

Guardameta Robótico

1st Samantha López Vizcarra
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México.
A01742394@tec.mx

2nd Dulce María Naranjo
Sarmiento
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México.
A00832765@tec.mx

5th Jesús Antonio López Malacón
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México.
A01742257@tec.mx

3rd Fátima Pamela Ramón
Quintana
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México.
00833076@tec.mx

6th Diego Flores Martínez
Escuela de Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México
A01571205@tec.mx

4th Juan Alberto Moreno Cantú
Ingeniería y Ciencias
Instituto Tecnológico y de
Estudios Superiores de
Monterrey
Monterrey,N.L. México.
A00833076@tec.mx

Resumen — Este documento pretende mostrar una implementación de las diferentes áreas que integran la ingeniería mecatrónica, tales como electrónica, programación, diseño y control. Esto se logra a través de un prototipo físico de un jugador de fútbol 2-DOF, hecho a través de programación Arduino, impresión 3D, diseño SolidWorks y otras habilidades enseñadas durante la materia de Implementación de sistemas mecatrónicos.

Palabras clave — *micro-controlador, control, impresión 3D, tolerancias, electrónica.*

I. INTRODUCCIÓN

Este trabajo tiene como objetivo el diseño y construcción de un robot portero de fútbol de 2 grados de libertad traslacionales capaz de bloquear los disparos de un balón, esto a través de los conocimientos adquiridos en el bloque de Implementación de sistemas mecatrónicos, que integró conocimientos en las áreas de la electrónica, programación, diseño e impresión 3D.

La carrera a la que pertenece este proyecto es ingeniería mecatrónica, y se pretende obtener una noción y profundización en la carrera a través del modelado físico de este dispositivo, así como la implementación de metodologías utilizadas en el modelado y diseño de sistemas mecatrónicos que existen en el mundo profesional.

Es importante mencionar que la realización de este trabajo se realizó de manera colaborativa, con un equipo de trabajo conformado por los integrantes mencionados anteriormente, y con la asesoría de los profesores encargados de cada módulo que compone al bloque..

II. MÓDULO MICROCONTROLADORES

A. Arduino

El Arduino es un microcontrolador, el cual ayuda a ejecutar los programas cargados desde un ordenador funcionando de manera independiente de este, a su vez controla y alimenta diferentes dispositivos electrónicos, tomando decisiones de acuerdo al programa ejecutado y guardado en él, gracias a sensores y actuadores, es que este microcontrolador puede interactuar con el entorno físico.

En la Fig.1 se puede observar el diagrama esquemático del Arduino UNO Rev3.

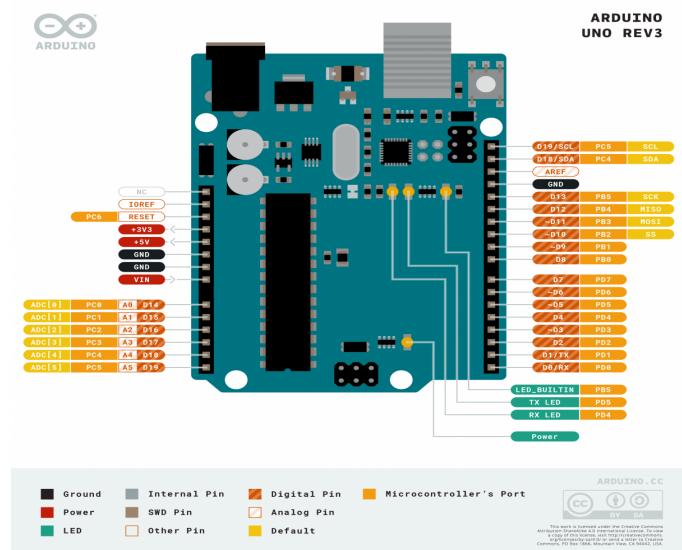


Fig. 1. Esquema de Arduino UNO Rev 3 [1]

Cabe mencionar, que la placa de Arduino UNO cuenta con una serie de características y componentes que la hacen versátil, así como que su manejo sea fácil.

Algunas de sus características incluyen:

- Microcontrolador:** El ATMega328P que se encuentra en el Arduino Uno tiene una velocidad de reloj de 16MHz y una capacidad de memoria de programa de 32 KB, lo que permite la ejecución de instrucciones y el almacenamiento de programas.
- Puertos de E/S:** El Arduino UNO tiene 14 pines digitales de entrada/salida, de los cuales 6 pueden utilizarse como salidas PWM (modulación por ancho de pulso) y 6 como entradas analógicas. Estos pines permiten la conexión y control de componentes electrónicos como sensores, actuadores y pantallas.
- Conectividad:** La placa cuenta con un puerto USB que permite la conexión con una computadora para la programación y la comunicación serial. También tiene un conector de alimentación, pines de alimentación de 5V y 3.3V, y pines de tierra para una fácil conexión de los componentes externos.
- Programación:** El Arduino UNO se programa utilizando el lenguaje de programación de Arduino, que es una variante simplificada de C/C++. Se proporciona un entorno de desarrollo integrado (IDE) gratuito y de código abierto que facilita la escritura, carga y depuración del código en la placa.
- Compatibilidad:** El Arduino UNO es compatible con una amplia gama de shields (placas de expansión) que permiten agregar funcionalidades adicionales a la placa, como conectividad WiFi, Ethernet, Bluetooth, controladores de motor, entre otros.

En la TABLA I se presenta la documentación del Arduino UNO para su funcionamiento, mientras que en la TABLA II el consumo de energía que este necesita.

TABLA I. CONDICIONES DE FUNCIONAMIENTO RECOMENDADAS

Condiciones de Funcionamiento Recomendadas		
Descripción	Mínimo	Máximo
Límites térmicos conservadores para todo el tablero:	-40 °C (-40°F)	85 °C (185°F)

TABLA II. CONSUMO DE ENERGÍA

Símbolo	Consumo de energía			
	Descripción	Mínimo	Máximo	Unidad
VINMax	Voltaje máximo de entrada desde la almohadilla VIN	6	20	V

Símbolo	Consumo de energía			
	Descripción	Mínimo	Máximo	Unidad
VUSBMax	Voltaje máximo de entrada desde el conector USB	-	5.5	V
Pmax	Consumo máximo de energía	-	XX	mA

B. Motor STEPPER

El motor NEMA 17 es conocido por su tamaño compacto y su alta precisión. Es ampliamente utilizado en aplicaciones que requieren un control preciso del movimiento, como impresoras 3D, máquinas CNC, robótica, sistemas de posicionamiento, automatización industrial y otras aplicaciones similares.

Es un motor paso a paso, lo que significa que se mueve en pasos discretos en respuesta a señales eléctricas. Estos motores constan de un rotor y un estator, y el movimiento se produce mediante la activación secuencial de las bobinas del estator. Cada vez que se activa una bobina, el rotor gira una fracción de paso predefinida, lo que permite un control preciso del movimiento y la posición. A su vez, cabe mencionar que cuenta con un ángulo de paso de 1.8 grados por paso, lo que significa que se requieren 200 pasos para completar una rotación completa.

En la Fig.2. se puede observar el diagrama esquemático del Motor NEMA 17.

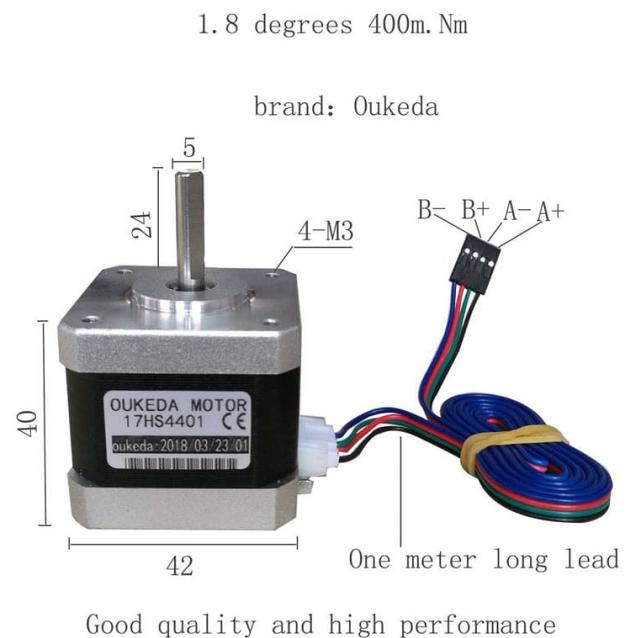


Fig. 2. Esquema de Motor NEMA 17 [2]

A continuación en la TABLA III. y TABLA IV. Se presentan los datos del motor NEMA 17.

TABLA III. MOTOR NEMA 17

Motor NEMA 17		
Descripción	Unidad	Rango de valores
Potencia de entrada, Nominal ($\pm 10\%$)	VDC	14-48
Potencia de entrada auxiliar, Nominal ($\pm 10\%$)	VDC	6-24
Potencia de entrada auxiliar, Máximo	W	1
Torque de detención	mNm	15
Límite de carga de empuje	Kg	0.28
Límite de carga radial (desde el extremo del eje)	N	20
Inercia del rotor	g-cm ²	57
Torque de retención en corriente continua	Nm	0.35
Torque de retención en la corriente pico	Nm	0.5
Corriente de salida continua	A	1.8
Corriente de salida máxima (depende de la aplicación)	A	3.5
Ángulo de paso	Deg	1.8
Codificador,Magnético resolución	ppr	4096
Pérdida de circuito	W	6
Peso	Kg	0.37
Hardware de conexión Tamaño/par de tornillo	Nm	0.63
Protección de sobretensión	VDC	Lógico

TABLA IV. CONTROL MOTOR NEMA 17

Motor NEMA 17		
Elemento	Descripción	Especificación
Modos de operación	Selezionable	Posición de perfil, Velocidad, Velocidad de perfil, Par de perfil, Homing, Posición síncrona cíclica
Display	-	LED bicolor

Motor NEMA 17		
Elemento	Descripción	Especificación
	Interfaz de usuario	ServoStudio, basado en Windows
Herramientas de software	Funciones	Configuración de conexión, información de la unidad, información de energía, configuración de E/S, configuración y ajuste de movimiento, historial/pantalla de fallas
	Posición	Cuenta
Unidades rotativas	Velocidad	rpm/100
	Acc/Dec	rpm/100/s

C. Driver

El driver A4988 permite controlar con precisión y facilidad el movimiento de motores paso a paso bipolares. Algunas de sus características más destacadas son las siguientes:

- **Control de corriente:** El A4988 incorpora un sistema de regulación de corriente que permite ajustar la intensidad que fluye a través de las bobinas del motor paso a paso. Esto es importante para garantizar el correcto funcionamiento del motor y evitar el sobrecalentamiento.
- **Micropasos:** El A4988 es capaz de realizar micropasos, una técnica que permite subdividir cada paso del motor en fracciones más pequeñas. Esto se traduce en un movimiento más suave y preciso, así como en una menor vibración y ruido del motor. El A4988 puede realizar microstepping de hasta 1/16 de paso.
- **Control de dirección:** El driver A4988 permite controlar la dirección del motor paso a paso mediante un pin específico. Esto facilita la programación y el control del motor en proyectos.
- **Interfaz de control:** El A4988 utiliza una interfaz de control sencilla que se puede conectar directamente a un microcontrolador, como Arduino, mediante pines digitales. El control de la velocidad y el movimiento se realiza a través de la secuencia de pulsos enviados a los pines de control del A4988.

