[2]==num2[2] &&
dig2B[3]==num2[3] && dig2B[4]==num2[4] && dig2B[5]==num2[5]){
    ver2B=1;
    digito2b=2;
    vac2B=0;
}

else if(dig2B[0]==num3[0] && dig2B[1]==num3[1] && dig2B[2]==num3[2] &&
dig2B[3]==num3[3] && dig2B[4]==num3[4] && dig2B[5]==num3[5]){
    ver2B=1;
    digito2b=3;
    vac2B=0;
}

else if(dig2B[0]==num4[0] && dig2B[1]==num4[1] && dig2B[2]==num4[2] &&
dig2B[3]==num4[3] && dig2B[4]==num4[4] && dig2B[5]==num4[5]){
    ver2B=1;
    digito2b=4;
    vac2B=0;
}
```

```
else if(dig2B[0]==num5[0] && dig2B[1]==num5[1] && dig2B[2]==num5[2] &&
dig2B[3]==num5[3] && dig2B[4]==num5[4] && dig2B[5]==num5[5]) {
    ver2B=1;
    digito2b=5;
    vac2B=0;
}

else if(dig2B[0]==num6[0] && dig2B[1]==num6[1] && dig2B[2]==num6[2] &&
dig2B[3]==num6[3] && dig2B[4]==num6[4] && dig2B[5]==num6[5]) {
    ver2B=1;
    digito2b=6;
    vac2B=0;
}

else if(dig2B[0]==num7[0] && dig2B[1]==num7[1] && dig2B[2]==num7[2] &&
dig2B[3]==num7[3] && dig2B[4]==num7[4] && dig2B[5]==num7[5]) {
    ver2B=1;
    digito2b=7;
    vac2B=0;
}

else if(dig2B[0]==num8[0] && dig2B[1]==num8[1] && dig2B[2]==num8[2] &&
dig2B[3]==num8[3] && dig2B[4]==num8[4] && dig2B[5]==num8[5]) {
    ver2B=1;
    digito2b=8;
    vac2B=0;
}

else if(dig2B[0]==num9[0] && dig2B[1]==num9[1] && dig2B[2]==num9[2] &&
dig2B[3]==num9[3] && dig2B[4]==num9[4] && dig2B[5]==num9[5]) {
    ver2B=1;
    digito2b=9;
    vac2B=0;
}

else if(dig2B[0]==num0[0] && dig2B[1]==num0[1] && dig2B[2]==num0[2] &&
dig2B[3]==num0[3] && dig2B[4]==num0[4] && dig2B[5]==num0[5]) {
    ver2B=1;
    digito2b=0;
    vac2B=0;
}

else if(dig2B[0]==0 && dig2B[1]==0 && dig2B[2]==0 && dig2B[3]==0 && dig2B[4]==0
&& dig2B[5]==0) {
    vac2B=1;
    ver2B=0;
}
```

```

    }
else{
    ver2B=0;
    }

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//////////////////////////////////////RESULTADO
RA////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

    if(res1[0]==0 && res1[1]==0 && res1[2]==0 && res1[3]==0 && res1[4]==0 &&
res1[5]==0) {
    verr1=0;
    vacr1=1;
    }

    else if(res1[0]==num1[0] && res1[1]==num1[1] && res1[2]==num1[2] &&
res1[3]==num1[3] && res1[4]==num1[4] && res1[5]==res1[5]){
    verr1=1;
    resul1=1;
    vacr1=0;
//    Serial.print(res1[0]);
//    Serial.print(res1[1]);
//    Serial.print(res1[2]);
//    Serial.print(res1[3]);
//    Serial.print(res1[4]);
//    Serial.println(res1[5]);
//    Serial.println("Digito RA");
//    Serial.println(resul1);
    }

    else if(res1[0]==num2[0] && res1[1]==num2[1] && res1[2]==num2[2] &&
res1[3]==num2[3] && res1[4]==num2[4] && res1[5]==num2[5]){
    verr1=1;
    resul1=2;
    vacr1=0;
    }

    else if(res1[0]==num3[0] && res1[1]==num3[1] && res1[2]==num3[2] &&
res1[3]==num3[3] && res1[4]==num3[4] && res1[5]==num3[5]){
    verr1=1;

```

```
    resul1=3;
    vacr1=0;
}

else if(res1[0]==num4[0] && res1[1]==num4[1] && res1[2]==num4[2] &&
res1[3]==num4[3] && res1[4]==num4[4] && res1[5]==num4[5]){
    verr1=1;
    resul1=4;
    vacr1=0;
}

else if(res1[0]==num5[0] && res1[1]==num5[1] && res1[2]==num5[2] &&
res1[3]==num5[3] && res1[4]==num5[4] && res1[5]==num5[5]){
    verr1=1;
    resul1=5;
    vacr1=0;
}

else if(res1[0]==num6[0] && res1[1]==num6[1] && res1[2]==num6[2] &&
res1[3]==num6[3] && res1[4]==num6[4] && res1[5]==num6[5]){
    verr1=1;
    resul1=6;
}

else if(res1[0]==num7[0] && res1[1]==num7[1] && res1[2]==num7[2] &&
res1[3]==num7[3] && res1[4]==num7[4] && res1[5]==num7[5]){
    verr1=1;
    resul1=7;
    vacr1=0;
}

else if(res1[0]==num8[0] && res1[1]==num8[1] && res1[2]==num8[2] &&
res1[3]==num8[3] && res1[4]==num8[4] && res1[5]==num8[5]){
    verr1=1;
    resul1=8;
    vacr1=0;
}

else if(res1[0]==num9[0] && res1[1]==num9[1] && res1[2]==num9[2] &&
res1[3]==num9[3] && res1[4]==num9[4] && res1[5]==num9[5]){
    verr1=1;
    resul1=9;
    vacr1=0;
}

else if(res1[0]==num0[0] && res1[1]==num0[1] && res1[2]==num0[2] &&
```

```
res1[3]==num0[3] && res1[4]==num0[4] && res1[5]==num0[5]){
  verr1=1;
  resul1=0;
  vacr1=0;
}
else{
  verr1=0;
}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////RESULTADO
RB////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

  if(res2[0]==0 && res2[1]==0 && res2[2]==0 && res2[3]==0 && res2[4]==0 &&
res2[5]==0) {
  verr2=0;
  vacr2=1;
}

  else if(res2[0]==num1[0] && res2[1]==num1[1] && res2[2]==num1[2] &&
res2[3]==num1[3] && res2[4]==num1[4] && res2[5]==num1[5]){
  verr2=1;
  resul2=1;
  vacr2=0;
//  Serial.print(res2[0]);
//  Serial.print(res2[1]);
//  Serial.print(res2[2]);
//  Serial.print(res2[3]);
//  Serial.print(res2[4]);
//  Serial.println(res2[5]);
//  Serial.println("Digito RB");
//  Serial.println(resul2);
}

  else if(res2[0]==num2[0] && res2[1]==num2[1] && res2[2]==num2[2] &&
res2[3]==num2[3] && res2[4]==num2[4] && res2[5]==num2[5]){
  verr2=1;
  resul2=2;
  vacr2=0;
}

  else if(res2[0]==num3[0] && res2[1]==num3[1] && res2[2]==num3[2] &&
```



```
res2[3]==num3[3] && res2[4]==num3[4] && res2[5]==num3[5]){
    verr2=1;
    resul2=3;
    vacr2=0;
}

else if(res2[0]==num4[0] && res2[1]==num4[1] && res2[2]==num4[2] &&
res2[3]==num4[3] && res2[4]==num4[4] && res2[5]==num4[5]){
    verr2=1;
    resul2=4;
    vacr2=0;
}

else if(res2[0]==num5[0] && res2[1]==num5[1] && res2[2]==num5[2] &&
res2[3]==num5[3] && res2[4]==num5[4] && res2[5]==num5[5]){
    verr2=1;
    resul2=5;
    vacr2=0;
}

else if(res2[0]==num6[0] && res2[1]==num6[1] && res2[2]==num6[2] &&
res2[3]==num6[3] && res2[4]==num6[4] && res2[5]==num6[5]){
    verr2=1;
    resul2=6;
    vacr2=0;
}

else if(res2[0]==num7[0] && res2[1]==num7[1] && res2[2]==num7[2] &&
res2[3]==num7[3] && res2[4]==num7[4] && res2[5]==num7[5]){
    verr2=1;
    resul2=7;
    vacr2=0;
}

else if(res2[0]==num8[0] && res2[1]==num8[1] && res2[2]==num8[2] &&
res2[3]==num8[3] && res2[4]==num8[4] && res2[5]==num8[5]){
    verr2=1;
    resul2=8;
    vacr2=0;
}

else if(res2[0]==num9[0] && res2[1]==num9[1] && res2[2]==num9[2] &&
res2[3]==num9[3] && res2[4]==num9[4] && res2[5]==num9[5]){
    verr2=1;
    resul2=9;
    vacr2=0;
}
```

```

}

else if(res2[0]==num0[0] && res2[1]==num0[1] && res2[2]==num0[2] &&
res2[3]==num0[3] && res2[4]==num0[4] && res2[5]==num0[5]){
  verr2=1;
  resul2=0;
  vacr2=0;
}
else{
  verr2=0;
}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////RESULTADO
RC////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

if(res3[0]==0 && res3[1]==0 && res3[2]==0 && res3[3]==0 && res3[4]==0) {
  verr3=0;
  vacr3=1;
}

else if(res3[0]==num1[0] && res3[1]==num1[1] && res3[2]==num1[2] &&
res3[3]==num1[3] && res3[4]==num1[4]) {
  verr3=1;
  resul3=1;
  vacr3=0;
// Serial.print(res3[0]);
// Serial.print(res3[1]);
// Serial.print(res3[2]);
// Serial.print(res3[3]);
// Serial.print(res3[4]);
// Serial.println(res3[5]);
// Serial.println("Digito RC");
// Serial.println(resul2);
}

else if(res3[0]==num2[0] && res3[1]==num2[1] && res3[2]==num2[2] &&
res3[3]==num2[3] && res3[4]==num2[4]) {
  verr3=1;
  resul3=2;
  vacr3=0;
}

```

```
else if(res3[0]==num3[0] && res3[1]==num3[1] && res3[2]==num3[2] &&
res3[3]==num3[3] && res3[4]==num3[4]){
    verr3=1;
    resul3=3;
}

else if(res3[0]==num4[0] && res3[1]==num4[1] && res3[2]==num4[2] &&
res3[3]==num4[3] && res3[4]==num4[4]){
    verr3=1;
    resul3=4;
    vacr3=0;
}

else if(res3[0]==num5[0] && res3[1]==num5[1] && res3[2]==num5[2] &&
res3[3]==num5[3] && res3[4]==num5[4]){
    verr3=1;
    resul3=5;
    vacr3=0;
}

else if(res3[0]==num6[0] && res3[1]==num6[1] && res3[2]==num6[2] &&
res3[3]==num6[3] && res3[4]==num6[4]){
    verr3=1;
    resul3=6;
    vacr3=0;
}

else if(res3[0]==num7[0] && res3[1]==num7[1] && res3[2]==num7[2] &&
res3[3]==num7[3] && res3[4]==num7[4]){
    verr3=1;
    resul3=7;
    vacr3=0;
}

else if(res3[0]==num8[0] && res3[1]==num8[1] && res3[2]==num8[2] &&
res3[3]==num8[3] && res3[4]==num8[4]){
    verr3=1;
    resul3=8;
    vacr3=0;
}

else if(res3[0]==num9[0] && res3[1]==num9[1] && res3[2]==num9[2] &&
res3[3]==num9[3] && res3[4]==num9[4]){
    verr3=1;
    resul3=9;
    vacr3=0;
}
```

```
}

else if(res3[0]==num0[0] && res3[1]==num0[1] && res3[2]==num0[2] &&
res3[3]==num0[3] && res3[4]==num0[4]){
    verr3=1;
    resul3=0;
    vacr3=0;
}
else{
    verr3=0;
}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////OPERADOR////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

if(ope[0]==vsuma[0] && ope[1]==vsuma[1] && ope[2]==vsuma[2] && ope[3]==vsuma[3]){
    finop[0]='+';
}

else if(ope[0]==vresta[0] && ope[1]==vresta[1] && ope[2]==vresta[2] &&
ope[3]==vresta[3]){
    finop[0]='-';
}

else {
    finop[0]='N';
}

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////OPERACIONES////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

if (Serial.available()){

    if (Serial.read()=='X'){

        if (ver1B==1 && ver1A==1)
        {
            ing1=(digitola*10)+digitolb;
```

```

Serial.print(ing1);
}
else if (vac1B==1 && ver1A==1){
  ing1=digitola;
Serial.print(ing1);
}
else if (vac1A==1){
  ver1B=0;
Serial.print('N');
}
else if (ver1A==0){
  ver1B=0;
Serial.print('N');
}
else {
Serial.print('N');
}

Serial.print(',');
Serial.print(finop[0]);
Serial.print(',');

if (ver2B==1 && ver2A==1)
{
  ing2=(digito2a*10)+digito2b;
  Serial.print(ing2);
}
else if (ver2A==1 && vac2B==1){
  ing2=digito2a;
Serial.print(ing2);
}
else if (vac2A==1){
  ver2B=0;
Serial.print('N');
}
else if (ver2A==0){
  ver2B=0;
Serial.print('N');
}
else {
Serial.print('N');
}
Serial.print(',');

if (verr3==1 && verr2==1 && verr1==1)
{

```

```
sol=(resul1*100)+(resul2*10)+resul3;
Serial.println(sol);
}
else if (vacr3==1 && verr2==1 && verr1==1){
sol=(resul1*10)+resul2;
Serial.println(sol);
}
else if (vacr3==1 && vacr2==1 && verr1==1){
sol=resul1;
Serial.println(sol);
}
else{
Serial.println('N');
}
}
}
```

Anexo C

Algoritmo de corrección

```

#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <Encoder.h>

// Definir los pines del encoder rotativo
#define CLK 2
#define DT 3
#define SW 4

// Inicializar el encoder
Encoder myEnc(CLK, DT);

// Crear una instancia del controlador PWM
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

#define SERVOMIN 600 // Valor mínimo del pulso del servo
#define SERVOMAX 150 // Valor máximo del pulso del servo

// Variables del menú
int menuIndex = 0;
String menuItems[] = { "Practica", "Entretenimiento" };
const int numMenuItems = 2;
bool buttonPressed = false;
int previousMenuIndex = -1; // Para rastrear cambios en el menú

// Variables para los datos recibidos
String cifra1, signo, cifra2, respuesta;
int expectedRes; // Variable para la respuesta correcta

// Variables para el submenú de práctica
int practicaMenuIndex = 0;
String practicaMenuItems[] = { "Validar", "Salir" };
const int numPracticaMenuItems = 2;

int digits[3] = { 0, 0, 0 };

void setup() {
  pinMode(SW, INPUT_PULLUP); // Configurar el botón del encoder como entrada
                             // con resistencia pull-up
  Serial.begin(9600); // Serial para depuración
  Serial1.begin(9600); // Serial1 para comunicación con el otro Arduino
  Serial2.begin(9600); // Serial2 para el módulo Bluetooth

  pwm.begin();

```



```

pwm.setPWMPFreq(60); // Frecuencia de 60 Hz para los servos

Serial.println("Menu Inicializado");
displayMenu();
encero();
}

void loop() {
// Leer el valor del encoder
long newPosition = myEnc.read();
static long oldPosition = -999;

if (newPosition != oldPosition) {
    oldPosition = newPosition;
    // Calcular el índice del menú en función de la posición del encoder
    menuIndex = (newPosition / 4) % numMenuItems; // Ajusta el divisor según
la sensibilidad que desees
    if (menuIndex < 0) {
        menuIndex += numMenuItems;
    }
    if (menuIndex != previousMenuIndex) {
        displayMenu();
        sendMenuAudio(); // Enviar el mensaje correspondiente al módulo
Bluetooth
        previousMenuIndex = menuIndex;
    }
}

// Leer el estado del botón del encoder
if (digitalRead(SW) == LOW) {
    if (!buttonPressed) {
        buttonPressed = true;
        selectMenuItem();
    }
} else {
    buttonPressed = false;
}

delay(100); // Pequeño retardo para evitar lecturas erráticas
}

void displayMenu() {
    Serial.println("----- MENU -----");
    for (int i = 0; i < numMenuItems; i++) {

```

```

if (i == menuIndex) {
    Serial.print("> ");
} else {
    Serial.print(" ");
}
Serial.println(menuItems[i]);
}
}

void sendMenuAudio() {
String message = menuItems[menuIndex];
Serial2.println(message);
Serial.print("Enviando por Bluetooth: ");
Serial.println(message);
}

void selectMenuItem() {
Serial.print("Seleccionado: ");
Serial.println(menuItems[menuIndex]);

// Enviar el menú seleccionado al módulo Bluetooth
String message = "Has elegido: " + menuItems[menuIndex];
Serial2.println(message);
delay(500);

// Si se selecciona "Practica", activar el modo de práctica
if (menuItems[menuIndex] == "Practica") {
    practicaMode();
}
}

void practicaMode() {
Serial.println("Modo Practica Iniciado");
Serial2.println("Modo Practica Iniciado");

while (true) {
    // Leer el valor del encoder en el modo de práctica
    long newPosition = myEnc.read();
    static long oldPosition = -999;

    if (newPosition != oldPosition) {
        oldPosition = newPosition;
        // Calcular el índice del menú de práctica en función de la posición
del encoder

```

```

    practicaMenuIndex = (newPosition / 4) % numPracticaMenuItems; //
Ajusta el divisor según la sensibilidad que desees
    if (practicaMenuIndex < 0) {
        practicaMenuIndex += numPracticaMenuItems;
    }
    displayPracticaMenu(practicaMenuIndex);
}

// Leer el estado del botón del encoder en el modo de práctica
if (digitalRead(SW) == LOW) {
    if (!buttonPressed) {
        buttonPressed = true;
        // Manejar la selección en el submenú de práctica
        // handlePracticaSelection(practicaMenuIndex);
        Serial1.println("X");
        if (practicaMenuIndex == 1) { // Si selecciona "Salir"
            Serial.println("Saliendo del Modo Practica");
            Serial2.println("Saliendo del Modo Practica");
            break; // Salir del bucle while y volver al menú principal
        }
        delay(1000);
        //handlePracticaSelection(practicaMenuIndex);
    }
} else {
    buttonPressed = false;
}

// Comprobar si hay datos seriales disponibles en Serial1
if (Serial1.available() > 0) {
    String data = Serial1.readStringUntil('\n'); // Leer la cadena de
datos hasta el final de la línea
    parseData(data); // Llamar a la función
para analizar y procesar los datos recibidos
    handlePracticaSelection(practicaMenuIndex);
}

delay(100); // Pequeño retardo para evitar lecturas erráticas
}

Serial.println("Volviendo al Menu Principal");
Serial2.println("Volviendo al Menu Principal");
displayMenu();
}

```

```

void displayPracticaMenu(int practicaMenuIndex) {
    Serial.println("----- MENU PRACTICA -----");
    for (int i = 0; i < numPracticaMenuItems; i++) {
        if (i == practicaMenuIndex) {
            Serial.print("> ");
        } else {
            Serial.print("  ");
        }
        Serial.println(practicaMenuItems[i]);
        Serial2.println(practicaMenuItems[i]);
    }
}

void handlePracticaSelection(int practicaMenuIndex) {
    if (practicaMenuIndex == 0) { // Si selecciona "Validar"

