

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**CONSTRUCCIÓN DE 5 MÓDULOS DIDÁCTICOS PARA
CONECTAR Y CONTROLAR LAS ENTRADAS Y SALIDAS DE
ARDUINO DESDE MATLAB**

CONSTRUCCIÓN DE LOS MÓDULOS DIDÁCTICOS

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN ELECTROMECAÁNICA**

CHRISTOPHER MARIO GANSINO LLANOS

christopher.gansino@epn.edu.ec

DIRECTOR: CATALINA ELIZABETH ARMAS FREIRE

elizabeth.armas@epn.edu.ec

DMQ, agosto 2023

CERTIFICACIONES

Yo, CHRISTOPHER MARIO GANSINO LLANOS declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

CHRISTOPHER MARIO GANSINO LLANOS

christopher.gansino@epn.edu.ec

mario-gansino1996@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por CHRISTOPHER MARIO GANSINO LLANOS, bajo mi supervisión.

CATALINA ELIZABETH ARMAS FREIRE

DIRECTOR

elizabeth.armas@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como los productos resultantes del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CHRISTOPHER MARIO GANSINO LLANOS

DEDICATORIA

Dedicó la tesis a mi padre Mario Gansino que me ha guiado por el camino del bien para ser un mejor ser humano y lograr seguir en mis estudios, a mi madre Enriqueta Llanos que me ayudo en los momentos complicados de la vida y al igual que mi padre me ayudo a culminar seguir con mis estudios. A mis cuatro hermanas: Allison Gansino, Katherine Gansino, Micaela Gansino y Samanta Gansino, que a pesar de los conflictos siempre me han apoyado, a mi sobrio que me da felicidad el saber que tiene una gran persona como madre.

Mario Gansino

AGRADECIMIENTO

Agradezco a mis padres por ayudarme a culminar mi formación académica, de igual forma agradezco a toda persona dentro y fuera de la universidad que me ha apoyado en la culminación de este este proyecto.

Mario Gansino

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 OBJETIVO GENERAL.....	2
1.2 OBJETIVOS ESPECÍFICOS.....	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO	3
Conexión de Arduino y Matlab.....	3
Protocolo de comunicación I2C.....	4
2 METODOLOGÍA.....	5
2.1 CONSTRUCCIÓN DE MÓDULOS	5
Adquisición de elementos.....	5
Construcción de la placa PCB.....	13
Construcción de base de los módulos.....	14
2.2 HOJAS GUÍAS DE LABORATORIO	16
3 RESULTADOS	17
3.1 ALIMENTACIÓN DEL MÓDULO.....	17
Fuente de Arduino.....	17
Fuente externa.....	17
3.2 COMUNICACIÓN ENTRE EL COMPUTADOR Y EL MÓDULO	18
3.3 CONEXIONADO DE PERIFÉRICOS.	19
Periféricos de entrada.....	19
Periféricos de salida	19
3.4 CONTROL DE PERIFÉRICOS DESDE <i>COMMAND WINDOW</i> DE <i>MATLAB</i>	21
Lectura de voltaje del sensor de temperatura LM35.....	21
Lectura de voltaje del sensor fotorresistor.....	21
Lectura de voltaje del potenciómetro.....	22
Lectura de pines del teclado matricial	22
Escritura para el encendido del LED.....	23
Escritura para el encendido del módulo relé, foco y ventilador	25

Escritura para la posición del servomotor	26
3.5 CONTROL DE PERIFÉRICOS DESDE LA INTERFAZ DE MATLAB	28
Control del LED.....	28
Control de la bocina	28
Escrituras de ondas PWM en el LED.....	28
Posicionamiento del servomotor	29
Lectura de voltaje del sensor fotorresistor	29
Lectura de temperatura del sensor LM35.....	30
3.6 CONTROL DE PERIFÉRICOS DESDE ARDUINO	31
Lectura de temperatura del sensor LM35 y accionamiento del módulo relé.....	31
Posición del servomotor.....	32
Encendido del LED con el potenciómetro	32
Ingreso de valores alfanuméricos a través del teclado matricial.....	32
Lectura del sensor fotorresistor para dar tonos en la bocina	33
4 CONCLUSIONES	34
5 RECOMENDACIONES.....	35
6 REFERENCIAS BIBLIOGRÁFICAS	36
7 ANEXOS.....	37
ANEXO I. REPORTE DE SIMILITUD GENERADO POR TURNITIN.....	1
ANEXO II. CERTIFICADO DE FUNCIONAMIENTO DEL TRABAJO DE INTEGRACIÓN CURRICULAR	2
ANEXO III. DIMENSIONES DE CORTE PARA LA PLACA ELECTRÓNICA.	3
ANEXO IV. CIRCUITO IMPRESO.....	1
ANEXO V. DISTRIBUCIÓN DE PERIFÉRICOS.....	1
ANEXO VI. DIMENSIONES DE CORTE PARA LA BASE DEL MÓDULO.....	1
ANEXO VII. DIAGRAMA DE CONEXIÓN DEL MÓDULO.....	1
ANEXO VIII. INTERFAZ APP DESIGNER.	1

RESUMEN

Este proyecto tiene como objetivo desarrollar 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab, para que los estudiantes aprendan de la importancia de vincular el lenguaje de programación, la electrónica digital y los microprocesadores [1]. Los estudiantes identificarán cada elemento de entrada y salida (digitales y analógicas) como: sensor de temperatura (LM35), potenciómetros, motores, diodo emisor de luz (LED), sirenas, módulo Arduino, fotorresistor, entre otros.

En el componente de diseño de los módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab, se muestran los planos para la construcción de los módulos, donde se identifican las dimensiones, la disposición de espacio en el que se ubican todos los dispositivos del módulo. Además, el diseño del circuito impreso (PCB) [2].

Este proyecto se enfoca en la construcción del módulo didáctico a partir de los planos elaborados en el componente anteriormente mencionado. En la etapa de construcción se especifica el material y se elabora la placa PCB que facilita la distribución de la conexión de las fuentes de alimentación y de cada dispositivo de entrada y salida, así como de la placa Arduino.

Los módulos didácticos funcionan de dos formas: conexión de Matlab con Arduino y únicamente el módulo Arduino. En la primera forma de trabajo, Arduino se conecta a través del bus universal en serie (USB), lo que permite la transferencia de datos desde Matlab hacia los pines de la placa Arduino [3]. En la segunda forma de trabajo, se carga el código de programación en el microcontrolador de Arduino, lo que permite controlar las entradas y salidas sin la necesidad de conexión a Matlab.

Se crearon hojas guías que proporcionan instrucciones detalladas sobre cómo utilizar los módulos y realizar experimentos en el laboratorio. Estas hojas guías permiten familiarizar y desarrollar habilidades en los estudiantes acerca de Arduino y Matlab.

PALABRAS CLAVE: Arduino, Matlab, LM35, fotorresistor, LED.

ABSTRACT

This project aims to develop 5 didactic modules to connect and control Arduino inputs and outputs from Matlab, so that students learn the importance of linking the programming language, digital electronics and microprocessors [1]. Students will identify each input and output element (digital and analog) as: temperature sensor (LM35), potentiometers, motors, light emitting diode (LED), sirens, Arduino module, photoresistor, among others.

In the design component of the didactic modules to connect and control the inputs and outputs of Arduino from Matlab, the plans for the construction of the modules are shown, where the dimensions are identified, the layout of the space in which all the module devices. In addition, the design of the printed circuit (PCB) [2].

This project focuses on the construction of the didactic module from the plans elaborated in the aforementioned component. In the construction stage, the material is specified and the PCB board is made, which facilitates the distribution of the connection of the power supplies and each input and output device, as well as the Arduino board.

The didactic modules work in two ways: connection of Matlab with Arduino and only the Arduino module. In the first way of working, Arduino is connected via the Universal Serial Bus (USB), which allows data transfer from Matlab to the Arduino board gantries [3]. In the second way of working, the programming code is loaded into the Arduino microcontroller, which allows controlling the inputs and outputs without the need to connect to Matlab.

Guide sheets were created that provide clear and detailed instructions on how to use the modules and perform experiments in the laboratory. These guide sheets allow students to familiarize and develop skills about Arduino and Matlab.

KEYWORDS: Arduino, Matlab, LM35, photoresistor, LED.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El proyecto se plantea con el objetivo de construir 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab, enlazando el lenguaje de programación, la electrónica digital y microprocesadores. Para seleccionar los periféricos de entrada y salida, se consideran las características y especificaciones técnicas y físicas, siendo óptimos los siguientes dispositivos:

Periféricos de entrada:

- Sensor de temperatura LM35.
- Sensor fototransistor.
- Potenciómetro.
- Teclado matricial.

Periféricos de salida:

- LED.
- Relés.
- Servomotor.
- Bocina.
- LCD.

Cargas de los relés:

- Bombilla incandescente.
- Ventilador.

Se elaboró una placa PCB para facilitar la conexión de los pines de cada dispositivo, para ello se soldó cada pin de cada dispositivo en la placa, considerando la distribución del componente de diseño.

Los módulos didácticos funcionan de dos formas: conexión de Matlab con Arduino y únicamente el módulo Arduino. En la primera forma de trabajo, Arduino se conecta a través del bus universal en serie (USB), lo que permite el accionar de los pódicos de la placa Arduino Uno desde Matlab. En la segunda forma de trabajo, se carga el código de

programación en el microcontrolador de Arduino, lo que permite controlar los pódicos de entrada y salida sin la necesidad de conexión a Matlab.

El módulo didáctico dispone de dos alimentaciones con tierras aisladas. La primera alimentación es proporcionada por el Arduino con voltajes de 5 (V_{DC}) que son entregados desde el computador, y la segunda alimentación es proporcionada por un cargador externo con voltajes de 5 (V_{DC}), cada fuente es distribuida a distintos actuadores donde satisfacen el amperaje requerido para el funcionamiento.

Se creó material instructivo en forma de hojas guías que proporcionan instrucciones claras y detalladas sobre cómo utilizar los módulos y realizar prácticas en el laboratorio. Estas hojas guías permiten familiarizar y desarrollar habilidades en los estudiantes acerca de Arduino y Matlab.

1.1 Objetivo general

Construir 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab.

1.2 Objetivos específicos

1. Construir los módulos para conectar y controlar las entradas y salidas de Arduino.
2. Elaborar hojas guías de laboratorio para familiarizar al estudiante con los módulos.

1.3 Alcance

El proyecto se enfoca en la construcción de 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab. El objetivo es diseñar y programar un sistema capaz de adquirir señales desde dispositivos de entrada (sensor de temperatura LM35, pulsadores, potenciómetro, sensor fototransistor y teclado matricial) para procesarlas y enviar señales a los dispositivos de salida (pantalla de cristal líquido LCD; un servomotor; bocina; LED y dos relés que encienden una bombilla incandescente y un ventilador, respectivamente.)

Para enlazar Arduino con Matlab se utilizan los paquetes disponibles (arduinoio.mlpkinstall). La interfaz gráfica de Matlab que permite controlar los dispositivos

físicos y se usa la opción *App Designer*. Se utilizan códigos de programación del componente de Diseño de módulos, tanto en el programa Arduino como en Matlab para realizar las siguientes acciones: receptar las señales de un pulsador, potenciómetro y teclado, realizar mediciones de temperatura y cantidad de iluminación, controlar el encendido y apagado de un LED, posicionar un servomotor, variar el nivel de frecuencia de una bocina, accionar los relés; y, enviar datos por el puerto serial I2C para mostrar información en un LCD.

Se construyó la PCB que se elaboró en Proteus, que permite la conexión de los periféricos con la placa del Arduino Uno.

Los módulos se construyen basándose en el diseño en dos y tres dimensiones creado en el software Inventor, para distribuir el espacio de los periféricos, la placa Arduino Uno, protecciones eléctricas y el cableado.

Se elaboran dos hojas guías de laboratorio que permiten al estudiante familiarizarse con la conexión y manejo de cada módulo.

1.4 Marco teórico

Conexión de Arduino y Matlab

Matlab es un entorno desarrollador de computación que realiza procesos de programación mediante aplicaciones remotas como lo es *App Designer* y *Command Window*, controlando las entradas y salidas (digitales y analógicas) de la placa Arduino a través de la conexión USB del computador [4].

Arduino es un entorno de diseño de programación que facilita la interacción entre el código de programación cargado desde el computador en el microcontrolador para el control de periféricos, accionando las entradas y salidas (digitales y analógicas) de Arduino.

Mediante el soporte de Matlab para Arduino (`arduinoio.mlpkginstall`) [5], es posible comunicarse de manera interactiva con el módulo de arduino de la Figura 1.1. De tal manera que permite realizar acciones como:

- Recibir datos de sensores conectados a los puertos analógicos.
- Controlar los puertos digitales como salidas y/o entradas; además de los pines PWM para el uso del servomotor.

- Conexión mediante la comunicación de circuito inter- integrado (I2C) para el uso del LCD.

El paquete de soporte es compatible con versiones de Matlab a partir de la versión R2014a en adelante [5].

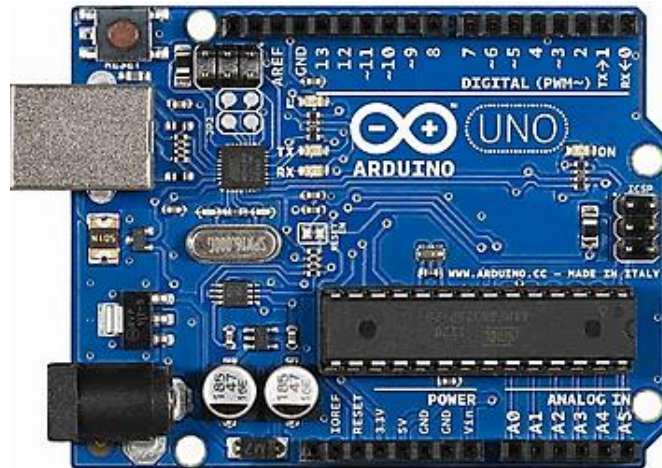


Figura 1.1. Arduino Uno

Protocolo de comunicación I2C.

