

# 5. Arduino

- Microcontrolador: Arduino
- Sensores
- Actuadores

# Microcontrolador: Arduino

Visitando la [web](#) de Arduino, podemos leer:

*"Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla "*

Todos los proyectos que se van a desarrollar en este curso, lo harán sobre la placa de Arduino UNO, por tanto, esta placa va a contener el cerebro de nuestros proyectos y hará de interfaz de entrada/salida de nuestros sensores y actuadores.

A lo largo de los siguientes *ítems*, abordaremos qué es y cómo podemos utilizar esta placa para adentrarnos en el mundo de la robótica.

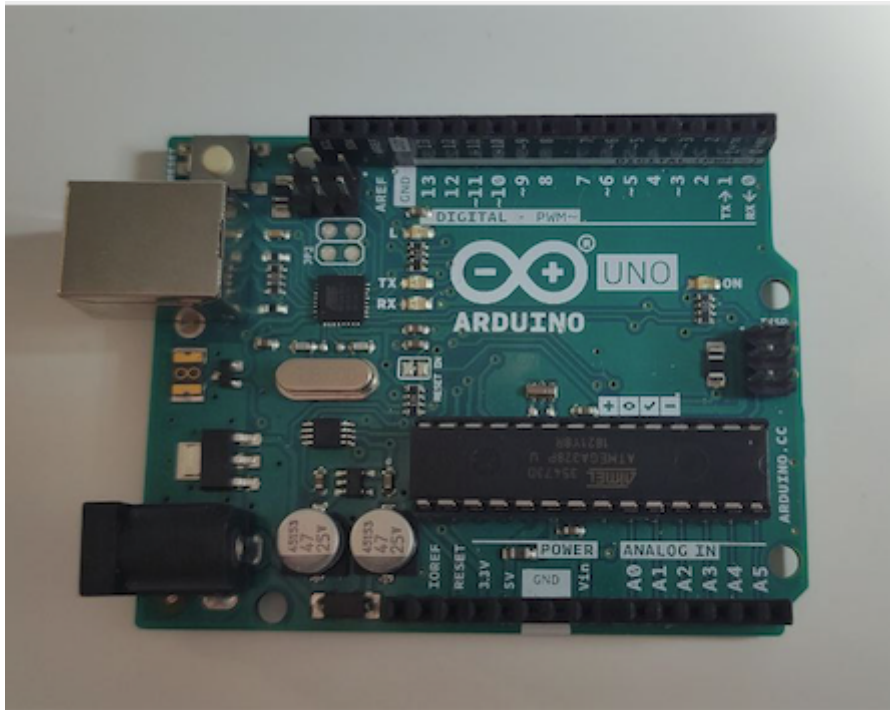
## Un poco de historia

Hoy en día existen numerosas plataformas, con microprocesadores similares al de Arduino que pueden ser utilizadas en el aula. Todas ellas son compatibles con el IDE de Arduino, y esto no es casual, sino más bien obligatorio. El IDE es el entorno de desarrollo integrado que ofrece dicha plataforma y en este caso, la de Arduino es la más completa y compatible con otros lenguajes de programación.

Esto es así porque Arduino fue la primera plataforma que se creó en el año 2005 el Instituto de Diseño Interactivo de Ivrea (Italia). Arduino apareció por la necesidad de contar con un dispositivo para utilizar en aulas que fuera de bajo coste. La idea original fue, fabricar una placa para uso interno de la escuela.

Por último, es obligatorio mencionar en este apartado a David Cuartielles, uno de los cofundadores de Arduino, por ser el único español en el equipo, en concreto, Zaragozano.

## Arduino desde la mirada del alumnado



Una definición simple de robótica podría ser el resultado de combinar sensores y actuadores para que en su conjunto hagan determinadas acciones. En este sentido, para que una alarma se dispare cuando detecta una intrusión, un robot de limpieza gire al detectar un obstáculo, o un limpiaparabrisas se active al caer las primeras gotas de lluvia, requieren además de los sensores y actuadores que componen estos sistemas, una unidad central capaz de procesar el algoritmo lógico que va desencadenando las acciones.

Arduino reúne todas estas características. Cuenta con una serie de pines que serán configurados como entradas, en el caso de querer leer información procedente de los sensores, o serán configuradas como salidas en el caso de querer accionar un actuador. Más adelante explicaremos en profundidad los sensores y actuadores.

