



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales

No Renovables

Carrera de Electromecánica

Diseño y construcción de un banco didáctico para el análisis del par resistente de máquinas.

**Trabajo de Integración Curricular,
previo a la obtención del título de
Ingeniero Electromecánico.**

AUTOR:

Marco Danilo Aguilar Loiza

DIRECTOR:

Ing. José Fabricio Cuenca Granda, Mg. Sc

Loja-Ecuador

2024



unl

Universidad
Nacional
de Loja

Sistema de Información Académico
Administrativo y Financiero - SIAAF

CERTIFICADO DE CULMINACIÓN Y APROBACIÓN DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo, **CUENCA GRANDA JOSE FABRICIO**, director del Trabajo de Integración Curricular denominado **DISEÑO Y CONSTRUCCIÓN DE UN BANCO DIDÁCTICO PARA EL ANÁLISIS DEL PAR RESISTENTE DE MÁQUINAS**, perteneciente al estudiante **MARCO DANILO AGUILAR LOAIZA**, con cédula de identidad N° **0705642726**.

Certifico:

Que luego de haber dirigido el **Trabajo de Integración Curricular**, habiendo realizado una revisión exhaustiva para prevenir y eliminar cualquier forma de plagio, garantizando la debida honestidad académica, se encuentra concluido, aprobado y está en condiciones para ser presentado ante las instancias correspondientes.

Es lo que puedo certificar en honor a la verdad, a fin de que, de así considerarlo pertinente, el/la señor/a docente de la asignatura de **Integración Curricular**, proceda al registro del mismo en el Sistema de Gestión Académico como parte de los requisitos de acreditación de la Unidad de Integración Curricular del mencionado estudiante.

Loja, 31 de Julio de 2024



Firmado electrónicamente por:
JOSE FABRICIO
CUENCA GRANDA

F) _____
DIRECTOR DE TRABAJO DE INTEGRACIÓN CURRICULAR



Certificado TIC/TT.: UNL-2024-001519

Autoría

Yo, **Marco Danilo Aguilar Loaiza**, declaro ser autor/a del presente trabajo de integración curricular o de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mí del trabajo de integración curricular o de titulación en el Repositorio Digital Institucional – Biblioteca Virtual.



Marco Danilo Aguilar Loaiza

Cédula de Identidad: 0705642726

Fecha: Loja 25 de febrero de 2025.

Correo electrónico: marco.aguilar@unl.edu.ec

Teléfono o Celular: 0992738855

Autorización

Carta de autorización del trabajo de integración curricular por parte del autor para la consulta de producción parcial o total, y publicación electrónica de texto completo

Yo **Marco Danilo Aguilar Loaiza** declaro ser autor del trabajo de integración curricular o de titulación titulado **Diseño y construcción de un banco didáctico para el análisis del par resistente de máquinas**, como requisito para optar el título de **Ingeniero Electromecánico**; autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de integración curricular o de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 25 días del (mes) de diciembre del dos mil veinticinco.

Firma:



Autor: Marco Danilo Aguilar Loaiza

Cedula: 0705642726

Dirección: Argelia

Correo electrónico: marco.aguilar@unl.edu.ec

Celular: 0992738855

DATOS COMPLEMENTARIOS:

Director del trabajo de integración curricular o de titulación: Ing. José Fabricio Cuenca Granda

Dedicatoria

El presente trabajo lo dedico de todo corazón a mis padres, Marco Aguilar y Liria Loaiza, por su incansable esfuerzo, por cada madrugada compartida, por sus consejos llenos de sabiduría, su amor incondicional y la fortaleza con la que siempre me han guiado por el camino correcto, sin rendirse jamás. Este logro, y los que aún están por venir, son también de ustedes.

A mis hermanos, Santiago y María, quienes han estado presentes en mis mayores alegrías y también en los momentos más difíciles, brindándome siempre su apoyo, su motivación y su cariño.

A mis abuelitos, Juvenal y Esthela, que han sido un pilar fundamental en mi vida. Gracias por su amor incondicional, por sus palabras llenas de sabiduría, por sus enseñanzas transmitidas con cariño y paciencia, y por motivarme siempre a seguir adelante. Su ejemplo de trabajo, perseverancia y valores ha sido una guía constante que me ha acompañado en cada etapa de este camino académico.

Dedico también este trabajo a Maritza Avemañay, por su apoyo incondicional a lo largo de este proceso. Su amor, dedicación y constante respaldo han sido fundamentales en mi camino académico.

Y, finalmente, a toda mi familia en general, que con sus consejos, compañía y afecto, han sido parte esencial de este logro.

Agradecimiento

Agradezco de manera especial a la Universidad Nacional de Loja y a la carrera de Ingeniería Electromecánica por ser el espacio donde, a lo largo de estos cinco años, pude crecer personal y profesionalmente. Mi más sincero reconocimiento a todos los docentes que, con vocación y compromiso, aportaron desinteresadamente a la construcción de mi conocimiento, fortaleciendo mi formación académica y técnica.

Extiendo un agradecimiento profundo al Ing. José Cuenca, por su acertada dirección, guía y colaboración invaluable en el desarrollo de este trabajo. Su apoyo ha sido fundamental para alcanzar este logro.

A mis compañeros y amigos de carrera, gracias por compartir conmigo cada etapa de esta experiencia universitaria. Juntos enfrentamos desafíos, celebramos logros y forjamos recuerdos imborrables.

Y, por supuesto, a mis padres, Marco Aguilar y Liria Loaiza, por ser mi mayor fuente de motivación. Gracias por su amor, sacrificio y apoyo incondicional, que han sido la base sobre la cual he construido cada uno de mis sueños.

Índice de contenidos

CERTIFICACIÓN	II
AUTORÍA	III
AUTORIZACIÓN	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE ANEXOS	XVII
SIMBOLOGÍA	XVIII

1. TÍTULO **1**

2. RESUMEN **2**

3. INTRODUCCIÓN **4**

4. MARCO TEÓRICO **6**

4.1 CAPÍTULO I: FUNDAMENTOS TEÓRICOS **6**

4.1.1 BANCO DIDÁCTICO 6

4.1.2 MOTORES DE AC 6

4.1.3 MOTORES DE DC 6

4.1.4 CURVAS TACODINÁMICAS 6

4.1.5 TORQUE 7

4.1.5.1 Torque constante 7

4.1.5.2 Torque lineal 7

4.1.5.3 Torque parabólico 8

4.1.5.4 Torque hiperbólico 9

4.2 CAPÍTULO II: SOFTWARE DE DISEÑO **9**

4.2.1 AUTOCAD 10

4.2.2 INVENTOR 10

4.2.3 ULTIMAKER CURA 10

4.3 CAPÍTULO III: SOFTWARE DE OBTENCIÓN Y PROCESAMIENTO DE DATOS **10**

4.3.1	SOFTWARE DE ADQUISICIÓN DE DATOS	10
4.3.1.1	Arduino	11
4.3.1.1.1	Modulo relé	11
4.3.1.1.2	Modulo sensor de velocidad	12
4.3.1.1.3	Modulo sensor de corriente	13
4.3.1.1.4	Modulo sensor de voltaje	14
4.3.2	SOFTWARE DE PROGRAMACIÓN.	14
4.3.2.1	Matlab	14
4.3.2.1.1	Simulink	15
4.3.2.1.2	Design App	15
4.4	CAPÍTULO IV: MODELADOS PARA MOTORES DC CON ESCOBILLAS O DE IMANES PERMANENTES.	16
4.4.1	MÉTODO TEÓRICO	16
4.4.1.1	Resistencia de armadura	16
4.4.1.2	Inductancia de armadura	16
4.4.1.3	Constante electromotriz	16
4.4.1.1	Constante de torque	17
4.4.1.2	Constante de tiempo mecánica	17
4.4.1.3	Momento de inercia (J)	17
4.4.1.4	Corriente de arranque y torque de fricción	18
4.4.1.5	Constante de Fricción de Coulomb	18
4.4.2	MODELO ELECTROMECAÁNICO.	18
4.4.3	FUNCIONES DE TRANSFERENCIA DEL MOTOR DC	20
4.5	CAPÍTULO V:	21
4.5.1	METODOLOGÍA PARA EL DESARROLLO DE ACTIVIDADES PRÁCTICAS Y DE INVESTIGACIÓN.	21
4.5.1.1	Preparatorio	21
4.5.1.2	Desarrollo	22
4.5.1.3	Informe	22
5.	<u>METODOLOGÍA</u>	23
5.1	ÁREA DE TRABAJO	23
5.2	EQUIPOS Y MATERIALES	23

5.2.1	SOFTWARE	23
5.2.2	INSTRUMENTOS DE MEDICIÓN.	23
5.3	PROCEDIMIENTO	24
5.3.1	PRIMER OBJETIVO	24
5.3.1.1	Exploración de componentes necesarios para la implementación en el banco	24
5.3.1.2	Diseño	24
5.3.1.3	Construcción del banco didáctico	25
5.3.2	SEGUNDO OBJETIVO	25
5.3.2.1	Recopilación de datos	25
5.3.2.2	Desarrollo de interfaces gráficas	25
5.3.3	TERCER OBJETIVO	26
5.3.3.1	Análisis de requerimientos de las guías prácticas	26
5.4	PROCESAMIENTO Y ANÁLISIS DE DATOS.	27
6.	RESULTADOS	28
<hr/>		
6.1	EXPLORACIÓN DE COMPONENTES NECESARIOS PARA LA IMPLEMENTACIÓN EN EL BANCO	28
6.2	REQUISITOS DE DISEÑO	28
6.2.1	DESCRIPCIÓN DE LAS CARACTERÍSTICAS DE LOS TIPOS DE MOTORES.	28
6.2.1.1	Motores adaptados para el accionamiento de grúas:	28
6.2.1.2	Motores para bombas de desplazamiento positivo:	29
6.2.1.1	Motores para ventiladores:	29
6.2.2	DESCRIPCIÓN DE LAS MÁQUINAS.	29
6.2.3	REQUERIMIENTOS DE DISEÑO	29
6.2.4	MÁQUINAS	29
6.2.4.1	Grúa	29
6.2.4.1.1	Bomba de diafragma	30
6.2.4.2	Ventilador	31
6.2.4.3	Elementos motrices	31
6.2.4.3.1	Motor acoplado a reductor (Motor para accionar grúa)	31
6.2.4.3.2	Motor RS775 (Motor para accionar bomba)	32
6.2.4.3.3	Motor doble eje (Motor para ventilador)	33
6.3	DISEÑO DEL BANCO DIDÁCTICO	33

6.3.1	ESQUEMA DE SISTEMA DE ELEVACIÓN	34
6.3.2	ESQUEMA DE SISTEMA DE BOMBEO	35
6.3.3	ESQUEMA DE SISTEMA DE VENTILACIÓN.	36
6.3.4	DETALLES DE LOS ELEMENTOS QUE CONFORMAN EL SISTEMA DE ELEVACIÓN, BOMBEO Y VENTILACIÓN.	36
6.3.4.1	Ejes	37
6.3.4.2	Poleas	38
6.3.4.2.1	Relación de transmisión	39
6.3.4.3	Correas	39
6.3.4.4	Chumaceras	40
6.3.5	DISEÑO DE PIEZAS	40
6.4	CONSTRUCCIÓN DEL BANCO DIDÁCTICO.	45
6.4.1	FUNDICIÓN DE PESAS.	47
6.4.2	ENSAMBLE DE LOS DIFERENTES COMPONENTES ELÉCTRICOS.	48
6.5	COSTOS DE CONSTRUCCIÓN E IMPLEMENTACIÓN DEL BANCO DIDÁCTICO.	52
6.6	VALIDACIÓN DE CARACTERÍSTICAS DE MOTORES	54
6.6.1	MEDICIÓN DE LOS PARÁMETROS ELÉCTRICOS Y MECÁNICOS DE LOS MOTORES.	54
6.6.2	ANÁLISIS COMPARATIVO ENTRE PARÁMETROS DE FÁBRICA Y MEDICIONES EXPERIMENTALES.	55
6.7	CREACIÓN DE INTERFACES GRÁFICAS.	56
6.7.1	CREACIÓN DEL APLICATIVO PASO A PASO	58
6.7.2	FLUJOGRAMA DE FUNCIONAMIENTO DEL APLICATIVO.	66
6.7.2.1	Validación de interfaces gráficas	66
6.7.2.1.1	Modelo fenomenológico	67
6.7.2.1.2	Funciones de transferencia	69
6.8	CALIBRACIÓN DE SENSORES	71
6.8.1	VOLTAJE	71
6.8.2	AMPERAJE	72
6.8.3	SENSOR INFRARROJO FOTOELÉCTRICO DE VELOCIDAD ENCODER	75
6.8.4	RELÉ PARA EL MOTOR DE LA BOMBA	78
6.8.5	RELÉS PARA MOTORREDUCTOR	79
6.8.6	CALIBRACIÓN SENSOR MAF	81
6.8.7	CALIBRACIÓN SENSOR DE PRESIÓN	81
6.8.8	CALIBRACIÓN DE FINAL DE CARRERA	82

6.9	VALIDACIÓN DE FUNCIONAMIENTO	83
6.10	CARACTERIZACIÓN Y ANÁLISIS EL COMPORTAMIENTO DEL PAR MOTOR Y RESISTENTE	89
6.10.1	SISTEMA DE ELEVACIÓN DE CARGAS.	89
6.10.2	SISTEMA DE BOMBEO	93
6.10.3	SISTEMA DE VENTILACIÓN.	96
6.11	GUÍA DE INSTALACIÓN DEL APLICATIVO	99
6.12	GUÍAS PRÁCTICAS	100
6.13	MANUAL DE OPERACIÓN DEL BANCO DIDÁCTICO EXPERIMENTAL	100
<u>7.</u>	<u>DISCUSIÓN</u>	<u>102</u>
<u>8.</u>	<u>CONCLUSIONES</u>	<u>104</u>
<u>9.</u>	<u>RECOMENDACIONES</u>	<u>105</u>
<u>10.</u>	<u>BIBLIOGRAFÍA</u>	<u>106</u>
<u>11.</u>	<u>ANEXOS</u>	<u>108</u>

Índice de tablas

Tabla 1. Máquinas de posible implementación en el banco	28
Tabla 2. Características de ejes.	37
Tabla 3. Características de poleas	38
Tabla 4. Relación de transmisión.	39
Tabla 5. Características de correas.....	39
Tabla 6. Características de chumaceras	40
Tabla 7. Costo de Elementos.....	52
Tabla 8. Costos de manufactura.	54
Tabla 9. Costos totales	54
Tabla 10. Datos de motores.....	55
Tabla 11. Validación de parámetros.	56
Tabla 12. Medición de rpm con diferentes parámetros.....	76
Tabla 13. Pruebas de funcionamiento.	77
Tabla 14. Error porcentual en los valores voltaje medidos por los diferentes equipos.....	84
Tabla 15. Error porcentual en los valores corriente medidos por los diferentes equipos.	85
Tabla 16. Velocidad del motorreductor ante diferentes voltajes de alimentación.	87
Tabla 17. Validación del encoder.	89
Tabla 18. Ensayos del sistema de elevación de cargas.	90
Tabla 19. Ensayos del sistema de bombeo.....	93
Tabla 20. Ensayos del sistema de ventilación.....	97

Índice de figuras

Figura 1. Curva de torque constante.	7
Figura 2. Curva de torque lineal.	8
Figura 3. Curva de torque parabólico	8
Figura 4. Curva de torque constante.	9
Figura 5. Placa Arduino	11
Figura 6. Modulo relé.	12
Figura 7. Sensor de velocidad.....	13
Figura 8. Sensor de corriente.	13
Figura 9. Sensor de voltaje.	14
Figura 10. Constante electromotriz.....	16
Figura 11. Motor DC.	18
Figura 12. Ubicación de la FERNNIR.....	23
Figura 13. Resumen de procedimientos.....	26
Figura 14. Análisis de datos.....	27
Figura 15. Grúa.	30
Figura 16. Bomba de diafragma.	30
Figura 17. Ventilador.....	31
Figura 18. Motor acoplado a reductor con conmutación por escobillas	32
Figura 19. Motor RS775 tipo escobilla.....	32
Figura 20. Motor doble eje de metal con conmutación por escobillas	33
Figura 21. Diagrama cinemático del sistema de elevación.....	34
Figura 22. Sistema de elevación.	34
Figura 23. Diagrama cinemático del sistema de bombeo.	35
Figura 24. Sistema de bombeo.....	35
Figura 25. Diagrama de sistema de ventilación.....	36
Figura 26. Sistema de ventilación.....	36
Figura 27. Eje de motor acoplado a reductor.....	37
Figura 28. Eje de motor de bomba.....	37
Figura 29. Eje de motor de ventilador.	37
Figura 30. Eje de bomba.	37
Figura 31. Eje de tambor.....	37
Figura 32. Eje de grúa.....	37

Figura 33. Eje de ventilador.....	37
Figura 34. Polea 16 dientes. Fuente: Amazon, sf.	38
Figura 35. Polea de 40 dientes. Fuente: Amazon, sf.....	38
Figura 36. Polea de 36 dientes. Fuente: Amazon, sf.....	38
Figura 37. Polea de 30 dientes Fuente: Amazon, sf.....	38
Figura 38. Correa dentada.....	39
Figura 39. Chumacera de 8mm.....	40
Figura 40. Chumacera de 12mm.....	40
Figura 41. Base de motor de ventilador.	40
Figura 42. Base de bornera de derivación.....	41
Figura 43. Medidor de revoluciones por minuto.	41
Figura 44. Soportes para la grúa.	42
Figura 45. Bomba de diafragma.	42
Figura 46. Base para medición en grados.	42
Figura 47. Brida para obstrucción del aire.....	43
Figura 48. Ducto de ventilación.....	43
Figura 49. Enrolladora de cable.....	44
Figura 50. Base para sensores.....	44
Figura 51. Polea loca.	45
Figura 52. Impresión 3D.....	46
Figura 53. Torneado de pesas.	46
Figura 54. Construcción de carriles templadores para bandas.....	47
Figura 55. Proceso de función del plomo.	47
Figura 56. Esquema eléctrico en Proteus.....	48
Figura 57. Circuito eléctrico del banco.....	49
Figura 58. Circuito eléctrico del banco.....	51
Figura 59. Ensamble de los componentes físicos del banco.....	51
Figura 60. Medición de inductancia.	55
Figura 61. Código de Arduino.	56
Figura 62. Código en Script de Matlab.....	57
Figura 63. GUI de Matlab.....	58
Figura 64. Programación del menú práctica.	59
Figura 65. Ajuste del GUIDE.	59
Figura 66. Programación del grupo de botones de selección de gráfica.....	60

Figura 67. Programación que almacena los datos.....	60
Figura 68. Programación de la proyección de las gráficas.	61
Figura 69. Programa del grupo de botones para el estado del banco.....	61
Figura 70. Programación del botón bajar pesa.	62
Figura 71. Conexión entre Matlab y Arduino.....	63
Figura 72. Procesamiento de datos obtenidos del Arduino.	63
Figura 73. Programación del botón paro de emergencia.	64
Figura 74. Programación que permite graficar.	64
Figura 75. Programación que permite exportar los datos a un archivo Excel.	65
Figura 76. Flujograma del aplicativo.	66
Figura 77. Diagrama de bloques.....	67
Figura 78. Grafica de torque mecánico(Nm) en el eje y vs tiempo(s) en el eje x.	68
Figura 79. Grafica de velocidad angular(rpm) en el eje y vs tiempo(s) en el eje x.	68
Figura 80. Grafica de corriente de armadura(A) en el eje y vs tiempo(s) en el eje x.	69
Figura 81. Funciones de transferencia	69
Figura 82. Grafica de torque mecánico(Nm) en el eje y vs tiempo(s) en el eje x.	70
Figura 83. Grafica de corriente de armadura(A) en eje y vs tiempo(s) en el eje x.	70
Figura 84. Grafica de velocidad angular(rpm) en el eje y vs tiempo(s) en el eje x.	71
Figura 85. Calibración de voltímetro.	72
Figura 86. Conexión del sensor de voltaje al Arduino.	72
Figura 87. Señal real de sensor con A0.	73
Figura 88. Valor de voltaje medido desde el monitor serial de Arduino.....	73
Figura 89. Voltaje medido por el monitor serial cuando se aplica una carga de 1.622 A	74
Figura 90. Calibración del amperímetro.	74
Figura 91. Código de calibración.....	75
Figura 92. Conexión del sensor de corriente al Arduino.	75
Figura 93. Rueda de 60 agujeros.	76
Figura 94. Rueda de 16 agujeros	77
Figura 95. Conexión de los encoders.....	78
Figura 96. Código para la rueda de 16 agujeros.	78
Figura 97. Código para el control de motor de la bomba.	79
Figura 98. Conexión de relé del motor de la bomba.....	79
Figura 99. Código de los relés que controlan el sentido de giro del motorreductor.....	80
Figura 100. Inversión de giro de motorreductor.	80

Figura 101. Código del sensor MAF.	81
Figura 102. Código de sensor de presión.....	82
Figura 103. Código de calibración del final de carrera.....	82
Figura 104. Comparación de valores entre fuente y multímetro.	83
Figura 105. Gráfica de voltaje	84
Figura 106. Corriente.....	85
Figura 107. Características del motorreductor.....	86
Figura 108. Medición visual de vueltas del motor en un minuto.	86
Figura 109. Medición de la frecuencia de giro del motor por medio de un micrófono	88
Figura 110. Medición de la velocidad angular del motor por medio del encoder.	88
Figura 111. Gráfica de velocidad angular.....	89
Figura 112. Funcionamiento para toma de datos del sistema de elevación.....	90
Figura 113. Gráfica de análisis de datos de la velocidad, potencia voltaje y corriente con respecto al torque utilizando poleas de 36 -36 dientes.	91
Figura 114. Gráfica de análisis de datos de la velocidad, potencia, voltaje y corriente respecto al torque polea 36 – 40 dientes.	92
Figura 115. Gráfica de análisis de datos de la velocidad, potencia, voltaje y corriente respecto al torque polea 36 – 30 dientes.	92
Figura 116. Funcionamiento del sistema de bombeo para tomas de datos.....	93
Figura 117. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 -36 dientes.....	94
Figura 118. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 – 40 dientes.....	95
Figura 119. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 -30 dientes.....	95
Figura 120. Funcionamiento del sistema de ventilación para la toma de datos.....	96
Figura 121. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 -36 dientes.	98
Figura 122. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 – 40 dientes.	98
Figura 123. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 – 30 dientes.	99

Índice de anexos

Anexo 1. Fichas técnicas.	108
Anexo 2. Facturas.	118
Anexo 3. Certificado de la traducción del resumen.	121
Anexo 4. Archivos de respaldo.	122
Anexo 5. Guías prácticas.	123
Anexo 6. Códigos.	132
Anexo 7. Planos.	150

Simbología

V = Voltaje [V].

m = Sensibilidad.

I = Corriente [A].

R = Resistencia de armadura [Ohms].

L = Inductancia eléctrica [H].

J = Momento de inercia [$\text{kg}\cdot\text{m}^2$].

B = Constante de fricción viscosa [N.m.s].

Ka = Constante de fuerza electromotriz [V/rads].

Km = Constante del par motor [N.m/A].

tm_{ss} = Tiempo de estabilización mecánica [s].

tm = Constante de tiempo mecánica [s].

$iarr$ = Corriente de arranque [A].

Tf = Torque de fricción [N.m].

Tm = Torque mecánico [N.m].

w = Velocidad angular [rad/s].

1. Título

Diseño y construcción de un banco didáctico para el análisis del par resistente de máquinas.

2. Resumen

El presente trabajo permite estudiar el comportamiento de las curvas tacodinámicas de máquinas fundamentales con fines educativos consolidándose como un banco de prácticas.

Las curvas tacodinámicas de las máquinas electromecánicas fueron fundamentales para evidenciar la relación entre la potencia, el torque y la velocidad angular. En este trabajo se implementó un banco didáctico que permitió realizar prácticas experimentales para caracterizar y analizar el comportamiento del par resistente bajo distintas condiciones de operación, en sistemas como: bombeo, elevación de carga y ventilación.

A partir de información relacionada con los sistemas analizados, se definieron motores, máquinas, condiciones de operación y construcción del banco. A manera de ejemplo se expone el sistema de elevación de carga, permite variar la velocidad de la carga que se traslada. Para la adquisición de datos, se utilizó herramientas de programación como Matlab y Arduino junto con una tarjeta Arduino Mega, que permitió extraer información y graficar los resultados para su análisis.

Los resultados obtenidos reflejaron el comportamiento real de las máquinas, el cual coincidió con la literatura consultada. Aunque la adquisición de datos fue satisfactoria, existe la necesidad de mejorar la resolución para capturar fielmente los fenómenos instantáneos como picos de corriente, fluctuaciones de potencia, entre otros.

El banco desarrollado permitió la realización de tres tipos de prácticas bajo distintas condiciones, obteniendo curvas con una resolución aceptable. El banco fue diseñado para que los estudiantes de electromecánica puedan comprender de mejor manera cómo es afectada la potencia en distintas condiciones.

Palabras clave: banco didáctico, par motor resistente, curvas tacodinámicas.

Abstract

This work enables the study of the tachodynamic curves of fundamental machines for educational purposes, consolidating itself as a practice bench.

The tachodynamic curves of electromechanical machines were essential to demonstrate the relationship between power, torque, and angular velocity. In this project, a didactic test bench was implemented to conduct experimental practices aimed at characterizing and analyzing the behavior of the resistant torque under different operating conditions in systems such as pumping, load lifting, and ventilation.

Based on information related to the analyzed systems, motors, machines, operating conditions, and the bench's construction were defined. As an example, the load lifting system is presented, which allows for variation in the speed at which the load is moved. For data acquisition, programming tools such as Matlab and Arduino were used, along with an Arduino Mega board, which made it possible to extract data and plot results for analysis.

The results obtained reflected the actual behavior of the machines, which matched the information found in the literature. Although data acquisition was satisfactory, there remains a need to improve resolution in order to accurately capture instantaneous phenomena such as current peaks and power fluctuations, among others.

The developed bench enabled the execution of three types of experiments under different conditions, obtaining curves with acceptable resolution. The bench was designed to help electromechanical engineering students better understand how power is affected under various operating scenarios.

Keywords: didactic bench, resistant torque, tachodynamic curves.

3. Introducción

La potencia de las máquinas tiene una alta dependencia de la frecuencia de giro a la que operan. Dependiendo del tipo de máquina, estas pueden presentar un comportamiento constante, lineal, parabólico e hiperbólico, lo que motiva el presente trabajo, ya que se suele suponer que la relación entre potencia y velocidad es proporcional, cuando no lo es. Esta elección se realiza debido a la falta actual de un recurso educativo que facilite la comprensión de las curvas características de los motores de máquinas.

Para el desarrollo del trabajo, se partió de una búsqueda bibliográfica relacionada con las máquinas que se querían ejemplificar. A continuación, se realizó el estudio de los diferentes dispositivos que debía tener, las variables a medir y los comportamientos que se esperaban de los motores y de las máquinas bajo diferentes condiciones de operación. Mediante una tarjeta de adquisición Arduino y con el uso de Matlab, se realizó la adquisición de datos para el análisis de las curvas características de las máquinas.

Los objetivos propuestos y su cumplimiento se dieron de la siguiente manera:

- Diseñar y construir un banco didáctico experimental: Se construyó un banco de pruebas con tres sistemas principales: bombeo, elevación de carga y ventilación, incorporando motores, máquinas, sensores y un sistema de control y adquisición de datos basado en Matlab y Arduino.
- Caracterizar y analizar el comportamiento del par motor y resistente: Se analizaron las curvas características de los sistemas de bombeo, elevación de carga y ventilación mediante sensores y un sistema de adquisición de datos con Arduino Mega, identificando comportamientos de torque constante, lineal y parabólico. Estas curvas fueron comparadas con la búsqueda literaria para validar el desempeño de las máquinas en distintas condiciones de trabajo.
- Desarrollar guías y procedimientos para el procesamiento de datos y prácticas: Se elaboraron tres tipos de guías prácticas para las distintas condiciones de trabajo de los sistemas de bombeo, elevación de carga y ventilación, proporcionando instrucciones detalladas para su ejecución.

En cuanto al contenido del presente trabajo, en el primer capítulo se expone el marco teórico relacionado con máquinas con diferentes curvas tacodinámicas, software de diseño, sistema de adquisición de datos y modelados matemáticos para motores DC. En el segundo capítulo se describe la metodología paso a paso para el cumplimiento de los objetivos. En el tercer capítulo se exponen los resultados obtenidos en base a los objetivos planteados mediante la selección de motores, tarjetas y sensores, guías prácticas, valoración económica, entre otros.

Finalmente, se concluye y discuten los resultados, especialmente sobre la capacidad de adquisición de datos y fidelidad de las curvas en base a la literatura y otros aspectos.

4. Marco teórico

4.1 Capítulo I: Fundamentos teóricos

4.1.1 Banco didáctico

Los bancos de prueba son diseñados para evaluar el desempeño de una variedad de componentes y sistemas. Los cuales son fundamentales en la fabricación y el mantenimiento de piezas y sistemas en diversas industrias, como la automotriz, aeroespacial, energética, de maquinaria y de equipos eléctricos. Se utilizan para comprobar y confirmar la operación de motores, sistemas hidráulicos, transmisiones, sistemas eléctricos, equipos de comunicaciones y otros componentes.

Estos equipos pueden medir la fuerza, el par, la velocidad, el flujo, la presión, la temperatura y otras variables relevantes para el rendimiento del sistema o componente que se está probando (Mtorres, 2022).

4.1.2 Motores de AC

Son motores que funcionan a través de corriente alterna y cuentan con la particularidad de que transforman la energía eléctrica en energía mecánica. Su funcionamiento se basa en un campo magnético gravitatorio, una de las características más destacadas es el número de devanados del rotor, este dato nos dará el número de devanados de motor, que será doble de polos.

La principal razón por la cual estamos ante una alternativa muy recomendable se debe a que son significativamente más económicos y sufren un menor desgaste en comparación con los motores de corriente continua, además es ideal para aquellos proyectos en los que no se trabaja con un movimiento estable y de velocidad fija (ADIndustrial, 2022).

4.1.3 Motores de DC

Los motores DC son aquellos que, como su nombre indica, trabajan con corriente continua igualmente llamada corriente directa (en inglés direct current, DC) que genera un campo magnético que actúa sobre el rotor o inducido del motor convirtiendo de esta forma la energía eléctrica en energía mecánica (Zuendo, 2023).

4.1.4 Curvas tacodinámicas

En el siglo XVII, en Francia, el matemático y filósofo Blaise Pascal comenzó una investigación referente al principio mediante el cual la presión aplicada a un líquido contenido en un recipiente se transmite con la misma intensidad en todas direcciones, gracias a este principio se pueden obtener fuerzas muy grandes empleando otras relativamente pequeñas. Uno de los aparatos más comunes para alcanzar lo anteriormente mencionado es la prensa hidráulica, la cual está basada en el principio de Pascal (Torres, 2019).

4.1.5 Torque

La definición de torque se la puede dar como la fuerza aplicada para hacer que gire un objeto alrededor de un eje.

“El Torque es la fuerza de giro que puede ser suministrada por el eje de un motor hacia los ejes que estén acoplados y sometidos a cargas. En el sistema ingles la unidad de Torque es Libra fuerza-Pulgada (Lbf-Pulg) o Libra fuerza-Pie (Lbf-Pie), en el sistema internacional la unidad de torque viene expresada por Newton-Metro (N-m)” (Bavaresco, 2019).

4.1.5.1 Torque constante

El toque constante normalmente se origina en fuerzas de gravedad o fuerza de fricción seca en máquinas que pueden ser grúas, transportadores, hornos, molinos y varios procesos análogos de maquinado sobre todo en máquinas que son de arranque pesado siendo para ello necesario un torque pico del doble del nominal, en la **Figura 1** se representa comportamiento de la relación entre el torque y la velocidad.

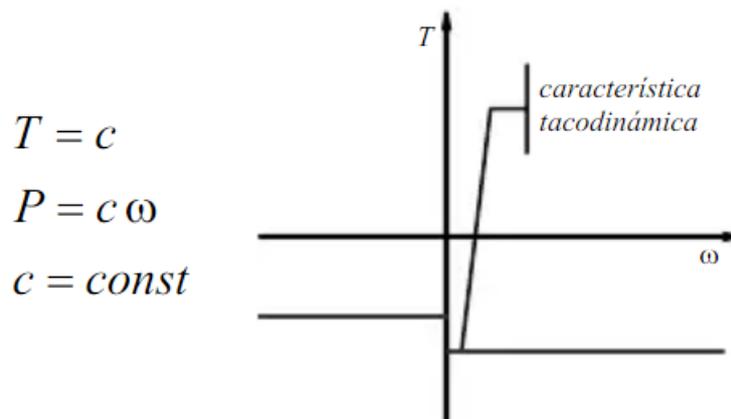


Figura 1. Curva de torque constante.

Fuente: (Martínez,2007).

4.1.5.2 Torque lineal

“Son máquinas de arranque ligero, pues a baja velocidad el torque es muy bajo, pero con la velocidad el torque crece linealmente y la potencia cuadráticamente”, en la **Figura 2** se representa comportamiento de la relación entre el torque y la velocidad (Martínez,2007).

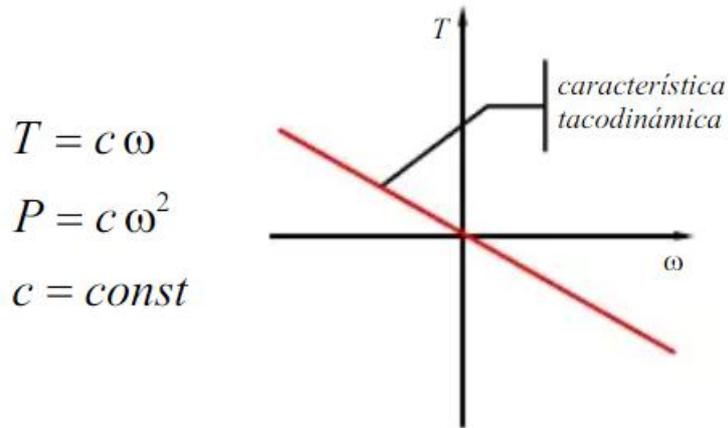


Figura 2. Curva de torque lineal.

Fuente: (Martínez,2007).

Este torque se genera a partir de fuerzas de fricción laminar ocasionadas por el rozamiento con fluidos en un régimen laminar. Entre este tipo de máquinas se incluyen bombas de desplazamiento positivo, mezcladores de líquidos y dispositivos utilizados en suspensiones de alta viscosidad, entre otras.

4.1.5.3 Torque parabólico

Son máquinas que son de arranque ligero debido que a baja velocidad el torque también disminuye, pero si la velocidad aumenta el torque aumenta cuadráticamente el cual se da debido al rozamiento o fricción viscosa ocasiona un régimen turbulento, en la **Figura 3** se representa comportamiento de la relación entre el torque y la velocidad.

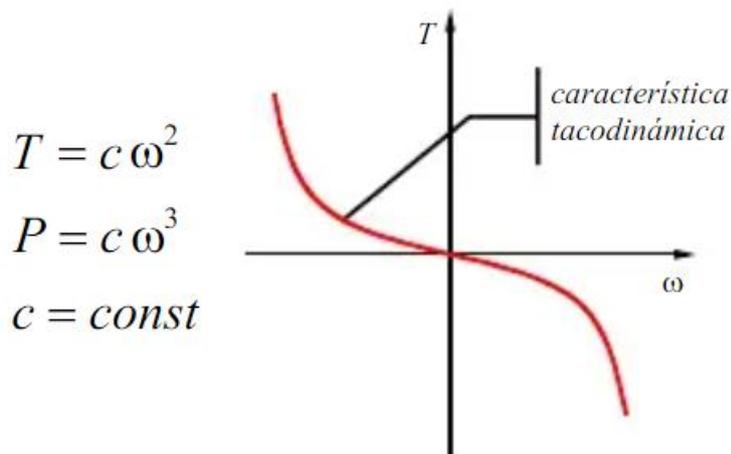


Figura 3. Curva de torque parabólico

Fuente: (Martínez,2007).

Entre los ejemplos más comunes de máquinas con torque parabólico encontramos: bombas de impelente, sopladores, ventiladores, turbomáquinas que trabajen con fluidos de baja viscosidad, entre otros.

4.1.5.4 Torque hiperbólico

Se origina en fuerzas constantes que actúan sobre puntos de velocidad constante, pero brazo de torque continuamente variable. Ejemplos: bobinadoras, enrolladoras, refrentado y procesos análogos de maquinado. Son máquinas de arranque pesado, pues a mínima velocidad el torque es máximo. Pero, con la velocidad, el torque decrece hiperbólicamente, y la potencia se mantiene constante; en la **Figura 4** se representa comportamiento de la relación entre el torque y la velocidad (Martínez,2007).

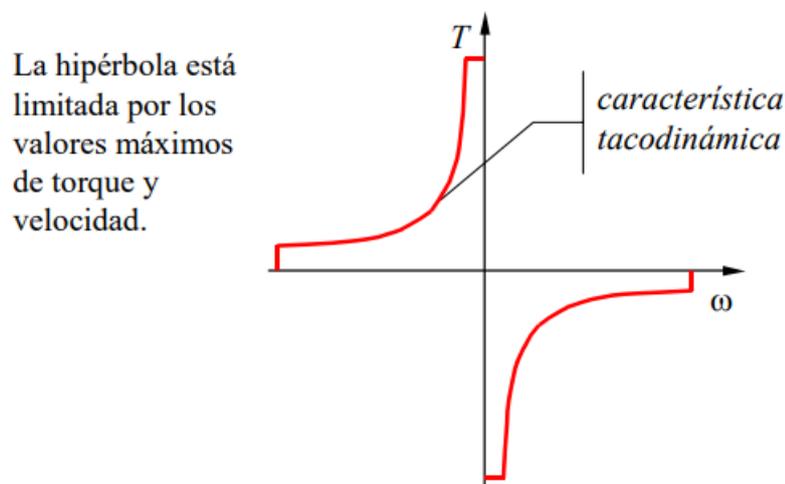


Figura 4. Curva de torque constante.

Fuente: (Martínez,2007).

4.2 Capítulo II: Software de diseño

Los software de diseño e ingeniería mecánica utilizados para crear, probar y comunicar ideas se han posicionado como una herramienta viable para ingenieros y diseñadores. En ocasiones, se hacen uso de herramientas adicionales en el software de CAD para fabricar productos con máquinas CNC o una impresora 3D. El software de ingeniería se utiliza en diferentes sectores, como el diseño de maquinaria industrial, la automoción y para los productos de consumo (Autodesk,2023).

Existen diferentes software como:

- AutoCAD.
- Inventor.
- Ultimaker Cura.

4.2.1 AutoCAD

AutoCAD es un programa que muchos arquitectos e ingenieros hacen uso para diseñar bocetos, planos y piezas que deben cumplir con los requisitos del cliente.

Además, AutoCAD es un programa de diseño versátil que permite desarrollar proyectos arquitectónicos, industriales, mecánicos y de ingeniería. Gracias a la facilidad de crear diseños en 2D y 3D, se ha convertido en uno de los programas de diseño digital más importante mercado (Quispe, 2020).

4.2.2 Inventor

Autodesk, es una empresa destacada por el desarrollo de software para arquitectos y diseñadores industriales, dado que ofrece un sin número de herramientas que facilitan y mejoran los diseños de los productos. Una de estas herramientas es Autodesk Inventor, conocido por su capacidad para mejorar el diseño mecánico y simulación tridimensional.

Inventor es un programa de modelado paramétrico de objetos en 3D que permite combinar datos 2D y 3D, lo que facilita la modificación directa en el diseño, simulación y validación del diseño (Perdomo, 2020).

4.2.3 Ultimaker Cura

Ultimaker Cura es el software más usado para la impresión 3D. Dado que este programa convierte diseños en 3D a programas como Fusion 360 o Blender, por medio de archivos CAD o STL, en formatos que las impresoras 3D pueden utilizar. Funciona con muchas impresoras como Creality, Ultimaker y Anet.

Para que una impresora 3D pueda imprimir correctamente las figuras diseñadas, se necesita software de preparación como Ultimaker Cura. Este programa permite ajustar varios parámetros para conseguir el resultado deseado.

Algunas de las opciones que se pueden controlar para la precisión es ajustando el ancho de cada capa, que va desde 0,16 mm hasta 0,4 mm. También, se puede definir el número de capas de pared para reforzar la pieza, el patrón y la cantidad de relleno interno, la velocidad de impresión y los soportes para áreas sin puntos de apoyo.

Una vez que la pieza está bien posicionada en el programa, solo hay que hacer clic en guardar. También, proporciona información útil como la cantidad de filamento necesario y el tiempo aproximado de impresión (García, 2023)

4.3 Capítulo III: Software de obtención y procesamiento de datos

4.3.1 Software de adquisición de datos

Los "Software de Adquisición de Datos" son herramientas que se encargan de recopilar y gestionar información de sensores y dispositivos. Además, facilitan la conexión y

comunicación permitiendo capturar datos en tiempo real. Estos programas no solo facilitan la adquisición de datos, sino que también ofrecen funciones para su análisis y visualización, siendo importantes en la investigación científica y el control industrial.

4.3.1.1 Arduino

Arduino es una herramienta para el desarrollo de hardware de código abierto que ofrece tanto hardware como software fáciles de usar y flexibles. La misma que está pensada para ayudar en la creación de proyectos electrónicos muy popular entre diseñadores y desarrolladores.

Las placas de Arduino son microcontroladores que tienen pines de entrada/salida y una interfaz para su programación. La mayoría de estas placas como se muestra en la **Figura 5** se programan con el entorno de desarrollo integrado (IDE) de Arduino, un software de código abierto que facilita la programación de los microcontroladores Arduino (Fernández, 2022).

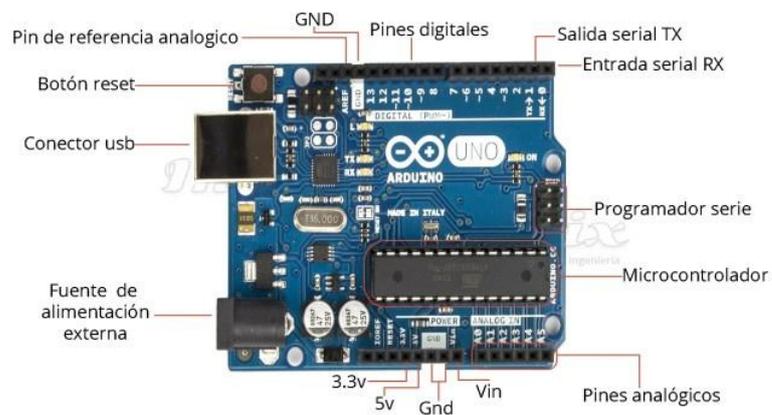


Figura 5. Placa Arduino

Fuente: (Fernández, 2022).

4.3.1.1.1 *Modulo relé*

Un relé electromagnético es un dispositivo que funciona como un interruptor controlado por un circuito eléctrico. Utiliza una bobina y un electroimán para manejar uno o varios contactos, lo que permite abrir o cerrar otros circuitos eléctricos de forma independiente.

Para activar un relé, se debe aplicar una tensión de 5 voltios al pin de control, el cual permite un flujo de corriente y cierra el circuito. Cuando se aplica una tensión de 0 voltios, la corriente se interrumpe y el circuito se abre. Existen relés disponibles para diferentes voltajes, como 12V, 24V, 110V y 220V (Del Valle, 2020).

Características del relé:

- Voltaje de alimentación: 5 VDC.
- Conexión directa a un microcontrolador u otro circuito integrado.

- Activa con una señal de 0 VDC (0 lógico).
- LED indicador de funcionamiento.
- Tiene tres pines: Positivo (VCC), Señal y Negativo (GND).
- Soporte de corriente: 10 A.
- Soporte de voltaje: 250 V.
- Diseño compacto para fácil integración.

En la **Figura 6**, se puede visualizar el aspecto del relé, proporcionando una representación gráfica para una mejor comprensión de su configuración y la ubicación de los pines.



Figura 6. *Modulo relé.*

Fuente: *(Del Valle, 2020).*

4.3.1.1.2 *Modulo sensor de velocidad*

El módulo sensor de velocidad IR con el comparador LM393 es una herramienta para medir la velocidad en diferentes aplicaciones, como las ruedas de un carro robot o como un interruptor óptico. Funciona interrumpiendo la luz infrarroja, lo que genera un pulso digital en el pin D0, que puede ser interpretado por microcontroladores como Arduino.

Este sensor de encoder óptico usa el interruptor infrarrojo MOCH22A, que consta de un emisor y un receptor IR. El cual mide el movimiento al interrumpir el haz de luz infrarroja con las ranuras de un disco acoplado al eje, generando pulsos TTL que son digitalizados. Por lo cual, se recomienda utilizar interrupciones por flancos de subida y bajada y añadir un capacitor de 100nF entre la salida D0 y tierra para evitar posibles falsos disparos.

Este modelo de encoder está diseñado para plataformas de robótica móvil, como seguidores de líneas y laberintos, así como para contar las revoluciones en motores DC/AC y como sensor de final de carrera. Es ideal para proyectos que requieren monitoreo y control exacto de la velocidad y posición de motores. La **Figura 7** se muestra el módulo, ofreciendo una representación gráfica que facilita la ubicación de los pines (Escobar, 2019).



Figura 7. *Sensor de velocidad.*

Fuente: (Escobar, 2019).

4.3.1.1.3 *Modulo sensor de corriente*

El módulo sensor de corriente ACS712-20A es útil para monitorear el consumo en proyectos de robótica y electrónica. Basado en el circuito integrado ACS712 de Allegro MicroSystems, este módulo mide con precisión la corriente en circuitos de corriente alterna (AC) o corriente directa (DC) (Escobar, 2019).

Características destacadas:

- Utilizar un sensor de efecto Hall que ofrece un voltaje de salida proporcional a la corriente en el circuito.
- Sensor lineal de efecto Hall de bajo offset.
- Alta precisión gracias a la proximidad del sensor al elemento de paso, ambos ubicados dentro del IC.
- Baja resistencia del elemento, que minimiza la disipación de potencia (1.2 mΩ típico).
- Soporta picos de corriente hasta 5 veces la corriente nominal de operación.
- Terminales del elemento conductor aisladas eléctricamente (hasta 2.1KV), eliminando la necesidad de aislamiento (optoacopladores).
- Funciona tanto con corriente alterna como corriente directa.

La **Figura 8** muestra una representación gráfica de este sensor, ayudando a comprender su disposición y componentes.



Figura 8. *Sensor de corriente.*

Fuente: (Escobar, 2019).

4.3.1.1.4 Modulo sensor de voltaje

El Sensor de Voltaje FZ0430 es un componente diseñado para medir con precisión el estado de una batería o verificar la alimentación de dispositivos de 12V o 24V, como tiras LED, electroimanes, ventiladores o células Peltier. Su diseño incluye un divisor de tensión simple con resistencias de 30K y 7.5K, lo que significa que la tensión que llega al módulo se divide por un factor de 5 ($7.5/(30+7.5)$).

Para garantizar un funcionamiento seguro, es importante tener en cuenta que la tensión máxima que se puede medir es de 25V con un procesador que tiene una tensión de alimentación de 5V (Vcc), y 16.5V con uno de 3.3V (Vcc). Superar estos límites podría dañar el pin analógico de un Arduino. La resolución del módulo es de 24.45 mV, lo que permite una medición detallada. La **Figura 9** proporciona una guía visual para entender la configuración y la ubicación de este sensor en su aplicación práctica.



Figura 9. Sensor de voltaje.

Fuente: (Escobar, 2019).

Características técnicas destacadas:

- Rango de entrada de voltaje: 0 V a 25 V DC.
- Voltaje de detección máximo: 25 V.
- Rango de detección de voltaje: 0.02445 V~25 V.
- Resolución analógica de tensión: 0.00489 V.
- Voltaje de detección mínimo: 0.02445 V.

4.3.2 Software de programación.

4.3.2.1 Matlab

Matlab, desarrollado por Mathworks, es una plataforma de programación y cálculo enfocada en ingeniería y matemáticas.

Su capacidad de calcular y representar gráficamente problemas complejos, lo que minimiza la necesidad de realizar cálculos manuales, como el determinante de una matriz. Este software utiliza su propio lenguaje y archivos con extensión, y su funcionalidad se amplía

mediante toolboxes. Matlab es útil para manejar operaciones comunes, como vectores y matrices, y también para programación orientada a objetos (Llamas, 2023).

Un script en Matlab es un archivo que contiene una secuencia de comandos Matlab, guardado en un archivo de texto plano con extensión ".m". Estos archivos contienen una serie de comandos que se ejecutan en orden secuencial cuando se llama al script.

Un script es una forma de organizar y ejecutar una serie de comandos en Matlab, facilitando la automatización de tareas y la reutilización de código. Los scripts son útiles para realizar operaciones repetitivas o para ejecutar un conjunto específico de acciones en un orden determinado (MathWorks, 2024).

4.3.2.1.1 Simulink

Simulink, creado por MathWorks, es una herramienta para modelar y simular sistemas mediante diagramas de bloques. Es ideal para el diseño, la simulación y el análisis de sistemas dinámicos en distintos dominios, como eléctrico, mecánico, térmico y de control.

Cuenta con una amplia biblioteca de bloques predefinidos para diversas aplicaciones, ofreciendo mucha flexibilidad. También permite a los usuarios diseñar bloques personalizados para necesidades específicas.

Entre las ventajas de Simulink es la facilidad de uso, gracias a una interfaz intuitiva, y su capacidad para acelerar el desarrollo permitiendo realizar pruebas y análisis antes de la implementación. Además, permite modelar sistemas que van desde los más simples hasta los más complejos. Presenta como una herramienta completa para ingenieros y profesionales que buscan una solución eficaz para el diseño y análisis de sistemas dinámicos (MathWorks, 2024).

4.3.2.1.2 Design App

App Designer de Matlab es una aplicación⁵ que posibilita a los usuarios crear aplicaciones profesionales sin necesidad de ser expertos en desarrollo de software. Integra la disposición visual de la interfaz gráfica de usuario (GUI) con la programación de la aplicación (MathWorks, 2024).

Características destacadas:

- Interfaz intuitiva: Facilita la creación de aplicaciones.
- Incluye botones, cuadros de texto, listas y gráficos, entre otros.
- Permite la escritura de código para controlar el comportamiento.
- Facilita la creación de escritorio o web.

Beneficios:

- Facilidad de uso.
- Adaptable a diversas aplicaciones.
- Acelera el desarrollo al diseñar la interfaz y programar simultáneamente.

4.4 Capítulo IV: Modelados para motores DC con escobillas o de imanes permanentes.

4.4.1 Método teórico

4.4.1.1 Resistencia de armadura

La resistencia de armadura se puede medir con un multímetro digital, conectando las terminales, sin alimentación de voltaje. (Bellón, 2023).

4.4.1.2 Inductancia de armadura

La inductancia de armadura (L_a) al igual que la resistencia de armadura se puede medir en los devanados del motor (en los cables rojo y negro), utilizando un instrumento que sirve para medir inductancias (Quijano y Hernández, 2009).

4.4.1.3 Constante electromotriz

Este tipo de constante se asocia con la contraelectromotriz (E_a), que es la fuerza electromotriz inducida en la bobina de un motor eléctrico. La relación entre la contraelectromotriz (E_a), la velocidad angular ($\omega(t)$), la tensión (v), la corriente (i) y la resistencia (R), como se observa en la **Figura 10** y en la **Ecuación 1**.

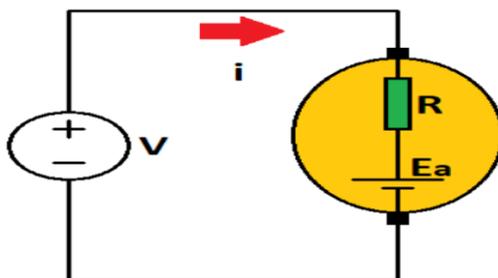


Figura 10. Constante electromotriz.

$$K_a = \frac{E_a}{\omega(t)} \quad (1)$$

Aplicando la ley de Kirchoff se obtiene que $E_a = v - iR$ sustituyendo en la **Ecuación 1**, se puede obtener la **Ecuación 2**.

$$K_a = \frac{v - iR}{\omega(t)} \quad (2)$$

4.4.1.1 Constante de torque

En motores de corriente continua constante de torque es igual a la constante electromotriz, como se muestra en la **Ecuación 3** (Castaño, 2019).

$$K_a = K_m \quad (3)$$

4.4.1.2 Constante de tiempo mecánica

Dada la conexión entre la tensión de armadura y la velocidad del eje del motor, al aplicar un cambio en el voltaje del motor, se puede verificar una respuesta transitoria de la velocidad en condiciones de carga constante. A través de esta respuesta, es posible identificar la constante de tiempo, que representa el 63.2% del proceso de ajuste hasta alcanzar el estado estacionario.

En el caso de un motor, es necesario conectar un osciloscopio a los bornes del motor y aplicar escalones de voltaje entre +5 V y 0 V para observar su dinámica. Se puede lograr utilizando un generador de señales o mediante la programación de un código, como en el caso de controlar motores con Arduino. También es posible conectar directamente el motor a la fuente de alimentación y capturar el instante preciso en que se produce la transición de voltaje (Castaño, 2019).

4.4.1.3 Momento de inercia (J)

El momento de inercia (J) de un motor de corriente continua (DC) determina la capacidad para cambiar su velocidad de rotación en respuesta a diferentes condiciones. La fórmula que modela esta relación se expresa en la **Ecuación 4**:

$$J = \frac{T_m * K_a * K_m}{R} \quad (4)$$

En esta ecuación, J representa el momento de inercia, una medida de la distribución de masa en relación con el eje de rotación. El término T_m denota el par de carga del motor, que es la fuerza motriz que impulsa su rotación. La constante de contraelectromotriz (K_a), también conocida como constante electromotriz (E_a), está relacionada con la generación de voltaje inducido en la bobina del motor durante su rotación. Por otro lado, la constante de torque (K_m) conecta la corriente que fluye a través del motor con el par que se produce. Finalmente, R es la resistencia eléctrica de la bobina del motor.

La fórmula muestra el momento de inercia que está influenciado por factores como el par de carga, la constante de contraelectromotriz, la constante de torque y la resistencia. Analizar esta relación proporciona una comprensión de la dinámica y la capacidad del motor frente a cambios en estas variables. Su aplicación es fundamental para entender el rendimiento de motores de corriente continua en diversas aplicaciones industriales y tecnológicas (Castaño, 2019).

4.4.1.4 Corriente de arranque y torque de fricción

Para iniciar el motor, se aplica una variación de voltaje de corriente continua (DC) a través de sus terminales. La tensión se incrementa gradualmente hasta que se observa el inicio del giro del eje del motor. En este punto, se detiene el aumento de voltaje y se registra la lectura del amperímetro para determinar la corriente necesaria para el arranque del motor, denotada como i_{arr} .

Con esta información, se puede calcular el torque de fricción (T_f) utilizando la **Ecuación 5**.

$$T_f = K_m * i_{arr} \quad (5)$$

K_m representa la constante de torque del motor. Este procedimiento permite cuantificar el torque de fricción, una medida para comprender la resistencia inicial que debe vencer el motor durante el proceso de arranque (Castaño, 2019).

4.4.1.5 Constante de Fricción de Coulomb

La constante de fricción de Coulomb representa la resistencia al movimiento entre superficies en contacto, para caracterizar la fricción seca. En la **Ecuación 6**, B (Constante de fricción de Coulomb) se calcula como el producto de la velocidad angular instantánea ($\omega(t)$) y el torque mecánico total (T_m), dividido por el torque de fricción (T_f).

$$B = \frac{T_m - T_f}{\omega(t)} \quad (6)$$

4.4.2 Modelo electromecánico.

El motor de corriente directa consiste con una armadura, escobas y bobinas de campo en serie, paralelo o la combinación de ellas, el cual viene siendo usado por muchos años como un convertidor básico de energía. Estos motores son usados en elevadores eléctricos, laminadores, vehículos eléctricos y algunas bombas donde se requiere de velocidad variable; Los elementos más importantes de un motor DC, la Reductora y la carga del motor, representado en la Figura 11.

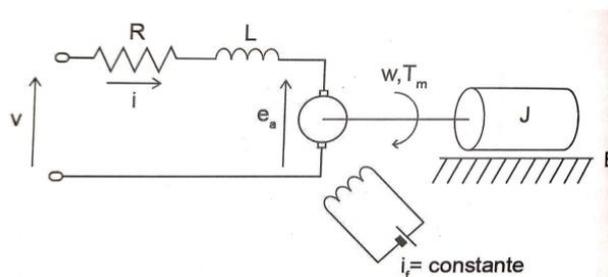


Figura 11. Motor DC.

Fuente: (Huelin, Gutiérrez y Larraga, 2022).

La armadura del motor DC se modela una resistencia constante R en serie con una inductancia constante L que representa la inductancia de la bobina de la armadura, y una fuente de alimentación v que representa la tensión generada en la armadura.

La primera ecuación se realiza haciendo un análisis de la malla del circuito, como se observa en la **Ecuación 7** (Castaño, 2019).

$$v(t) = Ri(t) + L \frac{di(t)}{dt} + E_a(t) \quad (7)$$

Despejando $L \frac{di(t)}{dt}$ obtenemos la **Ecuación 8**.

$$L \frac{di(t)}{dt} = v(t) - Ri(t) - E_a(t) \quad (8)$$

Donde $E_a(t)$ (Fuerza contraelectromotriz [volts]) es una tensión generada que resulta cuando los conductores de la armadura se mueven a través del flujo de campo establecido por la corriente del campo i_f

Naturalmente, en toda potencia mecánica desarrollada en el rotor se entrega a la carga mecánica conectado al eje del motor de CC.

Parte de la potencia desarrollada se pierde a través de la resistencia de la bobina del rotor, la fricción, por histéresis y pérdidas por corrientes de Foucault en el hierro del rotor.

Desde aquí las pérdidas por fricción y parte de la energía desarrollada es almacenada como energía cinética en la masa girante del rotor. La ecuación de la sección mecánica viene dada por el modelo que se observa en la **Ecuación 9**:

$$T_m(t) = J \frac{d\omega(t)}{dt} + B\omega(t) \quad (9)$$

Despejando $J \frac{d\omega(t)}{dt}$ se obtiene la **Ecuación 10**.

$$J \frac{d\omega(t)}{dt} = T_m(t) - B\omega(t) \quad (10)$$

Donde $T_m(t)$ es el torque del motor de corriente continua, B es el coeficiente de fricción equivalente al motor de CD (corriente continua) y la carga montados sobre el eje del motor, J es el momento de inercia total del rotor y de la carga con relación al eje del motor, $\omega(t)$ es la velocidad angular del motor y $\frac{d\omega(t)}{dt}$ es la aceleración angular.

Para poder lograr la interacción entre las ecuaciones anteriores se proponen las siguientes relaciones que asumen que existe una relación proporcional, K_a (Constante contraelectromotriz [v/rad s]), entre el voltaje inducido en la armadura y la velocidad angular del eje del motor como se muestra en la **Ecuación 11**.

$$E_a(t) = K_a \omega(t) \quad (11)$$

Se supone la siguiente relación electromecánica que establece que el torque mecánico es proporcional, K_m (Constante de Torque [Nm / A]), a la corriente eléctrica que circula por el motor DC, esto se lo puede expresar como se indica en la **Ecuación 12** (Castaño, 2019).

$$T_m(t) = K_m i(t) \quad (12)$$

4.4.3 Funciones de transferencia del motor DC

Para lograr la correcta derivación de las ecuaciones de transferencia del motor de corriente continua, se requiere emplear la transformada de Laplace en las Ecuaciones 8, 10, 11 y 12, lo que conduce a la obtención de las **Ecuaciones 13,14, 15, 16** respectivamente:

$$Lsi(s) = v(s) + Ri(s) + E_a(s) \quad (13)$$

$$Js\omega(s) = T_m(s) + B(s) \quad (14)$$

$$E_a(s) = K_a \omega(s) \quad (15)$$

$$T_m(s) = K_m i(s) \quad (16)$$

Luego de obtener estas ecuaciones, se procede a sustituir e integrar las ecuaciones 15 y 16 en la ecuación 13, obtenido como resultado la **Ecuación 17**.

$$Ls \frac{T_m(s)}{K_m} = v(s) - R \frac{T_m(s)}{K_m} - K_a \omega(s) \quad (17)$$

Despejando $v(s)$ de la ecuación 17, se llega a obtiene la formulación de la **Ecuación 18**.

$$v(s) = \frac{(R + Ls) T_m(s)}{K_m} + K_a \omega(s) \quad (18)$$

Al despejar $\omega(s)$ de la ecuación 14, se alcanza la expresión de la velocidad angular, la cual está representada en la **Ecuación 19**.

$$\omega(s) = \frac{T_m(s)}{Js + B} \quad (19)$$

Posteriormente, se procede a sustituir la ecuación obtenida en la ecuación 18, resultando la **Ecuación 20**.

$$v(s) = \frac{(R + Ls) T_m(s)}{K_m} + K_a \frac{T_m(s)}{Js + B} \quad (20)$$

Al aplicar la propiedad distributiva, se obtiene como resultado la **Ecuación 21**.

$$v(s) = \frac{(R + Ls)(Js + B) + K_a K_m}{K_m(Js + B)} T_m(s) \quad (21)$$

En base a la ecuación 21 se obtiene las funciones de transferencia con salida de torque, fuerza contraelectromotriz, corriente, velocidad angular y posición. Todas estas ecuaciones están íntegramente vinculadas con la entrada de voltaje como se puede evidencia en las **Ecuaciones 22, 23, 24, 25 y 26** (Castaño, 2019).

- Torque vs voltaje.

$$\frac{T_m(s)}{v(s)} = \frac{K_m(Js + B)}{LJs^2 + (RJ + LB)s + RB + K_m K_a} \quad (22)$$

- Fuerza contraelectromotriz vs voltaje.

$$\frac{E_a(s)}{v(s)} = \frac{K_m K_a}{LJs^2 + (RJ + LB)s + RB + K_m K_a} \quad (23)$$

- Corriente de armadura vs voltaje

$$\frac{i(s)}{v(s)} = \frac{Js + B}{LJs^2 + (RJ + LB)s + RB + K_m K_a} \quad (24)$$

- Velocidad angular vs voltaje.

$$\frac{\omega(s)}{v(s)} = \frac{K_m}{LJs^2 + (RJ + LB)s + RB + K_m K_a} \quad (25)$$

- Posición vs voltaje.

$$\frac{\theta(s)}{v(s)} = \frac{K_m}{s(LJs^2 + (RJ + LB)s + RB + K_m K_a)} \quad (26)$$

4.5 Capítulo V:

4.5.1 Metodología para el desarrollo de actividades prácticas y de investigación.

Las actividades prácticas y de investigación que desarrollen los estudiantes en el laboratorio como parte del currículo, deben guardar relación con el plan de estudios de la carrera, de manera que se demuestre la funcionalidad del laboratorio. Para ellos, los estudiantes deben disponer de una guía que oriente y garantice los resultados de aprendizaje previstos en los sílabos. La guía para la práctica de laboratorio forma parte del presente documento y es entregada al estudiante al inicio de clases durante el encuadre de la unidad.

La realización de la actividad práctica implica ejecutar las siguientes actividades: Preparatorio, desarrollo e informe.

4.5.1.1 Preparatorio

El preparatorio es la actividad previa que desarrolla el estudiante como trabajo autónomo antes de realizar la práctica.

Para asistir a la práctica es obligatorio que el estudiante revise la guía de la práctica a desarrollar y elabore un documento que contemple al marco teórico de la misma. La guía señala los temas a consultar.

4.5.1.2 Desarrollo

El desarrollo consiste en realizar actividades mediante una guía práctica que facilita la correcta ejecución de los procesos y permite obtener los resultados esperados.

4.5.1.3 Informe

El informe también es una actividad basada en los requerimientos de la guía práctica e incluye los resultados obtenidos durante el desarrollo.

5. Metodología

5.1 Área de trabajo

La presente investigación se desarrolló en las instalaciones de la Universidad Nacional de Loja, específicamente en el edificio de laboratorios de la Facultad de Energía, las Industrias y los Recursos No Renovables, su ubicación se encuentra visualizada en la **Figura 12**. El mismo que se sitúa en la ciudad de Loja, cantón Loja.

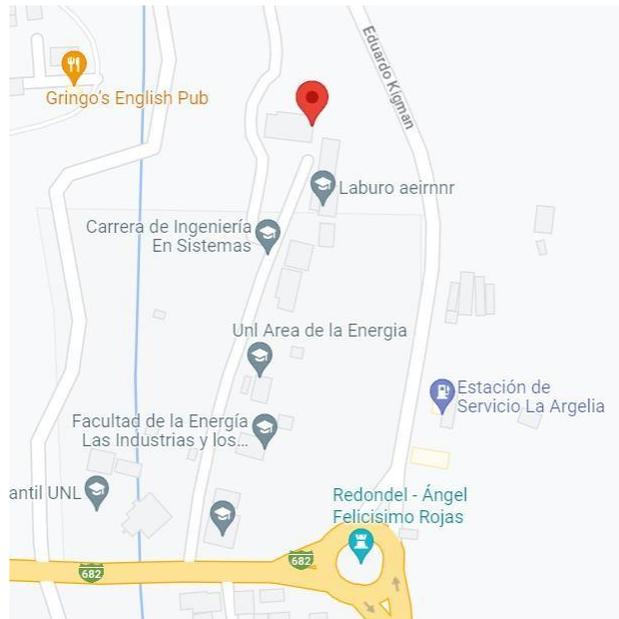


Figura 12. Ubicación de la FERNNIR.

Fuente: (Google Maps, s. f.).

5.2 Equipos y materiales

5.2.1 Software

- Autodesk: Inventor: Diseño 3D
- Mathcad: Realización de cálculos.
- Matlab: Programación y análisis matemático.
- Arduino: Utilizado para programar placas de Arduino.

5.2.2 Instrumentos de medición.

- Multímetro digital Truper MUT-830: Mide voltaje, corriente, resistencia, entre otros.
- Manómetro analógico: Mide presión de fluidos.
- Anemómetro UT363: Mide velocidad del aire.

5.3 Procedimiento

5.3.1 Primer objetivo

5.3.1.1 Exploración de componentes necesarios para la implementación en el banco

Para el desarrollo del primer objetivo se realizó una revisión de documentos bibliográficos relacionados con:

- Bancos didácticos
- Curvas tacodinámicas
- Generación de algoritmos con Matlab
- Adquisición de datos y control con Arduino.