Emplea dos líneas de comunicación, una línea para enviar datos y otra línea para recibir datos. Un pin de reloj en serie (SCL) de la placa Arduino que da una frecuencia de intervalos regulares y un pin de datos en serie (SDA) que transfiere datos como se muestra en la Figura 1.2. Para este proyecto se usa este protocolo solamente para la conexión de la pantalla de cristal líquido (LCD) que dispone de dirección única teniendo como conexión solamente dos pines de la placa Arduino Uno [6].

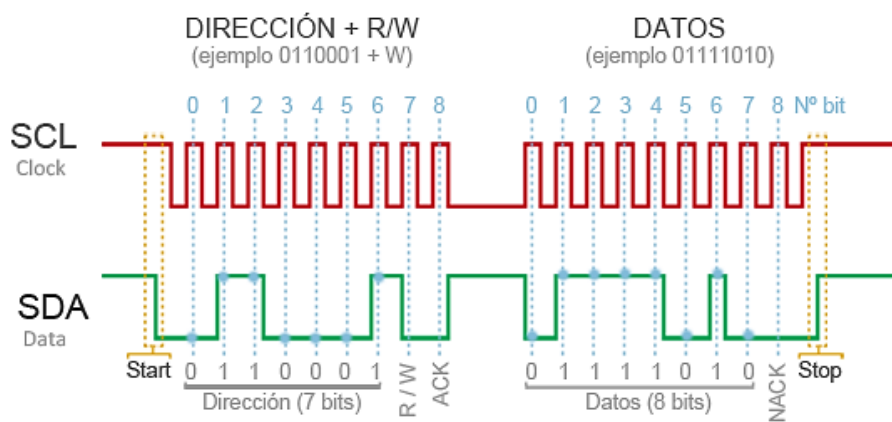


Figure 1.2. Transferencia de datos del protocolo I2C

2 METODOLOGÍA

El desarrollo del componente de construcción de los módulos didácticos se basa en la investigación aplicada, considerando las enseñanzas de las materias de Electrónica Digital y Microprocesadores (TEMD333) y Electricidad (TEMD245) de la carrera de Tecnología Superior en Electromecánica.

En la Figura 2.1 se indica la metodología aplicada para el desarrollo del componente de construcción del módulo didáctico.

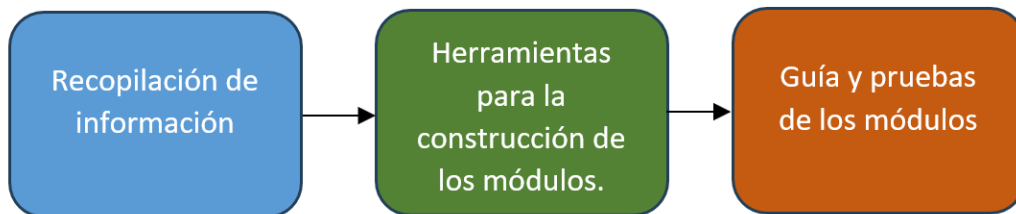


Figura 2.1. Esquema de metodología

2.1 Construcción de módulos

La etapa de implementación inicia con la adquisición de elementos, periféricos y materiales especificados; luego se construyeron las placas PCB; después se construyeron los módulos de modo que sean modulares, robustos y de fácil conexión; y, finalmente se realizan las siguientes pruebas de funcionamiento:

- Alimentación del módulo.
- Comunicación entre el computador y el módulo.
- Conexión de periféricos.
- Control de periféricos desde *Command Window* de Matlab.
- Control de periféricos desde la interfaz de Matlab.
- Control de periféricos desde Arduino.

Adquisición de elementos

- **Arduino Uno**

Para la adquisición de los dispositivos, se tomó como base las características físicas y electrónicas de los pines que se muestran en la Figura 2.2 de la placa del Arduino Uno, siendo las siguientes [7]:

- Dispone de catorce pines digitales (D0 - D13) en configuración *pull-up*, que dependiendo de la pre configuración se pueden usar como entradas o salidas, cada pin trabaja a 5 (V_{DC}) con una corriente de suministro máxima de 50 (mA).
- La placa Arduino tiene seis pines de entrada analógica (A0 - A5) que a su vez tiene una conexión interna hacia un conversor analógico digital, cada pin admite un voltaje de 5 (V_{DC}).
- Posee dos pines de comunicación serial denominados RX y TX (D0, D1 respectivamente).
- Dispone de dos pines (D2, D3) que permiten generar interrupciones y pueden activarse dependiendo de la configuración que se preseleccione.
- Dispone de seis salidas de modulación de ancho de pulso (PWM).
- Dispone de dos pines para a comunicación de datos I2C, con sus nombres SCL y SDA.

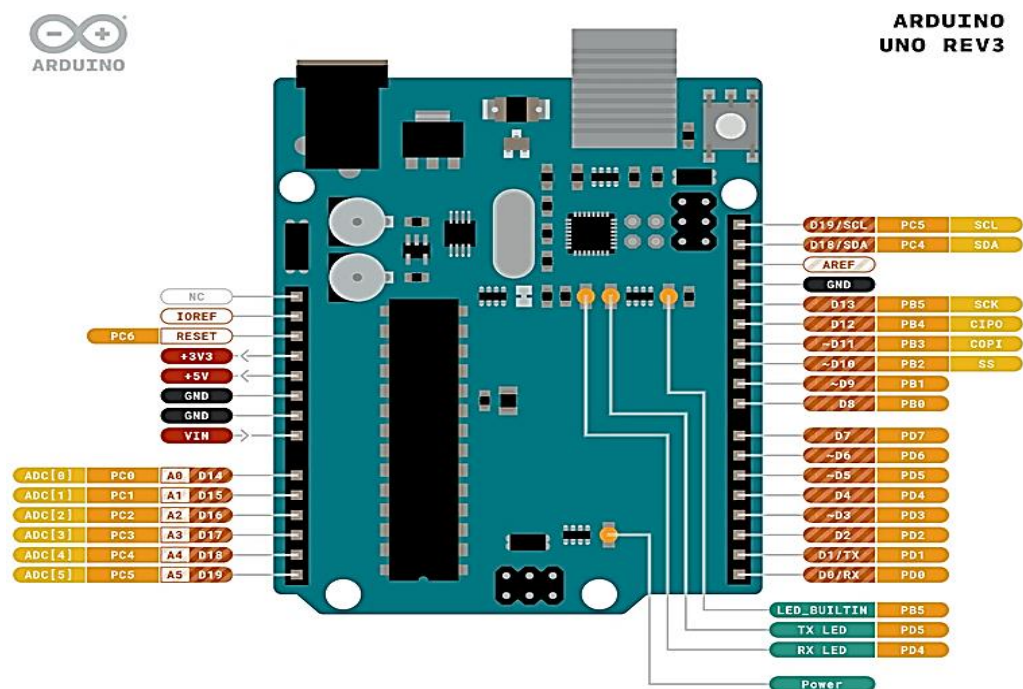


Figura 2. 2. Pines del Arduino Uno

- Fuentes de alimentación

Los voltajes que se suministraron en el módulo y el nombre usado se indican en la Tabla 2.1.

Tabla 2.1. Fuentes del módulo

Fuente de alimentación	Voltaje (V_{DC})	Nombre
Arduino Uno	5	Vcc
Fuente externa	5	Vce

- **Periféricos de entrada**

En la Tabla 2.2 se indican los pines de cada periférico de entrada.

Tabla 2.2. Pines de periféricos de entrada

Dispositivo	Pines
Sensor de temperatura LM35	Vcc, Control, Gnd
Módulo fotorresistor	Control, Gnd, Vcc.
Potenciómetro	Vce, Salida, Gnd.
Teclado matricial	Pines de salida (L1, L2, L3, L4, C1, C2, C3, C4).
Integrado 74c922 para conectar el teclado matricial	Pines de entrada del teclado matricial (L1, L2, L3, L4, C1, C2, C3, C4), Vce, Gnd, Salida A, Salida B, Salida C, Salida D, Datos A.

Se indican los pines del sensor LM35 en la Figura 2.3.

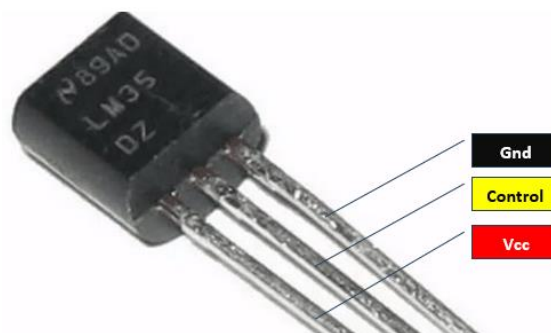


Figura 2.3. Pines del sensor LM35

Se consideran los pines de la Figura 2.4 del módulo fotorresistor.



Figura 2.4. Pines del módulo fotorresistor

En la Figura 2.5 se muestran los conectores del potenciómetro.

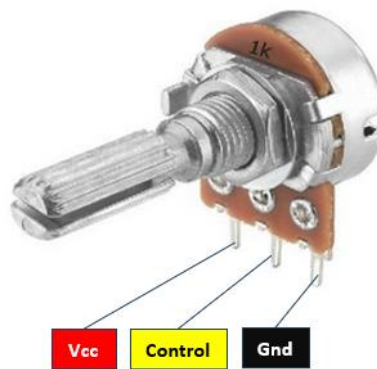


Figura 2.5. Pines del potenciómetro

El teclado matricial dispone de ocho pines que se indican en la Figura 2.6.



Figura 2.6. Pines del teclado matricial

El integrado 74c922 se indica en la Figura 2.7.

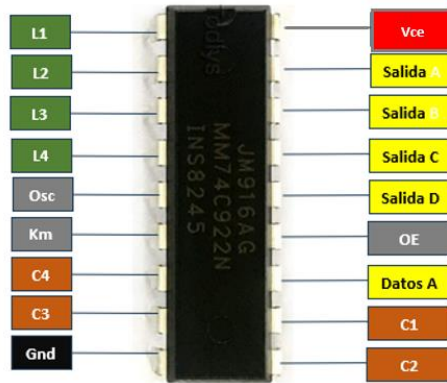


Figura 2.7. Pines del integrado 74c922

- **Periféricos de Salida**

Los pines de los periféricos de salida se visualizan en la Tabla 2.3.

Tabla 2.3. Pines de periféricos de salida

Dispositivo	Pines
LED	Control, Gnd.
Módulo relé	Vcc, Entrada 1 (IN1), Entrada 2 (IN2), Gnd, Relé 1 (Ventilador, Vce), Relé 2 (Foco, Vce).
Servomotor	Vcc, Control, Gnd.
Bocina	Control, Gnd.
LCD	Gnd, Vcc, 16 pines de conexión al módulo I2C (Vcc, Rs, R/W, EN, DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7, LED Vcc, LED Gnd).
I2C LCD	16 pines de conexión al LCD (Vcc, Rs, R/W, EN, DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7, LED Vcc, LED Gnd), Dos pines de conexión de encendido de luz LED, Gnd, Vcc o Vcc, SDA, SCL.

El LED dispone de los siguientes pines de la Figura 2.8, para este proyecto se implementó un LED de color rojo y un LED de color amarillo.

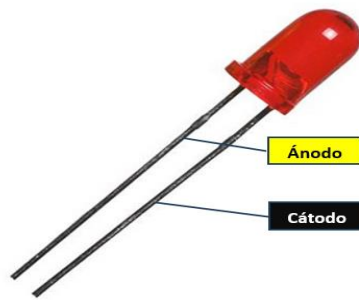


Figura 2.8. Pines de LED

Para el cálculo de la resistencia se implementó la Ecuación 2.1 de la ley de Ohm [8].

$$V = R * I$$

Ecuación 2.1. Ley de Ohm

Donde:

V: Voltaje (V_{DC}).

R: Resistencia (Ohm).

I: Corriente (A).

También se consideraron los valores de voltaje de operación que se muestra en la Tabla 2.3 y la corriente máxima de 50 (mA) que suministra los pines del Arduino uno.

Tabla 2.3. Voltaje de operación del LED

Color LED	Voltaje de operación (V_{DC})
Rojo	1,63 – 2,03
Amarillo	2,10 – 2,18

Dando como resultado:

Para el LED Rojo.

$$R = \frac{V}{I} = \frac{2,03}{0,05} = 40,6 \text{ (Ohm)}$$

Para el LED amarillo.

$$R = \frac{V}{I} = \frac{2,18}{0,05} = 43,6 \text{ (Ohm)}$$

Los pines del módulo relé se visualizan en la Figura 2.9.

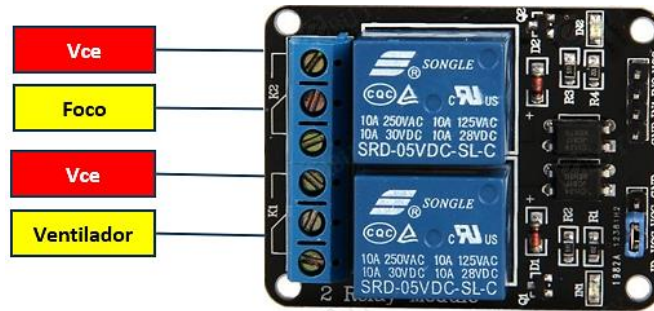


Figura 2.9. Módulo relé

En la Figura 2.10 se muestra el ventilador y el foco que se conectó al módulo relé.

