

# **UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ**

**EXTENSIÓN MULTIDISCIPLINARIA DE CIUDAD UNIVERSITARIA**

**Instituto de Ingeniería y Tecnología**

## **Diseño de un sistema hardware in the loop para un péndulo invertido sobre un riel**

**Ingeniería Mecatrónica**

### **Autores:**

Jose Antonio González Sánchez 201640

Daniel Adrián Pizarro Medina 214634

Samuel Alejandro Mireles Nava 215761

Jostin Osiel Medina Valerio 216002

Brayan Daniel Rivas Parra 215163

**Profesor:** Dr. Francesco José García Luna

Abril 2025

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivo general</b>	<b>4</b>
<b>3. Objetivos particulares</b>	<b>4</b>
<b>4. Materiales</b>	<b>4</b>
<b>5. Desarrollo</b>	<b>5</b>
5.1. Modelo . . . . .	5
5.2. Cinematica directa . . . . .	6
5.3. Modelo Dinamico . . . . .	8
5.4. Diseño de la placa de circuito impreso . . . . .	10
5.4.1. Componentes de la placa de circuito impreso . . . . .	10
5.5. Ros2 jazzy y Gazebo harmonic . . . . .	13
5.5.1. Visualizacion con RVIZ . . . . .	13
5.5.2. Simulacion en Gazebo . . . . .	13
5.5.3. Gazebo plugging . . . . .	14
5.5.4. Puente entre Gazebo y Rviz . . . . .	14
5.5.5. Micro-Ros . . . . .	15
<b>Appendices</b>	<b>17</b>
<b>A. Apendice</b>	<b>17</b>
A.1. Modelo de solid works (renderizados) y planos . . . . .	20

## Índice de figuras

1.	Pendulo de furuta lineal . . . . .	5
2.	Marcos de Referencia . . . . .	6
3.	Diagrama pictorico . . . . .	10
4.	Diagrama esquematico . . . . .	11
5.	Diseño de placa de circuito impreso . . . . .	12
6.	Visualizacion en RVIS . . . . .	13
7.	Simulacion en Gazebo . . . . .	13
8.	Masa del pendulo . . . . .	20
9.	Varilla del pendulo . . . . .	20
10.	Dispositivo Movil . . . . .	20
11.	Barra deslizante . . . . .	20
12.	Eje de las poleas . . . . .	20
13.	Riel del chasis . . . . .	20
14.	Placa que no sostiene el motor . . . . .	21
15.	Placa que sostiene el motor . . . . .	21
16.	Placa de los extremos . . . . .	21
17.	Polea . . . . .	21
18.	Motor . . . . .	21
19.	Primera parte del pendulo, base principal . . . . .	22
20.	Segunda parte del pendulo, dispositivo movil . . . . .	22
21.	Plano del Elemento Movil . . . . .	23
22.	Plano de la barilla guia . . . . .	24
23.	Plano del eje de las poleas . . . . .	25
24.	Plano de la masa del extremo del pendulo . . . . .	26
25.	Plano de la barilla pendulo . . . . .	27
26.	Plano de la placa que sostiene el motor. . . . .	28
27.	Plano del riel del chasis del pendulo de furuta lineal. . . . .	29
28.	Plano de la placa no motriz. . . . .	30
29.	Plano de la polea. . . . .	31
30.	Plano del motor. . . . .	32
31.	Plano placa extremo. . . . .	33
32.	Ensamble pendulo. . . . .	34

## 1. Introducción

El péndulo de invertido es un sistema mecánico interesante por su comportamiento inestable, lo que lo convierte en un buen caso de estudio dentro del área de control y la robótica. Analizar este tipo de mecanismos permite aplicar conceptos clave como la robotica, dinámica y la retroalimentación, además de poner en practica habilidades en modelado, simulación y programación, que son importantes en la ingeniería[1].

En este proyecto se va a modelar el péndulo lineal en SolidWorks, con la finalidad de generar una carpeta URDF que nos permita visualizar el sistema dentro del entorno de ROS (Robot Operating System). Una vez hecho esto, se definirán los marcos de referencia de cada uno de los cuerpos del sistema, lo cual es necesario para desarrollar la cinemática directa[1].

El péndulo que vamos a trabajar tiene dos grados de libertad, por lo que será necesario resolver las transformaciones homogéneas de cada eslabón para obtener la matriz homogénea total. Esta matriz nos ayudará a entender la relación entre la posición y orientación del efecto final, con respecto al sistema de coordenadas de la base de la estructura del péndulo[1]. Todo este proceso nos servirá para tener una mejor comprensión del comportamiento del sistema, y además tener mas conocimientos para futuras implementaciones en control y simulaciones[1].

## 2. Objetivo general

Diseñar un sistema de control para estabilizar un péndulo invertido montado sobre un riel.

## 3. Objetivos particulares

- Documentar el diseño mecánico del sistema.
- Analizar la cinemática directa del sistema.
- Modelar la dinámica del péndulo invertido.
- Implementar el sistema en un entorno virtual.
- Evaluar la estabilidad del péndulo invertido.

## 4. Materiales

- Estructura mecánica: estructura metálica de riel extruido de aluminio, poleas dentadas, banda dentada, pieza móvil, guías de acero, barra de acero y pesa de acero.
- Motor encoder MG513P20-12V: motor de corriente continua equipado con un encoder incorporado<sup>[2]</sup>.
- Potenciómetro WDD35D4-5K: potenciómetro de precisión diseñado para aplicaciones que requieren una medición exacta de ángulos o posiciones. Presenta una resistencia nominal de 5 kiloohms<sup>[3]</sup>.
- Circuito impreso: circuito impreso (PCB) diseñado específicamente para comunicar la ESP-32 Devkit v1 con el MG513P20-12V mediante un puente H TB6612 y leer la posición del WDD35D4-5K.
- ESP-32 Devkit v1: controlador utilizado para mover el MG513P20-12V mediante el puente H TB6612 y leer la posición del WDD35D4-5K<sup>[4]</sup>.
- Puente H TB6612: circuito integrado utilizado para controlar motores de corriente continua a través de una configuración tipo puente H<sup>[5]</sup>.
- Fuente de voltaje 12V: fuente de voltaje externa utilizada para alimentar el motor MG513P20-12V.

## 5. Desarrollo

### 5.1. Modelo

Se elaboró un modelo 3D del péndulo de Furuta sobre un riel en un software de diseño, se le asignó material específico a cada componente y aplicamos renderizado para una mejor visualización. Para crear una carpeta URDF, realizamos un ensamble de 4 partes principales. La primera está conformada por la base, los rieles, placas y poleas; la segunda, por el dispositivo móvil, la cual tiene el sensor de la posición angular del péndulo. Para las últimas dos partes, solamente es la barilla del péndulo y la masa del extremo. De esta manera se le asigna un marco a cada cuerpo.



Figura 1: Pendulo de furuta lineal

## 5.2. Cinematica directa

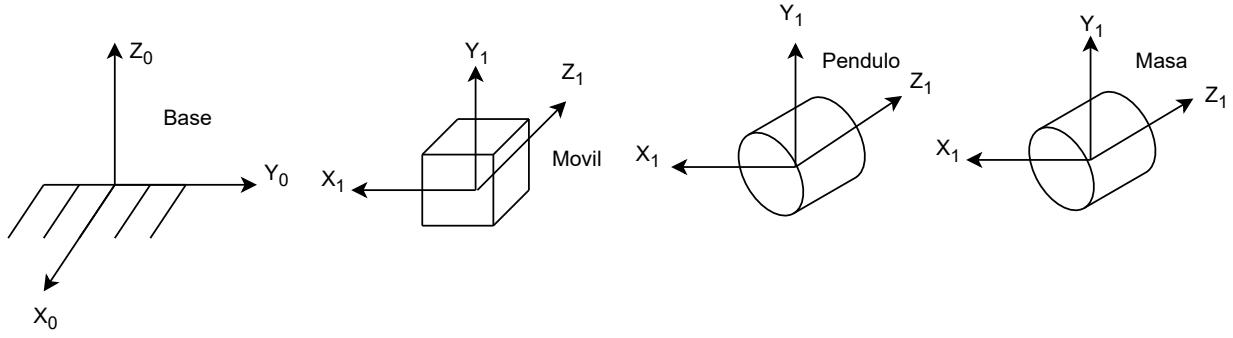


Figura 2: Marcos de Referencia

La cinemática directa del péndulo de Furuta se obtiene al encontrar las matrices homogéneas que siguen la siguiente estructura:

$$H = \begin{bmatrix} R_{3 \times 3} & d_{3 \times 1} \\ 0 & 1 \end{bmatrix}$$

### Nomenclatura de Variables

En este sistema de un péndulo invertido sobre un riel con 2 grados de libertad, se definen las siguientes variables:

- \$d\_1\$: Desplazamiento a lo largo del eje \$x\$ de la base, que define el movimiento del carro móvil.
- \$d\_2\$: Desplazamiento a lo largo del eje \$y\$ de la base, que es constante para este sistema.
- \$d\_3\$: Desplazamiento a lo largo del eje \$z\$ de la base, que es constante para este sistema.
- \$\theta\$: Ángulo de rotación del péndulo con respecto al carro móvil.
- \$d\_4\$: Desplazamiento a lo largo del eje \$x\$ del carro móvil debido al movimiento del péndulo.
- \$d\_5\$: Desplazamiento a lo largo del eje \$y\$ del carro móvil, que es constante.
- \$d\_6\$: Desplazamiento a lo largo del eje \$x\$ del péndulo.
- \$d\_7\$: Desplazamiento a lo largo del eje \$z\$ del péndulo.

### Matrices de Transformación

#### Matriz de Transformación de la Base al Carro Móvil

La matriz de transformación entre el marco de la base y el carro móvil es:

$${}^1H_0 = \begin{bmatrix} 0 & -1 & 0 & d_1 \\ 0 & 0 & 1 & d_2 \\ -1 & 0 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Matriz de Transformación del Carro Móvil al Péndulo

La matriz de transformación entre el carro móvil y el péndulo es:

$${}^2H_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & d_4 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Matriz de Transformación del Péndulo a la Masa del Péndulo

La matriz de transformación entre el péndulo y la masa del péndulo es:

$${}^3H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_6 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Multiplicación de las Matrices Homogéneas

Para obtener la posición y orientación de la masa del péndulo con respecto al marco base, las matrices homogéneas deben ser multiplicadas sucesivamente.