Esta información fue recopilada especialmente en:

- Tesis y Artículos científicos relacionados con el tema de estudio.
- Páginas web.
- Catálogos y fabricantes de motores.

5.3.1.2 Diseño

El diseño del banco se estructuró en dos fases: diseño conceptual y diseño detallado, cada una enfocada en diferentes aspectos para su desarrollo.

Diseño Conceptual del Banco

Se desarrollaron ideas de manera conceptual, como:

- Los tipos de torque involucrados.
- Las máquinas necesarias.
- La estructura general del banco para garantizar su correcto funcionamiento.

Diseño Detallado

Se determinaron los parámetros técnicos y operativos del banco, como:

- Potencia.
- Caudales.
- Pesas.
- Velocidad.
- Transmisiones.
- Fuente de alimentación.
- Software de diseño.
- Tipo de máquinas según las curvas de torque.
- Sensores para la medición de variables mecánicas y eléctricas.

5.3.1.3 Construcción del banco didáctico

El banco didáctico se divide en varias zonas según su funcionalidad:

- **Zona de máquinas:** Destinada a la instalación de las máquinas.
- **Zona de utilería:** Área donde se encuentra el equipo y herramientas adicionales.
- **Zona de alimentación eléctrica:** Espacio dedicado a la conexión y distribución de la energía eléctrica.
- **Estructura general:** Contiene y soporta todo el conjunto del banco.

Como estrategia de construcción, se utilizaron las siguientes técnicas:

- Impresión 3D.
- Fresado.
- Torneado.

5.3.2 Segundo objetivo

La recopilación de datos del banco se divide en recopilación de datos y desarrollo de interfaces gráficas.

5.3.2.1 Recopilación de datos

- Validación de características de motores
- Mediciones experimentales de voltaje, corriente.
- Calibración de sensores
- Desarrollo de modelo matemático

5.3.2.2 Desarrollo de interfaces gráficas

Para el desarrollo de interfaces gráficas y obtención de datos se utilizó Matlab y Arduino para el que se siguió los siguientes pasos:

- Instalar el paquete de soporte de Matlab y software Arduino.
- Configurar dispositivos de Arduino y sensores según sea necesario.
- Desarrollar una interfaz gráfica utilizando la herramienta GUIDE o código manual.
- Configurar la comunicación serial en Matlab con Arduino.
- Insertar el código en Matlab para enviar comandos al Arduino y recibir los datos adquiridos.
- Actualizar la interfaz gráfica en Matlab para mostrar los datos recibidos en tiempo real.
- Depurar cualquier error encontrado.
- Optimizar código.

5.3.3 Tercer objetivo

5.3.3.1 Análisis de requerimientos de las guías prácticas

El criterio de análisis de requerimientos de las guías prácticas se basa en el PLAN CURRICULAR del modelo general de evaluación de carreras del CACES (Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior) de la Universidad Nacional de Loja contempla el indicador “Prácticas en relación a las asignaturas”.

Las prácticas deben planificarse en correspondencia con el sílabo de cada asignatura como evidencias del indicador se debe tomar en cuenta que las “Guías de prácticas de laboratorios/ talleres / centros de simulación”.

En el desarrollo del diseño, construcción y obtención de datos para el banco didáctico orientado al análisis del par resistente de máquinas, se implementaron distintas fases de manera sistemática. Cada una de estas etapas fue planificada para alcanzar los objetivos planteados. Con el fin de proporcionar una comprensión más detallada de todo el proceso, se desarrolló un flujograma que se presenta en la **Figura 13**. Este diagrama ofrece una guía clara para la interpretación de las distintas etapas involucradas en el desarrollo del proyecto.

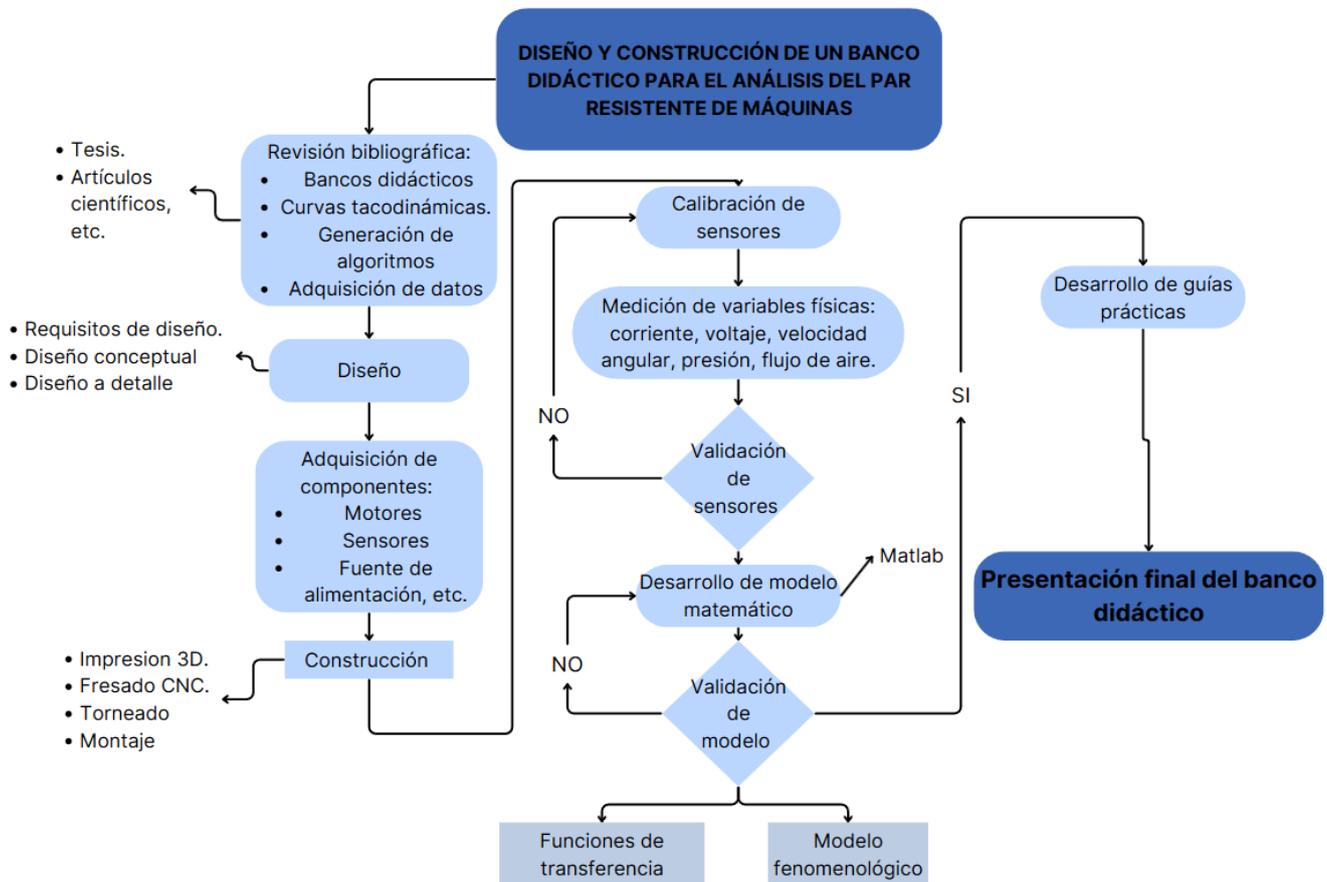


Figura 13. Resumen de procedimientos.

5.4 Procesamiento y análisis de datos.

La obtención y análisis de datos se realizó mediante la extracción de las variables físicas y eléctricas del sistema de adquisición para luego ser procesadas mediante la interfaz de Matlab con el fin de visualizar las curvas características de los motores. Para una mejor comprensión se desarrolló el flujograma que se visualiza en la **Figura 14**.

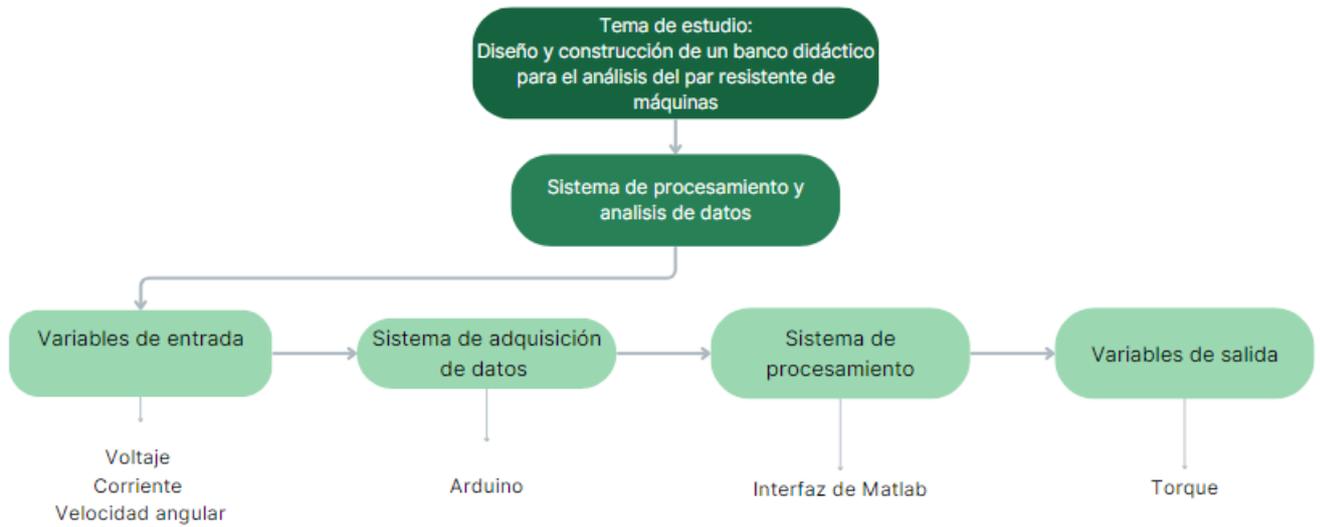


Figura 14. *Análisis de datos*

6. Resultados

6.1 Exploración de componentes necesarios para la implementación en el banco

Una vez explorado los componentes del banco didáctico para el análisis de par resistente de máquinas se desarrolló una tabla con las posibles máquinas que se pueden incorporar con el objetivo de mostrar los diferentes casos de torque como se visualiza en la **Tabla 1**.

Tabla 1. Máquinas de posible implementación en el banco

	Clase	Tipo	Dimensión	Impulsión	Capacidad	Referencia
Torque constante	Grúa	Mini Tudou Juguete	16.42" x 9.13" x 7.24"	No aplica	17.64 oz.	(Amazon, s.f.)
		Bomba de agua de diafragma	No aplica	300 cm	1.5 a 2.1 L/min	(Grupo Electrostore, s.f.)
Torque lineal	Bombas de desplazamiento positivo	CrocSee DC 12V	28"x 1.06" x 1.06"	198.12 cm	1.3 L/min	(Amazon, s.f.)
		VerRich-Bomba de diafragma	6.5"x 3"x 1.86"	no aplica	6 L/min	(Amazon, s.f.)
		Bomba de taladro	4.3" x 15"x 3"	No aplica	30-40 L/min	(Amazon, s.f.)
		Ventilador	Fenghao	0,98" x 4.72"x 4.72"	No aplica	no aplica
Torque parabólico	Bombas de impelente (centrifugas)	Vansuna	4.33" x 3.15" x 2.64"	22 pies	490 L/min	(Amazon, s.f.)
		Bordstract	2.48"x 2.17"x 4.96"	No aplica	20 a 30 L/min	(Amazon, s.f.)
		Brushless	2.16" x 2" x 1.57"	300cm	4 L/min	(Grupo Electrostore, s.f.)
		Brushless	1.46" x 1.1" x 1.85"	20 cm	2 L/min	(Grupo Electrostore, s.f.)

6.2 Requisitos de diseño

6.2.1 Descripción de las características de los tipos de motores.

6.2.1.1 Motores adaptados para el accionamiento de grúas:

Para el accionamiento de una máquina tipo grúa, se selecciona un motor acoplado a un sistema de transmisión para la reducción de velocidad angular conocido también como

motorreductor por las ventajas que ofrece, especialmente su elevado torque y alta reducción de velocidad.

6.2.1.2 Motores para bombas de desplazamiento positivo:

Los requisitos de diseño, que se ha identificado para el correcto funcionamiento de una bomba de desplazamiento positivo incluyen la necesidad de altas velocidades, lo que garantiza un desempeño constante y eficiente del flujo del fluido.

6.2.1.1 Motores para ventiladores:

Para accionar ventiladores de computadora el par de arranque es bajo y requiere altas velocidades de rotación superiores a 2000 rpm, por lo cual, se descarta el acoplamiento de un reductor para garantizar un rendimiento eficiente y un flujo de aire constante.

6.2.2 Descripción de las máquinas.

En la elección de los sistemas que se agregaron en el banco, una de las opciones más adecuadas para conseguir un torque constante fue el sistema de elevación de carga.

Por otro lado, para obtener un torque lineal, que se genera mediante la acción de succión y expulsión que impulsa el fluido a través del sistema, se seleccionó la bomba de diafragma.

Finalmente, se tomó la decisión de seleccionar un sistema de ventilación para obtener un torque cuadrático.

6.2.3 Requerimientos de diseño

Después de realizar una exploración y análisis de la bibliografía para la elección de los elementos adecuados para su funcionamiento, se procede a detallar las características de cada uno de los componentes que va a formar parte del banco didáctico.

6.2.4 Máquinas

6.2.4.1 Grúa

A continuación, se muestra los detalles de la grúa y una foto en la **Figura 16**.

- Material de soportes: PLA.
- Resistencia del material: 55 MPa.
- Material de eje: Acero inoxidable.
- Diámetro de eje: 8 mm.
- Material de tambor: PLA.



Figura 15. *Grúa.*

6.2.4.1.1 *Bomba de diafragma*

En la **Figura 15** se visualiza la bomba de diafragma y las características del mismo.

- Modelo: DP – 52.
- Presión: 0.48 MPa.
- Caudal: 3.5 L/min.
- Succión: 1.5 m verticales.
- Elevación: 4 – 5 m verticales.
- Material: Plástico ABS.
- Conexión de manguera: 10 mm.
- Dimensiones: 1.25 x 95.5 x 55 mm.



Figura 16. *Bomba de diafragma.*

6.2.4.2 Ventilador

En la **Figura 17** se indica una fotografía del ventilador y un listado de sus características.

- Material de ventilador: Plástico Termoestable.
- Voltaje: 12 voltios DC.
- Corriente: 0.50 A.
- Dimensiones: 4.75" x 1.5".



Figura 17. Ventilador.

6.2.4.3 Elementos motrices

6.2.4.3.1 Motor acoplado a reductor (Motor para accionar grúa)

A continuación, se indica las características del motorreductor que es utilizado para realizar el funcionamiento de la grúa (*Grupo Electrostore*, 2024) y en la **Figura 18** se visualiza:

- Voltaje: 3-12V.
- Velocidad: 1-1500 rpm.
- Par: 2.15 kgf x cm.
- Potencia de salida: 0.1-5 W.
- Relación de la caja: 1:73.6.
- Longitud del motor: 43.5 mm.
- Diámetro del eje: 6 mm.
- Consumo de corriente nominal: 500 mA.



Figura 18. Motor acoplado a reductor con conmutación por escobillas.

6.2.4.3.2 Motor RS775 (Motor para accionar bomba)

El motor RS 775 que se puede observar en la **Figura 19** es un dispositivo de alto par y alta velocidad, diseñado con un eje en forma “D” que posibilita su acople seguro a las poleas, piñones, ruedas, bielas y otros componentes, permitiendo la transmisión de su potencia. A continuación, se presentan las características detalladas de este motor (Julpin, 2022).

- Corriente sin carga: 0.27 A.
- Velocidad de giro sin carga: 4500 rpm.
- Potencia de salida: 24 W.
- Diámetro total del eje: 5 mm.
- Borde biselado del eje: 1.5 mm (3.5 mm en eje “D”).
- Diámetro del motor: 42 mm.
- Longitud del motor: 66.5 mm.



Figura 19. Motor RS775 tipo escobilla.

6.2.4.3.3 *Motor doble eje (Motor para ventilador)*

El motor doble eje que se indica en la **Figura 20**, posee un eje delantero que sale de la caja de cambios, en forma de D, baja velocidad y gran torsión.

En el otro lado, el eje trasero que sale del cuerpo del motor, eje redondo, alta velocidad y bajo par, adecuado para instalar codificador de posición giratoria.

Debido a las características del motor, cuenta con bajo ruido y alto par, voltaje múltiple y velocidad múltiple (Amazon, sf).

- Voltaje nominal: 12 V.
- Corriente nominal: 1.5 A.
- Velocidad sin carga: 29 rpm.
- Velocidad nominal: 23 rpm.
- Par nominal: 32 kgf.cm (3140 Nm) (440 oz.in).
- Relación de reducción: 1:270.



Figura 20. *Motor doble eje de metal con conmutación por escobillas*

6.3 **Diseño del banco didáctico**

Para la concepción del sistema que constituirá el banco, se siguió la metodología previamente establecida.

Este sistema fue diseñado para llevar a cabo tres prácticas distintas: una destinada a medir el torque lineal mediante el empleo de una bomba, la otra orientada a evaluar el torque constante mediante el funcionamiento de una grúa y finalmente la que es destinada a evaluar el torque parabólico a través del sistema de ventilación. Con el fin de poder visualizar los componentes se desarrolló el diseño en el software Inventor.

A su vez se elaboró un diseño más detallado que cuenta con dimensiones de cada elemento que se puede visualizar en el **Anexo 7**.

6.3.1 Esquema de sistema de elevación

En la **Figura 21** se visualiza el sistema de transmisión de potencia mecánica entre el motor y la grúa. La construcción en físico se encuentra en la **Figura 22**.

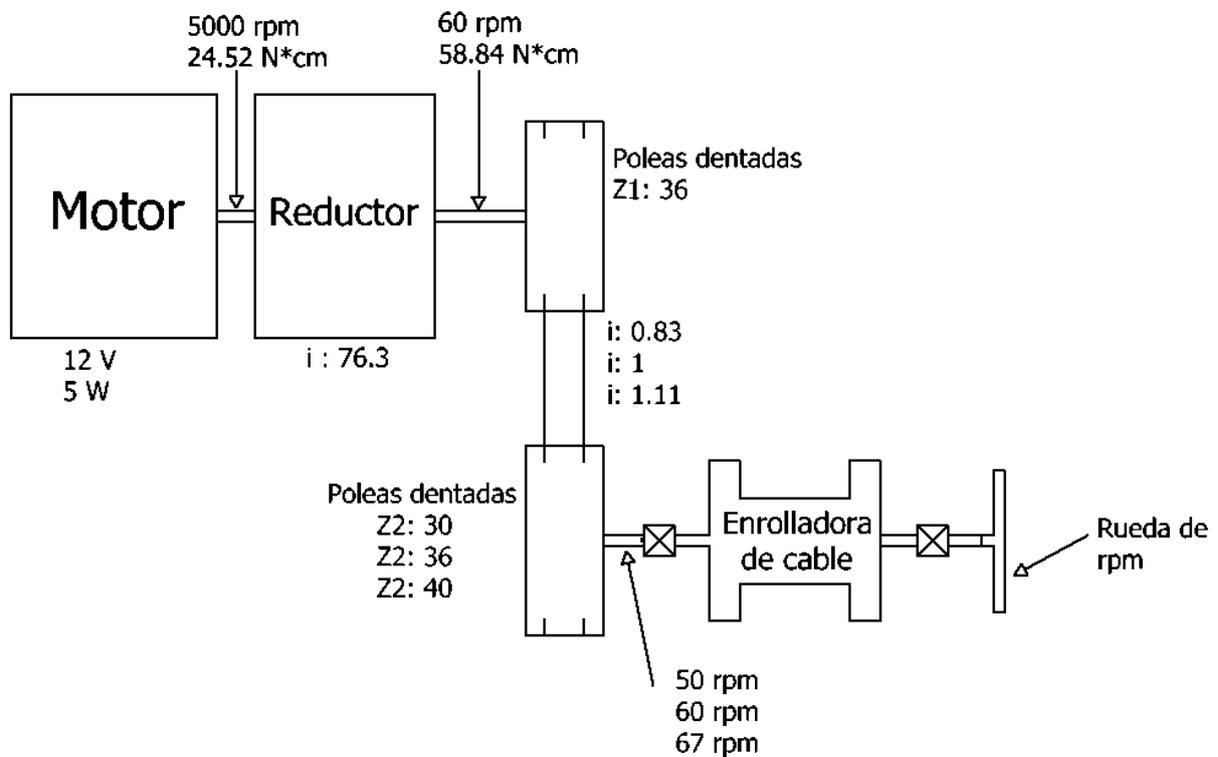


Figura 21. Diagrama cinemático del sistema de elevación.

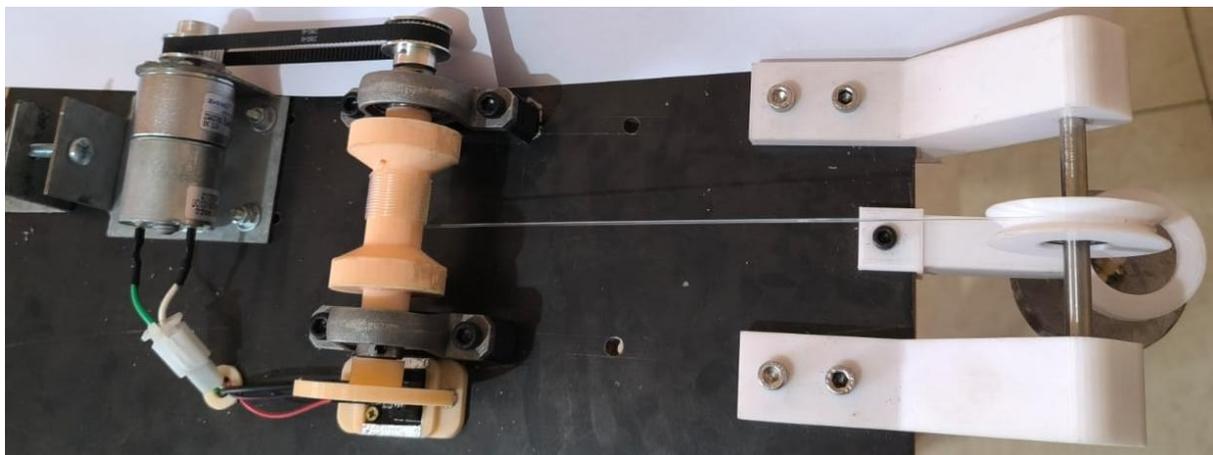


Figura 22. Sistema de elevación.

6.3.2 Esquema de sistema de bombeo

En la **Figura 23** se muestra el diagrama cinemático. En la **Figura 24** se visualiza el sistema construido.

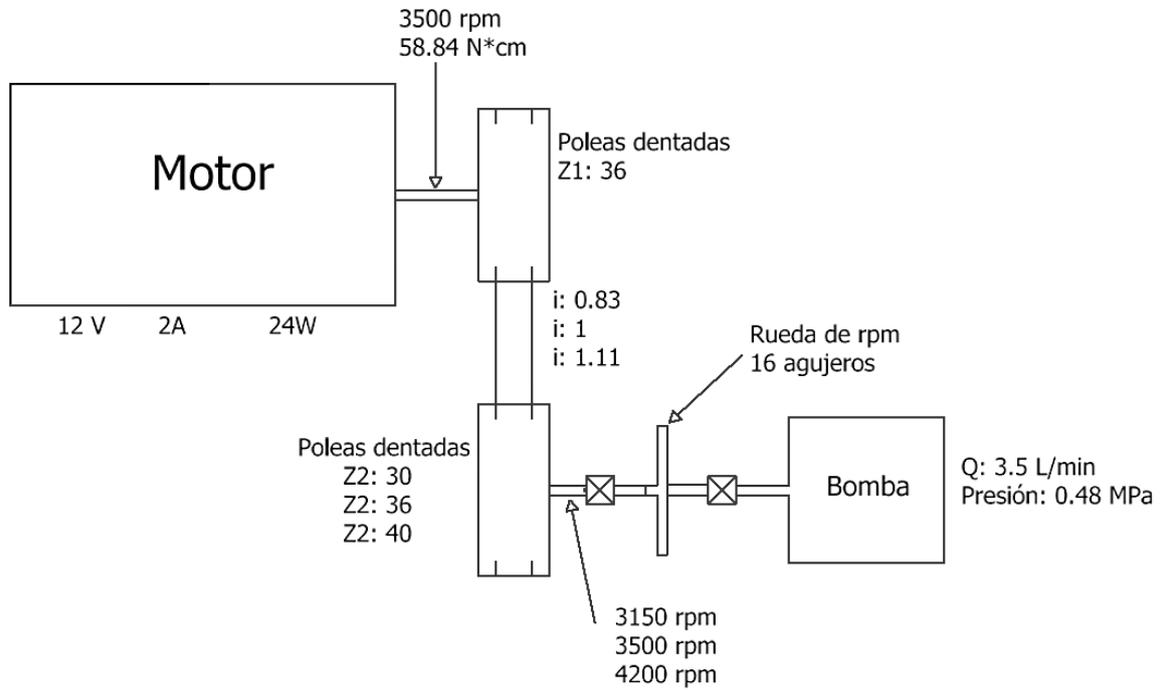


Figura 23. Diagrama cinemático del sistema de bombeo.

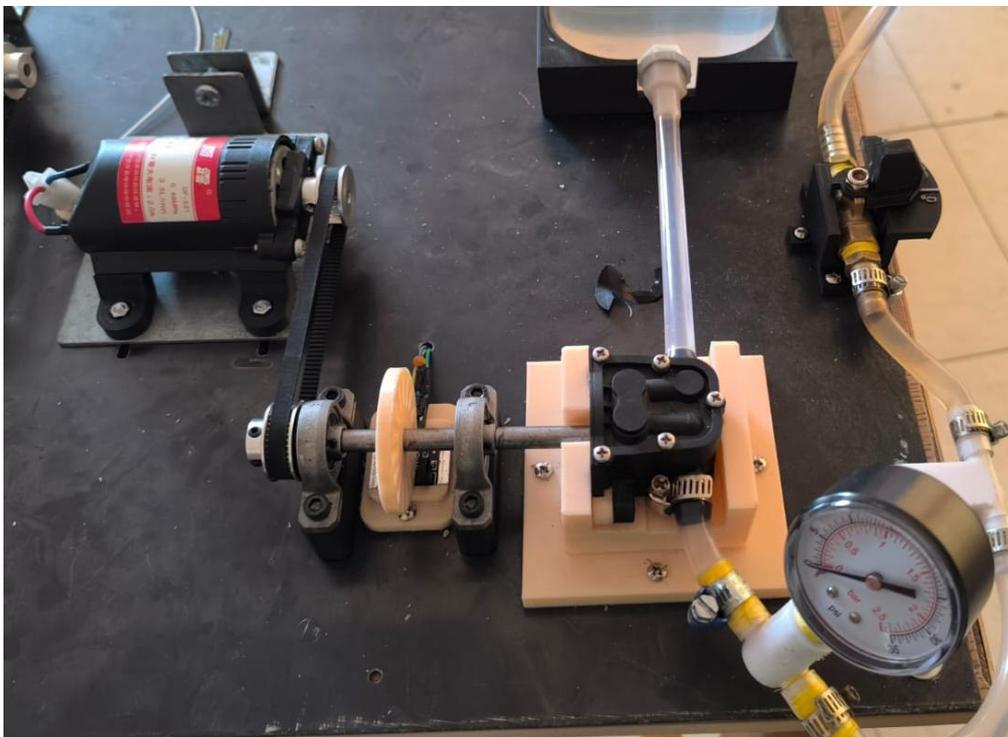


Figura 24. Sistema de bombeo.

6.3.3 Esquema de sistema de ventilación.

A continuación, en la **Figura 25** se observa el diagrama cinemático. En la **Figura 26** su construcción.

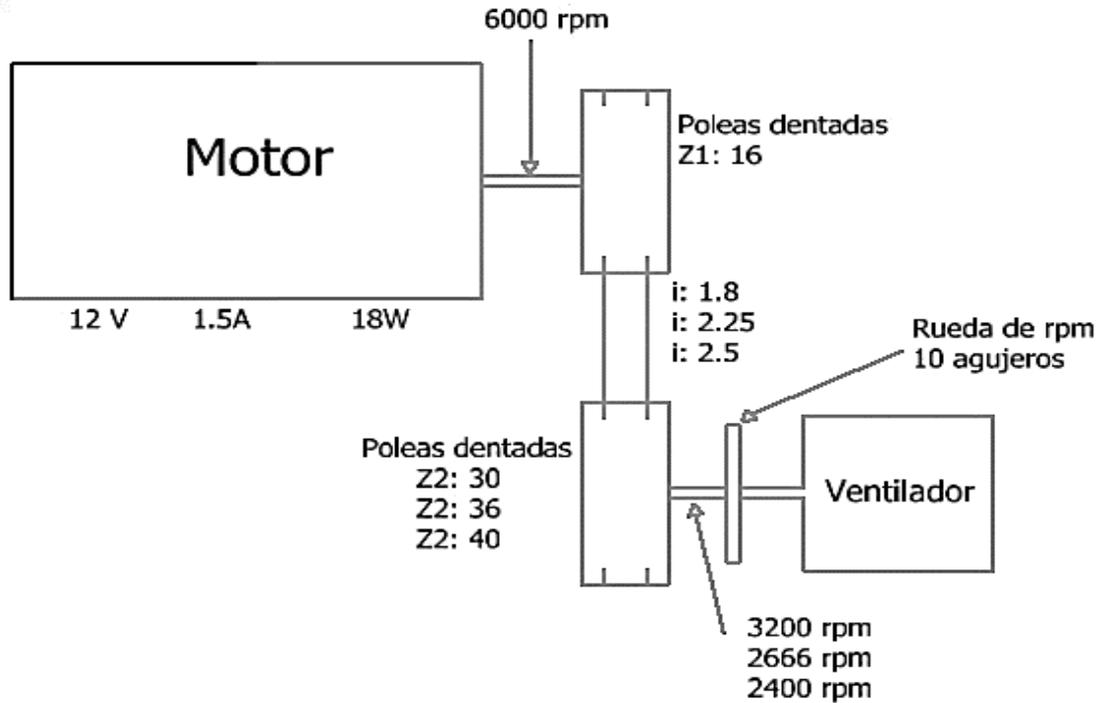


Figura 25. Diagrama de sistema de ventilación.

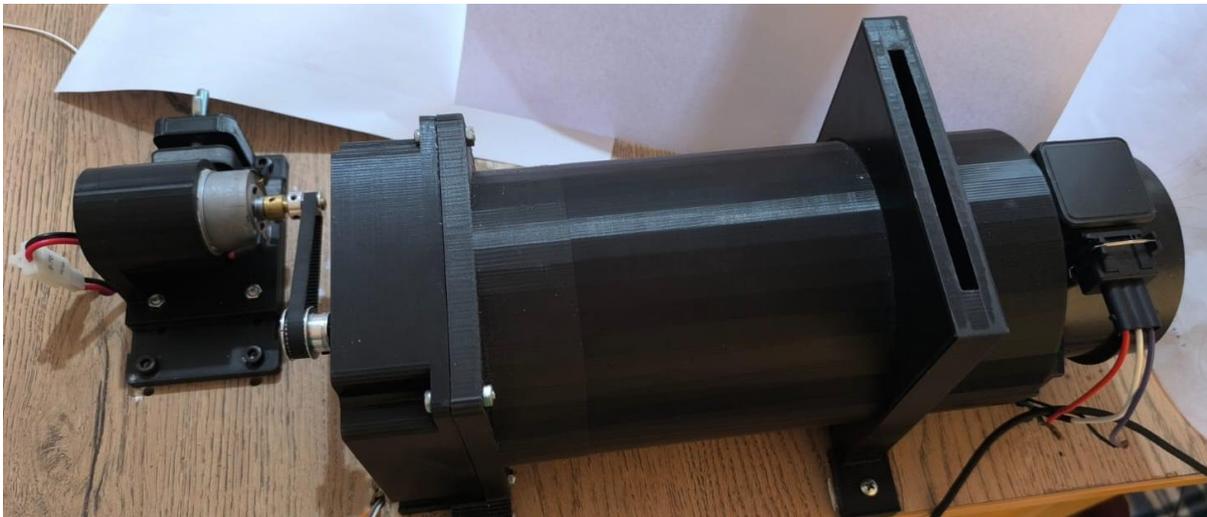


Figura 26. Sistema de ventilación.

6.3.4 Detalles de los elementos que conforman el sistema de elevación, bombeo y ventilación.

Una vez visualizado el sistema de elevación, bombeo y ventilación se procede a desarrollar mediante las **Tablas 2, 3, 4, 5 y 6** detalles sobre implementos del banco como son:

poleas, correas, chumaceras y ejes que se indica de la **Figura 27** a la **Figura 40**, necesarios para transmitir potencia.

6.3.4.1 Ejes

Tabla 2. Características de ejes.

Características de ejes			
Imagen	Nombre	Diámetro	Material
	Eje de motorreductor	6 mm	Acero de transmisión
Figura 27. Eje de motor acoplado a reductor.			
	Eje de motor de bomba	5 mm	Acero de transmisión
Figura 28. Eje de motor de bomba.			
	Eje motor de ventilador	6 mm	Bronce
Figura 29. Eje de motor de ventilador.			
	Eje de bomba	8 mm	Acero de transmisión
Figura 30. Eje de bomba.			
	Eje de tambor	8 mm	PLA
Figura 31. Eje de tambor.			
	Eje de grúa	12 mm	Acero de transmisión
Figura 32. Eje de grúa.			
	Eje de ventilador	8 mm	PLA
Figura 33. Eje de ventilador.			

6.3.4.2 Poleas

Tabla 3. Características de poleas

Características de poleas

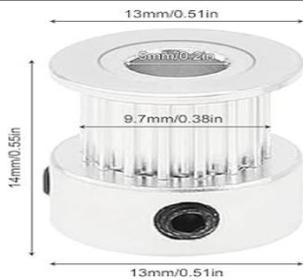
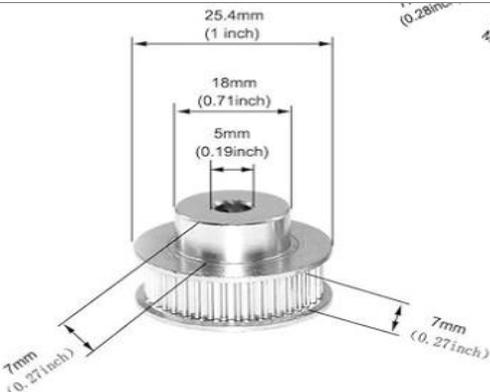
	Dientes	Diámetro exterior	Tipo
	16	0.197"	
	40	1.08"	
	36	1"	Dentada GT2
	30	0.85"	

Figura 34. Polea 16 dientes. Fuente: Amazon, sf.

Figura 35. Polea de 40 dientes. Fuente: Amazon, sf.

Figura 36. Polea de 36 dientes. Fuente: Amazon, sf.

Figura 37. Polea de 30 dientes Fuente: Amazon, sf.

6.3.4.2.1 Relación de transmisión

El emparejamiento de poleas genera una relación de transmisión específica según la combinación utilizada, como se observa en la **Tabla 4**. Se considera una polea conductora de 16 dientes para el sistema de ventilación y de 36 dientes para los otros dos sistemas, además de poleas conducidas de 30, 36 y 40 dientes para los tres sistemas.

Tabla 4. Relación de transmisión.

Relación de transmisión de poleas			
Sistema	Polea 1 conductora	Polea 2 Conducida	Relación
Ventilación	16	30	1.875
		36	2.25
		40	2.5
Bombeo y grúa	36	30	1.8
		36	1
		40	0.9

6.3.4.3 Correas

Tabla 5. Características de correas

Características de correas			
Correa	Longitud	Ancho	Paso
Dentada GT2	280 mm	6 mm	2 mm
	200 mm	6 mm	2 mm



Figura 38. Correa dentada.

6.3.4.4 Chumaceras

Tabla 6. Características de chumaceras

Características de chumaceras				
Chumacera	Material de la carcasa	Material del rodamiento	Diámetro interior	Tamaño de paquete
KP08	Aleación de Zinc	Acero cromado	8 mm	5 x 10 x 5 cm



Figura 39. Chumacera de 8mm.

Fuente: Amazon, sf.

KP001	Aleación de Zinc	Acero cromado	12 mm	7.7 x 6.6 x 2.2 cm
-------	------------------	---------------	-------	--------------------



Figura 40. Chumacera de 12mm.

Fuente: Amazon, sf.

6.3.5 Diseño de piezas

En la **Figura 41**, se visualiza el diseño de la base para el motor del ventilador, previamente seleccionado y ajustado a las dimensiones del banco mediante el software Inventor, lo que facilita la fase de construcción.

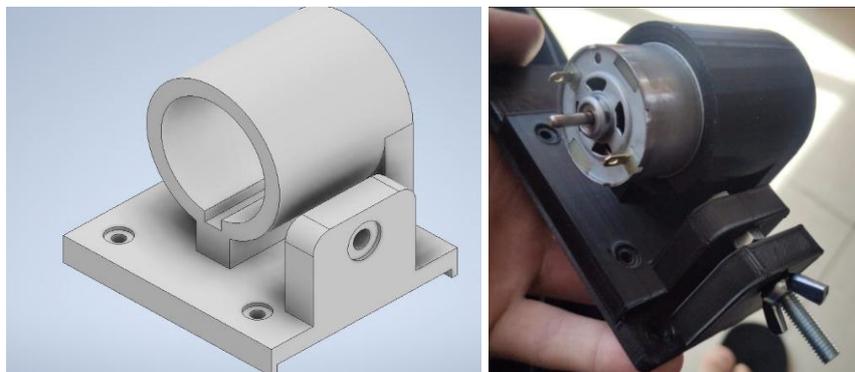


Figura 41. Base de motor de ventilador.

El siguiente componente, representado visualmente en la **Figura 42**, corresponde a una base de bornera de derivación. Este componente está destinado a ser integrado en el banco para proporcionar funcionalidad a todos los sistemas, permitiendo realizar derivaciones rápidas de voltaje hacia los distintos sensores y facilitando así su operación.

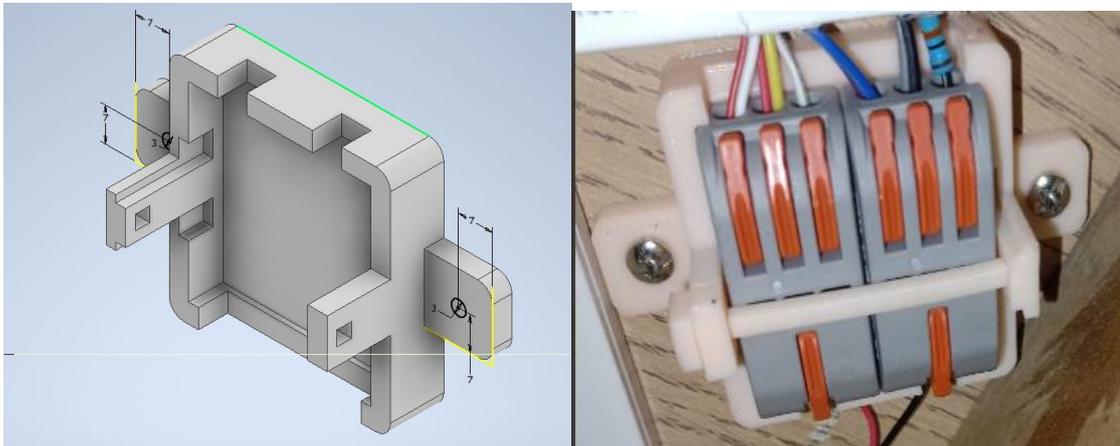


Figura 42. Base de bornera de derivación.

En la representación gráfica proporcionada en la **Figura 43**, se destaca el diseño de la rueda destinada a medir de velocidad angular del motor. La implementación de este componente permite a monitorizar las revoluciones por minuto, al momento de variar la velocidad de los motores mediante el cambio de poleas dentadas en los sistemas.

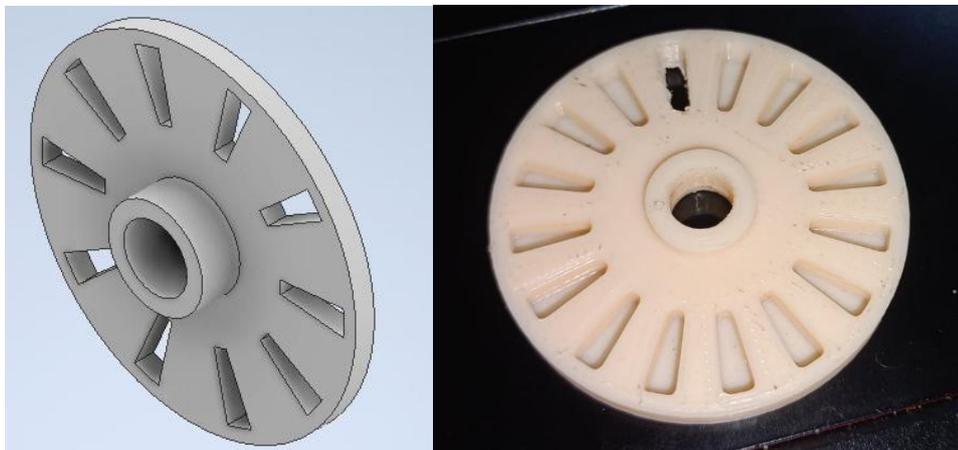


Figura 43. Medidor de revoluciones por minuto.

En la presentación visual proporcionado en la **Figura 44**, se muestra la base del sistema de grúa, elaborado en el software Inventor. Este componente se ensambla en el banco para el funcionamiento del sistema de elevación de carga.

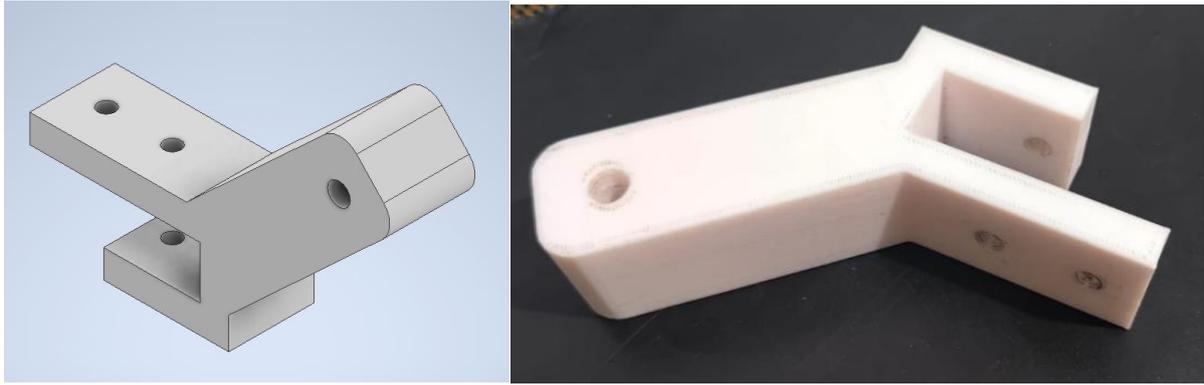


Figura 44. Soportes para la grúa.

En la **Figura 45**, se presenta la base de la bomba, la cual permite fijarla a la mesa, evitando que se desplace cuando el diafragma de la bomba se accione.

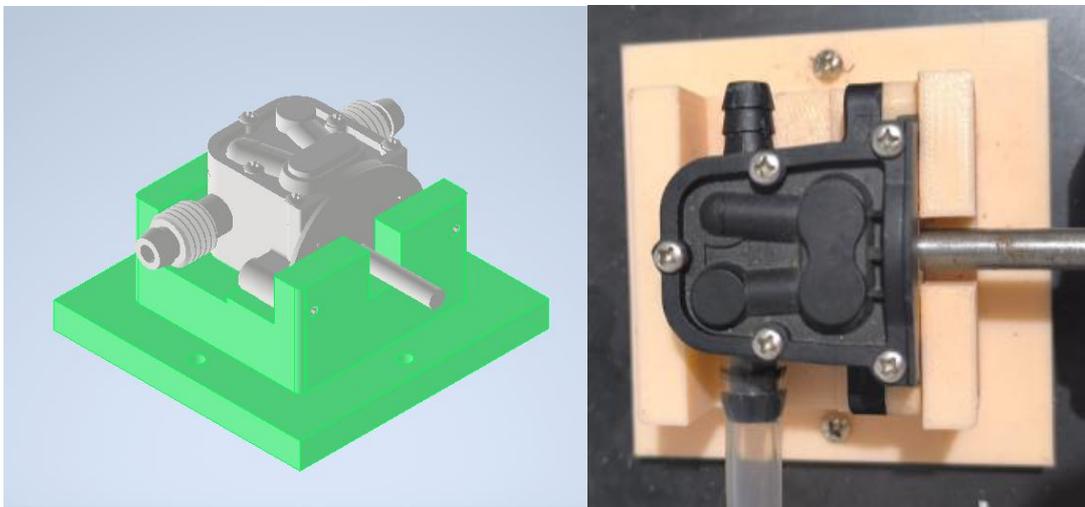


Figura 45. Bomba de diafragma.

Otro elemento que se puede visualizar en la **Figura 46** es la base encargada de regular los grados de apertura de la llave, la mismo que facilita la toma de datos teniendo en cuenta si está abierta, semiabierta o cerrada, para controlar el paso de agua en el sistema de bombeo.

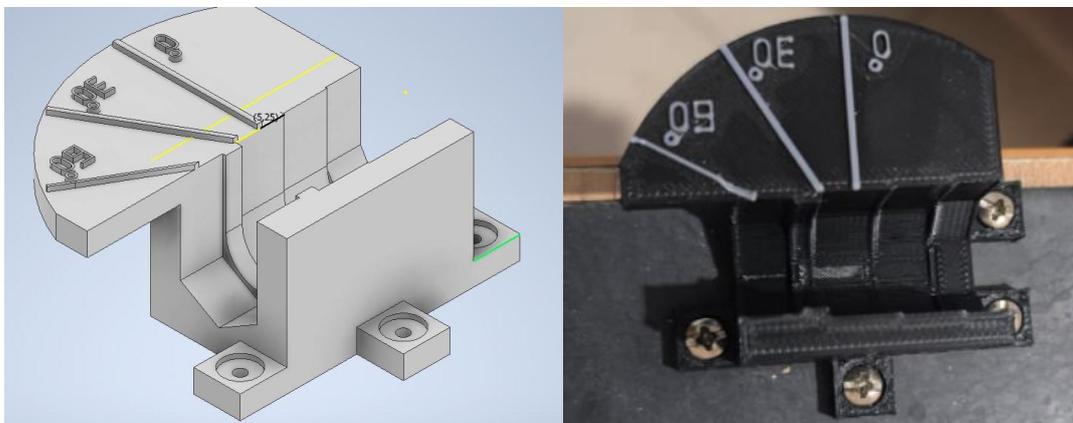


Figura 46. Base para medición en grados.

En la **Figura 47** se presenta el diseño de las bridas utilizadas para realizar mediciones con diferentes áreas de flujo. Estas bridas permiten configurar el sistema con una obstrucción del 50%, del 25% o una obstrucción total, regulando el flujo de aire en el sistema de ventilación.

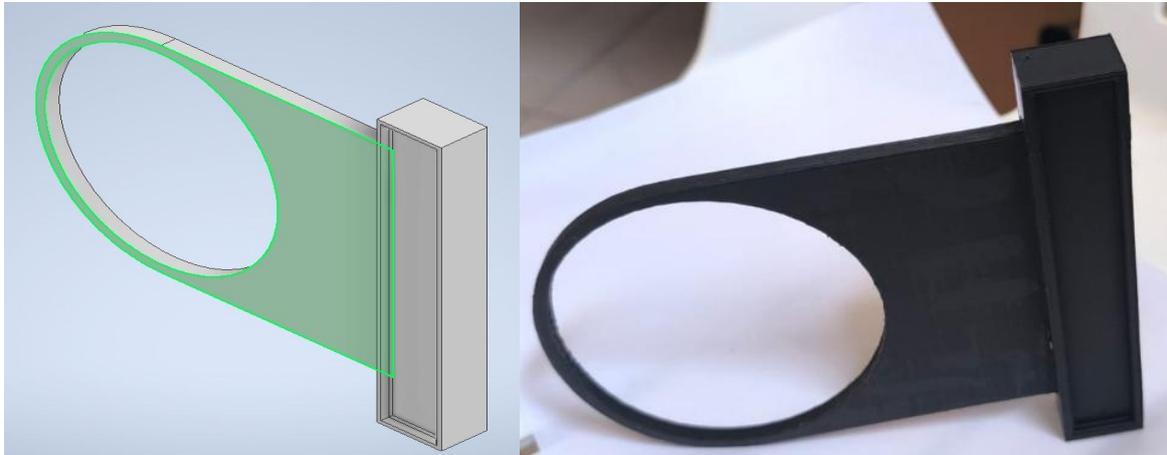


Figura 47. *Brida para obstrucción del aire.*

Otro elemento importante a diseñar es el ducto de ventilación, como se puede visualizar en la **Figura 48**. Este componente dirige el flujo de aire del ventilador hacia el sensor MAF.

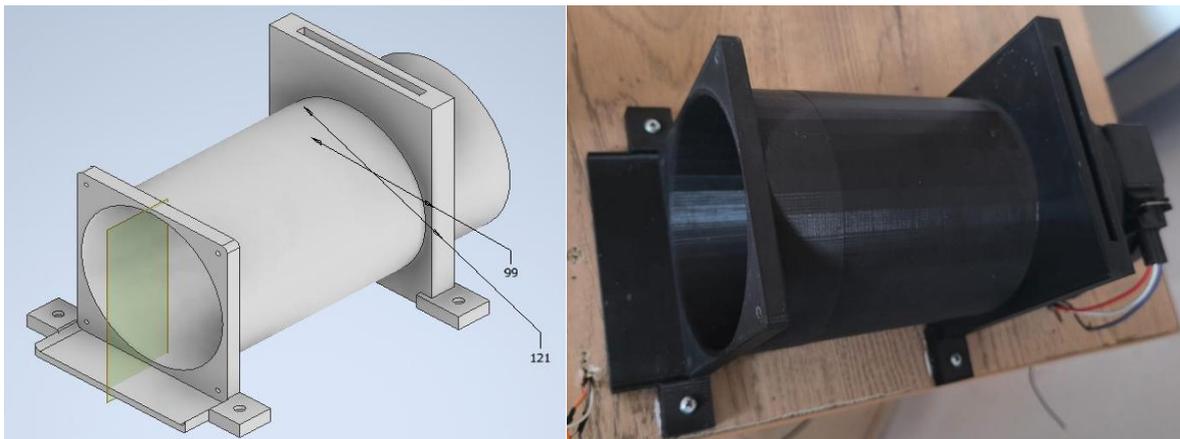


Figura 48. *Ducto de ventilación.*

Otro componente importante a diseñar es el tambor del sistema de elevación, mostrado en la **Figura 49**. En un extremo del eje se encuentra el sistema de transmisión por poleas, mientras que en el otro extremo se ubica una rueda utilizada para medir su velocidad angular. En la sección central del tambor se enrolla el cable encargado de elevar los objetos.

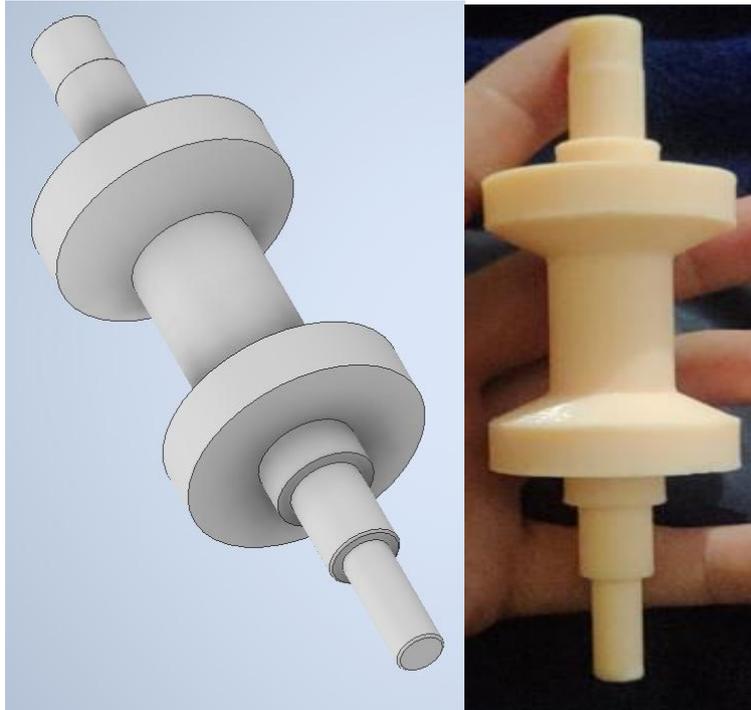


Figura 49. Enrolladora de cable.

El diseño de las bases para los sensores, mostradas en la **Figura 50**, es importante para dar una buena presentación y garantizar que los sensores estén a la altura adecuada para medir la velocidad angular. Además, permite fijar los sensores de manera segura en la mesa, junto con su respectivo sistema eléctrico.

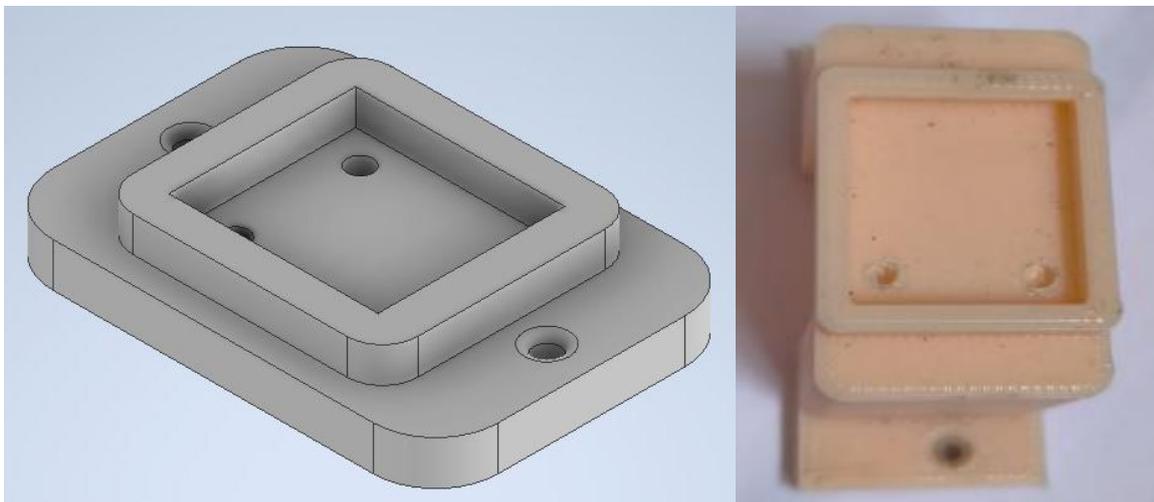


Figura 50. Base para sensores.

Para el sistema de elevación, también es importante el desarrollo de una polea loca, como se muestra en la **Figura 51**, la cual funciona como guía para el cable de la grúa.

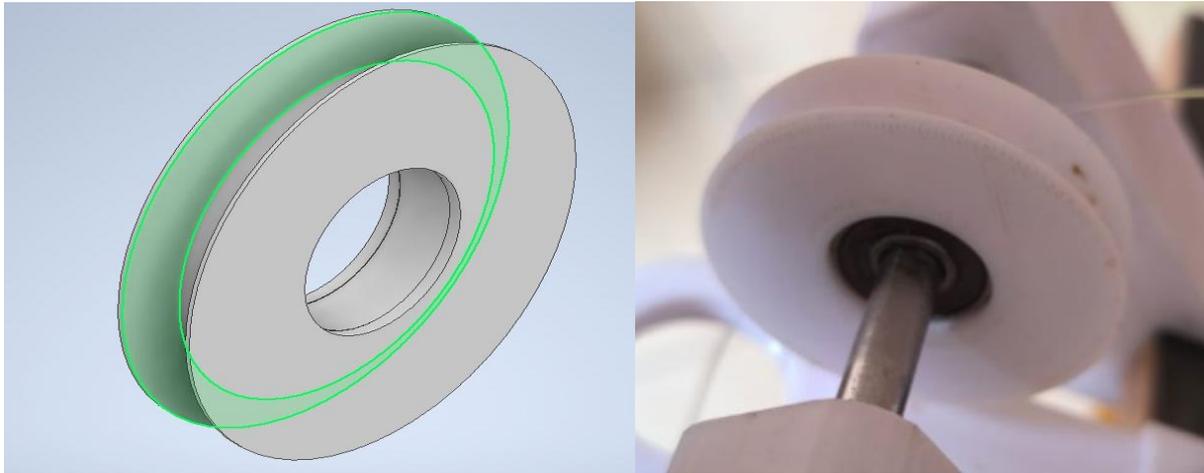


Figura 51. Polea loca.

6.4 Construcción del banco didáctico.

Finalizado el diseño y la adquisición de materiales, se inicia la construcción del banco didáctico. La impresión 3D permite fabricar diversas piezas, como se muestra en la **Figura 52**, incluyendo:

- Bases para los sensores y Arduino.
- Pilares para sostener el eje de la grúa.
- Base para bomba.
- Base para contenedor.
- Aspas para medir rpm.
- Polea loca.
- Ducto de ventilación.
- Bridas.
- Enrolladora de cable.
- Gancho.



Figura 52. *Impresión 3D.*

Además, se utiliza el torno para el maquinado de materiales como plomo, aluminio y acero de transmisión, con el fin de adecuar pesas, poleas y los ejes de transmisión para los sistemas, como se visualiza en la **Figura 53**.

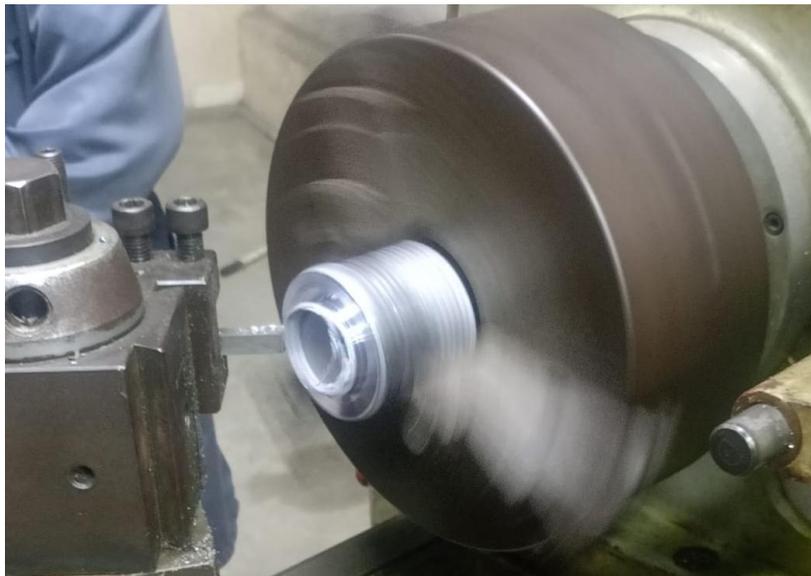


Figura 53. *Torneado de pesas.*

Para la adecuación de la plancha base del sistema bombeo y elevación de carga, ubicada en la parte posterior de la mesa, se utiliza una fresadora CNC, como se muestra en la **Figura 54**, debido a su ventaja de precisión de corte y taladrado. Esto permite la creación de carriles que facilitan el templado de las bandas entre los diferentes componentes, como motores y máquinas.

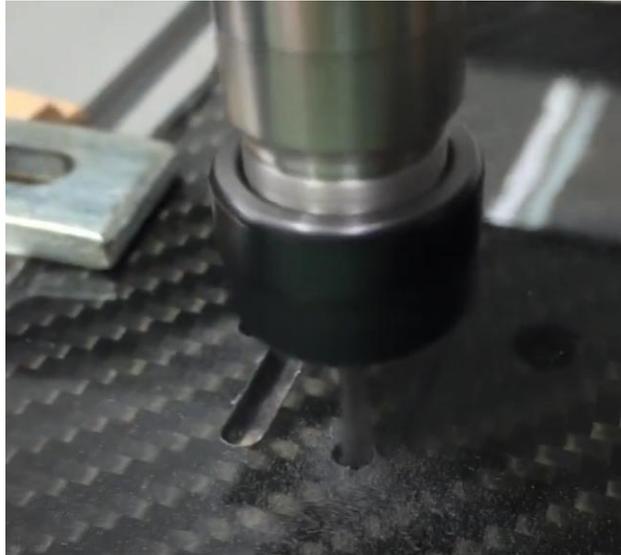


Figura 54. *Construcción de carriles templadores para bandas.*

6.4.1 *Fundición de pesas.*

Para la caracterización del motor en el sistema de elevación de carga, se considera necesario la creación de pesas. El material más adecuado es el plomo, ya que cumple con las características necesarias para la construcción de las diferentes pesas. El plomo cuenta con propiedades de fácil maleabilidad, lo que permite obtener pesas de diversos tamaños, el proceso de fundición se lo muestra en la **Figura 55**.



Figura 55. *Proceso de fundición del plomo.*

6.4.2 Ensamble de los diferentes componentes eléctricos.

En esta etapa, se incorpora una mesa diseñada específicamente para albergar todos los sistemas y complementos necesarios en el banco didáctico. Entre estos se incluyen elementos manufacturados mediante fresadora y torno, piezas impresas en 3D y componentes previamente adquiridos con el objetivo de garantizar el funcionamiento del conjunto.

Esta mesa se configura como un componente que cumple la función de organizar y contener los diversos elementos, además de facilitar la interconexión entre los sensores y los dispositivos de adquisición de datos. Para su implementación, es necesario desarrollar un esquema eléctrico en la plataforma Proteus, como se detalla en la **Figura 56** y se aplica en la **Figura 57**.

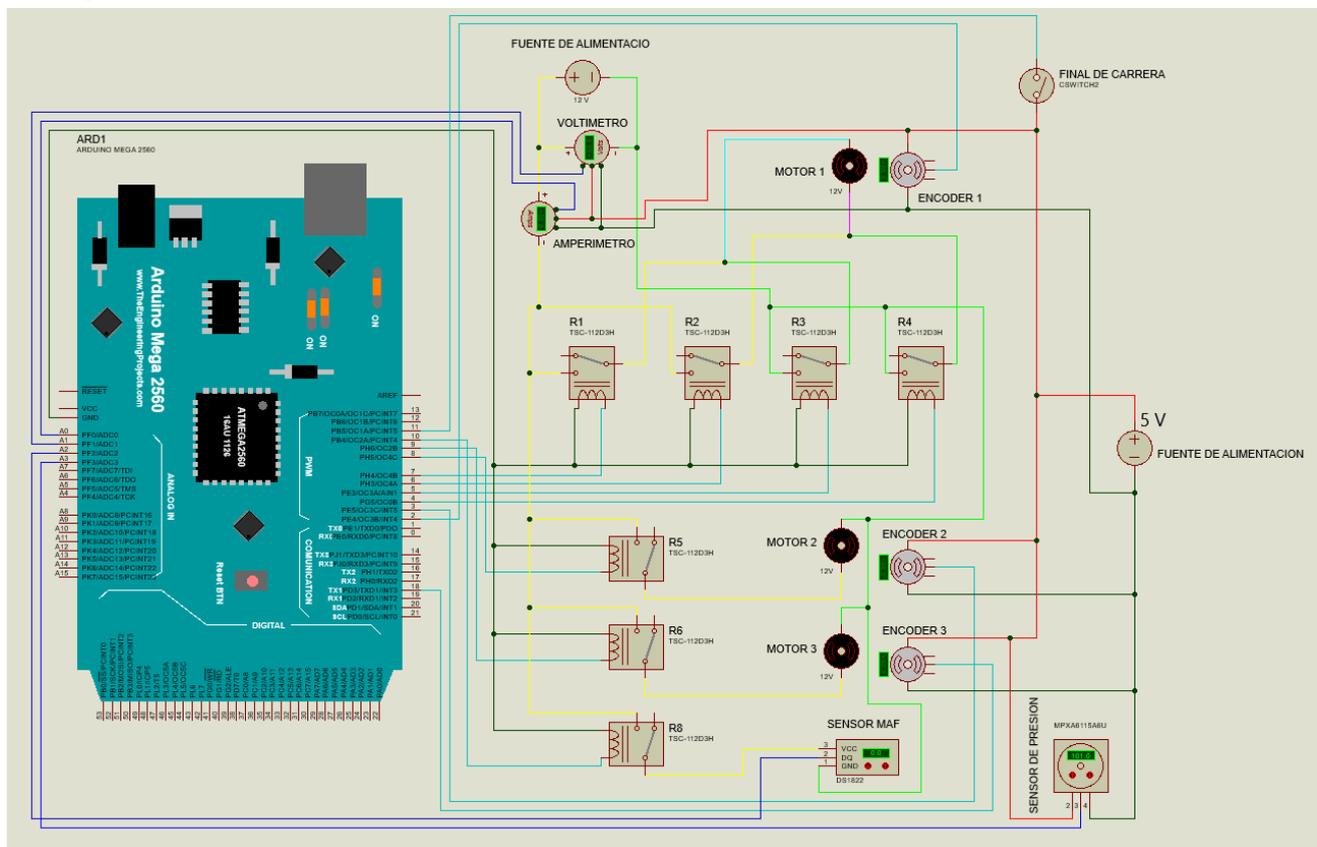


Figura 56. Esquema eléctrico en Proteus.



Figura 57. *Circuito eléctrico del banco.*

Para dimensionar la fuente de alimentación a la que se conecta los dispositivos eléctricos y que pueda manejar los picos de corriente al arrancar todos los motores al mismo tiempo, debemos tener en cuenta los picos de arranque de los motores y los consumos de los demás componentes. Vamos a hacer un cálculo conservador para asegurar que la fuente tenga suficiente capacidad para manejar todo sin problemas.

