Laboratorio Arduino IDE.

Sistemas teleinformáticas 3141769.

Oswaldo Perez.

Geraldine Sierra Parra,

Karol Suaza,

Yesid Martinez,

Andres Cardenas,

SENA 2025.

Contenido

[1. ¿Qué es un Arduino? 3](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970487)

[2. Partes de un Arduino: 4](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970488)

[3. Antes de empezar 8](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970489)

[: 8](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970490)

[4. Materiales Utilizados en el laboratorio: 12](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970491)

[Primera medición: 12](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970492)

[Ilustración 1Partes del Arduino 4](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970532)

[Ilustración 2Pines 5](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970533)

[Ilustración 3Pines A1-A5 6](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970534)

[Ilustración 4Alimentación 6](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970535)

[Ilustración 5Cristal Cuarzo 7](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970536)

[Ilustración 6Microcotrolador 8](file:///C:\Users\CASA\Desktop\Laboratorio%20Arduino.docx#_Toc192970537)

# ¿Qué es un Arduino?

Es una placa electrónica de hardware libre, con un microcontrolador (Es un circuito integrado que contiene un procesador, memoria y periféricos de entrada y salida) se puede reprogramar con una serie de pines. (arduino, 2023)

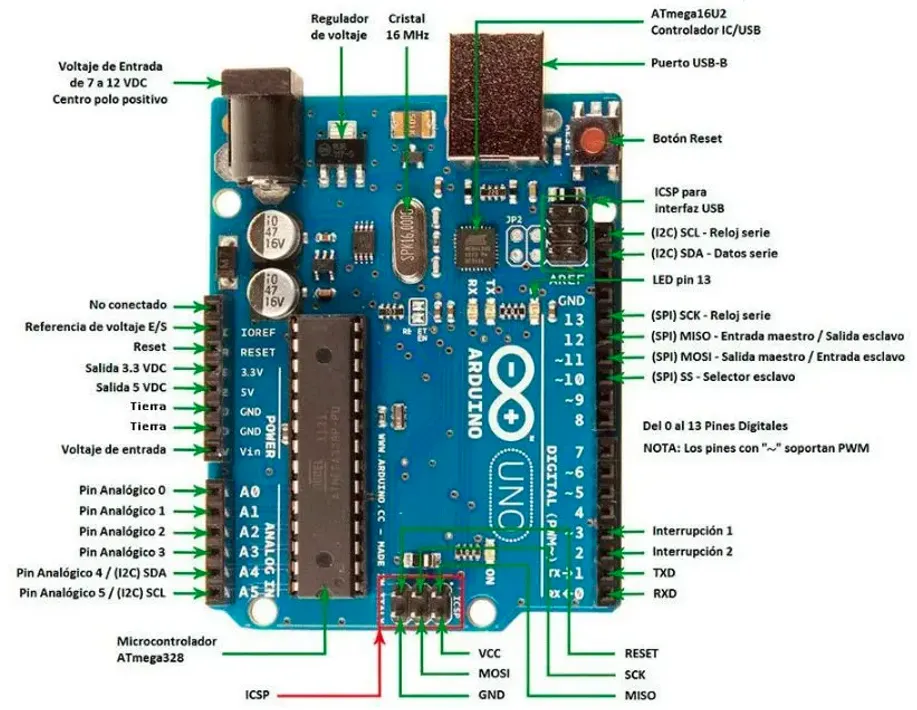


Ilustración 1Partes del Arduino

# Partes de un Arduino:

Para comenzar definiremos nuestro Arduino como una caja negra. Caja negra haciendo alusión al sistema de registro de los aviones para accidentes, que en realidad no son negras, sino que se les denomina así porque no se sabe lo que contiene, solo sabemos qué hace una función. Lo mismo sucede con nuestra tarjeta, será una caja con entradas, que son procesadas, y salidas.

Nos daremos cuenta que básicamente todo se comporta de esta manera, algo toma referencias (entrada), las analiza (proceso) y ejecuta una acción (salida). Un ejemplo real seria nuestro baño diario. Giramos la llave, tocamos el agua (entrada), la comparamos con una sensación adecuada para nosotros (procesamos la información) y giramos la llave para agua fría o caliente (salida). Haz el ejercicio y piensa cuantos sistemas que conoces realizan la misma lógica.