        // Enviar los datos cifrados por comas al módulo Bluetooth
        String message = "#," + cifra1 + "," + signo + "," + cifra2 + "," +
respuesta;
        Serial2.println(message);
        Serial.print("Enviando datos por Bluetooth: ");
        Serial.println(message);

        // Llamar a la función de corrección si es necesario
        validateData();
    }
}

void parseData(String data) {
    int firstComma = data.indexOf(',');
    int secondComma = data.indexOf(',', firstComma + 1);
    int thirdComma = data.indexOf(',', secondComma + 1);

    cifra1 = data.substring(0, firstComma);
    signo = data.substring(firstComma + 1, secondComma);
    cifra2 = data.substring(secondComma + 1, thirdComma);
    respuesta = data.substring(thirdComma + 1);

    Serial.println("Datos recibidos:");
    Serial.println("Cifra 1: " + cifra1);
    Serial.println("Signo: " + signo);
    Serial.println("Cifra 2: " + cifra2);
    Serial.println("Respuesta: " + respuesta);
}

```

```

//Mover los servos basados en los datos recibidos
//moveServos();
}

void validateData() {
    bool error = false;

    int num1 = cifra1.toInt();
    int num2 = cifra2.toInt();
    int res = respuesta.toInt();
    expectedRes = (signo == "+") ? (num1 + num2) : (num1 - num2);

    if (res != expectedRes) {
        error = true;
    }

    if (error) {
        // Mover los servos para mostrar la respuesta correcta en Braille
        moveServosToCorrectAnswer();
        delay(15000); // Esperar 7 segundos
        encero(); // Limpiar la respuesta
    }
}

void moveServos() {
    // Ejemplo simple: mover servos basados en los valores de cifra1, cifra2 y
    // respuesta
    // Aquí puedes agregar la lógica específica para mover tus servos

    int servoPos1 = map(cifra1.toInt(), 0, 9, SERVOMIN, SERVOMAX); //
    // Asumiendo cifra1 es un dígito del 0 al 9
    int servoPos2 = map(cifra2.toInt(), 0, 9, SERVOMIN, SERVOMAX); //
    // Asumiendo cifra2 es un dígito del 0 al 9
    int servoPosRes = map(respuesta.toInt(), 0, 999, SERVOMIN, SERVOMAX); //
    // Asumiendo respuesta es un número de 0 a 999

    pwm.setPWM(0, 0, servoPos1); // Mover el primer servo
    pwm.setPWM(1, 0, servoPos2); // Mover el segundo servo
    pwm.setPWM(2, 0, servoPosRes); // Mover el tercer servo
}

void moveServosToCorrectAnswer() {
    // Mover los servos para mostrar la respuesta correcta en Braille
    if(expectedRes < 0)

```

```

{
    expectedRes=expectedRes*-1;
}
if (expectedRes < 10) {
    digits[0] = expectedRes;
    matriz0();
} else if (expectedRes < 100) {
    digits[0] = expectedRes / 10;    // Decenas
    digits[1] = expectedRes % 10;   // Unidades
    matriz0();
    matriz1();
} else {
    digits[0] = expectedRes / 100;   // Centenas
    digits[1] = (expectedRes / 10) % 10; // Decenas
    digits[2] = expectedRes % 10;   // Unidades
    matriz0();
    matriz1();
    matriz2();
}
}
}

```

```

void matriz0() {
    if (digits[0] == 1) {
        pwm.setPWM(0, 0, SERVOMAX);
        pwm.setPWM(1, 0, SERVOMIN);
        pwm.setPWM(2, 0, SERVOMIN);
        pwm.setPWM(3, 0, SERVOMIN);
    } else if (digits[0] == 2) {
        pwm.setPWM(0, 0, SERVOMAX);
        pwm.setPWM(1, 0, SERVOMAX);
        pwm.setPWM(2, 0, SERVOMIN);
        pwm.setPWM(3, 0, SERVOMIN);
    } else if (digits[0] == 3) {
        pwm.setPWM(0, 0, SERVOMAX);
        pwm.setPWM(1, 0, SERVOMIN);
        pwm.setPWM(2, 0, SERVOMAX);
        pwm.setPWM(3, 0, SERVOMIN);
    } else if (digits[0] == 4) {
        pwm.setPWM(0, 0, SERVOMAX);
        pwm.setPWM(1, 0, SERVOMIN);
        pwm.setPWM(2, 0, SERVOMAX);
        pwm.setPWM(3, 0, SERVOMAX);
    } else if (digits[0] == 5) {
        pwm.setPWM(0, 0, SERVOMAX);
    }
}

```

```
pwm.setPWM(1, 0, SERVOMIN);
pwm.setPWM(2, 0, SERVOMIN);
pwm.setPWM(3, 0, SERVOMAX);
} else if (digits[0] == 6) {
pwm.setPWM(0, 0, SERVOMAX);
pwm.setPWM(1, 0, SERVOMAX);
pwm.setPWM(2, 0, SERVOMAX);
pwm.setPWM(3, 0, SERVOMIN);
} else if (digits[0] == 7) {
pwm.setPWM(0, 0, SERVOMAX);
pwm.setPWM(1, 0, SERVOMAX);
pwm.setPWM(2, 0, SERVOMAX);
pwm.setPWM(3, 0, SERVOMAX);
} else if (digits[0] == 8) {
pwm.setPWM(0, 0, SERVOMAX);
pwm.setPWM(1, 0, SERVOMAX);
pwm.setPWM(2, 0, SERVOMIN);
pwm.setPWM(3, 0, SERVOMAX);
} else if (digits[0] == 9) {
pwm.setPWM(0, 0, SERVOMIN);
pwm.setPWM(1, 0, SERVOMAX);
pwm.setPWM(2, 0, SERVOMAX);
pwm.setPWM(3, 0, SERVOMIN);
} else if (digits[0] == 0) {
pwm.setPWM(0, 0, SERVOMIN);
pwm.setPWM(1, 0, SERVOMAX);
pwm.setPWM(2, 0, SERVOMAX);
pwm.setPWM(3, 0, SERVOMAX);
}
}
```

```
void matriz1() {
if (digits[1] == 1) {
pwm.setPWM(4, 0, SERVOMAX);
pwm.setPWM(5, 0, SERVOMIN);
pwm.setPWM(6, 0, SERVOMIN);
pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 2) {
pwm.setPWM(4, 0, SERVOMAX);
pwm.setPWM(5, 0, SERVOMAX);
pwm.setPWM(6, 0, SERVOMIN);
pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 3) {
pwm.setPWM(4, 0, SERVOMAX);
```

```

    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 4) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMAX);
} else if (digits[1] == 5) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMIN);
    pwm.setPWM(6, 0, SERVOMIN);
    pwm.setPWM(7, 0, SERVOMAX);
} else if (digits[1] == 6) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 7) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMAX);
} else if (digits[1] == 8) {
    pwm.setPWM(4, 0, SERVOMAX);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMIN);
    pwm.setPWM(7, 0, SERVOMAX);
} else if (digits[1] == 9) {
    pwm.setPWM(4, 0, SERVOMIN);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMIN);
} else if (digits[1] == 0) {
    pwm.setPWM(4, 0, SERVOMIN);
    pwm.setPWM(5, 0, SERVOMAX);
    pwm.setPWM(6, 0, SERVOMAX);
    pwm.setPWM(7, 0, SERVOMAX);
}
}
}

```

```

void matriz2() {
    if (digits[2] == 1) {
        pwm.setPWM(8, 0, SERVOMAX);
    }
}

```



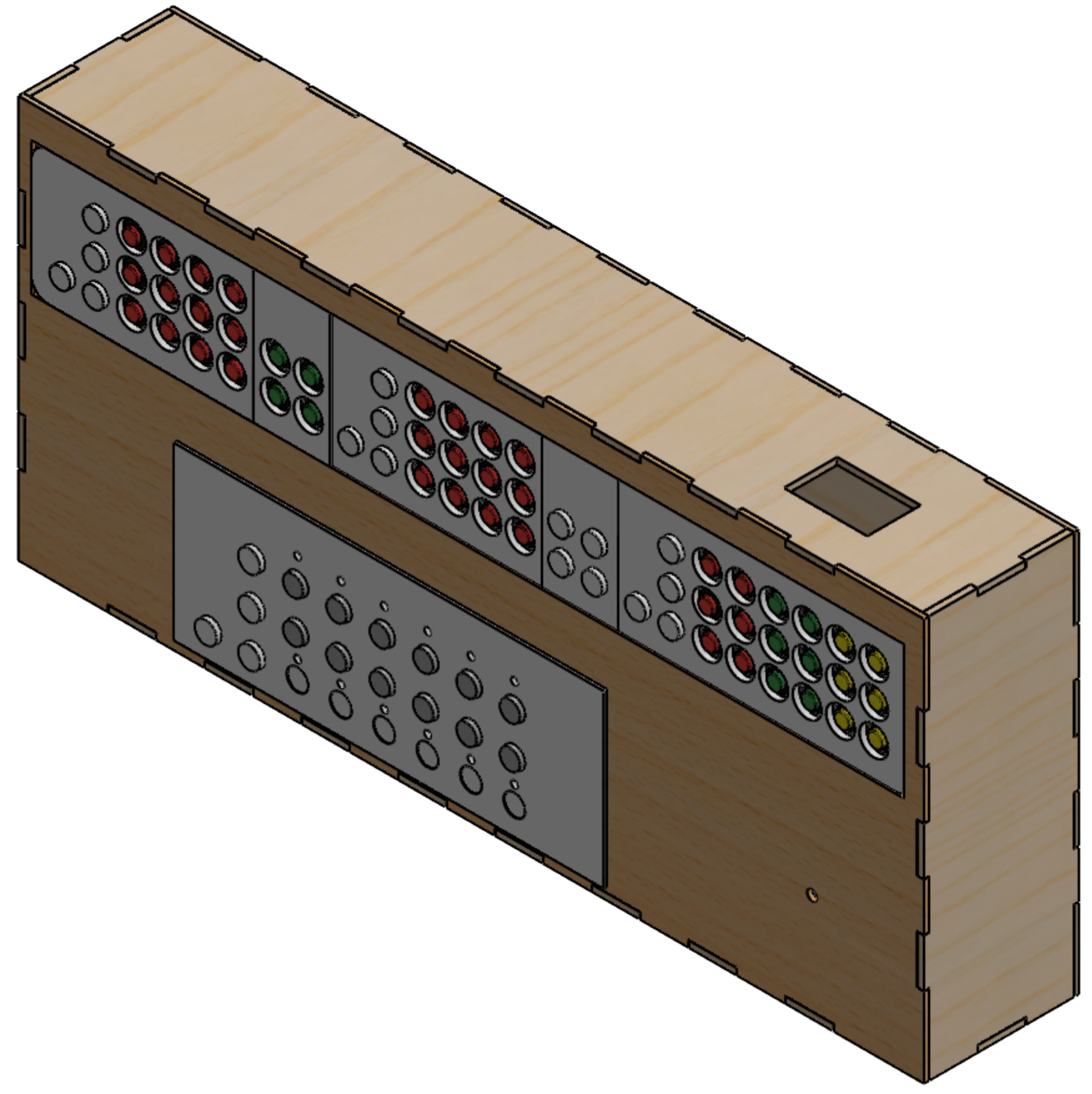
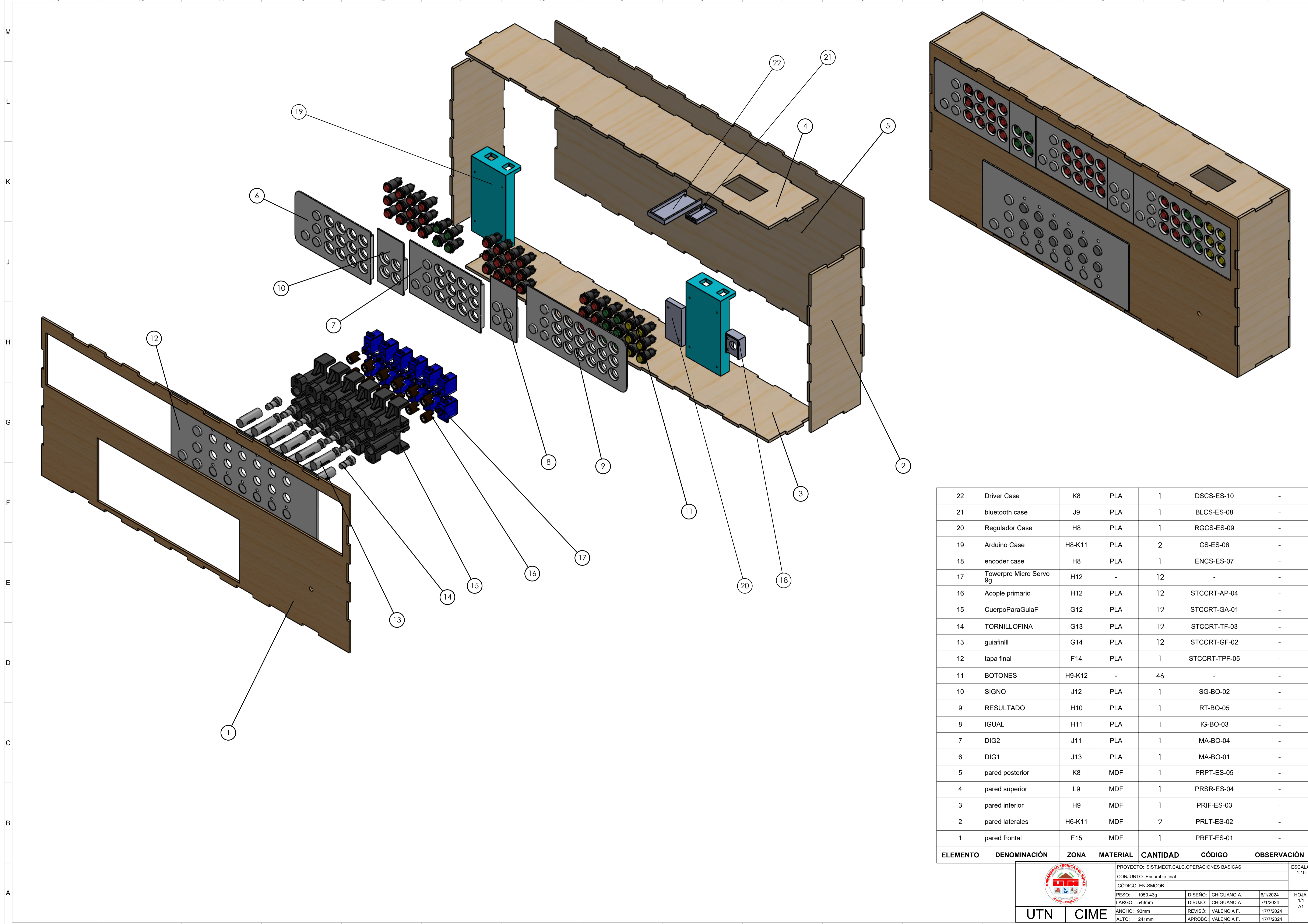
```
pwm.setPWM(9, 0, SERVOMIN);
pwm.setPWM(10, 0, SERVOMIN);
pwm.setPWM(11, 0, SERVOMIN);
} else if (digits[2] == 2) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMAX);
pwm.setPWM(10, 0, SERVOMIN);
pwm.setPWM(11, 0, SERVOMIN);
} else if (digits[2] == 3) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMIN);
pwm.setPWM(10, 0, SERVOMAX);
pwm.setPWM(11, 0, SERVOMIN);
} else if (digits[2] == 4) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMIN);
pwm.setPWM(10, 0, SERVOMAX);
pwm.setPWM(11, 0, SERVOMAX);
} else if (digits[2] == 5) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMIN);
pwm.setPWM(10, 0, SERVOMIN);
pwm.setPWM(11, 0, SERVOMAX);
} else if (digits[2] == 6) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMAX);
pwm.setPWM(10, 0, SERVOMAX);
pwm.setPWM(11, 0, SERVOMIN);
} else if (digits[2] == 7) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMAX);
pwm.setPWM(10, 0, SERVOMAX);
pwm.setPWM(11, 0, SERVOMAX);
} else if (digits[2] == 8) {
pwm.setPWM(8, 0, SERVOMAX);
pwm.setPWM(9, 0, SERVOMAX);
pwm.setPWM(10, 0, SERVOMIN);
pwm.setPWM(11, 0, SERVOMAX);
} else if (digits[2] == 9) {
pwm.setPWM(8, 0, SERVOMIN);
pwm.setPWM(9, 0, SERVOMAX);
pwm.setPWM(10, 0, SERVOMAX);
pwm.setPWM(11, 0, SERVOMIN);
} else if (digits[2] == 0) {
```

```
pwm.setPWM(8, 0, SERVOMIN);  
pwm.setPWM(9, 0, SERVOMAX);  
pwm.setPWM(10, 0, SERVOMAX);  
pwm.setPWM(11, 0, SERVOMAX);  
}  
}
```