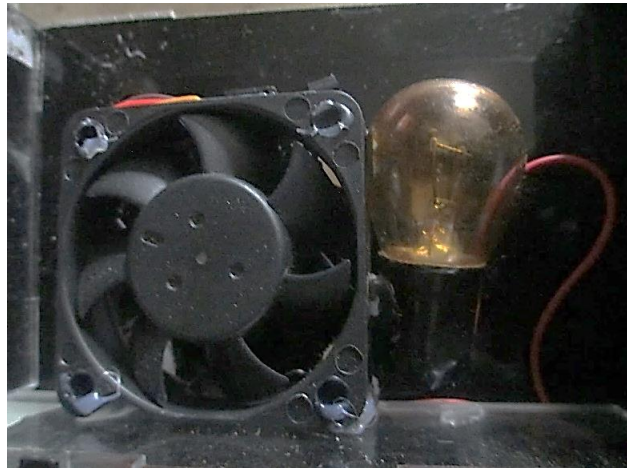


Figura 2.10. Ventilador y foco

El servomotor tiene tres pines como se ve en la Figura 2.11



Figura 2.11. Pines del servomotor

En la Figura 2.12 se muestra la bocina.



Figura 2.12. Pines de la bocina

Los pines del módulo I2C se indican en la Figura 2.13.

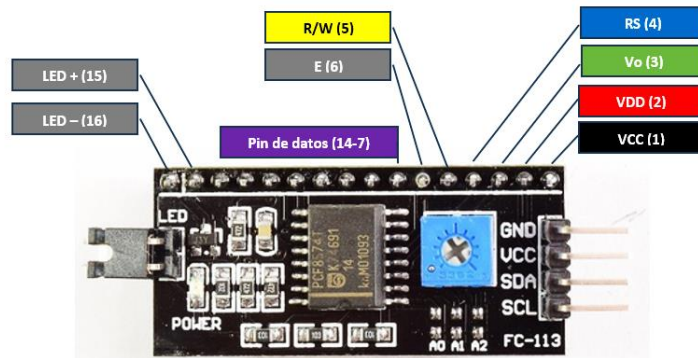


Figura 2.13. Módulo I2C

Los pines del LCD se indican en la Figura 2.14.

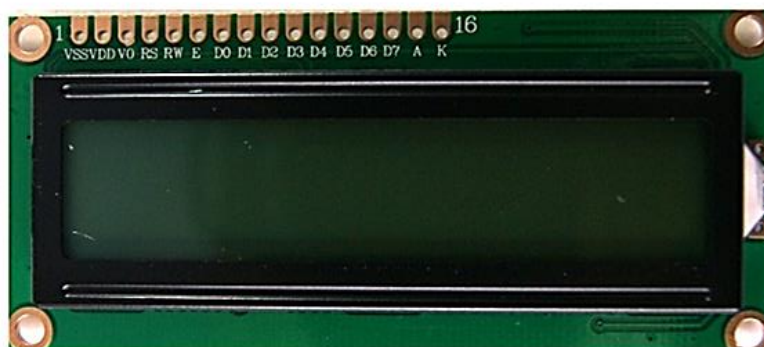


Figura 2.14. Distribución de pines LCD.

En la Figura 2.15 se indica la conexión del LCD y el módulo I2C.



Figura 2.15. Conexión LCD y Módulo I2C

Construcción de la placa PCB

Para la construcción de la placa PCB, se requiere del esquema del componente de diseño de los módulos. Por lo que, se consideraron las dimensiones que se muestra en el Anexo III.

Se utilizó el método de planchado para obtener las pistas en la baquelita, en el Anexo IV se muestra el diseño de circuito impreso.

Una vez elaborada la placa se colocó pintura negra como protección y para el acabado.

Finalmente, se soldaron los dispositivos necesarios de acuerdo con la distribución de periféricos que se muestra en el Anexo V. La placa final se indica en la Figura 2.20.

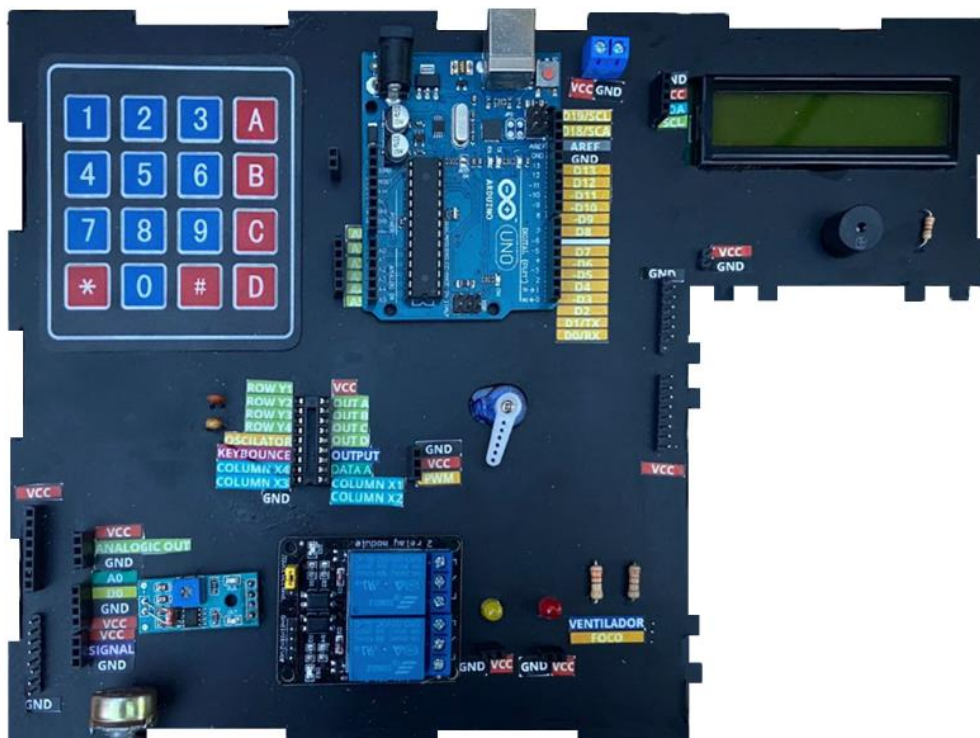


Figura 2.20. Placa del módulo terminada.

Construcción de base de los módulos

Para la construcción de la base de los módulos, se seleccionó el material acrílico debido a que posee importantes características como son:

- Posee una dureza similar a la de algunos metales como lo es el cobre y latón.
- Al ser una aleación de plástico, es resistente al paso de corriente eléctrica, evitando así poner en riesgo el daño de los dispositivos electrónicos.
- Soporta altas temperaturas y es un buen aislante de temperatura.
- Es resistente a la intemperie y manipulación.

Se utilizó el esquema del componente de diseño de los módulos, para realizar los cortes necesarios en la plancha de acrílico, en el Anexo VI se muestra las dimensiones de corte para cada parte de la base, dando como resultado el modelado de la Figura 2.21.

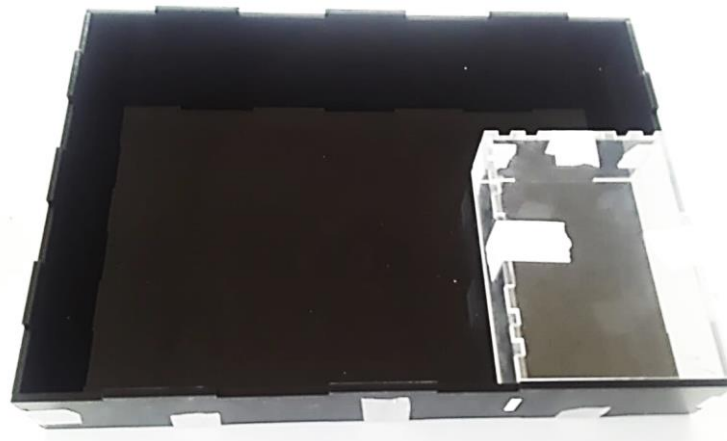


Figura 2.21. Base de los módulos.

Para finalizar la etapa de construcción, se ubicó el ventilador y el foco en la base transparente del módulo de la Figura 2.10, teniendo en cuenta su distribución de pines que se conectaron a la placa PCB. El módulo terminado se obtiene al juntar la placa PCB que de la Figura 2.20 y la base de acrílico de la Figura 2.21 obteniendo el resultado de la Figura 2.22.



Figura 2.22. Módulo didáctico.

En la Tabla 2.4 se muestran las partes del módulo.

Tabla 2.4. Partes del módulo

No.	Descripción	No.	Descripción
1	Placa Arduino Uno	10	LCD con pines de conexión
2	Teclado matricial	11	Pines de alimentación de fuente externa
3	Integrado 74c922	12	Servomotor con pines de conexión
4	Módulo relé	13	Focos LED con pines de conexión
5	Módulo fotorresistor con pines de conexión	14	Potenciómetro con pines de conexión
6	Sensor LM35	15	Pines de alimentación de Arduino
7	Ventilador	16	Pines de conexión del LM35
8	Foco	17	Pines de alimentación de Arduino
9	Bocina con pines de conexión	18	Pines de salida del integrado 74c922

2.2 Hojas guías de laboratorio

Para indicar la forma correcta del uso de los módulos. Se elaboraron hojas guías, en la Figura 2.22 se muestra un código QR que permite acceder al contenido de las hojas guías.



Figura 2.22. Hojas guías

Enlace: <https://1drv.ms/f/s!Auh36VlvAW8DrUCeQN6ldG2GTJhV?e=nYsnff>.

Para visualizar el video del funcionamiento del módulo, se accede mediante el código QR de la Figura 2.23 o mediante el enlace de acceso.



Figura 2.23. Uso del módulo

Enlace: <https://1drv.ms/f/s!Auh36VlvAW8DrT-DSEL0ZeS2tz7p?e=cAz4VI>.

3 RESULTADOS

3.1 Alimentación del módulo

Fuente de Arduino

El módulo dispone de dos alimentaciones, la primera fuente corresponde a Vcc. Se recopilaron datos de mediciones de los voltajes en los terminales de alimentación de Arduino, obteniendo la Tabla 3.1.

Tabla 3.1. Voltaje del Arduino

Fuente	Voltaje ideal (V_{DC})	Voltaje medido (V_{DC})	Error %	Resultado
Vcc	5	5,02	0,4	✓
		5,01	0,2	✓
		4,99	0,2	✓
		5,00	0	✓
		5,01	0,2	✓

Los valores de la tabla anterior comprueban que el voltaje permanece con valores próximos a 5 (V_{DC}), dado que se encuentra dentro del rango de funcionamiento del voltaje requerido.

Fuente externa

La segunda fuente de alimentación corresponde a Vce y es suministrada por un cargador externo que proporciona un voltaje de 5 (V_{DC}). La Tabla 3.2 muestra los valores de los voltajes medidos.

Tabla 3.2. Voltaje de la fuente de alimentación

Fuente	Voltaje ideal (V_{DC})	Voltaje medido (V_{DC})	Error %	Resultado
Vce	5	4,99	0,2	✓
		4,99	0,2	✓
		5,00	0	✓
		5,01	0,2	✓
		5,00	0	✓

Los valores de voltaje de la tabla anterior, son óptimos dado que están dentro del voltaje ideal proporcionado por la fuente.

3.2 Comunicación entre el computador y el módulo

Para comprobar la comunicación del módulo Arduino y el computador, se utilizó la pantalla de *Command Window* de Matlab. En la Figura 3.1, al conectar la placa Arduino al puerto USB del computador y asignar una variable, inmediatamente se muestran las propiedades de la placa conectada.

```
Arduino Uno detected.  
This device is ready for use with MATLAB Support Package for Arduino Hardware. Get started with examples and other documentation.  
To use this device with Simulink, install Simulink Support Package for Arduino Hardware.  
  
a =  
  
arduino with properties:  
  
    Port: 'COM5'  
    Board: 'Uno'  
    AvailablePins: {'D2-D13', 'A0-A5'}  
    AvailableDigitalPins: {'D2-D13', 'A0-A5'}  
    AvailablePWMPins: {'D3', 'D5-D6', 'D9-D11'}  
    AvailableAnalogPins: {'A0-A5'}  
    AvailableI2CBusIDs: [0]  
    Libraries: {'I2C', 'SPI', 'Servo'}  
Show all properties
```

Figura 3.1. Comprobación de comunicación de Arduino

Otra manera que se comprobó que la placa se comunique de manera correcta con el computador, es enviar una señal o comando a través del *Command Window*, activando uno de los pines de salida. Al enviar los datos, las luces LED RX y TX parpadean y encienden el LED L, se puede apreciar en la Figura 3.2.

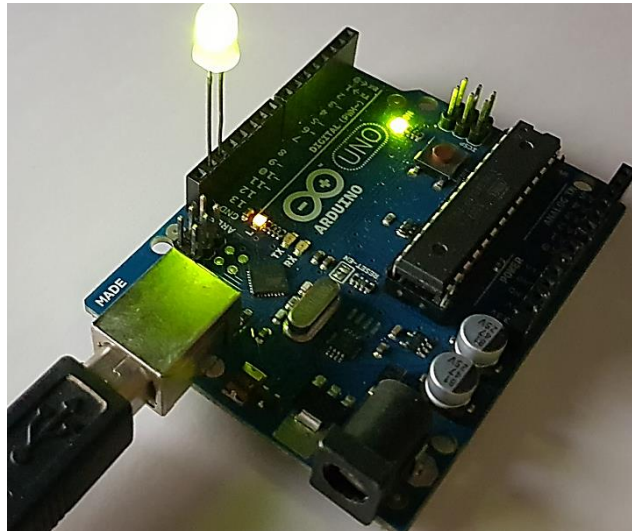


Figura 3.2. Comunicación Arduino y computador.

