

# 1. Preparando el terreno

- [Conoce Arduino](#)
- [Hardware del Arduino](#)
- [Software del Arduino](#)
- [Instalando mBlock](#)
- [Dos formas de programar con mBlock](#)
- [Preparando mBlock y el Arduino](#)
- [Sensores](#)
- [Actuadores y otras salidas](#)
- [Kit de préstamo de CATEDU Arduino](#)
- [Resistencias](#)
- [Placa Protoboard](#)
- [Pulsadores](#)
- [LED](#)
- [Condensador](#)
- [LDR](#)
- [Sensor ultrasonidos](#)
- [Métodos para interactuar los objetos y el robot \(dispositivo\)](#)

# Conoce Arduino

## ¿Qué es Arduino?

Arduino es una tarjeta electrónica que integra básicamente a un microcontrolador y un conjunto de pines de conexión de entradas y salidas que permiten, mediante un determinado programa, interaccionar con el medio físico mediante sensores y actuadores electrónicos. De esta forma podrás crear tus propios proyectos tecnológicos, dotarlos de sensores que detecten magnitudes físicas como luz, calor, fuerza, etc... y en base a esa información, escribiendo un programa, activar otros dispositivos (actuadores) como pequeñas bombillas, ledes, servomotores, pequeños motores DC, relés, etc... Los sensores se conectan a los pines de entrada y los actuadores a los de salida.

**¿Sabías que.... ?** Uno de los co-creadores de Arduino es Español, de Zaragoza: **David Cuartielles** [+info](#)

## ¿Qué es un microcontrolador?

Es un circuito integrado que se puede programar, o sea que puede ejecutar las órdenes que tenga almacenadas en su memoria. Tiene las tres funciones principales de un computador: la unidad central de proceso, memoria y entradas y salidas.

Arduino utiliza la marca ATMEL, y el modelo de microcontrolador depende del tipo de tarjeta, por ejemplo la tarjeta Arduino Uno utiliza el micro ATMEL MEGA 328P. Si quieres saber las entrañas de esta placa [aquí](#)

## ¿Qué se puede hacer con Arduino? ¿Algún ejemplo?

Realmente el límite lo marca tu imaginación pero por dar alguna pista, podrías diseñar un sistema para la apertura y cierre de la puerta de un garaje, hacer un robot móvil que detecte objetos o que siga una línea negra, crear un detector de luz y oscuridad, implementar un termómetro, controlar un cilindro neumático, etc...

En este manual tienes múltiples ejemplos de pequeños proyectos para el aula, aunque Arduino es una herramienta que también se utiliza en el ámbito profesional para monitorización de sensores y automatización a pequeña escala por su flexibilidad, fiabilidad y precio.

## ¿Qué son las entradas y salidas?

Mediante los conectores de Arduino correspondientes a las entradas y salidas podemos comunicar nuestros programas con el “mundo exterior”. Si queremos leer el valor de la magnitud física medida por un sensor, por ejemplo una LDR que detecta el nivel de luminosidad, lo tendremos que hacer conectando el sensor a uno de los pines de entrada (en este caso analógicas) de la tarjeta.

De esta forma con una simple instrucción de lectura en el programa, podremos obtener el valor de la magnitud física. Si nuestra intención es actuar o “hacer algo” una vez leído el valor del sensor, por ejemplo encender un led si el sensor de luminosidad detecta oscuridad, tendremos que conectar el actuador (en este caso el led) a un pin de salida que proporcionará la corriente necesaria para activarlo.

En Arduino las entradas pueden ser analógicas o digitales y las salidas sólo digitales. Cada pin digital tiene doble función entrada o salida. En la zona de configuración del programa hay que indicar explícitamente mediante una instrucción cuál es función desempeña un determinado pin.

## ¿Dónde se conectan los sensores? ¿A las entradas analógicas o digitales?

La mayoría de sensores miden señales analógicas y proporcionan una variación de voltaje dentro de un rango (normalmente de 0 a +5V) dependiendo de lo que varíe la magnitud física medida. Muchos sensores son resistivos a algo (luz, temperatura, humedad,...), es decir que varían su resistencia eléctrica con la magnitud física, pero mediante un sencillo montaje de divisor de tensión conseguimos una variación de voltaje apta para Arduino. Estos montajes los veremos en las prácticas.

Veamos este ejemplo:

**El sensor LDR** es una resistencia que cambia según la intensidad de la luz. La estrategia es colocar el LDR en [un divisor de tensión](#) con otra resistencia de valor parecido al promedio del que da el LDR (por ej 10k), y el valor del punto de unión proporciona una tensión entre 0 y 5V. Como es una señal analógica, la conectamos a una entrada analógica (en la figura al A6)

[embedded-image-YfsLGvOe.jpegldrana.png](#)

Una vez realizadas las conexiones, si midieramos la salida del sensor con un voltímetro nos daría un valor decimal, por ejemplo un nivel de luz “intermedio” (rango de 0 a 5V) de un sensor de luz podría dar 3,3 voltios. Este tipo de información el microcontrolador **no la entiende tal cual**, sólo es capaz de interpretar números binarios (“0” ó “1”) por lo que para traducir los valores analógicos dispone internamente de un conversor analógico - digital que hará la conversión entre los dos sistemas, de forma que podremos tener valores discretos de la medida de los sensores analógicos. En el Arduino **las entradas analógicas** leen valores analógicos entre 0V y la alimentación (normalmente 5V) y los convierten en números entre **0 y 1023** (porque lo codifica en 10 dígitos binarios proporcionan  $2^{10} = 1024$  combinaciones).

Por ejemplo, si la entrada analógica lee un valor de 3,3V y la fuente de alimentación es 5V, la señal analógica que lee Arduino, haciendo una regla de 3, tiene un valor de  $3,3 * 1023 / 5 = 675,18 = 675$

### Mapeo

Para convertir estos valores 0 -1023 a valores más legibles, por ejemplo 0 - 100 para representarlo en % o 0-5 para que represente la medida en voltios ... veremos en programación la función **mapear**

**La mayoría de los sensores nos lo venden ya preparados** montados en una pequeña placa electrónica y con circuitos integrados auxiliares para no tener que estar haciendo divisores de tensión. Pueden tener salida analógica o **salida digital**, que en este caso lo tenemos que conectar a cualquier entrada digital D0 hasta D13.

Veamos el mismo ejemplo del LDR: Podemos comprar este módulo:

[ldrdig1.png](#)

Estos módulos proporcionan 3 pines: dos que son la alimentación, (0V, GND o - ) y (+5V V+o Vcc) y el pin que proporciona la lectura (Vout o D0 o I/O). En el caso de que sea un sensor que mida una magnitud analógica como en este caso la luz, suelen proporcionar un potenciómetro para determinar qué luminosidad se considera un 0 o un 1.

## ¿Hay sensores digitales que no estén en una placa electrónica?

Las entradas digitales sin una placa electrónica son cuando las señales a leer son valores discretos. Por ejemplo queremos poner un pulsador o un interruptor que encienda un led. Hacemos un montaje que cuando se pulse, entren 5 voltios en el pin digital de entrada y cuando no se pulse que “entren” 0 voltios. De esta manera la lectura del pin digital de entrada será “HIGH” con 5 voltios o “LOW” con 0 voltios.

Veremos más adelante que un interruptor no es un simple cable que conectamos a +5V o a 0V pues ¿Qué valor lee Arduino mientras levantamos el cable de un sitio a otro?, para ello veremos [configuraciones Pull-up o Pull-down](#) que se repiten en muchos sensores.

## ¿Qué son las salidas digitales etiquetadas con PWM (~)?

Son salidas digitales que **simulan una salida analógica**. Las siglas significan *Modulación por Ancho de Pulso (Pulse Width Modulation)* es decir, proporcionan una onda cuadrada con un nivel alto (+5V) de “cierta” duración.

[pinesPWM.png](#)

**Los valores PWM** que podemos proporcionar pueden ir desde **0 a 255**.

- Un valor bajo es una señal cuadrada donde el pulso es pequeño comparado con el periodo (ver figura primera señal) por lo tanto el Valor medio es bajo.
- Un valor mitad, es decir  $255/2 = 127$  o  $128$ , corresponde a una señal cuadrada perfecta (ver figura segunda señal) por lo que el Valor medio es la mitad, si 5V es la tensión de alimentación, sería 2,5V.
- Un valor alto es una señal cuadrada donde el pulso es grande comparado con el periodo (ver figura la tercera señal) por lo tanto el Valor medio es alto
- Un valor 0 corresponde a un 0V analógico y sería una señal sin pulso
- Un valor 255 corresponde a la máxima tensión (la de alimentación (normalmente 5V)).
- La frecuencia es de 490Hz para los pines 3, 9, 10, 11 y de 980Hz para los pines 5 y 6 en un Arduino UNO o Nano

[pwmsenal.png](#)

De esta manera podemos simular señales analógicas, esto es muy útil para activar servomotores y llevarlos a una posición determinada o variar la luminosidad de un led o en los motores de los robots que vayan más deprisa o más despacio

## ¿Puedo accionar motores DC con Arduino?

Si son motores muy, muy pequeños sí sería posible aunque no es recomendable, pueden dañar la placa. Los motores necesitan un consumo alto de corriente, sobre todo si tienen que mover cierta carga, por lo que se recomienda o bien utilizar una tarjeta Shield o extensión de Arduino que dispone de circuitería apta para proporcionar dicha corriente (transistores).

- En el [curso Arduino con código](#) utiliza una Shield llamada *Edubásica* que dispone de un [transistor](#) y un [circuito integrado LM293](#) para realizar esta función, además de otras ventajas para el aprendizaje de Arduino.
- En el curso [Rover con Arduino](#) utiliza Shield motor para NodeMCU

# Hardware del Arduino

## Placa Arduino

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores.

<https://sketchfab.com/models/286e0d188c184d51b7abccd89d774a28/embed>

[Arduino UNO R3](#) by [GmedranoTIC](#) on [Sketchfab](#)  
embedded-image-UBSH5GVW.png

## Entradas y salidas

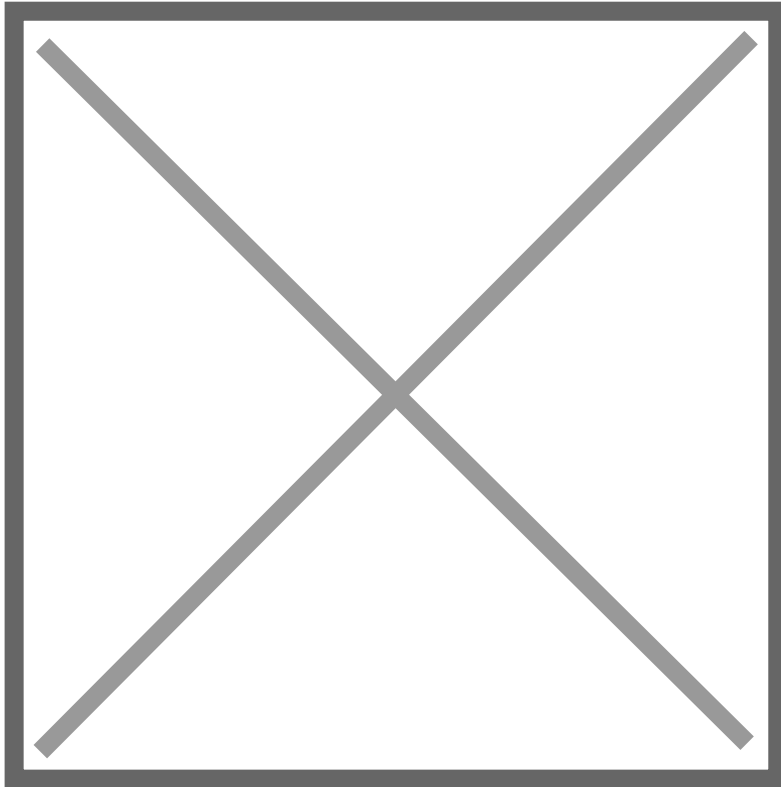
La placa Arduino UNO consta de:

- **DIGITALES:**
  - 14 entradas/salidas digitales D0-D13 previamente hay que configurarlas o entradas o salidas. Que operan a 5 ó 0 voltios. Cada pin puede proporcionar o recibir como máximo 40 mA.
    - Los pines D0 y D1 son compartidos por el puerto USB por lo tanto no se pueden utilizar si se está comunicando con el ordenador, están marcados como TX/RX (transmisión y recepción puerto serie).
- **ANALÓGICAS**
  - 6 entradas analógicas A0 al A5 con una resolución de 10 bits que proporcionan un número entero de 0 a 1023. Por defecto miden de 0 voltios (masa) hasta 5 voltios.
  - 6 salidas pseudo-analógicas, en los pines digitales 3, 5, 6, 8, 10 y 11 son PWM (Pulse Width Modulation) y como hemos visto, pueden proporcionar una salida cuadrada que su valor medio puede estar entre 0 y 5V

La intensidad de corriente que pueden proporcionar como salida son 20mA.

