

OTROS RECURSOS EDUCATIVOS ABIERTOS



**Universidad de Jaén**

Escuela Politécnica Superior de Jaén

# **Curso básico de Programación Hardware con NodeMCU. Actuadores, sensores e híbridos**

Autores: José Luis López Ruiz y Alicia Montoro Lendínez

Fecha: 2024

**Programación Hardware  
Sistemas empotrados y ubicuos**



CREA

# Curso básico de Programación Hardware con NodeMCU. Actuadores, sensores e híbridos

José Luis López Ruiz y Alicia Montoro Lendínez



## Índice

1.	Presentación del curso .....	7
2.	Introducción .....	7
3.	Instalación IDE Arduino.....	10
3.1.	ESP8266.....	11
3.2.	ESP32.....	14
3.3.	Elevador de tensión desde 3.3V hacia 5V.....	15
4.	Dispositivos .....	19
4.1.	Actuadores.....	19
4.1.1.	LED de un solo color.....	19
4.1.2.	LED RGB.....	23
4.1.3.	Láser .....	25
4.1.4.	Pantalla LCD con I2C .....	27
4.1.5.	Servomotor.....	29
4.1.6.	Pulsador .....	31
4.1.7.	Zumbador pasivo .....	34
4.2.	Sensores.....	36
4.2.1.	Sensor de ultrasonido .....	37
4.2.2.	Sensor ambiental .....	39
4.2.3.	Sensor de fuego.....	41
4.2.4.	Sensor de gas.....	43
4.2.5.	Sensor de vibraciones .....	48
4.2.6.	Sensor de movimiento infrarrojo pasivo .....	50
4.2.7.	Sensor fotovoltaico.....	52
4.2.8.	Sensor infrarrojo.....	54
4.2.9.	Sensor de sonido.....	56
4.2.10.	Sensor para detección de líneas .....	58
4.2.11.	Reloj y calendario .....	60
4.2.12.	Sensor de humedad en suelo o higrómetro .....	62
4.2.13.	Sensor de lluvia.....	64
4.2.14.	Acelerómetro triaxial.....	66
4.2.15.	Cámara termal.....	68
4.3.	Híbridos.....	70
4.3.1.	Comunicador unidireccional: emisor y receptor.....	70
4.3.2.	Mando a distancia infrarrojo .....	73



5. Conclusiones ..... 75

Bibliografía ..... 77

## Índice de ilustraciones

<b>Ilustración 1.</b> NodeMCU v0.9. Fuente: <a href="https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard">https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard</a> .....	8
<b>Ilustración 2.</b> NodeMCU V2 Amica. Fuente: <a href="https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard">https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard</a> .....	8
<b>Ilustración 3.</b> Pinout de la placa de desarrollo V2/V3. Fuente: <a href="https://www.luisllamas.es/esp8266-nodemcu/">https://www.luisllamas.es/esp8266-nodemcu/</a> .....	9
<b>Ilustración 4.</b> NodeMCU V3 Lolin/Wemos.....	9
<b>Ilustración 5:</b> Interfaz de la aplicación Arduino IDE.....	10
<b>Ilustración 6.</b> Instalación del paquete para gestionar placas de desarrollo ESP8266 .....	12
<b>Ilustración 7.</b> Selección de la placa y el puerto.....	13
<b>Ilustración 8.</b> Instalación del paquete para gestionar placas de desarrollo ESP32 .....	15
<b>Ilustración 9.</b> Elevador de tensión de 3.3V a 5V.....	16
<b>Ilustración 10.</b> Esquemático del circuito utilizando un elevador de voltaje para alimentar una pantalla LCD.....	16
<b>Ilustración 11.</b> Ejemplo de un multímetro.....	17
<b>Ilustración 12.</b> Medida de V0 (5V).....	17
<b>Ilustración 13.</b> Medición V1/VI (3.3V).....	18
<b>Ilustración 14.</b> Pines de cualquier LED (fuente de <a href="https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/actuadores-y-otras-salidas">https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/actuadores-y-otras-salidas</a> ).....	20
<b>Ilustración 15.</b> Ejemplo de un listado de resistencias comerciales dado por un fabricante .....	21
<b>Ilustración 16.</b> Esquemático del circuito del ejercicio 1.....	23
<b>Ilustración 17.</b> LED RGB .....	23
<b>Ilustración 18.</b> Esquemático del ejercicio 2.....	25
<b>Ilustración 19.</b> Láser modelo ky-008.....	25
<b>Ilustración 20.</b> Esquemático ejercicio 3.....	26
<b>Ilustración 21.</b> LCD con conversor I2C.....	27
<b>Ilustración 22.</b> Instalación de la biblioteca LiquidCrystal_I2C .....	28
<b>Ilustración 23.</b> Esquemático del ejercicio 4.....	29
<b>Ilustración 24.</b> Servomotor SG90.....	29
<b>Ilustración 25.</b> Instalación de la librería Servo.....	30
<b>Ilustración 26.</b> Esquemático del ejercicio 5 .....	31
<b>Ilustración 27.</b> Pulsador .....	31
<b>Ilustración 28.</b> Diagrama de un botón pulsador (fuente de <a href="https://www.automatizacionparatodos.com/push-button-con-arduino/">https://www.automatizacionparatodos.com/push-button-con-arduino/</a> ).....	32
<b>Ilustración 29.</b> Esquema del montaje con cortocircuito (fuente de <a href="https://www.luisllamas.es/leer-un-pulsador-con-arduino/">https://www.luisllamas.es/leer-un-pulsador-con-arduino/</a> ) .....	33
<b>Ilustración 30.</b> Esquema del circuito para evitar un cortocircuito (fuente de <a href="https://www.luisllamas.es/leer-un-pulsador-con-arduino/">https://www.luisllamas.es/leer-un-pulsador-con-arduino/</a> ).....	33
<b>Ilustración 31.</b> Esquemático del ejercicio 6.....	34

<b>Ilustración 32.</b> Zumbador pasivo modelo KY-006.....	35
<b>Ilustración 33.</b> Esquemático del ejercicio 7.....	36
<b>Ilustración 34.</b> Sensor de ultrasonidos modelo HC-SR04.....	37
<b>Ilustración 35.</b> Esquemático del ejercicio 8.....	38
<b>Ilustración 36.</b> Sensor de humedad modelo DHT11.....	39
<b>Ilustración 37.</b> Instalación de la biblioteca DHT sensor library.....	40
<b>Ilustración 38.</b> Esquemático del ejercicio 9.....	40
<b>Ilustración 39.</b> Espectro de emisión (fuente de <a href="https://www.luisllamas.es/detector-llama-arduino/">https://www.luisllamas.es/detector-llama-arduino/</a> ).....	41
<b>Ilustración 40.</b> Sensor de fuego modelo HL-01.....	41
<b>Ilustración 41.</b> Esquemático del ejercicio 10.....	43
<b>Ilustración 42.</b> Sensor de gas MQ-2.....	43
<b>Ilustración 43.</b> Tabla con el tipo de sensores de gas y su tipo de detección.....	45
<b>Ilustración 44.</b> Esquema para la reducción de voltaje en la entrada analógica (fuente de <a href="https://github.com/miguel5612/MQsensorsLib">https://github.com/miguel5612/MQsensorsLib</a> ).....	46
<b>Ilustración 45.</b> Circuito divisor del voltaje (fuente de <a href="https://www.learningaboutelectronics.com/Articulos/Como-reducir-voltaje-con-resistencias.php">https://www.learningaboutelectronics.com/Articulos/Como-reducir-voltaje-con-resistencias.php</a> ).....	46
<b>Ilustración 46.</b> Instalación de la biblioteca MQsensorsLib.....	47
<b>Ilustración 47.</b> Esquemático del ejercicio 11.....	48
<b>Ilustración 48.</b> Sensor de vibraciones modelo SW18010P.....	48
<b>Ilustración 49.</b> Sensor de vibraciones modelo SW-420.....	48
<b>Ilustración 50.</b> Esquemático del ejercicio 12.....	49
<b>Ilustración 51.</b> Sensor de movimiento PIR modelo HC-SR501.....	50
<b>Ilustración 52.</b> Ejemplo de funcionamiento (fuente de <a href="https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/">https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/</a> ).....	50
<b>Ilustración 53.</b> Potenciómetros para ajustar tiempo entre las mediciones y la sensibilidad del sensor (fuente de <a href="https://www.puntoflotante.net/MODULO-SENSOR-PASIVO-INFRRARROJO-PIR-HC-SR501.htm">https://www.puntoflotante.net/MODULO-SENSOR-PASIVO-INFRRARROJO-PIR-HC-SR501.htm</a> ).....	51
<b>Ilustración 54.</b> Esquemático del ejercicio 13.....	52
<b>Ilustración 55.</b> Sensor fotovoltaico.....	53
<b>Ilustración 56.</b> 75. Esquemático del ejercicio 14.....	54
<b>Ilustración 57.</b> Sensor infrarrojo modelo HW-201.....	54
<b>Ilustración 58.</b> Funcionamiento del sensor (fuente de <a href="https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/">https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/</a> ).....	55
<b>Ilustración 59.</b> Esquemático del ejercicio 15.....	56
<b>Ilustración 60.</b> Sensor de sonido modelo KY-038.....	56
<b>Ilustración 61.</b> Esquemático del ejercicio 16.....	58
<b>Ilustración 62.</b> Sensor para detección de líneas KY-033.....	58
<b>Ilustración 63.</b> Esquemático del ejercicio 17.....	60
<b>Ilustración 64.</b> Reloj y calendario modelo RTC DS1302.....	60
<b>Ilustración 65.</b> Instalación de la biblioteca RTC.....	61
<b>Ilustración 66.</b> Esquemático del ejercicio 18.....	62
<b>Ilustración 67.</b> Sensor de humedad en el suelo modelo HW-080.....	63

<b>Ilustración 68.</b> Esquemático del ejercicio 19.....	64
<b>Ilustración 69.</b> Sensor de lluvia modelo HL-83.....	65
<b>Ilustración 70.</b> Esquemático del ejercicio 20.....	66
<b>Ilustración 71.</b> Acelerómetro triaxial modelo HW-616.....	67
<b>Ilustración 72.</b> Esquemático del ejercicio 21.....	68
<b>Ilustración 73.</b> Cámara termal modelo AMG8833.....	68
<b>Ilustración 74.</b> Instalación de la biblioteca AMG88xx .....	69
<b>Ilustración 75.</b> Esquema del ejercicio 22.....	69
<b>Ilustración 76.</b> Comunicador unidireccional (emisor y receptor respectivamente).....	70
<b>Ilustración 77.</b> Instalación de la librería rc-switch .....	72
<b>Ilustración 78.</b> Esquemático del ejercicio 23.....	73
<b>Ilustración 79.</b> Mando a distancia con receptor infrarrojo.....	73
<b>Ilustración 80.</b> Instalación d ela librería IRremoteESP8266.....	74
<b>Ilustración 81.</b> Esquemático del ejercicio 24.....	75

## 1. Presentación del curso

En este documento, se va a presentar un curso básico de programación hardware orientado a placas NodeMCU. El objetivo principal es iniciar al lector en la programación básica de placas de bajo coste a través del uso de elementos de entrada y salida (actuadores), sensores y dispositivos híbridos.

Como objetivos específicos tenemos:

- Programación a bajo nivel de placas de desarrollo NodeMCU.
- Programación de interfaces de E/S en placas de desarrollo NodeMCU
- Utilización de sensores avanzados para recogida de muestras.

Aunque está orientado a placas de desarrollo de la familia NodeMCU, también puede ser extrapolable a cualquier tipo de placa comercial, como por ejemplo una placa de desarrollo Arduino.

La dinámica del curso es la siguientes:

1. Inicialmente se presentan una serie de modelos de placas NodeMCU que pueden ser usadas en el curso, y se define la estructura de cualquier proyecto.
2. A continuación, se van presentando una serie de dispositivos: entrada y salida (atuadores), sensores e híbridos:
  - a. Breve explicación del dispositivo.
  - b. Presentación de los pines y características principales.
  - c. Ejercicio.
  - d. Solución.

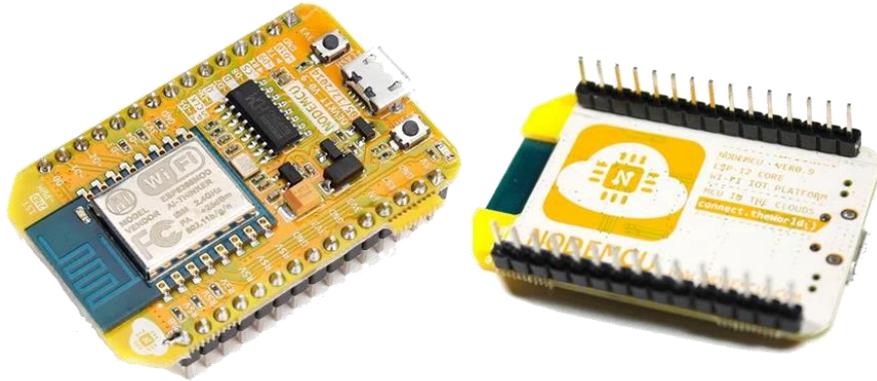
Todas las soluciones se encuentran disponibles también en un repositorio público en GitHub.

<https://github.com/jossellr/CBP-NodeMCU-ASH.git>

## 2. Introducción

Las placas de desarrollo NodeMCU son de bajo coste y son muy utilizadas como tecnología aplicada al concepto de Internet de las Cosas o también denominado Internet of Things (IoT) en inglés.

Este tipo de placas ha evolucionado a través de varias versiones desde su inicio. La primera versión, conocida como devkit v0.9, se ensamblaba con un ESP12 y 4MB de memoria flash. El ESP12 es parecido al ESP12E, pero no tiene una fila de pines, lo que resulta en menos GPIO disponibles. Actualmente, este modelo no está en el mercado.



*Ilustración 1. NodeMCU v0.9. Fuente: <https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>*

La siguiente versión se nombró como V2 aunque también se le suele llamar Amica, creada por el alemán Gerwin Janssen, pero el fabricante la adoptó como placa oficial. A diferencia de la anterior está se monta sobre un ESP12E por lo que dispone de más pines.



*Ilustración 2. NodeMCU V2 Amica. Fuente: <https://www.electronicwings.com/nodemcu/nodemcu-development-kitboard>*

## NODEMCU V2 PINOUT

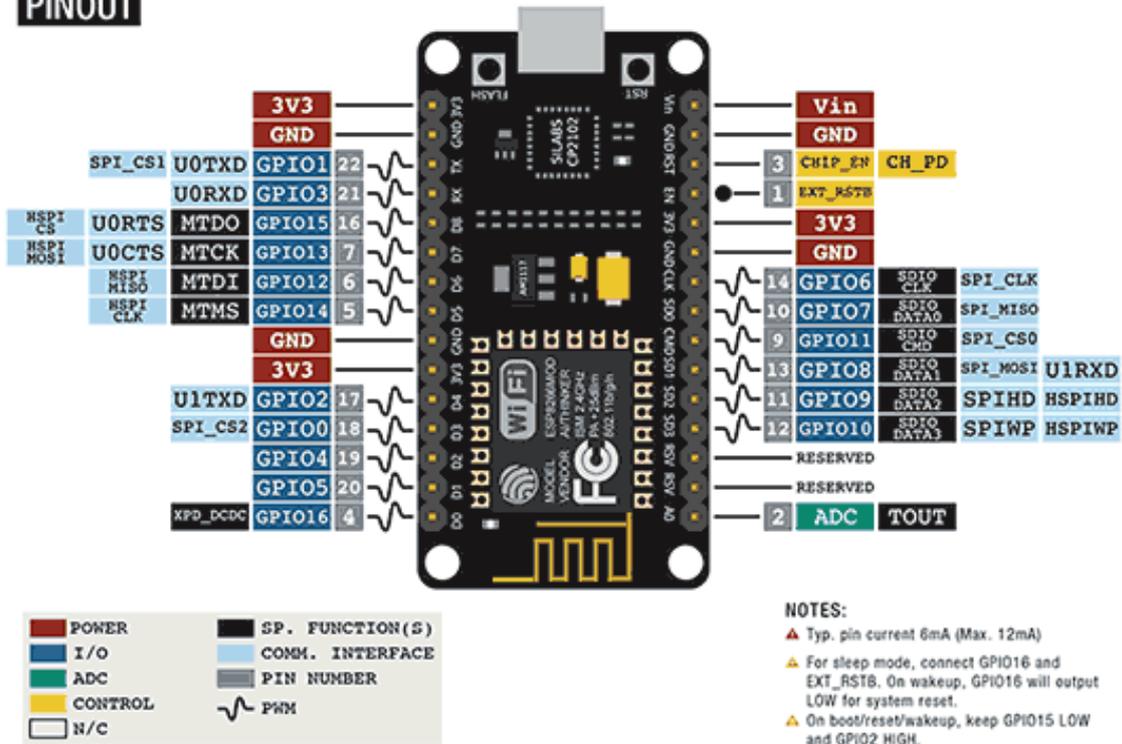


Ilustración 3. Pinout de la placa de desarrollo V2/V3. Fuente: <https://www.luisllamas.es/esp8266-nodemcu/>

Por último, la versión V3 o también llamada Lolin/Wemos, creada como un diseño mejorado por el propio fabricante.



Ilustración 4. NodeMCU V3 Lolin/Wemos

Su principal diferencia sería en converso serial que en este caso es el CH340G en vez del CP2102, para asegurar un USB más robusto. Además, en esta versión mejorada utilizaron los pines RSV de la V2 para añadir un GND y una salida de 5V proveniente del USB, VUSB.

La manera más fácil de diferenciar las dos últimas versiones que se encuentran a la venta sería echando un vistazo rápido al conversor serial. El referente al V2 es más cuadrado mientras que el del V3 es más alargado.

### 3. Instalación IDE Arduino

Para la realización del curso se utilizará el entorno de desarrollo Arduino IDE. Este software permitirá programar, verificar y subir los diferentes códigos en la placa desarrollo.

Para instalarlo simplemente necesitamos descargar<sup>1</sup> la última versión relacionada con nuestro sistema operativo y ejecutar el instalador. Todo el código relacionado con este curso ha sido compilado y probado en el sistema operativo Windows 11 con la versión de Arduino IDE 2.2.1.

Una vez instalado, al ejecutarlo, aparecerá la interfaz definida en la Ilustración 5.

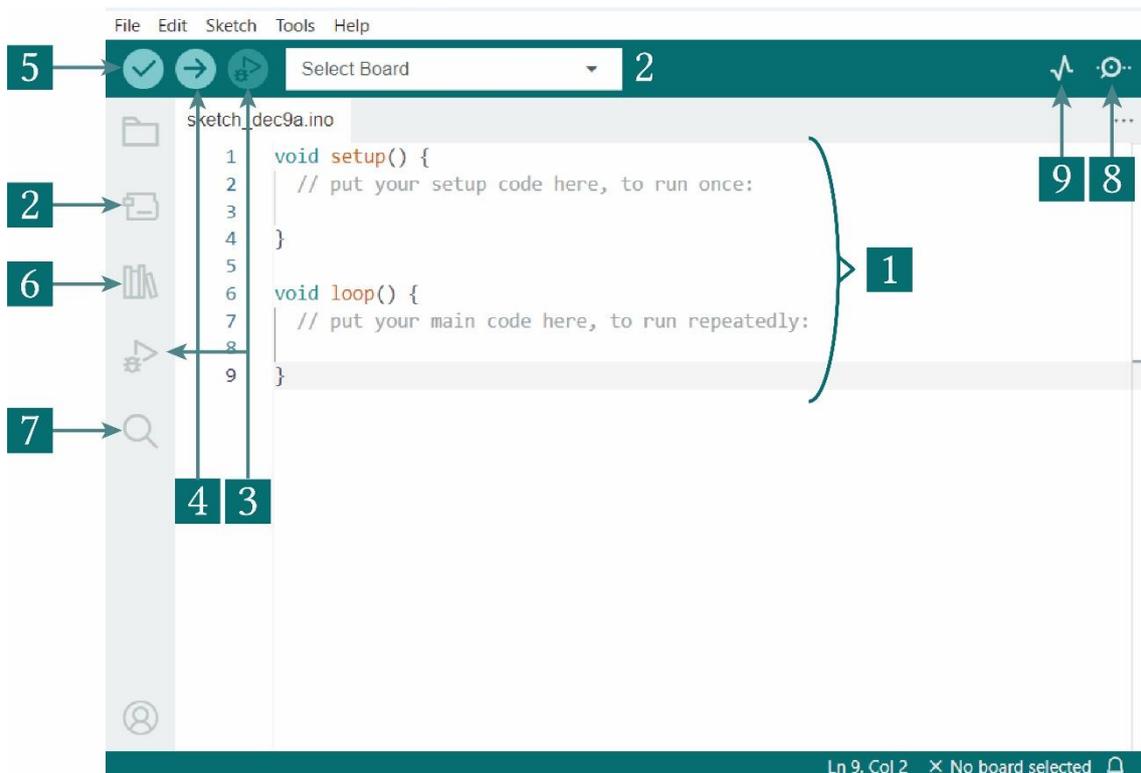


Ilustración 5: Interfaz de la aplicación Arduino IDE

<sup>1</sup> <https://www.arduino.cc/en/software>

En la interfaz gráfica se pueden distinguir una serie de elementos principales que serán muy útiles a lo largo del curso:

1. Zona de código.
2. Selector del tipo de placa de desarrollo.
3. Ejecución y depurador.
4. Subida de código a la placa.
5. Verificación del código.
6. Instalación de bibliotecas de terceros.
7. Búsqueda y reemplazo de texto en el código.
8. Serial Plotter (salida a través de gráficas).
9. Serial Monitor (salida por consola)

Para seguir el curso es indispensable tener conocimientos básicos del lenguaje de programación C. En caso de no contar con un nivel básico es recomendable realizar previamente un curso relacionado. La web W3Schools proporciona un curso en el siguiente [enlace](#). También se recomienda revisar algunos conceptos básicos de electrotecnia.

