

DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO ENTRENADOR A BAJO COSTO
PARA EL LABORATORIO DEL CURSO SISTEMAS DIGITALES
MICROCONTROLADOS

LUIS ANGEL DÁVILA CAUSIL

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
INGENIERÍA ELECTRÓNICA
MONTERÍA
2023

DISEÑO Y CONSTRUCCIÓN DE UN MÓDULO ENTRENADOR A BAJO COSTO
PARA EL LABORATORIO DEL CURSO SISTEMAS DIGITALES
MICROCONTROLADOS

LUIS ANGEL DÁVILA CAUSIL

Trabajo de grado para optar al título de
Ingeniero Electrónico

Asesor
RICHARD ADOLFO CRAWFORD VIDAL
Ingeniero Electrónico

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
INGENIERÍA ELECTRÓNICA
MONTERÍA
2023

ÍNDICE:

RESUMEN	6
ABSTRACT.....	6
INTRODUCCIÓN	7
1. MARCO TEÓRICO Y ESTADO DEL ARTE	9
1.1 Marco conceptual.....	10
1.1.1 Microcontrolador.....	10
1.1.1.1 Partes de un Microcontrolador	10
1.1.1.2 ¿Qué se puede hacer con un microcontrolador?	11
1.1.2 Microcontrolador PIC16F84A	11
1.1.2.1 Características.....	11
1.1.3 ¿Qué se necesita para grabar un programa en un microcontrolador? ...	11
1.1.4 PCB o circuito impreso	12
1.1.4.1 Diseño de la PCB mediante software	13
1.1.5. Kit o tarjeta de desarrollo.....	14
1.1.6. Dispositivos y elementos usados con los microcontroladores	19
1.2 Conclusión del capítulo.....	24
2. METODOLOGÍA	25
2.1. Determinar los componentes necesarios a usar en el Módulo Entrenador cumpliendo lo requerimientos de la academia:	26
2.1.1. Etapa de elección del microcontrolador.....	26
2.1.2. Elección de módulos	29
2.1.3. Etapa de estudio del lenguaje c y librerías soportadas por el ESP32 .	32
2.1.4. Cantidad de dispositivos.....	36
2.2. Etapa del diseño del Módulo entrenador utilizando de los elementos con el microcontrolador y dispositivos a usar.	37
2.2.1. Etapa de conexión de los elementos con el Microcontrolador.....	38
2.2.2. Prueba conjunta de los elementos	39
2.2.3. Diseño del esquemático de la conexión de los elementos con el Microcontrolador.....	40
2.3. Etapa de implementación del diseño del Módulo entrenador en una PCB acoplado cada uno de los elementos.	42
2.3.1. Diseño de la PCB a partir del esquemático del circuito.	42
2.3.2. Inclusión del PIC 16F84A	45
2.3.3. Impresión de la PCB.....	46
2.4. Validación del Módulo Entrenador.....	51
2.4.1. Realización del taller donde se usan varios elementos del micro y se observa el funcionamiento del kit.	51
2.4.2. Resultados del taller.....	53
3. RESULTADOS.....	57

3.1.	Estructura de la tarjeta de desarrollo.....	57
3.1.1.	Ubicación de los microcontroladores.....	58
3.1.2.	Salidas a los elementos de la placa.	58
3.1.3.	Entrada de corriente directa (DC).....	59
3.1.4.	Suiches.....	60
3.1.5.	Entradas analógicas.....	61
3.1.6.	Pulsadores	62
3.1.7.	LEDS	62
3.1.8.	Sensor de temperatura.....	63
3.1.9.	Buzzer	63
3.1.10.	Display 7 segmentos.	64
3.1.11.	Pantalla OLED.....	65
3.1.12.	Salidas auxiliares del ESP32.....	66
3.1.13.	Entradas auxiliares PIC 16F84A.....	67
3.1.14.	Suiches de encendido y selección de voltaje.	68
3.2.	Costos.....	68
4.	Manual de usuario.....	70
4.1.	Partes del Módulo Entrenador.....	70
4.1.1.	Alimentación DC.....	71
4.1.2.	Suiches.....	72
4.1.3.	Entradas analógicas.....	74
4.1.4.	LEDS.....	75
4.1.5.	Buzzer y sensor de temperatura.....	76
4.1.6.	Pantalla OLED.....	77
4.1.7.	Display 7 segmentos.	78
4.2.	Manual guía del Módulo Entrenador ESP32 Y PIC16F84A.....	80
4.2.1	Programación del PIC16F84A.....	80
4.2.1.1	Configuración del Entorno de Desarrollo	80
4.2.1.2	Creación de un Nuevo Proyecto	80
4.2.1.3	Programación en C.....	80
4.2.1.4	Compilación	80
4.2.1.5	Programación del PIC16F84A	80
4.2.2	Programación del ESP32 con el IDE Arduino.	81
4.2.2.1	Configuración del Entorno	81
4.2.2.2	Configuración del Arduino para ESP32	81
4.2.3	Librerías para el sensor de temperatura y pantalla OLED.....	81
4.2.3.1	Sensor de Temperatura DS18B20	81
4.2.3.2	Pantalla OLED.....	82
4.2.4	Consideraciones al momento de conectar los microcontroladores.	82
5.	CONCLUSIONES.....	85
6.	Bibliografía.....	856

LISTA DE ANEXOS

	Pág.
ANEXO A. Tabla de precios de los elementos usados con el módulo entrenador.....	87
ANEXO B. Participantes en el taller realizado en el curso de sistemas digitales microcontrolados.....	88
ANEXO C. Encuesta realizada durante el taller.....	89
ANEXO D. Manual de prácticas módulo entrenador pic16f84a y esp32.....	90

RESUMEN

Este trabajo presenta el diseño y construcción de un Módulo Entrenador de bajo costo para el laboratorio del curso de Sistemas Digitales Microcontrolados. El enfoque principal del proyecto es la utilización de un microcontrolador tipo PIC de Microchip, el ESP32, que amplía la compatibilidad con el PIC 16F84A. El objetivo es proporcionar a los estudiantes una plataforma versátil y accesible para realizar prácticas de laboratorio, incluso aquellas que sean más largas o complejas.

El Módulo Entrenador diseñado ofrece soporte para diversos elementos comúnmente utilizados con microprocesadores, como potenciómetros, LEDs, suiches, pulsadores, interfaz serial, bocinas, pantallas OLED, conversores analógico-digital y display 7 segmentos. Estos componentes permiten realizar montajes y pruebas de laboratorio de manera eficiente.

Para validar su funcionamiento, el Módulo Entrenador se prueba con ambos microcontroladores, verificando todas sus funciones. Además, se ha desarrollado un manual de uso detallado que servirá como guía para los estudiantes que utilicen este recurso en su proceso de aprendizaje.

PALABRAS CLAVE: ESP32, MICROCONTROLADOR, CIRCUITO IMPRESO, MÓDULO ENTRENADOR, PIC16F84A, PANTALLA OLED.

ABSTRACT

This work presents the design and construction of a low-cost Trainer Module for the laboratory of the Microcontrolled Digital Systems course. The main focus of the project is the use of Microchip's PIC-type microcontroller, the ESP32, which extends compatibility with the PIC 16F84A. The goal is to provide students with a versatile and accessible platform to perform labs, even those that are longer or more complex.

The designed Trainer Module offers support for various elements commonly used with microprocessors, such as potentiometers, LEDs, switches, pushbuttons, serial interface, speakers, OLED screens, analog-digital converters, and 7-segment display. These components allow for efficient laboratory assembly and testing.

To validate its operation, the Trainer Module was tested with both microcontrollers, verifying all its functions. In addition, a detailed user manual has been developed that will serve as a guide for students who use this resource in their learning process.

KEYWORDS: ESP32, MICROCONTROLLER, PRINTED CIRCUIT, TRAINER MODULE, PIC16F84A, OLED SCREEN.

INTRODUCCIÓN

La realización de prácticas de laboratorio son fundamentales en el ámbito académico, ya que permite a los estudiantes aplicar los conocimientos teóricos adquiridos en un entorno práctico y real. En el campo de la Ingeniería Electrónica, estas prácticas son especialmente importantes, ya que implican el montaje y la interconexión de circuitos electrónicos.

Sin embargo, en ciertas materias como Sistemas Digitales Microcontrolados, donde la programación es fundamental, surge un desafío adicional. En lugar de centrarse en el montaje físico de los circuitos, se enfoca en la programación y aplicación de microcontroladores. Esto implica que el tiempo dedicado al montaje de los circuitos puede ser prolongado, lo que limita el tiempo disponible para el desarrollo de la programación y la exploración de diferentes aplicaciones.

En este contexto, es crucial contar con un Módulo Entrenador que permita a los estudiantes avanzar en la programación de microcontroladores de manera eficiente y sin contratiempos en la configuración física de los circuitos. Si bien existen tarjetas de desarrollo disponibles en el mercado, su costo puede ser alto y pueden no ofrecer todas las funcionalidades necesarias para el curso.

Por lo tanto, surge la pregunta: ¿Es posible disponer de un Módulo Entrenador económico que cumpla con los requisitos del curso de Sistemas Digitales Microcontrolados? Este proyecto tiene como objetivo responder a esta interrogante, diseñando y construyendo un Módulo Entrenador a bajo costo que amplíe las capacidades de un microcontrolador tipo PIC y el ESP32, que además permita realizar prácticas de manera más eficiente, tanto en términos de tiempo como de recursos.

OBJETIVOS

Objetivo general

Diseñar y construir un Módulo Entrenador de bajo costo para el laboratorio del curso de Sistemas Digitales Microcontrolados que permita realizar prototipos, sea económico y se pueda usar en las prácticas de laboratorio de microprocesadores o materias a fines.

Objetivos específicos.

- Determinar los componentes necesarios a usar en el Módulo Entrenador cumpliendo lo requerimientos de la academia.
- Diseñar el Módulo Entrenador para el microcontrolador utilizando los elementos y dispositivos a usar.
- Implementar el diseño del Módulo Entrenador en una PCB acoplado cada uno de los elementos.
- Documentar un manual de uso del Módulo, junto con un ejemplo de aplicación y manual guía para ponerlo al servicio de la universidad.

1. MARCO TEÓRICO Y ESTADO DEL ARTE

Esta sección tiene como objetivo presentar los resultados de las consultas realizadas con el fin de conocer los distintos trabajos, publicaciones, tesis y proyectos que involucran Módulos Entrenadores de microcontroladores. Para ello, se realizaron búsquedas en bases de datos e internet, obteniendo un total de cuatro trabajos destacados. Entre ellos, tres son de carácter nacional y uno internacional.

A nivel internacional, en Ecuador en la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL se realizó el proyecto de **Diseño de tarjetas para el diseño de aplicaciones con dsPIC**, el cual fue el diseño y construcción de una tarjeta de entrenamiento y otra de aplicaciones básicas para el desarrollo de diferentes aplicaciones con dsPIC, como resultado diseñaron una tarjeta de entrenamiento en donde se aprovechó buena parte de los recursos que ofrece el dsPIC30F4011.(Barcia et al., 2008)

En el ámbito nacional e internamente, la Universidad Pontificia Bolivariana en la sede de Bucaramanga se realizó el **DISEÑO Y FABRICACION DE UN MÓDULO ENTRENADOR IMPLEMENTANDO LA FAMILIA DE MICROCONTROLADORES dsPIC33F**; el cual es una herramienta de desarrollo, que ayuda al estudiante en el proceso de aprendizaje de la familia de microcontroladores dsPIC33F, y ofrece múltiples opciones al momento de diseñar un ejercicio, ejemplo o aplicación, donde se implementan con algunos elementos como: diodos emisores de luz (LEDs), display de cristal líquido (LCD), pulsadores, dip-switch, display siete segmentos, memoria en una microSD, puerto serial, módulo bluetooth, potenciómetro y borneras de entrada y salida para tratar señales analógicas. Además la tarjeta cuenta con un programador para que el usuario tenga la facilidad de comprobar los códigos de programación de cada una de las prácticas.(Gonzales Rojas & Caviativa Mendez, 2011)

A nivel nacional en Medellín, se realizó una tesis en la FACULTAD DE EDUCACIÓN VIRTUAL Y A DISTANCIA de la corporación UNIMINUTO DE DIOS, llamada **DISEÑO E IMPLEMENTACIÓN DE UN KIT PARA ROBÓTICA EDUCATIVA** con la intención de que el kit inducirá a los estudiantes a desarrollar su creatividad, imaginación y su capacidad de investigación en diferentes áreas como la electrónica, la lógica de programación y la mecánica, existe la idea de que la implementación de la robótica es demasiado costosa, y los costos disminuyen considerablemente, cuando se puede programar nuevamente el microcontrolador de acuerdo a la creatividad de cada persona.(Arroyave Zuluaga & Mosquera Rentería, 2015)

En otra investigación a nivel nacional en la FACULTAD DE INGENIERÍA de la Universidad Autónoma de Occidente de Cali, se realizó el **DESARROLLO DE UN KIT MODULAR DE BAJO COSTO, QUE FACILITE LA REALIZACIÓN DE PRÁCTICAS DE FÍSICA EN COLEGIOS**, donde presenta el desarrollo de un kit de bajo costo para el desarrollo de experimentos de física para colegios de bajos

recursos económicos, lograron como resultado el desarrollo de un kit modular de bajos costo (UAOLab), que facilita la realización de prácticas de física en los colegios. (Sanchez Naranjo, 2020)

1.1 Marco conceptual.

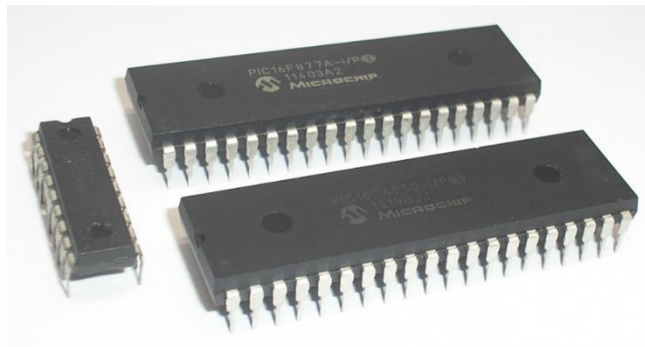
En esta sección se presentan los conceptos necesarios para comprender el proyecto, desde los más básicos hasta más complejos.

1.1.1 Microcontrolador

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida como se observa en la *Figura 1*. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información. (Hernandez, 2021)

Figura 1.

Algunos microcontroladores de la familia Microchip



Nota. Tomado de ¿Qué es un microcontrolador? [Fotografía], por Sherlin Hernández, 2021, <http://sherlin.xbot.es/microcontroladores/>

1.1.1.1 Partes de un Microcontrolador

Un microcontrolador se compone de varias partes, como lo son un procesador incluye al menos tres elementos, ALU, unidad de control y registros.

- ALU. También conocida como Unidad Aritmética y Lógica. Esta unidad está compuesta por los circuitos electrónicos digitales del tipo combinatorios (compuertas, sumadores, multiplicadores), cuya principal función es el realizar operaciones.
- Unidad de control. La unidad de control es el conjunto de sistemas digitales secuenciales (aquellos que tienen memoria) que permiten distribuir la lógica de las señales.

- Registros. Los registros son las memorias principales de los procesadores, ya que funcionan a la misma velocidad que el procesador a diferencia de otras memorias un tanto más lentas (como la RAM, FLASH o la CACHE). Los registros están contruidos por Flip-Flops. Los Flip-Flops son circuitos digitales secuenciales internos en un microcontrolador. (Marmolejo, 2017)

1.1.1.2 ¿Qué se puede hacer con un microcontrolador?

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas, el cual puede usarse para muchas aplicaciones algunas de ellas son: manejo de sensores, controladores, juegos, calculadoras, agendas, avisos lumínicos, secuenciador de luces, cerrojos electrónicos, control de motores, relojes, alarmas, robots, entre otros. El límite es la imaginación. (Hernández, 2021)

1.1.2 Microcontrolador PIC16F84A

El PIC16F84A es un microcontrolador gama media de 8 bits que físicamente consta de 18 pines, permite grabarse hasta unas 10000 veces debido a que su memoria es de tipo flash, trabaja de 2 a 5.5v y por terminal maneja 25mah de corriente, a continuación, se presentan algunas características. (Microchip, 2023)

1.1.2.1 Características

Set de 35 instrucciones

Memoria de programa de 1 KB (equivale a 1024 instrucciones)

Máxima velocidad de operación: 20 MHz

68 Bytes de RAM

64 Bytes de EEPROM

4 fuentes de interrupción

2 puertos de salida

13 Líneas de I/O configurables individualmente

25 mA de corriente por pin

1.1.3 ¿Qué se necesita para grabar un programa en un microcontrolador?

Se necesita básicamente tres cosas:

1. Una computadora
2. Software de programación (incluyendo un compilador).
3. Un circuito programador (ver Figura 2).

Y obviamente también se necesita un microcontrolador

La computadora, a través del software de grabación, se encarga de enviar el programa que se desea grabar al microcontrolador por intermedio del circuito programador. (Marmolejo, 2017)

Figura 2.

Circuito programador de Microcontroladores tipo PIC



Nota. Adaptado de Adaptador de programación y sistema de desarrollo para PIC, 2021, tomado de https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/microchip2016-02-07-20-36-39_/

1.1.4 PCB o circuito impreso

PCB son las siglas de Placa de Circuito Impreso, pero se utilizan las siglas en inglés (Printed Circuit Board) para no confundirla por ejemplo con las ranuras PCI de nuestro PC, en electrónica una superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora, pues una PCB básicamente es un soporte físico en donde se instalan componentes electrónicos y eléctricos y se interconectan entre ellos. Estos componentes pueden ser, chips, condensadores, diodos, resistencias, conectores, etc. Si echas un vistazo a un ordenador por dentro, verás que hay múltiples placas planas con un montón de componentes pegados a ella, se trata de una placa base y está compuesta por una PCB y los componentes que hemos citado

Para conectar cada elemento en una PCB utilizamos una serie de pistas conductoras de cobre extremadamente finas y que general un carril, conductor, como si de un cable se tratase. En los circuitos más sencillos, solamente tenemos pistas conductoras en una cara o las dos visibles de la PCB, pero en otros más completos tenemos pistas eléctricas e incluso componentes apilados en múltiples capas de ellas.

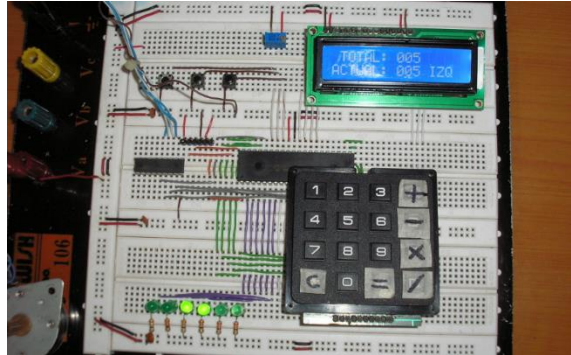
El soporte principal para estas pistas y componentes es una combinación de fibra de vidrio reforzada con materiales cerámicos, resinas, plástico y otros elementos no conductores. Aunque actualmente se están utilizando componentes como celuloide y pistas de pintura conductora para fabricar PCB flexibles. (Castillo, 2019)

1.1.4.1 Diseño de la PCB mediante software

Todo comienza con el diseño de la PCB desde la protoboard, como se observa en la *Figura 3*, de aquí todo lo montado se realiza en un software el esquemático trazando digitalmente las pistas eléctricas necesarias para conectar los componentes, así como enumerar cuántas capas van a ser necesarias para poder generar todas las conexiones que van a ser necesarias para los componentes.

Figura 3.

Montaje de un circuito en una protoboard



Nota. Tomado de ¿Qué es un microcontrolador? [Fotografía], por Sherlin Hernández, 2021, <http://sherlin.xbot.es/microcontroladores/>

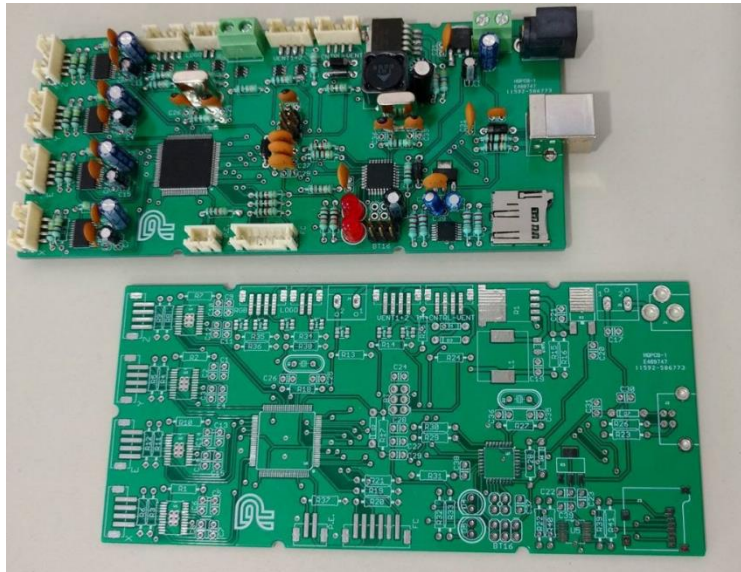
Este proceso se realiza mediante software de ordenador CAM como por ejemplo TinyCAD o DesignSpark PCB, Eagle, muy usado en las carreras de ingeniería. No solamente se diseñan las pistas eléctricas, sino que también se crean las distintas etiquetas para enumerar componentes instalados e identificar cada conector. (Castillo, 2019)

Se documentan todos los pasos necesarios del proceso de elaboración para que el fabricante sepa exactamente qué debe hacer cuando se le envíe el proyecto.

Después se procede a perforar la placa virgen, colocando y soldando cada uno de los elementos, al final el diseño queda como se puede observar en la siguiente figura.

Figura 4.

Circuito impreso con sus elementos



Nota. Adaptada de Lección #1: Tarjeta de Circuito Impreso [Fotografía], por Doñate Raul, 2018, <https://pcbcentral.com/tarjeta-de-circuito-impreso-definiciones-y-terminologia-utilizada-parte-i>

1.1.5. Kit o tarjeta de desarrollo

Es una placa o circuito que contiene un microcontrolador principal que corre o ejecuta una serie de instrucciones de un programa suministrado. Alrededor de este procesador o unidad principal se ha creado un diseño electrónico que permite: la programación del componente, suministra el voltaje adecuado para el correcto funcionamiento del controlador y proporciona acceso a las entradas y salidas del microcontrolador para la conexión de sensores y actuadores. (Concepción, 2019)

La mayoría de las aplicaciones requieren más que simplemente agregar un sensor a un objeto físico. Cuando se habla de 'objetos inteligentes', generalmente se habla de la adición de un microcontrolador conectado a Internet (también conocido como MCU).

Las MCU tienen menos capacidad que un procesador de computadora estándar, su bajo costo las convierte en una opción más práctica para agregar capacidades de computación a un objeto, espacio o proceso que no las tiene (Lemus, 2019).

Hay muchos MCU para el desarrollo y microcontroladores disponibles de una variedad de compañías: TI, Samsung, Arduino, Raspberry Pi y más. Elegir cuál es el adecuado depende de una serie de factores que varían según la naturaleza de la aplicación.

Las MCU comúnmente van junto con lo que se conoce como una "placa de desarrollo". Las placas de desarrollo son un punto de partida para todos los

proyectos integrados mediante el desarrollo de una prueba rápida de concepto antes de migrar a un diseño real. Una placa de desarrollo proporciona todo lo necesario para programar la MCU; incluyen periféricos como una fuente de alimentación, soporte para conectar sensores y, a veces, incluso sensores y actuadores a bordo. También permiten a los usuarios conectar rápidamente sensores y actuadores (si no están ya incluidos en la placa) y el software que los acompaña facilita la creación y el despliegue del código, normalmente una tarjeta de desarrollo es donde se trabaja primeramente el microcontrolador con los elementos para después consecuentemente implementarla en una PCB, algunas placas y kits de desarrollo en el mercado son:

1.1.5.1. Arduino

El proyecto Arduino inició en 2005 y es sin duda la tarjeta más conocida a nivel mundial. Este proyecto es hardware libre, esto quiere decir que ellos comparten el diagrama de las tarjetas para que cualquiera pueda construirlas. La mayoría de las placas están diseñadas con un microcontrolador AVR Atmel-8 bits, en muchos casos el Atmega328. En la *Figura 5* se presenta una placa de desarrollo Arduino uno (Lemus, 2019).

Figura 5.

Placa de desarrollo Arduino uno



Nota. Adaptada de ¿Qué es una tarjeta de desarrollo microcontrolador para IOT? [Figura] por Isaac Lemus, 2019, <https://conocimientolibre.mx/microcontrolador/>

1.1.5.2. ESP32

Una de las innovaciones más importante en el ámbito de las tarjetas de desarrollo fue la aparición del ESP32, el cual se observa en la *Figura 6*. Este microcontrolador de la marca ESPRESSIF fue uno de los primeros en permitir la conexión a una red wifi, lo que impulsó el internet de las cosas. Su bajo costo y diminuto tamaño permite la realización de proyectos con la posibilidad de conectarse a una red Wifi y de este modo compartir información a través de Internet. (Lemus, 2019)

Figura 6.
Tarjeta ESP32



Nota. Adaptada de ¿Qué es una tarjeta de desarrollo microcontrolador para IOT? [Figura] por Isaac Lemus, 2019, <https://conocimientolibre.mx/microcontrolador/>

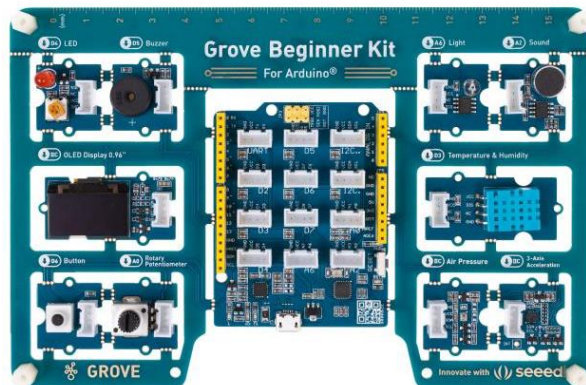
1.1.5.3. Kit Grove Beginner para Arduino

Es un kit de desarrollo basado en el Microcontrolador: ATmega320p (Compatible con Arduino™) se observa en la *Figura 7*, cuenta con 10 de los sensores más comunes; los sensores que traen conector Grove lo que permite retirarlos de la tarjeta principal y usarlos de manera individual, los sensores que trae son:

Elementos y sensores (Seeedstudio,2022)

- LED
- Buzzer
- Display OLED 0.96" (SSD1315)
- Pulsador
- Potenciómetro rotativo
- Sensor de sonido
- Sensor de luz
- Sensor de humedad y temperatura (DHT11)
- Sensor acelerómetro de 3 ejes (LIS3DHTR)
- Sensor de presión (BMP280)
- Tiene un costo de \$193.000 pesos colombianos en el mercado

Figura 7.
Kit Grove Beginner para Arduino



Nota: Adaptada de Grove Beginner Kit for Arduino - All-in-one [Figura] , 2022, <https://www.seeedstudio.com/Grove-Beginner-Kit-for-Arduino-p-4549.html>

1.1.5.4. Sistema de desarrollo para microcontroladores SDM_R5.

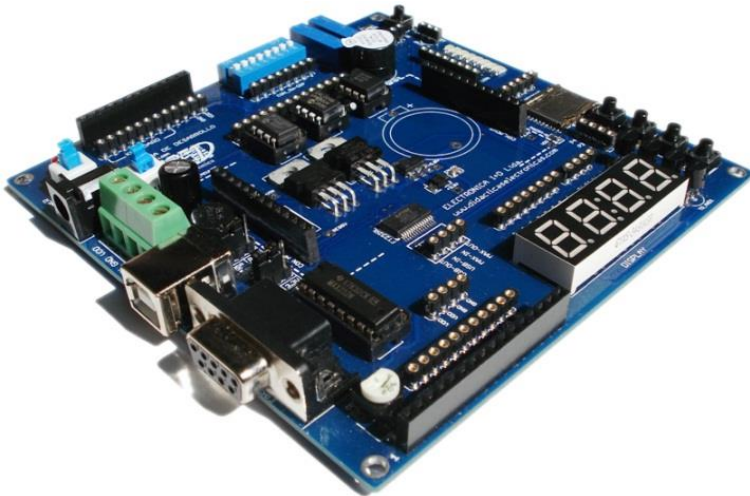
El sistema de desarrollo para microcontroladores R5 de Microchip, Freescale y otras marcas provee un set de elementos que permiten probar la mayoría de los módulos de un microcontrolador. En la *Figura 8* se presenta la vista superior del Kit de desarrollo SDM_R5 (Electrónica I+D, 2021).

La tarjeta principal con todos los elementos soldados y entre sus sensores incluye:

- Pulsadores, suiches, Buzzer
- LEDs, sensor de luz
- Displays, conector para LCD
- Potenciómetros, potenciómetro digital
- RTC, EEPROM
- Comunicación Serial, comunicación USB, conversor USB a Serial
- Conector para teclado matricial
- Base para memoria micro SD

Figura 8.

Kit de desarrollo SDM_R5



Nota: Adaptada de Sistema de desarrollo para microcontroladores, [Figura], 2021, <https://didacticasselectronicas.com/index.php/sistemas-de-desarrollo/otros-sistemas/tarjetas-desarrollo/sistema-de-desarrollo-para-microcontroladores-detail>

1.1.5.5. Sistema de desarrollo Renesas RL78G13

Sistema de desarrollo y depuración con múltiples sensores y periféricos para la serie de microcontroladores de bajo voltaje Renesas RL78G13 de 32MHz y 41 DMIPS, ROM flash 64kB, RAM 4K. En la *Figura 9* se presenta la vista superior del sistema de desarrollo Renesas RL78G13. Algunos de los elementos y sensores son los siguientes:

- LCD gráfico de 96x64 puntos
- 3 pulsadores para empleo del usuario y 1 de RESET
- 10 LED's (6 disponibles para empleo del usuario)
- Lector de tarjetas de memoria microSD
- Memoria serial EEPROM
- Acelerómetro digital de 3 ejes
- Sensor de temperatura lineal digital
- Sensor de luz ambiental digital
- Sensor receptor infrarrojo universal
- LED emisor infrarrojo

Figura 9.
Renesas RL78/G13



Nota: Adaptada de Renesas RL78/G13, [Figura], 2022,
<https://www.electronicoscaldas.com/es/sistemas-de-desarrollo-para-renesas/168-sistema-desarrollo-renesas-rl78-g13-yrdkrl78g13.html>

1.1.5.6. Placa de desarrollo Explorer 16/32

La placa de desarrollo Explorer 16/32 es un sistema de desarrollo flexible para microcontroladores PIC® de 16 bits y 32 bits de Microchip (Ver *Figura 10*). Es una versión fresca y económica de la placa de desarrollo Explorer 16 y soporta dispositivos de las familias de MCU PIC24 y PIC32 y las familias dsPIC® DSC como módulos plug-in de procesador (PIM), que permiten al usuario desarrollar rápidamente aplicaciones diversas. (Digikey, s. f.)

Figura 10.
Placa de desarrollo Explorer16/32



Nota. Adaptada de Herramientas de desarrollo Explorer 16/32 [Figura] por Microchip, 2021, <https://www.digikey.com/es/product-highlight/m/microchip-technology/explorer-16-32-development-board>

1.1.5.7. Otras tarjetas de desarrollo

Tarjeta FPGA diseñada por Xilinx Spartan 3E, con chip XC3S500E y varias interfaces integradas. Soporta tarjetas accesorias de expansión, consiste de una tarjeta DVK601 y un núcleo FPGA EP4CE6 (ver Figura 11).

Características:

- Con 8 diferentes puertos para conectar tarjetas o módulos de expansión
- Conector de expansión FPGA
- Con interfaz LCD
- Contiene Buzzer, joystick, potenciómetro y conector Jack de 5V.

Figura 11.

Tarjeta de desarrollo Xilinx Open3S500E



Nota: Adaptada de Tarjeta de desarrollo Xilinx Open3S500E, [Figura], <https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/cpld-y-fpga/tarjeta-de-desarrollo-xilinx-open3s500e-tarjetas-sistemas-boards-de-desarrollo-fpga-open3s500e-xilinx-spartan-waveshare-detail>

1.1.6. Dispositivos y elementos usados con los microcontroladores

En el ámbito de la electrónica y la programación, los microcontroladores son dispositivos clave para el desarrollo de diversos proyectos. Existen una gran variedad de microcontroladores, cada uno con características y funcionalidades distintas que se adaptan a distintas necesidades. En este sentido, es importante

conocer cuáles son los dispositivos y elementos más utilizados en la actualidad con microcontroladores, ya que esto permitirá elegir el que mejor se adapte a las necesidades del proyecto en cuestión. A continuación, se presentan algunos de los microcontroladores y elementos más utilizados en la actualidad en el mundo de la electrónica y la programación.