- **Cálculo de los picos de arranque**

Los motores pueden consumir entre 3 y 5 veces su corriente nominal durante el arranque. Usando los valores nominales de corriente de los motores y asumiendo un pico de 5 veces la corriente nominal:

Motores (12V):

Motor de 5W (0.42A nominal):

$$\text{Pico de arranque} = 0.42 \text{ A} \times 5 = \mathbf{2.1 \text{ A}}$$

Motor de 18W (1.5A nominal):

$$\text{Pico de arranque} = 1.5 \text{ A} \times 5 = \mathbf{7.5 \text{ A}}$$

Motor de 24W (2A nominal):

$$\text{Pico de arranque} = 2\text{A} \times 5 = \mathbf{10 \text{ A}}$$

Sensor MAF:

El sensor MAF consume 2A nominalmente. No hay pico de corriente significativo, ya que su consumo es constante.

Relés (5V):

Los 7 relés a 5V consumen aproximadamente 0.7 A nominalmente. Esto no suele tener picos de corriente elevados.

- **Cálculo del total de corriente de arranque**

Ahora, sumamos el pico de arranque de los motores y el consumo de los demás componentes:

Motores: $2.1 \text{ A} + 7.5 \text{ A} + 10 \text{ A} = \mathbf{19.6 \text{ A}}$

Sensor MAF: 2A (sin picos, consumo constante)

Relés : 0.7 A

Total, de corriente de arranque (suponiendo que todos los motores y componentes arrancan simultáneamente):

$19.6 \text{ A} + 2 \text{ A} + 0.7 \text{ A} = \mathbf{22.30 \text{ A}}$

3. Margen de seguridad

Es recomendable agregar un margen de seguridad del 30 - 40% para tener en cuenta variaciones en los picos de arranque o posibles incrementos en la carga. Esto nos dará un margen de sobrecarga para evitar problemas:

$22.30 \text{ A} \times 1.4 = \mathbf{31.22 \text{ A}}$

Para manejar correctamente los arranques simultáneos de los motores, sensores y relés, es necesaria una fuente de 12V de al menos 32A. Esta capacidad asegurará que no haya problemas al arrancar todos los dispositivos al mismo tiempo.

- **Potencia total**

Para calcular la potencia total que debe tener la fuente de alimentación, utilizamos la siguiente fórmula:

$$P = V \times I = V$$

Ya calculamos que la corriente máxima necesaria es 32A para asegurar un buen margen de seguridad.

$$P = 12V \times 32A = \mathbf{384W}$$

Las Figuras 58 y 59 presentan el ensamble físico final del banco didáctico diseñado para el análisis del par resistente en máquinas, junto con su representación en 3D, creada utilizando el software Inventor.



Figura 58. Circuito eléctrico del banco.



Figura 59. Ensamble de los componentes físicos del banco.

6.5 Costos de construcción e implementación del banco didáctico.

Tras completar la fase de construcción e instalación de los componentes del banco didáctico, se procede a realizar una tabla detallada de todos los costos de adquisición y construcción, como se indica en la **Tabla 7, 8 y 9**.

Tabla 7. Costo de Elementos.

Elementos para implementar en el banco		
Elemento	Cantidad	Precio (\$)
Motor reductor 37d 12v DC 60rpm zga37rg	1 unidad	22.99
Módulo sensor de corriente acs712 - 20a	1 unidad	3.99
Módulo sensor de corriente acs712 - 5a	1 unidad	3.99
Sensor infrarrojo fotoeléctrico de velocidad h encoder	3 unidad	7.47
Sensor de voltaje DC 0-25vDC fz0430	1 unidad	1.99
Sensor de peso 5kg +módulo 24 bit a/d convertidor hx711	1 unidad	5.99
Kit de 40 cables dupont 20cm de longitud hembra-macho	1 unidad	1.99
Pug banana 4mm	10 unidades	3.00
Jack banana 4mm js-910b	10 unidades	5.00
Módulo wifi esp8266-01 esp-01s	1 unidad	3.99
Adaptador cargador 110v/9v 1a plug DC005	1 unidad	3.49
Cable dupont 2,54mm arcoiris 40 pines 24awg arduino (metro)	1 unidad	3.99
Plug dupont 2,54mm 1 pin (incluye terminal)	30 unidades	1.50
Espadines en ángulo macho 40 pines 2,54mm	1 unidad	0.40
Espadines rectos macho 40 pines 2,54mm mecanizado redondos	1 unidad	0.75
Par de soportes metálicos para motor 37d mm	1 unidad	4.99
Sensor ultrasónico hc-sr04+	1 unidad	2.99
Eje diámetro 12mm largo 800mm	1 unidad	15.50
Kp001 diámetro 12mm	5 unidades	33.75
Bomba de agua de diafragma 12vDC dp-521	1 unidad	29.99
Polea gt2 20 dientes diámetro 6.35mm	1 unidad	2.99
Tubo termoencogible - tubo termoencogible 2.0mm negro	2 m	0.40
Tubo termoencogible - tubo termoencogible 2.5mm negro	4 m	0.80
Tubo termoencogible - tubo termoencogible 3.0mm negro	2 m	0.50
Tubo termoencogible - tubo termoencogible 5.0mm rojo	2 m	1.00
Chumacera Kp08 diámetro 8mm	4 unidades	11.96
Polea gt2 20 dientes diámetro 5mm	2 unidades	5.80
Polea gt2 20 dientes diámetro 8mm	2 unidades	5.80
Polea gt2 16 dientes diámetro 5mm	1 unidad	2.90
Polea gt2 36 dientes diámetro 5mm	1 unidad	6.50
Polea gt2 40 dientes diámetro 5mm	1 unidad	7.50
Correa dentada gt2 cerrada 280mm	1 unidad	3.50
Arduino mega	1 unidad	20.00
Módulo de 4 relés integrados	1 unidad	6.00
Relé	3 unidad	10.50
Estaño	1 rollo	10.00

Elementos para implementar en el banco

Elemento	Cantidad	Precio (\$)
Grilón	0.4 m	6.00
Filamento para impresora 3d	3 rollo	60.00
Mesa	1 unidad	80.00
Eje de transmisión	0.6 m	3.00
Plancha galvanizada de 2 mm	1 plancha	100.00
Polea fija	1 unidad	3.00
Plomo	20 lb	20.00
Manguera	4 m	1.00
Llave esférica 1/4" paso total	1 unidad	4.50
Acoples para cortadora	2 unidad	2.00
Recipiente para agua	1 unidad	5.00
Conector eléctrico de 2 vías	3 unidades	6.00
Fuente de alimentación	1 unidad	30.00
Brocas para taladro (4mm y 6mm)	2 unidades	5.00
Fresas (3mm, 4mm y 6mm)	3 unidades	19.00
Cuchilla de torno	1 unidad	7.00
Interruptor balancín negro de 2 pines, 2 posiciones	1 unidad	1.00
Regleta de 20 x12X2000	1 unidad	3.50
Regleta de 10 x 10 x 2000	1 unidad	2.50
Regleta de 15 x 10 x 2000	1 unidad	3.00
Cable #22	10 m	3.00
Cable #18	10 m	5.00
Cable para Arduino	10 m	2.00
Tornillo y tuerca milimétrico hexagonal frange 6mm	10 unidades	4.50
Tornillo y tuerca m4 metálico 10 mm	20 unidades	1.50
Tornillo y tuerca mariposa de 6 mm	2 unidades	0.50
Sensor de presión de 0.5 MPa para aire, aceite y agua	1 unidad	32.59
Motor para ventilador	1 unidad	10.00
Medidor de sensor de flujo de aire masivo MAF	1 unidad	21.95
Juego de 2 ruedas síncronas GT2 de 20 y 30 dientes	1 unidad	11.89
Correa de distribución 2GT-6	1 unidad	9.99
Adaptador de ajuste en T de bloqueo de presión	1 unidad	9.99
Acople de bronce para motor de eje 3 mm	1 unidad	3.98
Cable USB	1 unidad	5.00
Manómetro analógico	1 unidad	5.00
Bornera de derivación	1 unidad	4.50
Ventilador	1 unidad	10.00
Costos de importación		39.99
	Total	796.78

Tabla 8. Costos de manufactura.

Proceso de manufactura			
Actividad	Tiempo de uso (horas)	Precio por hora (\$)	Total (\$)
Torneado	5	40	200
Fresado	10	40	400
Impresión 3D	200	2	400
Fundición	2	30	60
Total:			1060

Tabla 9. Costos totales

Costos totales	
Actividad	Costo (\$)
Adquisición de elementos	796.78
Proceso de manufactura	1060
Costos de diseño y construcción	500
Total	2356.78

6.6 Validación de características de motores

6.6.1 Medición de los parámetros eléctricos y mecánicos de los motores.

La validación de las variables dinámicas mecánicas de los diversos motores se realiza mediante la utilización de información recopilada en el capítulo IV titulado “Modelados para motor DC”. Este capítulo proporciona los conocimientos para llevar a cabo la medición de los valores físicos que influyen en el funcionamiento de los motores de corriente continua, tales como:

- Resistencia de armadura (R): Se obtiene mediante mediciones realizadas con la ayuda de un multímetro para determinar la resistencia eléctrica en la armadura del motor, lo cual es necesario para el análisis de su rendimiento.
- Inductancia de armadura (L): Se mide para evaluar la inductancia en la armadura del motor, un factor clave para entender la respuesta del motor a cambios en la corriente.
- Tiempo de estabilización mecánica (tmss): Este parámetro hace referencia al tiempo que el sistema mecánico de cada motor tarda en alcanzar la estabilidad después de un cambio en las condiciones de operación.

- Corriente de arranque (i_{arr}): Es la cantidad de corriente eléctrica que un motor consume en el momento que comienza a funcionar.
- Constante electromotriz (V, I, W): Para su obtención, se consideran las variaciones en el voltaje (V), la corriente (I) y la potencia (W), lo que permite obtener una comprensión de la eficiencia del motor.

En la **Figura 60** y en la **Tabla 10** se presentan de manera gráfica las evidencias de los valores de estas variables dinámicas correspondientes a cada motor. Estas representaciones visuales ofrecen una comparación y permiten una interpretación de los resultados obtenidos.

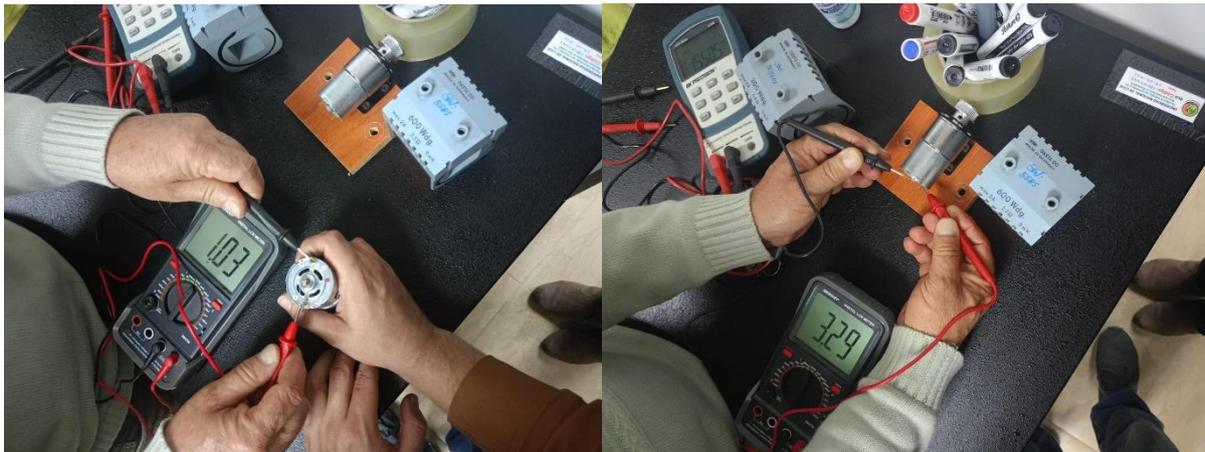


Figura 60. Medición de inductancia.

Tabla 10. Datos de motores

	Parámetros mecánicas y eléctricas		
	Motorreductor	Motor de Bomba	Motor de ventilador
Resistencia de armadura (R)	8.5 ohm	4 ohm	1.6 ohm
Inductancia de armadura (L)	3.29 mH	1.034 mH	1.18 mH
Tiempo de estabilización mecánica (tmss)	0.1 s	0.15 s	0.13 s
Corriente de arranque (i_{arr})	0.14 A	0.2 A	0.54 A

6.6.2 Análisis comparativo entre parámetros de fábrica y mediciones experimentales.

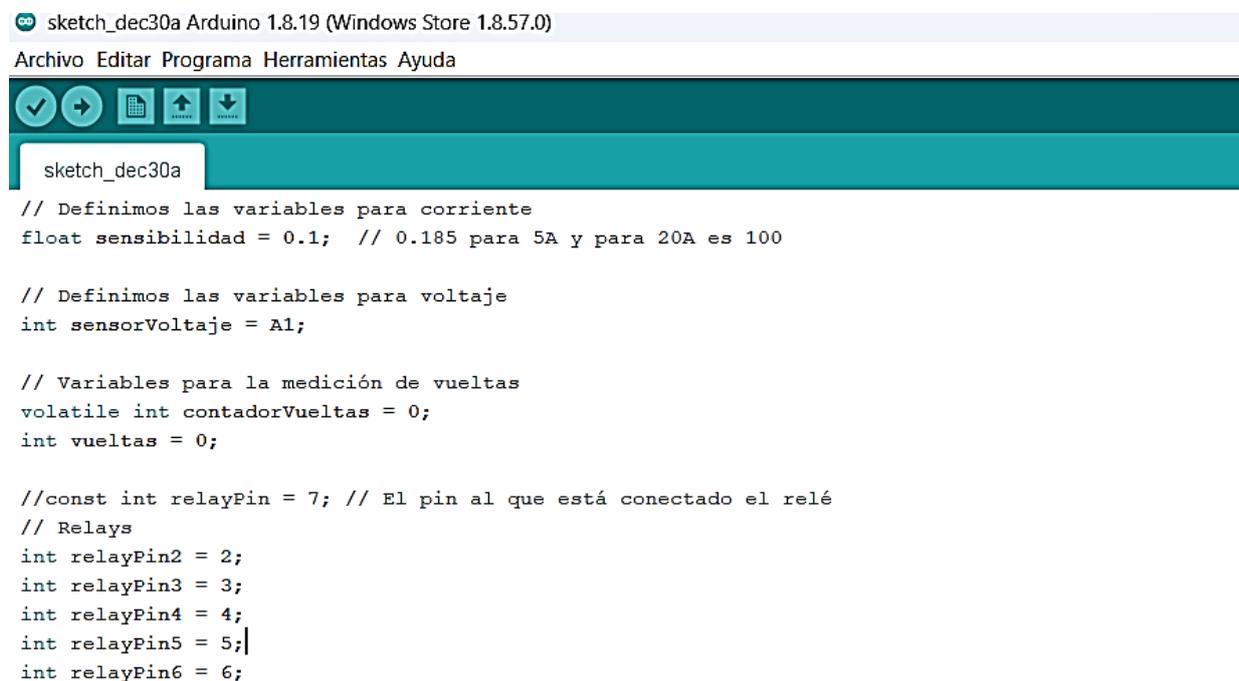
En esta validación, se comparan los datos proporcionados por el fabricante de los motores con los datos obtenidos a través de mediciones físicas realizadas durante las pruebas. Esta comparación permite evaluar las diferencias y confirmar la precisión de las especificaciones del fabricante, así como analizar el comportamiento real de los motores en las condiciones de operación del banco didáctico. Los resultados de esta comparación se presentan en la **Tabla 11**.

Tabla 11. Validación de parámetros.

Validación de los parámetros en los motores					
Valores a validar		Datos del Fabricante		Datos medidos	
		En vacío	Nominal	En vacío	Nominal
Motor de bomba	Corriente	0.4 A	1.5 A	0.44 A	1.55 A
	Revoluciones	7830 rpm	6210 rpm	7800 rpm	6000 rpm
Motor de ventilador	Corriente	0.29 A	0.63 A	0.3 A	0.61 A
	Revoluciones	3748 rpm	3620 rpm	3800	3500
Motorreductor	Corriente	0.12 A	0.72 A	0.14 A	0.52 A
	Revoluciones	68 rpm	57 rpm	69 rpm	57 rpm

6.7 Creación de interfaces gráficas.

El código para la interfaz de Arduino se encarga de adquirir, interpretar y transmitir los diversos datos generados por el propio Arduino, permitiendo el correcto funcionamiento del banco de pruebas, como se muestra en la **Figura 61**.



```

sketch_dec30a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda
sketch_dec30a
// Definimos las variables para corriente
float sensibilidad = 0.1; // 0.185 para 5A y para 20A es 100

// Definimos las variables para voltaje
int sensorVoltaje = A1;

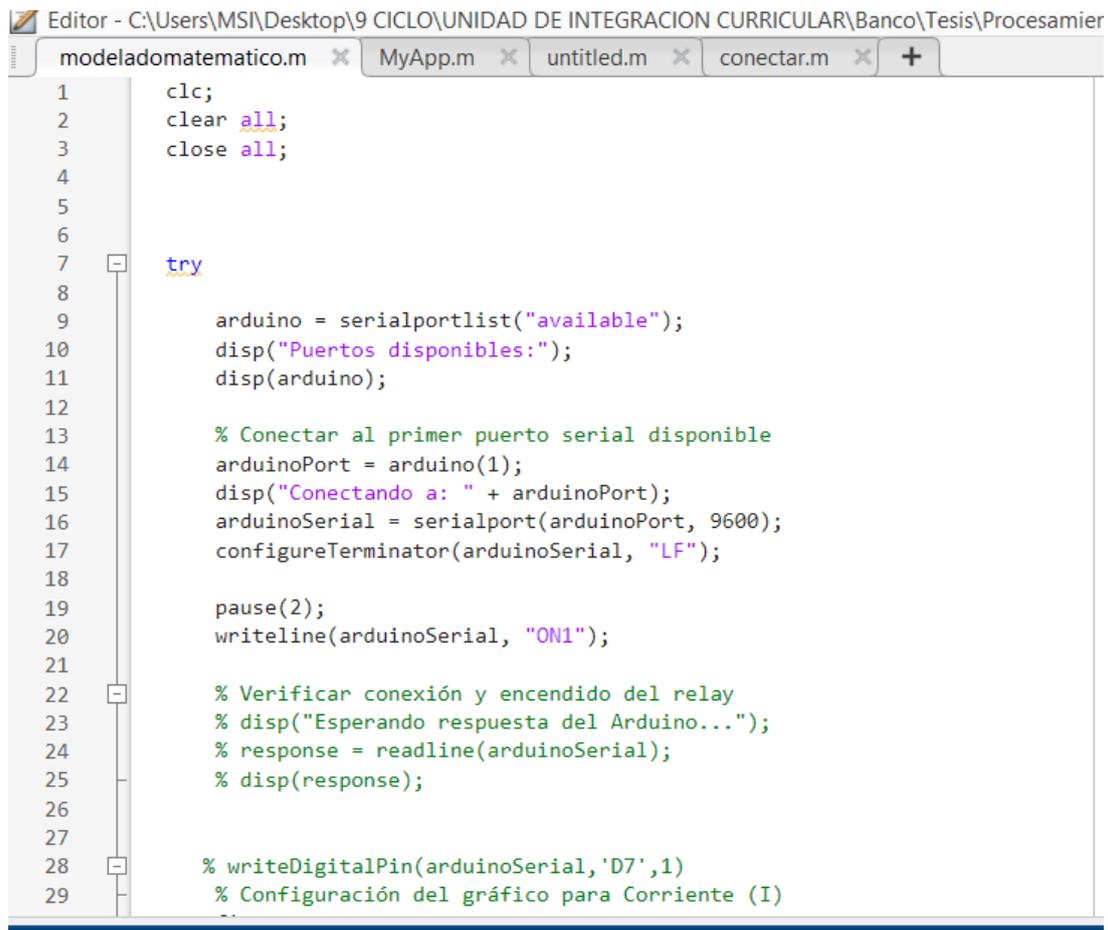
// Variables para la medición de vueltas
volatile int contadorVueltas = 0;
int vueltas = 0;

//const int relayPin = 7; // El pin al que está conectado el relé
// Relays
int relayPin2 = 2;
int relayPin3 = 3;
int relayPin4 = 4;
int relayPin5 = 5;
int relayPin6 = 6;
    
```

Figura 61. Código de Arduino.

En la siguiente fase del proyecto, se crea un código script en la interfaz de Matlab, que permite la comunicación entre la placa Arduino y este software, permitiendo el correcto funcionamiento de los diferentes sistemas del banco de pruebas. Posteriormente, se incorpora un mecanismo de cálculo para el torque mecánico de los motores, mediante el procesamiento

de los valores de voltaje, corriente y velocidad angular, lo que posibilita un monitoreo instantáneo durante pruebas y experimentos, como se muestra en la **Figura 62**.



```
Editor - C:\Users\MSI\Desktop\9 CICLO\UNIDAD DE INTEGRACION CURRICULAR\Banco\Tesis\Procesamier
modeladomatematico.m x MyApp.m x untitled.m x conectar.m x +
1      clc;
2      clear all;
3      close all;
4
5
6
7      try
8
9          arduino = serialportlist("available");
10         disp("Puertos disponibles:");
11         disp(arduino);
12
13         % Conectar al primer puerto serial disponible
14         arduinoPort = arduino(1);
15         disp("Conectando a: " + arduinoPort);
16         arduinoSerial = serialport(arduinoPort, 9600);
17         configureTerminator(arduinoSerial, "LF");
18
19         pause(2);
20         writeline(arduinoSerial, "ON1");
21
22         % Verificar conexión y encendido del relay
23         % disp("Esperando respuesta del Arduino...");
24         % response = readline(arduinoSerial);
25         % disp(response);
26
27
28         % writeDigitalPin(arduinoSerial,'D7',1)
29         % Configuración del gráfico para Corriente (I)
```

Figura 62. Código en Script de Matlab.

Tras la obtención de datos y gráficas, se procede a la implementación de un archivo en App Designer, como se muestra en la **Figura 63**, con el objetivo de proporcionar al usuario una interfaz intuitiva y fácil de usar. Esta decisión se toma considerando las ventajas de este software, como la retroalimentación en tiempo real y la visualización de datos.

Durante la programación de la aplicación, se tuvo en cuenta el funcionamiento del banco de pruebas para incluir todas las opciones y solicitudes que el usuario puede configurar según las distintas prácticas. Además, se garantiza que las diferentes configuraciones y variables no entren en conflictos, dada la presencia de múltiples opciones en la interfaz gráfica de usuario (GUI). Este enfoque garantiza una interacción fluida y sin inconvenientes para el usuario.

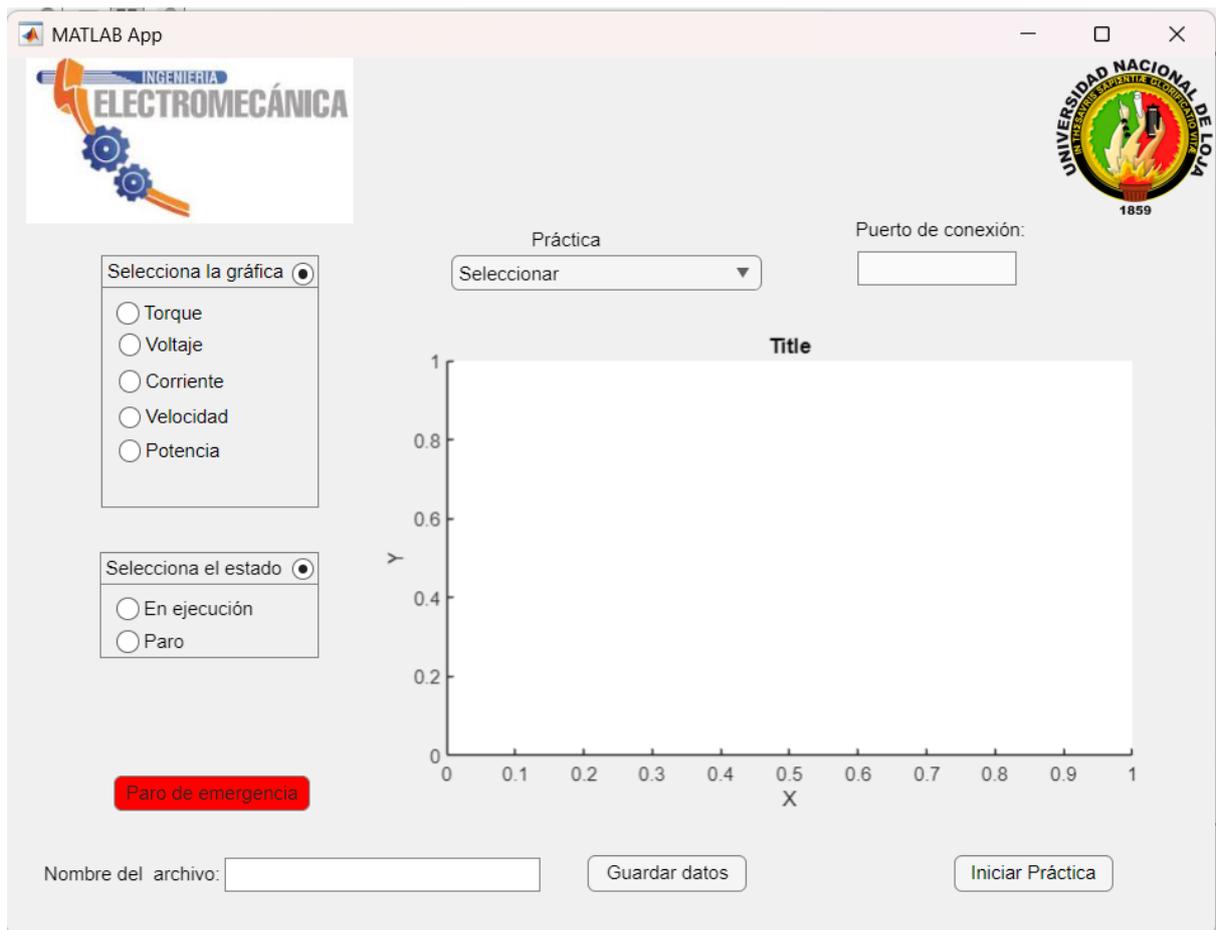


Figura 63. GUI de Matlab.

6.7.1 Creación del aplicativo paso a paso

PASO 1

Se seleccionan los elementos que conforman el GUIDE, como botones (Button), listas desplegables (Drop Down), ejes gráficos (Axes) y campos de edición (EditField), según lo mostrado en la **Figura 63**.

PASO 2

Se programan los componentes del GUIDE. El primer componente programado fue el "Drop Down", denominado práctica que permite seleccionar la práctica a realizar. Al elegir una opción, el código ejecuta internamente las instrucciones correspondientes, siguiendo la lógica de programación mostrada en la **Figura 64**.

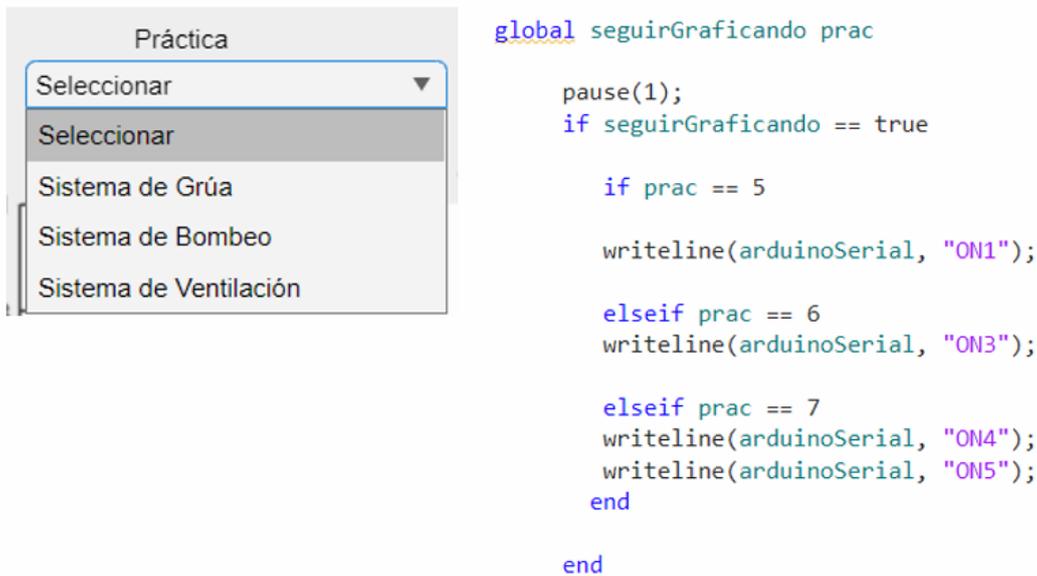


Figura 64. Programación del menú práctica.

De la misma manera, al seleccionar una opción en el menú, se pueden mostrar u ocultar botones complementarios según los requisitos de cada práctica, como se observa en la programación de la **Figura 65**.

```

function PrcticaDropDownValueChanged(app, event)
    value = app.PrcticaDropDown.Value;
    %disp(value)
    global R L tmss iarr prac

    prac = value;

    if value == 5
        app.BajarpesaButton.Visible = 'on';
        app.PresinButton_2.Visible = 'off';
        app.VelocidaddelaireButton.Visible = 'off';

    elseif value == 6
        app.BajarpesaButton.Visible = 'off';
        app.PresinButton_2.Visible = 'on';
        app.VelocidaddelaireButton.Visible = 'off';

    elseif value == 7
        app.BajarpesaButton.Visible = 'off';
        app.PresinButton_2.Visible = 'off';
        app.VelocidaddelaireButton.Visible = 'on';

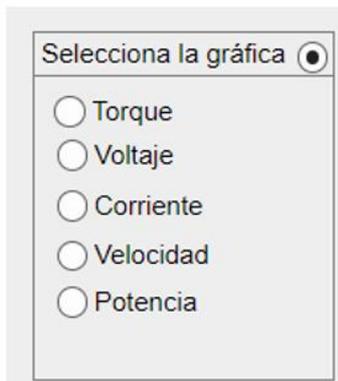
    end
end

```

Figura 65. Ajuste del GUIDE.

PASO 3

El siguiente componente a programar es el “Drop Down” denominado "Selecciona la gráfica", que permite elegir la gráfica que se desea visualizar en el “Axes” de la parte derecha del GUIDE en tiempo real. La programación de este componente se muestra en la **Figura 66**.



```

if dim == 6
    plot(app.UIAxes, historial.Torque, '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Torque (Nm)');
    title(app.UIAxes, 'Gráfica de Torque');
    drawnow;
elseif dim == 7
    plot(app.UIAxes, historial.Voltaje, '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Voltaje (V)');
    title(app.UIAxes, 'Gráfica de Voltaje');
    drawnow;
elseif dim == 9
    plot(app.UIAxes, historial.Corriente, '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Corriente (A)');
    title(app.UIAxes, 'Gráfica de Corriente');
    drawnow;
elseif dim == 10
    plot(app.UIAxes, historial.Velocidad, '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Velocidad (rpm)');
    title(app.UIAxes, 'Gráfica de Velocidad');
    drawnow;

```

Figura 66. Programación del grupo de botones de selección de gráfica.

Adicionalmente, cumple la función de guardar todos los valores obtenidos para los diferentes parámetros de los sistemas y, automáticamente, sigue mostrándolos en el “Axes” incluso después de culminar la conexión entre el software Matlab y Arduino. Esto se logra mediante el código de la **Figura 67**, que permite guardar los datos, y el código de la **Figura 68**, que permite proyectar las diferentes curvas tacodinámicas e intercambiarlas entre sí.

```

%variables de los datos
global tor vol corr vel tim pot pres fluj tiempoDeSubida

tor = historial.Torque;
pot = historial.Potencia;
pres = historial.Presion;
fluj = historial.Flujo;
vol = historial.Voltaje;
corr = historial.Corriente;
vel = historial.Velocidad;
tim = tiempo;

```

Figura 67. Programación que almacena los datos.

```

global ce
ce=true;
while ishandle(app.UIAxes)&& ce
global dim tor vol corr vel tim fin pot pres fluj

if dim == 6
plot(app.UIAxes, tim, tor , '-o', 'LineWidth', 2);
xlabel(app.UIAxes, 'Tiempo (s)');
ylabel(app.UIAxes, 'Torque (N.cm)');
title(app.UIAxes, 'Gráfica de Torque');
drawnow;
elseif dim == 7
plot(app.UIAxes, tim, vol , '-o', 'LineWidth', 2);
xlabel(app.UIAxes, 'Tiempo (s)');
ylabel(app.UIAxes, 'Voltaje (V)');
title(app.UIAxes, 'Gráfica de Voltaje');
drawnow;
elseif dim == 9
plot(app.UIAxes, tim, corr , '-o', 'LineWidth', 2);
xlabel(app.UIAxes, 'Tiempo (s)');
ylabel(app.UIAxes, 'Corriente (A)');
title(app.UIAxes, 'Gráfica de Corriente');
drawnow;

```

Figura 68. Programación de la proyección de las gráficas.

PASO 4

El “Drop Down” titulado "Seleccione el estado" es determinante para realizar las prácticas, ya que este menú permite que Matlab se ejecute cuando el menú está en la opción “En ejecución”. En caso contrario, si el menú se encuentra en la opción "Paro", el motor de la práctica se detendrá y no se podrá encender hasta que se deseccione esta opción. La programación de este menú se indica en la **Figura 69**.

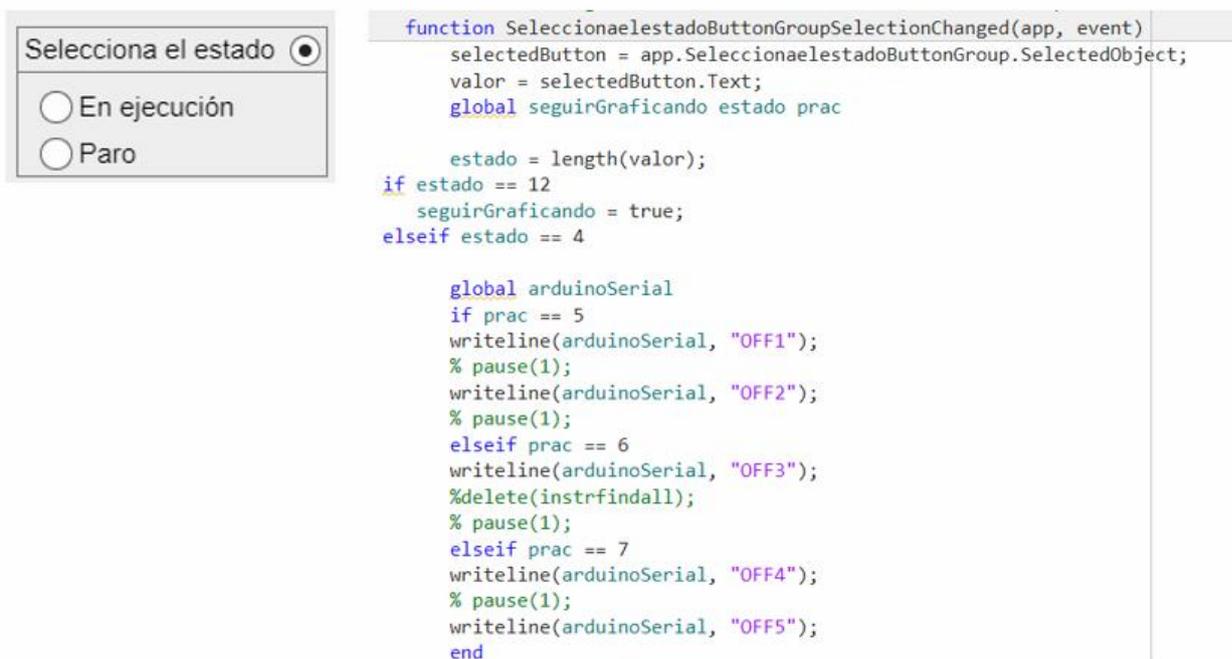


Figura 69. Programa del grupo de botones para el estado del banco.

PASO 5

El “Button” titulado "Bajar Pesa" aparece en la pantalla GUIDE al seleccionar la práctica "Sistema de elevación de cargas". Este botón permite invertir el giro del motor reductor acoplado al sistema de elevación, permitiendo que los pesos que han sido elevados regresen a su posición inicial para realizar la siguiente práctica con un peso diferente. La programación de este botón se muestra a continuación en la **Figura 70**.

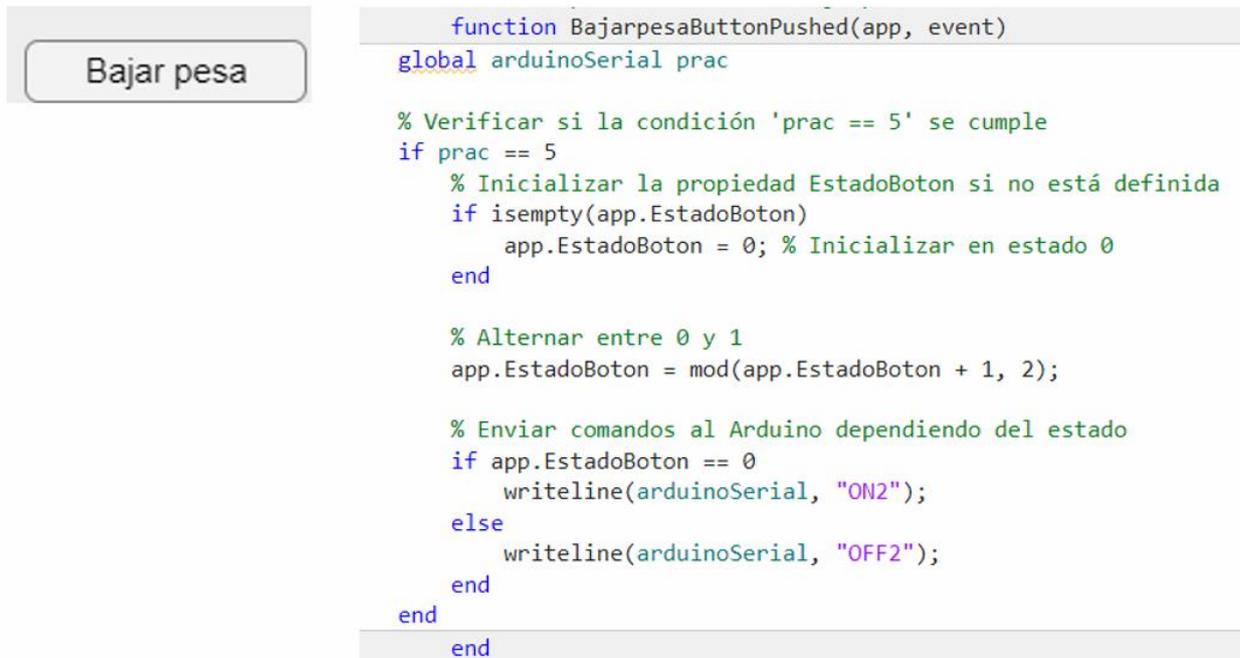


Figura 70. Programación del botón bajar pesa.

PASO 6

El “Button” titulado "Iniciar Práctica" cumple la función de conectar automáticamente al Arduino implementado en el banco, habilitando el canal serial que permite recibir y enviar los datos desde el Arduino hacia Matlab y viceversa. Además, en la programación de este botón se encuentra un bucle “while”, el cual se encarga de recibir los datos y procesarlos para su posterior graficado, siguiendo los diferentes parámetros previamente seleccionados por los menús de opciones y botones. Su programación se muestra en la **Figura 71** y la **Figura 72**.

Iniciar Práctica

```
function IniciarPrcticaButtonPushed(app, event)
global arduinoSerial

%fclose(arduinoSerial);
delete(arduinoSerial);

% Conectar con Arduino
arduino = serialportlist("available");
disp("Puertos disponibles:");
disp(arduino);

% Conectar al primer puerto serial disponible
arduinoPort = arduino(1);
disp("Conectando a: " + arduinoPort);
```

Figura 71. Conexión entre Matlab y Arduino.

```
% Bucle principal
while ishandle(app.UIAxes) && seguirGraficando % Mientras el axes esté activo
% Leer los tres datos desde Arduino
data = str2double(split(readline(arduinoSerial), ','));

global R L tmss iarr dim fin prac ca estado

% Actualizar los gráficos
if ~isempty(data) && length(data) == 8
% Obtener los valores individuales
Ca = data(1); %final de carrera
I = data(2); % Corriente
V = data(3); % Voltaje
W1 = data(4); % Vueltas1
W2 = data(5); % Vueltas2
W3 = data(6); % Vueltas3
F= data(7); % Flujo de aire del ventilador
Pr = data(8); % Presión de la bomba
```

Figura 72. Procesamiento de datos obtenidos del Arduino.

PASO 7

El "Button" titulado "Paro de Emergencia" permite apagar todos los relés involucrados en el funcionamiento del banco, ayudando a detener inmediatamente todo el sistema ante un acontecimiento inesperado. Este código se muestra en la **Figura 73**.

Paro de emergencia

```
function ParodeemergenciaButtonPushed(app, event)
    global arduinoSerial seguirGraficando ce

    writeline(arduinoSerial, "OFF1");
    % pause(1);
    writeline(arduinoSerial, "OFF2");
    % pause(1);

    writeline(arduinoSerial, "OFF3");
    %delete(instrfindall);
    % pause(1);

    writeline(arduinoSerial, "OFF4");
    % pause(1);
    writeline(arduinoSerial, "OFF5");

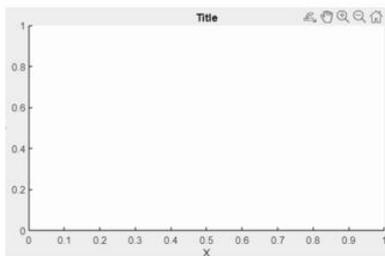
    seguirGraficando = false;

    ce=true
end
end
```

Figura 73. Programación del botón paro de emergencia.

PASO 8

El “Axes” permite la graficado y visualización de los diferentes datos obtenidos y procesados por el código de Matlab mencionado anteriormente. En la **Figura 74** se visualiza el código utilizado para este procedimiento.



```
if dim == 6
    plot(app.UIAxes, tim, tor , '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Torque (N.cm)');
    title(app.UIAxes, 'Gráfica de Torque');
    drawnow;
elseif dim == 7
    plot(app.UIAxes, tim, vol , '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Voltaje (V)');
    title(app.UIAxes, 'Gráfica de Voltaje');
    drawnow;
elseif dim == 9
    plot(app.UIAxes, tim, corr , '-o', 'LineWidth', 2);
    xlabel(app.UIAxes, 'Tiempo (s)');
    ylabel(app.UIAxes, 'Corriente (A)');
    title(app.UIAxes, 'Gráfica de Corriente');
```

Figura 74. Programación que permite graficar.

PASO 9

Finalmente, se utilizó un “Edit Field” para que el usuario pueda introducir el nombre correspondiente a cada práctica realizada. Además, este campo está relacionado con el “Button” titulado "Guardar Datos", lo que permite tomar el nombre introducido en el cuadro de texto y crear un archivo Excel con todos los datos que el programa ha obtenido hasta ese momento. La programación para este botón se muestra en la **Figura 75**.



```
function GuardardatosButtonPushed(app, event)
    global tor vol corr vel tim pot nombreArchivo prac pres fluj

    if prac == 5 %%grua
        datos =[corr;vol;vel;tor;pot;tim];
        writematrix(datos', [nombreArchivo, '.xls']);
        % writematrix(datos', 'datos.xls');
        % %
    elseif prac == 6 %%Bomba
        datos =[corr;vol;vel;tor;pot;pres;tim];
        writematrix(datos', [nombreArchivo, '.xls']);
        writematrix(datos', 'datos.xls');
        % %
        % %
    elseif prac == 7 %%ventilador
        datos =[corr;vol;vel;tor;pot;fluj;tim];
        writematrix(datos', [nombreArchivo, '.xls']);
        % writematrix(datos', 'datos.xls');
        % %
    end
end
```

Figura 75. Programación que permite exportar los datos a un archivo Excel.

6.7.2 Flujograma de funcionamiento del aplicativo.

En la **Figura 76** se muestra el flujograma del funcionamiento del aplicativo para la realización de las prácticas.

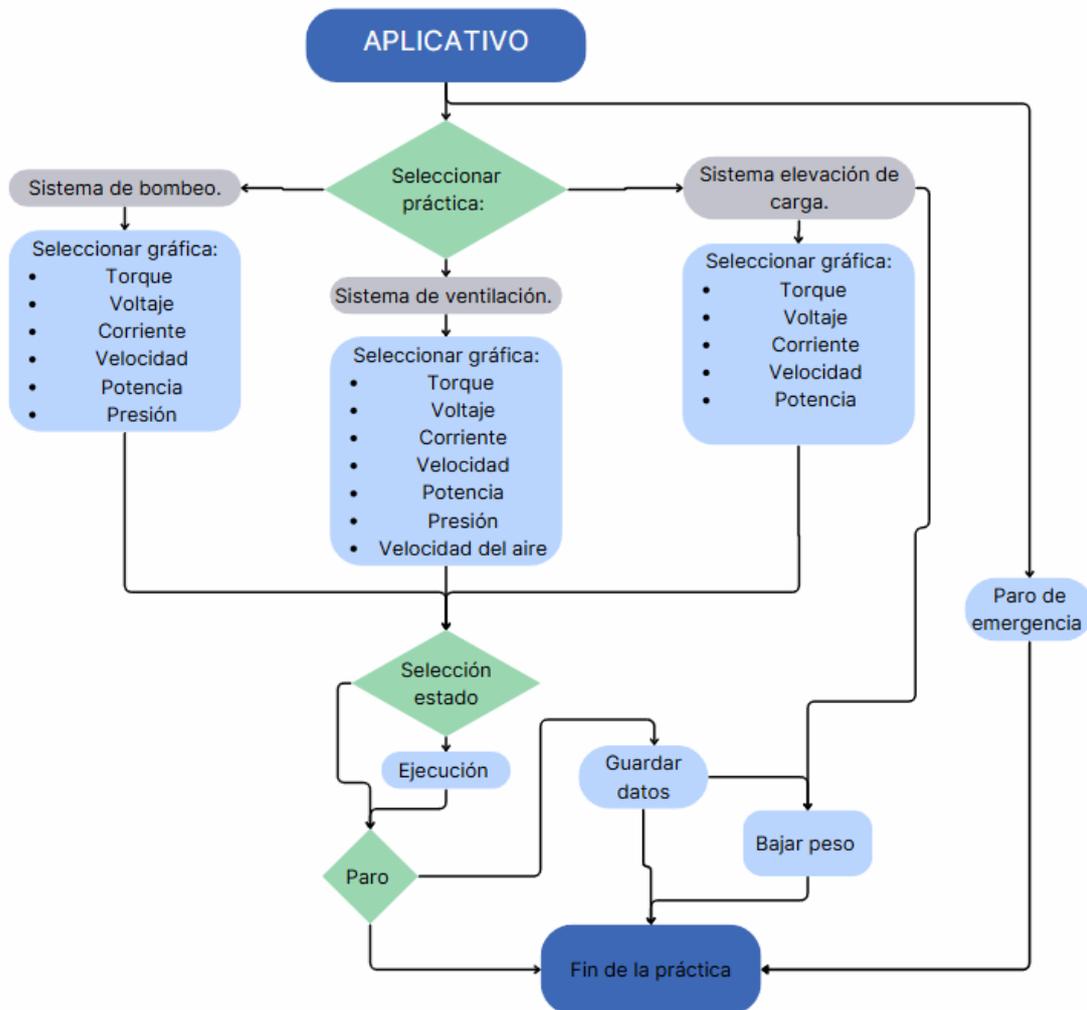


Figura 76. Flujograma del aplicativo.

6.7.2.1 Validación de interfaces gráficas

En el proceso de validación de las interfaces gráficas, se emplean diversos modelados matemáticos con el fin de obtener gráficos comparativos. Lo cual implica la utilización de simulaciones basadas en modelos fenomenológicos y funciones de transferencia específicas, para la caracterización de los motores de escobillas.

La implementación de estos modelos permite obtener gráficas simuladas que se comparan con los datos obtenidos de manera experimental. De esta manera, se establece la confiabilidad de las interfaces gráficas, respaldada por la comparación entre las representaciones teóricas y las observaciones prácticas. El mismo que contribuye a garantizar la coherencia entre la simulación y la realidad, validando las interfaces gráficas desarrolladas.

6.7.2.1.1 Modelo fenomenológico

Para el modelo fenomenológico de los motores de escobillas, es necesario definir constantes dinámicas, mecánicas y eléctricas. En estas constantes van introducidos los valores medidos experimentalmente en el motor y sirven como base para formular las ecuaciones diferenciales que describen su comportamiento.

A partir de estas constantes, se construye un diagrama de bloques en el software Simulink, utilizando las **Ecuaciones 8, 10, 11 y 12** descritas en el capítulo IV del marco teórico.

Para ello, se ha implementado diversas salidas en el sistema detallado en la **Figura 77**. Cada una de ellas ha sido diseñada con una entrada única de voltaje, con el fin de realizar predicciones en el comportamiento del motor en diversas situaciones.

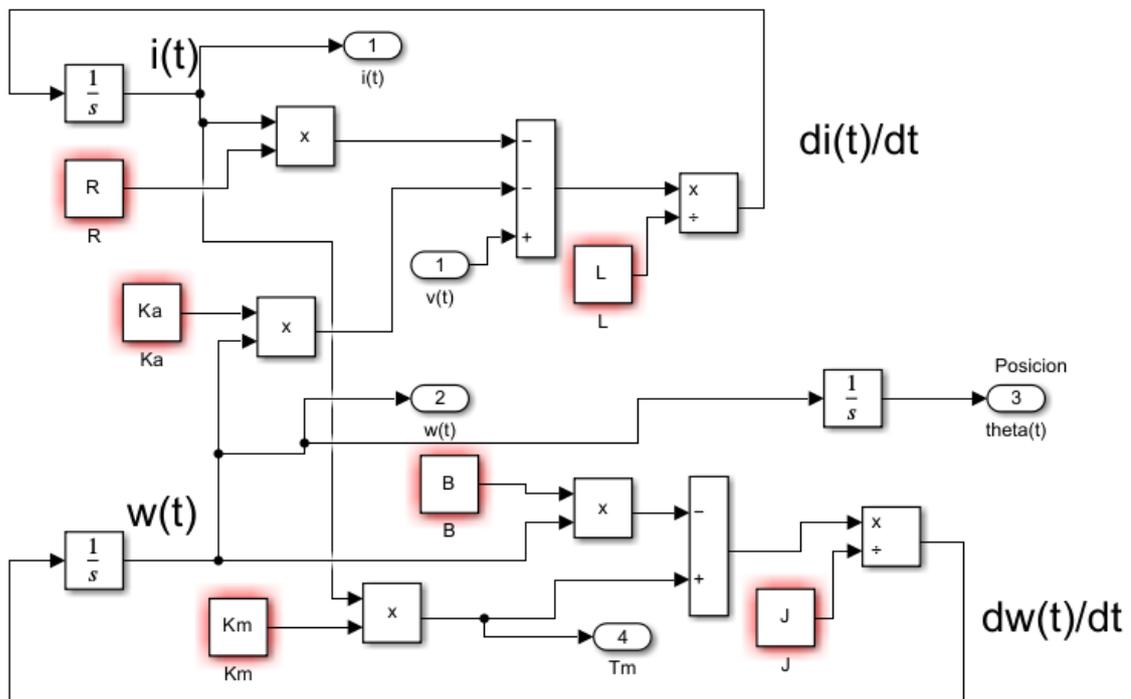


Figura 77. Diagrama de bloques.

La primera salida se enfoca en la predicción de la corriente, la segunda aborda la velocidad angular, la tercera se centra en la predicción de la posición, y la cuarta tiene como objetivo anticipar el torque medio del motor de corriente continua las mismas que se pueden observar en la **Figura 78, 79 y 80**. Estas predicciones son importantes para entender cómo se deben comportar estos parámetros ante el correcto funcionamiento de los sistemas.

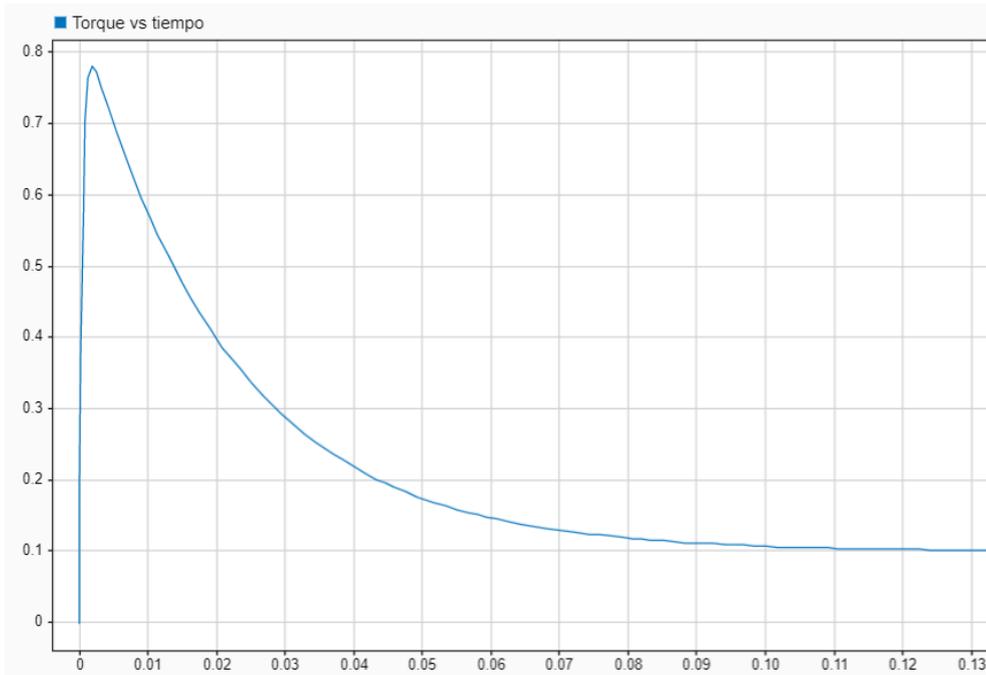


Figura 78. Grafica de torque mecánico(Nm) en el eje y vs tiempo(s) en el eje x.

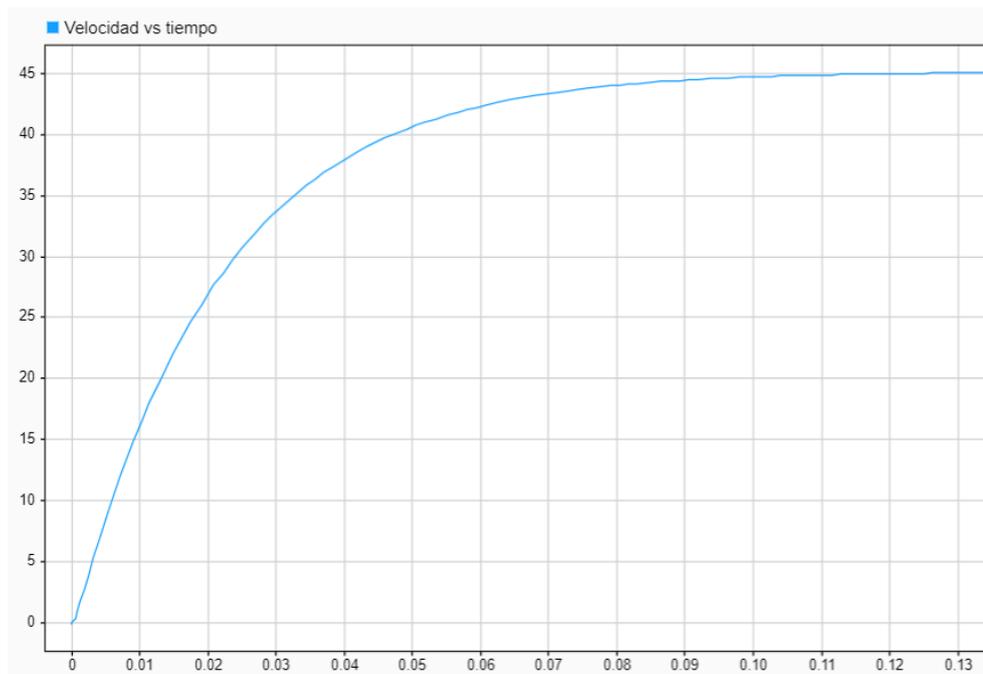


Figura 79. Grafica de velocidad angular(rpm) en el eje y vs tiempo(s) en el eje x .

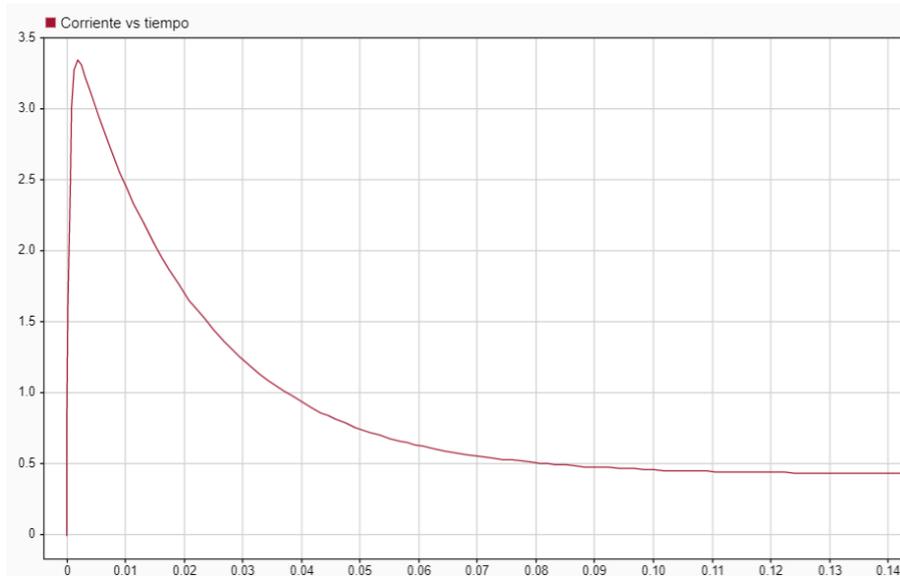


Figura 80. Grafica de corriente de armadura(A) en el eje y vs tiempo(s) en el eje x.

6.7.2.1.2 Funciones de transferencia

Mediante las funciones de transferencia encontradas en el modelado matemático para un motor DC (ecuaciones 22, 23, 24, 25 y 26), se realiza su resolución en Simulink mediante el bloque Transfer Fnc. Este bloque es una herramienta matemática lineal que utiliza la transformada de Laplace para describir el comportamiento dinámico y estacionario de un sistema. En detalle, proporciona una relación algebraica entre la entrada y la salida de un sistema, facilitando el análisis y diseño de sistemas dinámicos, este proceso se lo implementa como se muestra en la **Figura 81**.

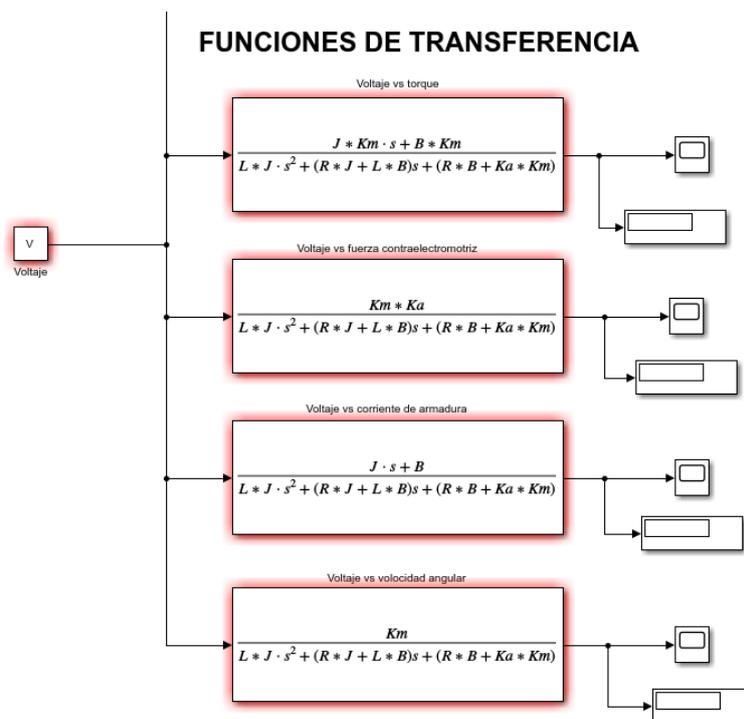


Figura 81. Funciones de transferencia

Las funciones de transferencia permiten obtener como resultados el torque, la corriente de armadura y la velocidad angular, considerando un voltaje común como entrada para predecir el comportamiento de los motores. Las gráficas correspondientes se presentan en las **Figuras 82, 83 y 84.**

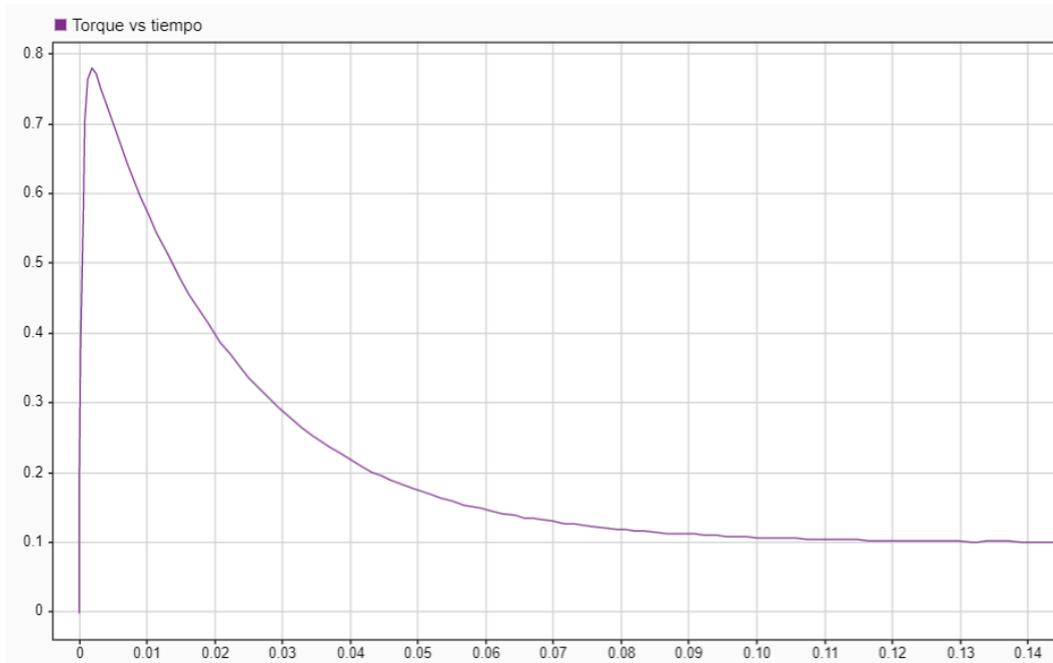


Figura 82. Grafica de torque mecánico(Nm) en el eje y vs tiempo(s) en el eje x.

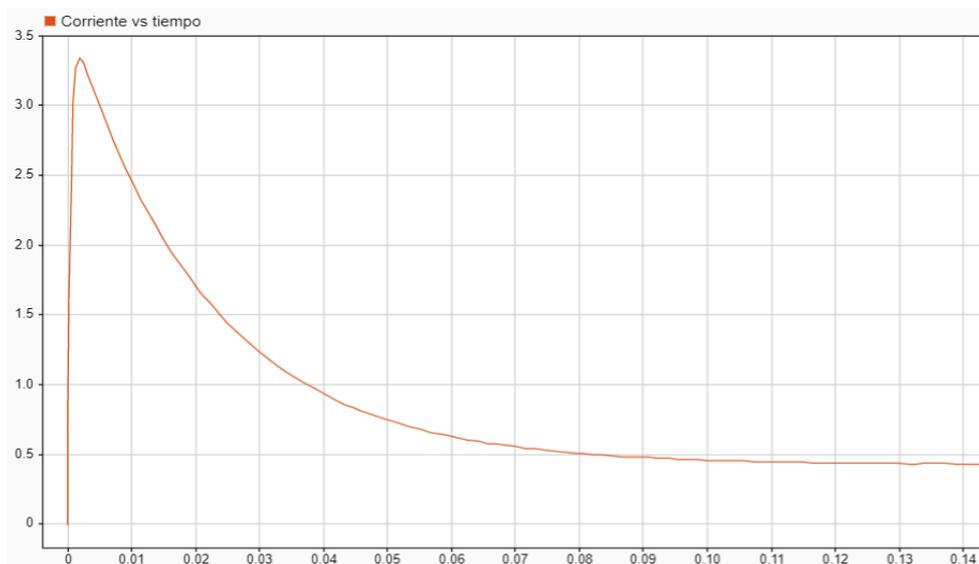


Figura 83. Grafica de corriente de armadura(A) en eje y vs tiempo(s) en el eje x.

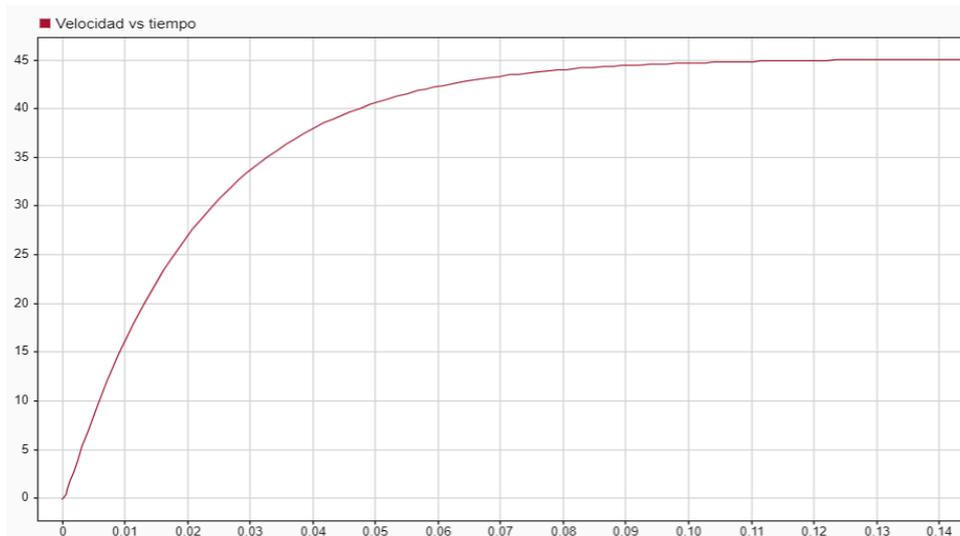


Figura 84. Grafica de velocidad angular(rpm) en el eje y vs tiempo(s) en el eje x.