Además, cuenta con un microprocesador reprogramable sobre el cual vamos a poder subir o cargar nuestro programa y se encargará de leer la información de los sensores conectados en los pines de entrada, y activar (poner en estado ALTO o 5 voltios) o desactivar (poner en estado BAJO o 0 voltios) los pines de salida.

Dicho de otro modo, y tomando uno de los ejemplos anteriormente mencionados, podríamos configurar nuestro Arduino para crear un sistema de seguridad simple, del siguiente modo:

Definiremos el Pin2 de nuestro Arduino como una entrada. En dicho pin, conectaremos nuestro sensor de presencia que siempre se encuentra a 0 voltios, excepto cuando detecta una presencia y entonces se eleva a 5 voltios. Por tanto es muy sencillo: únicamente tenemos que leer repetidamente (en bucle) el valor del Pin2 de nuestro Arduino, y preguntarnos si vale 5 voltios, o en su defecto se encuentra en estado alto. Si esta condición se cumple es que hay una intrusión y por tanto deberemos activar la sirena para alarmar de ello.

La sirena se comporta como un actuador. Se activa cuando se cumple la condición de haber una intrusión. Para conseguir esto, únicamente debemos definir el Pin3 de nuestro Arduino como salida y mediante la programación la activaremos en el momento correspondiente.

Como puedes ver, el mayor peso va a estar siempre en nuestra programación. Debemos tener mucho cuidado con la secuenciación y contemplar toda la lógica que va a definir el comportamiento, y por tanto buen funcionamiento de nuestros proyectos. También va a ser de suma importancia como definimos en cada uno de ellos, los pines de nuestro Arduino, para lo cual tendremos que tener siempre claro cuales vamos a utilizar, y si van a ser de entrada o de salida. En el último punto, trabajaremos unos ejercicios sencillos que comenzarán a abrir tu imaginación y creatividad que junto a la del alumnado serán los únicos límites de vuestros proyectos, pero antes de llegar ahí, veamos en profundidad los sensores y actuadores y como programar nuestro Arduino desde un entorno gráfico.

Para conocer más sobre Arduino puedes acudir a los siguientes enlaces, pertenecientes al libro [Programa Arduino Mediante Código](#) de **CATEDU**

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



# Sensores

Esta sección es una visión rápida de las posibles entradas del Arduino y está adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

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**.

The image shows the Arduino IDE's block programming interface. On the left is a sidebar with a category list: Lógica, Control, Matemáticas, Texto, Variables, Listas, Funciones, Entrada/Salida, Tiempo, Puerto serie, Bluetooth, **Sensores** (highlighted), Actuadores, Pantalla LCD, Memoria, Motor, Motor-Shield, Keypad, Reloj RTC, and GPS. The main workspace contains several sensor blocks stacked vertically:

- Potenciómetro**: Includes a potentiometer icon, a pin dropdown set to 'A0', and a unit dropdown set to '%'. It is connected to an 'Enviar' block with the text 'ArduinoBlocks!'.
- Pulsador**: Includes a pushbutton icon, a pin dropdown set to '2', and a 'Lógica invertida' checkbox.
- Pulsador (filtrado)**: Includes a pushbutton icon, a pin dropdown set to '0-RX', and a dropdown set to 'se ha pulsado'. It is connected to a 'Bucle' block.
- Pulsador táctil**: Includes a pushbutton icon, a pin dropdown set to '2', and a dropdown set to '13' with an 'OFF' button and a 'milisegundos' label.
- Detector de movimiento (PIR)**: Includes a PIR sensor icon and a pin dropdown set to '2'.
- DHT-11**: Includes a temperature sensor icon, a dropdown set to 'Temperatura °C', and a pin dropdown set to '2'.
- DHT-22**: Includes a temperature sensor icon, a dropdown set to 'Temperatura °C', and a pin dropdown set to '2'.

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 cardiacos, sensores de efecto HALL,...

La interfaz de conexión de un sensor con Arduino lo podemos clasificar en tres tipos: **DIGITAL**, **ANALÓGICO** o **DATOS**.

- **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.