### Multiplicación de ${}^1H_0 \cdot {}^2H_1$

$${}^2H_0 = \begin{bmatrix} -\sin \theta & -\cos \theta & 0 & -d_4 + d_1 \\ 0 & 0 & 1 & d_5 + d_2 \\ -\cos \theta & \sin \theta & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### Multiplicación de ${}^2H_0 \cdot {}^3H_2$

$${}^3H_0 = \begin{bmatrix} -\sin \theta & -\cos \theta & 0 & -\cos \theta \cdot d_6 + (-d_4 + d_1) \\ 0 & 0 & 1 & d_7 + d_5 + d_2 \\ -\cos \theta & \sin \theta & 0 & \sin \theta \cdot d_6 + d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 5.3. Modelo Dinámico

#### Descripción del sistema físico

#### Definiciones y Parámetros

- $x(t)$ : posición del carro
- $\theta(t)$ : ángulo del péndulo desde la vertical hacia arriba
- $m$ : masa del péndulo
- $M$ : masa del carro
- $\ell$ : distancia desde el eje hasta el centro de masa del péndulo
- $I$ : momento de inercia del péndulo respecto a su centro de masa
- $u(t)$ : fuerza externa aplicada al carro
- $g$ : aceleración gravitacional

#### Cinemática

Posición del centro de masa del péndulo:

$$x_p = x + \ell \sin \theta, \quad z_p = \ell \cos \theta$$

Velocidades:

$$\dot{x}_p = \dot{x} + \ell \cos \theta \dot{\theta}, \quad \dot{z}_p = -\ell \sin \theta \dot{\theta}$$

#### Energías

##### Energía Cinética

Carro:

$$T_c = \frac{1}{2} M \dot{x}^2$$

Péndulo (traslacional + rotacional):

$$T_p = \frac{1}{2} m \left[ (\dot{x} + \ell \cos \theta \dot{\theta})^2 + (-\ell \sin \theta \dot{\theta})^2 \right] + \frac{1}{2} I \dot{\theta}^2$$

$$T_p = \frac{1}{2} m \left( \dot{x}^2 + 2\ell \dot{x} \cos \theta \dot{\theta} + \ell^2 \dot{\theta}^2 \right) + \frac{1}{2} I \dot{\theta}^2$$

##### Energía Potencial

$$V = mg\ell \cos \theta$$

#### Lagrangiano

$$\mathcal{L} = T - V = \frac{1}{2} (M + m) \dot{x}^2 + m\ell \cos \theta \dot{x} \dot{\theta} + \frac{1}{2} (I + m\ell^2) \dot{\theta}^2 - mg\ell \cos \theta$$

## Ecuaciones de Euler–Lagrange

Para  $x$ :

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{x}} \right) - \frac{\partial \mathcal{L}}{\partial x} = u(t)$$

$$(M+m)\ddot{x} + m\ell \cos \theta \ddot{\theta} - m\ell \sin \theta \dot{\theta}^2 = u(t)$$

Para  $\theta$ :

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right) - \frac{\partial \mathcal{L}}{\partial \theta} = 0$$

$$(I+m\ell^2)\ddot{\theta} + m\ell \cos \theta \ddot{x} - mg\ell \sin \theta = 0$$

## Modelo Dinámico No Lineal

$$\begin{cases} (M+m)\ddot{x} + m\ell \cos \theta \ddot{\theta} - m\ell \sin \theta \dot{\theta}^2 = u(t) \\ (I+m\ell^2)\ddot{\theta} + m\ell \cos \theta \ddot{x} - mg\ell \sin \theta = 0 \end{cases}$$

## Linealización para $\theta \approx 0$

Aproximaciones:

$$\sin \theta \approx \theta, \quad \cos \theta \approx 1, \quad \dot{\theta}^2 \approx 0$$

Sistema linealizado:

$$\begin{cases} (M+m)\ddot{x} + m\ell \ddot{\theta} = u(t) \\ (I+m\ell^2)\ddot{\theta} + m\ell \ddot{x} - mg\ell \theta = 0 \end{cases}$$

## Despeje Final (caso $I = 0$ )

$$\begin{aligned} \ddot{x} &= -\frac{m}{M}g\theta + \frac{1}{M}u(t) \\ \ddot{\theta} &= \frac{M-m}{M\ell}g\theta - \frac{1}{M\ell}u(t) \end{aligned}$$

#### 5.4. Diseño de la placa de circuito impreso

Se realizó un diseño de la placa de circuito impreso de doble capa en el software Fritzing, específico para este proyecto, con el objetivo de controlar un motor con encoder para el movimiento de la pieza móvil mediante un puente H, leer la posición del potenciómetro que indica la ubicación del péndulo y alimentar el motor mediante una fuente de voltaje externa. Todo esto es controlado por un ESP32.

#### 5.4.1. Componentes de la placa de circuito impreso

- ESP-32 Devkit v1: placa de desarrollo que integra un potente microcontrolador con capacidades de conexión Wi-Fi y Bluetooth.
  - puente H TB6612: circuito integrado utilizado para controlar motores de corriente continua a través de una configuración tipo puente H.