En la Fig.3. Se puede observar el diagrama esquemático del Driver A4988.

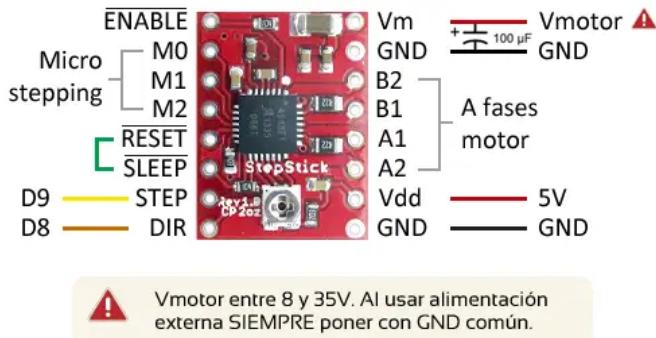


Fig. 3. Esquema de Driver A498 [3]

TABLA V. ÍNDICES ABSOLUTOS MÁXIMOS

Índices absolutos máximos			
Características	Sim	Rango	Unidad
Voltaje de suministro de carga	VBB	35	V
Corriente de salida	IOUT	± 2	A
Voltaje de entrada lógica	VIN	-0.3 to 5.5	V
Voltaje de alimentación lógica	VDD	-0.3 to 5.5	V
Voltaje de salida del motor		-2.0 to 37	V
Voltaje de detección	VSENSE	-0.5 to 0.5	V
Voltaje de referencia	VREF	5.5	V
Temperatura ambiente de funcionamiento	TA	-20 to 85	°C
Unión máxima	TJ(max)	150	°C
Temperatura de almacenamiento	Tstg	-55 to 150	°C

TABLA VI. CARACTERÍSTICAS ELÉCTRICAS

Características eléctricas						
Características	Sim	Condición de prueba	Min	Tipo 2	Máx	Unidad
Controladores de salida						
Rango de tensión de alimentación lógica	VBB	Operando	8	—	35	V
Rango de tensión de alimentación lógica	VD D	Operando	3.0	—	5.5	V
Salida de la resistencia	RDSON	Fuente Conducto, IOUT = -1.5 A	—	320	430	mΩ
		Sink Driver, IOUT = 1.5 A	—	320	430	mΩ
Tensión directa del diodo		Source Diode, IF = -1.5 A	—	—	1.2	V
		Sink Diode, IF = 1.5 A	—	—	1.2	V
Fuente de alimentación del motor		fPWM < 50 kHz	—	—	4	mA
		En funcionamiento, salidas desactivadas	—	—	2	mA

Características eléctricas						
Características	Sim	Condición de prueba	Min	Tipo 2	Máx	Unidad
Controladores de salida						
Corriente de alimentación lógica		fPWM < 50 kHz	–	–	8	mA
		Salidas apagadas	–	–	5	mA
Control Logic						
Voltaje de entrada lógica	VIN(1)		VDD' 0.7	–	–	V
	VIN(0)		–	–	VDD' 0.3	V
Corriente de entrada lógica	IIN(1)	VIN = VDD'0.7	–20	<1.0	20	µA
	IIN(0)	VIN = VDD'0.3	–20	<1.0	20	µA
Seleccionar micropaso	RMS1	MS1 pin	–	100	–	kΩ
	RMS2	MS2 pin	–	50	–	kΩ
	RMS3	MS3 pin	–	100	–	kΩ
Histeresis de entrada lógica	VHYS (IN)	As a % of VDD	5	11	19	%
Tiempo en blanco	tBLANK		0.7	1	1.3	µs
Fuera de tiempo fijo		OSC = VDD or GND	0	30	40	µs

Características eléctricas						
Características	Sim	Condición de prueba	Min	Tipo 2	Máx	Unidad
Controladores de salida						
		ROSC = 25 kΩ	2 3	30	37	µs
Rango de tensión de entrada de referencia	VREF		0	–	4	V
Corriente de entrada de referencia	IREF		–3	0	3	µA
Error de nivel de disparo actual		VREF = 2 V, %ITripMAX = 38.27%	–	–	±15	%
		VREF = 2 V, %ITripMAX = 70.71%	–	–	±5	%
		VREF = 2 V, %ITripMAX = 100.00%	–	–	±5	%
Tiempo muerto cruzado	tDT		100	475	800	ns
Protection						
Umbral de protección de sobreintensidad	IOCPST		2.1	–	–	A
Temperatura de desconexión térmica	TTSD		–	165	–	°C

Características eléctricas						
Características	Sim	Condición de prueba	Min	Tipo 2	Máx	Unidad
Controladores de salida						
Histéresis de apagado térmico	TTSD HYS		-	15	-	°C
Bloqueo de subtensión VDD	VDD UVLO	Subida de VDD	2.7	2.8	2.9	V
VDD Histéresis de subtensión	VDD UVLOHYS		-	90	-	mV

TABLA VII. CARACTERÍSTICAS TÉRMICAS

Características térmicas				
Características	Sim	Condición de prueba	Rango	Unidad
Resistencia térmica del paquete	R _{θJA}	PCB de cuatro capas, basado en el estándar JEDEC	32	C/W

D. Joystick

Se refiere a un módulo o componente electrónico que se utiliza para controlar la posición o movimiento en proyectos y aplicaciones basados en Arduino. El joystick para Arduino es una interfaz de entrada analógica que permite al usuario interactuar con el microcontrolador y controlar el comportamiento de un programa o sistema.

El joystick para Arduino generalmente consta de dos potenciómetros (*uno para el eje X y otro para el eje Y*) que registran el movimiento en dos dimensiones. Estos potenciómetros generan señales analógicas proporcionales a la posición del joystick en cada eje.

Al conectar el joystick a las entradas analógicas de Arduino, se pueden leer estos valores para determinar la posición del joystick en tiempo real.

En la Fig.4. Se puede observar el diagrama esquemático del Joystick.

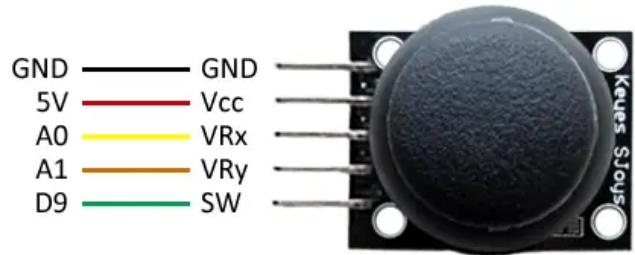


Fig. 4. Esquema de Joystick [4]

TABLA VIII. DATOS JOYSTICK

Datos Joystick		
Número de pin	Nombre del pin	Descripción
1	GND	Tierra de terminal a módulo
2	+5V	Alimentación de la terminal al módulo
3	VRX	Voltaje Proporcional al módulo x
4	VRY	Voltaje proporcional al módulo y
5	SW	Switch

E. Código Arduino Pantalla

```
// include the library code:  
#include <LiquidCrystal.h>
```

```
const int IR = A0; // Pin analógico para el sensor infrarrojo  
const int buzzerPin = 6; // Pin digital para el buzzer  
int goles = 0; //contador  
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
//pins del display  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
//Icons  
byte face[8] = { // Array of bytes  
  B00000, // B stands for binary formatter and the five  
  numbers are the pixels  
  B00000,  
  B01010,  
  B00000,  
  B01110,  
  B10001,  
  B00000,
```

```

B00000
};

byte smile[8] = {
B00000,
B00000,
B01010,
B00000,
B10001,
B01110,
B00000,
B00000
};

//Inicializar entradas
void setup() {
  Serial.begin(9600);
  pinMode(IR,INPUT) ; //Sensor infrarrojo como entrada
  pinMode(buzzerPin, OUTPUT); // Configurar el pin del
buzzer como salida
  lcd.begin(16, 2); // set up the LCD's number of columns
and rows
  lcd.createChar(0,face); // Create a custom character :(
  lcd.createChar(1, smile); //:)

}

// LOOP
void loop() {
//SENSOR
  int value = analogRead(IR); // Lectura analógica del sensor
  Serial.println(value);

// Lógica del sensor infrarrojo y buzzer
  if(value < 550) {
    goles = goles+1; // Incrementar el número de goles
    tone(buzzerPin, 500, 1000); // Sonido Buzzer Frecuencia:
500Hz, Duración: 1000ms
    delay(1000); // Esperar para incrementar goles
  }

  else {
    noTone(buzzerPin); // Detener el sonido del buzzer si no
hay obstáculo
  }

//DISPLAY
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with
0):
  lcd.setCursor(0, 0);
  lcd.print("Goles:");
  lcd.setCursor(0, 1);
  lcd.print(goles);
  if(goles == 0)
  {

    lcd.setCursor(4, 1);
    lcd.write(byte(1)); // Display the custom character 1, the
face :(
  }
  else
  {
    lcd.setCursor(4, 1);
    lcd.write(byte(0)); // Display the custom character 0, the
face :(
  }
}

```

F. Código Arduino Final

```

// include libraries
#include <LiquidCrystal.h> //Display library
#include <AccelStepper.h> //accelstepper library

//-----
//Pins
//Stepper
const byte Analog_Y_pin = A1; //y-axis readings
const int pinJoyButton = 9;
const int MaxSpeed = 2000; //Velocidad máxima

//Display
const int IR = A0; // Pin analógico para el sensor infrarrojo
const int buzzerPin = 6; // Pin digital para el buzzer
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
//pins del display

//-----
//Components
AccelStepper stepper(1, 3, 2); // direction 2, Step 3
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

//-----
//Variables
//Stepper
int Analog_Y = 0; //y-axis value
//Display-Buzzer-IR
byte face[8] = { // Array of bytes
  B00000,           // B stands for binary formatter and the five
numbers are the pixels
  int v = 0; //velocidad
  int goles = 0; //contador

//-----
//Icons
  B00000,
  B01010,
  B00000,
  B01110,
  B10001,