En cuanto a la placa de desarrollo podremos utilizar cualquiera de estas dos versiones:

- ESP8266.
- ESP32.

Aunque estas placas son las recomendables para el curso, es posible utilizar otro tipo de placas de desarrollo comerciales siempre y cuando incluyan pines equivalentes a los que se utilizan e incluyan un módulo WiFi.

A continuación, se desglosan las diferentes placas de desarrollo recomendadas para el curso.

### 3.1. ESP8266

Para la placa de desarrollo ESP8266 (NodeMCU v2) utilizaremos los siguientes pines:

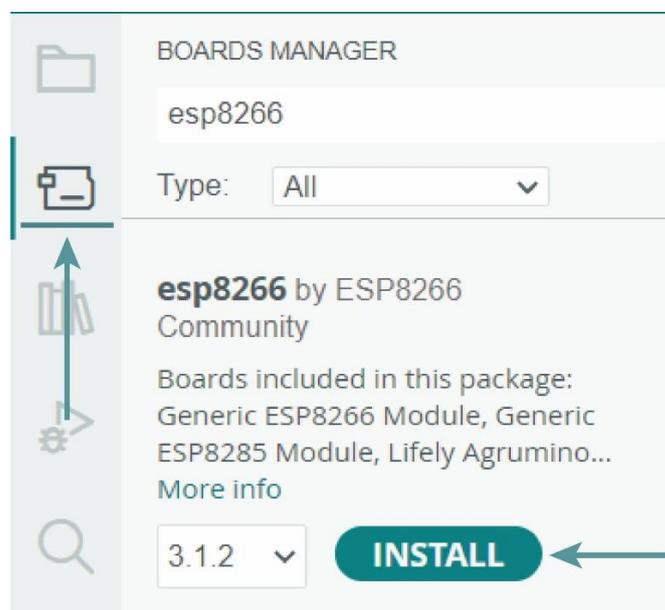
- **3V3**: alimentación a 3.3V.
- **Vin**: alimentación a 5V.
- **GND**: tierra.
- **D0-D8**: entradas digitales.
- **A0**: entrada analógica.

En este tipo de placa debemos tener en cuenta que cualquier pin D0–D8 proporciona un voltaje de 3.3V y la entrada analógica sólo puede recibir un voltaje entre el rango de (0V – 3.3V). Esto es muy importante ya que algunos sensores proporcionan una salida de 5V.

Este tipo de placa de desarrollo no está incluida por defecto en el entorno de desarrollo. Por tanto, debemos acceder a la opción de *File* → *Preferences* → *Settings* y en la opción **Additional boards manager URLs** incluir el siguiente enlace:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

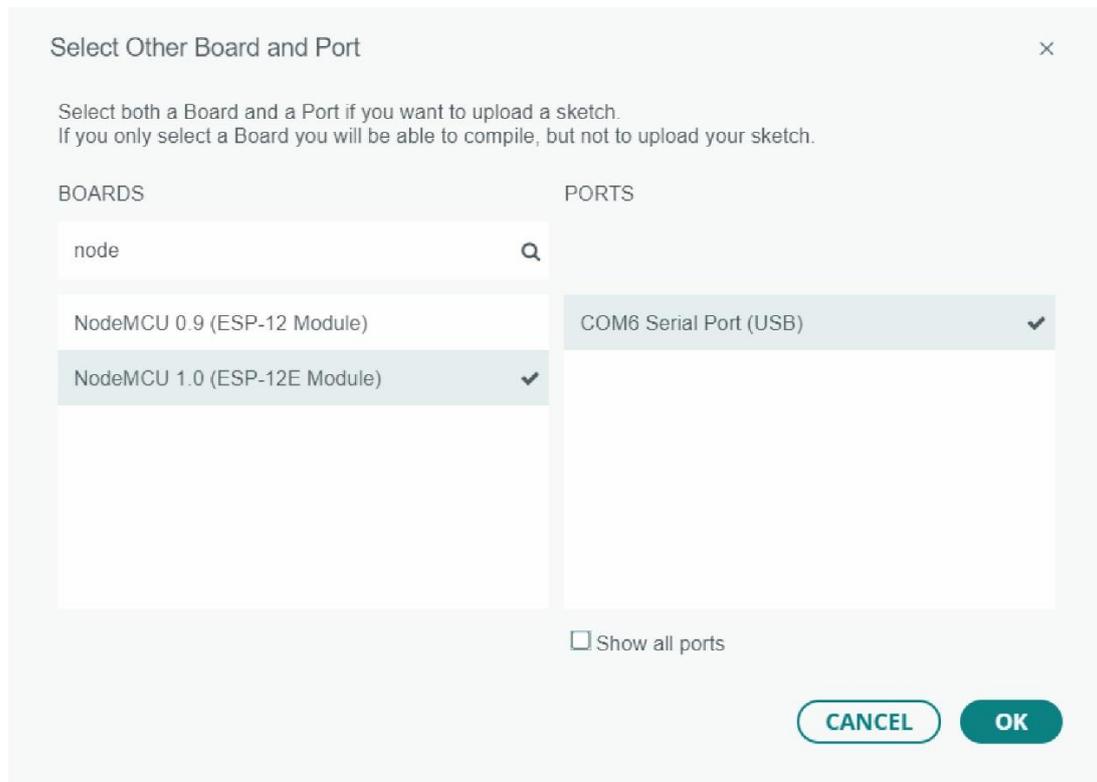
Una vez incluido nos dirigimos al gestor de tarjetas en el lateral izquierdo, buscamos **esp8266** e instalamos la última versión tal y como aparece en la Ilustración 6.



*Ilustración 6. Instalación del paquete para gestionar placas de desarrollo ESP8266*

Una vez se ha instalado, si nos dirigimos al selector de placas (justo al lado del botón de ejecución con depuración) debemos seleccionar la placa **NodeMCU 1.0 (ESP-**

12E Module) y el puerto USB en el que se encuentra conectada la placa tal y como aparece en la Ilustración 7.



*Ilustración 7. Selección de la placa y el puerto*

Para hacer una prueba rápida y comprobar si funciona incluye el siguiente código de ejemplo. En este caso, el código simplemente imprime por pantalla “¡Hola mundo!” infinitamente. Para ejecutarlo:

1. Verifica el código 
2. Sube el código a la placa 
3. Por último, para ver la salida debes pulsar en la opción Serial Monitor 

### Código de ejemplo

```

void setup() {
  Serial.begin(9600); // Define el canal de salida.
}

void loop() {
  Serial.print("¡Hola mundo!\n"); // Imprime por pantalla.
}

```

}

Todos los proyectos están divididos en dos funciones principales:

- **setup ( )**: se ejecuta siempre una vez al inicio, y suele utilizarse para inicializar recursos.
- **loop ( )**: se ejecuta de forma indefinida e implementa la funcionalidad principal de nuestra placa de desarrollo.

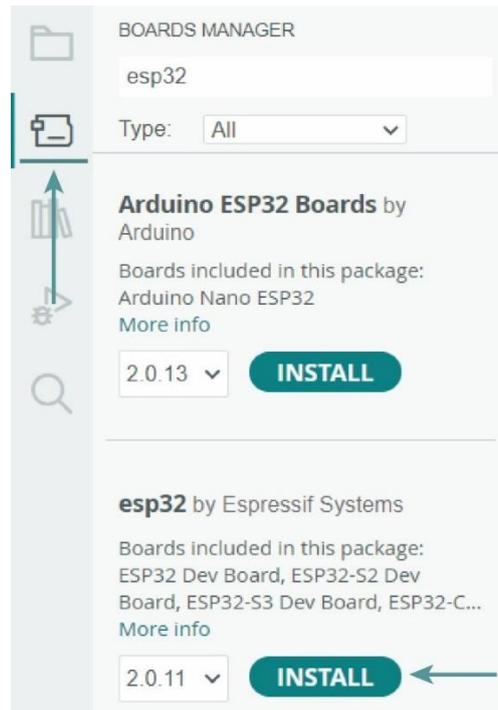
En ambos casos no devuelven nada y tampoco reciben ningún tipo de parámetro. También, es muy importante definir el canal de salida para imprimir por consola resultados y depurar de forma más sencilla el código.

### 3.2. ESP32

En este caso, vamos a utilizar los mismos pines, pero es necesario instalar los paquetes para este tipo de placa. Para ello, de igual forma que en la sección anterior (sección 3.1), debemos incluir un enlace personalizado para poder instalarlos. En enlace es el siguiente:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

Una vez incluido nos dirigimos de la misma forma al gestor de placas, buscamos *esp32* e instalamos la segunda opción tal y como aparece en la Ilustración 8.

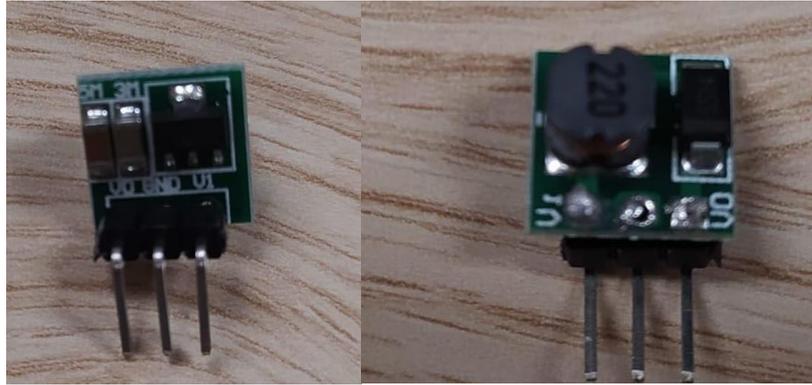


*Ilustración 8. Instalación del paquete para gestionar placas de desarrollo ESP32*

Una vez instalado seleccionamos la placa **ESP32 Dev Module** y probamos el mismo código de la Sección 3.1.

### 3.3. Elevador de tensión desde 3.3V hacia 5V.

En las placas de desarrollo presentadas encontramos un problema bastante común y es que su salida de tensión para poder alimentar a sensores o actuadores sería solo de 3.3V y hay muchos dispositivos que necesitan una mayor tensión de alimentación, en concreto a 5V. Para ello, se propone utilizar un elevador de tensión que no necesitaría de código software. En la Ilustración 9 se incluye un ejemplo del elevador utilizado para este curso.



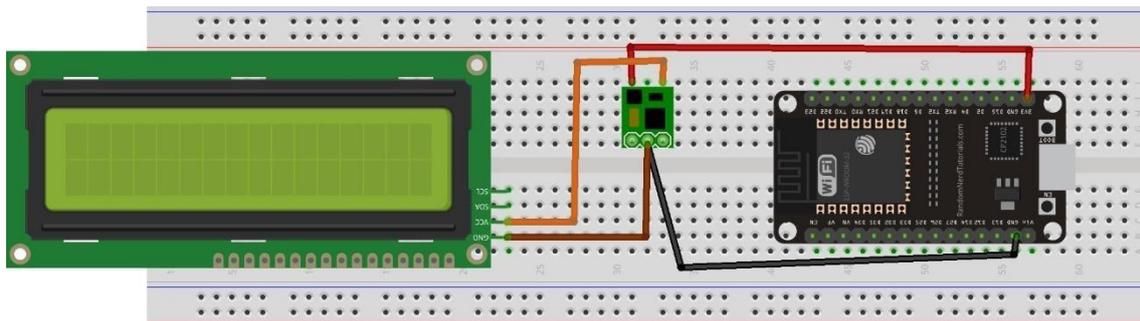
*Ilustración 9. Elevador de tensión de 3.3V a 5V*

En este caso, el elevador cuenta con los siguientes pines:

- **V0:** tensión de salida del elevador a 5V.
- **V1 o VI:** tensión de entrada al elevador de 3.3V.
- **GND:** tierra.

A continuación, se incluye el esquemático para realizar el cableado. Se puede observar que el cable rojo corresponde a la salida de 3.3V de la placa e iría al V1/VI, la salida del elevador de 5V (V0) se uniría con la alimentación de la pantalla LCD. Además, recordemos que todos los elementos deben ir a tierra.

Este circuito se encuentra incompleto, ya que faltaría conectar una serie de pines desde la placa de desarrollo hasta la LCD para suministrar datos de salida. A lo largo del curso veremos qué falta y cómo conectarlo.



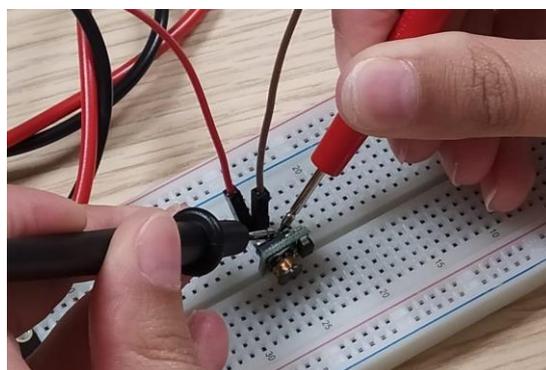
*Ilustración 10. Esquemático del circuito utilizando un elevador de voltaje para alimentar una pantalla LCD*

Para comprobar que la salida que se está obteniendo es de 5V, antes de conectar cualquier dispositivo al pin V0 del elevador, es necesario comprobar con un multímetro (ver Ilustración 11) el voltaje en este punto.



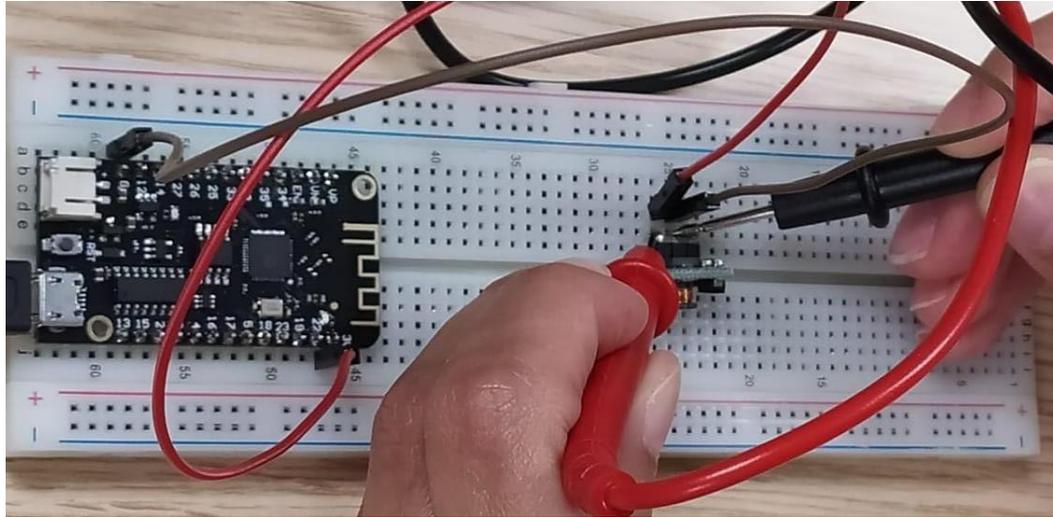
*Ilustración 11. Ejemplo de un multímetro*

En este caso, para realizar dicha comprobación debemos colocar la sonda roja sobre el pin V0 y la sonda negra en el pin GND del elevador (ver **Ilustración 12**).



*Ilustración 12. Medida de V0 (5V).*

Para comprobar también que la entrada que estamos obteniendo de la placa es de 3.3V, se conecta la sonda roja sobre el pin V1 y la sonda negra en el pin GND del elevador (ver Ilustración 12).



*Ilustración 13. Medición V1/V1 (3.3V)*

## 4. Dispositivos

Una placa de desarrollo por sí sola no tiene mucha utilidad. Por lo general, este tipo de placas de bajo coste suelen utilizarse dentro del campo del IoT y utilizan un conjunto de elementos para interactuar con el entorno y el resto de dispositivos inteligentes.

Los dispositivos suelen dividirse en tres categorías:

- Dispositivos de entrada y salida, también llamados actuadores.
- Sensores.
- Dispositivos híbridos.

A lo largo de las secciones trataremos diferentes dispositivos y los utilizaremos a través de ejercicios básicos para manipularlos.

### 4.1. Actuadores

Un actuador es un mecanismo que transforma energía, ya sea hidráulica, neumática o eléctrica, en la activación de un proceso con el propósito de generar un cambio en un sistema automatizado.

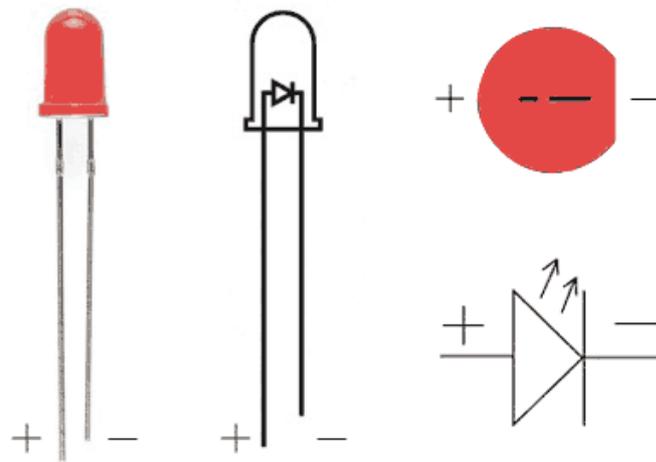
En esta sección, aprenderemos a conectar y utilizar diferentes tipos de actuadores:

- LED con un único color.
- LED al cual podemos modificarle el color.
- Láser.
- Pantalla LCD.
- Servomotor.
- Pulsador.
- Zumbador pasivo.

#### 4.1.1. LED de un solo color

Un LED, que es la abreviatura de diodo emisor de luz en inglés, es un tipo de diodo que emite luz. Contiene un semiconductor que, cuando se le aplica una corriente continua, produce luz, un fenómeno conocido como electroluminiscencia (ver

Ilustración 14). Hay varios tipos de LED, que se diferencian por las tecnologías empleadas en su fabricación y su integración en circuitos electrónicos.



*Ilustración 14. Pines de cualquier LED (fuente de <https://libros.catedu.es/books/programa-arduino-mediante-codigo/page/actuadores-y-otras-salidas>)*

Para alargar la vida de un LED, normalmente se suele incluir una resistencia para reducir la intensidad de la corriente. Las resistencias utilizadas suelen variar entre los  $180\Omega$  y  $330\Omega$  (ver Ilustración 27). Las tensiones umbrales y corrientes máximas, correspondientes a cada diodo LED según su color, se muestran en la Tabla 1.

*Tabla 1. Intervalos de corriente y tensión soportados para cada tipo de color*

Color	Tension umbral (V)	Tensión máxima (V)	Corriente media (mA)	Corriente máxima (mA)
Rojo	1.8	2.2	5-10	20
Verde	2	3.5	5-10	20
Amarillo	2	2.5	5-10	20
Naranja	2.1	2.2	5-10	20

A continuación, se define la fórmula para saber de forma exacta el valor de la resistencia.

$$R = \frac{V_f - V_l}{I}$$

*donde  $V_f$  es el valor de voltaje de la fuente,  $V_l$  el valor del voltaje que se desea en el LED e  $I$  es la intensidad de la corriente de la fuente*

Se supone un ejemplo para un diodo LED de color rojo y se debe tener en cuenta que la salida de tensión de un pin digital correspondiente a nuestras placas es de 3.3V. Por tanto, el valor de la resistencia será el que se muestra a continuación.

$$R = \frac{V_{fuente} - V_{led}}{I} = \frac{3.3 - 1.8}{0.02} = 70\Omega$$

Por tanto, se debe buscar un valor un poco mayor o un poco menor de una resistencia normalizada. En la Ilustración 15 se muestra el código de colores para elegir la resistencia adecuada.