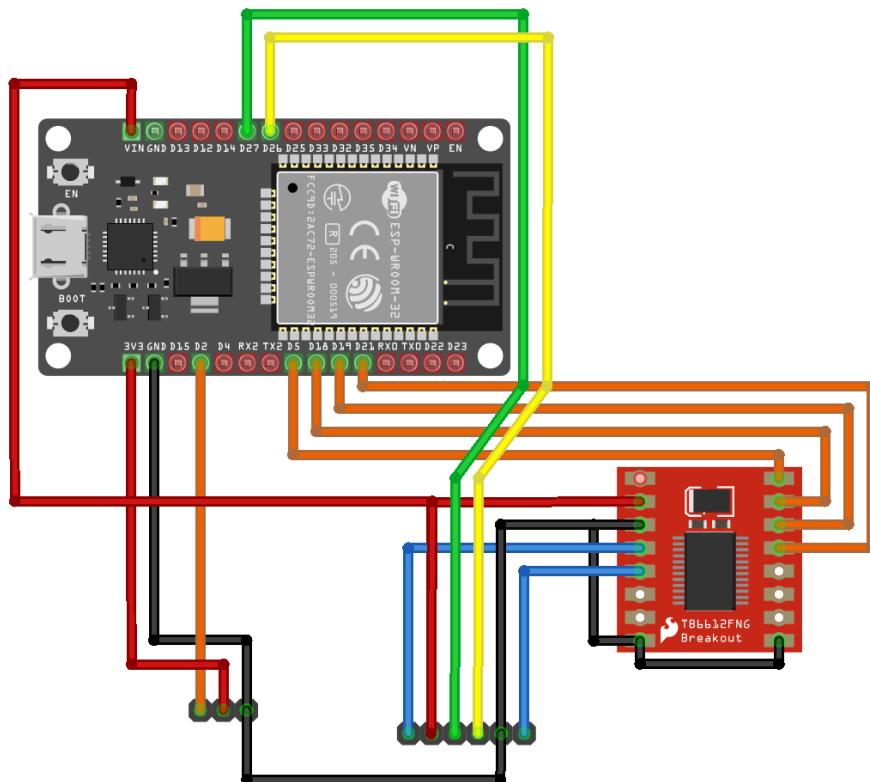


Figura 3: Diagrama pictórico

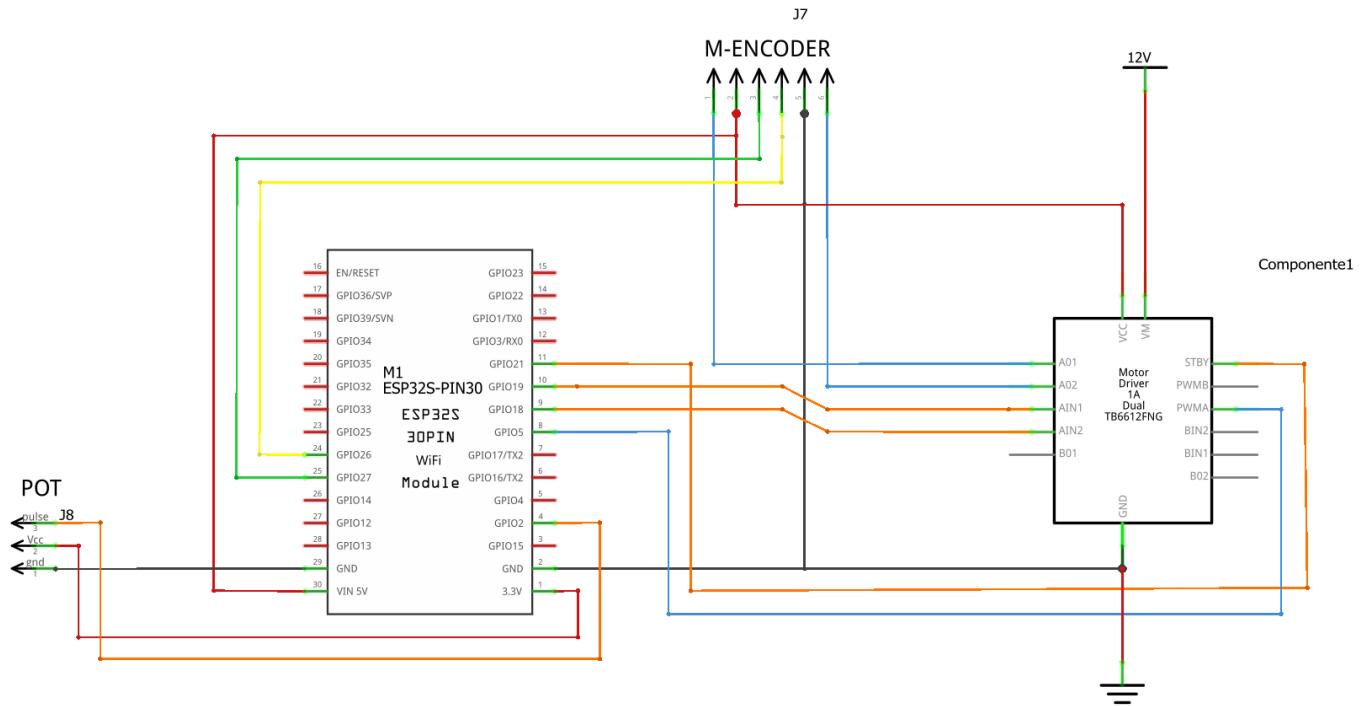


Figura 4: Diagrama esquematico

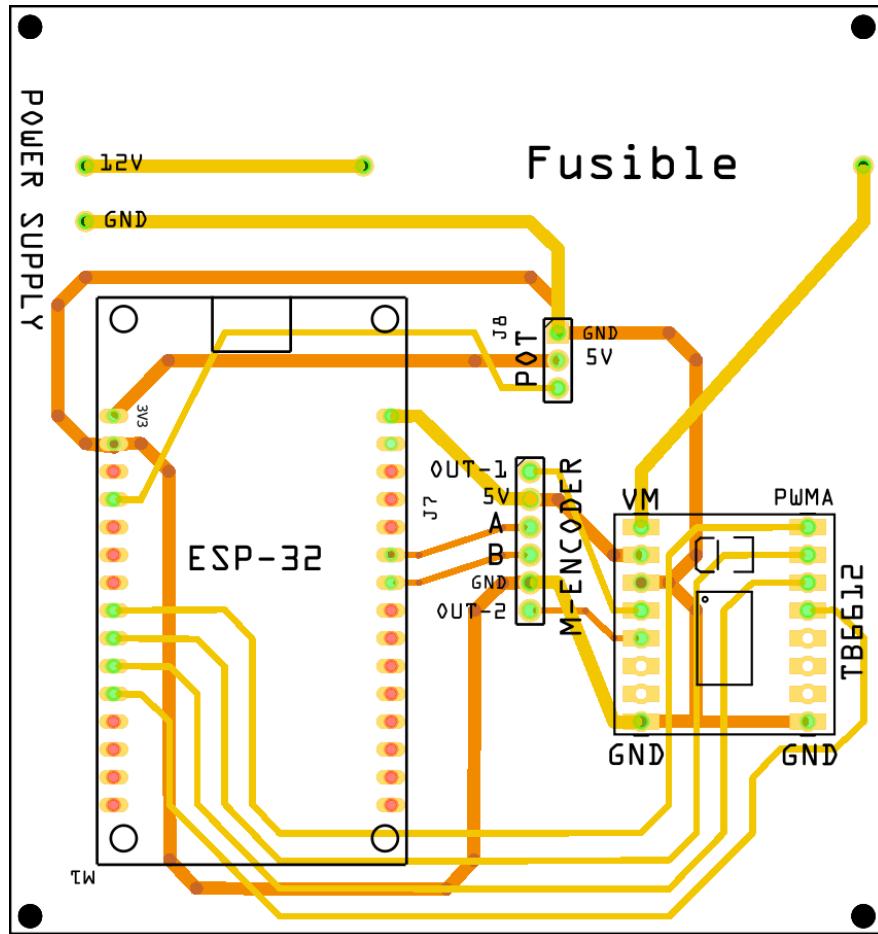


Figura 5: Diseño de placa de circuito impreso

## 5.5. Ros2 jazzy y Gazebo harmonic

Se utilizaron estas herramientas para poder visualizar y simular nuestro modelo en rviz2 y gazebo.

### 5.5.1. Visualizacion con RVIZ

Se utilizo rviz2 para visualizar el urdf de nuestro modelo y asi poder ver las estructura de transformaciones, que sirve para entender los movimientos del pendulo

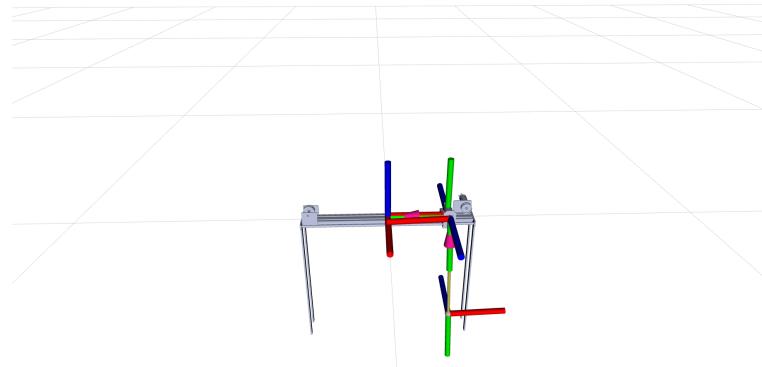


Figura 6: Visualizacion en RVIS

### 5.5.2. Simulacion en Gazebo

Se utilizo gazebo para hacer una simulacion fisica realista de nuestro pendulo

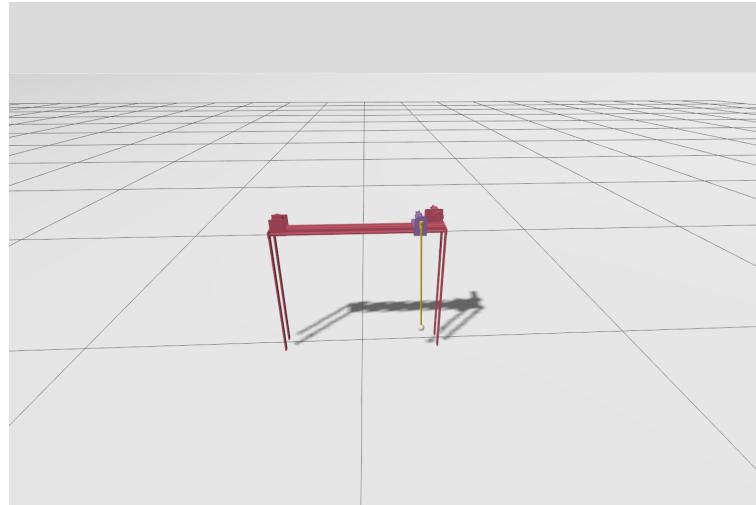


Figura 7: Simulacion en Gazebo

### 5.5.3. Gazebo plugging

Se llevó a cabo la conexión mediante la implementación y análisis de un *plugin* nativo de Gazebo, específicamente:

Listing 1: Plugin utilizado para el control de posición de la articulación

```
<plugin name="gz::sim::systems::JointPositionController">
  filename="gz-sim-joint-position-controller-system">
    <joint_name>movil_joint</joint_name>
    <topic>movil_cmd</topic>
</plugin>
```

### 5.5.4. Puente entre Gazebo y Rviz

Para establecer la comunicación entre ROS 2 y Gazebo, se implementó el nodo `parameter_bridge` perteneciente al paquete `ros_gz_bridge`. Este nodo actúa como un puente que permite la conversión de mensajes entre los formatos de Gazebo y ROS 2, habilitando la comunicación entre ambos entornos.

La configuración utilizada se muestra a continuación:

Listing 2: Configuración del nodo bridge entre ROS 2 y Gazebo

```
gz_bridge = Node(
  package='ros_gz_bridge',
  executable='parameter_bridge',
  arguments=[
    "/movil_cmd@std_msgs/msg/Float64 [gz.msgs.Double]",
    "/odom@nav_msgs/msg/Odometry [gz.msgs.Odometry]",
    "/tf@tf2_msgs/msg/TFMessage [gz.msgs.Pose_V]",
    "/joint_states@sensor_msgs/msg/JointState [gz.msgs.Model]",
  ],
  output='screen',
  parameters=[{'use_sim_time': True}],
)
```

A continuación, se detalla el propósito de cada uno de los tópicos enlazados:

- `/movil_cmd`: Transfiere comandos de posición o esfuerzo al modelo en simulación, utilizando el tipo `Float64` en ROS y `Double` en Gazebo.
- `/odom`: Permite recibir información de odometría desde Gazebo, convertida al tipo estándar `Odometry` en ROS 2.
- `/tf`: Gestiona la transmisión de transformaciones de referencia espacial mediante el tipo `TFMessage` en ROS, traducido desde `Pose_V` en Gazebo.
- `/joint_states`: Sincroniza el estado de las articulaciones del modelo simulado usando el tipo `JointState` en ROS 2, vinculado a `Model` en Gazebo.

La opción `use_sim_time: True` habilita el uso del tiempo simulado de Gazebo como fuente de temporización en ROS 2, garantizando una correcta sincronización temporal durante la ejecución de nodos que dependen del reloj.

Esta configuración es fundamental para integrar correctamente la simulación física del modelo con los nodos ROS, permitiendo tanto el control como la visualización del estado del sistema en tiempo real.

### 5.5.5. Micro-Ros

En el sistema físico del péndulo invertido sobre riel, se utilizó un microcontrolador ESP32 programada con el entorno Arduino IDE. Se configuraron dos sensores: un potenciómetro para medir el ángulo del péndulo y un encoder para medir la posición lineal del carro. La comunicación con ROS 2 se realiza mediante micro-ROS, lo que permite exponer estos datos al resto del sistema y el acoplamiento de los topicos a la simulacion.

Se crearon dos tópicos:

- `/cart_position`: Nos proporciona la posición del carro en el riel, medida por el encoder.
- `/pendulum_angle`: Nos proporciona el ángulo del péndulo, medido por el potenciómetro.

Para visualizar los datos de los sensores en ROS 2, se utilizan los siguientes comandos en una terminal:

- `ros2 topic echo /cart_position`
- `ros2 topic echo /pendulum_angle`

Estos comandos permiten monitorear en tiempo real los valores publicados por los sensores en el sistema.

## Referencias

- [1] [https://www.researchgate.net/publication/376596779\\_DISENO\\_Y\\_ANALISIS\\_DE\\_LA\\_TECNICA\\_DE\\_CONTROL\\_PID PARA\\_PENDULO\\_INVERSO\\_LINEAL\\_EN\\_V-REALMR](https://www.researchgate.net/publication/376596779_DISENO_Y_ANALISIS_DE_LA_TECNICA_DE_CONTROL_PID PARA_PENDULO_INVERSO_LINEAL_EN_V-REALMR). [Accessed 27-04-2025].
- [2] “Mg513 dc 12v.” <https://es.aliexpress.com/i/4000996252848.html?gatewayAdapt=glo2esp>. [Accessed 27-04-2025].
- [3] “ANGULAR DISPLACEMENT SENSOR / POTENTIOMETER / PRECISION-WDD35D4M - Zhejiang Huiren Electronics Co. Ltd. - PDF Catalogs — Technical Documentation — Brochure — pdf.directindustry.com.” [https://pdf.directindustry.com/pdf/zhejiang-huirenl-electronics-co-ltd/angular-displacement-sensor-potentiometer-precision-wdd35d4m/198850-763670.html?fbclid=IwY2xjawJ71k5leHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR5dmxQZR\\_28WzSH61ALJoYa3d7N340Ug0oddcpFK\\_nrLdTrYCDhE\\_\\_aus6Cg\\_aem\\_5EQgpqyw1zv3l-3Xf0\\_1eQ](https://pdf.directindustry.com/pdf/zhejiang-huirenl-electronics-co-ltd/angular-displacement-sensor-potentiometer-precision-wdd35d4m/198850-763670.html?fbclid=IwY2xjawJ71k5leHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR5dmxQZR_28WzSH61ALJoYa3d7N340Ug0oddcpFK_nrLdTrYCDhE__aus6Cg_aem_5EQgpqyw1zv3l-3Xf0_1eQ). [Accessed 28-04-2025].
- [4] R. Mischianti, “DOIT ESP32 DEV KIT v1: high resolution pinout and specs — mischianti.org.” [https://mischianti.org/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/?fbclid=IwY2xjawJ7141leHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR4XKd731G10pWY985HFCFeE8YYmXXCoUKNV51WwaxZsTj\\_Q\\_aem\\_1q1cwkMvCat6xofAmiNQkA](https://mischianti.org/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/?fbclid=IwY2xjawJ7141leHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR4XKd731G10pWY985HFCFeE8YYmXXCoUKNV51WwaxZsTj_Q_aem_1q1cwkMvCat6xofAmiNQkA). [Accessed 28-04-2025].
- [5] “pololu.com.” [https://www.pololu.com/file/0J86/TB6612FNG.pdf?fbclid=IwY2xjawJ71GJ1eHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR5dmxQZR\\_28WzSH61ALJoYa3d7N340Ug0oddcpFK\\_nrLdTrYCDhE\\_\\_aus6Cg\\_aem\\_5EQgpqyw1zv3l-3Xf0\\_1eQ](https://www.pololu.com/file/0J86/TB6612FNG.pdf?fbclid=IwY2xjawJ71GJ1eHRuA2FlbQIxMABicmlkETF3SnprVnQ1Q1psa2dNc2pVAR5dmxQZR_28WzSH61ALJoYa3d7N340Ug0oddcpFK_nrLdTrYCDhE__aus6Cg_aem_5EQgpqyw1zv3l-3Xf0_1eQ). [Accessed 28-04-2025].