Básicamente, las señales de entrada pueden ser de dos tipos, digitales y analógicas. Las señales digitales solo tienen 2 estados, serán reconocidas como 1 o 0, todo o nada, alto o bajo (5V o 0V), como la luz de una habitación que esta encendida o apagada. Arduino Uno tiene 13 señales digitales que pueden funcionar como entrada o salida (definidas en el programa), o sea, leerá 5V o 0V, o entregará 5V o 0V. Otros modelos como el Arduino Mega tienen 54 señales digitales, que funcionan de la misma forma.

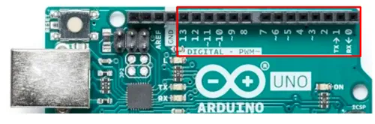


Ilustración 2Pines

Por otro lado, las entradas analógicas pueden tener infinitos valores entre amplitudes definidas con el tiempo, como las señales de audio. Infinito siempre es teórico, en la práctica las entradas analógicas medirán valores entre 0V y 5V y serán comprendidas por la tarjeta como valores entre 0 y 1023 (1024 valores ya que se considera el 0 como un valor más). Esto ocurre porque se debe digitalizar el valor analógico para que sea comprendido en el mismo idioma de nuestra tarjeta (lenguaje digital de unos y ceros, como en matrix) y para ello se ocupan 10 bit y como son 2 valores posibles en cada bit (1 y 0) tendremos combinaciones, 1024 valores a repartir entre 0V y 5V, de infinito nada. Quien digitaliza la señal analógica al lenguaje de unos y ceros es el Conversor analógico digital, CAD o DAC en inglés.

El Arduino Uno tiene 6 entradas analógicas, que soporta una tensión de hasta 5V y pueden ser escaladas a valores menores, pero no superiores. Otras placas pueden tener más o menos entradas, un CAD de mayor o menor cantidad de bit, pero se comportarán de la misma forma.

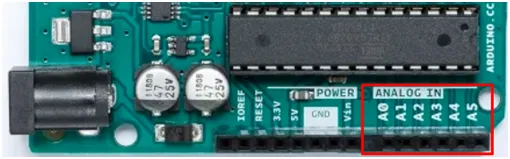


Ilustración 3Pines A1-A5

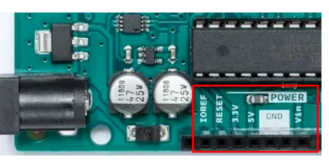


Ilustración 4Alimentación



La tarjeta tiene dos conexiones para comunicarse por ICSP, un tipo de comunicación serial. El que se encuentra cerca de las entradas analógicas, específicamente sirve para programar el Bootloader del microcontrolador que utiliza Arduino. El Bootloader es un gestor de arranque que interpreta los programas, permite enviar y recibir datos por los puertos, hace posible la comunicación USB. El ICSP que está cerca del conector usb de la tarjeta, es justamente el que nos permite programar el microcontrolador encargado de la comunicación usb.

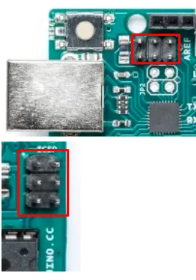




Ilustración 5Cristal Cuarzo



Ilustración 6Microcotrolador

Llegamos al cerebro de esta tarjeta, el microcontrolador ATmega 328. Como se mencionó al comienzo, el Arduino es un microcontrolador con las entradas y salidas listas para usar. El micro es el que ejecuta nuestros programas, lee nuestras entradas, envía señales para controlar dispositivos a la velocidad que el cristal le permite.