1.1.6.1. LED

Un LED (Light Emitting Diode, por sus siglas en inglés) es un dispositivo semiconductor que convierte la energía eléctrica en luz visible. Los LED están diseñados para emitir luz en un rango de longitudes de onda estrechas y específicas, lo que les permite ser utilizados para aplicaciones de iluminación y señalización de alta eficiencia. Los materiales utilizados en la fabricación de LED pueden ser de diferentes tipos, incluyendo arseniuro de galio (GaAs), fosfuro de aluminio (AlP), nitruro de galio (GaN), entre otros. El proceso de emisión de luz en un LED se produce cuando los electrones y los huecos cargados positivamente se recombinan en la zona de unión del material semiconductor. A medida que los electrones se mueven de una banda de energía superior a una inferior, liberan energía en forma de fotones que emiten luz visible. La emisión de luz del LED es muy eficiente y tiene una larga vida útil en comparación con otras fuentes de luz (Ng, K. K. 2007).

1.1.6.2. Suiche

Un switch o suiche en electrónica es un dispositivo que se utiliza para abrir o cerrar un circuito eléctrico, permitiendo o interrumpiendo el flujo de corriente eléctrica en el mismo. Es un componente fundamental en la mayoría de los circuitos electrónicos, y existen diferentes tipos de switches que se utilizan en función de las necesidades específicas del circuito en cuestión.

Entre los tipos de switches más comunes se encuentran los interruptores de pulsador o pulsadores, los interruptores de palanca, los interruptores de deslizamiento, los interruptores de tipo push-pull, los interruptores de posición, los interruptores rotativos, entre otros.

La elección del tipo de switch a utilizar dependerá de factores como la corriente y voltaje que se manejen en el circuito, la durabilidad necesaria, el tipo de accionamiento (manual, mecánico, eléctrico), entre otros (Smith, 2004).

1.1.6.3. Pulsador

Un pulsador o push button es un tipo de interruptor momentáneo que se utiliza para realizar acciones en un circuito eléctrico. Estos pulsadores se caracterizan por su diseño compacto y su capacidad de ser presionados temporalmente para enviar una señal eléctrica al circuito.

Los pulsadores se utilizan en una amplia variedad de aplicaciones electrónicas, como en sistemas de control, dispositivos de entrada para ordenadores y dispositivos móviles, sistemas de seguridad, entre otros. A menudo se colocan en una placa de circuito impreso y se conectan a través de contactos eléctricos.

Los pulsadores pueden estar diseñados para realizar diferentes acciones, como encender o apagar un dispositivo, cambiar de modo o seleccionar opciones, entre otros. Los pulsadores pueden tener una única posición o varias posiciones, y pueden tener diferentes diseños, como pulsadores, pulsadores de tipo basculante, pulsadores de tipo momentáneo, entre otros (Horowitz, 2015).

1.1.6.4. Buzzer

Un buzzer, también conocido como zumbador o generador de tonos, es un dispositivo electromecánico que produce sonidos audibles a través de vibraciones mecánicas o electrónicas. En electrónica, los buzzers se utilizan comúnmente para indicar una alarma o señalización, como una alerta de advertencia o una indicación de confirmación.

Los buzzers piezoeléctricos utilizan un cristal piezoeléctrico para producir vibraciones mecánicas cuando se aplica una corriente eléctrica, mientras que los buzzers electromagnéticos utilizan un electroimán para producir un campo magnético que hace vibrar una membrana o una placa metálica (Murata).

Los buzzers se pueden controlar a través de un microcontrolador utilizando un puerto de salida digital y una señal de onda cuadrada, que se genera en el software del microcontrolador. La frecuencia de la señal determina la frecuencia del tono producido por el buzzer, mientras que la duración de la señal determina la duración del tono.

En resumen, los buzzers son dispositivos electromecánicos que se utilizan en electrónica para producir sonidos audibles para indicar una alarma o señalización (Jameco, Electromagnetic Buzzer).

1.1.6.5. Potenciómetro

Un potenciómetro es un dispositivo electrónico pasivo que se utiliza para controlar el flujo de corriente en un circuito eléctrico. Consiste en un resistor de tres terminales que se puede ajustar para variar la resistencia eléctrica en un rango determinado. Al girar el eje central del potenciómetro, se puede ajustar el valor de resistencia entre dos de los terminales, mientras que el tercer terminal permanece conectado en todo momento. Los potenciómetros se utilizan comúnmente para ajustar el volumen en sistemas de audio, para controlar la intensidad de luz en pantallas, en sistemas de control de motores, y en otras aplicaciones donde se requiere ajustar una resistencia eléctrica (Nashelsky, 2008).

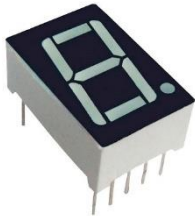
1.1.6.6. Display 7 segmentos

El display de 7 segmentos es un componente electrónico utilizado para mostrar números, letras y algunos símbolos en dispositivos electrónicos. Este dispositivo, el cual se muestra en la *Figura 12*, consiste en siete segmentos iluminados que se combinan para formar caracteres alfanuméricos. Los segmentos están dispuestos en una configuración común de ánodo o cátodo, lo que significa que todos los ánodos o cátodos de los segmentos se conectan entre sí, respectivamente. Para mostrar un número o letra, se activan los segmentos correspondientes, y para mostrar varios caracteres, se utilizan varios displays. Estos displays pueden ser controlados por microcontroladores, circuitos integrados específicos o lógica

combinacional, lo que los hace muy versátiles y fáciles de integrar en diferentes aplicaciones electrónicas. Los displays de 7 segmentos se utilizan comúnmente en dispositivos electrónicos como relojes, termostatos, medidores de voltaje, contadores, entre otros. Estos displays son económicos y están disponibles en diferentes tamaños y colores, lo que los hace populares en muchas aplicaciones.

Figura 12.

Display 7 segmentos



Nota: Adaptada de Display 7 Segmentos 0.56", [Figura], 2022, <https://ferretronica.com/products/display-7-segmentos-0-56-pulgadas-anodo-comun-color-rojo>

1.1.6.7. Pantalla LCD

Una pantalla LCD (Liquid Crystal Display, por sus siglas en inglés) es un dispositivo de visualización utilizado en muchos dispositivos electrónicos para mostrar información de texto o gráficos. La pantalla, que se muestra en la *Figura 13*, está compuesta por una capa de cristal líquido que cambia su polarización cuando se aplica una corriente eléctrica. Esta capa se encuentra entre dos placas polarizadas y se controla mediante una red de electrodos para producir los patrones de visualización deseados.

La mayoría de las pantallas LCD están retroiluminadas con LED para mejorar la legibilidad en condiciones de poca luz. Los controladores LCD integrados en los microcontroladores o circuitos integrados permiten la programación de la visualización de texto y gráficos. La resolución y el tamaño de la pantalla varían según el dispositivo y la aplicación (Gagné, 2002).

Las pantallas LCD se utilizan en una amplia gama de dispositivos electrónicos, desde relojes y calculadoras hasta teléfonos móviles y televisores. También se utilizan en proyectos de electrónica, robótica y microcontroladores para mostrar información y proporcionar interacción con el usuario.

En resumen, una pantalla LCD es un dispositivo visual que utiliza cristales líquidos polarizados para mostrar información de texto o gráficos en una amplia gama de dispositivos electrónicos y proyectos de electrónica.

Figura 13.
Pantalla LCD



Nota: Adaptada de Pantalla LCD 2x16, [Figura], 2022,
<https://tdelectronica.com/producto/pantallas-y-lcd/lcd-2x16/>

1.1.6.8. Pantalla OLED

Una pantalla OLED (Organic Light Emitting Diode) es un tipo de pantalla que utiliza una capa de materiales orgánicos que emiten luz cuando se aplica una corriente eléctrica (ver *Figura 14*). En contraste con las pantallas LCD que utilizan luz de fondo para iluminar los píxeles, los OLED emiten luz directamente, lo que resulta en una imagen con colores más intensos, contrastes más altos y un menor consumo de energía.

Cada píxel de una pantalla OLED es una pequeña celda que emite luz. Al aplicar una corriente eléctrica a través de la celda, se emite luz en una variedad de colores. Los OLED se utilizan en una amplia gama de dispositivos electrónicos, como televisores, monitores, teléfonos inteligentes y reproductores de MP3.

Las pantallas OLED tienen varias ventajas sobre otros tipos de displays, como un tiempo de respuesta rápido, un bajo consumo de energía, una amplia gama de ángulos de visión y la capacidad de producir imágenes nítidas y brillantes. Sin embargo, también tienen algunas desventajas, como una vida útil más corta y un mayor costo en comparación con otros tipos de pantallas (Bhushan, 2014).

En resumen, una pantalla OLED es un tipo de display que utiliza materiales orgánicos que emiten luz cuando se aplica una corriente eléctrica y se utilizan en una amplia gama de dispositivos electrónicos.

Figura 14.
Pantalla OLED



Nota: Adaptada de Pantalla OLED 128X64, [Figura],
<https://www.mactronica.com.co/lcd-oled-128x64-096-pulgadas-soporte>

1.1.6.9. Sensor de temperatura

Un sensor de temperatura es un dispositivo que mide la temperatura de su entorno y produce una señal eléctrica proporcional a dicha temperatura. Estos sensores utilizan diferentes principios físicos, como la variación de la resistencia, la tensión o la frecuencia en respuesta a los cambios de temperatura. Son ampliamente utilizados en una variedad de aplicaciones, desde el monitoreo de la temperatura ambiente en edificios hasta el control de temperatura en sistemas de refrigeración y calefacción. Proporcionan información crucial para garantizar el funcionamiento adecuado de los dispositivos y procesos que dependen de una temperatura controlada.

1.2 Conclusión del capítulo.

En este capítulo, se han abordado los conceptos fundamentales relacionados con los componentes electrónicos esenciales, incluyendo los microcontroladores, las placas de circuito, los interruptores, los suiches, las pantallas OLED y los displays de 7 segmentos. Estos componentes desempeñan un papel crucial en el ámbito de la electrónica y son piezas fundamentales en la creación de sistemas y dispositivos electrónicos. El conocimiento adquirido sobre estos elementos individuales sienta las bases necesarias para comprender y abordar desarrollos posteriores en el presente estudio, permitiendo a los investigadores apreciar la versatilidad de estos componentes y la amplia gama de aplicaciones que pueden lograrse a través de su integración y manipulación en proyectos electrónicos.

2. METODOLOGÍA

En este capítulo se presenta el proceso en la elaboración del proyecto, el cual se divide en varias etapas, verificando el cumplimiento en cada una de ellas con el resultado esperado como se observa en la *Tabla 1*. Esta metodología permitirá llevar a cabo de manera sistemática y ordenada el diseño, construcción y verificación del Módulo Entrenador para el laboratorio de Sistemas Digitales Microcontrolados, asegurando la calidad y funcionalidad del kit desarrollado.

Tabla 1.
Objetivos y resultados esperados.

Objetivo Específico	Actividades	Resultado esperado
Determinar los componentes necesarios a usar en Módulo Entrenador cumpliendo los requerimientos de la academia.	1. Seleccionar el microcontrolador a usar.	Hallar un microcontrolador económico y con muchas funcionalidades.
	2. Investigar qué componentes son de vital importancia para un microprocesador teniendo en cuenta el plan del curso Sistemas digitales microprocesadores y el estado del arte.	Obtener una lista completa de los componentes esenciales para el funcionamiento del Módulo Entrenador.
	3. Establecer la cantidad de dispositivos a usar con el Módulo Entrenador.	Definir la cantidad precisa de elementos necesarios para garantizar una experiencia de aprendizaje adecuada.
Diseñar el Módulo Entrenador para el microcontrolador utilizando los elementos y dispositivos a usar.	1. Realizar las conexiones de cada elemento con el microcontrolador en una board.	Establecer las conexiones físicas entre los componentes y el microcontrolador en una placa de pruebas.
	2. Probar la funcionalidad de cada elemento mediante el microcontrolador observando físicamente.	Verificar que cada elemento funcione correctamente al interactuar con el microcontrolador, realizando pruebas y observando los resultados.
	3. Realizar un esquemático de las conexiones con los elementos de acuerdo a las librerías	Crear un esquemático detallado que muestre las conexiones eléctricas entre los elementos y el microcontrolador, siguiendo las especificaciones de las librerías utilizadas.
Implementar el diseño del Módulo Entrenador en una PCB acoplado cada uno de los elementos.	1. Las conexiones realizadas en la board pasarla a un diseño en Eagle	Realizar las conexiones previamente establecidas en la placa de pruebas a un diseño digital utilizando el software Eagle.
	2. Acoplar a la PCB el PIC 16f84 y comprobar el funcionamiento.	Conectar el microcontrolador en la placa de circuito impreso y obtener un correcto funcionamiento.

	3. Imprimir en una tarjeta PCB el diseño	Generar una placa de circuito impreso a partir del diseño y montar todos los componentes en la PCB final con un correcto funcionamiento
Validar el uso del kit, junto con un ejemplo de aplicación y manual guía para ponerlo al servicio de la universidad.	1. Realizar un taller donde se usen varios elementos del micro y se observe el funcionamiento del kit	Comprobar el funcionamiento de los elementos con los estudiantes y recibir una buena aceptación del Módulo Entrenador
	2. Realizar un manual de la conexiones y uso de cada módulo del kit	Obtener un manual completo y detallado que explique las conexiones y el uso adecuado de cada uno de los módulos presentes en el Módulo Entrenador.

2.1. Determinar los componentes necesarios a usar en el Módulo Entrenador cumpliendo lo requerimientos de la academia:

Esta sección describe como fueron la selección de elementos para el Módulo Entrenador, así como los elementos necesarios partiendo del estado del arte y consultas en la academia, partiendo de la elección del microcontrolador hasta los elementos necesarios para su correcto funcionamiento.

2.1.1. Etapa de elección del microcontrolador.

Tras un análisis detallado de los Kits de desarrollo en el estado del arte, se ha podido observar que estos han sido diseñados de manera que sean capaces de manejar una gran cantidad de periféricos, permitiendo al usuario adaptar el circuito a diferentes problemas y situaciones. Esto se logra a través de la programación del microcontrolador, la cual permite una amplia variedad de posibilidades en la solución de problemas electrónicos. De esta forma, se puede garantizar que el usuario tenga una amplia gama de herramientas a su disposición para la realización de proyectos y aplicaciones en el campo de la electrónica.

Se tiene que en el laboratorio de Microcontroladores se trabaja con el PIC16f84, pero este a veces es algo limitado en prestaciones, con lo cual es necesario un microcontrolador que tenga los suficientes pines y soporten una gran variedad de periféricos para realizar programas más completos, partiendo del estado del arte y marco teórico se observan que existen varios microcontroladores, y de diferentes tipos, en la siguiente tabla se observan las principales diferencias:

Tabla 2.

CARACTERÍSTICAS	ARDUINO UNO	ESP32	KL25Z
<i>Fabricante</i>	Arduino	Espressif Systems	NXP Semiconductors
<i>Arquitectura</i>	AVR	Xtensa Dual-Core 32-bit LX6	ARM Cortex-M0+
<i>Velocidad de reloj</i>	16 MHz	160/240 MHz	48 MHz

<i>Memoria flash</i>	32 KB	4 MB	128 KB
<i>SRAM</i>	2 KB	520 KB	16 KB
<i>EEPROM</i>	1 KB	N/A	N/A
<i>Número de pines digitales</i>	14	28	27
<i>Número de pines analógicos</i>	6	18	15
<i>Comunicación serial</i>	UART, SPI, I2C	UART, SPI, I2C, Bluetooth, Wi-Fi	UART, SPI, I2C, USB
<i>Comunicación inalámbrica</i>	N/A	Bluetooth, Wi-Fi	N/A
<i>Integración con IDE</i>	Arduino IDE	Arduino IDE, ESP-IDF	MCUXpresso IDE

Uno de los microcontroladores más accesibles en el mercado que cumple con los objetivos requeridos es el ESP32. Este microcontrolador se caracteriza por su amplia variedad de pines y conectividad inalámbrica, incluyendo Wi-Fi y Bluetooth. A diferencia de otros microcontroladores como Arduino, el ESP32 puede resultar menos común para algunos estudiantes, lo que implica que deberán realizar investigaciones adicionales para llevar a cabo ciertas configuraciones. Por otro lado, una de las ventajas del ESP32 es su capacidad para manejar múltiples tareas simultáneamente, gracias a su arquitectura de dos núcleos. Además, este microcontrolador tiene una memoria Flash integrada y soporte para almacenamiento externo, lo que lo hace ideal para proyectos de Internet de las cosas y aplicaciones de bajo consumo de energía.

2.1.1.1. Elección de versión del ESP32

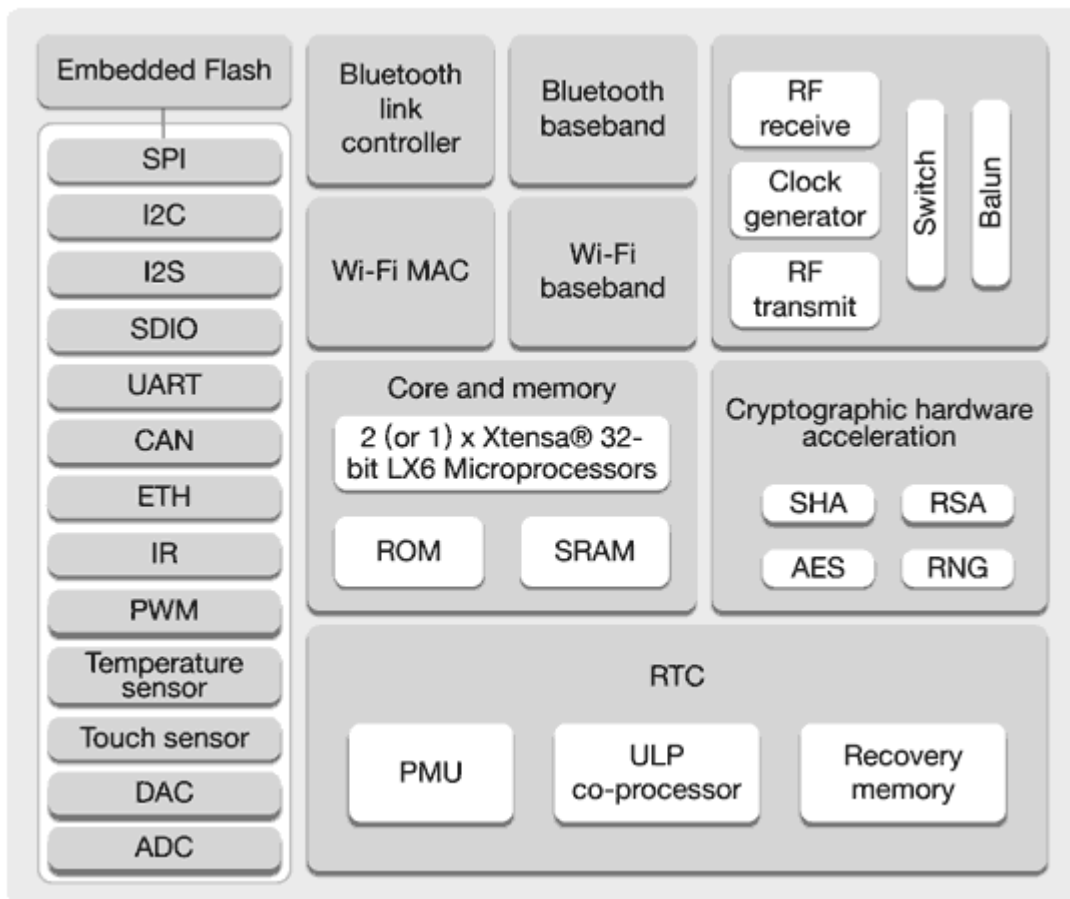
Se tiene que el ESP32 tiene varias versiones, una versión de 30 pines y una de 36 pines, se entró en consideración el uso de la versión de 36 pines pero tienen prácticamente los mismos pines de entradas y salidas, por lo tanto se opta por el ESP32 de 30 pines.

Considerando la cantidad de pines involucrados en esta instancia, específicamente 30, se han analizado exhaustivamente sus configuraciones y compatibilidades. Es destacable que el microcontrolador en cuestión ya cuenta con la integración de dos elementos fundamentales, a saber, el protocolo WiFi y Bluetooth. Este hecho convierte al dispositivo en una herramienta de gran utilidad para brindar soporte al desarrollo del Internet de las cosas (IoT).

Además de esto tiene un buen conjunto de instrucciones y métodos de comunicación como lo son el SPI, I2C, UART, entre otros, como se observa en la *Figura 15*.

Figura 15.

Instrucciones y métodos de comunicación del ESP32.



Nota: Adaptada de ESP32, el "hermano mayor" del ESP8266 con WiFi y Bluetooth, [Figura], 2018, <https://www.luisllamas.es/esp32/>

En este sentido, se puede observar que este microcontrolador posee diversas funcionalidades adicionales que lo hacen altamente completo. Entre estas, se destacan la modulación por ancho de pulso (PWM), la cual es una técnica que consiste en variar el ancho de un pulso de señal digital para controlar la cantidad de energía que se entrega a un dispositivo. Esta técnica es ampliamente utilizada en aplicaciones como el control de velocidad de motores eléctricos, el control de luminosidad en pantallas y en la regulación de voltaje en fuentes de alimentación. La capacidad de este microcontrolador para realizar esta función lo hace altamente versátil y adecuado para abordar una amplia gama de proyectos.

Por otro lado, los conversores analógico-digitales (ADC) y digitales-analógicos (DAC) son herramientas fundamentales en aplicaciones donde se requiere el procesamiento de señales analógicas. El ADC convierte la señal analógica en una señal digital, mientras que el DAC convierte la señal digital en una señal analógica. Estas funciones permiten el uso de sensores y actuadores analógicos en sistemas

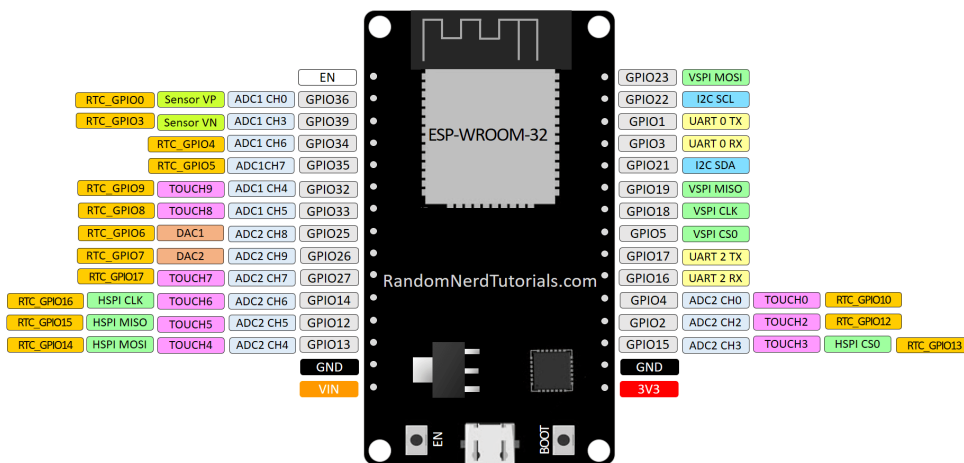
digitales, así como la generación de señales analógicas para controlar dispositivos de salida. La inclusión de estas funcionalidades en el microcontrolador evaluado lo convierte en una herramienta altamente adecuada para el desarrollo de proyectos que requieran la conversión de señales analógicas y digitales.

2.1.1.2. Cantidad de pines disponibles del ESP32

Para determinar la capacidad de pines para cada elemento, se ha considerado cuidadosamente la configuración de pines específica para cada uno de ellos. Esto se puede apreciar en la *Figura 16*, donde se detallan las conexiones y configuraciones de los pines del ESP32.

Figura 16.

Configuración de los pines del ESP32.



Nota: Adaptado de Módulo ESP32 - DEVKITV1, [FIGURA], 2022, <https://sumador.com/collections/radiocomunicaciones/products/modulo-esp32>

Por consiguiente, se observa que de los 30 pines disponibles en el microcontrolador solo dos de ellos corresponden a salidas de tipo DAC, los cuales deben ser utilizados obligatoriamente. Además, de acuerdo al análisis previo, se ha determinado que un total de 15 pines son requeridos para los elementos básicos del sistema.

Sin embargo, es importante tener en cuenta que para realizar prácticas y utilizar múltiples elementos como pulsadores, suiches y LEDs, se necesitarán más pines disponibles. Por tanto, del diagrama del ESP32 realizando un pequeño análisis se verifica que el total de pines usables son 25, y con estos es posible satisfacer las necesidades de conexión de todos los elementos en el proyecto.

2.1.2. Elección de módulos

Posteriormente se llevó a cabo una recopilación de información sobre los módulos necesarios mediante la revisión de prácticas previas realizadas en el curso, así como la identificación de los elementos utilizados en ellas. Para complementar esta

información, se consultó con los docentes encargados del curso y se recopiló información de fuentes bibliográficas tales como tesis, trabajos de grado y prácticas previas. También se investigaron los elementos más comúnmente utilizados en conjunción con los microcontroladores para asegurar que el diseño del sistema de desarrollo considerara las necesidades de los usuarios y cumpliera con los estándares de la industria, con lo cual se llegó a la conclusión de que los elementos más usados en las prácticas académicas son los Pulsadores, suiches, leds y buzzers.

Para comprobar la selección de los elementos a usar en el Módulo Entrenador, se consultan los kits de desarrollo comerciales actuales en el mercado, con lo cual, durante el análisis de los diferentes sistemas de desarrollo para microcontroladores, se encontró una coincidencia en el uso de algunos elementos, siendo los pulsadores, LEDS, pantallas OLED o LCD, buzzers y potenciómetros los más utilizados. Esto se puede observar en la *Tabla 3*, donde se muestra la presencia de estos componentes en cada uno de los sistemas de desarrollo evaluados. Es importante destacar que la presencia de estos elementos se debe a su importancia en la realización de pruebas y experimentos con microcontroladores, ya que permiten interactuar con el sistema y obtener información relevante de su funcionamiento. Además, se pudo identificar que estos elementos son comúnmente utilizados en los laboratorios de microcontroladores, por lo que su inclusión en los sistemas de desarrollo permite una mayor compatibilidad y familiaridad con los equipos utilizados en la educación y formación en este campo.

Tabla 3
Dispositivos usados en Módulos Entrenadores

Dispositivo	Renasas	SDM_R5	Groove	FPGA
LED	Sí	Sí	Sí	Sí
Pantalla	Sí	Sí	Sí	Sí
Pulsador	No	Sí	Sí	Sí
Switch	No	Sí	No	No
Lector de memoria	No	Sí	No	Sí
Buzzer	Sí	Sí	Sí	No
Serial	No	Sí	No	Sí
Acelerómetro	No	No	Sí	Sí
Sensor de temperatura	No	No	Sí	Sí
Sensor de luz ambiental	No	Sí	Sí	Sí
Infrarrojo	No	No	No	Sí
Potenciómetro	Si	Si	No	Sí

A partir del análisis de la tabla, se identificaron los elementos más utilizados en los sistemas de desarrollo, destacando entre ellos los leds, pulsadores, suiches, pantallas y buzzers. Estos elementos coinciden con aquellos empleados en los laboratorios de microcontroladores, lo que sugiere una convergencia en cuanto a la selección de componentes utilizados en ambas prácticas. Cabe resaltar que, además de los elementos básicos que se emplean en el desarrollo de las prácticas, los buzzers también son comúnmente implementados en los sistemas de desarrollo de microcontroladores, lo que sugiere su importancia en la implementación de este en el Módulo Entrenador para generación de señales audibles para diversos fines.

2.1.2.1. Elección entre la pantalla OLED vs LCD.

En la elección de la pantalla se había seleccionado inicialmente la pantalla LCD pero esta al necesitar muchos pines para su funcionamiento y al tener la pantalla oled como una tecnología nueva se analiza la inclusión de la pantalla OLED en vez de la LCD, con lo cual se hace el análisis en la siguiente tabla:

Tabla 4.

Diferencias entre las pantallas OLED y LCD.

Característica	Pantallas OLED	Pantallas LCD
Tecnología	Emplea LEDs orgánicos que emiten luz propia	Emplea cristales líquidos que requieren de una fuente de luz externa
Resolución	Mayor resolución debido a la individualidad de los píxeles	Resolución limitada por el número de píxeles
Consumo de energía	Consumo bajo de energía debido a que solo los píxeles encendidos consumen energía	Consumo alto de energía debido a que requiere de una fuente de luz externa
Programación	Más sencilla y directa debido a que se controla cada píxel individualmente	Requiere programación más compleja ya que se deben enviar instrucciones para mostrar caracteres o gráficos
Costo	Suele ser más costoso que las pantallas LCD	Las pantallas LCD son más económicas

Nota:

En resumen, la programación de pantallas OLED en microcontroladores es más sencilla y directa debido a que se controla cada píxel individualmente, lo que permite mayor flexibilidad en la visualización de información. Además, las pantallas OLED utilizan una tecnología más actual y tienen mayor resolución, aunque pueden ser más costosas que las pantallas LCD. Por otro lado, la programación de pantallas LCD requiere enviar instrucciones para mostrar caracteres o gráficos, lo que puede hacerla más compleja, y requieren de una fuente de luz externa, lo que aumenta su consumo de energía. Finalmente, se opta por la pantalla OLED como la definitiva debido a que el ESP32 es enfocado al bajo consumo de energía.

Con esto seleccionan los elementos que fueron comunes en todas las investigaciones y se analizan otros elementos que pueden complementar el Módulo Entrenador como lo son los los LED'S, Display 7 segmentos, pantalla OLED y como entradas pulsadores, suiches y potenciómetro, además se aprovechan los beneficios inalámbricos del microcontrolador usado, el cual cuenta con dos periféricos los cuales son el Wifi y Bluetooth, los más usados y los cuales se encuentran en la mayoría de los dispositivos actuales.

2.1.3. Etapa de estudio del lenguaje c y librerías soportadas por el ESP32

El ESP32 es soportado por el IDE Arduino, el cual cuenta con muchas librerías, en este caso para generar la compatibilidad con el software se debe agregar la librería VCP (Virtual COM Port), la cual como su nombre indica configura de manera virtual los puertos para poder comunicarse en este caso con la ESP32, al momento de instalarse se debe configurar y mediante este usar las mayorías de las librerías que se encuentran en Arduino, así con muchas similitudes las configuraciones de los pines e instancias.

Con lo cual se presentan las configuraciones básicas y conexiones que se usan con el ESP32, el cómo debe ser la conexión de cada elemento y su funcionamiento con el microcontrolador para su correcto uso, entre los cuales hay diferentes tipos de entradas o salidas.

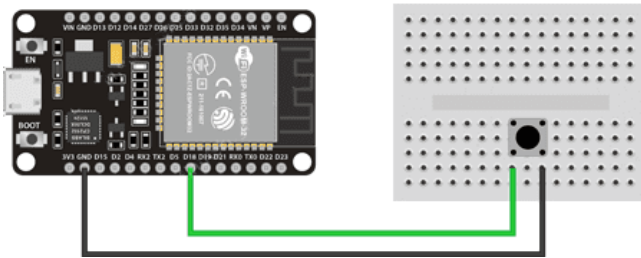
2.1.3.1. Entradas digitales

Las entradas digitales de un microcontrolador son pines que pueden detectar y leer señales digitales provenientes de dispositivos externos. Estas entradas se utilizan para recibir información o datos digitales, como pulsaciones de pulsadores, estados lógicos, señales de sensores, entre otros. El microcontrolador puede leer el estado de una entrada digital y utilizar esa información para tomar decisiones o realizar acciones en el programa, entre los dispositivos más usados como entradas digitales se encuentran los suiches y pulsadores, para leer el estado de estos se utiliza la técnica de lectura digital, donde se lee el valor lógico del pin de entrada.