## Pines de la placa

Elementos con los que podemos interactuar: (tomando como ejemplo la placa USB). Empezando en el sentido de las agujas del reloj desde el centro de la parte superior:



- Pin de referencia analógica (naranja).
- Señal de tierra digital (verde claro).
- Pines digitales 2-13 (verde).
- Pines digitales 0-1 / entrada y salida del puerto serie: TX/RX (azul) (estándar de comunicación serie IC2).
- Botón de reset (negro).
- Entrada del circuito del programador serie (marrón).
- Pines de entrada analógica 0-5 (azul oscuro).
- Pines de alimentación y tierra (naranja y naranja claro).
- Entrada de la fuente de alimentación externa (9-12V DC)
- X1 (gris). - Entrada fuente de alimentación externa
- SV1: En las placas antiguas existen estos pines para conmutar si la alimentación es por el puerto USB, ahora lo realiza automáticamente un transistor MOSFET
- Puerto USB (rojo).

Las placas: Arduino Diecimila, Arduino Duemilanove o UNO y Arduino Mega están basados en los microcontroladores Atmega168, Atmega 328 y Atmega1280 respectivamente. Trabajan a 16MHz, una velocidad suficiente para proyectos de robótica. Su capacidad de memoria Flash van desde 16k el Atmega168 hasta 128k Atmega1280, suficientes para que quepan nuestros programas de robótica.

## SHIELDS para Arduino

Las llamadas Shields (escudos) para Arduino son tarjetas que se acoplan a la placa Arduino y añaden funciones y potencialidad a la placa Arduino. Según mi criterio hay dos tipos de Shields: Las educativas y las no educativas

## Shields educativas

Hay muchas Shields Educativas, vamos a ver unos pocos ejemplos

- La shield del [curso Arduino con código EDUBASICA](#), que integra algunos sensores y actuadores, pero lo que destaca es que además tiene la opción de conexión de motores o actuadores que requieran potencia, pues la placa Arduino por sí sola no puede proporcionar la suficiente intensidad para alimentar motores, relés o electroválvulas. El límite de intensidad que proporciona cada una de las salidas digitales es de 40 mA. Para poder activar estos dispositivos tendremos que montar un circuito externo adicional con transistores o circuitos integrados específicos para motores, como es el caso del LM293, que entregan la intensidad suficiente. Para facilitarnos la tarea esta placa integra un LM293 y un transistor de potencia, en total 5 salidas de potencia.

### [EDUBASICA-PEQ.png](#)

- La shield del [curso ECHIDNA](#) que tiene integrados sensores y actuadores como el acelerómetro, sensor de luz, pulsadores, potenciómetro, altavoz... y destaca el Joystick con la posibilidades de gamificación que se puede hacer con este elemento. También destaca el modo MAKEY que permite convertir los pines analógicos y los digitales D2 D3 en pines sensibles al tacto, con las mismas posibilidades que la placa MAKEY MAKEY ([ver curso Makey Makey](#)). Actualmente Echidna ya no fabrica esta Shield sino comercializa la ECHIDNA BLACK que no es una shield sino es autónoma (es decir, NO necesita la placa Arduino, la tiene integrada)  
. [2022-07-23 00\\_12\\_07-Editing Page 1.3 Hardware \\_ Librería CATEDU.png](#)

- La shield del curso [ARDUINO EN EL AULA, la TDR-STEAM](#) que tiene muchas posibilidades didácticas por su sensores (temperatura, humedad, luz, potenciómetro, pulsador, receptor IR).

### [tdrsteam.png](#)

- La [Shield k5864195](#) muy sencilla y barata, si lo que se busca son sensores sencillos (pulsadores, potenciómetro) y actuadores sencillos (altavoz, leds y display 7 segmentos).

### [VISUALINO.png](#)

- Shield **Protoboard** que simplemente tiene una placa para hacer los prototipos. Por ejemplo [Curso de Domótica con Arduino shieldproto.png](#)

## Shields no educativas

Es difícil elegir unos ejemplos de toda la variedad comercial que existe y tanta vitalidad de versiones ([ver lista](#)), preferimos que en caso de que necesites alguna función extra a tu Arduino, busques en ese momento cual es la mejor Shield que se adapta a tu proyecto. Es importante que sepas que existen Shields para todo, por ejemplo para...

- Agregar **conectividad** al Arduino, ya sea por red móvil (3G, 4G), por Wifi, Bluetooth o por cable Ethernet o incluso localización GPS.
- Agregar **potencia** a las salidas para conectar por ejemplo motores, como el de Edubásica pero más especializado en esta función y con más salidas pero por ejemplo [la L293D](#)  [7€ permite 4 motores, motor paso a paso, servo....](#)
- Shields con **relés** para conectar sensores y actuadores con tensión y así no tienes problemas de quemar el Arduino.
- Shields con **batería** que en el caso de corte de alimentación, permiten que el Arduino no se apague.
- Entrada y salida de **imagen** con pantalla líquida y cámara

Y encima se pueden apilar !! consiguiendo aumentar la capacidad de tu Arduino tanto como necesites (sin pasarte, pues la alimentación del Arduino es limitada al menos que la Shield tenga su propia alimentación).

<https://giphy.com/embed/O71r0jHck9jB6>

[via GIPHY](#)

# Alimentación eléctrica de Arduino

Uno de los aspectos claves para el buen funcionamiento de proyectos con Arduino, es que no incluyan elementos que consuman una intensidad superior a 200 mA como motores, relés, electroválvulas, etc.. Si los incluyen, entonces todo falla, debido a que no se ha alimentado correctamente la placa.

Tenemos dos posibilidades para alimentar Arduino:



#### **A .- Mediante el cable USB conectado al ordenador:**

- Cada pin proporciona 40 mA.
- El límite proporcionado por el USB es de 500 mA en total.

Si conectamos demasiada carga, la placa Arduino suele tener un comportamiento anómalo pudiéndose resetear el micro.

#### **B .- Utilizando una fuente de alimentación externa conectada al Jack de Arduino (fuente de voltaje, adaptador de corriente, batería o portapilas) :**

- El voltaje recomendado de la fuente externa está entre 7 y 12V.
- Tiene un diodo de protección para proteger la placa de cambio de polaridad accidental.
- La intensidad máxima que puede entregar Arduino a los actuadores que queramos controlar (servos, motores, relés,...) es de 1A, aunque una exposición prolongada a esta corriente puede estropear la placa. Lo recomendable son 500 mA.
- El pin serigrafiado con Vin proporciona directamente el voltaje de la fuente conectada al Jack de Arduino (menos la caída de tensión del diodo de protección) Desde ese pin podemos sacar un cable y alimentar a los actuadores que necesitemos. Por ejemplo, si alimentamos con una pila externa de 9V conectada al jack, en el pin Vin tendremos aproximadamente 9V (hay que restar la caída de tensión del diodo de protección que será de medio voltio). Además en los pines 5V y 3.3V dispondremos también de dichos voltajes aunque la fuente externa sea de 9V.

#### **C.- No recomendable: Conectando el positivo (+Vcc) de la fuente externa a Vin y el negativo a GND:**

Podemos alimentar Arduino externamente si necesidad de conector Jack a través de Vin y GND el problema es que nos saltamos un diodo de protección que evita que se quemé el circuito por un exceso de corriente.

#### **CONCLUSIÓN:**

- Si necesitamos hacer funcionar actuadores de bajo consumo (luces, zumbadores, etc...) podremos trabajar directamente con el USB conectado al ordenador o conectado a un

PowerBank.

- Si necesitamos mover cargas, excitar bobinas u otros elementos de mayor consumo lo recomendable es alimentar externamente Arduino desde el Jack con un rango de 9 a 12V.

Si vas a utilizar elementos que requieran más de 800mA **tienes que usar la alimentación externa del Jack** y alimentar esos elementos por el **pin Vin** del Arduino.

Si esos elementos van a ser alimentados de forma independiente. ES IMPORTANTE **UNIR LAS MASAS**, En caso contrario, romperás la placa Arduino.

En este ejemplo se ha conectado el motor a una tensión externa (la pila) y la placa Arduino al USB **fíjate como se han unido las masas**

Opcionalmente se ha optado por unir el + de la pila con Vin. Eso equivale a conectar la pila al Jack. Esto hace que si quitamos la alimentación USB, la placa Arduino sigue alimentada.

Es recomendable unir la fuente externa (pila) al jack **y no como se muestra**, pues así tienes un **diodo de protección** que te protege la placa Arduino en el caso de que cambies la polaridad de la pila de forma accidental. ¿Porqué en este circuito no se ha hecho así? Porque en Tinkercad no he encontrado un Jack.

<https://www.tinkercad.com/embed/5jwVKqxbivr?editbtn=1>

# Software del Arduino

Las placas electrónicas educativas (Echidna, Micro:bit, Arduino, ESP32 ...) se programan mediante **varias opciones** :

[https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm\\_tCQkeAgg/pubembed?start=false&loop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm_tCQkeAgg/pubembed?start=false&loop=false&delayms=3000)

## OPCIÓN LENGUAJE GRÁFICO POR BLOQUES

Recomendado para primaria. Tenemos muchas posibilidades de lenguajes gráficos. Destacamos dos:

- **ECHIDNASCRATCH**
  - **Específico para Echidna e integra la IA [CURSO DE ECHIDNA](#)**
- **MBLOCK** Basado en Scratch. Aunque es un programa especializado en el robot comercial mBot, (basado en Arduino), el mismo programa está adaptado para programar Arduino.
  - **[CURSO ARDUINO CON MBLOCK](#)** se utiliza Arduino y placa Protoboard
  - **[CURSO DE ECHIDNA](#)** se utiliza la Shield Echidnam y EchidnaBlack
  - **[CURSO DE MBOT](#)** se utiliza el robot mBot
- **STEAMAKERBLOCKS (antes ARDUINOBLOCKS)** se trabaja online, muy visual y muy amigable. Está adaptado tanto para trabajar tanto Arduino como muchas placas controladoras y en el aula. Podemos verlo en los siguientes cursos:
  - **[CURSO ROVER CON ARDUINO](#)** aunque no se utiliza un Arduino, sino un NodeMCU pero la programación es exactamente igual
  - **[CURSO DE ARDUINO CON ARDUINOBLOCKS](#)** donde se utiliza el Arduino con una placa protoboard
  - **[CURSO ARDUINOBLOCKS EN EL AULA](#)** donde se utiliza la Shield TDR-STEAM
  - **[CURSO ESP32 EN EL AULA](#)** donde también utiliza la Shiedl TDR Steam pero la placa no es un Arduino, sino ESP32, la programación es exactamente igual.



- **Microbloks** <https://microblocks.fun/> placas: Arduino, Microbit, ESP32, RaspberryPico, ...  
[ver minitutorial](#)

Otros softwares para programar con bloques

- **Snap4Arduino** <https://snap4arduino.rocks/run/> Online, libre... ver [compartiva vs mBlock](#)
- **S4A** <https://s4a.cat/>

### EN VIVO ¿Qué es eso?

Existe una posibilidad de utilizar la placa "en vivo" frente a "cargar" el programa en la placa. Es decir, interactuando con el ordenador. El programa está en el PC. En la placa hay un **firmware** que le dice que este a las órdenes del PC. De esta manera podemos por ejemplo:

- Enviar órdenes desde el ordenador a la placa.