```
void encero() {  
  pwm.setPWM(0, 0, SERVOMIN);  
  pwm.setPWM(1, 0, SERVOMIN);  
  pwm.setPWM(2, 0, SERVOMIN);  
  pwm.setPWM(3, 0, SERVOMIN);  
  delay(1000);  
  pwm.setPWM(4, 0, SERVOMIN);  
  pwm.setPWM(5, 0, SERVOMIN);  
  pwm.setPWM(6, 0, SERVOMIN);  
  pwm.setPWM(7, 0, SERVOMIN);  
  delay(1000);  
  pwm.setPWM(8, 0, SERVOMIN);  
  pwm.setPWM(9, 0, SERVOMIN);  
  pwm.setPWM(10, 0, SERVOMIN);  
  pwm.setPWM(11, 0, SERVOMIN);  
  delay(1000);  
}
```

Anexo D

Planos



22	Driver Case	K8	PLA	1	DSCS-ES-10	-
21	bluetooth case	J9	PLA	1	BLCS-ES-08	-
20	Regulador Case	H8	PLA	1	RGCS-ES-09	-
19	Arduino Case	H8-K11	PLA	2	CS-ES-06	-
18	encoder case	H8	PLA	1	ENCS-ES-07	-
17	Towerpro Micro Servo 9g	H12	-	12	-	-
16	Acople primario	H12	PLA	12	STCCRT-AP-04	-
15	CuerpoParaGuiaF	G12	PLA	12	STCCRT-GA-01	-
14	TORNILLOFINA	G13	PLA	12	STCCRT-TF-03	-
13	guiafinill	G14	PLA	12	STCCRT-GF-02	-
12	tapa final	F14	PLA	1	STCCRT-TPF-05	-
11	BOTONES	H9-K12	-	46	-	-
10	SIGNO	J12	PLA	1	SG-BO-02	-
9	RESULTADO	H10	PLA	1	RT-BO-05	-
8	IGUAL	H11	PLA	1	IG-BO-03	-
7	DIG2	J11	PLA	1	MA-BO-04	-
6	DIG1	J13	PLA	1	MA-BO-01	-
5	pared posterior	K8	MDF	1	PRPT-ES-05	-
4	pared superior	L9	MDF	1	PRSR-ES-04	-
3	pared inferior	H9	MDF	1	PRIF-ES-03	-
2	pared laterales	H6-K11	MDF	2	PRLT-ES-02	-
1	pared frontal	F15	MDF	1	PRFT-ES-01	-
ELEMENTO	DENOMINACIÓN	ZONA	MATERIAL	CANTIDAD	CÓDIGO	OBSERVACIÓN



UTN
CIME

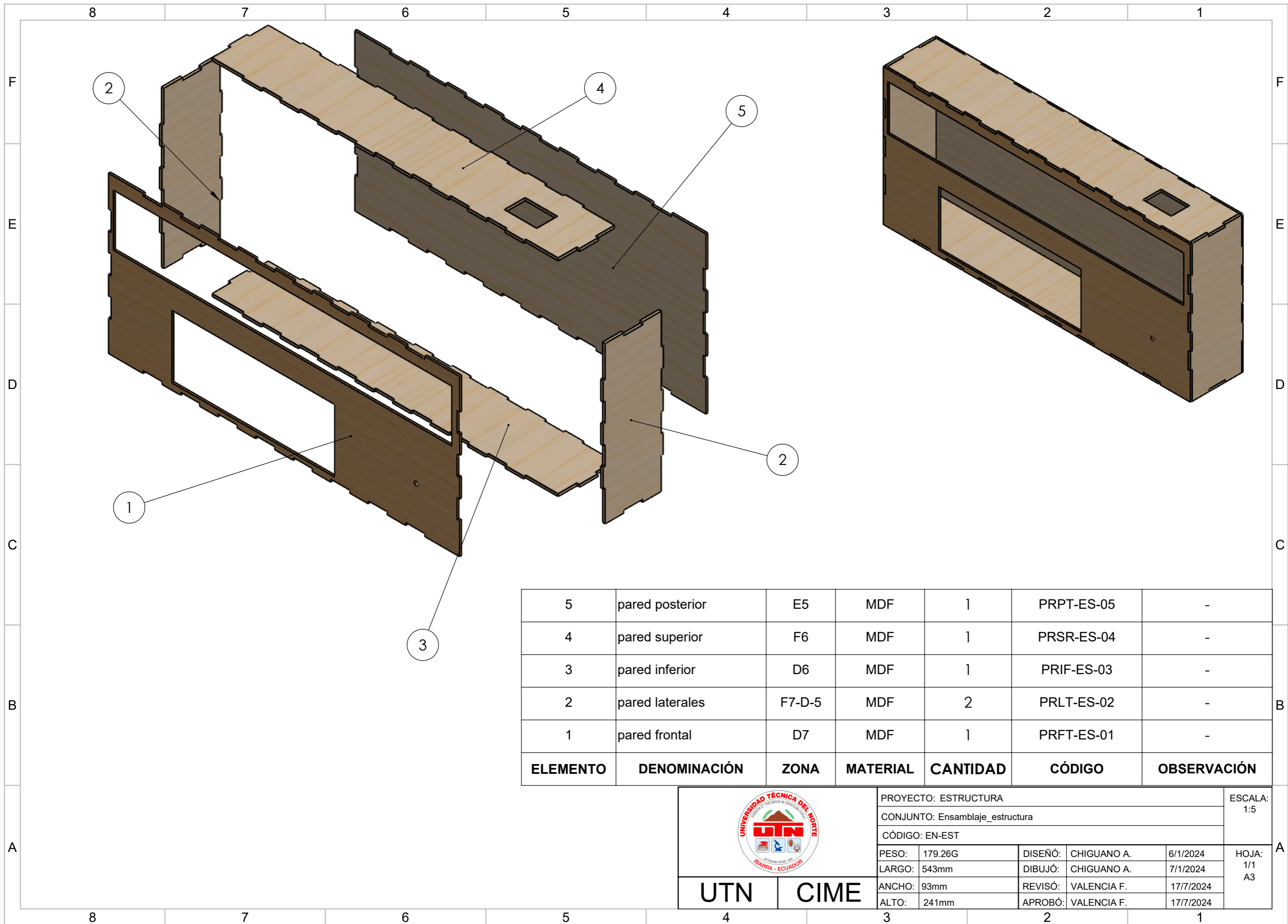
PROYECTO: SIST.MECT.CALC.OPERACIONES BASICAS

CONJUNTO: Ensamble final

CÓDIGO: EN-SMCOB

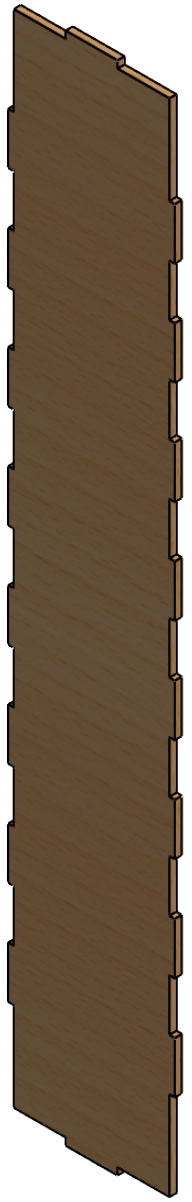
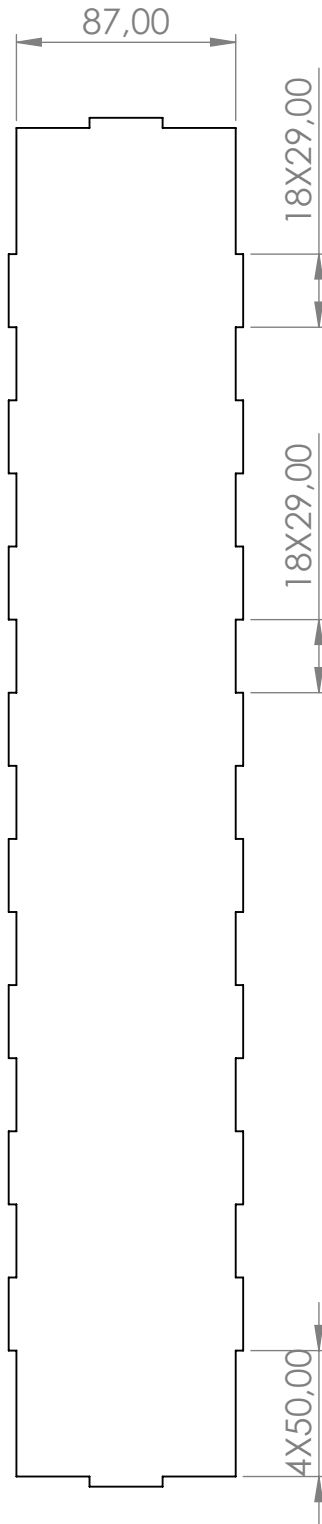
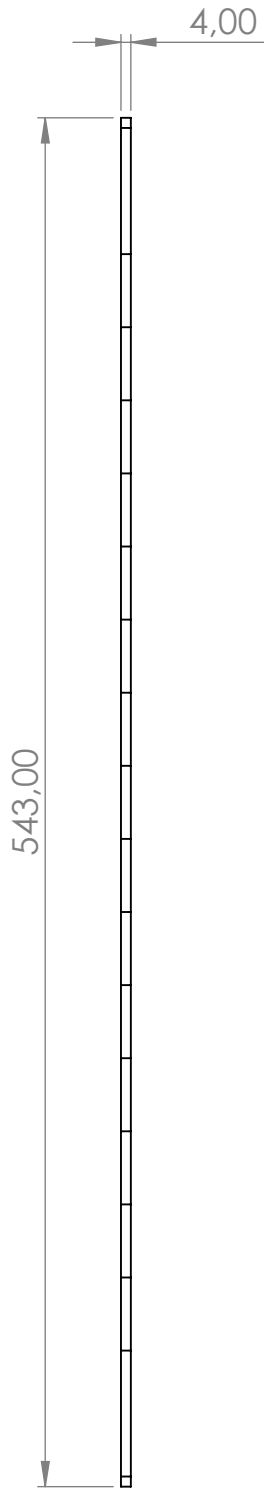
PESO: 1050.43g	DISEÑO: CHIGUANO A.	6/1/2024	HOJA: A1
LARGO: 543mm	DIBUJO: CHIGUANO A.	7/1/2024	1/1
ANCHO: 93mm	REVISO: VALENCIA F.	17/7/2024	
ALTO: 241mm	APROBO: VALENCIA F.	17/7/2024	


ESCALA: 1:10

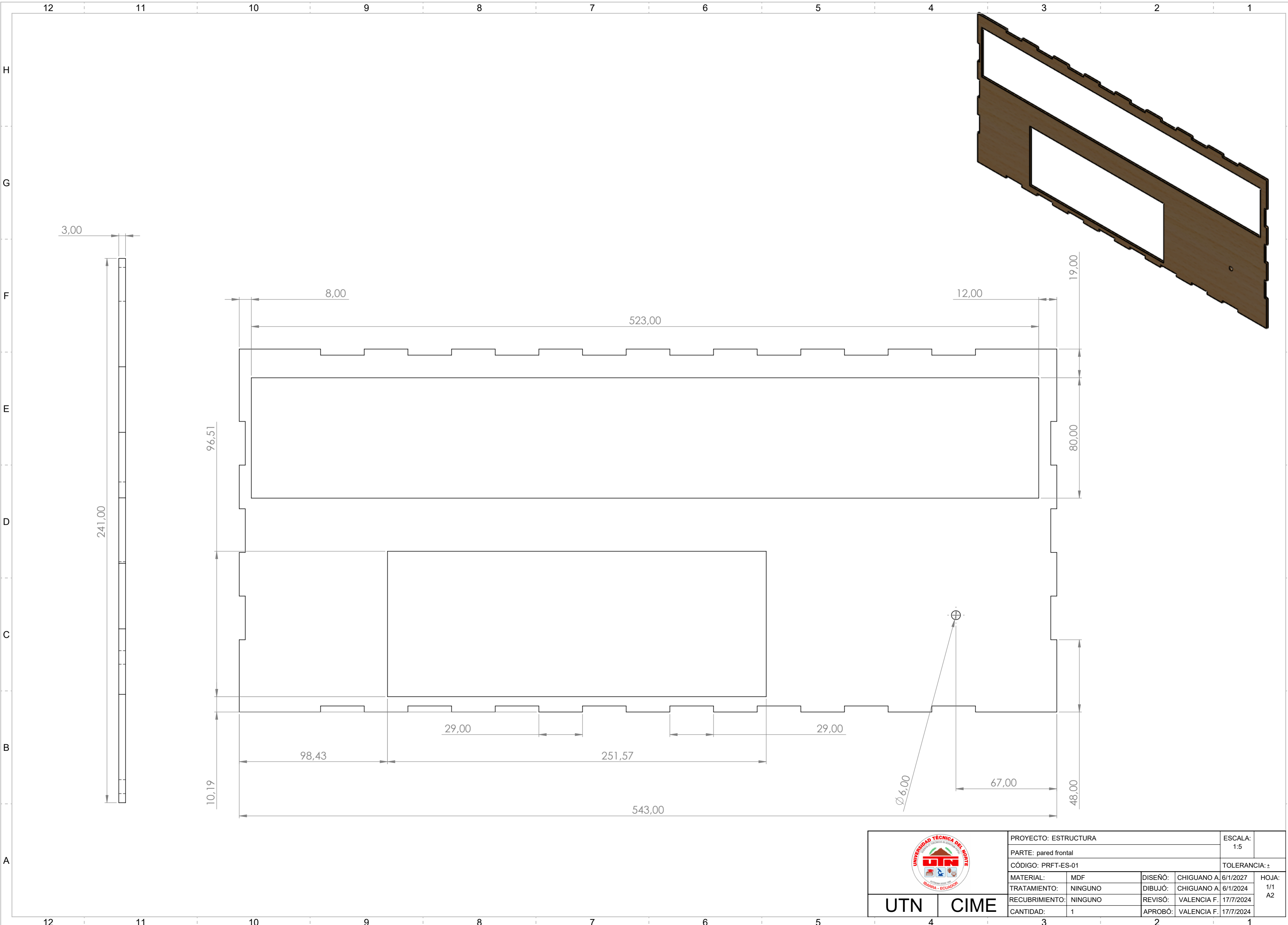


5	wall posterior	E5	MDF	1	PRPT-ES-05	-
4	wall superior	F6	MDF	1	PRSR-ES-04	-
3	wall inferior	D6	MDF	1	PRIF-ES-03	-
2	wall laterales	F7-D-5	MDF	2	PRLT-ES-02	-
1	wall frontal	D7	MDF	1	PRFT-ES-01	-
ELEMENTO	DENOMINACIÓN	ZONA	MATERIAL	CANTIDAD	CÓDIGO	OBSERVACIÓN

	PROYECTO: ESTRUCTURA				ESCALA: 1:5	
	CONJUNTO: Ensamblaje_estructura					
	CÓDIGO: EN-EST					
	PESO:	179.26G	DISEÑO:	CHIGUANO A.	6/1/2024	HOJA: 1/1 A3
	LARGO:	543mm	DIBUJÓ:	CHIGUANO A.	7/1/2024	
ANCHO:	93mm	REVISÓ:	VALENCIA F.	17/7/2024		
ALTO:	241mm	APROBÓ:	VALENCIA F.	17/7/2024		
UTN		CIME				

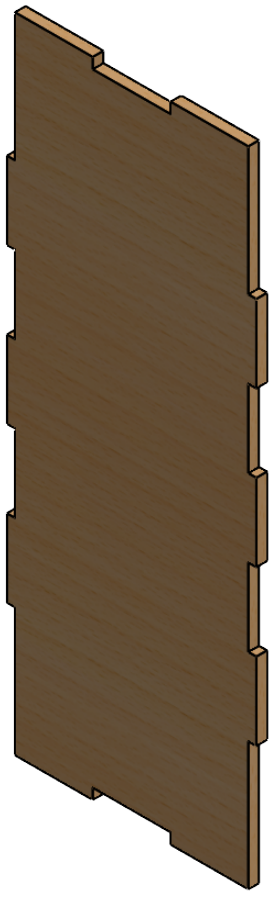
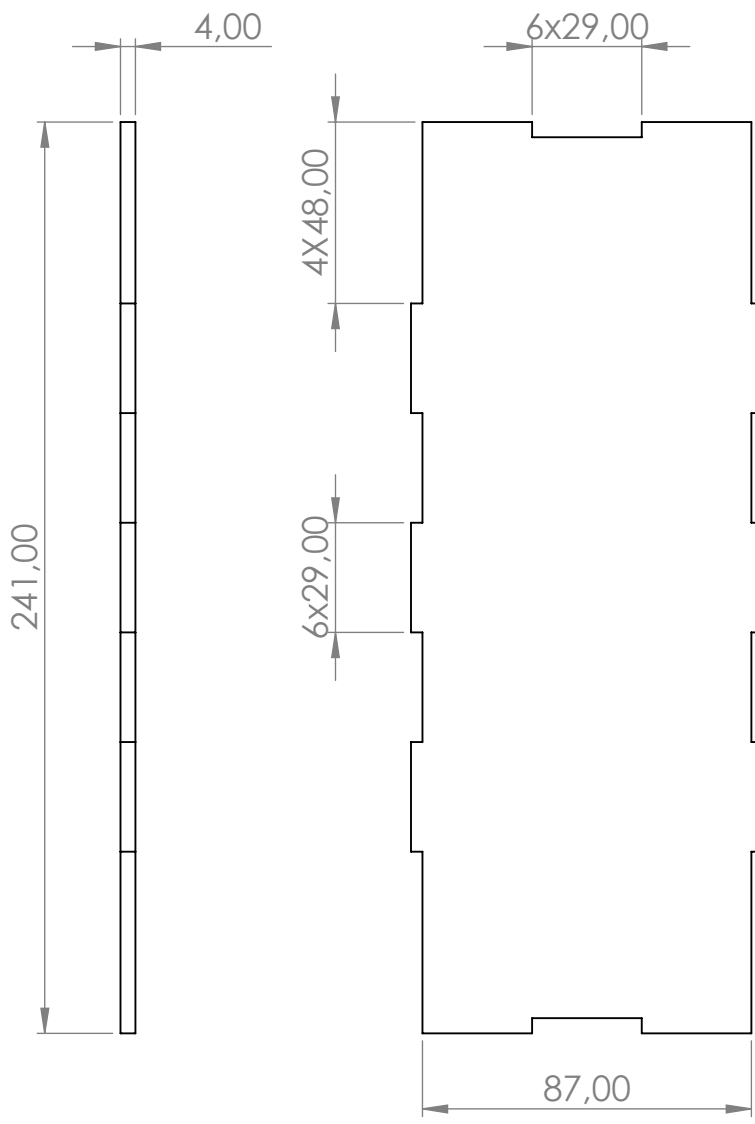


	PROYECTO: ESTRUCTURA				ESCALA:	
	PARTE: pared inferior				1:5	
	CÓDIGO: PRIF-ES-03				TOLERANCIA: ±	