En el caso de los suiches, estos se conectan a pines de entrada del microcontrolador y permiten interactuar con el sistema a través de señales eléctricas, manteniendo el estado seleccionado. Por otro lado, los pulsadores también se utilizan para enviar señales al microcontrolador. Cuando se presiona un pulsador, los contactos eléctricos se cierran temporalmente y permite que la corriente fluya hacia tierra. Al soltar el pulsador, los contactos vuelven a su posición original. La conexión del pulsador al microcontrolador se realiza conectando un pin del microcontrolador con el pulsador y conectando la otra terminal del pulsador a tierra, en la Figura 17 se observa la conexión de un pulsador, la cual es la misma configuración para los suiches.

Figura 17.

Conexión del microcontrolador con un pulsador

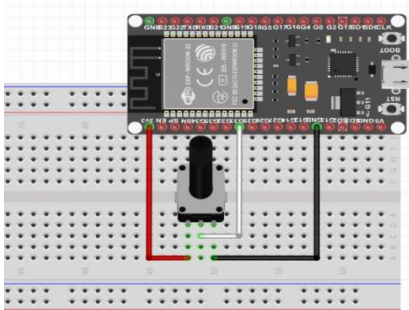


Nota: Adaptado de Configuración y manejo de las interrupciones del ESP32 GPIO en el IDE de Arduino, [Figura], 2022, <https://descubrearduino.com/interrupciones-esp32-gpio/>

2.1.3.2. Entradas analógicas

Las entradas analógicas de un microcontrolador permiten la lectura de señales analógicas provenientes de sensores u otros dispositivos. Estas entradas son capaces de medir y convertir señales continuas en valores digitales, como lo son los potenciómetros, los cuales se conectan a los microcontroladores mediante uno de sus terminales a la fuente de alimentación, otro terminal a la tierra del circuito y el tercer terminal al pin de entrada/salida del microcontrolador. El valor de la resistencia total varía según la posición del cursor, lo que produce una variación en el voltaje de entrada o salida del microcontrolador, los pines a usar con este son definidos como entradas analógicas y la conexión es como se observa en la *Figura 18*.

Figura 18.
Conexión del ESP32 con un Potenciómetro.



Nota: Adaptado de Básicos ESP32: Lectura de valor analógico [Figura], 2021, <https://tecnotizate.es/basicos-esp32-lectura-de-valor-analogico/>

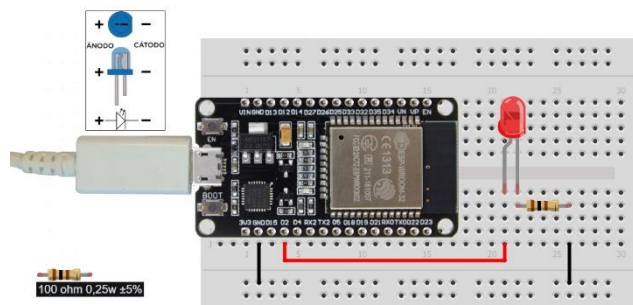
2.1.3.3. Salidas digitales.

Las salidas digitales de un microcontrolador son pines que pueden generar señales digitales de dos estados: alto (1) o bajo (0). Estas salidas se utilizan para controlar dispositivos externos, como LEDs, buzzers, actuadores, entre otros. Al establecer el estado de una salida digital, se puede encender o apagar un componente conectado, permitiendo el control y la interacción del microcontrolador con el entorno.

En el caso de los LEDs se conectan al microcontrolador a través de uno de sus pines digitales y puede ser encendido o apagado mediante el envío de una señal eléctrica al pin correspondiente. Además de su función como fuente de luz, los LED también pueden ser utilizados como indicadores de estado para mostrar si un proceso está en curso, si se ha completado con éxito, o si se ha producido algún error.

Para realizar la configuración de un LED mediante un microcontrolador la conexión se debe realizar como se observa en la *Figura 19*, donde el pin del microcontrolador se conecta al ánodo del LED, y al cátodo una resistencia que se recomienda mayor a 100ohm y no mayor a 1k, finalizando se realiza la conexión a tierra.

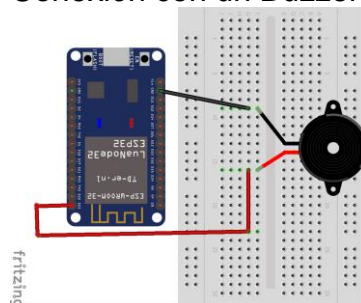
Figura 19.
Conexión del ESP32 con un LED



Nota: Adaptado de Micropython Esp32 [Figura] ,2019,
<https://www.esploradores.com/while/>

La configuración de un buzzer es de forma más sencilla, se deben conectar a un pin de salida digital o a un PWM del microcontrolador para modificar su tonalidad y producir sonidos audibles, la conexión se realiza como se observa en la *Figura 20*.

Figura 20.
Conexión con un Buzzer



Nota: Adaptado de zumbador con la tarjeta ESP32, [Figura], 2021,
<https://www.robotique.tech/robotics/control-a-buzzer-by-the-esp32-card/>

2.1.3.4. Elementos visualizadores

Los elementos visuales permiten transmitir información de manera visual, lo cual resulta especialmente útil en aplicaciones donde se requiere una representación gráfica o una visualización clara de datos, en esta mini sección se presentan los dos usados en el proyecto.

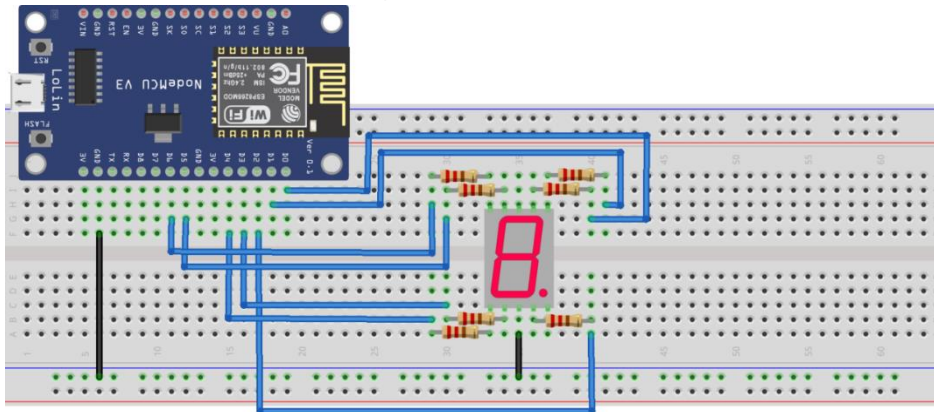
2.1.3.4.1. Display 7 segmentos

El funcionamiento del display 7 segmentos en microcontroladores es sencillo, se deben conectar los pines de cada segmento a los puertos de salida del microcontrolador y configurarlos como salidas digitales. Luego, se deben enviar los valores correspondientes para cada segmento según el número o letra que se desee mostrar.

Es posible utilizar múltiples displays 7 segmentos en cascada para mostrar información más compleja, como números decimales con punto flotante o texto. Para ello, se deben utilizar técnicas de multiplexado y controlar cada display individualmente. La conexión se realiza como se observa en la *Figura 21*.

Figura 21.

Conexión del display 7 segmentos



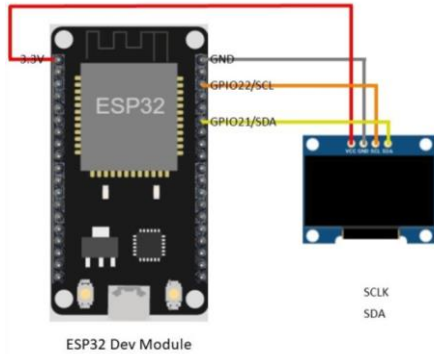
Nota: Adaptado de Display 7 Segmentos [Figura], 2020, <https://controlautomaticoeducacion.com/micropython/display-7-segmentos-pico/>

2.1.3.4.2. Pantalla OLED

En los microcontroladores, la programación de la pantalla OLED se realiza mediante el uso de librerías específicas para el modelo de pantalla y el microcontrolador utilizado. Además, la comunicación con el microcontrolador se lleva a cabo a través de interfaces de comunicación como I2C, con los cuales cuenta el microcontrolador elegido, con lo cual se deben elegir dos puertos que tengan SPI o I2C, su conexión se observa en la *Figura 22*.

Figura 22.

Conexión del ESP32 con la pantalla OLED.



Nota: Adaptado de MicroPython ESP32, [Figura], 2019, https://www.esploradores.com/oled_ssd1306/

El diagrama de conexiones muestra la cantidad de pines requeridos para establecer la conexión y garantizar el funcionamiento adecuado de cada elemento con el microcontrolador ESP32. Con el objetivo de simplificar y optimizar la configuración, se ha elaborado la *Tabla 5* que indica el número mínimo total de pines necesarios en el microcontrolador para la conexión de todos los elementos.

Tabla 5
Cantidad de pines por elemento.

ELEMENTO	PINES
LED	1
SWITCH	1
DISPLAY 7 SEGMENTOS	7
PANTALLA OLED	2
POTENCIOMETRO	1
PULSADORES	1
BUZZERS	1
Sensor de temperatura	1
TOTAL PINES	15

Nota: Cantidad de pines necesarios en el microcontrolador para el uso de cada elemento.

La tabla proporciona una visión general de la distribución de pines necesarios para cada componente, considerando factores como la comunicación, la alimentación y otros requisitos específicos de cada elemento. Al conocer el número mínimo total de pines requeridos, en este caso 15, se facilita la planificación y asignación de recursos del microcontrolador, lo que contribuye a una implementación más eficiente y organizada del sistema.

2.1.4. Cantidad de dispositivos.

En esta sección se establece la cantidad de elementos a utilizar en cada dispositivo, teniendo en cuenta la investigación del estado del arte y los usos comunes en

prácticas de laboratorio. A continuación, se presenta la cantidad preliminar de elementos seleccionados para cada componente:

- Suiches: Se ha observado en el mercado que existen suiches conocidos como dip suiches que vienen en grupos. En este caso, se ha seleccionado un suiche de 4 bits, lo que implica la utilización de cuatro pines.
- Pulsadores: Los pulsadores, por lo general, se presentan como elementos individuales. En este caso, se han elegido dos pulsadores para su integración en el proyecto.
- Entrada analógica: Para la entrada analógica, se ha optado por utilizar un potenciómetro lineal, el cual requiere un único pin para su conexión.
- Display 7 segmentos: Tras consultar con la comunidad académica y considerando que el uso principal de estos displays es mostrar la hora, se ha determinado que se necesitan al menos cuatro dígitos. Para simplificar la implementación y aprovechar las ventajas de la multiplexación, se ha seleccionado un par de display de 2 dígitos que requieren 11 pines entre los dos.
- Pantalla oled: La pantalla oled seleccionada cuenta con el protocolo I2C y permite una cantidad de 128x64 pixeles.

Para los demás elementos, se ha seleccionado un elemento individual para cada uno, siguiendo criterios de funcionalidad y aplicabilidad. En la *Tabla 6* se observa la cantidad y el total de pines necesarios con lo cual se ocupan los 25 pines disponibles del ESP32.

Tabla 6
Cantidad de pines por el total de elementos

ELEMENTOS	CANTIDAD	PINES NECESARIOS	PINES TOTALES
LED	4	1	4
SWITCH	4	1	4
DISPLAY 7 SEGMENTOS	1	11	11
OLED	1	2	2
POTENCIOMETRO	1	1	1
PULSADORES	2	1	2
BUZZERS	1	1	1
TOTAL		15	25

2.2. Etapa del diseño del Módulo entrenador utilizando de los elementos con el microcontrolador y dispositivos a usar.

Con el objetivo de asegurar el óptimo rendimiento de todos los componentes en conjunto, se procedió a realizar el ensamblaje individual de cada elemento en una protoboard, utilizando el microcontrolador ESP32 para realizar pruebas exhaustivas. Durante este proceso, se tuvieron en cuenta las limitaciones de disponibilidad de pines en el ESP32, lo cual implicó realizar ajustes y consideraciones especiales para garantizar la correcta conexión y funcionalidad de todos los elementos involucrados.

Al tener en cuenta las limitaciones de pines del ESP32, se evaluó cuidadosamente la distribución de pines para cada componente, asegurando que no se excediera la capacidad del microcontrolador y se cumplieran los requisitos de conexión necesarios. Esto permitió una configuración adecuada de los pines disponibles, evitando conflictos y asegurando un funcionamiento armonioso del sistema en su conjunto.

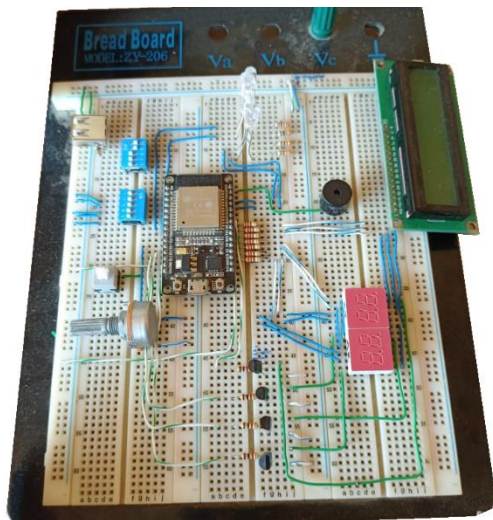
Es importante destacar que esta planificación y ajuste de conexiones considera las características específicas del ESP32 y los componentes utilizados en este proyecto en particular. Por tanto, es recomendable adaptar y ajustar la configuración de pines según las necesidades individuales de otros proyectos o sistemas similares.

2.2.1. Etapa de conexión de los elementos con el Microcontrolador

Inicialmente se realiza el armado completo del circuito en una protoboard, considerando la ubicación y conexión de cada uno de los pines de los elementos, siguiendo el esquema de conexión mostrado en la *Figura 23*. Además, se llevaron a cabo diversas pruebas para comprobar la funcionalidad y confiabilidad del circuito en diferentes escenarios. Es importante destacar que la ubicación de los componentes en la protoboard puede variar según la preferencia del usuario, siempre y cuando se respeten las conexiones establecidas en el esquema de conexión como se observa en la siguiente figura:

Figura 23.

Montaje de los elementos en la protoboard



Nota: Imagen tomada por el autor

2.2.1.1. *Ejemplo de prueba de los elementos.*

Se realizó la definición de los pines de entrada (sw) como suiches, los cuales se encargan de activar o desactivar un LED determinado, con el uso del lenguaje de programación C y el IDE de Arduino. Para ello, se llevó a cabo la escritura de un código en el cual se asignaron los pines correspondientes y se programó la lógica

necesaria para el encendido o apagado del LED; en la *Figura 24* se observa el código.

Figura 24.

Código para encender un LED con un Suiche

```
void setup() {
  //Configuración de pines
  pinMode(LED_PIN, OUTPUT);
  pinMode(SWITCH_PIN, INPUT);
}

void loop() {
  //Control del LED mediante el switch
  if(digitalRead(SWITCH_PIN) == HIGH) {
    digitalWrite(LED_PIN, HIGH);
    delay(500);
  }
  else {
    digitalWrite(LED_PIN, LOW);
  }
}
```

Nota: Imagen hecha por el autor.

Se observa en el código anterior es escrito en C, donde se definió el pin 23 como salida (led1) y el pin 36 como entrada (sw1). Al activarse el switch (sw1) a través del código, se enciende el led (led1) mediante la función digitalWrite(). El tiempo que el led permanece encendido es de 500 milisegundos, gracias a la función delay(). Es importante destacar que este es un ejemplo sencillo de cómo se pueden utilizar los pines de entrada y salida de un microcontrolador como el esp32 y que puede ser aplicado en la implementación de sistemas más complejos en el futuro. Además, es fundamental tener en cuenta la importancia de conocer los pines y las funciones que se utilizan en la programación para asegurar un correcto funcionamiento del sistema y evitar posibles errores en la implementación.

2.2.2. Prueba conjunta de los elementos

La etapa de conexión conjunta de todos los componentes en la protoboard con el ESP32 fue crucial para lograr la integración completa y el funcionamiento armonioso del sistema. Se prestó especial atención a las limitaciones de configuración presentes en algunos pines del ESP32, con el fin de garantizar una correcta interacción y compatibilidad entre todos los elementos involucrados.

Para llevar a cabo esta tarea, se realizó un exhaustivo análisis de la hoja de datos del ESP32, así como de las especificaciones técnicas de los demás componentes, con el objetivo de ahondar en las características y requerimientos de cada uno de los componentes del Módulo entrenador.

Esto permitió identificar los pines del ESP32 que requerían una configuración específica, ya sea debido a sus funciones especiales o a posibles conflictos con otros componentes.

Una vez identificadas las limitaciones de configuración, se procedió a realizar los ajustes necesarios en el esquema de conexiones. Esto implicó redistribuir algunos cables y cambiar la asignación de pines, de manera que se resolvieran posibles conflictos y se garantizara la correcta comunicación entre el ESP32 y los demás componentes, además, se tuvo en cuenta la distribución óptima de los cables, evitando cruces y enredos que pudieran generar interferencias o cortocircuitos.

Una vez finalizada la etapa de conexión conjunta, se llevaron a cabo pruebas exhaustivas para verificar el correcto funcionamiento del sistema en su conjunto. Se realizaron mediciones de voltaje y corriente en diferentes puntos, se evaluó la transmisión de datos entre los componentes y se comprobó el rendimiento general del sistema. En caso de detectar algún problema o inconsistencia, se realizaron ajustes adicionales para corregirlos y asegurar un funcionamiento óptimo, en la *Figura 25* se observan los elementos a usados en el esquemático.

Figura 25.

Elementos usados en la protoboard.

ENTRADAS DIGITALES		DISPLAY			SALIDAS	
SWITCHES		A	B	C	DISPLAY OLED	
DIP SWITCH X5		D	E	F	128X64 PIXELS	
			G			
BOTONES		ESP32			LEDS	
P1	P2	WIFI	ALIMENTACIÓN	BLUETOOTH	L1	L2
		802.1gbn	Micro USB	4.2	L3	L4
ENTRADA ANALOGA						
POTENCIOMETRO		ALIMENTACIÓN EXTERNA			BUZZER	
POT1		DC 5V			DIGITAL MONO	

Nota: Gráfico hecho por el autor.

En conclusión, la etapa de conexión conjunta en la protoboard con el esp32 representó un desafío técnico importante que requirió un análisis detallado, ajustes precisos y pruebas rigurosas. Sin embargo, gracias a estos esfuerzos, se logró una integración exitosa de todos los componentes, asegurando un funcionamiento armonioso y eficiente del sistema en su conjunto.

2.2.3. Diseño del esquemático de la conexión de los elementos con el Microcontrolador

Una vez que se han definido los elementos necesarios y sus conexiones, se procede a crear el esquemático del proyecto. Para esto, se utiliza el software Eagle, el cual permite diseñar el esquema electrónico de manera precisa y organizada. El esquemático resultante se muestra en la Figura 26.

El diseño del esquemático es crucial, ya que representa la representación visual de las conexiones y componentes utilizados en el proyecto. Proporciona una guía clara

2.3. Etapa de implementación del diseño del Módulo entrenador en una PCB acoplando cada uno de los elementos.

Esta sección consiste en tomar el esquemático del circuito del Módulo Entrenador y mediante un software especializado realizar la PCB, teniendo en cuenta el número de capas y los elementos a usar, así como la verificación de su funcionamiento.

2.3.1. Diseño de la PCB a partir del esquemático del circuito.

La elaboración de una PCB implica una serie de etapas que comienzan con la creación del esquemático en un software especializado, en este caso Eagle. Una vez elaborado el esquemático, se procede a la creación de la placa en Eagle, en la cual se busca optimizar la conexión entre los distintos componentes. Es importante destacar que, durante esta etapa, se lleva a cabo una ubicación manual estratégica con el fin de lograr una adecuada funcionalidad de la placa, sin descuidar su estética. Por tanto, se procura que los dispositivos de igual naturaleza estén colocados en una ubicación cercana, de modo que el usuario tenga una idea implícita de los elementos y su uso. De esta manera, se logra una PCB eficiente y bien organizada, que cumple con los requisitos funcionales y estéticos.

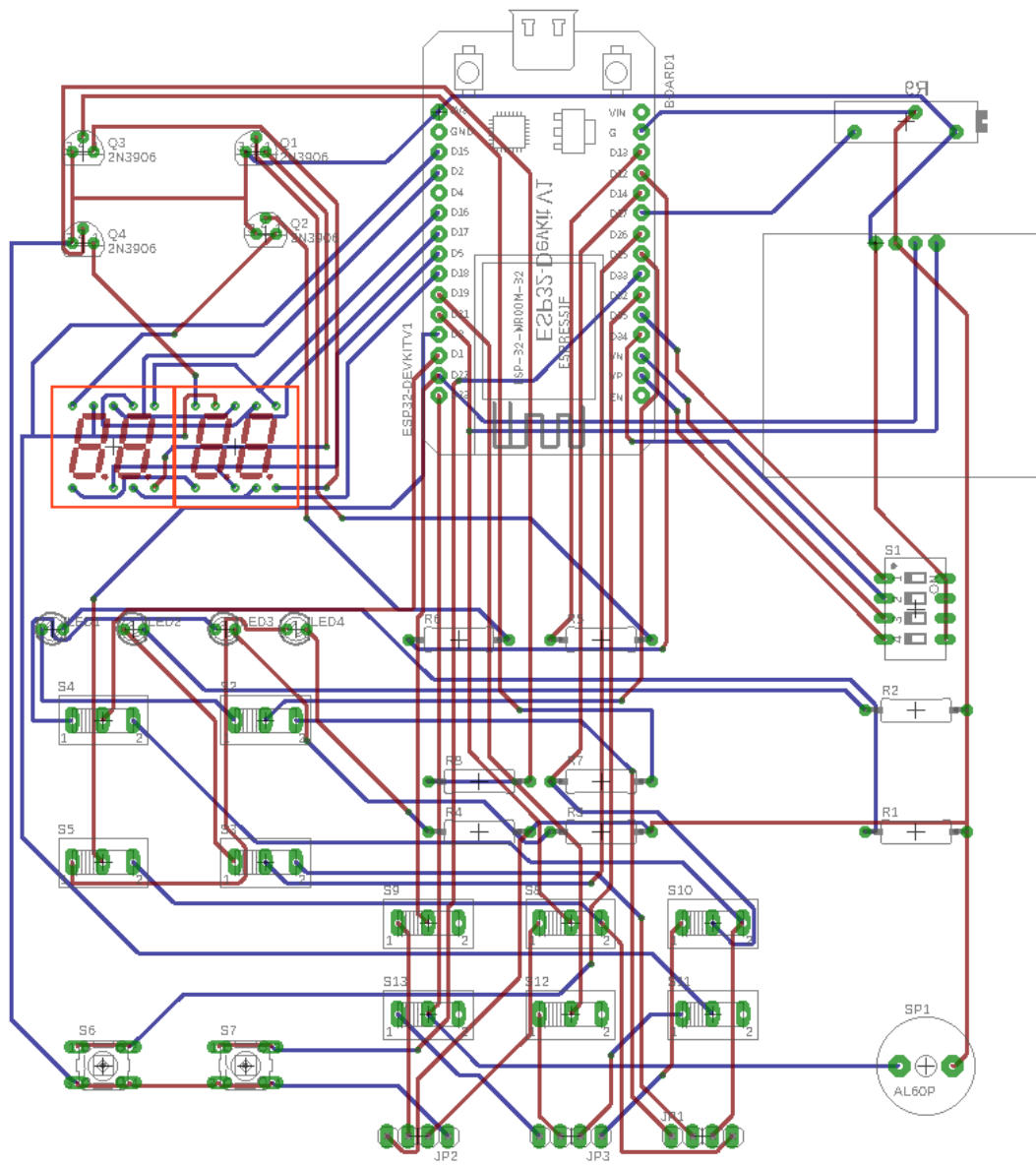
Para este proyecto, se ha optado por una PCB de doble cara, lo cual permite aprovechar al máximo el espacio y facilitar las interconexiones entre los componentes.

En la *Figura 27* se observa el diseño final de la PCB, donde las conexiones en rojo son la capa superior y la azul la capa inferior. Posteriormente se realiza la ubicación estratégica de los elementos, donde se agrupan los componentes relacionados cercanos entre sí, lo cual facilita la interconexión y mejora el rendimiento del circuito. Esta disposición ayuda a organizar y optimizar las conexiones entre los elementos del circuito.

2.3.1.1. Rediseño de la PCB

Para la fabricación del circuito impreso en doble capa, se identificó la necesidad de utilizar maquinaria especializada. Inicialmente, se planificó utilizar la ruteadora disponible en la Universidad Pontificia Bolivariana (UPB) sede Montería para llevar a cabo este proceso. Sin embargo, se enfrentaron dificultades técnicas relacionadas con los controladores de la máquina, lo que imposibilitó su utilización. Ante esta situación, se decidió buscar alternativas comerciales para la impresión del circuito.

Figura 27.
Diseño de la PCB a doble capa.



Nota: Imagen hecha por el autor.

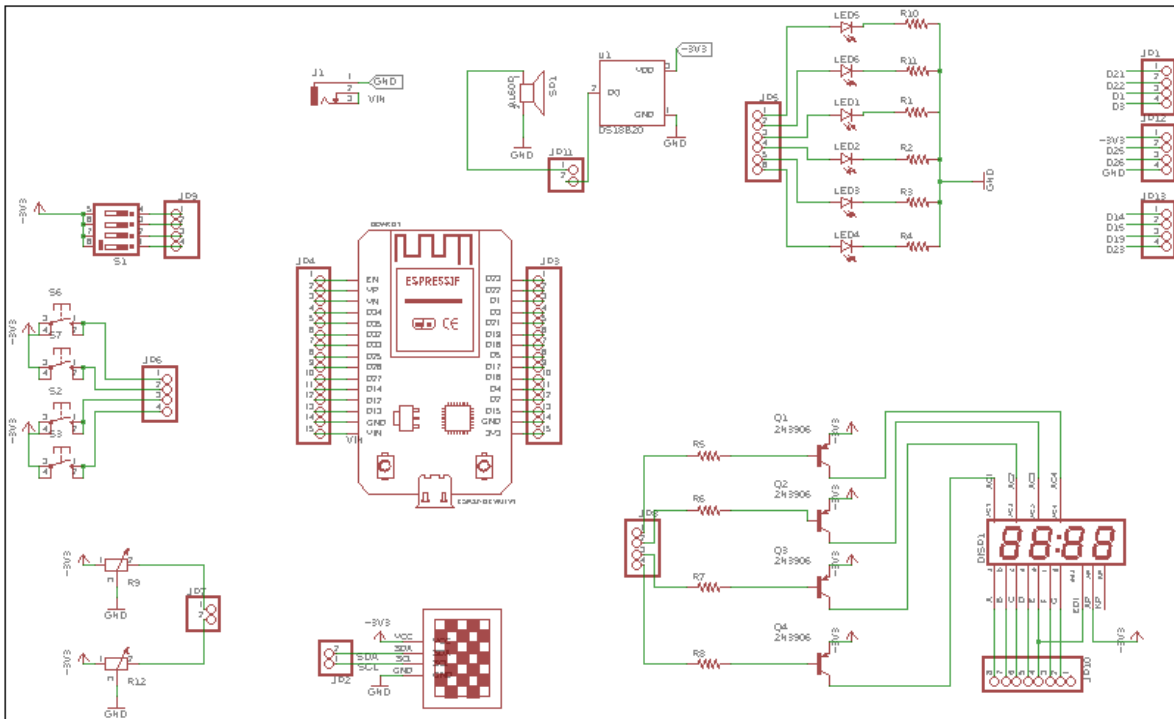
Se realizó una exhaustiva búsqueda a nivel nacional con el objetivo de encontrar proveedores de servicios de impresión de circuitos en doble capa. Lamentablemente, se constató que aún no existe una oferta formal de este tipo de servicio en el país. Ante esta limitación, se tomó la determinación de realizar un rediseño en la estructura del circuito, buscando una configuración más modular que pudiera ser impresa por cualquier persona con conocimientos en la materia.

El nuevo esquema del circuito resultante refleja esta adaptación, presentando una configuración más accesible y factible de ser fabricada. Con este enfoque, se espera

maximizar la posibilidad de llevar a cabo la impresión del circuito en doble capa, brindando una solución que se ajuste a las limitaciones actuales y facilite la replicación del diseño por parte de otros interesados en el campo.

Con lo cual el esquema del nuevo circuito queda como se observa en la *Figura 28*:

Figura 28.
Resultado del circuito modular.



Nota: Esquemático hecho en EAGLE por el autor

La necesidad de adaptar el diseño del circuito a una configuración más modular, es decir, el usuario debe elegir qué usar y realizar la conexión, condujo a una reducción significativa en las interconexiones requeridas, lo que a su vez permitió optimizar el diseño para que pudiera ser implementado en una sola capa. Además, se realizaron ajustes y mejoras en algunos elementos del circuito, siempre considerando la necesidad de mantener un equilibrio entre funcionalidad y costo.

El nuevo esquema resultante refleja estas modificaciones y mejoras. El diseño ahora presenta una distribución más eficiente de los componentes, minimizando la cantidad de rutas y simplificando el proceso de fabricación. Asimismo, se incorporaron elementos adicionales que fueron identificados como necesarios para dar un plus al proyecto, asegurando que estos ajustes no impactaran significativamente en el costo final del circuito, los cuales se observan en la *Figura 29*.

Figura 29.
Elementos finales.

ENTRADAS DIGITALES		DISPLAY			SALIDAS	
SWITCHES		A	B	C	DISPLAY OLED	
DIP SWITCH X5		D	E	F	128X64 PIXELS	
		G				
BOTONES		ESP32			LEDS	
P1	P2	WIFI	ALIMENTACIÓN	BLUETOOTH	L1	L2
		802.1gbn	Micro USB	4.2	L3	L4
ENTRADA ANALOGA						
POTENCIOMETRO		ALIMENTACIÓN EXTERNA			BUZZER	
POT1		DC 5V			DIGITAL MONO	

Nota: Imagen hecha por el autor.

Ahora el diseño final del circuito adopta una estructura modular, optimizada para su implementación en una sola capa. Se han realizado ajustes y mejoras estratégicas, tanto en la distribución de los componentes como en la incorporación de nuevos elementos, con el objetivo de garantizar un rendimiento óptimo y una fabricación más eficiente del circuito, sin comprometer excesivamente el costo asociado al proyecto.

2.3.2. Inclusión del PIC 16F84A

Debido a que en el laboratorio del curso de Sistemas Digitales Microcontrolados se trabaja inicialmente con el PIC 16F84A, con lo cual es necesario un circuito o tarjeta adicional que lo hiciera compatible con la placa diseñada. Se considera como solución una placa adicional para no modificar el circuito, pero realizando un análisis este tendría que ser a doble capa, con lo cual se decide implementar en esta misma. Con lo cual se observa que es posible implementarlo entre los pines del ESP32 en la placa de desarrollo, como se observa en la *Figura 30*.

Figura 30.
Inclusión del PIC16F84

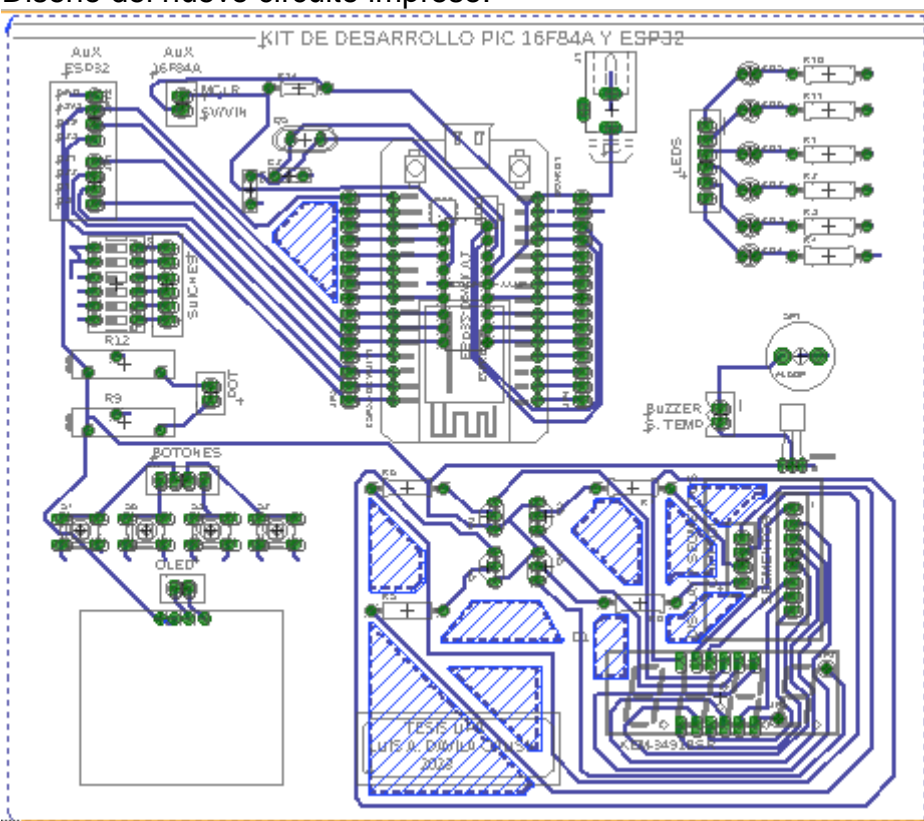


Nota: Esquemático hecho por el autor.