```

```

B00000,
B00000
};
byte smile[8] = {
B00000,
B00000,
B01010,
B00000,
B10001,
B01110,
B00000,
B00000
};

//-----
-----  

void setup()
{
  Serial.begin(9600);
  //IR
  pinMode(IR,INPUT) ; //Sensor infrarrojo como entrada

  //BUZZER
  pinMode(buzzerPin, OUTPUT); // Configurar el pin del
  buzzer como salida

  //DISPLAY
  lcd.begin(16, 2); // set up the LCD's number of columns
  and rows
  lcd.createChar(0, face); // Create a custom character :(
  lcd.createChar(1, smile); //:)

  //JOYSTICK
  pinMode(Analog_Y_pin, INPUT);

  //STEPPER PARAMETERS
  stepper.setMaxSpeed(MaxSpeed); //SPEED = Steps / second
  stepper.setAcceleration(500); //ACCELERATION = Steps
  /(second)^2
  stepper.setSpeed(0);
}

//-----
-----  

void loop()
{
  //VELOCIDAD
  ReadAnalog();
  stepper.runSpeed(); //step the motor (this will step the motor
  by 1 step at each loop indefinitely)

  //-----
-----  

//SENSOR
int value = analogRead(IR); // Lectura analógica del sensor
Serial.println(value);

// Lógica del sensor infrarrojo y buzzer
if (value < 550) {
  goles = goles+1; // Incrementar el número de goles
  tone(buzzerPin, 500, 1000); // Sonido Buzzer Frecuencia:
  500Hz, Duración: 1000ms
  delay(1000); // Esperar para incrementar goles
}
else {
  noTone(buzzerPin); // Detener el sonido del buzzer si no
  hay obstáculo
}

//DISPLAY
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with
0):
lcd.setCursor(0, 0);
lcd.print("Goles:");
lcd.setCursor(0, 1);
lcd.print(goles);
if (goles == 0)
{
  lcd.setCursor(4, 1);
  lcd.write(byte(1)); // Display the custom character 1, the
  face :)
}
else
{
  lcd.setCursor(4, 1);
  lcd.write(byte(0)); // Display the custom character 0, the
  face :(
}

//-----
-----  

//Función Variables Joystick a Velocidad
void ReadAnalog()
{
  Analog_Y = analogRead(Analog_Y_pin);

  if(abs(Analog_Y)>600 || abs(Analog_Y)<400)
  {
    v = map(Analog_Y, 0, 1023, -MaxSpeed, MaxSpeed);
    stepper.setSpeed(v);
  }
  else
  {

```

```

stepper.setSpeed(0);
}

```

}

G. Esquemático de la interconexión de estos elementos

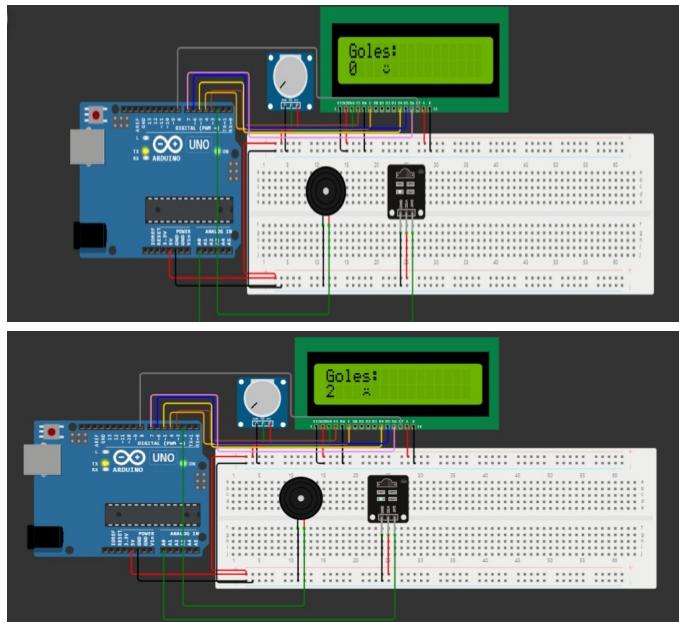


Fig. 5. Pantalla Contador de Goles con Sensor de Obstáculos

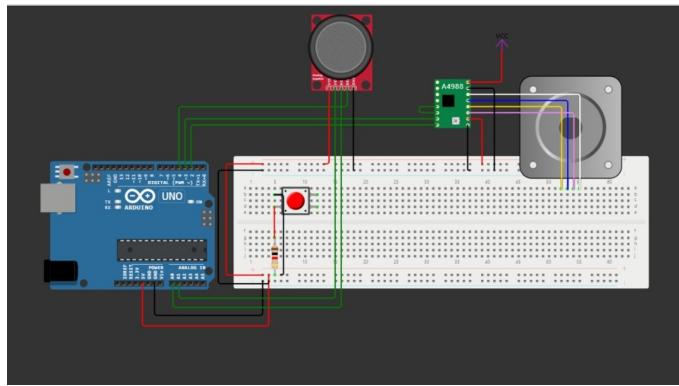


Fig. 6. Conexión del motor, botón y Joystick

III. MÓDULO DESARROLLO DE PROTOTIPOS Y SUS TECNOLOGÍAS

En esta sección se plantea todo lo necesario para el prototipo final del reto del guardameta robótico, para el cual se implementaron las medidas y tolerancias requeridas expuestas en el documento del proyecto, las cuales se pueden visualizar en el apartado IV. *MÓDULO ELEMENTOS DE MÁQUINAS*, esto con la intención de que no sea muy repetitiva la información presentada.

La TABLA IX, presenta los materiales, junto con los precios de los elementos requeridos para el prototipo.

TABLA IX. MATERIALES Y PRECIOS

Inversión en el prototipo		
Elemento	Lugar	Precio
Perfil V	Amazon	\$125.25 c/u
Carrito	Amazon	\$214
Motor A pasos	Mercado Libre	\$178
Driver De Motor A Pasos	Mercado Libre	\$59
Modulo Joystick Arduino Dos Ejes (xy)	NA	\$45
Solenoides	Mercado Libre	\$135
Arduino	NA	NA
Limit Switch	Mercado Libre	\$79 2 pzs
Filamento	Amazon	\$548.63
Tornillo botón milimétrico acero inox	BIRLOS Y TORNILLOS	\$3.30
Tornillo botón m.m inox 5x12		\$15.60
Tornillo botón m.m inox 4x16		\$7.14
Tornillo botón m.m inox 5x16		\$10.80
Tuerca para perfil 5mm		\$43.20
Tornillo botón m.m inox 4x50		\$34.32
Tuerca hex inox 4mm		\$3.84
Correa Dentada	Amazon	\$327.00
Transistores	Prov Local	...
MDF	Planno	\$180 2 laminas de 1 m x 1.2 m

Cabe mencionar que algunas piezas se tomó la decisión de imprimirlas en 3D por los detalles que estas presentaban, así como el hecho de agilizar tiempos y minimizar gastos, mientras que para la carcasa del prototipo se tomó la decisión de que fuera de MDF ya que iba a tener que soportar el peso de algunos elementos a implementar, así como que sus dimensiones son muchísimo mayor y corriámos el riesgo de que en impresión 3D se presentarán algunos errores, aparte de

que también minimizamos tiempos y costos al utilizar el material elegido.

A. Montura Limit Switch

En la Fig.7 se puede observar la pieza donde el Limit Switch será montado en el prototipo construido, la cual se decidió imprimir en 3D por todos los detalles con los que cuenta, aparte de que es una pieza pequeña por lo que no necesita demasiado material.

Mientras que en la Fig.8 se muestra el tiempo (*1hr 22min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes en la base solamente, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

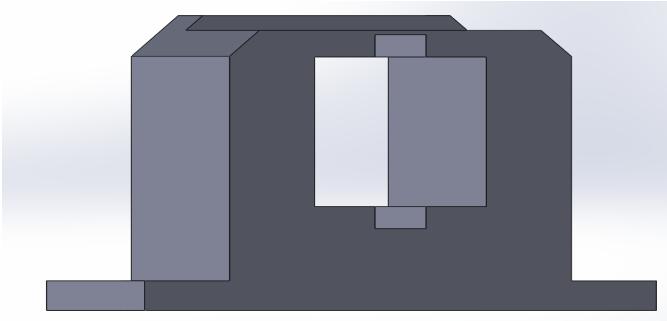


Fig. 7. Pieza Montura Limit Switch

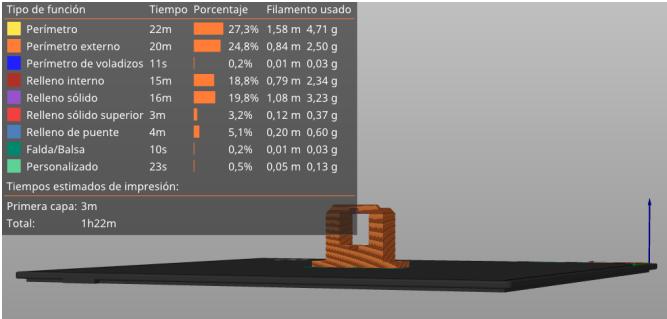


Fig. 8. Tiempo de impresión de Montura Limit Switch

B. Montura Motor NEMA 17

En la Fig.9 se puede observar la pieza donde el Motor NEMA 17 será montado en el prototipo construido, la cual se decidió imprimir en 3D por todos los detalles con los que cuenta, aparte de que es una pieza pequeña por lo que no necesita demasiado material.

Para sostener la pieza en el prototipo se utilizaron tornillos de distintas medidas.

Mientras que en la Fig.10 se muestra el tiempo (*2 hr 42 min*) en el que se tardó en imprimir la pieza en su totalidad, la cual cuenta con soportes en la base solamente, cabe mencionar

que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*), el resultado de la impresión fue bueno, ya que no tenía hilos, ni fue necesario hacer un retrabajo.

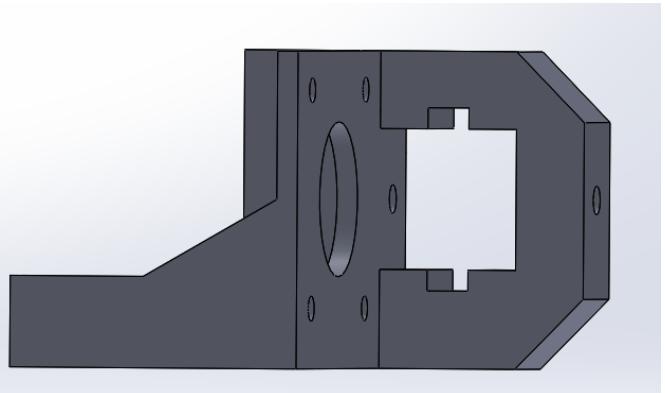


Fig. 9. Pieza Montura Motor NEMA 17



Fig. 10. Tiempo de impresión Montura Motor NEMA 17

C. Montura Polea A

En la Fig.11 se puede observar la pieza donde la polea será montada en el prototipo construido, se tomó la decisión de realizarla en impresión en 3D por todos los detalles con los que cuenta, aparte de que es una pieza pequeña por lo que no necesita demasiado material. Para sostenerla en el prototipo se utilizaron tornillos.