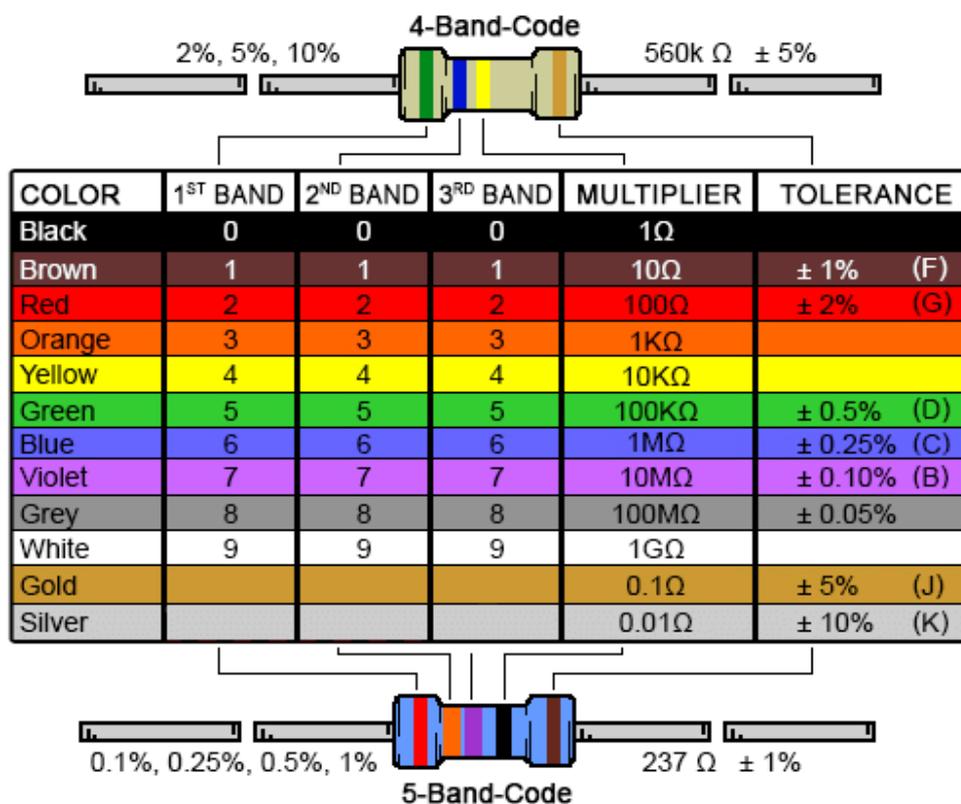


Ilustración 15. Ejemplo de un listado de resistencias comerciales dado por un fabricante

Los LEDs de este tipo se caracterizan por ser resistentes a la lluvia y suelen trabajar de forma general entre los siguientes valores:

- **Corriente máxima:** 20 mA.
- **Voltaje:** 1.8V – 2.4V (depende del color).
- **Temperatura de operación:** -30° a 85° C.

En este caso, nuestro objetivo es codificar la placa de desarrollo para encender un LED. Ya que este tipo de placas no cuenta con ningún dispositivo de salida, es fundamental saber trabajar con estos actuadores básicos. De esta forma es posible utilizar los LEDs para notificar cambios en el sistema o situaciones indeseadas. Por ejemplo, podemos utilizar la iluminación de un LED para indicar que la temperatura es demasiado alta.

## Ejercicio 1

*Apaga y enciende un LED de cualquier color cada segundo con el objetivo de simular un parpadeo.*

Información adicional: para resolver este ejercicio es fundamental utilizar las funciones **pinMode** [1], **digitalWrite** [2] y **delay** [3]. Adicionalmente se puede utilizar la función **digitalRead** [4], pero no es obligatoria.

## Solución del ejercicio 1

Para la resolución del ejercicio vamos a utilizar diversas funciones:

- **pinMode:** se utiliza para establecer un pin como entrada (recepción de datos) o salida (comunicación de datos). Recibe dos parámetros: un número entero que representa el PIN utilizado y el modo (input o output).
- **digitalWrite:** se utiliza para comunicarle al pin un valor. Para encender y apagar un LED utilizamos dos constantes:
  - **HIGH:** para pines de 3.3V es un valor superior a 2V. Equivalente a un 1.
  - **LOW:** para pines de 3.3V es un valor inferior a 2V. Equivalente a un 0.

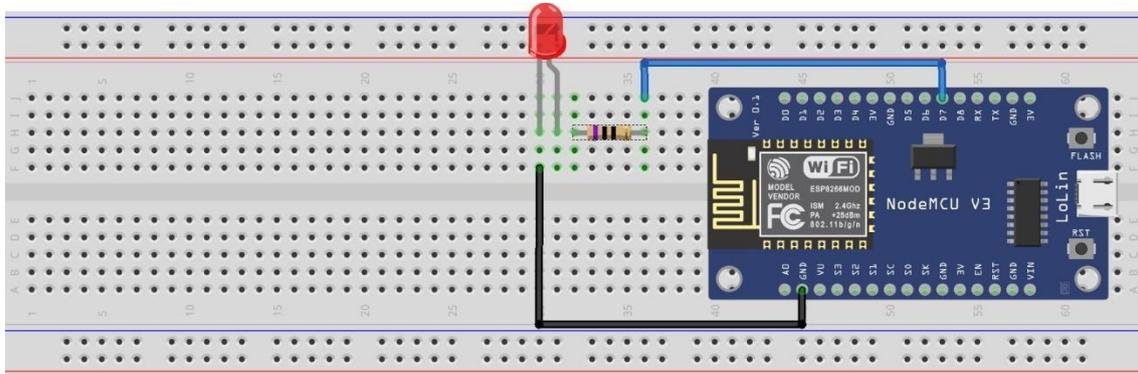
De igual forma se le indica el valor del PIN y el dato en forma de voltaje con las constantes LOW o HIGH.

- **digitalRead:** opcionalmente se puede utilizar esta función para leer el valor actual y no guardar el estado del LED (encendido o apagado).

El código del ejercicio se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_1/Ejercicio\\_1.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_1/Ejercicio_1.ino)

Y el esquemático asociado a este ejercicio se muestra en la Ilustración 16.



*Ilustración 16. Esquemático del circuito del ejercicio 1*

#### 4.1.2.LED RGB

Un LED RGB, siglas en inglés que representan los colores Rojo, Verde y Azul, tiene la habilidad de emitir estos tres colores. Este dispositivo alberga internamente tres diodos emisores, cada uno diseñado para producir uno de estos colores específicos. A diferencia del LED anterior, este tiene cuatro patillas, y debemos usar tres resistencias, una para cada color. En la Ilustración 17 se puede ver un ejemplo de este tipo de actuador.



*Ilustración 17. LED RGB*

Está compuesto por los siguientes pines:

- **Vin:** alimentación a 3.3V.
- **R:** pin para color rojo.
- **G:** pin para color verde.

- **B:** pin para color azul.

En este caso, las características de este tipo de LED son las siguientes:

- **Corriente máxima:** 20 mA.
- **Voltajes:**
  - Color rojo: 2V.
  - Color azul: 3.2V.
  - Color verde: 3.2V.
- **Temperatura de operación:** -25° a 85° C.

## Ejercicio 2

*Utiliza el LED RGB, enciéndelo y cambia su color cada segundo*

Información adicional: se puede iluminar de forma analógica o digital. Hay funciones analógicas equivalentes a las del ejercicio anterior. Por ejemplo, **analogWrite** [5].

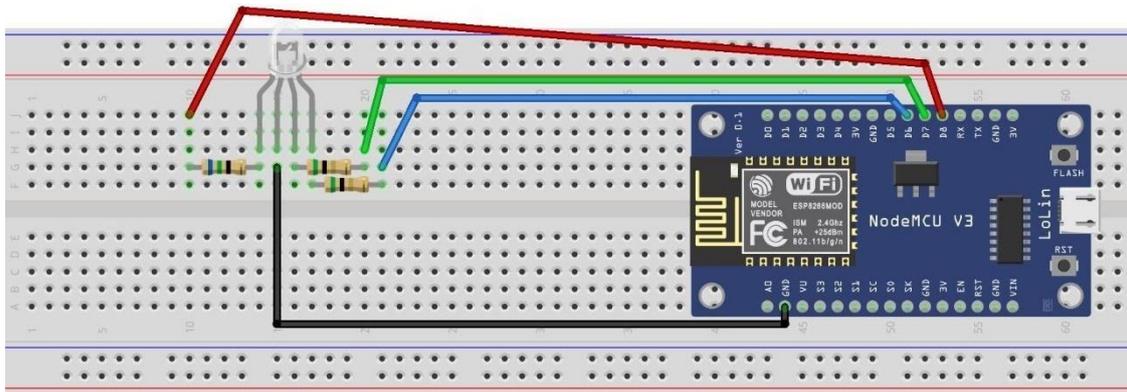
## Solución del ejercicio 2

Para la solución de este ejercicio se ha optado por usar entradas digitales. Por tanto, con este proyecto sólo podremos visualizar  $2^3 = 8$  colores.

El código del ejercicio se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio 2/Ejercicio 2.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio%202/Ejercicio%202.ino)

Y en la Ilustración 18 se muestra el circuito utilizado para el código del ejercicio 2.

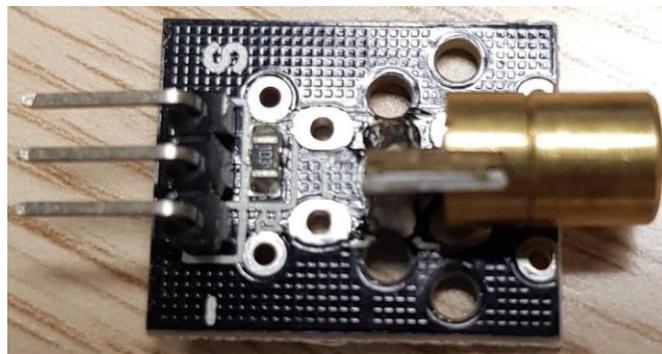


*Ilustración 18. Esquemático del ejercicio 2*

#### 4.1.3. Láser

Un dispositivo láser (ver Ilustración 19) emite una luz de alta potencia de color rojo. Este dispositivo tiene un cabezal para darle mayor resistencia y disipar el calor producido por este.

**MUY IMPORTANTE:** este dispositivo provoca quemaduras y daño temporal o permanente en los ojos.



*Ilustración 19. Láser modelo ky-008*

Sus pines son los siguientes:

- **S:** activación del láser (como un led).
- **VCC:** alimentación a 5V.
- **GND.**

Sus principales características son las siguientes:

- **Voltaje de funcionamiento:** 5 V.
- **Longitud de onda:** 650nm.

- **Distancia:** 2 metros.

### Ejercicio 3

*Realiza un parpadeo similar, pero con una temporalidad diferente:*

1. *Haz que el láser emita luz durante 5 segundos.*
2. *Pasado este tiempo apágalo durante 1 segundo y vuelve al punto 1.*

*Comprueba si la distancia efectiva es de 2 metros de forma aproximada. ¿Crees que se puede alimentar el láser directamente con una salida de corriente de la placa de desarrollo esp8266 o esp32?*

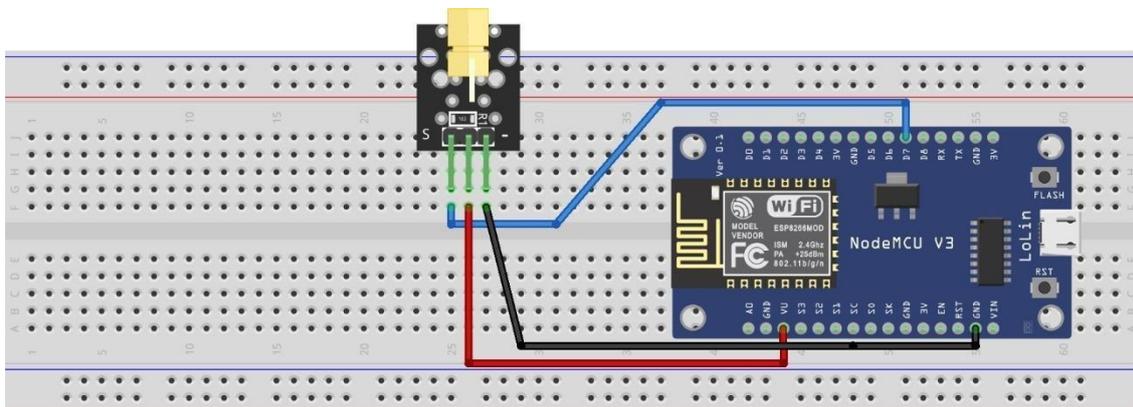
### Solución del ejercicio 3

En este proyecto hacemos algo parecido al ejercicio del LED, un parpadeo con el láser. Lo más importante es que este dispositivo necesita un voltaje de 5V y las salidas de corriente de nuestra placa sólo tienen 3.3V. Por tanto, es necesario utilizar un elevador de voltaje tal y como se explicó en la sección 3.3.

El código del ejercicio se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_3/Ejercicio\\_3.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_3/Ejercicio_3.ino)

Y el esquemático del ejercicio 3 se encuentra en la Ilustración 20. Para simplificar el circuito no se incluye el elevador de voltaje.



*Ilustración 20. Esquemático ejercicio 3*

#### 4.1.4. Pantalla LCD con I2C

La pantalla LCD de 16x2 basada en el controlador HD44780 de Hitachi es un accesorio muy utilizado en proyectos con Arduino y otros microcontroladores. No obstante, este tipo de pantalla necesita una gran cantidad de pines del microcontrolador para su funcionamiento, ya que se comunica a través de un bus paralelo. Afortunadamente, hay una solución a este inconveniente: un adaptador basado en el PCF8574 que permite conectar la pantalla a la placa usando solo dos líneas digitales a través del bus I2C (ver Ilustración 21).



*Ilustración 21. LCD con convertor I2C*

Los pines de este dispositivo son los siguientes:

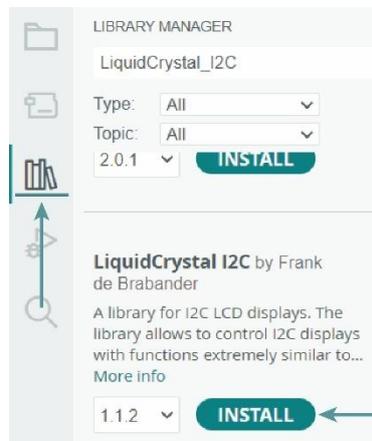
- **VCC:** alimentación a 5V.
- **GND.**
- **SDA, SCL:** pines para alimentar la matriz 16x2.

Sus principales características son las siguientes:

- **Voltaje de funcionamiento:** 5V.
- **Zona de escritura:** 2 filas de 16 caracteres.
- **Tamaño de cada carácter:** 5.23 x 3 mm
- **Gran variedad de caracteres:** es capaz de mostrar letras, números, símbolos especiales, y hasta ocho caracteres personalizados por el usuario en color blanco.
- **Interfaz paralela:** puede funcionar en un modo de 8 bits, o en un modo de 4 bits para conservar los pines del microcontrolador.

Para mandar información a la LCD, necesitamos instalar y utilizar una biblioteca de terceros llamada [LiquidCrystal I2C](#). Para instalarla debemos buscarla por su nombre en el gestor de bibliotecas en el panel del lateral izquierdo tal y como aparece en la

Ilustración 22. No obstante, existen otro tipo de bibliotecas similares que también pueden ser utilizadas.



*Ilustración 22. Instalación de la biblioteca LiquidCrystal\_I2C*

Para su correcto funcionamiento, es necesario conectar el pin **SDA** a **D2**, y el pin **SCL** a **D1** (la biblioteca utiliza esos pines en el caso de la placa ESP8266).

## Ejercicio 4

*Muestra en la pantalla LCD en la primera fila la palabra “Hola” y en la segunda “mundo!” y, además, incluye un carácter especial y modificalo cada segundo (haz una especie de animación).*

*Información adicional: consulta la documentación de la librería [LiquidCrystal\\_I2C](#) [6].*

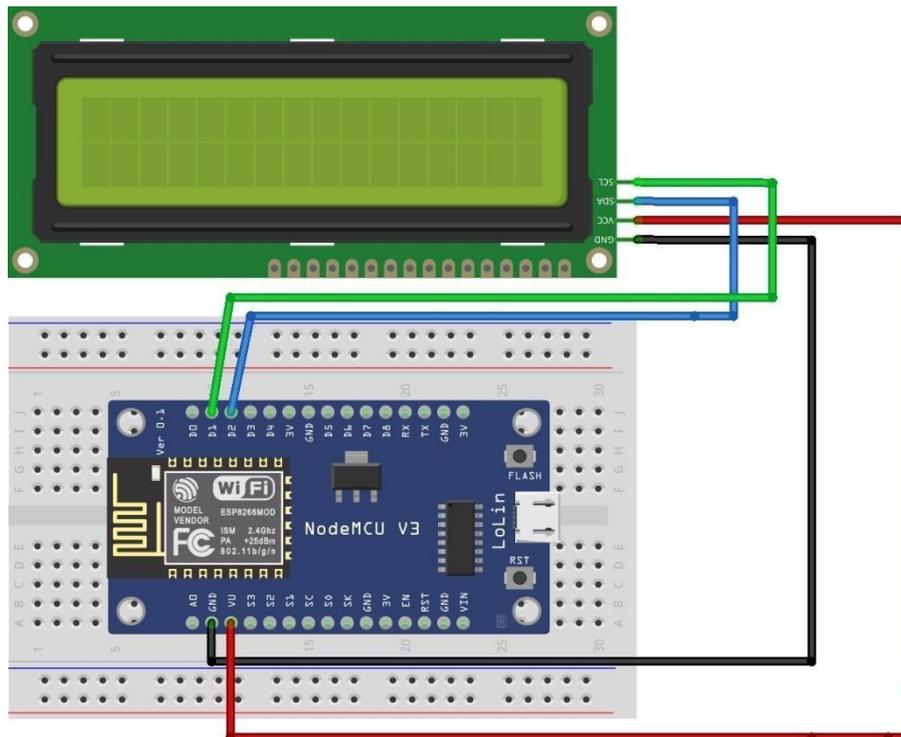
## Solución del ejercicio 4

De igual forma, es necesario alimentar la pantalla LCD con un voltaje de 5V. Por tanto, es necesario utilizar un elevador de voltaje tal y como se explicó en la sección 3.3.

El código del ejercicio se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_4/Ejercicio\\_4.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_4/Ejercicio_4.ino)

Y el esquemático de este ejercicio se encuentra en la Ilustración 23.



*Ilustración 23. Esquemático del ejercicio 4*

#### 4.1.5. Servomotor

Un servomotor es un pequeño dispositivo rotativo o motor que proporciona un control exacto de la posición angular. Este tipo de dispositivos es muy común encontrarlos y en este curso utilizaremos el modelo SG-90 (ver Ilustración 24).



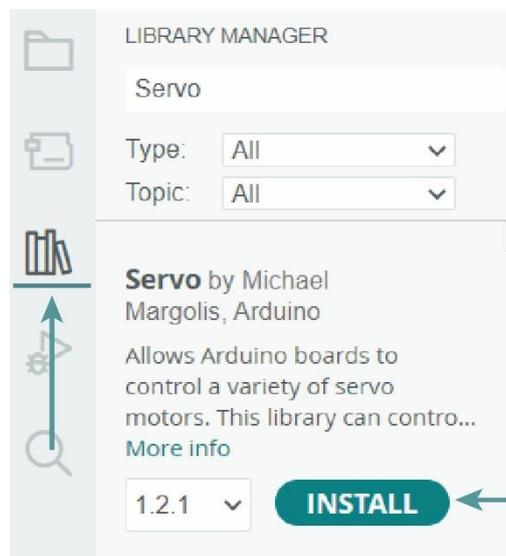
*Ilustración 24. Servomotor SG90*

Este actuador incluye tres brazos, y sus características más importantes, son las siguientes:

- **Tensión de operación:** entre 4.8 y 6V.

- **Capacidad de rotación:** de 0° a 180°.
- **Velocidad de operación:** 0.1s/60°
- **Conexiones:**
  - **Vu:** alimentación (cable rojo del servomotor)
  - **GND:** tierra (cable marrón del servomotor)
  - **PIN DIGITAL:** señal (cable naranja del servomotor)

Para poder interactuar con el servomotor es necesario instalar y utilizar la biblioteca [Servo](#). Para ello, lo instalaremos de la misma forma que en secciones anteriores (ver Ilustración 25).



*Ilustración 25. Instalación de la librería Servo*

## Ejercicio 5

*Mueve el servomotor de forma infinita primeramente de 0° a 180° realizando esperas de 15 ms entre cada ángulo, y posteriormente a la inversa de 180° a 0°.*

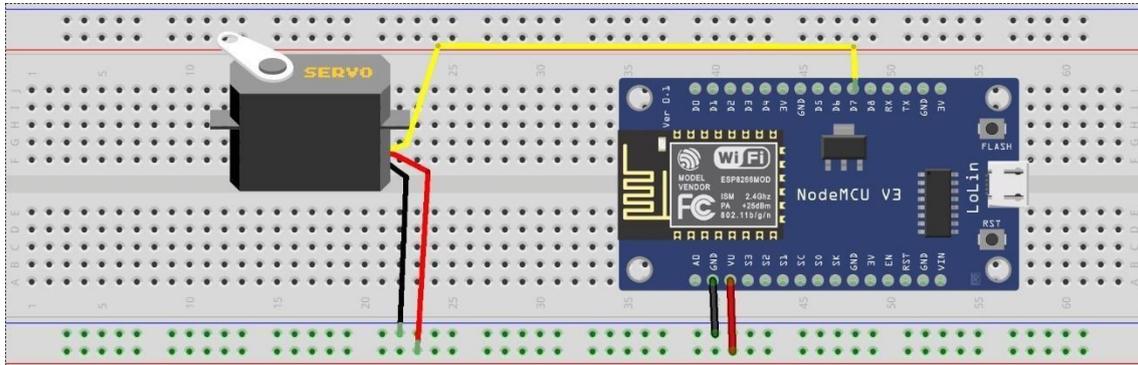
*Información adicional: consulta la documentación de la librería [Servo](#) [7].*

## Solución del ejercicio 5

De igual forma, es necesario alimentar el servomotor con un voltaje de 5V. Por tanto, es necesario utilizar un elevador de voltaje tal y como se explicó en la sección 3.3. A continuación, en la Ilustración 26 se incluye el esquemático del circuito. Para simplificarlo no se incluye el elevador de voltaje.