	MATERIAL:	MDF	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					

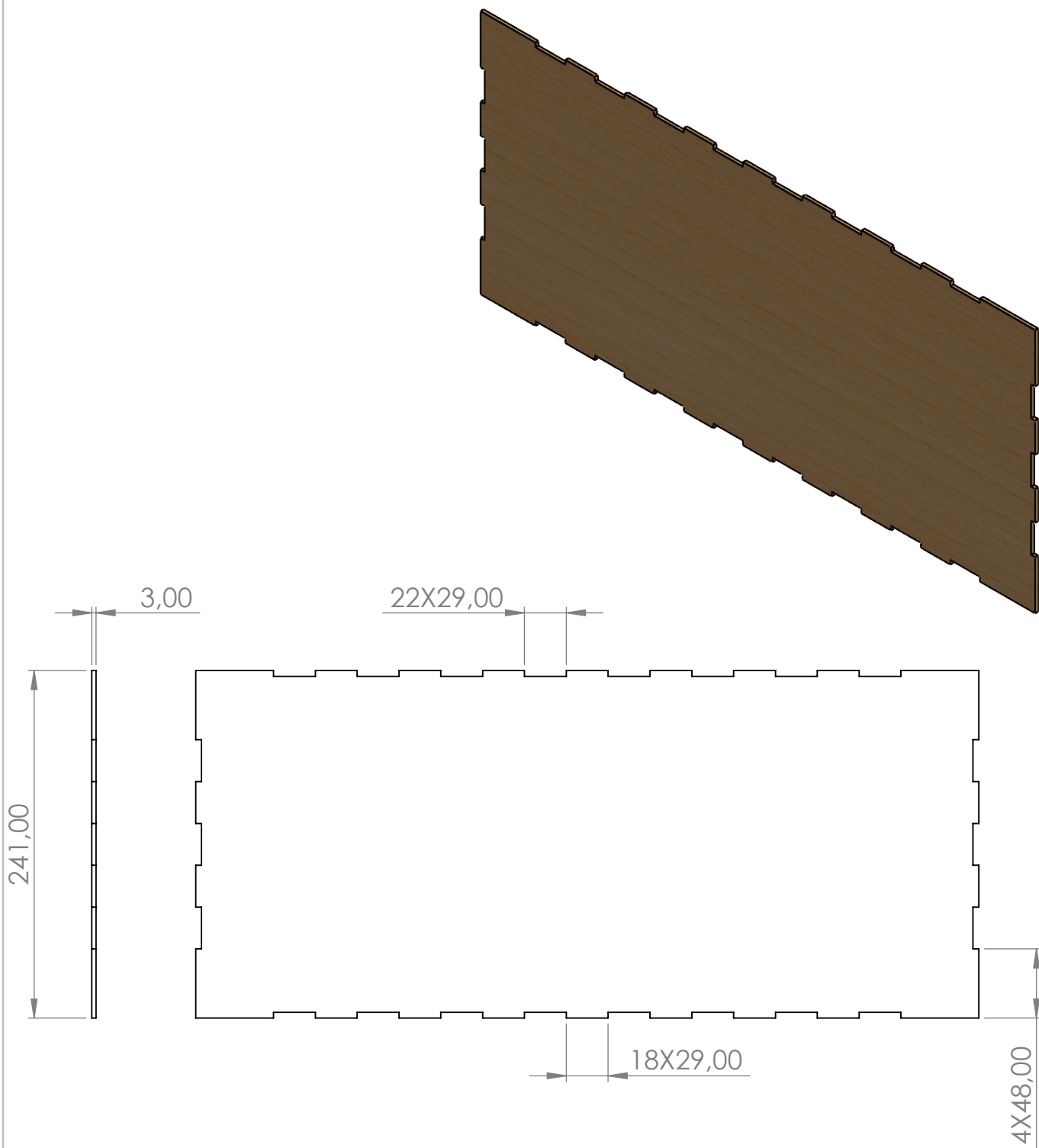



PROYECTO: ESTRUCTURA				ESCALA:	1:5
PARTE: pared frontal				TOLERANCIA: ±	
CÓDIGO: PRFT-ES-01				HOJA:	
MATERIAL:	MDF	DISEÑO:	CHIGUANO A.	6/1/2027	1/1
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	A2
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024	
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024	

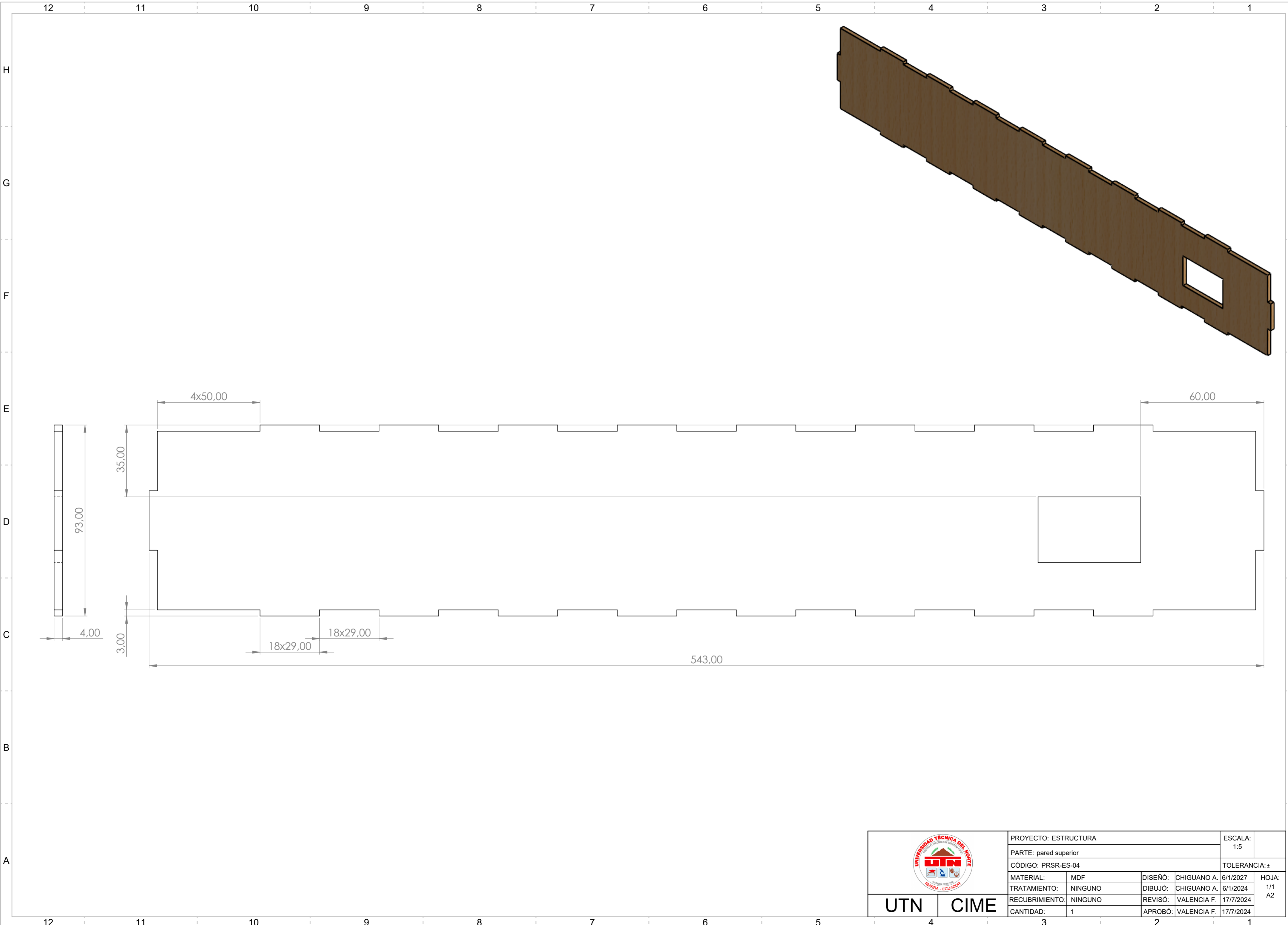
UTN | **CIME**




	PROYECTO: ESTRUCTURA				ESCALA:	
	PARTE: pared laterales				1:2	
	CÓDIGO: PRLT-ES-02				TOLERANCIA: ±	
	MATERIAL:	MDF	DISEÑÓ:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	2	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					

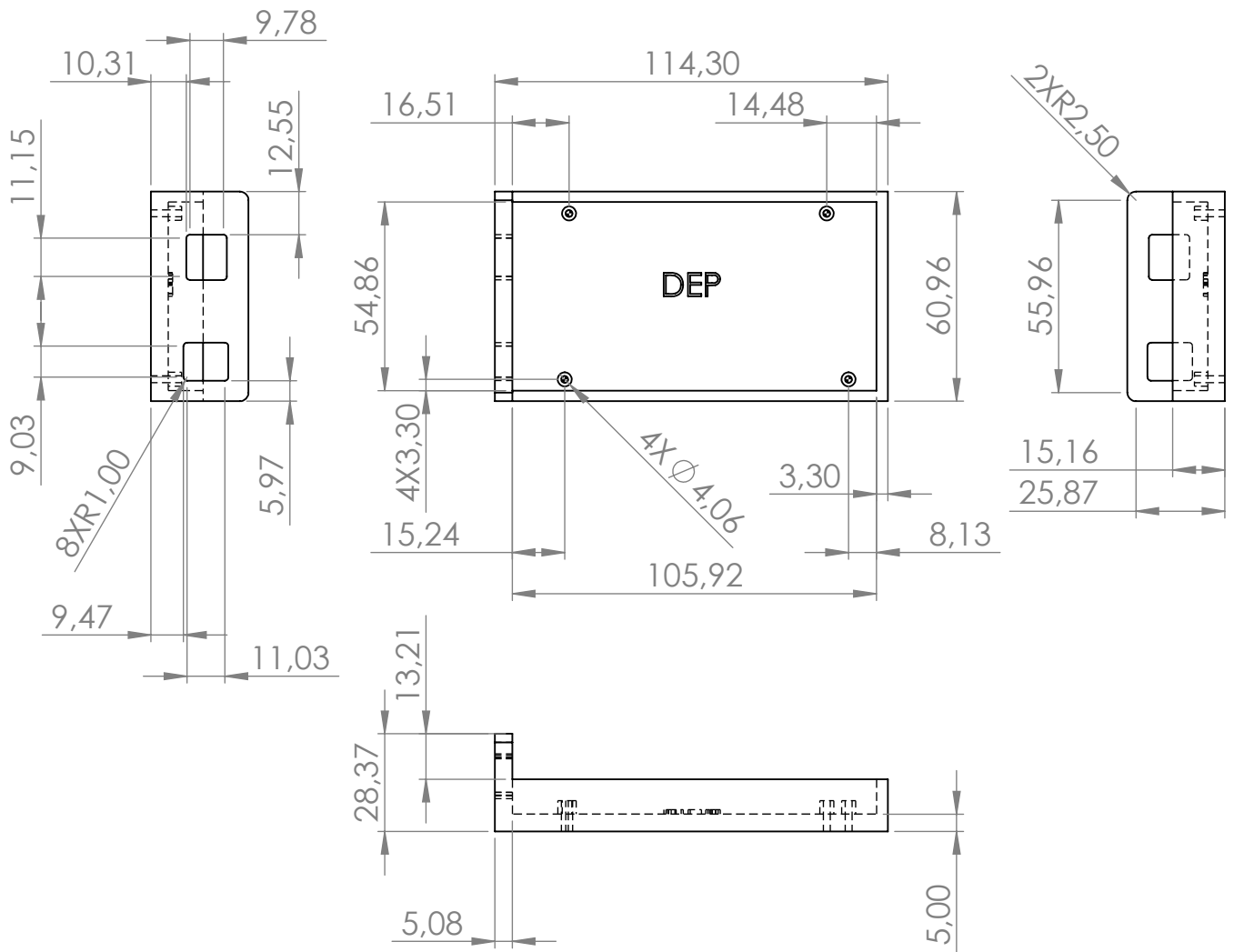
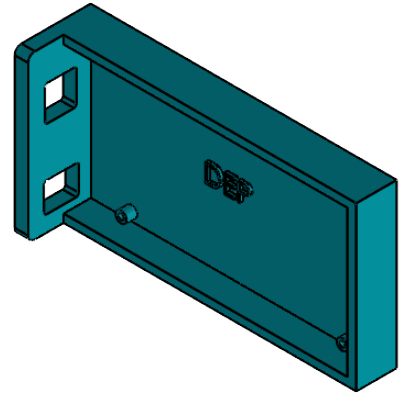


	PROYECTO: ESTRUCTURA				ESCALA:	
	PARTE: pared posterior				1:5	
	CÓDIGO: PRPT-ES-05				TOLERANCIA: ±	
	MATERIAL:	MDF	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					

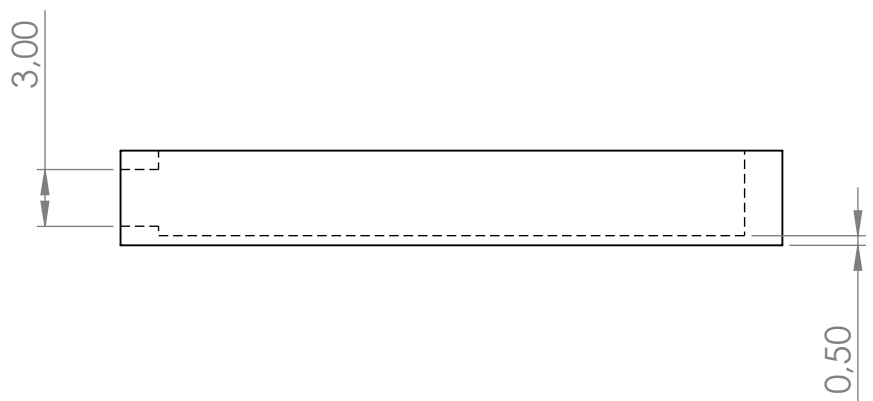
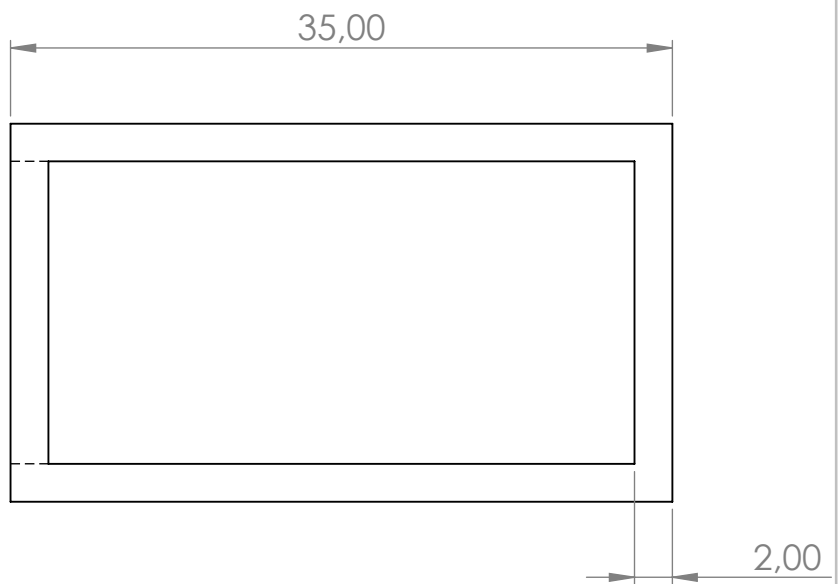
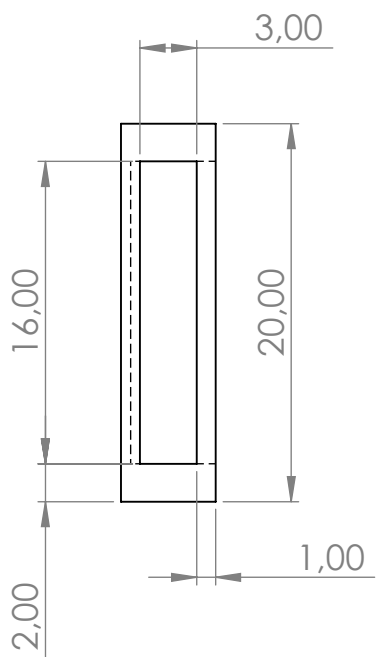
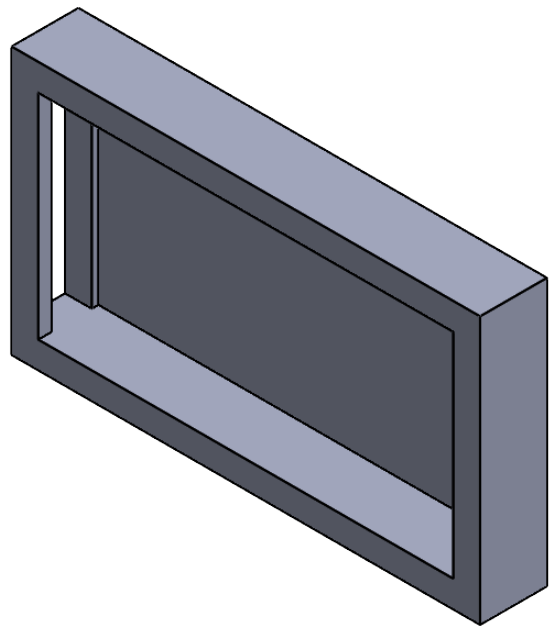


	PROYECTO: ESTRUCTURA				ESCALA:	
	PARTE: pared superior				1:5	
	CÓDIGO: PRSR-ES-04				TOLERANCIA: ±	
	MATERIAL:	MDF	DISEÑO:	CHIGUANO A. 6/1/2027	HOJA:	
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A. 6/1/2024	1/1	A2
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F. 17/7/2024			
CANTIDAD:	1	APROBÓ:	VALENCIA F. 17/7/2024			

UTN CIME



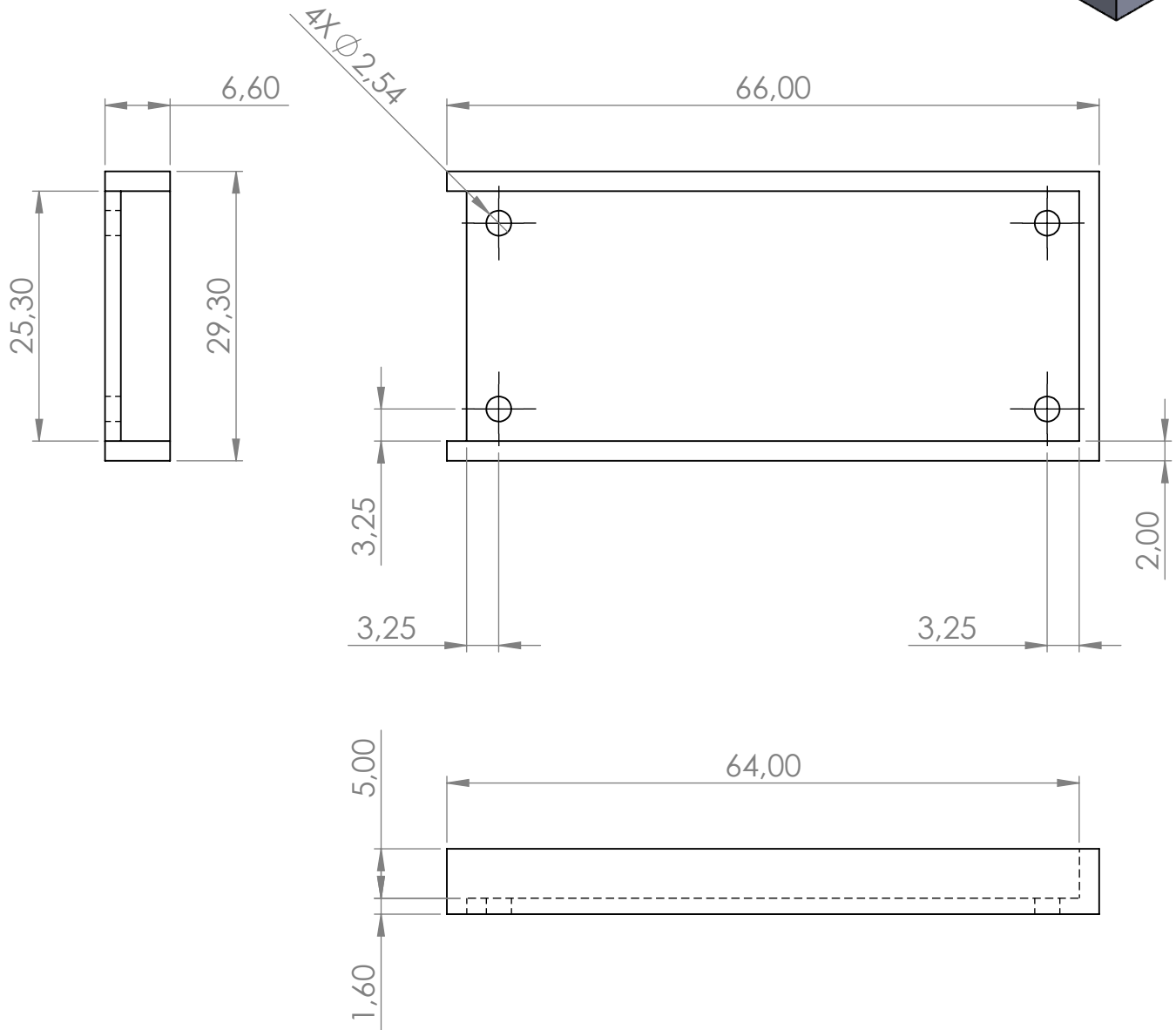
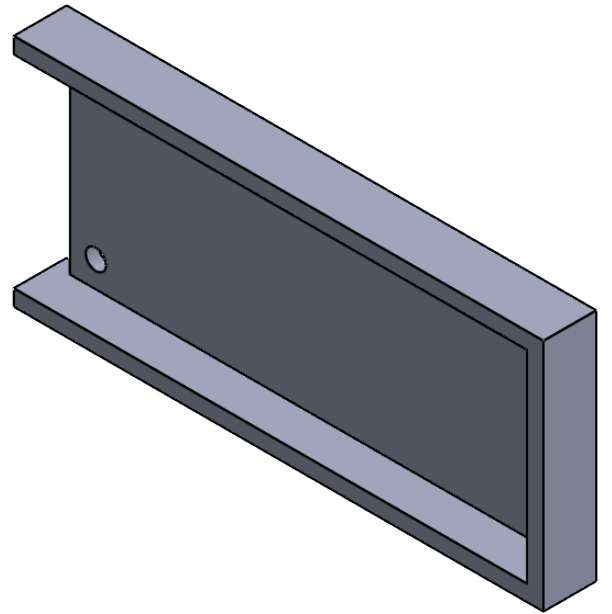
	PROYECTO: Estructura				ESCALA:	
	PARTE: Arduino Case				1:2	
	CÓDIGO: ARCS-ES-06				TOLERANCIA: ±	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	2	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



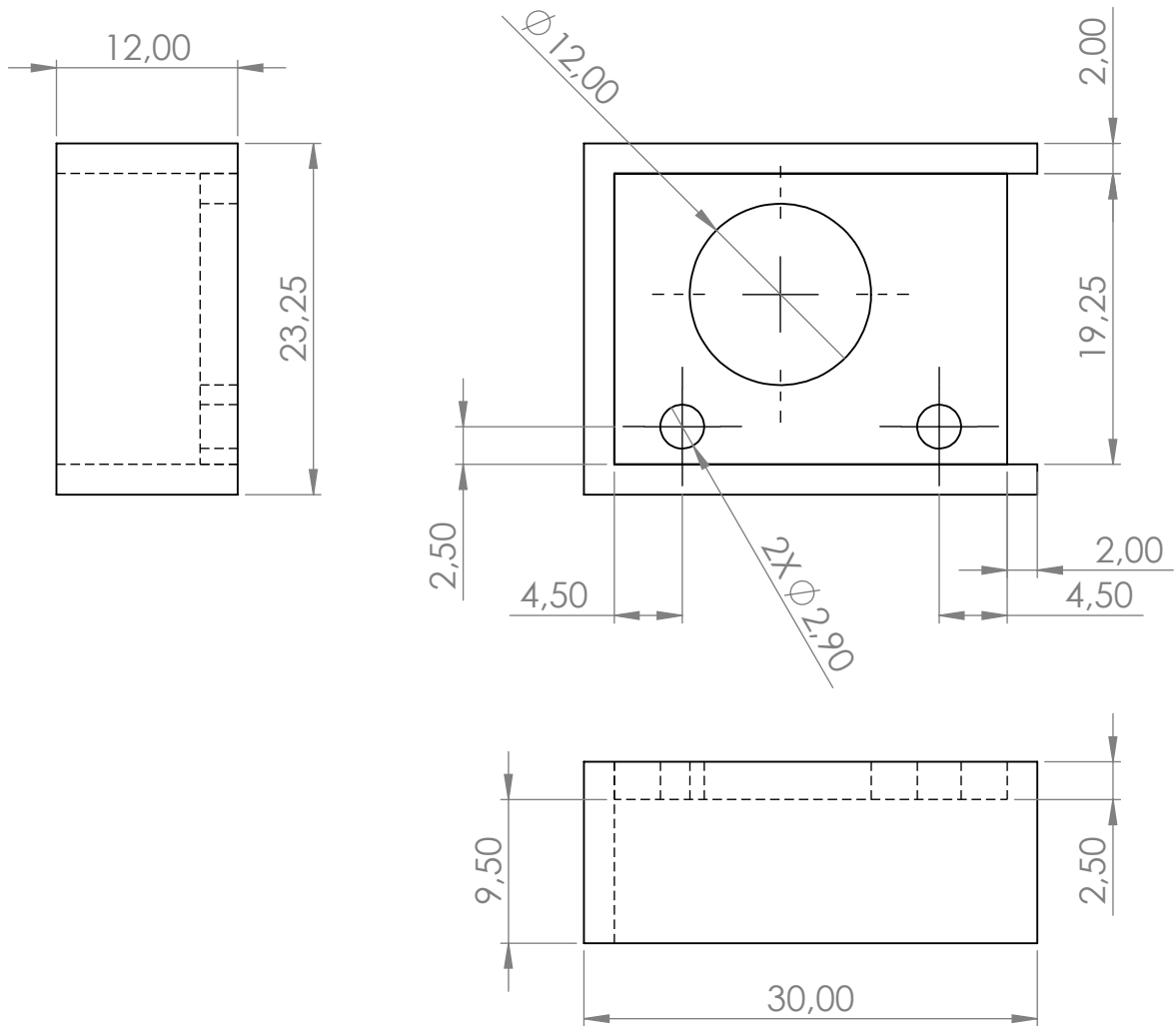
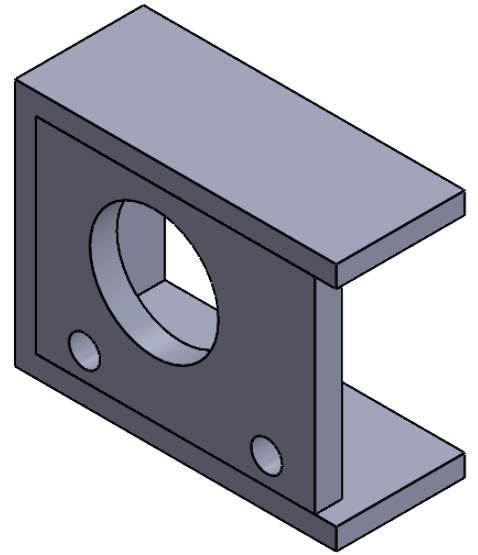
PROYECTO: Estructura				ESCALA: 2:1	
PARTE: bluetooth case					
CÓDIGO: BLCS-ES-08				TOLERANCIA: ±	
MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024	
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024	

UTN

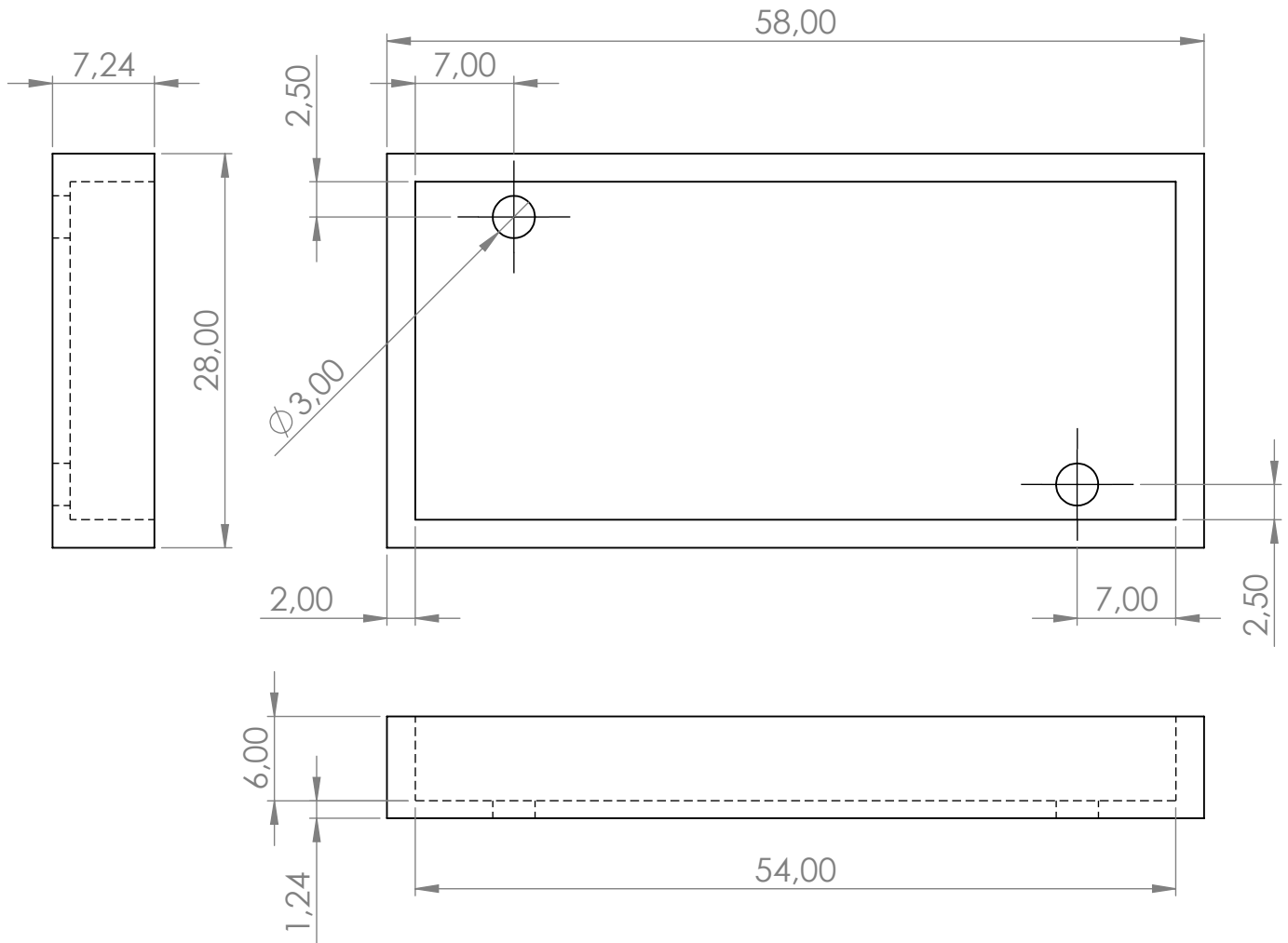
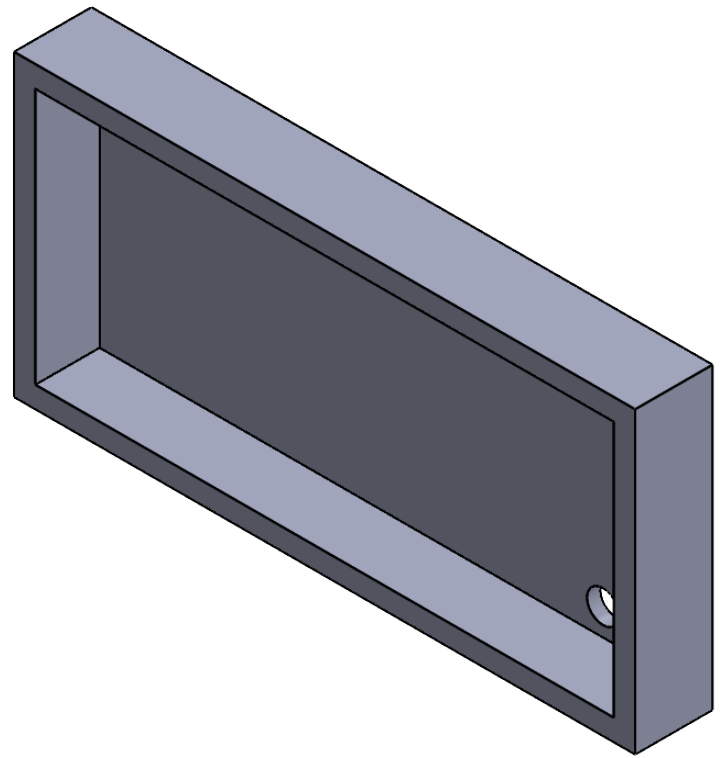
CIME



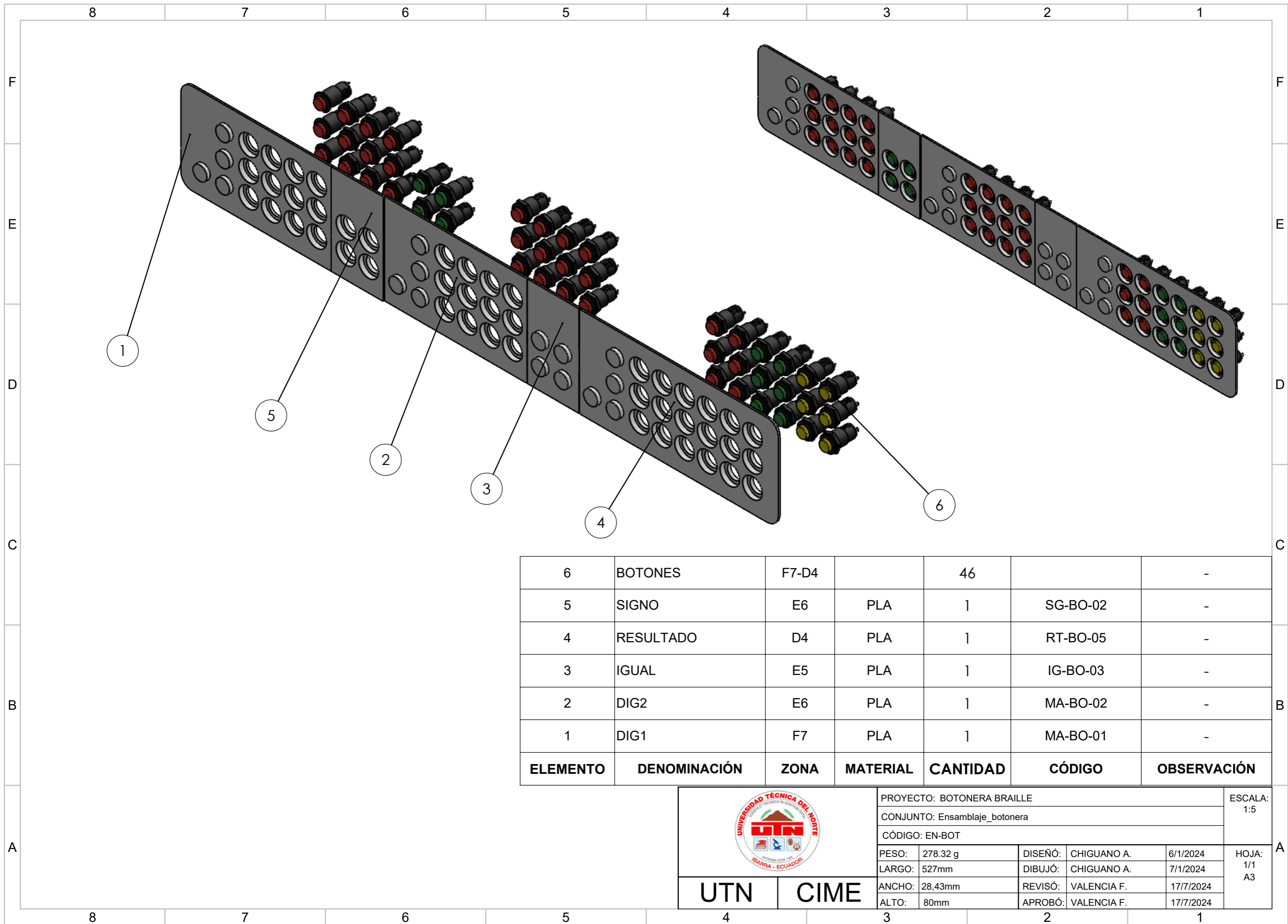
	PROYECTO: Estructura				ESCALA:	
					1:1	
	PARTE: Driver Case				TOLERANCIA: ±	
	CÓDIGO: DSCS-ES-10					
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024		
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



	PROYECTO: Estructura				ESCALA:	
	PARTE: encoder case				2:1	
	CÓDIGO: ENCS-ES-07				TOLERANCIA: \pm	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN		CIME				