Mientras que en la Fig.12 se muestra el tiempo (*1 hr 17 min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes en la base solamente, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

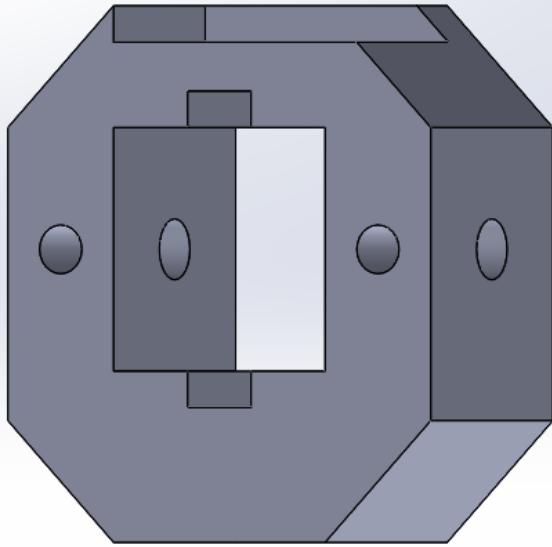


Fig. 11. Pieza Montura Polea A

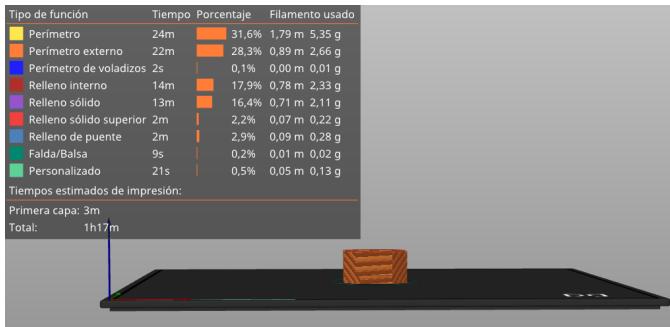


Fig. 12. Tiempo de impresión Montura Polea A

D. Montura Polea B

En la Fig.13 se puede observar la pieza donde la polea será montada en el prototipo construido, la cual se imprimió en 3D, debido a los detalles con la que esta cuenta, aparte de la complejidad que sería hacer estas piezas en MDF, aparte de que es una pieza pequeña por lo que no necesita demasiado material.

Mientras que en la Fig.14 se muestra el tiempo (*36 min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes en la base solamente, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

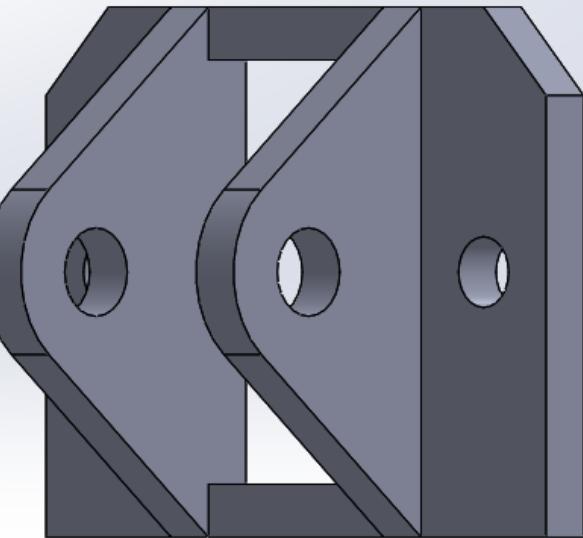


Fig. 13. Pieza Montura Polea B

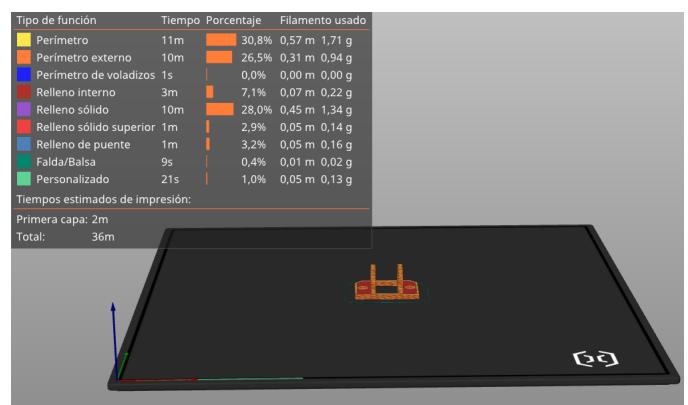


Fig. 14. Tiempo de impresión de la Polea B

E. Soporte de Portero

En la Fig.15 se puede observar la pieza que es el soporte donde el portero será montado en el prototipo construido, se tomó la decisión de realizarla en impresión en 3D por todos los detalles con los que cuenta, aparte de que es una pieza regularmente pequeña por lo que no necesita demasiado material. Para sostenerla en el prototipo se utilizaron tornillos.

Mientras que en la Fig.16 se muestra el tiempo (*3 hr 58 min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes en todos los sitios, esto con la intención de disminuir los riesgos de que haya hilos en la impresión o cualquier otro defecto, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

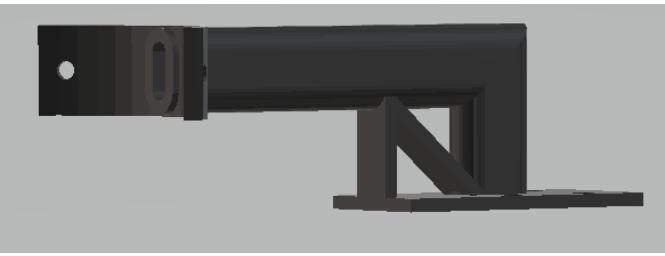


Fig. 15. Pieza Soporte de Portero

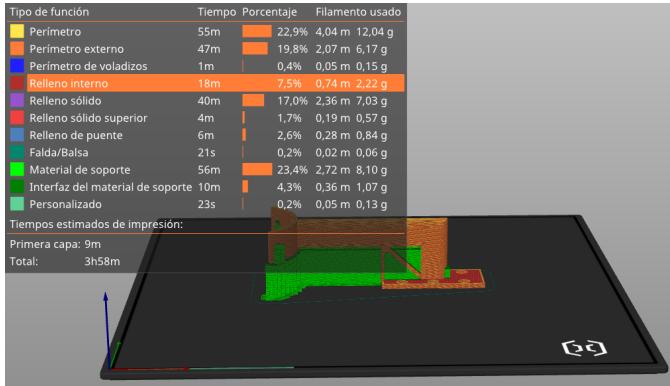


Fig. 16. Tiempo de impresión de Soporte de Portero

F. Base del portero

En la Fig.17 se puede observar la pieza que es la base del portero, esta cuenta con agujeros en la parte de abajo y en el costado para sostenerse al soporte del portero mediante tornillos. La pieza se imprimió en 3D debido a los detalles con lo que cuenta.

Mientras que en la Fig.18 se muestra el tiempo (*1 hr 46 min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes solamente en la base, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

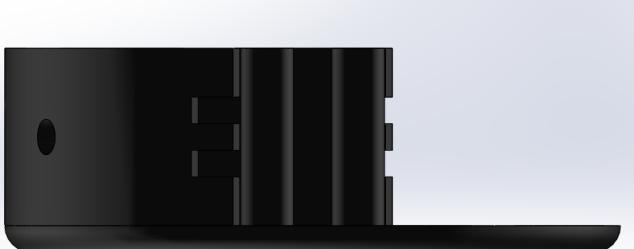


Fig. 17. Pieza Base del Portero

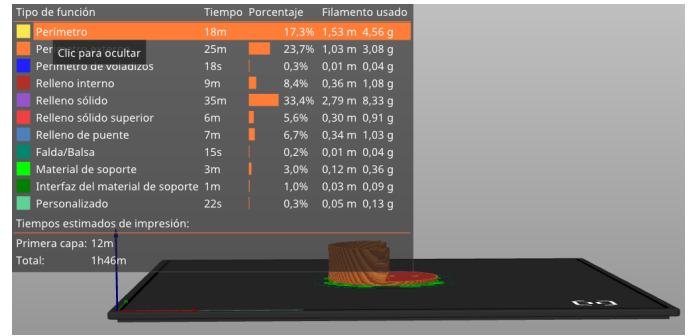


Fig. 18. Tiempo de impresión de la base del portero

G. Parte del Portero - Golpe

En la Fig.19 se puede observar la pieza encargada de realizar el golpe del portero, la cual cuenta con un espacio para el solenoide, la manera en la que esta será sostenido con la base del portero y la tapadera de este, es por medio de tornillos. Al igual que las piezas anteriores esta también se realizó en impresión en 3D.

Mientras que en la Fig.20 se muestra el tiempo (*1 hr 32 min*) en el que se tardó en imprimir la pieza, la cual cuenta con soportes solamente en la base, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

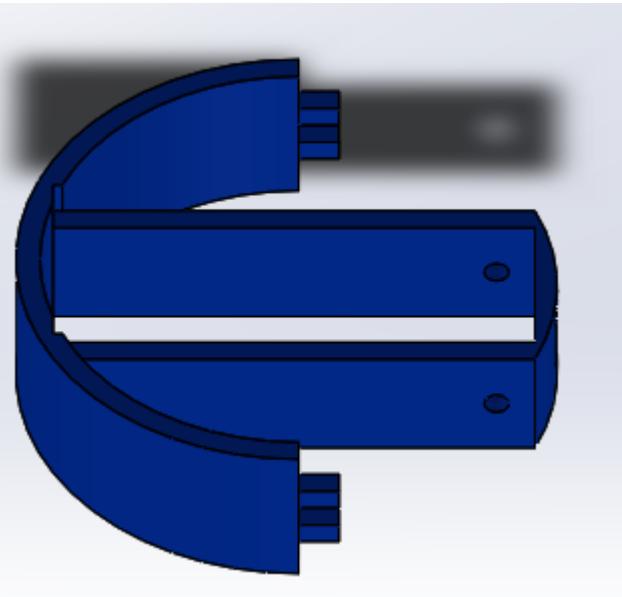


Fig. 19. Pieza del Portero que Golpea

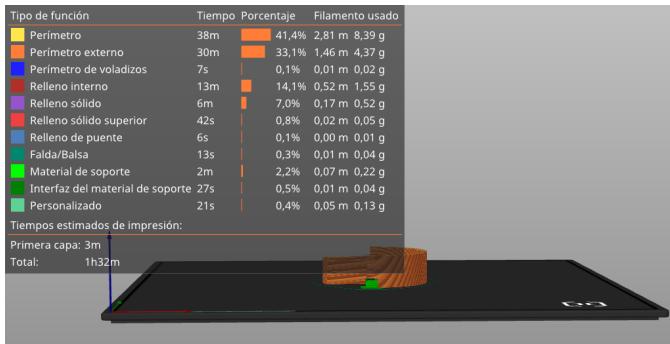


Fig. 20. Tiempo de impresión de Portero - Golpe

H. Tapa del Portero

En la Fig.21 se puede observar la pieza de la tapa del portero, la cual será sostenida al portero mediante tornillos, se realizó en impresión en 3D.