## A. Apendice

### Codigo Arduino IDE

Listing 3: Código de control PID para péndulo invertido con micro-ROS

```
#include <PID_v1.h>
#include <math.h>
#include <micro_ros_arduino.h>
#include <rcl/rcl.h>
#include <rclc/rclc.h>
#include <rclc/executor.h>
#include <std_msgs/msg/float32.h>

const int potPin = 2;
const int IN1 = 18;
const int IN2 = 19;
const int PWMA = 5;
const int STBY = 21;
const int encoderPinA = 27;
const int encoderPinB = 26;

double angleInput = 0, angleOutput = 0, angleSetpoint = 0;
double angleKp = 250, angleKi = 110, angleKd = 17;

double carroInput = 0, carroOutput = 0, carroSetpoint = 0;
double carroKp = 13, carroKi = 1.0, carroKd = 0.01;

PID anglePID(&angleInput, &angleOutput,
&angleSetpoint, angleKp, angleKi, angleKd, DIRECT);
PID carroPID(&carroInput, &carroOutput,
&carroSetpoint, carroKp, carroKi, carroKd, DIRECT);

volatile long encoderCount = 0;
int lastEncoded = 0;

const double pasosPorCm = 180;
const double limiteMin = -14.0;
const double limiteMax = 10;

rcl_node_t node;
rclc_support_t support;
rcl_allocator_t allocator;
rcl_publisher_t angle_pub;
rcl_publisher_t pos_pub;
std_msgs_msg_Float32 angle_msg;
std_msgs_msg_Float32 pos_msg;

void updateEncoder() {
    int MSB = digitalRead(encoderPinA);
    int LSB = digitalRead(encoderPinB);
    int encoded = (MSB << 1) | LSB;
    int sum = (lastEncoded << 2) | encoded;

    if (sum == 0b1101 || sum == 0b0100 ||
```

```

| sum == 0b0010 || sum == 0b1011) encoderCount++;
if (sum == 0b1110 || sum == 0b0111 |
| sum == 0b0001 || sum == 0b1000) encoderCount--;
lastEncoded = encoded;
}

void setup() {
Serial.begin(115200);

pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);
pinMode(PWMA, OUTPUT); pinMode(STBY, OUTPUT);
pinMode(potPin, INPUT);
pinMode(encoderPinA, INPUT); pinMode(encoderPinB, INPUT);

attachInterrupt(digitalPinToInterrupt(encoderPinA),
updateEncoder, CHANGE);
attachInterrupt(digitalPinToInterrupt(encoderPinB),
updateEncoder, CHANGE);
digitalWrite(STBY, HIGH);

int lecturaRaw = analogRead(potPin);
angleSetpoint = map(lecturaRaw, 0, 4095, -9000, 9000) / 100.0;
noInterrupts();
long encoderCopy = encoderCount;
interrupts();
carroSetpoint = encoderCopy / pasosPorCm;

anglePID.SetMode(AUTOMATIC);
anglePID.SetOutputLimits(-150, 150);
carroPID.SetMode(AUTOMATIC);
carroPID.SetOutputLimits(-100, 100);

set_microros_transports();
allocator = rcl_get_default_allocator();
rclc_support_init(&support, 0, NULL, &allocator);
rclc_node_init_default(&node, "pendulo_node", "", &support);

rclc_publisher_init_default(
    &angle_pub,
    &node,
    ROSIDL_GET_MSG_TYPE_SUPPORT(std_msgs, msg, Float32),
    "pendulum_angle"
);

rclc_publisher_init_default(
    &pos_pub,
    &node,
    ROSIDL_GET_MSG_TYPE_SUPPORT(std_msgs, msg, Float32),
    "cart_position"
);
}

void loop() {

```

```

int lecturaRaw = analogRead(potPin);
angleInput = map(lecturaRaw, 0, 4095, -9000, 9000) / 100.0;

noInterrupts();
long encoderCopy = encoderCount;
interrupts();
carroInput = encoderCopy / pasosPorCm;

anglePID.Compute();
carroPID.Compute();
double controlTotal = angleOutput + carroOutput;

if ((carroInput <= limiteMin && controlTotal < 0) ||
    (carroInput >= limiteMax && controlTotal > 0)) {
    controlTotal = 0;
}

controlTotal = 150.0 * tanh(controlTotal / 150.0);
static double outputFiltrado = 0;
double alpha = 0.3;
outputFiltrado = alpha * controlTotal + (1 - alpha) * outputFiltrado;

if (abs(angleInput - angleSetpoint) < 1.5) {
    digitalWrite(PWMA, 0);
    digitalWrite(IN1, LOW); digitalWrite(IN2, LOW);
} else {
    if (outputFiltrado > 0) {
        digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
    } else {
        digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
    }
    analogWrite(PWMA, abs(outputFiltrado));
}

angle_msg.data = angleInput;
pos_msg.data = carroInput;
rcl_publish(&angle_pub, &angle_msg, NULL);
rcl_publish(&pos_pub, &pos_msg, NULL);

Serial.print("Setpoint ngulo ( ): "); Serial.print(angleSetpoint);
Serial.print(" | ngulo actual ( ): "); Serial.print(angleInput);
Serial.print(" | Pos. carro (cm): "); Serial.print(carroInput);
Serial.print(" | Setpoint carro (cm): "); Serial.print(carroSetpoint);
Serial.print(" | PWM: "); Serial.println(outputFiltrado);

delay(10);
}