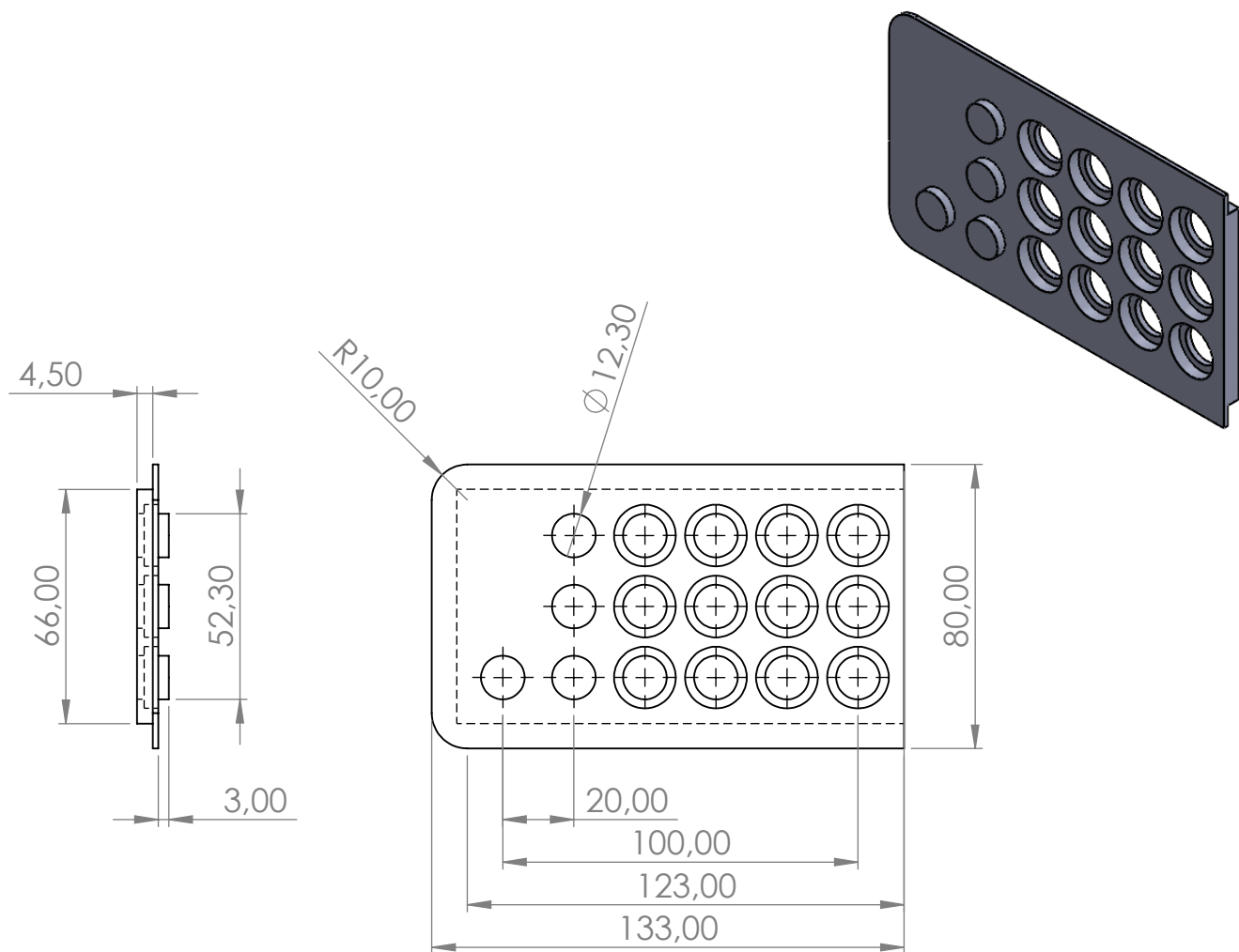
	PROYECTO: Estructura				ESCALA:	
	PARTE: Regulador Case				1:1	
	CÓDIGO: RGCS-ES-09				TOLERANCIA: \pm	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



6	BOTONES	F7-D4		46		-
5	SIGNO	E6	PLA	1	SG-BO-02	-
4	RESULTADO	D4	PLA	1	RT-BO-05	-
3	IGUAL	E5	PLA	1	IG-BO-03	-
2	DIG2	E6	PLA	1	MA-BO-02	-
1	DIG1	F7	PLA	1	MA-BO-01	-
ELEMENTO	DENOMINACIÓN	ZONA	MATERIAL	CANTIDAD	CÓDIGO	OBSERVACIÓN

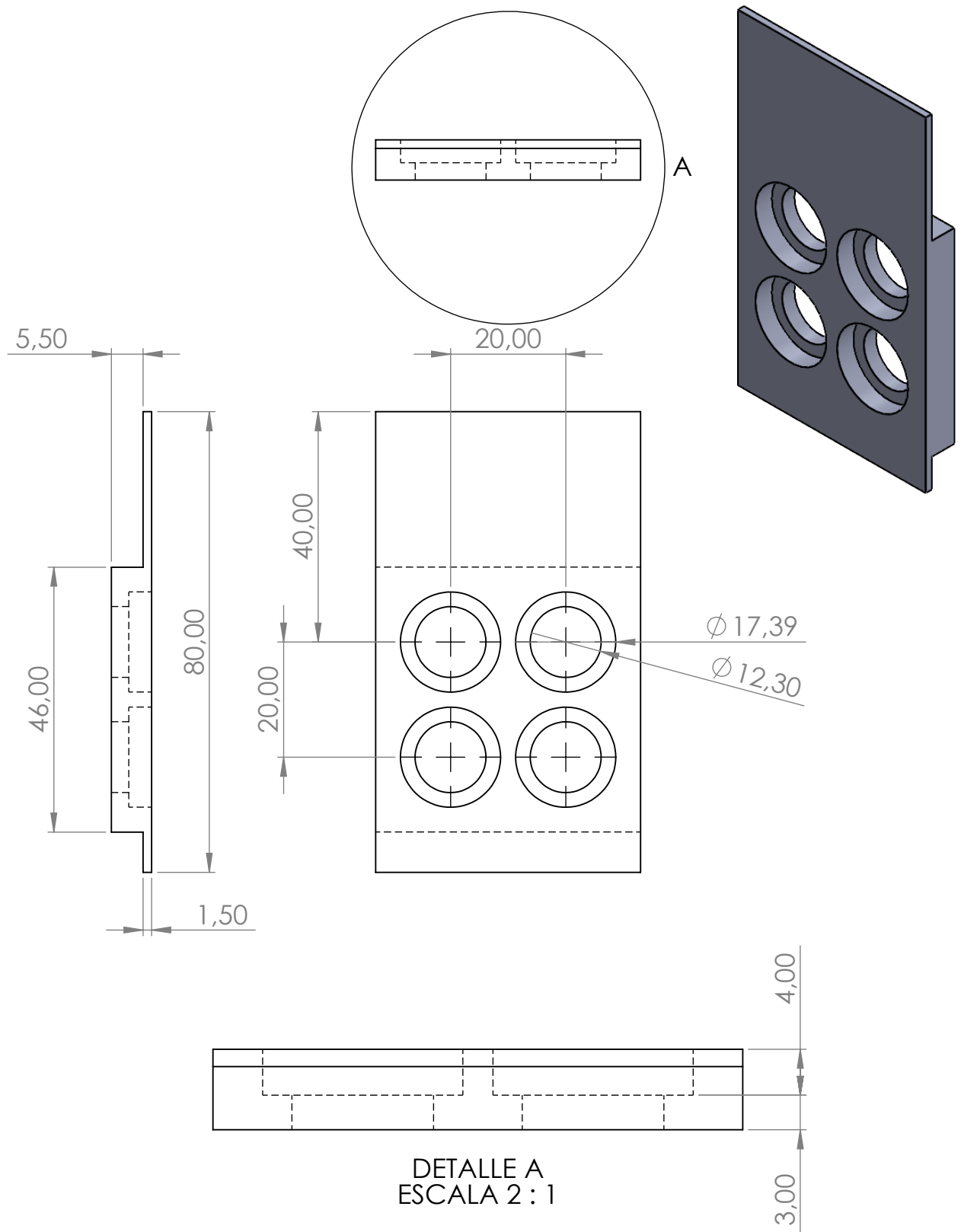


UTN				CIME				PROYECTO: BOTONERA BRAILLE CONJUNTO: Ensamblaje_botonera CÓDIGO: EN-BOT				ESCALA: 1:5	
PESO:		278.32 g		DISEÑO:		CHIGUANO A.		6/1/2024		HOJA: 1/1 A3			
LARGO:		527mm		DIBUJÓ:		CHIGUANO A.		7/1/2024					
ANCHO:		28,43mm		REVISÓ:		VALENCIA F.		17/7/2024					
ALTO:		80mm		APROBÓ:		VALENCIA F.		17/7/2024					

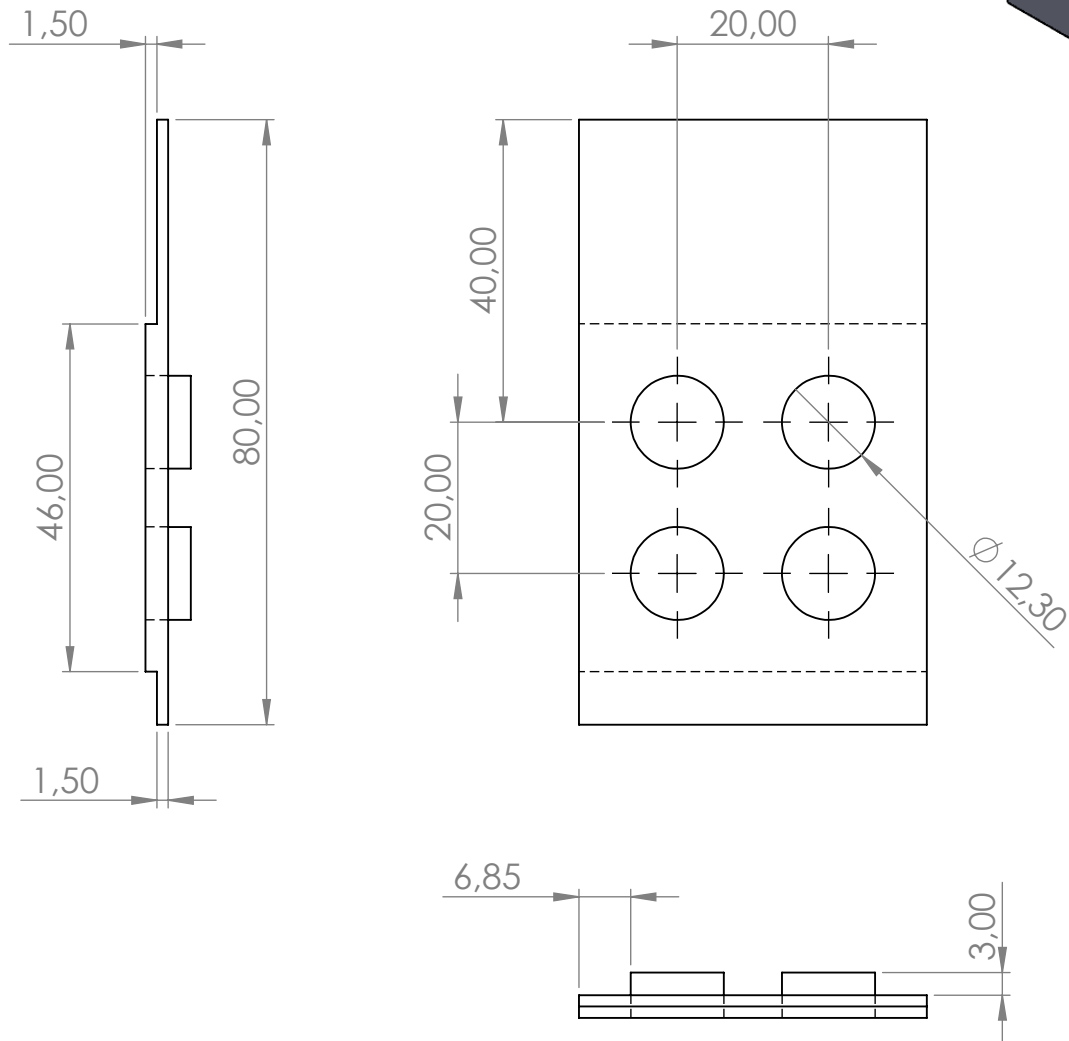
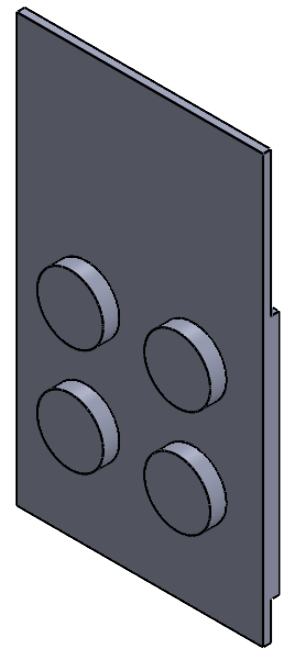


DETALLE A
ESCALA 2 : 1

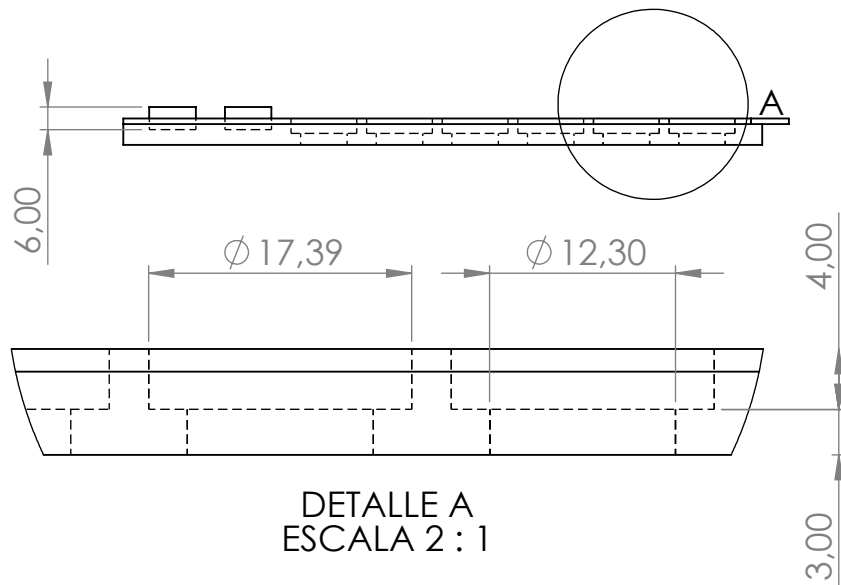
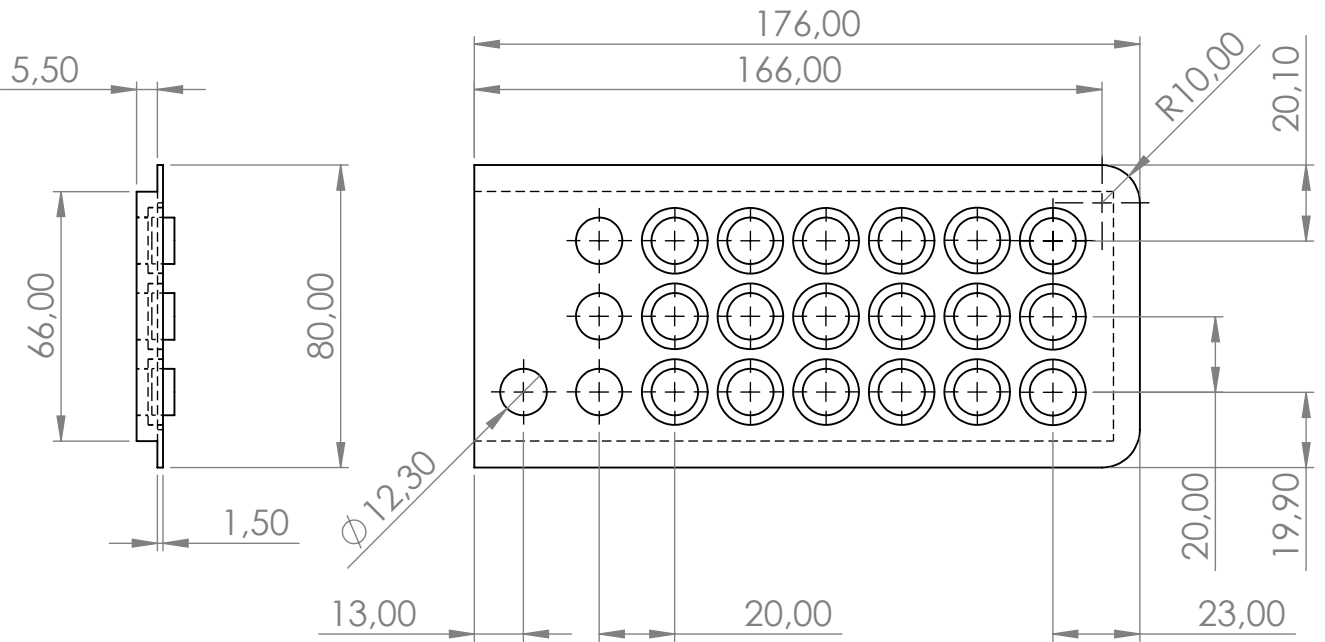
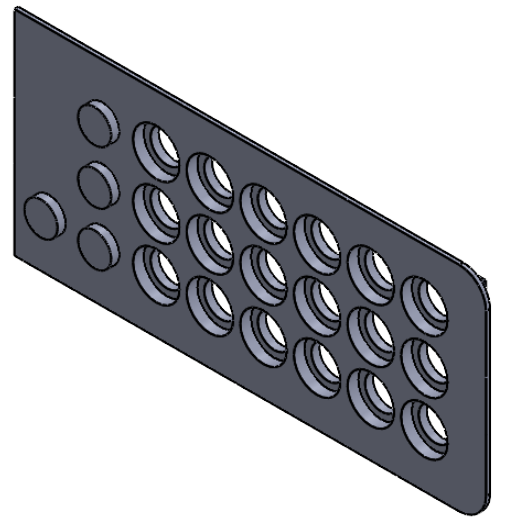
	PROYECTO: BOTONERA BRAILLE				ESCALA:	
	PARTE: DIG1				1:2	
	CÓDIGO: MA-BO-01				TOLERANCIA: ±	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



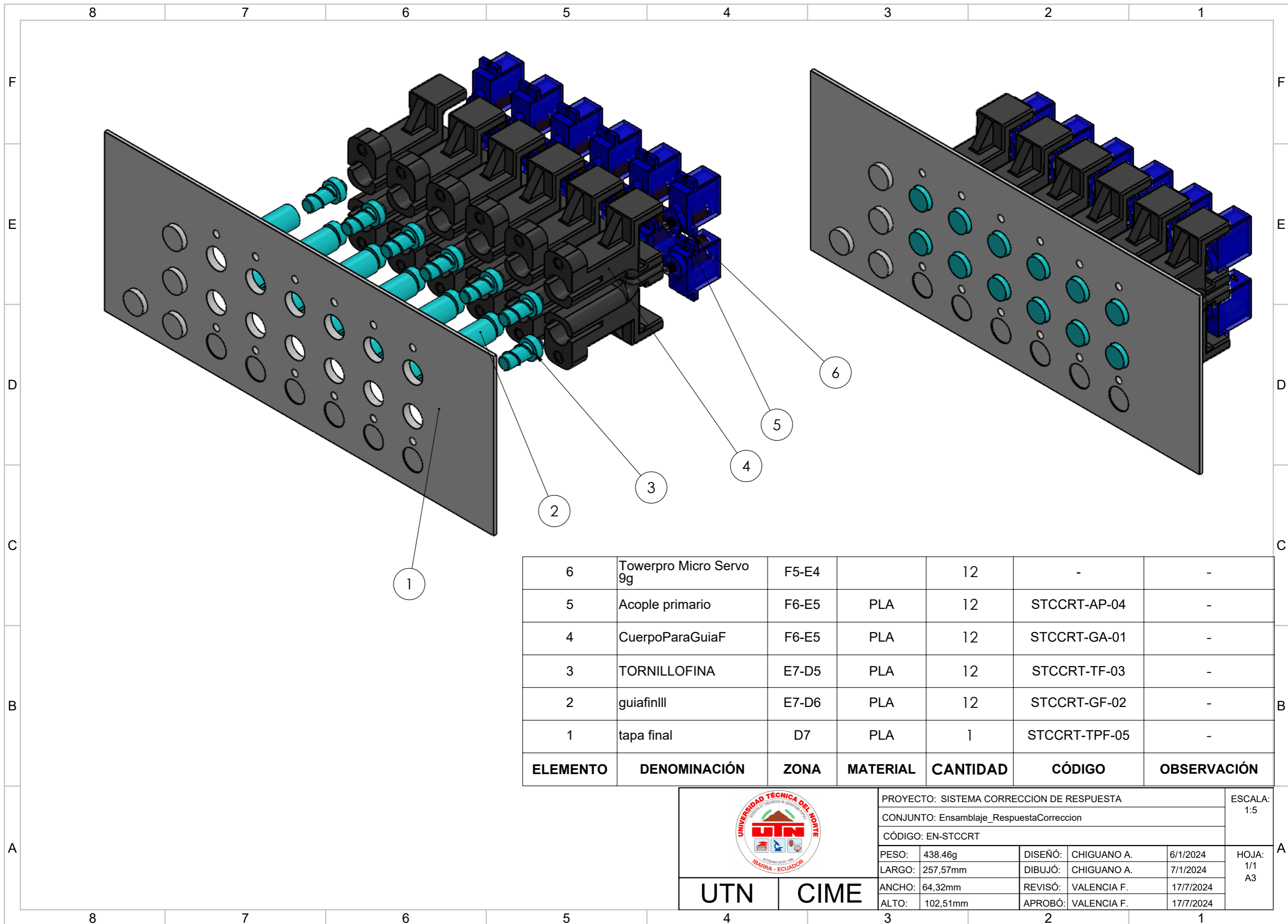
	PROYECTO: BOTONERA BRAILLE				ESCALA:	
	PARTE: SIGNO				1:1	
	CÓDIGO: SG-BO-02				TOLERANCIA: ±	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



	PROYECTO: BOTONERA BRAILLE				ESCALA:	
	PARTE: IGUAL				1:1	
	CÓDIGO: IG-BO-03				TOLERANCIA: ±	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



	PROYECTO: BOTONERA BRAILLE				ESCALA:		
	PARTE: RESULTADO				1:2		
	CÓDIGO: RT-BO-05				TOLERANCIA: ±		
	MATERIAL:	PLA	DISEÑÓ:	CHIGUANO A.	6/1/2027	HOJA:	
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024		
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024			
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024			
UTN	CIME					1/1	A4

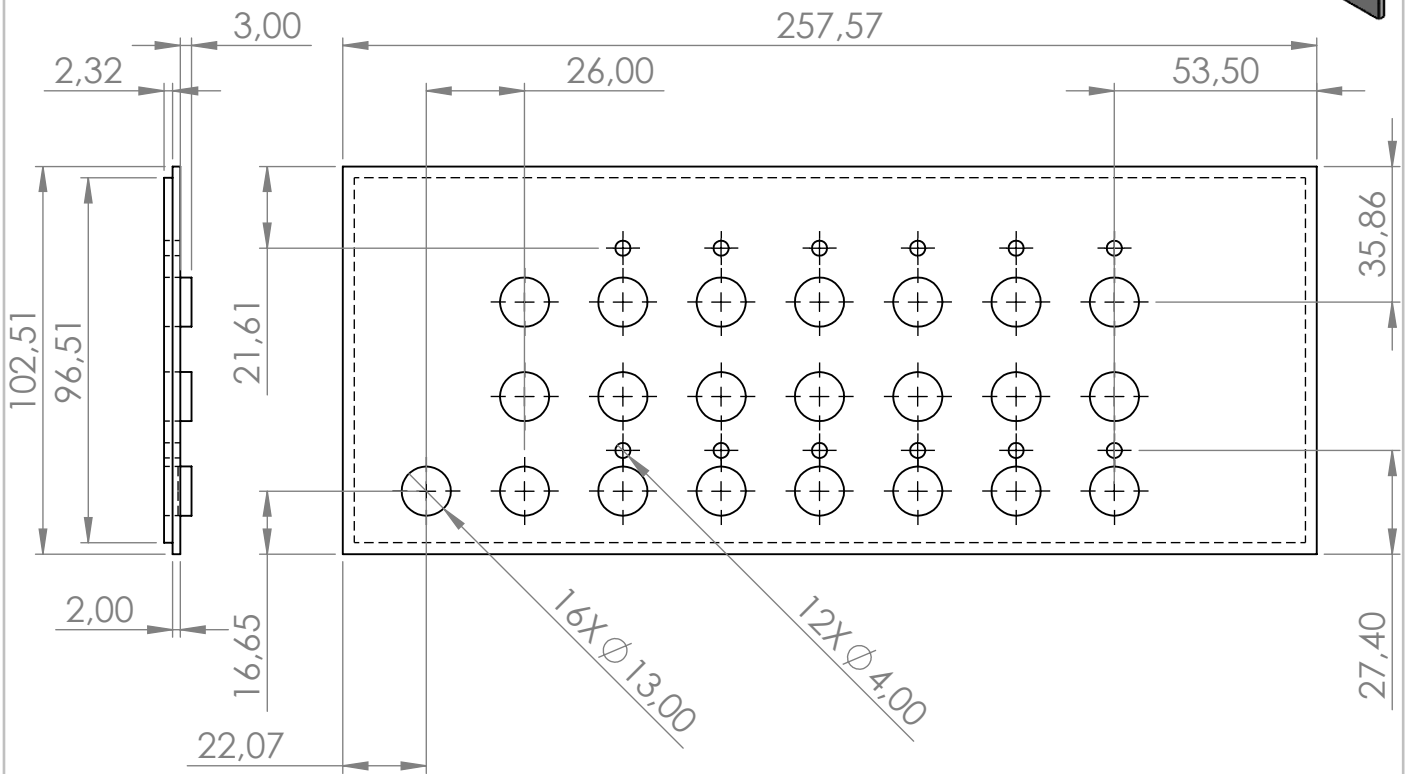
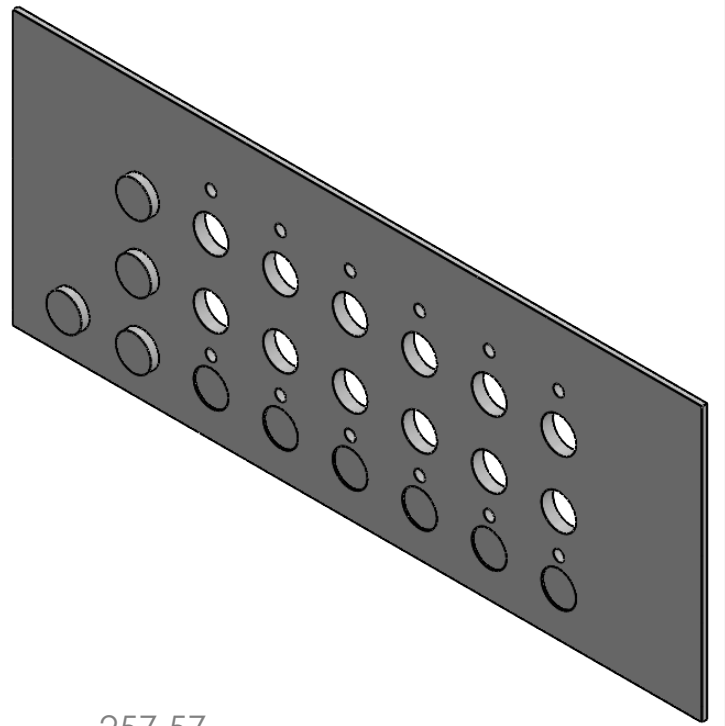


6	Towerpro Micro Servo 9g	F5-E4		12	-	-
5	Acople primario	F6-E5	PLA	12	STCCRT-AP-04	-
4	CuerpoParaGuiaF	F6-E5	PLA	12	STCCRT-GA-01	-
3	TORNILLOFINA	E7-D5	PLA	12	STCCRT-TF-03	-
2	guiafinIII	E7-D6	PLA	12	STCCRT-GF-02	-
1	tapa final	D7	PLA	1	STCCRT-TPF-05	-
ELEMENTO	DENOMINACIÓN	ZONA	MATERIAL	CANTIDAD	CÓDIGO	OBSERVACIÓN

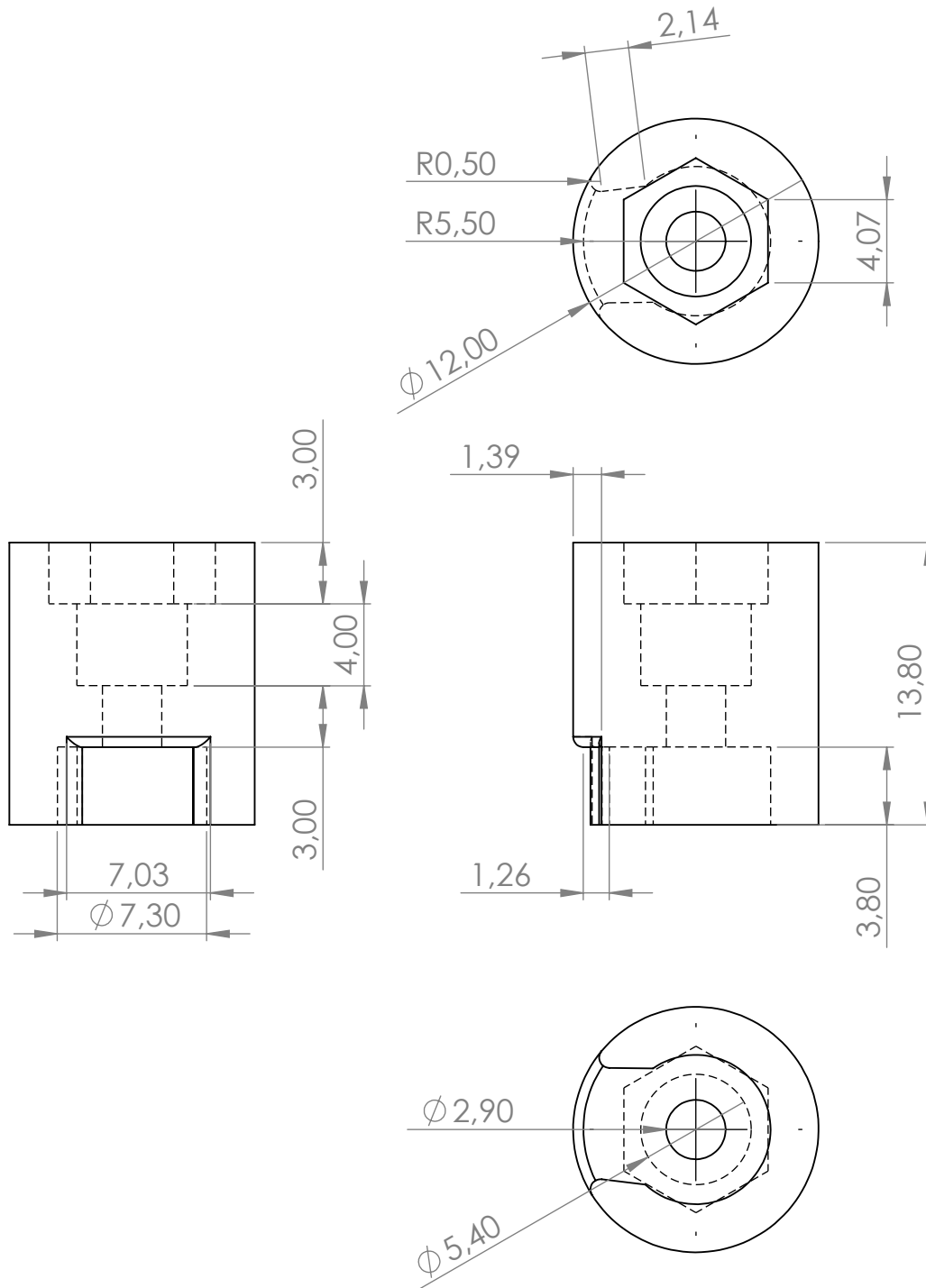
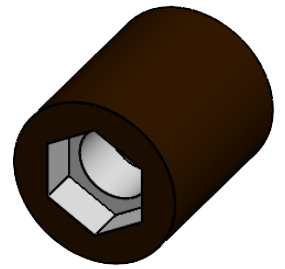



UTN CIME

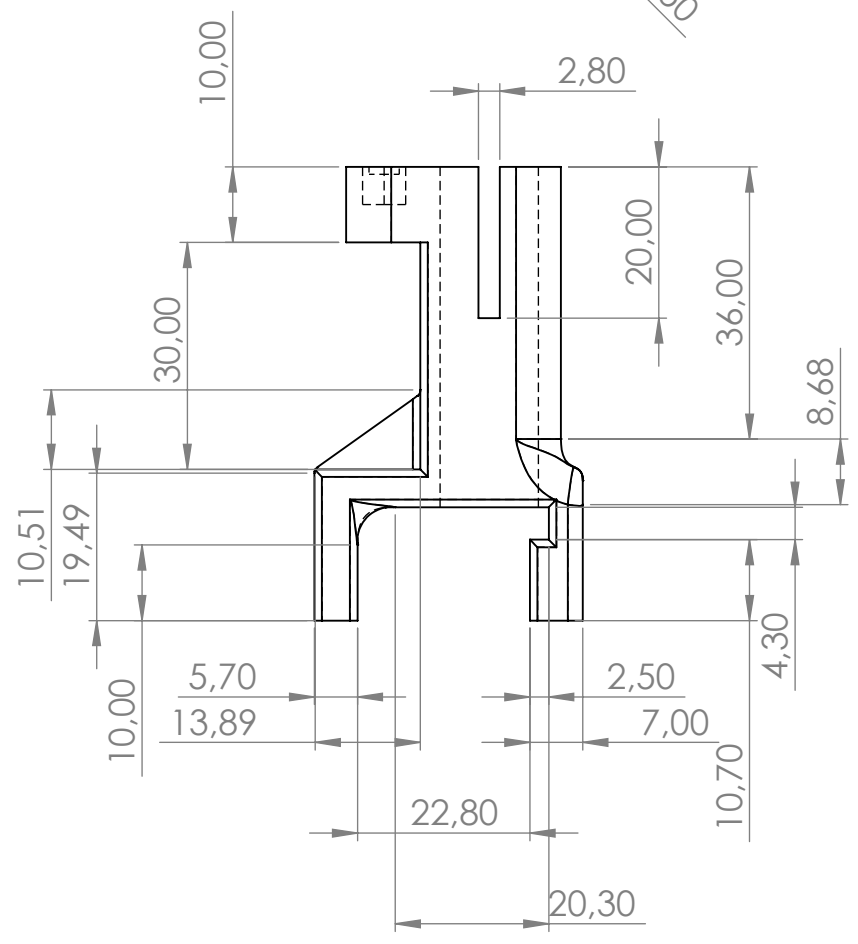
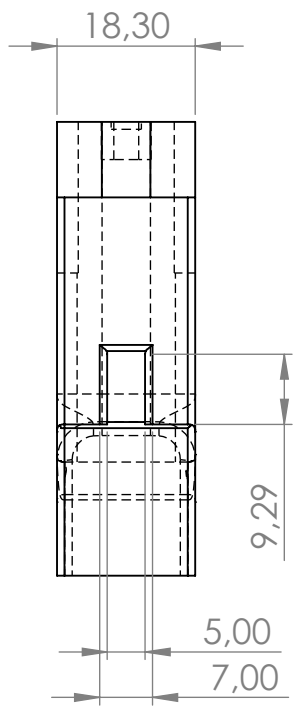
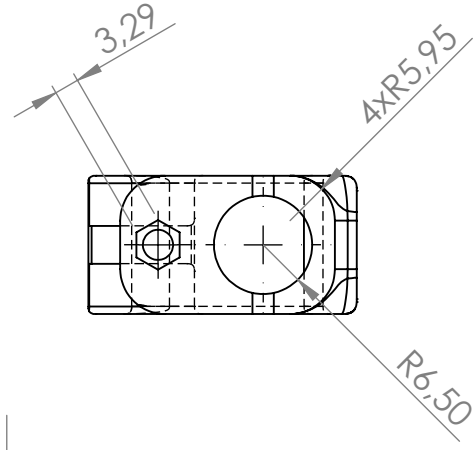
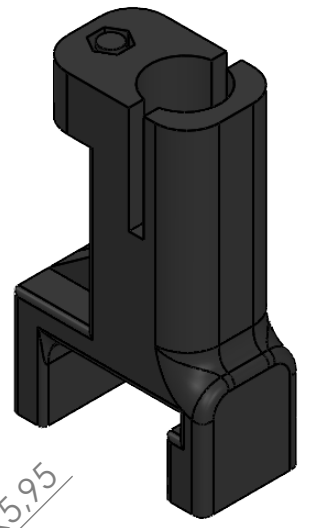
PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA: 1:5	