2.3.3. Impresión de la PCB

En el proceso de diseño y fabricación de la nueva PCB, se ha prestado especial atención a la ubicación estratégica de los elementos para garantizar un esquema eficiente y funcional. Mediante un exhaustivo análisis y considerando las características y requerimientos de cada componente, se ha logrado una disposición óptima en la placa de circuito impreso. La *Figura 31* ilustra de manera clara y concisa la distribución final de los elementos en la PCB, reflejando un enfoque cuidadoso y meticuloso en el diseño del circuito. Esta configuración ha sido cuidadosamente planificada para optimizar la interconexión de los componentes y maximizar el rendimiento del sistema en su conjunto.

Figura 31.
Diseño del nuevo circuito impreso.



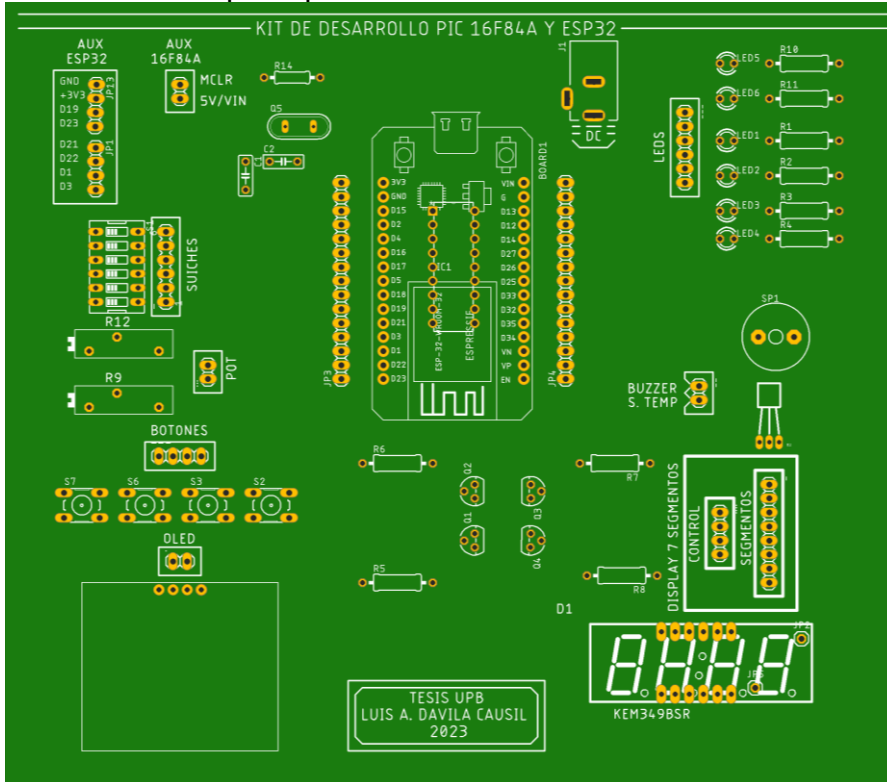
Nota: Imagen hecha por el autor.

2.3.3.1. Capa superior.

En la capa superior de la PCB se muestra la disposición de los elementos sin que haya conexiones o ensamblajes directos en esta capa. A diferencia de la capa inferior, donde se llevaron a cabo las interconexiones entre los componentes, en la capa superior se priorizaron las leyendas descriptivas y el espacio adecuado para cada elemento. Esta disposición permite una mejor visualización de los componentes y facilita el proceso de montaje y soldadura en la capa inferior. Es importante destacar que aunque en la capa superior no se realicen conexiones directas, su diseño y leyendas descriptivas son fundamentales para asegurar el

correcto funcionamiento y la eficiencia del circuito en su conjunto, su diseño se observa en la *Figura 32*.

Figura 32.
Diseño de la capa superior de la PCB.

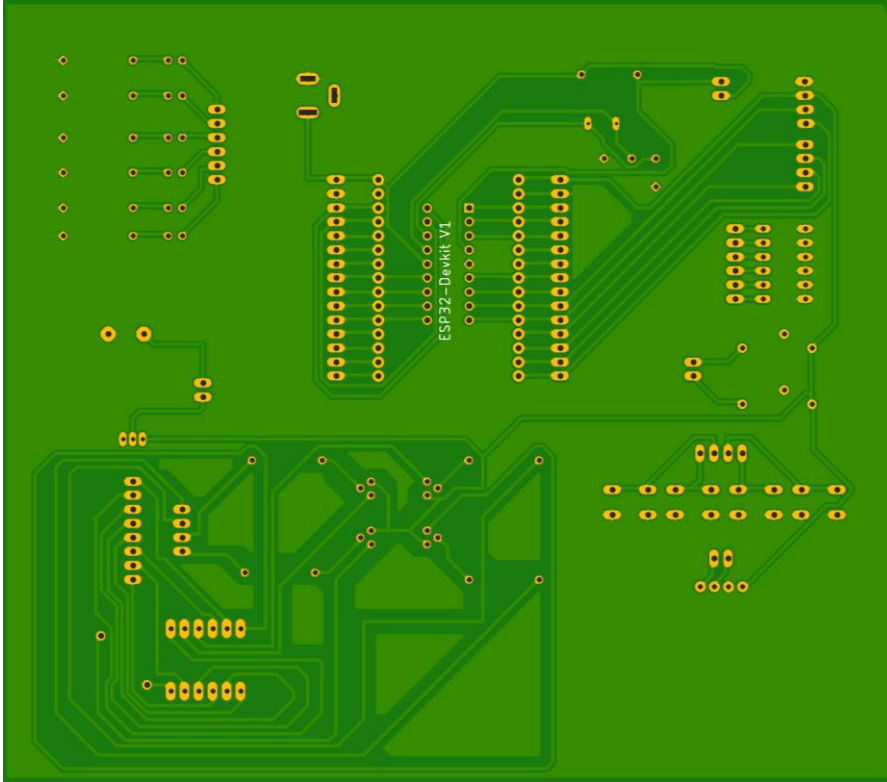


Nota: Imagen hecha por el autor.

2.3.3.2. Capa inferior

La capa inferior de la PCB se diseñó de manera estratégica para satisfacer el objetivo de tener una estructura de una sola capa, considerando la disposición de elementos que atraviesan ambas placas y no son superficiales. Esta capa resultó ser la opción más adecuada para la interconexión de los dispositivos. Además, se tuvo en cuenta la importancia de tener un plano de tierra (GND) extenso en toda la placa, con el propósito de reducir las impedancias presentes en el circuito. Esta elección garantiza una distribución uniforme de la referencia de tierra, minimizando las variaciones de voltaje y mitigando posibles problemas de ruido e interferencia electromagnética. La figura adjunta ilustra el resultado de este diseño, resaltando la disposición de la capa inferior y la distribución de la conexión a tierra para una mejor integridad de las señales y un rendimiento óptimo del circuito, como se observa en la *Figura 33*.

Figura 33.
Diseño capa inferior de la PCB.



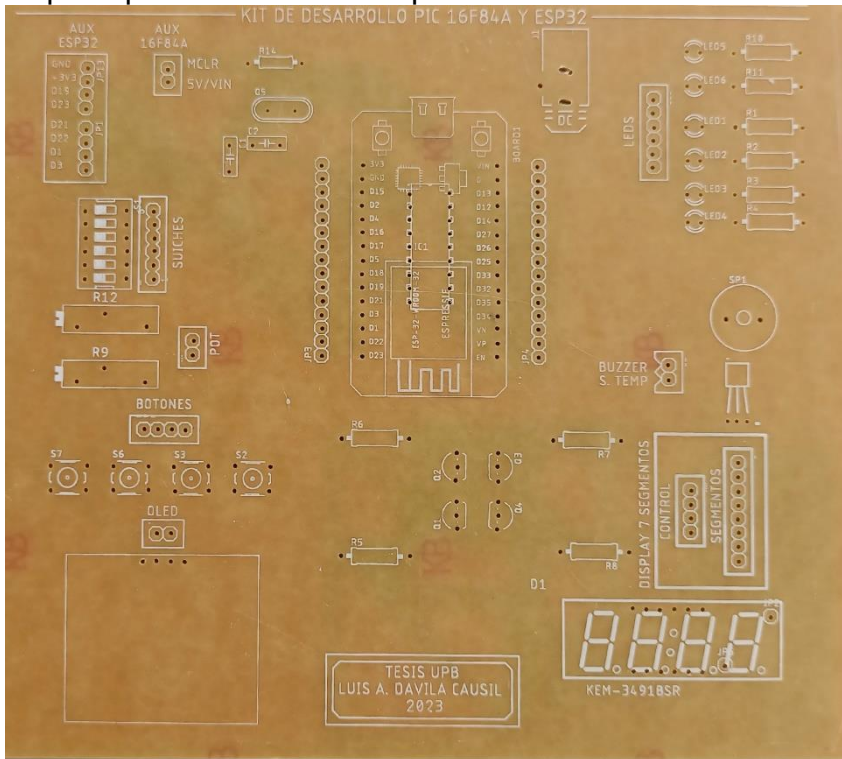
Nota: Imagen hecha por el autor

2.3.3.3. Resultado del circuito impreso

En esta sección se presenta el resultado de la impresión del circuito, con la ausencia de los elementos en el circuito lo que permite una mejor visualización de la estructura y las rutas de interconexión, lo que resulta útil para comprender la lógica y el flujo de señales dentro del circuito.

En la *Figura 34* se observa el resultado de la capa superior, la inclusión de las descripciones brinda una guía práctica para los estudiantes y facilita la identificación de los componentes y su ubicación correcta durante el proceso de ensamblaje. Esto fomenta un aprendizaje más efectivo y una mayor comprensión de los principios de electrónica involucrados en el funcionamiento del circuito.

Figura 34.
Capa superior del circuito impreso sin los elementos.



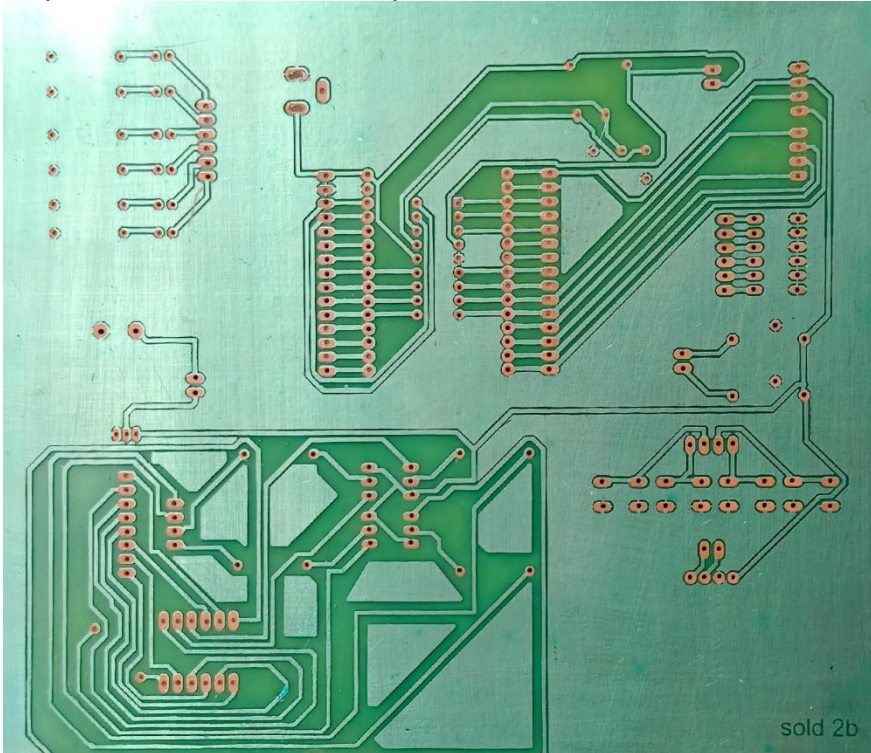
Nota: Foto tomada por el autor

En el diseño del circuito impreso la capa inferior desempeña un papel fundamental en la realización de las conexiones necesarias para el correcto funcionamiento del circuito. Esta capa alberga las rutas de interconexión que permiten el flujo de señales entre los diferentes componentes.

La disposición y organización de las conexiones en la capa inferior se realiza siguiendo criterios de diseño y optimización, buscando reducir la longitud de las rutas, minimizar el cruce de pistas y evitar interferencias no deseadas. Esto se logra a través de una cuidadosa planificación y enrutamiento de las conexiones.

En la *Figura 35* se pueden apreciar los trazados de cobre que conectan los distintos elementos del circuito, tales como resistencias, capacitores, transistores, entre otros. Estas conexiones están diseñadas de manera estratégica para asegurar una adecuada transmisión de señales y minimizar la interferencia electromagnética.

Figura 35.
Capa inferior del circuito impreso.



Nota: Foto tomada por el autor.

En resumen, la capa inferior del circuito impreso juega un papel crucial en la creación de las conexiones necesarias para el funcionamiento del circuito. Su diseño y enrutamiento adecuado aseguran la integridad de las señales y la correcta operación del sistema electrónico.

2.4. Validación del Módulo Entrenador.

Con el propósito de evaluar minuciosamente y verificar el correcto funcionamiento de la placa de desarrollo, se llevó a cabo un taller el día viernes 12 de mayo del 2023 con estudiantes del curso Sistemas Digitales Microcontrolados bajo la supervisión del Ingeniero Luis Manuel León Monterrosa actual docente del curso, como se observa en la figura; en el cual se realizaron diversas pruebas y pruebas de concepto. El objetivo central de esta actividad consistió en asegurar que todos los componentes y elementos integrados en la placa estuvieran adecuadamente configurados y operativos, permitiendo así su correcto desempeño en futuros proyectos. Además, se documentaron los resultados obtenidos para su posterior análisis y seguimiento.

Figura 36.

Realización del taller.



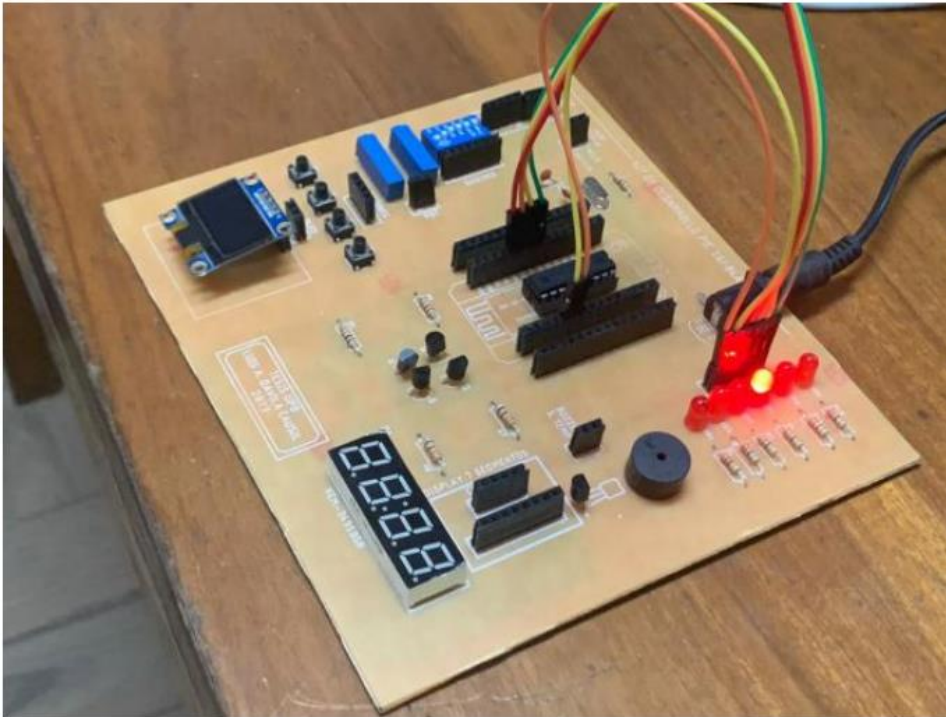
Nota: Imagen tomada por el maestro del curso.

2.4.1. Realización del taller donde se usan varios elementos del micro y se observa el funcionamiento del kit.

En el taller se trabajó con una placa prototipo, con la cual se pudieron realizar distintas conexiones y montajes prácticos con ambos microcontroladores, como se observa en la *Figura 37* el encendido de LEDs con el PIC 16F84A. Se realizaron pruebas individuales de cada componente, tales como suiches, leds, display de 7 segmentos y pantalla OLED, a fin de verificar su correcto funcionamiento y su respuesta ante las instrucciones enviadas desde el microcontrolador. Estas pruebas permitieron detectar posibles irregularidades y realizar ajustes necesarios para optimizar el rendimiento de la placa.

Los suiches permitieron a los estudiantes comprender y experimentar con las configuraciones de entradas y salidas digitales. Los leds, por otro lado, brindaron una forma visual de verificar el funcionamiento de los circuitos. Asimismo, el display de 7 segmentos les permitió mostrar información numérica de manera clara y concisa.

Figura 37.
Módulo Entrenador funcionando con el PIC16F84A.



Nota: Imagen tomada por el autor durante la realización del taller.

En la *Figura 38* se aprecia a los estudiantes inmersos en el taller, mostrando su compromiso e interés por aprender y explorar las funcionalidades de la placa de desarrollo. Su participación activa y su capacidad para plantear preguntas y discutir los resultados obtenidos contribuyeron a enriquecer la experiencia y a generar un ambiente propicio para el aprendizaje colaborativo.

Figura 38.
Estudiante realizando un montaje en el Módulo Entrenador.

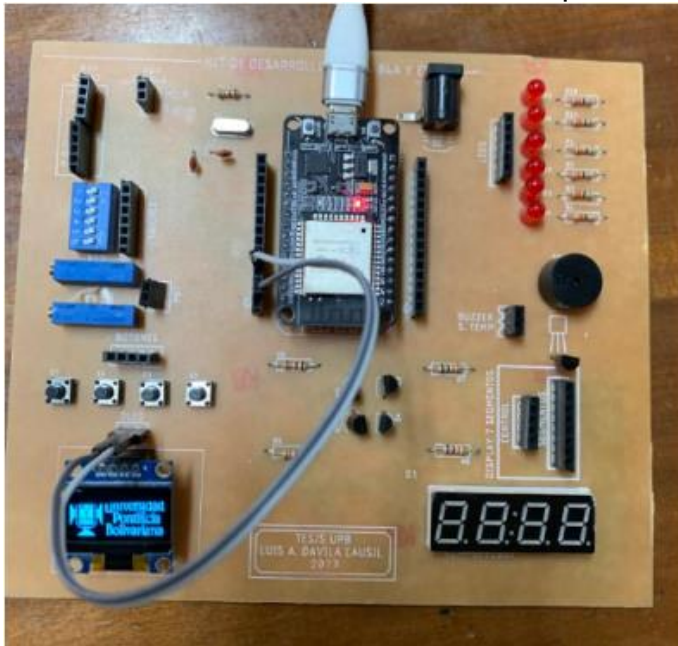


Nota: Imagen tomada por el autor durante la realización del taller.

Sin embargo, uno de los puntos más destacados del taller fue la introducción a la pantalla OLED, una tecnología más actual que despertó un gran interés entre los participantes. Su calidad de imagen y la flexibilidad en la programación con el ESP32 fueron aspectos que capturaron la atención de los estudiantes. A través de la pantalla oled, pudieron explorar la visualización de textos y gráficos como se observa en la figura, sumergiéndose en un entorno interactivo y dinámico.

Figura 39.

Módulo Entrenador funcionando con la pantalla OLED y el ESP32



Nota: Imagen tomada por el autor durante la realización del taller.

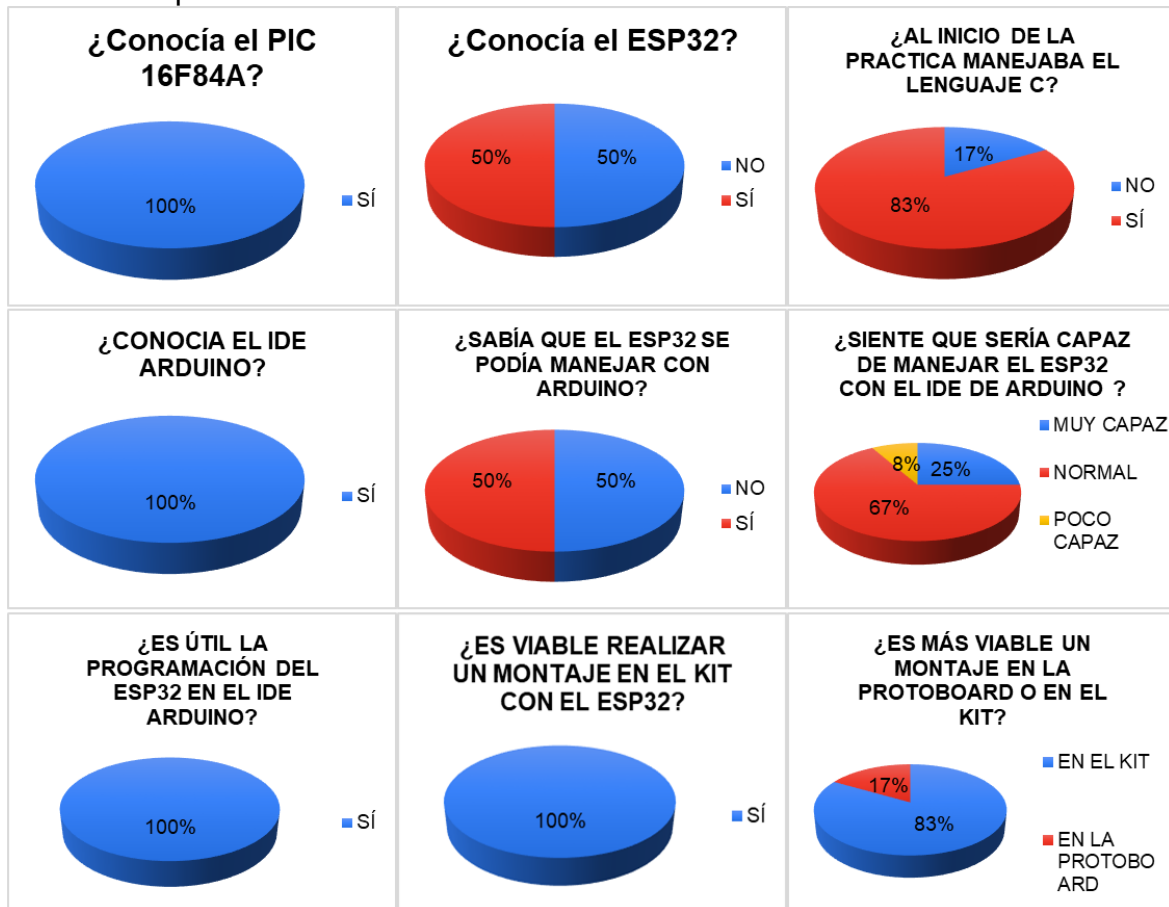
En síntesis, el taller organizado para evaluar y comprobar el funcionamiento de la placa de desarrollo fue una instancia esencial para asegurar su correcta configuración y operatividad. Mediante pruebas rigurosas y la participación activa de los estudiantes, se logró identificar y corregir posibles fallos, garantizando así la calidad y confiabilidad de la placa para futuros proyectos y aplicaciones. Este taller representa un valioso aporte al conocimiento y desarrollo de habilidades prácticas a futuro con el Módulo Entrenador, fortaleciendo adicionalmente la formación académica de los estudiantes involucrados.

2.4.2. Resultados del taller.

Al concluir el taller, se llevó a cabo una encuesta (anexo C) con el objetivo de evaluar el nivel de conocimiento que poseían los estudiantes sobre los elementos utilizados durante el desarrollo de las actividades. Esta encuesta permitió recopilar información relevante acerca de la familiaridad de los participantes con los componentes del kit, así como determinar si existe la necesidad de incorporar nuevos elementos y recursos en el laboratorio.

A partir de los resultados preliminares obtenidos de la encuesta realizada se obtuvo la *Figura 40*, donde se desprenden diversos hallazgos y conclusiones de relevancia. En primer lugar, se evidencia que los participantes poseen un conocimiento sólido y familiaridad con el microcontrolador 16F84A, el cual fue objeto de estudio durante el curso. No obstante, se destaca que varios de ellos presentan limitado conocimiento acerca del ESP32, tanto en su configuración como en su programación. Esta brecha en el conocimiento sugiere la necesidad de brindar mayor atención y recursos para familiarizar a los estudiantes con este microcontrolador y aprovechar su potencial en futuros proyectos.

Figura 40.
Resultados preliminares de la encuesta



Nota: Imagen hecha con los resultados de la encuesta, [Figura], 2023.

Asimismo, en la *Figura 41* se constata que la mayoría de los elementos proporcionados en el Módulo Entrenador, como los suiches, leds y displays de 7 segmentos, fueron ampliamente trabajados y utilizados en el entorno del laboratorio. Sin embargo, otros elementos como el buzzer, el sensor de temperatura y la pantalla OLED no fueron abordados en profundidad en la materia por los participantes. Es importante resaltar que el Módulo ofrece la posibilidad de trabajar con estos componentes adicionales, y durante el taller se realizaron pruebas y exploraciones con el fin de fomentar su empleo en futuros proyectos. Esta experiencia permitió a

los estudiantes ampliar su conocimiento y experimentar con nuevas funcionalidades que el módulo entrenador ofrece. En términos de percepciones y valoraciones, los participantes coinciden en que el Módulo Entrenador presenta una ventaja significativa en comparación con el uso exclusivo de una protoboard.

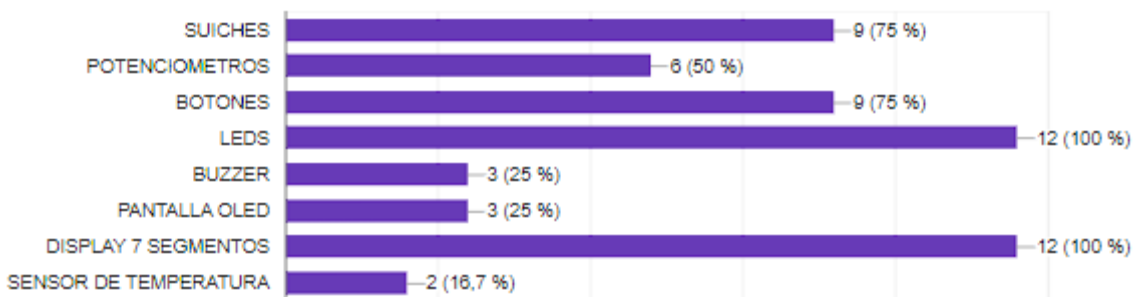
Figura 41.

Elementos conocidos y usados por los estudiantes en el curso.

DE LOS ELEMENTOS EN EL KIT, ¿CUÁLES CONOCÍA ANTERIORMENTE?



DE LOS ELEMENTOS EN EL KIT, ¿CUÁLES SE USARON EN LA MATERIA?

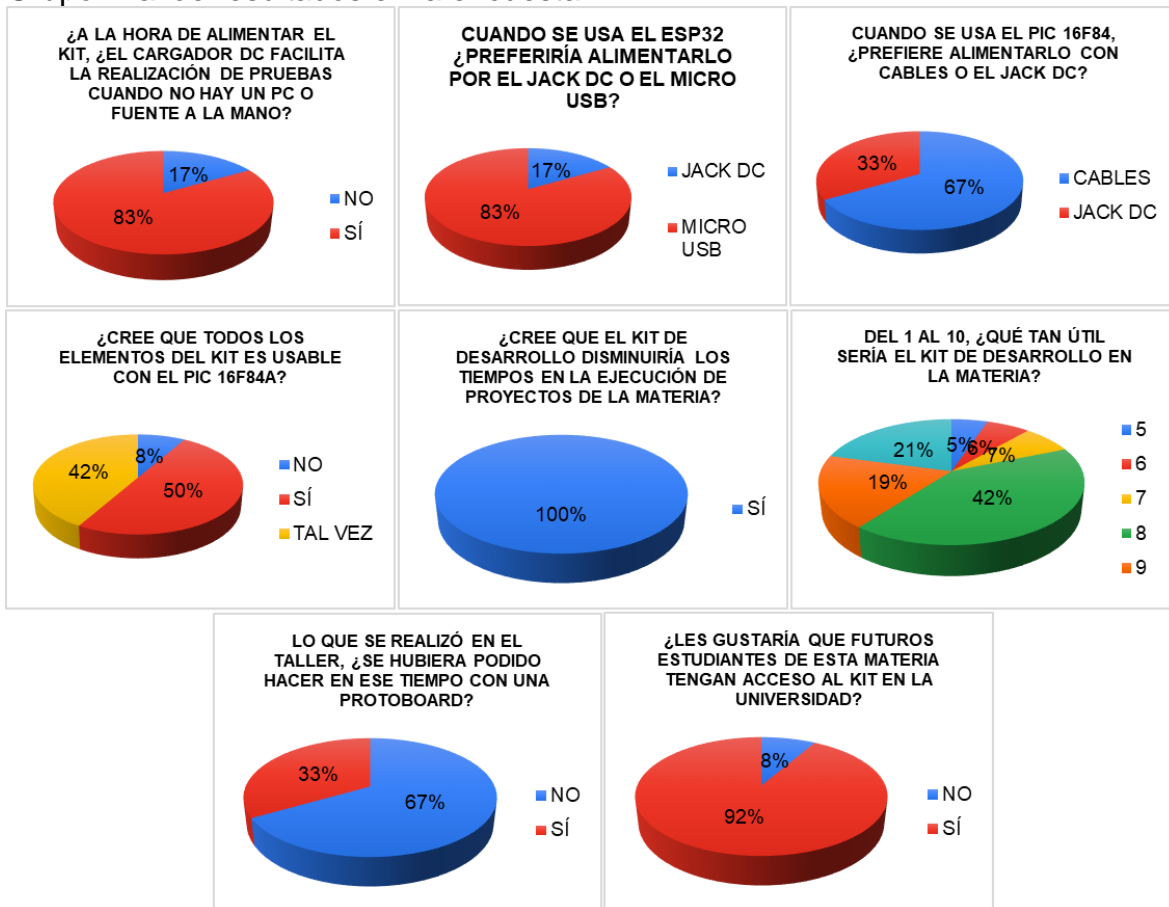


Nota: Resultados de la encuesta de los elementos conocidos, [Figura], 2023.

La posibilidad de contar con una fuente de alimentación mediante el Jack DC brinda mayor comodidad y eficiencia en la conexión y operación de los componentes. Además, los estudiantes expresan su recomendación de implementar el uso del kit en futuras actividades y proyectos dentro del ámbito universitario. Esta retroalimentación positiva respalda la utilidad y pertinencia del Módulo Entrenador en el contexto académico, destacando su potencial para fortalecer la formación teórica y práctica de los estudiantes en el campo de los microcontroladores, donde recomiendan su uso a futuro en la academia, en la *Figura 42* se observan estos resultados.

Figura 42.

Grupo final de resultados en la encuesta



Nota: Tomado de los resultados de la encuesta. [Figura], 2023.