```

### A.1. Modelo de solid works (renderizados) y planos



Figura 8: Masa del pendulo



Figura 9: Varilla del pendulo

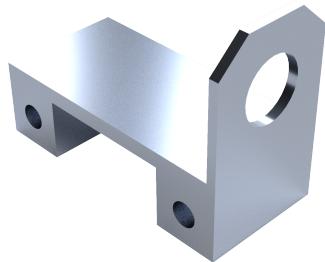


Figura 10: Dispositivo Movil



Figura 11: Barra deslizante



Figura 12: Eje de las poleas



Figura 13: Riel del chsis



Figura 14: Placa que no sostiene el motor



Figura 15: Placa que sostiene el motor



Figura 16: Placa de los extremos



Figura 17: Polea



Figura 18: Motor

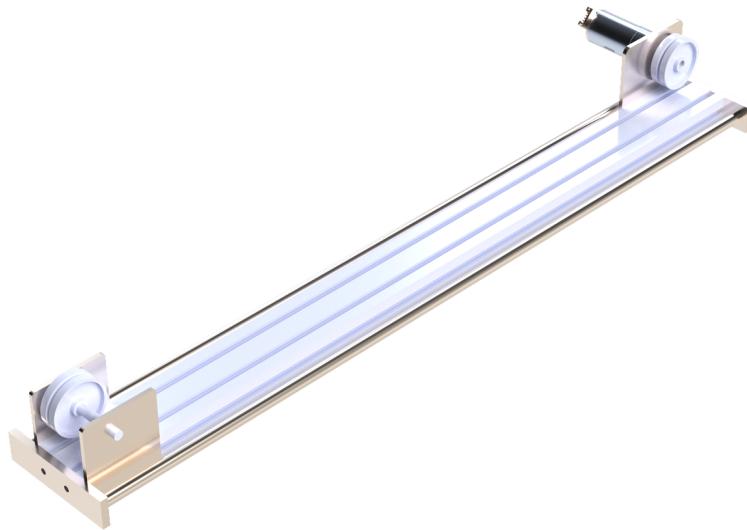