CONJUNTO: Ensamblaje_RespuestaCorreccion					
CÓDIGO: EN-STCCRT					
PESO:	438.46g	DISEÑO:	CHIGUANO A.	6/1/2024	HOJA: 1/1 A3
LARGO:	257,57mm	DIBUJÓ:	CHIGUANO A.	7/1/2024	
ANCHO:	64,32mm	REVISÓ:	VALENCIA F.	17/7/2024	
ALTO:	102,51mm	APROBÓ:	VALENCIA F.	17/7/2024	



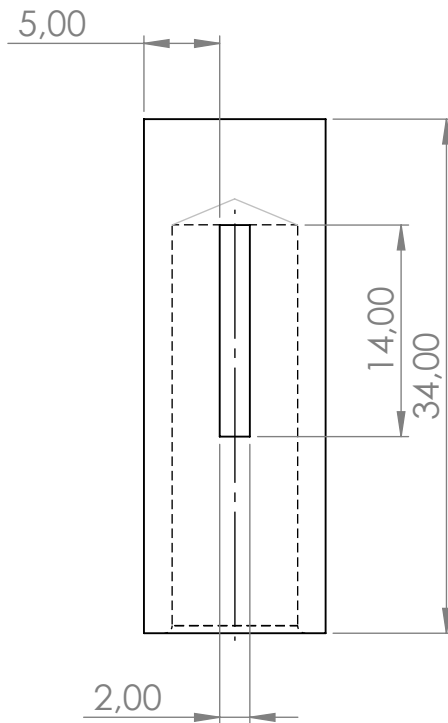
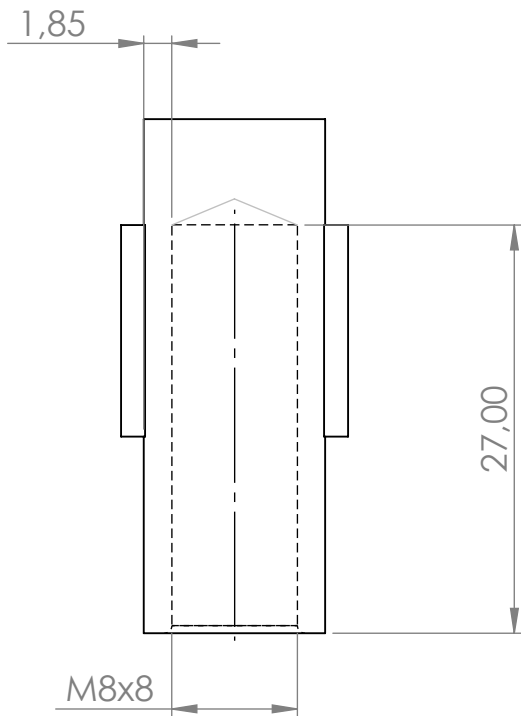
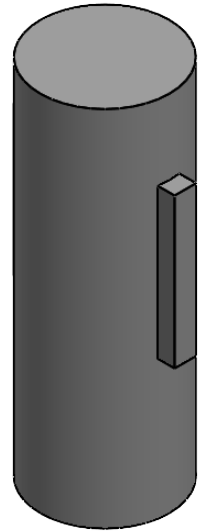
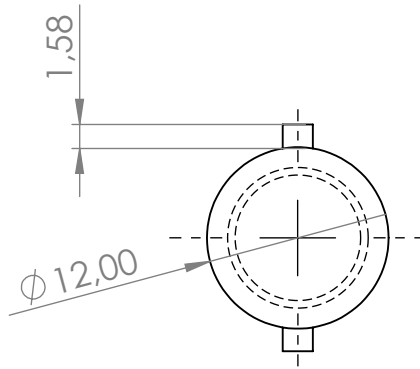
	PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA:	HOJA:
	PARTE: tapa final				1:5	
CÓDIGO: STCCRT-TPF-05				TOLERANCIA: ±		
MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027		
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024		
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	1	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



	PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA:	
	PARTE: Acople primario				2:1	
	CÓDIGO: STCCRT-AP-04				TOLERANCIA: \pm	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	12	APROBÓ:	VALENCIA F.	17/7/2024		
UTN	CIME					



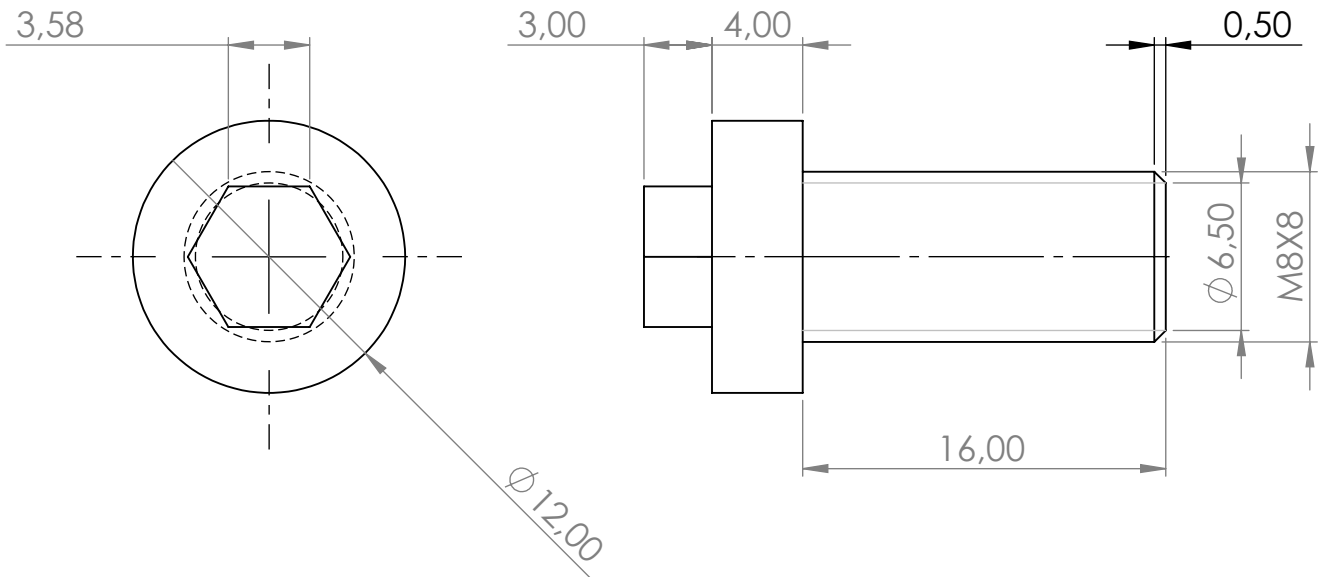
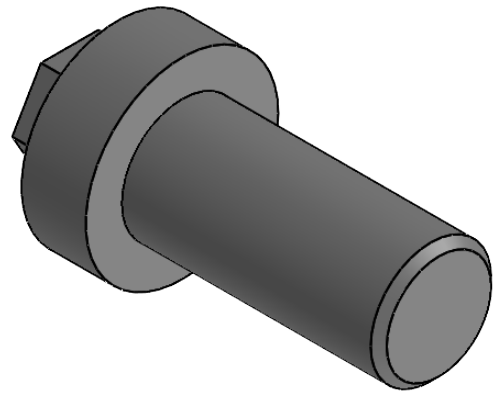
	PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA:	
	PARTE: CuerpoParaGuiaF				1:1	
	CÓDIGO: STCCRT-GA-01				TOLERANCIA: ±	
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024		
CANTIDAD:	12	APROBÓ:	VALENCIA F.	17/7/2024		
UTN		CIME				




PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA: 2:1	
PARTE: guiafinIII					
CÓDIGO: STCCRT-GF-02				TOLERANCIA: ±	
MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA: 1/1 A4
TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024	
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024	
CANTIDAD:	12	APROBÓ:	VALENCIA F.	17/7/2024	

UTN

CIME



	PROYECTO: SISTEMA CORRECCION DE RESPUESTA				ESCALA:		
	PARTE: TORNILLOFINA				2:1		
	CÓDIGO: STCCRT-TF-03				TOLERANCIA: \pm		
	MATERIAL:	PLA	DISEÑO:	CHIGUANO A.	6/1/2027	HOJA:	
	TRATAMIENTO:	NINGUNO	DIBUJÓ:	CHIGUANO A.	6/1/2024		
RECUBRIMIENTO:	NINGUNO	REVISÓ:	VALENCIA F.	17/7/2024			
CANTIDAD:	12	APROBÓ:	VALENCIA F.	17/7/2024			
UTN	CIME					1/1	A4

Anexo E

Manual de Usuario



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

Manual de Usuario

Paso 1.- Se realiza la conexión del cable de poder a un tomacorriente como se observa en la Fig. Anexo E.1.

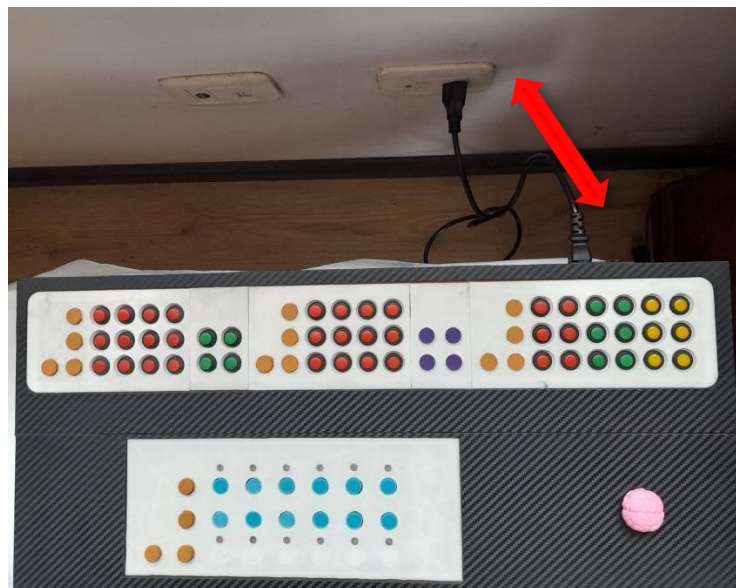


Fig. Anexo E.1

Paso 2.- Luego se presiona el botón de encendido del dispositivo como se muestra en la Fig. Anexo E.2.

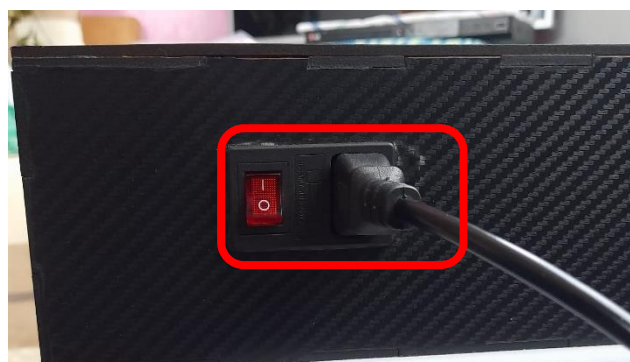


Fig. Anexo E.2

Paso 3.- Para dar inicio al modo Practica se presiona el pulsador que sobre sale de la estructura. (ver Fig. Anexo E.3).