Mientras que en la Fig.22 se muestra el tiempo (58 min) en el que se tardó en imprimir la pieza, la cual cuenta con soportes solamente en la base, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).



Fig. 21. Pieza Tapa de Portero

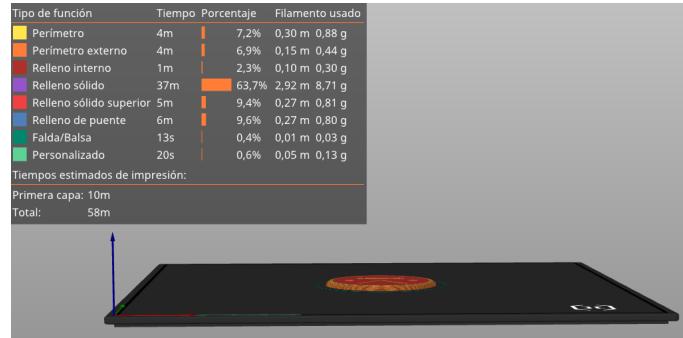


Fig. 22. Tiempo de impresión de la Tapa del Portero

I. Ensamblaje Portero con Solenoide

A continuación en la Fig.23 y Fig.24 , se puede apreciar el ensamblaje del portero junto con el solenoide quien es el que se encarga de la acción de golpear del portero.

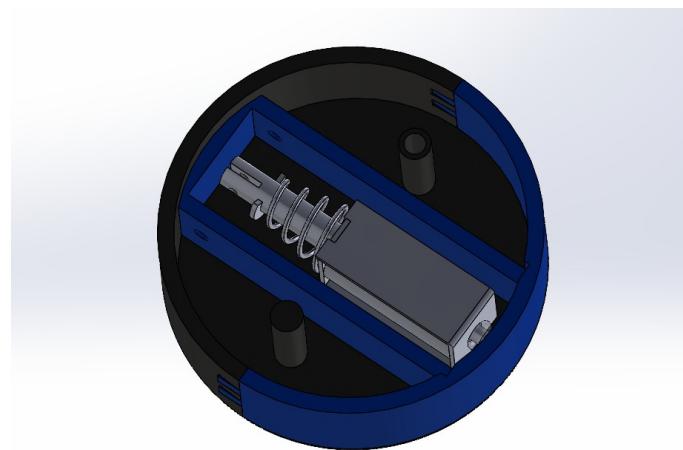


Fig. 23. Ensamblaje del portero sin tapa con el solenoide

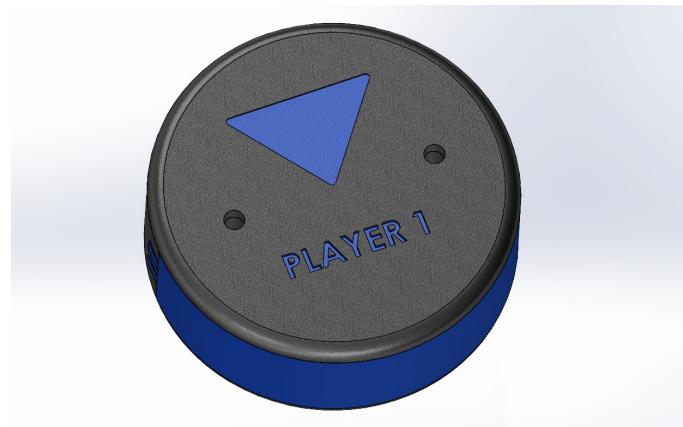


Fig. 24. Ensamblaje del portero completo con el solenoide

J. Control del Prototipo

En la Fig.25 y Fig.27, se puede apreciar la pieza de la caja de control del prototipo, así como la tapa, ambas piezas se decidieron imprimir en 3D, ya que se contaba con el material y por los detalles con las que estas cuentan también era la opción más viable.

Por errores de diseño, la caja y tapa del control no se utilizaron al momento de utilizar el prototipo, se recomienda que para próximos proyectos se tomen medidas de los componentes que necesitan ir en el control, administrando bien los espacios, así como tomar en cuenta los cables. Un error visible fue la profundidad del control, y que el botón a implementar era muy pequeño.

Mientras que en la Fig.26 y Fig.28 se muestra el tiempo (*5 h 09 min - 2 hr 03 min*) en el que se tardó en imprimir ambas piezas, las cuales cuentan con soportes solamente en la base, cabe mencionar que se utilizó el software Prusa Slicer para generar el código g (*lenguaje de programación más utilizado en el mundo para programar máquinas herramienta CNC*).

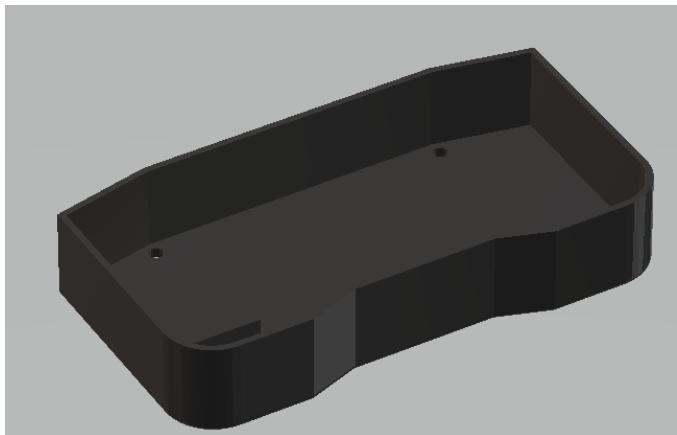


Fig. 25. Pieza Caja del Control

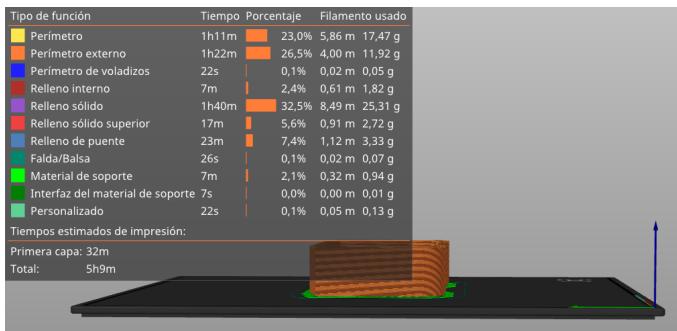


Fig. 26. Tiempo de impresión de la Caja del Control

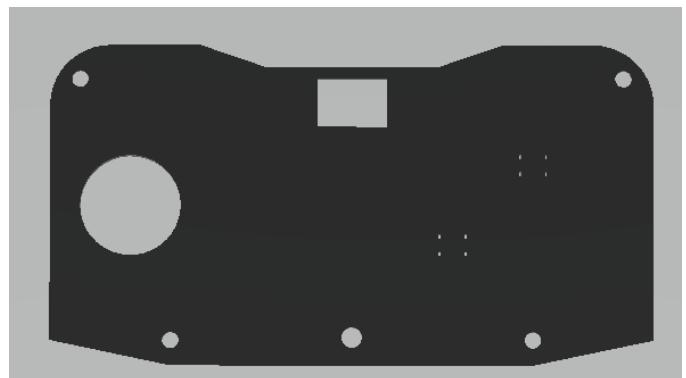


Fig. 27. Pieza Tapa de Control

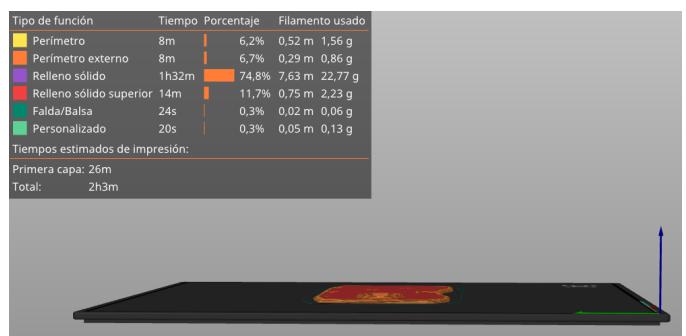


Fig. 28. Tiempo de impresión de Tapa de Control

K. Carcasa del Prototipo

De la Fig.29 a la Fig. 38 se pueden observar las diferentes piezas que se necesitaron para la carcasa del prototipo, la cual se decidió realizar en MDF de 6 mm por su fácil manufactura, así como su accesible precio, esta parte del prototipo no se pudo haber realizado en impresión 3D, de acuerdo a que son piezas de grandes dimensiones a comparación de las anteriores piezas, por lo que se iba a invertir demasiado PLA, así como tiempo, a su vez habría la posibilidad de que no resistiera los pesos de los demás componentes como el motor NEMA 17, el perfil V, así como los movimientos del portero al estar en uso.

El corte total de todas las piezas tuvo una duración de 25 min con un costo de \$250 pesos mexicanos.