Por ejemplo que al pulsar la tecla espacio que se encienda el led D13

- Enviar información desde la placa al ordenador

Por ejemplo que muestre por pantalla nos muestre la cantidad de luz, que registra el sensor LDR, etc...

Que nosotros sepamos, estos programas permiten la programación en vivo :

- **mBlock** placas: Arduino, Microbit, Raspberry Pi, ... robots de Makeblock: mBot, Cyberpi...

- **EchidnaScratch** [CURSO DE ECHIDNA](#)

- **Microblocks**

**VENTAJAS** LA PROGRAMACIÓN EN VIVO PERMITE MUCHO JUEGO Y POSIBILIDADES A LA HORA DE ELABORAR PROYECTOS

**INCONVENIENTES:** Necesitas el ordenador encendido y conectado al robot.

### TAMBIÉN ES EN VIVO PERO ....

Hay otros softwares que técnicamente trabajan en vivo, es decir, que el programa se ejecuta desde el ordenador, no se ejecuta en la placa, son :

- **Snap4Arduino** para placas Arduino
- **Picobriks** blocks para Picobrick board

**Pero no permiten trabajar utilizando los elementos del ordenador** (teclado, webcam, pantalla, sprite o objetos,,,) )

Es **importante** que entiendas que para trabajar en vivo, la placa tiene que tener cargado un **"firmware"** para que interactúe con el ordenador.



**P: ¿Qué es eso de "firmware"?**

R: No es más que un software que se graba en los chips de la placa.

**P ¿Y por qué se llama así, y no se llama software o programa y en paz?**

R: Digamos que como se graba en los chips, es un medio camino entre software y hardware, para diferenciarlo del software habitual.

**EN CARGA ¿Qué es eso?**

Simplemente el programa que estas haciendo **se carga** en la placa

VENTAJAS: Eres independiente del ordenador, tu robot funciona independiente

DESVENTAJAS Pierdes todas las posibilidades de utilizar los recursos de un ordenador, teclado, pantalla, webcam, altavoces...

Es **importante** que si **cargas** tu programa en la placa, **pierdes** lo que había antes

Es decir, si quieres volver a trabajar EN VIVO tienes que cargar el firmware correspondiente.

## OPCIÓN LENGUAJE POR CÓDIGO

Recomendable a partir de secundaria.

### LENGUAJE POR CÓDIGO ARDUINO IDE

Es un lenguaje basado en Wiring y permite la programación del Arduino en un entorno de desarrollo (basado en Processing). El programa se llama **ARDUINO IDE** y se puede descargar desde la página oficial de Arduino: <https://www.arduino.cc/en/software>.

Hay otra posibilidad que es utilizarlo online, con la ventaja de tener tus proyectos "en la nube" y no depender del equipo. OJO, TIENES QUE TENER INSTALADO EL SOFTWARE

**CREATE AGENT**

<https://create.arduino.cc/getting-started/plugin/welcome>

arduinoide.png

En los cursos de CATEDU se ha utilizado el lenguaje por código empezando desde cero en:

- **CURSO ARDUINO CON CÓDIGO** donde se trabaja con el Arduino y con diferentes sensores y actuadores, con o sin placa Shield Edubásica.



- **CURSO DE DOMOTICA CON ARDUINO** donde se realiza una maqueta de una casa controlada con domótica. También el curso ofrece la versión de hacer la misma maqueta utilizando lenguaje gráfico por bloques.

Recomendamos estas hojas resumen si vas a trabajar con código:

- En Español: [enlaceDrive](#), [enlaceGithub](#)
- En Inglés: [enlaceDrive](#), [enlaceGithub](#), [enlaceSpakrfun](#)

## LENGUAJE POR CÓDIGO PYTHON

Es un lenguaje más amigable que ArduinoIDE y tiene muchos campos de aplicación aparte de la robótica

- Arduino tiene su propia versión para trabajar con sus placas compatibles: **Arduino Lab for Micropython**
  - [Ver curso Arduino ALVIK](#)
- Un compilador universal **Thonny**
  - [Ver curso Pico-briks](#)

## VENTAJAS E INCONVENIENTES LENGUAJE GRÁFICO POR BLOQUES vs CÓDIGO

El lenguaje gráfico por bloques es un lenguaje **sencillo de utilizar**, nos evita tener en cuenta muchas **librerías** y **cálculos**.

Otra ventaja, es que el lenguaje por bloques es el único que permite programación "en vivo"

Por ejemplo, la instrucción leer valor distancia el sensor ultrasonidos, mediante programación por bloques es

[leer-us.jpg](#)

mientras que en código es

```
double distancia;

double fnc_ultrasonic_distance(int _t, int _e){
  unsigned long dur=0;
```

```

digitalWrite(_t, LOW);
delayMicroseconds(5);
digitalWrite(_t, HIGH);
delayMicroseconds(10);
digitalWrite(_t, LOW);
dur = pulseIn(_e, HIGH, 18000);
    // devuelve cuanto tarda el pulso alto en microseg; 18000 es el tiempo a esperar limite
if(dur==0)return 999.0;
return (dur/57);
    // la velocidad del sonido es 344m/s = 34400 cm/seg = 0,0344 cm/microseg
    // como v=e/t luego e = v*t y como cuenta la ida y la vuelta distancia = v*t/2
    // luego distancia = 0,0344/2 * dur = dur/57
}

void setup()
{
    pinMode(6, OUTPUT);
    pinMode(5, INPUT);
}

void loop()
{

    distancia = fnc_ultrasonic_distance(6,5);
}

```

Como se puede ver en **código**, tiene que calcular la distancia haciendo cálculos del tiempo de rebote del eco, mientras que el **gráfico** es sumamente sencillo y se centra en el objetivo del algoritmo a crear, no en lo accesorio. Esto hace que un lenguaje gráfico por bloques se puede aplicar **desde los 8 años**.

Por otra parte, el lenguaje **código tiene todo el potencial**, es decir, no todo está en los lenguajes gráficos. Si se quiere cosas más avanzadas, hay que recurrir al código.



Un lenguaje **gráfico** se convierte en lenguaje **código**, pero **al revés no se puede**, debido a que el código es más depurado y no tiene la información necesaria para volver a su origen en bloques, ya lo has visto con el anterior ejemplo, el código tiene más información.

**¿No te lo crees?** Haz la prueba, métete en <https://www.tinkercad.com/> crea un programa con bloques, dale a la pestaña de código y te aparecerá una advertencia que perderás el programa con bloques ! **no puedes volver atrás!**

Curiosamente, tiene una opción *bloques+código* que traduce cada bloque con un código, es decir, traduce cada bloque sin perder información, sólo de esa manera se puede pasar de bloques a código y viceversa.

2024-09-23 10\_51\_49-Circuit design PIN0-CORE-CREW - Tinkercad.png

Hay [herramientas para pasar de bloques a código](#) pero no al revés

### bloques-codigo.png

Imagen Federico Coca [Notas sobre ESP32 STEAMakers](#) CC-BY-SA

El lenguaje **código** se traduce en lenguaje **máquina** (ceros y unos) entendible para el microprocesador, **pero al revés no se puede**.

En [este vídeo](#), en mi opinión se olvida de mBlock, Snap4Arduino, S4A pero puedes ver un vistazo de los diferentes editores

## OPCIÓN SIMULACIÓN

Incluimos dentro del apartado de Software los diferentes programas que hay para simular placas electrónicas como Arduino, ESP32, Micro:bit etc...

### Tinkercad

Esta herramienta <https://www.tinkercad.com> aparece en el [Curso Arduino con código](#) en la práctica [Comunicación entre dos Arduinos](#), pero también es una plataforma que sirve para hacer los diseños de elementos 3D, ver curso [Impresión 3D con Tinkercad](#)

Tiene la ventaja que es aplicación **online**, muy **visual** y buscan un reflejo de la práctica **real**, además de estar la herramienta **adaptada al aula** (gestión de alumnos y proyectos). Como

desventajas podemos decir que no tiene mucha variedad de componentes electrónicos y la simulación es algo lenta.

<https://www.youtube.com/embed/pXEv0wxW9Jo?rel=0>

Además permite embeber, **pruébalo !!!**      Pulsa **Iniciar simulación** y luego pulsa el **botón A** de cualquiera de los dos micro:bits

<https://www.tinkercad.com/embed/8vBL2E8wWDx?editbtn=1>

## Wokwi

Si Tinkercad se queda corto, puedes probar esta plataforma <https://wokwi.com/> con muchas posibilidades. Es **online** y puede trabajar con **multitud de placas**: ArduinoUno, ESP32, Raspberry,,,,

Como única desventaja que encontramos, es que echamos de menos la realidad de Tinkercad, por ejemplo no puedes poner una placa protoboard para realizar las conexiones, pero a cambio se gana simplicidad de cableado.

[2022-12-05 10\\_14\\_42-Knob.ino - Wokwi Arduino and ESP32 Simulator.png](#)

## UnoArduSim

Es una **aplicación local**. [UnoArduSim](#) además es una aplicación portable fácil de instalar y con los elementos de leds, motores servos ya preparados, ideal para ejemplos sencillos y para examinar señales y no depender de Internet, pero no es tan versátil.

[https://www.youtube.com/embed/WLJ\\_l4uGjXg?rel=0](https://www.youtube.com/embed/WLJ_l4uGjXg?rel=0)

## SIMULIDE

En <https://simulide.com/> podemos encontrar un programa local de software libre genérico para electrónica, incluido Arduino. En esta captura se puede ver uno de los ejemplos que incorpora el programa:

[2025-02-07 16\\_09\\_42-DesktopWindowXamlSource.png](#)

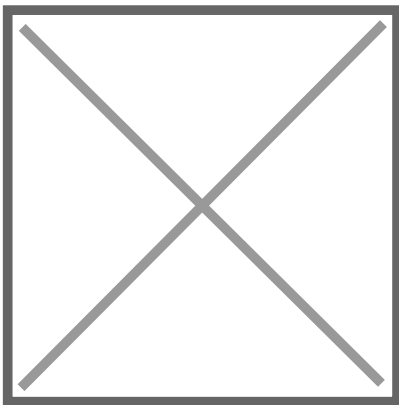


## OPCIÓN SÓLO DIBUJAR

- **TinkerCad** es un buen programa para dibujar los planos
  - permite también la simulación
  - permite embeber y compartir
  - no tiene muchos componentes
- **SimulIDE** es software libre. Es un programa portable.
  - Tiene muchos componentes
  - permite también la simulación
  - le faltan algunos sensores, pero van incorporando
- **Fritzing** es un clásico. Es un programa portable.
  - Tiene muchos componentes
  - no es gratis, hay que pagar 8€
- **Circuit canvas**
  - puede compartir [por ejemplo](#)
  - tiene buenos tutoriales sobre electrónica
  - todo en inglés

# Instalando mBlock

**mBlock** es un programa especializado en el manejo de los robots de Makeblock ([ver cursos de mBot en Aularagon](#)), pero permiten muchas arquitecturas de placas pues los mismos robots de Makeblock estan basados en placas de hardware libre Arduino, ESP32...



Hay dos versiones, **online y offline**, las dos son buenas, no obstante

- Si tienes Windows o Mac preferimos la **versión offline** para no saturar la red en los centros, rapidez y comodidad.

- Si utilizas **Linux** o **Chromebook** tienes que usar la **versión online** obligatoriamente pues no hay versión instalable.

Tanto una opción como la otra, permite tener los proyectos en la nube de Makeblock, compartir, embeberlos, etc...