Además, el código se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_5/Ejercicio\\_5.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_5/Ejercicio_5.ino)



*Ilustración 26. Esquemático del ejercicio 5*

#### 4.1.6. Pulsador

Los pulsadores son elementos externos que permiten controlar de manera física las acciones que ejecutará nuestro sistema, ya sea permitiendo o interrumpiendo la señal de entrada donde están instalados. Son componentes cruciales en la mayoría de los proyectos y son sencillos de utilizar y programar, aunque es importante prestar atención a ciertos detalles al emplearlos. Estos dispositivos también se conocen como actuadores eléctricos.

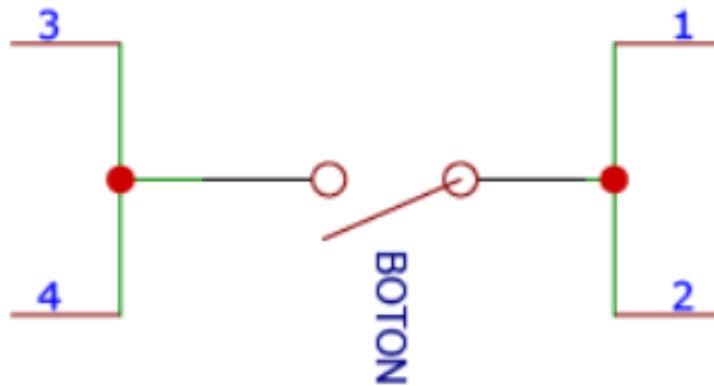


*Ilustración 27. Pulsador*

Estos dispositivos constan de cuatro pines (ver Ilustración 27):

- **Pin 1, 2:** están unidos internamente.
- **Pin 3, 4:** están unidos internamente.

Para que el sistema funcione correctamente, se alimenta el positivo y el negativo por diferentes pares de patillas. Por ejemplo, el pin 1 en el positivo y el pin 3 en el negativo, y normalmente estos pines son los que se encuentran juntos en el mismo lateral tal.



*Ilustración 28. Diagrama de un botón pulsador (fuente de <https://www.automatizacionparatodos.com/push-button-con-arduino/>)*

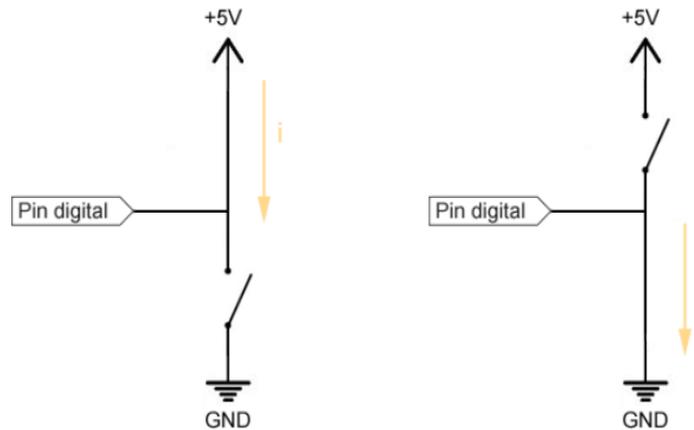
Las características principales de un pulsador son la siguientes:

- **Temperatura de operación:** -30° a + 70°C.
- **Voltaje de funcionamiento:** 3.3-5V.

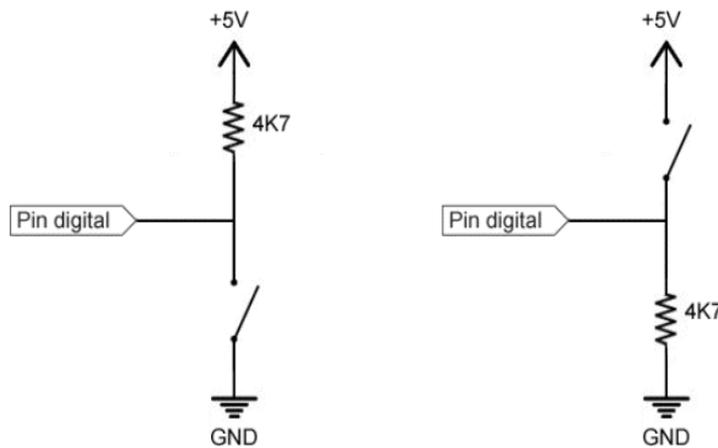
Para la utilización de un pulsador se debe de tener en cuenta algunos conceptos. Un pulsador es posible utilizarlo de dos modos diferentes:

- **PULL UP:** la obtención del valor a través de una entrada digital se realiza en la entrada positiva del botón. Cuando se pulsa el botón recibimos en la entrada un valor LOW.
- **PULL DOWN:** la obtención del valor a través de una entrada digital se realiza en la salida negativa del botón. Cuando se pulsa el botón recibimos en la entrada un valor HIGH.

Aunque se utilice un esquema PULL UP o PULL DOWN (ver Ilustración 29), al puentear con un cable para recoger la señal del botón, estamos creando un cortocircuito. Por tanto, debemos incluir una resistencia antes de conectar el positivo al botón en el caso de la opción pull up, y una en la parte negativa si estamos usando la versión pull down (ver Ilustración 30).



*Ilustración 29. Esquema del montaje con cortocircuito (fuente de <https://www.luisllamas.es/leer-un-pulsador-con-arduino/>)*



*Ilustración 30. Esquema del circuito para evitar un cortocircuito (fuente de <https://www.luisllamas.es/leer-un-pulsador-con-arduino/>)*

## Ejercicio 6

Ilumina un LED con la señal de un pulsador. ¿Qué tipo deberíamos utilizar? ¿PULL UP o PULL DOWN? ¿Se podría encender el LED sin código?

## Solución del ejercicio 6

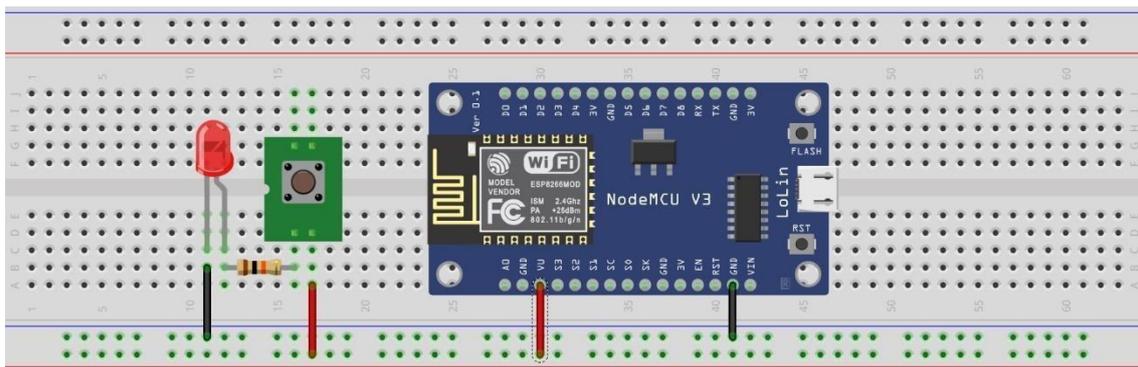
En este caso, el proyecto consiste en iluminar un LED con la señal del pulsador. En el esquema se utiliza el sistema PULL DOWN (tiene más sentido iluminar un led cuando se pulsa el botón), pero se puede utilizar perfectamente el otro sistema.

Como norma general, la resistencia debe ser al menos 10 veces menor que la impedancia del pin digital, que en el caso de la placa ESP8266 es entorno a  $1M\Omega$ . Además, se ha de tener en cuenta el consumo y la corriente que se le va a suministrar a la placa, mientras menor sea la resistencia, mayor será la corriente y como resultado mayor será el consumo y la corriente que entrará a la placa. Sin embargo, no se puede poner una resistencia excesivamente alta para tener un consumo bajo porque si la corriente es muy pequeña puede ser que no llegué a accionar los pines de la placa.

A continuación, se incluye el enlace con el código del ejercicio resuelto:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_6/Ejercicio\\_6.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_6/Ejercicio_6.ino)

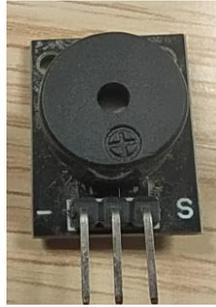
En el esquemático de la Ilustración 31 se va a utilizar una resistencia de  $10K\Omega$  sin necesidad de código se comprueba el pulsador simplemente utilizando el pin Vu que suministra una alimentación de 5V.



*Ilustración 31. Esquemático del ejercicio 6*

#### 4.1.7. Zumbador pasivo

Un zumbador pasivo o un altavoz son dispositivos que transforman una señal eléctrica en sonido. Estos dispositivos no tienen circuitos internos, por lo que necesitamos proporcionar una señal eléctrica para producir el sonido que queremos. En este caso, vamos a usar el modelo KY-006 (ver Ilustración 32).



*Ilustración 32. Zumbador pasivo modelo KY-006*

El modelo cuenta con los siguientes pines:

- **S:** señal digital.
- **Pin central:** alimentación.
- **-:** tierra.

Sus características principales son las siguientes:

- **Voltaje de funcionamiento:** 1.5 - 5V.
- **Rango de generación de tonos:** 1.5 Hz– 2.5 kHz
- **Dimensiones:** 18mm x 15mm
- **Corriente de trabajo:** < 25mA

## Ejercicio 7

*Conecta el zumbador pasivo y emite tonos para crear una canción. Recrea con el zumbador la canción de tu película favorita.*

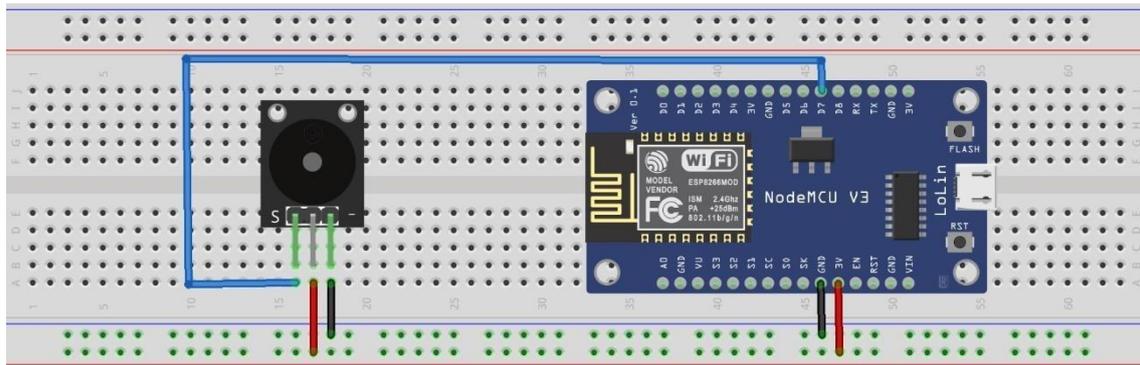
*Información adicional: para poder realizar este ejercicio es necesario dos funciones completamente nuevas **tone** [8] y **noTone** [9].*

## Solución del ejercicio 7

A continuación, en el siguiente enlace, se incluye el código solución del ejercicio:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_7/Ejercicio\\_7.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_7/Ejercicio_7.ino)

En este caso, hemos utilizado la función **tone** para emitir sonidos, y la función **noTone** para dejar de emitirlos. Además, se incluye en la Ilustración 33.



*Ilustración 33. Esquemático del ejercicio 7*

## 4.2. Sensores

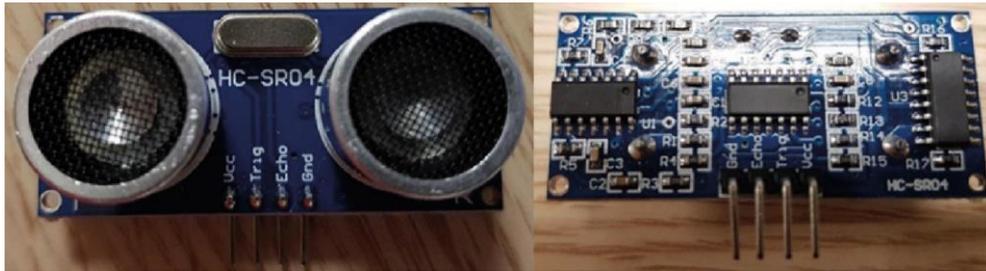
En esta sección, vamos a aprender a utilizar sensores. Estos dispositivos se caracterizan por cambiar un parámetro físico a una salida eléctrica a diferencia de los actuadores que cambian una señal eléctrica en una salida física.

En esta sección, aprenderemos a conectar y utilizar diferentes tipos de sensores:

- Sensor de ultrasonidos.
- Sensor de ambiental.
- Sensor de fuego.
- Sensor de gas.
- Sensor de vibraciones.
- Sensor de movimiento infrarrojo pasivo.
- Sensor fotovoltaico.
- Sensor infrarrojo.
- Sensor de sonido.
- Sensor para detección de líneas.
- Reloj y calendario.
- Sensor de humedad en el suelo o higrómetro.
- Sensor de lluvia.
- Acelerómetro triaxial.
- Cámara termal.

#### 4.2.1. Sensor de ultrasonido

Los sensores de ultrasonido, también conocidos como sensores ultrasónicos, son detectores de proximidad que operan sin contacto mecánico y pueden identificar objetos a distancias que varían desde unos pocos centímetros hasta varios metros. El sensor emite un sonido y calcula el tiempo que tarda en regresar la señal. Un ejemplo de este tipo de sensores se puede ver en la Ilustración 34.



*Ilustración 34. Sensor de ultrasonidos modelo HC-SR04*

En este curso utilizaremos el modelo HC-SR04, el cual cuenta con los siguientes pines:

- **VCC:** alimentado a 5V.
- **Trig:** emite el sonido.
- **Echo:** nos proporciona el tiempo que tardado en volver en microsegundos ( $\mu$ s).
- **GND:** tierra.

Es importante saber que la velocidad el sonido es 340 m/s. Dado que el pin echo nos devuelve el tiempo que ha tardado en volver en  $\mu$ s, tenemos que cambiar de unidad la velocidad del sonido, siendo 0.034 cm/ $\mu$ s la correcta. Para calcular la distancia que ha recorrido, debemos despejar la S de la fórmula. Se debe tener en cuenta que la distancia S es la suma de la distancia de ida más vuelta, por tanto, si se quiere saber la distancia a la que se encuentra el objeto se tiene que dividir S entre 2.

$$T \text{ (tiempo del echo)} = \frac{S \text{ (distancia total)}}{V \text{ (velocidad del sonido)}}$$

Además, el sensor cuenta con las siguientes características:

- **Voltaje de funcionamiento:** 3–5V.

- **Ángulo de medición:** 30°.
- **Ángulo de medición efectivo:** 15°.
- **Rango mínimo de detección:** 1.7 cm.
- **Rango máximo de detección:** 4 m (puede ser utilizado a más de 4m, pero el fabricante no garantiza una buena medición).
- La **precisión** puede variar entre los 3mm.
- **Frecuencia de trabajo:** 40KHz.
- **Duración mínima del pulso de disparo** (nivel TTL): 10  $\mu$ S.
- **Duración del pulso eco de salida** (nivel TTL): 100-25000  $\mu$ S.
- **Tiempo mínimo de espera entre una medida y el inicio de otra:** 20 mS.

## Ejercicio 8

Lee los valores del sensor y muestra por pantalla cuando haya un obstáculo entre el rango mínimo y máximo del sensor y, además, muestra también la distancia en centímetros.

## Solución del ejercicio 8

A continuación, se incluye el código para la solución de este ejercicio y el esquemático (ver Ilustración 35):

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_8/Ejercicio\\_8.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_8/Ejercicio_8.ino)

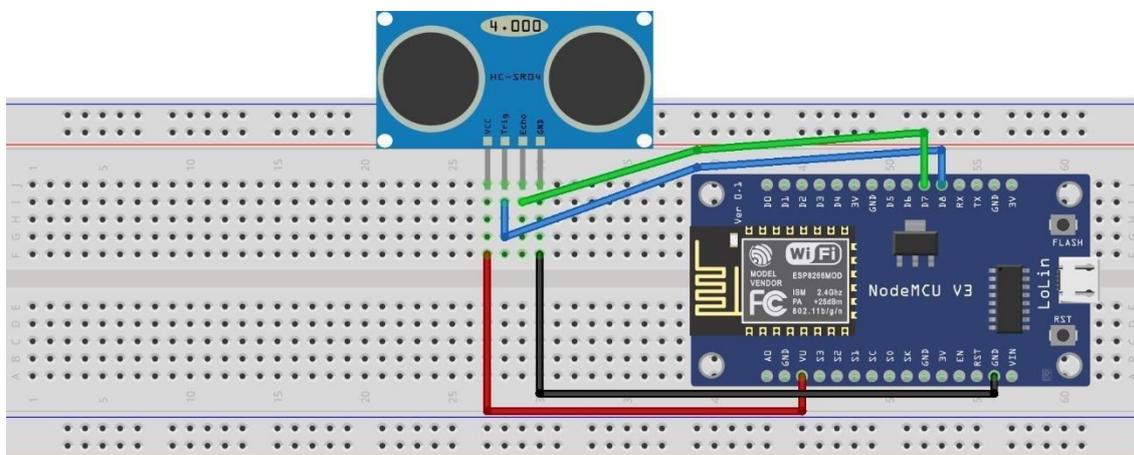
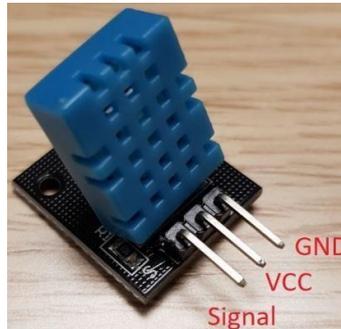


Ilustración 35. Esquemático del ejercicio 8

#### 4.2.2. Sensor ambiental

Los sensores ambientales digitales suelen medir diferentes variables como la humedad, la temperatura o incluso la presión atmosférica. En este caso, vamos a utilizar el modelo DHT11 de la Ilustración 36 que se encarga de medir temperatura y humedad.



*Ilustración 36. Sensor de humedad modelo DHT11*

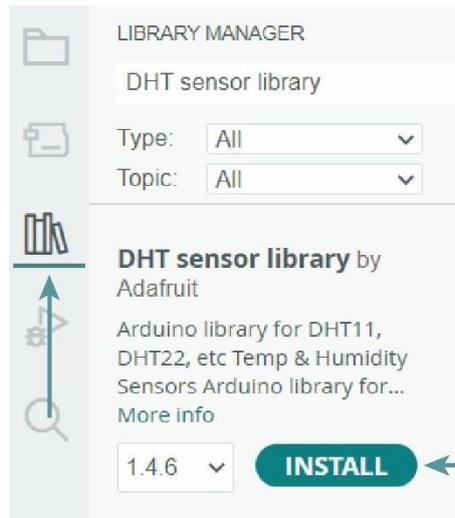
Este modelo consta de tres pines:

- **VCC:** alimentación.
- **Data:** a través del cual obtenemos los datos.
- **GND:** tierra.

Sus principales características son las siguientes:

- **Voltaje de operación:** 3-5V.
- **Rango de medición de temperatura:** 0 a 50 °C
- **Precisión de medición de temperatura:**  $\pm 2.0$  °C
- **Resolución Temperatura:** 0.1°C
- **Rango de medición de humedad:** 20% a 90% RH.
- **Precisión de medición de humedad:** 5% RH.
- **Resolución Humedad:** 1% RH
- **Tiempo entre medición:** 1 segundo.

Para poder recoger datos del sensor es necesario instalar y utilizar la biblioteca de terceros denominada [DHT sensor library](#). Se instala de la misma forma que las anteriores.



*Ilustración 37. Instalación de la biblioteca DHT sensor library*

## Ejercicio 9

Mide con el sensor ambiental la temperatura y humedad del ambiente. Cada dos segundos consulta sus valores y muéstralos por pantalla.

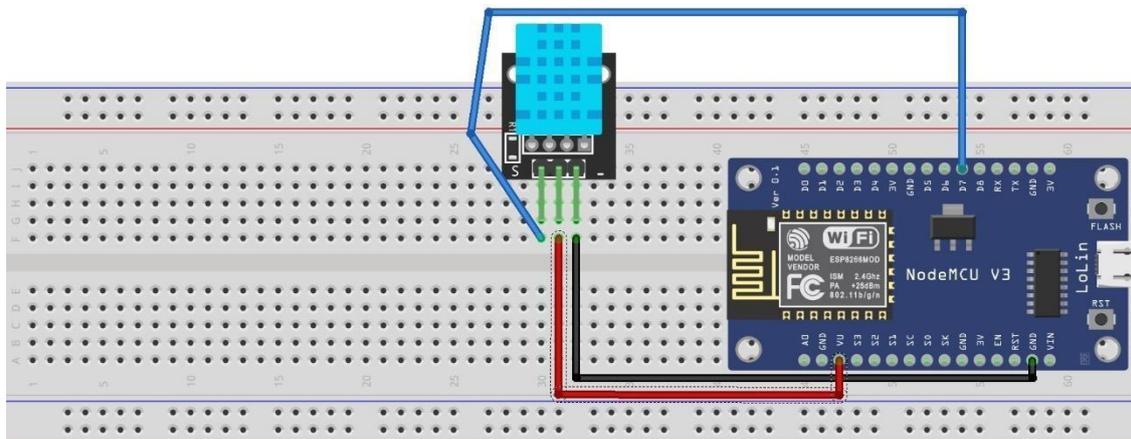
Información adicional: consulta la documentación de la librería **DHT sensor library** [10].