6.8 Calibración de sensores

Para la adquisición de los datos de las diferentes variables involucradas en el funcionamiento del banco didáctico de curvas taco-dinámicas, es necesario calibrar adecuadamente los diversos sensores que se van a encargar de medir el voltaje, corriente y velocidad de los motores.

6.8.1 Voltaje

Para adquirir el valor de voltaje utilizado en la alimentación de los motores ante diferentes condiciones de carga se utiliza un sensor de voltaje FZ0430 DC, este sensor cuenta con un rango de medición de 0 a 25 V. Este sensor es analógico, lo que significa que su salida va a comunicar 1024 datos al Arduino, dependiendo de las condiciones de medida.

En el modelado matemático para la calibración de sensor es necesario seguir los siguientes pasos:

- Multiplicar el rango de medida al que llega este sensor (25 V)
- Multiplicar el pin analógico del Arduino que va a recibir los valores del sensor
- Dividir el número total de valores que mide la placa Arduino o su resolución que es de 10 bits (1024 datos)
- Crear el código Arduino.

A continuación, en la **Figura 85** se exhibe el código creado en la interfaz de Arduino.

```

// Conectar en puerto A1

// Definimos las variables para VOLTAJE
int Sensorvol = A1;

void setup() {

  Serial.begin(9600);
}

void loop () {

  float voltaje = (float)25*analogRead(A1)/1023;
  Serial.print("Voltaje medido = ");
  Serial.println(voltaje);
  delay(500);
}

```

Figura 85. Calibración de voltímetro.

La conexión física de este sensor al Arduino se realiza como se muestra en la **Figura 86**.

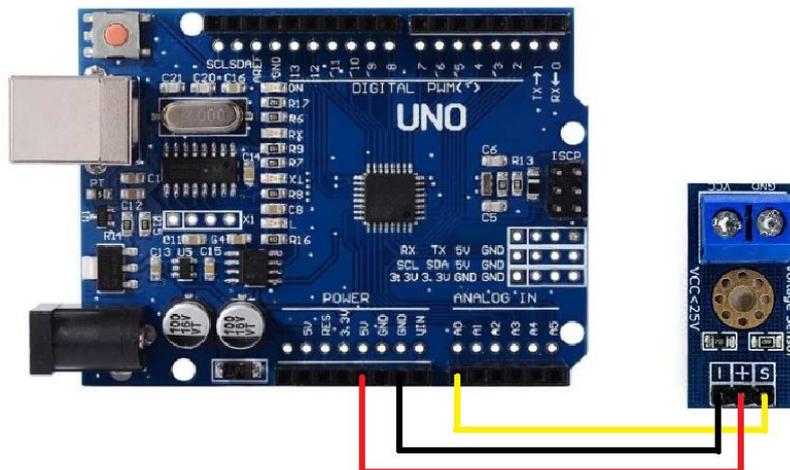


Figura 86. Conexión del sensor de voltaje al Arduino.

6.8.2 Amperaje

Para determinar el valor de amperaje utilizado en la alimentación de los motores ante diferentes circunstancias de carga, se utiliza un sensor de corriente ACS712. Este sensor dispone un rango de medición de 0 a 20 A, además se debe tomar en cuenta que el sensor es analógico, lo que significa su salida comunica de 0 a 1024 datos al Arduino.

El primer paso para calibrar correctamente el sensor es identificando su sensibilidad. Este valor se obtiene utilizando de la **Ecuación 27**, la cual cuenta con un valor constante de 2.5, que corresponde al valor estándar del sensor cuando la corriente es 0.

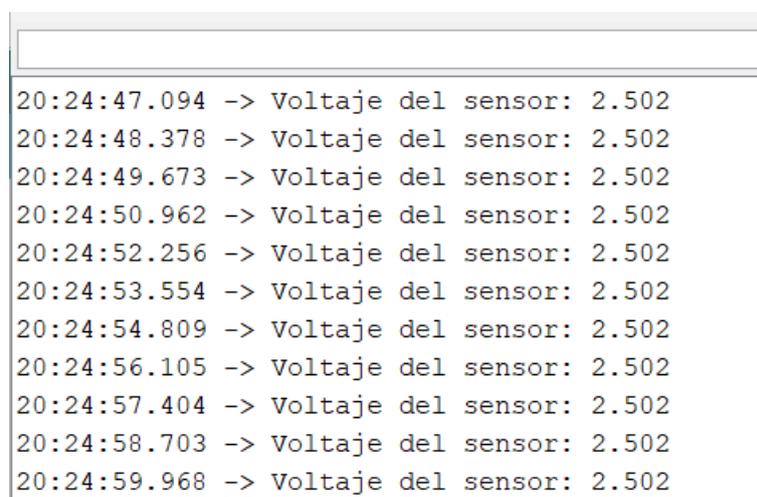
$$I = \frac{V - 2.5}{Sensibilidad} \quad (27)$$

La sensibilidad es equivalente a la recta de voltaje vs corriente. Basándose en esta afirmación, se puede obtener la recta por medio de dos puntos medidos adecuadamente. Para ello, se prosigue a obtener estos dos puntos reales.

El primer punto de la recta se obtiene mediante una medición realizada con la ayuda de la interfaz de Arduino, en la cual se cumple con las características de sistema sin carga (0 A). El código para la obtención de la señal real se muestra en la **Figura 87** y en la **Figura 88** se muestra el valor de la señal medida.

```
void setup() {  
  
  Serial.begin(9600);  
}  
  
void loop() {  
  
  float voltajeSensor =get_voltage(10000);//obtenemos voltaje del sensor(10000 muestras)  
  Serial.print("Voltaje del sensor: ");  
  Serial.println(voltajeSensor ,3);  
}  
  
float get_voltage(int n_muestras)  
{  
  float voltage=0;  
  
  for(int i=0;i<n_muestras;i++)  
  {  
    voltage =voltage+analogRead(A0) * (5.0 / 1023.0);  
  }  
  voltage=voltage/n_muestras;  
  return(voltage);  
}
```

Figura 87. Señal real de sensor con A0.



```
20:24:47.094 -> Voltaje del sensor: 2.502  
20:24:48.378 -> Voltaje del sensor: 2.502  
20:24:49.673 -> Voltaje del sensor: 2.502  
20:24:50.962 -> Voltaje del sensor: 2.502  
20:24:52.256 -> Voltaje del sensor: 2.502  
20:24:53.554 -> Voltaje del sensor: 2.502  
20:24:54.809 -> Voltaje del sensor: 2.502  
20:24:56.105 -> Voltaje del sensor: 2.502  
20:24:57.404 -> Voltaje del sensor: 2.502  
20:24:58.703 -> Voltaje del sensor: 2.502  
20:24:59.968 -> Voltaje del sensor: 2.502
```

Figura 88. Valor de voltaje medido desde el monitor serial de Arduino.

Para determinar el segundo punto de la recta, se utiliza el mismo código mencionado anteriormente, con la única diferencia de que la medición se realiza con una carga superior a 1 amperio. En este caso, se aplicó una carga de 1.622 A, cuyo resultado se muestra en la **Figura 89**.

```
20:34:18.482 -> Voltaje del sensor: 2.799
20:34:19.775 -> Voltaje del sensor: 2.808
20:34:21.071 -> Voltaje del sensor: 2.803
20:34:22.367 -> Voltaje del sensor: 2.798
20:34:23.668 -> Voltaje del sensor: 2.795
20:34:24.924 -> Voltaje del sensor: 2.801
```

Figura 89. Voltaje medido por el monitor serial cuando se aplica una carga de 1.622 A

Finalmente, para obtener el valor real de la sensibilidad del sensor, se utiliza la **Ecuación 28**. Esta ecuación permite calcular la pendiente entre el punto 1 y 2, proporcionando el valor preciso de la sensibilidad.

$$\text{Sensibilidad} = m = \frac{V_2 - V_1}{I_2 - I_1} = \frac{2.801 - 2.502}{1.622 - 0} = 0.184 \quad (28)$$

Una vez adquirido adecuadamente la sensibilidad del amperímetro, se prosigue a la calibración del programa dedicado a la obtención de los valores de corriente a ser consumidos por los motores. Para la ecuación que nos ayuda a traducir la señal que nos entrega el sensor es necesario multiplicar el valor que nos proporciona en el puerto del Arduino por el voltaje de alimentación del sensor (Vcc). Este resultado se divide entre la resolución del Arduino, que es de 10 bits (1024 datos).

Además, al valor obtenido anteriormente es necesario restarle 2.5, que corresponde al voltaje base emitido por el sensor durante la medición. Finalmente, el valor obtenido se divide entre la sensibilidad, como se muestra en la **Figura 90**.

```
for(int i=0;i<n_muestras;i++)
{
  voltajeSensor = analogRead(A1) * (5.0 / 1023.0);////lectura del sensor
  corriente=corriente+(voltajeSensor-2.5)/Sensibilidad; //Ecuación para obtener la corriente
}
```

Figura 90. Calibración del amperímetro.

Para obtener datos más precisos, es necesario la implementación de un filtro que mejore las mediciones de amperaje obtenidas por el sensor. Este proceso de filtrado se lo realiza obteniendo 500 datos de medición y calculando su promedio. En la **Figura 91** se muestra el código completo para la obtención del amperaje.

```

void loop() {

    float I=get_corriente(200);//obtenemos la corriente promedio de 500 muestras
    Serial.print("Corriente: ");
    Serial.println(I,3);
    delay(100);
}

float get_corriente(int n_muestras)
{
    float voltajeSensor;
    float corriente=0;
    for(int i=0;i<n_muestras;i++)
    {
        voltajeSensor = analogRead(A1) * (5.0 / 1023.0);////lectura del sensor
        corriente=corriente+(voltajeSensor-2.5)/Sensibilidad; //Ecuación para obtener la corriente
    }
    corriente=corriente/n_muestras;
    return(corriente);
}

```

Figura 91. Código de calibración.

La conexión física de este sensor al Arduino se realiza como se muestra en la **Figura 92**.

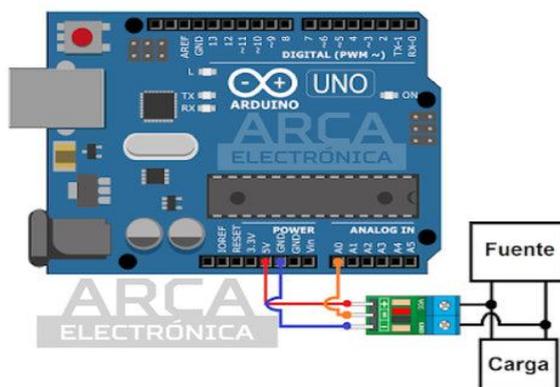


Figura 92. Conexión del sensor de corriente al Arduino.

Fuente: (Arca electrónica)

6.8.3 Sensor infrarrojo fotoeléctrico de velocidad encoder

Para medir correctamente la velocidad angular de los diferentes motores, el primer paso es comprender que la velocidad de cada motor es distinta. Esta información se obtiene a partir de los datasheets de los motores.

En el caso de motorreductor, la velocidad nominal es de 60 rpm. Para este caso, se realizan diferentes pruebas de funcionamiento utilizando ruedas con distintos números de agujeros destinados a medir las rpm, como se muestra en la **Tabla 12**.

Tabla 12. Medición de rpm con diferentes parámetros.

Velocidad angular	Intervalo de vuelta	Pruebas de funcionamiento			
		Velocidad angular medidos			
		1 agujero	2 agujeros	20 agujeros	60 agujeros
59 rpm	1/2 vuelta	60	60	57	59
	1 vuelta	0	30	63	59
60 rpm	1/2 vuelta	60	60	60	60
	1 vuelta	60	60	60	60
61 rpm	1/2 vuelta	120	90	63	61
	1 vuelta	60	60	66	61

En base a lo observado en la tabla 11, la precisión del sensor mejora a medida que aumenta el número de agujeros en la rueda. Por este motivo, se decide construir una rueda con 60 agujeros, cómo se observaba en la **Figura 93**.



Figura 93. Rueda de 60 agujeros.

En el caso del motor RS-755, que gira a altas revoluciones por minuto, es necesario implementar una rueda con un número de agujeros adecuado. Esta debe evitar tener muy pocos agujeros para que la precisión no sea baja, pero tampoco debe tener demasiados ya que esto puede dificultar su medición al ser un motor de alta velocidad.

Según el datasheet de motor, se conoce que su velocidad nominal es de 3500 rpm, lo que equivale a un aproximado de 58 vueltas por segundo. En la **Tabla 13** se muestra los valores obtenidos al medir la variación de rpm utilizando diferentes números de agujeros.

Tabla 13. *Pruebas de funcionamiento.*

Pruebas de funcionamiento		
Numero de agujeros en la rueda	Interrupciones en el sensor	Variación de rpm*min
1 agujero	58	60
2 agujeros	116	60
6 agujeros	348	60
10 agujeros	580	60
16 agujeros	928	60
20 agujeros	1160	60
58 agujeros	3480	60

Se llega a la conclusión de que el número de agujeros en las ruedas no afecta significativamente en la variación entre las mediciones. Esto se debe a las altas velocidades de giro del motor, que alcanza 58 vueltas por segundo en condiciones nominales. Por esta razón, se decide utilizar una rueda con número de agujeros intermedia, como se muestra en la **Figura 94.**



Figura 94. *Rueda de 16 agujeros*

La conexión física de los encoders al Arduino se realiza como se muestra en la **Figura 95.**

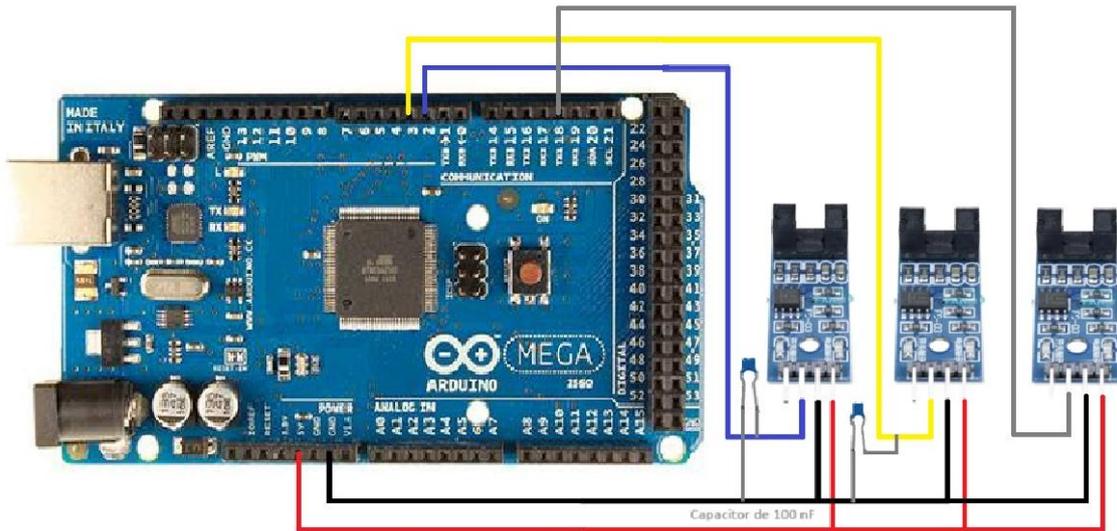


Figura 95. *Conexión de los encoders.*

El código necesario para medir con precisión las revoluciones del motor, utilizando la rueda de 16 agujeros del encoder del motor RS-755, se muestra en la **Figura 96**.

```
void loop() {
  delay(1000);          // retardo de casi 1 segundo
  Serial.print((contador/16)*60); // Son 16 interrupciones
  Serial.println(" RPM"); // El numero 16 depende del numero aspas de la helise del motor en prueba
  contador = 0;
}

void interrupcion0() // Funcion que se ejecuta durante cada interrupcion
{
  contador++; // Se incrementa en uno el contador
}
```

Figura 96. *Código para la rueda de 16 agujeros.*

6.8.4 Relé para el motor de la bomba

El encendido y apagado de motor que se encarga de poner en funcionamiento la bomba de diafragma, se realiza por medio un relé mediante la utilización del pin 8 del Arduino. Además, el control se lo realiza mediante la interfaz de Matlab la cual se encarga de enviar la señal al Arduino para activar y desactivar este dispositivo, como se muestra en la **Figura 97**.

```

// Si hay datos disponibles en el puerto serial desde MATLAB
if (Serial.available() > 0) {
  String command = Serial.readStringUntil('\n');

  if (command == "ON3") {      ///BOMBA AAAAAAAAAAAAA
    digitalWrite(relayPin8, HIGH);

    //Serial.println("OK_ON3");
  } else if (command == "OFF3") {
    digitalWrite(relayPin8, LOW);

    //Serial.println("OK_ON");
  }
}

```

Figura 97. Código para el control de motor de la bomba.

La conexión física del relé con el Arduino se realiza como se indica en la **Figura 98**.

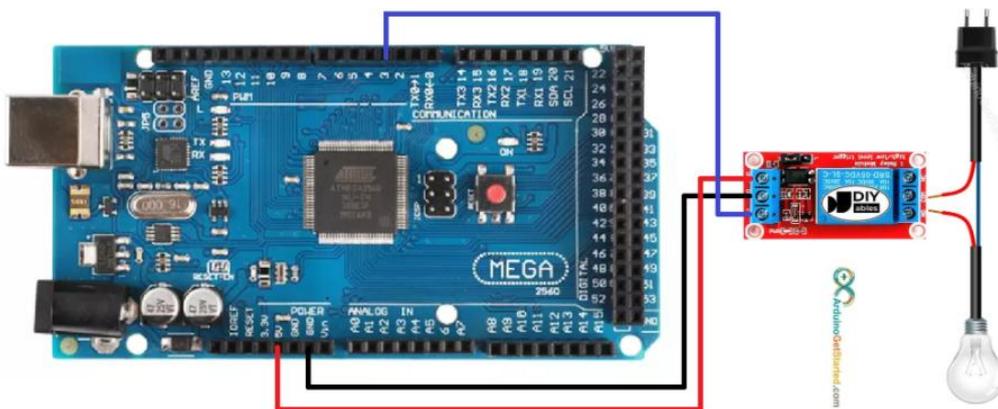


Figura 98. Conexión de relé del motor de la bomba.

6.8.5 Relés para motorreductor

Para el fácil y correcto funcionamiento del elevador de carga, se utiliza un módulo de relés. Estos cumplen la función de alimentar al motorreductor y de invertir la polaridad de la alimentación con el propósito de efectuar una inversión de giro, esta inversión de giro ayuda a elevar y a bajar la carga, para realizar el cambio de la masa e iniciar nuevamente con el ensayo, la **Figura 99** muestra el código implementado en la interfaz de Arduino.

```

if (Serial.available() > 0) {
  String command = Serial.readStringUntil('\n');

  if (command == "ON1") {
    digitalWrite(relayPin4, LOW);
    digitalWrite(relayPin5, LOW);
    //Serial.println("OK_ON");

  } else if (command == "OFF1") {
    digitalWrite(relayPin4, HIGH);
    digitalWrite(relayPin5, HIGH);
    //Serial.println("OK_OFF");

  } else if (command == "ON2") {
    digitalWrite(relayPin6, LOW);
    digitalWrite(relayPin7, LOW);
    //Serial.println("OK_ON");
  } else if (command == "OFF2") {
    digitalWrite(relayPin6, HIGH);
    digitalWrite(relayPin7, HIGH);
    //Serial.println("OK_ON")
  }
}

```

Figura 99. Código de los relés que controlan el sentido de giro del motorreductor.

El funcionamiento del código inicia cuando, en la interfaz de Matlab, se pulsa un botón llamado iniciar práctica, simultáneamente Matlab envía una señal a Arduino para encender el pin 4 y 5, que enciende los relés 1 y 2 permitiendo el giro derecho en el motor. Cuando la carga llega a la altura deseada, estos relés se apagan automáticamente.

Para descender la carga suspendida es necesario colocar en Matlab "bajar carga". Este botón envía una señal que activa el pin 6 y 7 del Arduino, invirtiendo la polaridad de la alimentación del motor. Esto permite que el motor gire en sentido izquierdo y baje la carga hasta el punto que sea necesario, la conexión de este circuito se observa en la **Figura 100**.

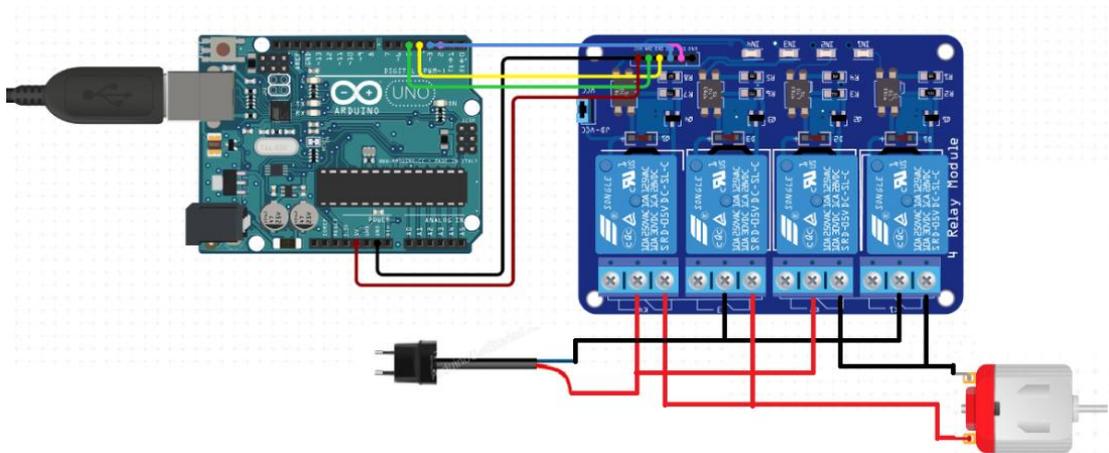


Figura 100. Inversión de giro de motorreductor.

6.8.6 Calibración sensor MAF

Para la calibración del sensor MAF(sensor de flujo de masa de aire), se utiliza un anemómetro digital. Este dispositivo permite comparar la señal de voltaje producida por el sensor frente a los diferentes cambios en los parámetros de velocidad del ventilador y las modificaciones en el área obstruida por la brida. De esta manera, se calibra el sensor y se obtienen valores aproximados a los registrados por el anemómetro.

El código de Arduino que interpreta y transforma la señal en flujo de aire se presenta en la **Figura 101**.

```
// Definir los pines
const int mafPin = A2; // Pin analógico donde está conectado el sensor MAF
const int relayPin1 = 9;
const int relayPin2 = 10;

// Arrays para los datos de calibración
float voltajes[] = {0, 0.35, 0.38, 0.6, 0.86, 1.04, 1.08, 1.19, 1.3, 1.35, 1.45, 1.5, 1.59};
float velocidades[] = {0, 0.8, 1, 1.5, 1.9, 2, 2.5, 3.2, 3.4, 4, 5.8, 6.4, 6.7};

// Parámetros del filtro de media móvil
const int numSamples = 10;
float samples[numSamples];
int sampleIndex = 0;
bool samplesFilled = false;

void setup() {
  Serial.begin(9600);
  // Configurar los pines de los relés como salida
  pinMode(relayPin1, OUTPUT);
  pinMode(relayPin2, OUTPUT);

  // Inicializar los relés apagados
  digitalWrite(relayPin1, LOW); //HIGH MOTOR
  digitalWrite(relayPin2, HIGH); //LOW SENSPR MAF

  // Inicializar el array de muestra
  for (int i = 0; i < numSamples; i++) {
    samples[i] = 0;
  }
}
```

Figura 101. Código del sensor MAF.

6.8.7 Calibración sensor de presión

En la calibración del sensor de presión digital, se utiliza un presostato analógico para comparar las mediciones de la presión ejercida por la bomba bajo diferentes factores físicos. Estos incluyen el grado de apertura de la llave que regula el paso de agua al contenedor y los cambios en la velocidad de la bomba al ajustar la relación de transmisión mediante el uso de distintas poleas. La **Figura 102** muestra el código desarrollado para la correcta interpretación de la presión.

```

1 // Constantes
2 const int sensorPin = A3; // Pin analógico al que está conectado el sensor
3 const float sensorMinVoltage = 0.5; // Voltaje mínimo del sensor (0 psi)
4 const float sensorMaxVoltage = 4.5; // Voltaje máximo del sensor (30 psi)
5 const float sensorMinPressure = 0.0; // Presión mínima (0 psi)
6 const float sensorMaxPressure = 72.5189; // Presión máxima (30 psi)
7
8 // Variables
9 int sensorValue = 0; // Valor analógico leído del sensor
10 float voltage = 0.0; // Voltaje calculado a partir del valor analógico
11 float pressure = 0.0; // Presión calculada
12 float lastValidPressure = 0.0; // Último valor de presión positivo válido
13
14 void setup() {
15     Serial.begin(9600); // Inicia la comunicación serie a 9600 bps
16 }
17
18 void loop() {
19     // Leer el valor analógico del sensor
20     sensorValue = analogRead(sensorPin);
21
22     // Convertir el valor analógico a voltaje
23     voltage = sensorValue * (5.0 / 1023.0);
24
25     // Mapear el voltaje a la presión correspondiente
26     pressure = mapVoltageToPressure(voltage, sensorMinVoltage, sensorMaxVoltage, sensorMinPressure, sensorMaxPressure);
27
28     // Verificar si la presión es negativa
29     if (pressure < 0) {
30         // Mantener el último valor positivo válido
31         pressure = lastValidPressure;

```

Figura 102. Código de sensor de presión.

6.8.8 Calibración de final de carrera

El final de carrera es un instrumento diseñado para protección del sistema de elevación, dado que permite la detención automáticamente del motor cuando la carga entra en contacto con este sensor. La **Figura 103** muestra el código para la calibración del final de carrera.

```

finaldecarrera.ino
1 const int pinFinalCarrera = 5; // Pin del final de carrera
2 const int pinRelay1 = 6; // Pin del primer relé
3 const int pinRelay2 = 7; // Pin del segundo relé
4
5 void setup() {
6     pinMode(pinFinalCarrera, INPUT); // Configura el pin del final de carrera como entrada
7     pinMode(pinRelay1, OUTPUT); // Configura el pin del primer relé como salida
8     pinMode(pinRelay2, OUTPUT); // Configura el pin del segundo relé como salida
9     digitalWrite(pinRelay1, HIGH); // Inicializa el primer relé apagado
10    digitalWrite(pinRelay2, HIGH); // Inicializa el segundo relé apagado
11 }
12
13 void loop() {
14     int estadoFinalCarrera = digitalRead(pinFinalCarrera); // Lee el estado del final de carrera
15
16     if (estadoFinalCarrera == HIGH) { // Si el final de carrera está accionado (5V)
17         digitalWrite(pinRelay1, LOW); // Apaga el primer relé
18         digitalWrite(pinRelay2, LOW); // Apaga el segundo relé
19     } else {
20         digitalWrite(pinRelay1, HIGH); // Enciende el primer relé
21         digitalWrite(pinRelay2, HIGH); // Enciende el segundo relé
22     }
23 }
24

```

Figura 103. Código de calibración del final de carrera.

6.9 Validación de funcionamiento

La validación del funcionamiento se realiza mediante la verificación física de las mediciones realizadas por los diversos sensores. En el caso específico de los sensores de voltaje y corriente, se efectúa una comprobación comparativa entre los valores obtenidos por estos sensores, los valores medidos mediante un multímetro y los valores configurados en una fuente reguladora de voltaje y corriente. Esta comparación asegura la precisión de las mediciones y valida la fiabilidad de los datos obtenidos.

Las **Figura 104** se ilustra como se realizó la medición de los valores correspondientes al voltaje y corriente entre los diferentes equipos empleados.

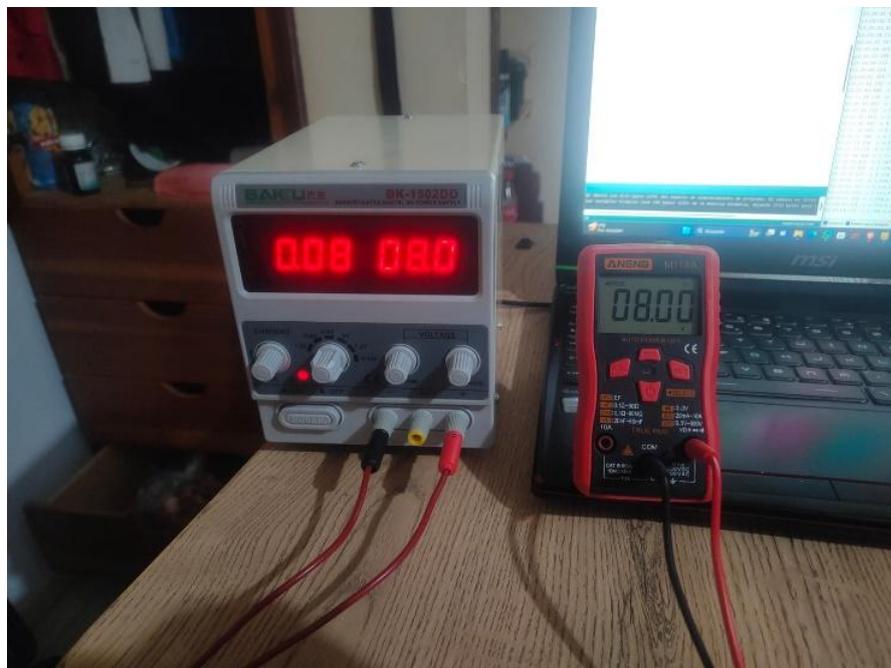


Figura 104. Comparación de valores entre fuente y multímetro.

A continuación, se calcula el porcentaje de error entre los valores medidos por los sensores incorporados en el banco, los valores obtenidos con un multímetro y los valores proporcionados por una fuente variable. Este cálculo se realiza utilizando la **Ecuación 29**, considerando que los valores más confiables son aquellos medidos por el multímetro. Los errores calculados se presentan en la **Tabla 14 y 15**.

$$\text{error relativo} = \frac{|valor\ aproximado - valor\ exacto|}{valor\ exacto} * 100 \quad (29)$$

Tabla 14. Error porcentual en los valores voltaje medidos por los diferentes equipos.

Voltaje (V)		
Sensor (V)	Multímetro (V)	Error %
11.95	11.96	0.08
10.97	10.97	0
10	9.97	0.30
8.94	8.92	0.22
8.09	8.05	0.50
6.96	6.93	0.43
6.06	6.02	0.66
5.06	5.01	0.99

A continuación, en la **Figura 105** se presenta la gráfica sobre la medición de voltaje y su error porcentual.

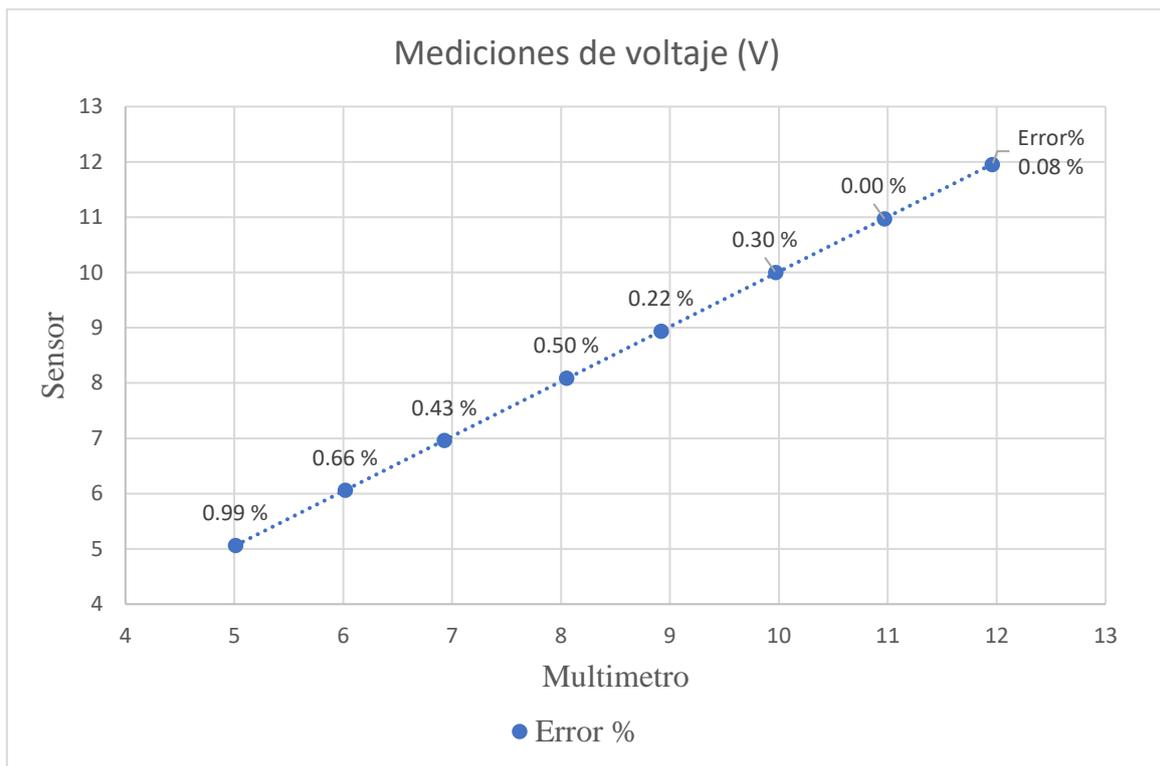


Figura 105. Gráfica de voltaje

Tabla 15. Error porcentual en los valores corriente medidos por los diferentes equipos.

Corriente (I)		
Sensor (A)	Multímetro (A)	Error %
0.083	0.084	1.19
0.225	0.227	0.88
0.296	0.297	0.34
0.453	0.455	0.44
0.562	0.565	0.53
0.663	0.667	0.60
0.723	0.725	0.28
0.851	0.856	0.58
0.955	0.96	0.52
1.408	1.42	0.85

A continuación, en la **Figura 106** se presenta la gráfica sobre la medición de corriente y su error porcentual.

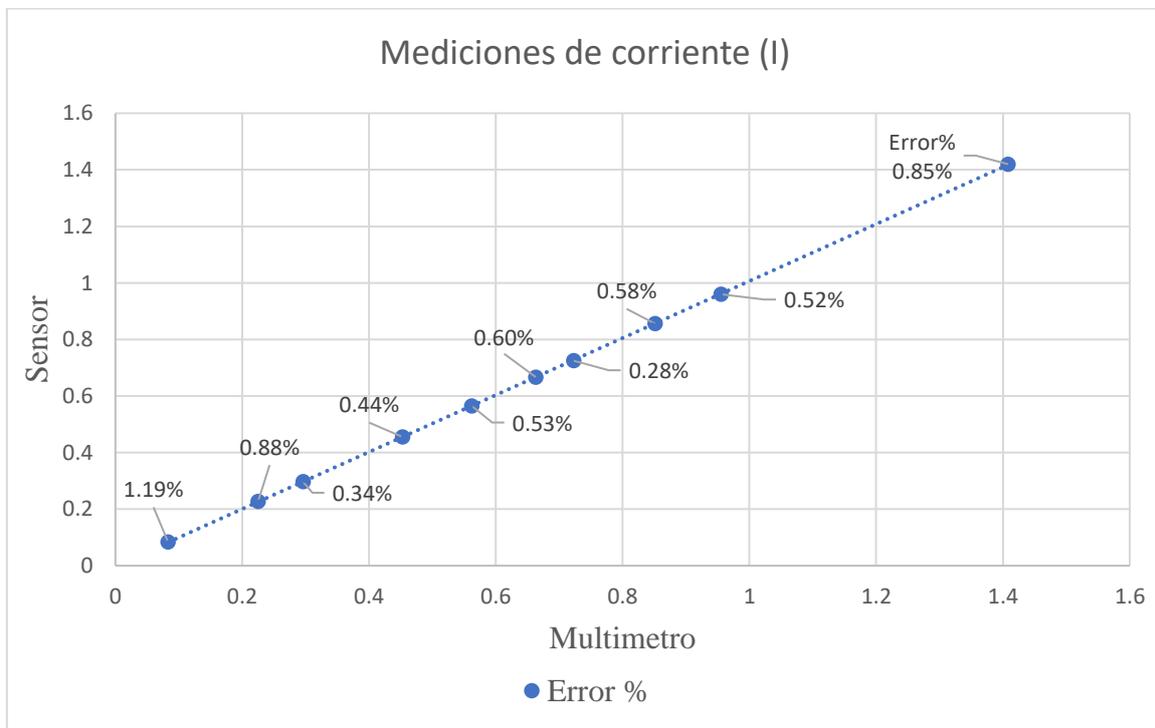


Figura 106. Corriente.

Para validar el funcionamiento del encoder o sensor encargado de medir las revoluciones generadas por el motor bajo diversas condiciones de carga, se lleva a cabo un proceso de evaluación. Este proceso consiste en medir las revoluciones del motor mientras está conectado al sistema de la grúa sin carga. Durante esta evaluación, se varía el voltaje de alimentación y se analiza el comportamiento del motor ante distintas situaciones de trabajo. Además, los valores obtenidos se comparan con los valores especificados en la ficha técnica del motor, como se ilustra en la **Figura 107**.

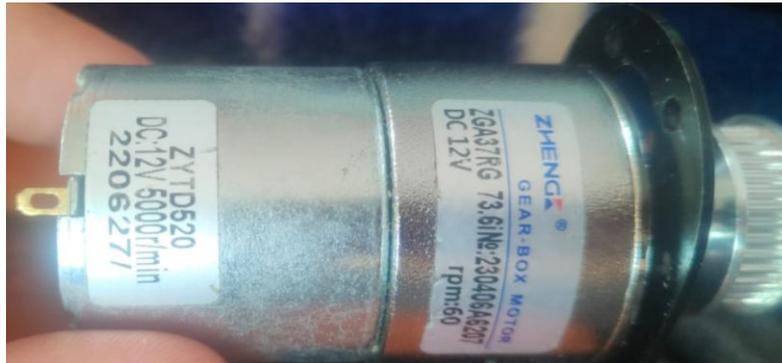


Figura 107. Características del motorreductor.

Adicionalmente, se verifica la precisión de las mediciones del encoder mediante un método visual. Este consiste en utilizar un cronómetro y colocar una marca en la polea para facilitar el conteo de las revoluciones por minuto en el motor, como se muestra en la **Figura 108**.

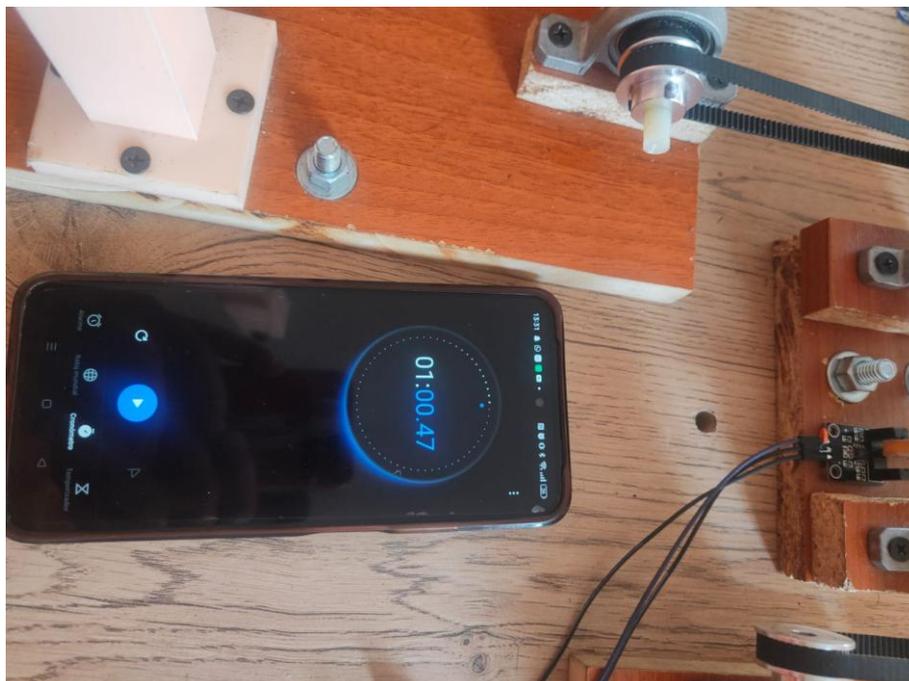


Figura 108. Medición visual de vueltas del motor en un minuto.

Finalmente, se emplea la **Tabla 16** para registrar y comparar los valores obtenidos por el encoder y el método visual bajo diferentes condiciones de voltaje. Este análisis permite calcular el error porcentual en las mediciones de rpm, proporcionando una evaluación detallada de la precisión del sensor.

Tabla 16. Velocidad del motorreductor ante diferentes voltajes de alimentación.

Velocidad angular (W)			
Voltaje (V)	Medición de sensor (rpm)	Medición visual (rpm)	Error %
12	69	69	0
11	62	62	0
10	55	55	0
9	47	47	0
8	42	42	0
7	36	36	0
6	33	33	0
5	24	24	0

Para validar el correcto funcionamiento del encoder encargado de medir las revoluciones por minuto producidas por el motor RS775, se utiliza un procedimiento que emplea un micrófono, el cual se encarga de receptar todas las ondas de frecuencia cercanas. La medición realizarse en un ambiente libre ruidos externos que puedan interferir con la frecuencia de giro del motor. Los pasos que se deben seguir para este método son los siguientes:

- Conectar el micrófono a la entrada de audio de la computadora.
- Descargar e instalar la aplicación Audacity.
- Configurar las opciones de la aplicación como: frecuencia de muestreo (44100 Hz) y el formato de muestra en 16 bits.
- Grabar el sonido emitido por el motor al girar
- Exportar el archivo de audio
- Utilizar Matlab para mejorar e interpretar la señal medida.

Las **Figuras 109 y 110** muestran la comparación de los valores medidos para diferentes niveles de voltaje, tanto con el encoder como con el micrófono, permitiendo generar la **Tabla 17**.

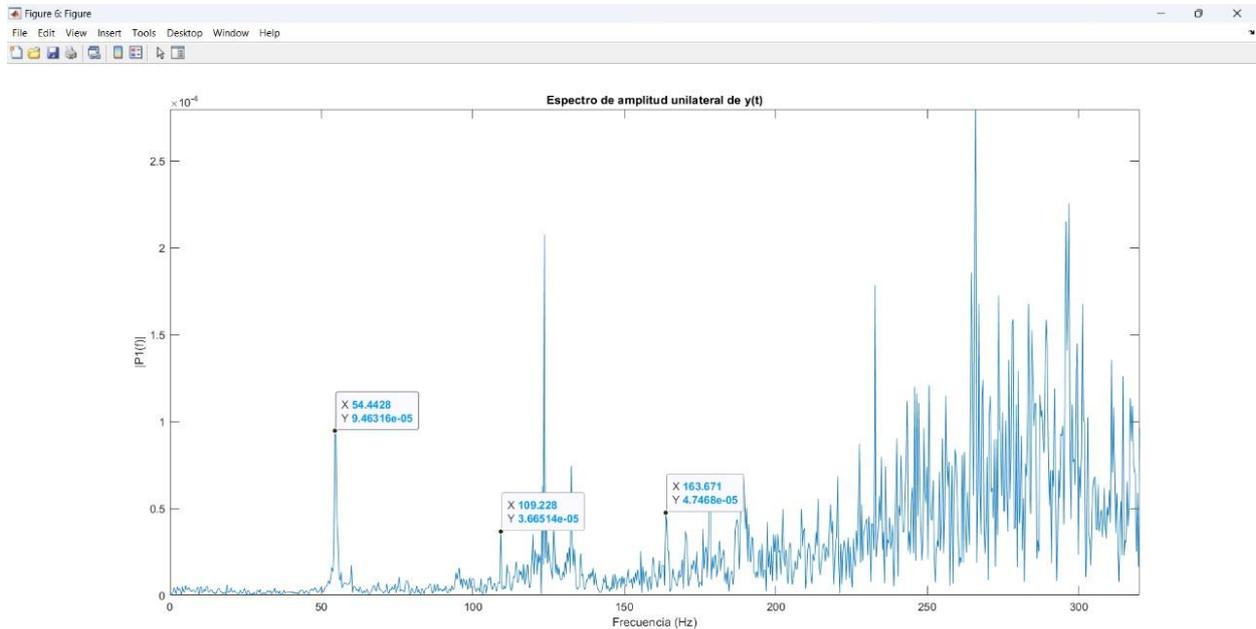


Figura 109. Medición de la frecuencia de giro del motor por medio de un micrófono

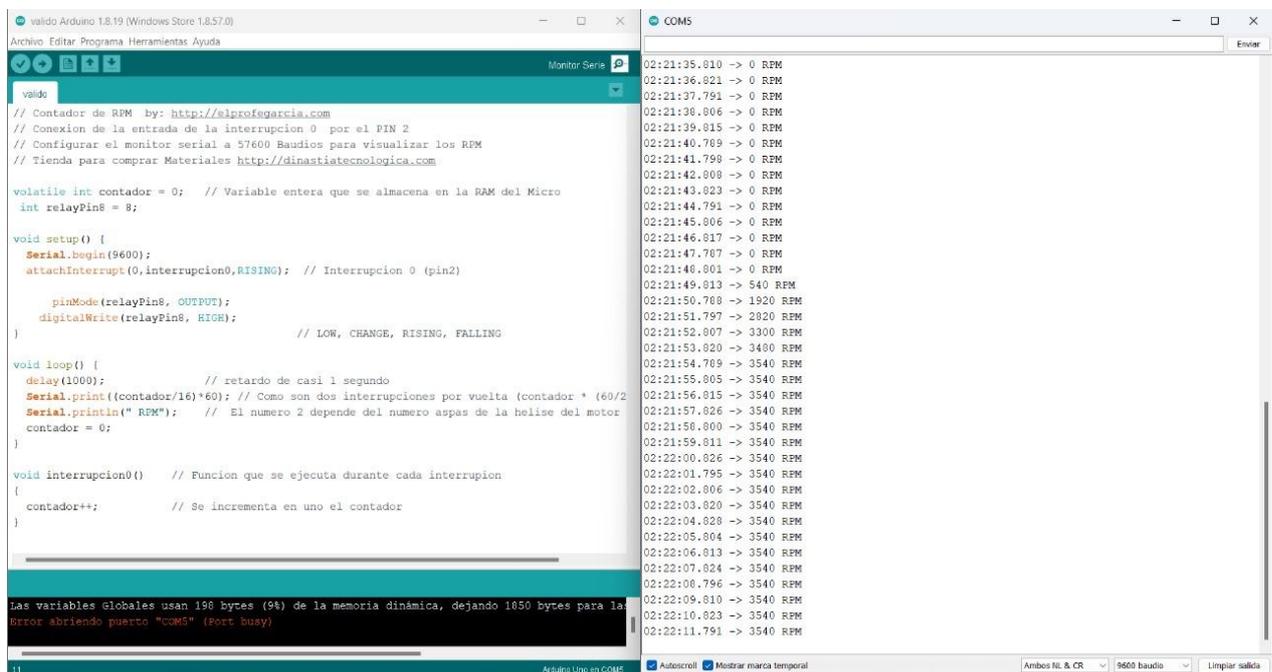


Figura 110. Medición de la velocidad angular del motor por medio del encoder.

Tabla 17. Validación del encoder.

Validación de encoder para el motor RS77					
Voltaje	Valores de medidos por el micrófono		Valores medidos por el encoder		Error %
V	rpm*s	rpm*min	rpm*s	rpm*min	%
14	69.8393	4190.358	73	4380	4.52
13	64.0491	3842.946	64	3840	0.07
12	58.7743	3526.458	59	3540	0.38
11	54.0846	3245.076	54	3240	0.15
10	48.6845	2921.07	49	2940	0.64
9	43.6918	2621.508	44	2640	0.70
8	38.7122	2322.732	38	2280	1.83
7	33.0517	1983.102	32	1920	3.18
6	28.2412	1694.472	27	1620	4.39
5	23.2429	1394.574	22	1320	5.34

Mediante el análisis de la **Tabla 17**, se ha representado gráficamente el comportamiento del error de medición frente a distintos cambios de voltaje, lo que genera variaciones en la frecuencia de giro en el motor. Este análisis se refleja de manera visual en la **Figura 111**.

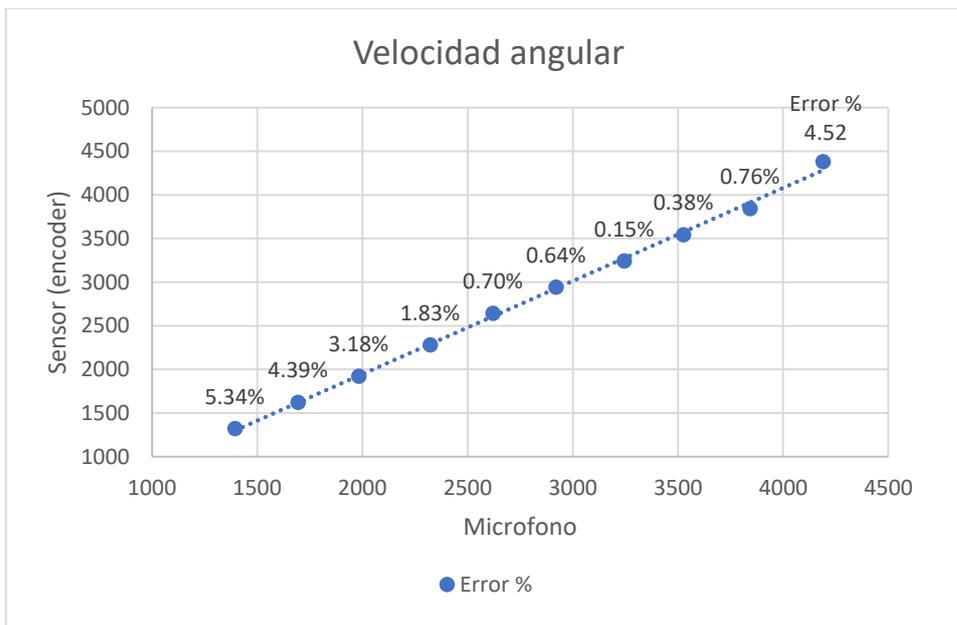


Figura 111. Gráfica de velocidad angular.

6.10 Caracterización y análisis el comportamiento del par motor y resistente

6.10.1 Sistema de elevación de cargas.

El comportamiento del par motor del motor-reductor al poner en funcionamiento la grúa se muestra en la **Figura 112**. En el caso sin carga, el par motor tiene el siguiente comportamiento: al iniciar, el par motor se incrementa para vencer la resistencia inicial, que

incluye la fricción en los mecanismos de la grúa y la masa inercial de la carga. A medida que la carga se eleva, este proceso continúa hasta alcanzar un punto de estabilización, donde el par motor se equilibra con la carga y se logra un movimiento uniforme.



Figura 112. Funcionamiento para toma de datos del sistema de elevación.

En la **Tabla 18** se exhiben los datos recopilados de los múltiples ensayos realizados en el sistema de grúa. Durante estos ensayos, se modificaron los parámetros de las cargas a ser elevadas, con el propósito de evaluar el comportamiento del motor frente a estas variaciones en las condiciones de trabajo.

Tabla 18. Ensayos del sistema de elevación de cargas.

Sistema de grúa							
Primera tabla de ensayos							
Polea 36 - 36	Voltaje (V)	Corriente (A)	Velocidad angular (rpm)	Potencia eléctrica (W)	Potencia mecánica (W)	Rendimiento	Torque (N*cm)
Sin carga	12.10	0.16	73	1.93	0	0	22.37
Carga de 1,5 lb	12.07	0.19	73	2.29	0.51	22.24	25.99
Carga de 3,5 lb	12.03	0.31	69	3.73	1.13	30.17	40.31
Carga de 5,5 lb	11.97	0.40	66	4.79	1.69	35.32	49.60
Carga de 7,5 lb	11.97	0.53	64	6.34	2.24	35.25	59.03
Segunda tabla de ensayos							
Polea 36 - 40	Voltaje (V)	Corriente (A)	Velocidad angular (rpm)	Potencia eléctrica (W)	Potencia mecánica (W)	Rendimiento	Torque (N*cm)
Sin carga	12.10	0.13	69	1.57	0	0	19.78
Carga de 1,5 lb	12.01	0.15	67	1.80	0.47	25.99	22.95
Carga de 3,5 lb	12.00	0.26	64	3.12	1.04	33.45	37.98
Carga de 5,5 lb	12.00	0.37	61	4.44	1.56	35.20	51.29
Carga de 7,5 lb	11.97	0.48	57	5.75	1.99	34.66	63.45

Tercera tabla de ensayos

Polea 36 - 30	Voltaje (V)	Corriente (A)	Velocidad angular (rpm)	Potencia eléctrica (W)	Potencia mecánica (W)	Rendimiento	Torque (N*cm)
Sin carga	12.07	0.15	93	1.81	0	0	16.63
Carga de 1,5 lb	12.06	0.19	88	2.29	0.61	26.84	21.54
Carga de 3,5 lb	12.01	0.32	83	3.84	1.35	35.22	34.20
Carga de 5,5 lb	11.97	0.47	78	5.63	2.00	35.53	45.89
Carga de 7,5 lb	11.95	0.62	73	7.41	2.55	34.43	54.18

Con los datos obtenidos en la tabla anterior, se generan las curvas características del motorreductor, como se muestra en **Figura 113**, donde se toma en cuenta la velocidad, potencia, voltaje y corriente con respecto al torque para el sistema de transmisión conformado por poleas dentadas de 36 a 36 dientes.

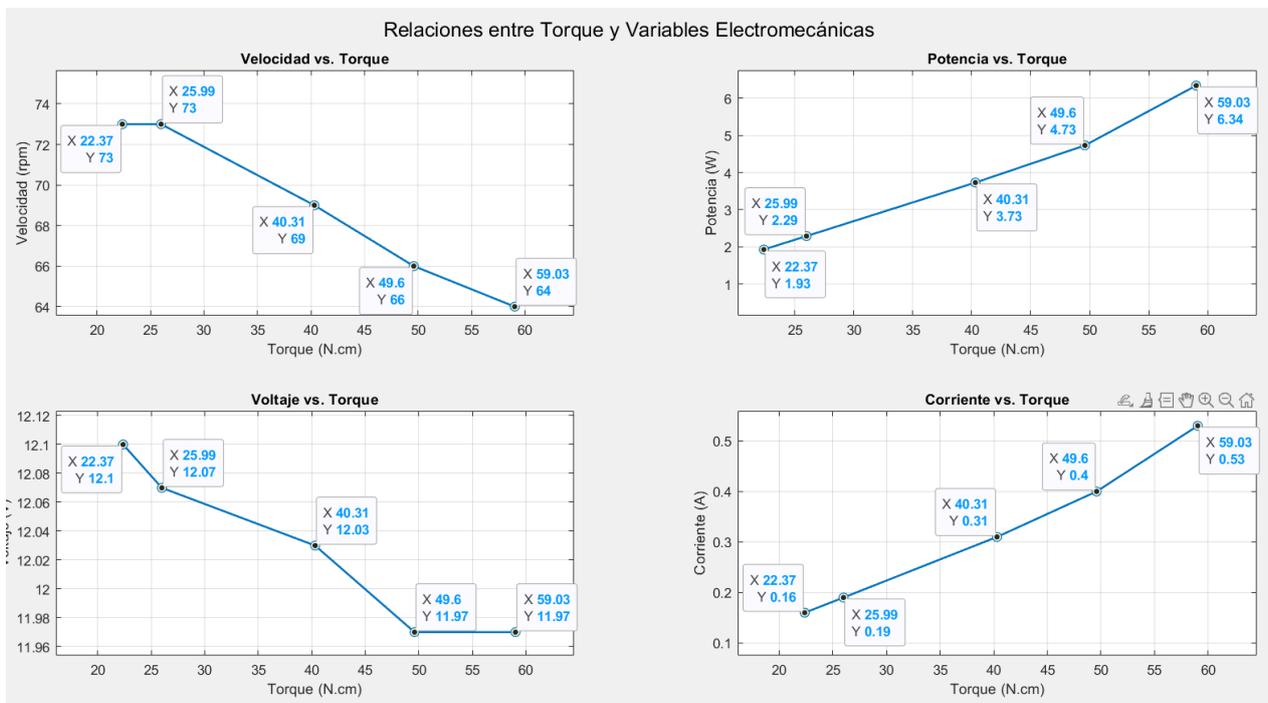


Figura 113. Gráfica de análisis de datos de la velocidad, potencia voltaje y corriente con respecto al torque utilizando poleas de 36 -36 dientes.

Para la segunda prueba, se realizan las mismas curvas del motorreductor como se muestra en la **Figura 114**, con la diferencia de que la relación de transmisión se modifica de la siguiente manera: en el eje del motorreductor se conserva la polea de 36 dientes, mientras que en el eje de enrolladora de cable se utiliza una polea de 40 dientes.

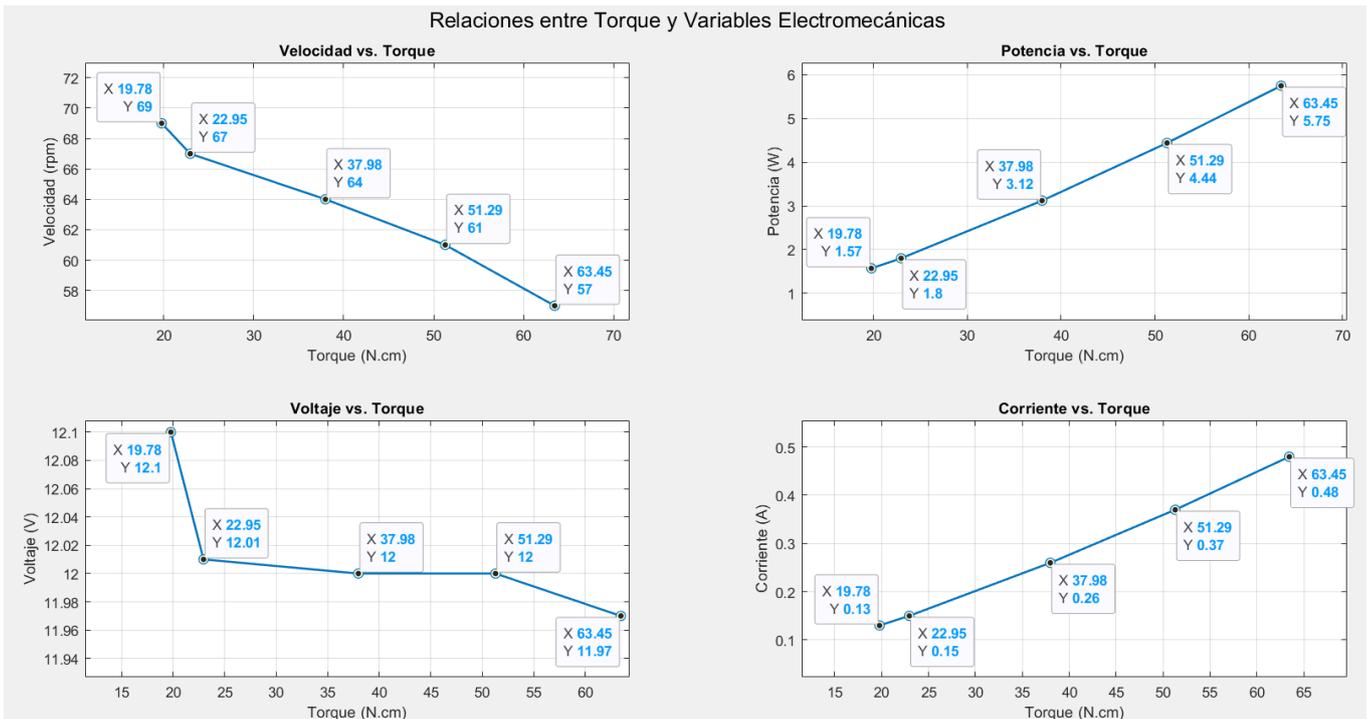


Figura 114. Gráfica de análisis de datos de la velocidad, potencia, voltaje y corriente respecto al torque polea 36 – 40 dientes.

En la tercera prueba se elaboran las mismas curvas anteriores, como se indica en la **Figura 115**, manteniendo la polea que se encuentra en el eje del reductor y cambiando la polea del eje de la enrolladora de cable por una de 30 dientes.

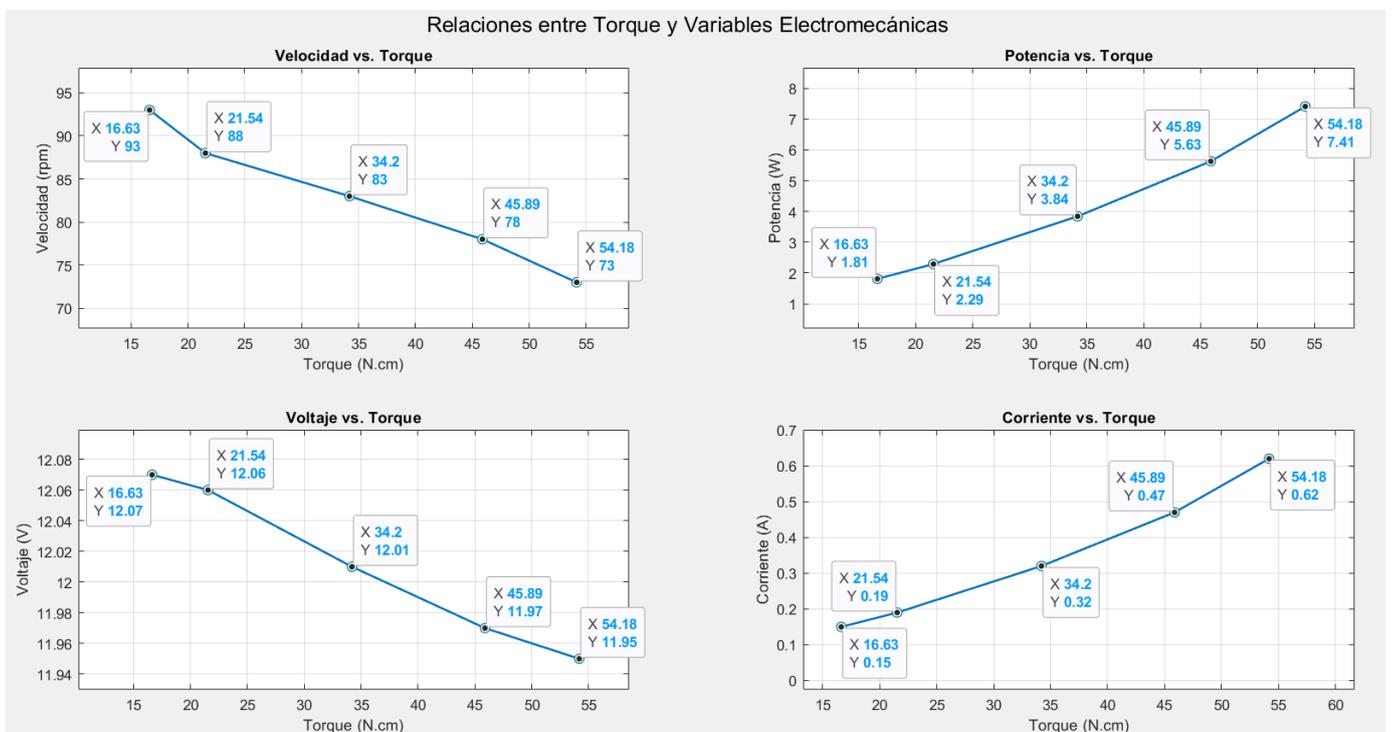


Figura 115. Gráfica de análisis de datos de la velocidad, potencia, voltaje y corriente respecto al torque polea 36 – 30 dientes.

6.10.2 Sistema de bombeo

El comportamiento del par motor de este motor DC, encargado de impulsar la bomba de diafragma, se manifiesta de la siguiente manera: al iniciar, el par motor aumenta para superar el par resistente del sistema, que incluye la fricción en los mecanismos de la bomba (expansión del diafragma) y el par necesario para desplazar el líquido fuera de la cámara de bombeo hacia la línea de descarga. A medida que la bomba supera la fase de arranque, el par disminuye hasta alcanzar un punto en el cual se mantiene constante. Este comportamiento se evalúa en la **Figura 116**.

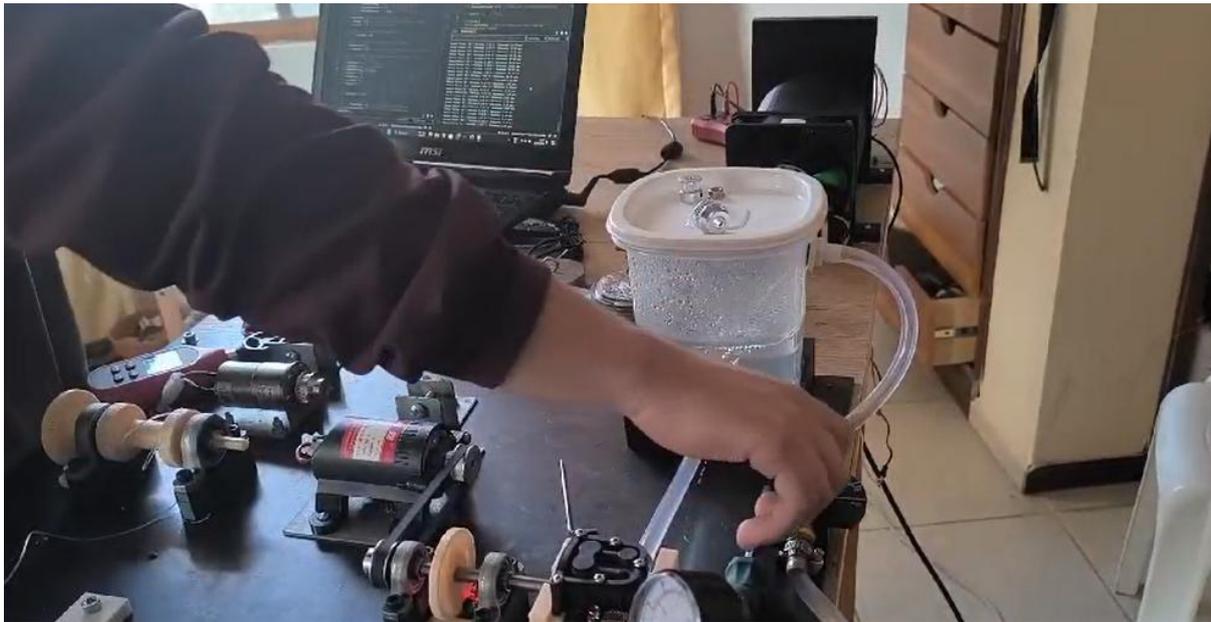


Figura 116. Funcionamiento del sistema de bombeo para tomas de datos.