## Versión online

---

Para la versión web **primero hay que instalar mLink driver** nos vamos a

<https://mblock.cc/pages/downloads> y abajo nos encontramos mLink para descargar, **descarga el mLink correspondiente a tu sistema operativo** :

[mlink-1.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Lo instalamos, y entonces pedirá instalar los **drivers** que son pequeños programas que comunican el PC con el hardware del robot



[mlink-drivers.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Si utilizas **Chromebook** lee la siguiente [guía](#) para conectar el robot

Una vez instalado **ya podemos entrar en la web** <https://ide.mblock.cc/> para programar en bloques nuestro robot

Nota: La página <https://ide.mblock.cc/> no suele estar bien situada en los buscadores si ponemos *mBlock*, la página que suele salir es esta <https://www.mblock.cc/en/> y nos da dos opciones, trabajar con bloque, o con Python, elegimos bloques

2024-05-16 23\_50\_10-mBlock - One-Stop Coding Platform for Teaching and Learning.png

Si has entrado sin querer en la URL <https://ide.makeblock.com/> no permite el logueo en Europa

En resumen, la URL correcta es <https://ide.mblock.cc/>

En este editor **tenemos que loguearnos** podemos crear un nuevo usuario, utilizando una cuenta de Google/Apple o registrarnos con un correo electrónico

2024-05-16 23\_53\_10-mBlock Block-Based IDE- Coding for Beginners.png

## Versión offline Windows o Mac

---

Vamos a <https://mblock.cc/pages/downloads> y descargamos la versión correcta a nuestro sistema operativo.

[download-version.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

En windows hacemos doble click en el archivo ejecutable descargado

[installing-download.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>



Lo instalamos, y entonces pedirá instalar los **drivers** que son pequeños programas que comunican el PC con el hardware del robot

[install-drivers.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

Algunas veces la ventana de driver de la figura anterior esta escondida en el fondo, hay que minimizar ventanas para acceder a ella.

En MAC el archivo será un fichero mount con la extensión .dmg, por lo tanto hay que arrastar el icono a la carpeta de aplicaciones

[install-mac.png](#)

Licencia CC-BY -SA <https://docs.arduino.cc/tutorials/alvik/getting-started-mblock/>

# Dos formas de programar con mBlock

## DOS FORMAS DE PROGRAMAR EN MBLOCK

Como vemos, mBlock permite dos formas de programar :

[https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm\\_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vQb1Dv9wN9QK-F6V7yvwDoyzquqwWlGvlyVjr83Yk56kAoYD7bXlnYDm_tCQkeAgg/pubembed?start=falseloop=false&delayms=3000)

## OPCIÓN Programación en vivo

El programa reside en el ordenador, y en la placa hay instalado un Firmware para ir escuchando y ejecutando lo que manda el ordenador.

- **VENTAJAS**

- Te permite interactuar el robot y el ordenador, por ejemplo podemos hacer que cuando el detector de humedad detecte agua, que salga por pantalla un fondo acuático, o que pulsando una tecla del teclado se encienda un LED en la placa...

- **DESVENTAJAS**

- **hay que cargar dentro del robot el Firmware exclusivo de mBlock** para que Arduino haga caso a mBlock
- Hay que tener **nuestro ordenador como intermediario, se come los recursos** y puede que nuestro programa vaya lento
- Por supuesto necesita tener ordenador conectado al robot, o sea, trabaja como un esclavo del ordenador.

## OPCIÓN Programación cargar a la placa

Todos los programas editores de Arduino (tanto los que programan con código como el Arduino IDE) como los editores de programas gráficos en bloque (mBlock, Snap4Arduino, Arduinoblocks, ...) permiten cargar el programa en la placa. Las ventajas y desventajas son las opuestas de trabajar en vivo.



## EN VIVO ¿Qué es eso?

---

Existe una posibilidad de utilizar la placa "en vivo" frente a "cargar" el programa en la placa. Es decir, interactuando con el ordenador. El programa está en el PC. En la placa hay un **firmware** que le dice que este a las órdenes del PC. De esta manera podemos por ejemplo:

- Enviar órdenes desde el ordenador a la placa.

Por ejemplo que al pulsar la tecla espacio que se encienda el led D13

- Enviar información desde la placa al ordenador

Por ejemplo que muestre por pantalla nos muestre la cantidad de luz, que registra el sensor LDR, etc...

Que nosotros sepamos, estos programas permiten la programación en vivo :

- **mBlock** placas: Arduino, Microbit, Raspberry Pi, ... robots de Makeblock: mBot, Cyberpi...

- **EchidnaScratch** CURSO DE ECHIDNA

- **Microblocks**

**VENTAJAS** LA PROGRAMACIÓN EN VIVO PERMITE MUCHO JUEGO Y POSIBILIDADES A LA HORA DE ELABORAR PROYECTOS

**INCONVENIENTES:** Necesitas el ordenador encendido y conectado al robot.

# Preparando mBlock y el Arduino

Hay que hacer dos cosas

- Cuando te descargas **mBlock** por defecto está preparado para utilizarlo con los robots (mBot), hay que decirle que no vamos a conectar un robot, que la placa será **Arduino UNO**
- Nuestro **ARDUINO** no pueden entender mBlock por lo que hay que grabarle un software dentro (un software que se graba en una placa hardware se llama **FIRMWARE**, no se borra cuando se apaga) y este firmware entiende lo que le manda mBlock, es decir, si conectamos otro Arduino y no le hemos grabado el firmware, no entenderá a mBlock.

Lo mejor un tutorial muy breve:

[https://docs.google.com/presentation/d/e/2PACX-1vRLqEoJCT355xMCeCXsd0Wc7JcJRk9JkwLCyzEPb\\_h1S2IbYKK7OdUf33yWRIXq216Zgh9Da7-gIOq1/embed?start=false&loop=false&delayms=3000](https://docs.google.com/presentation/d/e/2PACX-1vRLqEoJCT355xMCeCXsd0Wc7JcJRk9JkwLCyzEPb_h1S2IbYKK7OdUf33yWRIXq216Zgh9Da7-gIOq1/embed?start=false&loop=false&delayms=3000)

# Sensores

Esta sección es una visión rápida de las posibles **entradas** de los robots.

**NO LEAS TODOS SINO LOS QUE TIENE TU ROBOT**

## Un poco de teoría...

Cualquier sistema de control podríamos decir que funciona de una manera similar a un ser humano, salvando las distancias. Nosotros recibimos la información del mundo exterior gracias a nuestros sentidos (oído, olfato, gusto, vista y tacto), nuestro cerebro procesa esa información y a través de nuestros músculos o de nuestra voz realizamos diferentes acciones. Pues lo mismo sucede con los sistemas de control, reciben información del exterior gracias a los diferentes **SENSORES**, procesan esa información en sus PLACAS CONTROLADORAS (sus cerebros) tales como Arduino y dan una respuesta utilizando sus diferentes **ACTUADORES**.

[2022-04-13 19\\_23\\_12-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Un sensor es un objeto capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas. Los sensores o periféricos de entrada nos permiten obtener información del mundo real para utilizarla desde el programa de Arduino.

En la actualidad la cantidad de sensores disponibles es tan extensa como las variables que queramos medir, desde sensores de temperatura, humedad, luminosidad,... hasta acelerómetros, giroscopios, GPS,... pasando por detectores de gases, de pulsos cardíacos, sensores de efecto HALL,...

## Tipos de sensores

- **DIGITAL:** un sensor digital sólo tiene dos estados: activado/desactivado, ON/OFF, 1/0, Alto/Bajo, ... En este caso conectaremos el sensor a una de las entradas digitales de Arduino para leer el estado.

Ejemplo: un pulsador es un tipo de sensor sencillo que sólo nos da dos estados, "pulsado o no pulsado". Conectado a la placa Arduino debe generar 0v en reposo y 5v al pulsarlo. De esta forma desde el programa de Arduino podremos leer el estado del botón.



[2022-04-13 19\\_25\\_30-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **ANALÓGICO:** el sensor nos puede dar un rango de valores, normalmente se traduce en un valor de tensión o de corriente variable en función de la señal captada al sensor. En este caso conectaremos el sensor a una de las entradas analógicas de Arduino (A0,..., A5). El rango de entrada será una tensión entre 0v (GND) y 5v.

Ejemplo: Una fotorresistencia es un componente electrónico cuya resistencia disminuye con el aumento de intensidad de luz incidente. Su valor varía entre 0 y 5 v. la cantidad de valores que pueden leer las entradas analógicas de Arduinio son de 10 bits es decir 1024 valores. De tal modo que 0 = 0 v. y 1023 = 5V.

[2022-04-13 19\\_26\\_06-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

- **DATOS:** el sensor ofrece su información a través de una interfaz de comunicación. La forma de comunicación puede ser por sistemas estándar como **I2C** o SPI o algunos sensores usan su propio protocolo para codificar la información y debemos realizar desde el software la decodificación correcta para interpretar los datos del sensor (normalmente los desarrolladores de este tipo de sensores ofrecen una librería software para Arduino que hace todo el trabajo).

Ejemplo: el sensor DHT11. Por un solo pin envía los datos de temperatura y humedad.

[2022-04-13 19\\_27\\_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

## Sensores modulares.

En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

### Sensor pulsador

Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso



- [en el robot mClon](#)

2022-04-13 21\_58\_38-Modulo pulsador con tapa, 6uds.pngimage-1649922433118.png

Otra manera más "barata" de sustituir este módulo pulsador es poner un pulsador normal y una resistencia ( $\pm 10k$ ), al pulsar se produce una entrada en el Arduino, hay dos configuraciones, que al pulsar se emita un 0 lógico (configuración **Pull up**) o que al pulsar emita un 1 lógico (configuración **Pull down**) [¿Por qué hay que poner una resistencia?](#)

image-1650097141548.pngimage-1650097157889.png

Lo "normal" es que al pulsar se emita un '1' configuración **Pull down**, pero hay pulsadores que funcionan **Pull up** y los llaman **lógica invertida**, por eso en la programación por bloques podemos encontrar esto:

[logicainvertida.png](#)

## Sensor Táctil Capacitivo.

Este pequeño sensor puede "sentir" a las personas y el tacto y la retroalimentación de metales a un nivel de voltaje alto / bajo. Incluso aislado por alguna tela y papel, todavía puede sentir el tacto. Su sensibilidad disminuye a medida que la capa de aislamiento se hace más gruesa. En nuestra opinión lo preferimos frente al *Sensor pulsador* pues es muy económico, duradero y fiable.

Un ejemplo de uso en

- [Disparo láser en Arduino con ArduinoBlocks](#)
- [Apertura de puerta en Domotica con Arduino](#)

2022-04-13 21\_59\_29-Interruptor táctil TTP223B.png

## Potenciómetro y joystick

Un potenciómetro es una resistencia variable, es decir, cambia de valor mecánicamente, lo tenemos en multitud de dispositivos. El **joystick** es internamente dos potenciómetros con un pulsador integrado en un solo mando.

potenciometro-joystick.png2023-11-24 18\_45\_14-Quiero una - EchidnaShield.png

Este sensor es **analógico**, su salida puede ser cualquier valor entre Vcc y GND (si está en divisor de tensión como en la placa Edubásica no llega a esos valores extremos), por lo tanto hay que conectarlo a una entrada analógica de Arduino y como cualquier entrada analógica, proporcionará

valores entre 0 y 1023.

Ejemplos de uso:

- Arduino con código: [Mapeo del potenciómetro](#)
- Arduino con código: [Regular la luz con potenciómetro](#)
- [Arduinoblocks en el aula](#)
- En [Arduino con Echidna](#), con joystick
- [Domótica con Arduino con joystick](#)

## Sensor Focélula LDR.

El uso de fotocélulas es muy común en nuestras vidas, las encontramos en el encendido automático de farolas, apertura de puertas,... La fotocélula es un semiconductor. Es ampliamente utilizado en campos de interruptores de control automático como cámaras, luces solares de jardín, lámparas de césped, detectores de dinero, relojes de cuarzo, tazas de música, cajas de regalo, mini luces nocturnas, interruptores de control de luz y sonido, etc. Es un sensor analógico dando valores entre 0 y 5V y como entrada analógica de un Arduino se traduce en un rango de 0 a 1023 valores.

Un ejemplos de uso :

- [el interruptor crepuscular del curso Arduino con ArduinoBlocks](#)
- [Medir la luz en Rover con Arduino](#)
- [Medir la oscuridad en Arduino con mBlock](#)
- [Hinchar un balón en Arduino con mBlock](#)

Una manera más económica de montar este sensor es utilizar una resistencia y un LDR:

- El LDR cuando más oscuridad, más resistencia
- En una configuración **PULL DOWN**, cuanto **más** luz, la resistencia del LDR baja, por lo tanto **más** tensión en A6

[embedded-image-YfsLGvOe.jpeg](#)  
[2022-04-14 08\\_04\\_14-3.5 Cableado sensores \\_ Librería CATEDU.png](#)

Los módulos LDR que se venden suelen esta configuración Pull down, es decir, cuanto **más** luz, **más** tensión:

[image-1650370178627.png](#)

La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.