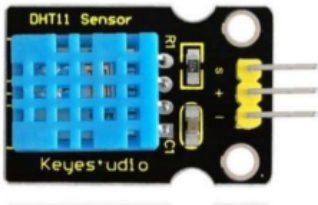
- **ANALÓGICO**: el sensor nos puede dar un rango de valores, normalmente se traduce en un valor de tensión o de corriente variable en función de la señal captada al sensor. En este caso conectaremos el sensor a una de las entradas analógicas de Arduino (A0,..., A5). El rango de entrada será una tensión 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 Arduino son de 10 bits es decir 1024 valores. De tal modo que  $0 = 0 \text{ v.}$  y  $1023 = 5 \text{ V.}$



- **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 que mide temperatura y humedad.



## Sensores modulares más comunes.

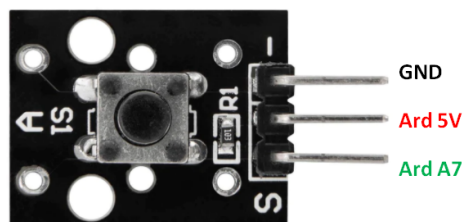
En la actualidad existen infinidad de sensores que los fabricantes presentan en forma modular. Esto hace que su conexión y utilización sea mucho más sencilla que la tradicional, olvidándonos de resistencias, polaridades, cableados,... para su correcto funcionamiento.

### Sensor pulsador

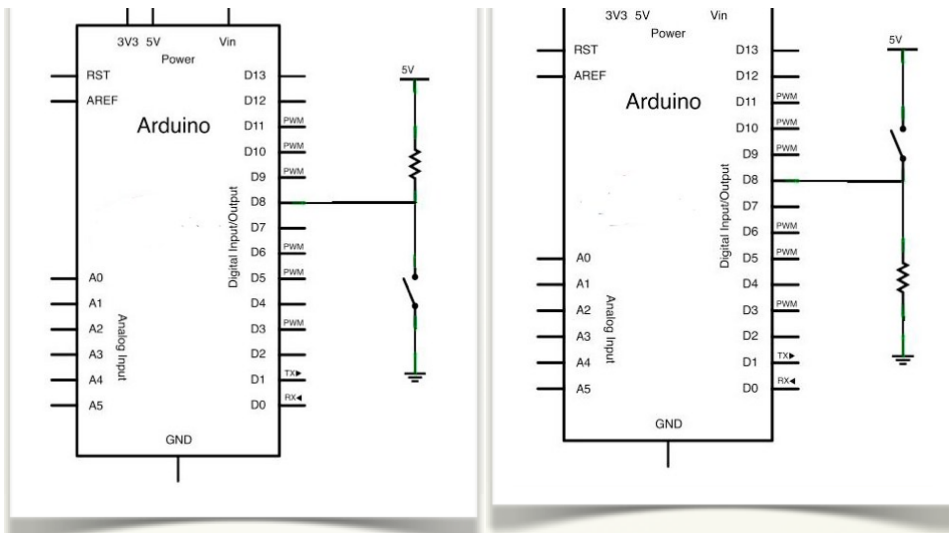
Es un sensor digital, que presenta dos estados; cuando se presiona el botón, emite una señal de bajo (0V), cuando suelta el botón, emite una señal de bajo alto (5V). [Datasheet](#)