Al momento de simular carga o mayor altura de descarga para la bomba, se observa que el par motor sufre un aumento en comparación con el caso sin carga o con altura mínima. Sin embargo, el comportamiento general es parecidas, ya que después del intervalo de arranque, el torque tiende a estabilizarse. En la **Tabla 19** se presenta el comportamiento de la bomba frente a diferentes circunstancias de carga.

Tabla 19. Ensayos del sistema de bombeo.

Sistema de bombeo							
Primera tabla de ensayos							
Polea 36 - 36	Voltaje (V)	Corriente (A)	Presión de sensor (PSI)	Presión de manómetro (PSI)	Velocidad angular (rpm)	Potencia eléctrica (W)	Torque (N*cm)
Llave abierta 90°	11.82	1.26	3.77	3.00	3423	14.89	2.38
Llave abierta 60°	11.80	1.28	4.23	4.43	3363	15.10	2.43
Llave abierta 30°	11.79	1.31	13.03	13.03	3314	15.44	2.47
llave 0°	11.78	1.55	55.44	55.13	3229	18.26	2.56

Segunda tabla de ensayos

Polea 36 - 40	Voltaje (V)	Corriente (A)	Presión de sensor (PSI)	Presión de manómetro (PSI)	Velocidad angular (rpm)	Potencia eléctrica (W)	Torque (N*cm)
Llave abierta 90°	11.83	1.12	2.33	2.00	3188	13.25	2.47
Llave abierta 60°	11.82	1.13	5.59	5.00	3173	13.36	2.48
llave abierta 30°	11.80	1.22	11.80	12.00	3031	14.40	2.66
llave 0°	11.79	1.24	81.35	81.00	3018	14.62	2.68

Tercera tabla de ensayos

Polea 36 - 30	Voltaje (V)	Corriente (A)	Presión de sensor (PSI)	Presión de manómetro (PSI)	Velocidad angular (rpm)	Potencia eléctrica (W)	Torque (N*cm)
Llave abierta 90°	11.81	1.57	5.79	6.00	3865	18.54	2.15
Llave abierta 60°	11.74	1.66	6.33	6.00	3774	19.49	2.14
llave abierta 30°	11.73	1.85	9.93	10.00	3668	21.70	2.09
llave 0°	11.71	1.87	70.43	70.00	3663	21.90	2.06

Para el caso del sistema de bombeo, una vez obtenidos los datos mediante el cambio de flujo de agua, se procede a graficar los resultados como se observa en la **figura 117**. Este procedimiento sigue la misma metodología aplicada anteriormente en el sistema de elevación de carga, con el objetivo de obtener las curvas de comportamiento del motor al modificarse la presión, en este primer caso se utilizan poleas dentadas de 36 a 36 en los ejes respectivos.

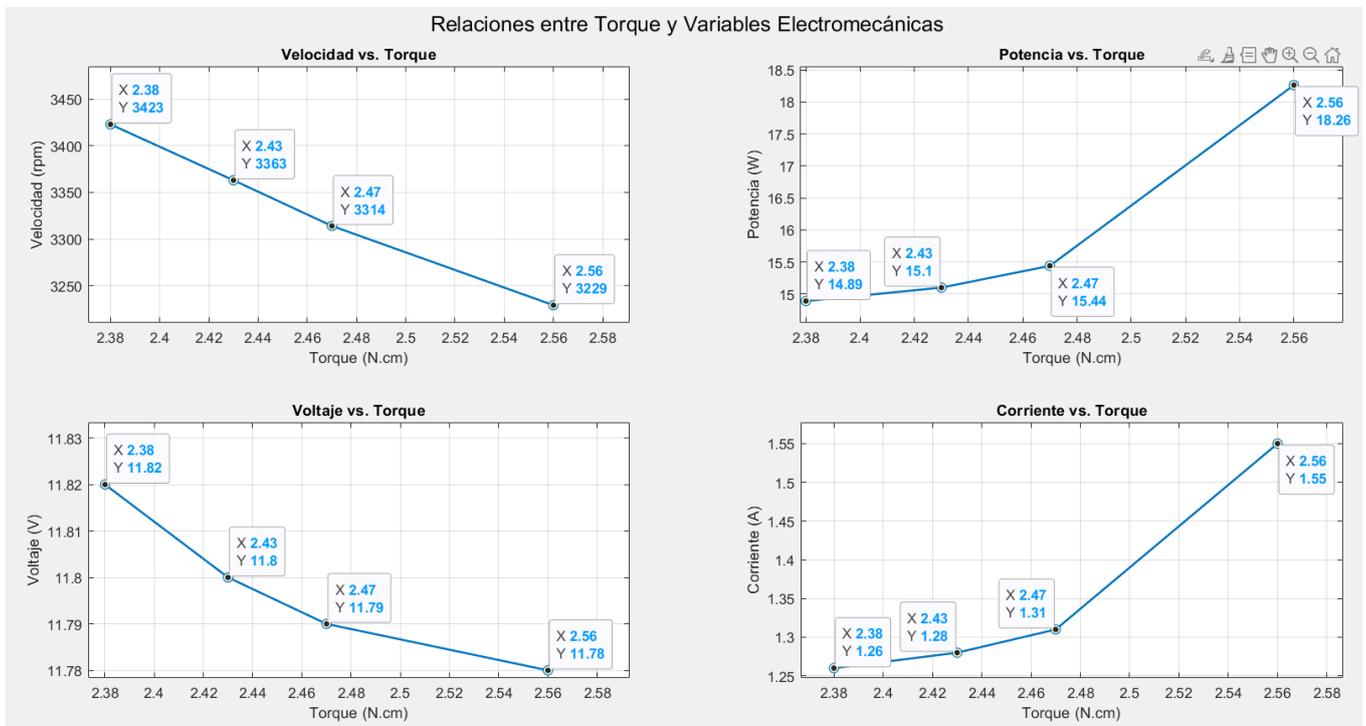


Figura 117. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 -36 dientes.

Como se muestra en la **Figura 118**, para la obtención de las curvas características del motor en este segundo caso, se mantiene la polea de 36 dientes en el eje del motor, mientras que en el eje de la bomba se utiliza una polea de 40 dientes. Esta configuración provoca que las rpm bajen en comparación con la transmisión anterior.

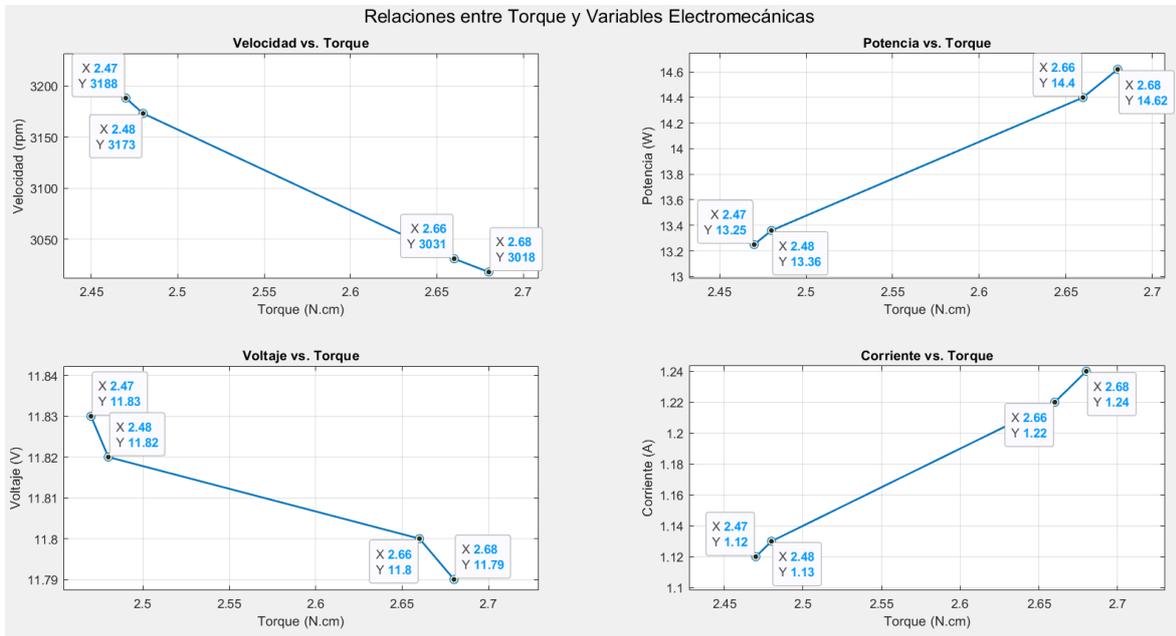


Figura 118. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 – 40 dientes.

En la **Figura 119** se muestra las curvas características del sistema de bombeo, donde se mantiene la polea del eje del motor y reemplazando la polea de la bomba por una de 30 dientes, aumentando así los rpm.

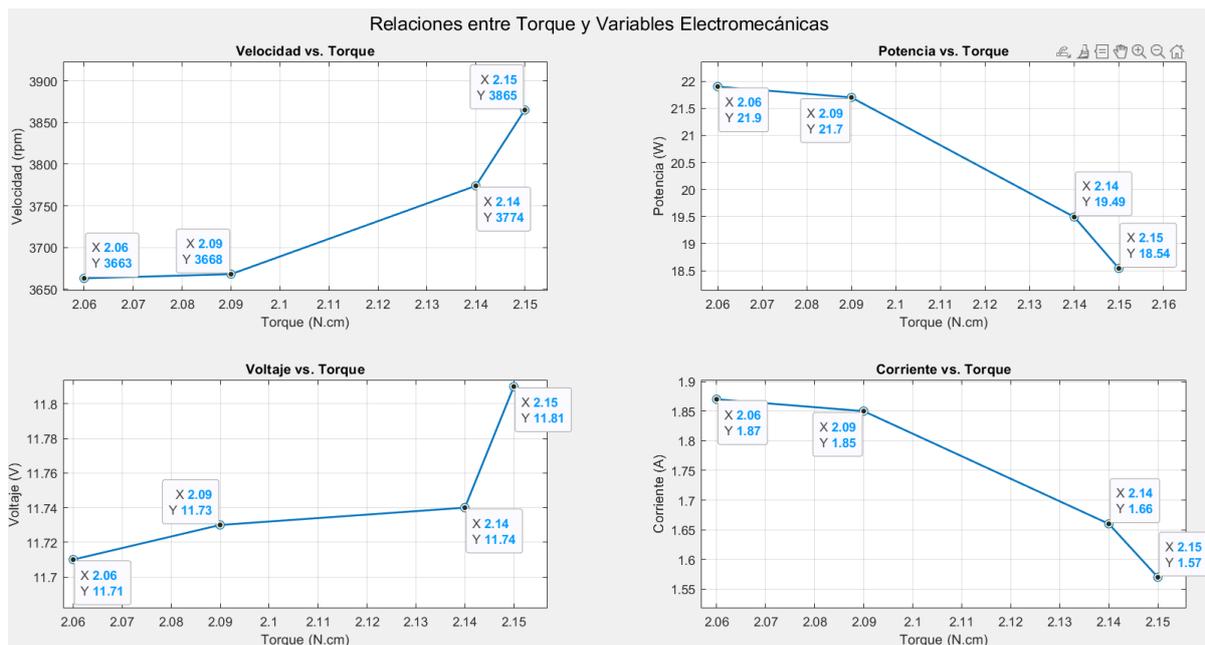


Figura 119. Análisis de datos de velocidad, potencia, voltaje y corriente con referente al torque del motor RS775 polea 36 -30 dientes.

6.10.3 Sistema de ventilación.

El comportamiento del par motor, encargado de impulsar un sistema de ventilación, se manifiesta de la siguiente manera: al iniciar, el par motor se eleva para superar el par resistente del sistema, que incluye la fricción en los componentes del ventilador (como los cojinetes y el rotor) y el par necesario para iniciar el movimiento del aire a través del sistema de ventilación. A medida que el ventilador supera la fase de arranque, el par disminuye hasta alcanzar un punto en el cual se mantiene constante. Para analizar este comportamiento se realiza la toma de datos como se muestra en la **Figura 120**.



Figura 120. *Funcionamiento del sistema de ventilación para la toma de datos.*

Al momento de simular una mayor carga, como un incremento en la resistencia al flujo de aire dentro del el sistema de ventilación, se observa que el par motor sufre un aumento durante el arranque en comparación con las condiciones trabajo sin carga adicional. En la **Tabla 20** se presenta el comportamiento del sistema de ventilación frente a diferentes circunstancias de carga.

Tabla 20. *Ensayos del sistema de ventilación.*

Sistema de ventilación							
Primera tabla de ensayos							
Polea 16 - 36	Velocidad angular (rpm)	flujo de aire	Señal de sensor MAF	Voltaje (V)	Corriente (A)	Potencia eléctrica (W)	Torque (N*cm)
Brida sin obstrucción	3300	3.40	1.30	11.88	1.61	18.29	4.09
Brida con 25% de obstrucción	3240	2.00	1.04	11.49	1.65	18.96	4.30
Brida con 50% de obstrucción	3240	1.00	0.38	11.49	1.70	19.53	4.39
Brida con obstrucción	3180	0	0	11.36	1.83	21.74	4.92
Segunda tabla de ensayos							
Polea 16 - 40	Velocidad angular (rpm)	flujo de aire	Señal de sensor MAF	Voltaje (V)	Corriente (A)	Potencia eléctrica (W)	Torque (N*cm)
Brida sin obstrucción	3060	3.20	1.19	11.55	1.42	16.34	4.09
Brida con 25% de obstrucción	3000	1.90	0.86	11.51	1.55	17.69	4.41
Brida con 50% de obstrucción	2940	0.80	0.35	11.46	1.57	17.99	4.56
Brida con obstrucción	2940	0	0	11.41	1.59	18.36	4.65
Tercera tabla de ensayos							
Polea 16 - 30	Velocidad angular (rpm)	flujo de aire	Señal de sensor MAF	Voltaje (V)	Corriente (A)	Potencia eléctrica (W)	Torque (N*cm)
Brida sin obstrucción	3600	4.00	1.35	11.93	2.26	25.88	4.70
Brida con 25% de obstrucción	3410	2.50	1.08	11.93	2.43	28.99	5.47
Brida con 50% de obstrucción	3320	1.50	0.60	11.90	2.47	29.47	5.67
Brida con obstrucción	3230	0	0	11.45	2.65	31.54	6.00

Para el caso del sistema de ventilación, los datos se obtienen variando la cantidad de flujo que pasa a través del conducto mediante el uso de diferentes bridas. Esto permite registrar los valores mostrados en la tabla anterior y proceder a generar las curvas del motor, de manera similar a los sistemas anteriores. En este caso, se utiliza una polea de 16 dientes en el eje del motor, ya que con esta se logran las relaciones de transmisión más idóneas para que el ventilador opere dentro del rango de revoluciones por minuto recomendadas por el fabricante. Además, en este caso se utilizó una polea de 36 dientes en el eje del ventilador, en la **Figura 121** se visualizan las gráficas obtenidas con esta relación de transmisión.

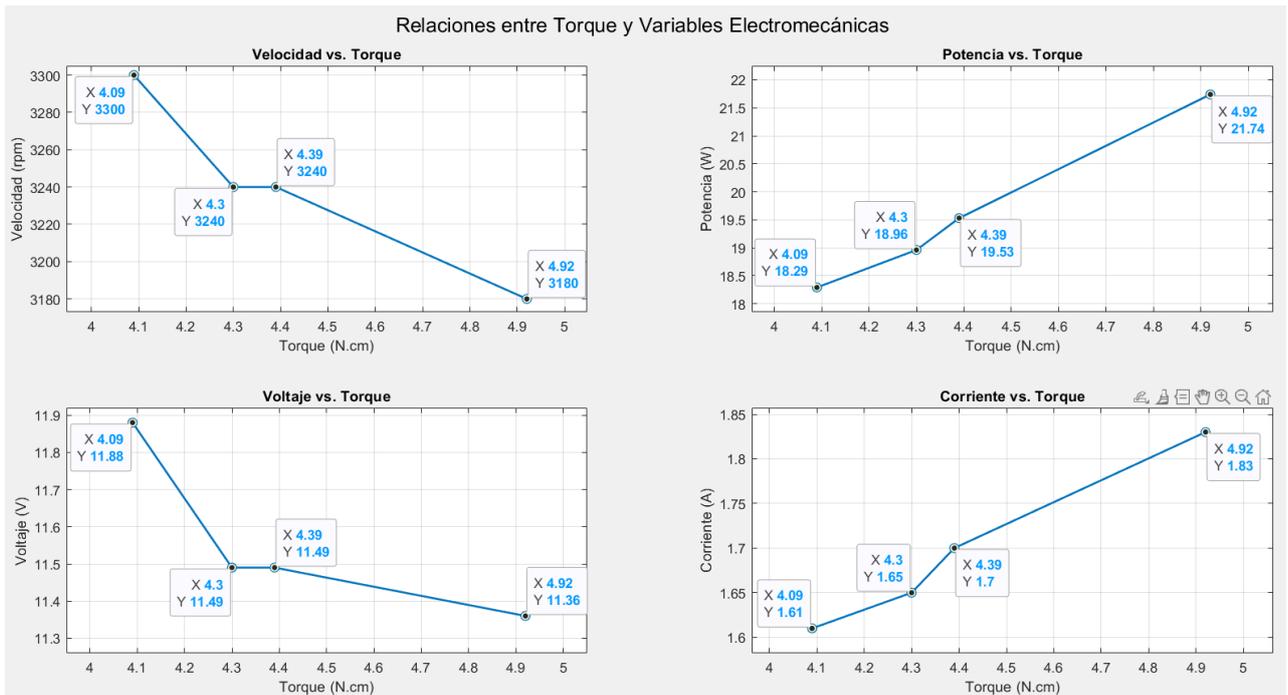


Figura 121. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 -36 dientes.

Los resultados del segundo ensayo realizado en el sistema de ventilación se muestran en la **Figura 122**, donde se utiliza una polea de 40 dientes en el eje del ventilador con el objetivo de disminuir la velocidad de trabajo.

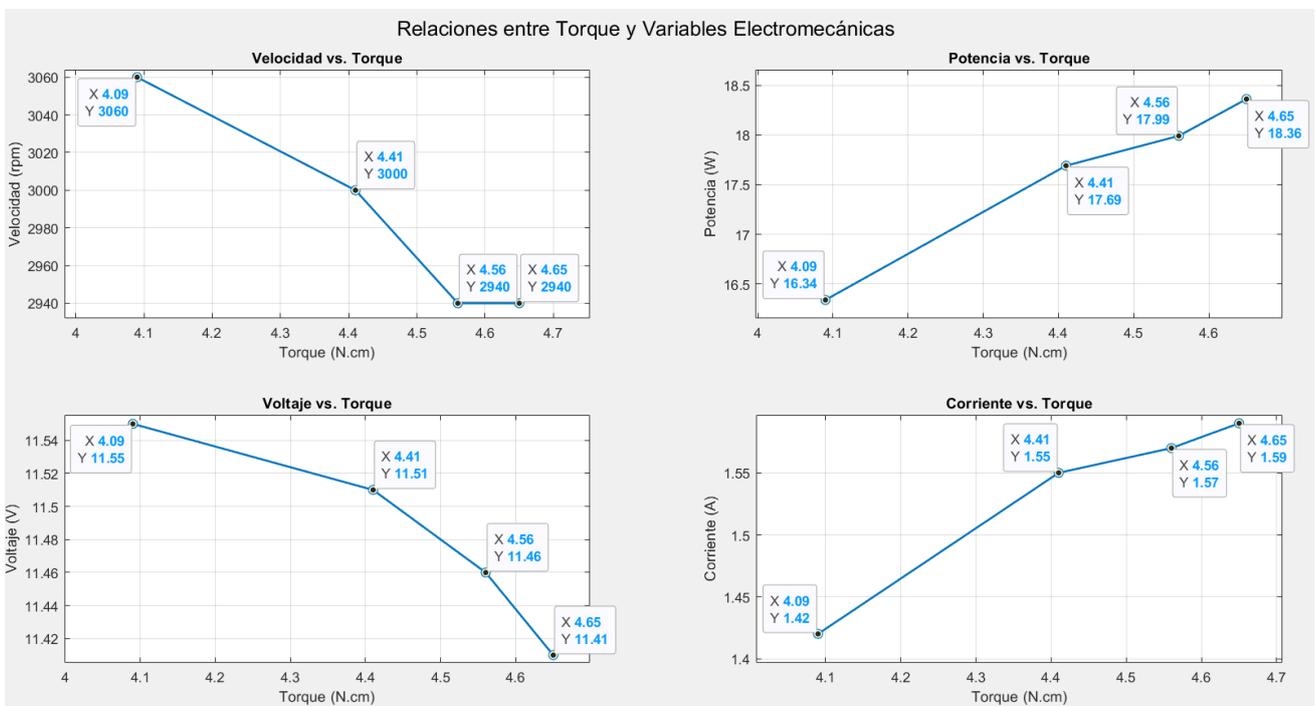


Figura 122. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 – 40 dientes.

La última variación del sistema de ventilación realizada para las pruebas se la observa en la **Figura 123**. En este caso, se utiliza una polea de 30 dientes en el eje del ventilador, con el propósito de incrementar la velocidad de trabajo mediante el sistema de transmisión, en comparación con los casos anteriores.

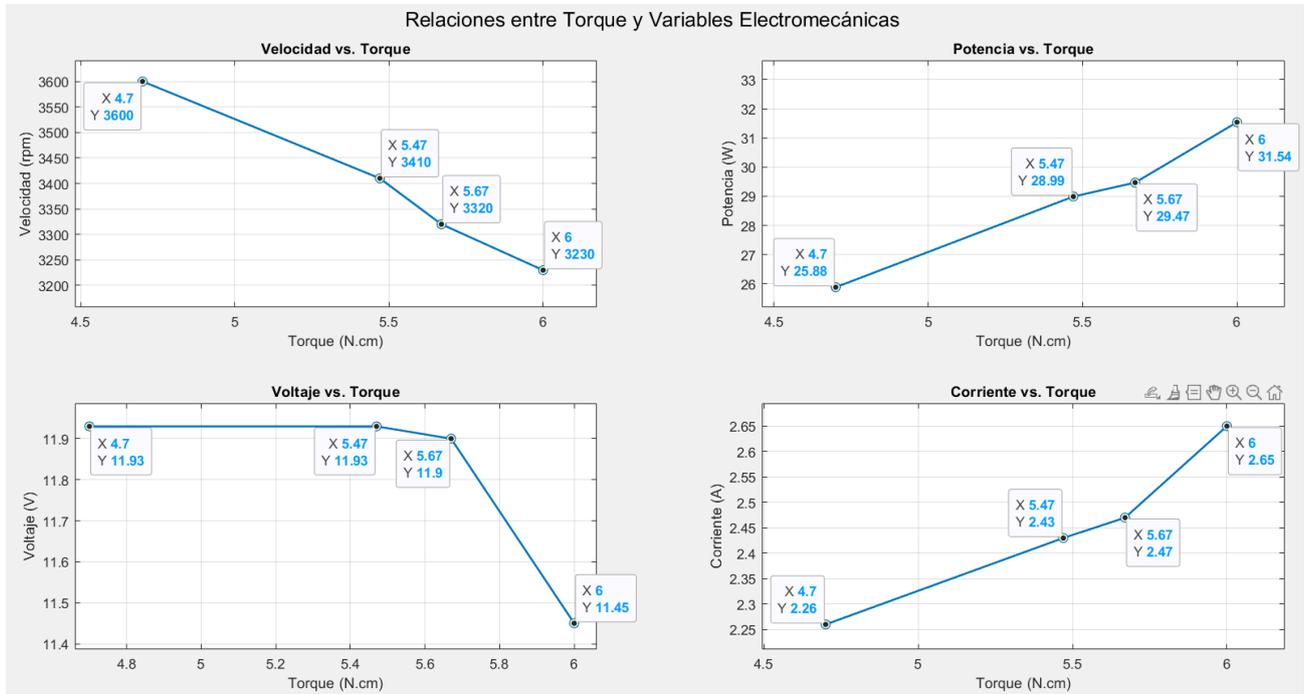


Figura 123. Velocidad, potencia, voltaje y corriente con referente al torque del motor del ventilador polea 16 – 30 dientes.

6.11 Guía de instalación del aplicativo

- **Instalación para dispositivos que no cuentan con Matlab.**

PASO 1:

Descargar el archivo de la ampliación a instalar que se encuentra en la carpeta drive en el siguiente enlace:

https://drive.google.com/drive/folders/1HP_njywC9MuaL5RMKF53kgMt8F3wsju0?usp=sharing

PASO 2

Abrir la carpeta nombrada for_redistribution, dar clip derecho y ejecutar como administrador al archivo denominado MyAppInstaller_mcr.

PASO 3

Aceptar los requisitos solicitados por el programa.

- **Instalación para dispositivos que cuentan con Matlab 2023.**

PASO 1:

Descargar el archivo de la ampliación a instalar que se encuentra en la carpeta drive en el siguiente enlace:

https://drive.google.com/drive/folders/1HP_njywC9MuaL5RMKF53kgMt8F3wsju0?usp=sharing

PASO 2

Verificar que el dispositivo cuente con Matlab versión 2023a.

PASO 3

Abrir la carpeta nombrada for_redistribution_files_only, dar clip derecho y ejecutar como administrador al archivo denominado Graficador de curvas características.

PASO 4

Aceptar los requisitos solicitados por el programa.

6.12 Guías prácticas

Después de llevar a cabo una investigación preliminar sobre los requisitos necesarios para la ejecución de las prácticas, se presenta las directrices para el desarrollo en el **Anexo 4**.

Se encuentra dividido en:

- **Práctica 1:** Sistema elevación de carga.
- **Práctica 2:** Sistema de bombeo
- **Práctica 3:** sistema de ventilación.

6.13 Manual de operación del banco didáctico experimental

1. Controles previos a la puesta en funcionamiento

Antes de operar el banco, es fundamental realizar las siguientes verificaciones:

- Verificar que todos los **componentes mecánicos estén firmemente sujetos** a la mesa de madera.
- Confirmar que el **sistema eléctrico** esté bien conectado, sin cables sueltos ni signos de cortocircuito.
- En el **sistema de bombeo**, revisar que el depósito contenga suficiente **refrigerante** y que no existan fugas.
- **Asegurarse de que no haya objetos extraños** dentro de los sistemas que puedan interferir con el movimiento.
- Girar manualmente cada mecanismo (bombeo, ventilación, elevación) para verificar un movimiento suave.
- **Revisar que las poleas estén bien ajustadas** al eje y no presenten juego o desplazamiento.

- **Comprobar que las poleas estén correctamente alineadas** entre sí, garantizando una transmisión eficiente y sin desvíos.
 - Verificar que los **sensores de torque, velocidad y adquisición de datos** estén conectados correctamente.
-

2. Operación durante el funcionamiento

Durante el uso del banco:

- **No forzar los mecanismos.** Si hay ruidos inusuales o resistencia, detener la operación inmediatamente.
 - **Evitar tocar** elementos en movimiento, como ventiladores, correas o poleas
 - **Evitar** acercar las manos al ventilador.
 - No **mover** las pesas de la grúa al estar en funcionamiento.
 - Detener el funcionamiento de todos los motores al momento de realizar cualquier variación en los elementos del banco (bridas, pesas y grados de apertura de lleve)
 - Utilizar la interfaz de monitoreo para observar en tiempo real las **curvas tacodinámicas** de cada sistema.
 - Alternar entre los sistemas de manera ordenada para evitar interferencias en los datos.
-

3. Procedimiento después de las prácticas

Una vez finalizada la sesión experimental:

- **Limpiar toda la superficie** del banco, retirando residuos y restos de refrigerante.
- Verificar visualmente el estado de cada sistema, en especial las poleas y correas.
- En el sistema de bombeo, limpiar el depósito y comprobar si necesita recarga o cambio de líquido.
- Apagar el software y **desconectar la alimentación eléctrica** de forma segura.

7. Discusión

El desarrollo del banco didáctico para el análisis del par resistente en motores se llevó a cabo en cuatro etapas: diseño, construcción, adquisición de datos y elaboración de guías prácticas.

El siguiente banco permite realizar 3 prácticas orientadas al sistema de ventilación, elevación y bombeo, para ello se permite regular la velocidad de funcionamiento y condiciones de carga. El sistema de adquisición de datos está integrado por un Arduino Mega, el cual posee una resolución de 10 bits, e incorpora sensores para la medición de corriente, voltaje, velocidad angular, presión y flujo de aire. De manera complementaria, se integra un sistema de control por medio de relés que permiten el encendido o apagado de los sistemas.

Para la construcción del banco didáctico, se fabricaron diversas piezas, incluyendo bases para grúa, motores, ducto de ventilación, entre otros. Estas piezas fueron elaboradas mediante impresión 3D, lo que permitió una fabricación adaptada a las especificaciones del diseño. Adicionalmente, para la creación de las pesas, se utilizó el proceso de fundición, garantizando elementos diseñados específicamente para las pruebas de carga. En cuanto a la base intermedia entre la mesa y la base de los motores, se empleó una fresadora CNC para asegurar acabados de alta precisión y un ensamblaje sólido y estable dentro de los sistemas.

En comparación con la tesis de Diez (2018), quien determinó experimentalmente la curva característica de un motor eléctrico con un objetivo similar al de este banco didáctico, los sistemas utilizados en su proyecto difieren de los empleados aquí, ya que su ensayo lo realizó con un motor accionado en vacío.

Por otro lado, con la revisión de bancos didácticos de las marcas Edibon e Insur, se pudo evidenciar que, a diferencia del banco desarrollado, estos requieren la adquisición de módulos separados para realizar pruebas con los tres tipos de sistemas (ventilación, elevación y bombeo). En comparación, el banco didáctico construido integra los tres sistemas en una sola unidad, lo que permite reducir los costos. Sin embargo, presenta una velocidad limitada en su sistema de adquisición de datos, lo que dificulta el registro preciso de las curvas características durante la etapa inicial de funcionamiento de las máquinas. Por su parte, los bancos comerciales cuentan con un sistema de adquisición más rápidos y robustos.

Tomando en cuenta los pros y los contras del banco se determina que es un prototipo que da una primera solución para la obtención de las curvas características de los sistemas frente a diferentes condiciones de trabajo.

8. Conclusiones

- Se logró construir y diseñar un banco didáctico experimental, que permite realizar un conjunto de prácticas en tres diferentes sistemas: elevación, bombeo y ventilación. Este banco está diseñado para medir voltaje, corriente, velocidad, flujo de aire y presión, con el objetivo de obtener las curvas características de los sistemas accionados que componen el banco.

- El banco permite obtener las curvas características de los sistemas, los datos obtenidos representan fielmente el comportamiento de las máquinas. El sistema de elevación permite ajustar las condiciones de operación como velocidad y carga en 4 niveles. El sistema de bombeo permite reconfigurar la velocidad de operación a través de las diferentes transmisiones permitidas y el flujo del agua a través de una compuerta. Por su parte, el sistema de ventilación permite interactuar con la velocidad a través de poleas dentadas y el control de diferentes niveles de obstrucción en la descarga.

- Se desarrollaron tres guías prácticas, una para cada sistema, dentro de ellas la primera guía, del sistema de elevación de carga, permite desarrollar 15 condiciones de operación distintas. Por otro lado, la guía general del sistema de bombeo permite obtener 12 configuraciones distintas de trabajo, finalmente, la guía general del sistema de ventilación, permite realizar 12 condiciones de prácticas diferentes. Todas las prácticas están basadas en el plan curricular del Modelo General de Evaluación de Carreras del CACES de la Universidad Nacional de Loja, lo que permite su replicabilidad en el laboratorio para estudiantes.

9. Recomendaciones

- Para futuras investigaciones se recomienda mejorar el sistema de medición y adquisición de datos, incorporando sensores y controladores más robustos, con mayor velocidad de adquisición. Debido, a que el sistema actual presenta una limitación de 8 bits, lo que representó un problema en la detección de los picos de corriente inicial.
- Aunque el banco actual permite realizar diversas pruebas con diferentes configuraciones, se recomienda ampliar el banco e incorporar una máquina de torque parabólico, propia de máquinas de arranque pesado. Lo que ampliaría las prestaciones del banco logrado.
- Para futuros trabajos se recomienda investigar sobre bancos certificados con el fin de desarrollar mejoras al presente banco y someterlo a un proceso de certificación.

10. Bibliografía

- (12V-24V) Motor RS775 de alto par y velocidad. (s. f.). J Lupin Electrónica. Recuperado 29 de enero de 2025, de <https://www.julpin.com.co/inicio/motores-dc-y-paso-a-paso/1374-12v-24v-motor-rs775-de-alto-par-y-velocidad.html>
- CASTAÑO GIRALDOC, S. A. (2019, marzo 3). ▷ Modelo de Motor DC - [diciembre, 2023]. Control Automático Educación. <https://controlautomaticoeducacion.com/analisis-de-sistemas/modelo-de-motor-dc/>
- China 12v 24v RS 775 Motor eléctrico de CC Imán permanente de alto par para automóviles Freno de mano, Taladro, Fabricantes de herramientas de jardín Fábrica de proveedores. (s. f.). Recuperado 29 de enero de 2025, de <https://es.songgangmotor.com/mini-dc-motor/12v-24v-rs-775-electric-dc-motor-high-torqu.html>
- Facultad de Energía y los Recursos Naturales no Renovables. (2023). DOCUMENTO DE GESTIÓN DEL LABORATORIO DE ENERGÍA Y FLUIDOS.
- Félix Monasterio-Huelin, Alvaro Gutiérrez y Blanca Larraga. (2023). Modelado de un motor DC.
- Llamas, J. (2023, febrero 9). *Matlab*. Economipedia. <https://economipedia.com/definiciones/tecnologia/matlab.html>
- Manuel Guillermo Quijano Ortega y Carlos Gerardo Hernández Capacho. (2009). Obtención experimental de los parámetros del motor que se utilizará en el sistema de localización de una esfera rodante.
- MATLAB App Designer*. (s. f.). Recuperado 29 de enero de 2025, de <https://la.mathworks.com/products/matlab/app-designer.html>
- MOTOR REDUCTOR 37D 12V DC 60RPM ZGA37RG* – Grupo Electrostore. (s. f.). Recuperado 29 de enero de 2025, de <https://grupoelectrostore.com/shop/motores/motor-reductor-37d-12v-dc-60rpm-zga37rg/>

MSc. Ing. Oscar Alexander Bellón Hernández. (2023). *DESCRIPCIÓN TEÓRICA DE UN PROCEDIMIENTO PARA DETERMINAR LOS PARÁMETROS DE UN MOTOR DE CORRIENTE CONTINUA*. 13.

Perdomo, S. (2020, julio 8). *Autodesk Inventor: Qué es y sus funciones* | Deusto Formación. Deusto. <https://www.deustoformacion.com/blog/diseño-arquitectónico/que-es-para-que-sirve-autodesk-inventor-diseño-producto>

Programación y scripts. (s. f.). Recuperado 29 de enero de 2025, de https://la.mathworks.com/help/matlab/learn_matlab/scripts.html

Qué es Arduino, cómo funciona y qué puedes hacer con uno. (s. f.). Recuperado 29 de enero de 2025, de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

¿Qué es AutoCAD y para qué sirve? (2020, junio 16). Arcux. <https://arcux.net/blog/que-es-autocad-y-para-que-sirve/>

Relé con Arduino y ESP8266 para crear una lámpara inteligente. (2020, octubre 14). <https://programarfacil.com/blog/arduino-blog/rele-con-arduino-lampara/>

SENSOR INFRARROJO FOTOELÉCTRICO DE VELOCIDAD H ENCODER – Grupo Electrostore. (s. f.). Recuperado 29 de enero de 2025, de <https://grupoelectrostore.com/shop/sensores/infrarrojos/sensor-infrarrojo-fotoelectrico-de-velocidad-h-encoder/>

Simulación y diseño basado en modelos con Simulink. (s. f.). Recuperado 29 de enero de 2025, de <https://la.mathworks.com/products/simulink.html>

SL, U. T. (s. f.). *Ultimaker Cura (Windows)*. Uptodown. Recuperado 29 de enero de 2025, de <https://ultimaker-cura.uptodown.com/windows>

11. Anexos

Anexo 1. Fichas técnicas.



RS-755VC/WC

MABUCHI MOTOR

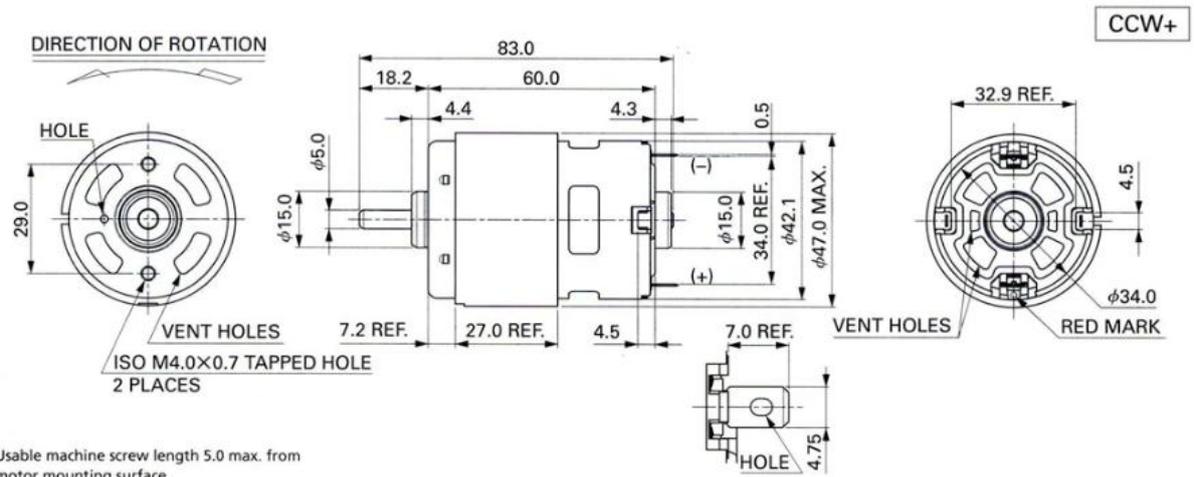
Carbon-brush motors

OUTPUT : 8.0W~160W (APPROX)

WEIGHT : 286g (APPROX)

Typical Applications Cordless Power Tools : Drill / Cordless Garden Tool / Circular Saw
Office Automation Equipment : Printer / Copy Machine

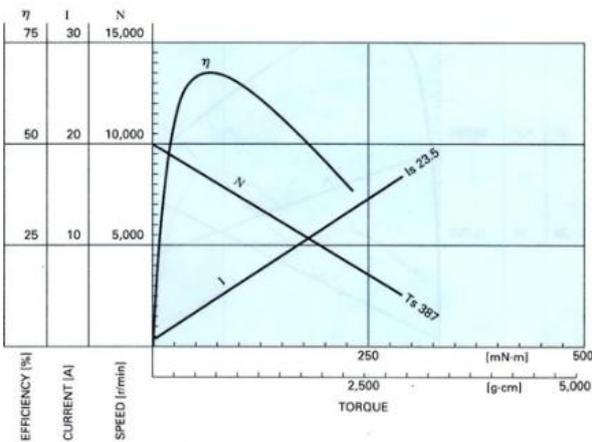
MODEL		VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY				STALL			
		OPERATING RANGE	NOMINAL	SPEED r/min	CURRENT A	SPEED r/min	CURRENT A	TORQUE mN·m	g·cm	OUTPUT W	TORQUE mN·m	g·cm	CURRENT A
RS-755VC-4540	(*1)	9~36	18V CONSTANT	10000	0.85	8400	4.47	61.8	630	54.3	387	3945	23.5
RS-755VC-8016	(*1)	6.0~14.4	14.4V CONSTANT	20500	2.40	17730	15.3	94.2	960	175	696	7095	98.0
RS-755VC-24120	(*2)	12~27	24V CONSTANT	3500	0.22	2840	0.95	45.9	468	13.6	244	2487	4.10
RS-755WC-8017	(*3)	6~20	18V CONSTANT	19800	2.30	17320	16.0	126	1289	229	1009	10285	112



Usable machine screw length 5.0 max. from motor mounting surface.

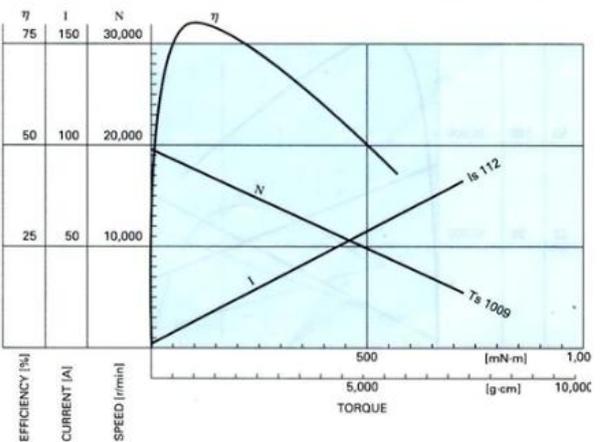
RS-755VC-4540

18.0V



RS-755WC-8017

18.0V



Fuente: (Mabuchi motor, s.f)

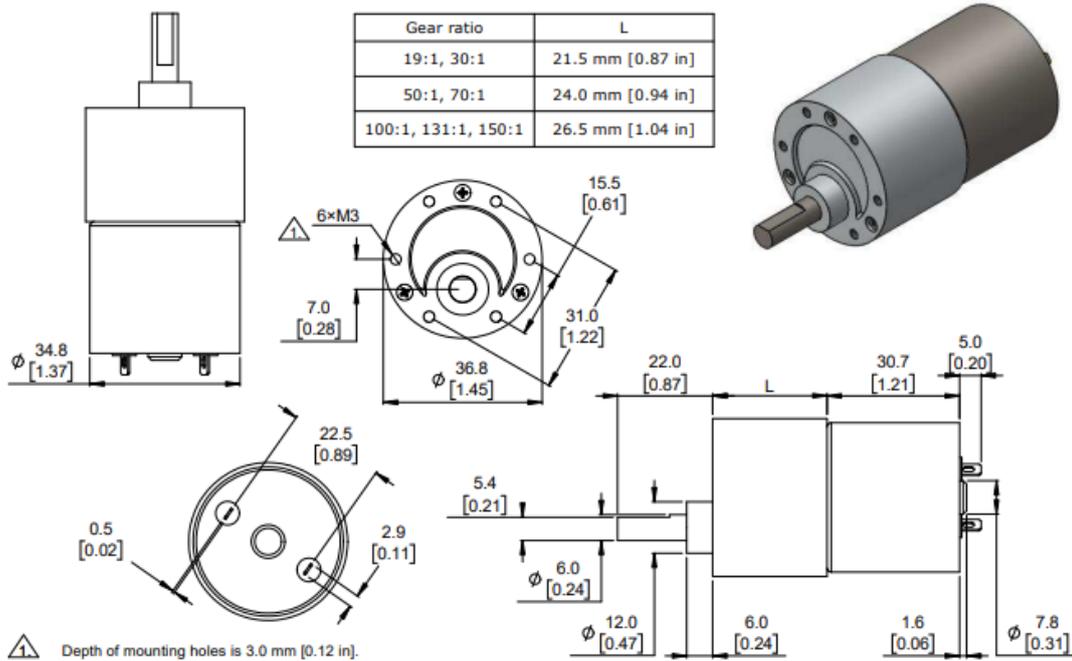
Performance summary and table of contents

Rated Voltage	Pololu Item #	Gear Ratio	No Load		At Maximum Efficiency				Max Power	Stall Extrapolation		Graph Page	
			Speed	Current	Speed	Torque	Current	Output		Torque	Current	6 V	12 V
			:1	RPM	A	RPM	kg-mm	A		W	W	kg-mm	A
12 V	2821, 4750	1	10,000	0.15						5			
	1102, 2822 4741, 4751	18.75	540		470	10	0.76	5.0	12	85		5	6
	1103, 2823 4742, 4752	30	330		280	18	0.78	5.1	12	140		7	8
	1104, 2824 4743, 4753	50	200		180	22	0.66	4.0	10	210		9	10
	1105, 2825 4744, 4754	70	150		130	32	0.68	4.2	10	270		11	12
	1106, 2826 4745, 4755	102.08	100		87	42	0.72	3.8	8	340		13	14
	1107, 2827 4746, 4756	131.25	76		66	60	0.74	4.1	6	450		15	16
	2828, 2829	150	67		58	65	0.72	3.8	6	490		17	18

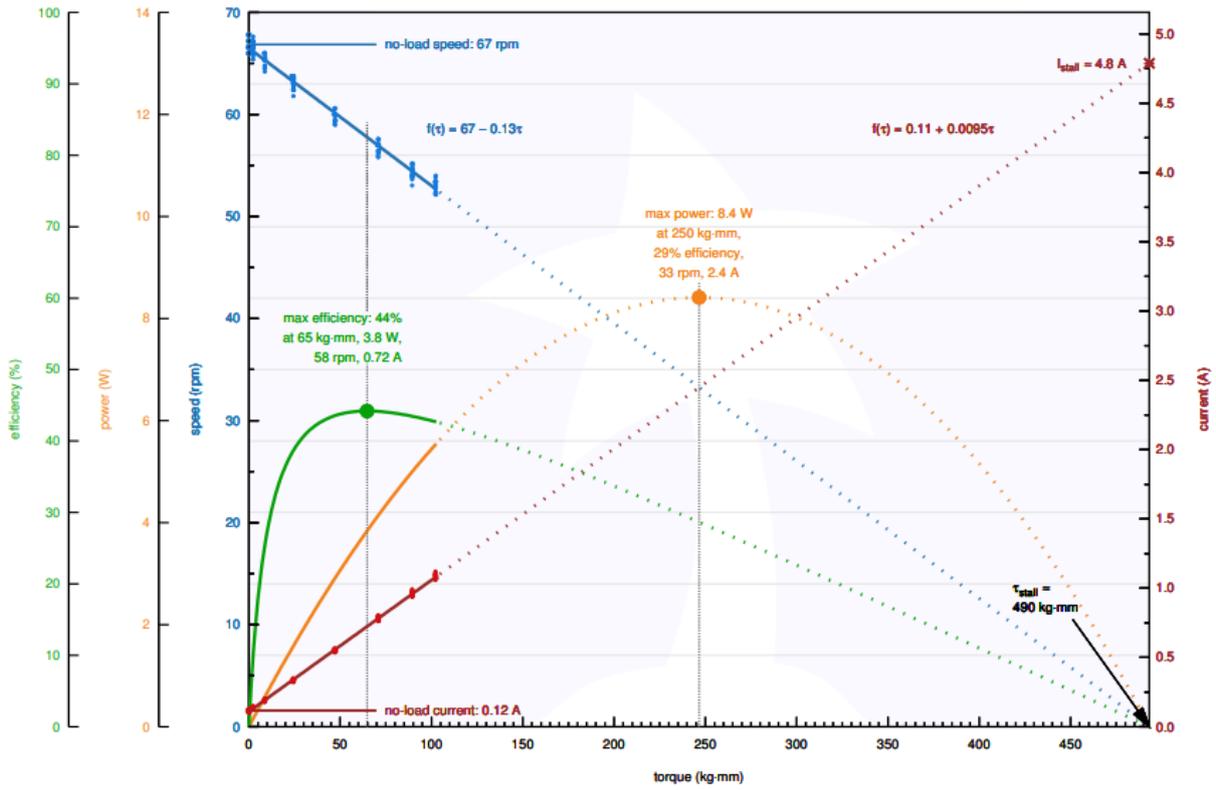
Dimensions (units: mm over [inches])

Gearmotor versions without encoders (items #1102-1107, 2829, 4741-4746)

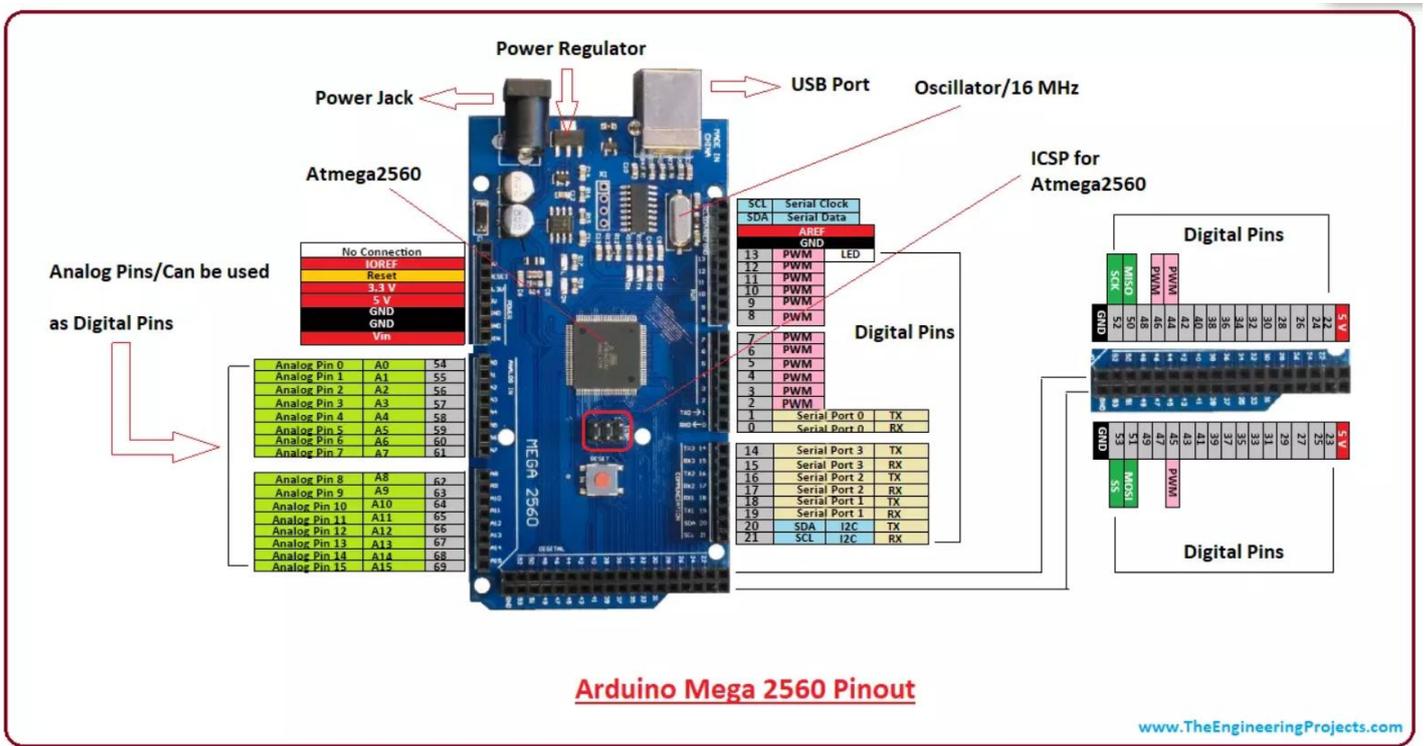
weight: 185 g to 195 g



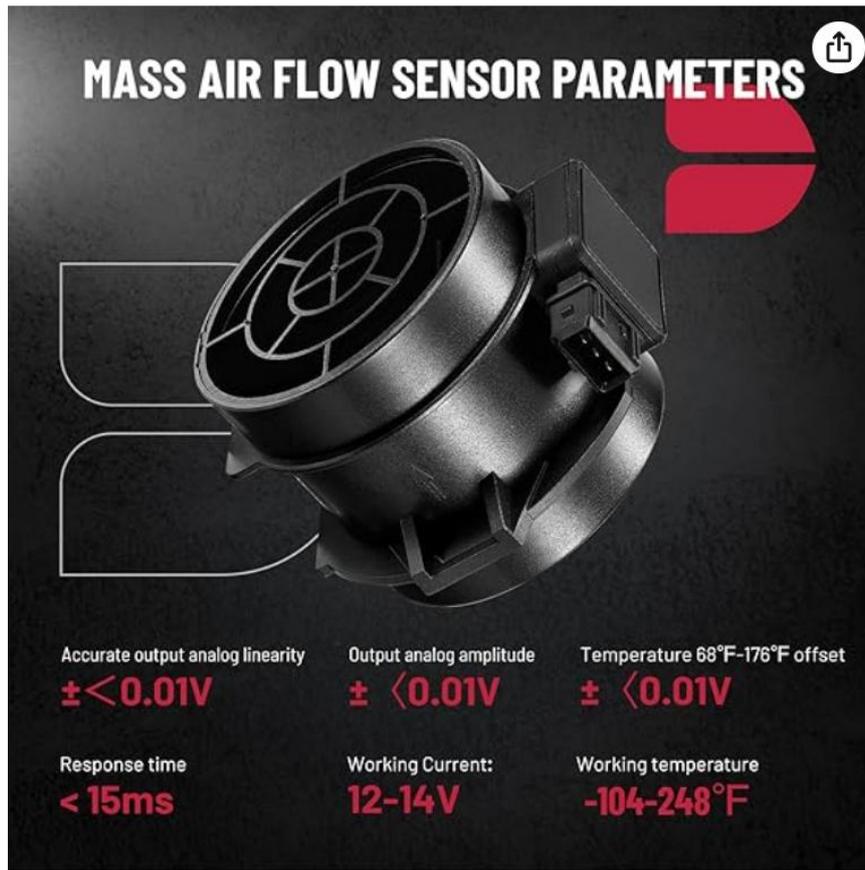
Pololu Items #2829, #2828, #150:1 Metal Gearmotor 37D , # () Performance at 12 V



Fuente: (Pololu, s.f)



Fuente: (Proyecto Aduino, s.f)



Fuente: (Amazon, s.f)

Especificaciones técnicas

Marca	BDFHYK
Material	Metal Plástico
Estilo	Sensor de aire de masa
Rango de medición:	up to 1000 SCFM
Tipo de montaje	Montaje en Brida
Tipo de salida	PNP, Push-pull, NPN
Usos específicos del producto	Automotriz, específicamente para motores BMW
Clasificación de temperatura superior	1.2E+2 Grados Celsius
Fabricante	BDFHYK
Dimensiones del paquete	5,35 x 4,88 x 4,57 pulgadas
País de origen	China
Número de modelo del producto	7410055
Número de pieza del fabricante	BD-05-07034BK

Fuente: (Amazon, s.f)

Sensor Detector Óptico Infrarrojo lector encoder FC-03

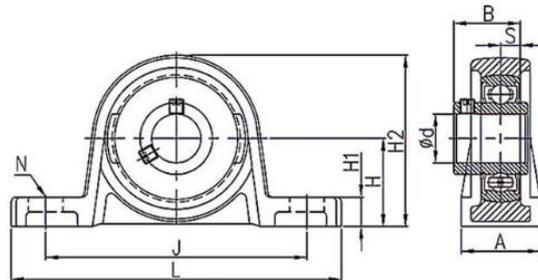


FC03 - emisor fotodiodo IR - detector fototransistor	
Tensión de trabajo	3,3 a 5V
Basado en comparador LM393	
Salida digital	ON/OFF
Salida analógica	0 - VCC
Led indicador de alimentación y detección	
Aplicación: equipos robótica, consolas	

Fuente: (FREE ELECTRON, s.f)



KP Pillow Block Bearing



RONICAL

Bearing Unit Model No.	Bore Diameter	Dimension (mm)											Bolt
		KP	d	H	L	J	A	N	H1	H2	B	S	
KP08	8	15	55	42	12	4.5	5	29	-	-	-	-	M4
KP000	10	18	67	53	16	7	6	35	14	4	5.56	33	M6
KP001	12	19	71	56	16	7	6	38	14.5	4	5	33	M6
KP002	15	22	80	63	16	7	7	43	16.5	4.5	4	34	M6
KP003	17	24	85	67	18	7	7	47	17.5	5	5.5	38	M6
KP004	20	28	100	80	20	10	9	55	21	6	7	46	M8
KP005	25	32	112	90	20	10	10	62	22.5	6	6	47	M8
KP006	30	36	132	106	26	13	11	70	24.5	6.5	6	50	M10
KP007	35	42	150	118	33	13	14	76	33	8	8	55	M12

Fuente: (Ronical, s.f)



Pump Information

Suction	Maximum / 1.5 m
Head	About 4-5 m
Greatest pressure	0.48MPa
Maximum flow	3.5L/min
Operating Voltage	12V DC
Maximum current	2A

Fuente: (Amazon, s.f)

Teeth Type: 2GT 30 Teeth



Fuente: (Amazon, s.f)



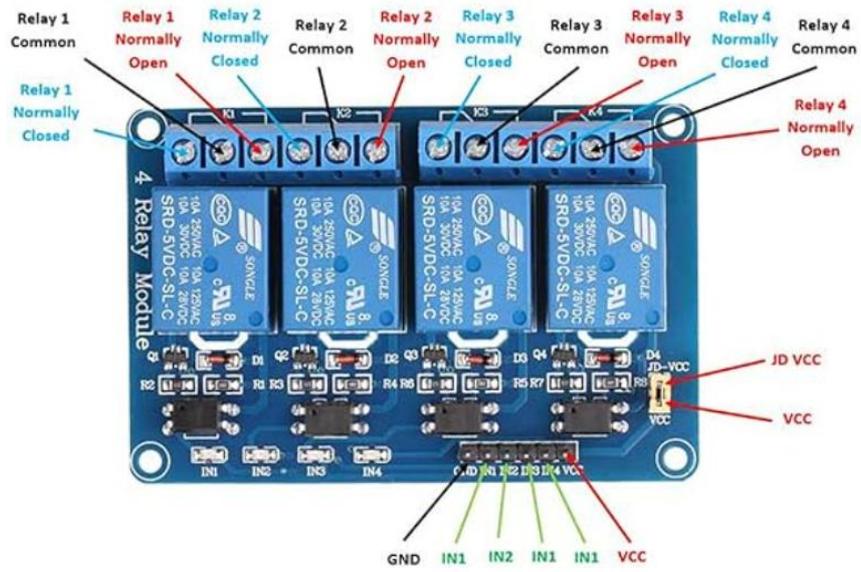
Fuente: (Amazon, s.f)



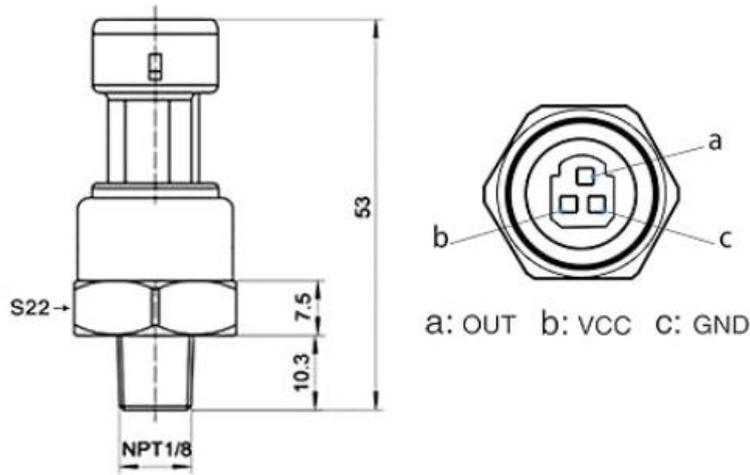
Fuente: (Amazon, s.f)



Fuente: (Amazon, s.f)



Fuente: (Amazon, s.f)

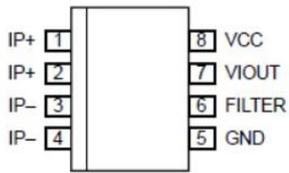


a: OUT b: VCC c: GND

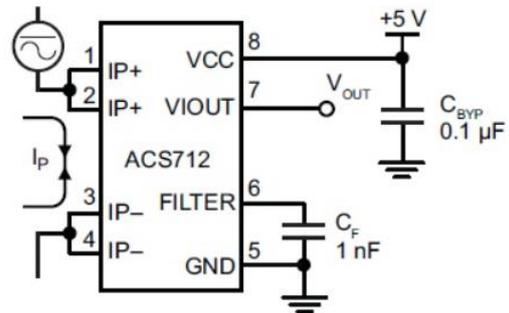
Accuracy:	±1.0%FS	Pressure range:	0-100 PSI
Working Power:	5V DC	Output signal:	0.5-4.5V
Operating temperature:	-40 ~ 105°C	Compensation temperature:	-20 ~ 80°C
Operating current:	≤3mA	Pressure connection:	1/8" -27NPT

Fuente: (Amazon, s.f)

Pin-out Diagram



Typical Application



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sampled; fused internally
3 and 4	IP-	Terminals for current being sampled; fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VIOUT	Analog output signal
8	VCC	Device power supply terminal

Fuente: (NAFCOM, s.f)

Anexo 2. Facturas.



ROBOTICS ECUADOR

SIMON ERNESTO RIOS HIDALGO

Dirección Matriz: VENEZUELA N11 - 249 ENTRE CALDAS Y MATOVELLE
 Dirección Sucursal: VENEZUELA N11 249 ENTRE CALDAS Y

Obligado a llevar Contabilidad NO
 CONTRIBUYENTE RÉGIMEN RIMPE

Ruc: 1722646682001

FACTURA

No: 001-010-000008195

1901202401172264668200120010100000081950000000011

Ambiente: producción

CLAVE DE ACCESO



1901202401172264668200120010100000081950000000011

Razón Social: AGUILAR LOAIZA MARCO DANILO

Fecha Emisión: 19/01/2024

Identificación: 0705642726

Dirección: PIÑAS

Código	Descripción	Cant	P.unit	% IVA	Subsidio	Desc	Total
8A1087	CAPACITOR CERAMICO 1nF 102	2	0.04	12	0	0.00	0.09
8A1095	CAPACITOR CERAMICO 100nF 104	2	0.04	12	0	0.00	0.09
8A1043	CAPACITOR ELECTROLITICO 1uF 50V	4	0.09	12	0	0.00	0.36
3C224	CORREA DENTADA GT2 CERRADA 200mm	1	2.46	12	0	0.00	2.46
3C212	POLEA GT2 36 DIENTES DIAMETRO 5mm	2	5.80	12	0	0.00	11.61

Información Adicional

correo: marco.aguilar@unl.edu.ec
 Régimen: Contribuyente Regimen RIMPE

FORMA DE PAGO

Nombre	Valor
SIN UTILIZACION DEL SISTEMA FINANCIERO	16.35

Subtotal 0%	0
Subtotal 12	14.60
Descuento	0.00
ICE	0
IVA 12	1.75
TOTAL	16.35
Total sin subsidio	0
Ahorro por subsidio	0

Otros Datos

*Tipo: Contribuyente Regimen RIMPE



ROBOTICS ECUADOR

SIMON ERNESTO RIOS HIDALGO

Dirección Matriz: VENEZUELA N11 - 249 ENTRE CALDAS Y MATOVELLE

Dirección Sucursal: VENEZUELA N11 249 ENTRE CALDAS Y

Obligado a llevar Contabilidad NO
CONTRIBUYENTE RÉGIMEN RIMPE

Ruc: 1722646682001

FACTURA

No: 001-010-000007975

1412202301172264668200120010100000079750000000016

Ambiente: producción

CLAVE DE ACCESO



1412202301172264668200120010100000079750000000016

Razón Social: AGUILAR LOAIZA MARCO DANILLO

Fecha Emisión: 14/12/2023

Identificación: 0705642726

Dirección: PIÑAS

Código	Descripción	Cant	P.unit	% IVA	Subsidio	Desc	Total
3C210	POLEA GT2 16 DIENTES DIAMETRO 5mm	1	2.59	12	0	0.00	2.59
3C211	POLEA GT2 20 DIENTES DIAMETRO 5mm	2	2.59	12	0	0.00	5.18
3C216	POLEA GT2 20 DIENTES DIAMETRO 8mm	2	2.59	12	0	0.00	5.18
3C904	KP08 DIAMETRO 8mm	4	2.68	12	0	0.00	10.71
3C225	CORREA DENTADA GT2 CERRADA 280mm	1	3.13	12	0	0.00	3.13
3C212	POLEA GT2 36 DIENTES DIAMETRO 5mm	1	5.80	12	0	0.00	5.80
3C213	POLEA GT2 40 DIENTES DIAMETRO 5mm	1	6.70	12	0	0.00	6.70

Información Adicional

correo:
marco.aguilard@uni.edu.ec
Régimen:
Contribuyente Regimen RIMPE

FORMA DE PAGO

Nombre	Valor
SIN UTILIZACION DEL SISTEMA FINANCIERO	44.00

Subtotal 0%	0
Subtotal 12	39.29
Descuento	0.00
ICE	0
IVA 12	4.71
TOTAL	44.00
Total sin subsidio	0
Ahorro por subsidio	0

Otros Datos

*Tipo: Contribuyente Regimen RIMPE



ROBOTICS ECUADOR

SIMON ERNESTO RIOS HIDALGO

Dirección Matriz: VENEZUELA N11 - 249 ENTRE CALDAS Y MATOVELLE
 Dirección Sucursal: VENEZUELA N11 249 ENTRE CALDAS Y

Obligado a llevar Contabilidad NO
 CONTRIBUYENTE RÉGIMEN RIMPE

Ruc: 1722646682001

FACTURA

No: 001-010-000007922

0612202301172264668200120010100000079220000000012

Ambiente: producción

CLAVE DE ACCESO



0612202301172264668200120010100000079220000000012

Razón Social: AGUILAR LOAIZA MARCO DANILO

Fecha Emisión: 06/12/2023

Identificación: 0705642726

Dirección: PIÑAS

Código	Descripción	Cant	P.unit	% IVA	Subsidio	Desc	Total
1A801	TUBO TERMOENCOGIBLE 2.0mm NEGRO (PRECIO POR METRO)	2	0.18	12	0	0.00	0.36
1A802	TUBO TERMOENCOGIBLE 2.5mm NEGRO (PRECIO POR METRO)	4	0.18	12	0	0.00	0.71
1A803	TUBO TERMOENCOGIBLE 3.0mm NEGRO (PRECIO POR METRO)	2	0.22	12	0	0.00	0.45
1A804	TUBO TERMOENCOGIBLE 3.0mm ROJO (PRECIO POR METRO)	2	0.22	12	0	0.00	0.45
1A805	TUBO TERMOENCOGIBLE 5.0mm NEGRO (PRECIO POR METRO)	2	0.45	12	0	0.00	0.89
5A221	SOPORTE DE PLASTICO PARA SENSOR ULTRASONICO HC-SR04	1	0.89	12	0	0.00	0.89
5A217	SENSOR ULTRASONICO HC-SR04	1	2.23	12	0	0.00	2.23
3C215	POLEA GT2 20 DIENTES DIAMETRO 6.35mm	1	2.68	12	0	0.00	2.68
3C906	KP001 DIAMETRO 12mm	5	6.03	12	0	0.00	30.13
2A1603	MINI BOMBA DE AGUA SUMERGIBLE 12VDC AD20P-1230E	1	10.00	12	0	0.00	10.00
3C711	EJE DIAMETRO 12mm LARGO 800mm	1	13.84	12	0	0.00	13.84
2A1604	BOMBA DE AGUA DE DIAFRAGMA 12VDC DP-521	1	26.79	12	0	0.00	26.79
3C222	CORREA DENTADA GT2 ANCHO 6mm (PRECIO POR METRO)	2	2.68	12	0	0.00	5.36

Información Adicional

correo: marco.aguiar@uni.edu.ec
 Régimen: Contribuyente Regimen RIMPE

FORMA DE PAGO

Nombre	Valor
SIN UTILIZACION DEL SISTEMA FINANCIERO	106.15

Subtotal 0%	0
Subtotal 12	94.78
Descuento	0.00
ICE	0
IVA 12	11.37
TOTAL	106.15
Total sin subsidio	0
Ahorro por subsidio	0

Otros Datos

*Tipo: Contribuyente Regimen RIMPE

Anexo 3. Certificado de la traducción del resumen.