[2022-04-19 14\\_05\\_05-ArduinoBlocks.png](#)

Hay módulos LDR ya montados, que tienen componentes **activos** es decir, llevan circuitos electrónicos, transistores que amplifican etc... y dan la salida **digital** con un potenciómetro para definir el rango de luz que cambia de estado lógico. Puedes ver en la figura que tiene una salida digital **D0**.

[sensorluzarduinoigital.jpg](#)

O hay [algunos que tienen 4 pines](#) como en la figura que ofrecen las dos cosas: salida analógica **A0** y digital **D0**.

[sensorluzarduino.jpg](#)

Nosotros aconsejamos el divisor de tensión por tres razones: más barato, no implica gran circuitería y es visible su funcionamiento frente a estos encapsulados.

## Sensor de Ultrasonidos.

Es un sensor digital de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 350 cm. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno.

**No es un sensor preciso**, con una ligera inclinación de la superficie ya da lecturas erróneas pero es muy barato

El más común es el **HC-SR04** que tiene 4 pines de conexión: **VCC** **Trig** (Disparo del ultrasonido)

**Echo** (Recepción del ultrasonido) y **GND** aunque en algunos modelos como el de [ElecFreaks](#) tiene 3 pines. Integra Trig y Echo en uno sólo.

La distancia se calcula con esta fórmula:

**Distancia en cm = {(Tiempo en segundos entre Trig y el Echo) \* (V.Sonido 34000 en cm/s)} / 2**

Si programas en código, tienes que utilizar la fórmula anterior, previamente tienes que programar el cálculo del tiempo entre una emisión de un pulso en Trg y la respuesta en Echo.

Si utilizas la programación en bloques, no es necesario, seguro que hay un bloque que lo hace todo por ti ☐

[2022-04-14 08\\_10\\_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

[2022-04-14 08\\_11\\_21-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

Ejemplos de uso:

- [Alarma en Domótica con Arduino](#)
- [Piano invisible en Arduino con ArduinoBlocks,](#)
- [Sensor parking en Arduino con ArduinoBlocks](#)
- [Piano invisible en Arduino con mBlock](#)
- [Sensor parking en Arduino con mBlock](#)
- [Sensor de distancia de ultrasonidos con Picobricks](#)

## Sensor DHT11 (Temperatura y Humedad).

Este sensor de temperatura y humedad **DHT11** nos permite [determinar las zonas de confort](#) para un rango de temperaturas entre 0°C y 50°C con un error de  $\pm 2^\circ\text{C}$  y un rango de humedad entre 20 y 90 %  $\pm 5\%$ . Una salida digital para dos variables cómo lo hace? Tiene dentro un pequeño microprocesador que lanza por el bit de datos 40 bits en serie, los 16 primeros son la humedad (en BCD) y los 16 restantes es la temperatura (en BCD) los 8 restantes son de comprobación

**Checksum** (en binario) como la letra del DNI. Por ejemplo `0100 0111 0000 0011 0001 1001 0000 0000 0001 1000` es `0100 0111 0000 0011` = 47.03% de humedad y `0001 1001 0000 0000` = 19.00°C y la comprobación es la suma de  $4+7+0+3+1+9+0+0=24=11000$

Ejemplos de uso:

- [Medir H y T con Blink en Rover con Arduino](#)
- [Estación meteorológica Arduino con Arduinoblocks](#)
- [Arduinoblocks en el aula](#)
- [SMART HOME con Micro:bit](#)

2022-04-13 19\_27\_43-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

No es un sensor con gran sensibilidad, pero para propósitos educativos cumple sus funciones. Por dentro tiene una resistencia NTC que decremента su resistencia si aumenta la temperatura. Hay otros que van al revés, los PTC. Tanto los NTC como los PTC se llaman **thermistores**. Para la humedad, mide la capacidad de un condensador que es sensible a la humedad, o sea, un **sensor capacitivo**.

Tenemos dos opciones comerciales: **Encapsulado** que lo tienes preparado para conectar la alimentación y leer por el pin de datos, o **sin encapsular**, que hay que colocar una resistencia de aproximadamente 10k entre Vcc y Data

[dht11-pines.png](#)

Ejemplo de uso de un DHT11 sin encapsular:

[dht11sinencapsular.png](#)

Fuente Luis LLamas CC-BY-NC-SA <https://www.luisllamas.es/arduino-dht11-dht22/>

Si queremos mejorar la sensibilidad, podemos utilizar el **DHT22** que es igual pero de color blanco y más caro. Si lo que queremos es sólo la temperatura es mejor utilizar el **LM35D** que tiene un rango de temperaturas desde 0°C a 100°C con una sensibilidad de 2mV/°C

[lm35.jpg](#)

Es un sensor bastante mediocre, si necesitas una precisión el doble, te recomendamos el DHT22 que funciona exactamente igual pero es de color blanco y más caro ~3€. Ver <https://www.luisllamas.es/arduino-dht11-dht22/>

## Sensor IR

Es un sensor para distancias cortas hasta 2cm y no da la distancia, simplemente si hay o no hay obstáculo, pero son muy baratos, unos 0.30€. [Aquí tienes un ejemplo de evita obstáculos en un rover marciano con Raspberry](#) Para saber más te recomendamos [esta página de Luis Llamas](#)

<https://sketchfab.com/models/6ad4f3afb83940fea95cd3846aa68a18/embed>

[IR Sensor Module for Arduino Projects | 3D Model](#) by [Veer AI](#) on [Sketchfab](#)

## Sensor llama

Este sensor de llama se puede utilizar para detectar fuego u otras luces cuya longitud de onda se encuentra entre 760 nm ~ 1100nm.

Un ejemplo de su uso:

- [Alarma por fuego en Domótica con Arduino](#)

2022-04-14 08\_06\_56-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

2022-04-14 08\_07\_38-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

## Sensor de Gas (MQ2).

Detecta gases inflamables : GLP, I-butano, propano, metano, alcohol, hidrógeno, humo... con más sensibilidad en algunos que en otros. Siempre detecta el conjunto. Son usados en electrónica de consumo y mercados industriales.

- **Sensibilidad** Tiene alta sensibilidad y se puede ajustar girando el potenciómetro.
- **Tiempo de respuesta:** Internamente posee un calentador para aumentar su temperatura y que estos gases reaccionen con la resistencia interna que tiene, por lo tanto tardan algo en responder la primera vez que se conectan, incluso horas en algunos modelos. Una vez calentados son rápidos en la respuesta.
- **Tipo de salida:** Analógico pero si tiene 4 pines como el de la figura, incorpora un pin digital.
- Ejemplos de uso:
  - [Smart Home Microbit](#)
  - [Smart Home ESP32](#)

2022-04-14 11\_05\_36-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

## Sensor de humedad de suelo.

“ Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. Son ampliamente empleados en sistemas automáticos de riego para detectar cuando es necesario activar el sistema de bombeo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. No tiene la precisión suficiente para realizar una medición absoluta de la humedad del suelo, pero tampoco es necesario para controlar un sistema de riego. Los valores obtenidos van desde 0 sumergido en agua, a 1023 en el aire (o en un suelo muy seco). Un suelo ligeramente húmedo daría valores típicos de 600-700. Un suelo seco tendrá valores de 800-1023.

Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>

[Se puede utilizar este sensor para hacer un dispositivo de riego automático, puede detectar si las plantas “tienen sed” y evitar que se marchiten.](#)

La corriente de trabajo del sensor es menor de 20mA. El voltaje de salida es de 0V (en el aire) a 2,3V (totalmente sumergido en agua).

- [Smart Agriculture Kit micro:bit](#)



2022-04-14 08\_15\_34-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png

## Sensor de humedad.

Este sensor **analógico** está diseñado para identificar y detectar la presencia de agua y su cantidad. Puede servir para detectar el nivel de agua, para disparar una alarma en caso de una fuga de agua, también para hacer un limpiapalabrisas automático.... puedes ver un ejemplo de uso en :

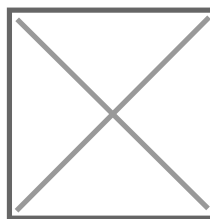
- Curso de [Domótica con Arduino](#)
- [Smart Agriculture Kit micro:bit](#)

Mide el volumen de agua caída a través de una serie de rastros de cables paralelos expuestos.

2022-04-14 08\_19\_22-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png  
2025-11-08 18\_13\_20-.png

## Sensor de efecto Hall.

Este es un sensor de inducción magnética. Detecta los materiales magnéticos dentro de un rango de detección de hasta 3 cm. El rango de detección y la fuerza del campo magnético son proporcionales. La salida es digital.

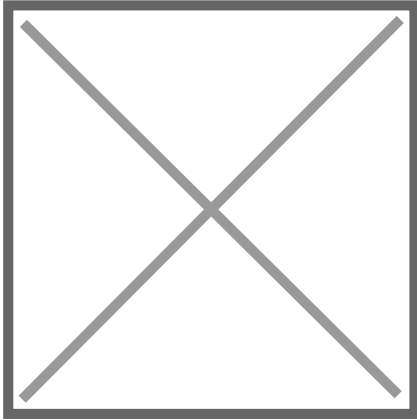


2022-04-14 10\_07\_05-1.3 Conexiones \_ Librería CATEDU.png

Sensor Hall.

*Edwin Helber Hall De Desconocido - [Popular Science Monthly Volume 64, Dominio público](#)*

[Edwin Helbert Hall](#) descubrió en 1879 que en presencia de un campo magnético, un conductor que conduzca una corriente se le producía un campo eléctrico porque las cargas eléctricas se desviaban de su trayectoria principal, nuestro sensor simplemente mide ese campo eléctrico:



De [Luis Llamas](#) CC-BY-NC

El sensor tiene un led de color rojo que indica que hay una lectura de campo magnético. Un ejemplo de uso lo puedes ver aquí: [medir rocas magnéticas con el Rover con Arduino](#)

## Sensor inclinación

Este sensor funciona al hacerle vibrar, emitiendo una señal digital de todo o nada. El módulo del sensor viene provisto de un potenciómetro para poder regularlo.

[2022-04-14 11\\_01\\_13-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

## Sensor de golpe

Es un sensor digital que al ser golpeado este sensor envía una señal momentánea.

[2022-04-14 11\\_02\\_09-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

## Sensor de pulso cardíaco.

Este módulo utiliza un LED infrarrojo (IR) ultrabrillante y un fototransistor para detectar el pulso en el dedo. Principio de funcionamiento: Se debe colocar el dedo entre el LED infrarrojo ultrabrillante (parte superior) mientras que el fototransistor, que queda en el otro lado, recoge la cantidad de luz transmitida. La resistencia del fototransistor variará levemente a medida que la sangre pase a través de su dedo.

[2022-04-14 11\\_03\\_47-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

## Sensor de Alcohol.

Este sensor de gas analógico MQ-3 es adecuado para detectar alcohol. Se puede usar en un analizador de aliento. También tiene una alta sensibilidad al alcohol y baja sensibilidad a la bencina

(éter de petróleo). La sensibilidad se puede ajustar con el potenciómetro.