Un ejemplo de uso

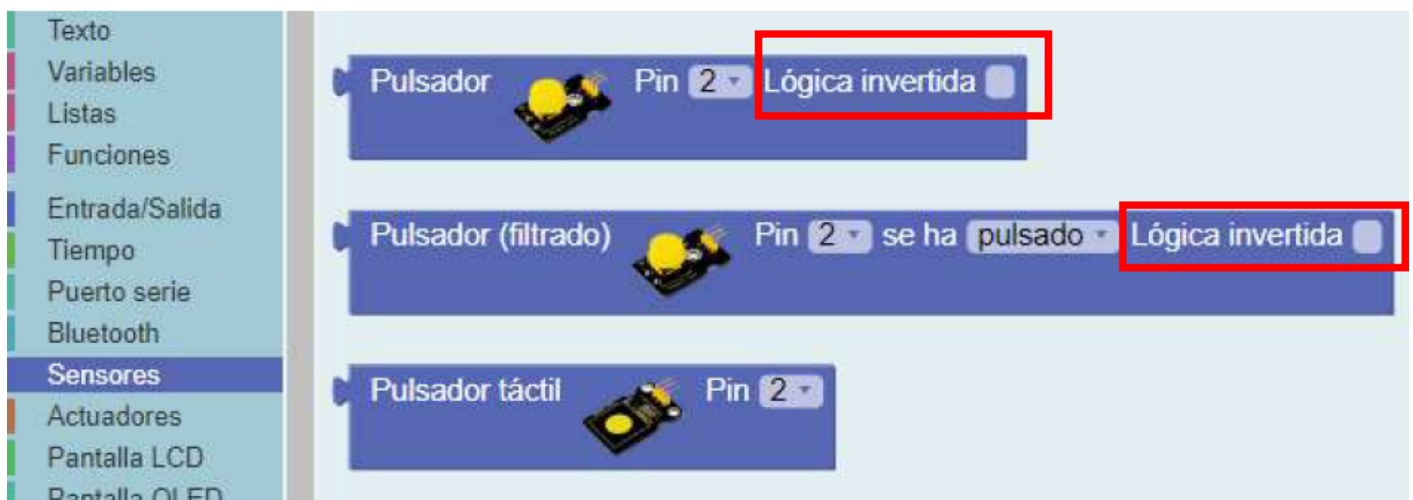
• en el robot mClen



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?



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



## 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.

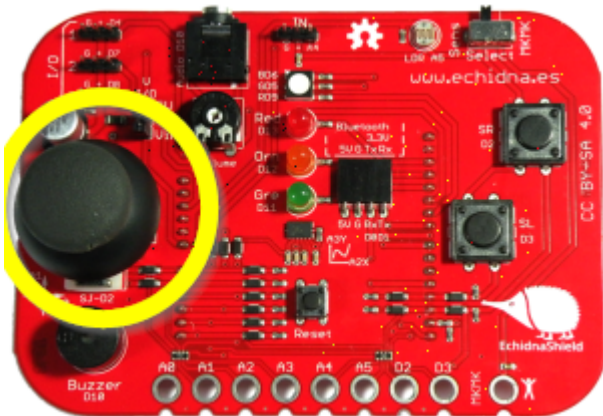
Un ejemplo de uso en

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



## 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.



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 Fotocélula 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 ejemplo 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



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



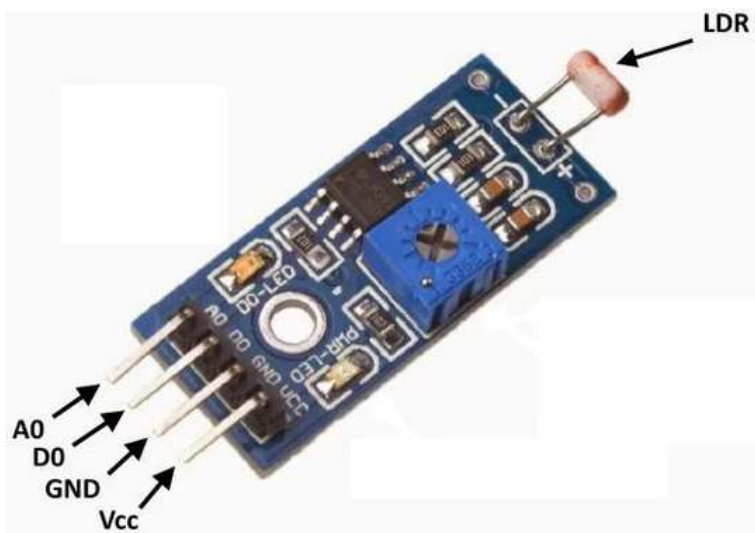
La instrucción con Arduinoblocks ya cuenta con esta configuración Pull downUp de que cuando **más** luz, **más** valor tiene la entrada analógica.



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**.



O como en la figura que ofrecen las dos cosas: salida analógica **A0** y digital **D0**.



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 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](#)

Radiación		Longitud de onda $\lambda$
Ultravioleta 100-400 nm	ultravioleta C	100 nm – 280 nm
	ultravioleta B	280 nm – 315 nm
	ultravioleta A	315 nm – 400 nm
Visible 400-780 nm	violeta	400 nm – 455 nm
	azul	455 nm – 490 nm
	verde	490 nm – 570 nm
	amarillo	570 nm – 590 nm
	anaranjado	590 nm – 620 nm
	rojo	620 nm – 780 nm
Infrarroja 780nm-1mm	infrarroja A	780 nm – 1400 nm
	infrarroja B	1400 nm – 3000 nm
	infrarroja C	3000 nm – 1 mm