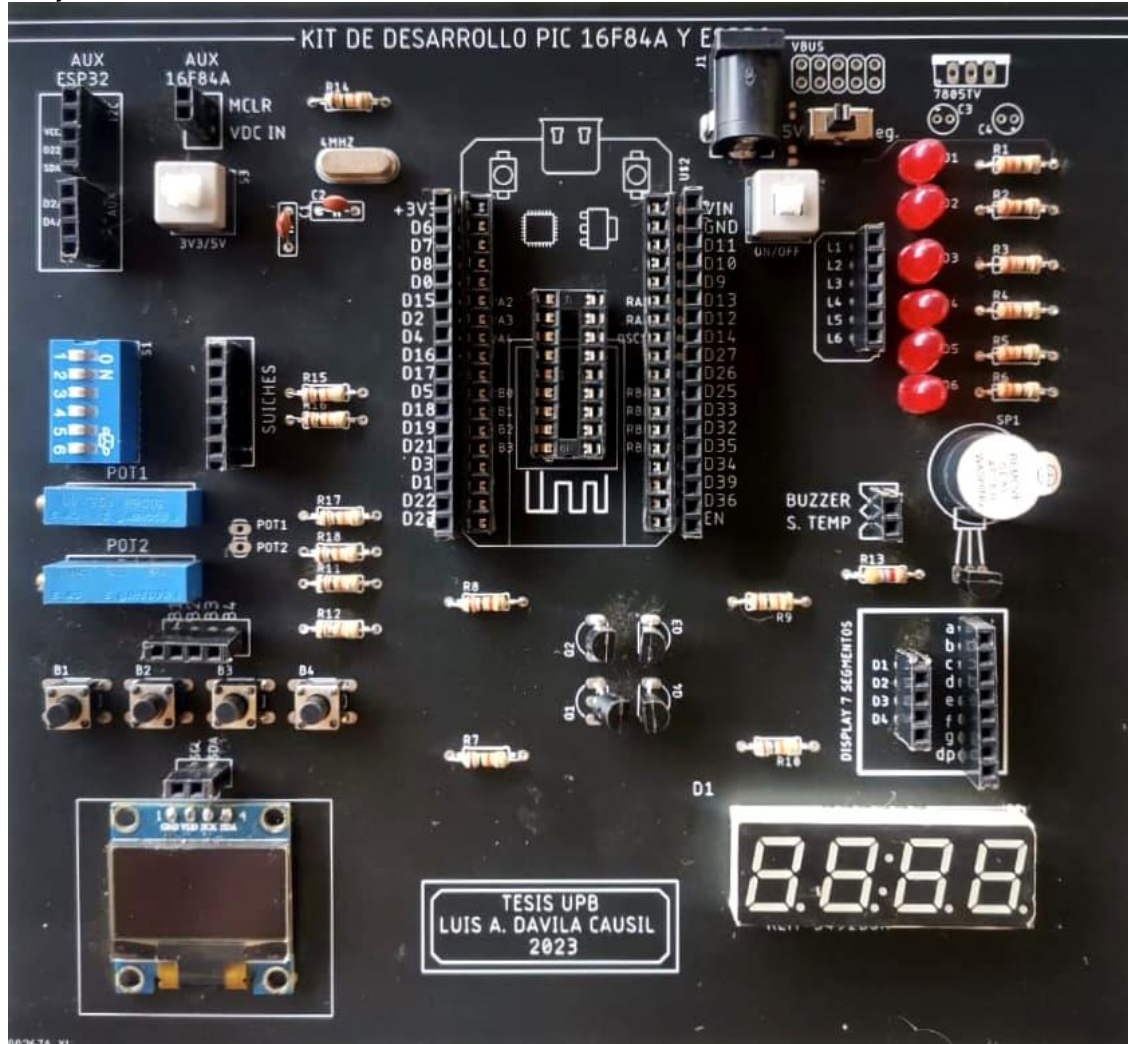
Los resultados de la encuesta revelan tanto fortalezas como áreas de oportunidad en relación al conocimiento y manejo de los microcontroladores, así como el aprovechamiento de los componentes disponibles en el Módulo Entrenador. La identificación de estas perspectivas contribuye a orientar futuras estrategias y acciones de enseñanza, con el propósito de mejorar la comprensión y habilidades prácticas de los estudiantes en el campo de la electrónica y los microcontroladores. Asimismo, la positiva percepción y recomendación del uso del Módulo Entrenador respaldan su valor como recurso didáctico en el contexto educativo, promoviendo un aprendizaje más significativo y enriquecedor.

3. RESULTADOS

En este capítulo, se expone el resultado definitivo de la placa impresa del Módulo entrenador, abordando dos aspectos principales. En la primera sección, se detallan la estructura del hardware de la tarjeta, incluyendo sus componentes. La segunda sección se enfoca en los costos asociados al kit, brindando un análisis detallado de los gastos involucrados en su fabricación. Estos elementos proporcionan una visión integral del Módulo Entrenador, el cual se observa en la *Figura 43*.

Figura 43.

Tarjeta final del Módulo Entrenador.



Nota: Imagen tomada por el autor

3.1. Estructura de la tarjeta de desarrollo.

En esta sección se expone la descripción detallada de los elementos utilizados en la tarjeta de desarrollo, destacando su ubicación precisa en el diseño y su funcionalidad específica. Se brinda una visión completa de cada componente, resaltando su importancia y rol dentro del sistema. Esta información resulta

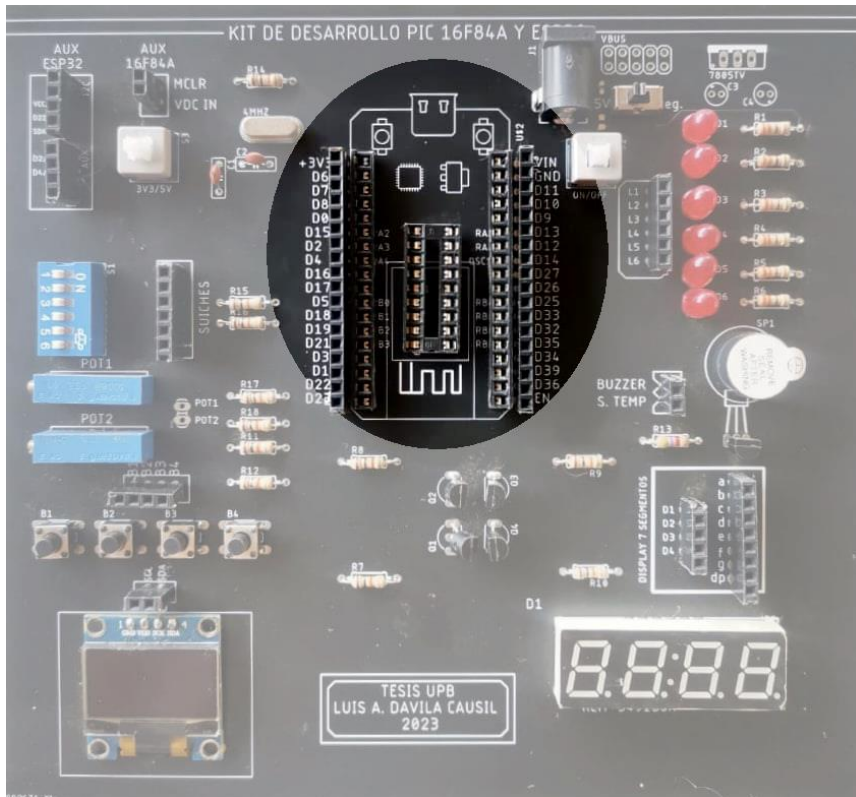
fundamental para comprender la estructura y configuración de la tarjeta, así como para identificar las posibilidades y aplicaciones que ofrece en diversos contextos.

3.1.1. Ubicación de los microcontroladores

Se presenta la ubicación de los microcontroladores PIC 16F84A y el ESP32, donde el 16f84a se monta sobre la base de circuitos integrados de 9 pines, y el ESP32 en los pin headers de 18 pines, con ayuda visual de la serigrafía en la capa superior de la placa de desarrollo, como se observa en la *Figura 44*.

Figura 44.

Ubicación de los microcontroladores.



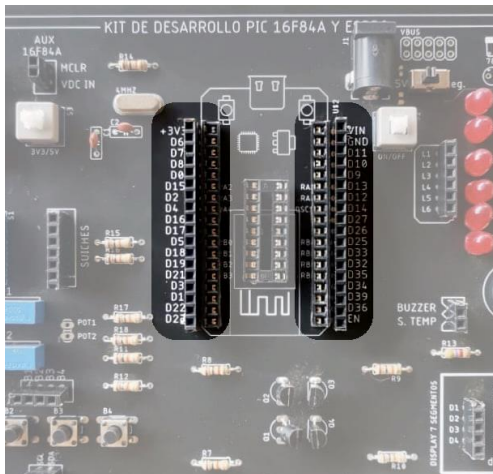
Nota: Imagen tomada por el autor.

3.1.2. Salidas a los elementos de la placa.

Las salidas de los microcontroladores están diseñadas de manera paralela a los pines, lo que posibilita una conexión directa y sencilla mediante jumpers hacia los dispositivos periféricos deseados. Estos dispositivos pueden incluir LEDs, switches y otros componentes utilizados en el proyecto. Esta disposición de las salidas facilita la interconexión entre el microcontrolador y los dispositivos externos, otorgando una mayor flexibilidad y versatilidad en la configuración del sistema. También pueden ser usados para conectar elementos externos de la placa.

En la *Figura 45* se observa la ubicación de estos pines en la placa.

Figura 45.
Salidas a los elementos en el Módulo Entrenador.



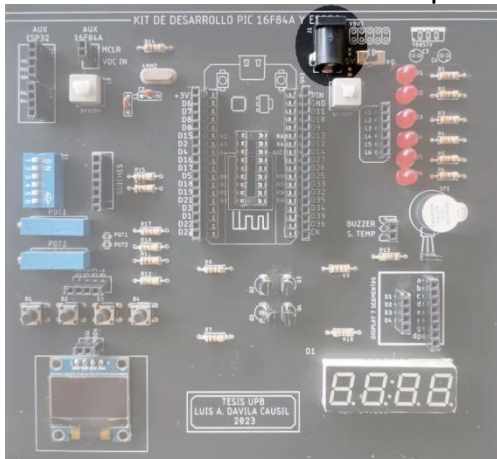
Nota: Imagen tomada por el autor.

3.1.3. Entrada de corriente directa (DC)

La tarjeta de desarrollo está diseñada para recibir una alimentación de 5V en corriente continua (DC) a través de un conector Jack en corriente directa de 2.5mm. Se establece una corriente máxima permitida de 2A y una corriente mínima de 1A. En el caso específico del ESP32, se alimenta a través del pin VIN, lo cual este JACK proporciona energía a este módulo por ese puerto. Por otro lado, en el microcontrolador 16F84A, la alimentación se realiza a través del pin VSS, suministrando energía al microcontrolador.

En la *Figura 46* se muestra la ubicación en la placa de desarrollo donde se encuentra el conector de alimentación, situado cerca de los bordes para facilitar su conexión externa y uso adecuado.

Figura 46.
Ubicación del JACK DC en la placa.



Nota: Imagen tomada por el autor.

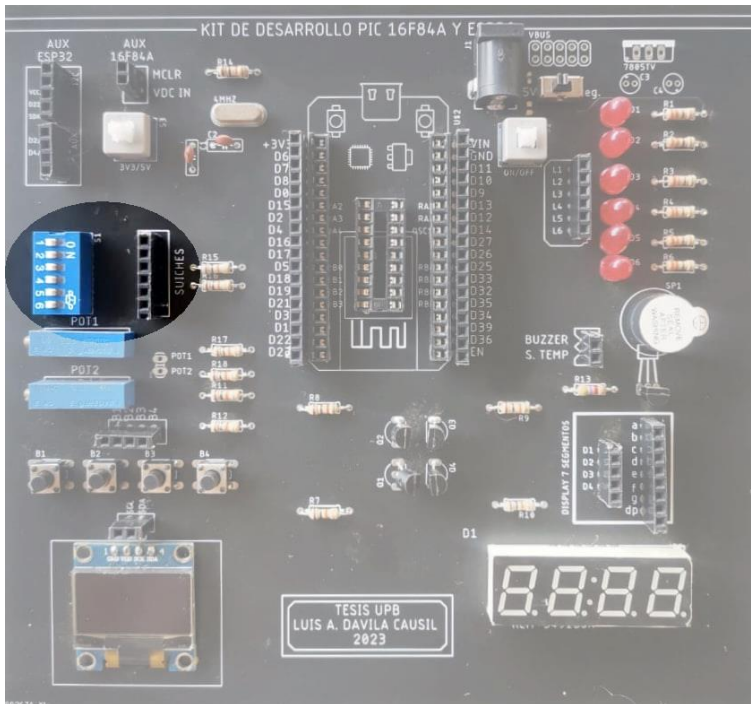
La correcta alimentación de la tarjeta es fundamental para el funcionamiento óptimo del microcontrolador y los demás componentes del sistema. La disposición estratégica del conector de alimentación en la placa de desarrollo garantiza una conexión segura y estable, evitando posibles problemas de suministro de energía.

3.1.4. Suiches

Los interruptores utilizados en este caso son conocidos como DIP-Switches, los cuales consisten en una serie de seis interruptores dispuestos en paralelo. Al activar o desactivar estos interruptores, generan una señal lógica de '0' o '1', dependiendo de su configuración. En el contexto de este proyecto, cuando el interruptor está en la posición de '1', se genera una señal de 0 voltios que representa un estado lógico '0'.

Estos DIP-Switches pueden ser conectados tanto al microcontrolador 16f84a como al ESP32. Los interruptores se pueden utilizar como elementos de control, permitiendo a los usuarios seleccionar opciones o configuraciones específicas en el sistema. Además, su disposición en paralelo facilita su conexión y configuración en el circuito, brindando flexibilidad en el diseño y adaptabilidad a las necesidades del proyecto, en la figura se observa su ubicación.

Figura 47.
DIP suiches



Nota: Imagen tomada por el autor.

La ubicación estratégica de los DIP-Switches en el circuito permite una fácil configuración y ajuste del sistema. Su utilización resulta especialmente útil en aplicaciones donde se requiere seleccionar opciones específicas o ajustar parámetros de funcionamiento. Gracias a la naturaleza de los DIP-Switches, la

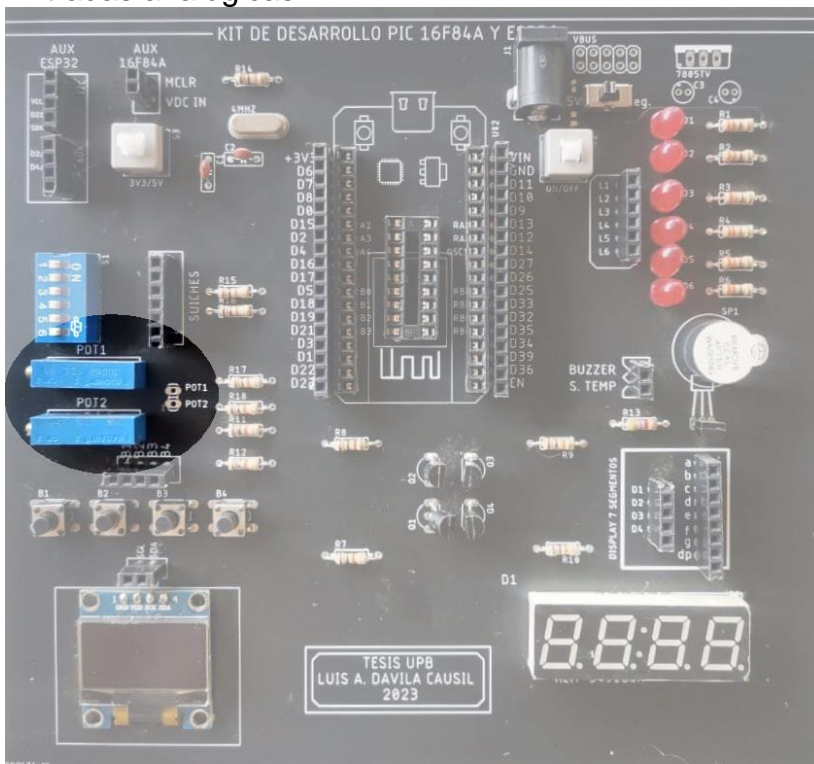
configuración es sencilla y se puede adaptar rápidamente según las necesidades del proyecto.

3.1.5. Entradas analógicas.

La placa cuenta con dos potenciómetros lineales, donde la resistencia varía de manera proporcional a la posición del cursor. Al ser conectados al ESP32 y al 16F84A, estos microcontroladores pueden leer el voltaje en el cursor y convertirlo en un valor digital. El ESP32, con una resolución de 12 bits, proporciona una mayor precisión en la lectura que el 16F84A, que tiene una resolución de 8 bits.

En la *Figura 48* se muestra la ubicación de los potenciómetros en la placa de desarrollo, lo cual permite ajustar manualmente su posición para controlar diferentes parámetros en un circuito. De esta manera, los potenciómetros se utilizan como una interfaz de entrada analógica para interactuar con el microcontrolador y ajustar valores de voltaje en tiempo real.

Figura 48.
Entradas analógicas.



Nota: Imagen tomada por el autor.

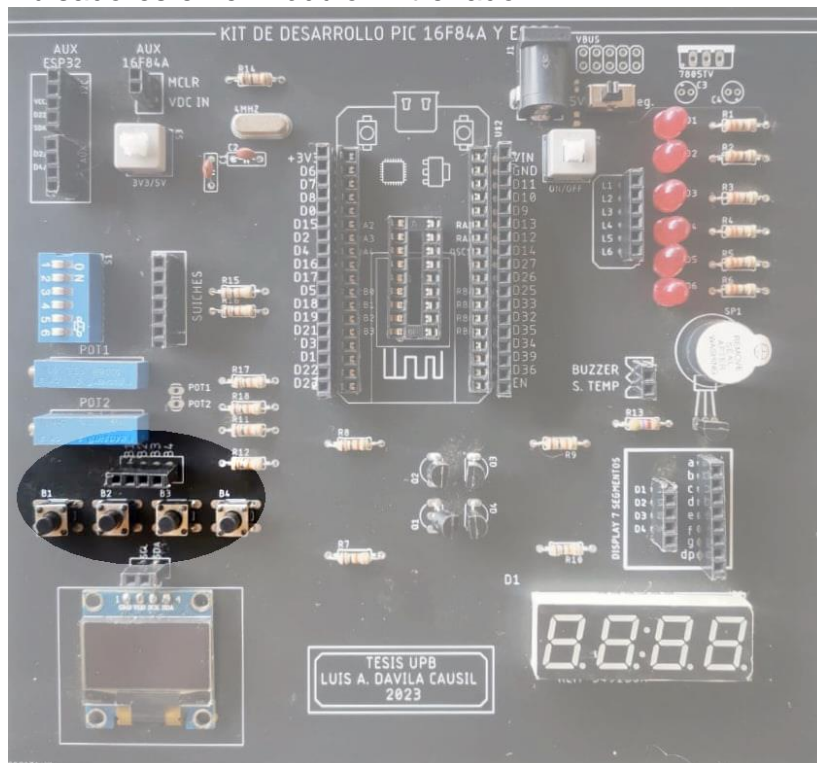
3.1.6. Pulsadores

En el caso específico de la placa de desarrollo, se usan pulsadores de 4 pines con una altura de 3mm, en el cual cuando los pulsadores son presionados, envían una señal de nivel lógico bajo, es decir, un valor de 0 voltios.

Es importante tener en cuenta que, para utilizar los pulsadores de manera efectiva, es necesario configurar internamente en el microcontrolador una resistencia pull-up virtual. Esta configuración garantiza que la señal se mantenga en un estado lógico alto (1) cuando el pulsador no está siendo presionado, y cambie a un estado lógico bajo (0) cuando se presiona. Los pulsadores pueden ser utilizados en diversas aplicaciones, como la detección de eventos, la interacción con el microcontrolador o la selección de opciones en un menú, entre otros. La ubicación de los pulsadores sobre el Módulo Entrenador se presenta en la *Figura 49*.

Figura 49.

Pulsadores en el Módulo Entrenador.



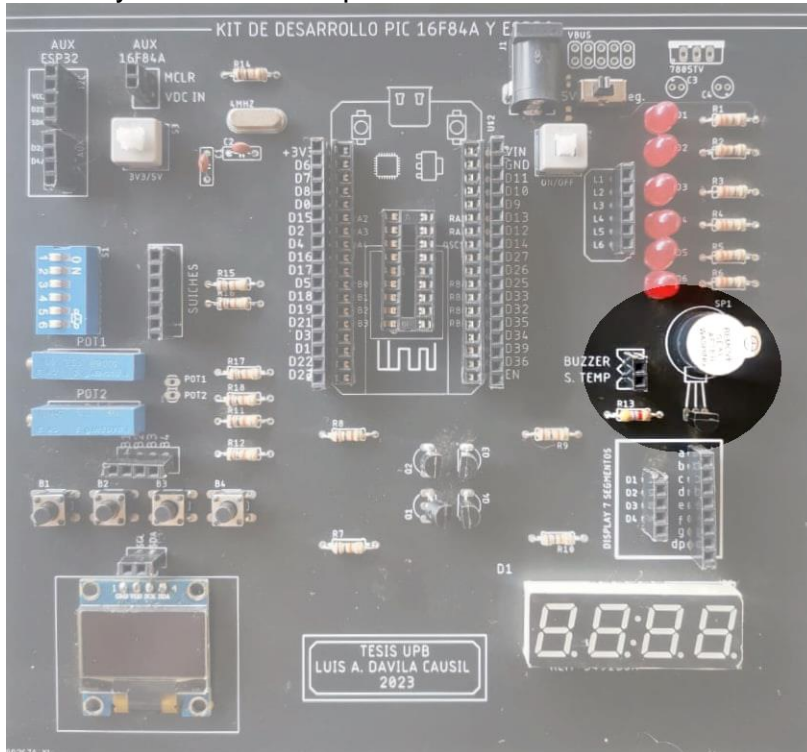
Nota: Imagen tomada por el autor.

3.1.7. LEDs

En el presente proyecto, los LEDs están conectados en serie con una resistencia de 330 ohmios para limitar el flujo de corriente y proteger los LEDs de posibles daños, por lo tanto, pueden ser utilizados como indicadores visuales en el sistema, permitiendo visualizar estados o señales específicas. La elección de la resistencia adecuada garantiza un flujo de corriente controlado, evitando daños en los LEDs y asegurando un funcionamiento óptimo.

En la *Figura 50* se observa la ubicación de los seis leds en paralelo con sus resistencias.

Figura 51.
Buzzer y sensor de temperatura.



Nota: Imagen tomada por el autor.

3.1.10. Display 7 segmentos.

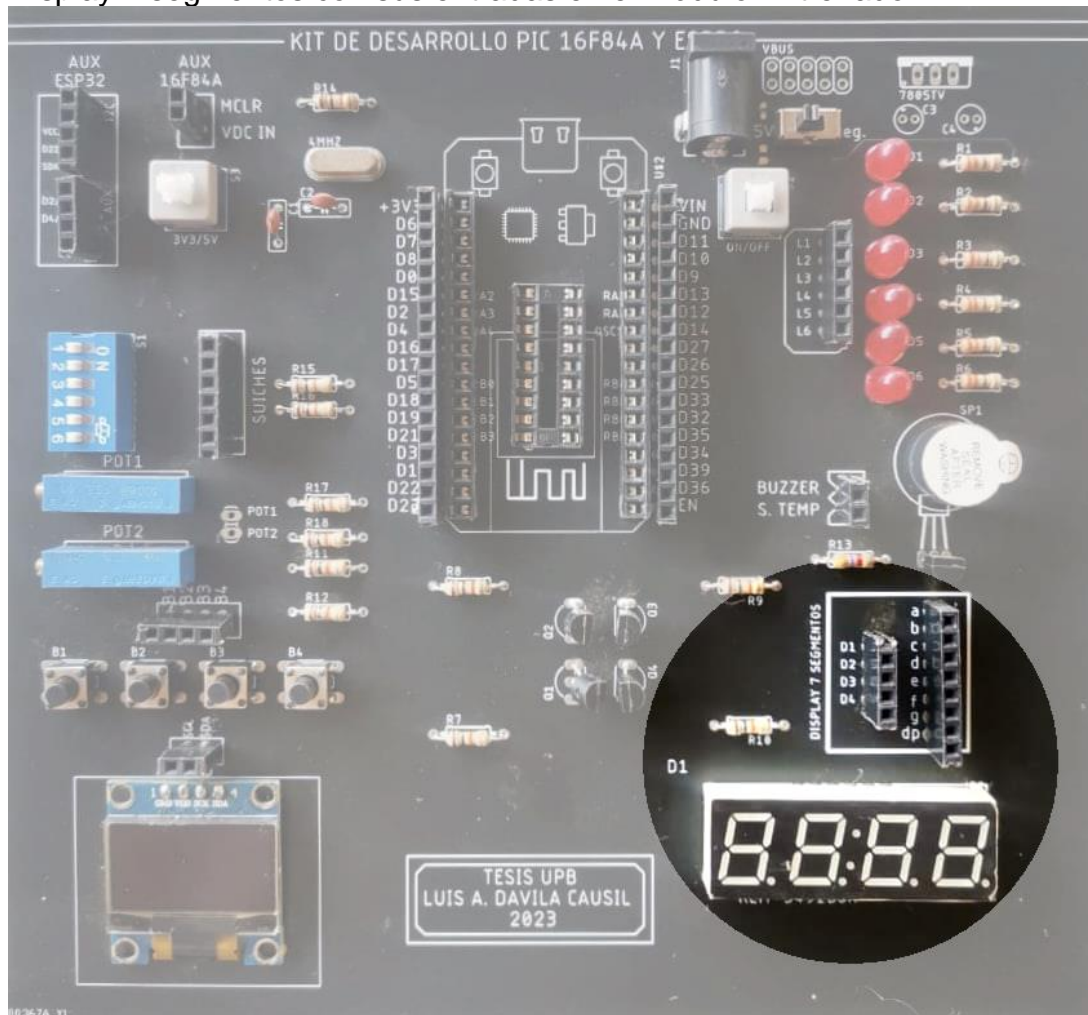
En este proyecto, la placa se encuentra un display 7 segmentos de 12 pines, con 4 dígitos, en el cual cada segmento está compuesto por un LED que se puede encender o apagar de forma independiente. Los dígitos se encuentran configurados en ánodo común, donde cada elemento puede encender de manera independiente. Se utilizan comúnmente para mostrar números y algunos caracteres alfanuméricos en dispositivos electrónicos como relojes digitales, calculadoras, termostatos, paneles de control, instrumentos de medición y otros sistemas de visualización. Debido a su simplicidad y claridad en la visualización, son ampliamente utilizados en aplicaciones donde se requiere una representación numérica o alfanumérica básica

La capacidad de multiplexación de este brinda a los estudiantes la oportunidad de adquirir conocimientos sobre circuitos digitales y comprender cómo se realiza la representación y visualización de datos. Estas aplicaciones no solo fomentan el aprendizaje, sino que también impulsan la creatividad y la experimentación, en la siguiente figura se observa su ubicación.

En la *Figura 52* se observa la ubicación del display 7 segmentos, junto con la entrada de los cuatro dígitos y los siete segmentos, más el control del punto dual.

Figura 52.

Display 7 segmentos con sus entradas en el Módulo Entrenador.



Nota: Imagen tomada por el autor.

3.1.11. Pantalla OLED.

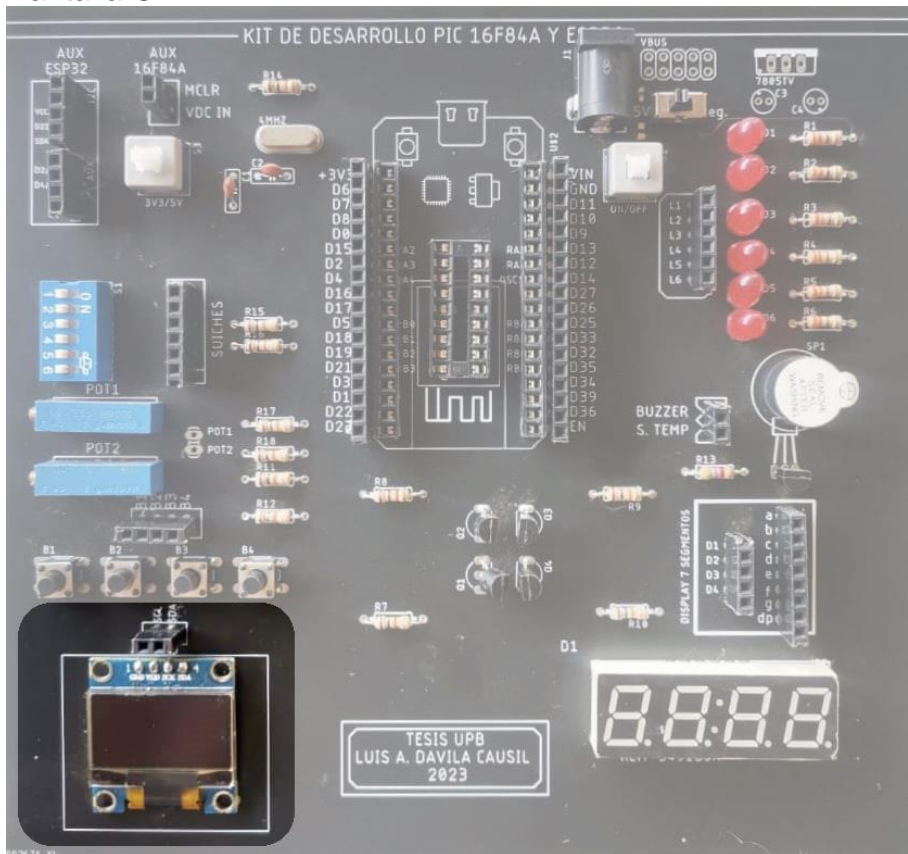
La pantalla OLED usada tiene una resolución de 128x64 bits, lo que se traduce en un total de 8192 píxeles que pueden ser utilizados para mostrar detalles visuales precisos mediante el protocolo I2C. La pantalla OLED ofrece la capacidad de representar textos, figuras e imágenes en una escala de dos colores, donde los píxeles encendidos representan una información específica y los píxeles apagados representan otra.

Estas características la hacen adecuada para una amplia gama de aplicaciones, como teléfonos móviles, tabletas, televisores y muchos otros dispositivos donde se requiere una representación visual de alta calidad. La versatilidad y calidad de imagen que ofrece la pantalla OLED la convierten en una elección popular en el campo de la tecnología de visualización.

En la *Figura 53* se observa la pantalla oled junto con sus dos entradas I2C (SDA y SCL).

Figura 53.

Pantalla OLED.



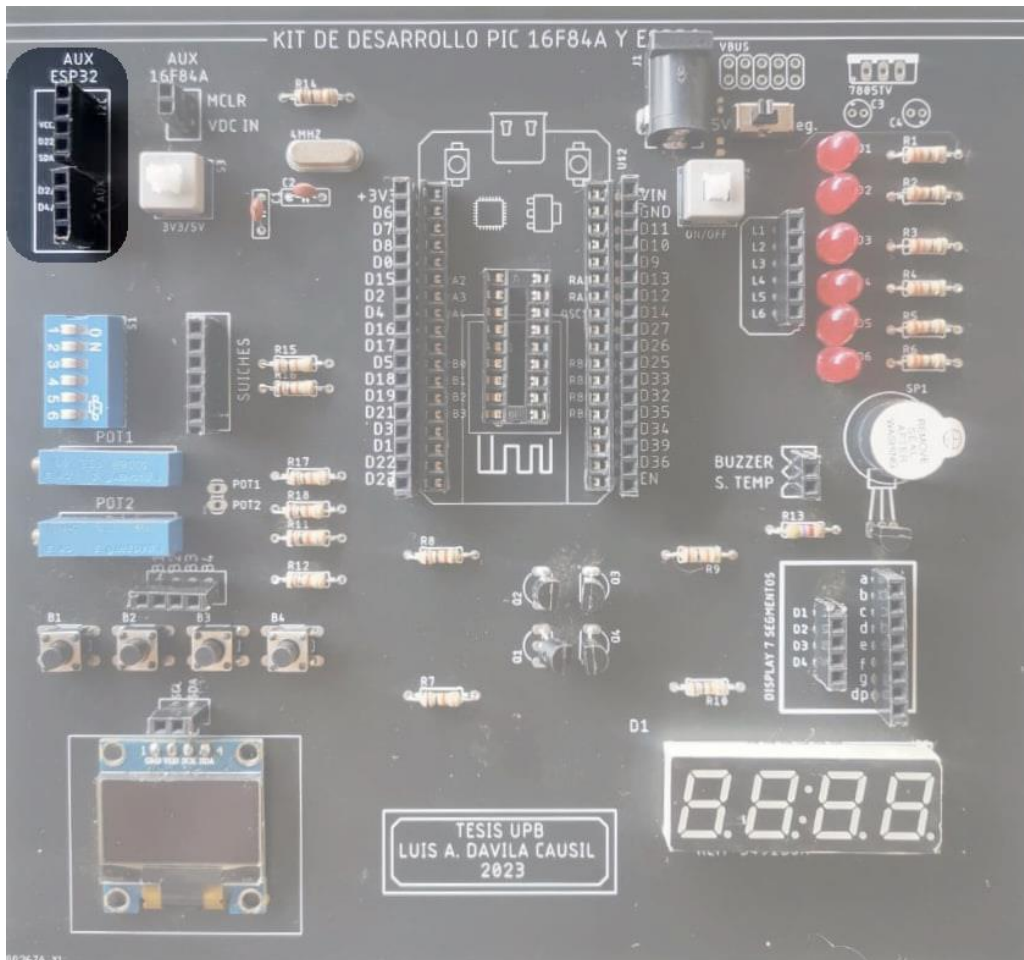
Nota: Imagen tomada por el autor.