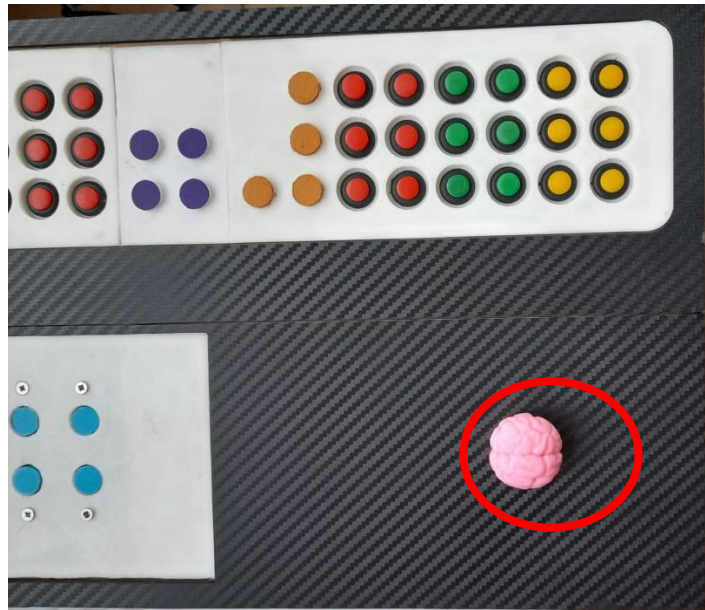


Fig. Anexo E.3

Paso 4.- Posterior se ingresa el primer número de hasta 2 dígitos mediante los botones de enclavamiento que representan el lenguaje Braille (véase Fig. Anexo E.4).

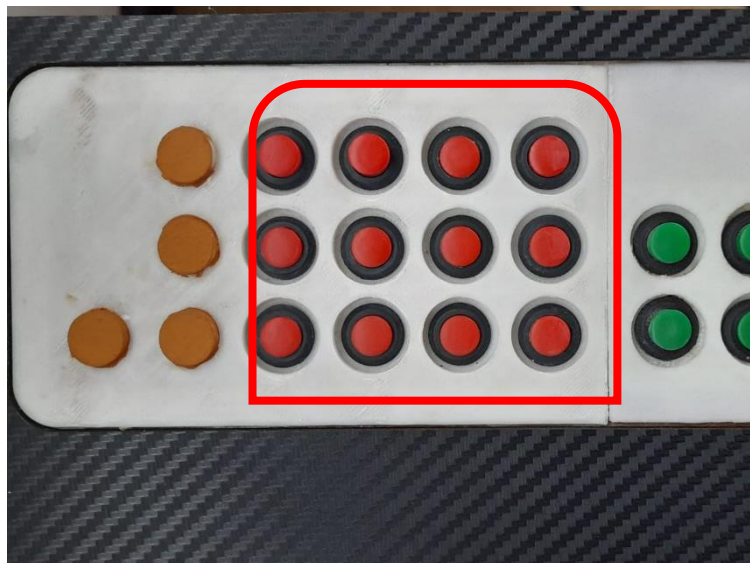


Fig. Anexo E.4

Paso 5.- A continuación, se ingresa el operador sea el caso suma o resta de igual manera a través de los botones de enclavamiento como se observa en la Fig. Anexo E.5.

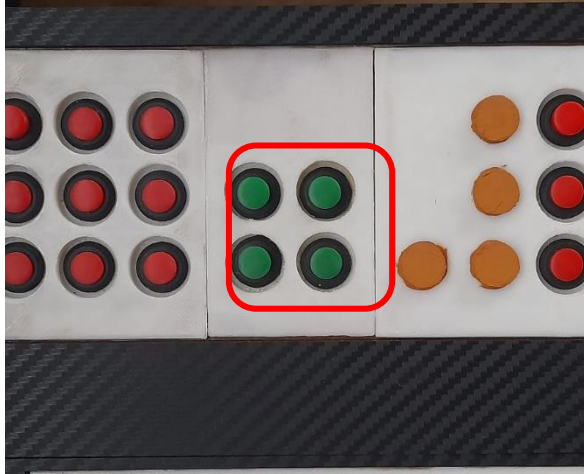


Fig. Anexo E.5

Paso 6.- Seguidamente se ingresa el segundo número de hasta 2 dígitos mediante los botones de enclavamiento como se muestra en la Fig. Anexo E.6.

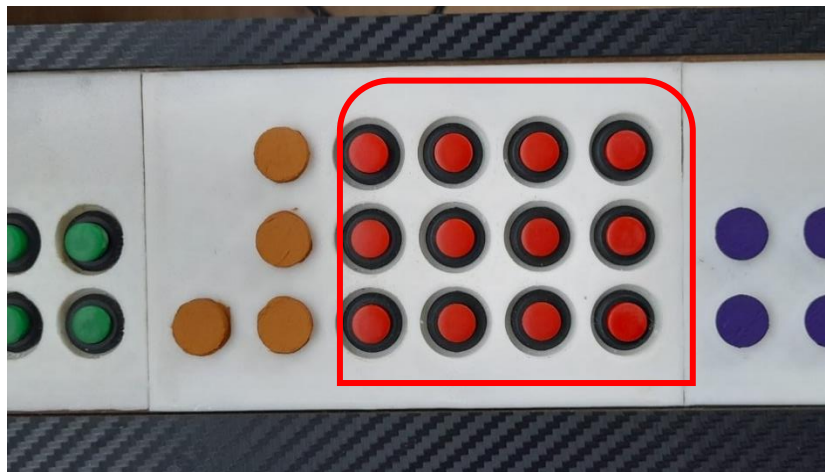


Fig. Anexo E.6

Paso 7.- Una vez ingresados los numero a operar se ingresa el resultado de hasta tres dígitos en la matriz de respuesta (ver Fig. Anexo E.7).

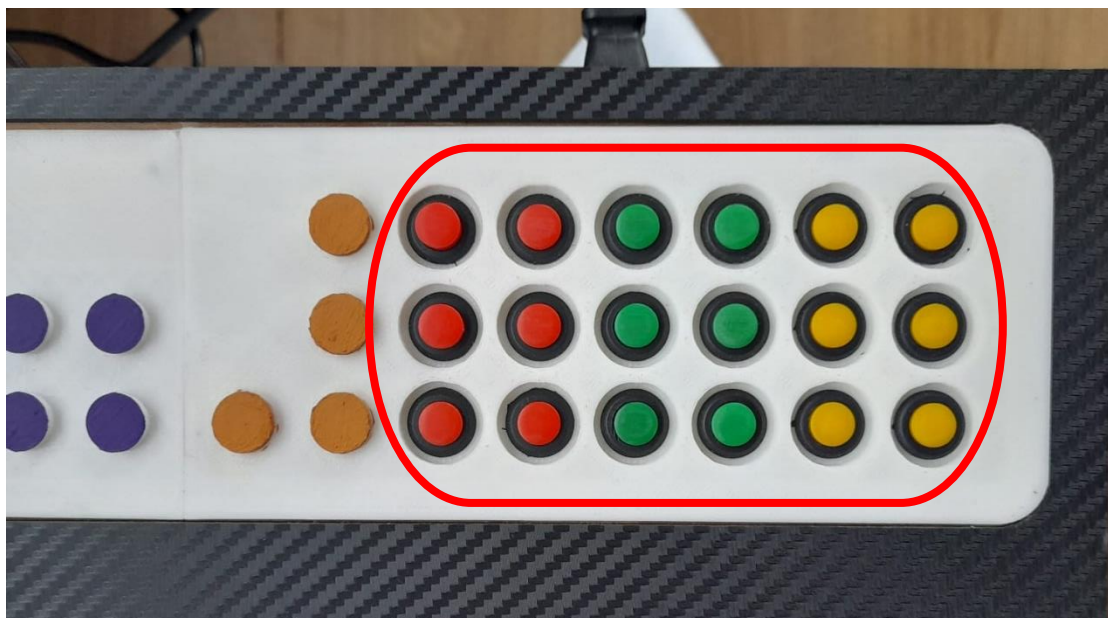


Fig. Anexo E.7

Paso 8.- A continuación, se realiza presiona el pulsador nuevamente esta vez para validar la respuesta ingresada.

- *Si la repuesta ingresada por el usuario es correcta la matriz de respuesta no se accionará se recomienda esperar un lapso de 5 segundos para ingresar la siguiente operación.*
- *En el caso de que la respuesta ingresada sea incorrecta la matriz de respuesta de manera automática corrige la respuesta plasmándola sobre la superficie el valor correcto como se puede apreciar en la Fig. Anexo E.8*

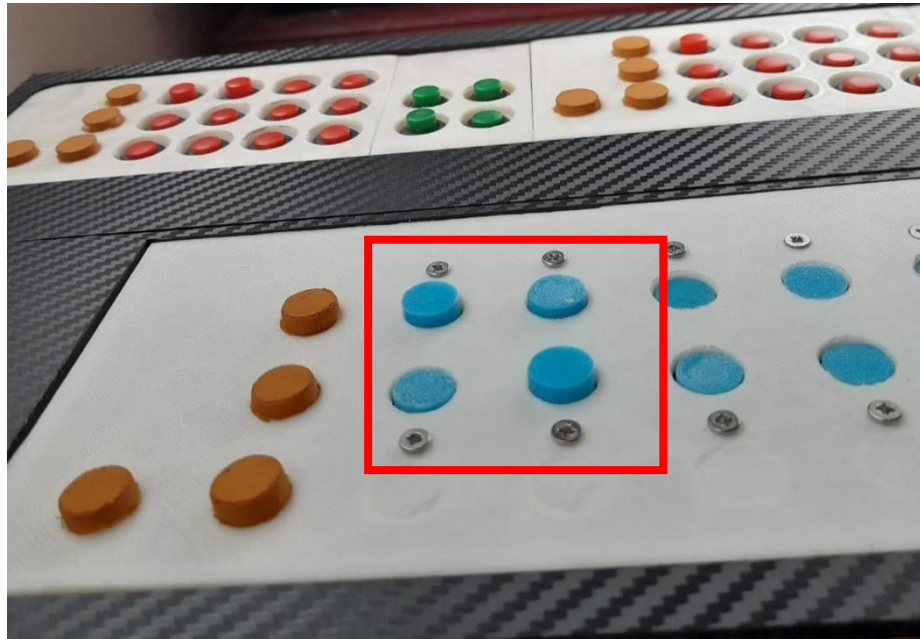


Fig. Anexo E.8

- *Después de 15 segundos la matriz de corrección se encera es decir todos los caracteres Brailles retroceden como se muestra en la Fig. Anexo E.9, para así poder acceder a una nueva operación si así lo requiere el usuario.*



Fig. Anexo E.9

Paso 9.- Se ingresar la nueva operación a practicar, creándose así un ciclo repetitivo de los pasos antes mencionados.

Paso 10.- Finalmente al culminar la práctica se apaga el dispositivo mediante el mismo botón de encendido como se muestra en la Fig. Anexo E.10 y se procede a desconectar del toma corriente.



Fig. Anexo E.10

Anexo F

Manual de Mantenimiento



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN MECATRÓNICA

Manual de Mantenimiento

Botones DS428

1. Vida útil de cien mil pulsaciones, en caso de encontrar resistencia inusual se recomienda lubricar las partes móviles. De persistir dicha resistencia o no retorno a su posición, se recomienda reemplazarlo, de preferencia del mismo color.
2. Se recomienda el uso de paños secos o ligeramente humedecidos con alcohol isopropílico para limpiar la superficie del botón, en caso de encontrarse en ambientes con polvo o grasa.