## Solución del ejercicio 9

A continuación, se incluye el código con la solución del ejercicio:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_9/Ejercicio\\_9.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_9/Ejercicio_9.ino)

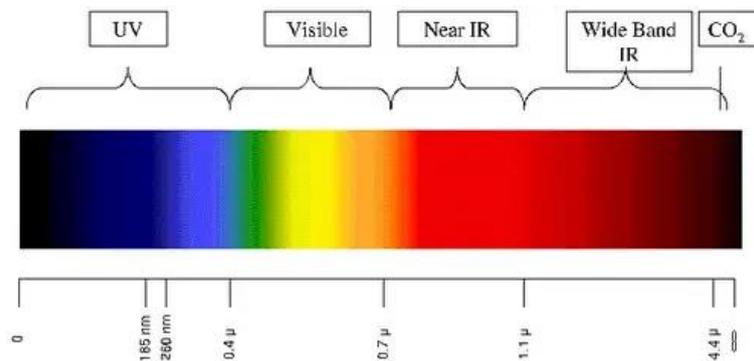
También, en la Ilustración 38, se incluye también el esquemático del mismo.



*Ilustración 38. Esquemático del ejercicio 9*

#### 4.2.3. Sensor de fuego

Estos tipos de sensores pueden identificar la presencia de fuego a una corta distancia. Funcionan detectando el espectro de una llama (ver Ilustración 39). Este espectro varía dependiendo de los elementos que participan en la reacción. Por ejemplo, en la combustión de productos con carbono en presencia de oxígeno, se observan dos picos característicos: uno en el rango ultravioleta con longitudes de onda de 185-260 nm (nanómetros) y otro en el infrarrojo con longitudes de onda de 4400-4600 nm



*Ilustración 39. Espectro de emisión (fuente de <https://www.luisllamas.es/detector-llama-arduino/>)*

Como veremos en las características, poco tiene que ver el rango correspondiente a la combustión con el rango que detecta el sensor, con lo cual nos dará numerosos falsos positivos. Para este curso utilizaremos el modelo HL-01 que aparece en la Ilustración 40.



*Ilustración 40. Sensor de fuego modelo HL-01*

En este caso, estos dispositivos son muy útiles para detectar fuego a corta distancia. Por ejemplo, se pueden situar cerca de maquinaria o zonas delicadas para que nos comuniquen de forma rápida si algo ha empezado a arder. De esta forma es posible actuar de forma más rápida. En una situación real este tipo de sensores es necesario sustituirlos por otros nuevos, ya que al estar en contacto con la llama el dispositivo deja de realizar su función

Este modelo consta de tres pines:

- **D0**: salida digital que puede ser regulada con un potenciómetro.
- **A0**: salida con los valores de voltaje.
- **VCC**: alimentación.
- **GND**: tierra.

Sus características son las siguientes:

- **Voltaje de funcionamiento**: 5V.
- **Distancia máxima**: 80 cm.
- **Ángulo de detección**: 60°.
- **Sensible a la radiación infrarroja en el rango de longitudes de onda**: 760 - 1100nm.

## Ejercicio 10

*Realiza un pequeño programa que sea capaz de detectar fuego y avisar por pantalla. Ahora que tenemos a nuestra disposición muchos actuadores, en vez de mostrar el aviso por pantalla puedes encender un LED hasta que no se detecte peligro.*

## Solución 10

A continuación, se incluye la solución a este ejercicio. La solución es totalmente digital y hemos regulado el umbral de detección a través del potenciómetro. Para probar su funcionamiento es necesario acercarse a una llama. Por ejemplo, se puede utilizar un encendedor, pero a una distancia prudencial para no quemar el sensor.

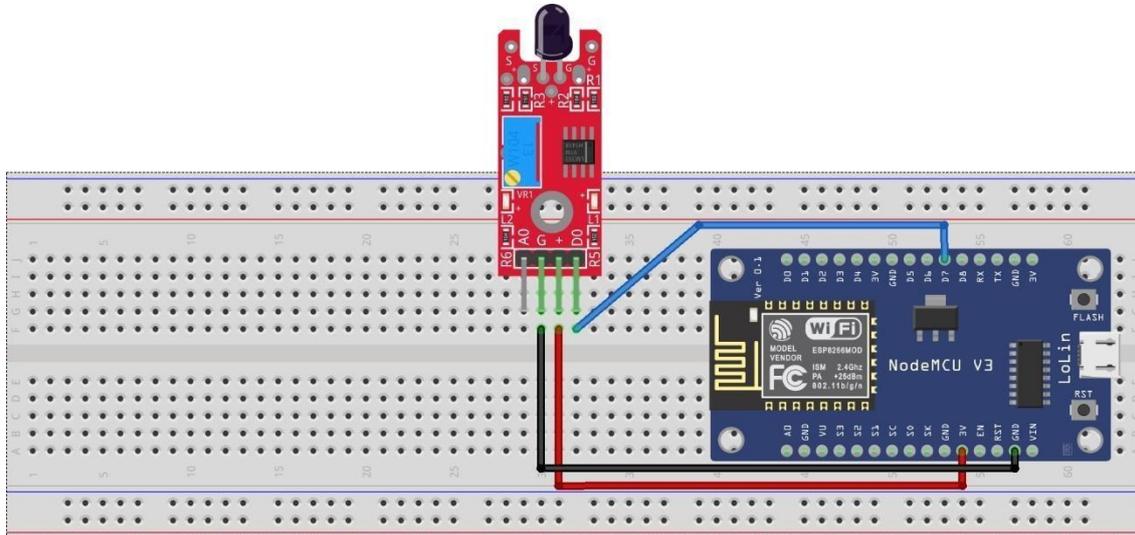
Si se opta por la solución analógica, es muy importante saber que trabajamos con los datos en crudo y, en este caso, nosotros deberíamos definir el umbral de detección. Al ser un sensor que trabaja en 5V, la salida analógica también estará en 5V, en cambio la entrada de pines soporta como máximo 3.3V, con lo cual estaríamos perdiendo información. Este caso avanzado se verá con más detenimiento en secciones posteriores.

En ambos casos, es necesario alimentar el sensor con un voltaje de 5V. Por tanto, es necesario utilizar un elevador de voltaje tal y como se explicó en la sección 3.3.

El código de este ejercicio se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_10/Ejercicio\\_10.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_10/Ejercicio_10.ino)

Y el esquemático se muestra en la Ilustración 41.



*Ilustración 41. Esquemático del ejercicio 10*

#### 4.2.4. Sensor de gas

Los sensores de gas son capaces de detectar concentraciones de gas natural en el aire, con rangos que van desde 300 hasta 10,000 partículas por millón. Existe una variedad de sensores de gas comerciales de bajo costo diseñados para medir gases específicos. Durante el curso, se empleará el sensor de gas MQ2 (ver Ilustración 42), que se utiliza para identificar fugas de gas tanto en aplicaciones domésticas como industriales.



*Ilustración 42. Sensor de gas MQ-2*

Este modelo consta de 4 pines:

- **A0:** salida analógica que proporciona un voltaje de salida en función de la concentración del humo o gas.
- **D0:** salida digital que se ajusta con el potenciómetro.
- **VCC:** alimentación.
- **GND:** tierra.

Sus características principales son las siguientes:

- **Voltaje de funcionamiento:** 5V.
- Respuesta rápida y alta sensibilidad.
- **Rango de detección:** 300 a 10000 ppm.
- **Gas característico:** 1000ppm, Isobutano.
- **Resistencia de sensado:** 1K $\Omega$  50ppm Tolueno a 20K $\Omega$ .
- **Tiempo de Respuesta:**  $\leq 10s$
- **Tiempo de recuperación:**  $\leq 30s$
- **Temperatura de trabajo:** -20 °C ~ +55 °C
- **Humedad:**  $\leq 95\%$  RH.
- **Contenido de oxígeno ambiental:** 21%
- **Consume:** menos de 150mA a 5V.

En el mercado hay una serie de sensores similares que son capaces de detectar otros tipos de gases. A continuación, en la Ilustración 43 se muestra una tabla donde se especifican todos los tipos de sensores y los gases que son capaces de detectar.

	MQ														
	2	3	4	5	6	7	8	9	135	214	216	303A	306A	307A	309A
Methane	x		x							x					
Butane	x				x								x		
LPG	x			x	x								x		
Smoke	x	x										x			
Alcohol		x										x			
Ethanol		x										x			
CNG Gas			x												
Natural Gas				x						x	x				
Carbon Monoxide						x		x						x	x
Hydrogen Gas							x								
Coal Gas								x			x				
Liquefied Gas								x							
Air Quality									x						
Flam-mable Gas															x

Ilustración 43. Tabla con el tipo de sensores de gas y su tipo de detección

## Ejercicio 11

Utiliza un sensor comercial de gas, y muestra el valor en crudo de voltaje. ¿Habría alguna forma de pasar de ese valor en crudo a Partículas Por Millón (ppm)? Dado que la salida analógica es a 5V, ¿crees que se puede conectar directamente a un pin de la placa de desarrollo?

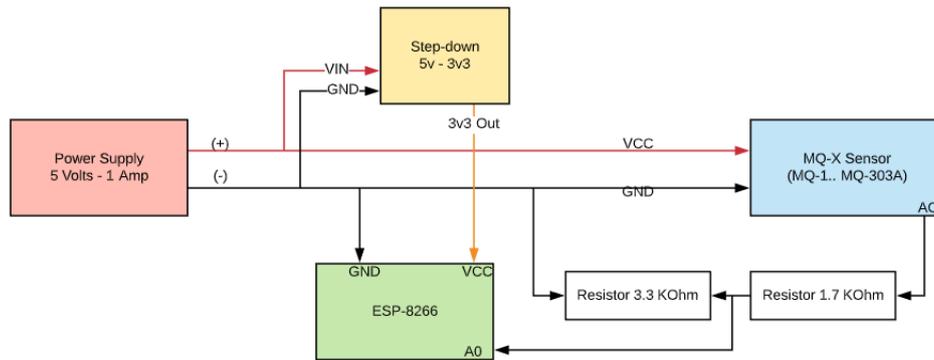
Información adicional: para realizar la conversión de voltaje a ppm existen bibliotecas de terceros como **MQSensorsLib** [11]. En este repositorio existe información para conectar debidamente un sensor de este tipo a una placa con entradas de 3.3V y la forma de calibrarlo en función de su tipo.

## Solución del ejercicio 11

En este caso, vamos a realizar dos tipos de soluciones: uno para la medición directa y otro para utilizar una biblioteca que es capaz de discernir entre gases y mostrar el valor como ppm de un gas en concreto. Tanto en un caso como en otro, debemos tener en cuenta que la entrada analógica en la placa ESP8266, sólo soporta voltajes de entre 0 y 3.3 V, y que el voltaje de entrada proveniente de la salida A0 en el sensor es

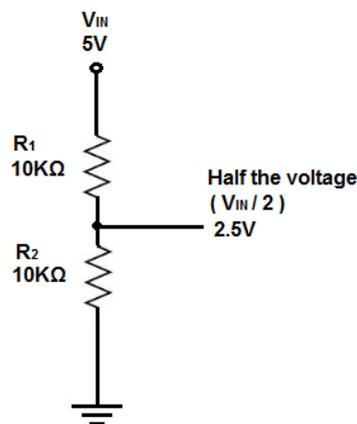
de 5V. Es decir, debemos modificar el voltaje de salida para que los valores medidos sean correctos.

Un modo de hacer esto es a través de dos resistencias: R2 será de 3.3K  $\Omega$  y R1 será de 1.7k  $\Omega$ , tal y como viene reflejado en la Ilustración 44.



*Ilustración 44. Esquema para la reducción de voltaje en la entrada analógica (fuente de <https://github.com/miguel5612/MQsensorsLib>)*

¿Cómo sabemos que este esquema nos reduce el voltaje de entrada? Porque utiliza un circuito divisor del voltaje (ver Ilustración 45).



*Ilustración 45. Circuito divisor del voltaje (fuente de <https://www.learningaboutelectronics.com/Articulos/Como-reducir-voltaje-con-resistencias.php>)*

El cálculo de la tensión de salida se corresponde con la siguiente ecuación.

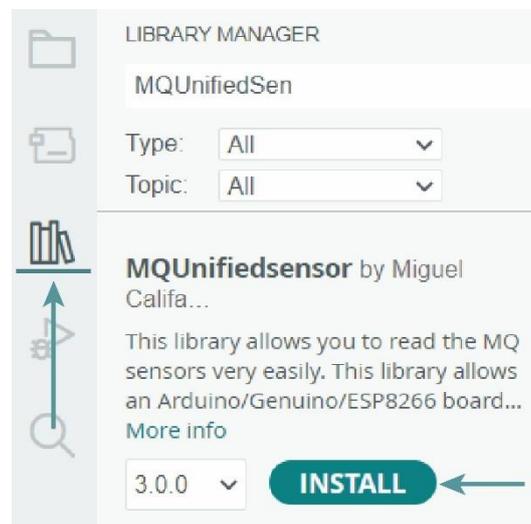
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

La primera solución consiste en leer el valor del voltaje y convertirlo a partículas por millón (ppm), sabiendo que el valor mínimo que es capaz de detectar son 300 ppm y el valor máximo son 10.000 ppm.

La solución de esta primera parte se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_11.1/Ejercicio\\_11.1.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_11.1/Ejercicio_11.1.ino)

Para la segunda solución utilizaremos la librería [MQSensorsLib](#), y mediremos el valor de ppm para el gas monóxido de carbono (CO). Para poder utilizarla es necesario instalar la biblioteca tal y como se muestra en la Ilustración 46.



*Ilustración 46. Instalación de la biblioteca MQSensorsLib*

Para discernir en la detección de un gas u otro, es necesario indicar los valores en los que está comprendido el gas. Para cada tipo de sensor la biblioteca nos proporciona una tabla con los gases que es capaz de detectar y sus valores (la tabla para el MQ2 viene definida en el código a modo de comentario). A continuación, se incluye el enlace al código asociado a esta segunda parte:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_11.2/Ejercicio\\_11.2.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_11.2/Ejercicio_11.2.ino)

Para ambas soluciones se ha utilizado el circuito de la Ilustración 47. En la imagen se ha utilizado un sensor equivalente, ya que en el software no se incluía el sensor MQ2. También, aunque por simplicidad no se incluya en el circuito, es necesario alimentar el sensor con un voltaje de 5V. Por tanto, es necesario utilizar un elevador de voltaje tal y como se explicó en la sección 3.3.

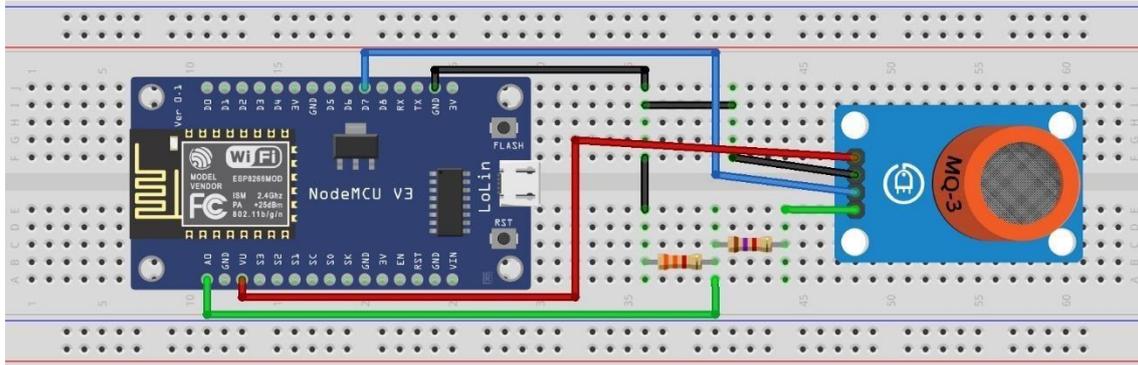


Ilustración 47. Esquemático del ejercicio 11

#### 4.2.5. Sensor de vibraciones

Un sensor de vibración es un dispositivo que responde a movimientos abruptos, impactos o vibraciones, pero no a movimientos constantes o graduales. Estos sensores son ampliamente utilizados para detectar cambios potenciales en el funcionamiento de maquinaria en la industria, o incluso para identificar si se han manipulado objetos.

En este curso utilizaremos el modelo SW18010P que se muestra en la Ilustración 48. Aunque existen sensores de vibración más precisos como es el caso del modelo SW-420 (ver Ilustración 49). Ambos modelos funcionan de la misma forma y tienen el mismo tipo de pines.

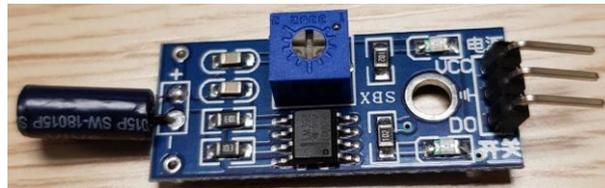


Ilustración 48. Sensor de vibraciones modelo SW18010P

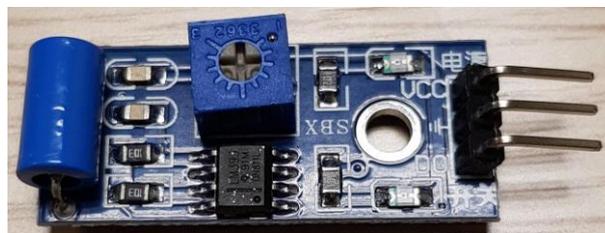


Ilustración 49. Sensor de vibraciones modelo SW-420

Estos sensores de vibraciones constan de los siguientes pines:

- VCC: alimentación.
- GND: tierra.

- **D0:** salida digital que se ajusta con el potenciómetro.

Estos modelos tienen como principales características:

- **Voltaje de funcionamiento:** 3.3-5V.
- Salida digital regulada por un potenciómetro.

## Ejercicio 12

*Conecta el sensor con la placa y haz que se ilumine un LED cuando detecte vibración. Para probar un caso real, sitúa el circuito en una taza y simula que vas a beber de ella. ¿Se ilumina el LED? Regula el potenciómetro para que seas capaz de detectar ese movimiento.*

## Solución del ejercicio 12

A continuación, se incluye el código para el ejercicio 12:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_12/Ejercicio\\_12.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_12/Ejercicio_12.ino)

Para adherirlo a una taza, puedes utilizar el mismo adhesivo de la protoboard. También puedes utilizar cinta de doble cara. Inicialmente deberas calibrar el potenciómetro para detectar de forma correcta este movimiento. En la Ilustración 50 se incluye el esquemático del circuito.

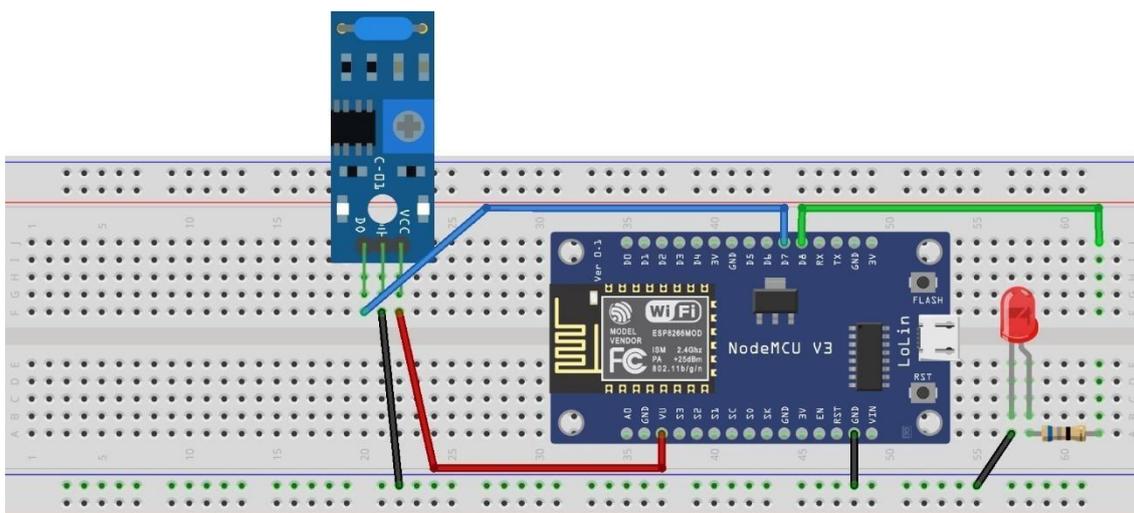


Ilustración 50. Esquemático del ejercicio 12

#### 4.2.6. Sensor de movimiento infrarrojo pasivo

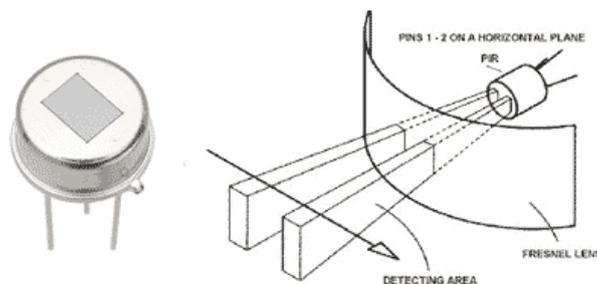
Los sensores infrarrojos pasivos (PIR) son dispositivos diseñados para detectar movimiento. Son económicos, compactos, de bajo consumo energético y fáciles de manejar. Por estas razones, se utilizan comúnmente en juguetes, aplicaciones de domótica y sistemas de seguridad. A continuación, se incluye una ilustración del modelo utilizado en el curso (ver Ilustración 51).