## 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

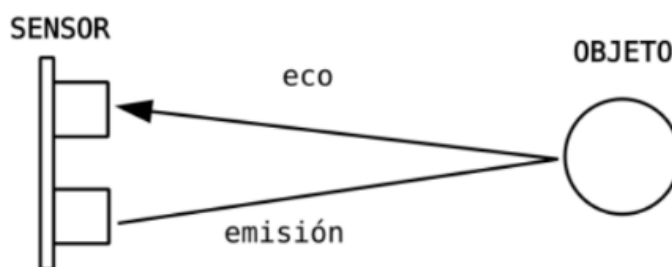
Ejemplos de uso:

- [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](#)

Pines de conexión:

- VCC
- Trig (Disparo del ultrasonido)
- Echo (Recepción del ultrasonido)
- GND

Distancia = {(Tiempo entre Trig y el Echo) \* (V.Sonido 340 m/s)}/2 por lo tanto en la programación os y calcular la distancia.



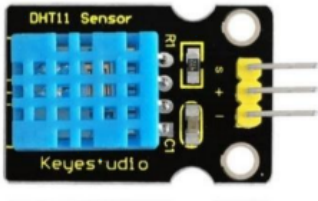
## Sensor DHT11 (Temperatura y Humedad).

Este sensor de temperatura y humedad **DHT11** tiene una salida de señal digital que funciona en un rango de temperaturas entre 0 y 50°C con un error de  $\pm 2^{\circ}\text{C}$  y un rango de humedad entre 20 y 90 %  $\pm 5\%$ . Tiene dentro un pequeño microprocesador que lanza por el bit de datos 40 bits en serie, los 16 primeros son la humedad y los 16 restantes es la temperatura los 8 restantes son de comprobación. 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

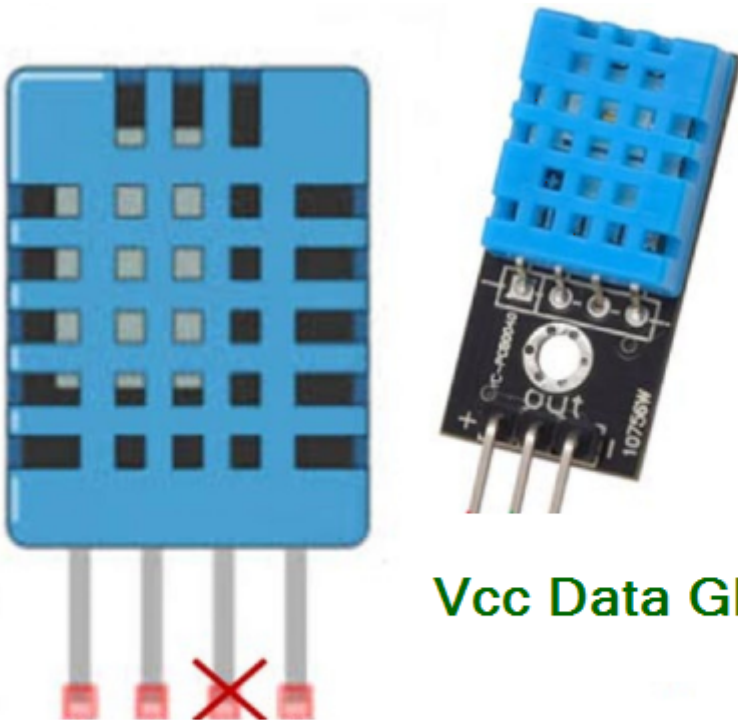
Ejemplos de uso:

- Medir H y T con Blink en Rover con Arduino
- Estación meteorológica Arduino con Arduinoblocks
- Arduinoblocks en el aula