3.1.12. Salidas auxiliares del ESP32.

Las salidas auxiliares del ESP32 se dividen en dos módulos distintos. Por un lado, se encuentra la salida I2C, la cual facilita la conexión de dispositivos externos a través de este protocolo de comunicación. Por otro lado, se encuentran dos pines DAC que permiten generar señales analógicas de salida, así como dos pines auxiliares adicionales que ofrecen mayor flexibilidad en la conectividad de componentes complementarios.

Estas salidas auxiliares del ESP32 amplían las capacidades del sistema, brindando opciones adicionales para la interacción con dispositivos externos y la generación de señales analógicas, lo que resulta fundamental en diversos proyectos y aplicaciones. En la Figura 54 se presenta la ubicación de las salidas auxiliares sobre el Módulo Entrenador.

Figura 54.
Salidas auxiliares del ESP32.

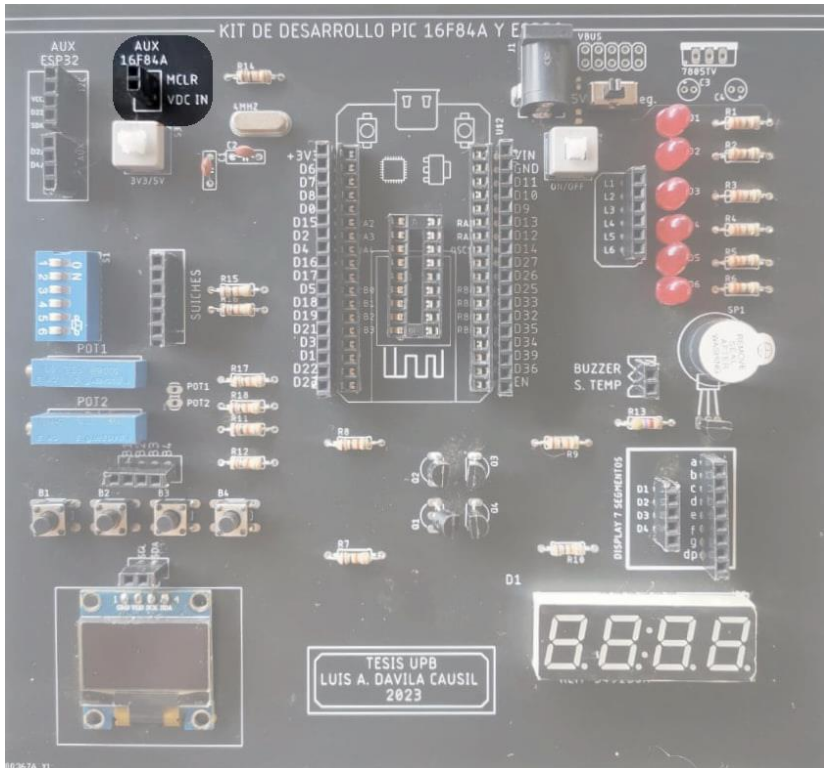


Nota: Imagen tomada por el autor.

3.1.13. Entradas auxiliares PIC 16F84A.

Las entradas auxiliares consisten en dos pines hembra que ofrecen la capacidad de conectar un pulsador al pin MCLR (Master Clear) del microcontrolador 16F84A. La entrada MCLR ya están configurada con una resistencia pull-up integrada, lo que permite mantener el nivel lógico alto cuando el pulsador no está presionado y así el 16F84A permanezca encendido. Los cinco voltios en su función como entrada, también puede funcionar como salida si se configura en el modo correspondiente. Estas entradas auxiliares proporcionan flexibilidad y versatilidad en el diseño del circuito, permitiendo la interacción con elementos externos de manera eficiente y segura. En la *Figura 55* se presenta la ubicación de las entradas auxiliares del PIC 16F84A sobre el Módulo Entrenador.

Figura 55.
Entradas auxiliares del PIC16F84A.



Nota: Imagen tomada por el autor.

3.1.14 Suiches de encendido y selección de voltaje.

Los interruptores de encendido cambian el estado de la placa entre "encendido" cuando se presionan y "apagado" cuando se liberan. De manera adicional, el interruptor de selección de voltaje de 5V actúa como una capa de seguridad adicional. Este interruptor permite elegir entre trabajar a 3.3V (en la posición "apagado") o 5V (en la posición "encendido") en conjunto con el ESP32 y los componentes de la placa. En contraste, en el caso del PIC 16F84A, este interruptor determina si se suministra energía a los dispositivos cuando está configurado en el modo 5V (en la posición "encendido") o si se desconecta la alimentación cuando se encuentra en la posición apagado.

3.2. Costos.

De acuerdo con los valores obtenidos de la Tabla del anexo A, se han identificado varios costos relevantes, considerando diferentes aspectos. En primer lugar, se ha determinado que el costo básico de los elementos utilizados en la etapa inicial del diseño, es decir, en las pruebas con la protoboard, la cual asciende a \$69.410 pesos colombianos. Por otro lado, al incluir los microcontroladores, se ha calculado un costo total de \$158.616 pesos colombianos.

En la siguiente fase del proceso, que implica la fabricación del circuito impreso (PCB), se han considerado diferentes opciones de impresión. A nivel local, en Montería, se ha obtenido un costo de \$60.000 pesos colombianos, con un tiempo de espera aproximado de 12 días. Por otro lado, a nivel internacional, con un tiempo de espera de 25 días, se ha encontrado un costo de \$35.000 pesos colombianos, lo

cual representa un ahorro significativo y la obtención de una PCB de mayor calidad y profesionalismo.

Considerando estos aspectos, el costo total del kit utilizado en el taller se estima en \$129.410 pesos colombianos. Es importante mencionar que esta cifra no incluye los microcontroladores, ya que la universidad cuenta con ellos. En caso de considerar los microcontroladores, el costo total del kit sería de \$218.616 pesos colombianos. Sin embargo, es posible adquirir algunos elementos a precios más económicos en tiendas especializadas, lo que podría reducir el costo total.

Si el objetivo es reducir aún más los costos, se puede optar por la tarjeta de menor precio, valorada en \$35.000 pesos colombianos, y prescindir de los microcontroladores disponibles en la universidad. De esta manera, el costo total se reduciría a \$104.410 pesos colombianos, lo cual resulta altamente asequible, considerando todas las funcionalidades y beneficios que ofrece el sistema, en la tabla 7 se observan los valores más relevantes.

Tabla 7
Costo neto del Módulo Entrenador

	Elementos	Microcontroladores	Tarjeta prototipo	Kit final	Kit + microcontroladores
Costos	\$69.410	\$89.206	\$129.410	\$104.410	\$193.616

En resumen, el análisis de costos realizado ha permitido identificar diferentes alternativas y consideraciones que influyen en el precio final de la placa. Esto proporciona información valiosa para la toma de decisiones y la optimización de recursos económicos, teniendo en cuenta las necesidades y posibilidades de los usuarios interesados en utilizar este sistema, se concluye teniendo en cuenta que la universidad cuenta con los microcontroladores, que el costo total del kit es de \$104.410 pesos colombianos, un precio muy asequible y cumple con los requerimientos.

4. MANUAL DE USUARIO.

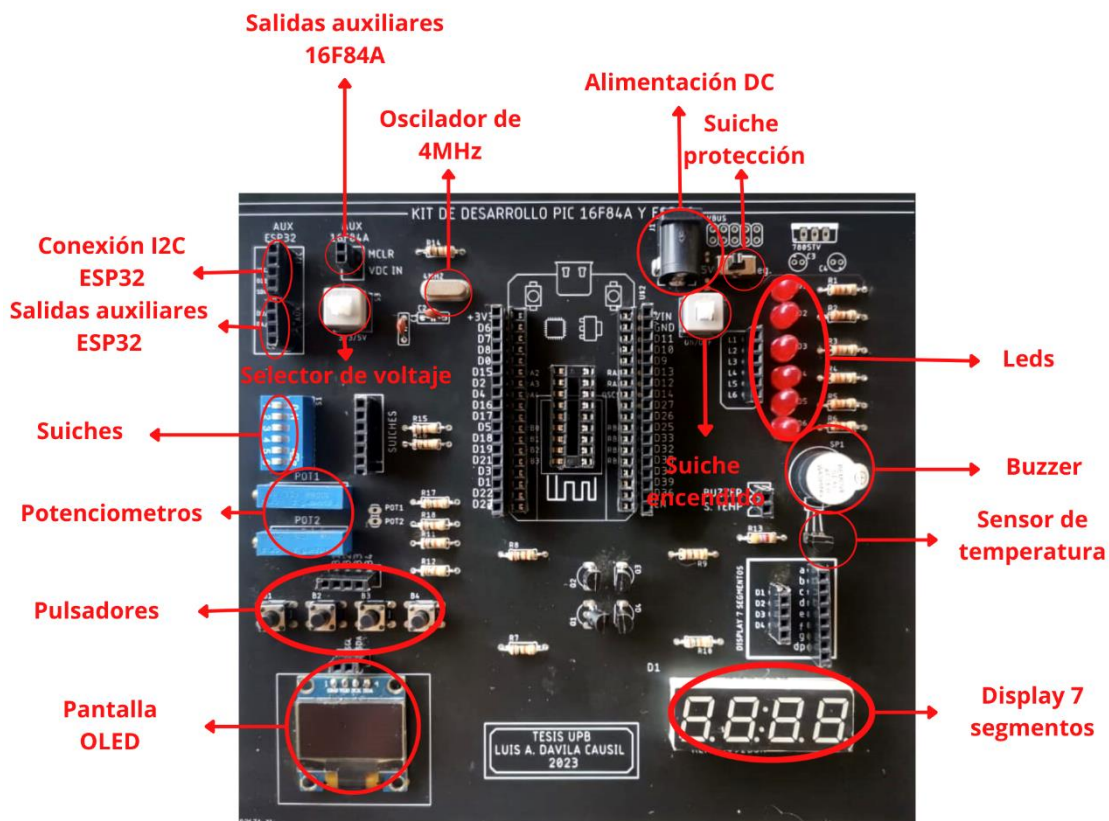
El presente manual de usuario se ha elaborado con el propósito de proporcionar a los estudiantes las herramientas necesarias para llevar a cabo proyectos utilizando el Módulo Entrenador. Su objetivo principal es minimizar posibles errores y dificultades que puedan surgir durante la utilización del sistema.

A través de este manual, los estudiantes encontrarán una guía detallada que les permitirá comprender y utilizar de manera eficiente todas las funcionalidades y componentes del kit. Se brindarán instrucciones paso a paso, ejemplos prácticos y recomendaciones relevantes para garantizar un proceso fluido y exitoso en el desarrollo de sus proyectos.

4.1. Partes del Módulo Entrenador

Antes de abordar los proyectos, se debe tener en cuenta las partes que conforman el kit y accesorios necesarios para su funcionamiento, los cuales se presentan en la *Figura 56*.

Figura 56.

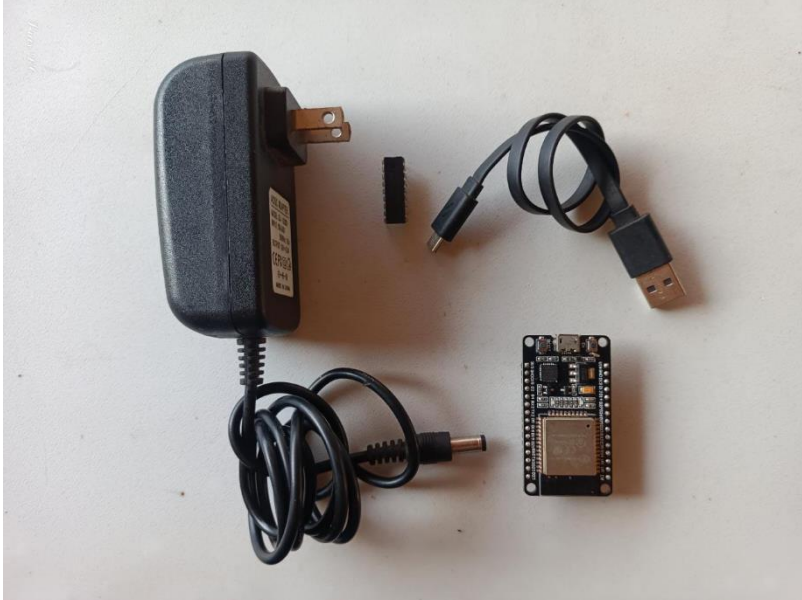


Nota: Imagen hecha por el autor.

A parte de la placa de desarrollo para el funcionamiento del módulo entrenador son necesarios los microcontroladores, cables y adaptadores para su funcionamiento, en la Figura 57 se observan los elementos.

Figura 57.

Elementos para el funcionamiento del Módulo Entrenador.



Nota: Imagen tomada por el autor.

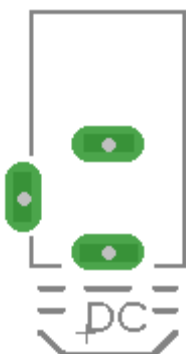
A continuación, se detallan los elementos utilizados en el Módulo Entrenador y las instrucciones para su conexión. Es importante tener en cuenta que los pines header deben ser conectados al microcontrolador mediante jumpers o cables. En este caso, se recomienda utilizar los pines header para una conexión más segura y estable.

4.1.1. Alimentación DC

Para la alimentación DC se tiene un JACK hembra de 2.5mm como se observa en la Figura 58, el cual se alimenta mediante el cargador AC/DC.

Figura 58.

Vista del elemento en EAGLE.



Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 59* se observa la conexión de los pines del Jack con los elementos del circuito, donde van a VIN en el caso del ESP32 y a VSS en el caso del PIC16F84A.

Figura 59.

Vista del esquemático en EAGLE.



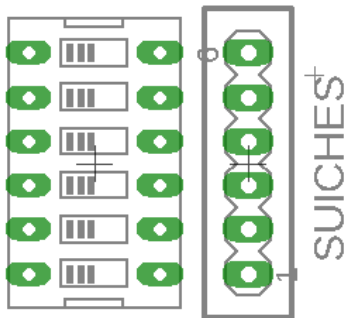
Nota: Imagen hecha por el autor en EAGLE.

4.1.2. Suiches

Son entradas digitales que permanecen fijos, sus conexiones son a los pin header llamado 'SUICHES', maquinados del 1 al 6 y son paralelos a sus suiches aledaños, como se observa en la *Figura 60*.

Figura 60.

Vista del elemento en EAGLE.

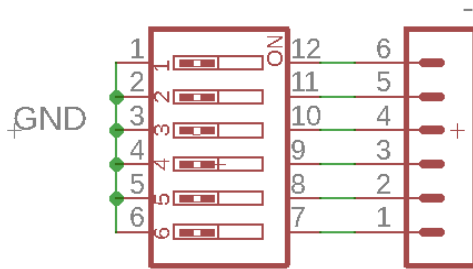


Nota: Imagen hecha por el autor en EAGLE.

En el esquemático se observa la conexión específica de los suiches, donde son conectados a tierra (ver *Figura 61*), es decir, para usarlos de deben configurar los pines como Pull-up en el microcontrolador.

Figura 61.

Vista del esquemático en EAGLE.



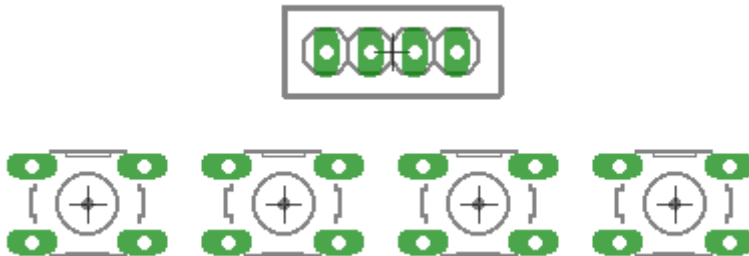
Nota: Imagen hecha por el autor en EAGLE.

Pulsadores

Son entradas digitales que se activan momentáneamente, en este caso al activarse envían cero voltios. Los cuatro pulsadores están ubicados de forma paralela a los pin headers como se observa en la *Figura 62*.

Figura 62.

Vista del elemento en EAGLE.

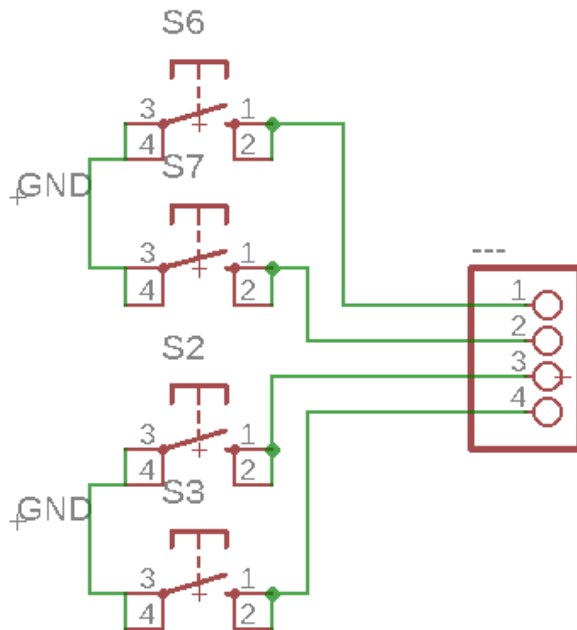


Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 63* se observa el esquemático los pulsadores, donse se observa que son activos en bajo, por lo tanto para usarse se debe activar el pull-up en los pines a usar.

Figura 63.

Vista del esquemático en EAGLE.



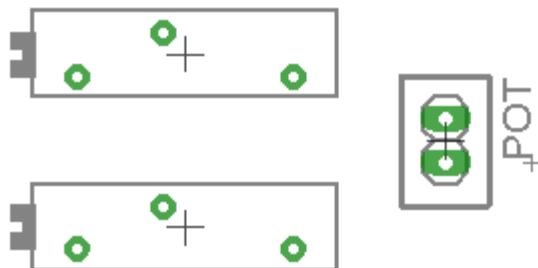
Nota: Imagen hecha por el autor en EAGLE.

4.1.3. Entradas analógicas

La placa cuenta con dos potenciómetros lineales en modo trimmer, están ubicados paralelamente a los header (ver *Figura 64*), en su funcionamiento varían el voltaje de 0v a 5v o 0v a 3.3 voltios, dependiendo del micro a usar.

Figura 64.

Vista del elemento en EAGLE.

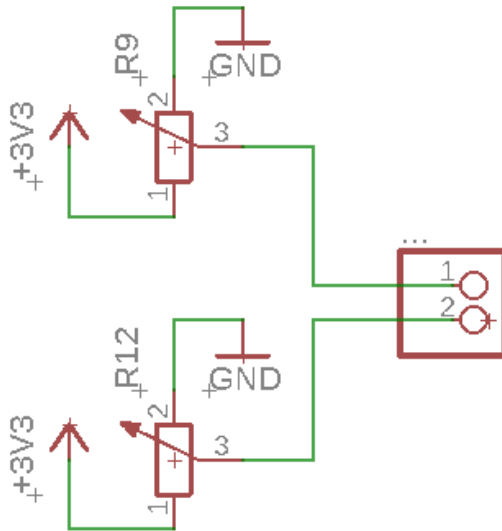


Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 65* se presenta el esquématico, depende de microcontrolador a usar como el PIC16F84A o el ESP32 puede representar en voltios los 256 niveles de resolución o 4096 niveles respectivamente.

Figura 65.

Vista del esquemático en EAGLE.



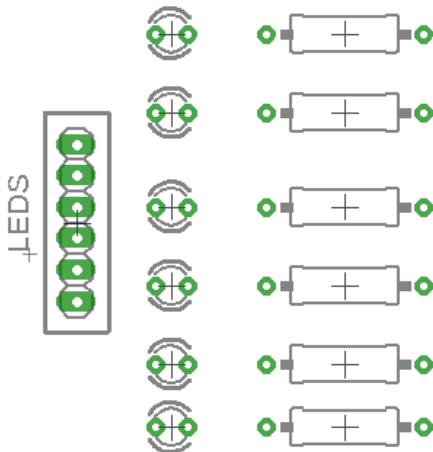
Nota: Imagen hecha por el autor en EAGLE.

4.1.4. LEDS

Los LEDS permiten visualizar datos digitales, estos están conectados horizontalmente con sus pin header y resistencias de 330ohm como se observa en la *Figura 66*.

Figura 66.

Vista del elemento en EAGLE.

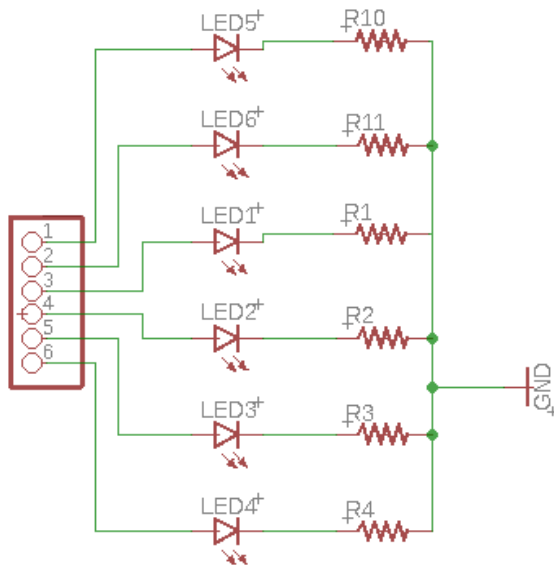


Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 67* se observa el esquemático, para encenderlos basta con enviar un voltaje, también se puede modificar su atenuación con un PWM.

Figura 67.

Vista del esquemático en EAGLE.



Nota: Imagen hecha por el autor en EAGLE.

4.1.5. Buzzer y sensor de temperatura.

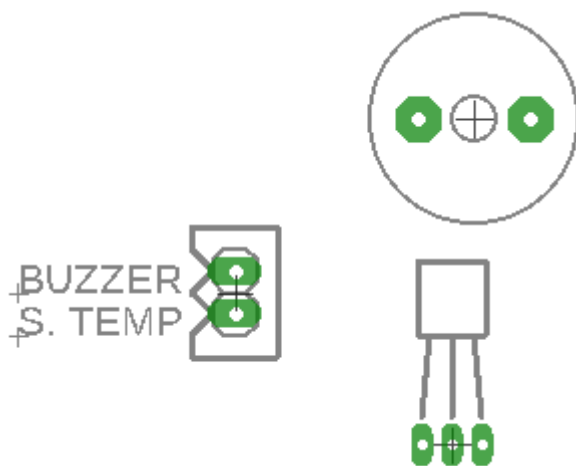
El buzzer permite reproducir sonidos audibles, su conexión es de forma directa y dependiendo del voltaje o frecuencia puede reproducir un sonido audible.

El sensor de temperatura digital es el DS18B20, su conexión es con el pin header hembra 'S. TEMP', para usarlo se debe recibir la información mediante la comunicación serial ONEWIRE, la cual es unidireccional.

En la *Figura 68* se presenta la parte visual del sensor de temperatura junto con el Buzzer.

Figura 68.

Vista de los elementos en EAGLE.

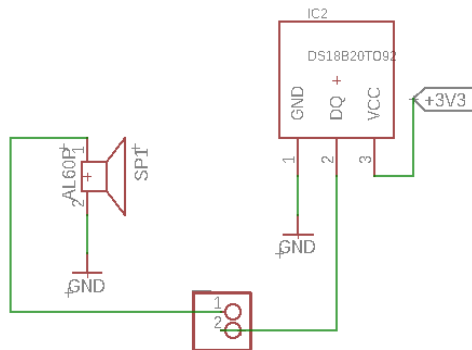


Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 69* se observa el esquemático, donde el buzzer se encuentra a la izquierda y el sensor de temperatura en el lado derecho.

Figura 69.

Vista del esquemático en EAGLE.



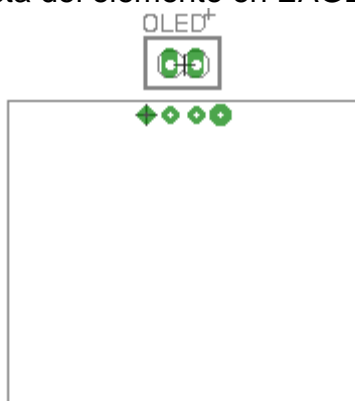
Nota: Imagen hecha por el autor en EAGLE.

4.1.6. Pantalla OLED

La pantalla oled está conectada a dos pines, los cuales son SCL y SDA del protocolo I2C, funciona tanto a 3.3v como a 5v, en la figura se observa el sensor de temperatura junto con su pin header de entradas, a la izquierda el SCL y a la derecha el SDA. En la *Figura 70* se presenta la capa superior de la pantalla OLED en EAGLE.

Figura 70.

Vista del elemento en EAGLE.

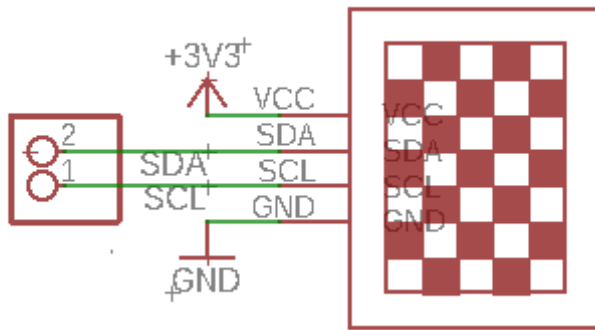


Nota: Imagen hecha por el autor en EAGLE.

En la *Figura 71* se observan las conexiones de la pantalla OLED con los puertos I2C.

Figura 71.

Vista del esquemático en EAGLE.



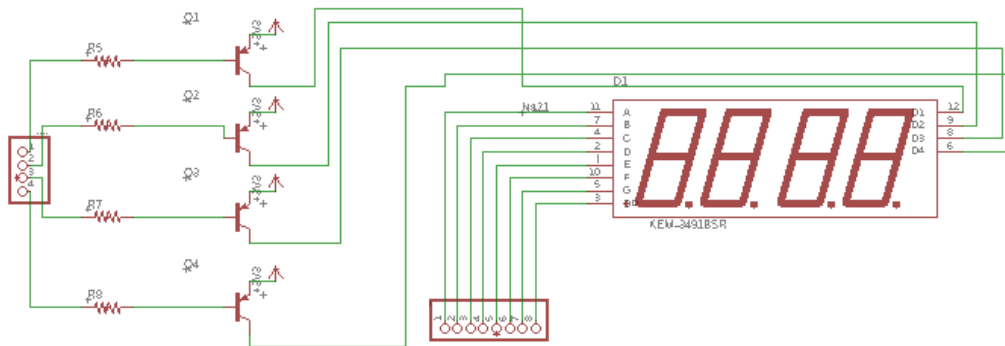
Nota: Imagen hecha por el autor en EAGLE.

4.1.7. Display 7 segmentos.

El display 7 segmentos de la placa es Ánodo Común, es decir, para encender los segmentos se deben colocar en bajo, y para encender un dígito en alto, en este caso como son transistores PNP se deben colocar en bajo para encender el dígito, ya que en alto se apagan los transistores. En la *Figura 72* se presenta el esquemático y diagrama de conexiones en EAGLE.

Figura 72.

Vista del esquemático en EAGLE.



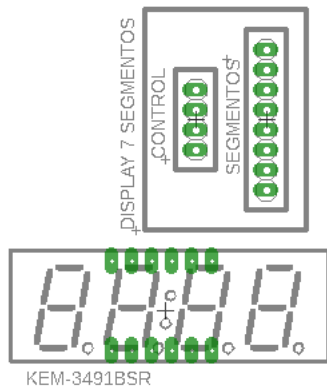
Nota: Imagen hecha por el autor en EAGLE.

En la parte de la PCB, se ha diseñado el esquema de control de manera que los dígitos de la pantalla estén ubicados en la parte de control de la placa. Esto facilita la visualización y lectura de los dígitos, ya que se encuentran en una posición de mayor visibilidad.

Asimismo, los segmentos que componen cada dígito están dispuestos de forma secuencial, de manera que el último segmento corresponde a los dos puntos. Esto sigue la convención comúnmente utilizada en las pantallas 7 segmentos, donde los dos puntos se encuentran en la posición más baja del dígito. En la *Figura 73* se presenta la vista en EAGLE.

Figura 73.

Vista del elemento en EAGLE.



Nota: Imagen hecha por el autor en EAGLE.

4.2 Manual guía del Módulo Entrenador ESP32 Y PIC16F84A

En esta guía se explorará el manejo del Módulo Entrenador en conjunción con dos microcontroladores altamente versátiles: el PIC16F84A y el ESP32. Se abordarán conceptos fundamentales y prácticos para operar y programar estos dispositivos, permitiendo a los usuarios desarrollar habilidades esenciales en la interacción hardware-software.

4.2.1 Programación del PIC16F84A

La programación de un microcontrolador PIC16F84A en lenguaje C es un proceso fundamental para desarrollar aplicaciones embebidas. Esta guía proporciona una visión general de cómo programar el PIC16F84A en lenguaje C, desde la configuración del entorno de desarrollo hasta la carga del programa en el microcontrolador.

4.2.1.1 Configuración del Entorno de Desarrollo

Antes de comenzar, es esencial configurar el entorno de desarrollo. Esto incluye la instalación del compilador C adecuado y de las herramientas de programación.

- Se descarga e instala el compilador XC8 de Microchip, que es una herramienta comúnmente utilizada para programar microcontroladores PIC.
- Se utiliza el software MPLAB X IDE, proporcionado por Microchip, como entorno de desarrollo para escribir y compilar el código en lenguaje C.

4.2.1.2 Creación de un Nuevo Proyecto

Para iniciar un proyecto de programación en C para el PIC16F84A, se deben seguir estos pasos:

- Se abre el MPLAB X IDE y se selecciona "File" > "New Project".
- Se elige la opción "Standalone Project" y se continúa.
- Se selecciona el dispositivo "PIC16F84A" como el objetivo del proyecto y se establece el nombre y la ubicación del proyecto.

4.2.1.3 Programación en C

Una vez que se haya configurado el proyecto, se puede proceder a escribir el código en lenguaje C que se desea cargar en el PIC16F84A.

- En la sección "main.c" del proyecto, se debe escribir el código C necesario para realizar las operaciones deseadas por el microcontrolador.
- Se deben utilizar funciones y librerías específicas para interactuar con los periféricos y sensores según los requerimientos del proyecto.

4.2.1.4 Compilación

- Se compila el código C seleccionando "Build" > "Build Project". Esto generará un archivo de código objeto (.hex) listo para ser programado en el PIC16F84A.

4.2.1.5 Programación del PIC16F84A

Para cargar el programa en el microcontrolador, se deben seguir estos pasos:

- Se conecta un programador compatible al PIC16F84A y al ordenador.
- En MPLAB X IDE, se selecciona el programador y el dispositivo objetivo.
- Se hace clic en "Programmer" > "Program" para cargar el archivo .hex en el PIC16F84A.
- Se verifica que la programación sea exitosa, lo que indicará que el microcontrolador está listo para funcionar según lo programado.

4.2.2 Programación del ESP32 con el IDE Arduino.

La programación de un ESP32 con el IDE Arduino es un proceso esencial para desarrollar proyectos de IoT y aplicaciones embebidas. Esta guía proporciona una descripción paso a paso, desde la configuración del entorno de desarrollo hasta antes de comenzar a escribir código.

4.2.2.1 Configuración del Entorno

Antes de programar el ESP32, es necesario configurar el entorno de desarrollo. A continuación, se describen los pasos a seguir:

- **Descargar del Arduino IDE:** En primer lugar, se debe descargar e instalar el Arduino IDE desde el sitio web oficial de Arduino.