Figura 19: Primera parte del pendulo, base principal

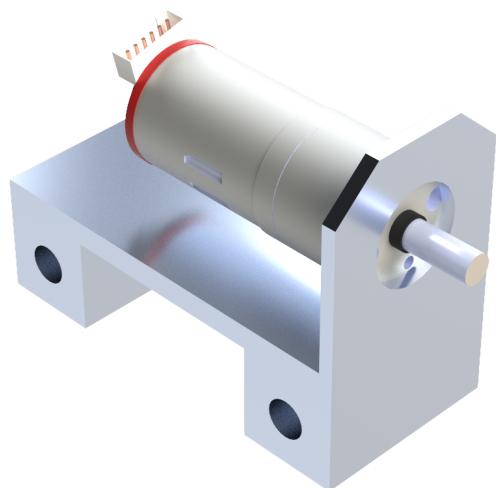


Figura 20: Segunda parte del pendulo, dispositivo movil

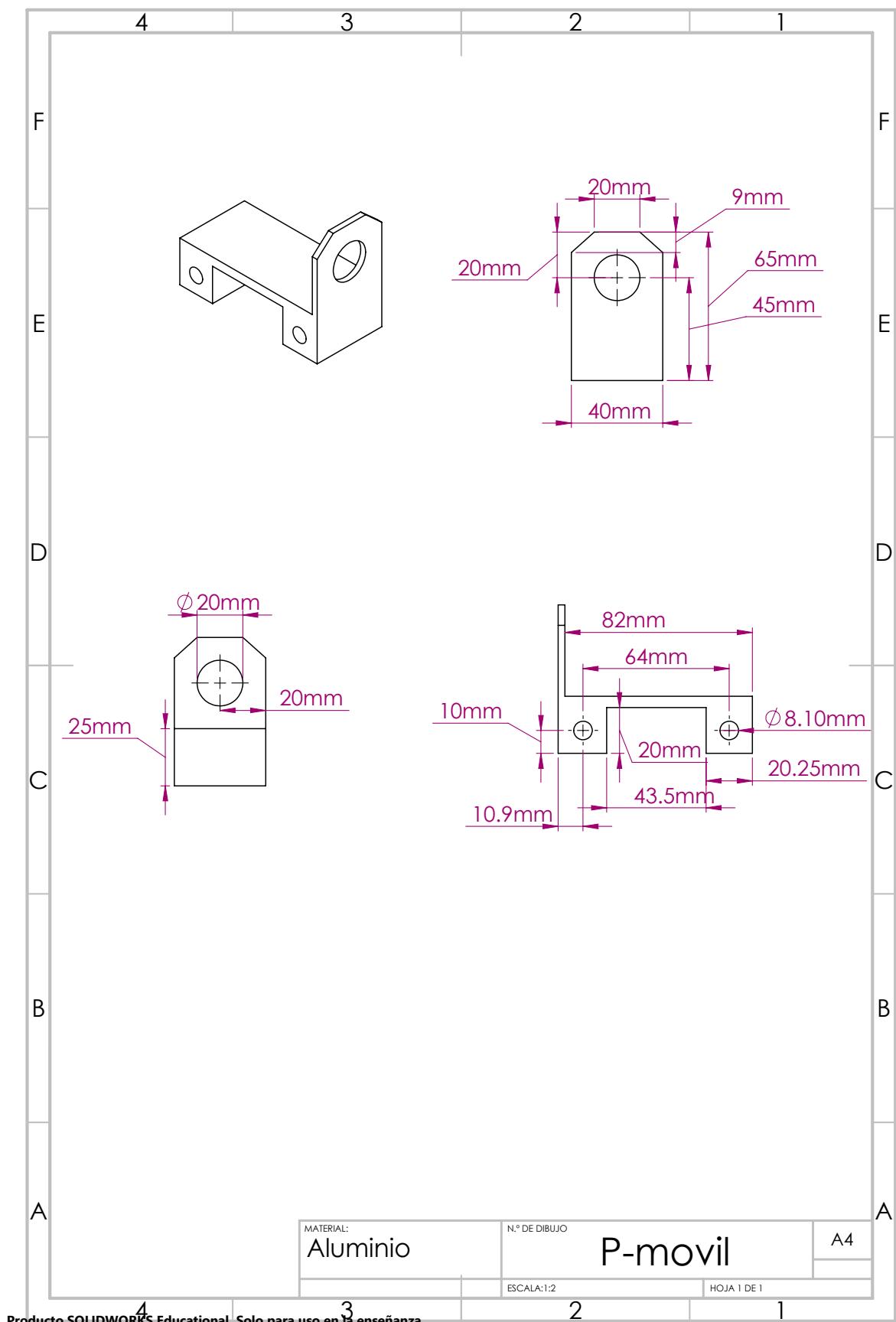


Figura 21: Plano del Elemento Movil <sup>23</sup>

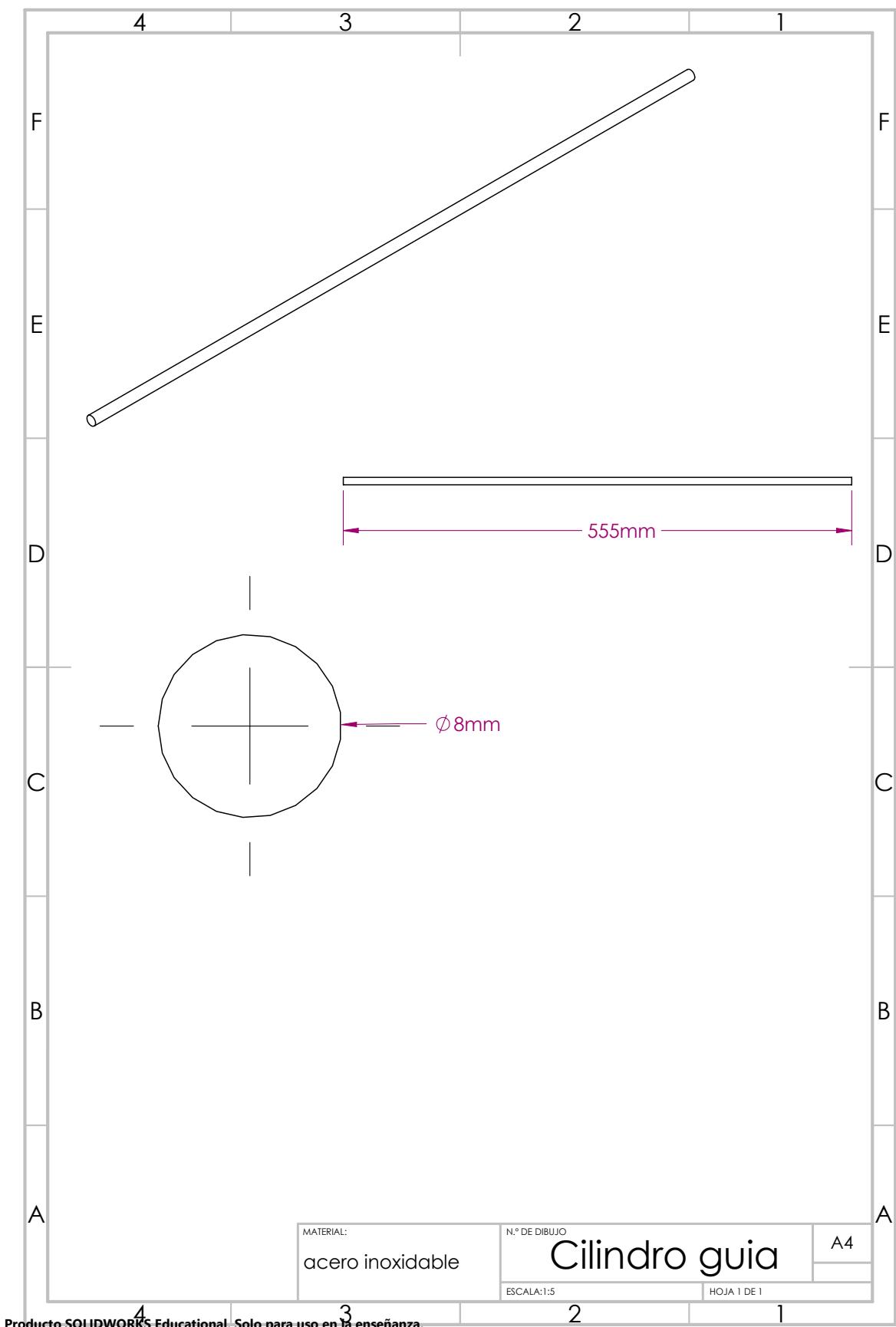


Figura 22: Plano de la barilla guia <sup>24</sup>

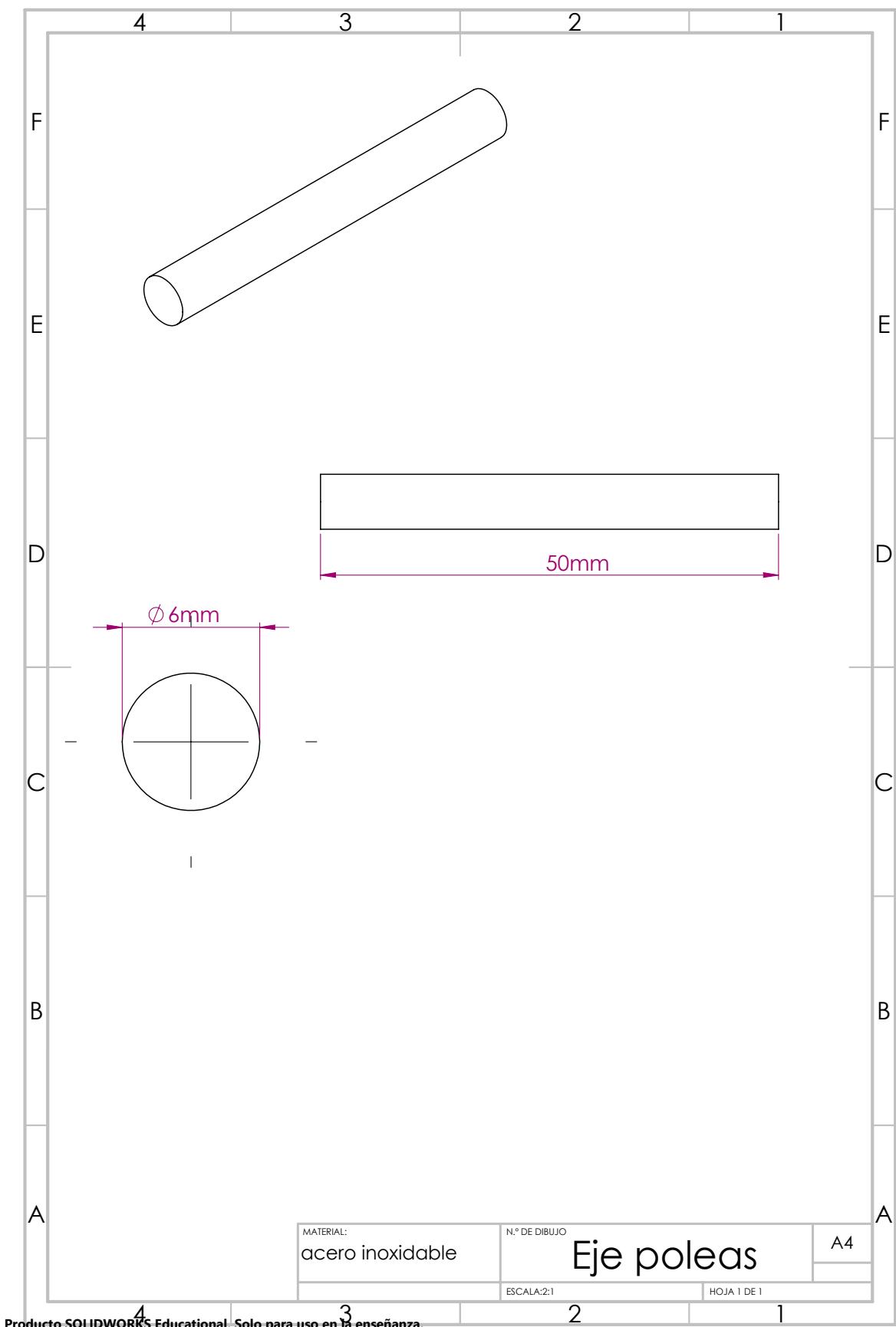


Figura 23: Plano del eje de las poleas <sup>25</sup>

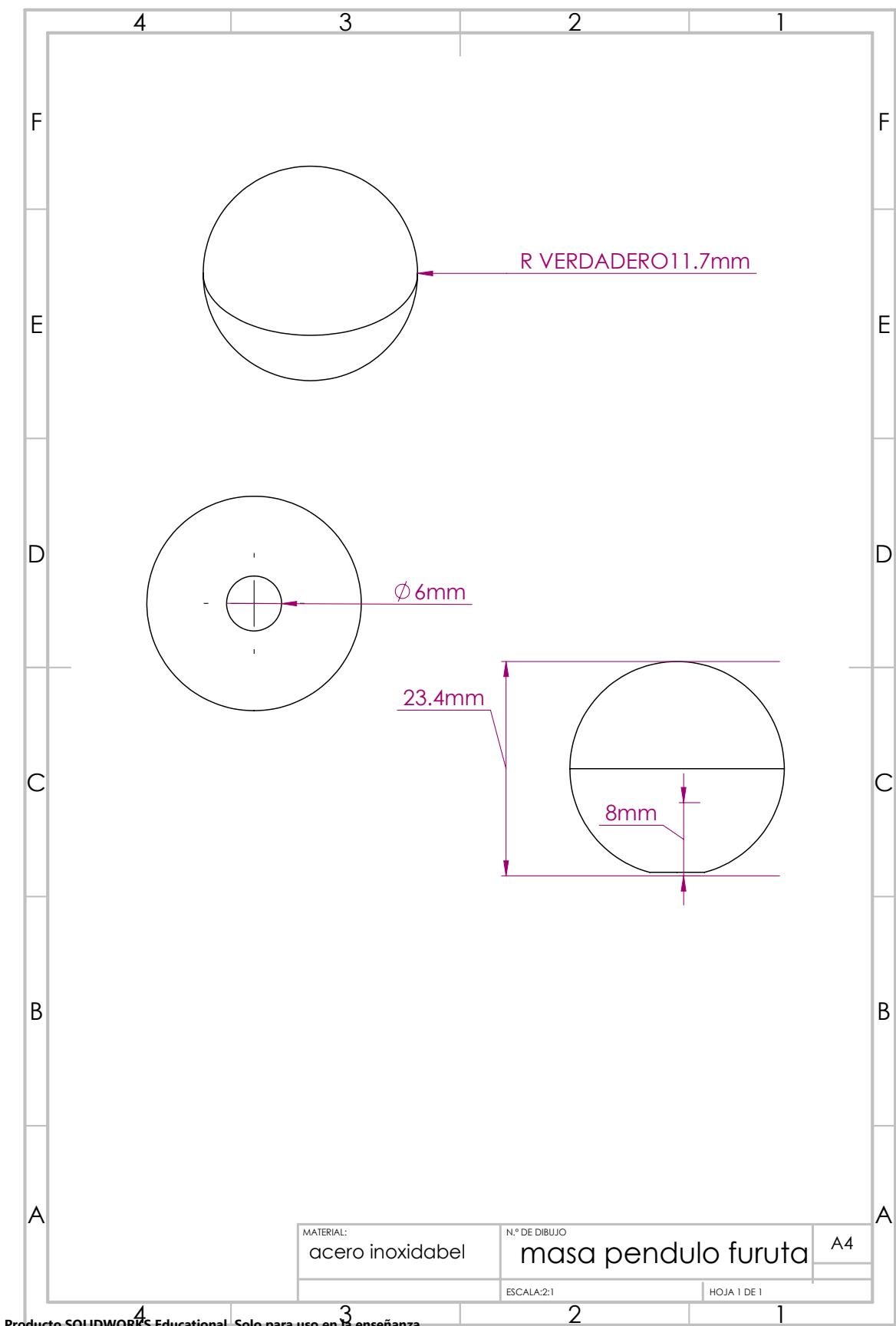