Lic. Andrea Sthefanía Carrión Mgs

0984079037

andrea.s.carrion@unl.edu.ec

Loja-Ecuador

Loja, 31 de enero del 2025

La suscrita, Andrea Sthefanía Carrión Fernández, Mgs, **DOCENTE EDUCACIÓN SUPERIOR** (registro de la SENESCYT número: 1008-12-1124463), **ÁREA DE INGLÉS-UNIVERSIDAD NACIONAL DE LOJA**, a petición de la parte interesada y en forma legal.

CERTIFICA:

Que la traducción del resumen del documento adjunto, solicitado por el señor: **Marco Danilo Aguilar Loaiza** con cédula de ciudadanía **No. 0705642726**, cuyo tema de investigación se titula: **“Diseño y construcción de un banco didáctico para el análisis del par resistente de máquinas”** ha sido realizado y aprobado por mi persona, Andrea Sthefanía Carrión Fernández, Mgs. en Pedagogía.

El apartado del Abstract es una traducción textual del Resumen aprobado en español.

Particular que comunico en honor a la verdad para los fines académicos pertinentes, facultando al portador del presente documento, hacer el uso legal pertinente.

ANDREA
STHEFANIA
CARRION
FERNANDEZ

Firmado digitalmente
por ANDREA STHEFANIA
CARRION FERNANDEZ
Fecha: 2025.01.31
18:36:45 -06'00'

Andrea Sthefanía Carrión Fernández. Mgs.

English Professor

Anexo 4. Archivos de respaldo.

A continuación, se presenta un código QR que dirige a una carpeta en Google Drive, la cual contiene los diferentes archivos esenciales para el correcto funcionamiento del banco de didáctico, tales como planos, diseños, programas y documentación técnica.



Anexo 5. Guías prácticas.

Práctica # 1

Asignatura: -----

Autores: -----

Número de estudiantes por grupo: 4 estudiantes

1. Tema: Análisis del par resistente del sistema de grúa.

2. Objetivos:

- Analizar y explicar gráficos que representan la variación del torque en función de la velocidad.
- Comparar y contrastar cómo diferentes tipos de máquinas responden a cambios en la velocidad de funcionamiento.
- Desarrollar estrategias para optimizar el rendimiento de las máquinas al ajustar la velocidad de funcionamiento para alcanzar niveles de torque específicos.

3. Materiales y reactivos – Por Grupo	4. Equipos y herramientas – Por Grupo
•	•

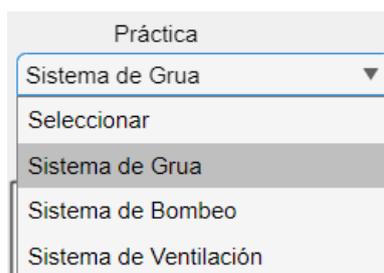
5. Instrucciones:

- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos
- Uso obligatorio del mandil

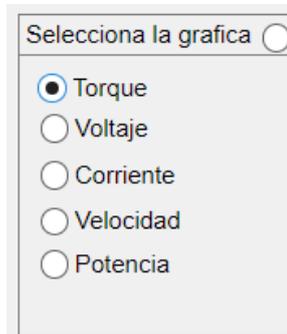
6. Actividades por desarrollar:

Para el desarrollo de la presente práctica se debe hacer lo siguiente:

1. Se debe seleccionar el tipo de práctica a realizar (Sistema de Grúa) cómo se observa en la siguiente figura.



2. Seleccionar la gráfica que se desea observar al iniciar la práctica (una vez iniciada la práctica se puede cambiar la gráfica), como se muestra en la siguiente figura.



Selecciona la grafica

Torque

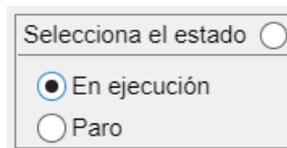
Voltaje

Corriente

Velocidad

Potencia

3. Para que la práctica pueda iniciar debe estar seleccionado en estado ejecución, como se muestra en la siguiente figura.



Selecciona el estado

En ejecución

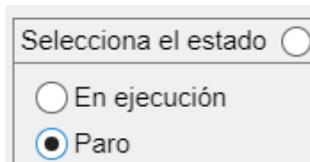
Paro

4. Pulsar el botón de iniciar práctica, como se muestra en la siguiente figura.



Iniciar Práctica

5. Es necesario detener la práctica antes de que la pesa alcance el punto máximo permitido en la grúa. Esto asegura que los datos recopilados no se vean afectados por distorsiones al accionar el resorte incorporado en el interruptor del final de carrera. Para lograrlo, se debe presionar el botón de parada, tal como se muestra en la figura siguiente.



Selecciona el estado

En ejecución

Paro

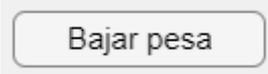
6. Si la práctica se realizó con éxito, se debe asignar un nombre al archivo de Excel donde se almacenarán todos los valores obtenidos. Luego, se debe presionar el botón (Guardar datos), como se muestra en la figura siguiente.



Nombre del archivo:

Los valores obtenidos en esta práctica se organizan en el archivo Excel en el siguiente orden, correspondiente a las diferentes columnas: corriente (A), voltaje (V), velocidad (rpm), torque (Nm), potencia (W) y tiempo (s).

7. Una vez que los datos han sido guardados, se debe pulsar el botón 'Bajar pesa'. Esta acción hará que el motorreductor gire en sentido contrario, permitiendo bajar la pesa hasta el punto adecuado para elevar la siguiente carga. Cuando la pesa llegue a la posición establecida, será necesario volver a presionar el botón para detener el motor. El botón correspondiente se muestra en la siguiente imagen.

A rectangular button with rounded corners and a thin border, containing the text 'Bajar pesa' in a sans-serif font.

8. Repetir este mismo proceso para los diferentes pesos a levantar y para las diferentes velocidades de la grúa (Intercambiar las poleas).
9. Una vez recopilados todos los valores correspondientes a las diferentes circunstancias de peso y relación de transmisión, se calculan los promedios y se procede a obtener las curvas características del sistema bajo las diferentes condiciones de funcionamiento.

7. Marco teórico: (a elaborar por el estudiante)

- Torque
- Tipos de torque
- Funcionamiento de una grúa.

8. Resultados obtenidos (a elaborar por el estudiante)

9. Discusión (a elaborar por el estudiante)

10. Conclusiones (a elaborar por el estudiante)

11. Recomendaciones (elaborar por el estudiante)

12. Preguntas de control (deben ser respondidas por el estudiante)

- ¿Qué sucede con el voltaje, corriente y torque al incrementar la velocidad de la máquina?
- ¿Qué sucede con el voltaje, corriente y torque al incrementar el peso a ser elevado por la máquina?

13. BIBLIOGRAFÍA:

Práctica # 2

Asignatura: -----

Autores: -----

Número de estudiantes por grupo: 4 estudiantes

1. Tema: Análisis del par resistente del sistema de bombeo.

2. Objetivos:

- Analizar y explicar gráficos que representan la variación del torque en función de la velocidad.
- Comparar y contrastar cómo diferentes tipos de máquinas responden a cambios en la velocidad de funcionamiento.
- Desarrollar estrategias para optimizar el rendimiento de las máquinas al ajustar la velocidad de funcionamiento para alcanzar niveles de torque específicos.

3. Materiales y reactivos – Por Grupo	4. Equipos y herramientas – Por Grupo
•	•

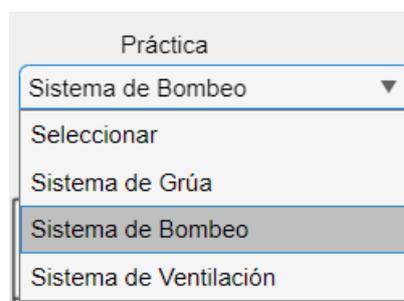
5. Instrucciones:

- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos
- Uso obligatorio del mandil

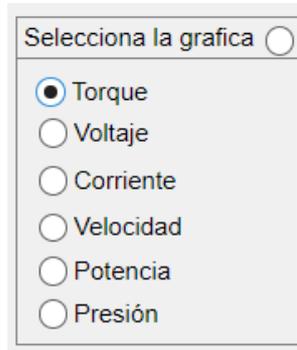
6. Actividades por desarrollar:

Para el desarrollo de la presente práctica se debe hacer lo siguiente:

1. Se debe seleccionar el tipo de práctica a realizar (Sistema de Bombeo) como se observa en la siguiente figura.



2. Seleccionar la gráfica que se desea observar al iniciar la práctica (una vez iniciada la práctica se puede cambiar la gráfica), como se muestra en la siguiente figura.



Selecciona la grafica

Torque

Voltaje

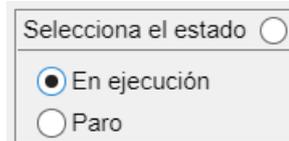
Corriente

Velocidad

Potencia

Presión

3. Para que la práctica pueda iniciar debe estar seleccionado en estado ejecución, como se muestra en la siguiente figura.



Selecciona el estado

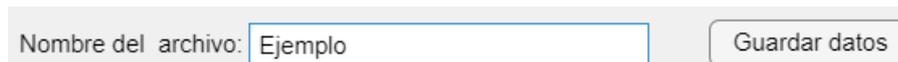
En ejecución

Paro

4. Pulsar el botón de iniciar práctica, como se muestra en la siguiente figura.

Iniciar Práctica

5. Si se desea parar la práctica se puede seleccionar el estado paro del paso 3.
6. Si la práctica fue un éxito, se debe colocar un nombre al archivo de Excel donde se van a guardar todos los valores obtenidos en esta práctica al presionar el botón guardar datos, como se muestra en la siguiente figura.



Nombre del archivo:

Los valores obtenidos en esta práctica se organizan en el archivo Excel en el siguiente orden, correspondiente a las diferentes columnas: corriente (A), voltaje (V), velocidad (rpm), torque (Nm), potencia (W), presión (PSI) y tiempo (s).

7. Repetir este mismo proceso para las diferentes velocidades de la bomba (Intercambiar las poleas) y para los diferentes grados de apertura de la llave.
8. Una vez recopilados todos los valores correspondientes a las diferentes circunstancias de apertura y relación de transmisión, se calculan los promedios y se procede a obtener las curvas características del sistema bajo las diferentes condiciones de funcionamiento.

7. Marco teórico: (a elaborar por el estudiante)

- Torque
- Tipos de torque
- Funcionamiento de una bomba de diafragma.
- Funcionamiento de un sistema de bombeo.

8. Resultados obtenidos (a elaborar por el estudiante)

9. Discusión (a elaborar por el estudiante)

10. Conclusiones (a elaborar por el estudiante)

11. Recomendaciones (elaborar por el estudiante)

12. Preguntas de control (deben ser respondidas por el estudiante)

- ¿Qué sucede con el voltaje, corriente y torque al incrementar la velocidad de la máquina?
- ¿Porque el sensor de presión no da una presión constante en sus mediciones?

13. BIBLIOGRAFÍA:

Práctica # 3

Asignatura: -----

Autores: -----

Número de estudiantes por grupo: 4 estudiantes

1. Tema: Análisis del par resistente del sistema de ventilación.

2. Objetivos:

- Analizar y explicar gráficos que representan la variación del torque en función de la velocidad.
- Comparar y contrastar cómo diferentes tipos de máquinas responden a cambios en la velocidad de funcionamiento.
- Desarrollar estrategias para optimizar el rendimiento de las máquinas al ajustar la velocidad de funcionamiento para alcanzar niveles de torque específicos.

3. Materiales y reactivos – Por Grupo	4. Equipos y herramientas – Por Grupo
•	•

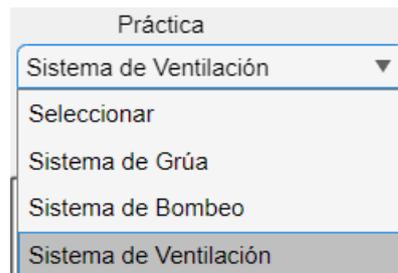
5. Instrucciones:

- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos
- Uso obligatorio del mandil

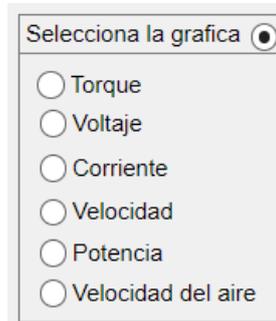
6. Actividades por desarrollar:

Para el desarrollo de la presente práctica se debe hacer lo siguiente:

1. Se debe seleccionar el tipo de práctica a realizar (Sistema de Ventilación) como se observa en la siguiente figura.



2. Seleccionar la gráfica que se desea observar al iniciar la práctica (una vez iniciada la práctica se puede cambiar la gráfica), como se muestra en la siguiente figura.



Selecciona la grafica

Torque

Voltaje

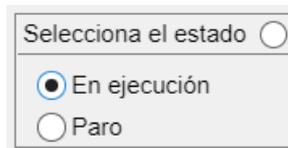
Corriente

Velocidad

Potencia

Velocidad del aire

3. Para que la práctica pueda iniciar debe estar seleccionado en estado ejecución, como se muestra en la siguiente figura.



Selecciona el estado

En ejecución

Paro

4. Pulsar el botón de iniciar práctica, como se muestra en la siguiente figura.

Iniciar Práctica

5. Si se desea parar la práctica se puede seleccionar el estado paro del paso 3.
6. Si la práctica fue un éxito, se debe colocar un nombre al archivo de Excel donde se van a guardar todos los valores obtenidos en esta práctica al presionar el botón guardar datos, como se muestra en la siguiente figura.



Nombre del archivo:

Los valores obtenidos en esta práctica se organizan en el archivo Excel en el siguiente orden, correspondiente a las diferentes columnas: corriente (A), voltaje (V), velocidad (rpm), torque (Nm), potencia (W), Velocidad del aire y tiempo (s).

7. Repetir este mismo proceso para las diferentes velocidades del sistema de ventilación (Intercambiar las poleas) y para las diferentes obstrucciones.
8. Una vez recopilados todos los valores correspondientes a las diferentes circunstancias de obstrucción y relación de transmisión, se calculan los promedios y se procede a obtener las curvas características del sistema bajo las diferentes condiciones de funcionamiento.

7. Marco teórico: (a elaborar por el estudiante)

- Torque
- Tipos de torque
- Funcionamiento de ventiladores.
- Sistema de hilo caliente para medir flujo de aire.

8. Resultados obtenidos (a elaborar por el estudiante)

9. Discusión (a elaborar por el estudiante)

10. Conclusiones (a elaborar por el estudiante)

11. Recomendaciones (elaborar por el estudiante)

12. Preguntas de control (deben ser respondidas por el estudiante)

- ¿Qué sucede con el voltaje, corriente y torque al incrementar la velocidad de las máquinas?
- ¿ Como es el comportamiento de las lecturas del sensor MAF ante las diferentes obstrucciones y transmisiones de velocidad?

13. BIBLIOGRAFÍA:

Anexo 6. Códigos

Código Arduino:

```
classdef final2 < matlab.apps.AppBase
```

```
    properties (Access = public)
```

```
        UIFigure          matlab.ui.Figure
```

```
        ParodeemergenciaButton    matlab.ui.control.Button
```

```
        Image2            matlab.ui.control.Image
```

```
        Image            matlab.ui.control.Image
```

```
        BajarpesaButton    matlab.ui.control.Button
```

```
        NombredelarchivoEditField    matlab.ui.control.EditField
```

```
        NombredelarchivoLabel    matlab.ui.control.Label
```

```
        GuardardatosButton    matlab.ui.control.Button
```

```
        SeleccionaelestadoButtonGroup    matlab.ui.container.ButtonGroup
```

```
        EnejecucinButton_2    matlab.ui.control.RadioButton
```

```
        ParoButton        matlab.ui.control.RadioButton
```

```
        Button_2        matlab.ui.control.RadioButton
```

```
        PuertodeconexinEditField    matlab.ui.control.EditField
```

```
        PuertodeconeccionLabel    matlab.ui.control.Label
```

```
        IniciarPrcticaButton    matlab.ui.control.Button
```

```
        SeleccionalagrificaButtonGroup    matlab.ui.container.ButtonGroup
```

```
        VelocidaddelaireButton    matlab.ui.control.RadioButton
```

```
        PresinButton_2    matlab.ui.control.RadioButton
```

```
        PotenciaButton    matlab.ui.control.RadioButton
```

```
        TorqueButton_2    matlab.ui.control.RadioButton
```

```
        VelocidadButton    matlab.ui.control.RadioButton
```

```
        CorrienteButton    matlab.ui.control.RadioButton
```

```
        VoltajeButton    matlab.ui.control.RadioButton
```

```
        Button        matlab.ui.control.RadioButton
```

```
        PrcticaDropDown    matlab.ui.control.DropDown
```

```
        PesoalevantarLabel    matlab.ui.control.Label
```

```
        UIAxes        matlab.ui.control.UIAxes
```

```
end
```

```
properties (Access = private)
```

```
    EstadoBoton
```

```
end
```

```
methods (Access = private)
```

```
function practicaValueChanged(app, event)
```

```
end
```

```
function PrcticaDropDownValueChanged(app, event)
```

```
    value = app.PrcticaDropDown.Value;
```

```
    global R L tmss iarr prac
```

```
    prac = value;
```

```
    if value == 5
```

```
        app.BajarpesaButton.Visible = 'on';
```

```
        app.PresinButton_2.Visible = 'off';
```

```
        app.VelocidaddelaireButton.Visible = 'off';
```

```
    elseif value == 6
```

```
        app.BajarpesaButton.Visible = 'off';
```

```
        app.PresinButton_2.Visible = 'on';
```

```
        app.VelocidaddelaireButton.Visible = 'off';
```

```
    elseif value == 7
```

```
        app.BajarpesaButton.Visible = 'off';
```

```
        app.PresinButton_2.Visible = 'off';
```

```
        app.VelocidaddelaireButton.Visible = 'on';
```

```
    end
```

```
end
```

```
function SeleccionalagrificaButtonGroupSelectionChanged(app, event)
```

```
    selectedButton = app.SeleccionalagrificaButtonGroup.SelectedObject;
```

```
    valor = selectedButton.Text;
```

```
    global dim
```

```
    dim = length(valor);
```

end

```
function IniciarPrcticaButtonPushed(app, event)
```

```
    global arduinoSerial
```

```
    delete(arduinoSerial);
```

```
    arduino = serialportlist("available");
```

```
    disp("Puertos disponibles:");
```

```
    disp(arduino);
```

```
    arduinoPort = arduino(1);
```

```
    disp("Conectando a: " + arduinoPort);
```

```
    delete(instrfindall);
```

```
    arduinoSerial = serialport(arduinoPort, 9600);
```

```
    configureTerminator(arduinoSerial, "LF");
```

```
    global seguirGraficando prac
```

```
    pause(1);
```

```
    if seguirGraficando
```

```
        if prac == 5
```

```
            writeline(arduinoSerial, "ON1");
```

```
        elseif prac == 6
```

```
            writeline(arduinoSerial, "ON3");
```

```
        elseif prac == 7
```

```
            writeline(arduinoSerial, "ON4");
```

```
            writeline(arduinoSerial, "ON5");
```

```
        end
```

```
    end
```

```
    serialPortName = arduinoSerial.Port;
```

```
    app.PuertodeconexinEditField.Value = serialPortName;
```

```
graficas = struct('Torque', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Voltaje', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Corriente', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Velocidad', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Potencia', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Presion', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2), ...
    'Flujo', plot(app.UIAxes, NaN, NaN, '-o', 'LineWidth', 2));
```

```
xlabel(app.UIAxes, 'Tiempo (s)');
ylabel(app.UIAxes, 'Valores');
title(app.UIAxes, 'Graficas desde Arduino');
grid(app.UIAxes, 'on');
```

```
tiempoMostrado = 10;
tiempoInicial = datetime('now', 'Format', 'ss.SSS');
```

```
historial = struct('Torque', [], 'Voltaje', [], 'Corriente', [], 'Velocidad', [], 'Potencia', [],
'Presion', [], 'Flujo', []);
tiempo = [];
```

```
while ishandle(app.UIAxes) && seguirGraficando
    data = str2double(split(readline(arduinoSerial), ','));
```

```
global R L tmss iarr dim fin prac ca estado
```

```
if ~isempty(data) && length(data) == 8
    I = data(1);
    V = data(2);
    W1 = data(3);
    W2 = data(4);
    W3 = data(5);
    F = data(6);
    Pr = data(7);
    Ca = data(8);
```

```

if Ca == 1
    seguirGraficando = false;
end

if prac == 5
    Wr = W1 * 0.10472;
    W = W1;
    R = 8.5;
elseif prac == 6
    Wr = W2 * 0.10472;
    W = W2;
    R = 4;
elseif prac == 7
    Wr = W3 * 0.10472;
    W = W3;
    R = 1.5;
end

Km = (V - I * R) / Wr;
Ka = Km;

if Wr == 0
    Tm = 0;
else
    Tm = (Km * I) * 100;
end

P = V * I;

historial.Torque = [historial.Torque, Tm];
historial.Potencia = [historial.Potencia, P];
historial.Voltaje = [historial.Voltaje, V];
historial.Corriente = [historial.Corriente, I];

```

```

historial.Velocidad = [historial.Velocidad, W];
historial.Presion = [historial.Presion, Pr];
historial.Flujo = [historial.Flujo, F];

tiempoTranscurrido = seconds(datetime('now', 'Format', 'ss.SSS') - tiempoInicial);
tiempo = [tiempo, tiempoTranscurrido];

global tor vol corr vel tim pot pres fluj tiempoDeSubida

tor = historial.Torque;
pot = historial.Potencia;
pres = historial.Presion;
fluj = historial.Flujo;
vol = historial.Voltaje;
corr = historial.Corriente;
vel = historial.Velocidad;
tim = tiempo;

if dim == 6
    plot(app.UIAxes, historial.Torque, '-o', 'LineWidth', 2);
elseif dim == 7
    plot(app.UIAxes, historial.Voltaje, '-o', 'LineWidth', 2);
elseif dim == 9
    plot(app.UIAxes, historial.Corriente, '-o', 'LineWidth', 2);
elseif dim == 10
    plot(app.UIAxes, historial.Velocidad, '-o', 'LineWidth', 2);
elseif dim == 8
    plot(app.UIAxes, historial.Potencia, '-o', 'LineWidth', 2);
elseif dim == 11
    plot(app.UIAxes, historial.Presion, '-o', 'LineWidth', 2);
elseif dim == 18
    plot(app.UIAxes, historial.Flujo, '-o', 'LineWidth', 2);
end

```

```

        if estado == 12
            seguirGraficando = true;
        elseif estado == 4
global arduinoSerial
if prac == 5
    writeline(arduinoSerial, "OFF1");
    writeline(arduinoSerial, "OFF2");
elif prac == 6
    writeline(arduinoSerial, "OFF3");
elif prac == 7
    writeline(arduinoSerial, "OFF4");
    writeline(arduinoSerial, "OFF5");
end
end

drawnow;

if fin == 100
    break;
end

end

function SeleccionaeleestadoButtonGroupSelectionChanged(app, event)
    selectedButton = app.SeleccionaeleestadoButtonGroup.SelectedObject;
    valor = selectedButton.Text;
    global seguirGraficando estado prac

    estado = length(valor);
    if estado == 12
        seguirGraficando = true;
    elseif estado == 4
        global arduinoSerial
        if prac == 5

```

```

        writeline(arduinoSerial, "OFF1");
        writeline(arduinoSerial, "OFF2");
elseif prac == 6
        writeline(arduinoSerial, "OFF3");
elseif prac == 7
        writeline(arduinoSerial, "OFF4");
        writeline(arduinoSerial, "OFF5");
end

seguirGraficando = false;
global ce
ce = true;
while ishandle(app.UIAxes) && ce
    global dim tor vol corr vel tim fin pot pres fluj

    if dim == 6
        plot(app.UIAxes, tim, tor, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Torque (Nm)');
        title(app.UIAxes, 'Gráfica de Torque');
        drawnow;
    elseif dim == 7
        plot(app.UIAxes, tim, vol, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Voltaje (V)');
        title(app.UIAxes, 'Gráfica de Voltaje');
        drawnow;
    elseif dim == 9
        plot(app.UIAxes, tim, corr, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Corriente (A)');
        title(app.UIAxes, 'Gráfica de Corriente');
        drawnow;
    elseif dim == 10

```

```

        plot(app.UIAxes, tim, vel, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Velocidad (rpm)');
        title(app.UIAxes, 'Gráfica de Velocidad');
        drawnow;
elseif dim == 8
        plot(app.UIAxes, tim, pot, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Potencia (W)');
        title(app.UIAxes, 'Gráfica de Potencia');
        drawnow;
elseif dim == 11
        plot(app.UIAxes, tim, pres, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Presión (PSI)');
        title(app.UIAxes, 'Gráfica de Presión');
        drawnow;
elseif dim == 18
        plot(app.UIAxes, tim, fluj, '-o', 'LineWidth', 2);
        xlabel(app.UIAxes, 'Tiempo (s)');
        ylabel(app.UIAxes, 'Velocidad del Aire (m/s)');
        title(app.UIAxes, 'Gráfica de la Velocidad del Aire');
        drawnow;
end

if fin == 100
    break;
end
end
end
end

function GuardardatosButtonPushed(app, event)
    global tor vol corr vel tim pot nombreArchivo prac pres fluj

```

```

if prac == 5
    datos = [corr; vol; vel; tor; pot; tim];
elseif prac == 6
    datos = [corr; vol; vel; tor; pot; pres; tim];
elseif prac == 7
    datos = [corr; vol; vel; tor; pot; fluj; tim];
end

writematrix(datos', [nombreArchivo, '.xls']);
end

function NombredelarchivoEditFieldValueChanged(app, event)
    global nombreArchivo
    nombreArchivo = app.NombredelarchivoEditField.Value;
end

function BajarpesaButtonPushed(app, event)
    global arduinoSerial prac

    if prac == 5
        if isempty(app.EstadoBoton)
            app.EstadoBoton = 0;
        end
        app.EstadoBoton = mod(app.EstadoBoton + 1, 2);
        if app.EstadoBoton == 0
            writeline(arduinoSerial, "ON2");
        else
            writeline(arduinoSerial, "OFF2");
        end
    end
end

function ParodeemergenciaButtonPushed(app, event)

```

```
global arduinoSerial seguirGraficando ce
```

```
writeline(arduinoSerial, "OFF1");
```

```
writeline(arduinoSerial, "OFF2");
```

```
writeline(arduinoSerial, "OFF3");
```

```
writeline(arduinoSerial, "OFF4");
```

```
writeline(arduinoSerial, "OFF5");
```

```
seguirGraficando = false;
```

```
ce = true;
```

```
end
```

```
end
```

```
methods (Access = private)
```

```
function createComponents(app)
```

```
    pathToMLAPP = fileparts(mfilename('fullpath'));
```

```
    app.UIFigure = uifigure('Visible', 'off');
```

```
    app.UIFigure.Position = [100 100 759 557];
```

```
    app.UIFigure.Name = 'MATLAB App';
```

```
    app.UIAxes = uiaxes(app.UIFigure);
```

```
    title(app.UIAxes, 'Title')
```

```
    xlabel(app.UIAxes, 'X')
```

```
    ylabel(app.UIAxes, 'Y')
```

```
    zlabel(app.UIAxes, 'Z')
```

```
    app.UIAxes.Position = [237 80 479 305];
```

```
    app.PesoalevantarLabel = uilabel(app.UIFigure);
```

```
    app.PesoalevantarLabel.HorizontalAlignment = 'right';
```

```
    app.PesoalevantarLabel.Position = [326 431 48 22];
```

```
    app.PesoalevantarLabel.Text = 'Práctica';
```

```
    app.PrcticaDropDown = uidropdown(app.UIFigure);
```

```
    app.PrcticaDropDown.Items = {'Seleccionar', 'Sistema de Grúa', 'Sistema de Bombeo',  
'Sistema de Ventilación'};
```

```
    app.PrcticaDropDown.ItemsData = [4 5 6 7 8];
```

```
    app.PrcticaDropDown.ValueChangedFcn = createCallbackFcn(app,  
@PrcticaDropDownValueChanged, true);
```

```

app.PrcticaDropDown.Position = [280 410 195 22];
app.PrcticaDropDown.Value = 4;
app.SeleccionalagrficaButtonGroup = uibuttongroup(app.UIFigure);
app.SeleccionalagrficaButtonGroup.SelectionChangedFcn = createCallbackFcn(app,
@SeleccionalagrficaButtonGroupSelectionChanged, true);
app.SeleccionalagrficaButtonGroup.Title = 'Selecciona la gráfica';
app.SeleccionalagrficaButtonGroup.Position = [60 272 137 160];
app.Button = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.Button.Text = '';
app.Button.Position = [120 137 25 22];
app.Button.Value = true;
app.VoltajeButton = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.VoltajeButton.Text = 'Voltaje';
app.VoltajeButton.Position = [11 92 65 22];
app.CorrienteButton = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.CorrienteButton.Text = 'Corriente';
app.CorrienteButton.Position = [11 69 71 22];
app.VelocidadButton = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.VelocidadButton.Text = 'Velocidad ';
app.VelocidadButton.Position = [11 46 77 22];
app.TorqueButton_2 = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.TorqueButton_2.Text = 'Torque';
app.TorqueButton_2.Position = [10 112 59 22];
app.PotenciaButton = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.PotenciaButton.Text = 'Potencia';
app.PotenciaButton.Position = [11 25 69 22];
app.PresinButton_2 = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.PresinButton_2.Visible = 'off';
app.PresinButton_2.Text = 'Presión  ';
app.PresinButton_2.Position = [11 4 76 22];
app.VelocidaddelaireButton = uiradiobutton(app.SeleccionalagrficaButtonGroup);
app.VelocidaddelaireButton.Visible = 'off';
app.VelocidaddelaireButton.Text = 'Velocidad del aire';
app.VelocidaddelaireButton.Position = [11 4 117 22];

```

```

app.IniciarPrcticaButton = uibutton(app.UIFigure, 'push');
app.IniciarPrcticaButton.ButtonPushedFcn = createCallbackFcn(app,
@IniciarPrcticaButtonPushed, true);
app.IniciarPrcticaButton.Position = [596 29 100 23];
app.IniciarPrcticaButton.Text = 'Iniciar Práctica';
app.PuertodeconexcionLabel = uilabel(app.UIFigure);
app.PuertodeconexcionLabel.HorizontalAlignment = 'right';
app.PuertodeconexcionLabel.Position = [529 437 112 22];
app.PuertodeconexcionLabel.Text = 'Puerto de conexión: ';
app.PuertodeconexinEditField = uieditfield(app.UIFigure, 'text');
app.PuertodeconexinEditField.Editable = 'off';
app.PuertodeconexinEditField.HorizontalAlignment = 'center';
app.PuertodeconexinEditField.Position = [535 413 100 22];
app.GuardardatosButton = uibutton(app.UIFigure, 'push');
app.GuardardatosButton.ButtonPushedFcn = createCallbackFcn(app,
@GuardardatosButtonPushed, true);
app.GuardardatosButton.Position = [366 29 100 23];
app.GuardardatosButton.Text = 'Guardar datos';
app.UIFigure.Visible = 'on';
end
end
methods (Access = public)
function app = final2
    createComponents(app)
    registerApp(app, app.UIFigure)
    if nargin == 0
        clear app
    end
end
function delete(app)
    delete(app.UIFigure)
end
end

```

Código Arduino:

```

#include <Arduino.h>

float sensibilidad = 0.183;
int sensorVoltaje = A1;
const int pinFinalCarrera = 5;
const int sensorPin = A3;
const float sensorMinVoltage = 0.5;
const float sensorMaxVoltage = 4.5;
const float sensorMinPressure = 0.0;
const float sensorMaxPressure = 72.5189;
int sensorValue = 0;
float voltage = 0.0;
float pressure = 0.0;
float lastValidPressure = 0.0;
const int mafPin = A2;
float voltajes[] = {0, 0.35, 0.38, 0.6, 0.86, 1.04, 1.08, 1.19, 1.3, 1.35, 1.45, 1.5, 1.59};
float velocidades[] = {0, 0.8, 1, 1.5, 1.9, 2, 2.5, 3.2, 3.4, 4, 5.8, 6.4, 6.7};
const int numSamples = 10;
float samples[numSamples];
int sampleIndex = 0;
bool samplesFilled = false;
int relayPin10 = 10;
int relayPin9 = 9;
int relayPin8 = 8;
int relayPin4 = 4;
int relayPin11 = 11;
int relayPin6 = 6;
int relayPin7 = 7;
bool relayState = false;
const int encoderPin1 = 2;
const int encoderPin2 = 3;
const int encoderPin3 = 18;
volatile int contadorVueltas1 = 0;
volatile int contadorVueltas2 = 0;

```

```

volatile int contadorVueltas3 = 0;
volatile int vueltas1 = 0;
volatile int vueltas2 = 0;
volatile int vueltas3 = 0;

void encoderInterrupt1() {
  int currentState1 = digitalRead(encoderPin1);
  if (currentState1 != vueltas1) {
    contadorVueltas1++;
  }
  vueltas1 = currentState1;
}

void encoderInterrupt2() {
  int currentState2 = digitalRead(encoderPin2);
  if (currentState2 != vueltas2) {
    contadorVueltas2++;
  }
  vueltas2 = currentState2;
}

void encoderInterrupt3() {
  int currentState3 = digitalRead(encoderPin3);
  if (currentState3 != vueltas3) {
    contadorVueltas3++;
  }
  vueltas3 = currentState3;
}

void setup() {
  Serial.begin(9600);
  pinMode(pinFinalCarrera, INPUT);
  pinMode(encoderPin1, INPUT);
  pinMode(encoderPin2, INPUT);
  pinMode(encoderPin3, INPUT);
  attachInterrupt(digitalPinToInterrupt(encoderPin1), encoderInterrupt1, HIGH);
}

```

```

attachInterrupt(digitalPinToInterrupt(encoderPin2), encoderInterrupt2, HIGH);
attachInterrupt(digitalPinToInterrupt(encoderPin3), encoderInterrupt3, HIGH);
pinMode(relayPin10, OUTPUT);
pinMode(relayPin9, OUTPUT);
pinMode(relayPin8, OUTPUT);
pinMode(relayPin4, OUTPUT);
pinMode(relayPin11, OUTPUT);
pinMode(relayPin6, OUTPUT);
pinMode(relayPin7, OUTPUT);
digitalWrite(relayPin10, LOW);
digitalWrite(relayPin9, HIGH);
digitalWrite(relayPin8, LOW);
digitalWrite(relayPin4, HIGH);
digitalWrite(relayPin11, HIGH);
digitalWrite(relayPin6, HIGH);
digitalWrite(relayPin7, HIGH);
for (int i = 0; i < numSamples; i++) {
    samples[i] = 0;
}
}

void loop() {
    float corriente = calcularCorriente(500);
    if (corriente < 1) {
        corriente += 0.10;
    } else if (corriente < 1.5) {
        corriente += 0.04;
    }
    if (corriente < 0) {
        corriente = 0;
    }
    float voltaje = ((float)25 * analogRead(sensorVoltaje) / 1023);
    int sensorValue = analogRead(mafPin);
    float voltage = sensorValue * (5.0 / 1023.0);
    samples[sampleIndex] = voltage;
}

```

```

sampleIndex++;
if (sampleIndex >= numSamples) {
    sampleIndex = 0;
    samplesFilled = true;
}
float voltageAvg = 0;
for (int i = 0; i < numSamples; i++) {
    voltageAvg += samples[i];
}
if (samplesFilled) {
    voltageAvg /= numSamples;
} else {
    voltageAvg /= (sampleIndex + 1);
}
float valormaf = calcularVelocidad(voltageAvg);
sensorValue = analogRead(sensorPin);
voltage = sensorValue * (5.0 / 1023.0);
pressure = map(voltage, sensorMinVoltage, sensorMaxVoltage, sensorMinPressure,
sensorMaxPressure);
if (pressure < 0) {
    pressure = lastValidPressure;
} else {
    lastValidPressure = pressure;
}
if (Serial.available() > 0) {
    String command = Serial.readStringUntil('\n');
    if (command == "ON2") {
        digitalWrite(relayPin4, LOW);
        digitalWrite(relayPin11, LOW);
    } else if (command == "OFF2") {
        digitalWrite(relayPin4, HIGH);
        digitalWrite(relayPin11, HIGH);
    } else if (command == "ON1") {
        digitalWrite(relayPin6, LOW);
    }
}

```

```

    digitalWrite(relayPin7, LOW);
} else if (command == "OFF1") {
    digitalWrite(relayPin6, HIGH);
    digitalWrite(relayPin7, HIGH);
} else if (command == "ON3") {
    digitalWrite(relayPin8, HIGH);
} else if (command == "OFF3") {
    digitalWrite(relayPin8, LOW);
} else if (command == "ON4") {
    digitalWrite(relayPin9, LOW);
} else if (command == "OFF4") {
    digitalWrite(relayPin9, HIGH);
} else if (command == "ON5") {
    digitalWrite(relayPin10, HIGH);
} else if (command == "OFF5") {
    digitalWrite(relayPin10, LOW);
}
}
int estadoFinalCarrera = digitalRead(pinFinalCarrera);
Serial.print(corriente, 2);
Serial.print(",");
Serial.print(voltaje);
Serial.print(",");
Serial.print(contadorVueltas1 / 2);
Serial.print(",");
Serial.print((((contadorVueltas2 / 2) / 16) * 60);
Serial.print(",");
Serial.print((((contadorVueltas3 / 2) / 14) * 60);
Serial.print(",");
Serial.print(valormaf);
Serial.print(",");
Serial.print(pressure);
Serial.print(",");
if (estadoFinalCarrera == HIGH) {

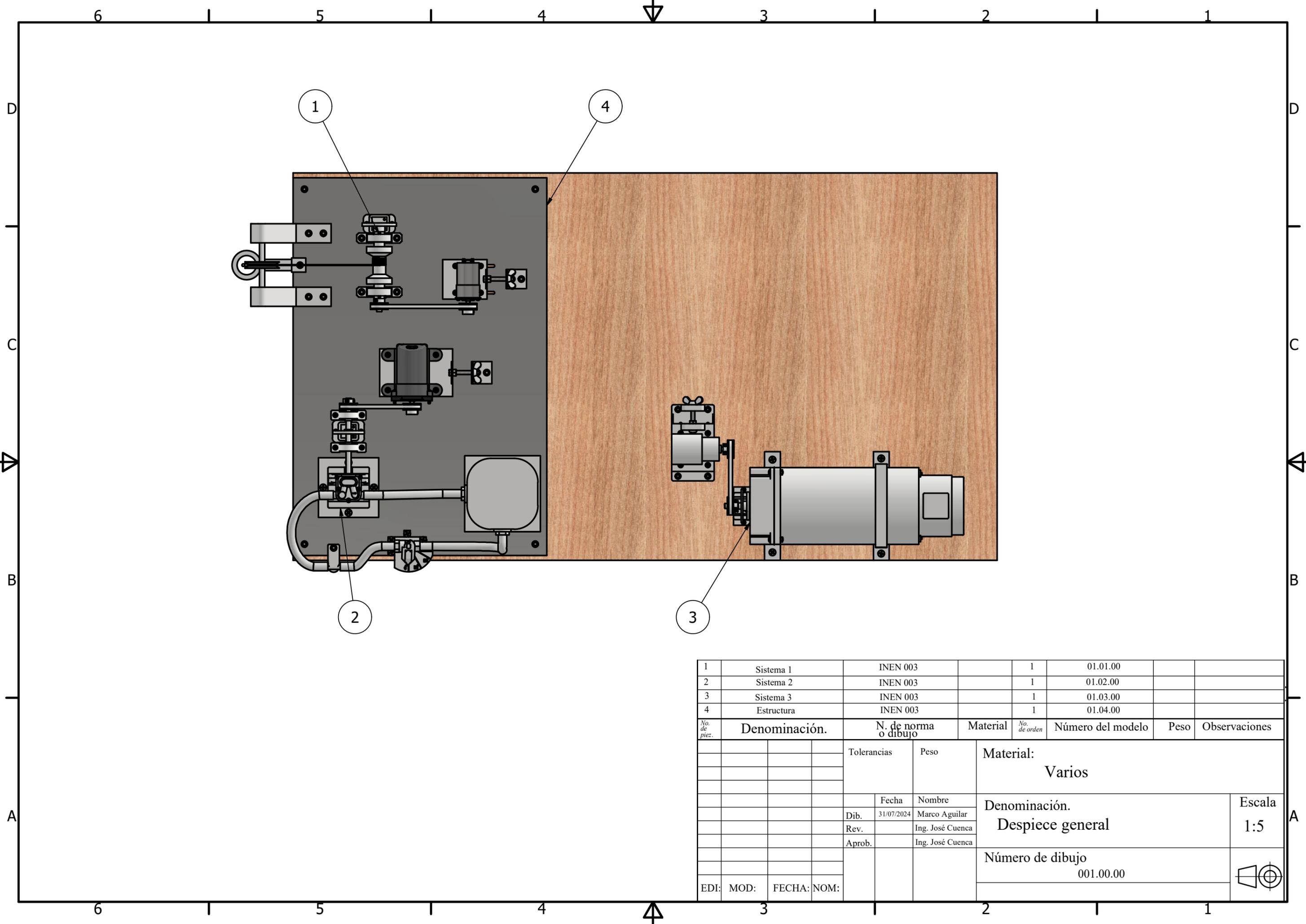
```

```

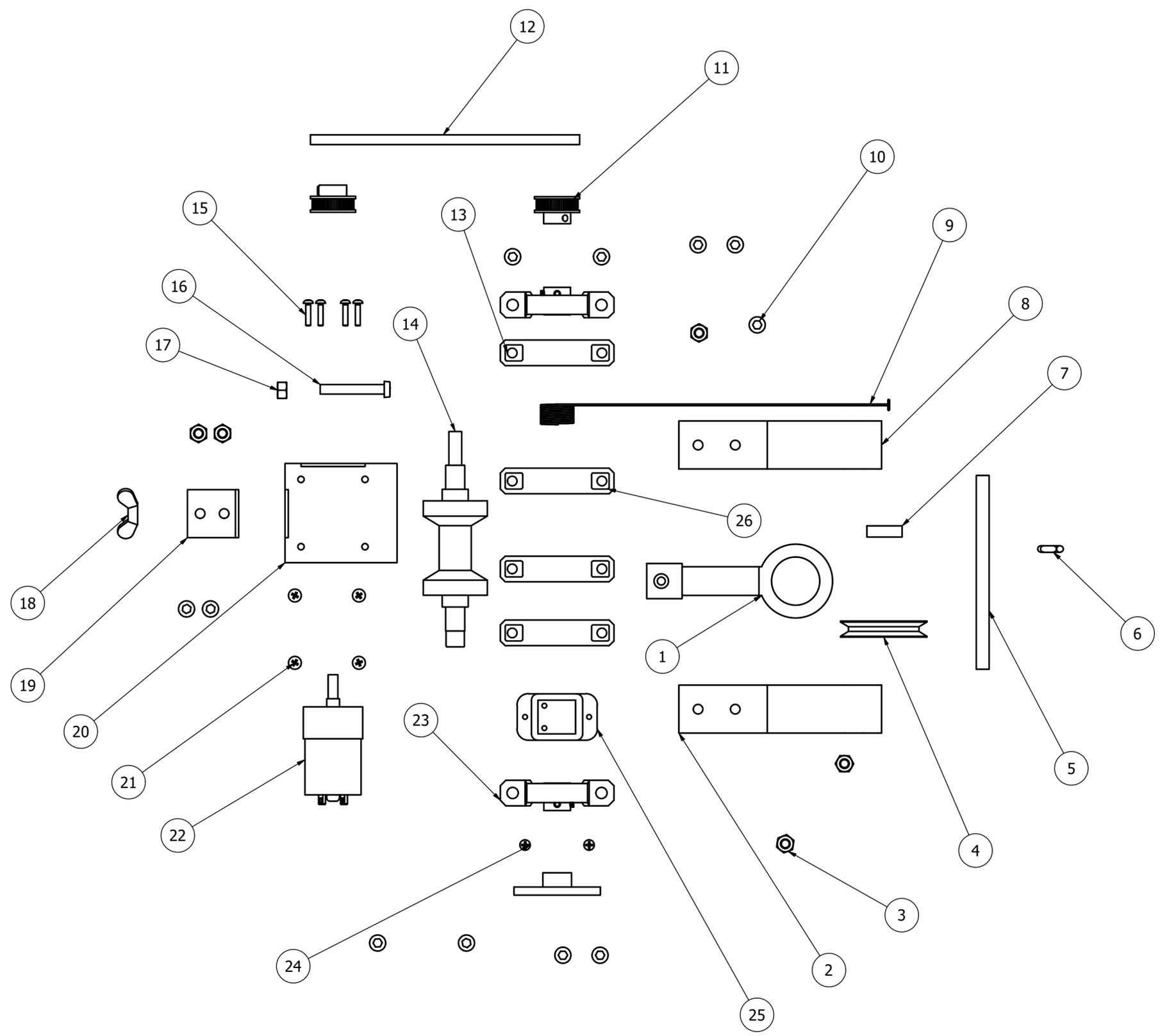
digitalWrite(relayPin6, HIGH);
digitalWrite(relayPin7, HIGH);
Serial.println(1);
} else {
    Serial.println(0)
}
contadorVueltas1 = 0;
contadorVueltas2 = 0;
contadorVueltas3 = 0;
delay(1000);
}
float calcularCorriente(int numeroMuestras) {
    float lecturaAcs712 = 0;
    float intensidad = 0;
    for (int i = 0; i < numeroMuestras; i++) {
        lecturaAcs712 = analogRead(A0) * (5 / 1023.0);
        intensidad = intensidad + ((lecturaAcs712 - 2.5) / sensibilidad);
    }
    intensidad =( intensidad / numeroMuestras);
    return intensidad;
}
float calcularVelocidad(float voltage) {
    for (int i = 0; i < 12; i++) {
        if (voltage >= voltajes[i] && voltage <= voltajes[i + 1]) {
            return velocidades[i] + (voltage - voltajes[i]) * (velocidades[i + 1] - velocidades[i]) /
(voltajes[i + 1] - voltajes[i]);
        }
    }
    return 0;
}
float map(float x, float in_min, float in_max, float out_min, float out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}

```

Anexo 7. Planos.

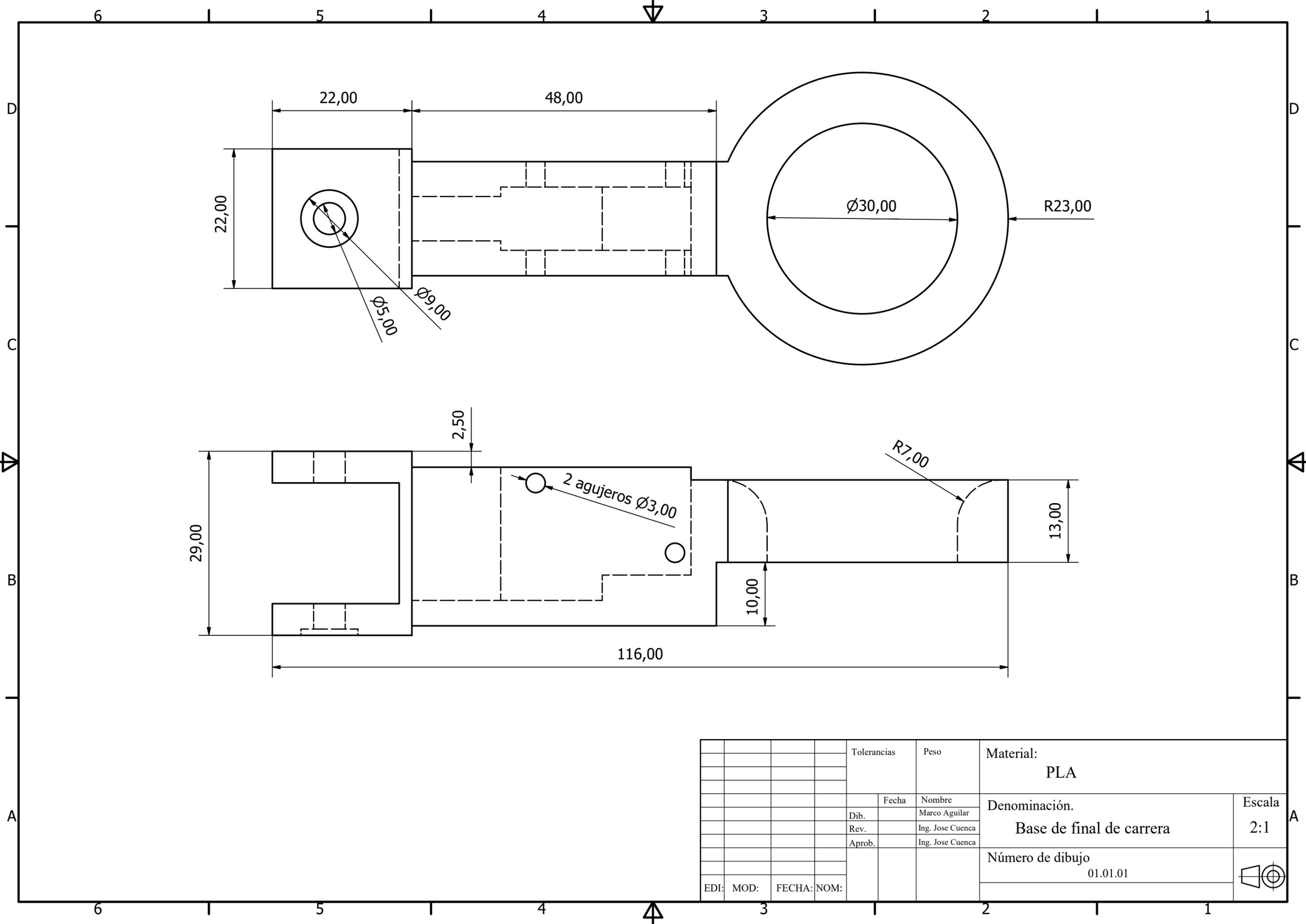


1	Sistema 1	INEN 003		1	01.01.00		
2	Sistema 2	INEN 003		1	01.02.00		
3	Sistema 3	INEN 003		1	01.03.00		
4	Estructura	INEN 003		1	01.04.00		
<i>No. de pieza.</i>	Denominación.	N. de norma o dibujo	Material	<i>No. de orden</i>	Número del modelo	Peso	Observaciones
		Tolerancias	Peso	Material: Varios			
		Fecha	Nombre	Denominación. Despiece general			Escala 1:5
	Dib.	31/07/2024	Marco Aguilar				
	Rev.		Ing. José Cuenca				
	Aprob.		Ing. José Cuenca				
				Número de dibujo 001.00.00			
EDI:	MOD:	FECHA:	NOM:				

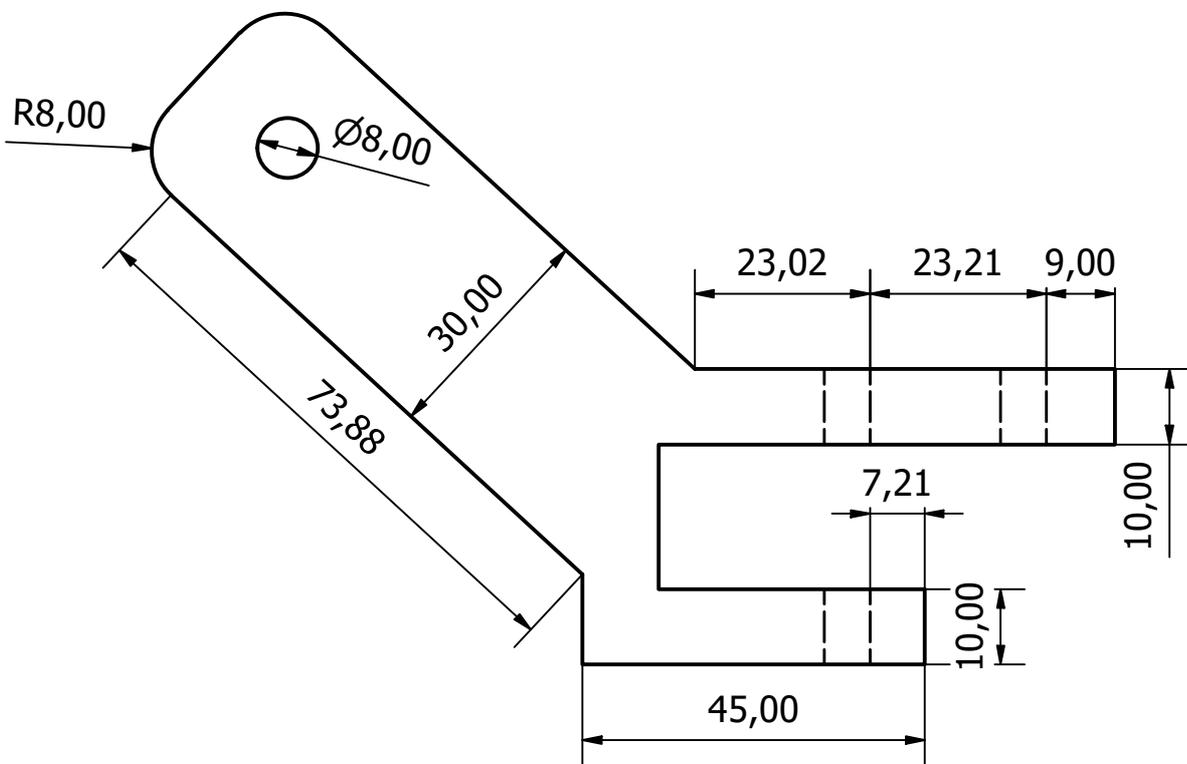
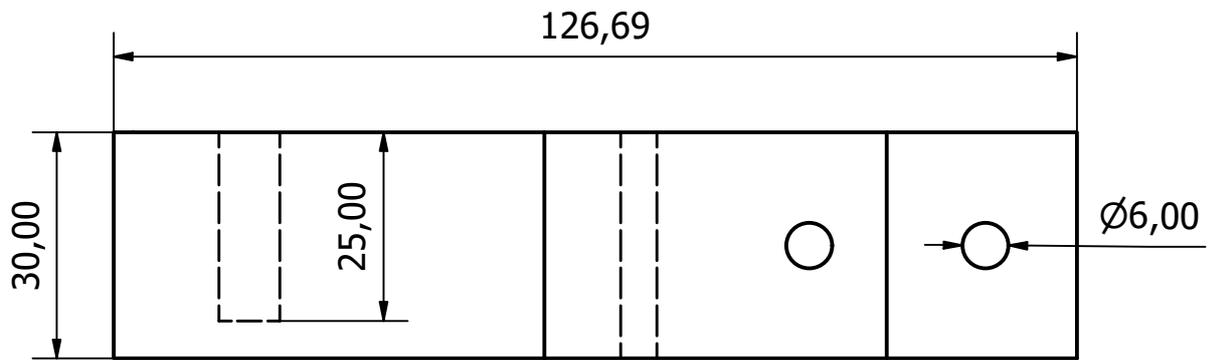


No. de pieza	Denominación	N. de norma o dibujo	Material	No. de orden	Número del modelo	Peso	Observaciones
1	Base de final de carrera	INEN 003		1	01.01.01		PLA
2	Brazo de grúa derecho	INEN 003		1	01.01.02		PLA
3	BS EN 24032 - Métrico M6:28	INEN 003		9			Acero Galvanizado
4	Polea loca	INEN 003		1	01.01.03		PLA
5	Eje de polea loca	INEN 003		1	01.01.04		Acero inoxidable
6	Gancho sujetador de pesas	INEN 003		1	01.01.05		PLA
7	Rodamiento de 8 mm	INEN 003		1			Acero, entre otros
8	Brazo de grúa izquierdo	INEN 003		1	01.01.06		PLA
9	Filamento de nylon	INEN 003		1			Nylon
10	AS 1420 - Métrico M6 x 40:9	INEN 003		11			Acero Galvanizado
11	Polea dentada	INEN 003		2			Aluminio
12	Banda dentada	INEN 003		1			Caucho, entre otros
13	Base de chumacera	INEN 003		2	01.01.07		Acero, entre otros
14	Enrolladora de cable	INEN 003		1	01.01.08		PLA
15	BS 4183. Serie métrica - Métrico M3 x 14:6	INEN 003		6			Acero Galvanizado
16	Screw GB/T 822-2000 Type H M6 x 40:2	INEN 003		1			Acero Galvanizado
17	ANSI B18.2.4.2M M6x1:2	INEN 003		1			Acero Galvanizado
18	Tuerca mariposa	INEN 003		1			Acero Galvanizado
19	Base de tensado	INEN 003		1	01.01.09		Acero Galvanizado
20	Base del motor-reductor	INEN 003		1	01.01.10		Acero Galvanizado
21	AS 1427 H - Métrico M4 x 16:11	INEN 003		1			Acero Galvanizado
22	Motor-reductor	INEN 003		4			Acero, entre otros
23	Chumacera	INEN 003		2			Acero, entre otros
24	CNS 4306 - AB M3,5 x 16:8	INEN 003		2			Acero Galvanizado
25	Base de encoder	INEN 003		2	01.01.11		PLA
26	Ajustador de altura de chumacera	INEN 003		1	01.01.12		PLA

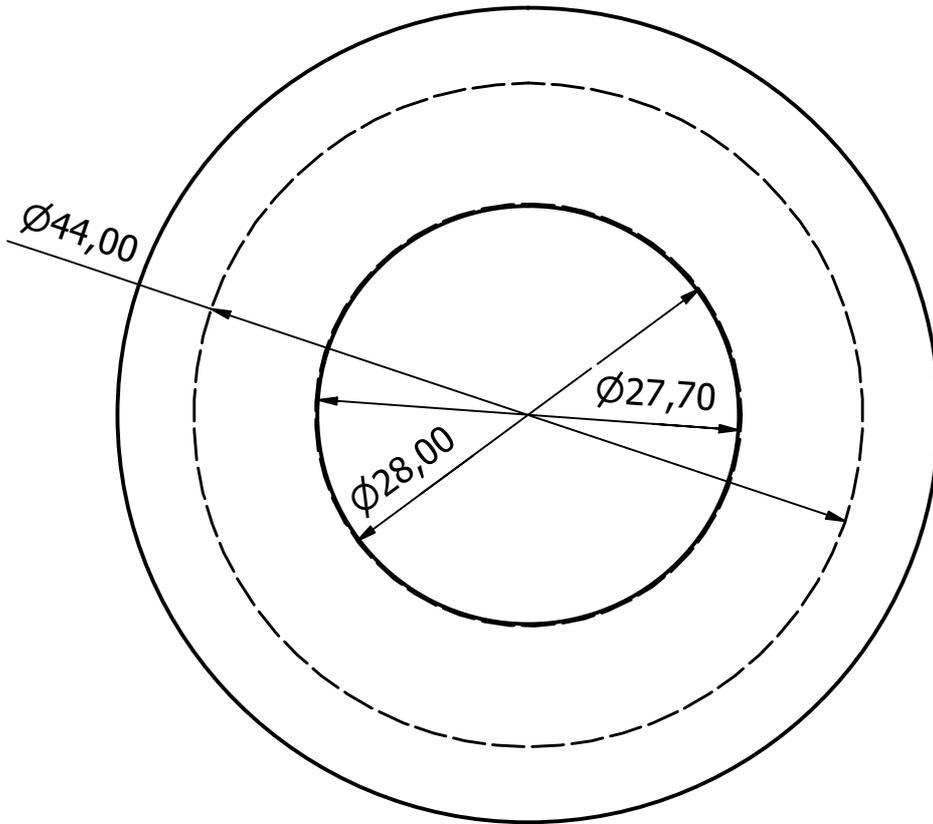
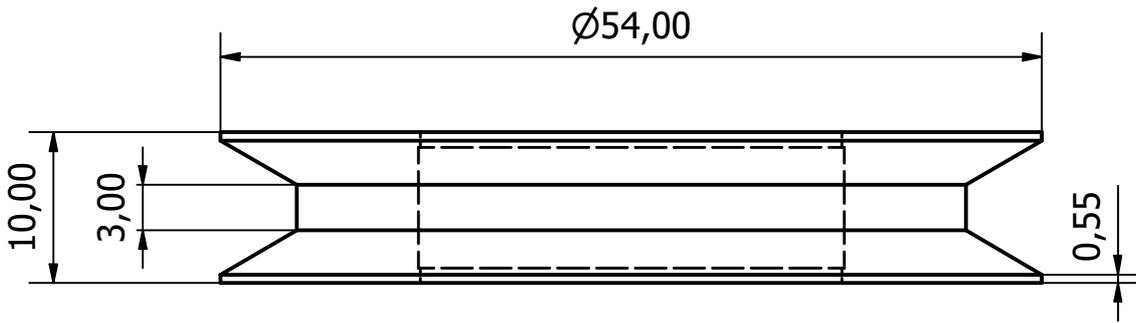
Tolerancias		Peso		Material:		
				Varios		
		Fecha	Nombre	Denominación. Despiece sistema 1		
		Dib.	31/07/2024			Marco Aguilar
		Rev.				Ing. José Cuenca
		Aprob.		Ing. José Cuenca	Escala 1:2	
Número de dibujo 001.01.00						
EDI:	MOD:	FECHA:	NOM:			



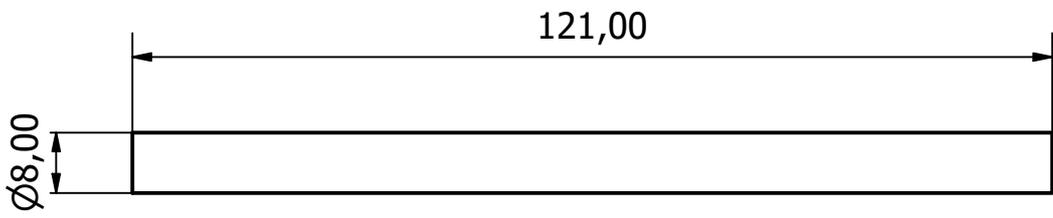
				Tolerancias	Peso	Material: PLA	
						Denominación: Base de final de carrera	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. Jose Cuenca	Número de dibujo 01.01.01	
				Aprob.	Ing. Jose Cuenca		
EDI:	MOD:	FECHA:	NOM:				



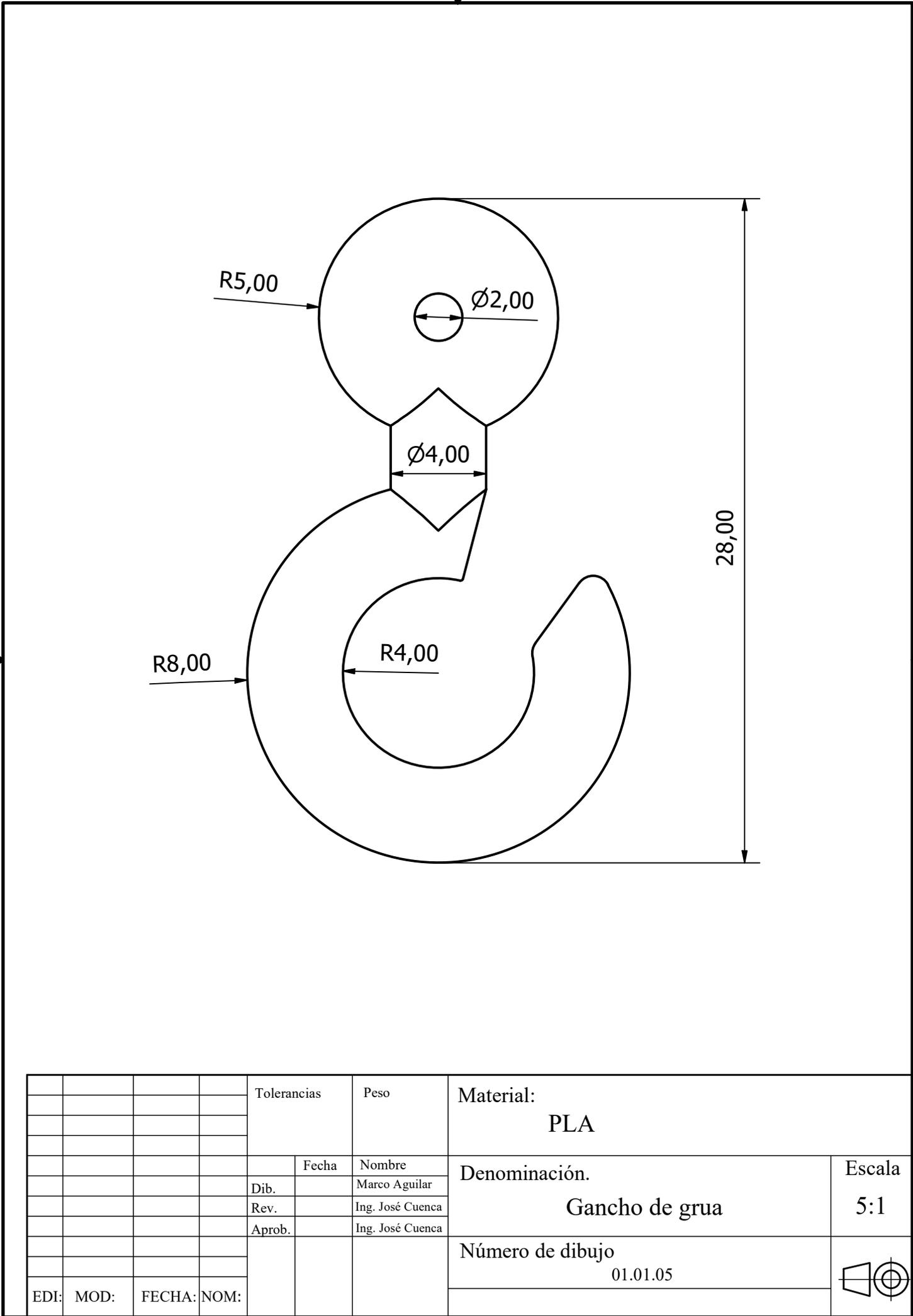
				Tolerancias	Peso	Material: PLA	
						Denominación. Brazo de grúa derecho	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.01.02	
EDI:	MOD:	FECHA:	NOM:				