# Antes de empezar

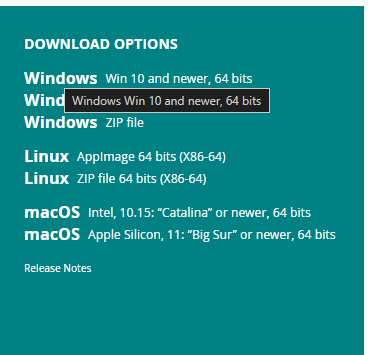
# :

Estos fueron algunos de los materiales utilizados para el laboratorio:

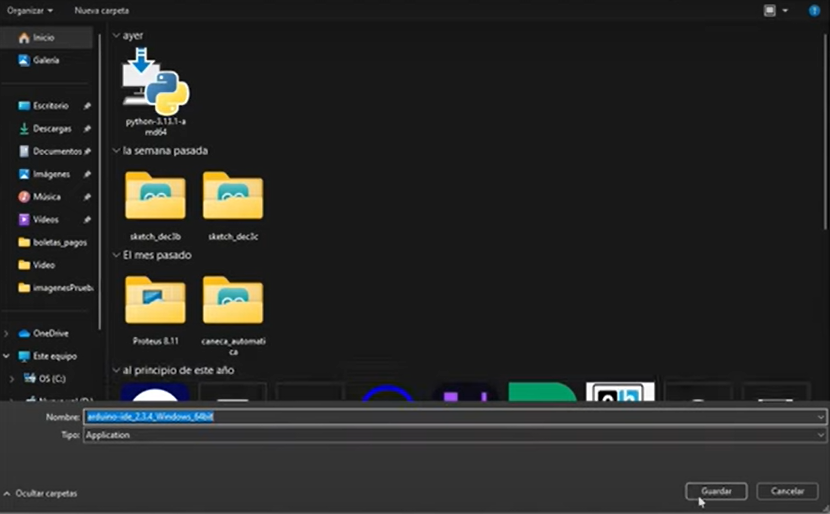
Primero vamos a descargar el programa Arduino IDE para generar el código programable para los proyectos que requieran de manejo de Arduino,

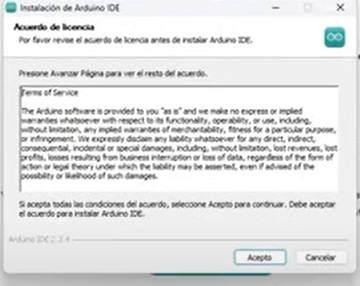
Se descarga por medio de la pagina web: <https://www.arduino.cc/en/software>

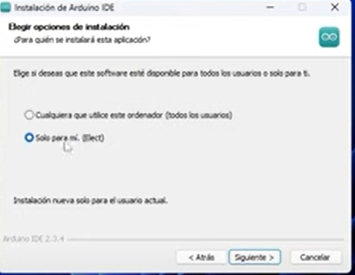
Versión Arduino IDE 2.3.4 para Windows



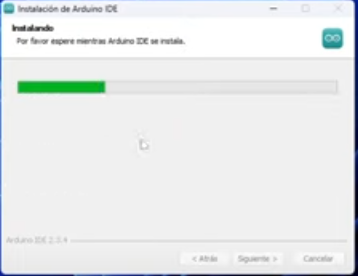
Damos Click en Just Download dos veces.





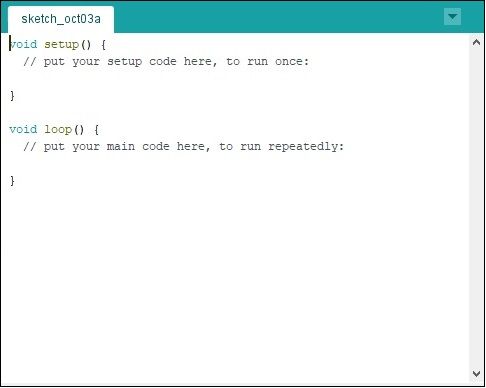


Solo para mi y le damos instalar esperamos a que cargue.





Terminar y el programa.