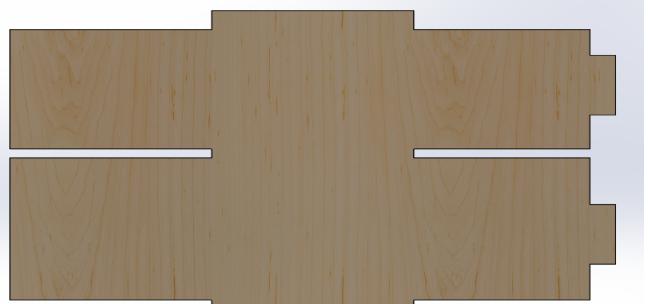


Fig. 29. Pieza Pared Ancho Caja del Prototipo

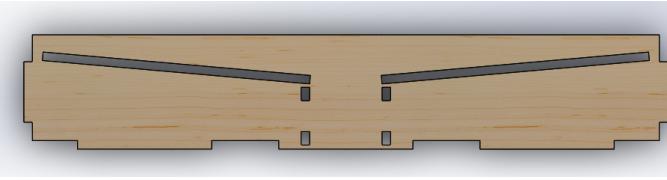


Fig. 30. Pieza Pared Interior de la Caja del Prototipo A

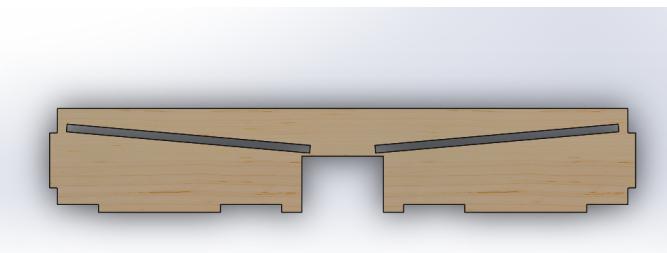


Fig. 31. Pieza Pared Interior de la Caja del Prototipo B



Fig. 32. Pieza Pared

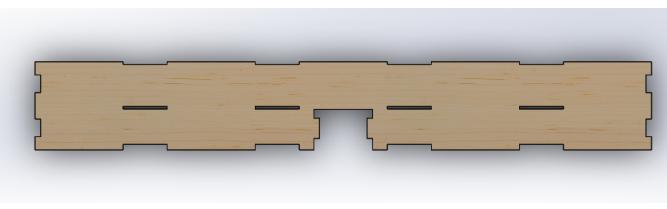


Fig. 33. Pieza Pared Larga del Prototipo



Fig. 34. Pieza Pared del Túnel del Prototipo

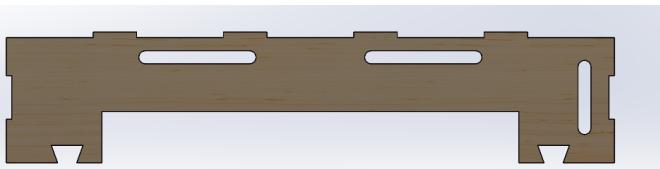


Fig. 35. Pieza Base del Prototipo



Fig. 36. Pieza Rampa del Prototipo

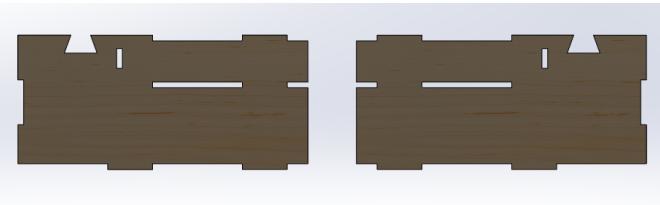


Fig. 37. Pieza Base del Prototipo

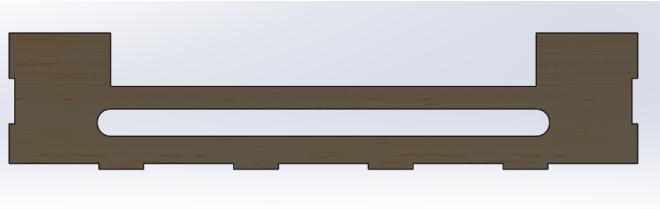


Fig. 38. Pieza Tapa de Caja del Prototipo

En la Fig. 39, se puede visualizar el ensamblaje completo de la carcasa del prototipo, mientras que en la Fig. 40, se puede visualizar finalmente el ensamblaje de todas los componentes del prototipo.

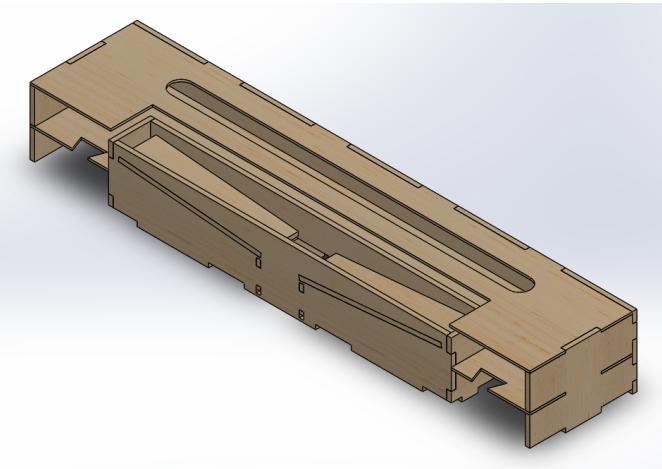


Fig. 39. Ensamblaje de la caja del Prototipo

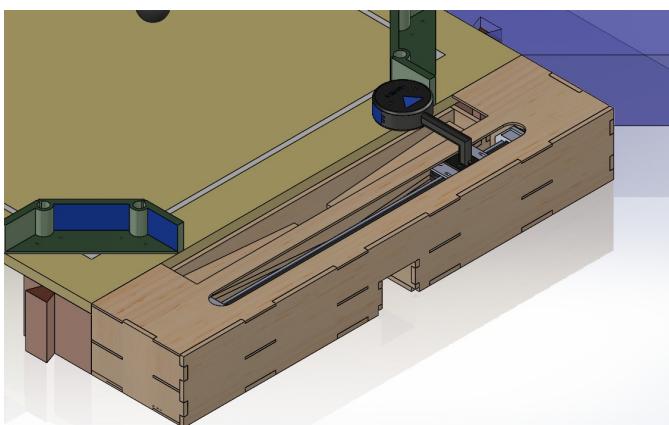


Fig. 40. Ensamblaje del Prototipo Completo

IV. ELEMENTOS DE LAS MAQUINAS

Para la realización del diseño de este proyecto, se tomaron en cuenta los siguientes requerimientos iniciales:

- El prototipo debe respetar las dimensiones previstas por el profesor.
- El prototipo debe ser capaz acoplarse y desacoplarse de manera segura y sencilla a la plataforma provista.
- Prototipo no sea excesivamente complejo ni incluya características innecesarias.
- Los estudiantes deben buscar una solución óptima y eficiente que cumpla con los objetivos del proyecto sin complicar innecesariamente el diseño o aumentar los costos.
- El prototipo debe ser lo más liviano posible.
- Los materiales y componentes deben ser ligeros y resistentes, optimizando así la movilidad y la eficiencia energética del robot.
- Los materiales deben de ser resistentes y la estructura mecánica debe ser adecuada para garantizar la durabilidad y la robustez del robot.

- Los componentes electrónicos deben de estar protegidos y ocultos para mantener una apariencia estética y evitar interferencias externas.

A. Prototipo inicial

Para el prototipo inicial cada miembro del equipo realizó una propuesta de diseño y materiales, con la información obtenida en clase del objetivo del proyecto, para esto cada integrante del equipo votó por una de las propuestas y finalmente se eligió una final, que en este caso fue la mostrada en la Fig. 41.

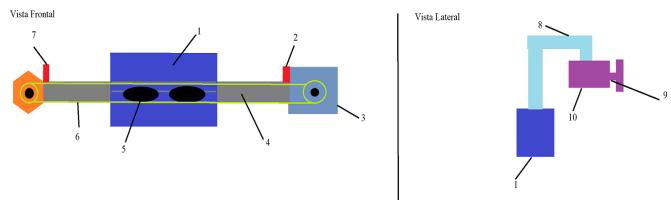


Fig. 41. Prototipo inicial

La lista de materiales propuesta fue la siguiente:

- 1 Carrito
- 2 - 7 Limit switch
- 3 Motor A
- 4 Perfil V de aluminio
- 5 Ruedas V
- 6 Banda y piñón
- 8 Unión carrito y jugador
- 9 Solenoide
- 10 Jugador

Una vez con esta propuesta, se realizó una tabla con los materiales propuestos y los elegidos finalmente para la compra, esto fue aprobado por el maestro, una vez implementados los cambios propuestos (Lista de materiales).

B. Diseño del Portero

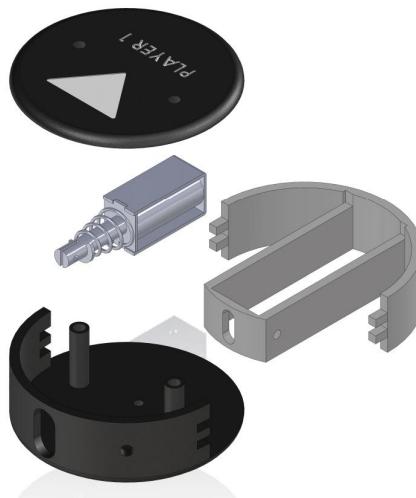


Fig. 42. Vista explosionada del Portero

El jugador mostrado en la Fig.42 fue diseñado en 3 piezas: la tapa, la base que se uniría a un soporte, y la pieza móvil, la cuál sería la que realizaría el golpe con el solenoide.

Para el diseño de cada pieza se tomaron en consideración los siguientes aspectos:

- Grosor de las paredes: las paredes del jugador tenían que tener un grosor de 3mm para la base del jugador.
- Espacio para el solenoide: nuestro solenoide media 5.7 mm, por lo tanto, se consideró dejar un margen lo suficientemente grande en su compartimento para que cupiera sin problemas, pero también lo suficientemente chico pero que el golpe no perdiera mucha carrera.
- Cobrar con agujeros para el atornillado, cableado y ensamble de la base con la pieza que recibiría el golpe.
- Tolerancias para impresión 3D reportadas en este documento.

C. Diseño de soportes para el perfil en V y Nema 17

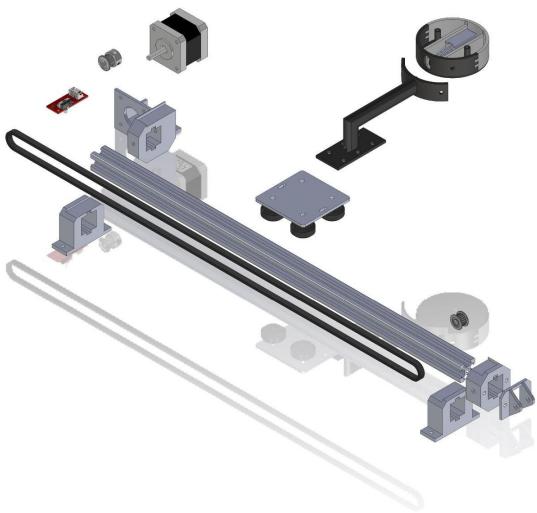


Fig. 43. Vista explosionada del Sistema Interno

En la Fig.43 se muestra la vista explosionada del mecanismo completo, en la parte superior derecha Fig.41 se muestran los soportes destinados para el motor a pasos, estos fueron diseñados con el objetivo de albergar el motor y fijarlo en el sistema, en la vista explosionada igualmente se puede observar la polea de sincronización, la cuál se une al motor a pasos.