4.2.2.2 Configuración del Arduino para ESP32

La configuración adecuada del Arduino IDE es fundamental para programar el ESP32. Estos son los pasos:

- **Agregar el Soporte para ESP32:** En el Arduino IDE, ve a "Archivo" > "Preferencias". En el campo "URLs Adicionales de Tarjetas", se debe ingresar la siguiente URL: https://dl.espressif.com/dl/package_esp32_index.json y hacer clic en "OK".
- **Instalación del Soporte para ESP32:** Posteriormente, se debe ir a "Herramientas" > "Tarjeta" > "Gestor de tarjetas". En el cuadro de búsqueda, se debe ingresar "esp32" y seleccionar "ESP32 by Espressif Systems". A continuación, se debe hacer clic en "Instalar".
- **Selección de la Tarjeta ESP32:** En "Herramientas" > "Tarjeta", se debe elegir "ESP32 Dev Module" como la tarjeta objetivo.
- **Configuración del Puerto:** En "Herramientas" > "Puerto", se debe seleccionar el puerto COM al que está conectado el ESP32.

A partir de este punto, se puede empezar a desarrollar proyectos, programar el ESP32 y utilizar las librerías necesarias para alcanzar los objetivos específicos del proyecto.

4.2.3 Librerías para el sensor de temperatura y pantalla OLED.

Para aprovechar plenamente las capacidades del Módulo Entrenador, es esencial incorporar diversas librerías que habilitan la interacción con sus componentes. A continuación, se detallan las librerías necesarias para emplear algunos de estos elementos.

4.2.3.1 Sensor de Temperatura DS18B20

Para utilizar el sensor de temperatura DS18B20 con éxito, se deben incorporar dos librerías fundamentales: "OneWire" y "DallasTemperature". Estas librerías permiten la comunicación con el sensor y simplifican la lectura precisa de las temperaturas. A continuación, se detalla cómo agregar y utilizar estas librerías:

- **Librería OneWire:**

- La librería "OneWire" facilita la comunicación con dispositivos que utilizan el protocolo OneWire, como el sensor DS18B20. Para agregar esta librería, sigue estos pasos:
 - Abrir el Arduino IDE.
 - Ir a "Sketch" > "Incluir Biblioteca" > "Gestionar Bibliotecas".
 - En el cuadro de búsqueda, ingresar "OneWire".
 - Seleccionar la librería "OneWire" de Paul Stoffregen y haz clic en "Instalar".
- **Librería DallasTemperature:**
 - La librería "DallasTemperature" simplifica la gestión de múltiples sensores de temperatura DS18B20 en un mismo bus OneWire. Para agregar esta librería, realiza lo siguiente:
 - Abre el Arduino IDE.
 - Ve a "Sketch" > "Incluir Biblioteca" > "Gestionar Bibliotecas".
 - En el cuadro de búsqueda, ingresa "DallasTemperature".
 - Selecciona la librería "DallasTemperature" de Miles Burton y haz clic en "Instalar".

Estas librerías combinadas permiten la comunicación efectiva con los sensores DS18B20 y facilitan la lectura precisa de las temperaturas. Una vez que se han agregado estas librerías al proyecto, se pueden utilizar en el código para acceder y gestionar los datos del sensor DS18B20. Es importante verificar que las conexiones del sensor se realicen correctamente y que se configure el bus OneWire según las especificaciones de hardware.

4.2.3.2 Pantalla OLED

Para interactuar con una pantalla OLED en el Módulo Entrenador, se deben incorporar las librerías "Adafruit_SSD1306" y "Adafruit_GFX". Estas librerías permiten controlar y mostrar información en pantallas OLED compatibles con el controlador SSD1306, que es ampliamente utilizado en diversas pantallas OLED. Al combinar estas dos librerías, se puede crear una experiencia de visualización rica y personalizada en la pantalla OLED, incluyendo texto, gráficos y otros elementos visuales.

La librería "Adafruit_SSD1306" se encarga de la comunicación y el control de la pantalla OLED, mientras que la librería "Adafruit_GFX" proporciona herramientas para el dibujo y la representación de gráficos en la pantalla. Juntas, estas librerías ofrecen una plataforma sólida para diseñar interfaces de usuario interactivas y visualmente atractivas en el entorno del Módulo Entrenador.

Al agregar estas librerías al proyecto, se podrán aprovechar al máximo las capacidades de la pantalla OLED, lo que permitirá mostrar datos de manera efectiva y crear interfaces de usuario dinámicas para una variedad de aplicaciones en el Módulo Entrenador.

4.2.4 Consideraciones al momento de conectar los microcontroladores.

Conectar apropiadamente el microcontrolador PIC16F84A y ESP32 es esencial para garantizar su funcionamiento y prevenir daños. Los interruptores de alimentación y selector de voltaje desempeñan un papel fundamental en este

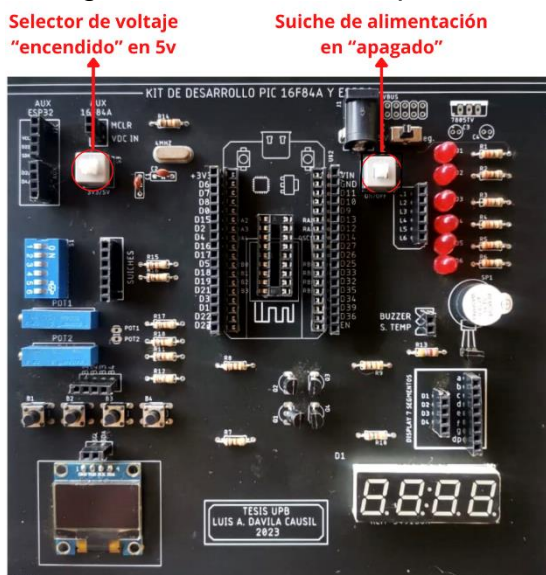
proceso. A continuación, se explican los pasos para lograr una conexión segura y eficaz:

- **Interruptor de Alimentación:** El interruptor de alimentación permite controlar la alimentación del PIC16F84A y otros elementos del sistema. Al encender el interruptor, se suministra la alimentación (5V) al microcontrolador y a los componentes del módulo entrenador. Al apagar el interruptor, se desconecta la alimentación. Asegúrate de que el interruptor esté en la posición adecuada antes de aplicar la alimentación al sistema.
- **Interruptor de Selector de Voltaje:** El interruptor de selector de voltaje permite alternar entre dos opciones de voltaje, en este caso, 5V y 3.3V. Esto es especialmente importante cuando se utiliza un microcontrolador como el ESP32, que requiere 3.3V en lugar de 5V. Si el interruptor de selector está configurado en 5V, asegúrate de que el interruptor de alimentación esté apagado. Esta configuración garantiza que la alimentación esté desconectada, lo que es esencial para evitar cortocircuitos y asegurar una programación segura.

En la *Figura 74* se muestra la configuración precisa de los interruptores para cargar un programa en el ESP32 cuando se encuentra montado en el Módulo Entrenador. En esta configuración, el interruptor de alimentación se encuentra en la posición "apagado", mientras que el interruptor de selección de voltaje está en "encendido", esta configuración específica evita cortocircuitos por el Módulo Entrenador en el ESP32.

Figura 74.

Configuración de los interruptores.



Nota: Imagen hecha por el autor.

5. CONCLUSIONES

En cuanto al diseño de PCBs, se ha observado que existen numerosas librerías disponibles en EAGLE, aunque la mayoría son creadas por personas particulares. Es importante tener en cuenta estas librerías, ya que proporcionan conocimientos valiosos para futuros diseños y fabricación de PCBs. Además, la impresión de PCBs resulta muy útil, ya que facilita la producción de prototipos y la corrección de errores para obtener módulos definitivos.

En el ámbito local, la realización de PCBs puede resultar complicada debido a la falta de servicios especializados. Por lo tanto, es necesario optar por prototipos de una capa para poder realizar múltiples iteraciones y corregir errores en los módulos definitivos del Módulo Entrenador.

El Módulo Entrenador ha sido diseñado con todos los elementos necesarios para su correcto funcionamiento. Incluye elementos clásicos como suiches, LEDs y potenciómetros, así como componentes innovadores como la pantalla OLED, que llaman la atención de la comunidad académica y fomentan el aprendizaje de manera efectiva.

Se ha observado que los montajes en el Módulo Entrenador son mucho más ágiles en comparación con el uso de protoboards, como revelan las encuestas realizadas. Este resultado es de gran importancia, ya que permite a los estudiantes realizar prácticas de manera más eficiente y comprender el funcionamiento de los circuitos de forma adecuada, proporcionándoles herramientas importantes para su desarrollo académico.

El uso del IDE de Arduino con el ESP32 resulta muy beneficioso en la programación del microcontrolador. Sin embargo, es importante destacar la importancia de trabajar con protocolos e interfaces ofrecidos por los microcontroladores, como interrupciones, configuraciones PULL-UP, PWM e interfaces I2C, para seguir buenas prácticas de programación.

En cuanto a los costos, se ha logrado cumplir con el objetivo de ofrecer un Módulo Entrenador a bajo costo, lo que lo hace muy accesible. Al tratarse de una capa, las personas interesadas en obtener uno pueden fabricarlo por sí mismas, lo que proporciona una opción económica y viable.

Finalmente, se ha comprobado que el circuito es compatible tanto con el PIC16F84A como con el ESP32 de 30 y 36 pines, asegurando que todos los elementos son compatibles con los voltajes de ambos microcontroladores (5v y 3.3v, respectivamente). Esto lo convierte en un componente crucial para el curso de Sistemas Digitales Microcontrolados y otras disciplinas relacionadas con los microcontroladores. Además, el diseño modular permite adaptar nuevos elementos en el futuro, evitando que el sistema se vuelva obsoleto y brindando una alternativa valiosa a la Universidad Pontificia Bolivariana Seccional Montería.

6. Bibliografía.

- Arroyave Zuluaga, L. A., & Mosquera Rentería, A. de J. (2015). *Diseño e implementación de un kit para robótica educativa*. Corporación Universitaria Minuto de Dios.
- Barcia, X., Caller, R., & Valdivieso, C. (2008). *Diseño de tarjetas para el desarrollo de aplicaciones con dsPIC*. Escuela Superior Politécnica Del Litoral.
- Castillo, J. A. (2019). *Qué es una PCB o Placa de Circuito Impreso. Uso, cómo se fabrica*. <https://www.profesionalreview.com/2019/02/11/pcb-que-es/>
- Concepción, R. (2019). *Tarjetas de Desarrollo (Episodio #8)*.
<https://www.rjconcepcion.com/podcast/tarjetas-de-desarrollo-episodio-8/>
- Digikey. (s. f.). *Herramientas de desarrollo Explorer 16/32*.
<https://www.digikey.com/es/product-highlight/m/microchip-technology/explorer-16-32-development-board>
- Electrónica I+D. (2021). Sistema de desarrollo para microcontroladores.
<https://didacticaselectronicas.com/index.php/sistemas-de-desarrollo/otros-sistemas/tarjetas-desarrollo/sistema-de-desarrollo-para-microcontroladores-detail>
- Gonzales Rojas, K. V., & Caviativa Mendez, J. A. (2011). *Diseño y fabricación de un módulo entrenador implementando la familia de microcontroladores dsPIC33F*.
- Hernandez, S. (2021). *¿Qué es un microcontrolador?*
<http://sherlin.xbot.es/microcontroladores/>
- Lemus, I. (2019). *¿Qué es una tarjeta de desarrollo microcontrolador para IOT?*
<https://conocimientolibre.mx/microcontrolador/>

Marmolejo, R. (2017). *Microcontrolador – qué es y para que sirve*. <https://hetpro-store.com/TUTORIALES/microcontrolador/>

Microchip. (2023). Datasheet PIC16f84A. <https://pdf1.alldatasheet.com/datasheet-pdf/view/74979/MICROCHIP/PIC16F84A.html>

Sanchez Naranjo, J. A. (2020). *Desarrollo de un kit modular de bajo costo, que facilite la realización de prácticas de física en colegios*. Universidad Autónoma De Occidente.

Seedstudio, (2022). Grove Beginner Kit For Arduino. <https://wiki.seeedstudio.com/Grove-Beginner-Kit-For-Arduino/>

ANEXO A.
TABLA DE PRECIOS DE LOS ELEMENTOS USADOS CON EL MÓDULO ENTRENADOR.

<i>CANTIDAD</i>	<i>ELEMENTO</i>	<i>PRECIO</i>	<i>TOTAL</i>
4	Pin header 2 pines	\$ 620	\$ 2.480
4	Pin header 4 pines	\$ 620	\$ 2.480
2	Pin header 6 pines	\$ 620	\$ 1.240
1	Pin header 8 pines	\$ 720	\$ 720
4	Pin header 18 pines	\$ 1.268	\$ 5.072
1	Base para C.I. 16 pines	\$ 320	\$ 320
1	Dip-switch 6 pines	\$ 1.023	\$ 1.023
2	Potenciometro lineal 5k	\$ 1.500	\$ 3.000
4	Pulsador 4 pines	\$ 649	\$ 2.596
1	Oled 0.96"	\$ 24.633	\$ 24.633
1	Jack DC 2.5mm	\$ 537	\$ 537
1	Convertidor AC/DC 5V 1A	\$ 7.200	\$ 7.200
1	Display 7 segmentos A.C. 12 pin	\$ 4.751	\$ 4.751
1	Buzzer activo voltaje variable	\$ 1.428	\$ 1.428
5	Resistencias 10kohm	\$ 37	\$ 185
2	Capacitor ceramico 22pf	\$ 61	\$ 122
6	Resistencias 330ohm	\$ 37	\$ 222
1	Sensor de18b20	\$ 9.758	\$ 9.758
1	Resistencia 4.7kohm	\$ 37	\$ 37
1	Oscilador 4MHz	\$ 620	\$ 620
4	Transistor PNP 2N3906	\$ 86	\$ 344
6	Led rojo difuso	\$ 107	\$ 642
1	Impresión PCB	\$ 60.000	\$ 60.000
		Subtotal	\$ 129.410
1	ESP32	\$ 34.800	\$ 34.800
1	16F84A	\$ 54.406	\$ 54.406
		TOTAL	\$ 218.616

Nota: Precios tomados de <https://didacticaselectronicas.com/>

ANEXO B.
PARTICIPANTES EN EL TALLER REALIZADO EN EL CURSO DE SISTEMAS DIGITALES MICROCONTROLADOS.



TALLER DE TRABAJO DE GRADO

Participantes en el taller de la tesis de grado "Diseño y construcción de un kit de desarrollo a bajo costo para el laboratorio de microprocesadores" el día 12 de mayo de 2023

N°	NOMBRES Y APELLIDOS	FIRMA
1	Juan Alberto Sánchez Saiz	Juan Sánchez
2	Kevin Jose Enamorado Tardocilla	Kevin Enamorado
3	Jairo David Rodríguez Rosinos	Jairo Rosinos
4	Juan Andres Gonzalez Acosta	Juan Gonzalez
5	Jesús Daniel Hoyos Navarro	Jesús Hoyos
6	Jairo Chirre	Jairo Chirre
7	Jaiden Hoyos Villalobos	Jaiden Hoyos
8	Samuel Ricardo Anata Lopez	Samuel Anata
9	Yeison Lopez	Yeison Lopez
10	Miguel Ángel Enamorado Burón	Miguel Enamorado
11	William Morales Blanco	William Morales
12		
13		

ANEXO C. ENCUESTA REALIZADA DURANTE EL TALLER.

<p>¿CONOCÍA EL 16F84A? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>	<p>¿ES MÁS VIABLE UN MONTAJE EN LA PROTOBOARD O EN EL KIT? *</p> <p><input type="radio"/> EN EL KIT</p> <p><input type="radio"/> EN LA PROTOBOARD</p>	<p>CUANDO SE USA EL PIC 16F84, ¿PREFIERE ALIMENTARLO CON CABLES O EL JACK DC?</p> <p><input type="radio"/> CABLES</p> <p><input type="radio"/> JACK DC</p>
<p>¿CONOCÍA EL ESP32? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>	<p>DE LOS ELEMENTOS EN EL KIT, ¿CUÁLES CONOCÍA ANTERIORMENTE? *</p> <p><input type="checkbox"/> SUICHES</p> <p><input type="checkbox"/> POTENCIOMETROS</p> <p><input type="checkbox"/> BOTONES</p> <p><input type="checkbox"/> LEDS</p> <p><input type="checkbox"/> BUZZER</p> <p><input type="checkbox"/> PANTALLA OLED</p> <p><input type="checkbox"/> DISPLAY 7 SEGMENTOS</p> <p><input type="checkbox"/> SENSOR DE TEMPERATURA</p>	<p>¿CREE QUE TODOS LOS ELEMENTOS DEL KIT ES USABLE CON EL PIC 16F84A?</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p> <p><input type="radio"/> TAL VEZ</p>
<p>¿AL INICIO DE LA PRACTICA MANEJABA EL LENGUAJE C? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>		<p>¿CREE QUE EL KIT DE DESARROLLO DISMINUIRÍA LOS TIEMPOS EN LA EJECUCIÓN DE PROYECTOS DE LA MATERIA?</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>
<p>¿CONOCIA EL IDE ARDUINO? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>	<p>DE LOS ELEMENTOS EN EL KIT, ¿CUÁLES SE USARON EN LA MATERIA? *</p> <p><input type="checkbox"/> SUICHES</p> <p><input type="checkbox"/> POTENCIOMETROS</p> <p><input type="checkbox"/> BOTONES</p> <p><input type="checkbox"/> LEDS</p> <p><input type="checkbox"/> BUZZER</p> <p><input type="checkbox"/> PANTALLA OLED</p> <p><input type="checkbox"/> DISPLAY 7 SEGMENTOS</p> <p><input type="checkbox"/> SENSOR DE TEMPERATURA</p>	<p>LO QUE SE REALIZÓ EN EL TALLER, ¿SE HUBIERA PODIDO HACER EN ESE TIEMPO CON UNA PROTOBOARD?</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>
<p>¿SABÍA QUE EL ESP32 SE PODÍA MANEJAR CON ARDUINO? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>		<p>DEL 1 AL 10, ¿QUÉ TAN ÚTIL SERÍA EL KIT DE DESARROLLO EN LA MATERIA?</p> <p>POCO ÚTIL</p> <p>1 <input type="radio"/></p> <p>2 <input type="radio"/></p> <p>3 <input type="radio"/></p> <p>4 <input type="radio"/></p> <p>5 <input type="radio"/></p> <p>6 <input type="radio"/></p> <p>7 <input type="radio"/></p> <p>8 <input type="radio"/></p> <p>9 <input type="radio"/></p> <p>10 <input type="radio"/></p> <p>MUY ÚTIL</p>
<p>¿SIENTE QUE SERÍA CAPAZ DE MANEJAR EL ESP32 CON EL IDE DE ARDUINO ? *</p> <p><input type="radio"/> MUY CAPAZ</p> <p><input type="radio"/> NORMAL</p> <p><input type="radio"/> POCO CAPAZ</p>	<p>A LA HORA DE ALIMENTAR EL KIT, ¿EL CARGADOR DC FACILITA LA REALIZACIÓN DE PRUEBAS CUANDO NO HAY UN COMPU O FUENTE A LA MANO?</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>	
<p>¿ES ÚTIL LA PROGRAMACIÓN DEL ESP32 EN EL IDE ARDUINO? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>	<p>CUANDO SE USA EL ESP32 ¿PREFERIRÍA ALIMENTARLO POR EL JACK DC O EL MICRO USB? *</p> <p><input type="radio"/> MICRO USB</p> <p><input type="radio"/> JACK DC</p>	
<p>¿LES GUSTARÍA QUE FUTUROS ESTUDIANTES DE ESTA MATERIA TENGAN ACCESO AL KIT EN LA UNIVERSIDAD? *</p> <p><input type="radio"/> SÍ</p> <p><input type="radio"/> NO</p>		

ANEXO D.
MANUAL DE PRÁCTICAS MÓDULO ENTRENADOR PIC16F84A Y ESP32.



Manual de prácticas Módulo Entrenador

LUIS A. DAVILA CASUIL

PIC16F84A y ESP32

Práctica 1: Secuencia de LEDS con el PIC16F84A.

Enunciado: En esta práctica, se propone a los estudiantes trabajar con tres interruptores para crear secuencias de LEDs en el Módulo Entrenador. Cada interruptor desencadenará una secuencia de LEDs diferente en el conjunto de luces. El objetivo es que los estudiantes comprendan cómo configurar y controlar interruptores para lograr efectos visuales con LEDs, lo que les permitirá adquirir habilidades prácticas en la interacción hardware-software.

Materiales:

- Módulo entrenador
- PIC16F84A
- Cableado para conexiones.

Elementos a usar del Módulo entrenador:

- 2 suiches.
- 6 LEDS

Objetivo General: El objetivo principal de este laboratorio es proporcionar a los estudiantes una experiencia práctica en la interacción con hardware y software al utilizar interruptores para controlar secuencias de LEDs. Se busca fomentar la comprensión de cómo se configuran y controlan los interruptores en el entorno de desarrollo, y cómo se generan efectos visuales con LEDs en función de la entrada del usuario.

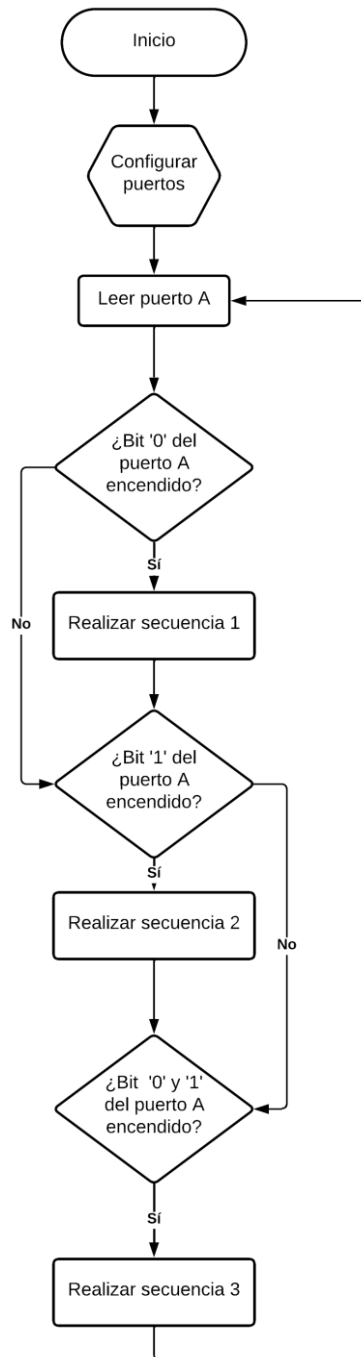
Objetivos Específicos:

1. Configurar los interruptores como entradas digitales en la placa de desarrollo.
2. Programar la placa de desarrollo para detectar el estado de los interruptores.
3. Diseñar y programar tres secuencias de LEDs diferentes que se activan según el interruptor presionado.
4. Comprender cómo controlar las salidas de LEDs para lograr efectos visuales deseados.
5. Implementar una lógica de temporización para crear secuencias visuales coherentes.

Diagrama de Flujo:

El siguiente diagrama de flujo (ver Figura 1) describe el funcionamiento general del programa:

Figura 1
Diagrama de flujo propuesto.



Solución al ejercicio: A continuación, se presenta la tabla de conexiones del ejercicio.

TABLA 1

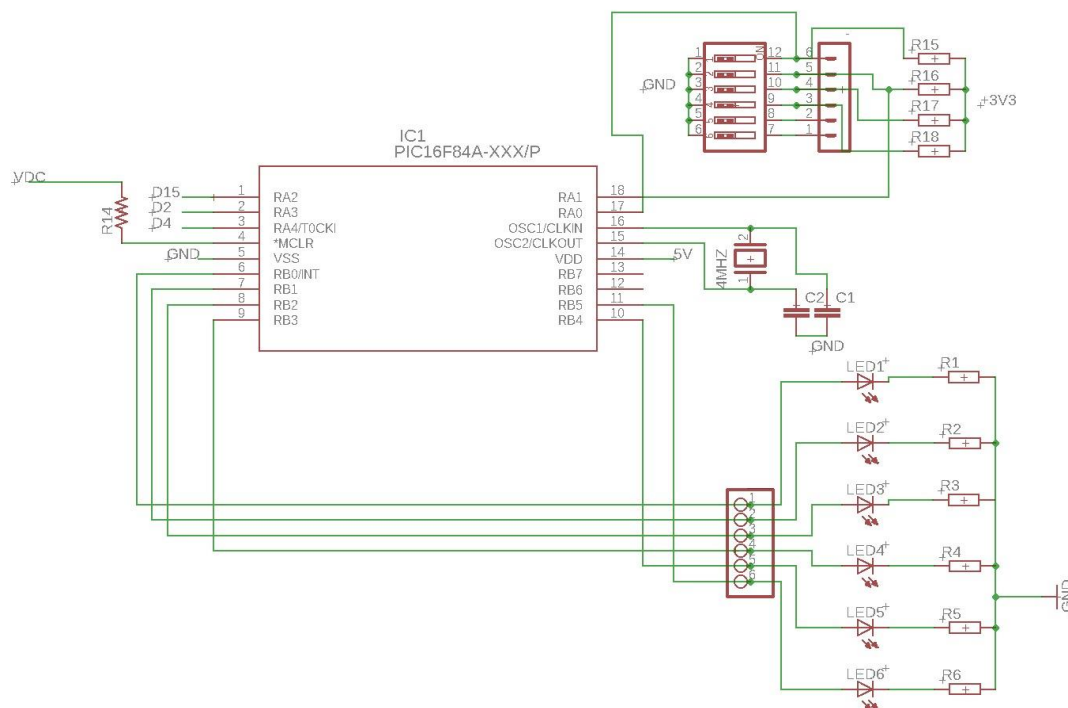
Conexiones de los elementos.

Componente	Conexión al PIC16F84A	Descripción
LED (Salida)	Puerto B (RB0 – RB5)	Conectar los pines del LED a los pines de salida del Puerto B (RB0 – RB5) del PIC16F84A.
Interruptores (Entrada)	Puerto A (RA0 – RA1)	Conecte los interruptores a los pines del Puerto A (RA0 – RA1) del PIC16F84A para leer sus estados como entradas digitales. Es importante que utilice resistencias de pull-up o pull-down para mantener un estado lógico definido cuando no se presionan los interruptores.

A través de la tabla de conexiones, se obtiene el siguiente esquemático.

Figura 2

Esquemático del circuito.



Para abordar el ejercicio propuesto, se puede implementar una solución utilizando el siguiente código, el cual resuelve la problemática de manera efectiva.

```

/*
 * File: principal.c
 * Author: Luis León
 *
 * Created on 25 de abril de 2023, 03:49 AM
 */

```

```

// CONFIG
#pragma config FOSC = HS      // Oscillator Selection bits (HS oscillator)
#pragma config WDTE = OFF     // Watchdog Timer (WDT enabled)
#pragma config PWRTE = OFF    // Power-up Timer Enable bit (Power-up
Timer is disabled)
#pragma config CP = OFF       // Code Protection bit (Code protection disabled)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.
#include <xc.h>
#define _XTAL_FREQ 4000000

void secuencia1(){
    PORTB=0b00000001;
    __delay_ms(250);
    PORTB=0b00000010;
    __delay_ms(250);
    PORTB=0b00000100;
    __delay_ms(250);
    PORTB=0b00001000;
    __delay_ms(250);
    PORTB=0b00010000;
    __delay_ms(250);
    PORTB=0b00100000;
    __delay_ms(250);
}

void secuencia2(){
    PORTB=0b00100000;
    __delay_ms(250);
    PORTB=0b00010000;
    __delay_ms(250);
    PORTB=0b00001000;
    __delay_ms(250);
    PORTB=0b00000100;
    __delay_ms(250);
    PORTB=0b00000010;
    __delay_ms(250);
    PORTB=0b00000001;
    __delay_ms(250);
}

void secuencia3(){
    PORTB=0b00100001;
    __delay_ms(250);
}

```

```
PORTB=0b00010010;
__delay_ms(250);
PORTB=0b00001100;
__delay_ms(250);
PORTB=0b00001100;
__delay_ms(250);
PORTB=0b00010010;
__delay_ms(250);
PORTB=0b00100001;
__delay_ms(250);
}
```

```
void main(void) {
    TRISB=0x00;//Configura pin B como salida
    TRISA0 = 1; // Configura el pin RA0 como entrada
    TRISA1 = 1; // Configura el pin RA1 como entrada
    while (1) {
        if(A0==0){
            secuencia1();
        }
        if(A1==0){
            secuencia2();
        }
        if(A0==0 && A1==0){
            secuencia3();
        }
    }
    return;
}
```

Práctica 2: Utilización de suiches y leds con el ESP32.

Objetivo General:

- El objetivo general de este proyecto de laboratorio es desarrollar un sistema que permita controlar cuatro LEDs mediante la interacción de cuatro suiches. Además, cuando los cuatro interruptores estén encendidos, se activará una función que hará parpadear los LEDs cada segundo.

Objetivos Específicos:

- Configurar correctamente los pines de entrada y salida en una placa de desarrollo.
- Utilizar dos interruptores con pull-up por hardware y dos con pull-up por software para controlar el estado de los LEDs.
- Implementar una función de temporización que haga parpadear los LEDs cuando los cuatro interruptores estén activados.
- Comprender y aplicar los conceptos de lectura de pines digitales y manejo de tiempos en programación.

Materiales

- Módulo Entrenador ESP32
- Jumpers
- Cable MicroUSB

Diagrama de flujo: El siguiente diagrama de flujo (ver Figura 3) describe el funcionamiento general del programa:

Solución al ejercicio: A continuación, se presenta la tabla de conexiones del ejercicio donde se realiza la conexión del ESP32 con el Módulo Entrenador.

Figura 3
Diagrama de flujo de la práctica propuesta.

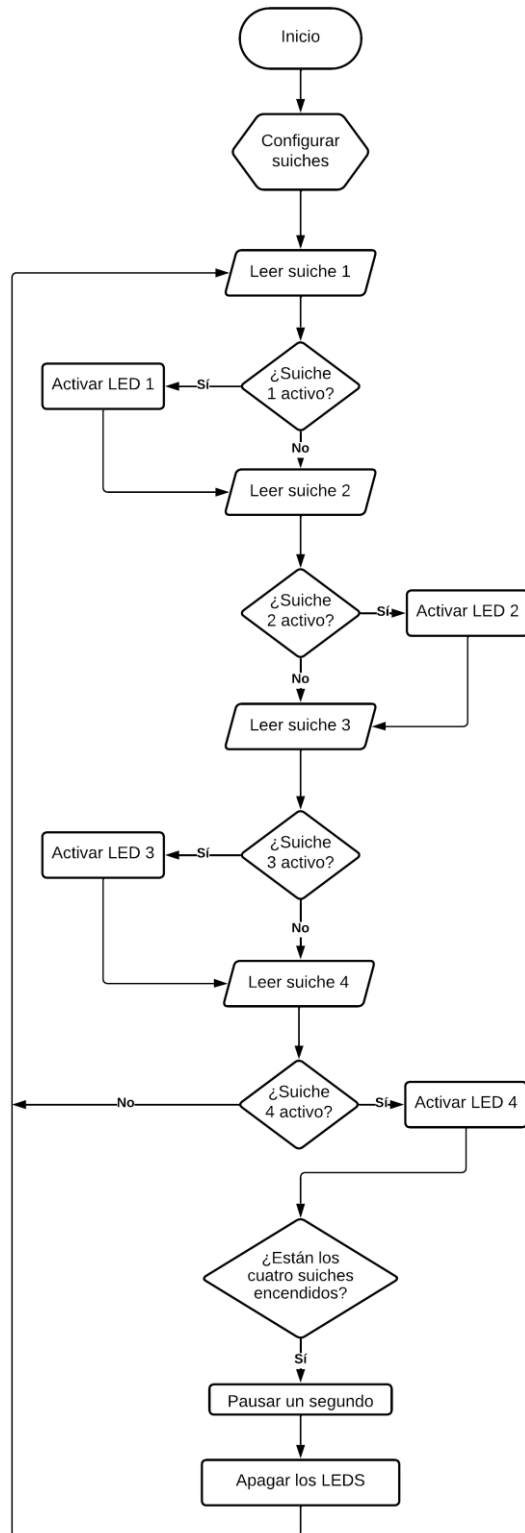


TABLA 2

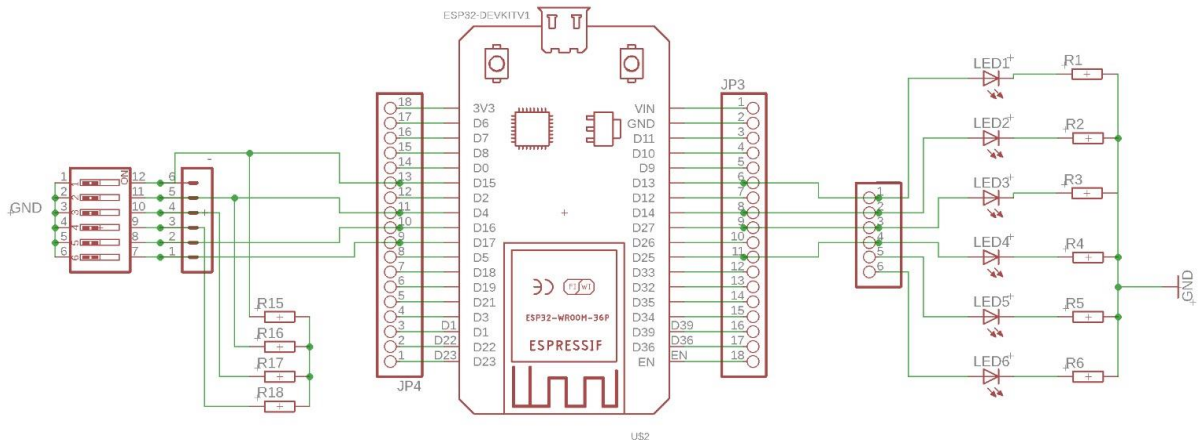
Conexiones del ESP32 con el Módulo Entrenador

Componente	Pin en ESP32
Suiche 1	Pin 15 (con pull-up por hardware)
Suiche 2	Pin 4 (con pull-up por hardware)
Suiche 3	Pin 16 (con pull-up por software)
Suiche 4	Pin 17 (con pull-up por software)
LED 1	Pin 13
LED 2	Pin 14
LED 3	Pin 27
LED 4	Pin 25

Se deben conectar los interruptores y LEDs a los pines indicados en la placa ESP32. El programa utiliza tanto pines con pull-up por hardware como pines con pull-up por software para los interruptores. Los LED se controlan a través de los pines especificados, con lo cual se obtiene el esquemático de la *Figura 4*.