# Materiales Utilizados en el laboratorio:

* Placa de Arduino,
* Cable de USB,
* Pila de 9 Voltios,
* Protoboard,
* Multimetro,
* Resistencias de varios Ohmnios en especial de (10 Kilohmnios),
* Leds de diferentes colores,
* Cables,
* Pulsador
* Bipper
* Potenciometro

# Primera medición:

Quitamos el estuche de pasta del multímetro, destornillamos, conectamos la pila de 9 Voltios al multímetro, conectamos el cable positivo y negativo después tomamos el mutimetro y configuramos la perilla a diodos, tomamos un led

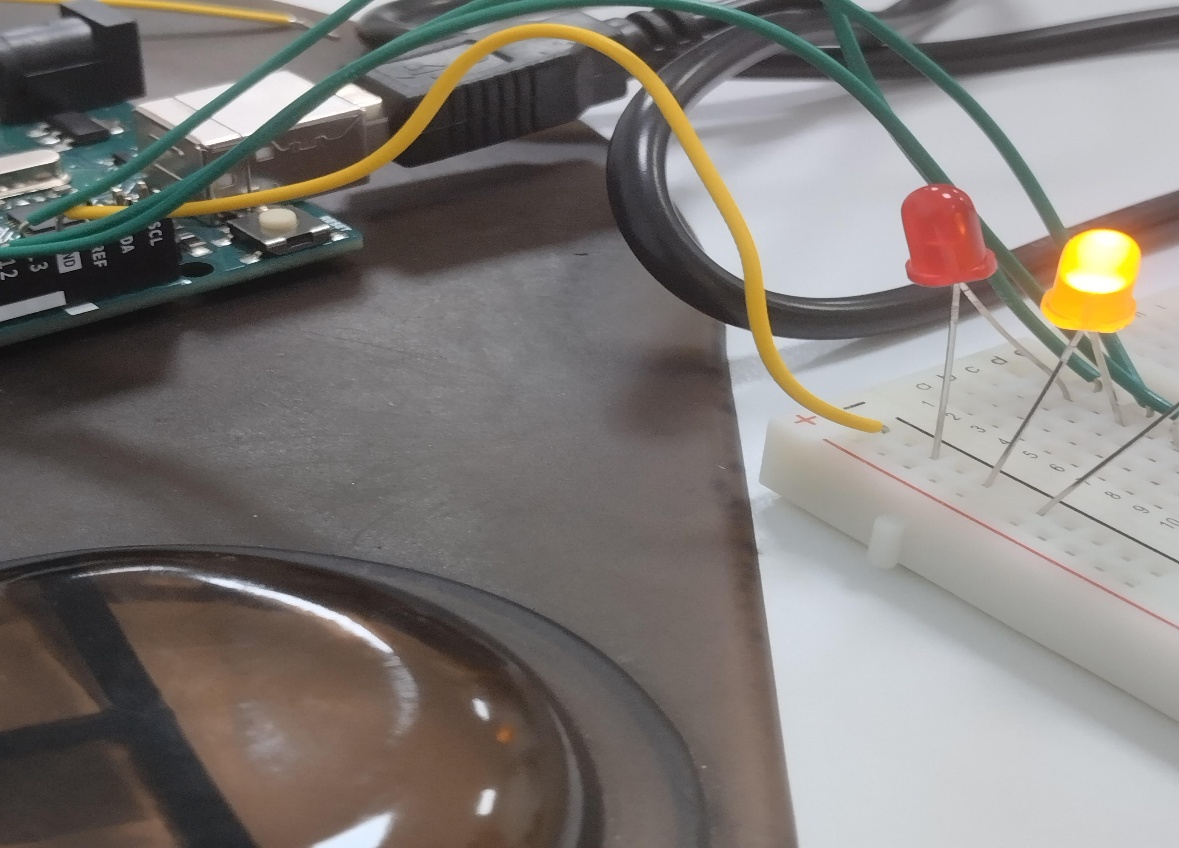


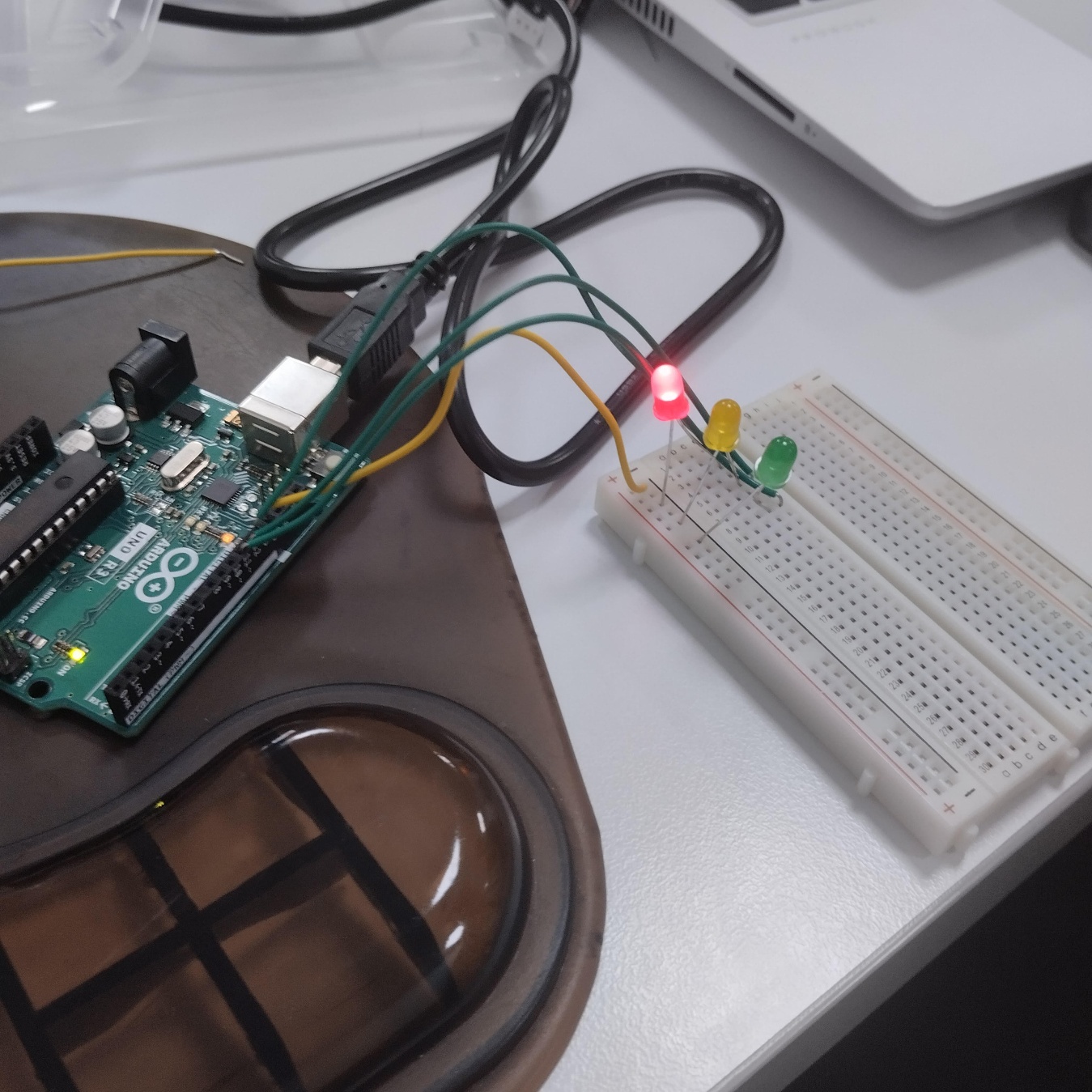


* **Proyecto Semáforo con luces Led:**

Lo primero es comprender el problema. Vamos a simular un cruce de calles con múltiples LEDS, que harán de semáforos, y dos pulsadores, que harán de sensores de presencia. El objetivo es que cuando se sitúe un coche en un semáforo que está en rojo, permita que se ponga en verde y cierre el otro semáforo.

Para empezar tenemos que partir de unas condiciones iniciales, ¿qué semáforo comienza abierto y qué semáforo comienza cerrado? Empezaremos con el semáforo 1 abierto y el semáforo 2 cerrado. Para entendernos, en el esquema del circuito eléctrico, de izquierda a derecha será semáforo 1, semáforo 2, pulsador 1 (simula sensor de presencia del semáforo 1) y pulsador 2 (simula sensor de presencia del semáforo 2).