*Ilustración 51. Sensor de movimiento PIR modelo HC-SR501*

Los sensores PIR funcionan midiendo la radiación infrarroja. Todos los cuerpos, ya sean vivos o inanimados, emiten cierta cantidad de energía infrarroja, que aumenta con la temperatura. Estos dispositivos cuentan con un sensor piroeléctrico que puede captar esta radiación y transformarla en una señal eléctrica.

En realidad, cada sensor PIR se divide en dos áreas y cuenta con un circuito eléctrico que equilibra ambas mediciones. Si ambas áreas reciben la misma cantidad de radiación infrarroja, la señal eléctrica resultante es nula. Sin embargo, si las dos áreas registran una medición diferente, se genera una señal eléctrica

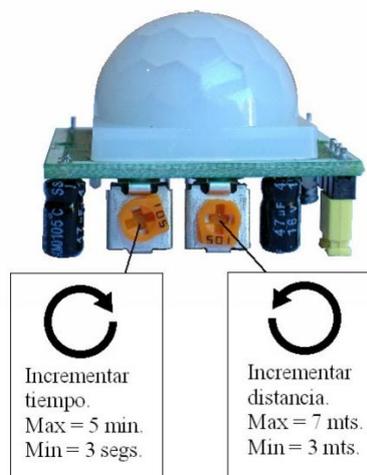


*Ilustración 52. Ejemplo de funcionamiento (fuente de <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>)*

Así, si un objeto cruza uno de los campos, se genera una señal eléctrica diferencial que es detectada por el sensor y se emite una señal digital.

El modelo utilizado está compuesto por los siguientes pines:

- **VCC:** alimentación.
- **GND:** tierra.
- **OUT:** salida digital que se ajusta con dos potenciómetros (ver Ilustración 53).



*Ilustración 53. Potenciómetros para ajustar tiempo entre las mediciones y la sensibilidad del sensor (fuente de <https://www.puntoflotante.net/MODULO-SENSOR-PASIVO-INFRARROJO-PIR-HC-SR501.htm>)*

Sus principales características son las siguientes:

- **Voltaje de funcionamiento:** 5-12V.
- **Ajuste de sensibilidad:** 3 a 7 metros.
- **Ajuste de retraso:** 3 segundos a 5 minutos.
- **Ángulo de medición:** 110°.
- **Temperatura de operación:** -15° a 70° C.
- **Tiempo de inicialización:** tras encender el módulo HC-SR05, es necesario esperar un minuto antes de que comience su funcionamiento normal. Durante este periodo inicial, es posible que el módulo active su salida dos o tres veces.

## Ejercicio 13

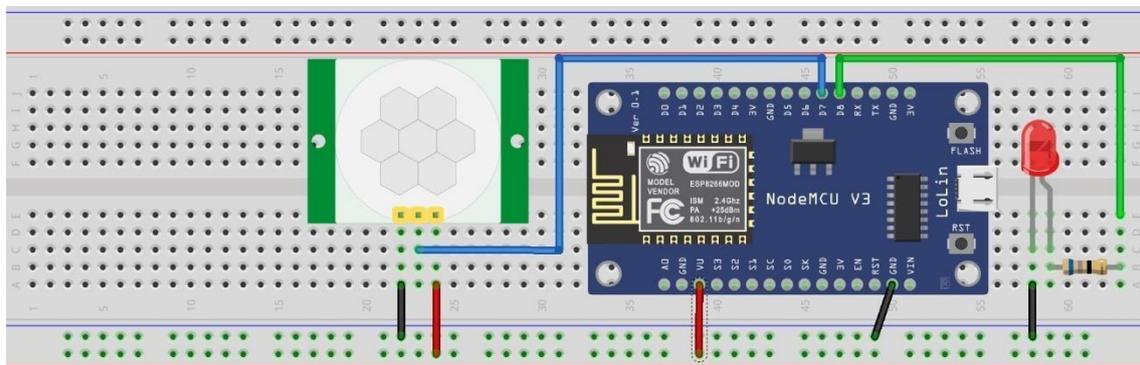
Conectar el sensor y regúlarlo para que detecte movimiento cada 3 segundos como máximo a 5 metros. Utiliza de igual forma que en los ejercicios anteriores un LED para indicar si ha detectado movimiento o no.

## Solución ejercicio 13

A continuación, en el siguiente enlace, se incluye el código asociado a este ejercicio:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_13/Ejercicio\\_13.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_13/Ejercicio_13.ino)

En este caso, es necesario alimentar el sensor con un voltaje de 5V. Por tanto, se debe utilizar un elevador de voltaje tal y como se explicó en la sección 3.3. En la **Ilustración 54** se incluye un esquemático del circuito de la solución. No se incluye el elevador de voltaje.



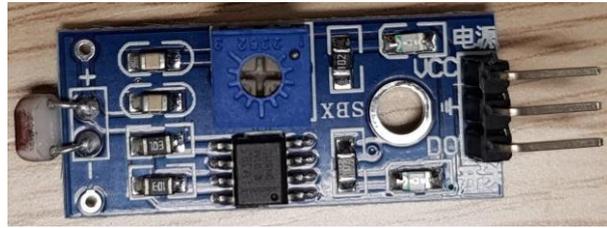
*Ilustración 54. Esquemático del ejercicio 13*

### 4.2.7. Sensor fotovoltaico

El sensor fotovoltaico es capaz de detectar la cantidad de luz que le llega variando su resistencia interna. A mayor resistencia menor luz hay y viceversa. Su comportamiento es el siguiente:

- Más luz = menor resistencia eléctrica.
- Menos luz = mayor resistencia eléctrica.

Este tipo de sensores (ver Ilustración 55) son muy utilizados en domótica, ya que nos permiten saber si hay suficiente luz y en caso contrario accionar bombillas.



*Ilustración 55. Sensor fotovoltaico*

En este caso los pines son los siguientes:

- **VCC:** alimentación.
- **GND:** tierra.
- **D0:** salida digital que se ajusta con el potenciómetro.

Este sensor cuenta con las siguientes características:

- **Voltaje de funcionamiento:** 3.3 - 5 V
- **Corriente de entrada:** 50 mA
- Potenciómetro ajustable.
- **Temperatura de funcionamiento:** -25°C - 80°C
- **Temperatura de almacenamiento:** -65°C - 150°C

## Ejercicio 14

*Conecta el sensor de luz y simula que enciendes una bombilla cuando no hay luz suficiente utilizando un LED. Por ejemplo, a partir de las 21:00h.*

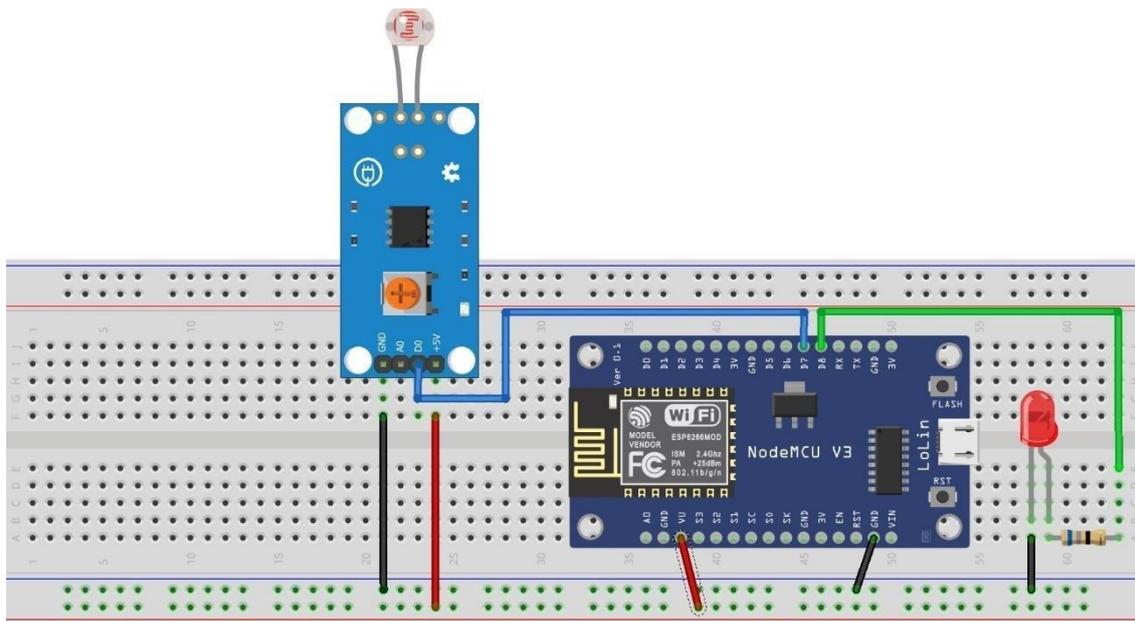
## Solución del ejercicio 14

El código es muy sencillo, de nuevo la complejidad está asociada a la regulación del potenciómetro. En este caso, en la hora indicada debemos realizar las comprobaciones necesarias para determinar el ajuste.

El código está incluido en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_14/Ejercicio\\_14.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_14/Ejercicio_14.ino)

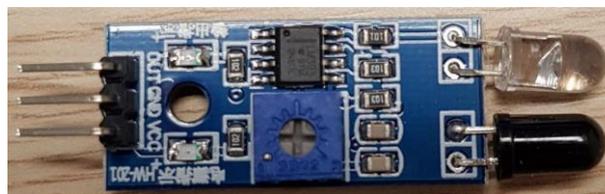
Además, el esquemático de la solución se encuentra en la Ilustración 56.



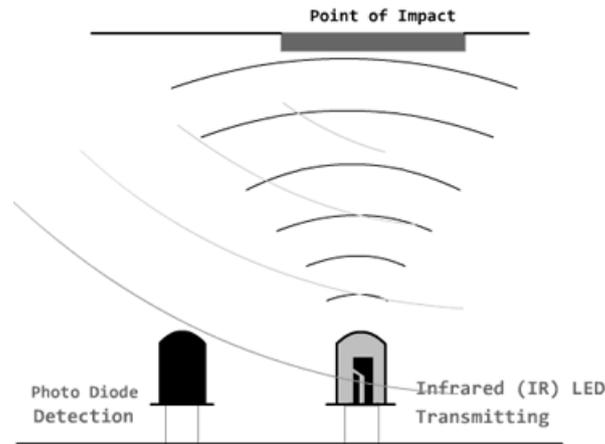
*Ilustración 56. 75. Esquemático del ejercicio 14*

#### 4.2.8. Sensor infrarrojo

Un detector de obstáculos infrarrojo es un aparato que identifica la existencia de un objeto a través de la reflexión de la luz. Se utiliza luz infrarroja (IR) para que no sea perceptible a los ojos humanos. Estos sensores son básicos en su constitución. Incluyen un LED que emite luz infrarroja y un fotodiodo (como el BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo (ver Ilustración 58). En este curso, usaremos el modelo HW-201 (ver Ilustración 57).



*Ilustración 57. Sensor infrarrojo modelo HW-201*



*Ilustración 58. Funcionamiento del sensor (fuente de <https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>)*

El modelo consta de los siguientes pines:

- **VCC:** alimentación.
- **GND:** tierra.
- **OUT:** salida digital que se ajusta con el potenciómetro.

Sus características principales son las siguientes:

- **Voltaje de funcionamiento:** 3.3 - 5V.
- **Ángulo de cobertura:** 35°.
- **Rango de detección:** 2 – 30 cm (ajustable con el potenciómetro).
- **Consumo:**
  - A 3.3V → 23 mA.
  - A 5V → 43 mA.

## Ejercicio 15

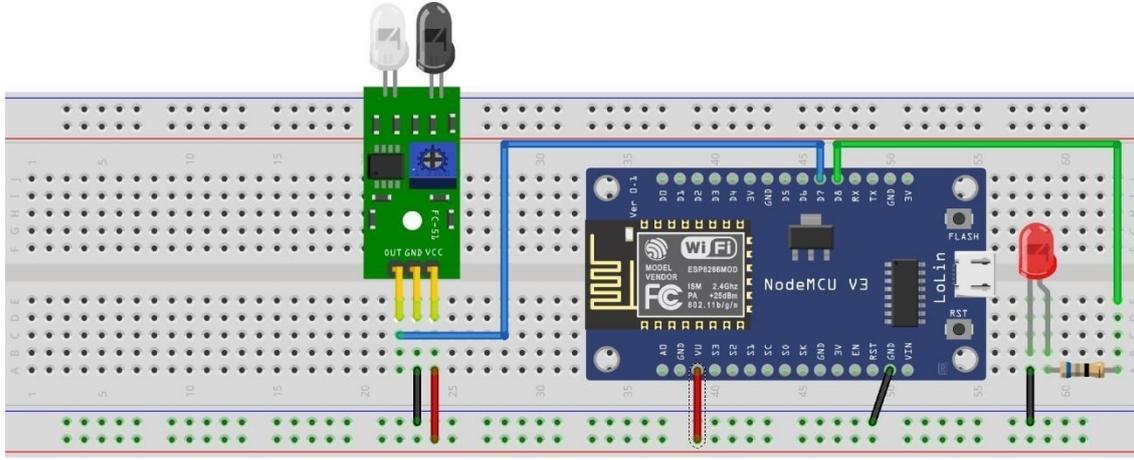
*Vamos a simular que el sensor se utiliza para detectar obstáculos. Normalmente la detección de un obstáculo desencadena una acción. Por ejemplo, en un vehículo autónomo pararía el motor de este. En este caso, un LED o cualquier otro actuador puede servirnos para este cometido (por ejemplo, un servomotor).*

## Solución del ejercicio 15

La solución a este ejercicio se encuentra a continuación:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_15/Ejercicio\\_15.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_15/Ejercicio_15.ino)

En nuestro caso, simplemente hemos utilizado un LED. Además, se incluye el esquemático del circuito en la Ilustración 59.



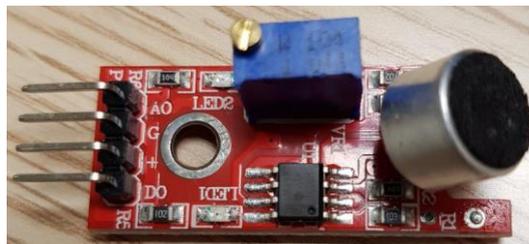
*Ilustración 59. Esquemático del ejercicio 15*

#### 4.2.9. Sensor de sonido

Los sensores de sonido para Arduino están equipados con un micrófono diminuto que detecta las fluctuaciones de ruido en el entorno. Entre sus ventajas más notables se encuentra su resistencia a la humedad y al calor, ya que incluso si la humedad y las partículas provocan un cortocircuito en parte del diafragma, siempre se produce una señal eléctrica en la salida. Sin embargo, el polvo es un inconveniente principal para los micrófonos, ya que puede degradar su rendimiento con el tiempo.

Estos sensores se pueden utilizar para realizar en diversos proyectos, por ejemplo, para medir la contaminación acústica.

En este curso el sensor de sonido es el modelo KY-038.



*Ilustración 60. Sensor de sonido modelo KY-038*

El modelo consta de los siguientes pines:

- **A0**: pin analógico que recibe un valor de ruido.
- **G**: tierra.
- **+**: alimentación.
- **D0**: salida digital que se ajusta con el potenciómetro.

Sus principales características son las siguientes:

- **Voltaje de funcionamiento**: 4 - 6V.
- **Micrófono**: Electret.
- **Gama de frecuencias**: 100 – 10.000 Hz.
- **Sensibilidad**:  $-46 \pm 2,0$ , (0dB = 1V / Pa) a 1KHz.
- **La sensibilidad mínima a ruido**: 58 dB.

## Ejercicio 16

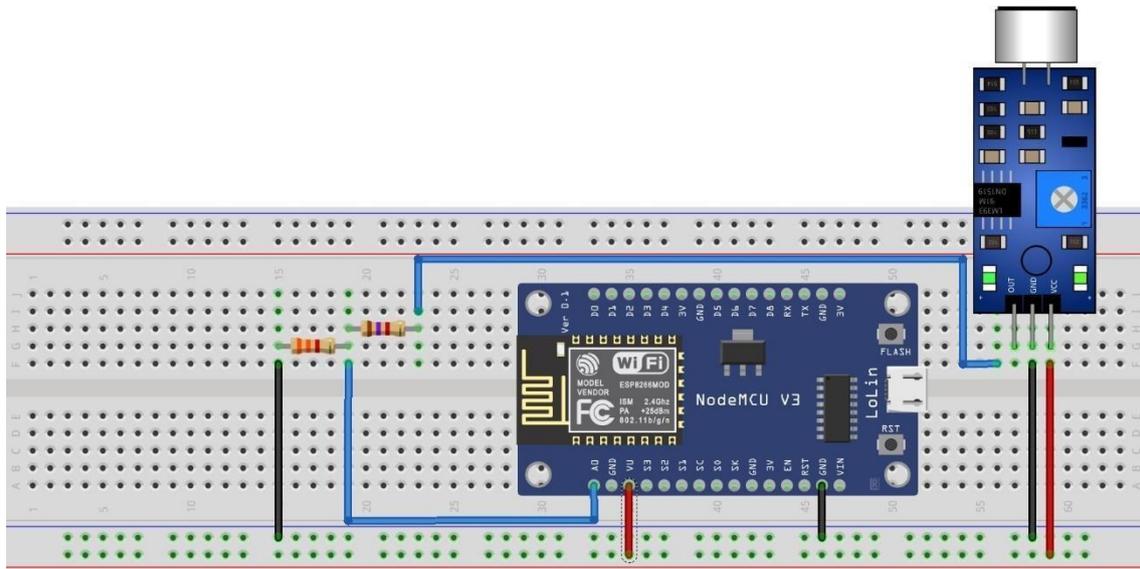
*Conecta el sensor y realiza una lectura analógica de los valores. Realiza diferentes pruebas de ruido para comprobar la sensibilidad del mismo. ¿Qué valores obtienes en una conversación entre dos personas a una distancia de 2 metros?*

## Solución del ejercicio 16

Para realizar el ejercicio es necesario alimentar el sensor con un voltaje de 5V. Por tanto, se debe utilizar un elevador de voltaje tal y como se explicó en la sección 3.3. Además, también se debe aplicar el divisor de tensión en la salida de 5V para adaptarla a un pin de 3.3V (ver la sección Sensor de gas).

A continuación, se incluye el código utilizado y el esquemático del circuito (ver Ilustración 61):

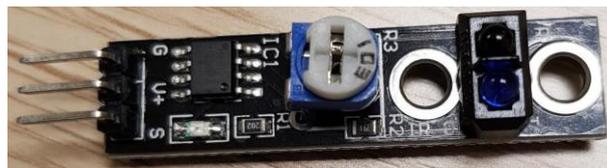
[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_16/Ejercicio\\_16.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_16/Ejercicio_16.ino)



*Ilustración 61. Esquemático del ejercicio 16*

#### 4.2.10. Sensor para detección de líneas

Estos sensores son bastante simples ya que cuentan con un LED que emite luz infrarroja y un fotodiodo (como el BPV10NF o similar) que recibe la luz reflejada por un posible obstáculo. Estos sensores suelen venir con una placa de medición estándar con el comparador LM393, que permite obtener la lectura como un valor digital cuando se supera un cierto umbral, que se ajusta a través de un potenciómetro en la placa. En este curso, usaremos el modelo KY-033 (ver Ilustración 62).



*Ilustración 62. Sensor para detección de líneas KY-033*

Este módulo, diseñado específicamente para Arduino, facilita la detección de líneas de manera sencilla, rápida y precisa, lo que es útil para aplicaciones como la detección de líneas. Por ejemplo, puede ser incorporado en vehículos autónomos para permitirles seguir un conjunto de líneas trazadas en el suelo.

El modelo consta de los siguientes pines:

- G: tierra.
- V+: alimentación.

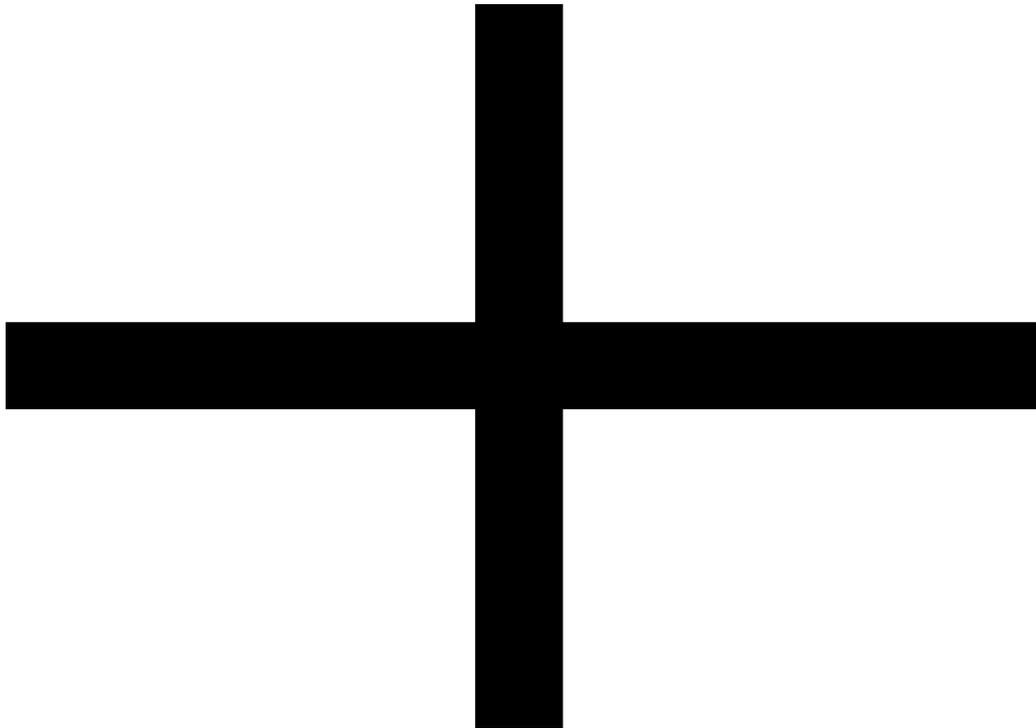
- S: salida digital que se ajusta con el potenciómetro.