3.3 Conexión de periféricos.

Para la conexión de los periféricos se emplea el diagrama de conexión del Anexo VII. Para los periféricos se realizó la comprobación de continuidad.

Periféricos de entrada

La comprobación del conexionado mediante continuidad se indica en la Tabla 3.3, desde los pines de cada periférico de entrada hacia los pines de la placa Arduino.

Tabla 3.3. Conexión de los periféricos de entrada

Periféricos	Pines de los periféricos conectados a Arduino	Pines de Arduino	Continuidad	Resultado
Sensor LM35	Vcc	5V	✓	✓
	Control	A1	✓	✓
	Gnd	Gnd	✓	✓
Sensor fotorresistor	Vcc	5V	✓	✓
	Control	A3	✓	✓
	Gnd	Gnd	✓	✓
Potenciómetro	Vce	5V	✓	✓
	Control	A0	✓	✓
	Gnd	Gnd	✓	✓
Teclado Matricial	Salida A	D8	✓	✓
	Salida B	D9	✓	✓
	Salida C	D10	✓	✓
	Salida D	D11	✓	✓
	Datos A	D2	✓	✓

La tabla anterior indica que las conexiones son correctas.

Periféricos de salida

La comprobación del conexionado de los periféricos de salida mediante continuidad se indica en la Tabla 3.4.

Tabla 3.4. Conexionado de los periféricos de salida

Periféricos	Pines de los periféricos conectados a Arduino	Pines de Arduino	Continuidad	Resultado
LED	Control	D3	✓	✓
	Gnd	Gnd	✓	✓
LED 2	Control	D13	✓	✓
	Gnd	Gnd	✓	✓
Módulo relé	Vcc	5V	✓	✓
	IN1	D7	✓	✓
	IN2	D4	✓	✓
	Gnd	Gnd	✓	✓
Servomotor	Vcc	5V	✓	✓
	Control	D5	✓	✓
	Gnd	Gnd	✓	✓
Bocina	Control	D11	✓	✓
	Gnd	D2	✓	✓
LCD	Vce	5V	✓	✓
	SCL	SCL	✓	✓
	SDA	SDA	✓	✓
	Gnd	Gnd	✓	✓

El conexionado de cada periférico de salida es correcto, de acuerdo a la comprobación de la tabla anterior.

En la Tabla 3.5 se muestra la comprobación del conexionado de los pines del foco y del ventilador con los pines del módulo relé.

Tabla 3.5. Cargas del módulo relé

Cargas	Pines de las cargas conectadas al módulo relé	Pines del módulo relé	Continuidad	Resultado
Foco	Control	Relé 1, NO	✓	✓
	Gnd	Gnd	✓	✓
Ventilador	Control	Relé 2, NO	✓	✓
	Gnd	Gnd	✓	✓

De acuerdo a los datos de la tabla anterior, el conexionado del módulo relé con las cargas es correcto.

3.4 Control de periféricos desde *Command Window* de Matlab

Para la comprobación del módulo mediante la utilización de *Command Window* de Matlab, se asignó la variable (a) para vincular la placa Arduino.

Se procedió a realizar las siguientes pruebas:

Lectura de voltaje del sensor de temperatura LM35

El comando `readVoltage(a,'pin')`, se usó para la lectura del pin A1 que tiene como periférico de entrada el sensor LM35. Los valores medidos en el *Command Window* de Matlab se registra en la Tabla 3.6.

Tabla 3.6. Lectura en el pin A1

Comando	Voltaje medido (V _{DC})	Resultado
readVoltage(a,'A1')	0,154	✓
	0,201	✓
	0,209	✓
	0,251	✓

La lectura del sensor se evidencia con los datos de la tabla anterior.

Lectura de voltaje del sensor fotorresistor

Se usó el comando `readVoltage(a,'pin')`. En la Tabla 3.7 se muestran las mediciones tomadas del módulo fotorresistor, donde se implementaron las siguientes condiciones:

- Condición uno: Fotorresistor tapado.
- Condición dos: Fotorresistor sin tapar.
- Condición tres: Fotorresistor iluminado.

Tabla 3.7. Lectura en el pin A3

Comando	Condiciones	Voltaje medido (V _{DC})	Resultado
readVoltage(a,'A3')	Uno	4,1	✓
		4,07	✓
		4,05	✓

Comando	Condiciones	Voltaje medido (V_{DC})	Resultado
readVoltage(a,'A3')	Dos	3,03	✓
		3,1	✓
		3,01	✓
readVoltage(a,'A3')	Tres	0,5	✓
		0,56	✓
		0,49	✓

Los voltajes varían dependiendo la forma de implementación de cada condición, la lectura se evidencia con los datos de la tabla anterior.

Lectura de voltaje del potenciómetro

La lectura del pin A0 se realizó mediante el comando readVoltage(a,'pin'). En la Tabla 3.8 se registran los datos obtenidos del pin de control del potenciómetro en el *Command Window*.

Tabla 3.8. Comprobación del funcionamiento del potenciómetro

Comando	Nivel del potenciómetro (%)	Voltaje ideal (V_{DC})	Voltaje medido (V_{DC})	Error %	Resultado
readVoltage(a,'A0')	0	0	0,01	0,01	✓
	50	2,5	2,51	0,39	✓
	100	5	4,98	0,40	✓

De acuerdo a la posición de la perilla del potenciómetro, el voltaje varió. Por lo que los registros son correctos.

Lectura de pines del teclado matricial

En la Tabla 3.9 se registran los datos obtenidos al presionar los botones del teclado matricial, para ello se usó el comando readDigitalPin(a, 'pin').

Tabla 3.9. Lectura de datos del teclado matricial

No.	Salida D	Salida C	Salida B	Salida A	Código binario 4 bits ideal	Resultado
	readDigital Pin(a, 'D11')	readDigital Pin(a, 'D10')	readDigital Pin(a, 'D9')	readDigital Pin(a, 'D8')		
0	0	0	0	0	0000	✓
1	0	0	0	1	0001	✓
2	0	0	1	0	0010	✓
3	0	0	1	1	0011	✓
4	0	1	0	0	0100	✓
5	0	1	0	1	0101	✓
6	0	1	1	0	0110	✓
7	0	1	1	1	0111	✓
8	1	0	0	0	1000	✓
9	1	0	0	1	1001	✓
A	1	0	1	0	1010	✓
B	1	0	1	1	1011	✓
C	1	1	0	0	1100	✓
D	1	1	0	1	1101	✓

De esta forma se comprueba cual es el valor alfanumérico que se obtuvo al presionar cada botón. Dado que los valores obtenidos en binario son similares a los valores binarios ideales, la comprobación es correcta.

Escritura para el encendido del LED

Mediante el comando `writeDigitalPin(a,'D2',n)`, donde:

- “n” igual a 1, se enciende el LED
- “n” igual a 0, se apaga el LED

Para ello se tomaron los datos del encendido y apagado. En la Tabla 3.10 se registraron los estados del LED.

Tabla 3.10. Encendido y apagado del LED

Comando	Voltaje entregado (V _{DC})	Voltaje ideal (V _{DC})	Error %	Estado del LED	Resultado
writeDigitalPin(a,'D13',1)	5,01	5	0,2	Encendido	✓
writeDigitalPin(a,'D13',0)	0,0	0	0	Apagado	✓
writeDigitalPin(a,'D13',1)	5,02	5	0,4	Encendido	✓

Se comprueba el correcto funcionamiento del encendido y apagado del LED. En la Figura 3.3 se muestran los estados del LED.

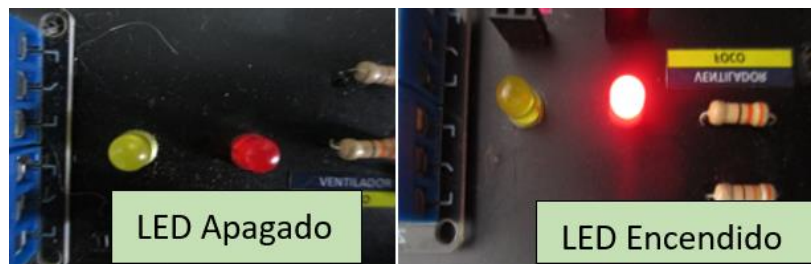


Figura 3.3. Encendido y apagado del LED

Con el comando writePWMPulse(a, 'D3', n), donde:

- “n” es el valor de voltaje.

Se generó una onda PWM que reguló el voltaje a la salida del pin D3, los registros se verifican en la Tabla 3.11.

Tabla 3.11. PWM en el pin D3.

Comando	Voltaje (V _{DC}) entregado	Voltaje ideal (V _{DC})	Error %	Estado del LED	Resultado
writePWMPulse(a, 'D3', 0)	0	0	0	Apagado	✓
writePWMPulse(a, 'D3', 2.4)	2,4	2,41	0,41	Encendido	✓

Comando	Voltaje (V _{DC}) entregado	Voltaje ideal (V _{DC})	Error %	Estado del LED	Resultado
writePWMPWMVoltage(a, 'D3', 4.9)	4,9	4,91	0,20	Encendido	✓

Se comprueba el correcto funcionamiento del LED en la Tabla anterior. El LED varía su intensidad de acuerdo al voltaje aplicado como se muestra en la Figura 3.4.



Figura 3.4. Funcionamiento del LED.

Escritura para el encendido del módulo relé, foco y ventilador

La Tabla 3.12 muestra los accionamientos de los relés al suministrar un voltaje en los pines D4 y D7 con el comando writeDigitalPin(a,'pin',n), donde:

- “n” igual a 1, se enciende el pin.
- “n” igual a 0, se apaga el pin.

Dando como resultado el encendido y apagado del ventilador y foco, respectivamente.

Tabla 3.12. Comprobación del accionamiento del módulo relé

Comando	Relé accionado	Estado del relé	Estado del relé ideal	Resultado
writeDigitalPin(a,'D4',1)	Dos	Foco encendido	Foco encendido	✓
writeDigitalPin(a,'D4',0)		Foco apagado	Foco apagado	✓

Comando	Relé accionado	Estado del relé	Estado del relé ideal	Resultado
writeDigitalPin (a,'D7',1)	Uno	Ventilador encendido	Ventilador encendido	✓
writeDigitalPin (a,'D7',0)		Ventilador apagado	Ventilador apagado	✓

De acuerdo a la tabla anterior, se comprueba el encendido del ventilador en la Figura 3.5. En la Figura 3.6 se comprueba el apagado del ventilador.



Figura 3.5. Ventilador encendido.

De acuerdo a la tabla anterior, se comprueba el encendido del foco en la Figura 3.6. En la Figura 3.5 se comprueba el apagado del foco.

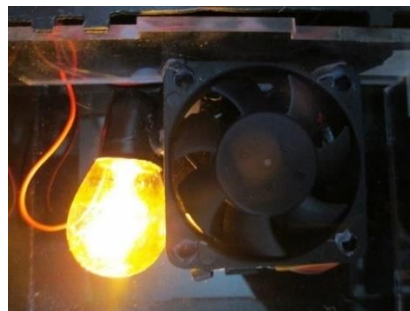


Figura 3.6. Foco encendido.

Escritura para la posición del servomotor

En la Tabla 3.13 se muestran las distintas posiciones del servomotor utilizando el comando writePosition(a,n), donde:

- “n” igual a 1 equivale a 180°.
- “n” igual a 0 equivale a 0°.

El servomotor se puede posicionar entre el rango de 0 a 1. Las mediciones de los ángulos del servomotor se realizaron mediante una escuadra angular.

Tabla 3.13. Posición del servomotor.

Comando	Posición servomotor n	Posición ideal	Resultado
writePosition(a,0)	0°	0°	✓
writePosition(a,0.5)	90°	90°	✓
writePosition(a,1)	180°	180°	✓

En la tabla anterior se comprueba la correcta posición del servomotor, los ángulos se visualizan en la Figura 3.7.

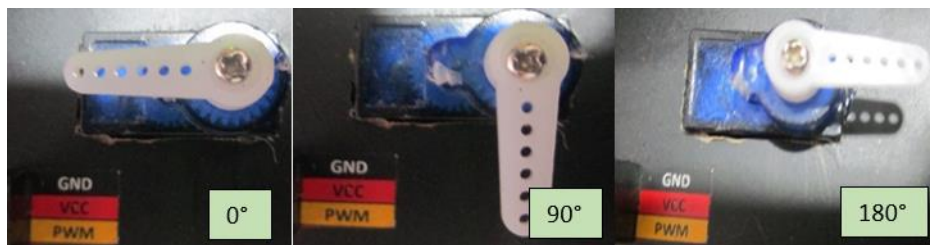


Figura 3.7. Posición del servomotor

Escritura para generar sonido en la Bocina

Mediante el comando playTone(a,'pin',f,t), donde:

- “f” es la frecuencia que genera el pin.
- “t” es el tiempo que dura el sonido.

Se registran los datos en la Tabla 3.14 que se tomaron al enviar el comando. La comprobación de sonido se realizó mediante una aplicación de celular llamada dB Meter.

Tabla 3.14. Control de bocina.

Comando	Frecuencia (Hz)	Tiempo (s)	Sonido (dB)	Resultado
playTone(a,'pin',218,5)	218	5	56	✓
playTone(a,'pin',786,5)	786		55	✓
playTone(a,'pin',2031,5)	2 031		63	✓
playTone(a,'pin',5000,5)	5 000		91	✓

La tabla anterior verifica que la bocina generó sonido.

3.5 Control de periféricos desde la interfaz de Matlab

Mediante el uso de la interfaz didáctica *App Designer*, se controlaron los periféricos de entrada y salida. En el ANEXO VIII, se muestra la pantalla de interfaz que se creó en el componente de diseño.

Para este caso se realizaron las siguientes pruebas:

Control del LED.