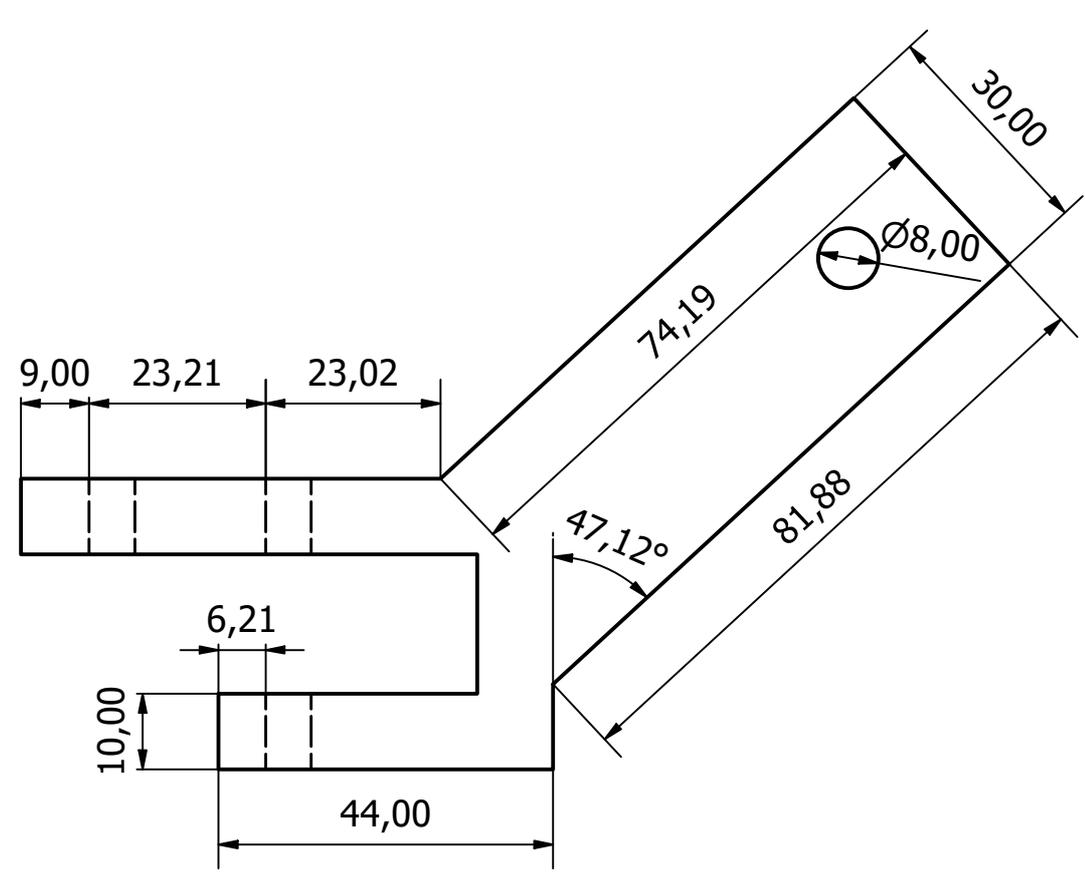
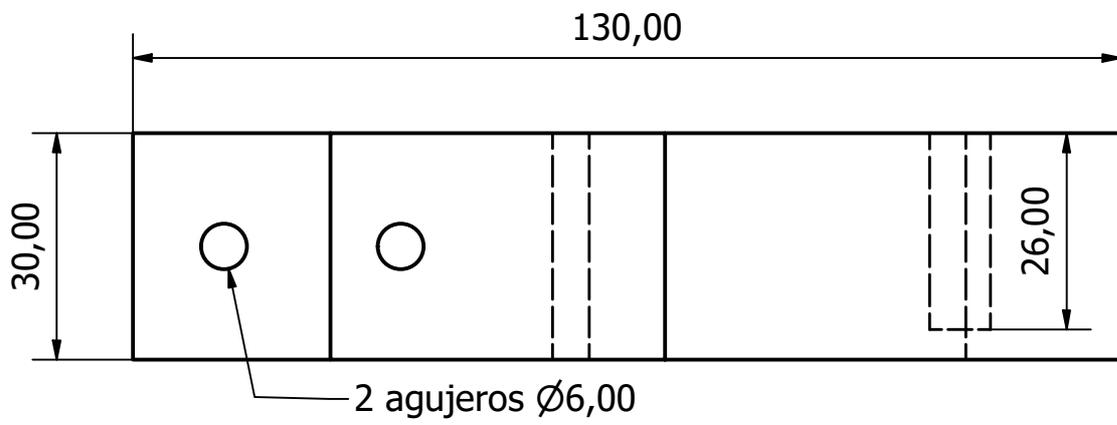


					Tolerancias	Peso	Material: PLA	
							Denominación. Polea loca	Escala 2:1
					Fecha	Nombre		
					Dib.	Marco Aguilar		
					Rev.	Ing. José Cuenca		
					Aprob.	Ing. José Cuenca		
							Número de dibujo 01.02.03	
EDI:	MOD:	FECHA:	NOM:					

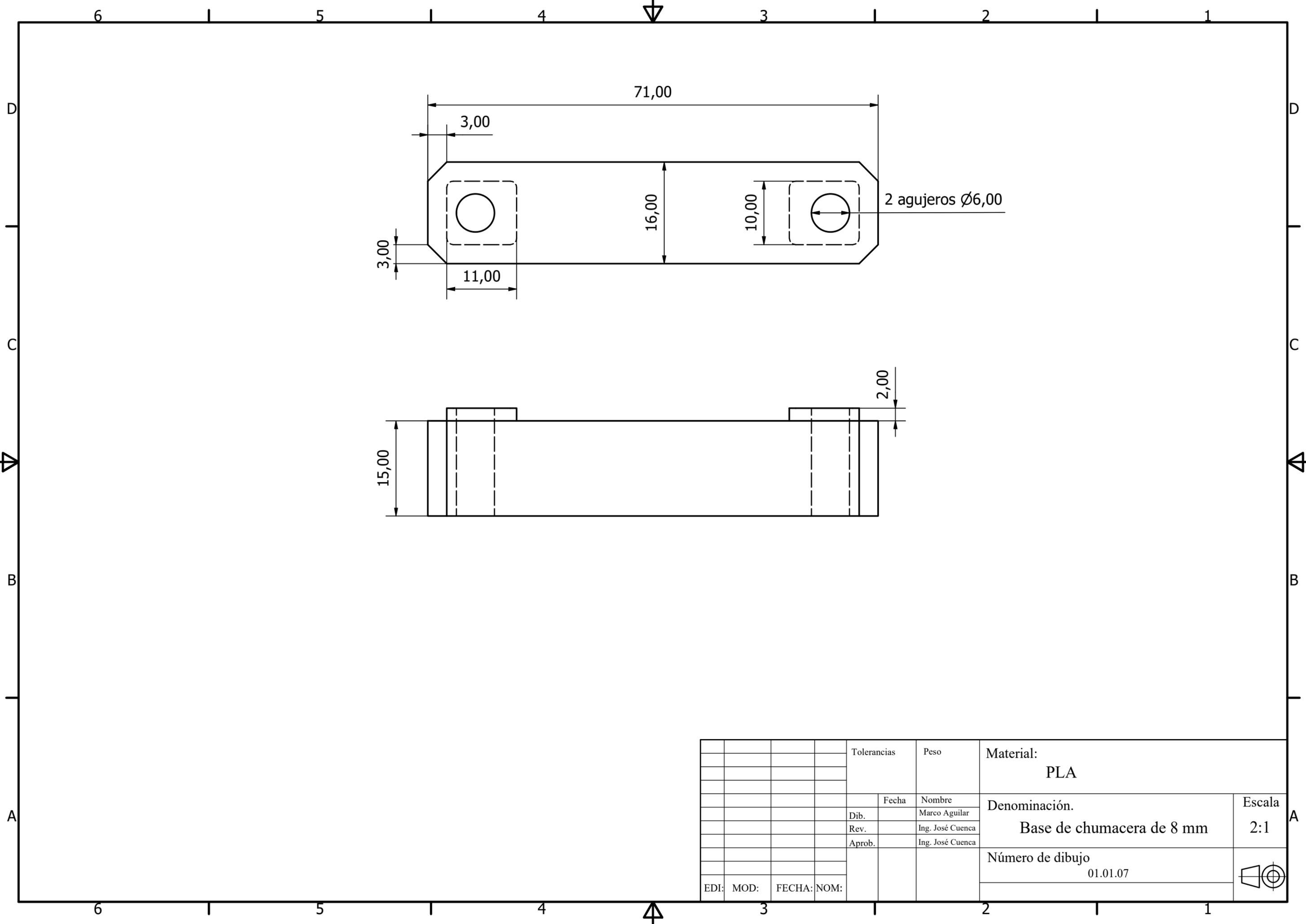


				Tolerancias	Peso	Material: Acero inoxidable	
						Denominación. Eje de polea loca	Escala 1:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.01.04	
EDI:	MOD:	FECHA:	NOM:				

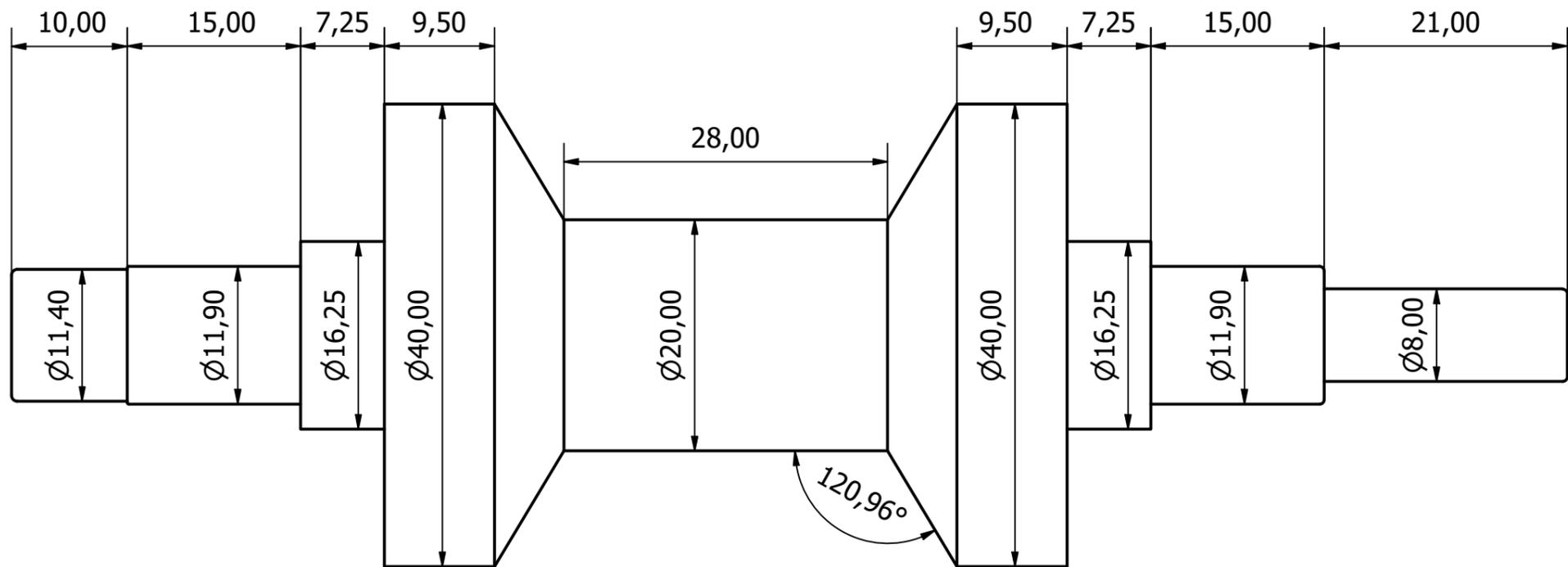


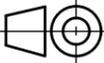


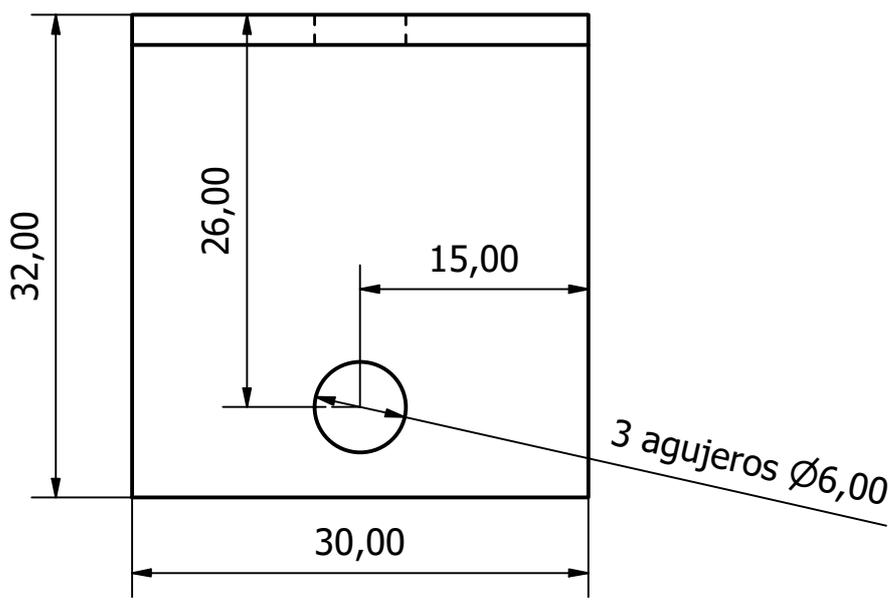
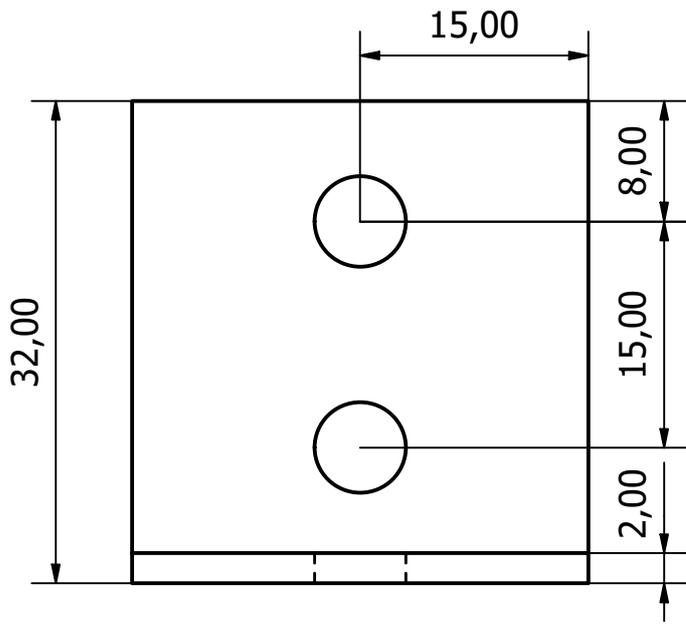
				Tolerancias	Peso	Material: PLA	
						Denominación. Brazo de grúa izquierdo	Escala 1:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.01.06	
EDI:	MOD:	FECHA:	NOM:				



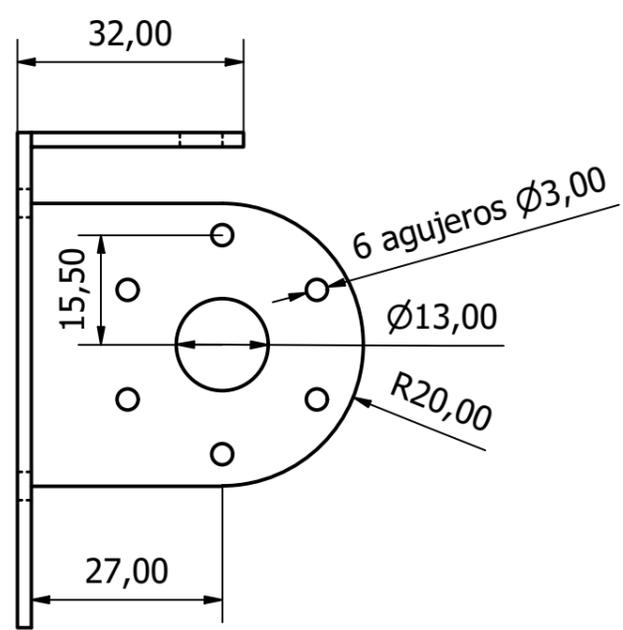
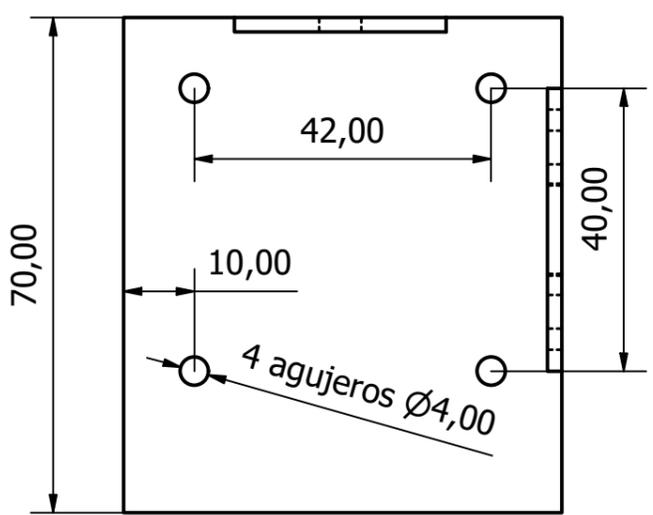
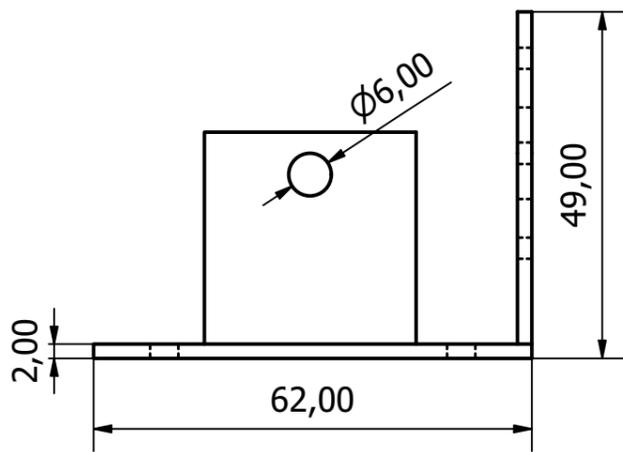
				Tolerancias	Peso	Material: PLA		
						Denominación. Base de chumacera de 8 mm	Escala 2:1	
					Fecha			Nombre
				Dib.				Marco Aguilar
				Rev.		Ing. José Cuenca		
				Aprob.		Ing. José Cuenca		
						Número de dibujo 01.01.07		
EDI:	MOD:	FECHA:	NOM:					



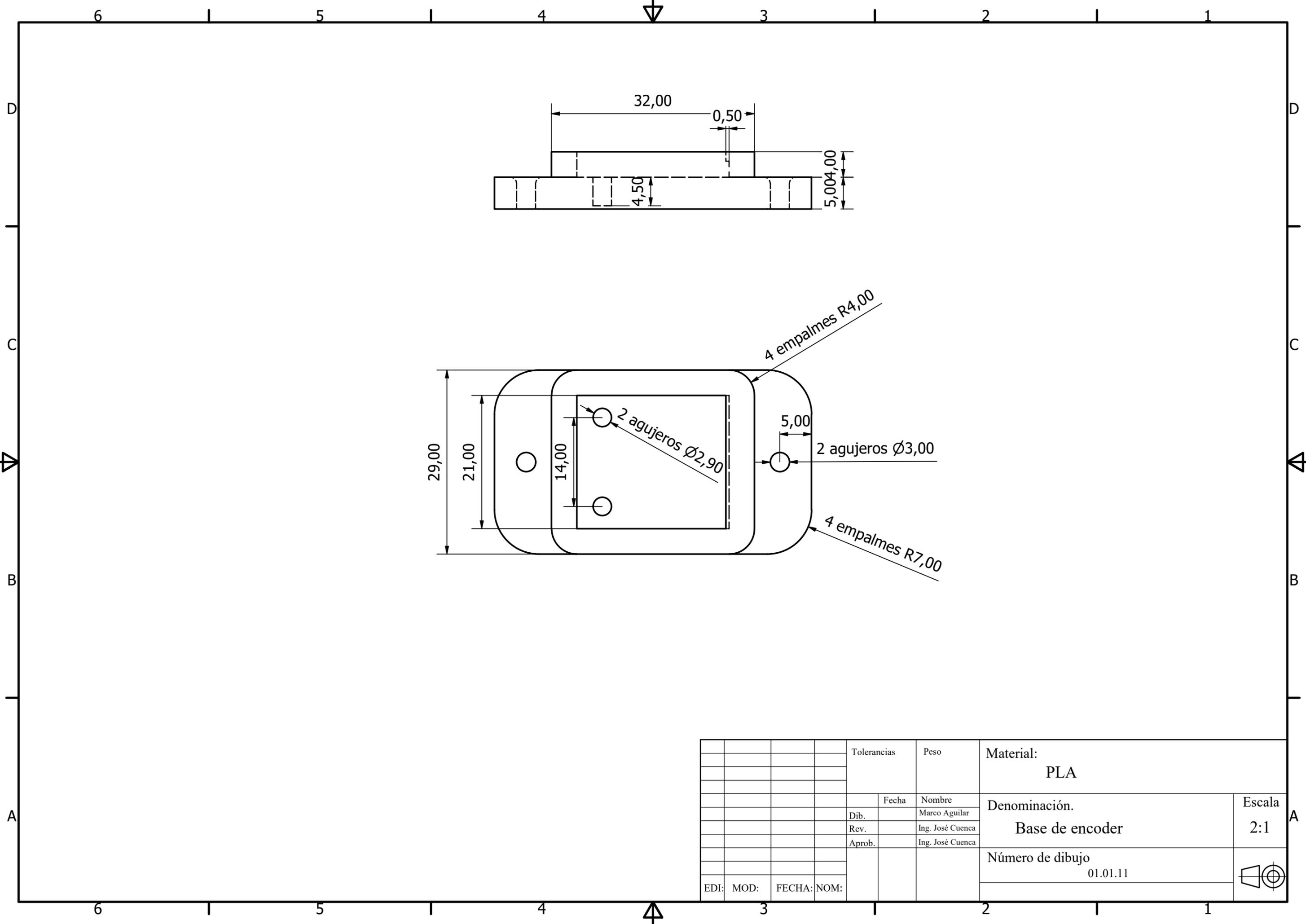
				Tolerancias	Peso	Material:	
						PLA	
					Fecha	Nombre	Escala
				Dib.		Marco Aguilar	2:1
				Rev.		Ing. José Cuenca	
				Aprob.		Ing. José Cuenca	
						Número de dibujo	
						01.01.08	
EDI:	MOD:	FECHA:	NOM:				



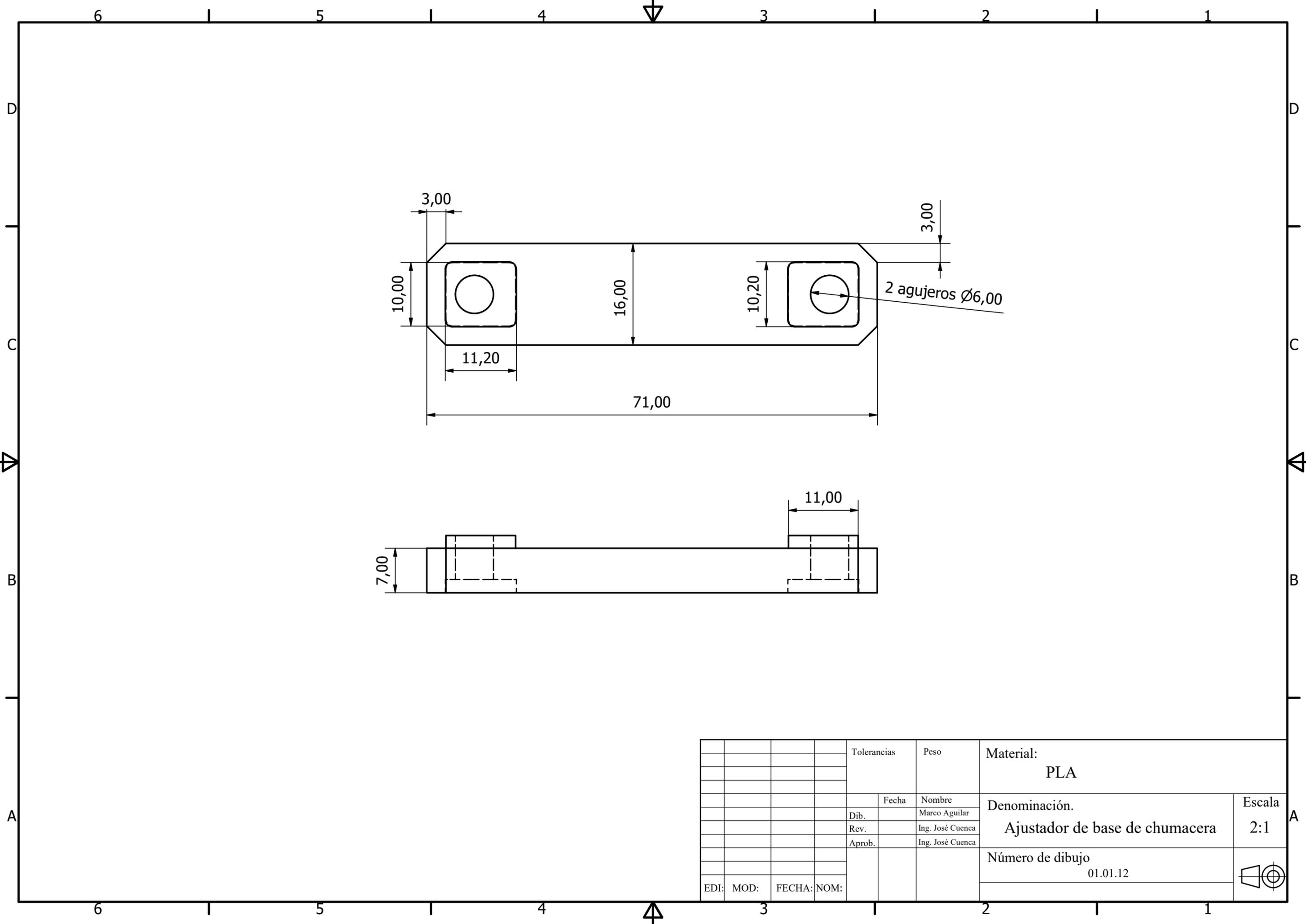
				Tolerancias	Peso	Material: Acero galvanizado	
						Denominación. Base de tensado	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.01.09	
EDI:	MOD:	FECHA:	NOM:				



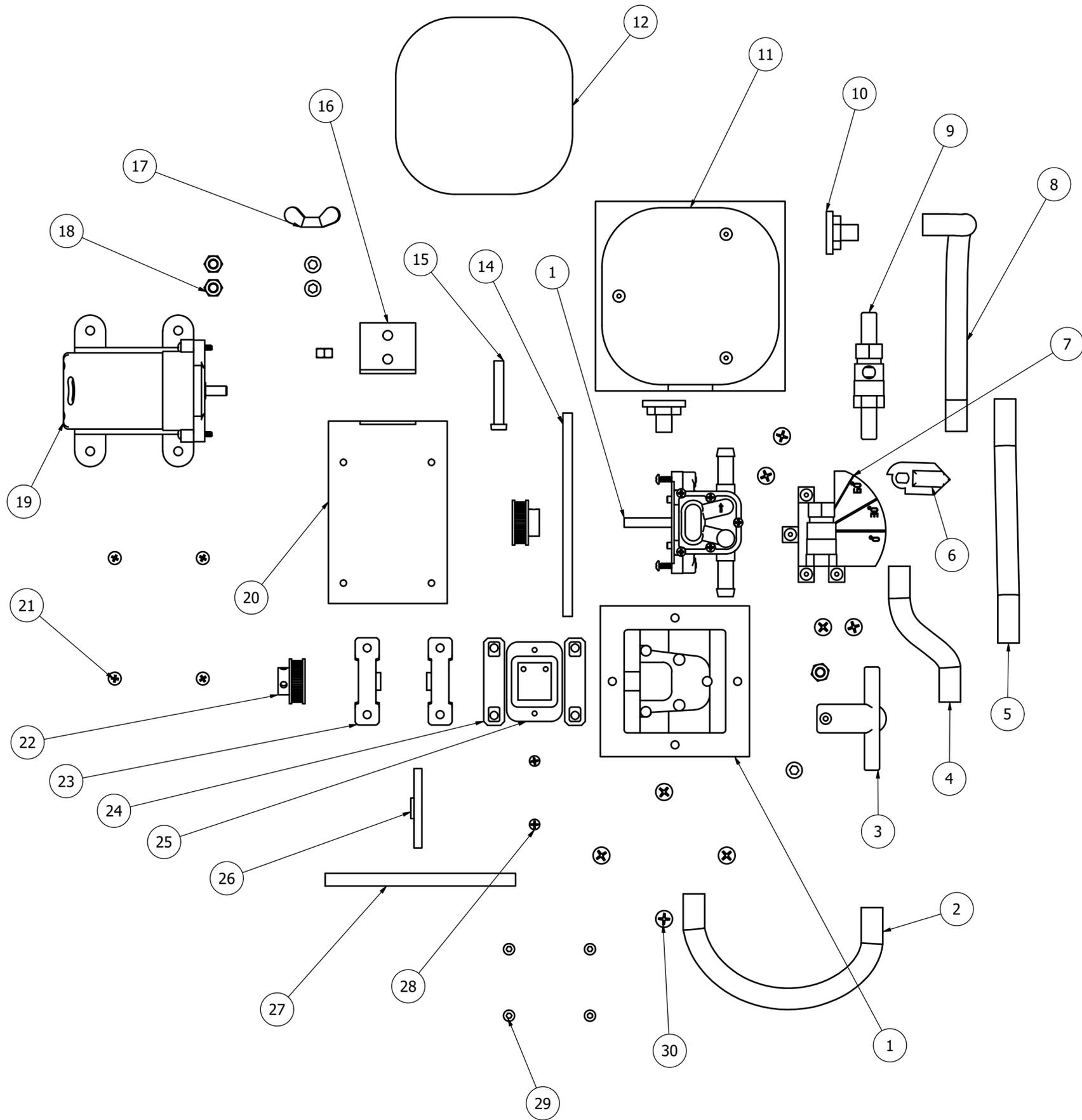
				Tolerancias	Peso	Material: Acero galvanizado	
						Denominación. Base de motor-reductor	Escala 1:1
						Número de dibujo 01.01.10	
EDI:	MOD:	FECHA:	NOM:				



				Tolerancias	Peso	Material: PLA	
						Denominación. Base de encoder	Escala 2:1
						Número de dibujo 01.01.11	
EDI:	MOD:	FECHA:	NOM:				

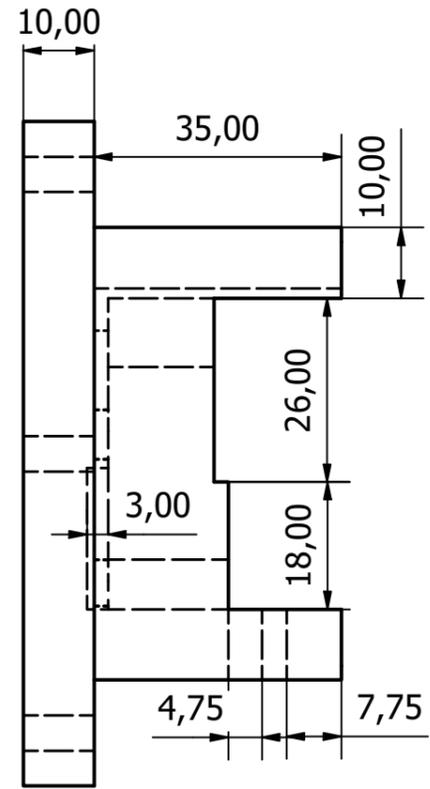
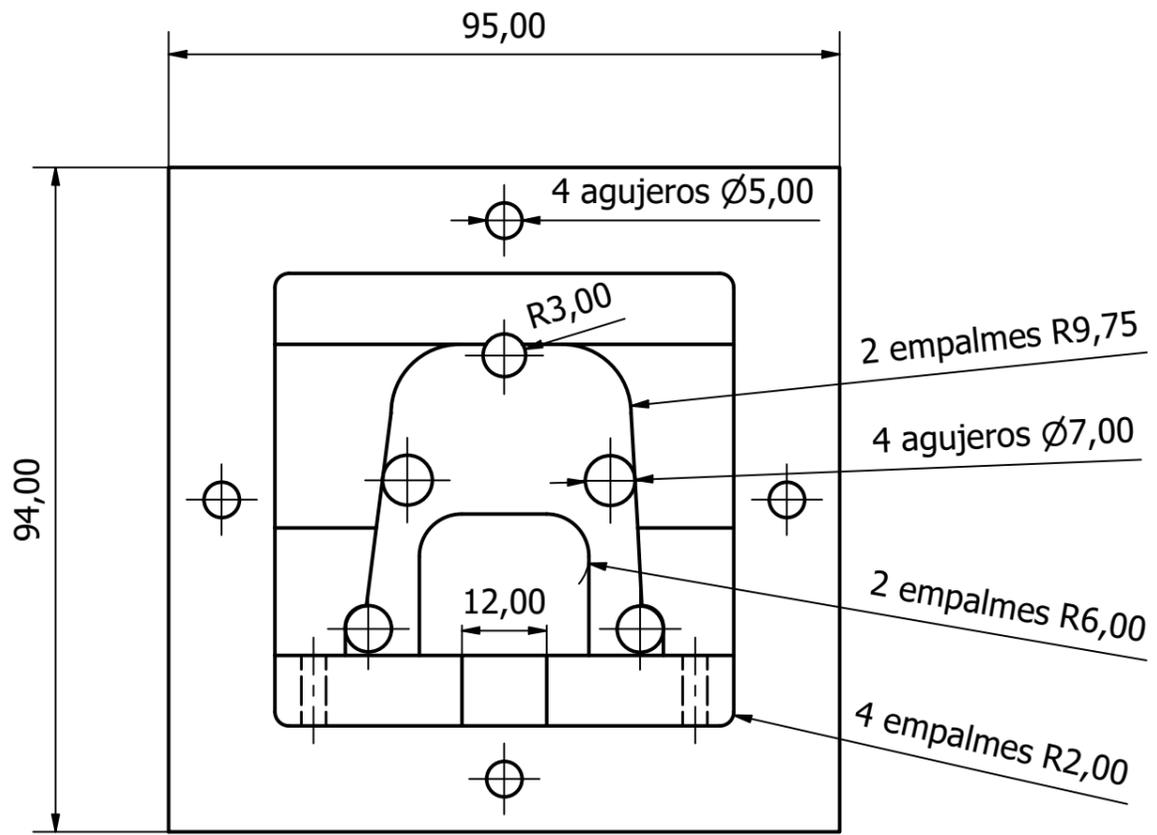


				Tolerancias	Peso	Material: PLA	
						Denominación. Ajustador de base de chumacera	Escala 2:1
						Número de dibujo 01.01.12	
EDI:	MOD:	FECHA:	NOM:				

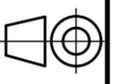


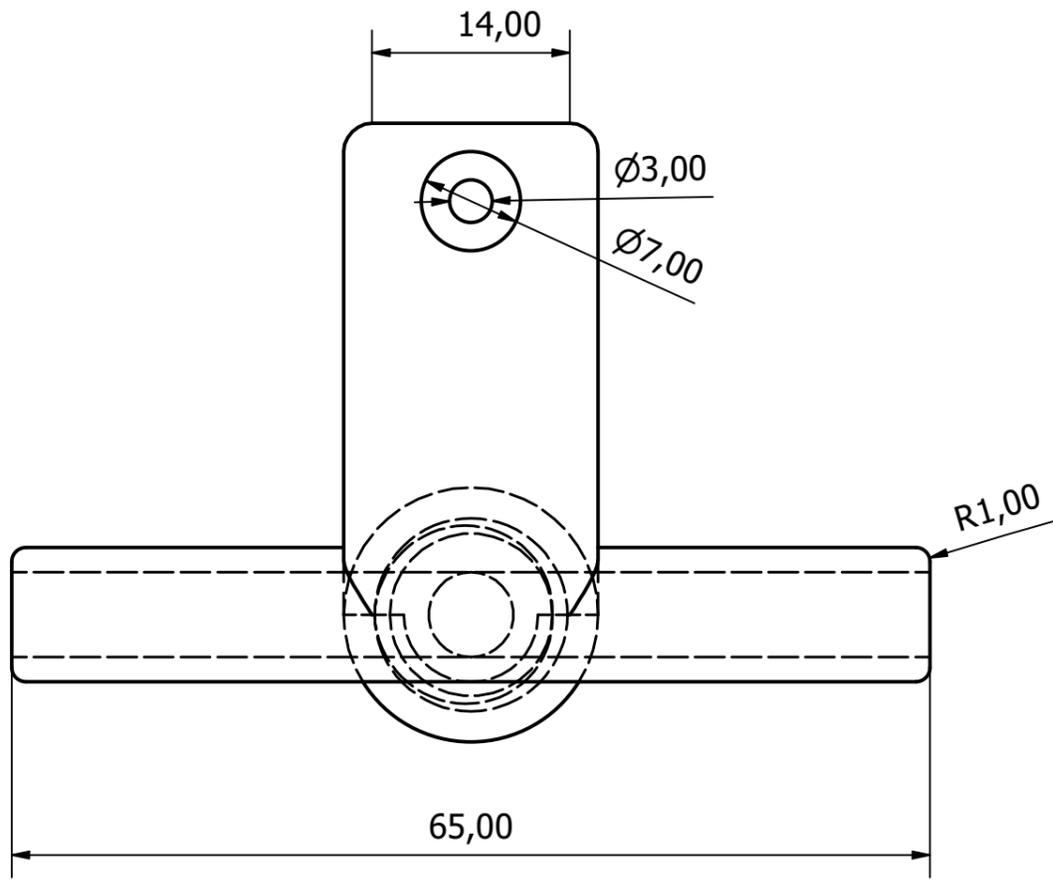
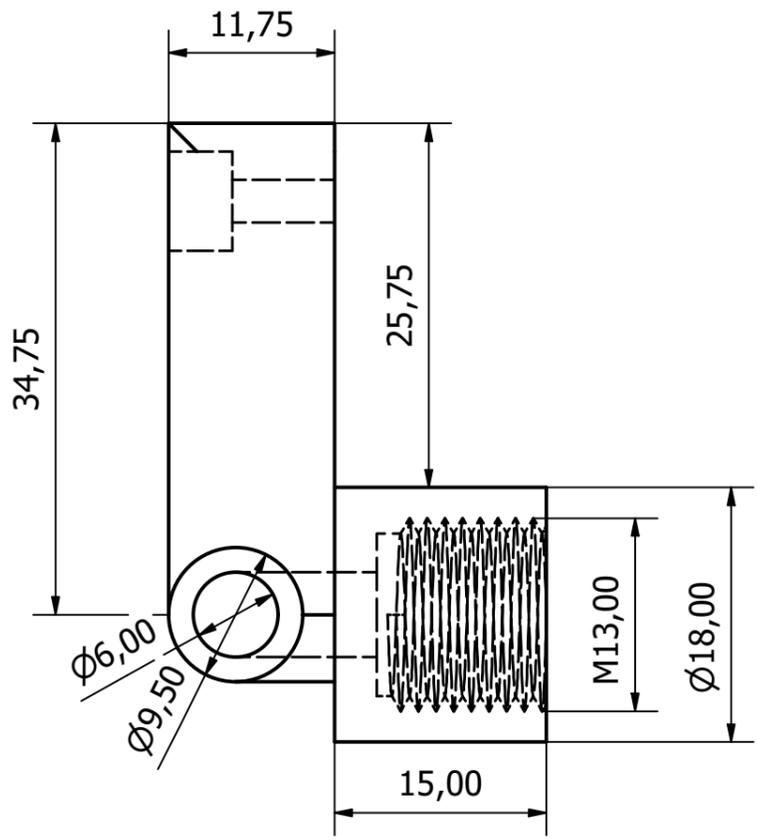
No. de pieza	Denominación	N. de norma o dibujo	Material	No. de orden	Número del modelo	Peso	Observaciones
1	Base de bomba	INEN 003		1	01.02.01		PLA
2	Tubo de descarga, tramo 1	INEN 003		1			PVC
3	Acople para sensor de presión	INEN 003		1	01.02.02		PLA
4	Tubo de descarga, tramo 2	INEN 003		1			PVC
5	Tubo de carga	INEN 003		1			PVC
6	Volante de llave	INEN 003		1	01.02.03		PLA
7	Graduador de apertura de llave	INEN 003		1	01.02.04		PLA
8	Tubo de descarga, tramo 3	INEN 003		1			PVC
9	Llave 1/4	INEN 003		1			Latón
10	Conector de tanque	INEN 003		2	01.02.05		PVC
11	Base de recipiente para agua	INEN 003		1	01.02.06		PLA
12	Recipiente para agua	INEN 003		1			PP
13	Bomba	INEN 003		1			ABS
14	Banda GT2	INEN 003		1			Caucho, entre otros
15	Screw GB/T 822-2000 Type H M6 x 40:1	INEN 003		1			Acero Inoxidable
16	Base del sistema de tensado	INEN 003		1	01.02.07		Acero Galvanizado
17	Tuerca mariposa	INEN 003		1			Acero Galvanizado
18	BS EN 24032 - Métrico M6:23	INEN 003		7			Acero Estructural
19	Motor de bomba	INEN 003		1			Cobre, entre otros
20	Base de motor	INEN 003		1	01.02.08		Acero Galvanizado
21	AS 1427 H - Métrico M4 x 16:8	INEN 003		4			Acero Estructural
22	Polea de 35 dientes	INEN 003		2			Aluminio
23	Chumasera	INEN 003		2			Acero, entre otros
24	Base de chumasera	INEN 003		2	01.02.09		PLA
25	Base de encoder	INEN 003		1	01.02.10		PLA
26	Rueda de velocidad	INEN 003		1	01.02.11		PLA
27	Eje de bomba	INEN 003		1			Acero de Transmisión
28	CNS 4306 - AB M3,5 x 16:7	INEN 003		1			Acero Galvanizado
29	AS 1420 - Métrico M4 x 30:5	INEN 003		7			Acero Galvanizado
30	CNS 4306 - AB M5,5 x 25:2	INEN 003		11			Acero Galvanizado

Denominación		N. de norma o dibujo	Material	No. de orden	Número del modelo	Peso	Observaciones
Tolerancias		Peso		Material: Varios			
Fecha		Nombre		Denominación.			Escala
Dib.	31/07/2024	Marco Aguilar		Despiece sistema 2			1:2
Rev.		Ing. José Cuenca		Número de dibujo			
Aprob.		Ing. José Cuenca		001.02.00			
EDI:	MOD:	FECHA:	NOM:				

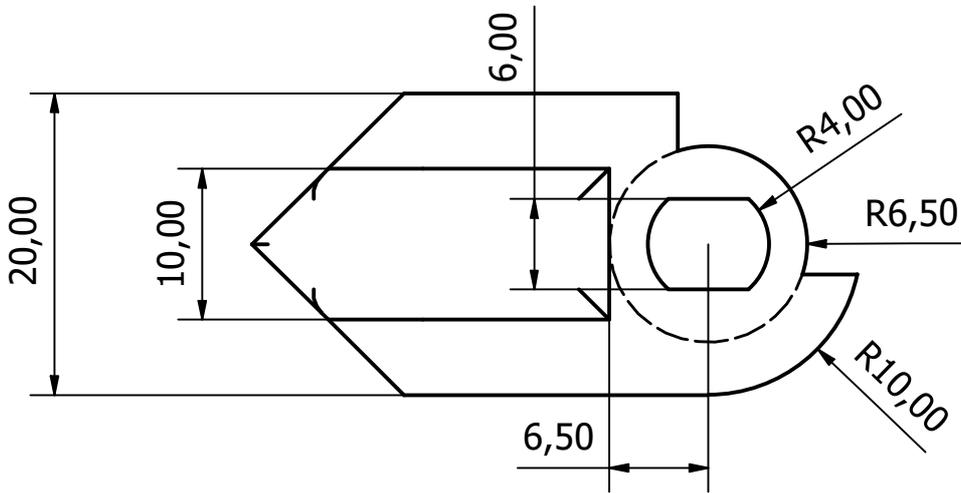
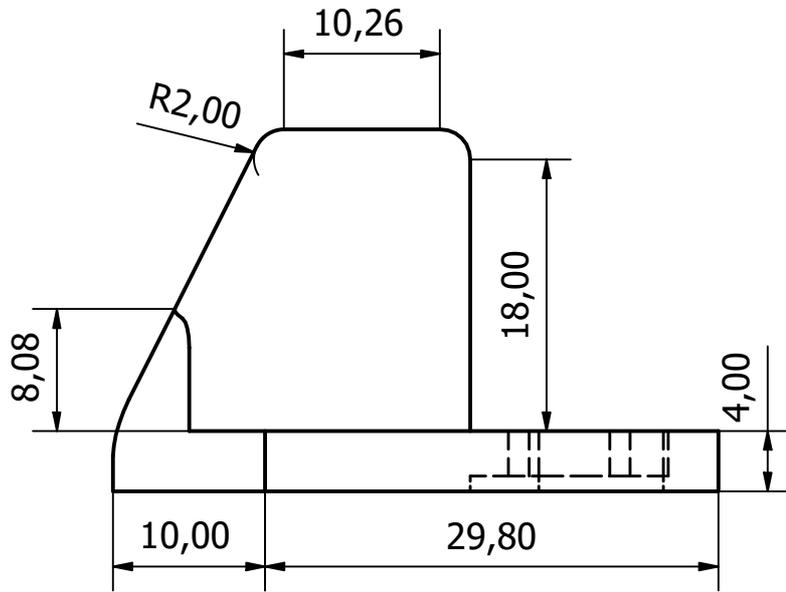


				Tolerancias	Peso	Material: PLA	
						Denominación. Base de bomba	Escala 1:1
						Número de dibujo 01.02.01	
EDI:	MOD:	FECHA:	NOM:				

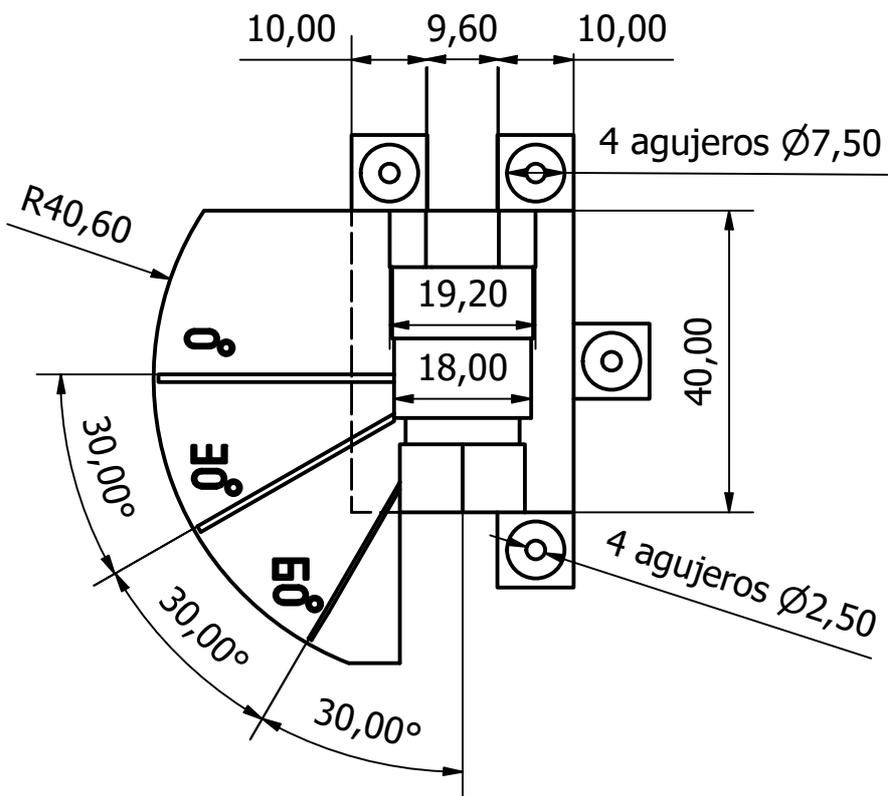
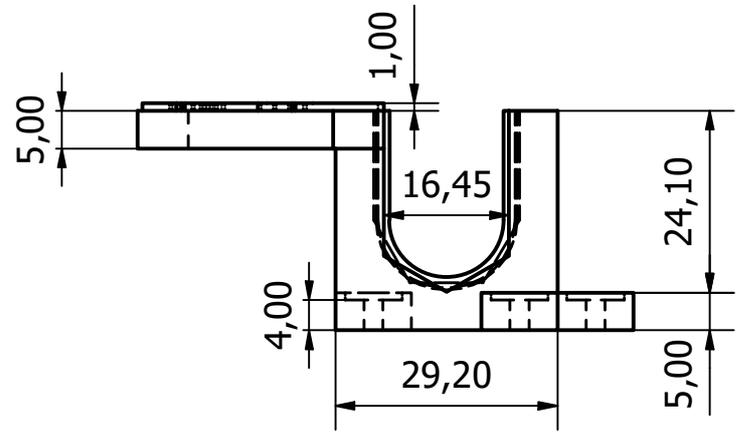




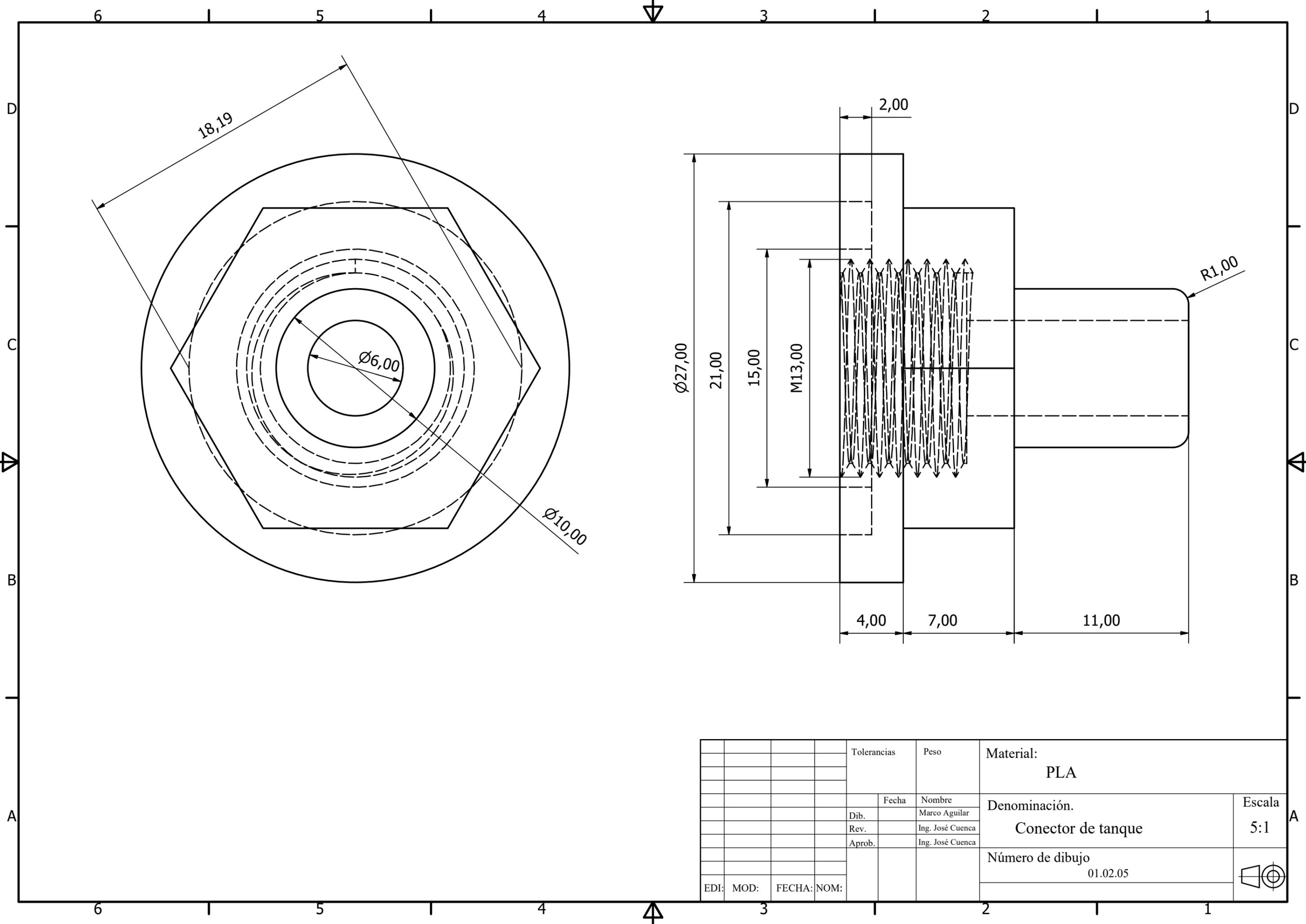
				Tolerancias	Peso	Material: PLA	
						Denominación. Acople para sensor de presión	Escala 2:1
						Número de dibujo 01.02.02	
EDI:	MOD:	FECHA:	NOM:				



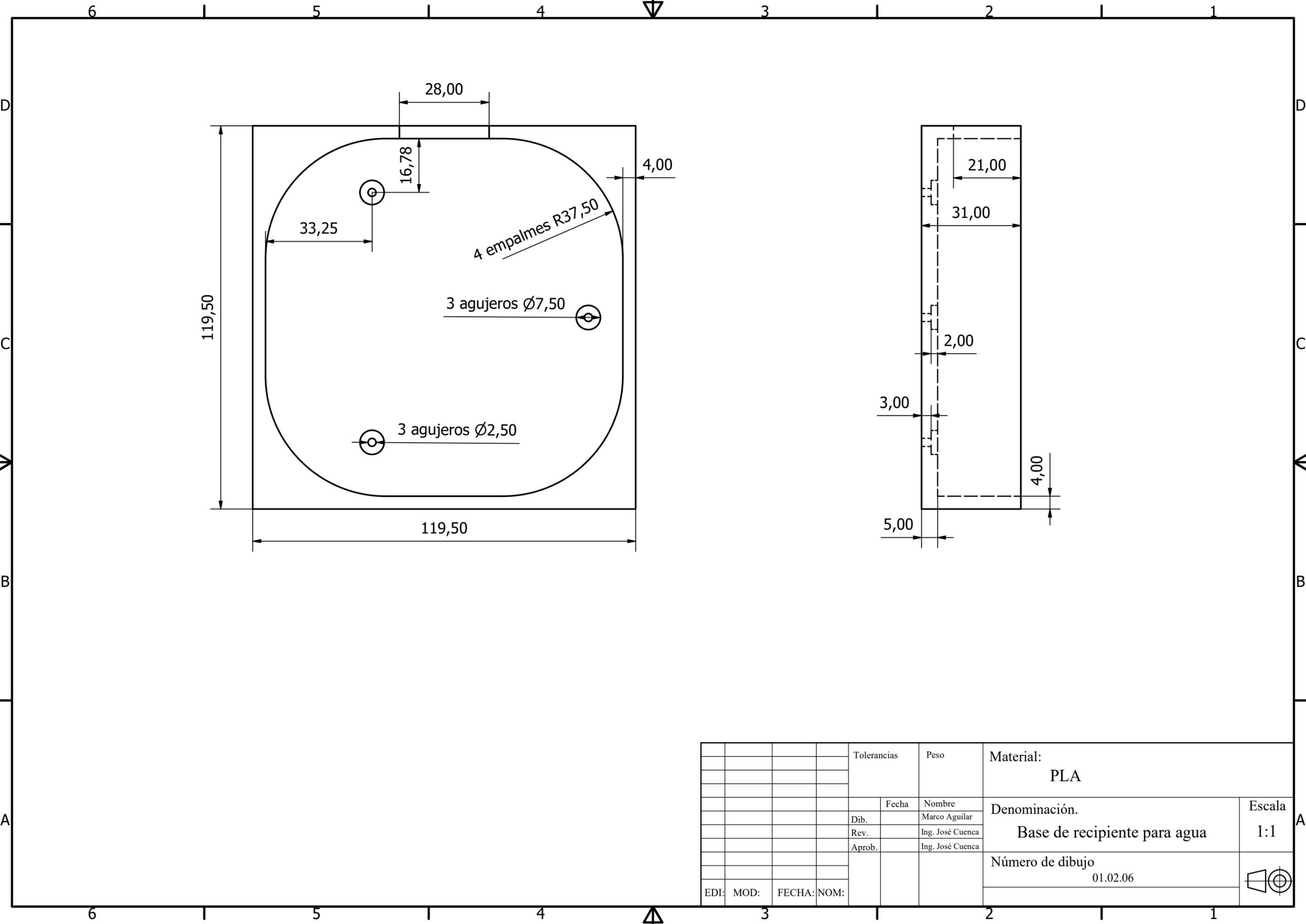
				Tolerancias	Peso	Material: PLA	
						Denominación. Bolante de llave	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.02.03	
EDI:	MOD:	FECHA:	NOM:				



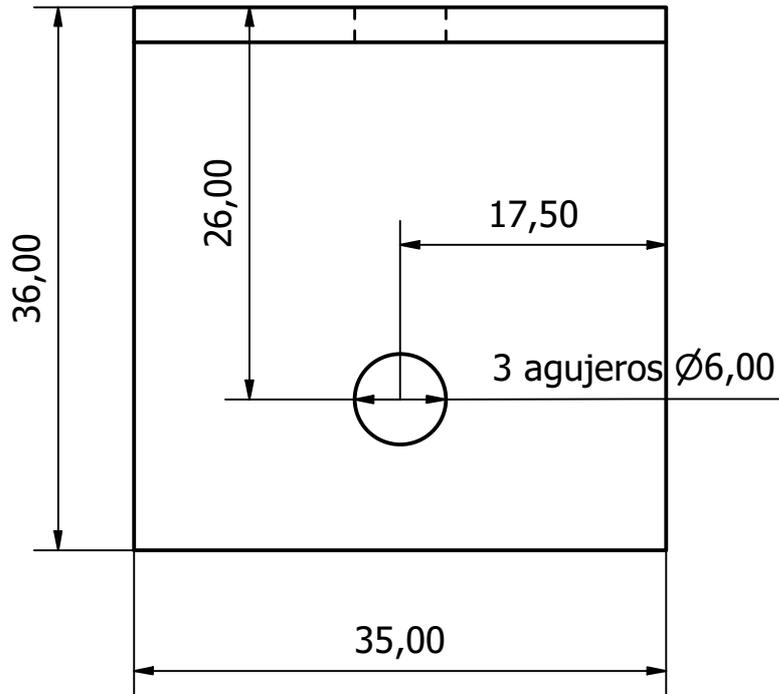
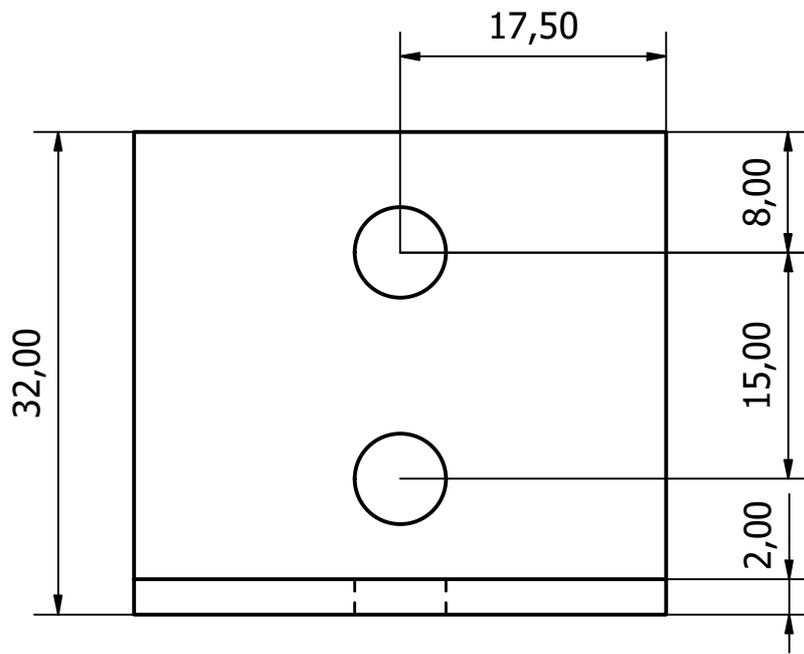
				Tolerancias	Peso	Material: PLA	
						Denominación. Graduador de apertura de llave	Escala 1:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.02.04	
EDI:	MOD:	FECHA:	NOM:				



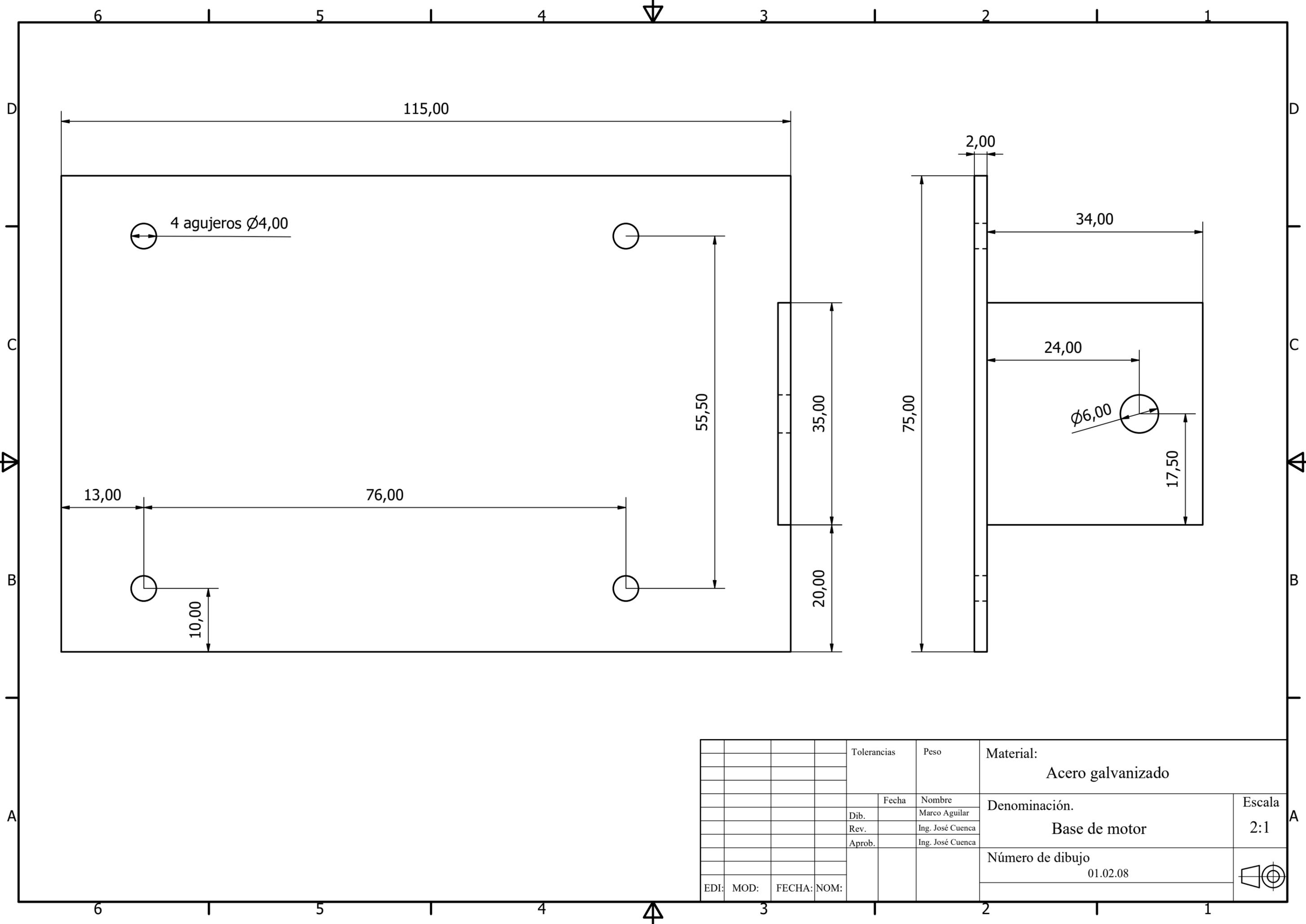
				Tolerancias	Peso	Material: PLA	
						Denominación. Conector de tanque	Escala 5:1
						Número de dibujo 01.02.05	
EDI:	MOD:	FECHA:	NOM:				