Sus características principales son las siguientes:

- **Voltaje de funcionamiento:** 3.3-5V.
- **Corriente de trabajo:** 20 mA.
- **Distancia de detección:** 2 – 40 mm.
- **Comparador IC:** LM393.
- **Sensor IR:** TCRT5000L.
- **Temperatura de operación:** -10 a +50 °C.
- **Ángulo efectivo:** 35°.

## Ejercicio 17

*Conecta el sensor a la placa, imprime la imagen `LineasNegras.png` que se encuentra dentro de este ejercicio, y enciende un LED cada vez que el sensor detecte la línea.*

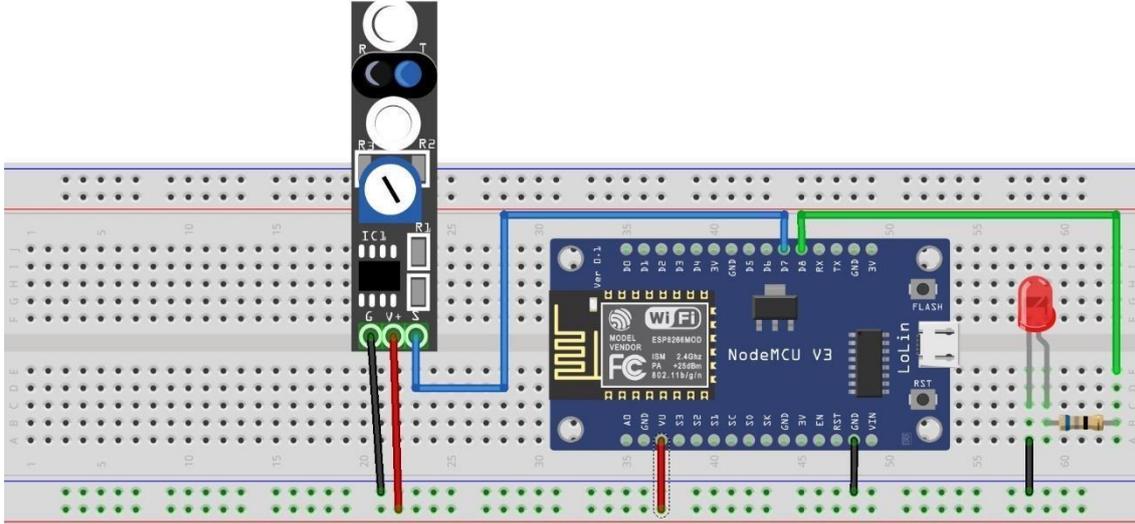


## Solución del ejercicio 17

A continuación, se enlace con la solución del código:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_17/Ejercicio\\_17.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_17/Ejercicio_17.ino)

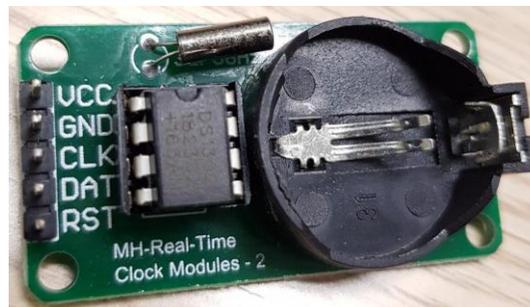
Y el esquemático del ejercicio se encuentra en la siguiente imagen (ver Ilustración 63).



*Ilustración 63. Esquemático del ejercicio 17*

#### 4.2.11. Reloj y calendario

Este dispositivo, que también puede ser clasificado como un tipo de sensor, utiliza el chip DS1302 (ver Ilustración 64), que incluye un reloj/calendario en tiempo real con 31 bytes de RAM estática. Este chip se comunica con un microprocesador a través de una interfaz serial simple. El DS1302 proporciona información sobre segundos, minutos, horas, día, fecha, mes y año. Ajusta automáticamente la fecha de fin de mes para los meses con menos de 31 días, incluyendo las correcciones para el año bisiesto. El reloj puede funcionar en formato de 24 o de 12 horas con un indicador AM/PM



*Ilustración 64. Reloj y calendario modelo RTC DS1302*

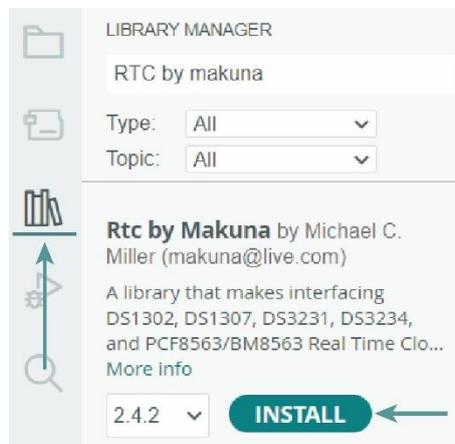
Consta de los siguientes pines:

- **VCC:** alimentación a 5V.
- **GND:** tierra.
- **CLK, DAT, RST:** pines para obtener información de hora y día.

Sus características principales son las siguientes:

- **Chip principal:** DS1302.
- **RAM para el almacenamiento de datos:** 31 x 8.
- **Tipo de interfaz:** serial interfaz I/O.
- **Voltaje de funcionamiento:** 2 - 5.5V
- **Temperatura de operación:** 0°C - 70°C
- Solo byte y transmisión multi-byte para el reloj o la memoria RAM de datos en lectura/escritura.
- **Batería requerida:** CR2032.

Para poder gestionar el reloj/calendario es necesaria la librería [Arduino Real Time Clock](#). Para ello debemos indicar en el gestor de librerías “*RTC by makuna*” (ver Ilustración 65).



*Ilustración 65. Instalación de la biblioteca RTC*

## Ejercicio 18

*Conecta el dispositivo reloj y calendario e imprime la fecha actual cada segundo con el siguiente formato: DD/MM/YYYY.*

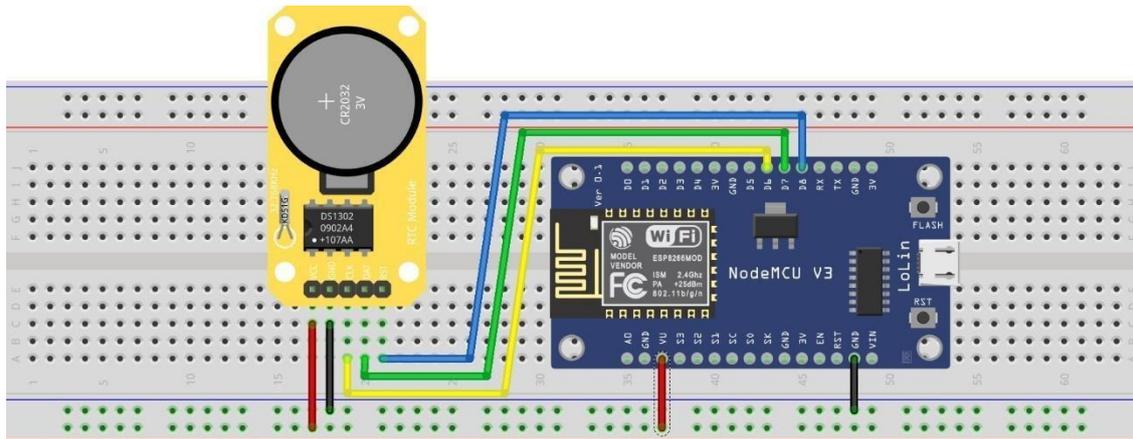
*Información adicional: consulta los ejemplos de la librería Arduino **Real Time Clock** [12] como punto de partida.*

## Solución del ejercicio 18

A continuación, se incluye el código con la solución del ejercicio:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_18/Ejercicio\\_18.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_18/Ejercicio_18.ino)

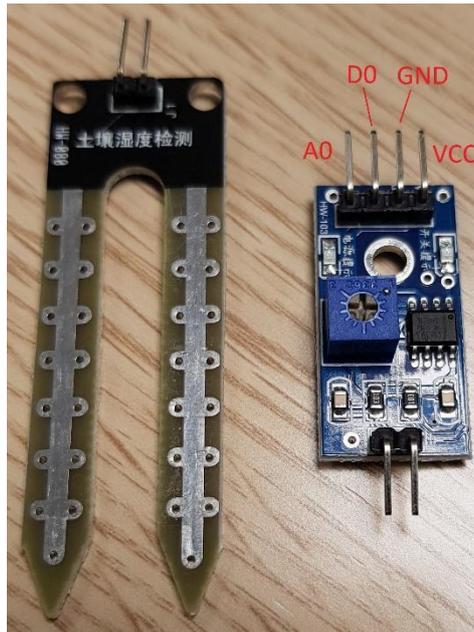
En la Ilustración 66 se muestra el esquemático del ejercicio.



*Ilustración 66. Esquemático del ejercicio 18*

### 4.2.12. Sensor de humedad en suelo o higrómetro

Un sensor de humedad del suelo, como el modelo HW-080 que usaremos en este curso, es un dispositivo que puede determinar si el suelo está húmedo o seco. Cuando el suelo está seco, el sensor emite un valor de 1023, y cuando está muy húmedo, emite un valor cercano a 0. Este sensor consta de una sonda con dos electrodos y un módulo convertidor que nos da una salida tanto analógica como digital, que puede ajustarse mediante un potenciómetro en la placa del módulo (ver Ilustración 67).



*Ilustración 67. Sensor de humedad en el suelo modelo HW-080*

Este tipo de sensores son muy utilizados para medir la humedad del suelo y activar el riego de forma automática cuando la humedad esté por debajo de un umbral.

Los pines de este modelo son los siguientes:

- **A0:** pin analógico que recibe un valor de humedad.
- **GND:** tierra.
- **VCC:** alimentación.
- **D0:** salida digital que se ajusta con el potenciómetro.

Sus principales características son:

- **Voltaje de funcionamiento:** 3.3-5V.
- **Salida:**
  - Disminuye la tensión cuanto mayor humedad detecte.
  - Aumenta la tensión cuanto menos humedad detecte.

## Ejercicio 19

*Conecta el sensor y extrae la humedad de forma analógica. Prueba a dejar caer una gota y a pasar un papel húmedo por el sensor. ¿En ambos casos se obtienen los mismos valores?*

## Solución del ejercicio 19

El proyecto consiste simplemente en visualizar los valores en crudo de la entrada analógica. Para poder probarlo humedece el sensor de forma directa con agua o pasa un papel húmedo por este. De las dos formas el sensor debería activarse y proporcionar valores altos

A continuación, se incluye el código para este ejercicio y el esquemático del mismo (ver Ilustración 68):

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_19/Ejercicio\\_19.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_19/Ejercicio_19.ino)

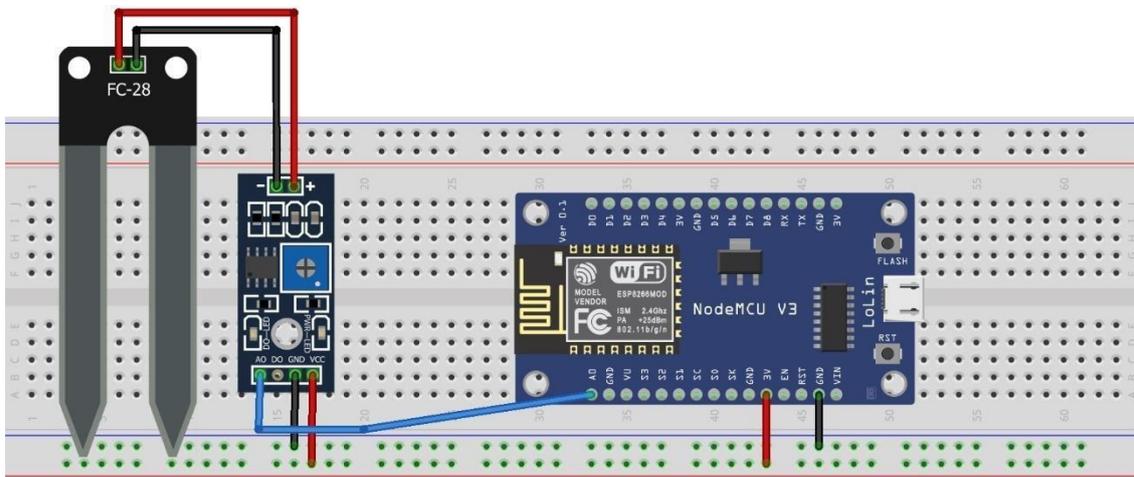
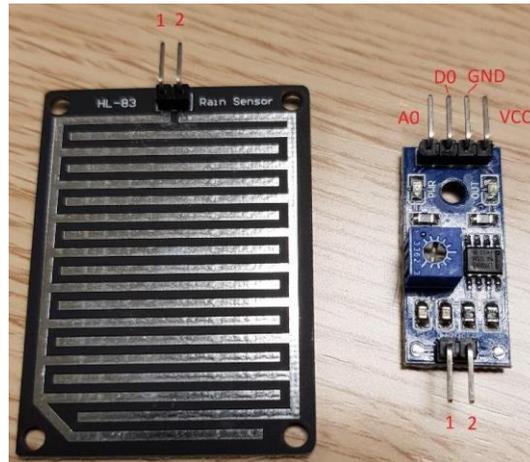


Ilustración 68. Esquemático del ejercicio 19

### 4.2.13. Sensor de lluvia

El sensor de lluvia es un dispositivo que nos permite identificar y reaccionar a la presencia de humedad, por ejemplo, activando una alarma. Estos dispositivos son ampliamente utilizados en domótica, la industria automotriz, sistemas de riego domésticos y otras áreas de nuestra vida diaria. En este curso, usaremos el modelo HL-83 (ver Ilustración 69).



*Ilustración 69. Sensor de lluvia modelo HL-83*

El modelo tiene los siguientes pines:

- **A0:** pin analógico que recibe un valor de humedad.
- **GND:** tierra.
- **VCC:** alimentación.
- **D0:** salida digital que se ajusta con el potenciómetro.

Sus principales características son las siguientes:

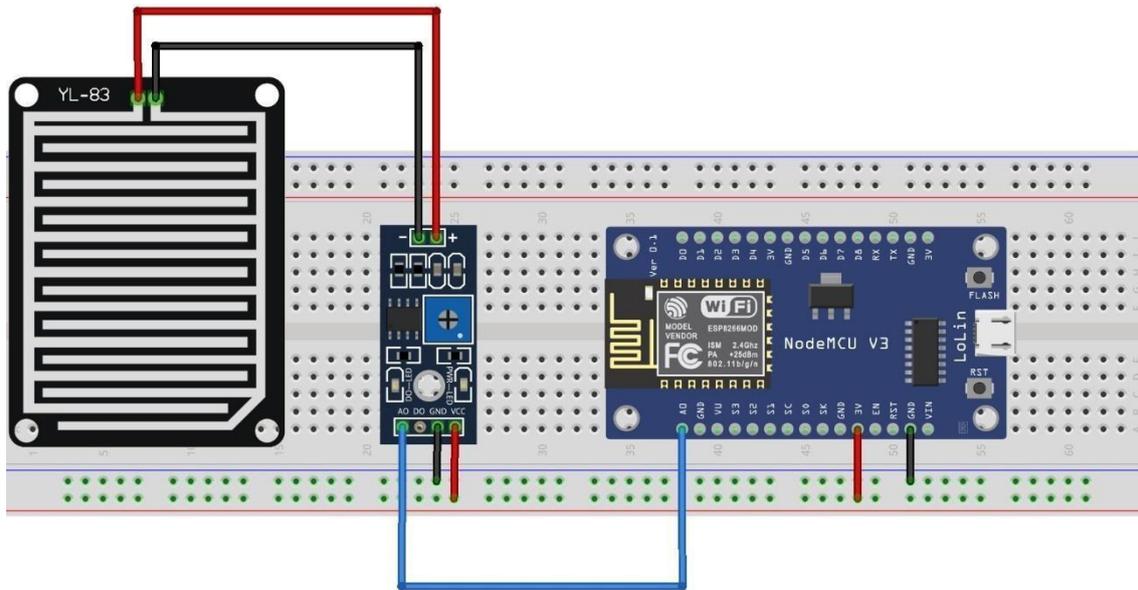
- **Voltaje de funcionamiento:** 3.3 - 5V.
- El valor de voltaje es similar al sensor anterior:
  - Valor de tensión alto si no detecta lluvia.
  - Valor de tensión bajo si detecta lluvia.
  - Los valores analógicos medidos varían desde 0 para una placa totalmente empapada, a 1023 para una placa totalmente seca.

## Ejercicio 20

*De igual forma que en el ejercicio anterior, conecta el sensor y extrae la humedad de forma analógica. Prueba a dejar caer una gota y a pasar un papel húmedo por el sensor. ¿En ambos casos se obtienen los mismos valores?*

## Solución del ejercicio 20

El código es idéntico al sensor anterior (ver sección Sensor de humedad en suelo o higrómetro). Por tanto, sólo se incluye el esquemático en la Ilustración 70.



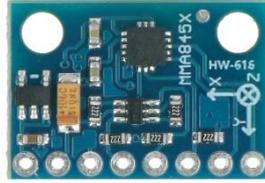
*Ilustración 70. Esquemático del ejercicio 20*

#### 4.2.14. Acelerómetro triaxial

Un acelerómetro triaxial es un dispositivo electromecánico que se utiliza para medir simultáneamente las vibraciones en tres direcciones o ejes perpendiculares entre sí. Estos ejes son generalmente identificados como X, Y y Z. Este tipo de acelerómetro proporciona más información de medida que un sensor uniaxial normal. Los acelerómetros triaxiales constan de tres acelerómetros individuales montados en una misma orientación, cada uno orientado a 90 grados respecto a los otros. Cuando se somete a una aceleración, el acelerómetro produce en sus terminales una salida eléctrica proporcional a esa aceleración,

Los acelerómetros triaxiales se utilizan en diversas aplicaciones, como la monitorización de la salud estructural de las construcciones. Existen varios tipos de acelerómetros triaxiales, cada uno basado en diferentes tecnologías como capacitivos, piezoeléctricos, térmicos, de efecto Hall, y ópticos. La elección de un acelerómetro triaxial debe basarse en las necesidades específicas de la aplicación.

En este caso, vamos a utilizar el modelo HW-616 (ver Ilustración 71).



*Ilustración 71. Acelerómetro triaxial modelo HW-616*

Los pines del modelo son los siguientes:

- **3.3V**: alimentación.
- **GND**: tierra.
- **SDA, SCL**: pines para la conexión I2C.

Sus principales características son las siguientes:

- **Voltaje de funcionamiento**: 3.3V.

Para poder gestionar y utilizar este sensor es necesario utilizar una biblioteca llamada Wire. Por defecto viene instalada.

## Ejercicio 21

*Obten la lectura de la aceleración en cada uno de sus ejes X, Y y Z, y muéstralo por pantalla.*

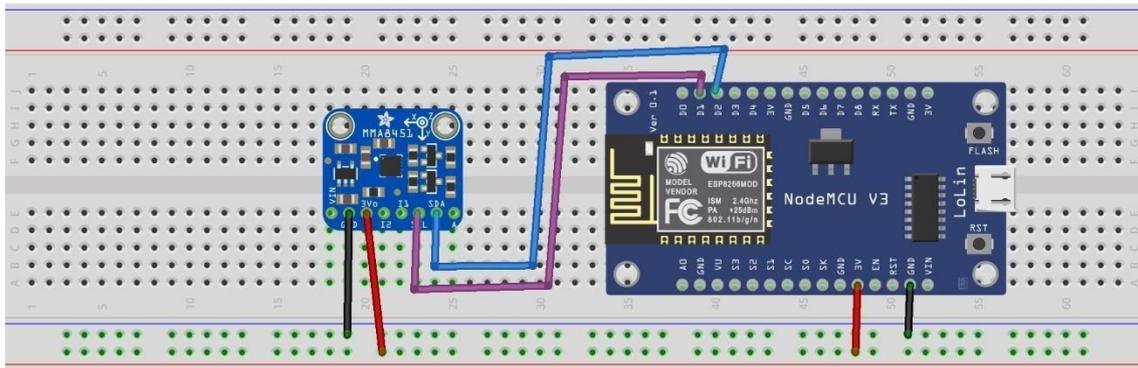
*Información adicional: para aprender a utilizar el acelerómetro consulta la documentación de la biblioteca [Wire](#) [13].*

## Solución del ejercicio 21

A continuación, se incluye el código del ejercicio resuelto:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_21/Ejercicio\\_21.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_21/Ejercicio_21.ino)

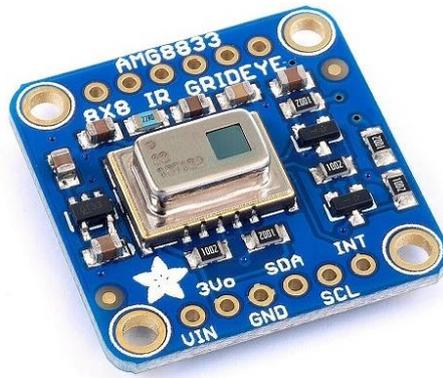
Además, se incluye un esquemático del circuito en la Ilustración 72. Por defecto, es necesario asociar el pin SDA del sensor con el pin D2 de la placa y el pin SCL con el pin D1.