Figura 24: Plano de la masa del extremo del pendulo <sup>26</sup>

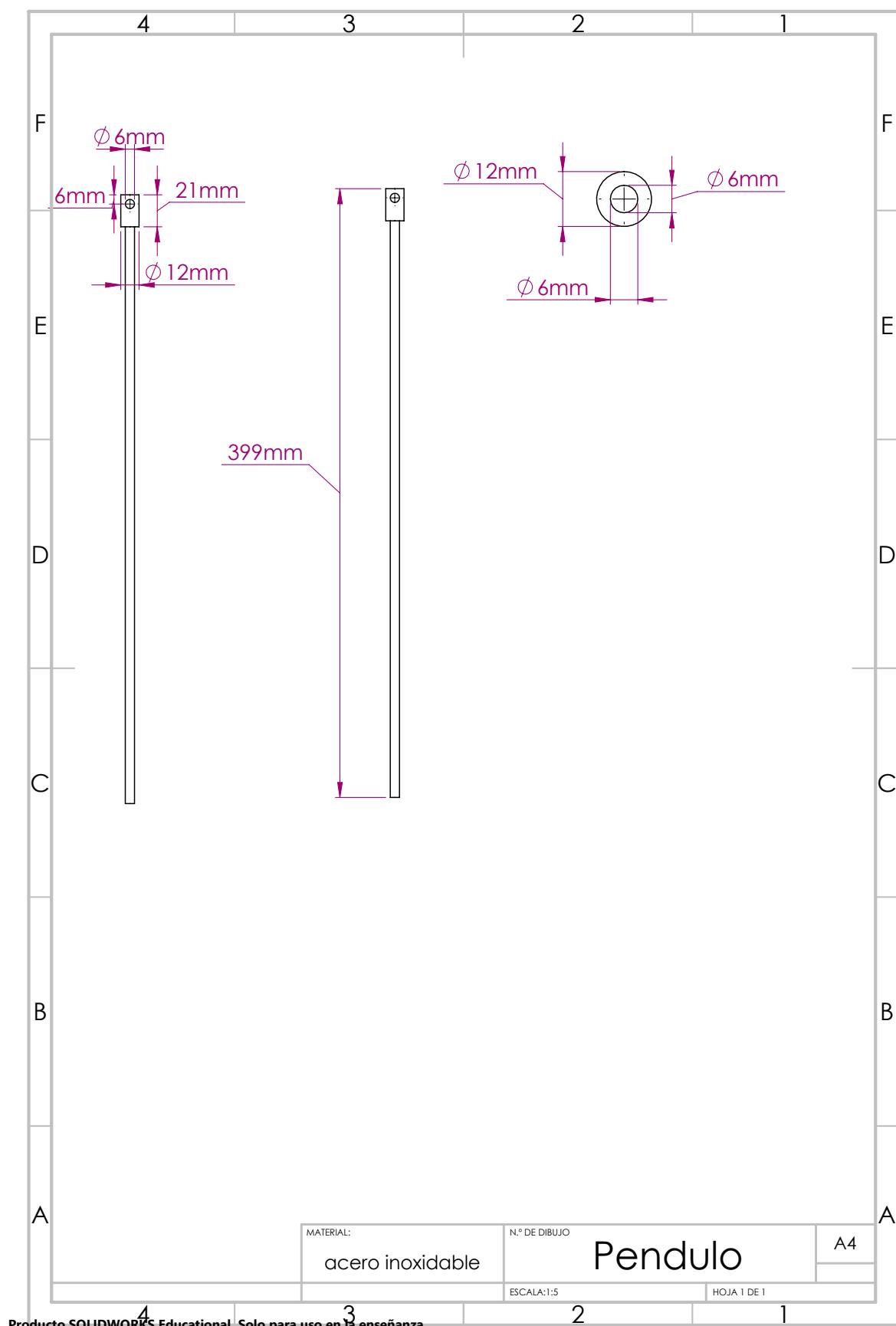


Figura 25: Plano de la barilla pendulo <sup>27</sup>

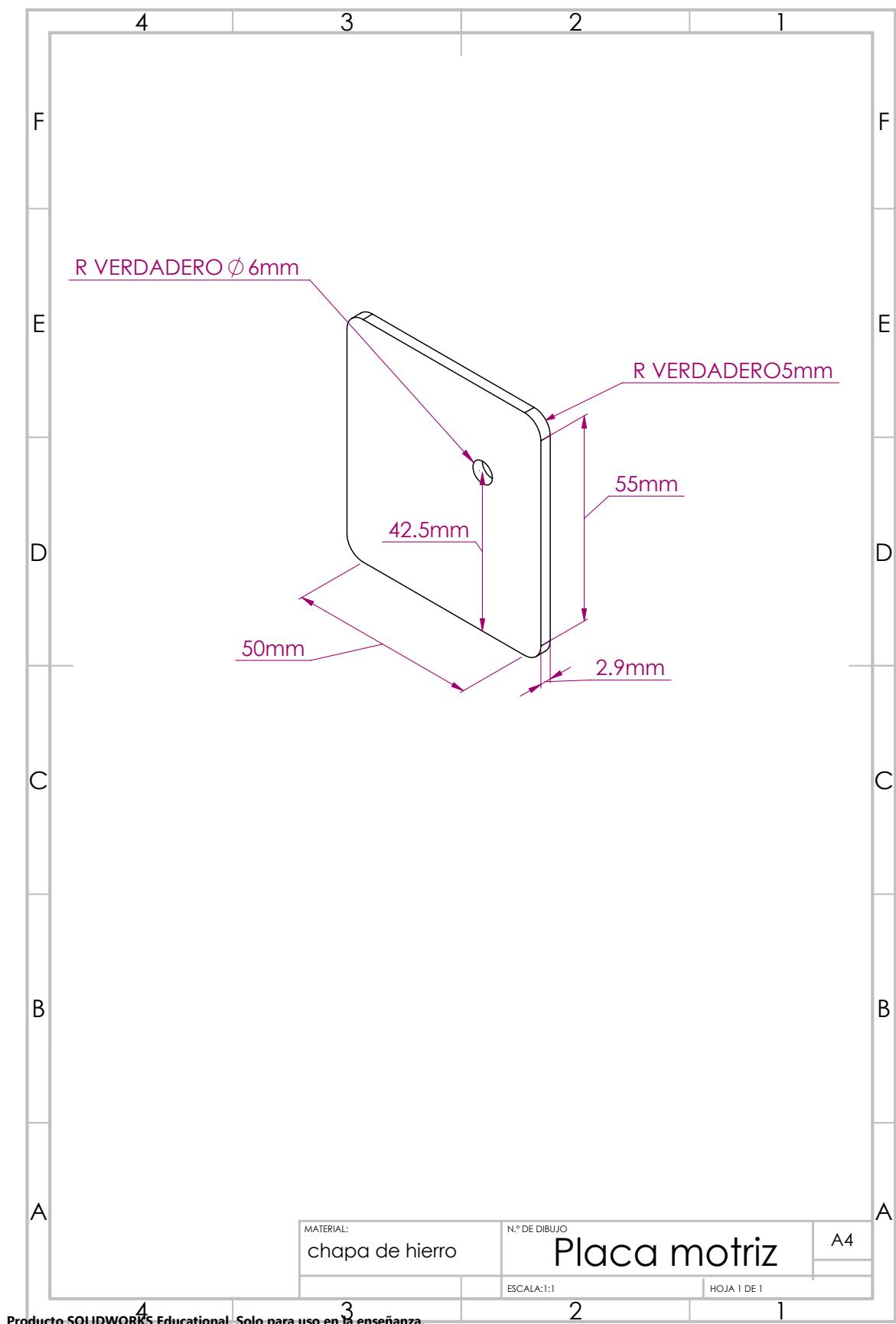


Figura 26: Plano de la placa que sostiene el motor.<sup>28</sup>

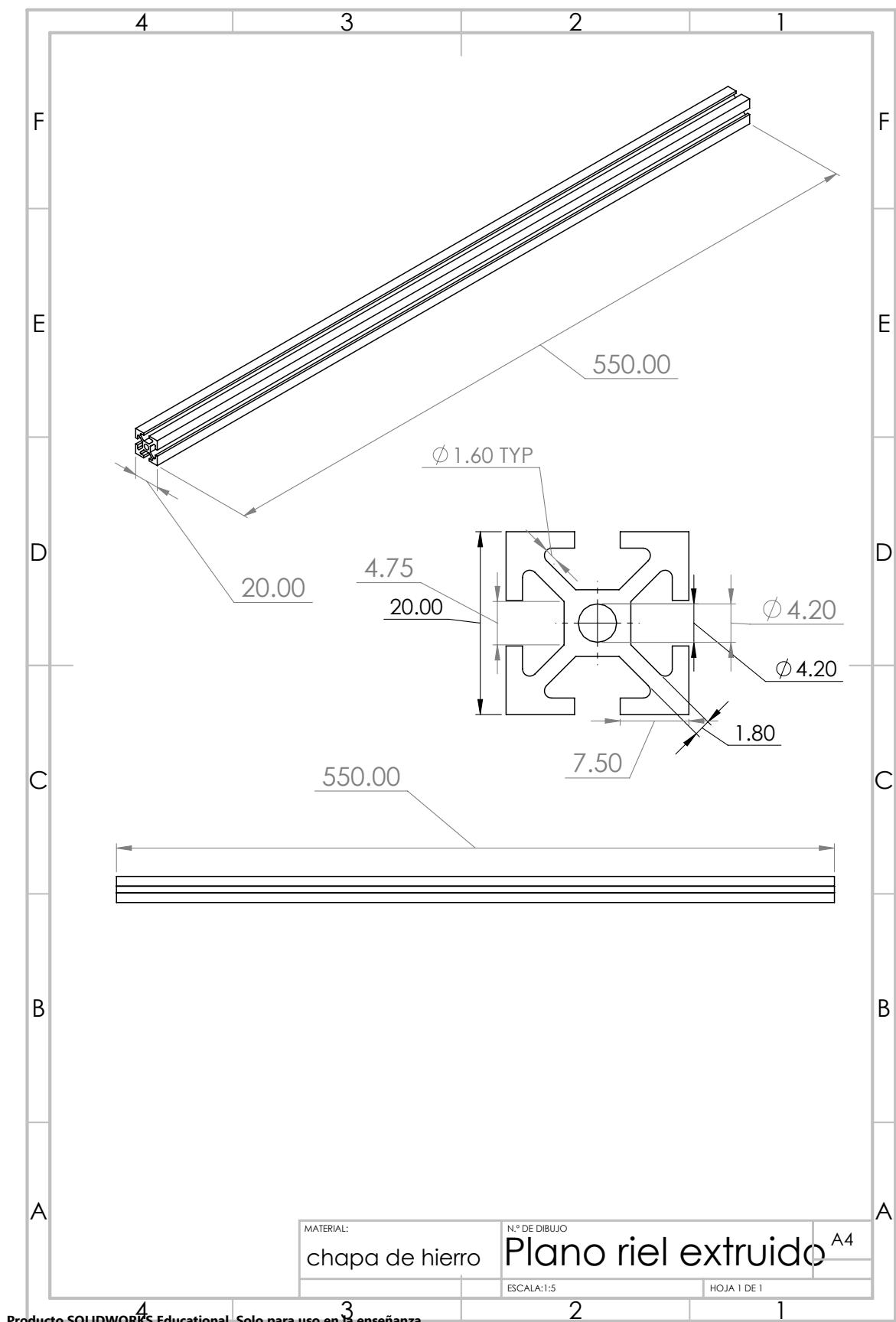


Figura 27: Plano del riel del chasis del pendulo de furuta lineal.  
29

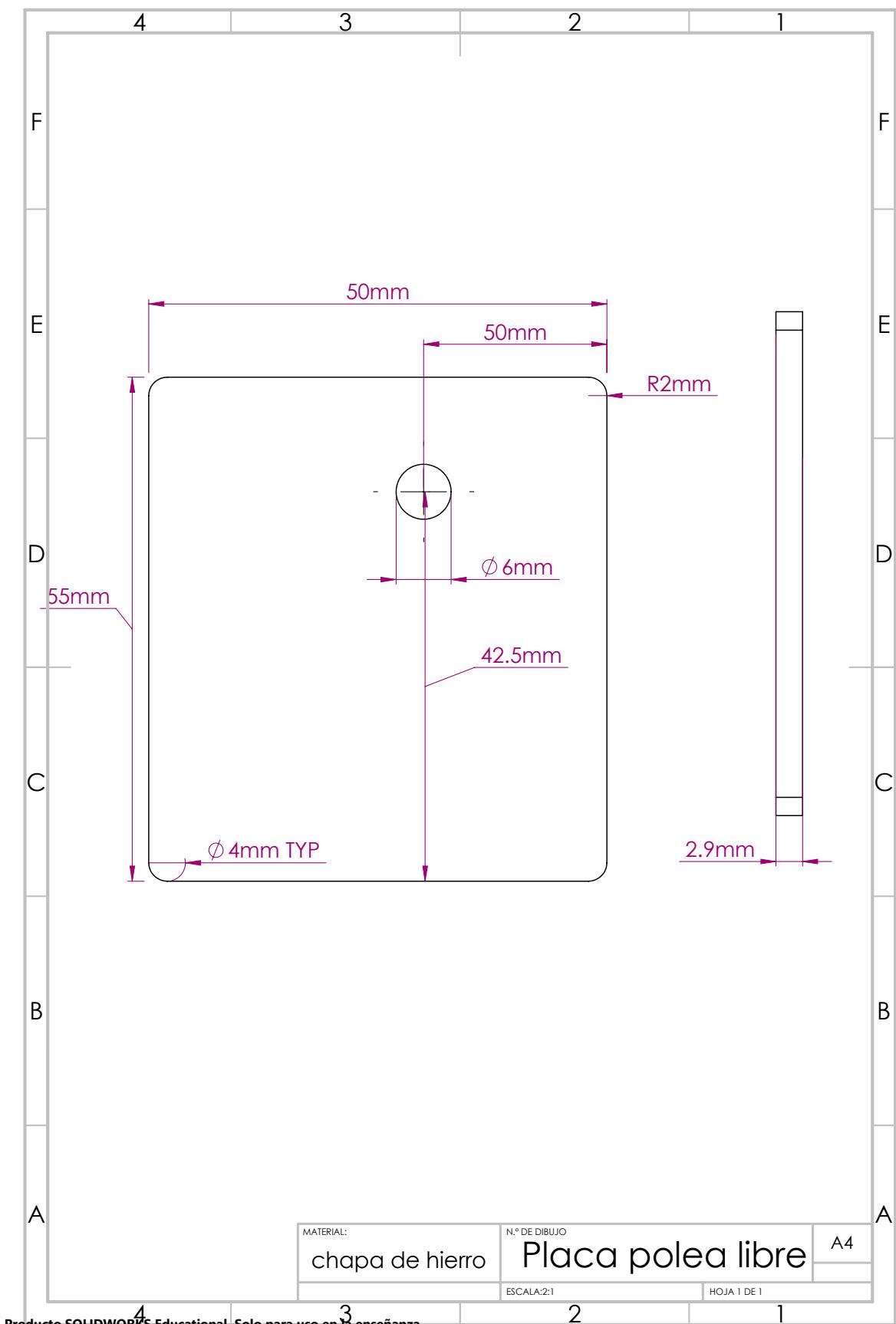


Figura 28: Plano de la placa no motriz.