En la Figura 3.8 se muestran dos estados del pulsador con el que se enciende y se apaga el LED, para la verificación se consideró la Tabla 3.10 debido a que se muestra el voltaje y estados del LED.



Figura 3.8. Pulsador del LED

Control de la bocina

En la Figura 3.9 se muestran las mediciones realizadas por el variador de frecuencia de la interfaz *App Designer*, para ello se consideraron las frecuencias y la comprobación del sonido de la Tabla 3.14.

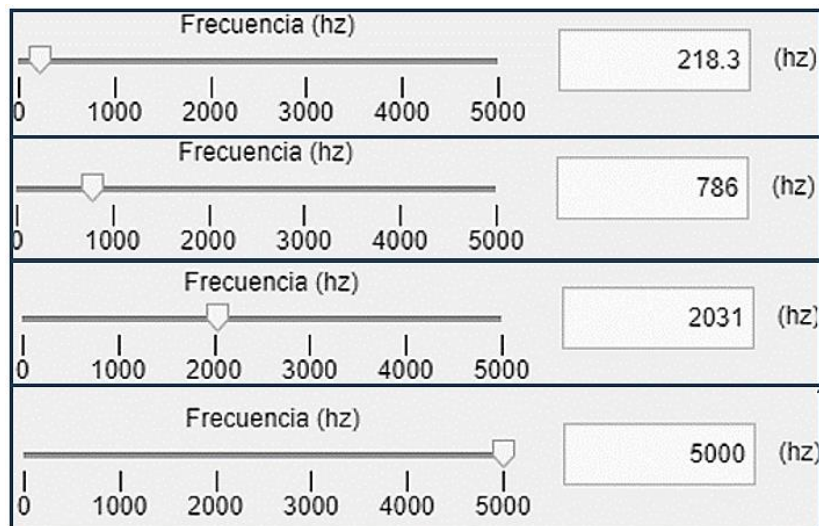


Figura 3.9. Variador de frecuencia de la bocina

Escrituras de ondas PWM en el LED

En la Tabla 3.15 se indican los datos de control del encendido del LED a diversas intensidades, como se muestran en la Figura 3.4 respecto a la variación del potenciómetro.

Tabla 3.15. Variación de voltaje con el potenciómetro

Potenciómetro %	Voltaje (V _{DC}) entregado	Voltaje ideal (V _{DC})	Error %	Estado del LED	Resultado
0	0,1	0	0,1	Apagado	✓
50	2,49	2,5	0,4	Encendido	✓
100	4,98	5	0,4	Encendido	✓

Se verificó con la tabla anterior el funcionamiento del LED y del potenciómetro al activarse en el *App Designer*.

Posicionamiento del servomotor

Los grados del variador se indican en la Figura 3.10, estos se usaron para controlar el servomotor. La Tabla 3.13 indica los datos y los resultados de las posiciones del servomotor.

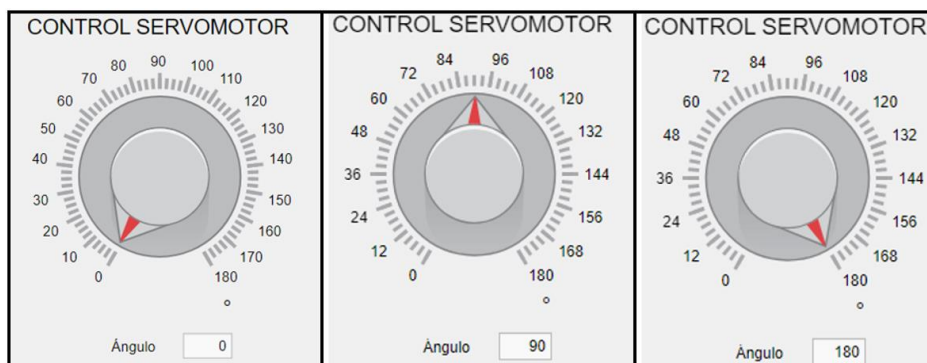


Figura 3.10. Controlador de servomotor

Lectura de voltaje del sensor fotorresistor

Los datos de voltaje obtenidos a través la interfaz de *App Designer* se indican en la Figura 3.11 al presionar el pulsador fotorresistor, para este caso se consideraron las condiciones y voltajes de la Tabla 3.7.

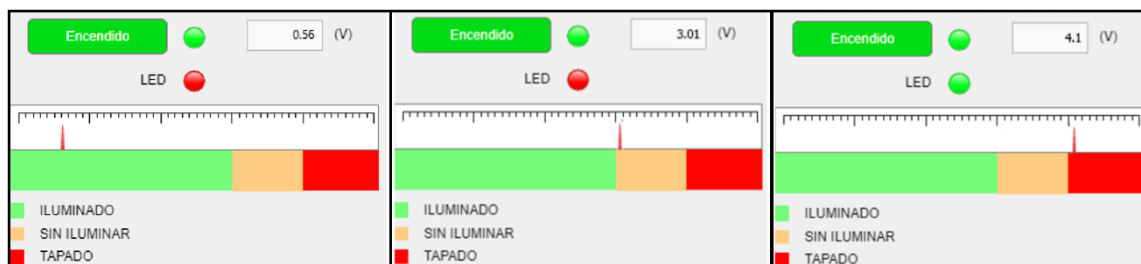


Figura 3.11. Lectura del fotorresistor

Cada que el medidor llegue al punto de TAPADO, el LED de la interfaz se enciende.

Lectura de temperatura del sensor LM35

Los valores de temperatura se indican en la Figura 3.12 , donde:

- La imagen 1 y 3 muestran el valor de temperatura de referencia.
- La imagen 2 y 4 muestran la temperatura medida por el sensor LM35.

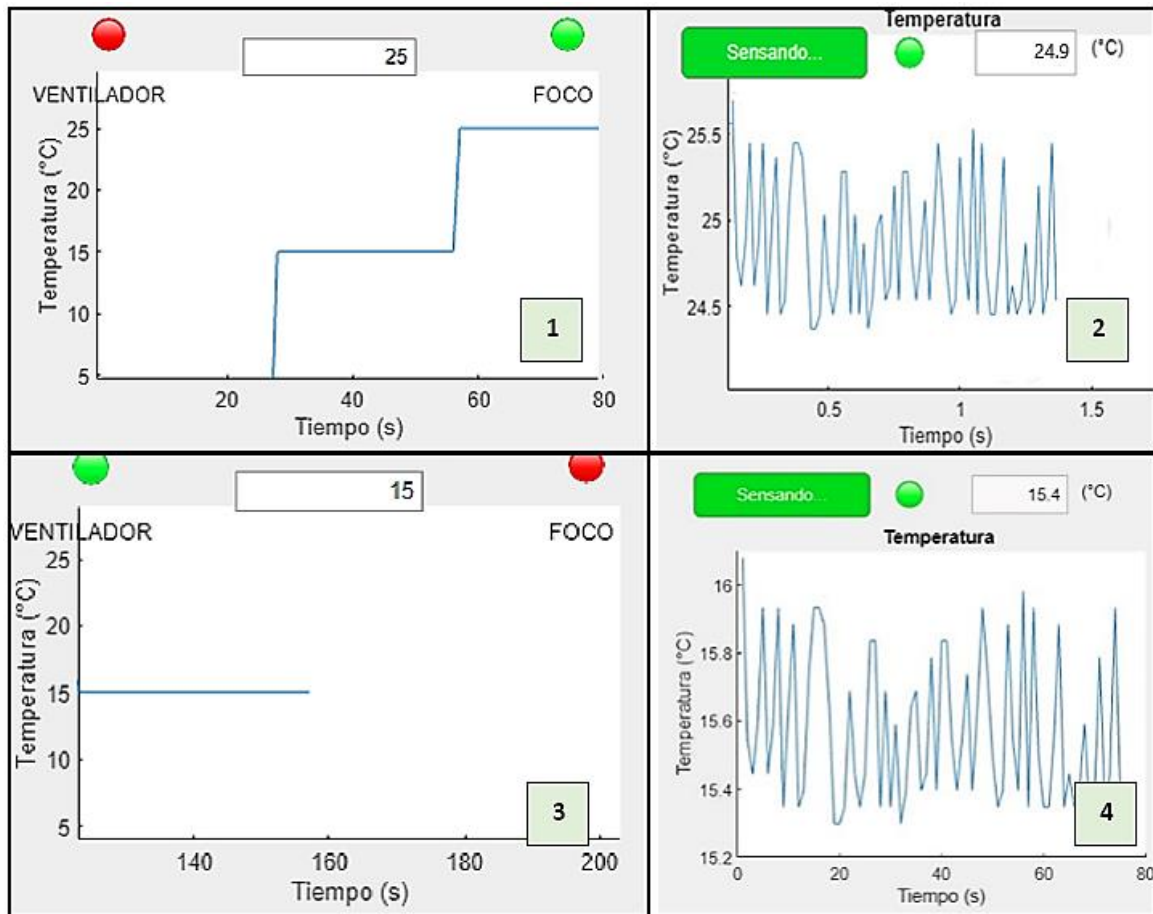


Figura 3.12. Mediciones del sensor LM35 y temperatura de referencia

Para este caso se usó un control por histéresis que permite un rango de tolerancia de $\pm 10\%$ del valor de temperatura de referencia considerando la sensibilidad del sensor. Para ello, si el valor medido está por debajo del valor de referencia, se encenderá el foco, caso contrario se encenderá el ventilador.

En la tabla 3.16 se registran los datos obtenidos de la interfaz *App Designer*.

Tabla 3.16. Mediciones del sensor de temperatura

Temperatura medida (°C)	Temperatura de referencia o ideal (°C)	Error %	Estado de la Carga	Resultado
15,4	15	2,66	Ventilador encendido y foco apagado	✓
24,6	25	1,6	Foco encendido y ventilador apagado	✓

Se comprueba el correcto funcionamiento de la interfaz para la lectura del sensor de temperatura en la figura y tabla anterior

3.6 Control de periféricos desde Arduino

Para este caso se utilizó el código de programación para Arduino del componente de diseño.

Se realizaron las siguientes pruebas:

Lectura de temperatura del sensor LM35 y accionamiento del módulo relé

Los datos de temperatura (T) y los datos de temperatura de referencia (SP) obtenidos a través de la pantalla LCD se registraron en la Tabla 3.17. El accionamiento del módulo relé y estado de las cargas se verifica de acuerdo a la Tabla 3.12.

Tabla 3.17. Datos del sensor LM35

Temperatura medida (°C)	Temperatura de referencia o ideal (°C)	Error %	Resultado
15,4	15	2,66	✓
19,8	20	1	✓
22,1	22	0,45	✓
24,9	25	0,4	✓

La comprobación de medición de temperatura respecto a la temperatura ideal de la tabla anterior, es correcta.

Los valores de SP varían de acuerdo a la posición de la perilla del potenciómetro.

Posición del servomotor

La posición del servomotor se controló mediante el cambio de la perilla del potenciómetro, los datos de la Tabla 3.18 se obtuvieron a través de la pantalla LCD.

Tabla 3.18. Posición del servomotor

Grados (GR)	Grados del servomotor	Resultado
0°	0°	✓
90	90	✓
180°	180°	✓

Se comprueba la correcta posición del servomotor de la tabla anterior. En la Figura 3.7 se visualiza la posición del servomotor.

Encendido del LED con el potenciómetro

En la tabla 3.19 se muestran los voltajes y el estado del LED al mover la perilla del potenciómetro.

Tabla 3.19. Encendido del LED

Potenciómetro	Voltaje (V _{DC}) entregado	Voltaje ideal	Error %	Estado del LED	Resultado
0%	0,02	0	0.02	Apagado	✓
50%	2,4	2,5	4	Encendido	✓
100%	4,9	5	2	Encendido	✓

De acuerdo a la tabla anterior y la Figura 3.4, se comprueba el correcto funcionamiento del LED.

Ingreso de valores alfanuméricos a través del teclado matricial

En la Tabla 3.20 se registraron los datos ingresados en el teclado matricial y el comportamiento al ingresar claves erróneas.

Tabla 3.20. Teclado matricial

Clave correcta	Claves ingresadas	Datos LCD	Resultado
1234	1246	Clave Incorrecta	✓
	1234	Clave Correcta	✓
	4567	Clave Incorrecta	✓

Como se muestra en la tabla anterior, al momento de ingresar claves erróneas en el teclado matricial, el LCD muestra el estado de contraseña y al ser incorrecto pide nuevamente la clave de ingreso. Por lo que, se comprueba el correcto funcionamiento.

Lectura del sensor fotorresistor para dar tonos en la bocina

La bocina generó sonido al mover la perilla del potenciómetro, el sonido se evidenció mediante la aplicación dB Meter y se registraron los datos en la Tabla 3.21.

Tabla 3.21. Sonido de la bocina

Potenciómetro	Frecuencia (Hz)	Sonido (dB)	Tiempo (s)	Resultado
4%	218	56	5	✓
16%	786	55		✓
41%	2 031	63		✓
100%	5 000	91		✓

La bocina suena con frecuencias entre 0 a 5 000 (Hz) dependiendo de la posición de la perilla del potenciómetro. De acuerdo a la tabla anterior, se comprueba el correcto funcionamiento de la bocina.