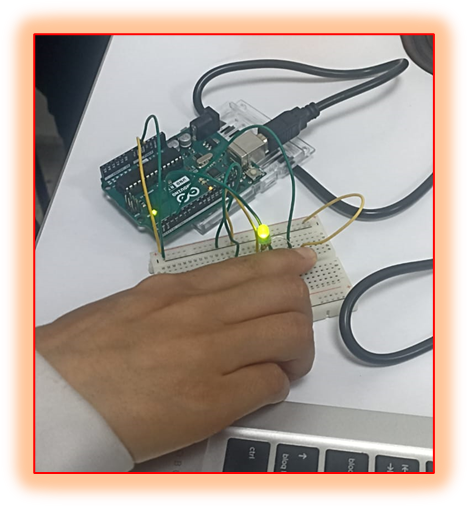


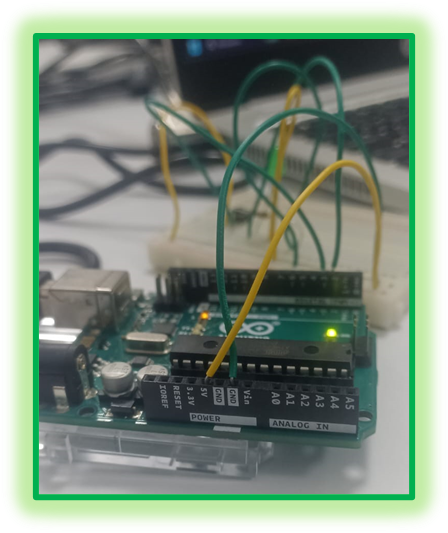


* **Proyecto enciende el Led cada vez que oprimamos el pulsador:**

En el siguiente video e imágenes profesor Osvaldo, podrá visualizar el proyecto de Arduino con un Led, un pulsador, una resistencia de 10 K Ohm, 6 cables de alimentación.

<https://youtube.com/shorts/WpPJVS8VkkM?feature=share>





* **Proyecto Buzzer y pulsador:**

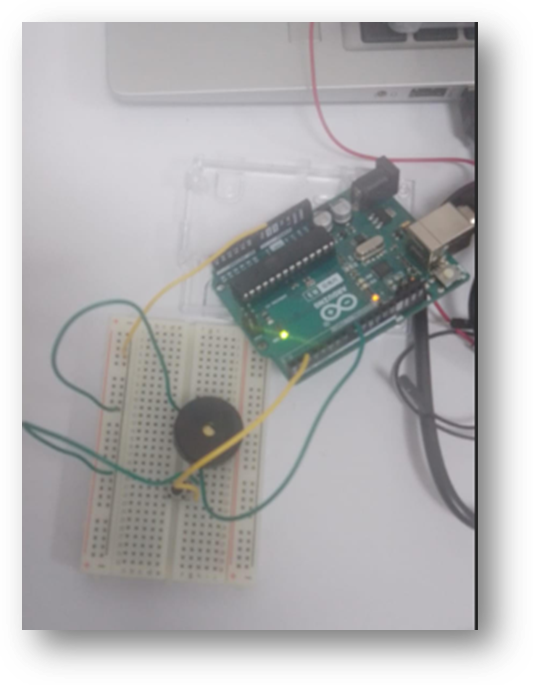
Un buzzer o zumbador es un dispositivo que es capaz de enviar avisos a través del sonido. Puede ser mecánico, electromecánico o piezoeléctrico.

Los zumbadores típicos son los piezoeléctricos vamos, los que vienen en cualquier kit de Arduino.

El zumbador piezoeléctrico se llama así porque en el centro del componente hay un elemento piezoeléctrico. Estos elementos sufren un fenómeno conocido como piezoelectricidad que ocurre en determinados cristales.

Consiste en que determinados materiales (cristales), al ser sometidos a tensiones mecánicas, producen una diferencia de potencial.

Esto también ocurre a la inversa, al someterse a un campo eléctrico, se deforman.



Bibliográficas:

<https://arduino.cl/?srsltid=AfmBOoquXjTPgpX9Jeu6U14CsPxIvmFi7TH42xOqLMZ6qh5EaBO7xvWH>