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

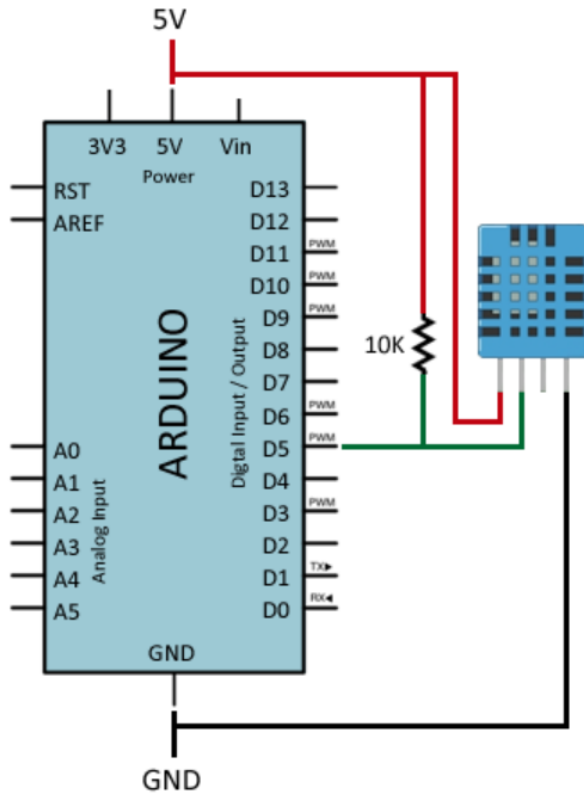
Tenemos dos opciones comerciales: **Encapsulado** que lo tienes preparado para conectar la alimentación y leer por el pin de datos, o **sin encapsular**, que hay que colocar una resistencia de aproximadamente 10k entre Vcc y Data



Vcc Data GND

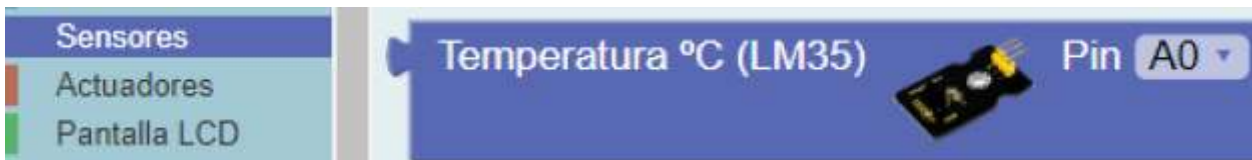
Vcc Data NC GND

Ejemplo de uso de un DHT11 sin encapsular:



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

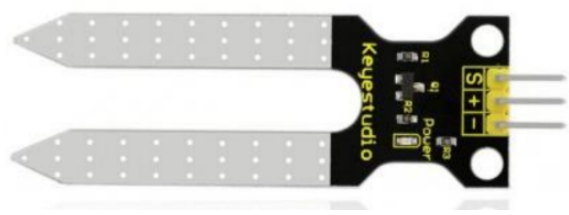


Es un sensor bastante mediocre, si necesitas una precisión el doble, te recomendamos el DHT22 que funciona exáctamente igual pero es de color blanco y más caro ~3€. Ver <https://www.luisllamas.es/arduino-dht11-dht22/>

## Sensor de humedad de suelo.

La función de este sensor es detectar la humedad del suelo. Si el suelo no tiene agua, el valor analógico emitido por el sensor disminuirá, de lo contrario, aumentará. Se puede utilizar este sensor para hacer un dispositivo de riego automático, puede detectar si las plantas “tienen sed” y evitar que se marchiten. El sensor se configura con dos sondas insertadas en el suelo, cuando la corriente atraviesa el suelo, el sensor obtendrá valor de resistencia al leer los cambios actuales entre las dos sondas y convertir dicho valor de resistencia en contenido de humedad. Cuanto mayor sea la humedad (menos resistencia), mayor será la conductividad del suelo. La corriente de

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



## Sensor de humedad.

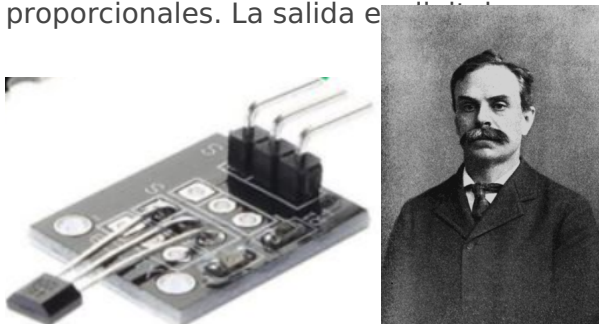
Este sensor analógico está diseñado para identificar y detectar la presencia de agua y su cantidad. Puede servir para detectar el nivel de agua, para disparar una alarma en caso de una fuga de agua, también para hacer un limpiapalabrisas automático.... puedes ver un ejemplo de uso en el curso de [Domótica con Arduino](#)