4 CONCLUSIONES

- El módulo contiene los siguientes periféricos que son:
 - Periféricos de entrada: Sensor de temperatura LM35, sensor fotorresistor, pulsadores, potenciómetro y teclado matricial.
 - Periféricos de salida: LED, relés, servomotor, bocina y LCD.
 - Cargas de los relés: foco y ventilador.
- En este proyecto se limitó el uso de periféricos adaptables para Arduino, dado que la placa Arduino Uno tiene solamente seis pines para entradas analógicas y catorce pines de entradas y salidas digitales, además, se consideró el suministro máximo de corriente igual a 50 (mA) que abastece la placa Arduino y la corriente de la fuente externa de 2 (A).
- Se considera importante el conocer cómo actúan los periféricos de entrada ante las variables de temperatura y luminosidad.
 - El sensor LM35 es utilizado en este proyecto para la medición de temperatura del ambiente y el control simple en lazo cerrado al activar y desactivar las cargas mediante el módulo relé.
 - El sensor fotorresistor al ser iluminado por una fuente de luz artificial da voltajes aproximados a 0 (V_{DC}), al tapar el sensor fotorresistor los voltajes de medición se aproximan a 5 (V_{DC}).
- Para la generación de sonido de la bocina, se ha considerado un rango de frecuencia de 0 a 5000 (Hz), dado que la audición óptima del ser humano se encuentra en rango de 20 a 20 000 (Hz) [9].
- Para construir la base del módulo se optó por un material que cumpliera con las exigencias de resistencia al manipularse, dado que al compararlo con otros materiales como: vidrio, papel prensado o madera MDF. El acrílico es ligero y duradero.

5 RECOMENDACIONES

- Como parte inicial del proyecto, se considera importante conocer cómo actúa cada fuente de alimentación, al disponer de dos fuentes en el módulo, se recomienda no mezclar las tierras, debido a que se puede ocasionar un cortocircuito en la placa Arduino o en los periféricos de entrada y salida.
- Para el cambio de conexión de cada dispositivo es recomendable desconectar las fuentes de alimentación y luego procede a hacer los cambios necesarios en el circuito, esto evita el daño en los pines del Arduino.
- El integrado de placa Arduino no soporta cargas excesivas en sus pines de alimentación, por lo que se requiere una alimentación externa para poder satisfacer esa demanda, para ello se debe conocer que dispositivo es necesario conectar exclusivamente a Arduino y que dispositivo es requerido conectar a la fuente externa.
- En los módulos no es posible trabajar simultáneamente con la interfaz de *App Designer*, la pantalla *Command Window* y Arduino. Por lo que, se recomienda usar primero *Command Window*, luego *App Designer* y finalmente Arduino IDE.
- Dado que el material con el que se construyó es acrílico, es vulnerable ante fuego intenso, a químicos corrosivos y a roturas por golpes bruscos. Para esto se recomienda manipular el módulo con cuidado.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. d. F. d. Tecnólogos, «Área Microprocesadores,» ESFOT, [En línea]. Available: <https://esfot.epn.edu.ec/index.php/esfot/445-area-microprocesadores>. [Último acceso: 15 07 2023].
- [2] O. G. Puertas, «Fabricación de placas de circuito impreso con Proteus,» Apuntes de electrónica., 09 2015. [En línea]. Available: <https://uvadoc.uva.es/bitstream/handle/10324/14625/TFG-P-312.pdf>. [Último acceso: 15 08 2023].
- [3] Rosi, «Características de Arduino,» 11 10 2018. [En línea]. Available: <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/CaracteristicasArduino.pdf>. [Último acceso: 15 08 2023].
- [4] J. Little, «MathWorks,» The MathWorks, inc, 1994-2023. [En línea]. Available: <https://la.mathworks.com/products/matlab.html>. [Último acceso: 15 07 2023].
- [5] MathWorks, «Hardware Support,» The MatWorks, inc, 1994-2023. [En línea]. Available: <https://la.mathworks.com/hardware-support/arduino-matlab.html>. [Último acceso: 16 07 2023].
- [6] K. Soberby, «A Guide to Arduino & the I2C Protocol (Two Wire),» Arduino Docs, 18 11 2021. [En línea]. Available: <https://docs.arduino.cc/learn/communication/wire>. [Último acceso: 16 07 2023].
- [7] Arduino, «Arduino UNO Rev3 with Long Pins,» Arduino Docs, 08 2022. [En línea]. Available: <https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins>. [Último acceso: 17 07 2023].
- [8] FLUKE, «Ley de ohm,» 2023. [En línea]. Available: <https://www.fluke.com/es-ec/informacion/blog/electrica/que-es-la-ley-de-ohm>. [Último acceso: 15 08 2023].
- [9] R. pujol, «Campo auditivo humano,» Maria Morell, 06 06 2018. [En línea]. Available: <http://www.cochlea.org/es/sonidos/campo-auditivo-humano>. [Último acceso: 19 08 2023].
- [10] P. Dore, «Arduino,» Arduino, 15 09 2021. [En línea]. Available: <https://www.arduino.cc/en/about>. [Último acceso: 15 07 2023].
- [11] Hubor, «Hubor-Proteus,» Hubor, 2015. [En línea]. Available: <https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2-proteus.html>. [Último acceso: 16 07 2023].
- [12] M. G, «¿Qué es Autodesk Inventor y cuáles son sus características principales?,» 3D natives, 13 06 2022. [En línea]. Available: <https://www.3dnatives.com/es/que-es-autodesk-inventor-13062022/>. [Último acceso: 16 07 2023].
- [13] Omar, «Caida de voltaje en leds segun el color,» pinterest, 2022. [En línea]. Available: <https://www.pinterest.com/pin/611363718137787530/>. [Último acceso: 17 07 2023].

7 ANEXOS

ANEXO I. Porcentaje de similitud generado por turnitin.

ANEXO II. Certificado de funcionamiento del trabajo de Integración Curricular.

ANEXO III. Dimensiones de corte para la placa electrónica.

ANEXO IV. Circuito Impreso.

ANEXO V. Distribución de periféricos.

ANEXO VI. Dimensiones de corte para la base del módulo.

ANEXO VII. Diagrama de conexión del módulo.

ANEXO VIII. Interfaz *App Designer*.

ANEXO I. REPORTE DE SIMILITUD GENERADO POR TURNITIN.

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de agosto de 2023

De mi consideración:

Yo, Catalina Elizabeth Armas Freire, en calidad de director del Trabajo de Integración Curricular, certifico que el siguiente es el resultado de la evaluación de similitud realizado por la plataforma Turnitin:

Fecha de entrega: 22-ago-2023 07:10a.m. (UTC-0500)

Identificador de la entrega: 2149403805

Nombre del archivo: Gansino.docx (8.81M)

Total de palabras: 6216

Total de caracteres: 31121

DocumentoGansino

INFORME DE ORIGINALIDAD

4%

INDICE DE SIMILITUD

4%

FUENTES DE INTERNET

1%

PUBLICACIONES

1%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

bibdigital.epn.edu.ec

Fuente de Internet

<1%

2

es.slideshare.net

Fuente de Internet

<1%

Director

Ing. Catalina Elizabeth Armas Freire. MSc.

ANEXO II. CERTIFICADO DE FUNCIONAMIENTO DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 22 de agosto de 2023

Yo, CATALINA ELIZABETH ARMAS FREIRE, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de integración curricular, certifico que he constatado la correcta construcción de los módulos didáctico, el cual fue implementado por el estudiante Christopher Mario Gansino Llanos.

El proyecto cumple con los requerimientos de construcción y parámetros necesarios para que los usuarios de la ESFOT puedan usar los módulos con seguridad.

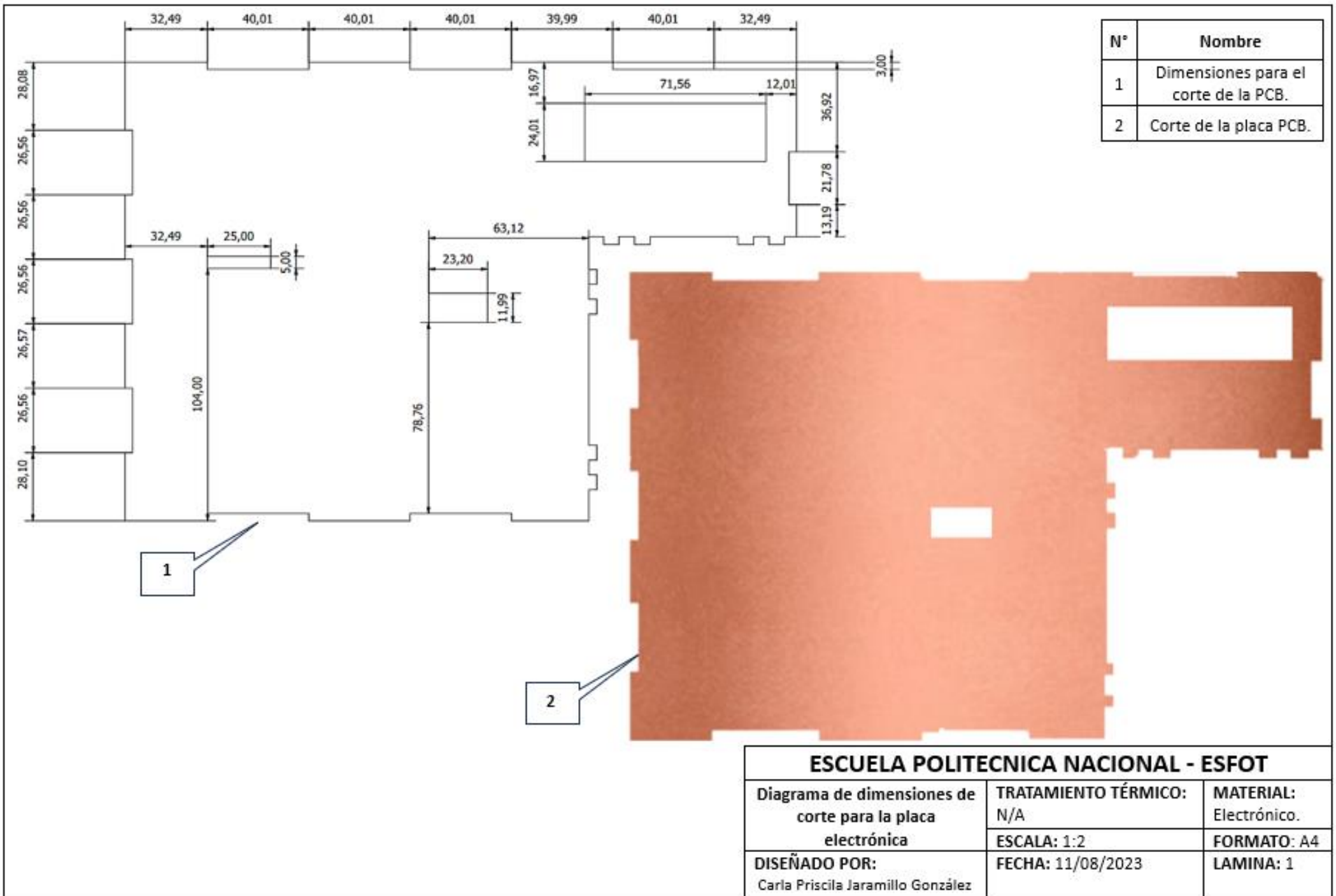
Director

Ing. Catalina Elizabeth Armas Freire. MSc.

Ladrón de Guevara E11-253 y Andalucía | Edificio N. 21 | Área 3 | Oficina 21

Correo: elizabeth.armas@epn.edu.ec

**ANEXO III. DIMENSIONES DE CORTE PARA LA PLACA
ELECTRÓNICA.**

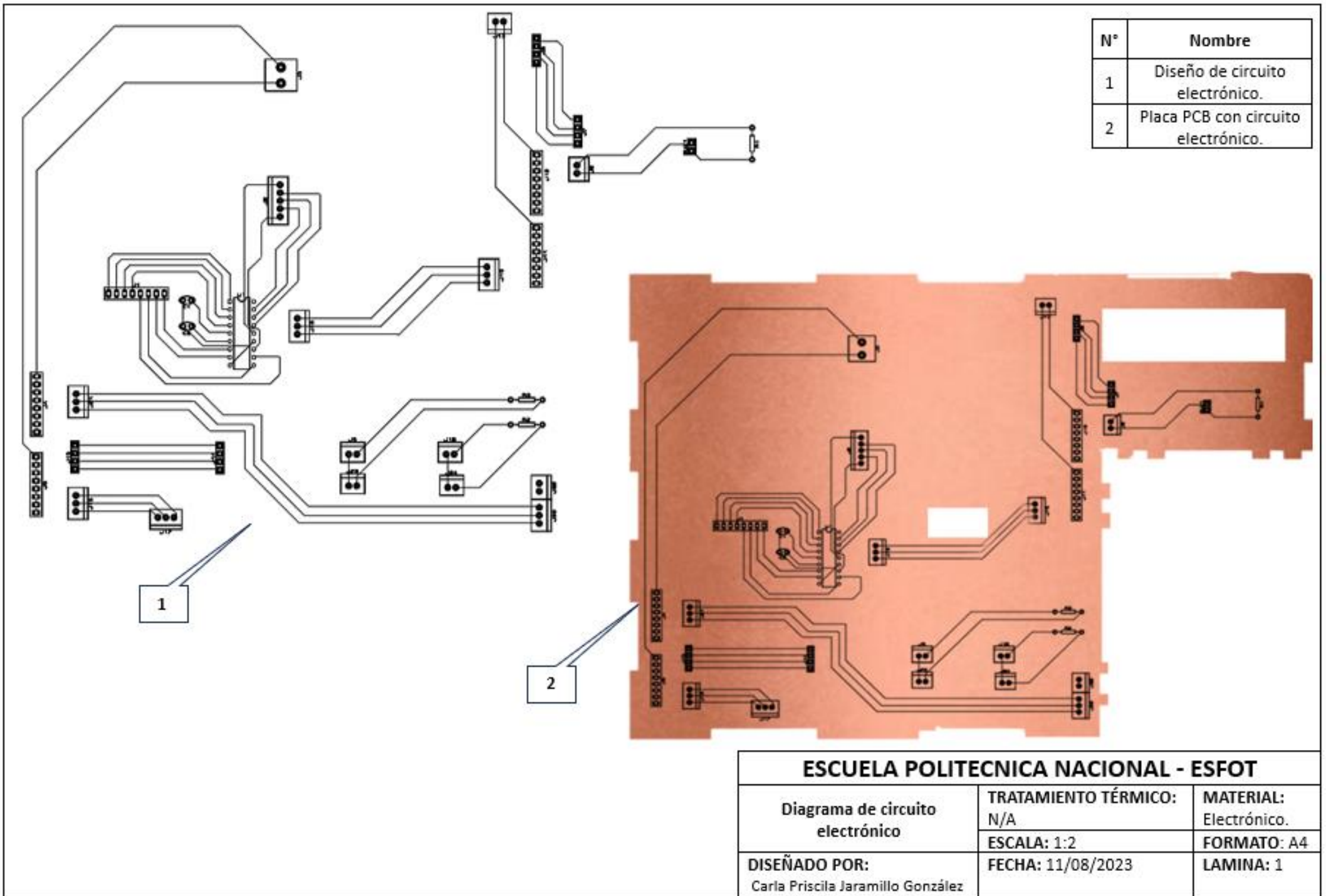


N°	Nombre
1	Dimensiones para el corte de la PCB.
2	Corte de la placa PCB.