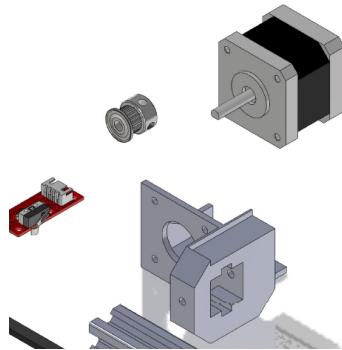


Fig. 44. Vista explosionada de soportes impresos en 3D para el motor a pasos.

Para su diseño fueron considerados los siguientes aspectos:

- Las tolerancias para impresión 3D descritas en este documento
- Posicionamiento de agujeros para fijación en la caja de MDF.
- Espacio necesario para evitar obstrucción con el movimiento rotatorio del motor.

En la figura Fig.43 se muestra la vista explosionada del mecanismo completo, en la parte inferior izquierda figura Fig.44 se muestran 3 piezas diferentes, la primera a la izquierda tiene como objetivo sostener el perfil en V, la segunda igualmente, a la vez que se une con la tercera pieza para sostener la polea de sincronización.

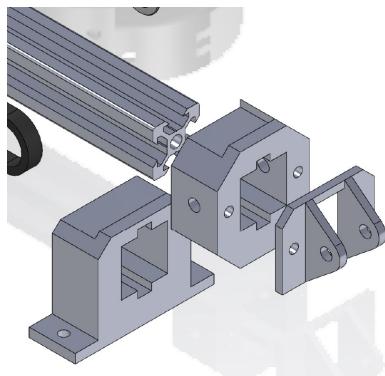


Fig. 45. Vista explosionada de soportes impresos en 3D

Para su diseño fueron considerados los siguientes aspectos:

- Las tolerancias para impresión 3D descritas en este documento.
- Posicionamiento de agujeros para fijación en la caja de MDF.
- Espacio necesario para evitar obstrucción con el movimiento de la banda.

- Espacio destinado para los Limit Switch, que debe estar situado en la parte superior de la pieza de soporte situada en medio.

D. Diseño de Caja de MDF

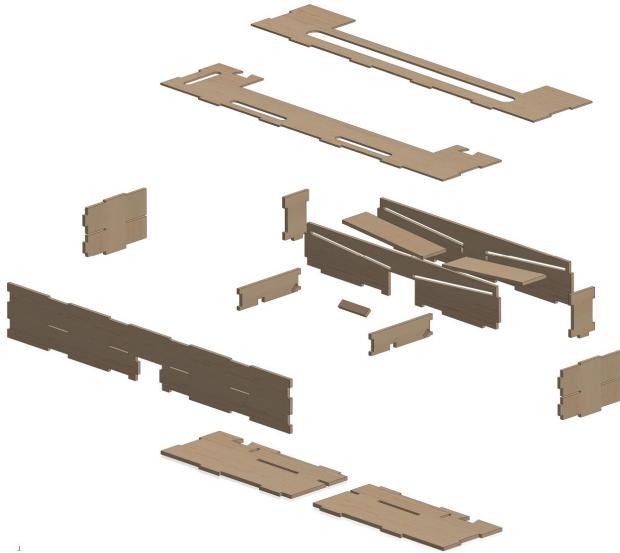


Fig. 46. Vista explosionada de caja MDF

Para la realización de la “caja” de MDF, explosionada en la Fig.46 anterior, fue necesario realizar primeramente el diseño de las piezas que iban a componerlas, esto fue dibujado y posteriormente prototipado en Solid Works, se tomaron en cuenta las siguientes consideraciones:

- Espacio en compartimentos para la electrónica y perfil en V.
- Ensamble de las piezas a través de alas de la suficiente longitud.
- Orificios destinados al cableado.
- Orificios del suficiente grosor para permitir la salida y movimiento del jugador.
- Tolerancias de 0.01 mm en las uniones de piezas, esto con el objetivo de que al momento de pegarlas, se evitaran retrabajos.
- Uso de material suficientemente fuerte, como es el MDF de 6 mm, para soportar el peso del sistema.

Se utilizó el CAD proporcionado por los maestros para el prototipado del ensamblaje Fig.47, en donde se muestra el modelo de la caja en MDF con las dimensiones apropiadas para encajar en la mesa de juego, además de esto, se fue a medir la mesa de juego antes y después del corte en láser para no errar en las medidas.

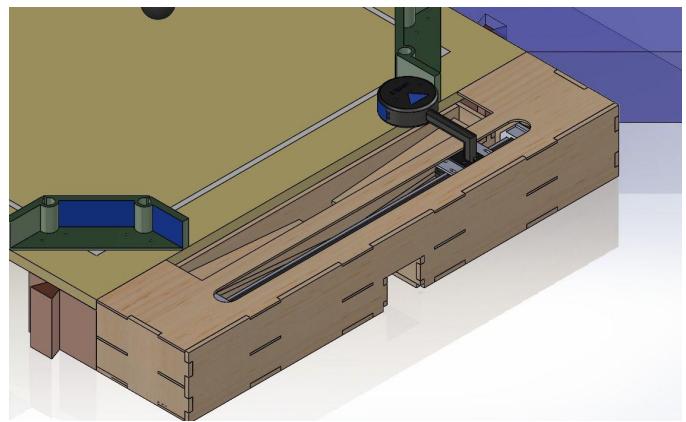


Fig. 47. Diseño de caja MDF modelado

E. Corte del diseño

Para el corte en láser optamos por ir a Innovaction Center el Tecnológico de Monterrey, en dónde seleccionamos cortamos las láminas de MDF de 6 mm para que tuvieran un tamaño más chico y cupieran en la cortadora, para los primeros cortes la máquina tardó un total aproximado de 2-4 minutos, para los siguientes cortes ocurrió un problema con el extractor y tuvimos que realizarlos en CIMA, en donde los siguientes cortes tuvieron una duración estimada de 5-10 minutos en realizarse en su totalidad. El resultado fue la siguiente caja, probada en la mesa de juego.

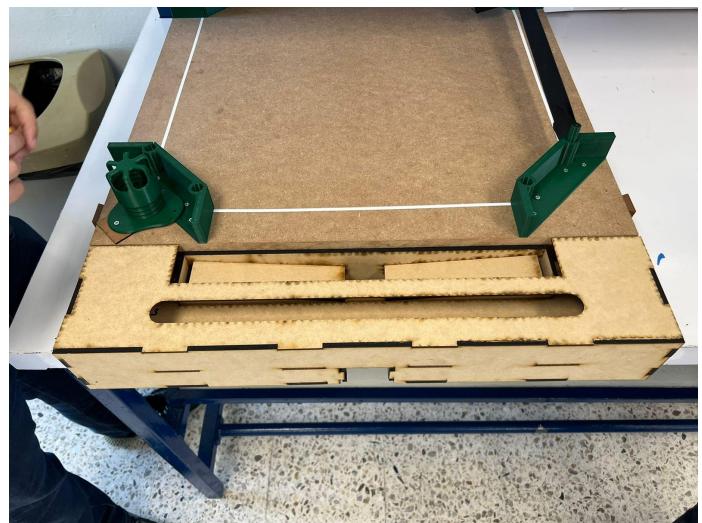


Fig. 48. Diseño de caja MDF colocado en mesa

F. Modelo Final

Una vez integrada la caja, la electrónica, y el diseño interno, se obtuvo el siguiente modelo final:



Fig. 49. Vista superior del modelo final



Fig. 50. Vista frontal del modelo final



Fig. 51. Vista lateral del modelo final



Fig. 52. Vista trasera del modelo final

G. Lista de Materiales

TABLA X. LISTA DE MATERIALES

Lista de materiales		
# Elemento (BOM)	Elemento	Descripción de uso
7	<u>Perfil V</u>	El perfil en V de 500 mm fue utilizado con la finalidad de movilizar el carrito a través del movimiento de la polea que estaba posicionada dentro de dicho perfil. Sus ranuras funcionaban a modo de riel para el deslizamiento de las ruedas en forma de V.
10	<u>Placa pótico</u> (carrito)	El carrito fue utilizado a modo de portero, sus ruedas en forma de V se movilizaban por la banda dentada a través de las ranuras del perfil en V. La base del carrito contaba con cortes en los cuáles fue posible atornillar el soporte o “palo” impreso en 3D del portero.
1	<u>Motor A pasos</u> (Nema 17)	El Nema 17 cumplió la función de realizar realizar un movimiento rotatorio a la punta del rotor de media luna, con este movimiento movilizaba a la polea de sincronización.
	Polea de sincronización	La polea se colocó en la punta del rotor y contaba con ranuras que movilizaban a la banda dentada, está a su vez movilizaba al carrito.
9	Correa Dentada	La correa dentada transformaba el movimiento rotatorio ejercido por el Nema 17 a la polea de sincronización en un movimiento tangencial, se colocó en las ranuras inferiores y superiores del perfil el V, en dónde se deslizaba y movía el carrito.
	Transistores	Los transistores funcionan para regular el voltaje proporcionado por la fuente, de esta manera nos fue posible alimentar elementos como el solenoide evitando accidentes.
	Modulo Joystick Arduino Dos Ejes (xy)	El joystick indicaba el movimiento del motor, se conectó a un driver que a su vez fue conectado al Nema.
1	Solenoide	El solenoide realizaba un “golpe” resultado de un impulso proporcionado por la fuente de poder, que interactuaba con su campo electromagnético, se posicionó detrás en la parte móvil del jugador, para proporcionarle el golpe, de igual manera, se conectó con de un transistor y un diodo que regularon el voltaje.
	Arduino UNO	El arduino es una placa de microcontrolador, con ella pudimos: 1. Comunicar por medio de programación al solenoide, joystick y Nema, para que se movieran en las direcciones indicadas y se

Lista de materiales		
# Elemento (BOM)	Elemento	Descripción de uso
		realizará el golpe. 2. Comunicar un infrarrojo con el display digital por medio de programación para indicar el número de goles.
	Driver De Motor A Pasos	El driver funciona para controlar motores de manera continua, se conectó a él Nema 17 para hacer rotar a la punta de media luna y cambiar su dirección de rotación.
14	Limit Switch	Los Limit Switch cumplieron la función de mandar una señal que se procesó por medio de programación para indicarle al Nema 17 cuando el carro topaba con una de las esquinas del mecanismo, de manera que no se siguiera movimiento en dicha dirección.
	Resistencias	Las resistencias controlaban el paso de corriente en nuestro circuito, de esta forma evitamos accidentes y la quema de nuestros elementos y dispositivos electrónicos.
	Protoboard	La protoboard nos permitió interconectar con componentes electrónicos, regular y proporcionar voltajes y comunicarnos por medio de programación. En la protoboard se conectaron todos los dispositivos electrónicos, en ella fue posible regular la corriente para el solenoide, Nema e interconectar el joystick, driver y motor a pasos.
	Tornillo Botón M.M INOX 5x12	Estos tornillos fueron los utilizados para unir la tapa del jugador con la parte inferior de este
	Tornillo Botón M.M INOX 4x16	Estos tornillos fueron los utilizados para unir el soporte del jugador o “palo” con el jugador impreso en 3D
	Tuerca para perfil 5mm	Estas tuercas fueron utilizadas con tornillos de 5mm para unir el soporte del jugador con el carro .
	Tuerca (Tuerca Hex. Inox 4mm)	Las tuercas fueron utilizadas para unir las monturas del Nema impresas en 3D
	Display	El display tiene como objetivo mostrar los goles anotados a nuestro equipo, funciona con la electrónica provista por el sensor infrarrojo.
	Sensor infrarrojo	El sensor infrarrojo tiene como objetivo detectar el movimiento en la parte inferior de la caja, donde se localiza el agujero destinado a guardar la pelota, de esta manera detecta cuando un gol es

Lista de materiales		
# Elemento (BOM)	Elemento	Descripción de uso
		metido a nuestro equipo.
	Botón	Al ser pulsado, activa al solenoide que produce el golpe a la pelota.
	Cables	Interconexión de elementos.