*Ilustración 72. Esquemático del ejercicio 21*

#### 4.2.15. Cámara termal

La cámara térmica es un dispositivo que permite la detección de calor. En este caso, utilizaremos el modelo AMG8833. Esta cámara, fabricada por Adafruit y basada en el sensor de Panasonic AMG8833, es un array de sensores térmicos infrarrojos de 8x8 que permite medir temperaturas entre 0°C y 80°C. Es comúnmente utilizada en proyectos que requieren la detección de variaciones de temperatura, y puede ser integrada con plataformas como Arduino para su funcionamiento.

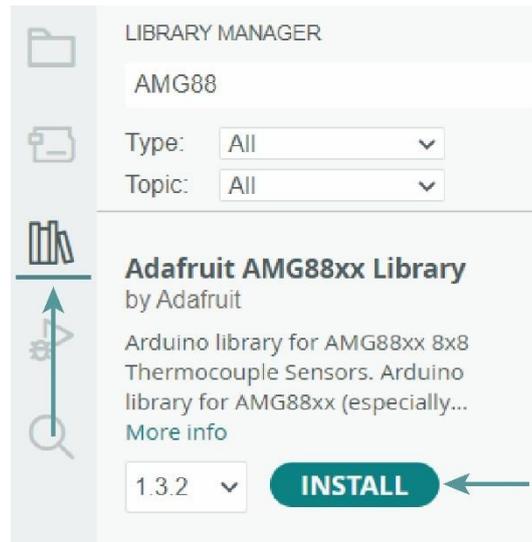


*Ilustración 73. Cámara termal modelo AMG8833*

Los pines a tener en cuenta son:

- **Vin**: alimentación por 5V.
- **3Vo**: alimentación por 3.3V
- **GND**: se conecta a tierra
- **SDA, SCL**: pines para la conexión I2C.

Para poder utilizar este sensor es necesario instalar la biblioteca [AMG88xx](#), tal y como viene en la Ilustración 74.



*Ilustración 74. Instalación de la biblioteca AMG88xx*

## Ejercicio 22

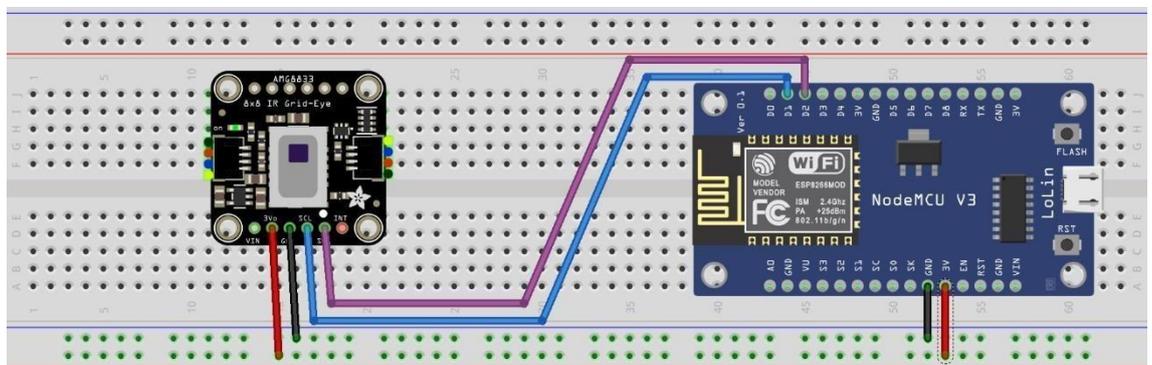
*Conecta el sensor termal y extrae la matriz correspondiente a los datos en crudo de temperatura. Imprímela por pantalla.*

*Información adicional: para resolver este ejercicio consulta la documentación de la biblioteca **AMG88xx** [14].*

## Solución del ejercicio 22

El esquemático de la solución de este ejercicio se muestra en la Ilustración 75, y el código con el ejercicio resuelto se encuentra en el siguiente enlace:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_22/Ejercicio\\_22.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_22/Ejercicio_22.ino)



*Ilustración 75. Esquema del ejercicio 22*

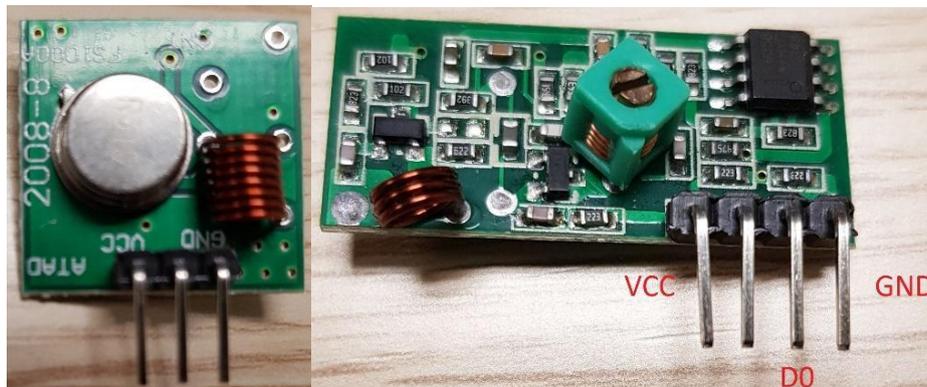
### 4.3. Híbridos

En esta última sección del curso, se incluyen dispositivos que no son actuadores o sensores como tales. En este caso, se han definido como dispositivos híbridos y son los siguientes:

- Comunicador unidireccional: emisor y receptor.
- Mando infrarrojo.

#### 4.3.1. Comunicador unidireccional: emisor y receptor

Los módulos de radiofrecuencia RF 433MHz son dispositivos inalámbricos que funcionan como transmisores y receptores, y que pueden ser utilizados para establecer comunicación entre diferentes procesadores. En este curso, utilizamos un módulo emisor modelo FS1000A y un receptor modelo XY-MK-5V (ver Ilustración 76). Este tipo de comunicación se ha hecho popular como medio de comunicación, principalmente, por su bajo coste.



*Ilustración 76. Comunicador unidireccional (emisor y receptor respectivamente)*

Los módulos de radiofrecuencia RF de 433MHz son dispositivos transmisores y receptores inalámbricos que se pueden utilizar para la comunicación entre procesadores. Aunque también hay módulos similares que operan a 315MHz, ambas frecuencias son de uso libre, lo que significa que no se incurre en costos por su utilización. El alcance de estos módulos depende del voltaje de alimentación y del tipo de antena utilizada; por ejemplo, a 5V y con la antena incluida en el módulo, el alcance raramente superará los 2 metros. Sin embargo, con una alimentación de 12V y una antena de cobre de 16.5cm, el alcance puede llegar hasta 300 metros en exteriores.

Los pines del módulo emisor son los siguientes:

- **VCC:** alimentación.
- **GND:** tierra.
- **OUT:** recepción de la señal.

Además, cuenta con las siguientes características:

- **Voltaje de funcionamiento:** 3.5 - 12V.
- **Alcance:** 20 - 200 metros (dependiendo del voltaje suministrado).
- **Tasa de transmisión:** 4 KB/s.
- **Potencia de transmisión:** 10 MW.
- **Frecuencia de transmisión:** 433 MHz.

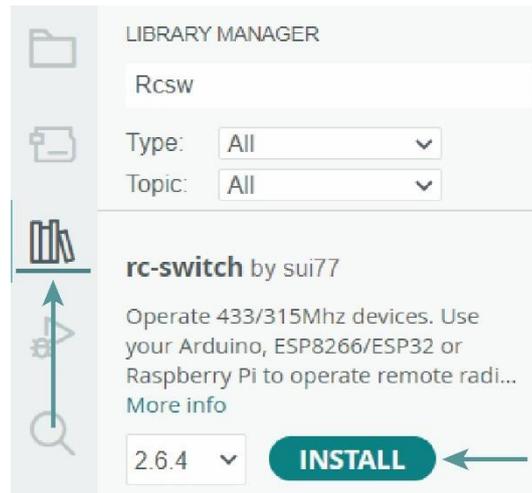
Los pines del módulo receptor son los siguientes:

- **VCC:** alimentación a 5V.
- **GND:** tierra.
- **OUT:** emisión de la señal.

Sus principales características son las siguientes:

- **Voltaje de funcionamiento:** 5V.
- **Frecuencia de recepción:** 433 MHz.

Para la transmisión entre los dos emisor y receptor, es necesario el uso de la librería [rc-switch](#) (ver **Ilustración 77**).



*Ilustración 77. Instalación de la librería rc-switch*

## Ejercicio 23

*Enciende un LED a distancia en el momento que se pulsa botón, utilizando el emisor y receptor para enviar la orden.*

*Información adicional: consulta la documentación de la biblioteca **rc-switch** [15].*

## Solución del ejercicio 23

Para este ejercicio es necesario dos placas de desarrollo para simular una comunicación a distancia. En este caso, es necesario alimentar ambos módulos con un voltaje de 5V. Por tanto, se debe utilizar un elevador de voltaje tal y como se explicó en la sección 2.3.

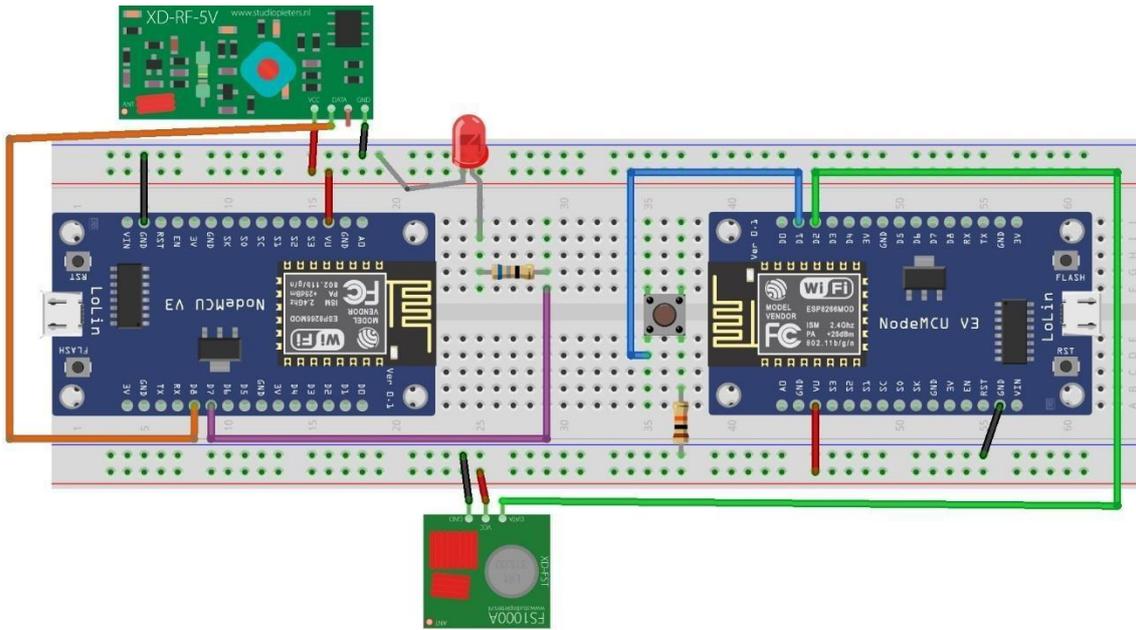
En primer lugar, se incluye el código de la placa que utiliza el emisor y hace uso de un botón para transmitir la orden:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_23\\_emisor/Ejercicio\\_23\\_emisor.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_23_emisor/Ejercicio_23_emisor.ino)

Por otro lado, se incluye el código del receptor:

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_23\\_receptor/Ejercicio\\_23\\_receptor.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_23_receptor/Ejercicio_23_receptor.ino)

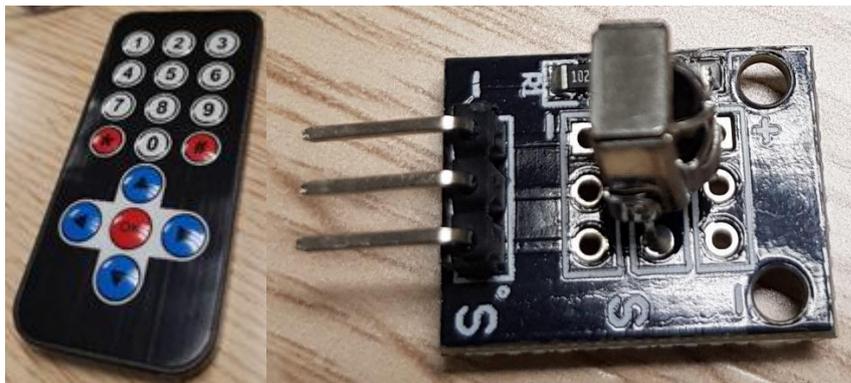
Por último, se incluye en la **Ilustración 78** el esquemático del ejercicio.



*Ilustración 78. Esquemático del ejercicio 23*

#### 4.3.2. Mando a distancia infrarrojo

Un mando a distancia infrarrojo (ver **Ilustración 79**) es un dispositivo de control que utiliza un LED infrarrojo para enviar una señal que puede ser detectada y procesada por un receptor. Este tipo de mando a distancia puede ser utilizado para controlar diversos proyectos de Arduino, como sistemas de luces, sonido, dispositivos por relé, robots o vehículos.



*Ilustración 79. Mando a distancia con receptor infrarrojo*

El control remoto infrarrojo opera mediante la emisión de luz en el espectro infrarrojo cercano, que es imperceptible para la vista humana, pero puede ser fácilmente detectada por un receptor infrarrojo. La distancia de operación de estos controles remotos es generalmente limitada, usualmente no supera los 3 metros, aunque puede llegar hasta 10 metros si no hay obstrucciones en la línea de vista. La distancia de

operación se ve fuertemente influenciada por el ángulo de emisión, disminuyendo rápidamente a medida que nos alejamos de la dirección frontal.

Para usar un control remoto infrarrojo, se requiere un módulo sensor infrarrojo que pueda recibir la señal emitida por el control. Este sensor tiene un filtro interno que solo detecta frecuencias infrarrojas cercanas a 38KHz, haciéndolo compatible con la mayoría de los controles remotos infrarrojos.

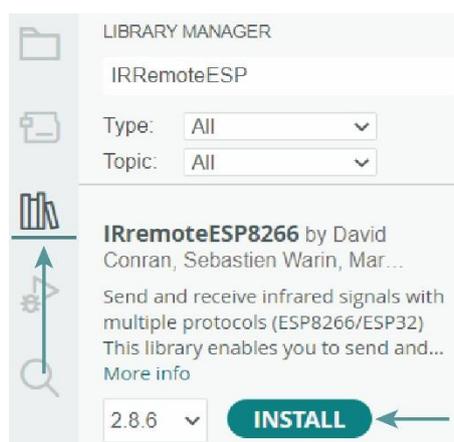
El receptor infrarrojo posee tres pines:

- -: tierra.
- S: pin por el que se recibe el código asociado a la tecla que ha pulsado.
- VCC: alimentación.

Sus características principales son las siguientes:

- **Voltaje de funcionamiento:** 2.7 - 5V.
- **Distancia de recepción:** 18 metros.
- **Ángulo de recepción:** - + 45°.
- **Frecuencia portadora:** 38 KHz.

Para poder integrar este dispositivo es necesario utilizar la librería de terceros llamada [IRremoteESP8266](#) (ver Ilustración 80).



*Ilustración 80. Instalación de la librería IRremoteESP8266*

## Ejercicio 24

*Integra el receptor IR en la placa y enciende un LED con un botón del mando y apágalo con otro diferente.*

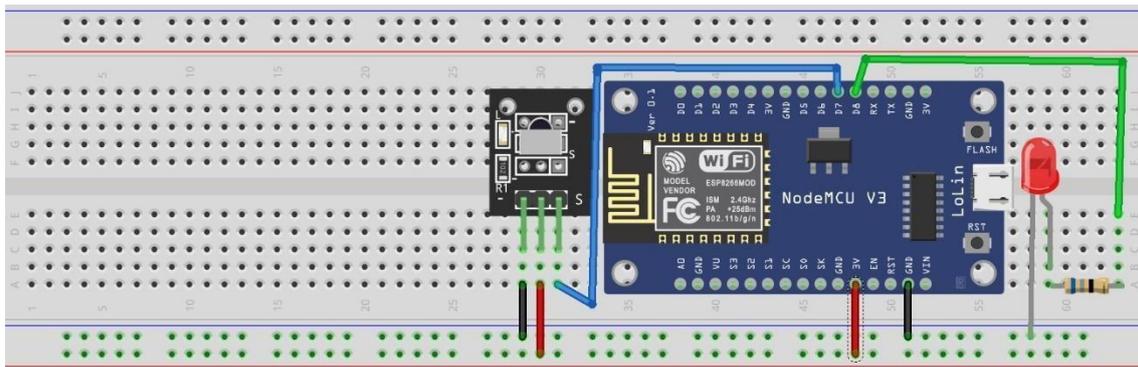
*Información adicional:* consulta la documentación de la librería de terceros *IRremoteESP8266* [16].

## Solución del ejercicio 24

A continuación, se incluye el código solución para este ejercicio.

[https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio\\_24/Ejercicio\\_24.ino](https://github.com/jossellr/CBP-NodeMCU-ASH/blob/main/Ejercicio_24/Ejercicio_24.ino)

Además, en la Ilustración 81 se incluye el esquemático del ejercicio.



*Ilustración 81. Esquemático del ejercicio 24*

## 5. Conclusiones

Este curso básico debe haber proporcionado al lector unas nociones básicas de la programación a bajo nivel de placas de desarrollo de la familia NodeMCU, así como aprendizaje en el uso de dispositivos de entrada y salida, sensores multimodales y dispositivos híbridos. El curso abarca la utilización de todos los dispositivos de bajo coste del mercado, permitiendo construir circuitos más complejos a partir de estos proyectos básicos.

A continuación, se incluyen una serie de propuestas más complejas para aplicar las competencias adquiridas:

1. Construye un vehículo en miniatura que sea capaz de moverse a través de líneas y, además, detecte obstáculos para detenerse y no colisionar.
2. Coloca un sensor en el cubo de la basura y contabiliza el número de veces que tiras la basura.
3. Coloca un sensor de temperatura en una habitación e indica si la temperatura es adecuada. Por ejemplo, puedes aplicar Lógica Difusa.

## Bibliografía

- [1] Arduino, «Documentación de la función pinMode,» [En línea]. Available:  
<https://www.arduino.cc/reference/es/language/functions/digital-io/pinmode/>.
- [2] Arduino, «Documentación de la función digitalWrite,» [En línea]. Available:  
<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>.
- [3] Arduino, «Documentación de la función delay,» [En línea]. Available:  
<https://www.arduino.cc/reference/en/language/functions/time/delay/>.
- [4] Arduino, «Documentación de la función digitalRead,» [En línea]. Available:  
<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>.
- [5] Arduino, «Documentación de la función analogWrite,» [En línea]. Available:  
<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>.
- [6] johnrickman, «Repositorio de la librería LiquidCrystal\_I2C,» [En línea]. Available: [https://github.com/johnrickman/LiquidCrystal\\_I2C](https://github.com/johnrickman/LiquidCrystal_I2C).
- [7] Arduino, «Documentación de la biblioteca Servo,» [En línea]. Available:  
<https://www.arduino.cc/reference/en/libraries/servo/>.

- [8] Arduino, «Documentación de la función tone,» [En línea]. Available: <https://www.arduino.cc/reference/en/language/functions/advanced-io/tone/>.
- [9] Arduino, «Documentación de la función noTone,» [En línea]. Available: <https://www.arduino.cc/reference/es/language/functions/advanced-io/notone/>.
- [10] Arduino, «Documentación de la biblioteca DHT sensor library,» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/dht-sensor-library/>.
- [11] miguel5612, «Repositorio de la librería MQSensorsLib,» [En línea]. Available: <https://github.com/miguel5612/MQSensorsLib>.
- [12] Makuna, «Repositorio de la biblioteca Arduino Real Time Clock,» [En línea]. Available: <https://github.com/Makuna/Rtc>.
- [13] Arduino, «Documentación de la biblioteca Wire,» [En línea]. Available: <https://www.arduino.cc/reference/en/language/functions/communication/wire/>.
- [14] Arduino, «Documentación de la biblioteca AMG88xx,» [En línea]. Available: <https://www.arduino.cc/reference/en/libraries/adafruit-amg88xx-library/>.
- [15] sui77, «Repositorio de la biblioteca rc-switch,» [En línea]. Available: <https://github.com/sui77/rc-switch>.

- [16] M. S. y. otros, «Repositorio de la lirbería IRremoteESP8266,» [En línea].  
Available: <https://github.com/crankyoldgit/IRremoteESP8266>.