Figura 4

Esquemático de la práctica.



Finalmente, el programa que permite realizar la práctica se presenta a continuación.

```

/ Definir pines para interruptores y LEDs
const int switchPin1 = 15; // Pin con pull-up por hardware
const int switchPin2 = 4; // Pin con pull-up por hardware
const int switchPin3 = 16; // Pin con pull-up por software
const int switchPin4 = 17; // Pin con pull-up por software
const int ledPin1 = 13;
const int ledPin2 = 14;
const int ledPin3 = 27;
const int ledPin4 = 25;

// Variables para el estado de los interruptores

```

```
int switchState1 = 0;
int switchState2 = 0;
int switchState3 = 0;
int switchState4 = 0;

void setup() {
  // Inicializar los pines de interruptores y LEDs
  pinMode(switchPin1, INPUT);
  pinMode(switchPin2, INPUT);
  pinMode(switchPin3, INPUT_PULLUP);
  pinMode(switchPin4, INPUT_PULLUP);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
}

void loop() {
  // Leer el estado de los interruptores
  switchState1 = !digitalRead(switchPin1);
  switchState2 = !digitalRead(switchPin2);
  switchState3 = !digitalRead(switchPin3);
  switchState4 = !digitalRead(switchPin4);

  // Controlar los LEDs en función del estado de los interruptores
  digitalWrite(ledPin1, switchState1);
  digitalWrite(ledPin2, switchState2);
  digitalWrite(ledPin3, switchState3);
  digitalWrite(ledPin4, switchState4);
  delay(10);
  // Si todos los interruptores están encendidos, hacer que los LEDs parpadeen
  if (switchState1 && switchState2 && switchState3 && switchState4) {
    delay(1000);
    // Invertir el estado de los LEDs
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, LOW);
    digitalWrite(ledPin4, LOW);
    delay(1000);
  }
}
```

Práctica 3: Utilización de los interruptores, display 7 segmentos y sensor de temperatura.

Objetivo General: El objetivo de este laboratorio es crear un sistema que permita visualizar la temperatura en grados Celsius (°C) o grados Fahrenheit (°F) en un display de 7 segmentos utilizando dos interruptores con pull-up. Además, se activará un buzzer cuando la temperatura supere los 35°C.

Objetivos Específicos:

1. Configurar dos interruptores como pull-up para seleccionar la visualización en °C o °F.
2. Leer la temperatura desde un sensor y realizar la conversión de °C a °F si es necesario.
3. Mostrar la temperatura en el display de 7 segmentos junto con un indicador de la unidad seleccionada (°C o °F).
4. Activar un buzzer cuando la temperatura supere los 35°C.

Materiales:

- Módulo Entrenador
- Cableado
- ESP32

Diagrama de flujo: Se presenta el diagrama de flujo (ver Figura 5) el cual describe el funcionamiento general del programa.

Solución al ejercicio: A continuación, se presenta la Tabla 3 de conexiones del ejercicio donde se realiza la conexión del ESP32 con el Módulo Entrenador.

Figura 5
Diagrama de flujo

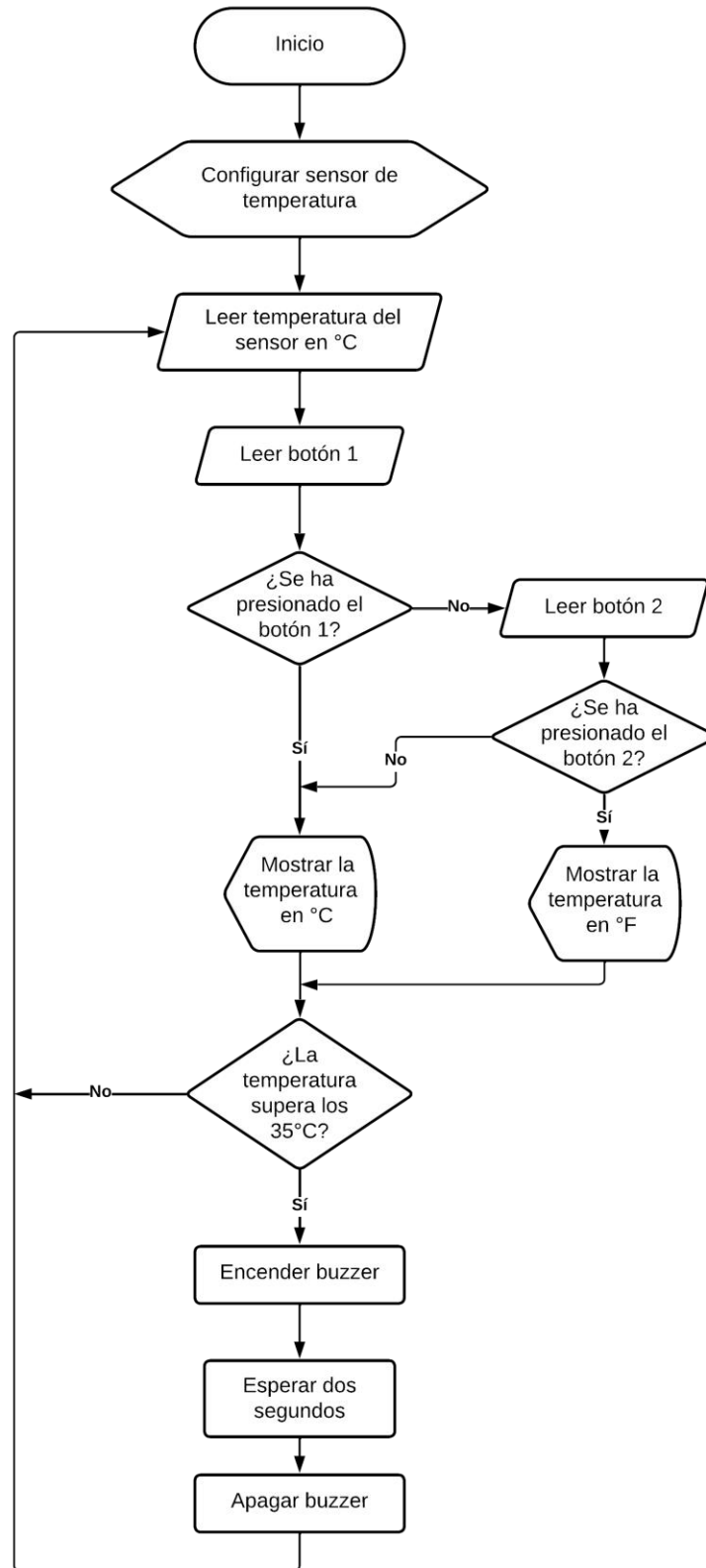


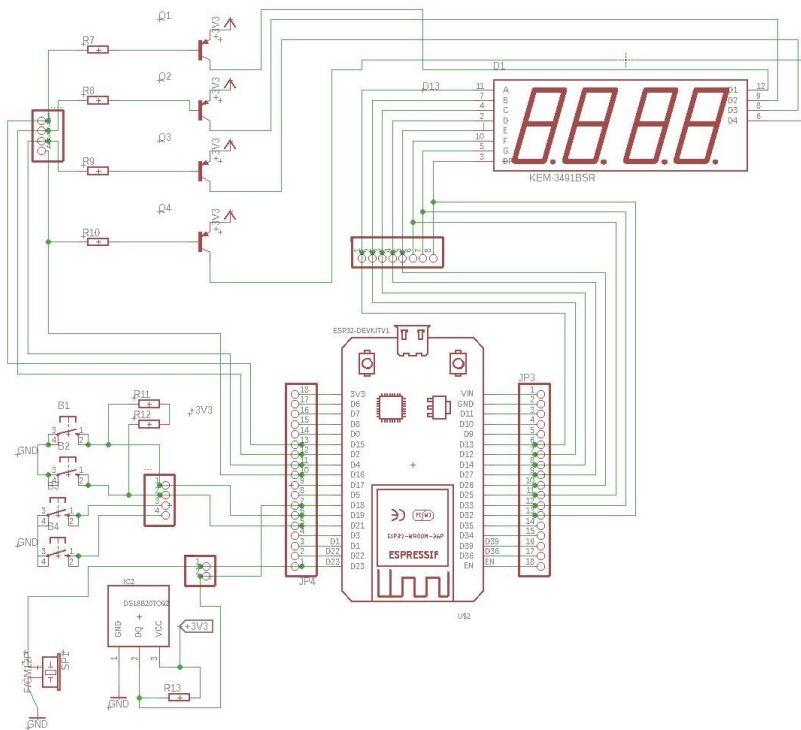
TABLA 3
Conexiones con el Módulo Entrenador

Componente	Pin en el ESP32
Sensor de Temperatura	Pin 18
Interruptor 1 (°C)	Pin 19
Interruptor 2 (°F)	Pin 21
Display a	Pin 13
Display b	Pin 12
Display c	Pin 14
Display d	Pin 27
Display e	Pin 26
Display f	Pin 25
Display g	Pin 33
Display dp	Pin 32
Buzzer	Pin 23
Display 1	Pin 15
Display 2	Pin 2
Display 3	Pin 4
Display 4	Pin 16

De la tabla de conexiones se realiza el esquemático (Ver figura 6).

Figura 6

Esquemático de la práctica



Finalmente, el código que realiza los requerimientos se observa a continuación.

```
//Incluir las bibliotecas OneWire y DallasTemperature

#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 18;

//Comunicar que vamos a utilizar la interfaz oneWire

OneWire oneWire(oneWireBus);

//Indica que el sensor utilizará la interfaz OneWire

DallasTemperature sensors (&oneWire);

// Definir pines para interruptores, display de 7 segmentos y buzzer
const int switchPin1 = 19; // Interruptor para °C
const int switchPin2 = 21; // Interruptor para °F
#define a 13 // Pines para el display de 7 segmentos, se encuentra en anodo común
#define b 12
#define c 14
#define d 27
#define e 26
#define f 25
#define g 33
#define dp 32
const int buzzerPin = 23; // Pin del buzzer

#define disp1 15//digitos del display 7 segmentos
#define disp2 2
#define disp3 4
#define disp4 16

// Variables para el estado de los interruptores, la temperatura y la unidad
seleccionada
int estado =0;
float temperatureC = 0;
float temperatureF = 0;
unsigned long varTiempo=0;

void setup() {
    //Ajustar la velocidad para el monitor serie
```

```

sensors.begin();
// Inicializar los pines de interruptores, display y buzzer
pinMode(switchPin1, INPUT);
pinMode(switchPin2, INPUT);

pinMode(a, OUTPUT);
pinMode(b, OUTPUT);
pinMode(c, OUTPUT);
pinMode(d, OUTPUT);
pinMode(e, OUTPUT);
pinMode(f, OUTPUT);
pinMode(g, OUTPUT);
pinMode(dis1, OUTPUT);
pinMode(dis2, OUTPUT);
pinMode(dis3, OUTPUT);
pinMode(dis4, OUTPUT);

pinMode(buzzerPin, OUTPUT);

}

void loop() {

if (millis() - varTiempo > 5000) {
  int parteEntera = (int) temperatureC;
  double parteDecimal = temperatureC*10 - parteEntera*10;
  digitalWrite(dis1, HIGH); // DIGITO 4 mostrar parte decimal
  digitalWrite(dis2, HIGH);
  digitalWrite(dis3, HIGH);
  digitalWrite(dis4, LOW);
  digitalWrite(dp, LOW);
  disp_num(parteDecimal);
  // Leer la temperatura en °C
  sensors.requestTemperatures();
  temperatureC = sensors.getTempCByIndex(0); // Supongamos que la relación
es de 10mV/°C
  varTiempo = millis();
}

if (millis()==0){
  // Leer la temperatura en °C
  sensors.requestTemperatures();
}
}

```

```

    temperatureC = sensors.getTempCByIndex(0); // Supongamos que la
relación es de 10mV/°C
}
temperatureF = (temperatureC * 9 / 5) + 32;

// Actualizar la unidad seleccionada
if ((digitalRead(switchPin1) == LOW) || (estado==0)) {
    estado =0;
    displayTemperature(temperatureC, 0);
}
if ((digitalRead(switchPin2) == LOW) || (estado==1)) {
    estado =1;
    displayTemperature(temperatureF, 1);
}

// Mostrar la temperatura en el display de 7 segmentos

// Comprobar si la temperatura supera los 35°C y activar el buzzer
if (( temperatureC > 35) || (temperatureF > 95)) {
    digitalWrite(buzzerPin, HIGH); // Encender el buzzer
} else {
    digitalWrite(buzzerPin, LOW); // Apagar el buzzer
}
}

void displayTemperature(float tempImp, int unitTemp) {
    int parteEntera1 = (int) tempImp/10;
    double parteDecimal1 = tempImp - parteEntera1*10;
    digitalWrite(dis1, LOW);//DIGITO 1
    digitalWrite(dis2, HIGH);
    digitalWrite(dis3, HIGH);
    digitalWrite(dis4, HIGH);
    digitalWrite(a, HIGH);
    digitalWrite(b, HIGH);
    digitalWrite(c, HIGH);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, LOW);
    digitalWrite(dp, HIGH);
    delay(2);
    digitalWrite(dis1, HIGH);//DIGITO 2
    digitalWrite(dis2, LOW);
    digitalWrite(dis3, HIGH);

```

```

digitalWrite(dis4, HIGH);
disp_num(parteEntera1);
delay(2);
  digitalWrite(dis1, HIGH);//DIGITO 3
digitalWrite(dis2, HIGH);
digitalWrite(dis3, LOW);
digitalWrite(dis4, HIGH);
digitalWrite(dp, HIGH);
disp_num(parteDecimal1);
delay(2);
  digitalWrite(dis1, HIGH);//DIGITO 4
digitalWrite(dis2, HIGH);
digitalWrite(dis3, HIGH);
digitalWrite(dis4, LOW);
if(unitTemp==0){
digitalWrite(a, LOW);//Letra C
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, HIGH);
digitalWrite(dp, HIGH);
}else{//Letra F
  digitalWrite(a, LOW);//Letra F
  digitalWrite(b, HIGH);
  digitalWrite(c, HIGH);
  digitalWrite(d, HIGH);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
  digitalWrite(dp, HIGH);
}
delay(2);
}

void disp_num(int num){//Imprimir número a 7 segmentos anodo comun
switch(num) {
  case 0:
    digitalWrite(a, LOW);
    digitalWrite(b, LOW);
    digitalWrite(c, LOW);
    digitalWrite(d, LOW);
    digitalWrite(e, LOW);
    digitalWrite(f, LOW);
    digitalWrite(g, HIGH);
    break;

```

```
case 1:
digitalWrite(a, HIGH);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
break;
case 2:
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, HIGH);
digitalWrite(g, LOW);
break;
case 3:
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, LOW);
break;
case 4:
digitalWrite(a, HIGH);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;
case 5:
digitalWrite(a, LOW);
digitalWrite(b, HIGH);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, HIGH);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;
case 6:
digitalWrite(a, LOW);
```

```
digitalWrite(b, HIGH);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;
case 7:
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, HIGH);
digitalWrite(g, HIGH);
break;
case 8:
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;
case 9:
digitalWrite(a, LOW);
digitalWrite(b, LOW);
digitalWrite(c, LOW);
digitalWrite(d, HIGH);
digitalWrite(e, HIGH);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
break;
default:
digitalWrite(a, LOW);
digitalWrite(b, HIGH);
digitalWrite(c, HIGH);
digitalWrite(d, LOW);
digitalWrite(e, LOW);
digitalWrite(f, LOW);
digitalWrite(g, LOW);
```

```
}
}
```

Ejercicio propuesto: Configuración de Conexión Wi-Fi con ESP32 y Pantalla OLED

Enunciado:

Realizar la configuración de un ESP32 para detectar y conectarse a redes Wi-Fi disponibles. Utilizar una pantalla OLED para mostrar las redes detectadas, e implementar un sistema de control mediante interruptores que permitirá al usuario navegar entre las redes y conectar el ESP32 a la red deseada.

Materiales:

- ESP32
- Pantalla OLED (incluida en el Módulo Entrenador
- Interruptores (tres en total)
- Cables de conexión
- Fuente de alimentación para el ESP32
- Computadora con el entorno de desarrollo Arduino y las librerías necesarias instaladas

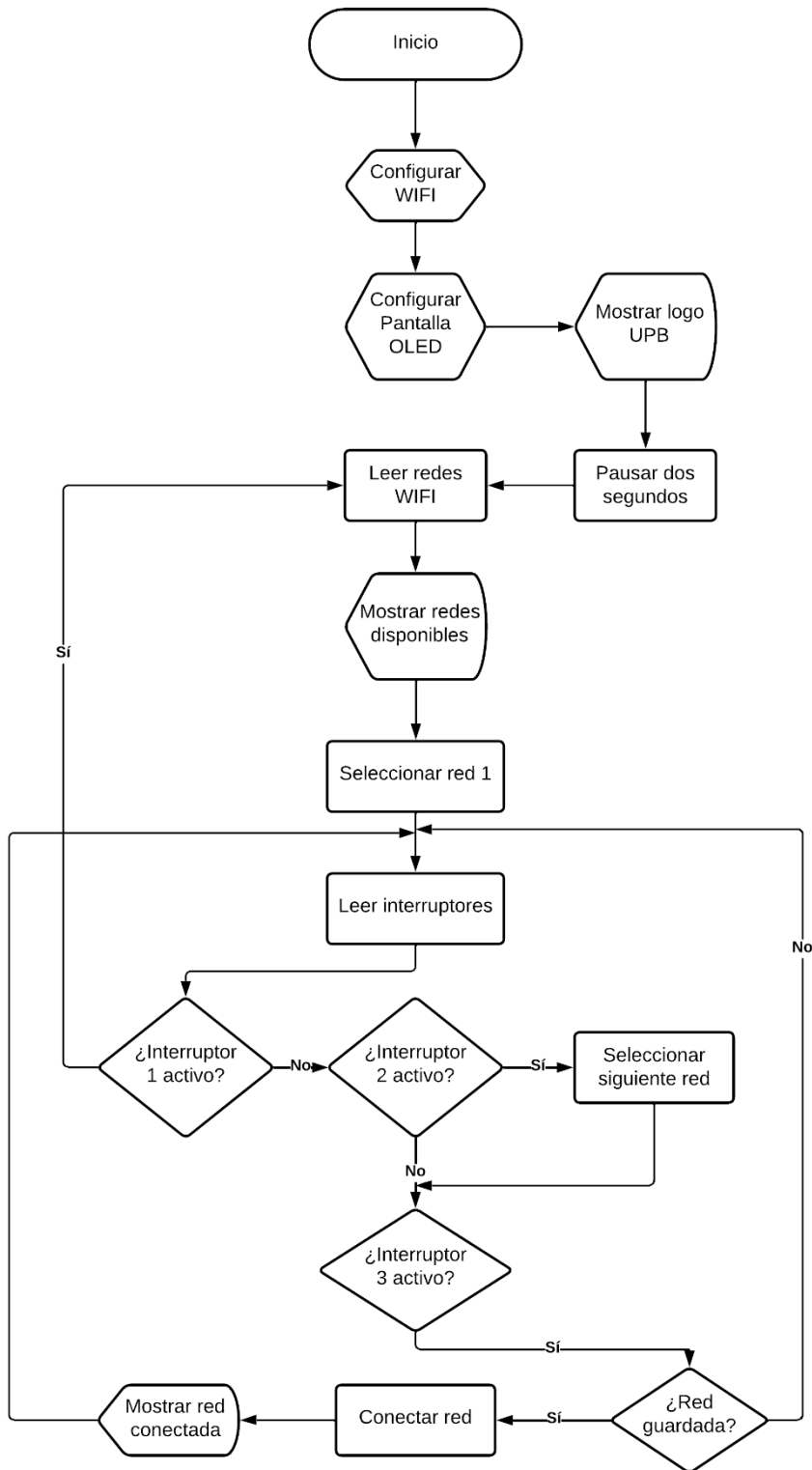
Objetivo General:

El objetivo principal de este laboratorio es adquirir habilidades en la configuración y gestión de conexiones Wi-Fi en un ESP32 y en la interacción con pantallas OLED para mostrar información relevante. Al finalizar el laboratorio, los participantes deberán ser capaces de programar un ESP32 para detectar, listar y conectarse a redes Wi-Fi disponibles, así como utilizar interruptores para controlar estas acciones.

Objetivos Específicos:

1. Configurar el entorno de desarrollo para programar el ESP32.
2. Utilizar una pantalla OLED para mostrar información sobre las redes Wi-Fi detectadas.
3. Implementar un sistema de control mediante interruptores para navegar entre las redes Wi-Fi y seleccionar una para la conexión.
4. Programar el ESP32 para conectarse a la red Wi-Fi seleccionada.
5. Mostrar un mensaje de "Conectado" en la pantalla OLED cuando la conexión sea exitosa.

Diagrama de Flujo:
Figura 7
Diagrama de flujo del ejercicio propuesto.



Finalmente, se presenta un código guía, el cual muestra las redes y permite seleccionarlas con un botón, compatible con el ESP32 y el Módulo Entrenador.

```
#include <WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128//Tamano pantalla
#define SCREEN_HEIGHT 64

// Definir pines para interruptores
const int switchPin1 = 5; // Interruptor para actualizar redes
const int switchPin2 = 18; // Interruptor para seleccionar red
const int switchPin3 = 19; // Interruptor para conectar red
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
int estado=0;
int redActual = 0;
int numNetworks = -1;
static const uint8_t image_data_Saraarray[1024] = {
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x7f, 0xff, 0xff, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x1f, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x0f, 0xff, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x03, 0xff, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x01, 0xff, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x7f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x3b, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
  0x00, 0xff, 0xff, 0xff, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x00,
```


0xff, 0x8f, 0xff, 0xfe, 0x3f, 0xe0, 0x00, 0x38, 0xf8, 0xdf, 0xef, 0x77, 0xb1, 0x3f,
0x36, 0x00,
0xff, 0x8f, 0xff, 0xfe, 0x3f, 0xe0, 0x00, 0x38, 0xf8, 0xec, 0x66, 0x33, 0x33,
0x13, 0x23, 0x00,
0xf8, 0x89, 0xb1, 0xce, 0x23, 0xe0, 0x00, 0x38, 0xf8, 0x6c, 0x66, 0x33, 0x32,
0x03, 0x03, 0x00,
0xf0, 0x8d, 0xba, 0xd6, 0x21, 0xe0, 0x00, 0x38, 0xf8, 0x6c, 0x66, 0x33, 0x32,
0x03, 0x3f, 0x00,
0xf0, 0x8d, 0xba, 0xd6, 0x21, 0xe0, 0x00, 0x39, 0xb8, 0x6c, 0x66, 0x33, 0x32,
0x03, 0x63, 0x00,
0xf0, 0x8d, 0xb9, 0xc6, 0x21, 0xe0, 0x00, 0x3f, 0x38, 0x6c, 0x66, 0x33, 0x33,
0x03, 0x63, 0x00,
0xe0, 0x8d, 0xbb, 0xd6, 0x20, 0xe0, 0x00, 0x38, 0x18, 0x6c, 0x66, 0x33, 0x33,
0x03, 0x63, 0x00,
0xe0, 0x8d, 0xbb, 0xd6, 0x20, 0xe0, 0x00, 0x38, 0x1c, 0x4c, 0x66, 0x33, 0x33,
0x8b, 0x67, 0x00,
0xc0, 0x8e, 0x73, 0xce, 0x20, 0x60, 0x00, 0x38, 0x0c, 0xcc, 0x67, 0xb3, 0x39,
0xf3, 0x7f, 0x00,
0xc0, 0x8f, 0xff, 0xfe, 0x20, 0x60, 0x00, 0x3c, 0x07, 0x1e, 0x73, 0x77, 0xb9,
0xe7, 0x3b, 0x00,
0xc0, 0x8f, 0xff, 0xfe, 0x20, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x80, 0x4f, 0xff, 0xfe, 0x40, 0x20, 0xfc, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x40, 0x04, 0x00, 0x40, 0x00, 0x7e, 0x00, 0x19, 0x80, 0x00, 0x00, 0x60,
0x00, 0x00, 0x00,
0x00, 0x20, 0x00, 0x00, 0x80, 0x00, 0x67, 0x00, 0x39, 0x80, 0x00, 0x00, 0x60,
0x00, 0x00, 0x00,
0x00, 0x10, 0x00, 0x01, 0x00, 0x00, 0x63, 0x00, 0x19, 0x80, 0x00, 0x00, 0x40,
0x00, 0x00, 0x00,
0x00, 0x0f, 0xff, 0xfe, 0x00, 0x00, 0x63, 0x04, 0x18, 0x00, 0x14, 0x00, 0x04,
0x00, 0x02, 0x00,
0x00, 0x00, 0x3d, 0x80, 0x00, 0x00, 0x66, 0x1f, 0x19, 0xb7, 0xdf, 0x37, 0xe7,
0xcd, 0xe3, 0xe0,
0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x7e, 0x31, 0x99, 0xff, 0x93, 0x3e, 0x64,
0xcf, 0x72, 0x70,
0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x63, 0x21, 0x99, 0x98, 0x83, 0x98, 0x60,
0x46, 0x30, 0x30,
0x00, 0x00, 0x39, 0x80, 0x00, 0x00, 0x63, 0xe0, 0xd9, 0x98, 0x8f, 0x98, 0x61,
0xc4, 0x30, 0xf0,
0x00, 0x00, 0x3f, 0x80, 0x00, 0x00, 0x61, 0xe0, 0xd9, 0x99, 0x3b, 0x98, 0x6e,
0xc4, 0x33, 0x30,
0x00, 0x00, 0xff, 0xe0, 0x00, 0x00, 0x61, 0xe0, 0xd9, 0x8d, 0x33, 0x98, 0x6c,
0x44, 0x36, 0x30,
0x00, 0x01, 0xff, 0xf0, 0x00, 0x00, 0x61, 0xf0, 0xd9, 0x8d, 0x33, 0x98, 0x6c,
0x44, 0x36, 0x30,

```

0x00, 0x03, 0xff, 0xfc, 0x00, 0x00, 0x63, 0xb0, 0x99, 0x8c, 0x33, 0x98, 0x6c,
0xc4, 0x36, 0x30,
0x00, 0x0f, 0xff, 0xfe, 0x00, 0x00, 0x63, 0x31, 0x99, 0x86, 0x33, 0x98, 0x6c,
0xc4, 0x36, 0x70,
0x00, 0x1f, 0xff, 0xff, 0x80, 0x00, 0xfe, 0x1f, 0x3f, 0xc6, 0x3d, 0xb8, 0xff, 0x6e,
0x77, 0xf0,
0x00, 0x3f, 0xff, 0xff, 0x80, 0x00, 0x00, 0x04, 0x00, 0x00, 0x08, 0x00, 0x06,
0x00, 0x01, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00
};

```

```

void setup() {

  pinMode(switchPin1, INPUT);
  analogReadResolution(9);
  Serial.begin(115200);
  // Iniciar comunicacion SDA = 5 y SCL = 4 SSD1306 OLED
  Wire.begin(21,22);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C, false, false)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  delay(2000); // Pause for 2 seconds

  // Limpiar display.
  display.clearDisplay();

  // Dibuja el bitmap en la pantalla oled
  display.drawBitmap(0, 0, image_data_Saraarray, 128, 64, 1);
  display.display();
  // Conectar al WiFi
  //WiFi.begin("nombre_de_red", "contraseña_de_red");

  // Otros ajustes y configuraciones
  delay(1000); // Pause for 3 seconds
  display.clearDisplay();
}

```

```

void loop() {

```

```

if ((digitalRead(switchPin1) == LOW) || (estado==0)) { //Entrar en primera
ejecución o cuando se presiona el interruptor
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE); // Imprime texto blanco
  display.setCursor(0,0); // Inicio
  display.setTextSize(2); //Tamaño del texto
  display.println("WIFI");
  display.setTextSize(1); //
  estado = 1; //Cambia a mostrar las redes actuales hastaa actualizar
  numNetworks = WiFi.scanNetworks();
  if (numNetworks == -1) { //No hay redes
    display.println("No hay redes WIFI");
    while (true);
  } else {
    display.print("Redes disponibles:");
    display.println(numNetworks);

for (int thisNet = 0; thisNet < numNetworks; thisNet++) {
  if(thisNet==0){
    display.setTextColor(SSD1306_BLACK, SSD1306_WHITE); // Draw 'inverse'
text
    redActual=0;
  } else {
    display.setTextColor(SSD1306_WHITE); // Draw white text
  }
  display.print(thisNet+1);
  display.print(" ");
  display.println(WiFi.SSID(thisNet));
  display.display();
}
}
}
int pin2=digitalRead(switchPin2);
if ( pin2== LOW || (estado==1)) { //estado o se presiona el interruptor seleccionar
estado=1;
if(redActual<numNetworks && pin2 == LOW){
  delay(20); //estabilidad
  redActual++;
  if(redActual<numNetworks){ //seleccionar máximo la ultima red
    display.clearDisplay();
    display.setTextColor(SSD1306_WHITE); // Draw white text
    display.setCursor(0,0); // Start at top-left corner
    display.setTextSize(2);
    display.println("WIFI");
    display.setTextSize(1); // Normal 1:1 pixel scale
    display.print("Redes disponibles:");

```

```

    display.println(numNetworks);
    for (int thisNet = 0; thisNet < numNetworks; thisNet++) { //imprime redes
actuales
        if(thisNet==redActual){
            display.setTextColor(SSD1306_BLACK, SSD1306_WHITE); // Draw
'inverse' text
        }else{
            display.setTextColor(SSD1306_WHITE); // Draw white text
        }
        display.print(thisNet+1);
        display.print(" ");
        display.println(WiFi.SSID(thisNet));
        display.display();
    }
}
}
}
display.display();

String connectedSSID = WiFi.SSID();//Red conectada

display.setTextColor(SSD1306_WHITE); // Draw white text
if (WiFi.status() == WL_CONNECTED) { //Si está conectada alguna red
    display.setCursor(0,56); // Start at top-left corner
    display.print("Conectado a: ");
    display.println(connectedSSID);

} else {
    display.setCursor(0,56); // Start at top-left corner
    display.println("No conectado");
}
display.display();
delay(20);
}

```

Reto: Implementar el movimiento de una red a otra mediante un potenciómetro, con los cuales cuenta el Módulo Entrenador.