[2022-04-14 11\\_04\\_42-PROYECTO 00 CONOCEMOS ARDUINO..pdf - Google Drive.png](#)

## Sensor de CO2

Hay sensores que utilizan **el protocolo I2C**, este protocolo permite conexiones serie y pueden compartir el mismo cable pues cada elemento tiene una dirección diferente. Esto lo veremos en el Display LCD. Se identifican por los pines SDA y SCL

CCS811-KEYSTUDUUDIO.png

## Resistencia Flex

Es una resistencia que cuanto más se dobla más resistencia ofrece, desde 25k hasta 125k

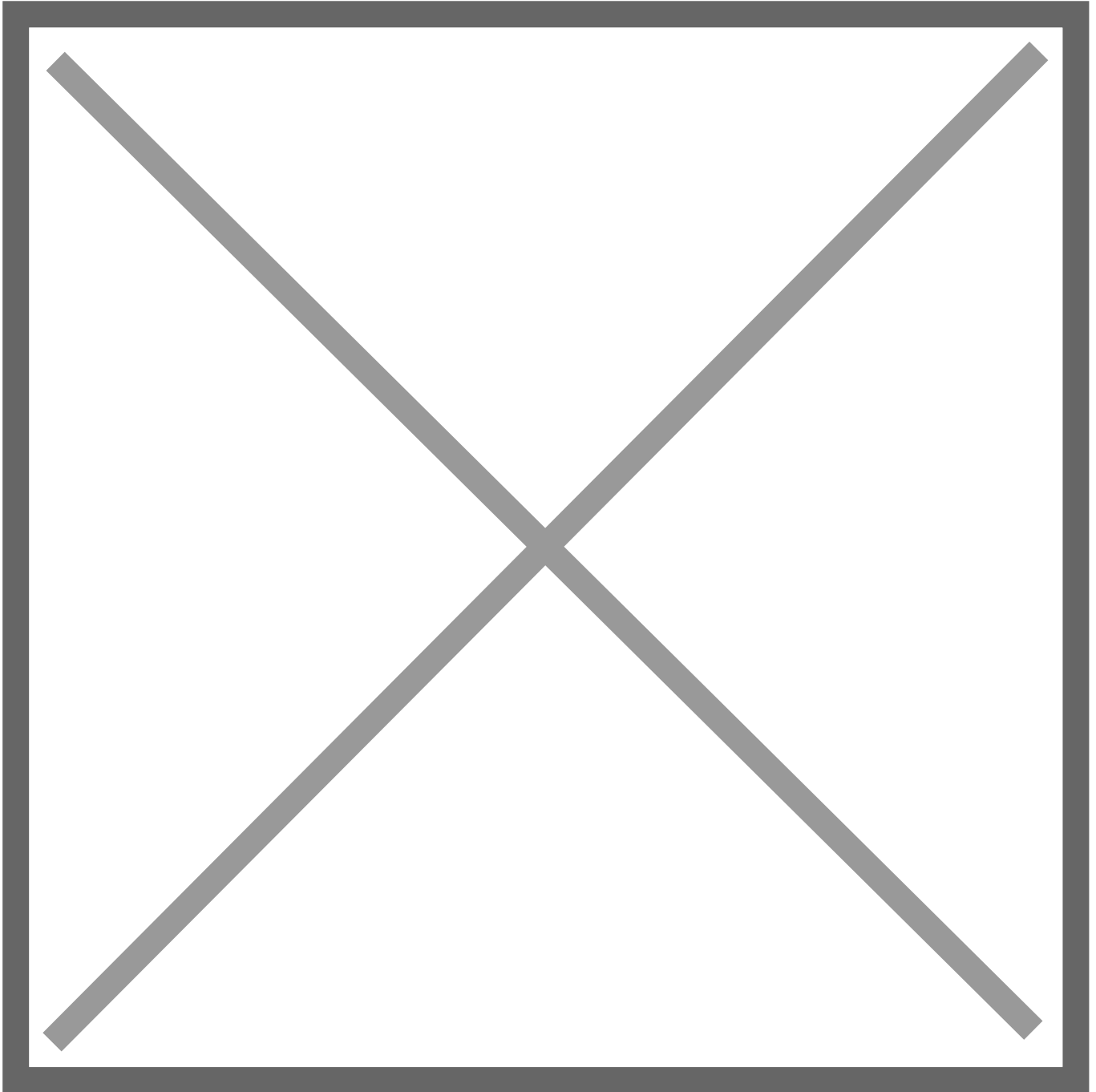
[Datasheet Sparkfun](#)

Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable, esa tensión es variable y ya tenemos la entrada **analógica**:



Es decir:

- La resistencia entre masa GND del ARDUINO (cable negro) y un punto en la placa protoboard
- ese punto medio conectarlo a una entrada analógica, por ejemplo A0 (cable amarillo)
- Una resistencia de valor parecida a la Flex de decenas de K entre ese punto y +5V (cable rojo en la foto)



Este sensor tiene posibilidades para usarlo en "ropa inteligente".

## Sensor de movimiento

“ Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta

razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

Los sensores PIR **se basan en la medición de la radiación infrarroja**. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piro eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica. En realidad **cada sensor está dividido en dos campos** y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

[2026-02-09 12\\_25\\_59-Greenshot.png](#)

El otro elemento restante para que todo funcione es **la óptica del sensor**. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

[2026-02-09 12\\_27\\_59-Inbox - jjquintana@educa.aragon.es - Mozilla Thunderbird.png](#)

Luis Llamas CC-BY-NC-SA <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>

Puedes ver ejemplos de uso en robótica en :

- [Smart Home para Microbit](#)
- [Smart Agriculture Kit para Microbit](#)

[2022-04-16 10\\_24\\_22-sensor pir arduino at DuckDuckGo.png](#)[2025-11-08 18\\_11\\_48-.png](#)

Más sensibles son los **sensores de microondas**. Son un radar que por efecto Doppler pueden captar cualquier objeto en movimiento dentro de un alcance de 5-7 metros en cualquier dirección e independiente de su temperatura. Es un buen sensor para alarmas, activación de luz por presencia..... Para saber más [ver la página de Luis Llamas](#)

[image-1650097316026.png](#)



Su conexión es muy sencilla, es un detector digital que hay que alimentarlo como el resto de sensores.

[image-1650097889205.png](#)

Curiosamente, la gran ventaja técnica de los de microondas **es un gran inconveniente para usarlo en el aula**, con cualquier movimiento se dispara, luego **para clase es mejor el sensor PIR**

Esta página esta adaptada de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA.

# Actuadores y otras salidas

Esta página es un repaso rápido de las posibles **salidas** de los robots  
**NO LEAS TODOS SÓLO LOS QUE TIENE TU ROBOT**

## ¿Qué es un actuador?

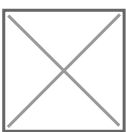
“ Un actuador es el nombre genérico que empleamos para cualquier **dispositivo capaz de realizar acciones en el mundo físico** y que podremos controlar desde un autómatas o procesador como Arduino. En particular, usamos el nombre actuador para los dispositivos que son capaces de generar movimiento.

[Luis LLamas](#) CC-BY-NC-SA

## ACTUADORES

### Motores baja potencia

Con el Arduino podemos usar motores de corriente continua de juguete como en la figura, aptos para poner un ventilador. Ejemplo de uso [Smart Home para microbit](#)

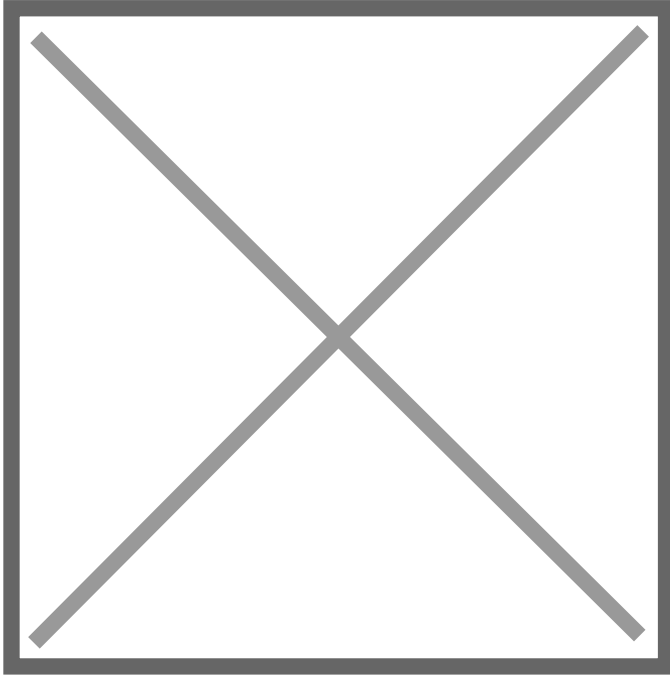


### Motores con algo de potencia

Si quieres usar un motor para mover unas ruedas, ya pide más corriente, no se puede conectar directamente a la placa (Arduino, esp32 etc..). El de la figura es típico en robótica y lleva reductores dentro para bajar la velocidad :

[2022-04-16 10\\_40\\_41-motor arduino at DuckDuckGo.png](#)

Necesita una alimentación extra, como el LM298N y las pilas., o el B6612FNG o una shield específica.



CC-BY-NC [Luis Llamas](#)

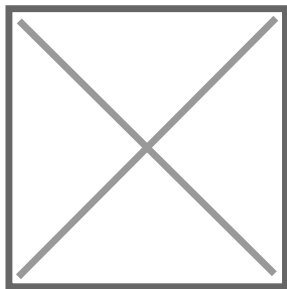
Ejemplos de uso lo puedes ver en

- [Curso mClon con nanoArduino con el B6612FNG](#)
- [Curso Rover con Arduino con la shield L293D ESP-12E para el NODE MCU que internamente tiene el LM298N](#)
- [Curso Arduino con código con LM298N](#) y también con un transistor

Los motores pueden (y deben) de estar conectados a **salidas PWM** de tal manera que se pueda regular la potencia y por lo tanto la velocidad.

## Servos

Un [servo convencional](#) es un motor donde fijamos el ángulo desde 0º a 180º, pero si queremos una rotación, existen [servos rotatorios](#) que simplemente tienen su velocidad de rotación controlada



Ejemplos de uso de servos:

- [Curso Arduino con código](#)
- [Curso brazo robótico del mClon con nanoArduino](#)
- [Apertura de barrera por ultrasonidos en curso Arduino con ArduinoBlocks](#)
- [Tractor entrando en el corral Arduino con EchidnaShield](#)
- [Apertura de puerta Domótica con Arduino](#)
- [Servo con PicoBrick](#)
- [Apertura ventana y puerta en Smarth Home para microbit](#)
- [Smart Agriculture Kit para micro:bit](#)

## Relés

Un relé es un interruptor activado por un electroimán, lo que permite independizar los circuitos. En el dibujo se ve que el circuito rojo de 220V está separado del verde, de sólo 1.5V. Pero es el circuito verde que al funcionar, hace que el electroimán mueva el interruptor del relé y encienda la bombilla. El objetivo es que he podido encender una bombilla de 220V sin tocar los 220V peligrosos. En el circuito verde, el interruptor puede ser un Arduino. Experimentalo en este [simulador](#).

- Un ejemplo con Arduino en [Luis LLamas](#)
- Un ejemplo de uso en [Picobricks](#)

2025-01-09 18\_42\_34-Simple Relay with bulb - Simulador De Circuito On Línea \_ DCACLab.png

## Electroimán

El electroimán no se puede conectar directamente, utilizaremos una etapa de potencia, por ejemplo el transistor el [IRF520N](#) que amplifique la señal del Arduino, pues las salidas de Arduino no tienen potencia para mover el electroimán. Para dar esa potencia utilizaremos otra fuente externa, unas pilas:

[image-1649970623453.png](#)

Las conexiones son :

- SIG del IRF520N a una salida digital por ejemplo D13
- VCC del IRF5020N al 5V del ARDUINO
- Los dos GND del IRF520N a GND del ARDUINO
- V+ y V- del IRF5020N al solenoide, da igual qué cable pues no tiene polaridad.
- VIN del IRF520N al VIN del ARDUINO (son los voltios de la pilas)

[image-1649970676310.png](#)

## Motor paso a paso

Igual que el [electroimán](#), necesitamos un controlador que nos de la potencia necesaria para mover el motor, el ULN2003