H. BOM de elementos y dibujo del ensamble

En la Fig. 53 se muestran varios de los elementos que componen al sistema modelado físicamente.

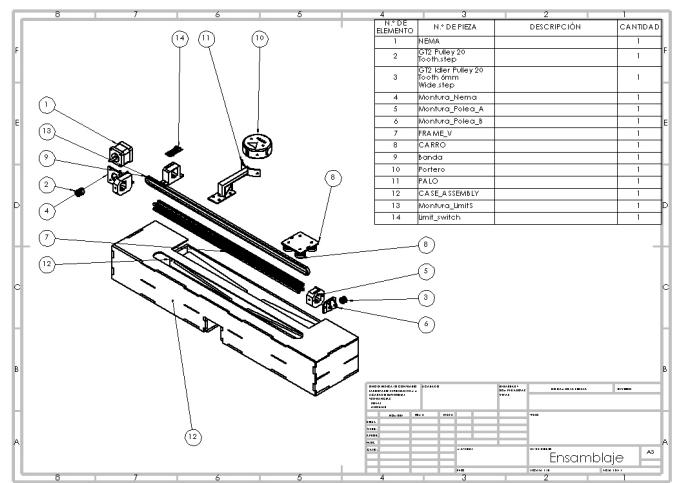


Fig. 53. Componentes del prototipo final

I. Análisis de Velocidad del Portero

Para el análisis del movimiento del portero analizamos la velocidad máxima de nuestro motor como 200 rpm, debido a documentación analizada con nuestro voltaje (12 V).

Debido a que tenemos dos poleas con igual número de dientes y dimensiones, sólo se tomarán en consideración los siguientes valores:

Radio de la polea (r): 8mm

RPM del motor: 200 rpm

Primeramente convertimos de RPM a rad/s

$$w_{polea} = 200 \text{ rpm}$$

$$w_{polea} = 200 \text{ rpm} * \frac{2\pi \text{ rad}}{\text{rev}} * \frac{\text{min}}{60 \text{ s}} = \frac{20}{3}\pi = 20.94 \text{ rad/s}$$

$$w_{polea} = 20.94 \text{ rad/s}$$

Para analizar la velocidad tangencial de la banda se realiza el producto cruz entre w_{polea} y r

$$v_{banda} = w_{polea} X r$$

$$v_{banda} = (20.94 \text{ rad/s})X(0.008m) = \frac{8}{150}\pi = 0.167 \text{ m/s}$$

La velocidad máxima de nuestro jugador puede alcanzar los 0.167m/s en su máximo valor.

V. CONCLUSIONES

A. Jesús Antonio López Malacón

Llevar a cabo la electrónica del prototipo fue mi aportación durante este reto, desde conseguir la fuente de voltaje y buscar suministrar nuestro solenoide de 24V y el stepper de 12V, al igual que suministrar poder al los dos arduinos que íbamos a utilizar. El realizar las conexiones por medio de cables duponts fue otra parte de mi trabajo en el reto, al igual que el código del solenoide junto al relay para mayor facilidad de manejo.

Los aprendizajes de eléctrica aprendidos en la realización de este fue algo que no contaba por lo que aprenderlo y realizarlo fue mi mayor reto en la materia, este fue superado con ayuda de los profesores e investigación en internet.

B. Fátima Pamela Ramón Quintana

La aportación que tuve en el equipo fue proponer el diseño elegido para llevarlo a físico, aunque en su mayoría todos eran muy similares, así como conseguir el lugar donde se iban a comprar varios de los elementos a implementar en el proyecto, también utilizar herramientas como SOLIDWORKS para proponer el portero, así como apoyar al equipo con las impresiones en 3D tanto del portero, control, etc, utilizando herramientas como Prusa Slicer para generar los códigos g, así como manejar la impresora Artillery X1, también pegamos la caja del prototipo, así como armar el portero y todo lo que iba montado en la caja, finalmente aporte en la realización en su mayoría de este reporte final.

Realmente los aprendizajes obtenidos durante esta materia fueron demasiado, tanto reforzar las habilidades de diseño en SOLIDWORKS, así como enseñarme generar los códigos necesarios para imprimir en 3D, también aprendí bastante de electrónica, así como de programación en arduino. Así como a reforzar algunos conceptos que había visto en materias anteriores.

El reto más difícil que enfrentamos fue que un día antes de entregar el proyecto uno de los códigos no corría adecuadamente, así como que nuestro driver se había descompuesto, por consiguiente se rompió nuestro portero y tuvimos que volverlo a imprimir y para nuestra suerte solo servicia una impresora del TEC, tuvimos varios contratiempos

pero al final todo salió mejor de lo que esperábamos. Otro reto grande fue la fuente de voltaje, ya que necesitábamos nuestra propia fuente, ya que nuestro solenoide necesitaba una alimentación de 24 v.

C. Samantha López Vizcarra

Mi aportación al equipo fué realizar, en su mayoría, el código del prototipo. El código para mover el motor junto con el joystick, así como las partes extras que agregamos relacionado al contador: display, buzzer y sensor infrarrojo. También realicé las conexiones de dichos componentes. Así mismo apoyé en la impresión 3D de las piezas y en conseguir lo que necesitamos para el proyecto.

Lo que aprendí durante este reto fué principalmente arduino, ya que era mi primera vez utilizándolo, así como aprendí a imprimir en 3D, manejar las impresoras y a utilizar un poco más SolidWorks.

Mi mayor reto fué conseguir que el stepper se moviera junto con el joystick rápido y fluido, ya que se movía a pasos o muy lento, lo que logré superar con la ayuda de los profesores e investigando en internet.

D. Juan Alberto Moreno Cantú

Mi aportación al equipo fue en su mayoría apoyar con el armado del prototipo, al igual que la investigación sobre los elementos que deberíamos usar y sus características, para así asegurarnos de no dañar ninguno de los componentes eléctricos, transversal a todo esto programe un código de visión artificial para poder controlar al stepper mediante una cámara web y que así este fuese automático, sin embargo por cuestión de tiempo no se pudo implementar, aun estando el código en un estado funcional.

Este bloque me llevo un gran número de aprendizajes ya que nunca había programado en arduino al igual de no estar muy familiarizado con las herramientas de diseño como lo es solidworks, ni de impresión como lo es prusa, y puedo decir con confianza que ya tengo un nivel de manejo intermedio a avanzado en estas habilidades.

E. Diego Flores Martínez

Mi aporte al equipo fue utilizar herramientas de diseño asistido por computadora para la realización de los prototipos digitales que hicimos. Modelé el mecanismo que realizamos para mover al portero, la caja de mdf donde lo colocamos, y al portero.

Esta clase me gustó mucho ya que aprendí conceptos mecánicos de los que no había escuchado. Reforcé mis conocimientos de electrónica y microcontroladores, además de que aprendí a como manipular el voltaje para aprovecharlo en futuros proyectos.

Lo más complicado de la materia y el proyecto fueron los conceptos eléctricos que vimos. Hacer la fuente de voltaje y aplicarla al proyecto fue complicado y la verdad es que tenía miedo de que fuera peligroso, pero al final resultó.

F. Dulce María Naranjo Sarmiento

Para este proyecto, considero que mi aporte individual fue realizar la propuesta del guardameta, al igual que parte del trabajo de físico, como es la impresión 3D, el cortado de madera a través de una cortadora láser y , parte de la implementación de los componentes físicos en un mismo modelo final.

Considero que la impresión 3D de los modelos fue una de las áreas en las que más trabajé y aprendí a lo largo de este proyecto, esto fue: el modelado de piezas en herramientas como Z-Suite, Prusa Slicer y Cura, así como el trabajado de estas piezas para su implementación en el modelo físico.

En este bloque aprendí de igual manera, conceptos sobre electrónica y diseño básico, tales son, el uso de herramientas como SolidWorks para el modelo de piezas más complejas de las que había modelado hasta el momento, como ensambles que hacen uso de tuercas, y la implementación de variables globales. Cabe mencionar que igualmente aprendí conceptos de electrónica básica, como el uso de fuentes de voltaje y la correcta elección de los componentes electrónicos.

Mi mayor reto en este bloque fue la comprensión de sistemas físicos y su modelado, debido a que considero entender las partes que integran el proyecto realizado, sin embargo fue difícil integrarlas como parte del proyecto final.

REFERENCIAS

- [1] “Guía de modelos Arduino y sus características | Arduino UNO | BricoGeek Lab,” Modelos de Arduino, 2023. <https://lab.bricogeek.com/tutoria/guia-de-modelos-arduino-y-sus-caracteristicas/arduino-uno> (Jun. 18, 2023).
- [2] “Nema_17_17HS4401 Motor a Pasos 1.7Amp 4kg/cm - UNIT Electronics,” UNIT Electronics, Jun. 15, 2023. <https://uelectronics.com/producto/nema-17-17hs4401-motor-a-pasos-1-7amp-4kg-cm/> (Jun. 18, 2023).
- [3] L. Llamas, “Motores paso a paso con Arduino y driver A4988 o DRV8825,” Luis Llamas, Aug. 23, 2016. <https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/> (Jun. 18, 2023).
- [4] L. Llamas, “Controla tus proyectos con Arduino y Joystick analógico,” Luis Llamas, Jul. 08, 2016. <https://www.luisllamas.es/arduino-joystick/> (Jun. 18, 2023).