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



## 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.

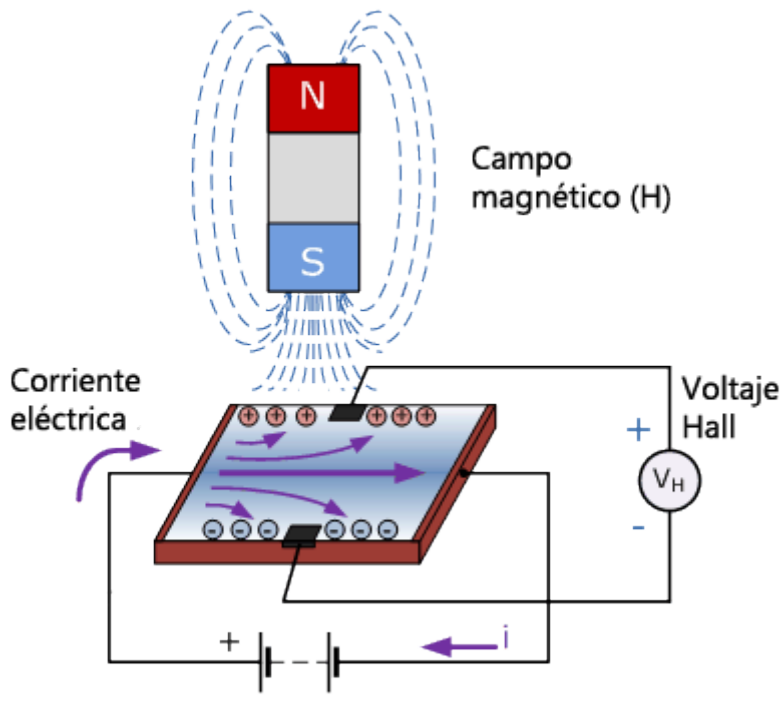


Sensor Hall.

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](#)

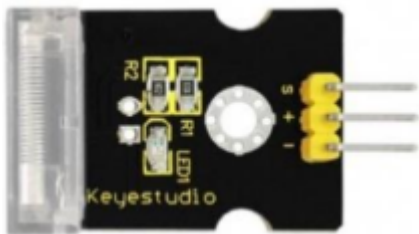
## Sensor inclinación

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.



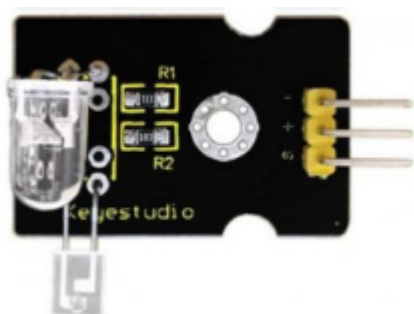
## Sensor de golpe

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



## 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.



## 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.



## Sensor de Gas (MQ2).

Este sensor analógico de gas MQ-2 se utiliza en equipos de detección de fugas de gas en electrónica de consumo y mercados industriales. Este sensor es adecuado para detectar GLP, I-butano, propano, metano, alcohol, hidrógeno y humo. Tiene alta sensibilidad y respuesta rápida. La sensibilidad se puede ajustar girando el potenciómetro.