ESCUELA POLITECNICA NACIONAL - ESFOT

Diagrama de dimensiones de corte para la placa electrónica	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Electrónico.
	ESCALA: 1:2	FORMATO: A4
DISEÑADO POR: Carla Priscila Jaramillo González	FECHA: 11/08/2023	LAMINA: 1

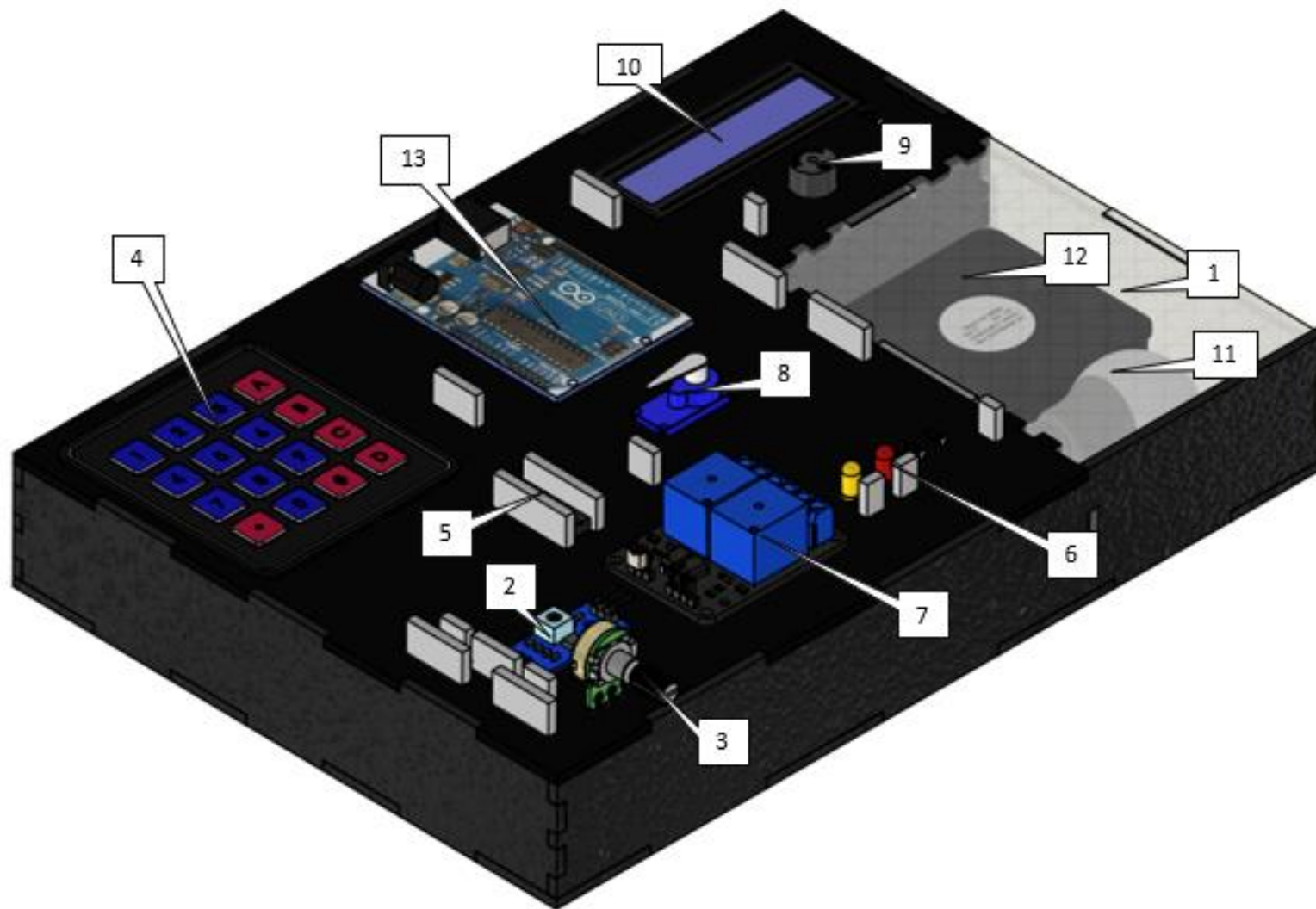
ANEXO IV. CIRCUITO IMPRESO.



N°	Nombre
1	Diseño de circuito electrónico.
2	Placa PCB con circuito electrónico.

ESCUELA POLITECNICA NACIONAL - ESFOT		
Diagrama de circuito electrónico	TRATAMIENTO TÉRMICO:	MATERIAL:
	N/A	Electrónico.
DISEÑADO POR: Carla Priscila Jaramillo González	ESCALA: 1:2	FORMATO: A4
	FECHA: 11/08/2023	LAMINA: 1

ANEXO V. DISTRIBUCIÓN DE PERIFÉRICOS.



Periféricos de Entrada	
N°	Nombre del componente
1	Sensor LM35
2	Sensor fotorresistor
3	Potenciómetro
4	Teclado matricial
5	Integrado para el teclado matricial

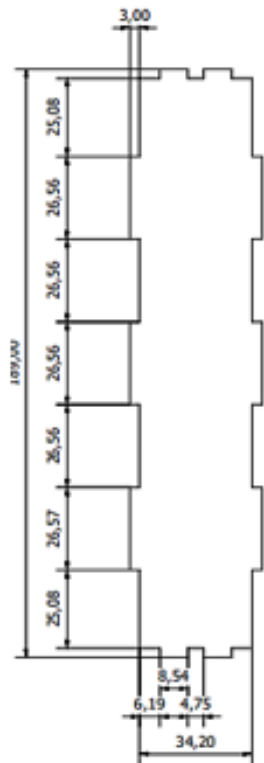
Periféricos de Salida	
N°	Nombre del componente
6	LED
7	Módulo relé
8	Servomotor
9	Bocina
10	LCD

Cargas del módulo relé	
N°	Nombre del componente
11	Bombilla incandescente
12	Ventilador
13	Modulo arduino

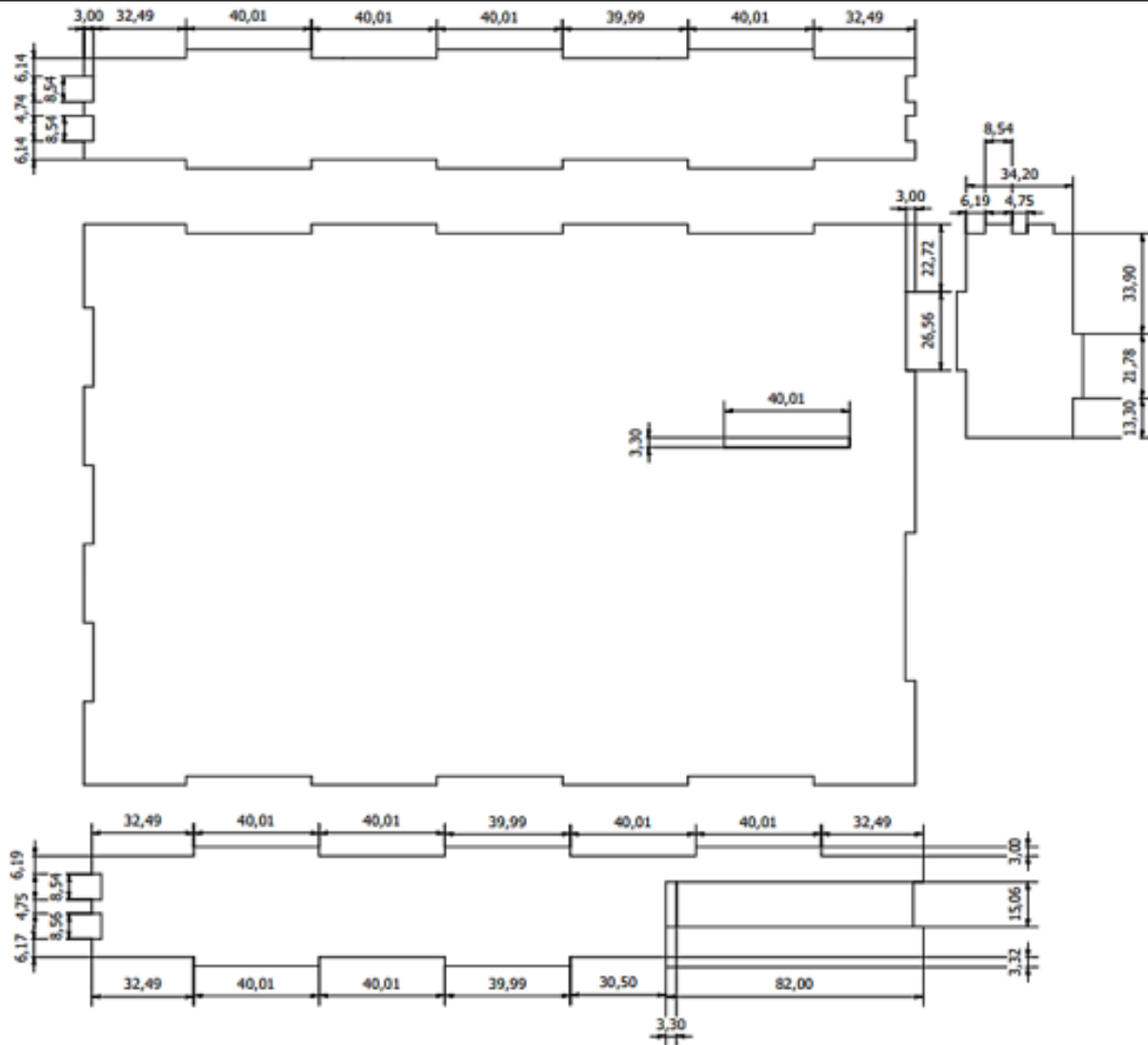
ESCUELA POLITECNICA NACIONAL - ESFOT

Disposición de periféricos	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Electrónico.
	ESCALA: 1:16	FORMATO: A3
DISEÑADO POR: Carla Priscila Jaramillo González	FECHA: 11/08/2023	LAMINA: 4

**ANEXO VI. DIMENSIONES DE CORTE PARA LA BASE DEL
MÓDULO.**

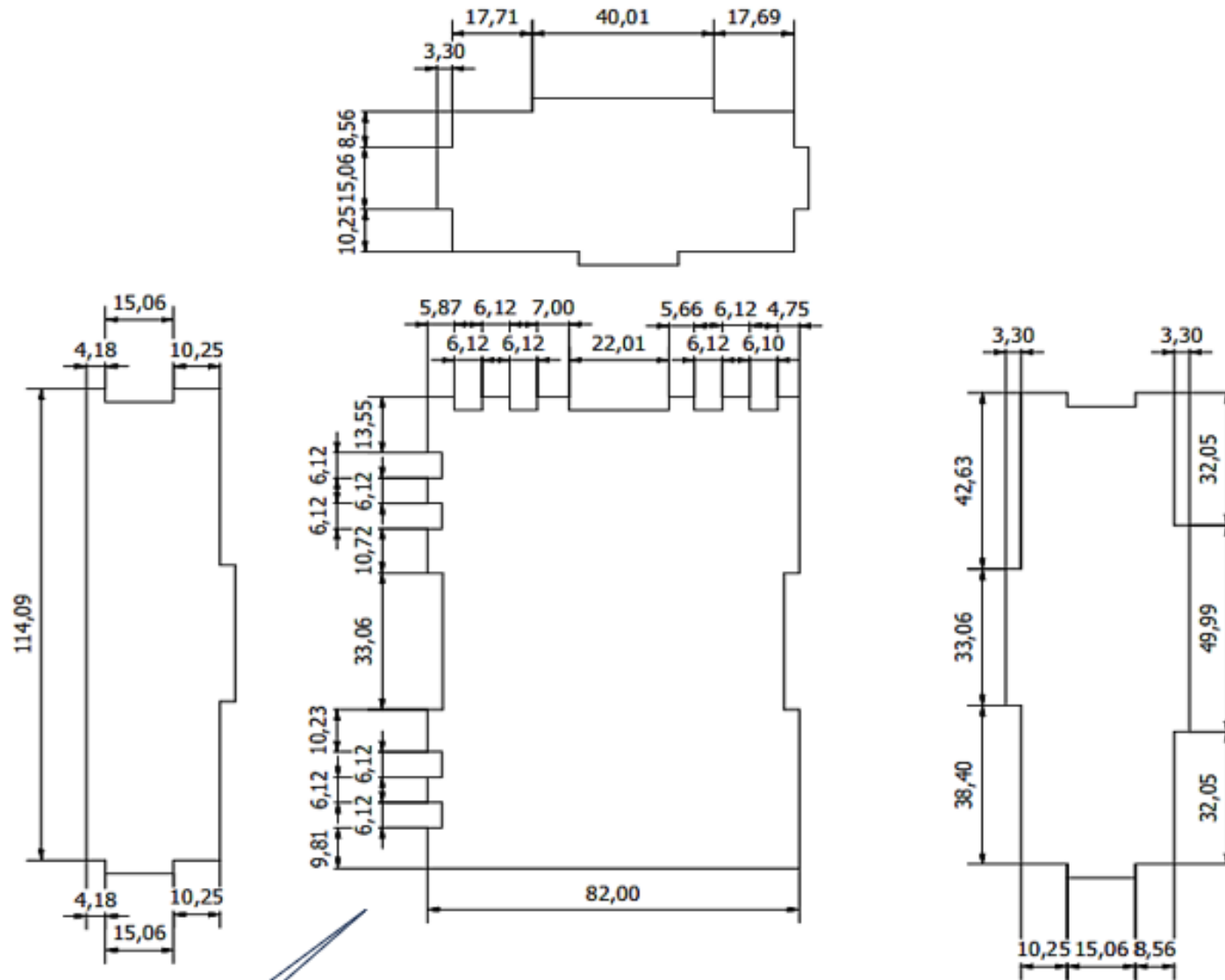


1



N°	Nombre
1	Diseño de la base del módulo.
2	Diseño de la caja para el ventilador y foco del módulo.