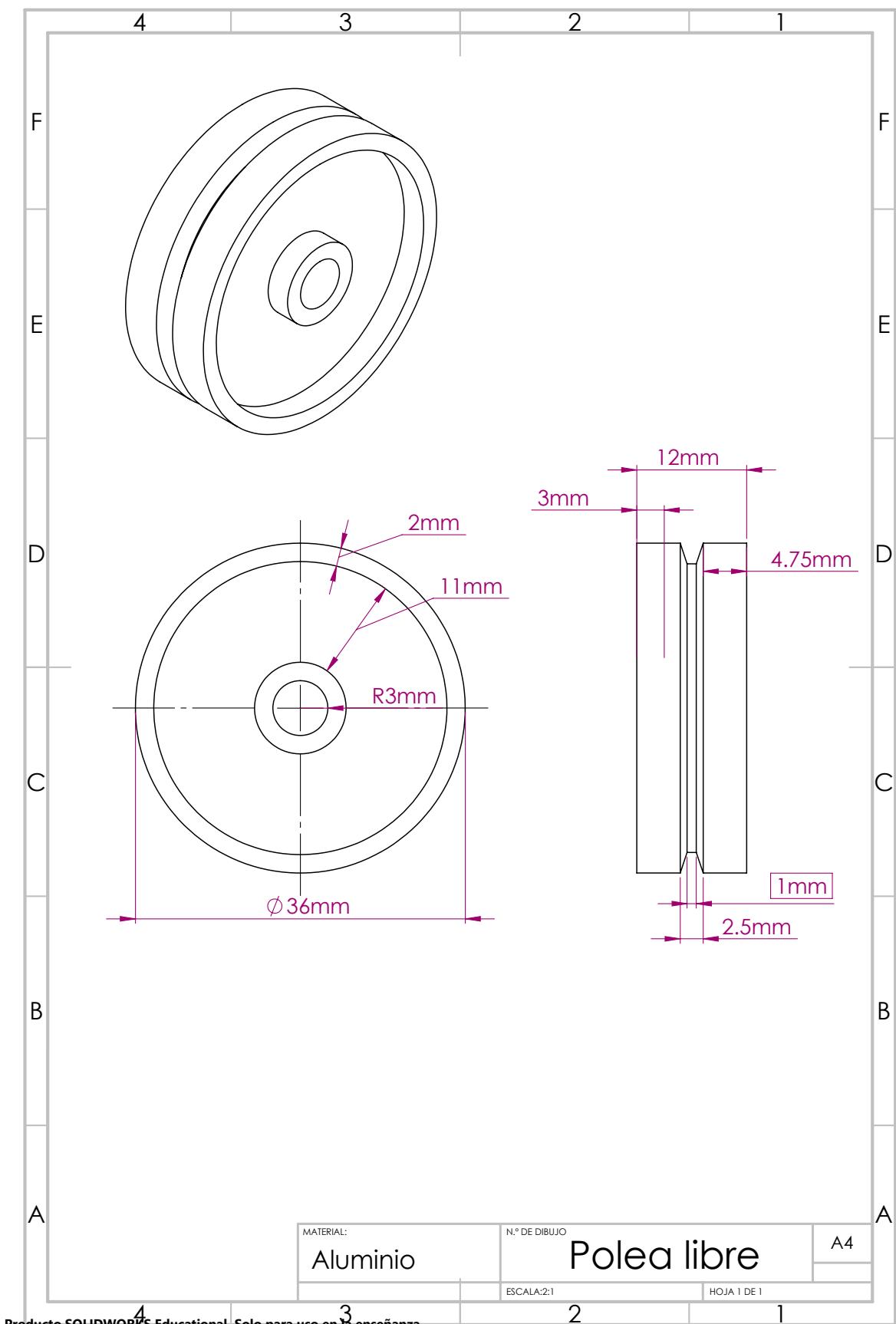


Figura 29: Plano de la polea.<sup>31</sup>

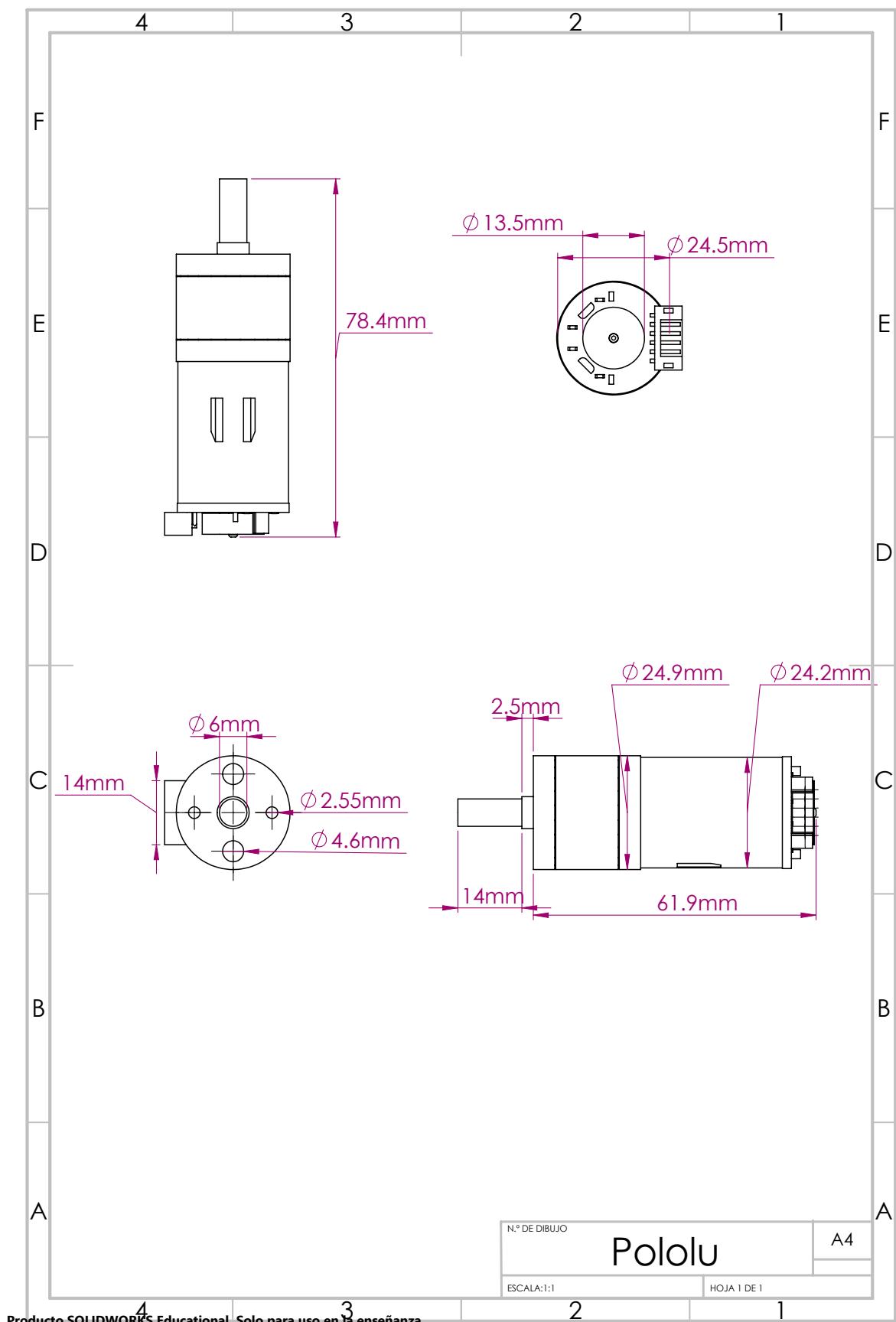
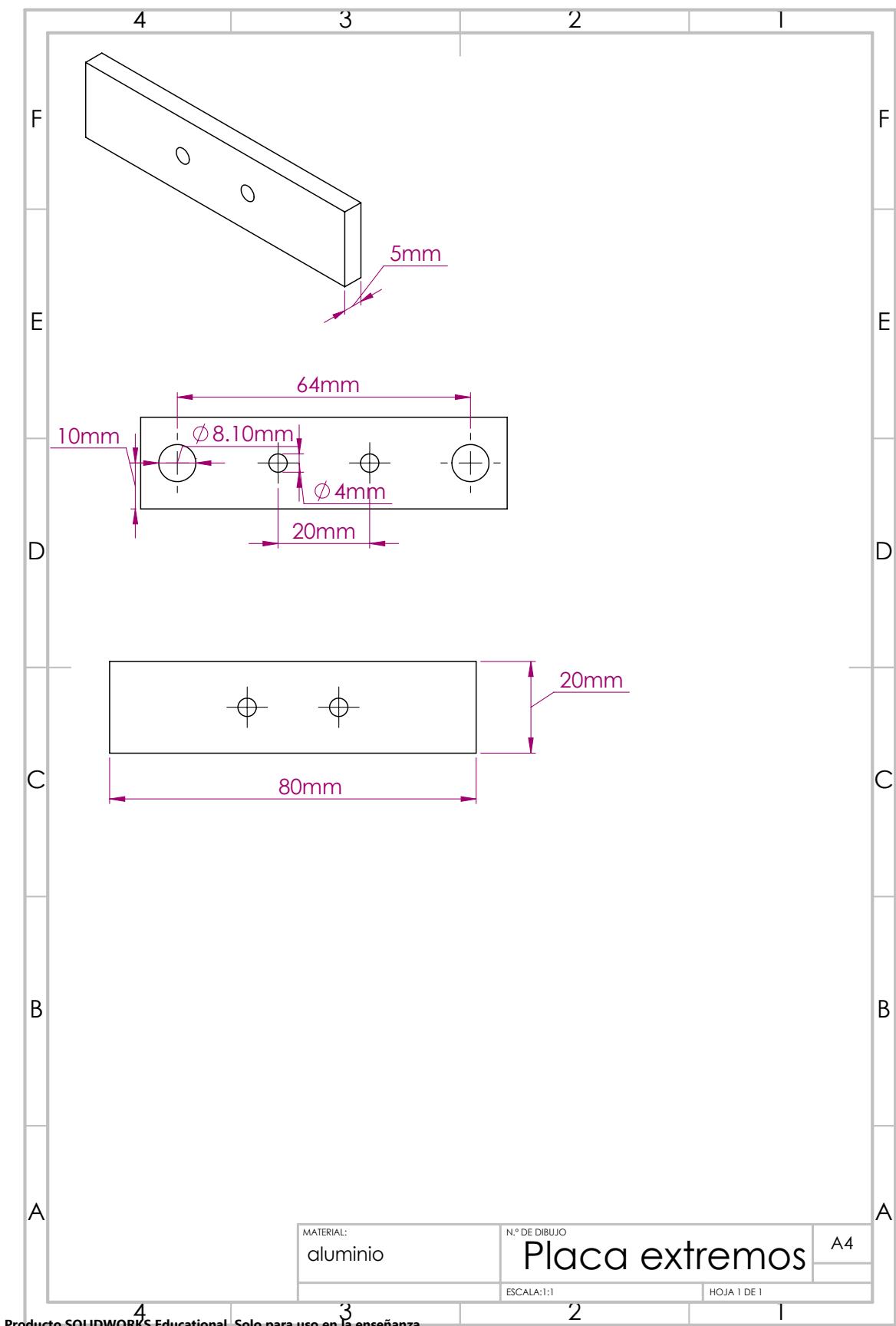


Figura 30: Plano del motor.<sup>32</sup>



Producto SOLIDWORKS Educational. Solo para uso en la enseñanza.

Figura 31: Plano placa extremo.<sup>33</sup>

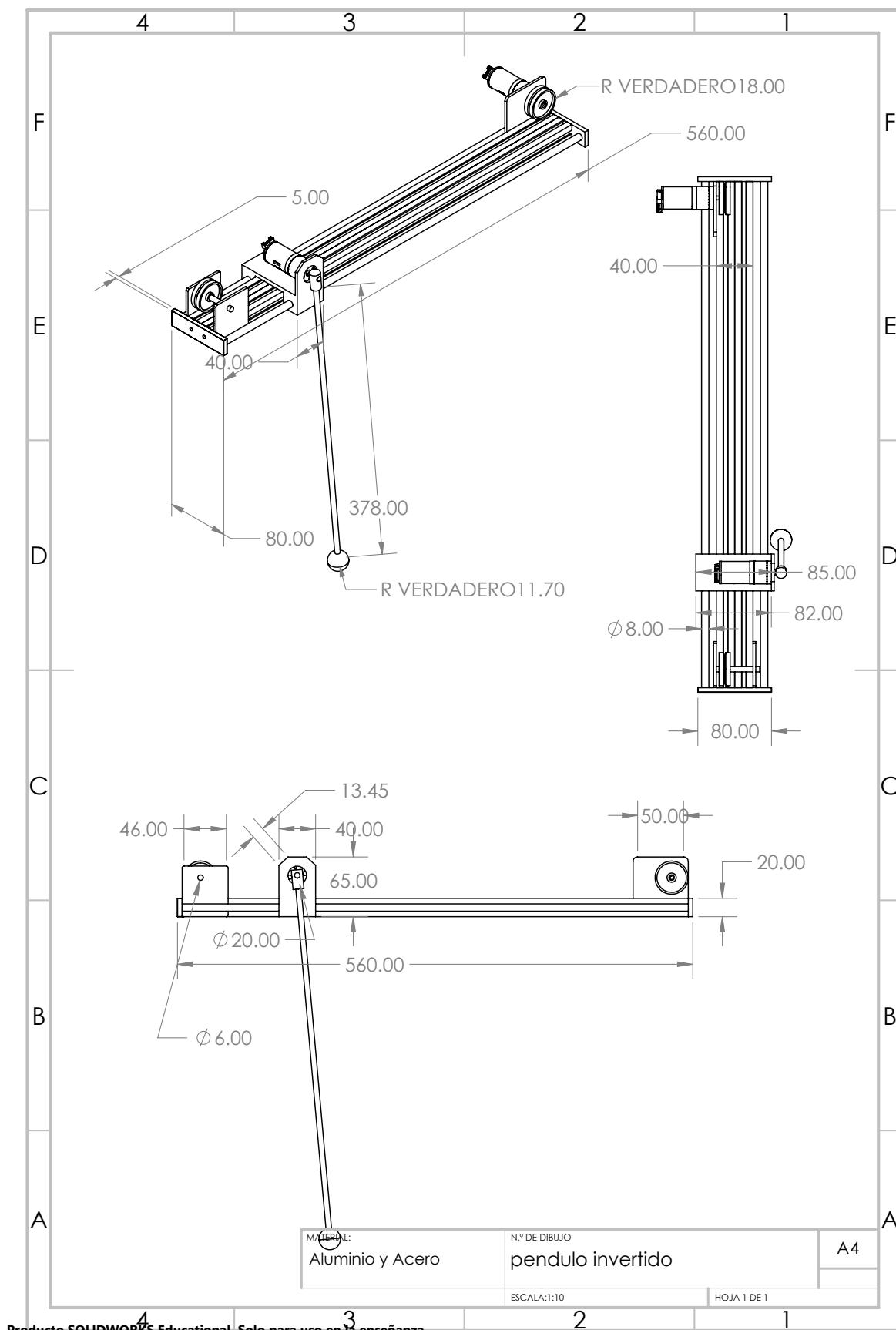


Figura 32: Ensamble pendulo.