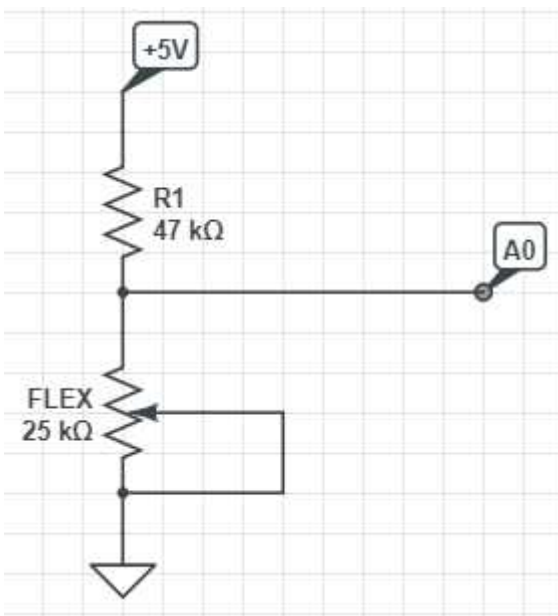


## Resistencia Flex

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

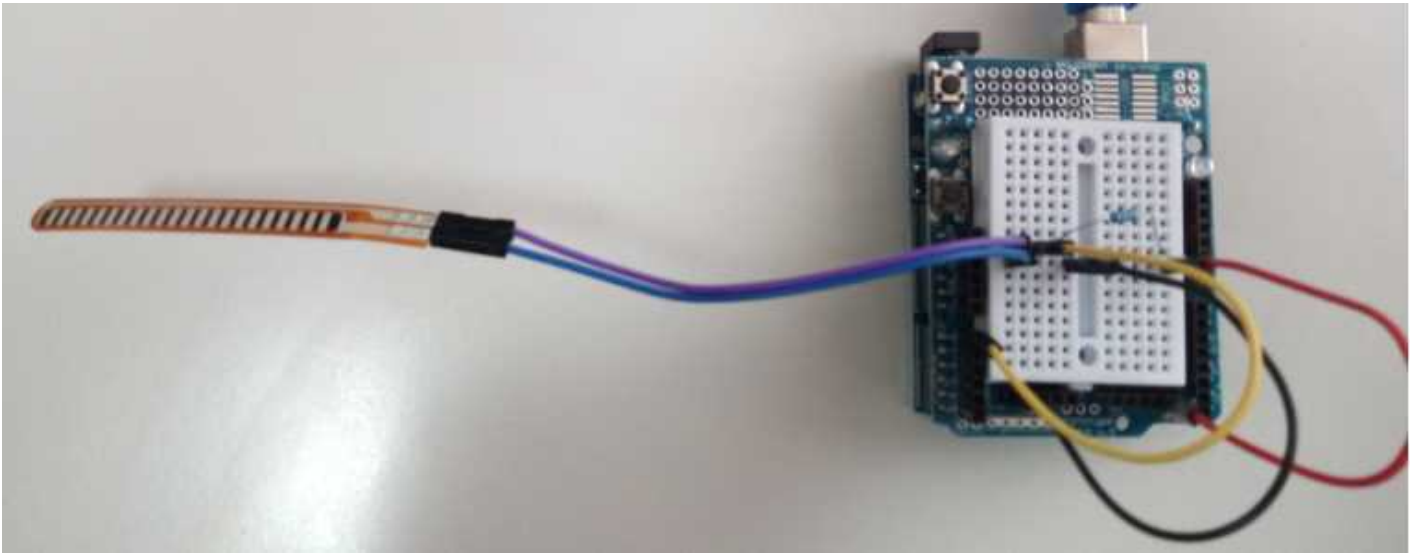


Para utilizar esta resistencia haremos un **DIVISOR DE TENSION** que consistirá en poner dos resistencias en serie y repartirá la tensión total entre 0V y 5V en las dos resistencias, **el punto medio** será un punto que tendrá una tensión variable en función de las dos resistencias, como la es variable,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

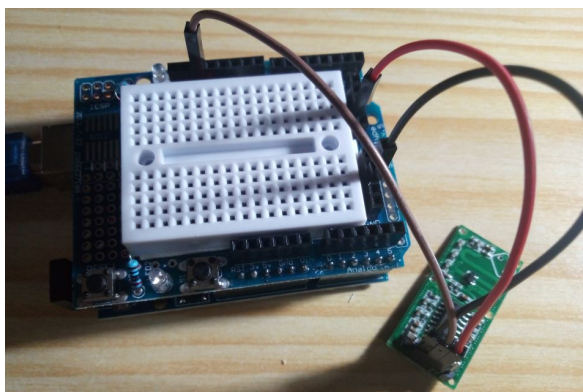
Tradicionalmente se utilizan **sensores PIR de infrarrojos** para detectar el movimiento, pero personalmente veo que tienen falsos positivos y negativos, si quieres utilizarlos, te recomendamos [esta página de Luis Llamas](#)



Personalmente prefiero 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.... Curiosamente, su gran ventaja técnica **es un gran inconveniente para usarlo en el aula**, con cualquier movimiento se dispara. Para saber más [ver la página de Luis Llamas](#)



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



Esta sección es una visión rápida de las posibles entradas del Arduino y está adaptado de [este enlace](#). José Andrés Echevarría @cantabRobots CC-BY-NC-SA

Los contenidos de esta página están extraídos del curso [Programa Arduino mediante Código](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



# Actuadores

Los contenidos de esta página están extraídos del curso [Programa Arduino mediante Código](#) de Aularagón.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