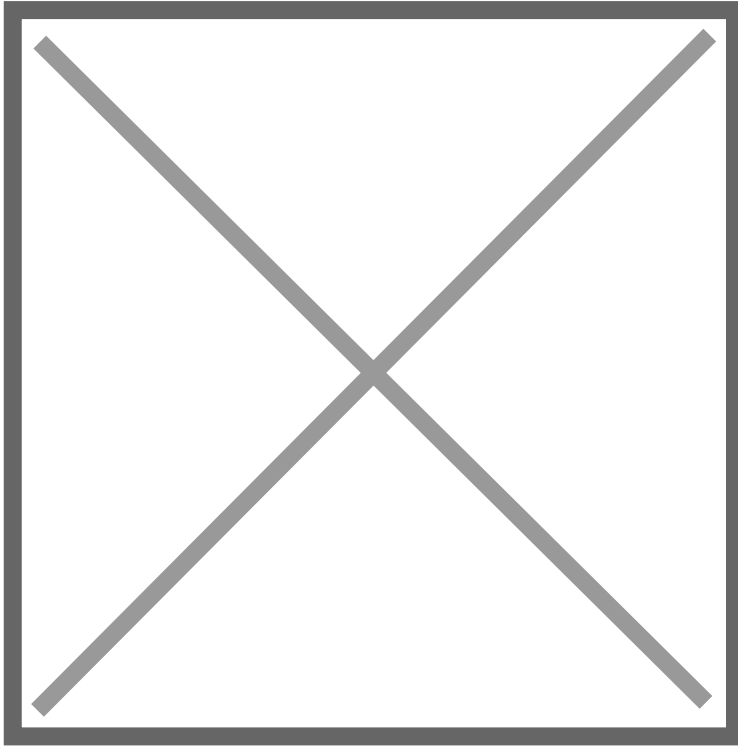
Imagen BY-NC-SA de [Luis Llamas](#)

También igual que el electroimán, necesitamos una potencia extra con las pilas:



Conexión :

- Cuatro pines digitales del ARDUINO al IN1,IN2,IN3,IN4 del ULN2003 por ejemplo D10,D11,D12,D13
- El conector blanco del ULN2003 al motor paso a paso
- El (+) del ULN2003 al Vin del ARDUINO
- El (-) del ULN2003 al GND del ARDUINO



La configuración más sencilla es la rotación simple en sentido horario (llamada fase1) :



Imagen BY-NC-SA de [Luis Llamas](#)

Es decir:

| Paso          | IN1=D10 | IN2=D11 | IN3=D12 | IN4=D13 |
|---------------|---------|---------|---------|---------|
| <b>Paso 1</b> | ON      | OFF     | OFF     | OFF     |
| <b>Paso 2</b> | OFF     | ON      | OFF     | OFF     |
| <b>Paso 3</b> | OFF     | OFF     | ON      | OFF     |
| <b>Paso 4</b> | OFF     | OFF     | OFF     | ON      |

Si has entendido este ejemplo serás capaz de realizar las configuraciones que desees.

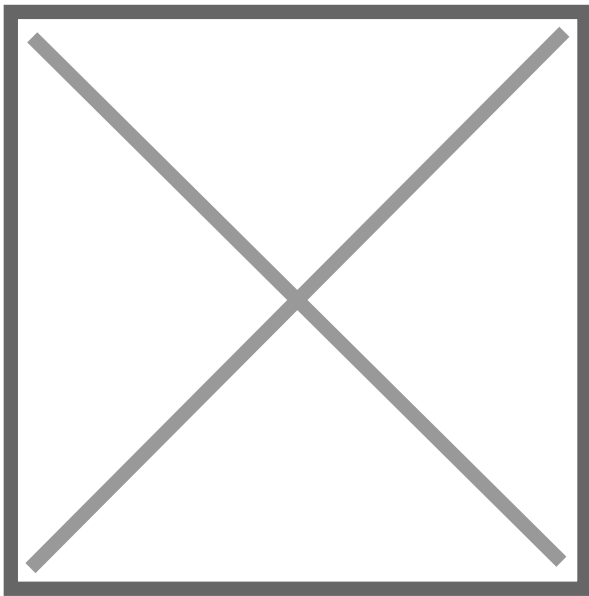
## OTRAS SALIDAS

No son actuadores pues no representan movimiento pero son también **salidas** del Arduino como los actuadores, luego lo vemos en esta página

## Buzzer activo

Reproduce un tono de una frecuencia determinada por el fabricante cuando recibe un '1' digital. Su conexión es muy simple:

- **GND** a GND y **VCC** a 5V (ojo que están a los dos extremos, marcados en rojo y en negro)
- **OUT** o también señalado como I/O a un pin digital, por ejemplo D13



Ejemplos de uso:

- Domótica con Arduino: [Apertura puerta con mBlock](#)
- Domótica con Arduino [Apertura puerta con código](#)

## Buzzer pasivo

La diferencia de un buzzer activo con el pasivo es que el pasivo hay que mandar la onda que se tiene que reproducir, como Arduino no puede reproducir onda puras (senoidales analógicas), se le envía ondas cuadradas con la frecuencia que se pretende reproducir. Como se puede ver en la figura, no tiene elementos auxiliares para reproducir un tono, es simplemente un altavoz.

[zumbadorpasivo.png](#)

Ejemplos de uso

- [Timbre Arduino con Echidna](#)
- [Arduino blocks en el aula](#)

## LEDS Y OTROS ELEMENTOS VISUALIZADORES

### Led normal

Un diodo es la unión de dos semiconductores P-N que sólo permite la corriente en un sentido. Un diodo LED es un diodo, que al pasar esta corriente emite una luz. Hay de 3mm y de 5mm de grosor, transparentes, traslúcidos y de colores (realmente son traslúcidos tintados)

Tenemos pues que respetar su polaridad para que funcione:

[arduino-led-patillaje.png](#)

Fuente Luis LLamas CC-BY-NC [Encender LED con Arduino](#)

Si alimentamos el LED con una tensión inferior a su tensión de polarización directa  $V_d \pm 1,4V-3,8V$  el led no luce. si alimentamos con una tensión superior, la corriente que circula se dispara por lo que se rompe. Conclusión: Hay que poner una resistencia limitante. Para calcularla te recomiendo la página de Luis Llamas [Encender LED con Arduino](#)

Aquí puedes ver un ejemplo de semáforo [en el curso Arduino con código](#)

[image-1650005625137.png](#)[image-1650005638493.png](#)

*Se puede conectar directamente pues las salidas de Arduino tienen una limitación interna de 20mA como en este ejemplo <https://www.youtube.com/watch?v=EFFSLvIF9rY>*

<https://www.youtube.com/embed/EFFSLvIF9rY>

Los LEDS igual que los motores pueden ser encendidos o todo o nada con salidas digitales 0 y 1 o salidas PWM de esta manera conseguimos luminosidades medias, en esta práctica con el potenciómetro se regula la intensidad de la luz de un LED

## Led RGB

Se trata de un encapsulado que integra 3 leds de los tres colores básicos. Cada color básico se puede regular desde 0 a 255 de intensidad, luego tenemos 16 millones de tonalidades diferentes.

[image-1650007895867.png](#)

Un ejemplo de su uso lo puedes ver en

- [Domótica con Arduino, interruptor crepuscular](#)
- [Intermitente RGB en Arduino con ArduinoBlocks](#)

Existen LEDS RGB que se comunican unos a otros, ([ver este ejemplo en el curso mClon](#)) como las tiras de leds RGB que cambian de color todos simultáneamente.

## Neopixel

La cinta LED RGB se basa en el chip controlador WS2812B de Neopixel con el cual podemos tener el control de cada pixel y de cada color, ya sea rojo, verde y azul a través de un solo cable de datos.

“ Los WS2811, WS2812 y WS2812B son LED que disponen de lógica integrada, por lo que es posible variar el color de cada LED de forma individual (a diferencia de las tiras RGB convencionales en las que todos los LED cambian de color de forma simultánea). Están basados en el LED 5050, llamado así porque tiene un tamaño de 5.0 x 5.0 mm.

Es un LED de bajo consumo y alto brillo, que incorpora en un único encapsulado los 3 colores RGB. La genial novedad del WS2812B (y resto de familia) es añadir un integrado dentro de cada LED, que permite acceder a cada pixel de forma individual. Por este motivo este tipo de LED se denominan “individual addressable”. Esto abre la puerta a un sinfín de aplicaciones y combinaciones, que van desde dotar de iluminaciones distintas zonas con una única tira, animaciones complejas, o incluso generar pantallas enteras de alta luminosidad....

A los LED WS2812B también se les denomina NeoPixel

Luis Llamas CC-BY-NS-SA <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>

[2025-11-08 18\\_27\\_31-.png](#)



- [Smart Home para Microbit con 4 leds RGB Neopixel](#)
- [Smart Home para ESP32](#)
- [Smart Agriculture Kit micro:bit](#)

## Láser

El diodo láser es un elemento motivador, barato y fácil de usar con el Arduino pues se activa digitalmente, si enviamos un 1 emite un rayo láser. Un ejemplo de uso lo puedes ver en la [Alarma por láser en Domótica con Arduino](#)



Si quieres saber más de este componente, te recomendamos [esta página de Luis Llamas](#).

Si tienes que comprar uno, es importante que no sea superior a 5mW, pues puede dañar permanentemente la retina del ojo [[+info](#)].

El modelo que proponemos es de **1mW**, no obstante, **EVITA SIEMPRE QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

[image-1649971311304.png](#)

## Display LCD

Si queremos visualizar datos de forma independiente del ordenador, existen matrices de led que puedes ver en los cursos de mBot o mClon pero lo mejor es una pantalla Liquid Crystal Display y la mejor forma de conectar esta interfaz es en serie utilizando el **protocolo I2C**.

Ejemplo de uso :

- en [Arduino con código](#)
- [Arduinoblocks en el aula](#)
- [Smart Home para Microbit](#)

[image-1650096059762.png](#)

CC-BY-SA [Luis Llamas](#)

No confundas Display LCD con matriz de LEDs , o una OLED

## Aclarando conceptos: Protocolo I2C

Es un protocolo de emitir los datos con sólo dos cables **SDA** (los datos) **y SCL** (reloj) y además pueden compartir varios dispositivos SLAVE) para ello la placa Arduino ESP32, microbit... (MASTER) tiene que saber **qué dirección tiene cada dispositivo IMPORTANTE:**

2025-11-08 18\_35\_50-.png

[Fuente Luis Llamas ArduinoI2C CC-BY-SA](#)

## Aclarando conceptos: Lógica invertida

Igual que vimos en sensores, con las configuraciones Pull up y Pull down, las salidas de cualquier actuador **PUEDEN SER LÓGICA INVERTIDA**, mira en este NodeMCU (básicamente es un Arduino con wifi, aquí está conectado con el servidor Blynk) como funciona al revés, cuando apretamos, por lo tanto enviamos un uno a la salida, **se apaga la salida !!**

<https://www.youtube.com/embed/GLvTBdcfKAI>

# Kit de préstamo de CATEDU Arduino

El presente curso parte de que el alumno tiene el kit de préstamo que facilita CATEDU:

[image-1648632032646.png](#)

## Arduino UNO

Se trata de una placa con un microcontrolador (**un micro ordenador**) y puertos digitales y analógicos de entrada y salida. ¿Por qué es tan famoso?

[image-1648632060408.jpg](#)

- Se trata de una placa de **hardware libre**
- Los programas para controlarlo son de **código abierto**
  - **Conexión por puerto USB** de esta manera, el Arduino obedece al programa del ordenador, pero si queremos: el programa lo grabamos dentro del Arduino y gracias a un conector, puede funcionar con pilas e independiente del ordenador

### Puertos:

- **14 pines digitales** numerados del **0 al 13** y se pueden ser:
  - De **salida** (sólo tiene dos estados LOW/HIGH)
  - De **entrada** (valores: mínimo 0V y máximo 5V)
- **6 pines de salida "analógicos"** 3-5-6-9-10-11 (*sí, has leído bien, se repiten con los anteriores*) tipo [PWM](#) (es decir, lo de *salida analógica es una mentirijilla*, es en formato de pulsos con duración proporcional al valor analógico que se quiera representar)
- **6 pines analógicos de entrada** numerados como **A0 hasta A5** (valores: mínimo 0V y máximo 5V) y ellos responden en formato de número entre 0 y 1023.

no está nada mal ¿no? y además barato !!!

# Resistencias

Las resistencias sirven para limitar la corriente. Lo utilizaremos en dos ocasiones:

- Limitar la corriente que circula en un diodo
- Limitar la corriente de carga de un condensador

También lo utilizaremos con el pulsador, para separar los voltios y para que tenga una tensión definida cuando el interruptor esté abierto (lo veremos en el siguiente capítulo)

Los valores de las resistencias en  $\Omega$  se determina mediante la siguiente tabla:

[image-1648632153105.jpg](#)

# Placa Protoboard

Placa protoboard o placa de pruebas es un tablero con orificios

embedded-image-mydxq7pe.png

Estos se encuentran conectados eléctricamente entre sí de manera interna, el patrón de conexión es de líneas según este esquema:

[image-1648632178895.png](#) [image-1648632186997.gif](#)

## ¿Cómo está conectado?

La línea X (pintada de rojo) está toda conectada y la línea Y (pintada de azul) también.