3. Inspeccionar periódicamente los puntos de conexión asegurándose de que los cables estén correctamente conectados y no exista corto circuitos
4. Verificar su correcta sujeción, asegurando que no exista corrosión o grietas en su estructura, caso contrario de existir corrosión limpiar con productos comerciales quita oxido como (WD-40).

Servomotores

1. Verificar que no exista suciedad externa como polvo o grasa, de existir dichos factores se recomienda limpiar con paño seco o ligeramente humedecido con alcohol isopropílico por su estructura exterior.
2. Visualizar que las conexiones (Señal, VCC y GND) se encuentre correctamente aisladas y unidas, que no existan corrosión ni suciedad, por

lo contrario de existir corrosión en sus pines limpiar con productos quita óxidos comerciales como (WD-40).

3. En caso de existir sonido de funcionamiento, se recomienda, desconectar y separar para dar una limpieza interna a sus engranajes con un pincel pequeño y seco. Por otro lado, el uso de un lubricante para engranajes de plástico es muy recomendado para alargar la vida útil de dicho componente.

Impresiones 3D

1. Se recomienda que no se almacene en lugares que exista humedad, ya que los componentes impresos en 3D pueden absorber dicha humedad y podrían perder rigidez y llevar a su deterioro progresivo.
2. Evitar en todo momento la exposición prolongada al sol o lugares donde pueden alcanzar temperaturas superiores a 50°C, ya que, del ser el caso, las piezas pueden generar deformaciones.
3. Tener una limpieza paulatina de las piezas que tengan contacto con el exterior, evitando la acumulación de polvo con la ayuda de paños secos o humedecidos en alcohol isopropílico.
4. De presentarse con fallos directos de roturas o agrietamientos de las diferentes piezas, se recomienda el cambio inmediato de piezas, basándose en los planos realizados adjuntos en el Anexo D.

Estructura MDF

1. En caso de encontrar polvo alrededor de la estructura, recomendable la limpieza con la ayuda de paños húmedos de alcohol isopropílico para evitar la acumulación prolongada de polvo.

2. Con respecto a presentar daños estructurales como golpes, o doblamientos de las paredes que lo recubren de manera muy extrema se deberán cambiar las partes afectadas, realizando un corte laser en una plancha de MDF basándose en las dimensiones de planos adjuntos.

Componentes Electrónicos

1. De ser necesario revisar cada 3 meses los diferentes componentes electrónicos de permanecer conectados correctamente y evitar la acumulación de polvo, con la ayuda de pinceles secos dar una limpieza suave a los componentes evitando desconectar los diferentes cables.