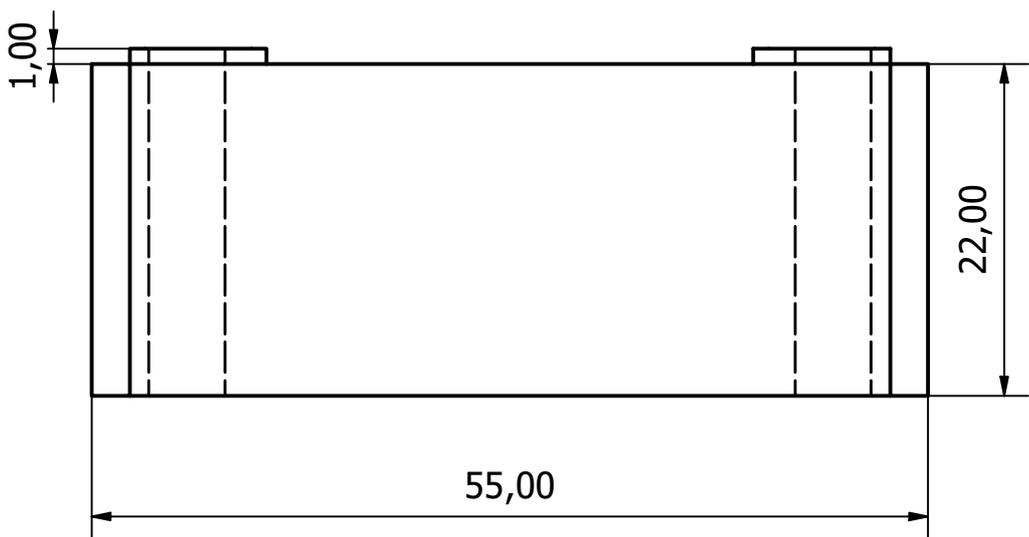
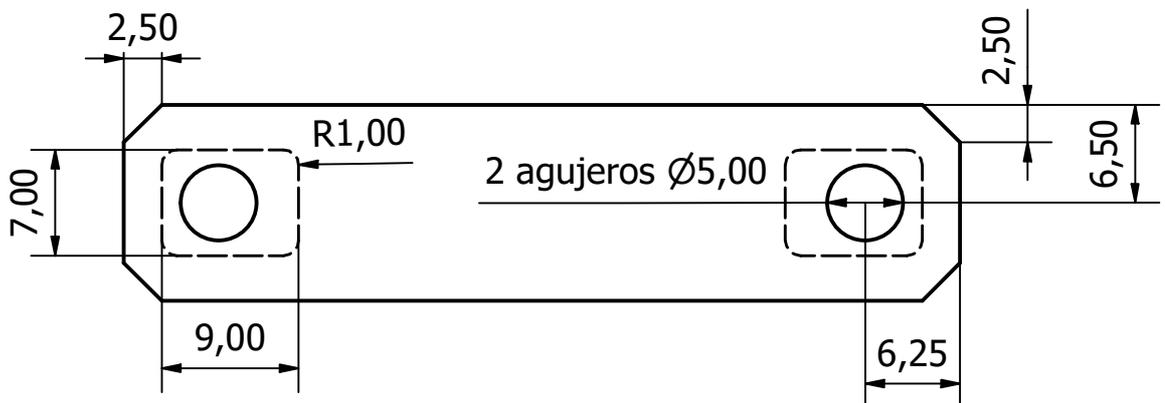
				Tolerancias	Peso	Material: PLA	
						Denominación. Base de recipiente para agua	Escala 1:1
						Número de dibujo 01.02.06	
EDI:	MOD:	FECHA:	NOM:				



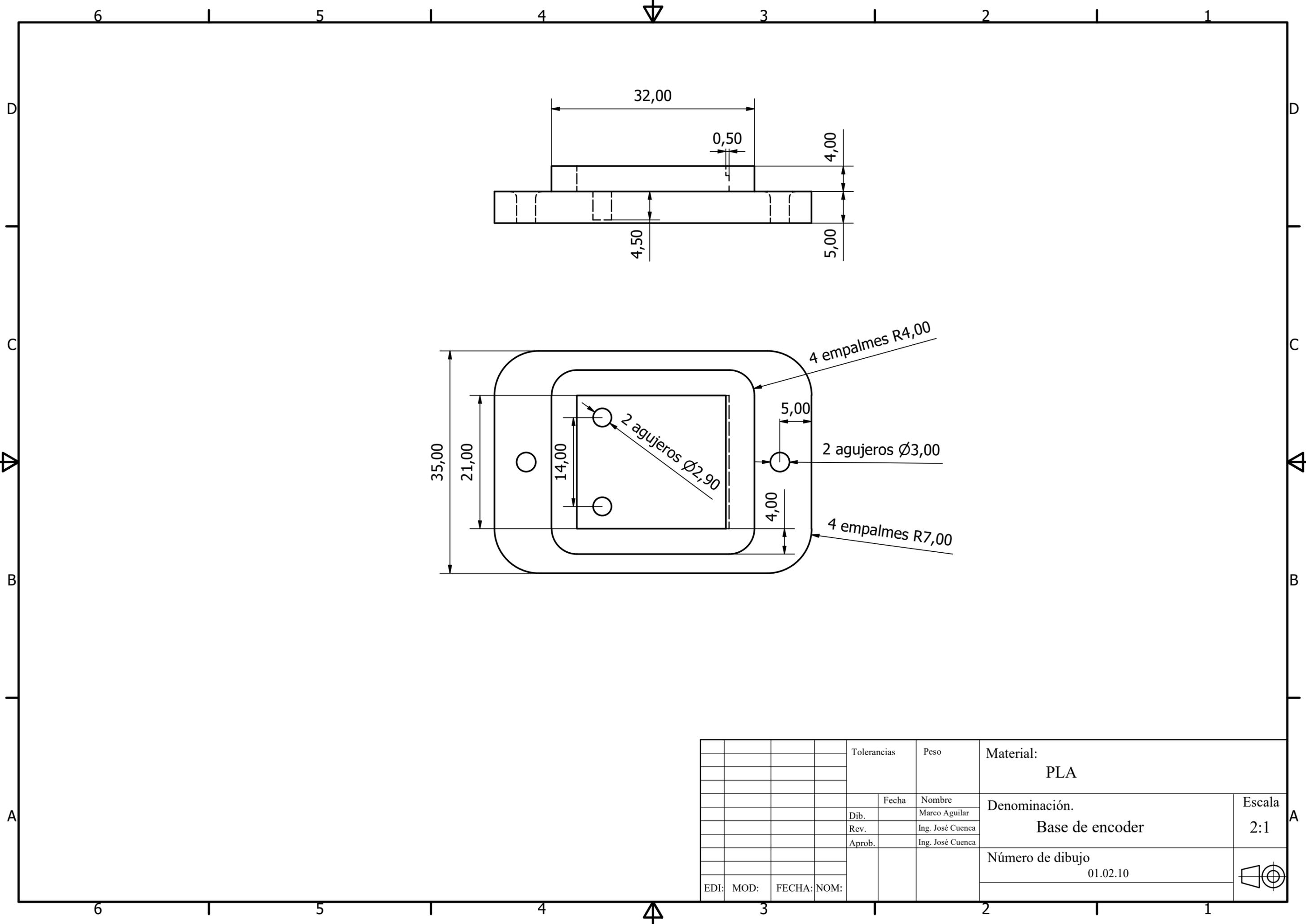
				Tolerancias	Peso	Material: Acero galvanizado	
						Denominación. Base de tensado	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.02.07	
EDI:	MOD:	FECHA:	NOM:				



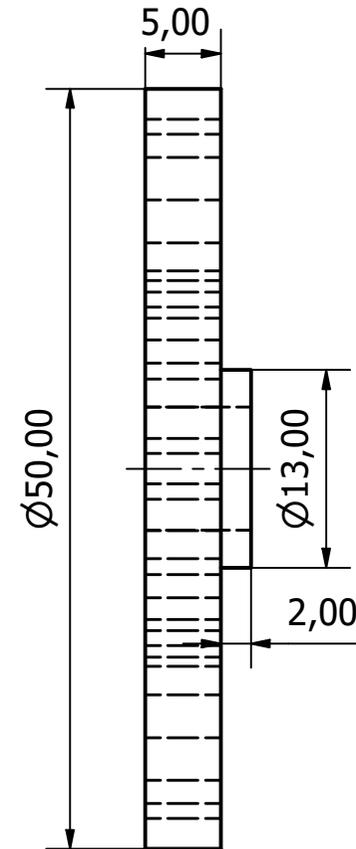
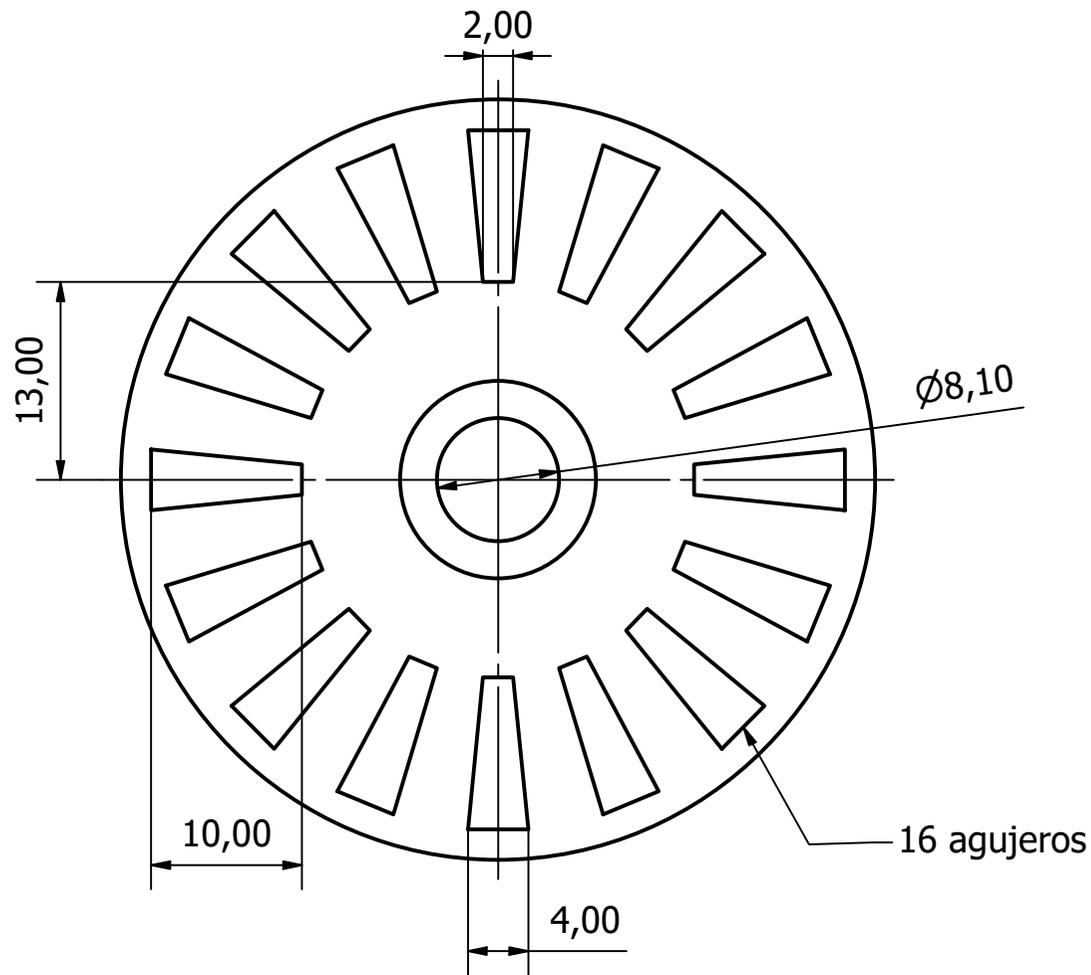
				Tolerancias	Peso	Material: Acero galvanizado	
						Denominación. Base de motor	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca	Número de dibujo 01.02.08	
				Aprob.	Ing. José Cuenca		
EDI:	MOD:	FECHA:	NOM:				



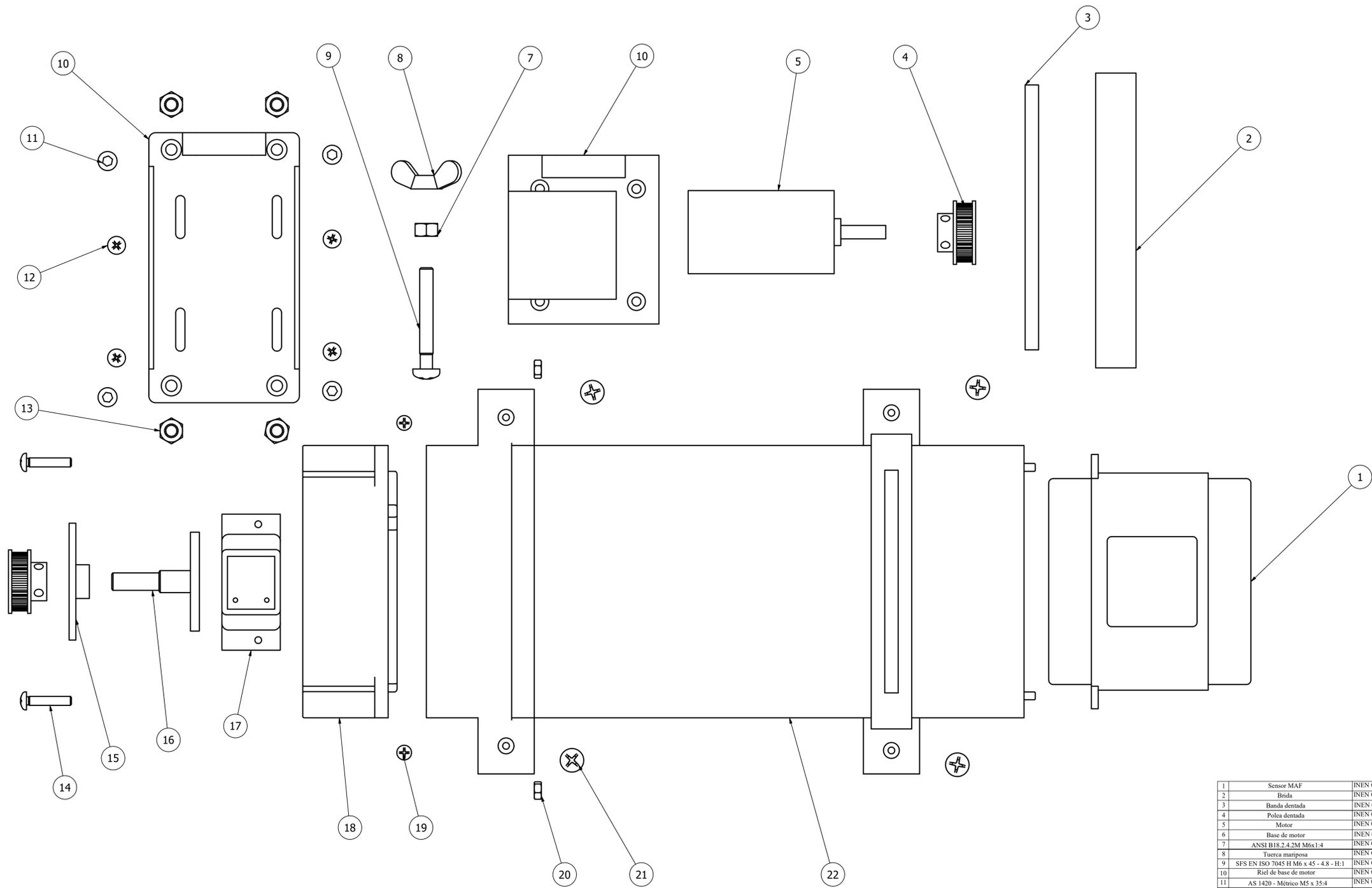
				Tolerancias	Peso	Material: PLA	
						Denominación. Base de chumacera	Escala 2:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca		
				Aprob.	Ing. José Cuenca		
						Número de dibujo 01.01.09	
EDI:	MOD:	FECHA:	NOM:				



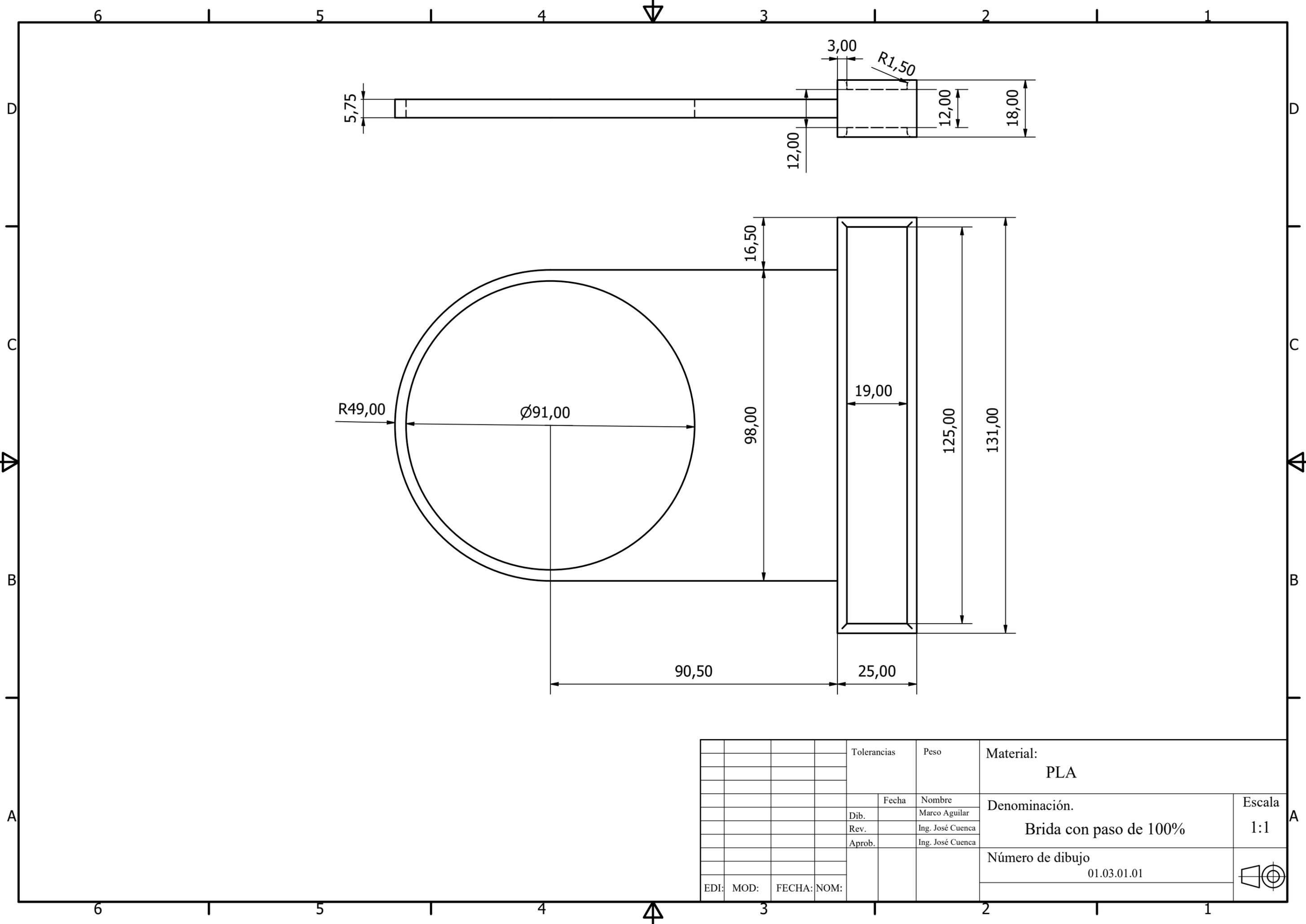
				Tolerancias	Peso	Material: PLA	
						Denominación: Base de encoder	Escala 2:1
						Número de dibujo 01.02.10	
EDI:	MOD:	FECHA:	NOM:				



					Tolerancias	Peso	Material: PLA	
							Denominación. Rueda de velocidad	Escala 2:1
					Fecha	Nombre		
				Dib.		Marco Aguilar		
				Rev.		Ing. José Cuenca		
					Aprob.	Ing. José Cuenca		
							Número de dibujo 01.02.11	
EDI:	MOD:	FECHA:	NOM:					



1	Sensor MAF	INEN 003	1			ABS
2	Brida	INEN 003	4	01.03.01		PLA
3	Banda dentada	INEN 003	9			Caucho, entre otros
4	Polea dentada	INEN 003	2			Aluminio
5	Motor	INEN 003	1			Acero, entre otros
6	Base de motor	INEN 003	1	01.03.02		PLA
7	ANSI B18.2.4.2M M6x1.4	INEN 003	1			Acero Galvanizado
8	Tuerca manposa	INEN 003	1			Acero Galvanizado
9	SFS EN ISO 7045 H M6 x 45 - 4.8 - H:1	INEN 003	1			Acero Galvanizado
10	Riel de base de motor	INEN 003	4	01.03.03		PLA
11	AS 1420 - Métrico M5 x 35:4	INEN 003	4			Acero Galvanizado
12	AS 1427 H - Métrico M4 x 16:6	INEN 003	4			Acero Galvanizado
13	BS EN 24032 - Métrico M6:24	INEN 003	4			Acero Galvanizado
14	AS 1427 Z - Métrico M4 x 20:4	INEN 003	4			Acero Galvanizado
15	Rueda de velocidad	INEN 003	1	01.01.04		PLA
16	Eje de ventilador	INEN 003	1	01.01.05		PLA
17	Base de encoder	INEN 003	1	01.01.06		PLA
18	Ventilador	INEN 003	1			PLA
19	CNS 4306 - AB M3,5 x 16:3	INEN 003	2			Acero Galvanizado
20	BS EN 24032 - Métrico M4:3	INEN 003	4			Acero Galvanizado
21	CNS 4306 - AB M5,5 x 25:3	INEN 003	4			Acero Galvanizado
22	Ducto de ventilación	INEN 003	1	01.01.07		PLA
Denominación.		N. de motor o dibujo	Material	Número del modelo	Peso	Observaciones
		Tolerancias	Peso	Material:		
				Varios		
		Fecha	Nombre	Denominación.		Escala
		31/07/2024	Marco Aguilar	Despiece sistema 3		1:2
		Rev.	Ing. José Cuenca	Número de dibujo		
		Approb.	Ing. José Cuenca	001.03.00		
EDI:	MOD:	FECHA:	NOM:			



R49,00

Ø91,00

16,50

98,00

19,00

125,00

131,00

90,50

25,00

3,00

R1,50

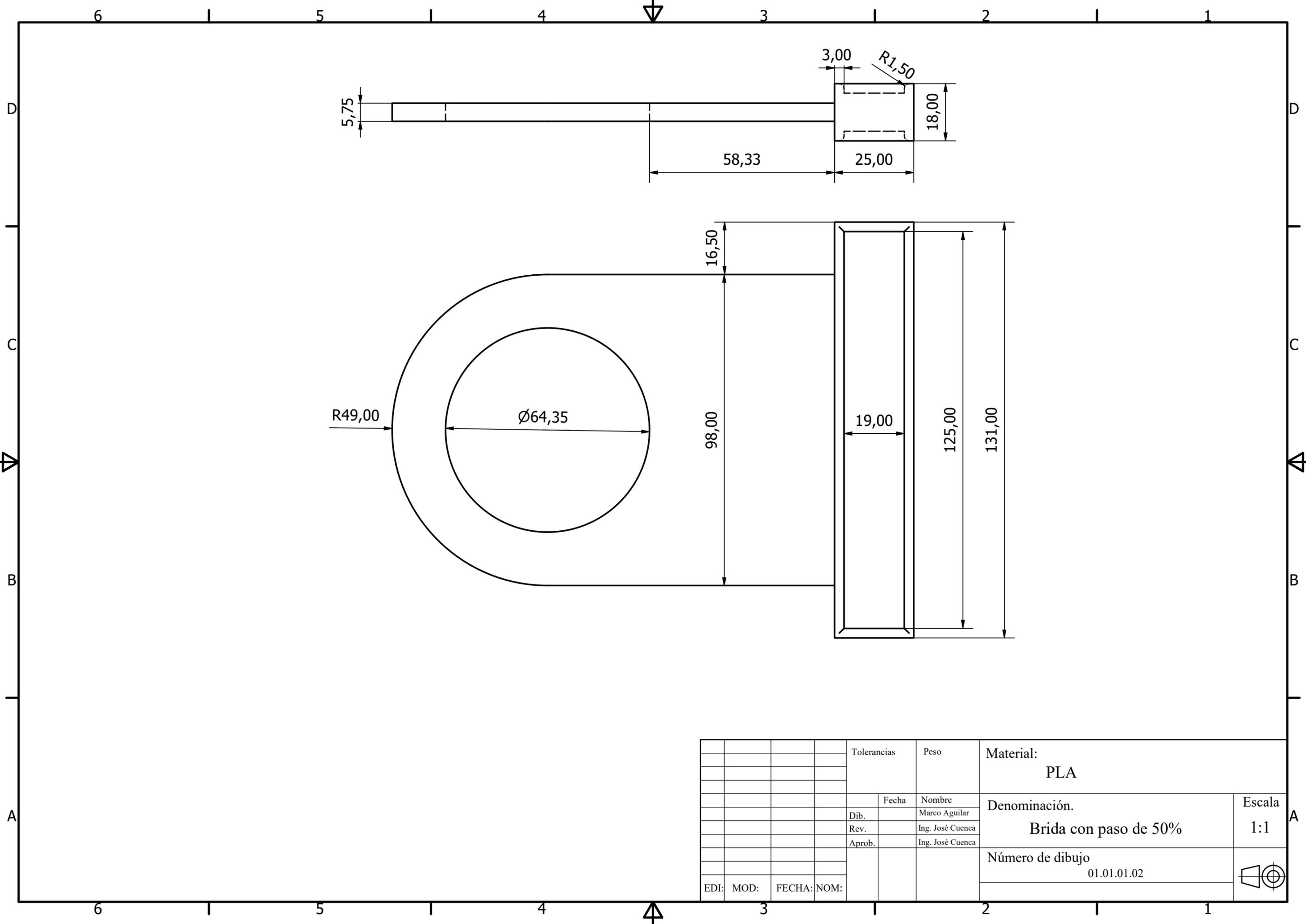
12,00

18,00

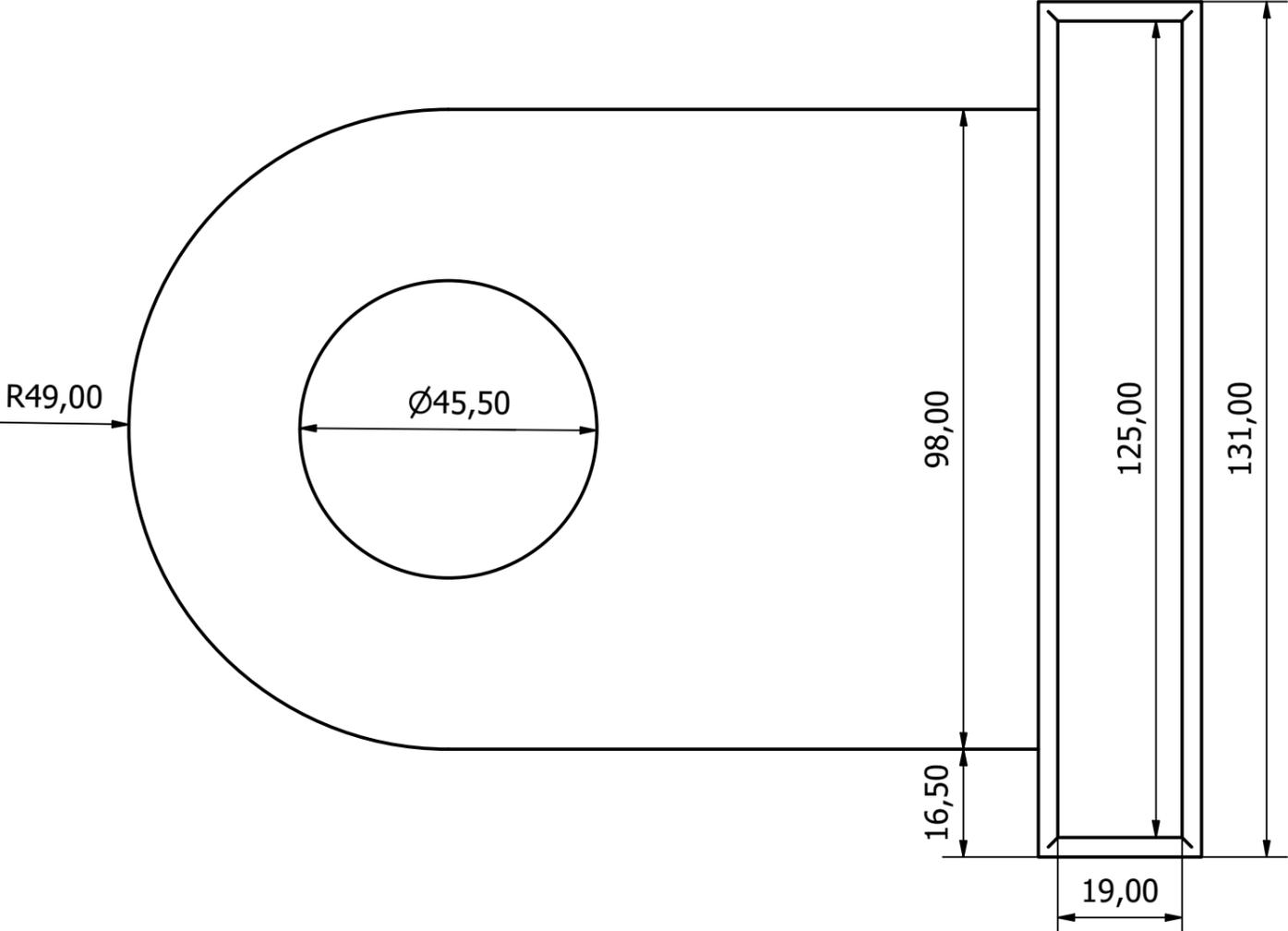
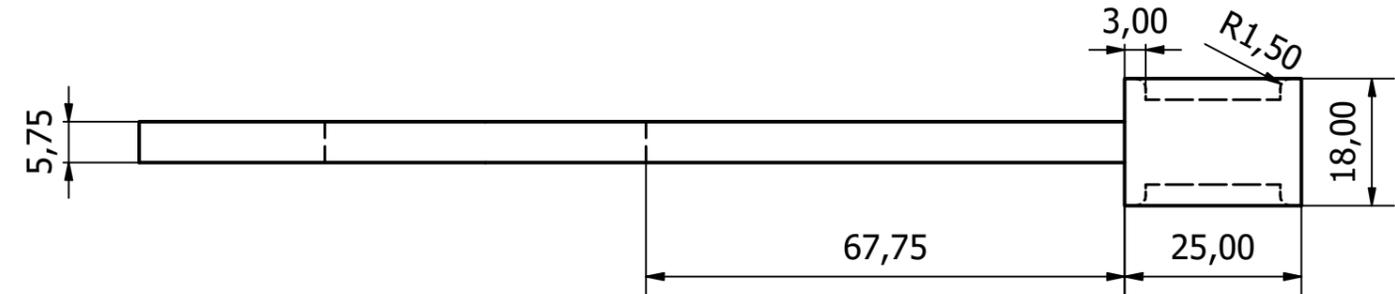
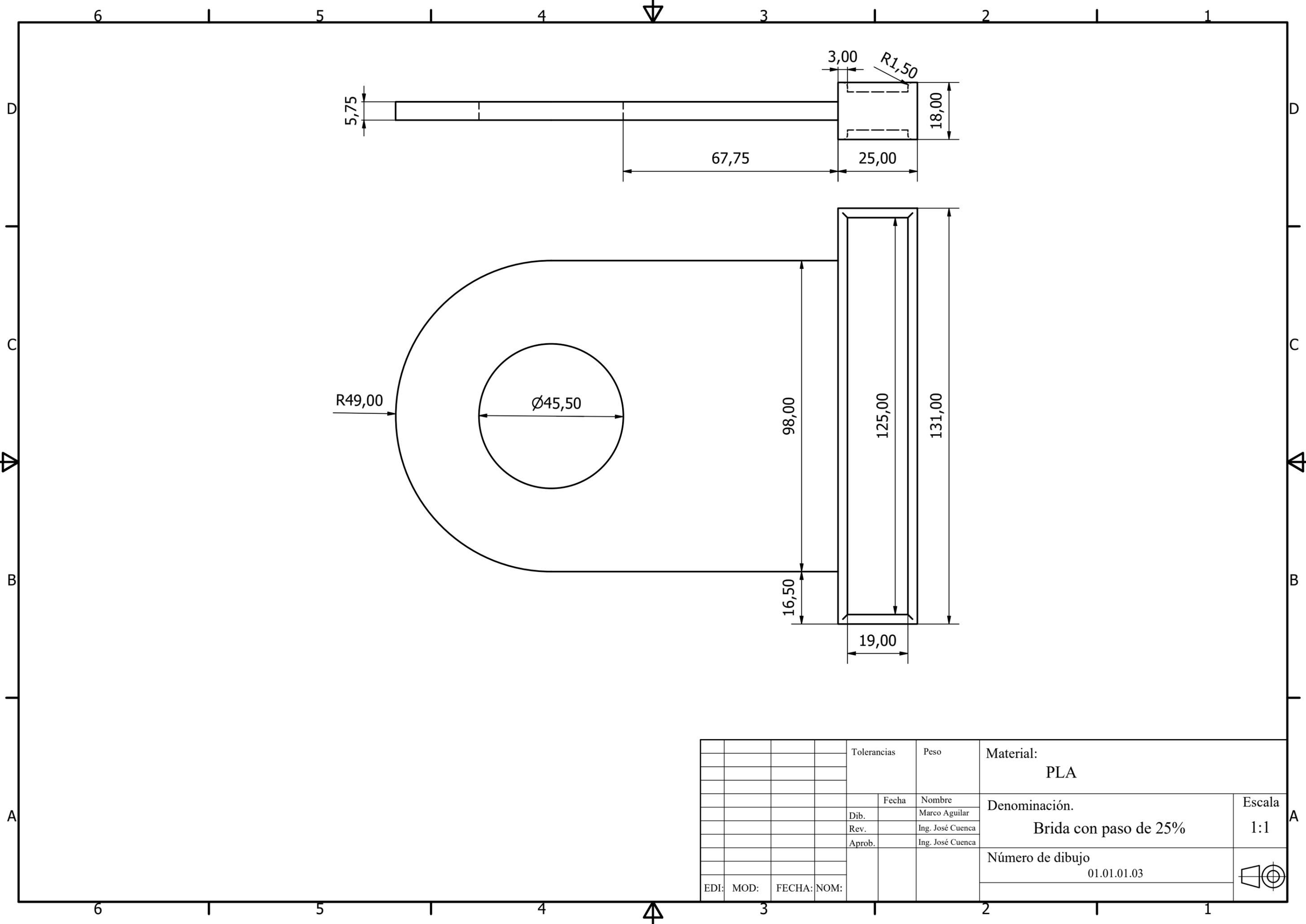
5,75

12,00

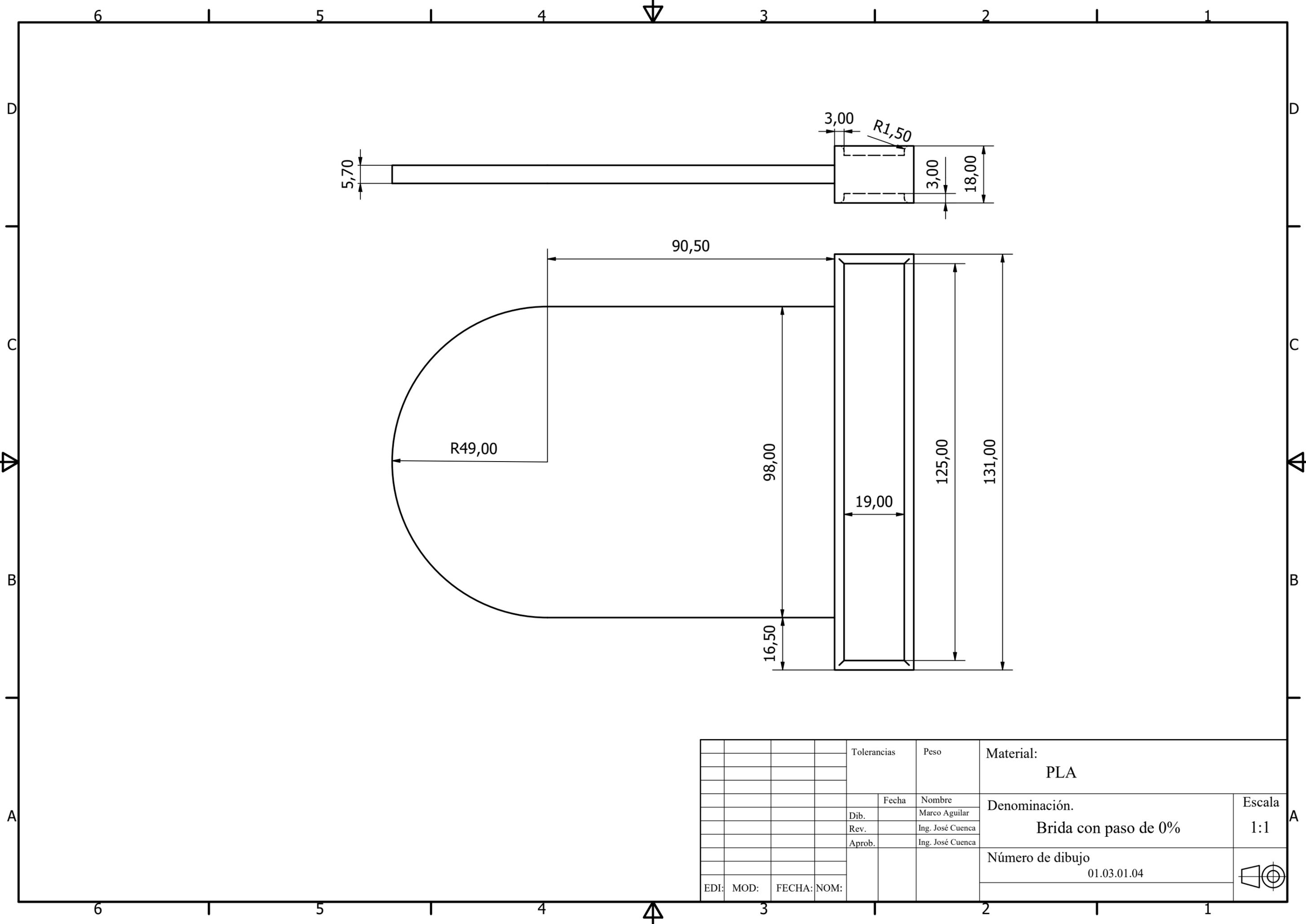
				Tolerancias	Peso	Material: PLA	
						Denominación. Brida con paso de 100%	Escala 1:1
						Número de dibujo 01.03.01.01	
EDI:	MOD:	FECHA:	NOM:				



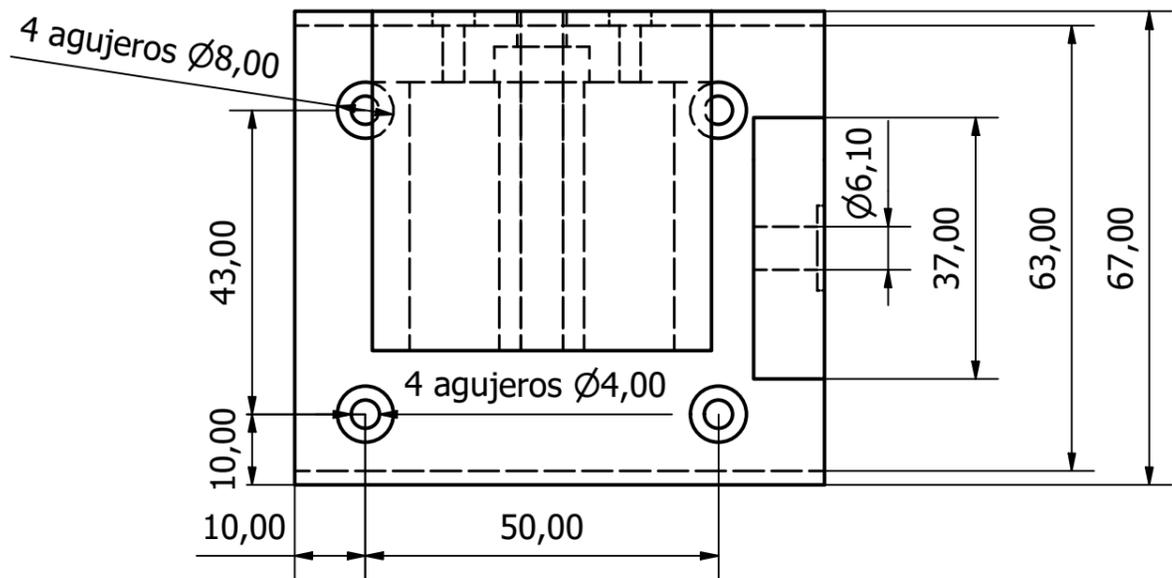
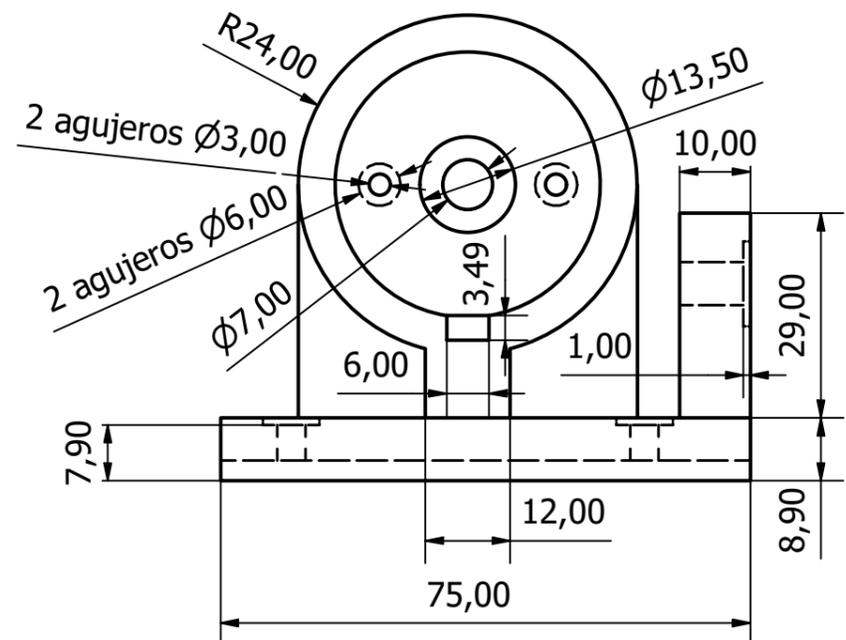
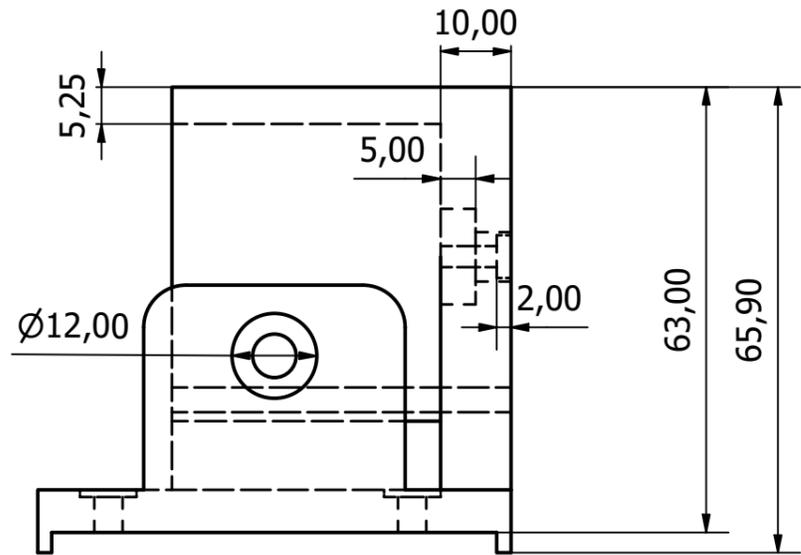
				Tolerancias	Peso	Material: PLA	
						Denominación. Brida con paso de 50%	Escala 1:1
						Número de dibujo 01.01.01.02	
EDI:	MOD:	FECHA:	NOM:				



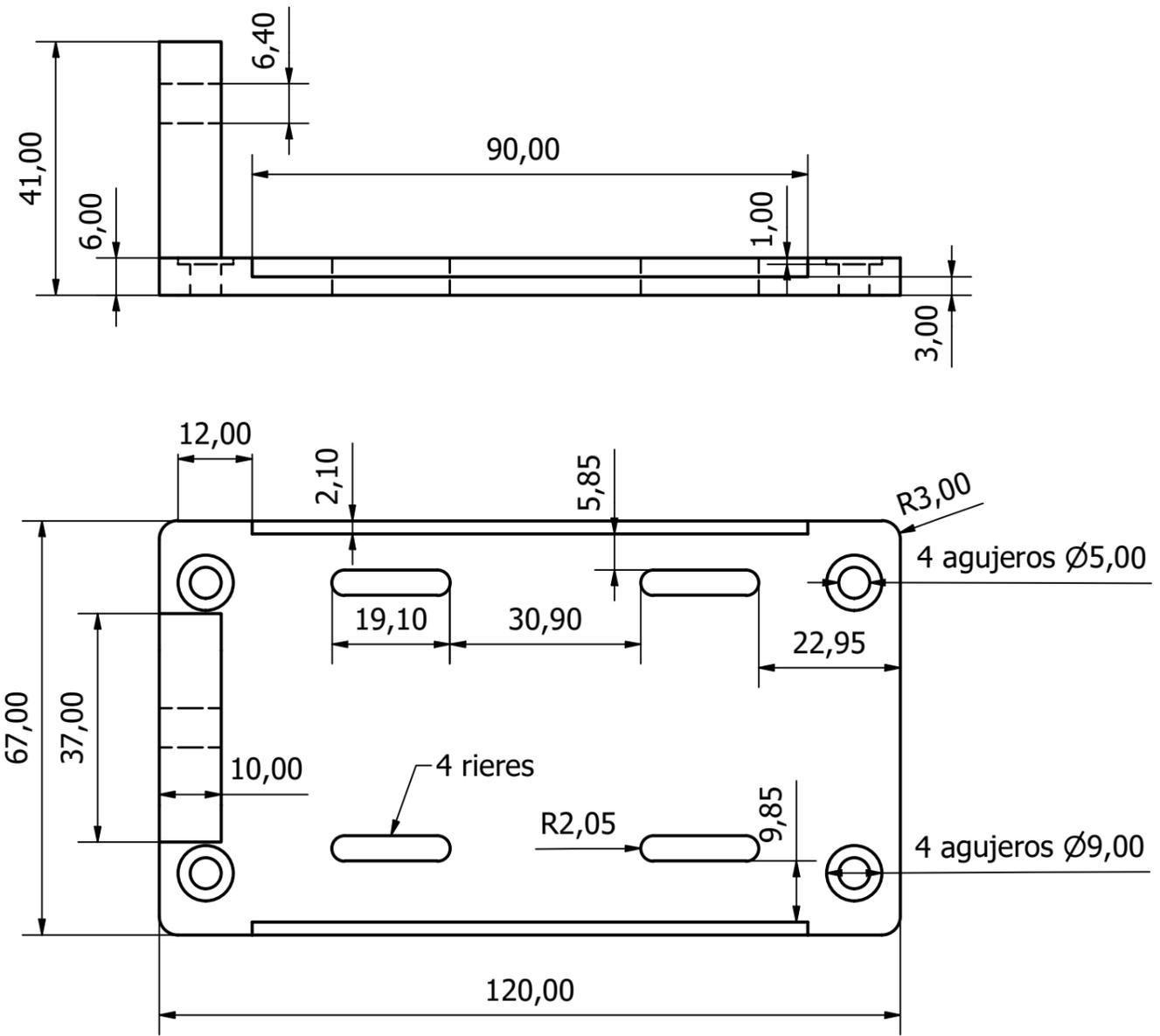
				Tolerancias	Peso	Material: PLA	
						Denominación. Brida con paso de 25%	Escala 1:1
						Número de dibujo 01.01.01.03	
EDI:	MOD:	FECHA:	NOM:				



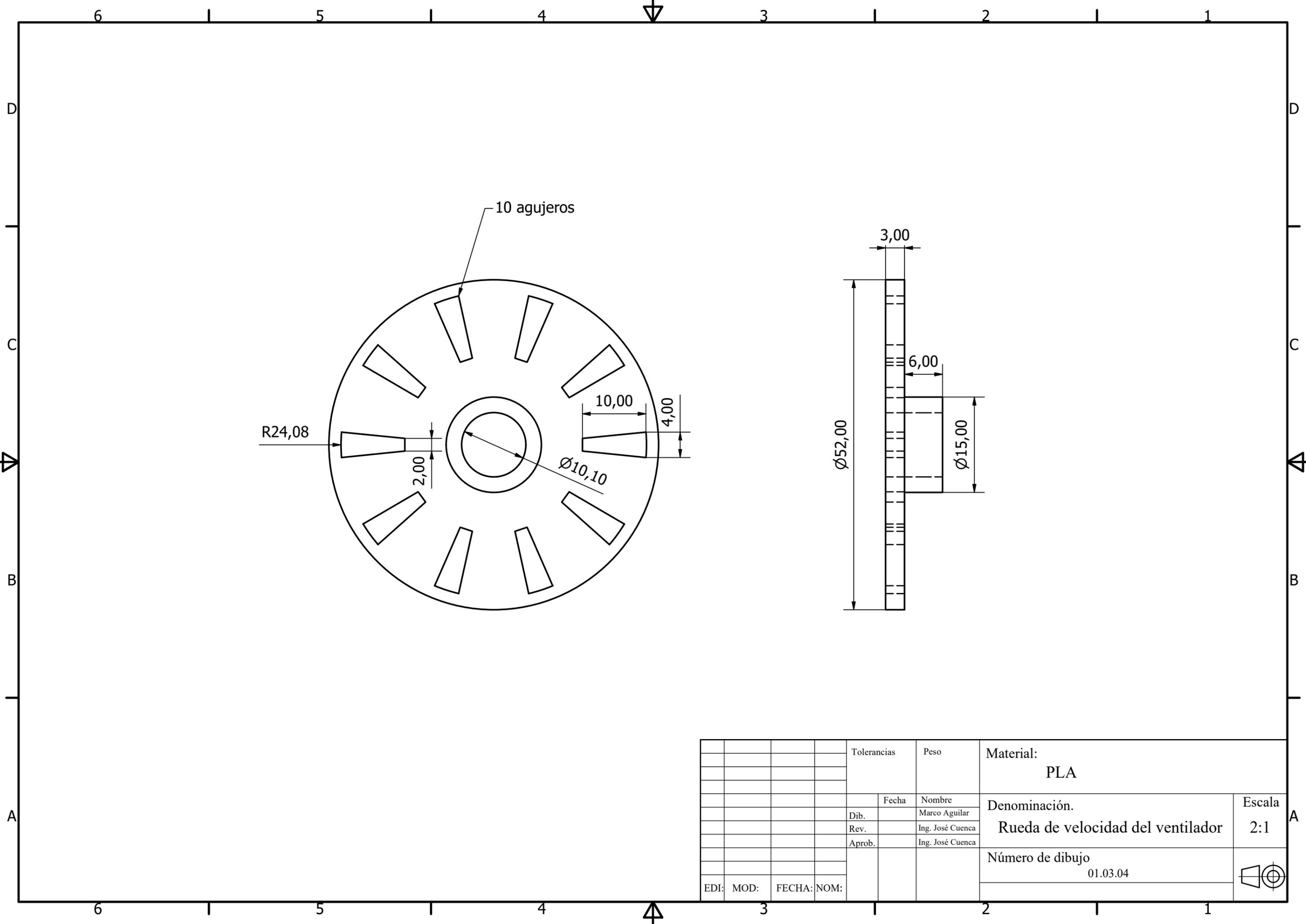
				Tolerancias	Peso	Material: PLA	
						Denominación. Brida con paso de 0%	Escala 1:1
						Número de dibujo 01.03.01.04	
EDI:	MOD:	FECHA:	NOM:				



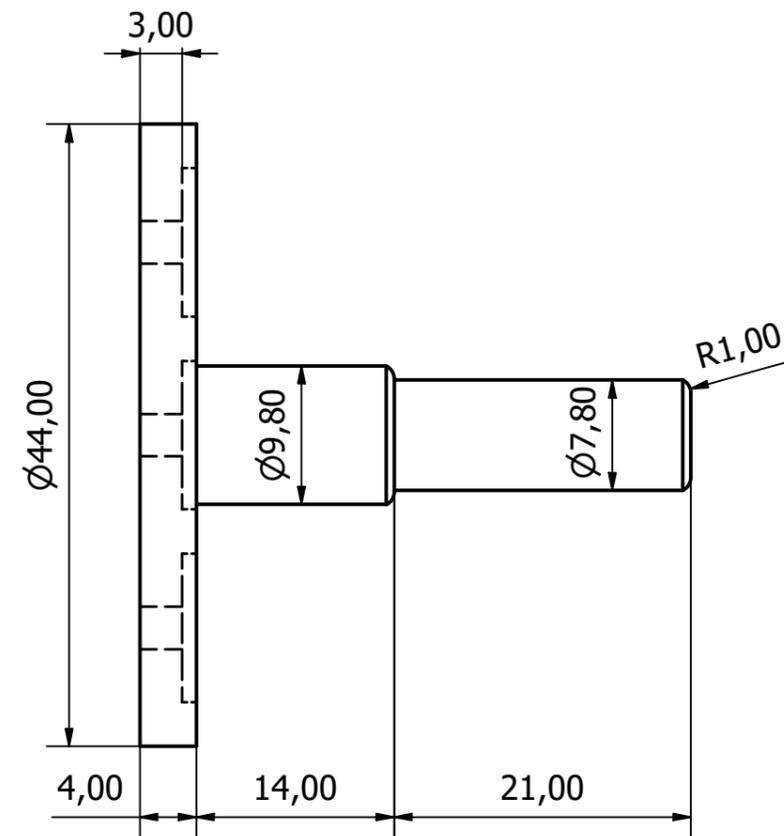
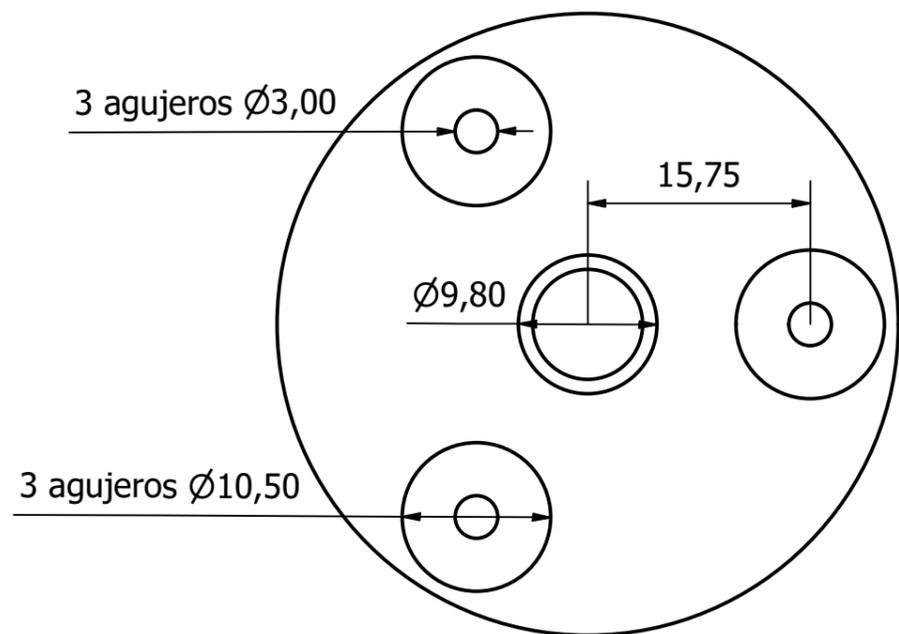
				Tolerancias	Peso	Material: PLA	
						Denominación. Base de motor del ventilador	Escala 1:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca	Número de dibujo 01.03.02	
				Aprob.	Ing. José Cuenca		
EDI:	MOD:	FECHA:	NOM:				



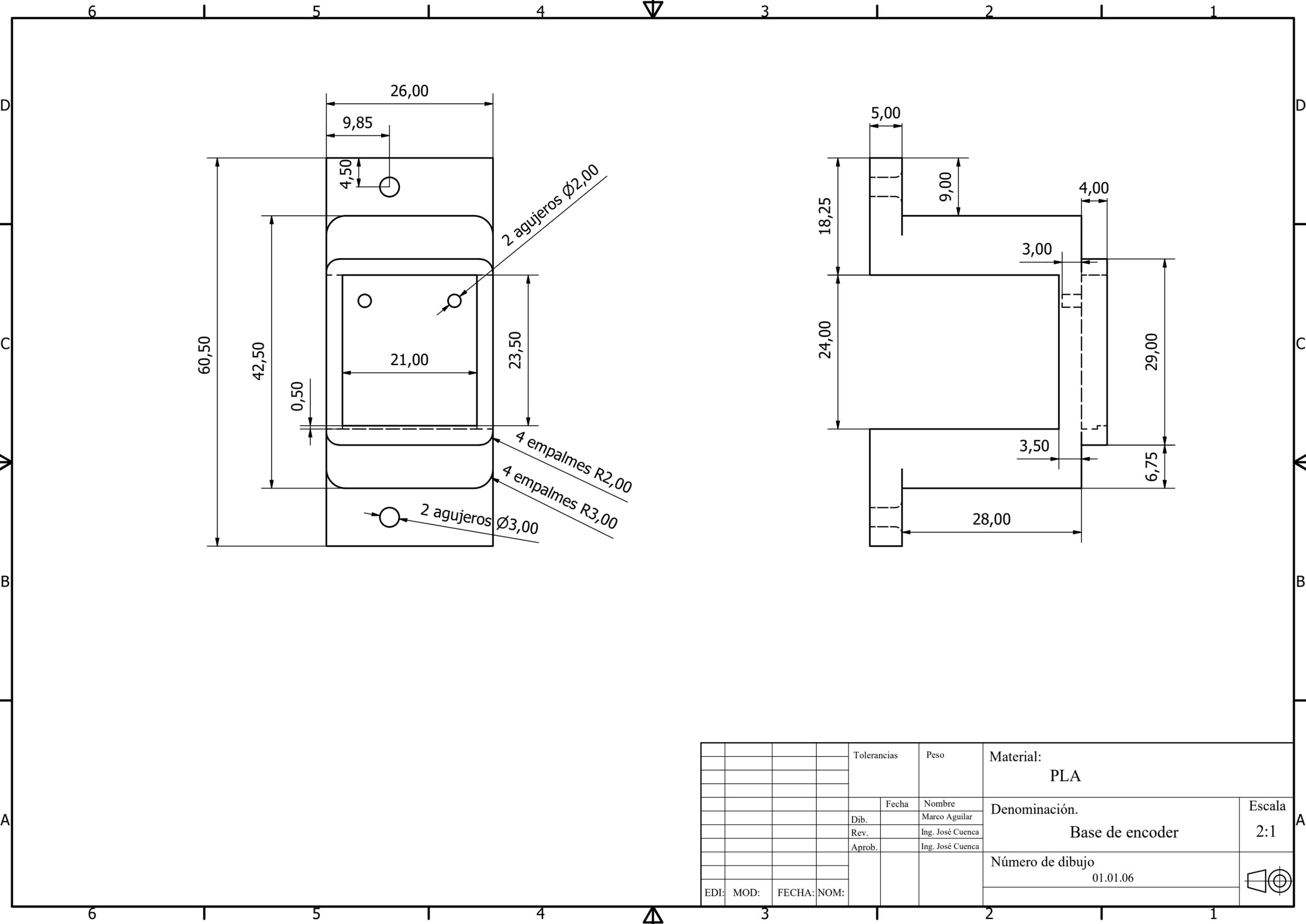
				Tolerancias	Peso	Material: PLA	
						Denominación. Riel de base motor	Escala 1:1
				Fecha	Nombre		
				Dib.	Marco Aguilar		
				Rev.	Ing. José Cuenca	Número de dibujo 01.03.03	
				Aprob.	Ing. José Cuenca		
EDI:	MOD:	FECHA:	NOM:				



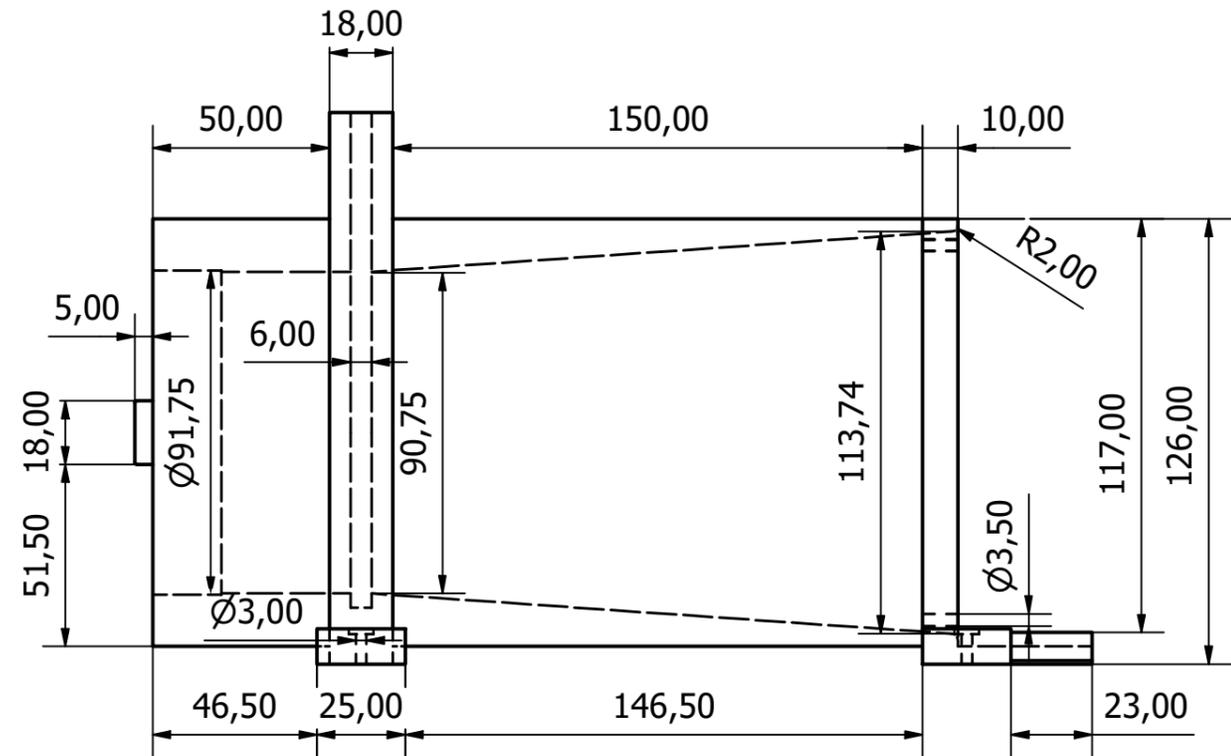
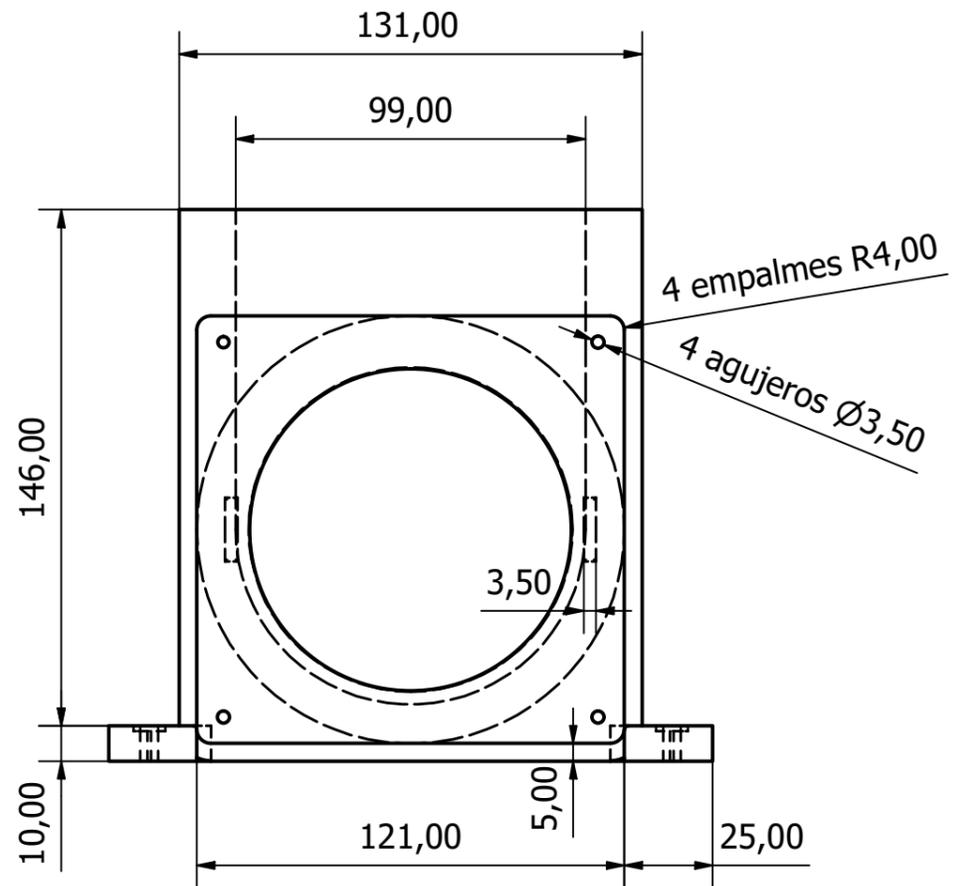
				Tolerancias	Peso	Material: PLA	
						Denominación. Rueda de velocidad del ventilador	Escala 2:1
						Número de dibujo 01.03.04	
EDI:	MOD:	FECHA:	NOM:				



				Tolerancias	Peso	Material: PLA	
						Denominación. Eje de ventilador	Escala 2:1
						Número de dibujo 01.01.05	
EDI:	MOD:	FECHA:	NOM:				



				Tolerancias	Peso	Material: PLA	
						Denominación. Base de encoder	Escala 2:1
						Número de dibujo 01.01.06	
EDI:	MOD:	FECHA:	NOM:				



				Tolerancias	Peso	Material:	
						PLA	
					Fecha	Nombre	Escala
				Dib.		Marco Aguilar	1:2
				Rev.		Ing. José Cuenca	
				Aprob.		Ing. José Cuenca	
						Número de dibujo	
						01.03.07	
EDI:	MOD:	FECHA:	NOM:				