Las líneas verticales 1 al 47 están conectadas verticalmente separados por la línea del medio, pero no horizontalmente (he pintado de verde las 4 primeras) es decir:

- A1 SI está conectado con B1C1D1E1
- A1 NO está conectado con A2B2C2D2E2
- A1 NO está conectado con F1G1H1I1J1

embedded-image-sexdzfob.png

## ¿Cómo se utiliza?

La línea X (pintada de rojo) se suele utilizar para poner la alimentación (5V en nuestro caso) y la línea Y (pintada de azul) se suele conectar a 0V

Las líneas verdes es donde se conectan los componentes de tal manera que se pueda construir el circuito

# Pulsadores

[image-1648632240664.webp](#)

Un pulsador nos permitirá crear entradas digitales al Arduino

Los pulsadores proporcionados abren y cierran los pines que están en vertical, pero no en horizontal pues están conectados (ver figura de la izquierda)

Luego tenemos que fijarnos en los pines, si están en línea son los pines que no están conectados cuando no se pulsa, por lo tanto son estos los que tienen que ir en líneas diferentes de la placa protoboard.

Los pines que están en paralelo están conectados, por lo tanto tienen que estar en la misma línea:

[image-1648632276077.png](#) [embedded-image-f9y5ddki.png](#)

Pero ¿Cómo se conecta? un poco de teoría en la página siguiente.

## Esto esta mal

[image-1648632330132.png](#) [Fuente Luis Llamas https://www.luisllamas.es/leer-un-pulsador-con-arduino/](https://www.luisllamas.es/leer-un-pulsador-con-arduino/)

La sencilla razón es la siguiente ¿cuando el interruptor o pulsador esté abierto qué tensión tendrá "Pin digital"?

En vista de protoboard sería así:

[embedded-image-bgbvz7ax.png](#)

El primer pulsador cuando se cierra sí que proporciona 5V al pin 12 por el cable azul pero... ¿y cuando no está pulsado?

El segundo pulsador cuando se cierra sí que proporciona 0V al pin 13 por el pin verde pero... ¿y cuando no está pulsado?

Ah!! y por supuesto ni se te ocurra esto, provocarías un cortocircuito:

image-1648632391355.png

Fuente Luis Llamas <https://www.luisllamas.es/leer-un-pulsador-con-arduino/>

## Esto esta bien

Lo correcto es hacerlo a través de resistencias, hay dos configuraciones, pull down o pull up según si la resistencia está abajo o arriba

image-1648632523286.png      embedded-image-noukkdmt.png

la más utilizada y lógica es la primera pull down, pues lo lógico es que cuando se cierra el pulsador, queremos transmitir un 1 lógico

*Resumiendo: lo que tienes que hacer es una de estas dos configuraciones, la opción recomendada es PULL UP para que OFF = 0 y ON = 1*

image-1648632493133.png

Fuente Luis Llamas <https://www.luisllamas.es/leer-un-pulsador-con-arduino/>

El valor de la resistencia puede ser de valores de algunos k

Visto en la placa el formato **pull down**:

embedded-image-ephsjoiq.png

Es decir (en la foto el cable verde del anterior dibujo es el blanco en la foto):

embedded-image-ui4hzkpo.png

## CONEXIÓN CON ARDUINO

Dos ejemplos sencillos para utilizar el pulsador, utilizaremos la resistencia configuración pull down y lo conectaremos al Arduino por A0

Utilizaremos la configuración PULL-DOWN es decir la resistencia "abajo" y el pulsador arriba, de esta manera:

- Cuando e pulsador está **abierto**, entonces enviamos un '**0**' lógico al Arduino.

- Cuando el pulsador está **cerrado**, entonces enviamos un '1' lógico al Arduino.



La forma de conectarlo en la placa Protoboard es (en este ejemplo lo conecta al pin digital 4):



Si quieres saber su contraria (Pull-up) [visita esta página.](#)

# LED

LED= light-emitting diode: es un diodo emisor de luz que tiene polaridad, es decir, que tiene + y -



## ¿Cómo se conecta?

Pues a través de una resistencia, de valor de algunos k o centenares de  $\Omega$ , cuanto más pequeña más se ilumina pero más peligro de fundir el led

En el ejemplo : conexión a la salida del pin 13 :

embedded-image-aomu56.png

NOTA: A veces conectaremos el LED directamente, por simplificar, pues el ARDUINO TIENE UNA RESISTENCIAS INTERNAS DE 30k pero no es una buena práctica

embedded-image-yf3siwlx.png

## Ejemplo de utilización

Mira el siguiente circuito **NO LO HAGAS SÓLO MÍRALO Y OBSERVA:**

### La alimentación

- La línea X **en esta placa está abajo del todo (para que veas otro modelo)** y se ha conectado con los 5V del Arduino con un cable rojo
- - La línea Y es la segunda empezando de abajo y se ha conectado a 0V (GND del Arduino) con un cable azul

### El pulsador

- Se ha conectado un pin a masa a través de una resistencia y un pequeño cable azul
- Ese pin será la entrada al Arduino y con un cable negro va al pin digital 8 del Arduino

- El otro extremo a 5V con un cable rojo

## EI LED

- Un extremo (el ánodo pata larga) a las diferentes salidas digitales del Arduino (pines 7-6-5-4-3-2) a través de una resistencia
- El otro extremo (cátodo pata corta) a masa-Linea Y

[image-1648632735275.png](#)

[image-1648632740892.jpg](#)

# Condensador

## ¿Qué es?

Es un elemento que consta de dos placas separadas que son capaces de cargarse eléctricamente, su capacidad se mide en Faradios. Se puede interpretar como un globo que nos permite hincharlo de aire, y la capacidad los litros. Pero también pueden **explotar** !!! luego lee con atención:

**ATENCIÓN** Los condensadores (*normalmente a partir de  $1\mu F$  son electrolíticos*) tienen **POLARIDAD QUE ES IMPORTANTE**

**RESPETARLA**, es decir, tienen un pin marcado con el signo (-) normalmente con franja blanca que tiene que ir al (-) o masa (GND 0V) de lo contrario, revientan, con el consecuente peligro, más peligrosos cuanto más capacidad tienen.

**ATENCIÓN** *Los condensadores tienen una tensión máxima de trabajo, marcado en el propio condensador, en nuestro caso los condensadores marcan 25V, luego se pueden utilizar sin problemas con el Arduino que la tensión máxima es 5V. Como si fuera la máxima presión que aguanta el globo antes de explotar.*

¿No te lo crees? pues mira..

<https://www.youtube.com/embed/sfAbv-0hQsU>

## Carga y descarga

Si lo hacemos a través de una resistencia, tarda un tiempo a cargarse y también a descargarse (como si en el globo lo deshincháramos pero estrechando la boca)

El circuito de carga sería

[image-1648632777199.png](#)  [embedded-image-o4heu7jo.png](#)



Fíjate que el lado (-) (la banda blanca) está conectado a GND

El **circuito de descarga** sería

[image-1648632819763.png](#)      [embedded-image-bmhxihhf.png](#)

## Cálculos

El tiempo de carga y de descarga se calcula con la fórmula :

$$T = 5 R C$$

En el caso de los circuitos de arriba  $T = 5 * 100k * 10\mu = 5 * 100\ 000 * 0.00001 = 5$  segundos

Igual que un globo, la carga evoluciona muy rápidamente al principio, pero a medida de que se va llenando, cuesta más, la gráfica de carga de un condensador es la siguiente:

[image-1648632850406.jpg](#)

Y para la descarga es la siguiente

[image-1648632859742.jpg](#)

# LDR

LDR= Light Depended Resistor es una resistencia que disminuye su valor cuando aumenta la luz.

Si se coloca en serie con una resistencia de un valor parecido, el punto de unión (cable verde) tiene una tensión variable con la luz que lo puede leer Arduino, en este esquema lo lee por la entrada A0

embedded-image-2xq9ysgy.png

Si hay luz, el LDR disminuye su resistencia por lo tanto disminuye el valor de la tensión en A0, y viceversa.

# Sensor ultrasonidos

El sensor dispone de un emisor T que emite una señal de ultrasonidos (a veces no es ultrasonidos, pues algunos lo pueden oír) y un receptor R que capta el eco del objeto que está delante de él. Su rango de detección va desde 2cm a 450cm

embedded-image-l1bcibrj.jpeg

Tiene cuatro pines, dos son la alimentación que conectaremos Vcc a 5V, GND a 0V y los otros dos son los importantes:

- Trigger que emite un pulso
- Echo que recibe el pulso

Según el tiempo entre el pulso y el eco (T), la distancia a la cual se encuentra el objeto es :  
distancia =  $T * \text{velocidad\_sonido} / 2$

Podemos conectar el pin trigger y echo donde queramos de las entradas y salidas digitales del Arduino, en este curso vamos a tomar como criterio:

- Trigger en el pin 4
- Echo en el pin 3

embedded-image-rekl7foj.png

**No tenemos que utilizar la fórmula para calcular la distancia, mBlock ya lo hace** y la instrucción es muy sencilla, sólo hay que proporcionarle qué pin hemos conectado el trigger y qué pin hemos conectado el echo y él nos devuelve la distancia en centímetros:

embedded-image-bgfltmpz.png

# Métodos para interactuar los objetos y el robot (dispositivo)

Si queremos que nuestro robot se pueda comunicar con el objeto o objetos que tenga mBlock, antes con mBlock 3.0 la comunicación era inmediata, fíjate en este script de una alarma:

[embedded-image-xGEexnUg.jpeg](#)

Mezcla en el mismo script:

- órdenes específicas de la placa arduino (set digital...)
- órdenes específicas del objeto que exista en mBlock (por defecto el oso panda) say ....  
switch costume to ....
- órdenes del fondo switch backdrop to ...

Con mBlock 5.0 **YA NO SE PUEDE**, pero tenemos unos trucos

## MÉTODO UTILIZAR VARIABLES GLOBALES

Se pueden crear variables, en cualquier objeto, y las lee cualquier objeto,

**Este método se utiliza con la opción EN VIVO**

[variablesglobales.jpg](#)

De esta manera si creamos una variable **frase** para todos los objetos:

[frase3.jpg](#)

Podemos usarla en el robot

[frase1.jpg](#)

y el programa del objeto que queramos, en este caso el oso panda lo puede visualizar

[frase2.jpg](#)

# MÉTODO UTILIZAR MENSAJES

Cualquier objeto tiene a su disposición enviar mensajes a los otros

**Este método se utiliza con la opción EN VIVO**

[mensajes.jpg](#)

# MÉTODO EXTENSIÓN BROADCAST = TRANSMITIR MENSAJES

**Este método se utiliza con la opción EN CARGA**

Es parecido al anterior, hay que ir al + que hay abajo para instalar extensiones

[arduino-pelotahinchadistancia3.png](#)

Buscar la extensión "Broadcast" e instalarla

[broadcast.jpg](#)

Se instala primero descargándola con el + aquí

[broadcast1.png](#)

Y luego añadir una vez descargada

[broadcast2.png](#)

Entonces aparecen unas nuevas instrucciones

[broadcast3.png](#)

y se instalan unas instrucciones extras parecidas a las anteriores pero más potentes

- En el mismo mensaje podemos transmitir valores asociados
- Funciona **EN MODO CARGA**
  - Esto es muy útil pues hay instrucciones que sólo se pueden utilizar en modo CARGA, de esta manera podemos pasar valores de la placa electrónica a los objetos del ordenador (oso panda o lo que sea) simplemente teniendo conectado la placa con el ordenador.

El resto de objetos trabajan en modo vivo, es decir, si cambias un bloque, automáticamente se ven los efectos

A lo otros objetos TAMBIÉN hay que instalar la extensión BROADCAST

**DESVENTAJA** no se pueden transmitir mensajes de objetos a la placa. Sólo de la placa a los objetos