ESCUELA POLITECNICA NACIONAL - ESFOT		
Dimensiones de corte para la base del módulo	TRATAMIENTO TÉRMICO:	MATERIAL:
	N/A	Electrónico.
DISEÑADO POR:	ESCALA: 1:2	FORMATO: A4
	Carla Priscila Jaramillo González	FECHA: 11/08/2023
		LAMINA: 1



2

ESCUELA POLITECNICA NACIONAL - ESFOT

Dimensiones de corte para la base del módulo	TRATAMIENTO TÉRMICO: N/A	MATERIAL: Electrónico.
	ESCALA: 1:2	FORMATO: A4
DISEÑADO POR: Carla Priscila Jaramillo González	FECHA: 11/08/2023	LAMINA: 2

ANEXO VII. DIAGRAMA DE CONEXIÓN DEL MÓDULO.

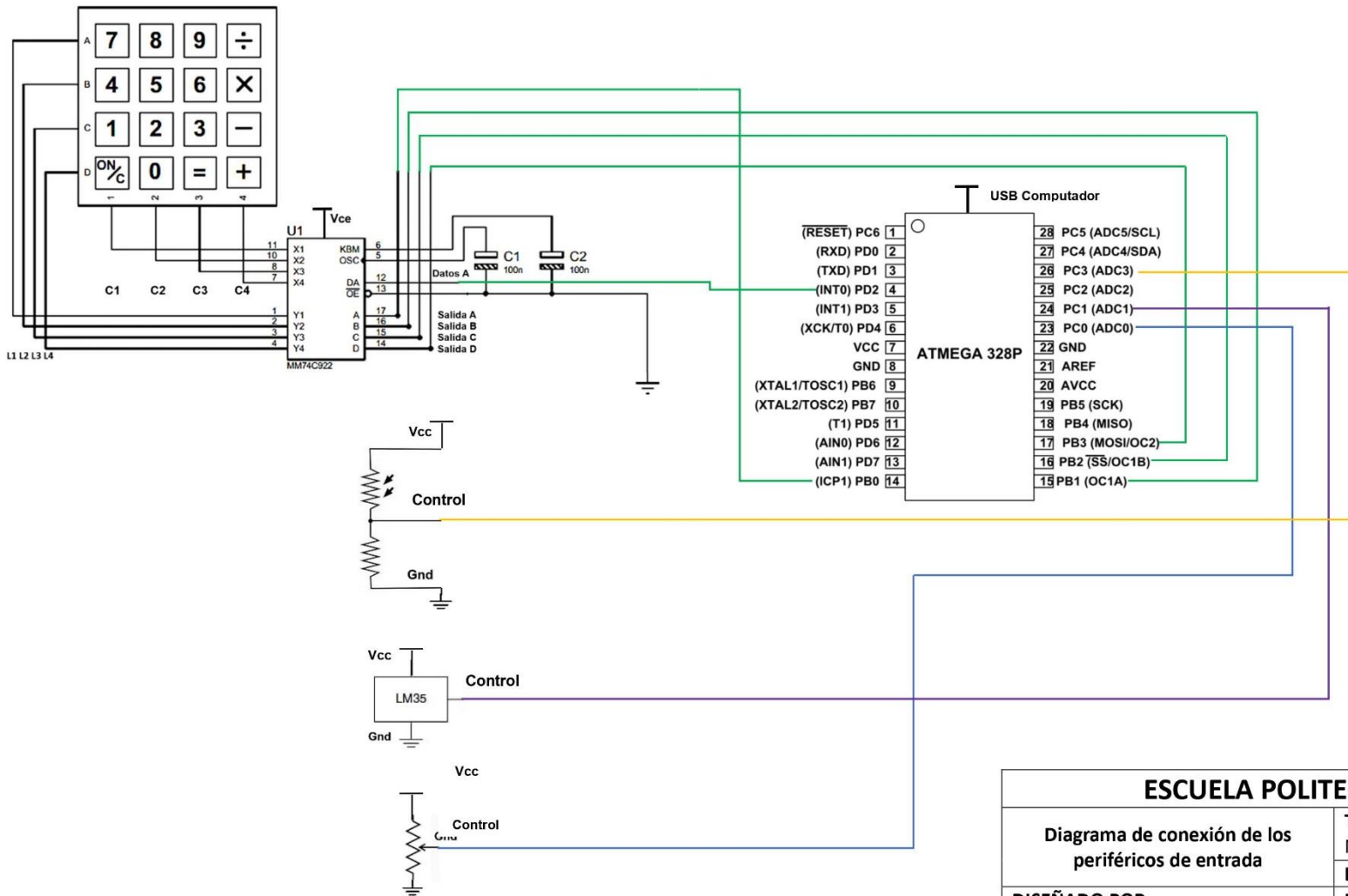


Tabla de simbología		
Nº	Símbolo	Nombre
1		Capacitor
2		Fotorresistor
3		Sensor de temperatura
4		Teclado matricial
5		Integrado para código binario de 4 dígitos del teclado matricial
6		Arduino Uno
7		Potenciómetro
8		Resistencia

ESCUELA POLITECNICA NACIONAL - ESFOT		
Diagrama de conexión de los periféricos de entrada	TRATAMIENTO TÉRMICO:	N/A
	ESCALA:	1:1
DISEÑADO POR:	FECHA:	11/08/2023
Christopher Mario Gansino Llanos		MATERIAL:
		Electrónico.
		FORMATO:
		A4
		LAMINA:
		1

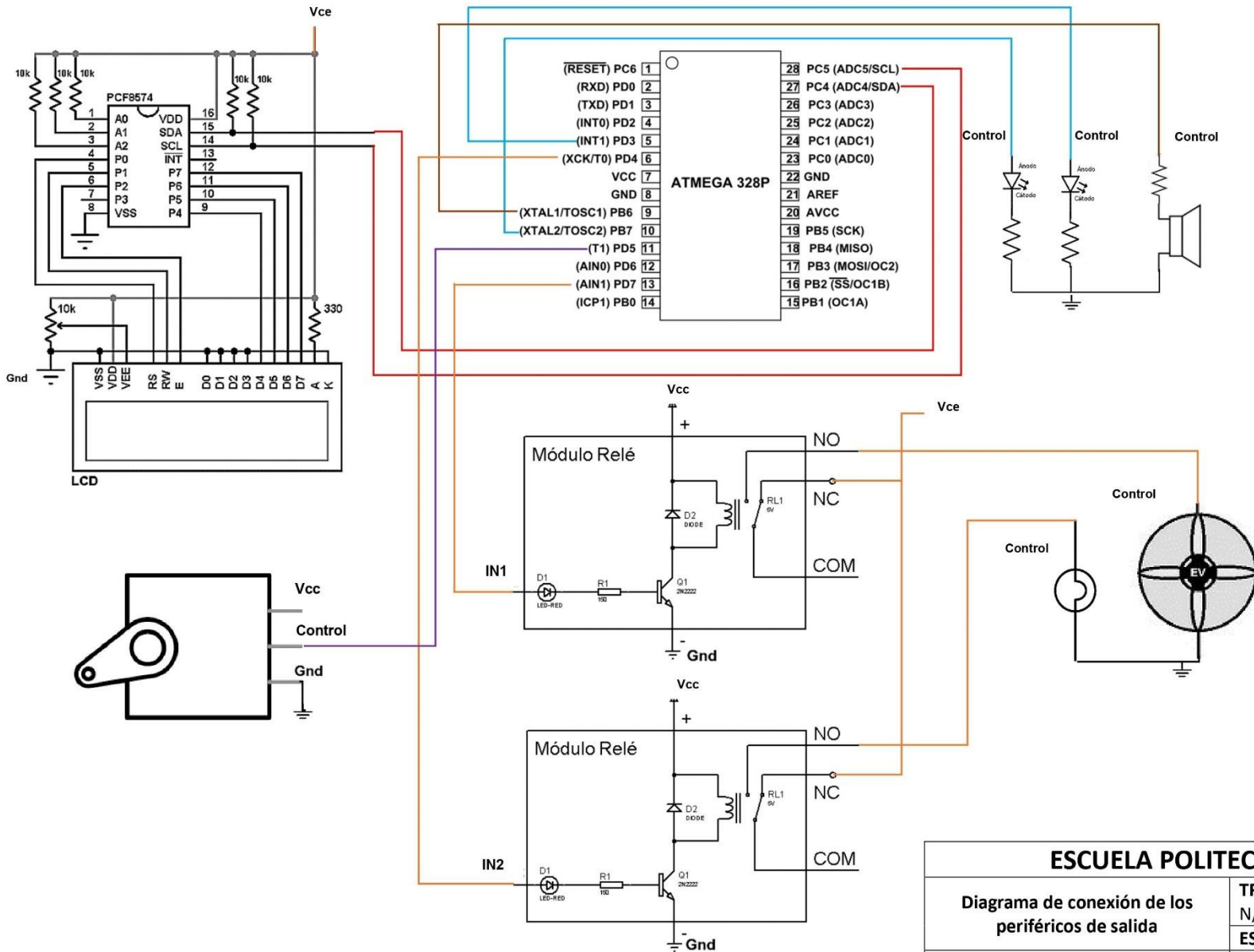


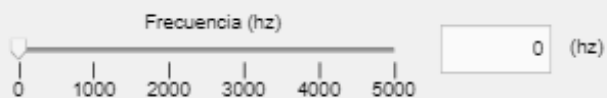
Tabla de simbología		
Nº	Símbolo	Nombre
1		Resistencia
2		LED
3		Bocina
4		Módulo Relé
5		Ventilador
6		Foco
		Servomotor
		LCD
		Arduino Uno
		I2C para el ICD

ESCUELA POLITECNICA NACIONAL - ESFOT		
Diagrama de conexión de los periféricos de salida	TRATAMIENTO TÉRMICO:	MATERIAL:
	N/A	Electrónico.
DISEÑADO POR: Christopher Mario Gansino Llanos	ESCALA: 1:1	FORMATO: A4
	FECHA: 11/08/2023	LAMINA: 1

ANEXO VIII. INTERFAZ APP DESIGNER.

CONTROL DE PUERTOS

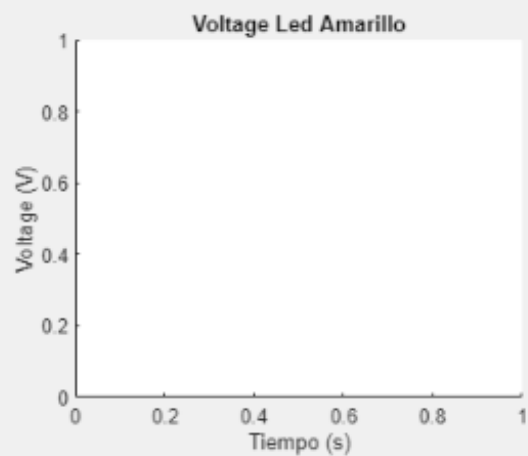
LED



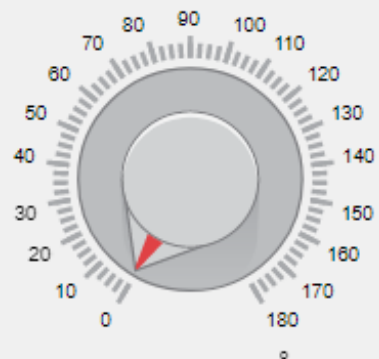
LED PWM



0 (V)



CONTROL SERVOMOTOR



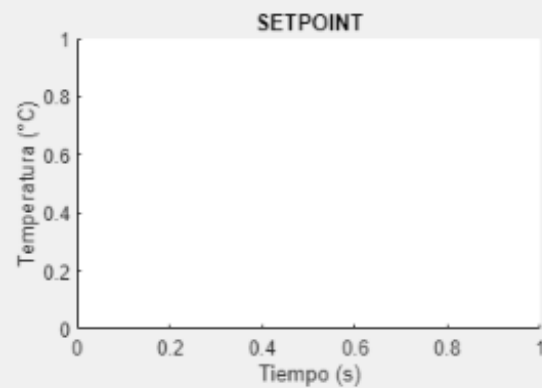
Ángulo 0

VENTILADOR



0

FOCO



SENSORES

FOTORESISTOR



0 (V)

LED



- SIN MOVIMIENTO
- MOVIMIENTO
- EMERGENCIA

LM35



0 (°C)

